

OMAP™

*Public Version*

# OMAP4430 Multimedia Device Silicon Revision 2.x

Texas Instruments OMAP™ Family of Products

Version K

## Technical Reference Manual



Literature Number: SWPU231K  
July 2010–Revised September 2010

## **WARNING: EXPORT NOTICE**

Recipient agrees to not knowingly export or re-export, directly or indirectly, any product or technical data (as defined by the U.S., EU, and other Export Administration Regulations) including software, or any controlled product restricted by other applicable national regulations, received from Disclosing party under this Agreement, or any direct product of such technology, to any destination to which such export or re-export is restricted or prohibited by U.S. or other applicable laws, without obtaining prior authorisation from U.S. Department of Commerce and other competent Government authorities to the extent required by those laws. This provision shall survive termination or expiration of this Agreement.

According to our best knowledge of the state and end-use of this product or technology, and in compliance with the export control regulations of dual-use goods in force in the origin and exporting countries, this technology is classified as follows:

US ECCN: 3E991

EU ECCN: EAR99

And may require export or re-export license for shipping it in compliance with the applicable regulations of certain countries.



# Contents

<b>Preface</b> .....	<b>245</b>
<b>1 Introduction</b> .....	<b>255</b>
1.1 Overview .....	256
1.2 Environment .....	257
1.3 Description .....	258
1.3.1 Cortex-A9 MPU Subsystem Description .....	260
1.3.2 IVA-HD Subsystem Description .....	260
1.3.3 Cortex-M3 MPU Subsystem Description .....	261
1.3.4 Display Subsystem Description .....	261
1.3.5 ABE Subsystem Description .....	262
1.3.6 ISS Description .....	263
1.3.7 2D/3D Graphics Accelerator Description .....	264
1.3.8 Face Detect Module Description .....	264
1.3.9 On-Chip Debug Support Description .....	264
1.3.10 Power, Reset, and Clock Management Description .....	265
1.3.11 On-Chip Memory Description .....	266
1.3.12 Memory Management Description .....	266
1.3.13 External Memory Interface Description .....	266
1.3.14 System and Connection Peripherals .....	266
1.3.14.1 System Peripherals .....	266
1.3.14.2 Connection Peripherals .....	267
1.4 Package-On-Package Concept .....	268
1.5 Device Identification .....	269
<b>2 Memory Mapping</b> .....	<b>271</b>
2.1 Introduction .....	272
2.2 L3 Memory Space Mapping .....	274
2.2.1 L3_EMU Memory Space Mapping .....	276
2.3 L4 Memory Space Mapping .....	277
2.3.1 L4_CFG Memory Space Mapping .....	277
2.3.2 L4_WKUP Memory Space Mapping .....	279
2.3.3 L4_PER Memory Space Mapping .....	280
2.3.4 L4_ABE Memory Space Mapping .....	282
2.4 Dual Cortex-M3 Subsystem Memory Space Mapping .....	286
2.5 DSP Subsystem Memory Space Mapping .....	287
2.6 Display Subsystem Memory Space Mapping .....	289
2.6.1 L3 Interconnect View of the Display Memory Space .....	289
2.6.2 L4 Interconnect View of the Display Memory Space .....	289
<b>3 Power, Reset and Clock Management</b> .....	<b>291</b>
3.1 Device Power Management Introduction .....	292
3.1.1 Device Power-Management Architecture Building Blocks .....	292
3.1.1.1 Clock Management .....	292
3.1.1.1.1 Module Interface and Functional Clocks .....	292
3.1.1.1.2 Module-Level Clock Management .....	293

3.1.1.1.3	Clock Domain .....	298
3.1.1.1.4	Clock Domain-Level Clock Management .....	299
3.1.1.1.5	Clock Domain HW_AUTO Mode Sequences .....	300
3.1.1.1.6	Clock Domain Sleep/Wakeup .....	303
3.1.1.1.7	Clock Domain Dependency .....	304
3.1.1.2	Power Management .....	307
3.1.1.2.1	Power Domain .....	307
3.1.1.2.2	Module Logic and Memory Context .....	308
3.1.1.2.3	Retention-Till-Access Memory Feature .....	309
3.1.1.2.4	Power Domain Management .....	309
3.1.1.3	Voltage Management .....	310
3.1.1.3.1	Voltage Domain .....	311
3.1.1.3.2	Voltage Domain Management .....	312
3.1.1.3.3	AVS Overview .....	313
3.1.2	Power-Management Techniques .....	315
3.1.2.1	Standby Leakage Management .....	315
3.1.2.2	Dynamic Voltage and Frequency Scaling .....	316
3.1.2.3	Dynamic Power Switching .....	316
3.1.2.4	Adaptive Voltage Scaling .....	317
3.1.2.5	Adaptive Body Bias .....	318
3.1.2.6	Combining Power-Management Techniques .....	318
3.1.2.6.1	DPS Versus SLM .....	319
3.2	PRCM Subsystem Overview .....	320
3.2.1	Introduction .....	320
3.2.2	Power-Management Framework Features .....	321
3.3	PRCM Subsystem Environment .....	322
3.3.1	External Clock Signals .....	322
3.3.2	External Reset Signals .....	323
3.3.3	External Power Control Signals .....	323
3.3.4	External Voltage Inputs .....	324
3.4	PRCM Subsystem Integration .....	325
3.4.1	Device Power-Management Layout .....	325
3.4.2	Power-Management Scheme, Reset, and Interrupt Requests .....	327
3.4.2.1	Power Domain .....	327
3.4.2.2	Resets .....	328
3.4.2.3	Interrupt Requests .....	328
3.5	Reset Management Functional Description .....	329
3.5.1	Overview .....	329
3.5.1.1	SCRM Reset Management Functional Description .....	329
3.5.1.1.1	Power-On Reset .....	329
3.5.1.1.2	Warm Reset .....	329
3.5.1.2	PRM Reset Management Functional Description .....	329
3.5.2	General Characteristics of Reset Signals .....	330
3.5.2.1	Scope .....	330
3.5.2.2	Occurrence .....	330
3.5.2.3	Source Type .....	330
3.5.2.4	Retention Type .....	331
3.5.3	Reset Sources .....	331
3.5.3.1	Global Reset Sources .....	331
3.5.3.2	Local Reset Sources .....	331
3.5.4	Reset Domains .....	332
3.5.5	Reset Logging .....	346
3.5.5.1	PRCM Module Reset Logging Mechanism .....	347

3.5.6	Reset Sequences .....	347
3.5.6.1	SCRM Power-on Reset Sequence .....	347
3.5.6.2	PRCM Module Power-On Reset Sequence .....	348
3.5.6.3	Cortex-A9 MPU Subsystem Power-On Reset Sequence .....	349
3.5.6.4	Cortex-A9 MPU Subsystem Warm Reset Sequence .....	350
3.5.6.5	Cortex-A9 MPU Subsystem Reset Sequence On Sleep and Wake-Up Transitions From RETENTION or Off STATE .....	351
3.5.6.6	IVAHD Subsystem Power-On Reset Sequence .....	352
3.5.6.7	IVAHD Subsystem Software Warm Reset Sequence .....	352
3.5.6.8	DSP Subsystem Power-On Reset Sequence .....	353
3.5.6.9	DSP Subsystem Software Warm Reset Sequence .....	354
3.5.6.10	DSP Subsystem Reset Sequence On Wake-Up Transitions From OSWR RETENTION State .....	355
3.5.6.11	Cortex-M3 MPU Subsystem Power-On Reset Sequence .....	356
3.5.6.12	Cortex-M3 MPU Subsystem Software Warm Reset Sequence .....	357
3.5.6.13	Global Warm Reset Sequence .....	359
3.6	Clock Management Functional Description .....	361
3.6.1	Overview .....	361
3.6.2	External Clock Inputs .....	363
3.6.2.1	sys_32k Clock Input .....	363
3.6.2.2	High-Frequency System Clock Input .....	363
3.6.3	Internal Clock Sources/Generators .....	363
3.6.3.1	PRM Clock Source .....	363
3.6.3.2	CM Clock Source .....	366
3.6.3.2.1	CM1 Clock Generator .....	366
3.6.3.2.2	CM1_USB Clock Generator .....	367
3.6.3.2.3	CM1_ABE Clock Generator .....	369
3.6.3.2.4	CM2 Clock Generator .....	370
3.6.3.3	Generic DPLL Overview .....	373
3.6.3.3.1	DPLLs Output Clocks Parameters .....	374
3.6.3.3.2	Enable Control, Status, and Low-Power Operation Mode .....	375
3.6.3.3.3	DPLL Power Modes .....	376
3.6.3.3.4	DPLL Recalibration .....	377
3.6.3.3.5	DPLL Spread Spectrum Clocking .....	378
3.6.3.3.6	DPLL Output Power Down .....	378
3.6.3.4	DPLL_PER Description .....	379
3.6.3.4.1	Overview .....	379
3.6.3.4.2	Synthesized Clock Parameters .....	379
3.6.3.4.3	Power Modes .....	380
3.6.3.4.4	Recalibration .....	381
3.6.3.4.5	Spread Spectrum Clocking .....	381
3.6.3.4.6	Output Power Down .....	381
3.6.3.5	DPLL_CORE Description .....	382
3.6.3.5.1	Overview .....	382
3.6.3.5.2	Synthesized Clock Parameters .....	382
3.6.3.5.3	Power Modes .....	383
3.6.3.5.4	Recalibration .....	384
3.6.3.5.5	Spread Spectrum Clocking .....	384
3.6.3.5.6	Output Power-Down .....	385
3.6.3.6	DPLL_ABE Description .....	385
3.6.3.6.1	Overview .....	385
3.6.3.6.2	Synthesized Clock Parameters .....	385
3.6.3.6.3	Power Modes .....	386
3.6.3.6.4	Recalibration .....	386

3.6.3.6.5	Spread Spectrum Clocking .....	387
3.6.3.7	DPLL_MPU Description .....	387
3.6.3.7.1	Overview .....	387
3.6.3.7.2	Synthesized Clock Parameters .....	387
3.6.3.7.3	Power Modes .....	388
3.6.3.7.4	Recalibration .....	388
3.6.3.7.5	Spread Spectrum Clocking .....	388
3.6.3.8	DPLL_IVA Description .....	389
3.6.3.8.1	Overview .....	389
3.6.3.8.2	Synthesized Clock Parameters .....	389
3.6.3.8.3	Power Modes .....	390
3.6.3.8.4	Recalibration .....	390
3.6.3.8.5	Spread Spectrum Clocking .....	390
3.6.3.8.6	Ouput Power Down .....	390
3.6.3.9	DPLL_USB Description .....	391
3.6.3.9.1	Overview .....	391
3.6.3.9.2	Synthesized Clock Parameters .....	391
3.6.3.9.3	Power Modes .....	391
3.6.3.9.4	Spread Spectrum Clocking .....	392
3.6.3.10	DPLLs Cascading .....	392
3.6.4	CD_WKUP Clock Domain .....	394
3.6.4.1	Overview .....	394
3.6.4.2	Clock Domain Modes .....	394
3.6.4.3	Clock Domain Dependency .....	395
3.6.4.3.1	Wake-Up Dependency .....	395
3.6.4.4	Clock Domain Module Attributes .....	395
3.6.5	CD_DSP Clock Domain .....	397
3.6.5.1	Overview .....	397
3.6.5.2	Clock Domain Modes .....	397
3.6.5.3	Clock Domain Dependency .....	398
3.6.5.3.1	Static Dependency .....	398
3.6.5.3.2	Dynamic Dependency .....	398
3.6.5.4	Clock Domain Module Attributes .....	398
3.6.6	CD_STD_EFUSE Clock Domain .....	399
3.6.6.1	Overview .....	399
3.6.6.2	Clock Domain Module Attributes .....	400
3.6.7	CD_CORTEXA9 Clock Domain .....	400
3.6.7.1	Overview .....	400
3.6.7.2	Clock Domain Modes .....	400
3.6.7.3	Clock Domain Dependency .....	401
3.6.7.3.1	Static Dependency .....	401
3.6.7.3.2	Dynamic Dependency .....	401
3.6.7.4	Clock Domain Module Attributes .....	402
3.6.8	CD_L4_PER Clock Domain .....	402
3.6.8.1	Overview .....	402
3.6.8.2	Clock Domain Modes .....	403
3.6.8.3	Clock Domain Dependency .....	404
3.6.8.3.1	Dynamic Dependency .....	404
3.6.8.3.2	Wake-Up Dependency .....	404
3.6.8.4	Clock Domain Module Attributes .....	409
3.6.9	CD_L3_INIT Clock Domain .....	414
3.6.9.1	Overview .....	414
3.6.9.2	Clock Domain Modes .....	415

3.6.9.3	Clock Domain Dependency .....	416
3.6.9.3.1	Static Dependency .....	416
3.6.9.3.2	Dynamic Dependency .....	416
3.6.9.3.3	Wake-Up Dependency .....	416
3.6.9.4	Clock Domain Module Attributes .....	418
3.6.10	CD_IVAHD Clock Domain .....	421
3.6.10.1	Overview .....	421
3.6.10.2	Clock Domain Modes .....	421
3.6.10.3	Clock Domain Dependency .....	421
3.6.10.3.1	Static Dependency .....	421
3.6.10.3.2	Dynamic Dependency .....	422
3.6.10.4	Clock Domain Module Attributes .....	422
3.6.11	CD_SGX Clock Domain .....	423
3.6.11.1	Overview .....	423
3.6.11.2	Clock Domain Modes .....	423
3.6.11.3	Clock Domain Dependency .....	423
3.6.11.3.1	Static Dependency .....	423
3.6.11.3.2	Dynamic Dependency .....	424
3.6.11.4	Clock Domain Module Attributes .....	424
3.6.12	CD_EMU Clock Domain .....	425
3.6.12.1	Overview .....	425
3.6.12.2	Clock Domain Modes .....	425
3.6.12.3	Clock Domain Dependency .....	426
3.6.12.3.1	Dynamic Dependency .....	426
3.6.12.4	Clock Domain Module Attributes .....	426
3.6.13	CD_DSS Clock Domain .....	426
3.6.13.1	Overview .....	426
3.6.13.2	Clock Domain Modes .....	427
3.6.13.3	Clock Domain Dependency .....	427
3.6.13.3.1	Static Dependency .....	427
3.6.13.3.2	Dynamic Dependency .....	428
3.6.13.3.3	Wake-Up Dependency .....	428
3.6.13.4	Clock Domain Module Attributes .....	429
3.6.14	CD_L4_CFG Clock Domain .....	430
3.6.14.1	Overview .....	430
3.6.14.2	Clock Domain Modes .....	430
3.6.14.3	Clock Domain Dependency .....	431
3.6.14.3.1	Dynamic Dependency .....	431
3.6.14.4	Clock Domain Module Attributes .....	431
3.6.15	CD_L3_INSTR Clock Domain .....	432
3.6.15.1	Overview .....	432
3.6.15.2	Clock Domain Modes .....	433
3.6.15.3	Clock Domain Dependency .....	433
3.6.15.4	Clock Domain Module Attributes .....	433
3.6.16	CD_L3_2 Clock Domain .....	434
3.6.16.1	Overview .....	434
3.6.16.2	Clock Domain Modes .....	435
3.6.16.3	Clock Domain Dependency .....	435
3.6.16.3.1	Dynamic Dependency .....	435
3.6.16.4	Clock Domain Module Attributes .....	436
3.6.17	CD_L3_1 Clock Domain .....	437
3.6.17.1	Overview .....	437
3.6.17.2	Clock Domain Modes .....	437

3.6.17.3	Clock Domain Dependency .....	437
3.6.17.3.1	Dynamic Dependency .....	437
3.6.17.4	Clock Domain Module Attributes .....	438
3.6.18	CD_EMIF Clock Domain .....	438
3.6.18.1	Overview .....	438
3.6.18.2	Clock Domain Modes .....	439
3.6.18.3	Clock Domain Dependency .....	439
3.6.18.4	Clock Domain Module Attributes .....	439
3.6.19	CD_CORTEXM3 Clock Domain .....	441
3.6.19.1	Overview .....	441
3.6.19.2	Clock Domain Modes .....	441
3.6.19.3	Clock Domain Dependency .....	441
3.6.19.3.1	Static Dependency .....	441
3.6.19.3.2	Dynamic Dependency .....	442
3.6.19.4	Clock Domain Module Attributes .....	442
3.6.20	CD_DMA Clock Domain .....	443
3.6.20.1	Overview .....	443
3.6.20.2	Clock Domain Modes .....	443
3.6.20.3	Clock Domain Dependency .....	444
3.6.20.4	Clock Domain Module Attributes .....	444
3.6.21	CD_C2C Clock Domain .....	444
3.6.21.1	Overview .....	444
3.6.21.2	Clock Domain Modes .....	445
3.6.21.3	Clock Domain Dependency .....	445
3.6.21.3.1	Static Dependency .....	445
3.6.21.3.2	Dynamic Dependency .....	446
3.6.21.4	Clock Domain Module Attributes .....	446
3.6.22	CD_CAM Clock Domain .....	447
3.6.22.1	Overview .....	447
3.6.22.2	Clock Domain Modes .....	447
3.6.22.3	Clock Domain Dependency .....	448
3.6.22.3.1	Static Dependency .....	448
3.6.22.3.2	Dynamic Dependency .....	448
3.6.22.4	Clock Domain Module Attributes .....	448
3.6.23	CD_ABE Clock Domain .....	449
3.6.23.1	Overview .....	449
3.6.23.2	Clock Domain Modes .....	449
3.6.23.3	Clock Domain Dependency .....	450
3.6.23.3.1	Wake-Up Dependency .....	450
3.6.23.4	Clock Domain Module Attributes .....	452
3.6.24	CD_L4_ALWON_CORE Clock Domain .....	455
3.6.24.1	Overview .....	455
3.6.24.2	Clock Domain Modes .....	456
3.6.24.3	Clock Domain Dependency .....	456
3.6.24.3.1	Wake-Up Dependency .....	456
3.6.24.4	Clock Domain Module Attributes .....	456
3.7	Power Management Functional Description .....	457
3.7.1	PD_WKUP Description .....	457
3.7.1.1	Power Domain Modes .....	458
3.7.2	PD_DSP Description .....	458
3.7.2.1	Power Domain Modes .....	458
3.7.2.1.1	Logic and Memory Area Power Modes .....	458
3.7.2.1.2	Logic and Memory Area Power Modes Control and Status .....	459



3.7.3	PD_STD_EFUSE Description .....	459
3.7.3.1	Power Domain Modes .....	460
3.7.4	PD_MPU Description .....	460
3.7.4.1	Power Domain Modes .....	460
3.7.4.1.1	Logic and Memory Area Power Modes .....	460
3.7.4.1.2	Logic and Memory Area Power Modes Control and Status .....	461
3.7.5	PD_L4_PER Description .....	461
3.7.5.1	Power Domain Modes .....	463
3.7.5.1.1	Logic and Memory Area Power Modes .....	463
3.7.5.1.2	Logic and Memory Area Power Modes Control and Status .....	463
3.7.6	PD_L3_INIT Description .....	464
3.7.6.1	Power Domain Modes .....	465
3.7.6.1.1	Logic and Memory Area Power Modes .....	465
3.7.6.1.2	Logic and Memory Area Power Modes Control and Status .....	465
3.7.7	PD_IVAHD Description .....	466
3.7.7.1	Power Domain Modes .....	466
3.7.7.1.1	Logic and Memory Area Power Modes .....	466
3.7.7.1.2	Logic and Memory Area Power Modes Control and Status .....	467
3.7.8	PD_SGX Description .....	468
3.7.8.1	Power Domain Modes .....	468
3.7.8.1.1	Logic and Memory Area Power Modes .....	468
3.7.8.1.2	Logic and Memory Area Power Modes Control and Status .....	468
3.7.9	PD_EMU Description .....	469
3.7.9.1	Power Domain Modes .....	469
3.7.9.1.1	Logic and Memory Area Power Modes .....	469
3.7.9.1.2	Logic and Memory Area Power Modes Control and Status .....	470
3.7.10	PD_DSS Description .....	470
3.7.10.1	Power Domain Modes .....	470
3.7.10.1.1	Logic and Memory Area Power Modes .....	470
3.7.10.1.2	Logic and Memory Area Power Modes Control and Status .....	471
3.7.11	PD_CORE Description .....	471
3.7.11.1	Power Domain Modes .....	473
3.7.11.1.1	Logic and Memory Area Power Modes .....	473
3.7.11.1.2	Logic and Memory Area Power Modes Control and Status .....	473
3.7.12	PD_CAM Description .....	475
3.7.12.1	Power Domain Modes .....	475
3.7.12.1.1	Logic and Memory Area Power Modes .....	475
3.7.12.1.2	Logic and Memory Area Power Modes Control and Status .....	475
3.7.13	PD_AUDIO Description .....	476
3.7.13.1	Power Domain Modes .....	477
3.7.13.1.1	Logic and Memory Area Power Modes .....	477
3.7.13.1.2	Logic and Memory Area Power Modes Control and Status .....	477
3.7.14	PD_ALWON_MPU Description .....	478
3.7.14.1	Power Domain Modes .....	478
3.7.15	PD_ALWON_DSP Description .....	478
3.7.15.1	Power Domain Modes .....	478
3.7.16	PD_ALWON_CORE Description .....	479
3.7.16.1	Power Domain Modes .....	479
3.8	Voltage Management Functional Description .....	480
3.8.1	Overview .....	480
3.8.2	Voltage-Control Architecture .....	480
3.8.3	VDD_MPU_L, VDD_CORE_L, and VDD_IVA_L Control .....	481
3.8.3.1	PRM Dedicated I <sup>2</sup> C Control .....	481

3.8.3.2	Adaptive Voltage Scaling .....	482
3.8.3.2.1	SmartReflex in the Device .....	482
3.8.3.2.2	SmartReflex Module .....	483
3.8.3.2.3	SmartReflex Submodules .....	484
3.8.3.2.4	Status Register .....	487
3.8.3.2.5	SmartReflex Parameters Set After Silicon Characterization .....	487
3.8.3.2.6	Voltage Processor Module .....	487
3.8.3.2.7	SMPS-Dependent Parameter Configuration .....	490
3.8.3.2.8	Communication Between SmartReflex, Voltage Processor, Voltage Controller, and SMPS .....	490
3.8.4	Internal LDOs Control .....	491
3.8.4.1	Memory LDOs .....	491
3.8.4.2	LDO_WAKEUP .....	491
3.8.4.3	ABB LDOs Control .....	492
3.8.4.4	Bandgap control .....	492
3.8.5	Voltage Domain State Transitions .....	492
3.8.5.1	VDD_x_L Transitions .....	493
3.8.5.2	Memory LDOs Transitions .....	493
3.8.5.3	LDO_WAKEUP Transitions .....	494
3.8.6	DVFS .....	494
3.8.6.1	EMIF Clocks Frequency Scaling Constraints .....	494
3.8.6.2	GPMC Clocks Frequency Scaling Constraints .....	496
3.8.6.3	CORE DVFS Versus Subsystem Functionality .....	497
3.8.6.3.1	Display Subsystem .....	497
3.8.6.3.2	IVAHD .....	497
3.8.6.3.3	Imaging Subsystem .....	497
3.8.6.4	DVFS Voting Mechanism (VDD_CORE_L and VDD_IVA_L Voltage Domains) .....	497
3.8.6.4.1	CORE OPP Voting Architecture .....	497
3.8.6.4.2	IVA OPP Voting Architecture .....	498
3.8.6.4.3	Voting Mechanism .....	498
3.9	Device Low-Power States .....	498
3.9.1	Device Wake-Up Source Summary .....	499
3.9.2	Device RETENTION State Management .....	499
3.9.3	Device OFF State Management .....	499
3.9.3.1	Device Off Mode Sleep Sequence .....	500
3.9.3.2	Device Off Mode Wake-Up Sequences .....	502
3.9.3.2.1	Wake Up From a Wake-Up Event .....	502
3.9.3.2.2	Wakeup Upon Global Warm Reset .....	504
3.9.3.3	Global Warm Reset During a Device Wake-Up Sequence .....	504
3.9.4	I/O Management .....	504
3.9.4.1	Hardware-Controlled I/O Isolation Sequences .....	506
3.9.4.2	Software-Controlled I/O Isolation .....	506
3.10	PRCM Module Programming Guide .....	507
3.10.1	DPLLs Low-Level Programming Models .....	507
3.10.1.1	Global Initialization .....	507
3.10.1.1.1	Surrounding Module Global Initialization .....	507
3.10.1.1.2	DPLL Global Initialization .....	507
3.10.1.2	DPLL Output Frequency Change .....	509
3.10.2	Clock Management Low-Level Programming Models .....	511
3.10.2.1	Global Initialization .....	511
3.10.2.1.1	Surrounding Module Global Initialization .....	511
3.10.2.1.2	Clock Management Global Initialization .....	511
3.10.2.2	Clock Domain Sleep Transition and Troubleshooting .....	512
3.10.2.3	Enable/Disable Software-Programmable Static Dependency .....	512

3.10.3	Power Management Low-Level Programming Models .....	513
3.10.3.1	Global Initialization .....	513
3.10.3.1.1	Surrounding Module Global Initialization .....	513
3.10.3.1.2	Power Management Global Initialization .....	513
3.10.3.2	Forced Memory Area State Change With Power Domain ON .....	513
3.10.3.3	Forced Power Domain Low-Power State Transition .....	513
3.10.4	Voltage Management Low-Level Programming Models .....	514
3.10.4.1	Global Initialization .....	514
3.10.4.1.1	Surrounding Module Global Initialization .....	514
3.10.4.1.2	Voltage Management Global Initialization .....	514
3.10.4.1.3	SmartReflex Module Initialization .....	514
3.10.4.1.4	Voltage Processor Initialization .....	516
3.10.4.1.5	Voltage Controller Initialization .....	517
3.10.4.2	Changing OPP .....	518
3.11	PRCM Register Manual .....	519
3.11.1	PRM Instance Summary .....	519
3.11.2	INTRCONN_SOCKET_PRM Registers .....	519
3.11.2.1	INTRCONN_SOCKET_PRM Register Summary .....	519
3.11.2.2	INTRCONN_SOCKET_PRM Register Description .....	519
3.11.3	CKGEN_PRM Registers .....	537
3.11.3.1	CKGEN_PRM Register Summary .....	537
3.11.3.2	CKGEN_PRM Register Description .....	537
3.11.4	MPU_PRM Registers .....	539
3.11.4.1	MPU_PRM Register Summary .....	539
3.11.4.2	MPU_PRM Register Description .....	539
3.11.5	DSP_PRM Registers .....	543
3.11.5.1	DSP_PRM Register Summary .....	543
3.11.5.2	DSP_PRM Register Description .....	544
3.11.6	ABE_PRM Registers .....	549
3.11.6.1	ABE_PRM Register Summary .....	549
3.11.6.2	ABE_PRM Register Description .....	549
3.11.7	ALWAYS_ON_PRM Registers .....	571
3.11.7.1	ALWAYS_ON_PRM Register Summary .....	571
3.11.7.2	ALWAYS_ON_PRM Register Description .....	571
3.11.8	CORE_PRM Registers .....	575
3.11.8.1	CORE_PRM Register Summary .....	575
3.11.8.2	CORE_PRM Register Description .....	576
3.11.9	IVAHD_PRM Registers .....	595
3.11.9.1	IVAHD_PRM Register Summary .....	595
3.11.9.2	IVAHD_PRM Register Description .....	596
3.11.10	CAM_PRM Registers .....	602
3.11.10.1	CAM_PRM Register Summary .....	602
3.11.10.2	CAM_PRM Register Description .....	602
3.11.11	DSS_PRM Registers .....	605
3.11.11.1	DSS_PRM Register Summary .....	605
3.11.11.2	DSS_PRM Register Description .....	606
3.11.12	SGX_PRM Registers .....	610
3.11.12.1	SGX_PRM Register Summary .....	610
3.11.12.2	SGX_PRM Register Description .....	611
3.11.13	L3INIT_PRM Registers .....	613
3.11.13.1	L3INIT_PRM Register Summary .....	613
3.11.13.2	L3INIT_PRM Register Description .....	614
3.11.14	L4PER_PRM Registers .....	628

3.11.14.1	L4PER_PRM Register Summary .....	628
3.11.14.2	L4PER_PRM Register Description .....	630
3.11.15	WKUP_PRM Registers .....	677
3.11.15.1	WKUP_PRM Register Summary .....	677
3.11.15.2	WKUP_PRM Register Description .....	677
3.11.16	WKUP_CM Registers .....	684
3.11.16.1	WKUP_CM Register Summary .....	684
3.11.16.2	WKUP_CM Register Description .....	685
3.11.17	EMU_PRM Registers .....	692
3.11.17.1	EMU_PRM Register Summary .....	692
3.11.17.2	EMU_PRM Register Description .....	693
3.11.18	EMU_CM Registers .....	695
3.11.18.1	EMU_CM Register Summary .....	695
3.11.18.2	EMU_CM Register Description .....	695
3.11.19	DEVICE_PRM Registers .....	698
3.11.19.1	DEVICE_PRM Register Summary .....	698
3.11.19.2	DEVICE_PRM Register Description .....	700
3.11.20	INSTR_PRM Registers .....	747
3.11.20.1	INSTR_PRM Register Summary .....	747
3.11.20.2	INSTR_PRM Register Description .....	747
3.11.21	CM1 Instance Summary .....	750
3.11.22	INTRCONN_SOCKET_CM1 Registers .....	751
3.11.22.1	INTRCONN_SOCKET_CM1 Register Summary .....	751
3.11.22.2	INTRCONN_SOCKET_CM1 Register Description .....	751
3.11.23	CKGEN_CM1 Registers .....	753
3.11.23.1	CKGEN_CM1 Register Summary .....	753
3.11.23.2	CKGEN_CM1 Register Description .....	754
3.11.24	MPU_CM1 Registers .....	791
3.11.24.1	MPU_CM1 Register Summary .....	791
3.11.24.2	MPU_CM1 Register Description .....	791
3.11.25	DSP_CM1 Registers .....	796
3.11.25.1	DSP_CM1 Register Summary .....	796
3.11.25.2	DSP_CM1 Register Description .....	796
3.11.26	ABE_CM1 Registers .....	800
3.11.26.1	ABE_CM1 Register Summary .....	800
3.11.26.2	ABE_CM1 Register Description .....	800
3.11.27	RESTORE_CM1 Registers .....	817
3.11.27.1	RESTORE_CM1 Register Summary .....	817
3.11.27.2	RESTORE_CM1 Register Description .....	818
3.11.28	INSTR_CM1 Registers .....	834
3.11.28.1	INSTR_CM1 Register Summary .....	834
3.11.28.2	INSTR_PRM Register Description .....	834
3.11.29	CM2 Instance Summary .....	838
3.11.30	INTRCONN_SOCKET_CM2 Registers .....	838
3.11.30.1	INTRCONN_SOCKET_CM2 Register Summary .....	838
3.11.30.2	INTRCONN_SOCKET_CM2 Register Description .....	838
3.11.31	CKGEN_CM2 Registers .....	840
3.11.31.1	CKGEN_CM2 Register Summary .....	840
3.11.31.2	CKGEN_CM2 Register Description .....	841
3.11.32	ALWAYS_ON_CM2 Registers .....	862
3.11.32.1	ALWAYS_ON_CM2 Register Summary .....	862
3.11.32.2	ALWAYS_ON_CM2 Register Description .....	863
3.11.33	CORE_CM2 Registers .....	867

3.11.33.1	CORE_CM2 Register Summary .....	867
3.11.33.2	CORE_CM2 Register Description .....	868
3.11.34	IVAHD_CM2 Registers .....	900
3.11.34.1	IVAHD_CM2 Register Summary .....	900
3.11.34.2	IVAHD_CM2 Register Description .....	901
3.11.35	CAM_CM2 Registers .....	905
3.11.35.1	CAM_CM2 Register Summary .....	905
3.11.35.2	CAM_CM2 Register Description .....	905
3.11.36	DSS_CM2 Registers .....	910
3.11.36.1	DSS_CM2 Register Summary .....	910
3.11.36.2	DSS_CM2 Register Description .....	910
3.11.37	SGX_CM2 Registers .....	914
3.11.37.1	SGX_CM2 Register Summary .....	914
3.11.37.2	SGX_CM2 Register Description .....	914
3.11.38	L3INIT_CM2 Registers .....	918
3.11.38.1	L3INIT_CM2 Register Summary .....	918
3.11.38.2	L3INIT_CM2 Register Description .....	918
3.11.39	L4PER_CM2 Registers .....	933
3.11.39.1	L4PER_CM2 Register Summary .....	933
3.11.39.2	L4PER_CM2 Register Description .....	934
3.11.40	RESTORE_CM2 Registers .....	966
3.11.40.1	RESTORE_CM2 Register Summary .....	966
3.11.40.2	RESTORE_CM2 Register Description .....	966
3.12	SCRM Register Manual .....	994
3.12.1	SCRM Instance Summary .....	994
3.12.2	SCRM Registers .....	994
3.12.2.1	SCRM Register Summary .....	994
3.12.2.2	SCRM Register Description .....	995
3.13	SR Register Manual .....	1018
3.13.1	SR Instance Summary .....	1018
3.13.2	SR Registers .....	1018
3.13.2.1	SR Register Summary .....	1018
3.13.2.2	SR Register Description .....	1018
<b>4</b>	<b>Dual Cortex-A9 MPU Subsystem .....</b>	<b>1029</b>
4.1	Dual Cortex-A9 MPU Subsystem Overview .....	1030
4.1.1	Introduction .....	1030
4.1.2	Features .....	1032
4.2	Dual Cortex-A9 MPU Subsystem Integration .....	1033
4.2.1	Clock Distribution .....	1034
4.2.2	Reset Distribution .....	1035
4.3	Dual Cortex-A9 MPU Subsystem Functional Description .....	1038
4.3.1	Cortex-A9 MPU Subsystem Block Diagram .....	1038
4.3.2	ARM Core .....	1039
4.3.3	Local Interconnect .....	1040
4.3.4	Power Management .....	1040
4.3.4.1	Power Domains .....	1040
4.3.4.2	Power States of CPU0 and CPU1 .....	1041
4.3.4.3	WUGEN .....	1043
4.3.4.4	Power Transition Sequence .....	1044
4.3.4.5	CPU0 and CPU1 Power Domains Description .....	1044
4.3.4.5.1	Power Domain Modes .....	1045
4.4	Dual Cortex-A9 MPU Subsystem Register Manual .....	1047
4.4.1	Cortex-A9 MPU Subsystem Instance Summary .....	1047

4.4.2	SCU Registers .....	1047
4.4.3	Interrupt Controller Registers .....	1047
4.4.4	Timer Registers .....	1047
4.4.5	PL310 Registers .....	1047
4.4.6	Local PRCM Revision Register .....	1047
4.4.6.1	Local PRCM Revision Register Summary .....	1047
4.4.6.2	Local PRCM Revision Register Description .....	1047
4.4.7	Local PRCM Registers .....	1048
4.4.7.1	Local PRCM Registers Summary .....	1048
4.4.7.2	Local PRCM Log Register Description .....	1048
4.4.8	Local PRCM CPU0 and CPU1 Registers .....	1049
4.4.8.1	Local PRCM CPU0 and CPU1 Register Summary .....	1049
4.4.8.2	Local PRCM CPU0 Register Description .....	1050
4.4.9	Wake-Up Generator Registers .....	1054
4.4.9.1	CORTEXA9_WUGEN Register Summary .....	1054
4.4.9.2	CORTEXA9_WUGEN Register Description .....	1055
<b>5</b>	<b>DSP Subsystem .....</b>	<b>1069</b>
<b>6</b>	<b>IVA-HD Subsystem .....</b>	<b>1071</b>
6.1	IVA-HD Overview .....	1072
6.1.1	IVA-HD Integration .....	1074
6.1.2	IVA-HD Functional Description .....	1075
6.1.2.1	IVA-HD Block Diagram .....	1075
6.1.2.1.1	SyncBox .....	1076
6.1.2.1.2	ICONTs .....	1076
6.1.2.1.3	vDMA .....	1076
6.1.2.1.4	iME3 .....	1076
6.1.2.1.5	iPE3 .....	1077
6.1.2.1.6	MC3 .....	1077
6.1.2.1.7	CALC3 .....	1077
6.1.2.1.8	iLF3 .....	1077
6.1.2.1.9	ECD3 .....	1077
6.1.2.1.10	SL2 Interface .....	1077
6.1.2.1.11	Message Bus .....	1078
6.1.2.1.12	IVA-HD Local Interconnect .....	1078
6.1.2.1.13	MailBox .....	1078
6.1.2.1.14	IVA-HD System Control .....	1078
6.1.2.2	IVA-HD Power Management .....	1078
6.1.2.3	IVA-HD Memory Mapping .....	1078
6.1.3	IVA-HD Register Manual .....	1082
6.1.3.1	IVA-HD Instance Summary .....	1083
6.1.3.2	SYSCTRL Registers .....	1083
6.1.3.2.1	SYSCTRL Register Summary .....	1083
6.1.3.2.2	SYSCTRL Register Description .....	1083
<b>7</b>	<b>Dual Cortex-M3 MPU Subsystem .....</b>	<b>1095</b>
7.1	Dual Cortex-M3 MPU Subsystem Overview .....	1096
7.1.1	Introduction .....	1096
7.1.2	Features .....	1097
7.2	Dual Cortex-M3 MPU Subsystem Integration .....	1098
7.2.1	Dual Cortex-M3 MPU Subsystem Clock and Reset Distribution .....	1099
7.2.1.1	Clock Distribution .....	1099
7.2.1.2	Reset Distribution .....	1100
7.3	Dual Cortex-M3 MPU Subsystem Functional Description .....	1102
7.3.1	Dual Cortex-M3 MPU Subsystem Block Diagram .....	1102

7.3.2	Power Management .....	1103
7.3.2.1	Local Power Management .....	1104
7.3.2.2	Power Domains .....	1104
7.3.2.3	Voltage Domain .....	1104
7.3.2.4	Power States and Modes .....	1105
7.3.2.5	Wake-Up Generator .....	1106
7.3.2.5.1	WUGEN Main Features .....	1106
7.3.3	Shared Cache .....	1107
7.3.4	Shared Cache MMU .....	1108
7.3.5	L2 MMU .....	1108
7.3.5.1	L2 MMU Behavior on Page-Fault in Dual Cortex-M3 MPU .....	1109
7.3.6	Interprocessor Communication (IPC) .....	1109
7.3.6.1	Use of WFE and SEV .....	1109
7.3.6.2	Use of Interrupt for IPC .....	1110
7.3.6.3	Use of the Bit-Band Feature for Semaphore Operations .....	1110
7.3.6.4	Private Memory Space .....	1110
7.4	Dual Cortex-M3 MPU Subsystem Register Manual .....	1112
7.4.1	Dual Cortex-M3 Subsystem Instance Summary .....	1112
7.4.2	Shared Cache Configuration Registers .....	1112
7.4.3	Shared Cache SCTM Registers .....	1112
7.4.4	Shared Cache MMU Registers .....	1112
7.4.5	Cortex-M3 L2 MMU Registers .....	1112
7.4.6	Cortex-M3 NVIC Registers .....	1113
7.4.7	Cortex-M3 Wake-Up Generator Registers .....	1113
7.4.7.1	CORTEXM3_WKUP Register Summary .....	1113
7.4.7.2	Dual Cortex-M3 MPU Wake-Up Generator Register Description .....	1113
7.4.8	Cortex-M3 RW Table Registers .....	1117
7.4.8.1	CM3_RW_Table Register Summary .....	1117
7.4.8.2	Cortex-M3 RW Table Register Description .....	1117
<b>8</b>	<b>Imaging Subsystem .....</b>	<b>1119</b>
8.1	ISS Overview .....	1120
8.1.1	ISS Integration .....	1123
8.1.1.1	ISS PRCM Interface Integration .....	1125
8.1.1.1.1	ISS Clock Domains .....	1125
8.1.2	ISS Functional Description .....	1126
8.1.2.1	ISS Interrupts .....	1126
8.1.2.1.1	ISS Interrupt Merger .....	1126
8.1.2.1.2	ISS Submodule Interrupts .....	1127
8.1.2.2	ISS Clocks .....	1136
8.1.2.3	ISS Reset .....	1137
8.1.2.4	ISS Power Management .....	1138
8.1.2.4.1	ISS Power-Management Infrastructure Overview .....	1138
8.1.2.4.2	ISS STANDBY Mechanism .....	1138
8.1.2.4.3	ISS IDLE Mechanism .....	1140
8.1.3	ISS Register Manual .....	1141
8.1.3.1	ISS Instance Summary .....	1141
8.1.3.2	ISS Registers .....	1141
8.1.3.2.1	ISS TOP Register Summary .....	1141
8.1.3.2.2	ISS TOP Register Description .....	1141
8.2	ISS Interfaces .....	1156
8.2.1	ISS Interfaces Overview .....	1156
8.2.1.1	ISS Interface Features .....	1157
8.2.2	ISS Interfaces Environment .....	1160

8.2.2.1	ISS Interfaces Signal Descriptions .....	1160
8.2.2.2	ISS Interface Modes .....	1161
8.2.3	ISS CSI2 PHY .....	1164
8.2.3.1	ISS CSI2 PHY Overview .....	1164
8.2.3.2	ISS CSI2 PHY Functional Description .....	1165
8.2.3.2.1	ISS CSI2 PHY Functional Configuration .....	1165
8.2.3.2.2	ISS CSI2 PHY and Link Initialization Sequence .....	1166
8.2.3.2.3	ISS CCP2 and Link Initialization Sequence .....	1170
8.2.3.2.4	ISS CSI PHY Error Signals .....	1170
8.2.3.3	ISS CSI2 PHY Register Manual .....	1171
8.2.3.3.1	ISS CSI2 PHY Instance Summary .....	1171
8.2.3.3.2	ISS CSI2 PHY Registers .....	1171
8.2.4	ISS CCP2 .....	1174
8.2.4.1	ISS CCP2 Environment .....	1174
8.2.4.1.1	ISS CCP2 Protocol and Data Formats .....	1174
8.2.4.2	ISS CCP2 Integration .....	1187
8.2.4.3	ISS CCP2 Functional Description .....	1189
8.2.4.3.1	ISS CCP2 Overview .....	1189
8.2.4.3.2	ISS CCP2 PHY .....	1189
8.2.4.3.3	ISS CCP2 VP Interface .....	1190
8.2.4.3.4	ISS CCP2 Data Compression .....	1192
8.2.4.3.5	ISS CCP2 Memory Read Channel .....	1192
8.2.4.4	ISS CCP2 Programming Model .....	1200
8.2.4.4.1	ISS CCP2 Hardware Setup/Initialization .....	1200
8.2.4.4.2	ISS CCP2 Event and Status Checking .....	1200
8.2.4.4.3	ISS CCP2 Register Accessibility During Frame Processing .....	1201
8.2.4.4.4	ISS CCP2 Enable/Disable the Hardware .....	1201
8.2.4.4.5	ISS CCP2 Select the Signaling Scheme .....	1202
8.2.4.4.6	ISS CCP2 Select the Mode: MIPI CSI1 or CCP2 .....	1202
8.2.4.4.7	ISS CCP2 Burst Settings .....	1202
8.2.4.4.8	ISS CCP2 Debug Mode .....	1202
8.2.4.4.9	ISS CCP2 Video Port .....	1203
8.2.4.4.10	ISS CCP2 Logical Channels .....	1204
8.2.4.4.11	ISS CCP2 Controls .....	1204
8.2.4.4.12	ISS CCP2 Region-of-Interest .....	1204
8.2.4.4.13	ISS CCP2 CRC .....	1204
8.2.4.4.14	ISS CCP2 Destination Format .....	1205
8.2.4.4.15	ISS CCP2 Frame Acquisition .....	1205
8.2.4.4.16	ISS CCP2 Synchronization Codes .....	1205
8.2.4.4.17	ISS CCP2 Status Data .....	1205
8.2.4.4.18	ISS CCP2 Pixel Data Region .....	1206
8.2.4.4.19	ISS CCP2 Memory Read Channel .....	1208
8.2.4.5	ISS CCP2 Register Manual .....	1209
8.2.4.5.1	ISS CCP2 Instance Summary .....	1209
8.2.4.5.2	ISS CCP2 Registers .....	1210
8.2.5	ISS CSI2 .....	1244
8.2.5.1	ISS CSI2 Environment .....	1244
8.2.5.1.1	ISS CSI2 Protocol and Data Format .....	1244
8.2.5.2	ISS CSI2 Integration .....	1278
8.2.5.3	ISS CSI2 Functional Description .....	1280
8.2.5.3.1	ISS CSI2 Overview .....	1280
8.2.5.3.2	ISS CSI2 Features .....	1280
8.2.5.3.3	ISS CSI2 Functional Description .....	1281



8.2.5.4	ISS CSI2 Programming Model .....	1290
8.2.5.4.1	ISS CSI2 Reset Management .....	1290
8.2.5.4.2	ISS CSI2 Enable Video/Picture Acquisition .....	1290
8.2.5.4.3	ISS CSI2 Disable Video/Picture Acquisition .....	1292
8.2.5.4.4	ISS CSI2 Capture a Finite Number of Frames .....	1292
8.2.5.4.5	ISS CSI2 Configure a Periodic Event During Frame Acquisition .....	1292
8.2.5.4.6	ISS CSI2 Linking a Context to a Virtual Channel and a Data Type .....	1292
8.2.5.4.7	ISS CSI2 Progressive and Interleaved Frame Configuration .....	1293
8.2.5.4.8	ISS CSI2 Progressive and Interleaved Frame Configuration .....	1293
8.2.5.5	ISS CSI2 Register Manual .....	1293
8.2.5.5.1	ISS CSI2 Instance Summary .....	1293
8.2.5.5.2	ISS CSI2 REGS1 Registers .....	1294
8.2.5.5.3	ISS CSI2 REGS2 Registers .....	1322
8.2.6	ISS TCTRL .....	1325
8.2.6.1	ISS TCTRL Environment .....	1325
8.2.6.2	ISS TCTRL Integration .....	1325
8.2.6.3	ISS TCTRL Functional Description .....	1327
8.2.6.3.1	ISS TCTRL Features .....	1327
8.2.6.3.2	ISS TCTRL Control-Signal Generator .....	1327
8.2.6.4	ISS TCTRL Programming Model .....	1330
8.2.6.4.1	ISS TCTRL Camera-Control Signal Generator .....	1330
8.2.6.5	ISS TCTRL Register Manual .....	1333
8.2.6.5.1	ISS TCTRL Instance Summary .....	1333
8.2.6.5.2	ISS TCTRL Registers .....	1333
8.2.7	ISS BTE .....	1343
8.2.7.1	ISS BTE Environment .....	1343
8.2.7.2	ISS BTE Integration .....	1344
8.2.7.2.1	ISS BTE PRCM Interface .....	1345
8.2.7.3	ISS BTE Functional Description .....	1345
8.2.7.3.1	ISS BTE Features .....	1345
8.2.7.3.2	ISS BTE Functional Description Details .....	1345
8.2.7.4	ISS BTE Programming Model .....	1356
8.2.7.4.1	ISS BTE Reset .....	1356
8.2.7.4.2	ISS BTE Interrupts .....	1356
8.2.7.4.3	ISS BTE Context Configuration .....	1357
8.2.7.4.4	ISS BTE Change Context Configuration .....	1357
8.2.7.5	ISS BTE Register Manual .....	1358
8.2.7.5.1	ISS BTE Instance Summary .....	1358
8.2.7.5.2	ISS BTE Registers .....	1358
8.2.8	ISS CBUFF .....	1380
8.2.8.1	ISS CBUFF Environment .....	1380
8.2.8.2	ISS CBUFF Integration .....	1380
8.2.8.2.1	ISS CBUFF PRCM Interface .....	1381
8.2.8.2.2	ISS CBUFF Interrupts .....	1382
8.2.8.3	ISS CBUFF Functional Description .....	1383
8.2.8.3.1	ISS CBUFF Features .....	1383
8.2.8.3.2	ISS CBUFF Functional Description .....	1383
8.2.8.4	ISS CBUFF Programming Model .....	1398
8.2.8.4.1	ISS CBUFF Reset Behavior .....	1398
8.2.8.4.2	ISS CBUFF Register Setup .....	1398
8.2.8.5	ISS CBUFF Register Manual .....	1398
8.2.8.5.1	ISS CBUFF Instance Summary .....	1398
8.2.8.5.2	ISS CBUFF Registers .....	1399

8.3	ISS ISP .....	1421
8.3.1	ISS ISP Overview .....	1421
8.3.1.1	ISS ISP Features .....	1421
8.3.1.2	ISS ISP Block Diagram .....	1422
8.3.2	ISS ISP Integration .....	1423
8.3.2.1	ISS ISP PRCM Interface .....	1423
8.3.2.1.1	ISS ISP Clocks .....	1423
8.3.2.1.2	ISS ISP Reset .....	1423
8.3.2.2	ISS ISP Interrupt Tree .....	1423
8.3.2.3	ISS ISP PG Integration .....	1424
8.3.2.4	ISS ISP IPIPEIF Integration .....	1424
8.3.2.4.1	ISS ISP IPIPEIF Interrupts .....	1424
8.3.2.5	ISS ISP IPIPE Integration .....	1425
8.3.2.5.1	ISS ISP IPIPE Interrupts .....	1425
8.3.2.5.2	ISS ISP DMA Requests .....	1425
8.3.2.6	ISS ISP RSZ Integration .....	1426
8.3.2.6.1	ISS ISP RSZ PRCM Interface .....	1426
8.3.2.6.2	ISS ISP RSZ Interrupts .....	1426
8.3.2.7	ISS ISP H3A Integration .....	1427
8.3.2.7.1	ISS ISP H3A Interrupts .....	1427
8.3.2.8	ISS ISP ISIF Integration .....	1427
8.3.2.8.1	ISS ISP ISIF Interrupts .....	1427
8.3.2.9	ISS ISP BL Integration .....	1428
8.3.3	ISS ISP Functional Description .....	1429
8.3.3.1	ISS ISP VP Functional Description .....	1429
8.3.3.1.1	ISS ISP VP Overview .....	1429
8.3.3.1.2	ISS ISP VP Data Formats .....	1430
8.3.3.1.3	ISS ISP VP Top-Level Communication With CCP2 RX and CSI2 RX .....	1431
8.3.3.1.4	ISS ISP VP Pixel Clock Inversion .....	1433
8.3.3.2	ISS ISP PG Functional Description .....	1434
8.3.3.2.1	ISS ISP PG Overview .....	1434
8.3.3.2.2	ISS ISP PG Functional Description .....	1435
8.3.3.3	ISS ISP IPIPEIF Functional Description .....	1436
8.3.3.3.1	ISS ISP IPIPEIF Overview .....	1436
8.3.3.3.2	ISS ISP IPIPEIF Top-Level Block Diagram .....	1437
8.3.3.3.3	ISS ISP IPIPEIF Input Interface .....	1438
8.3.3.3.4	ISS ISP IPIPEIF Data Path Selection .....	1441
8.3.3.3.5	ISS ISP IPIPEIF Timing Generation .....	1450
8.3.3.3.6	ISS ISP IPIPEIF Decompression (DCPM) Subblock: Unpack and Decompression Function .....	1451
8.3.3.3.7	ISS ISP IPIPEIF Dark-Frame Subtraction Functionality .....	1453
8.3.3.3.8	ISS ISP IPIPEIF (1, 2, 1) Averaging Filter for IPIPE Data Path .....	1454
8.3.3.3.9	ISS ISP IPIPEIF Horizontal Pixel Decimator (Downsizer) for IPIPE Data Path .....	1455
8.3.3.3.10	ISS ISP IPIPEIF RAW Data Gain for IPIPE Data Path .....	1456
8.3.3.3.11	ISS ISP IPIPEIF (1, 2, 1) Averaging Filter for H3A Data Path .....	1456
8.3.3.3.12	ISS ISP IPIPEIF Horizontal Pixel Decimator (Downsizer) for H3A Data Path .....	1456
8.3.3.3.13	ISS ISP IPIPEIF YUV4:2:2 8-bits packed Data Input Coming from ISIF Module .....	1456
8.3.3.3.14	ISS ISP IPIPEIF YUV4:2:0 Data Input for Memory-to-Memory Resize Operations .....	1457
8.3.3.3.15	ISS ISP IPIPEIF Module Events and Status Checking .....	1458
8.3.3.4	ISS ISP IPIPE Functional Description .....	1458
8.3.3.4.1	ISS ISP IPIPE Overview .....	1458
8.3.3.4.2	ISS ISP IPIPE Top-Level Block Diagram .....	1460
8.3.3.4.3	ISS ISP IPIPE Input Interface .....	1460
8.3.3.4.4	ISS ISP IPIPE Defect Pixel Correction .....	1462

8.3.3.4.5	ISS ISP IPIPE White Balance .....	1464
8.3.3.4.6	ISS ISP IPIPE RGB2RGB Blending Module .....	1464
8.3.3.4.7	ISS ISP IPIPE Gamma Correction Module .....	1465
8.3.3.4.8	ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix .....	1466
8.3.3.4.9	ISS ISP IPIPE RGB2YCbCr Conversion Matrix .....	1466
8.3.3.4.10	ISS ISP IPIPE 4:2:2 Conversion Module .....	1467
8.3.3.4.11	ISS ISP IPIPE 2D Edge Enhancer .....	1468
8.3.3.4.12	ISS ISP IPIPE Histogram .....	1471
8.3.3.4.13	ISS ISP IPIPE Boxcar .....	1472
8.3.3.5	ISS ISP RSZ Functional Description .....	1473
8.3.3.5.1	ISS ISP RSZ Overview .....	1473
8.3.3.5.2	ISS ISP RSZ Top-Level Block Diagram .....	1476
8.3.3.5.3	ISS ISP RSZ Interfaces .....	1476
8.3.3.5.4	ISS ISP RSZ Integration .....	1478
8.3.3.5.5	ISS ISP RSZ Functional Description .....	1480
8.3.3.6	ISS ISP H3A Functional Description .....	1494
8.3.3.6.1	ISS ISP H3A Overview .....	1494
8.3.3.6.2	ISS ISP H3A Top-Level Block Diagram .....	1496
8.3.3.6.3	ISS ISP H3A Line Framing Logic .....	1497
8.3.3.6.4	ISS ISP H3A Optional Preprocessing .....	1498
8.3.3.6.5	ISS ISP H3A Autofocus Engine .....	1498
8.3.3.6.6	ISS ISP H3A AE/AWB Engine .....	1503
8.3.3.6.7	ISS ISP H3A DMA Interface .....	1505
8.3.3.6.8	ISS ISP H3A Events and Status Checking .....	1512
8.3.3.7	ISS ISP ISIF Functional Description .....	1513
8.3.3.7.1	ISS ISP ISIF Overview .....	1513
8.3.3.7.2	ISS ISP ISIF Top-Level Block Diagram .....	1514
8.3.3.7.3	ISS ISP ISIF Input Interface .....	1516
8.3.3.7.4	ISS ISP ISIF Interface .....	1516
8.3.3.7.5	ISS ISP ISIF Sensor Linearization .....	1517
8.3.3.7.6	ISS ISP ISIF Input Data Formatter .....	1519
8.3.3.7.7	ISS ISP ISIF Color Space Converter .....	1527
8.3.3.7.8	ISS ISP ISIF Black Clamp .....	1530
8.3.3.7.9	ISS ISP ISIF Vertical Line Defect Correction (VDFC) .....	1534
8.3.3.7.10	ISS ISP ISIF Lens Shading Correction (2D-LSC) .....	1537
8.3.3.7.11	ISS ISP ISIF White Balance .....	1542
8.3.3.7.12	ISS ISP ISIF Low-Pass Filter (LPF) .....	1543
8.3.3.7.13	ISS ISP ISIF A-Law Compression .....	1543
8.3.3.7.14	ISS ISP ISIF Culling .....	1546
8.3.3.7.15	ISS ISP ISIF 12-to-8 bit DCPM Compression Block .....	1547
8.3.3.7.16	ISS ISP ISIF Storage Formatter .....	1548
8.3.3.7.17	ISS ISP ISIF YCbCr signal Processing .....	1551
8.3.3.7.18	ISS ISP ISIF Expected Bandwidth on BL Ports .....	1551
8.3.3.7.19	ISS ISP ISIF Events and Status Checking .....	1552
8.3.3.8	ISS ISP BL Functional Description .....	1553
8.3.3.8.1	ISS ISP BL Overview .....	1553
8.3.3.8.2	ISS ISP BL Functional Description .....	1554
8.3.3.8.3	ISS ISP BL Address Alignment .....	1557
8.3.3.8.4	ISS ISP BL Out-of-Order Responses .....	1557
8.3.3.8.5	ISS ISP BL Stalling .....	1557
8.3.3.8.6	ISS ISP BL Dynamic and Static MFlag Generation .....	1558
8.3.3.8.7	ISS ISP BL VBUSM2OCP Last Beat Command Delay .....	1559
8.3.3.8.8	ISS ISP BL Peak Memory Bandwidth Reduction .....	1559

8.3.3.9	ISS ISP Memory Mapping .....	1559
8.3.4	ISS ISP Programming Model .....	1561
8.3.4.1	ISS ISP PG Programming Model .....	1561
8.3.4.2	ISS ISP ISIF Programming Model .....	1561
8.3.4.2.1	ISS ISP ISIF Hardware Setup/Initialization .....	1561
8.3.4.2.2	ISS ISP ISIF Enable/Disable Hardware .....	1568
8.3.4.2.3	ISS ISP ISIF Register Accessibility During Frame Processing .....	1569
8.3.4.2.4	ISS ISP ISIF Interframe Operations .....	1569
8.3.4.2.5	ISS ISP ISIF Summary of Constraints .....	1569
8.3.4.3	ISS ISP IPIPEIF Programming Model .....	1570
8.3.4.3.1	ISS ISP IPIPEIF Hardware Setup/Initialization .....	1570
8.3.4.3.2	ISS ISP IPIPEIF Enable/Disable Hardware .....	1572
8.3.4.3.3	ISS ISP IPIPEIF Register Accessibility During Frame Processing .....	1572
8.3.4.3.4	ISS ISP IPIPEIF Interframe Operations .....	1573
8.3.4.3.5	ISS ISP IPIPEIF Summary of Constraints .....	1573
8.3.4.4	ISS ISP IPIPE Programming Model .....	1573
8.3.4.4.1	Global Initialization .....	1573
8.3.4.4.2	ISS ISP IPIPE Operational Modes Configuration .....	1577
8.3.4.5	ISS ISP RSZ Programming Model .....	1579
8.3.4.5.1	ISS ISP RSZ Hardware Setup/Initialization .....	1579
8.3.4.5.2	ISS ISP RSZ Global Image Processing Settings .....	1581
8.3.4.5.3	ISS ISP RSZ Engines Interframe Image Processing Settings .....	1584
8.3.4.5.4	ISS ISP RSZ Programming Constraints .....	1588
8.3.4.6	ISS ISP H3A Programming Model .....	1588
8.3.4.6.1	ISS ISP H3A Hardware Setup/Initialization .....	1588
8.3.4.6.2	ISS ISP H3A Enable/Disable Hardware .....	1590
8.3.4.6.3	ISS ISP H3A Register Accessibility During Frame Processing .....	1591
8.3.4.6.4	ISS ISP H3A Interframe Operations .....	1591
8.3.4.6.5	ISS ISP H3A Summary of Constraints .....	1591
8.3.4.7	ISS ISP BL Programming Model .....	1593
8.3.5	ISS ISP Register Manual .....	1594
8.3.5.1	ISS ISP Instance Summary .....	1594
8.3.5.2	ISS ISP5 SYS1 Registers .....	1594
8.3.5.2.1	ISS ISP5 SYS1 Register Summary .....	1594
8.3.5.2.2	ISS ISP5 SYS1 Register Description .....	1594
8.3.5.3	ISS ISP5 SYS2 Registers .....	1629
8.3.5.3.1	ISS ISP5 SYS2 Register Summary .....	1629
8.3.5.3.2	ISS ISP5 SYS2 Register Description .....	1630
8.3.5.4	ISS RESIZER Registers .....	1638
8.3.5.4.1	ISS RESIZER Register Summary .....	1638
8.3.5.4.2	ISS RESIZER Register Description .....	1640
8.3.5.5	ISS IPIPE registers .....	1695
8.3.5.5.1	ISS IPIPE Register Summary .....	1695
8.3.5.5.2	ISS IPIPE Register Description .....	1699
8.3.5.6	ISS ISIF Registers .....	1765
8.3.5.6.1	ISS ISIF Register Summary .....	1765
8.3.5.6.2	ISS ISIF Register Description .....	1768
8.3.5.7	ISS IPIPEIF Registers .....	1838
8.3.5.7.1	ISS IPIPEIF Register Summary .....	1838
8.3.5.7.2	ISS IPIPEIF Register Description .....	1838
8.3.5.8	ISS H3A Registers .....	1852
8.3.5.8.1	ISS H3A Register Summary .....	1852
8.3.5.8.2	ISS H3A Register Description .....	1853

8.4	ISS Still Image Coprocessor .....	1875
8.4.1	ISS SIMCOP Overview .....	1875
8.4.1.1	ISS SIMCOP Integration .....	1876
8.4.1.2	ISS SIMCOP Functional Description .....	1877
8.4.1.2.1	ISS SIMCOP Local Power and Clock Management .....	1877
8.4.1.2.2	Software Reset .....	1879
8.4.1.2.3	Interrupt Merger .....	1879
8.4.1.2.4	ISS SIMCOP Modules Description .....	1881
8.4.1.3	ISS SIMCOP Programming Models .....	1881
8.4.1.3.1	Global Initialization .....	1881
8.4.1.3.2	ISS SIMCOP Operational Modes Configuration .....	1882
8.4.1.4	ISS SIMCOP Register Manual .....	1884
8.4.1.4.1	SIMCOP Instance Summary .....	1884
8.4.1.4.2	SIMCOP Registers .....	1884
8.4.1.5	ISS SIMCOP Memory Mapping .....	1903
8.4.2	ISS SIMCOP Hardware Sequencer and Buffers .....	1903
8.4.2.1	ISS SIMCOP Hardware Sequencer and Buffers Overview .....	1903
8.4.2.2	ISS SIMCOP Hardware Sequencer and Buffer Integration .....	1904
8.4.2.3	ISS SIMCOP Hardware Sequencer and Buffers Functional Description .....	1907
8.4.2.3.1	ISS SIMCOP Hardware Sequencer and Buffers Software Reset .....	1907
8.4.2.3.2	ISS SIMCOP Hardware Sequencer and Buffers Power Management .....	1907
8.4.2.3.3	ISS SIMCOP Hardware Sequencer and Buffer Interrupt Requests .....	1907
8.4.2.3.4	ISS SIMCOP Hardware Sequencer Buffer Description .....	1907
8.4.2.3.5	ISS SIMCOP Hardware Sequencer .....	1914
8.4.2.4	ISS SIMCOP Hardware Sequencer and Buffers Basic Programming Model .....	1918
8.4.2.4.1	ISS SIMCOP Application Programming Principle .....	1918
8.4.2.4.2	External CPU Use for Data Processing .....	1919
8.4.2.4.3	Rotation Operational Mode Configuration .....	1919
8.4.2.4.4	LDC Operational Mode Configuration .....	1921
8.4.2.4.5	Concurrent Software and Hardware Sequencing .....	1922
8.4.2.5	ISS SIMCOP Hardware Sequencer and Buffers Register Manual .....	1923
8.4.2.5.1	Hardware Sequencer Instance Summary .....	1923
8.4.2.5.2	Hardware Sequencer Registers .....	1923
8.4.3	ISS SIMCOP DMA Module .....	1941
8.4.3.1	ISS SIMCOP DMA Overview .....	1941
8.4.3.2	ISS SIMCOP DMA Integration .....	1942
8.4.3.3	ISS SIMCOP DMA Functional Description .....	1944
8.4.3.3.1	ISS SIMCOP DMA Block Diagram .....	1944
8.4.3.3.2	ISS SIMCOP DMA Power Management .....	1944
8.4.3.3.3	ISS SIMCOP DMA Interrupt Requests .....	1944
8.4.3.3.4	ISS SIMCOP DMA Logical Channels .....	1945
8.4.3.3.5	Transaction Generation .....	1948
8.4.3.4	ISS SIMCOP DMA Basic Programming Model .....	1949
8.4.3.4.1	Initialization of Surrounding Modules .....	1949
8.4.3.4.2	ISS SIMCOP DMA Channel Configuration and Hardware Synchronization .....	1950
8.4.3.4.3	Software Synchronization .....	1952
8.4.3.5	ISS SIMCOP DMA Register Manual .....	1953
8.4.3.5.1	ISS SIMCOP DMA Instance Summary .....	1953
8.4.3.5.2	ISS SIMCOP DMA Registers .....	1953
8.4.4	ISS SIMCOP LDC Module .....	1972
8.4.4.1	ISS SIMCOP LDC Overview .....	1972
8.4.5	ISS SIMCOP Discrete Cosine Transform Module .....	1974
8.4.5.1	ISS SIMCOP DCT Overview .....	1974

8.4.5.2	ISS SIMCOP DCT Integration .....	1975
8.4.5.3	ISS SIMCOP DCT Functional Description .....	1977
8.4.5.3.1	ISS SIMCOP DCT Block Diagram .....	1977
8.4.5.3.2	ISS SIMCOP DCT Power Management .....	1977
8.4.5.3.3	ISS SIMCOP DCT Interrupt Requests .....	1977
8.4.5.3.4	Control and Status .....	1978
8.4.5.3.5	Data Format .....	1979
8.4.5.4	ISS SIMCOP DCT Basic Programming Model .....	1982
8.4.5.4.1	Initialization of Surrounding Modules .....	1982
8.4.5.4.2	Using the DCT for DCT Processing .....	1982
8.4.5.4.3	Using the DCT for IDCT Processing .....	1982
8.4.5.5	ISS SIMCOP DCT Register Manual .....	1984
8.4.5.5.1	ISS SIMCOP DCT Instance Summary .....	1984
8.4.5.5.2	DCT Registers .....	1984
8.4.6	ISS SIMCOP Variable Length Coder/Decoder for JPEG Module .....	1989
8.4.6.1	ISS SIMCOP VLCDJ Overview .....	1989
8.4.6.2	ISS SIMCOP VLCDJ Integration .....	1990
8.4.6.3	ISS SIMCOP VLCDJ Functional Description .....	1992
8.4.6.3.1	ISS SIMCOP VLCDJ Block Diagram .....	1992
8.4.6.3.2	ISS SIMCOP VLCDJ Power Management .....	1992
8.4.6.3.3	ISS SIMCOP VLCDJ Interrupt Requests .....	1992
8.4.6.3.4	ISS SIMCOP VLCDJ in JPEG Encoding Flow .....	1993
8.4.6.3.5	ISS SIMCOP VLCDJ in JPEG Decoding Flow .....	1993
8.4.6.3.6	Memory Interfaces .....	1994
8.4.6.3.7	Huffman Table Organization .....	1996
8.4.6.3.8	Restart Marker .....	1997
8.4.6.3.9	Encode/Decode Context Switching .....	1997
8.4.6.3.10	Using Sequential Block Mode for YUV4:4:4 Encoding/Decoding .....	1997
8.4.6.3.11	Error Reporting .....	1998
8.4.6.4	ISS SIMCOP VLCDJ Basic Programming Model .....	1999
8.4.6.4.1	Initialization of Surrounding Modules .....	1999
8.4.6.4.2	ISS SIMCOP VLCDJ Start and Register Modification During Processing .....	1999
8.4.6.4.3	JPEG Encoding .....	1999
8.4.6.4.4	JPEG Decoding .....	2000
8.4.6.4.5	Concurrent JPEG Encoding and Decoding .....	2002
8.4.6.4.6	Example of YUV4:4:4 Encoding .....	2004
8.4.6.4.7	Example of Huffman Encode and Decode Tables .....	2005
8.4.6.5	ISS SIMCOP VLCDJ Register Manual .....	2007
8.4.6.5.1	ISS SIMCOP VLCDJ Instance Summary .....	2007
8.4.6.5.2	VLCDJ Registers .....	2007
8.4.7	ISS SIMCOP Rotation Accelerator Module .....	2020
8.4.7.1	ISS SIMCOP ROT Overview .....	2020
8.4.7.2	ISS SIMCOP ROT Integration .....	2021
8.4.7.3	ISS SIMCOP ROT Functional Description .....	2024
8.4.7.3.1	ISS SIMCOP ROT Block Diagram .....	2024
8.4.7.3.2	ISS SIMCOP ROT Clock Configuration .....	2024
8.4.7.3.3	ISS SIMCOP ROT Reset .....	2024
8.4.7.3.4	ISS SIMCOP ROT Power Management .....	2024
8.4.7.3.5	ISS SIMCOP ROT Interrupt Requests .....	2024
8.4.7.3.6	ISS SIMCOP ROT DMA Requests .....	2024
8.4.7.3.7	ISS SIMCOP ROT Synchronization .....	2025
8.4.7.3.8	ISS SIMCOP ROT Formats and Operations .....	2025
8.4.7.4	ISS SIMCOP ROT Module Programming Guide .....	2031

8.4.7.4.1	ISS SIMCOP ROT Low-Level Programming Models .....	2031
8.4.7.5	ISS SIMCOP ROT Register Manual .....	2033
8.4.7.5.1	ISS SIMCOP ROT Instance Summary .....	2033
8.4.7.5.2	ISS SIMCOP ROT Registers .....	2033
<b>9</b>	<b>Face Detect .....</b>	<b>2041</b>
9.1	Face Detect Overview .....	2042
9.1.1	Main Features .....	2042
9.2	FDIF Integration .....	2044
9.3	FDIF Functional Description .....	2046
9.3.1	FDIF Block Description .....	2046
9.3.2	Software Reset .....	2046
9.3.3	System Power Management .....	2047
9.3.3.1	Autogating .....	2047
9.3.3.2	PRCM Hardware Handshake .....	2047
9.3.3.2.1	Protocol Transitions .....	2047
9.3.4	Interrupts and Events .....	2048
9.3.5	Typical Use .....	2049
9.3.6	Performance Parameters .....	2049
9.3.7	L3 Interconnect Parameters .....	2049
9.4	FDIF Programming Guide .....	2050
9.4.1	FDIF Low-level Programming Models .....	2050
9.4.1.1	Global Initialization .....	2050
9.4.1.1.1	Surrounding Modules Global Initialization .....	2050
9.4.1.1.2	FDIF Global Initialization .....	2050
9.4.1.2	FD Operational Modes Configuration .....	2051
9.4.1.2.1	FD Processing Modes .....	2051
9.5	FDIF Register Manual .....	2053
9.5.1	FDIF Instance Summary .....	2053
9.5.2	FDIF Registers .....	2053
9.5.2.1	FDIF Register Summary .....	2053
9.5.2.2	FDIF Register Description .....	2054
<b>10</b>	<b>Display Subsystem .....</b>	<b>2069</b>
10.1	Display Subsystem Overview .....	2070
10.1.1	DSS Environment .....	2072
10.1.1.1	DSS LCD Support .....	2073
10.1.1.1.1	Parallel Interface .....	2073
10.1.1.1.2	Serial Interfaces .....	2073
10.1.1.2	DSS TV Display Support .....	2074
10.1.1.2.1	Video Encoder .....	2074
10.1.1.2.2	HDMI .....	2074
10.1.2	DSS Integration .....	2075
10.1.2.1	DSS Clocks .....	2075
10.1.2.2	DSS Resets .....	2078
10.1.2.2.1	Hardware Reset .....	2079
10.1.2.3	DSS Power Management .....	2079
10.1.2.3.1	Standby Mode .....	2079
10.1.2.3.2	Wake-Up Mode .....	2080
10.1.2.4	DSS Interrupt Requests .....	2081
10.1.2.5	DSS DMA Requests .....	2083
10.1.3	DSS Register Manual .....	2085
10.1.3.1	DSS Instance Summary .....	2085
10.1.3.2	DSS Registers .....	2085
10.1.3.2.1	DSS Registers Mapping Summary .....	2085

10.1.3.2.2	DSS Register Description .....	2085
10.2	Display Controller .....	2091
10.2.1	Display Controller Overview .....	2091
10.2.2	Display Controller Environment .....	2095
10.2.2.1	LCD Output and Data Format for the Parallel Interface .....	2096
10.2.2.2	Transaction Timing Diagrams .....	2101
10.2.2.3	TV Output and Data Format for the Parallel Interface .....	2106
10.2.3	Display Controller Integration .....	2108
10.2.4	Display Controller Functional Description .....	2111
10.2.4.1	Clock Configuration .....	2112
10.2.4.2	Software Reset .....	2112
10.2.4.3	Power Management .....	2112
10.2.4.3.1	Idle Mode .....	2112
10.2.4.3.2	StandBy Mode .....	2113
10.2.4.3.3	Wakeup .....	2114
10.2.4.4	Interrupt Requests .....	2114
10.2.4.5	sDMA Requests .....	2116
10.2.4.6	DMA Engine .....	2116
10.2.4.6.1	Addressing and Bursts .....	2117
10.2.4.6.2	DMA Buffers .....	2118
10.2.4.6.3	Predecimation .....	2120
10.2.4.6.4	Arbitration .....	2120
10.2.4.6.5	DMA Power Modes .....	2120
10.2.4.7	Rotation and Mirroring .....	2121
10.2.4.8	Memory Format .....	2124
10.2.4.9	Graphics Pipeline .....	2126
10.2.4.9.1	Replication Logic .....	2127
10.2.4.9.2	Antiflicker Filter .....	2128
10.2.4.9.3	Color Look-Up Table (CLUT) .....	2128
10.2.4.9.4	Expansion to ARGB40 .....	2129
10.2.4.10	Video Pipelines .....	2130
10.2.4.10.1	Replication Logic .....	2131
10.2.4.10.2	VC-1 Range Mapping Unit .....	2132
10.2.4.10.3	Color Space Conversion Unit YUV to RGB .....	2132
10.2.4.10.4	Scaler Unit .....	2135
10.2.4.11	Write-back Pipeline .....	2144
10.2.4.11.1	Color Space Conversion Unit RGB to YUV .....	2145
10.2.4.11.2	Scaler Unit .....	2146
10.2.4.11.3	RGB Truncation Logic .....	2150
10.2.4.12	LCD Outputs .....	2150
10.2.4.12.1	Overlay Manager .....	2151
10.2.4.12.2	Gamma Correction Unit .....	2159
10.2.4.12.3	Color Phase Rotation Unit .....	2159
10.2.4.12.4	Passive Matrix .....	2160
10.2.4.12.5	Active Matrix .....	2161
10.2.4.12.6	Synchronized Buffer Update .....	2165
10.2.4.12.7	Timing Generator and Panel Settings .....	2166
10.2.4.12.8	Stall mode .....	2167
10.2.4.13	TV Output .....	2168
10.2.4.13.1	Overlay Manager .....	2169
10.2.4.13.2	Gamma Correction Unit .....	2169
10.2.4.13.3	Synchronized Buffer Update .....	2169
10.2.4.13.4	Timing and TV Format Settings .....	2169



10.2.4.14	Shadows Registers .....	2171
10.2.5	Display Controller Programming Guide .....	2177
10.2.5.1	Display Controller Low-level Programming Models .....	2177
10.2.5.1.1	Global Initialization .....	2177
10.2.5.1.2	Operational Modes Configuration .....	2178
10.2.6	Display Controller Use Cases and Tips .....	2192
10.2.6.1	Hardware Cursor .....	2192
10.2.7	Display Controller Register Manual .....	2192
10.2.7.1	Display Controller Instance Summary .....	2192
10.2.7.2	Display Controller Logical Register Mapping .....	2192
10.2.7.3	Display Controller Registers .....	2198
10.2.7.3.1	Display Controller Register Summary .....	2198
10.2.7.3.2	Display Controller Register Description .....	2203
10.3	MIPI Display Serial Interface .....	2329
10.3.1	MIPI DSI Overview .....	2329
10.3.2	DSI Environment .....	2332
10.3.2.1	DSI PHY .....	2333
10.3.2.1.1	Data/Clock Configuration .....	2334
10.3.2.1.2	ULPS .....	2336
10.3.2.2	DSI Protocol Layer .....	2336
10.3.2.2.1	Short Packet .....	2337
10.3.2.2.2	Long Packet .....	2337
10.3.2.2.3	Data Identifier .....	2338
10.3.2.2.4	Synchronization Codes .....	2339
10.3.2.2.5	Blanking .....	2340
10.3.2.2.6	Frame Structures .....	2343
10.3.2.2.7	VCs .....	2347
10.3.2.3	DSI Multilane Layer .....	2347
10.3.2.3.1	SoT and EoT in Multilane Configurations .....	2348
10.3.2.3.2	Lane Splitter .....	2348
10.3.2.4	DSI Pixel Data Formats .....	2351
10.3.2.4.1	Pixel Data Formats in Video Mode .....	2351
10.3.2.4.2	24 bpp – RGB Color Format, Long Packet (Video Mode) .....	2352
10.3.2.4.3	24 Bits per Pixel - RGB Color Format, Long Packet (Command Mode) .....	2352
10.3.2.4.4	18 bpp (Loosely Packed) – RGB Color Format, Long Packet (Transparent for DSI Protocol Engine) (Video Mode) .....	2353
10.3.2.4.5	18-bpp (Loosely Packed) – RGB Color Format, Long Packet (Command Mode) .....	2354
10.3.2.4.6	18-bpp (Packed) – RGB Color Format, Long Packet (Video Mode) .....	2355
10.3.2.4.7	16 bpp – RGB Color Format, Long Packet (Video Mode) .....	2356
10.3.2.4.8	16 bpp – RGB Color Format, Long Packet (Command Mode) .....	2357
10.3.3	DSI Integration .....	2359
10.3.4	DSI Functional Description .....	2362
10.3.4.1	DSI Block Diagram .....	2362
10.3.4.2	DSI Clock Configuration .....	2362
10.3.4.3	DSI Interrupt Requests .....	2364
10.3.4.4	DSI Protocol Engine .....	2365
10.3.4.4.1	DSI Protocol Overview .....	2365
10.3.4.4.2	DSI Global Register Controls .....	2365
10.3.4.4.3	DSI Transfer Modes .....	2366
10.3.4.4.4	Clock Requirements .....	2378
10.3.4.4.5	Video Port Interface .....	2382
10.3.4.4.6	SCP Interface .....	2391
10.3.4.4.7	Timers .....	2392
10.3.4.4.8	Bus Turnaround .....	2396

10.3.4.4.9	PHY Triggers .....	2397
10.3.4.4.10	ECC Generation .....	2400
10.3.4.4.11	Checksum Generation for Long Packet Payloads .....	2401
10.3.4.4.12	EOT Packet .....	2401
10.3.4.4.13	Software Reset .....	2402
10.3.4.4.14	Power Management .....	2402
10.3.4.4.15	Power Control of DSI PHY Complex I/O and DSI PLL .....	2402
10.3.4.4.16	Ultralow-Power State Configuration .....	2406
10.3.4.4.17	Interrupts .....	2407
10.3.4.5	DSI PLL Controllers .....	2408
10.3.4.5.1	DSI PLL Controllers Overview .....	2408
10.3.4.5.2	DSI PLL Controllers Architecture .....	2409
10.3.4.5.3	DSI PLL Operations .....	2410
10.3.4.5.4	DSI PLL Controllers Shadowing Mechanism .....	2411
10.3.4.5.5	Error Handling .....	2411
10.3.4.5.6	Software Reset .....	2411
10.3.4.5.7	Power Management .....	2411
10.3.4.5.8	DSI PLL Clock-Gating Sequence .....	2412
10.3.4.5.9	DSI PLL Clock Sequence .....	2414
10.3.4.5.10	DSI PLL Go Sequence .....	2416
10.3.4.5.11	DSI PLL Recommended Values .....	2417
10.3.4.6	DSI Complex I/O .....	2419
10.3.4.6.1	DSI Complex I/O Overview .....	2419
10.3.4.6.2	Software Reset .....	2419
10.3.4.6.3	Reset-Done Bits .....	2419
10.3.4.6.4	Pad Configuration .....	2419
10.3.4.6.5	Display Timing Configuration .....	2420
10.3.4.6.6	Power Management .....	2424
10.3.4.6.7	Error Handling .....	2425
10.3.5	DSI Programming Model .....	2428
10.3.5.1	DSI Global Initialization .....	2428
10.3.5.1.1	Surrounding Modules Global Initialization .....	2428
10.3.5.1.2	DSI Global Initialization .....	2428
10.3.5.2	DSI Operational Modes Configuration .....	2429
10.3.5.2.1	DSI Video Mode Using the DISPC Video Port .....	2429
10.3.5.2.2	DSI Command Mode Using the DISPC Video Port .....	2432
10.3.6	DSI Register Manual .....	2437
10.3.6.1	DSI Instance Summary .....	2437
10.3.6.2	DSI_PROTOCOL_ENGINE Registers .....	2437
10.3.6.2.1	DSI_PROTOCOL_ENGINE Register Summary .....	2437
10.3.6.2.2	DSI_PROTOCOL_ENGINE Register Description .....	2440
10.3.6.3	DSI_PHY Registers .....	2493
10.3.6.3.1	DSI_PHY Register Summary .....	2493
10.3.6.3.2	DSI_PHY Register Description .....	2494
10.3.6.4	DSI_PLLCTRL Registers .....	2499
10.3.6.4.1	DSI_PLLCTRL Register Summary .....	2499
10.3.6.4.2	DSI_PLLCTRL Register Description .....	2500
10.4	High-Definition Multimedia Interface .....	2509
10.4.1	High-Definition Multimedia Interface (HDMI) Overview .....	2509
10.4.1.1	HDMI Main Features .....	2509
10.4.1.2	HDMI Video Formats and Timings .....	2510
10.4.1.2.1	CEA-861-D video formats and timings .....	2510
10.4.1.2.2	VESA DMT video formats and timings .....	2510

10.5	Remote Frame Buffer Interface .....	2512
10.5.1	Remote Frame Buffer Interface Overview .....	2512
10.5.2	RFBI Environment .....	2514
10.5.2.1	RFBI Parallel Interface (MIPI DBI Protocol) .....	2514
10.5.2.1.1	Description of the Tearing Effect Pulse Signal .....	2515
10.5.2.2	Transaction Timing Diagrams .....	2516
10.5.3	RFBI Integration .....	2519
10.5.3.1	DMA Requests .....	2520
10.5.4	RFBI Functional Description .....	2521
10.5.4.1	Block Diagram .....	2521
10.5.4.2	Clock Configuration .....	2521
10.5.4.3	Software Reset .....	2521
10.5.4.4	Power Management .....	2521
10.5.4.5	Interrupt Requests .....	2522
10.5.4.6	DMA Requests .....	2523
10.5.4.7	Video Port FIFO .....	2523
10.5.4.7.1	Description .....	2523
10.5.4.7.2	RFBI Input Formats .....	2523
10.5.4.7.3	Stall Mechanism .....	2524
10.5.4.8	Interconnect FIFO .....	2525
10.5.4.8.1	Description .....	2525
10.5.4.8.2	Using DMA Request With Interconnect FIFO .....	2525
10.5.4.9	RFBI Output Parallel Modes .....	2527
10.5.4.9.1	Cycle Mode Selection .....	2527
10.5.4.9.2	Unmodified Bits .....	2529
10.5.4.9.3	Number of Pixels to Transfer .....	2530
10.5.4.10	RFBI Timing Generator .....	2530
10.5.4.10.1	Configuration Selection .....	2530
10.5.4.10.2	Read/Write .....	2530
10.5.4.10.3	RFBI State-Machine .....	2531
10.5.4.10.4	RFBI Timings .....	2531
10.5.4.11	Trigger Detection Logic .....	2533
10.5.4.11.1	Trigger Mode .....	2533
10.5.4.11.2	Internal Trigger Mode .....	2533
10.5.4.11.3	External Trigger Mode .....	2533
10.5.5	RFBI Programming Guide .....	2535
10.5.5.1	RFBI Low-Level Programming Models .....	2535
10.5.5.1.1	Global Initialization .....	2535
10.5.5.1.2	Operational Modes Configuration .....	2539
10.5.5.1.3	RFBI Events Servicing .....	2543
10.5.6	RFBI Hardware Status Features .....	2544
10.5.7	RFBI Register Manual .....	2545
10.5.7.1	RFBI Instance Summary .....	2545
10.5.7.2	RFBI Registers .....	2545
10.5.7.2.1	RFBI Register Summary .....	2545
10.5.7.2.2	RFBI Register Description .....	2546
10.6	Video Encoder .....	2558
10.6.1	Video Encoder Overview .....	2558
10.6.2	Video Encoder Environment .....	2560
10.6.3	Video Encoder Integration .....	2562
10.6.4	Video Encoder Functional Description .....	2565
10.6.4.1	Data Manager .....	2565
10.6.4.1.1	Color Space Converter .....	2565

10.6.4.1.2	Test Pattern Generation .....	2565
10.6.4.2	Luma Stage .....	2566
10.6.4.3	Chroma Stage .....	2566
10.6.4.4	Subcarrier and Burst Generation .....	2566
10.6.4.5	Vertical Blanking Interval .....	2567
10.6.4.5.1	Closed Caption Encoding .....	2567
10.6.4.5.2	Wide-Screen Signaling (WSS) Encoding .....	2569
10.6.4.6	Video DAC .....	2570
10.6.4.6.1	Video DC/AC Coupled TV Load .....	2571
10.6.4.6.2	TV Detection/Disconnection Pulse Generation and Use .....	2571
10.6.4.6.3	TV Short Detection .....	2574
10.6.4.6.4	Video DAC Normal Mode .....	2575
10.6.4.6.5	Video DAC Bypass Mode .....	2575
10.6.4.6.6	Video DAC Test Mode .....	2575
10.6.4.6.7	Video DAC Power Management .....	2576
10.6.5	Video Encoder Programming Guide .....	2577
10.6.5.1	Video Encoder Low-level Programming Models .....	2577
10.6.5.1.1	Surrounding Modules Global Initialization .....	2577
10.6.5.1.2	Video Encoder Global Initialization .....	2577
10.6.6	Video Encoder Use Case and Tips .....	2579
10.6.6.1	Video Encoder Register Settings .....	2579
10.6.7	Video Encoder Register Manual .....	2581
10.6.7.1	Video Encoder Instance Summary .....	2581
10.6.7.2	Video Encoder Registers .....	2581
10.6.7.2.1	Video Encoder Register Summary .....	2581
10.6.7.2.2	Video Encoder Register Description .....	2583
<b>11</b>	<b>2D/3D Graphics Accelerator .....</b>	<b>2609</b>
11.1	SGX Overview .....	2610
11.1.1	POWERVR SGX Main Features .....	2611
11.1.2	SGX 3D Features .....	2611
11.1.3	SGX Universal Scalable Shader Engine – Key Features .....	2612
11.2	SGX Integration .....	2613
11.3	SGX Functional Description .....	2615
11.3.1	SGX Block Diagram .....	2615
11.3.1.1	CGS .....	2615
11.3.1.2	Data Masters .....	2615
11.3.1.3	USSE .....	2616
11.3.1.4	Multilevel Cache .....	2616
11.3.1.5	Texturing Coprocessor .....	2616
11.3.1.6	Tiling Coprocessor .....	2616
11.3.1.7	Pixel Coprocessor .....	2616
11.3.2	SGX Clock Configuration .....	2616
11.3.3	SGX Software Reset .....	2617
11.3.4	SGX Power Management .....	2617
11.3.5	SGX Interrupt Requests .....	2617
11.4	SGX Register Manual .....	2618
11.4.1	SGX Instance Summary .....	2618
11.4.2	SGX OCP Registers .....	2618
11.4.2.1	SGX OCP Register Summary .....	2618
11.4.2.2	SGX OCP Register Description .....	2618
<b>12</b>	<b>Audio Back End .....</b>	<b>2635</b>
12.1	ABE Overview .....	2636
12.1.1	AE Subsystem .....	2637

12.1.2	Local Audio Interconnect .....	2638
12.1.3	GP Timers .....	2638
12.1.4	Watchdog Timer .....	2638
12.1.5	Digital Microphone Controller .....	2639
12.1.6	SLIMbus .....	2639
12.1.7	Multichannel Audio Serial Port .....	2640
12.1.8	Multichannel Buffered Serial Port .....	2640
12.1.9	Multichannel Pulse Density Modulation Module .....	2641
<b>13</b>	<b>Interconnect .....</b>	<b>2643</b>
13.1	Interconnect Overview .....	2644
13.1.1	Terminology .....	2644
13.1.2	Architecture Overview .....	2646
13.1.3	Module Distribution .....	2647
13.1.3.1	L3 Interconnect Module Distribution .....	2647
13.1.3.2	L4 Interconnect Module Distribution .....	2648
13.1.3.2.1	L4_PER Interconnect Agents .....	2648
13.1.3.2.2	L4_CFG Interconnect Agents .....	2649
13.1.3.2.3	L4_WKUP Interconnect Agents .....	2650
13.1.4	Connectivity Matrix .....	2651
13.2	L3 Interconnect .....	2652
13.2.1	L3 Interconnect Overview .....	2652
13.2.2	L3 Interconnect Integration .....	2654
13.2.3	L3 Interconnect Functional Description .....	2655
13.2.3.1	Module Use in L3 Interconnect .....	2655
13.2.3.2	Module Distribution .....	2655
13.2.3.2.1	L3 Interconnect Agents .....	2655
13.2.3.2.2	L3 Connectivity Matrix .....	2656
13.2.3.2.3	Master NIU Identification .....	2658
13.2.3.3	Bandwidth Regulators .....	2659
13.2.3.4	Flag Mux Generators .....	2659
13.2.3.5	Rate Adapter Generators .....	2660
13.2.3.6	Statistic Collectors Group .....	2660
13.2.3.7	L3 Protection and Firewalls .....	2660
13.2.3.7.1	L3 Firewall Reset .....	2661
13.2.3.7.2	Power Management .....	2662
13.2.3.7.3	L3 Firewall Functionality .....	2662
13.2.3.8	L3 Interconnect Error Handling .....	2670
13.2.3.8.1	Global Error-Routing Scheme .....	2670
13.2.3.8.2	Slave NIU Error Logging .....	2671
13.2.3.8.3	Flag Mux Error Logging .....	2673
13.2.3.8.4	Severity Level of Standard and Custom Errors .....	2673
13.2.4	L3 Interconnect Programming Guide .....	2674
13.2.4.1	L3 Interconnect Low-Level Programming Models .....	2674
13.2.4.1.1	Global Initialization .....	2674
13.2.4.2	Operational Modes Configuration .....	2674
13.2.4.2.1	L3 Interconnect Error Analysis Mode .....	2674
13.2.5	L3 Interconnect Register Manual .....	2678
13.2.5.1	L3 Register Group Summary .....	2678
13.2.5.1.1	L3 Firewall Registers Summary and Description .....	2678
13.2.5.1.2	L3 Host Register Summary and Description .....	2687
13.2.5.1.3	L3 TARG Register Summary and Description .....	2698
13.2.5.1.4	L3 PWR_DISC Register Summary and Description .....	2713
13.2.5.1.5	L3 FLAGMUX Register Summary and Description .....	2720

13.2.5.1.6	L3 BW Regulator Register Summary and Description .....	2724
13.2.5.1.7	L3 Rate Adapt Register Summary and Description .....	2728
13.2.5.1.8	L3 STATCOLL Register Summary and Description .....	2731
13.3	L4 Interconnects .....	2750
13.3.1	L4 Interconnect Overview .....	2750
13.3.2	L4 Interconnect Integration .....	2751
13.3.3	L4 Interconnect Functional Description .....	2752
13.3.3.1	Module Distribution .....	2752
13.3.3.1.1	L4_PER Interconnect Agents .....	2752
13.3.3.1.2	L4_CFG Interconnect Agents .....	2753
13.3.3.1.3	L4_WKUP Interconnect Agents .....	2754
13.3.3.2	Power Management .....	2754
13.3.3.3	L4 Firewalls .....	2754
13.3.3.3.1	Protection Group .....	2756
13.3.3.3.2	Segments and Regions .....	2759
13.3.3.3.3	L4 Firewall Address and Protection Register Settings .....	2764
13.3.3.4	L4 Error Detection and Reporting .....	2765
13.3.3.4.1	IA and TA Error Detection and Logging .....	2765
13.3.3.4.2	Time-Out .....	2766
13.3.3.4.3	Error Reporting .....	2767
13.3.3.4.4	Error Recovery .....	2768
13.3.3.4.5	Firewall Error Logging in the Control Module .....	2769
13.3.4	L4 Interconnect Programming Guide .....	2770
13.3.4.1	L4 Interconnect Low-level Programming Models .....	2770
13.3.4.1.1	Global Initialization .....	2770
13.3.4.1.2	Operational Modes Configuration .....	2770
13.3.5	L4 Interconnects Register Manual .....	2775
13.3.5.1	L4 Interconnects Instance Summary .....	2775
13.3.5.2	L4 Initiator Agent (L4 IA) .....	2777
13.3.5.2.1	L4 Initiator Agent (L4 IA) Register Summary .....	2777
13.3.5.2.2	L4 Initiator Agent (L4 IA) Register Description .....	2777
13.3.5.3	L4 Target Agent (L4 TA) .....	2783
13.3.5.3.1	L4 Target Agent (L4 TA) Register Summary .....	2783
13.3.5.3.2	L4 Target Agent (L4 TA) Register Description .....	2794
13.3.5.4	L4 Link Agent (L4 LA) .....	2797
13.3.5.4.1	L4 Link Agent (L4 LA) Register Summary .....	2797
13.3.5.4.2	L4 Link Agent (L4 LA) Register Description .....	2798
13.3.5.5	L4 Address Protection (L4 AP) .....	2803
13.3.5.5.1	L4 Address Protection (L4 AP) Register Summary .....	2803
13.3.5.5.2	L4 Address Protection (L4 AP) Register Description .....	2804
<b>14</b>	<b>Chip-to-Chip Interface (C2C) .....</b>	<b>2815</b>
14.1	C2C Overview .....	2816
14.2	C2C Integration .....	2817
14.3	C2C Power, Reset, and Clock Management .....	2818
14.3.1	PRCM Overview .....	2818
14.3.2	Reset Manager Functional Description .....	2818
14.3.2.1	Reset Sources .....	2818
14.3.2.1.1	Global Reset Sources .....	2818
14.3.2.2	Reset Domains .....	2818
14.3.3	Clock Manager Functional Description .....	2827
14.3.3.1	CD_C2C Clock Domain .....	2827
14.3.3.1.1	Overview .....	2827
14.3.3.1.2	Clock Domain Modes .....	2827

14.3.3.1.3	Clock Domain Dependency .....	2827
14.3.3.1.4	Clock Domain Module Attributes .....	2828
14.3.3.2	CD_CORTEXA9 Clock Domain .....	2829
14.3.3.2.1	Clock Domain Dependency .....	2829
14.3.3.3	CD_L4_CFG Clock Domain .....	2829
14.3.3.3.1	Clock Domain Dependency .....	2829
14.3.3.4	CD_L3_2 Clock Domain .....	2829
14.3.3.4.1	Clock Domain Dependency .....	2829
14.3.4	Power Management Functional Description .....	2830
14.3.4.1	PD_CORE Description .....	2830
14.3.5	PRCM Register Manual .....	2830
14.3.5.1	PRM Instance Summary .....	2830
14.3.5.2	CORE_PRM Registers .....	2831
14.3.5.2.1	CORE_PRM Register Summary .....	2831
14.3.5.2.2	CORE_PRM Register Description .....	2831
14.3.5.3	CORE_CM2 Registers .....	2832
14.3.5.3.1	CORE_CM2 Register Summary .....	2832
14.3.5.3.2	CORE_CM2 Register Description .....	2833
14.3.5.4	SCRM Instance Summary .....	2838
14.3.5.5	SCRM Registers .....	2838
14.3.5.5.1	SCRM Register Summary .....	2838
14.3.5.5.2	SCRM Register Description .....	2839
14.4	C2C L3 Interconnect .....	2844
14.4.1	C2C Interconnect Overview .....	2844
14.4.2	C2C Interconnect Integration .....	2846
14.4.2.1	Clocking, Reset, and Power-Management Scheme .....	2846
14.4.2.1.1	Clocks .....	2846
14.4.2.1.2	Resets .....	2846
14.4.2.1.3	Power Management .....	2846
14.4.3	C2C Interconnect Functional Description .....	2847
14.4.3.1	Slave NIU Configuration .....	2847
14.4.3.2	Global Error Routing Scheme .....	2847
14.5	C2C SSCM .....	2849
14.5.1	SSCM Overview .....	2849
14.5.2	SSCM Interrupts .....	2850
14.5.2.1	SSCM GPI Interrupt Mapping .....	2850
14.5.3	SSCM GPO .....	2850
14.6	Intersystem Communication Register Module .....	2851
14.6.1	intersystem Communication Register Overview .....	2851
14.6.2	ICR Functional Description .....	2851
14.6.2.1	ICR Clocking Scheme .....	2851
14.6.2.2	Flag Management .....	2851
14.6.2.3	Setting and Resetting a Flag .....	2852
14.6.2.4	Interrupt Management .....	2852
14.6.3	ICR Register Manual .....	2852
14.6.3.1	ICR Instance Summary .....	2852
14.6.3.2	ICR_MDM and ICR_MPU Registers .....	2853
14.6.3.2.1	ICR Register Summary .....	2853
14.6.3.2.2	ICR Register Description .....	2853
14.7	C2C Register Manual .....	2859
14.7.1	C2C Instance Summary .....	2859
14.7.2	C2C Registers .....	2859
14.7.2.1	C2C Register Summary .....	2859

14.7.2.2	C2C Register Description .....	2859
<b>15</b>	<b>Memory Subsystem .....</b>	<b>2889</b>
15.1	Memory Subsystem Overview .....	2890
15.1.1	DMM Overview .....	2890
15.1.1.1	DMM Features .....	2890
15.1.2	TILER Overview .....	2891
15.1.2.1	TILER Features .....	2891
15.1.3	EMIF Overview .....	2891
15.1.3.1	Main Features .....	2891
15.1.4	GPMC Overview .....	2892
15.1.4.1	GPMC Features .....	2892
15.1.5	ELM Overview .....	2893
15.1.5.1	ELM Features .....	2894
15.1.6	OCM Overview .....	2894
15.1.6.1	SAR ROM .....	2894
15.1.6.2	SAR RAM .....	2894
15.1.6.3	L3 OCM_RAM .....	2894
15.2	Dynamic Memory Manager .....	2895
15.2.1	DMM Overview .....	2895
15.2.2	DMM Integration .....	2896
15.2.2.1	DMM Configuration .....	2897
15.2.3	DMM Functional Description .....	2898
15.2.3.1	DMM Block Diagram .....	2898
15.2.3.2	DMM Clock Configuration .....	2899
15.2.3.3	DMM Power Management .....	2899
15.2.3.4	DMM Interrupt Requests .....	2900
15.2.3.5	DMM .....	2901
15.2.3.5.1	DMM Concepts .....	2901
15.2.3.5.2	DMM Transaction Flows .....	2906
15.2.3.5.3	DMM Internal Macro-Architecture .....	2907
15.2.3.6	TILER .....	2910
15.2.3.6.1	TILER Concepts .....	2910
15.2.3.6.2	TILER Macro-Architecture .....	2926
15.2.3.6.3	TILER Guidelines for Initiators .....	2929
15.2.4	DMM Use Cases and Tips .....	2931
15.2.4.1	PAT Use Cases .....	2931
15.2.4.1.1	Simple Manual Area Refill .....	2931
15.2.4.1.2	Single Auto-Configured Area Refill .....	2931
15.2.4.1.3	Chained Auto-Configured Area Refill .....	2932
15.2.4.1.4	Synchronized Auto-Configured Area Refill .....	2933
15.2.4.1.5	Cyclic Synchronized Auto-Configured Area Refill .....	2934
15.2.4.2	Addressing Management with LISA .....	2935
15.2.4.2.1	Case 1: Use of Only One Memory Controller .....	2935
15.2.4.2.2	Case 2: Use of the 2 Memory Controllers .....	2936
15.2.5	DMM Basic Programming Model .....	2939
15.2.5.1	Global Initialization .....	2939
15.2.5.2	DMM Module Global Initialization .....	2939
15.2.5.3	DMM Operational Modes Configuration .....	2939
15.2.5.3.1	Different Operational Modes .....	2939
15.2.5.3.2	Configuration Settings and LUT Refill .....	2939
15.2.5.3.3	Interleaving Settings .....	2940
15.2.5.3.4	Aliased Tiled View Orientation Settings and LUT Refill .....	2940
15.2.5.3.5	Priority Settings .....	2940



15.2.5.3.6	Error Handling .....	2941
15.2.5.3.7	PAT Programming Model .....	2941
15.2.5.4	Addressing an Object in Tiled Mode .....	2941
15.2.5.4.1	Frame Buffer Addressing .....	2942
15.2.5.4.2	TILER Page Mapping .....	2943
15.2.5.5	Addressing an Object in Page Mode .....	2943
15.2.5.6	Sharing Containers Between Different Modes .....	2944
15.2.6	DMM Register Manual .....	2945
15.2.6.1	DMM Instance Summary .....	2945
15.2.6.2	DMM Registers .....	2945
15.2.6.2.1	DMM Register Summary .....	2945
15.2.6.2.2	DMM Register Description .....	2946
15.3	EMIF Controller .....	2977
15.3.1	EMIF Module Overview .....	2977
15.3.1.1	EMIF Module Main Features .....	2977
15.3.1.2	Not supported Features .....	2978
15.3.1.3	EMIF Overall Architecture .....	2978
15.3.2	EMIF Environment .....	2978
15.3.2.1	EMIF Signals .....	2979
15.3.3	EMIF Integration .....	2981
15.3.4	EMIF Functional Description .....	2983
15.3.4.1	Block Diagram .....	2983
15.3.4.1.1	L3 Interface .....	2983
15.3.4.1.2	FIFOs Description .....	2983
15.3.4.1.3	Arbitration .....	2985
15.3.4.2	Clock Management .....	2986
15.3.4.3	Software Reset .....	2987
15.3.4.4	System Power Management .....	2987
15.3.4.4.1	Read Idle Window .....	2987
15.3.4.4.2	LPDDR2-SDRAM Power-Down Mode .....	2987
15.3.4.4.3	LPDDR2-SDRAM Deep Power-Down Mode .....	2987
15.3.4.4.4	LPDDR2-NVM Power-Down Mode .....	2988
15.3.4.4.5	LPDDR2-SDRAM Self-Refresh Mode .....	2988
15.3.4.4.6	Save and Restore .....	2989
15.3.4.4.7	Retention Mode .....	2989
15.3.4.5	Interrupt Requests .....	2989
15.3.4.6	SDRAM Output Impedance Calibration .....	2990
15.3.4.7	SDRAM Temperature Monitoring .....	2990
15.3.4.8	SDRAM Refresh Scheduling .....	2990
15.3.4.9	EMIF Access Cycles .....	2991
15.3.4.9.1	SDRAM Read .....	2992
15.3.4.9.2	SDRAM Write .....	2992
15.3.4.10	Turn Around Time .....	2992
15.3.4.11	SDRAM Address Mapping .....	2993
15.3.4.12	NVM Address Mapping .....	3000
15.3.4.13	LPDDR2 SDRAM Initialization .....	3000
15.3.5	External Memory Interface (EMIF) Programming Guide .....	3000
15.3.5.1	EMIF Low-level Programming Models .....	3000
15.3.5.1.1	Global Initialization .....	3000
15.3.5.1.2	Operational Modes Configuration .....	3003
15.3.5.1.3	EMIF Events Servicing .....	3007
15.3.6	EMIF Register Manual .....	3009
15.3.6.1	EMIF Instance Summary .....	3009

15.3.6.2	EMIF Registers .....	3009
15.3.6.2.1	EMIF Register Summary .....	3009
15.3.6.2.2	EMIF Register Description .....	3010
15.4	General-Purpose Memory Controller .....	3044
15.4.1	GPMC Overview .....	3044
15.4.2	GPMC Environment .....	3044
15.4.2.1	GPMC Modes .....	3044
15.4.2.2	GPMC Signals .....	3046
15.4.3	GPMC Integration .....	3049
15.4.4	GPMC Functional Description .....	3051
15.4.4.1	GPMC Block Diagram .....	3051
15.4.4.2	GPMC Clock Configuration .....	3052
15.4.4.3	GPMC Software Reset .....	3053
15.4.4.4	GPMC Power Management .....	3053
15.4.4.5	GPMC Interrupt Requests .....	3054
15.4.4.6	GPMC DMA Requests .....	3054
15.4.4.7	L3 Interconnect Interface .....	3054
15.4.4.8	GPMC Address and Data Bus .....	3055
15.4.4.8.1	GPMC I/O Configuration Setting .....	3055
15.4.4.8.2	GPMC CS0 Default Configuration at IC Reset .....	3056
15.4.4.9	Address Decoder and Chip-Select Configuration .....	3057
15.4.4.9.1	Chip-Select Base Address and Region Size .....	3057
15.4.4.9.2	Access Protocol .....	3059
15.4.4.9.3	External Signals .....	3059
15.4.4.9.4	Error Handling .....	3068
15.4.4.10	Timing Setting .....	3069
15.4.4.10.1	Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME) .....	3069
15.4.4.10.2	nCS: Chip-Select Signal Control Assertion/Deassertion Time (CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADELAY) .....	3069
15.4.4.10.3	nADV/ALE: Address Valid/Address Latch Enable Signal Control Assertion/Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME / ADVEXTRADELAY/ADVAADMUXONTIME/ADVAADMUXRDOFFTIME/ADVAADMUXWROFFTIME) .....	3070
15.4.4.10.4	nOE/nRE: Output Enable / Read Enable Signal Control Assertion / Deassertion Time (OEONTIME / OEOFFTIME / OEEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME) .....	3071
15.4.4.10.5	nWE: Write Enable Signal Control Assertion / Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY) .....	3071
15.4.4.10.6	GPMC_CLK .....	3071
15.4.4.10.7	GPMC_CLK and Control Signals Setup and Hold .....	3072
15.4.4.10.8	Access Time (RDACCESSTIME / WRACCESSTIME) .....	3072
15.4.4.10.9	Page Burst Access Time (PAGEBURSTACCESSTIME) .....	3073
15.4.4.10.10	Bus Keeping Support .....	3073
15.4.4.11	NOR Access Description .....	3074
15.4.4.11.1	Asynchronous Access Description .....	3074
15.4.4.11.2	Synchronous Access Description .....	3081
15.4.4.11.3	Asynchronous and Synchronous Accesses in Nonmultiplexed Mode .....	3090
15.4.4.11.4	Page and Burst Support .....	3094
15.4.4.11.5	System Burst Versus External Device Burst Support .....	3094
15.4.4.12	pSRAM Access Specificities .....	3095
15.4.4.13	NAND Access Description .....	3095
15.4.4.13.1	NAND Memory Device in Byte or 16-bit Word Stream Mode .....	3095
15.4.4.13.2	NAND Device-Ready Pin .....	3101
15.4.4.13.3	ECC Calculator .....	3102
15.4.4.13.4	Prefetch and Write-Posting Engine .....	3119

15.4.5	GPMC Basic Programming Model .....	3126
15.4.5.1	GPMC High Level Programming Model Overview .....	3126
15.4.5.2	GPMC Initialization .....	3128
15.4.5.3	GPMC Configuration in NOR Mode .....	3129
15.4.5.4	GPMC Configuration in NAND mode .....	3130
15.4.5.5	Set Memory Access .....	3132
15.4.5.6	GPMC Timing Parameters .....	3133
15.4.5.6.1	GPMC Timing Parameters Formulas .....	3136
15.4.6	GPMC Use Cases And Tips .....	3144
15.4.6.1	How to Set GPMC Timing Parameters for Typical Accesses .....	3144
15.4.6.1.1	External Memory Attached to the GPMC Module .....	3144
15.4.6.1.2	Typical GPMC Setup .....	3144
15.4.6.2	How to choose a suitable memory to use with the GPMC .....	3150
15.4.6.2.1	Supported Memories or Devices .....	3150
15.4.6.2.2	GPMC Features and Settings .....	3152
15.4.7	GPMC Register Manual .....	3153
15.4.7.1	GPMC Register Summary .....	3153
15.4.7.2	GPMC Register Descriptions .....	3154
15.5	Error Location Module .....	3187
15.5.1	Error Location Module Overview .....	3187
15.5.1.1	ELM Features .....	3187
15.5.2	ELM Integration .....	3188
15.5.3	ELM Functional Description .....	3189
15.5.3.1	ELM Software Reset .....	3189
15.5.3.2	ELM Power Management .....	3189
15.5.3.3	ELM Interrupt Requests .....	3190
15.5.3.4	Processing Initialization .....	3190
15.5.3.5	Processing Sequence .....	3191
15.5.3.6	Processing Completion .....	3191
15.5.4	ELM Basic Programming Model .....	3192
15.5.4.1	ELM Low Level Programming Model .....	3192
15.5.4.1.1	Processing Initialization .....	3192
15.5.4.1.2	Read Results .....	3193
15.5.4.2	Use Case: ELM Used in Continuous Mode .....	3194
15.5.4.3	Use Case: ELM Used in Page Mode .....	3195
15.5.5	ELM Register Manual .....	3197
15.5.5.1	ELM Instance Summary .....	3197
15.5.5.2	ELM Registers .....	3197
15.5.5.2.1	ELM Register Summary .....	3197
15.5.5.2.2	ELM Register Description .....	3198
15.6	On-Chip Memory (OCM) Subsystem .....	3216
15.6.1	OCM Subsystem Overview .....	3216
15.6.2	OCM Subsystem Integration .....	3216
15.6.3	OCM Subsystem Functional Description .....	3217
15.6.3.1	SAR ROM .....	3217
15.6.3.2	SAR RAM .....	3218
15.6.3.3	L3 OCM_RAM .....	3218
<b>16</b>	<b>sDMA .....</b>	<b>3219</b>
16.1	sDMA Module Overview .....	3220
16.2	sDMA Controller Environment .....	3222
16.3	sDMA Module Integration .....	3224
16.3.1	DMA Requests to the sDMA Controller .....	3226
16.4	sDMA Functional Description .....	3229

16.4.1	sDMA Controller Power Management .....	3229
16.4.2	sDMA Controller Interrupt Requests .....	3230
16.4.2.1	Interrupt Generation .....	3231
16.4.3	Logical Channel Transfer Overview .....	3231
16.4.4	FIFO Queue Memory Pool .....	3232
16.4.5	Addressing Modes .....	3233
16.4.6	Packed Accesses .....	3237
16.4.7	Burst Transactions .....	3238
16.4.8	Endianism Conversion .....	3238
16.4.9	Transfer Synchronization .....	3238
16.4.9.1	Software Synchronization .....	3238
16.4.9.2	Hardware Synchronization .....	3238
16.4.10	Thread Budget Allocation .....	3240
16.4.11	FIFO Budget Allocation .....	3241
16.4.12	Chained Logical Channel Transfers .....	3242
16.4.13	Reprogramming an Active Channel .....	3242
16.4.14	Packet Synchronization .....	3242
16.4.15	Graphics Acceleration Support .....	3243
16.4.16	Supervisor Modes .....	3244
16.4.17	Posted and Nonposted Writes .....	3244
16.4.18	Disabling a Channel During Transfer .....	3244
16.4.19	FIFO Draining Mechanism .....	3244
16.4.20	Linked List .....	3245
16.4.20.1	Overview .....	3245
16.4.20.2	Link-List Transfer Profile .....	3245
16.4.20.3	Descriptors .....	3245
16.4.20.3.1	Type 1 .....	3246
16.4.20.3.2	Type 2 .....	3247
16.4.20.3.3	Type 3 .....	3248
16.4.20.4	Linked-List Control and Monitoring .....	3248
16.4.20.4.1	Transfer Mode Setting .....	3248
16.4.20.4.2	Starting a Linked List .....	3249
16.4.20.4.3	Monitoring a Linked-List Progression .....	3249
16.4.20.4.4	Interrupt During Linked-List Execution .....	3249
16.4.20.4.5	Pause a Linked List .....	3249
16.4.20.4.6	Stop a Linked List (Abort or Drain) .....	3249
16.4.20.4.7	Status Bit Behavior .....	3250
16.4.20.4.8	Linked-List Channel Linking .....	3250
16.4.21	Auto-Restore Feature .....	3251
16.4.21.1	Save Process .....	3252
16.4.21.2	Restore Process .....	3253
16.4.21.2.1	PRCM and sDMA interaction During Restore Phase .....	3253
16.4.21.2.2	sDMA Descriptors Description .....	3253
16.4.21.3	SAR RAM .....	3254
16.4.21.4	SAR ROM .....	3254
16.4.21.5	SAR RAM Memory Allocation Reserved for Save and Restore .....	3254
16.4.21.6	Register Mapping in SAR RAM .....	3255
16.5	sDMA Basic Programming Model .....	3262
16.5.1	Setup Configuration .....	3262
16.5.2	Software-Triggered (Nonsynchronized) Transfer .....	3262
16.5.3	Hardware-Synchronized Transfer .....	3264
16.5.4	Synchronized Transfer Monitoring Using CDAC .....	3266
16.5.5	Concurrent Software and Hardware Synchronization .....	3266

16.5.6	Chained Transfer .....	3266
16.5.7	90-Degree Clockwise Image Rotation .....	3267
16.5.8	Graphic Operations .....	3268
16.5.9	Linked-List Programming Guidelines .....	3268
16.6	sDMA Register Manual .....	3269
16.6.1	sDMA Instance Summary .....	3269
16.6.2	sDMA Registers .....	3269
16.6.2.1	sDMA Register Summary .....	3269
16.6.2.2	sDMA Register Description .....	3270
<b>17</b>	<b>Interrupt Controllers .....</b>	<b>3299</b>
17.1	Interrupt Controllers Overview .....	3300
17.2	Interrupt Controllers Environment .....	3302
17.3	Interrupt Controllers Integration .....	3303
17.3.1	Interrupts Mapping .....	3303
17.3.2	Interrupt Requests to Cortex-A9 MPU INTC .....	3303
17.3.3	Interrupt Requests to Cortex-M3 MPU INTC .....	3306
17.4	Interrupt Controllers Functional Description .....	3309
17.4.1	Cortex-A9 MPU INTC Functional Description .....	3309
17.4.2	Cortex-M3 MPU INTC Functional Description .....	3309
17.5	Interrupt Controllers Register Manual .....	3310
17.5.1	Interrupt Controllers Instance Summary .....	3310
17.5.2	Cortex-A9 MPU INTC Registers .....	3310
17.5.3	Cortex-M3 MPU INTC Registers .....	3310
<b>18</b>	<b>Control Module .....</b>	<b>3311</b>
18.1	Control Module Overview .....	3312
18.2	Control Module Environment .....	3314
18.2.1	Control Module Signals .....	3314
18.3	Control Module Integration .....	3315
18.4	Control Module Functional Description .....	3318
18.4.1	Control Module Block Diagram .....	3318
18.4.2	Control Module Clock Configuration .....	3320
18.4.3	Control Module Software Reset .....	3321
18.4.4	Control Module Power Management .....	3321
18.4.4.1	Retention .....	3321
18.4.4.2	Power Management Protocols .....	3321
18.4.4.3	Save and Restore (SAR) Mechanism .....	3321
18.4.5	Hardware Requests .....	3321
18.4.6	Control Module Initialization .....	3321
18.4.7	Control Module Instances .....	3322
18.4.7.1	General Core Control Module Instance .....	3322
18.4.7.2	General Wakeup Control Module Instance .....	3322
18.4.7.3	Device Core Control Module Instance .....	3322
18.4.7.4	Device Wakeup Control Module Instance .....	3322
18.4.8	PAD Functional Multiplexing and Configuration .....	3323
18.4.8.1	Mode Selection .....	3326
18.4.8.2	Pull Selection .....	3327
18.4.8.3	Pad Multiplexing Register Fields .....	3327
18.4.8.4	Off Mode PAD Control Overview .....	3346
18.4.8.4.1	Wake-Up Event Detection .....	3346
18.4.8.4.2	I/O Power Optimization .....	3348
18.4.9	Extended-Drain I/O and PBIAS Cell .....	3349
18.4.9.1	PBIAS Cell .....	3352
18.4.9.2	Extended-Drain I/O .....	3352

18.4.9.3	PBIAS Error Generation .....	3356
18.4.9.4	Critical Timing Requirements .....	3357
18.4.10	Band Gap Voltage and Temperature Sensor .....	3357
18.4.10.1	Band Gap Voltage Reference .....	3358
18.4.10.2	Temperature Sensor .....	3359
18.4.10.2.1	Single Conversion Mode (CONTCONV = 0) .....	3359
18.4.10.2.2	Continuous Conversion Mode (CONTCONV = 1) .....	3359
18.4.10.2.3	ADC Codes Versus Temperature .....	3360
18.4.11	Hardware Observability .....	3361
18.4.11.1	Device Internal Signals Observability Overview .....	3361
18.4.11.2	Observability Gating Capabilities .....	3362
18.4.11.3	Observability Signals Multiplexing at General Core/Wake-Up Control Level .....	3363
18.4.11.4	Observability Signals Multiplexing at Device Core Control Level .....	3371
18.4.11.5	Observability Signals Sourced at Module Level .....	3371
18.4.11.5.1	Dual Cortex-A9 Subsystem Observable Signals .....	3371
18.4.11.5.2	IVAHD Subsystem Observable Signals .....	3377
18.4.11.5.3	Display Subsystem Observable Signals .....	3378
18.4.11.5.4	DSP Subsystem Observable Signals .....	3378
18.4.11.5.5	ABE Subsystem Observable Signals .....	3378
18.4.11.5.6	USBOTGHS Subsystem Observable Signals .....	3379
18.4.11.5.7	CM2 Subsystem Observable Signals .....	3379
18.4.11.5.8	CM1 Subsystem Observable Signals .....	3386
18.4.11.5.9	PRM Subsystem Observable Signals .....	3389
18.4.11.5.10	DPLL Clockview Channel Observable Signals Multiplexing .....	3393
18.4.11.5.11	Other CORE DPLL observable signals .....	3393
18.4.11.5.12	Other ABE DPLL observable signals .....	3393
18.4.11.5.13	Other PERIPH DPLL observable signals .....	3394
18.4.11.5.14	Other DSI1 DPLL observable signals .....	3394
18.4.11.5.15	Other DSI2 DPLL observable signals .....	3395
18.4.11.5.16	Other USB DPLL observable signals .....	3395
18.4.12	Functional Register Description .....	3396
18.4.12.1	DSP Boot Register .....	3396
18.4.12.2	PBIASLITE Control Register .....	3396
18.4.12.3	Temperature Sensor Control Register .....	3396
18.4.12.4	CSI Receiver Control Register .....	3396
18.4.12.5	Protection Status Registers .....	3397
18.4.12.6	Signal Integrity Parameter Control Registers with Pad Group Assignment .....	3398
18.4.12.6.1	Signal Integrity Parameter Controls Overview .....	3398
18.4.12.6.2	High-speed I/O Buffers with Impedance, Slew Rate and Weak Driver Settings .....	3399
18.4.12.6.3	Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings .....	3402
18.4.12.6.4	I/Os with Combined Mode and Load Settings .....	3404
18.4.12.6.5	50-Ω Output Buffer I/Os with Combined Mode and Load Settings .....	3405
18.4.12.6.6	I/O cells with Configurable Output Driver Impedance .....	3405
18.4.12.6.7	C2CI/O cells with load setting adjustable according to interface requirement / voltage mode .....	3406
18.4.12.6.8	SDMMC1 Pull-up Strength Control .....	3407
18.4.12.6.9	I2Cx I/Os Group Pullupresx Controls and Load Range Settings .....	3408
18.4.12.6.10	Device Interfaces Signal Group Controls Mapping .....	3408
18.4.12.7	Dual Voltage-Supplied Peripheral Controls .....	3424
18.5	Control Module Programming Guide .....	3425
18.5.1	Control Module Low-Level Programming Models .....	3425
18.5.1.1	Global Initialization .....	3425
18.5.1.1.1	Surrounding Modules Global Initialization .....	3425
18.5.1.1.2	Control Module Global Initialization .....	3425

18.5.1.2	Operational Modes Configuration .....	3425
18.5.1.2.1	Video Driver Configuration .....	3425
18.5.1.2.2	Extended-Drain I/Os and PBIAS Cells Programming Guide .....	3426
18.5.1.2.3	Hardware Observability Settings .....	3429
18.5.1.2.4	Off Mode Preliminary Settings .....	3430
18.5.1.2.5	Pad Configuration Programming Points .....	3430
18.6	Control Module Register Manual .....	3432
18.6.1	Control Module Instance Summary .....	3432
18.6.2	SYSCTRL_GENERAL_CORE Register Summary .....	3432
18.6.3	SYSCTRL_GENERAL_CORE Register Description .....	3435
18.6.4	SYSCTRL_GENERAL_WKUP Register Summary .....	3509
18.6.5	SYSCTRL_GENERAL_WKUP Register Description .....	3509
18.6.6	SYSCTRL_PADCONF_CORE Register Summary .....	3512
18.6.7	SYSCTRL_PADCONF_CORE Register Description .....	3518
18.6.8	SYSCTRL_PADCONF_WKUP Register Summary .....	3833
18.6.9	SYSCTRL_PADCONF_WKUP Register Description .....	3834
<b>19</b>	<b>Mailbox .....</b>	<b>3875</b>
19.1	Mailbox Overview .....	3876
19.2	Mailbox Integration .....	3878
19.2.1	System Mailbox Integration .....	3878
19.2.2	IVA-HD Mailbox Integration .....	3879
19.3	Mailbox Functional Description .....	3882
19.3.1	Mailbox Block Diagram .....	3883
19.3.2	Mailbox Software Reset .....	3883
19.3.3	Mailbox Power Management .....	3883
19.3.4	Mailbox Interrupt Requests .....	3884
19.3.5	Mailbox Assignment .....	3885
19.3.5.1	Description .....	3885
19.3.6	Sending and Receiving Messages .....	3886
19.3.6.1	Description .....	3886
19.3.7	16-Bit Register Access .....	3886
19.3.7.1	Description .....	3886
19.3.8	Example of Communication .....	3887
19.4	Mailbox Programming Guide .....	3888
19.4.1	Mailbox Low-level Programming Models .....	3888
19.4.1.1	Global Initialization .....	3888
19.4.1.1.1	Surrounding Modules Global Initialization .....	3888
19.4.1.1.2	Mailbox Global Initialization .....	3888
19.4.1.2	Mailbox Operational Modes Configuration .....	3889
19.4.1.2.1	Mailbox Processing modes .....	3889
19.4.1.3	Mailbox Events Servicing .....	3890
19.4.1.3.1	Events Servicing in Sending Mode .....	3890
19.4.1.3.2	Events Servicing in Receiving Mode .....	3890
19.5	Mailbox Register Manual .....	3891
19.5.1	Mailbox Instance Summary .....	3891
19.5.2	Mailbox Registers .....	3891
19.5.2.1	Mailbox Register Summary .....	3891
19.5.2.2	Mailbox Register Description .....	3892
<b>20</b>	<b>Memory Management Units .....</b>	<b>3907</b>
20.1	MMU Overview .....	3908
20.2	MMU Integration .....	3909
20.3	MMU Functional Description .....	3912
20.3.1	MMU Block Diagram .....	3912

20.3.1.1	MMU Address Translation Process .....	3912
20.3.1.2	Translation Tables .....	3913
20.3.1.2.1	Translation Table Hierarchy .....	3913
20.3.1.2.2	First-Level Translation Table .....	3914
20.3.1.2.3	Two-Level Translation .....	3917
20.3.1.3	Translation Lookaside Buffer .....	3920
20.3.1.3.1	TLB Entry Format .....	3921
20.3.2	MMU Clock Configuration .....	3922
20.3.3	MMU Software Reset .....	3922
20.3.4	MMU Power Management .....	3922
20.3.5	MMU Interrupt Requests .....	3923
20.3.6	MMU Error Handling .....	3923
20.4	MMU Low-level Programming Models .....	3925
20.4.1	Global Initialization .....	3925
20.4.1.1	Surrounding Modules Global Initialization .....	3925
20.4.1.2	MMU Global Initialization .....	3925
20.4.1.2.1	Main Sequence—MMU Global Initialization .....	3925
20.4.1.2.2	Subsequence—Configure a TLB entry .....	3926
20.4.1.3	Operational Modes Configuration .....	3926
20.4.1.3.1	Main Sequence—Writing TLB Entries Statically .....	3926
20.4.1.3.2	Main Sequence—Protecting TLB Entries .....	3927
20.4.1.3.3	Main Sequence—Deleting TLB Entries .....	3927
20.4.1.3.4	Main Sequence—Read TLB Entries .....	3927
20.5	MMU Register Manual .....	3928
20.5.1	MMU Instance Summary .....	3928
20.5.2	MMU Registers .....	3928
20.5.2.1	CORTEXM3_L2MMU and DSP MMU Register Summary .....	3928
20.5.2.2	MMU Register Description .....	3928
<b>21</b>	<b>Spinlock .....</b>	<b>3945</b>
21.1	Spinlock Overview .....	3946
21.2	Spinlock Integration .....	3947
21.3	Spinlock Functional Description .....	3948
21.3.1	Spinlock Software Reset .....	3948
21.3.2	Spinlock Power Management .....	3948
21.3.3	About Spinlocks .....	3949
21.3.4	Spinlock Functional Operation .....	3949
21.4	Spinlock Programming Guide .....	3951
21.4.1	Spinlock Low-level Programming Models .....	3951
21.4.1.1	Surrounding Modules Global Initialization .....	3951
21.4.1.2	Basic Spinlock Operations .....	3951
21.4.1.2.1	Spinlocks Clearing After a System Bug Recovery .....	3951
21.4.1.2.2	Take and Release Spinlock .....	3951
21.5	Spinlock Register Manual .....	3954
21.5.1	Spinlock Instance Summary .....	3954
21.5.2	Spinlock Registers .....	3954
21.5.2.1	Spinlock Register Summary .....	3954
21.5.2.2	Spinlock Register Description .....	3954
<b>22</b>	<b>Timers .....</b>	<b>3959</b>
22.1	Timers Overview .....	3960
22.2	General-Purpose Timers .....	3961
22.2.1	General-Purpose Timers Overview .....	3961
22.2.1.1	GP Timer Features .....	3961
22.2.2	GP Timer Environment .....	3963



22.2.2.1	GP Timer External System Interface .....	3963
22.2.3	GP Timer Integration .....	3965
22.2.4	GP Timer Functional Description .....	3968
22.2.4.1	GP Timer Block Diagram .....	3968
22.2.4.2	GPTIMER1, GPTIMER2, and GPTIMER10 Power Management .....	3971
22.2.4.2.1	Wake-Up Capability .....	3972
22.2.4.3	Power Management of Other GP Timers .....	3973
22.2.4.3.1	Wake-Up Capability .....	3973
22.2.4.4	Software Reset .....	3974
22.2.4.5	GP Timer Interrupts .....	3974
22.2.4.6	Timer Mode Functionality .....	3974
22.2.4.6.1	1-ms Tick Generation (Only GPTIMER1, GPTIMER2, and GPTIMER10) .....	3975
22.2.4.7	Capture Mode Functionality .....	3977
22.2.4.8	Compare Mode Functionality .....	3978
22.2.4.9	Prescaler Functionality .....	3979
22.2.4.10	Pulse-Width Modulation .....	3979
22.2.4.11	Timer Counting Rate .....	3980
22.2.4.12	Timer Under Emulation .....	3981
22.2.4.13	Accessing GP Timer Registers .....	3981
22.2.4.13.1	Writing to Timer Registers .....	3982
22.2.4.13.2	Reading From Timer Counter Registers .....	3984
22.2.5	GP Timer Low-Level Programming Models .....	3984
22.2.5.1	Global Initialization .....	3984
22.2.5.1.1	Global Initialization of Surrounding Modules .....	3984
22.2.5.1.2	GP Timer Module Global Initialization .....	3984
22.2.5.2	Operational Mode Configuration .....	3985
22.2.5.2.1	GP Timer Mode .....	3985
22.2.5.2.2	GP Timer Compare Mode .....	3985
22.2.5.2.3	GP Timer Capture Mode .....	3986
22.2.5.2.4	GP Timer PWM Mode .....	3986
22.2.6	GP Timer Register Manual .....	3988
22.2.6.1	GP Timer Instance Summary .....	3988
22.2.6.2	GP Timer Registers .....	3988
22.2.6.2.1	GP Timer Register Summary .....	3988
22.2.6.2.2	GP Timer Register Description .....	3992
22.2.6.2.3	GPTIMER1, GPTIMER2 , and GPTIMER10 Register Description .....	4005
22.3	Watchdog Timers .....	4011
22.3.1	Watchdog Timers Overview .....	4011
22.3.1.1	Watchdog Timers Features .....	4012
22.3.2	Watchdog Timer Environment .....	4012
22.3.3	Watchdog Timer Integration .....	4013
22.3.4	Watchdog Timers Functional Description .....	4015
22.3.4.1	Power Management .....	4015
22.3.4.1.1	Wake-Up Capability .....	4015
22.3.4.2	Interrupts .....	4015
22.3.4.3	General Watchdog Timer Operation .....	4016
22.3.4.4	Reset Context .....	4016
22.3.4.5	Overflow/Reset Generation .....	4017
22.3.4.6	Prescaler Value/Timer Reset Frequency .....	4017
22.3.4.7	Triggering a Timer Reload .....	4018
22.3.4.8	Start/Stop Sequence for Watchdog Timers (Using the WDT_WSPR Register) .....	4019
22.3.4.9	Modifying Timer Count/Load Values and Prescaler Setting .....	4019
22.3.4.10	Watchdog Counter Register Access Restriction (WDT_WCRR Register) .....	4019

22.3.4.11	Watchdog Timer Interrupt Generation .....	4019
22.3.4.12	Watchdog Timers Under Emulation .....	4020
22.3.4.13	Accessing Watchdog Timer Registers .....	4020
22.3.5	Watchdog Timer Low-Level Programming Model .....	4021
22.3.5.1	Global Initialization .....	4021
22.3.5.1.1	Surrounding Modules Global Initialization .....	4021
22.3.5.1.2	Watchdog Timer Module Global Initialization .....	4022
22.3.5.2	Operational Mode Configuration .....	4022
22.3.5.2.1	Watchdog Timer Basic Configuration .....	4022
22.3.6	Watchdog Timer Register Manual .....	4023
22.3.6.1	Watchdog Timer Instance Summary .....	4023
22.3.6.2	Watchdog Timer Registers .....	4023
22.3.6.2.1	Watchdog Timer Register Summary .....	4023
22.3.6.2.2	Watchdog Timer Register Description .....	4025
22.4	32-kHz Synchronized Timer .....	4037
22.4.1	32-kHz Synchronized Timer Overview .....	4037
22.4.1.1	32-kHz Synchronized Timer Features .....	4037
22.4.2	32-kHz Synchronized Timer Environment .....	4037
22.4.3	32-kHz Synchronized Timer Integration .....	4037
22.4.4	32-kHz Synchronized Timer Functional Description .....	4038
22.4.4.1	Reading the 32-kHz Synchronized Timer .....	4039
22.4.5	32KTimer Register Manual .....	4040
22.4.5.1	32KTimer Register Mapping Summary .....	4040
22.4.5.2	32KTimer Register Description .....	4040
<b>23</b>	<b>Serial Communication Interface .....</b>	<b>4043</b>
23.1	Multimaster High-Speed I <sup>2</sup> C Controller .....	4044
23.1.1	HS I <sup>2</sup> C Overview .....	4044
23.1.2	HS I <sup>2</sup> C Environment .....	4047
23.1.2.1	HS I <sup>2</sup> C in I <sup>2</sup> C Mode .....	4047
23.1.2.1.1	HS I <sup>2</sup> C Pins for Typical Connections in I <sup>2</sup> C Mode .....	4047
23.1.2.1.2	HS I <sup>2</sup> C Interface Typical Connections .....	4047
23.1.2.1.3	HS I <sup>2</sup> C Typical Connection Protocol and Data Format .....	4048
23.1.2.2	HS I <sup>2</sup> C in SCCB Mode .....	4052
23.1.2.2.1	HS I <sup>2</sup> C Pins for Typical Connections in SCCB Mode .....	4052
23.1.2.2.2	HS I <sup>2</sup> C SCCB Interface Typical Connections .....	4053
23.1.2.2.3	HS I <sup>2</sup> C SCCB Typical Connection Protocol and Data Format .....	4054
23.1.2.3	HS I <sup>2</sup> C for Communication With Power Chip(s) .....	4055
23.1.2.3.1	HS I <sup>2</sup> C I2C5 Pins for Typical Connections .....	4056
23.1.2.3.2	HS I <sup>2</sup> C I2C5 Interface Typical Connections .....	4056
23.1.2.3.3	HS I <sup>2</sup> C Typical Connections Protocol and Data Format for I2C5 .....	4057
23.1.3	HS I <sup>2</sup> C Integration .....	4060
23.1.4	HS I <sup>2</sup> C Functional Description .....	4063
23.1.4.1	HS I <sup>2</sup> C Block Diagram .....	4063
23.1.4.2	HS I <sup>2</sup> C Clocks .....	4064
23.1.4.2.1	HS I <sup>2</sup> C Clocking .....	4064
23.1.4.2.2	HS I <sup>2</sup> C Automatic Blocking of the I <sup>2</sup> C Clock Feature (I <sup>2</sup> C Mode Only) .....	4068
23.1.4.3	HS I <sup>2</sup> C Software Reset .....	4068
23.1.4.4	HS I <sup>2</sup> C Power Management .....	4068
23.1.4.5	HS I <sup>2</sup> C Interrupt Requests .....	4069
23.1.4.6	HS I <sup>2</sup> C DMA Requests .....	4070
23.1.4.7	HS I <sup>2</sup> C Programmable Multislave Channel Feature (I <sup>2</sup> C Mode Only) .....	4071
23.1.4.8	HS I <sup>2</sup> C FIFO Management .....	4071
23.1.4.8.1	HS I <sup>2</sup> C FIFO Interrupt Mode .....	4071

23.1.4.8.2	HS I <sup>2</sup> C FIFO Polling Mode .....	4073
23.1.4.8.3	HS I <sup>2</sup> C FIFO DMA Mode (I <sup>2</sup> C Mode Only) .....	4073
23.1.4.8.4	HS I <sup>2</sup> C Draining Feature (I <sup>2</sup> C Mode Only) .....	4074
23.1.4.9	HS I <sup>2</sup> C Noise Filter .....	4075
23.1.4.10	HS I <sup>2</sup> C Write and Read Operations in SCCB Mode .....	4075
23.1.4.11	HS I <sup>2</sup> C System Test Mode .....	4076
23.1.5	HS I <sup>2</sup> C Programming Guide .....	4077
23.1.5.1	HS I <sup>2</sup> C Low-Level Programming Models .....	4077
23.1.5.1.1	HS I <sup>2</sup> C Programming Model in I <sup>2</sup> C Mode .....	4077
23.1.5.1.2	HS I <sup>2</sup> C (I2C5) Programming Model for Communication With Power Chips .....	4093
23.1.5.1.3	HS I <sup>2</sup> C Basic Programming Model in SCCB Mode .....	4094
23.1.6	HS I <sup>2</sup> C Register Manual .....	4103
23.1.6.1	HS I <sup>2</sup> C Instance Summary .....	4103
23.1.6.2	HS I <sup>2</sup> C Registers .....	4103
23.1.6.2.1	HS I <sup>2</sup> C Register Summary .....	4103
23.1.6.2.2	HS I <sup>2</sup> C Register Description .....	4105
23.2	HDQ/1-Wire .....	4135
23.2.1	HDQ/1-Wire Overview .....	4135
23.2.2	HDQ/1-Wire Environment .....	4136
23.2.2.1	HDQ/1-Wire Functional Modes .....	4136
23.2.2.2	HDQ and 1-Wire (SDQ) Protocols .....	4136
23.2.2.2.1	HDQ Protocol Initialization (Default) .....	4136
23.2.2.2.2	1-Wire (SDQ) Protocol Initialization .....	4137
23.2.2.2.3	Communication Sequence (HDQ and 1-Wire Protocols) .....	4137
23.2.3	HDQ/1-Wire Functional Description .....	4139
23.2.3.1	HDQ/1-Wire Block Diagram .....	4139
23.2.3.2	HDQ/1-Wire Clocking Configuration .....	4140
23.2.3.2.1	HDQ/1-Wire Clocks .....	4140
23.2.3.3	HDQ/1-Wire Hardware and Software Reset .....	4141
23.2.3.4	HDQ/1-Wire Power Management .....	4141
23.2.3.4.1	Auto-Idle Mode .....	4141
23.2.3.4.2	Power-Down Mode .....	4141
23.2.3.5	HDQ Interrupt Requests .....	4142
23.2.3.6	HDQ Mode (Default) .....	4142
23.2.3.6.1	HDQ Mode Features .....	4142
23.2.3.6.2	Description .....	4142
23.2.3.6.3	Single-Bit Mode .....	4143
23.2.3.6.4	Interrupt Conditions .....	4143
23.2.3.7	1-Wire Mode .....	4144
23.2.3.7.1	1-Wire Mode Features .....	4144
23.2.3.7.2	Description .....	4144
23.2.3.7.3	1-Wire Single-Bit Mode Operation .....	4144
23.2.3.7.4	Interrupt Conditions .....	4145
23.2.3.7.5	Status Flags .....	4145
23.2.4	HDQ/1-Wire Integration .....	4145
23.2.5	HDQ/1-Wire Programming Models .....	4147
23.2.5.1	Global Initialization .....	4147
23.2.5.1.1	Surrounding Modules Global Initialization .....	4147
23.2.5.1.2	HDQ/1-Wire Module Global Initialization .....	4147
23.2.5.2	HDQ Operational Modes Configuration .....	4148
23.2.5.2.1	Main Sequence – HDQ Write Operation Mode .....	4148
23.2.5.2.2	Main Sequence - HDQ Read Operation Mode .....	4148
23.2.5.3	1-Wire Operational Modes Configuration .....	4148

23.2.5.3.1	Main Sequence - 1-Wire Write Operation Mode .....	4148
23.2.5.3.2	Main Sequence - 1-Wire Read Operation Mode .....	4149
23.2.5.3.3	Sub-sequence - Initialize 1-Wire Slave .....	4149
23.2.6	HDQ/1-Wire Register Manual .....	4149
23.2.6.1	HDQ/1-Wire Instance Summary .....	4149
23.2.6.2	HDQ/1-Wire Registers .....	4150
23.2.6.2.1	HDQ/1-Wire Register Summary .....	4150
23.2.6.2.2	HDQ/1-Wire Register Description .....	4150
23.3	UART/IrDA/CIR .....	4155
23.3.1	UART/IrDA/CIR Overview .....	4155
23.3.1.1	UART Features .....	4156
23.3.1.2	IrDA Features .....	4156
23.3.1.3	CIR Features .....	4157
23.3.2	UART/IrDA/CIR Environment .....	4158
23.3.2.1	System Using UART Communication With Hardware Handshake .....	4158
23.3.2.2	System Using IrDA Communication Protocol .....	4158
23.3.2.3	System Using CIR Communication Protocol With Remote Control .....	4158
23.3.2.4	UART Interface .....	4159
23.3.2.4.1	Description .....	4159
23.3.2.4.2	UART Protocol and Data Format .....	4159
23.3.2.5	IrDA Functional Interfaces .....	4160
23.3.2.5.1	UART3 Interface Description .....	4160
23.3.2.5.2	IrDA Protocol and Data Format .....	4161
23.3.2.6	CIR Functional Interfaces .....	4168
23.3.2.6.1	CIR Interface Description .....	4168
23.3.2.6.2	CIR Protocol and Data Format .....	4168
23.3.3	UART/IrDA/CIR Integration .....	4173
23.3.4	UART/IrDA/CIR Functional Description .....	4175
23.3.4.1	Block Diagram .....	4175
23.3.4.2	Clock Configuration .....	4176
23.3.4.3	Software Reset .....	4176
23.3.4.4	Power Management .....	4176
23.3.4.4.1	UART Mode Power Management .....	4176
23.3.4.4.2	IrDA Mode Power Management (UART3 Only) .....	4177
23.3.4.4.3	CIR Mode Power Management (UART3 Only) .....	4177
23.3.4.4.4	Local Power Management .....	4178
23.3.4.5	Interrupt Requests .....	4178
23.3.4.5.1	UART Mode Interrupt Management .....	4178
23.3.4.5.2	IrDA Mode Interrupt Management .....	4179
23.3.4.5.3	CIR Mode Interrupt Management .....	4180
23.3.4.6	FIFO Management .....	4181
23.3.4.6.1	FIFO Trigger .....	4183
23.3.4.6.2	FIFO Interrupt Mode .....	4183
23.3.4.6.3	FIFO Polled Mode Operation .....	4185
23.3.4.6.4	FIFO DMA Mode Operation .....	4185
23.3.4.7	Mode Selection .....	4190
23.3.4.7.1	Register Access Modes .....	4190
23.3.4.7.2	UART/IrDA (SIR, MIR, FIR)/CIR Mode Selection .....	4192
23.3.4.8	Protocol Formatting .....	4195
23.3.4.8.1	UART Mode .....	4195
23.3.4.8.2	IrDA Mode (UART3 Only) .....	4201
23.3.4.8.3	CIR Mode (UART3 Only) .....	4205
23.3.5	UART/IrDA/CIR Basic Programming Model .....	4207

23.3.5.1	UART Programming Model .....	4207
23.3.5.1.1	Quick Start .....	4207
23.3.5.1.2	Hardware and Software Flow Control Configuration .....	4210
23.3.5.2	IrDA Programming Model (UART3 Only) .....	4213
23.3.5.2.1	SIR Mode .....	4213
23.3.5.2.2	MIR Mode .....	4214
23.3.5.2.3	FIR Mode .....	4215
23.3.6	UART/IrDA/CIR Register Manual .....	4216
23.3.6.1	UART/IrDA/CIR Instance Summary .....	4216
23.3.6.2	UART/IrDA/CIR Registers .....	4216
23.3.6.2.1	UART/IrDA/CIR Register Summary .....	4216
23.3.6.2.2	UART/IrDA/CIR Registers .....	4218
23.4	Multichannel Serial Port Interface (MCSPI) .....	4264
23.4.1	MCSPI Overview .....	4264
23.4.2	MCSPI Environment .....	4266
23.4.2.1	MCSPI Interface .....	4268
23.4.2.1.1	Basic MCSPI Pins for Master Mode .....	4268
23.4.2.1.2	Basic MCSPI Pins for Slave Mode .....	4268
23.4.2.1.3	Multichannel SPI Protocol and Data Format .....	4269
23.4.2.2	SPI in Master Mode .....	4272
23.4.2.3	SPI in Slave Mode .....	4273
23.4.3	MCSPI Integration .....	4274
23.4.4	MCSPI Functional Description .....	4277
23.4.4.1	MCSPI Block Diagram .....	4277
23.4.4.2	Master Mode .....	4278
23.4.4.2.1	Master Mode Features .....	4278
23.4.4.2.2	Master Transmit-and-Receive Mode (Full Duplex) .....	4278
23.4.4.2.3	Master Transmit-Only Mode (Half Duplex) .....	4279
23.4.4.2.4	Master Receive-Only Mode (Half Duplex) .....	4279
23.4.4.2.5	Single-Channel Master Mode .....	4280
23.4.4.2.6	Start-Bit Mode .....	4282
23.4.4.2.7	Chip-Select Timing Control .....	4282
23.4.4.2.8	Programmable SPI Clock (mcs pim_clk) .....	4282
23.4.4.3	Slave Mode .....	4284
23.4.4.3.1	Dedicated Resources .....	4284
23.4.4.3.2	Slave Transmit-and-Receive Mode .....	4286
23.4.4.3.3	Slave Transmit-Only Mode .....	4286
23.4.4.3.4	Slave Receive-Only Mode .....	4287
23.4.4.4	FIFO Buffer Management .....	4288
23.4.4.4.1	Buffer Almost Full .....	4289
23.4.4.4.2	Buffer Almost Empty .....	4290
23.4.4.4.3	End of Transfer Management .....	4291
23.4.4.5	Interrupts .....	4291
23.4.4.5.1	Interrupt Events in Master Mode .....	4291
23.4.4.5.2	Interrupt Events in Slave Mode .....	4292
23.4.4.5.3	Interrupt-Driven Operation .....	4293
23.4.4.5.4	Polling .....	4294
23.4.4.6	DMA Requests .....	4294
23.4.4.7	Power Saving Management .....	4294
23.4.4.7.1	Normal Mode .....	4294
23.4.4.7.2	Idle Mode .....	4295
23.4.5	MCSPI Programming Guide .....	4298
23.4.5.1	MCSPI Low-Level Programming Models .....	4298

23.4.5.1.1	Global Initialization .....	4298
23.4.5.1.2	Operational Mode Configuration .....	4298
23.4.5.1.3	Common Transfer Procedures With FIFO – Polling Method .....	4300
23.4.6	MCSPi Register Manual .....	4303
23.4.6.1	MCSPi Instance Summary .....	4303
23.4.6.2	MCSPi Registers .....	4303
23.4.6.2.1	MCSPi Register Summary .....	4303
23.4.6.2.2	MCSPi Register Description .....	4304
23.5	Multichannel Buffered Serial Port (McBSP) .....	4327
23.5.1	McBSP Overview .....	4327
23.5.1.1	McBSP Features .....	4327
23.5.2	McBSP Environment .....	4329
23.5.2.1	McBSP Functions .....	4329
23.5.2.2	McBSP Signals Descriptions .....	4329
23.5.2.3	McBSP Functions Description .....	4331
23.5.2.3.1	McBSP Modes .....	4331
23.5.2.3.2	McBSP Functions .....	4332
23.5.2.4	McBSP Protocols and Data Formats .....	4333
23.5.2.4.1	Words, Frames, and Phases Definitions .....	4334
23.5.2.4.2	Serial Protocol and Data Formats .....	4335
23.5.2.4.3	Audio Protocol and Data Formats .....	4336
23.5.2.4.4	Voice Protocol and Data Formats .....	4338
23.5.3	McBSP Integration .....	4339
23.5.4	McBSP Functional Description .....	4346
23.5.4.1	Block Diagram .....	4346
23.5.4.2	Signal Source Control .....	4346
23.5.4.3	McBSP Clock Configuration .....	4347
23.5.4.3.1	McBSP1 Clocks .....	4347
23.5.4.3.2	McBSP2 Clocks .....	4347
23.5.4.3.3	McBSP3 Clocks .....	4348
23.5.4.3.4	McBSP4 Clocks .....	4349
23.5.4.4	McBSP Software Reset .....	4350
23.5.4.5	McBSP Power Management .....	4350
23.5.4.5.1	McBSP Operating States .....	4350
23.5.4.5.2	McBSP Acknowledgment Modes .....	4350
23.5.4.5.3	Wake-Up Capability .....	4352
23.5.4.5.4	Analysis of the Receiver Smart Idle Behavior .....	4353
23.5.4.6	McBSP Interrupt Requests .....	4355
23.5.4.7	McBSP DMA Requests .....	4355
23.5.4.8	McBSP Data Transfer Process .....	4356
23.5.4.8.1	Data Transfer Process for 8- / 12- / 16- / 20- / 24- / 32-bits Long Words .....	4356
23.5.4.8.2	Bit Reordering (Option to Transfer LSB First) .....	4356
23.5.4.8.3	Clocking and Framing Data .....	4357
23.5.4.8.4	Frame Phases (Dual-Phase Frame I2S Support) .....	4360
23.5.4.8.5	McBSP Reception .....	4363
23.5.4.8.6	McBSP Transmission .....	4364
23.5.4.8.7	Enable/Disable the Transmit and Receive Processes .....	4364
23.5.4.8.8	MCBSP Data Transfer Mode .....	4365
23.5.4.9	McBSP SRG .....	4367
23.5.4.9.1	Clock Generation in the SRG .....	4368
23.5.4.9.2	Frame Sync Generation in the SRG .....	4369
23.5.4.9.3	Synchronizing SRG Outputs to an External Clock .....	4370
23.5.4.10	McBSP Exception/Error Conditions .....	4372

23.5.4.10.1	Introduction .....	4372
23.5.4.10.2	Overrun in the Receiver .....	4373
23.5.4.10.3	Unexpected Receive Frame-sync Pulse .....	4373
23.5.4.10.4	Underflow in the Receiver .....	4375
23.5.4.10.5	Underflow in the Transmitter .....	4375
23.5.4.10.6	Unexpected Transmit Frame-sync Pulse .....	4375
23.5.4.10.7	Overflow in the Transmitter .....	4376
23.5.4.11	McBSP DMA Configuration .....	4377
23.5.4.12	Multi-channel Selection Modes .....	4377
23.5.4.12.1	Channels, Blocks and Partitions .....	4377
23.5.4.12.2	Multi-channel Selection .....	4378
23.5.4.12.3	Configuring a Frame for Multi-channel Selection .....	4378
23.5.4.12.4	Using Eight Partitions .....	4378
23.5.4.12.5	Receive Multi-channel Selection Mode .....	4379
23.5.4.12.6	Using Two Partitions (Legacy Only) .....	4379
23.5.4.12.7	Transmit Multi-channel Selection Modes .....	4380
23.5.5	McBSP Basic Programming Model .....	4384
23.5.5.1	McBSP Initialization Procedure .....	4384
23.5.5.2	Reset and Initialization Procedure for the Sample Rate Generator .....	4387
23.5.5.3	Data Transfer DMA Request Configuration .....	4389
23.5.5.4	Interrupt Configuration .....	4389
23.5.5.5	Receiver Configuration .....	4389
23.5.5.5.1	Place the receiver in reset (Step 1) .....	4389
23.5.5.5.2	Programming the McBSP Registers for the Desired Receiver Configuration (Step 2) .....	4390
23.5.5.5.3	Take the Receiver Out of Reset (Step 3) .....	4391
23.5.5.6	Transmitter Configuration .....	4391
23.5.5.6.1	Place the Transmitter in Reset (Step 1) .....	4391
23.5.5.6.2	Programming the McBSP Registers for the Desired Transmitter Operation (Step 2) .....	4391
23.5.5.6.3	Take the receiver out of reset (Step 3) .....	4393
23.5.5.7	General-Purpose I/O on the McBSP Pins (Legacy Only) .....	4393
23.5.6	MCBSP Register Manual .....	4394
23.5.6.1	MCBSP Instance Summary .....	4394
23.5.6.2	MCBSP Registers .....	4394
23.5.6.2.1	MCBSP Register Summary .....	4394
23.5.6.2.2	MCBSP Register Description .....	4398
23.6	Multichannel PDM Controller .....	4433
23.6.1	McPDM Overview .....	4433
23.6.2	McPDM Environment .....	4434
23.6.2.1	McPDM Modes .....	4434
23.6.2.2	McPDM Signals .....	4434
23.6.2.3	McPDM Configurations .....	4435
23.6.2.4	Protocols and Data Formats .....	4435
23.6.2.4.1	Protocols and Data Formats in Normal Mode .....	4435
23.6.2.4.2	Protocols and Data Formats in Command Mode .....	4436
23.6.2.4.3	Protocols and Data Formats in Status Mode .....	4437
23.6.3	McPDM Integration .....	4440
23.6.4	McPDM Functional Description .....	4442
23.6.4.1	McPDM Block Diagram .....	4442
23.6.4.2	McPDM Clock Configuration .....	4442
23.6.4.3	McPDM Software Reset .....	4442
23.6.4.4	McPDM Power Management .....	4443
23.6.4.5	McPDM Interrupt Requests .....	4443
23.6.4.6	McPDM DMA Requests .....	4446

23.6.4.7	Uplink Path .....	4446
23.6.4.7.1	Uplink Path Features .....	4448
23.6.4.7.2	Uplink Path Description .....	4448
23.6.4.8	Downlink Path .....	4451
23.6.4.8.1	Downlink Path Features .....	4452
23.6.4.8.2	Downlink Path Description .....	4452
23.6.4.9	Error Reporting .....	4455
23.6.5	McPDM Programming Guide .....	4456
23.6.5.1	McPDM Low-Level Programming Models .....	4456
23.6.5.1.1	Global Initialization .....	4456
23.6.5.1.2	McPDM Events Servicing .....	4459
23.6.6	McPDM Register Manual .....	4462
23.6.6.1	McPDM Instance Summary .....	4462
23.6.6.2	McPDM Registers .....	4462
23.6.6.2.1	McPDM Register Summary .....	4462
23.6.6.2.2	McPDM Register Description .....	4462
23.7	Digital Microphone Module .....	4477
23.7.1	DMIC Overview .....	4477
23.7.2	DMIC Environment .....	4479
23.7.2.1	DMIC Interface With External MIC .....	4479
23.7.2.1.1	DMIC Interface Overview .....	4479
23.7.2.1.2	DMIC Signals .....	4480
23.7.2.1.3	Serial Data Format .....	4480
23.7.2.2	DMIC Functional Path .....	4481
23.7.3	DMIC Integration .....	4482
23.7.4	DMIC Functional Description .....	4484
23.7.4.1	Block Diagram .....	4484
23.7.4.2	DMIC Clock Configuration .....	4484
23.7.4.3	DMIC Software Reset .....	4485
23.7.4.4	DMIC Power Management .....	4485
23.7.4.5	DMIC Interrupt Requests .....	4486
23.7.4.6	DMIC DMA Requests .....	4488
23.7.4.7	Audio Data Processing .....	4489
23.7.4.7.1	Filtering .....	4489
23.7.4.8	FIFO Management .....	4489
23.7.4.8.1	FIFO General Description .....	4489
23.7.4.8.2	FIFO Operations .....	4490
23.7.4.8.3	FIFO Output Range Definition .....	4491
23.7.5	DMIC Programming Guide .....	4492
23.7.5.1	DMIC Low-Level Programming Models .....	4492
23.7.5.1.1	Global Initialization .....	4492
23.7.5.1.2	Operational Mode Configuration .....	4493
23.7.5.1.3	DMIC Events Servicing .....	4494
23.7.6	DMIC Register Manual .....	4496
23.7.6.1	DMIC Instance Summary .....	4496
23.7.6.2	DMIC Registers .....	4496
23.7.6.2.1	DMIC Register Summary .....	4496
23.7.6.2.2	DMIC Register Descriptions .....	4497
23.8	Multichannel Audio Serial Port .....	4510
23.8.1	McASP Overview .....	4510
23.8.2	McASP Environment .....	4511
23.8.2.1	McASP Signals .....	4511
23.8.2.2	Protocols and Data Formats .....	4511



23.8.2.2.1	Protocols Supported .....	4511
23.8.2.2.2	Definition of Terms .....	4512
23.8.2.2.3	S/PDIF Coding Format .....	4514
23.8.3	McASP Integration .....	4517
23.8.4	McASP Functional Description .....	4519
23.8.4.1	Block Diagram .....	4519
23.8.4.2	McASP Clock and Frame-Sync Configuration .....	4519
23.8.4.2.1	Transmit Clock .....	4519
23.8.4.2.2	Frame-Sync Generator .....	4520
23.8.4.3	Serializers .....	4520
23.8.4.4	Format Unit .....	4521
23.8.4.5	Transmit State-Machine .....	4522
23.8.4.6	Transmit TDM Sequencer .....	4523
23.8.4.7	McASP Software Reset .....	4523
23.8.4.8	McASP Power Management .....	4523
23.8.4.9	Transfer Modes .....	4523
23.8.4.9.1	DIT Transfer Mode .....	4523
23.8.4.10	Data Transmission .....	4526
23.8.4.10.1	Data Ready Status and Event/Interrupt Generation .....	4526
23.8.4.11	Interrupt Requests .....	4528
23.8.4.11.1	Transmit Data Ready Interrupt .....	4529
23.8.4.11.2	Error Interrupt .....	4529
23.8.4.12	DMA Requests .....	4529
23.8.4.13	Audio Mute (AMUTE) Function .....	4529
23.8.4.14	Error Reporting .....	4530
23.8.4.14.1	Buffer Underrun Error –Transmitter .....	4531
23.8.4.14.2	DMA Error –Transmitter .....	4531
23.8.4.14.3	Clock Failure Detection .....	4531
23.8.5	McASP Low-Level Programming Model .....	4533
23.8.5.1	Global Initialization .....	4533
23.8.5.1.1	Surrounding Modules Global Initialization .....	4533
23.8.5.1.2	McASP Global Initialization .....	4533
23.8.5.2	Operational Modes Configuration .....	4536
23.8.5.2.1	McASP DIT Transmission Mode .....	4536
23.8.5.2.2	McASP Event Servicing .....	4540
23.8.6	McASP Register Manual .....	4543
23.8.6.1	McASP Instance Summary .....	4543
23.8.6.2	McASP Registers .....	4543
23.8.6.2.1	McASP Register Summary .....	4543
23.8.6.2.2	McASP Register Description .....	4544
23.9	Serial Low-Power Inter-Chip Media Bus Controller .....	4567
23.9.1	SLIMbus Overview .....	4567
23.9.1.1	Terminology .....	4568
23.9.1.2	Main Features .....	4569
23.9.2	SLIMbus Environment .....	4570
23.9.2.1	SLIMbus Component Connection Example .....	4570
23.9.2.2	SLIMbus Functional Interface .....	4571
23.9.2.3	SLIMbus Protocols and Data Formats .....	4571
23.9.2.3.1	Frame Structure .....	4571
23.9.2.3.2	Control and Data Channels .....	4572
23.9.2.3.3	SLIMbus Transfer Protocols .....	4573
23.9.2.3.4	Half-Duplex Versus Full-Duplex Asynchronous Configurations .....	4575
23.9.3	SLIMbus Integration .....	4576

23.9.4	SLIMbus Functional Description .....	4581
23.9.4.1	SLIMbus Block Description .....	4581
23.9.4.2	SLIMbus Clock Configuration .....	4582
23.9.4.3	Software Reset .....	4582
23.9.4.4	System Power Management and Wakeup .....	4582
23.9.4.4.1	Wake-Up Request .....	4582
23.9.4.4.2	Power Management .....	4582
23.9.4.4.3	SLIMbus Boot and Reset Procedures .....	4582
23.9.4.4.4	SLIMbus Reconfigure Parameters .....	4585
23.9.4.5	Data Channels (Ports) Management .....	4586
23.9.4.5.1	Data input/output: manual mode .....	4586
23.9.4.5.2	Data Input/Output: DMA Request Mode .....	4587
23.9.4.5.3	Data Channel Agent FIFO Control and Status .....	4588
23.9.4.6	FIFO Management .....	4588
23.9.4.6.1	FIFO Mapping .....	4588
23.9.4.6.2	FIFO Status .....	4589
23.9.4.6.3	FIFO Configuration .....	4590
23.9.4.7	Data Channel Segment Composition .....	4591
23.9.4.8	SLIMbus Transfer Protocols .....	4593
23.9.4.8.1	Isochronous Transfer Protocol .....	4593
23.9.4.8.2	Pushed Transfer Protocol .....	4594
23.9.4.8.3	Pulled Transfer Protocol .....	4595
23.9.4.8.4	Asynchronous Transfer Protocol .....	4596
23.9.4.8.5	Extended Asynchronous Transfer Protocol .....	4598
23.9.4.9	Paired-up channels .....	4600
23.9.4.9.1	Paired-up behaviour for Isochronous TP .....	4601
23.9.4.9.2	Paired-up behaviour for Pushed TP .....	4601
23.9.4.9.3	Paired-up behaviour for Pulled TP .....	4601
23.9.4.10	Device and Addressing Management .....	4601
23.9.4.11	Message Management .....	4602
23.9.4.11.1	Transmission .....	4602
23.9.4.11.2	Message reception .....	4603
23.9.4.12	Interrupt Service Routines .....	4604
23.9.4.12.1	Interrupt Requests .....	4605
23.9.4.12.2	Possible Interrupts .....	4605
23.9.4.12.3	Interface Device ISRs .....	4606
23.9.4.12.4	Framer Device ISRs .....	4609
23.9.4.12.5	Shared Message Channel Transmit ISRs .....	4610
23.9.4.12.6	Shared Message Channel Receive ISRs .....	4612
23.9.4.12.7	Data Channel Port ISRs .....	4614
23.9.5	SLIMbus Programming Guide .....	4616
23.9.5.1	SLIMbus Low-level Programming Models .....	4616
23.9.5.1.1	Global Initialization .....	4616
23.9.5.1.2	Operational Modes Configuration .....	4617
23.9.5.1.3	SLIMbus Transfer Procedures .....	4620
23.9.5.1.4	Event handling .....	4627
23.9.6	SLIMbus Register Manual .....	4628
23.9.6.1	SLIMbus Register Overview .....	4628
23.9.6.2	SLIMbus Instance Summary .....	4628
23.9.6.3	SLIMbus Registers .....	4628
23.9.6.3.1	SLIMBUS1 Register Summary .....	4628
23.9.6.3.2	SLIMBUS2 Register Summary .....	4630
23.9.6.3.3	SLIMbus Register Descriptions .....	4631

23.10	MIPI-HSI .....	4690
23.10.1	HSI Overview .....	4690
23.10.2	HSI Environment .....	4691
23.10.2.1	HSI Signals .....	4691
23.10.2.2	HSI Typical Application .....	4692
23.10.3	HSI Integration .....	4692
23.10.4	HSI Functional Description .....	4695
23.10.4.1	Block Diagram .....	4695
23.10.4.2	Clock Configuration .....	4696
23.10.4.2.1	Interface Clock Domain .....	4697
23.10.4.2.2	Functional Clock Domain .....	4697
23.10.4.2.3	Receive Clock Domain (Port 1 and Port 2) .....	4697
23.10.4.3	Software Reset .....	4698
23.10.4.4	Power Management .....	4698
23.10.4.4.1	HSI Module Power Management Modes .....	4698
23.10.4.5	Interrupt Requests .....	4699
23.10.4.5.1	Interrupts .....	4700
23.10.4.6	DMA Module .....	4702
23.10.4.6.1	DMA Overview .....	4702
23.10.4.6.2	DMA Memories .....	4702
23.10.4.6.3	DMA Basic Operation Outline .....	4702
23.10.4.6.4	DMA Configuration .....	4702
23.10.4.7	HSR Module .....	4703
23.10.4.7.1	Receive Data .....	4703
23.10.4.7.2	Transmission Detection .....	4704
23.10.4.7.3	Dynamic OPP Changes .....	4704
23.10.4.7.4	Configuration .....	4705
23.10.4.7.5	Receive Buffers .....	4706
23.10.4.7.6	Receive Operations .....	4706
23.10.4.7.7	Receive Exceptions .....	4707
23.10.4.8	HST Block .....	4708
23.10.4.8.1	Configuration .....	4708
23.10.4.8.2	Mapping .....	4709
23.10.4.8.3	Transmission Buffer .....	4710
23.10.4.8.4	Break Pattern Transmission .....	4710
23.10.4.8.5	Transmission Operations .....	4710
23.10.4.8.6	Transmission Exceptions .....	4711
23.10.4.9	HSI Wake Generator .....	4711
23.10.4.9.1	Software Management of HSI Wake Generator .....	4712
23.10.4.9.2	Automatic Management of HSI Wake Generator .....	4712
23.10.4.10	FIFO Subsystem .....	4712
23.10.4.10.1	FIFO Subsystem Architecture .....	4712
23.10.4.10.2	Transmit FIFO Architecture .....	4713
23.10.4.10.3	Receive FIFO Architecture .....	4715
23.10.4.10.4	FIFO Addressing .....	4716
23.10.4.11	Error Reporting .....	4716
23.10.4.11.1	Detection of Errors .....	4716
23.10.4.11.2	Error Counters .....	4717
23.10.4.11.3	Error Registers .....	4717
23.10.5	HSI Programming Guide .....	4719
23.10.5.1	HSI Low-level Programming Models .....	4719
23.10.5.1.1	Global Initialization .....	4719
23.10.5.1.2	HSI Operational Modes Configuration .....	4719

23.10.6	HSI Register Manual .....	4728
23.10.6.1	HSI Instance Summary .....	4728
23.10.6.2	HSI_TOP Registers .....	4728
23.10.6.2.1	HSI_TOP Register Summary .....	4728
23.10.6.2.2	HSI_TOP Register Description .....	4729
23.10.6.3	HSI_DMA_CHANNELS registers .....	4741
23.10.6.3.1	HSI_DMA_CHANNELS Register Summary .....	4741
23.10.6.3.2	HSI_DMA_CHANNELS Register Description .....	4742
23.10.6.4	HSI_PORTS Registers .....	4746
23.10.6.4.1	HSI_PORTS Register Summary .....	4746
23.10.6.4.2	HSI_PORTS Register Description .....	4747
23.11	High-Speed Multiport USB Host Subsystem .....	4765
23.11.1	High-Speed Multiport USB Host Subsystem Overview .....	4765
23.11.1.1	Main Features .....	4766
23.11.2	HS Multiport USB Host Subsystem Environment .....	4768
23.11.2.1	ULPIs .....	4768
23.11.2.1.1	Transceiver Interface Configuration in ULPI Mode .....	4770
23.11.2.1.2	TLL Interface Configuration in ULPI Mode .....	4772
23.11.2.2	FS/LS Serial Interfaces .....	4774
23.11.2.2.1	Encoding in Serial Mode .....	4775
23.11.2.2.2	Sideband Signals for Serial Modes .....	4777
23.11.2.2.3	External USB Transceiver Interface Configurations in Serial Modes .....	4778
23.11.2.2.4	TLL Interface Configurations in Serial Modes .....	4782
23.11.2.2.5	Interface Description in Serial Modes .....	4788
23.11.2.3	HSIC Interface .....	4790
23.11.3	HS Multiport USB Host Subsystem Integration .....	4791
23.11.4	HS Multiport USB Host Subsystem Functional Description .....	4794
23.11.4.1	USB Host Controller Functionality .....	4794
23.11.4.1.1	HS USB Host Controller Software Reset .....	4795
23.11.4.1.2	HS USB Host Controller Power Management .....	4796
23.11.4.1.3	HS USB Host Controller Save and Restore .....	4797
23.11.4.1.4	HS USB Host Controller Port Status .....	4797
23.11.4.1.5	HS USB Host Controller Burst Control .....	4797
23.11.4.1.6	OHCI Implementation Specifications .....	4797
23.11.4.1.7	UTMI Interface .....	4798
23.11.4.1.8	ULPI Interface .....	4798
23.11.4.2	USBTLL Module Functionality .....	4798
23.11.4.2.1	USBTLL Software Reset .....	4798
23.11.4.2.2	USBTLL Power Management .....	4798
23.11.4.2.3	USBTLL Save and Restore .....	4799
23.11.4.2.4	USBTLL Channels and Ports .....	4799
23.11.4.2.5	USBTLL Channel Configuration .....	4800
23.11.4.2.6	USBTLL VBUS Management and Emulations .....	4801
23.11.4.2.7	USBTLL Multimode Serial Port .....	4803
23.11.4.2.8	USBTLL Attach/Connect Emulation for Serial Modes .....	4805
23.11.5	HS Multiport USB Host Subsystem Low-Level Programming Model .....	4806
23.11.5.1	Global Initialization .....	4806
23.11.5.1.1	Surrounding Modules Global Initialization .....	4806
23.11.5.1.2	HS Multiport USB Host Subsystem Global Initialization .....	4806
23.11.5.2	Operational Modes Configuration (Selecting and Configuring USB Connectivity) .....	4807
23.11.5.3	USBTLL Registers .....	4807
23.11.5.3.1	TLL Control and Status Registers .....	4807
23.11.5.3.2	ULPI PHY-Side Registers .....	4808

23.11.6	HS USB Host Subsystem Register Manual .....	4809
23.11.6.1	USB_Host_HS Instance Summary .....	4809
23.11.6.2	USBTLLHS_config Registers .....	4809
23.11.6.2.1	USBTLLHS_config Register Summary .....	4809
23.11.6.2.2	USBTLLHS_config Register Description .....	4809
23.11.6.3	USBTLLHS_ULPI Registers .....	4817
23.11.6.3.1	USBTLLHS_ULPI Register Summary .....	4817
23.11.6.3.2	USBTLLHS_ULPI Register Description .....	4818
23.11.6.4	HSUSBHOST Controller Registers .....	4846
23.11.6.4.1	HSUSBHOST Controller Register Summary .....	4846
23.11.6.4.2	HSUSB Controller Register Description .....	4846
23.11.6.5	OHCI Registers .....	4852
23.11.6.5.1	OHCI Register Summary .....	4852
23.11.6.5.2	OHCI Register Description .....	4853
23.11.6.6	EHCI Registers .....	4869
23.11.6.6.1	EHCI Register Summary .....	4869
23.11.6.6.2	EHCI Register Description .....	4870
23.12	High-Speed USB OTG Controller .....	4888
23.12.1	High-Speed USB OTG Controller Overview .....	4888
23.12.1.1	Main Features .....	4890
23.12.2	High-Speed USB OTG Controller Environment .....	4891
23.12.2.1	High-Speed USB OTG Controller Functional Interfaces .....	4892
23.12.3	High-Speed USB OTG Controller Integration .....	4893
23.12.4	High-Speed USB OTG Controller Functional Description .....	4895
23.12.4.1	Software Reset .....	4895
23.12.4.2	Power Management .....	4895
23.12.4.2.1	Overview .....	4895
23.12.4.2.2	System Power Management .....	4895
23.12.4.2.3	Local Power Management .....	4897
23.12.4.3	Inventra MUSBMHDRC .....	4897
23.12.4.4	Configuration .....	4899
23.12.4.5	Basic Operation .....	4899
23.12.4.5.1	Module Initialization .....	4899
23.12.4.5.2	Transaction Handling .....	4899
23.12.4.6	Optional Features .....	4900
23.12.4.6.1	Double Packet Buffering .....	4900
23.12.4.6.2	USBOTGHS Support for Big Endian .....	4900
23.12.4.6.3	DMA .....	4901
23.12.4.7	Automatic Packet Splitting/Combining for Bulk Transfers .....	4902
23.12.4.8	High-Bandwidth Isochronous Endpoints .....	4902
23.12.4.9	Carkit Implementation .....	4903
23.12.4.10	USB PHY .....	4903
23.12.5	High-Speed USB OTG Controller Basic Programming Model .....	4904
23.12.5.1	Global Initialization .....	4904
23.12.5.1.1	Surrounding Modules Global Initialization .....	4904
23.12.5.1.2	High-Speed USB OTG Controller Global Initialization .....	4904
23.12.5.2	High-Speed USB OTG Controller Interface Modes .....	4905
23.12.5.3	High-Speed USB OTG Controller Power Management Configuration .....	4905
23.12.5.3.1	No Application Mode .....	4905
23.12.5.3.2	Host or Peripheral Mode .....	4905
23.12.5.3.3	Switching Between Host and Peripheral Mode .....	4905
23.12.6	High-Speed USB OTG Controller Register Manual .....	4905
23.12.6.1	HSUSBOTG Registers .....	4906

23.12.6.1.1	HSUSBOTG Register Summary .....	4906
23.12.6.1.2	HSUSBOTG Register Description .....	4906
23.12.6.2	USBPHY Registers .....	4910
23.12.6.2.1	USBPHY Register Summary .....	4910
23.12.6.2.2	USBPHY Register Description .....	4911
23.13	Full-Speed USB Host Controller .....	4930
23.13.1	FSUSB Host Controller Overview .....	4930
23.13.1.1	Main Features .....	4930
23.13.2	FSUSB Host Controller Environment .....	4932
23.13.2.1	FSUSB Host Controller Functional Interfaces Description .....	4932
23.13.3	FSUSB Host Controller Integration .....	4933
23.13.4	FSUSB Host Controller Functional Description .....	4935
23.13.4.1	Resets .....	4935
23.13.4.2	Power Management .....	4935
23.13.4.2.1	Features .....	4935
23.13.4.2.2	IDLE Protocol .....	4935
23.13.4.2.3	Standby Protocol .....	4936
23.13.4.3	FSUSB Host Controller Bidirectional Interface Modes .....	4936
23.13.5	FSUSB Host Controller Basic Programming Model .....	4937
23.13.5.1	Global Initialization .....	4937
23.13.5.1.1	Surrounding Modules Global Initialization .....	4937
23.13.5.1.2	FSUSB Host Controller Global Initialization .....	4937
23.13.6	USBFSHOST Register Manual .....	4939
23.13.6.1	USBFSHOST Instance Summary .....	4939
23.13.6.2	USBFSHOST Registers .....	4939
23.13.6.2.1	USBFSHOST Register Summary .....	4939
23.13.6.2.2	USBFSHOST Register Description .....	4940
<b>24</b>	<b>MMC/SD/SDIO .....</b>	<b>4957</b>
24.1	MMC/SD/SDIO Overview .....	4958
24.1.1	MMC/SD/SDIO Features .....	4961
24.2	MMC/SD/SDIO Environment .....	4964
24.2.1	MMC/SD/SDIO Functional Modes .....	4964
24.2.1.1	MMC/SD/SDIO Connected to an MMC, an SD, or an SDIO Card .....	4964
24.2.1.2	MMC/SD/SDIO Connected to an MMC, an SD, or an SDIO Card Through an External Transceiver Device .....	4965
24.2.2	Protocol and Data Format .....	4966
24.2.2.1	Protocol .....	4966
24.2.2.2	Data Format .....	4968
24.3	MMC/SD/SDIO Integration .....	4972
24.4	MMC/SD/SDIO Functional Description .....	4975
24.4.1	Block Diagram .....	4975
24.4.2	Resets .....	4975
24.4.2.1	Hardware Reset .....	4975
24.4.2.2	Software Reset .....	4975
24.4.3	Power Management .....	4976
24.4.4	Interrupt Requests .....	4979
24.4.4.1	Interrupt-Driven Operation .....	4982
24.4.4.2	Polling .....	4982
24.4.5	DMA Modes .....	4982
24.4.5.1	MASTER DMA Operations .....	4983
24.4.5.1.1	Descriptor Table Description .....	4984
24.4.5.1.2	Requirements for Descriptors .....	4985
24.4.5.1.3	Advanced DMA Description .....	4986

24.4.5.2	SLAVE DMA Operations .....	4987
24.4.5.2.1	DMA Receive Mode .....	4987
24.4.5.2.2	DMA Transmit Mode .....	4988
24.4.6	Mode Selection .....	4989
24.4.7	Buffer Management .....	4989
24.4.7.1	Data Buffer .....	4989
24.4.7.1.1	Memory Size, Block Length, and Buffer-Management Relationship .....	4992
24.4.7.1.2	Data Buffer Status .....	4993
24.4.8	Transfer Process .....	4993
24.4.8.1	Different Types of Commands .....	4993
24.4.8.2	Different Types of Responses .....	4993
24.4.9	Transfer or Command Status and Errors Reporting .....	4994
24.4.9.1	Busy Time-Out for R1b, R5b Response Type .....	4994
24.4.9.2	Busy Time-Out After Write CRC Status .....	4995
24.4.9.3	Write CRC Status Time-Out .....	4995
24.4.9.4	Read Data Time-Out .....	4996
24.4.9.5	Boot Acknowledge Time-Out .....	4997
24.4.10	Auto Command 12 Timings .....	4998
24.4.10.1	ACMD12 Timings During Write Transfer .....	4998
24.4.10.2	ACMD12 Timings During Read Transfer .....	4998
24.4.11	Transfer Stop .....	4998
24.4.12	Retention Mode .....	4999
24.4.13	Output Signals Generation .....	5000
24.4.13.1	Generation on Falling Edge of MMC Clock .....	5000
24.4.13.2	Generation on Rising Edge of MMC Clock .....	5000
24.4.14	Card Boot Mode Management .....	5001
24.4.14.1	Boot Mode Using CMD0 .....	5001
24.4.14.2	Boot Mode With CMD Line Tied to 0 .....	5001
24.4.15	MMC CE-ATA Command Completion Disable Management .....	5002
24.4.16	Test Registers .....	5002
24.4.17	MMC/SD/SDIO Hardware Status Features .....	5003
24.5	MMC/SD/SDIO Programming Guide .....	5004
24.5.1	Low-Level Programming Models .....	5004
24.5.1.1	Global Initialization .....	5004
24.5.1.1.1	Surrounding Modules Global Initialization .....	5004
24.5.1.1.2	MMC/SD/SDIO Host Controller Initialization Flow .....	5004
24.5.1.2	Operational Modes Configuration .....	5007
24.5.1.2.1	Basic Operations for MMC/SD/SDIO Host Controller .....	5007
24.5.1.2.2	Bus Voltage Selection .....	5024
24.5.1.2.3	Boot Mode Configuration .....	5026
24.6	MMC/SD/SDIO Register Manual .....	5030
24.6.1	MMC/SD/SDIO Instance Summary .....	5030
24.6.2	MMC/SD/SDIO Registers .....	5030
24.6.2.1	MMC/SD/SDIO Register Summary .....	5030
24.6.2.2	MMC/SD/SDIO Register Description .....	5031
<b>25</b>	<b>General-Purpose Interface .....</b>	<b>5087</b>
25.1	General-Purpose Interface Overview .....	5088
25.2	General-Purpose Interface Environment .....	5091
25.2.1	General-Purpose Interface as a Keyboard Interface .....	5091
25.2.2	General-Purpose Interface Signals .....	5092
25.3	General-Purpose Interface Integration .....	5095
25.4	General-Purpose Interface Functional Description .....	5099
25.4.1	General-Purpose Interface Block Diagram .....	5099

25.4.2	General-Purpose Interface Interrupt and Wake-Up Features .....	5100
25.4.2.1	Synchronous Path: Interrupt Request Generation .....	5100
25.4.2.2	Asynchronous Path: Wake-Up Request Generation .....	5101
25.4.2.3	Wake-Up Event Conditions During Transition To and From IDLE State .....	5102
25.4.2.4	Interrupt (or Wake-Up) Line Release .....	5103
25.4.3	General-Purpose Interface Clock Configuration .....	5104
25.4.3.1	Clocking .....	5104
25.4.4	General-Purpose Interface Hardware and Software Reset .....	5104
25.4.5	General-Purpose Interface Power Management .....	5105
25.4.5.1	Power Domain .....	5105
25.4.5.2	Power Management .....	5105
25.4.5.2.1	Idle Scheme .....	5105
25.4.5.2.2	Operating Modes .....	5105
25.4.5.2.3	System Power Management and Wakeup .....	5105
25.4.5.2.4	Module Power Saving .....	5106
25.4.6	General-Purpose Interface Interrupt and Wake-Up Requests .....	5108
25.4.6.1	Interrupt Request Generation .....	5108
25.4.6.2	Wake-Up Request Generation .....	5109
25.4.7	General-Purpose Interface Channel Description .....	5110
25.4.8	General-Purpose Interface Data Input/Output Capabilities .....	5112
25.4.9	General-Purpose Interface Set-and-Clear Protocol .....	5112
25.4.9.1	Description .....	5112
25.4.9.2	Clear Instruction .....	5113
25.4.9.2.1	Clear Register Addresses .....	5113
25.4.9.2.2	Clear Instruction Example .....	5113
25.4.9.3	Set Instruction .....	5113
25.4.9.3.1	Set Register Addresses .....	5113
25.4.9.3.2	Set Instruction Example .....	5114
25.5	General-Purpose Interface Programming Guide .....	5115
25.5.1	General-Purpose Interface Low-Level Programming Models .....	5115
25.5.1.1	Global Initialization .....	5115
25.5.1.1.1	Surrounding Modules Global Initialization .....	5115
25.5.1.1.2	General-Purpose Interface Module Global Initialization .....	5115
25.5.1.2	General-Purpose Interface Operational Modes Configuration .....	5116
25.5.1.2.1	General-Purpose Interface Read Input Register .....	5116
25.5.1.2.2	General-Purpose Interface Write Output Register .....	5116
25.5.1.2.3	General-Purpose Interface Set Bit Function .....	5116
25.5.1.2.4	General-Purpose Interface Clear Bit Function .....	5116
25.6	General-Purpose Interface Register Manual .....	5117
25.6.1	General-Purpose Interface Instance Summary .....	5117
25.6.2	General-Purpose Interface Registers .....	5117
25.6.2.1	General-Purpose Interface Register Summary .....	5117
25.6.2.2	General-Purpose Interface Register Description .....	5119
<b>26</b>	<b>Keyboard Controller .....</b>	<b>5141</b>
26.1	Keyboard Controller Overview .....	5142
26.2	Keyboard Controller Environment .....	5144
26.2.1	Keyboard Controller Functions/Modes .....	5144
26.2.2	Keyboard Controller Signals .....	5145
26.2.3	Protocols and Data Formats .....	5145
26.3	Keyboard Controller Integration .....	5147
26.4	Keyboard Controller Functional Description .....	5149
26.4.1	Keyboard Controller Block Diagram .....	5149
26.4.2	Keyboard Controller Software Reset .....	5150



26.4.3	Keyboard Controller Power Management .....	5150
26.4.4	Keyboard Controller Interrupt Requests .....	5151
26.4.5	Keyboard Controller Software Mode .....	5151
26.4.6	Keyboard Controller Hardware Decoding Modes .....	5151
26.4.6.1	Functional Modes .....	5151
26.4.6.2	Keyboard Controller Timer .....	5152
26.4.6.3	State-Machine Status .....	5153
26.4.6.4	Keyboard Controller Interrupt Generation .....	5154
26.4.6.4.1	Interrupt-Generation Scheme .....	5154
26.4.6.4.2	Keyboard Buffer and Missed Events (Overrun Feature) .....	5156
26.4.7	Keyboard Controller Key Coding Registers .....	5156
26.4.8	Keyboard Controller Register Access .....	5157
26.4.8.1	Write Registers Access .....	5157
26.4.8.2	Read Registers Access .....	5158
26.5	Keyboard Controller Programming Guide .....	5159
26.5.1	Keyboard Controller Low-Level Programming Models .....	5159
26.5.1.1	Global Initialization .....	5159
26.5.1.1.1	Surrounding Modules Global Initialization .....	5159
26.5.1.1.2	Keyboard Controller Global Initialization .....	5159
26.5.1.2	Operational Modes Configuration .....	5160
26.5.1.2.1	Keyboard Controller in Hardware Decoding Mode (Default Mode) .....	5160
26.5.1.2.2	Keyboard Controller Software Scanning Mode .....	5161
26.5.1.2.3	Using the Timer .....	5161
26.5.1.2.4	State-Machine Status Register .....	5161
26.5.1.3	Keyboard Controller Events Servicing .....	5161
26.6	Keyboard Controller Register Manual .....	5163
26.6.1	Keyboard Controller Instance Summary .....	5163
26.6.2	Keyboard Controller Registers .....	5163
26.6.2.1	Keyboard Controller Register Summary .....	5163
26.6.2.2	Keyboard Controller Register Description .....	5163
<b>27</b>	<b>Initialization .....</b>	<b>5179</b>
27.1	Initialization Overview .....	5180
27.1.1	Terminology .....	5180
27.1.2	Initialization Process .....	5180
27.2	Preinitialization .....	5182
27.2.1	Power Requirements .....	5182
27.2.2	Access to Power-Management IC .....	5185
27.2.2.1	Power-Management IC Detection .....	5186
27.2.2.2	Summary of the I <sup>2</sup> C Transactions .....	5186
27.2.3	Clock, Reset, and Control .....	5188
27.2.3.1	Overview .....	5188
27.2.3.2	Clocking Scheme .....	5189
27.2.3.2.1	Required System Input Clocks .....	5189
27.2.3.2.2	Optional System Output Clocks .....	5190
27.2.3.3	Reset Configuration .....	5190
27.2.3.3.1	ON/OFF Interconnect and Power-On-Reset .....	5190
27.2.3.3.2	Warm Reset .....	5190
27.2.3.3.3	Peripheral Reset by sys_pwron_reset_out .....	5190
27.2.3.3.4	Peripheral Reset by GPIO .....	5190
27.2.3.3.5	Warm Reset Impact on GPIOs .....	5191
27.2.3.4	TWL Power/Audio Devices Control .....	5191
27.2.3.5	TWL6030 Power Request Signals .....	5192
27.2.4	Boot Configuration .....	5192

27.2.4.1	Clock Source Selection .....	5192
27.2.4.2	Booting Device Order Selection .....	5193
27.2.4.3	Boot Peripheral Pin Multiplexing .....	5195
27.3	Reset Sequences .....	5197
27.4	Device Initialization by ROM Code .....	5198
27.4.1	Booting Overview .....	5198
27.4.1.1	Booting Types .....	5198
27.4.1.2	ROM Code Architecture .....	5198
27.4.2	Memory Maps .....	5200
27.4.2.1	ROM Memory Map .....	5200
27.4.2.2	RAM Memory Map .....	5201
27.4.3	Overall Booting Sequence .....	5203
27.4.4	Startup and Configuration .....	5205
27.4.4.1	Startup .....	5205
27.4.4.2	Clocking Configuration .....	5207
27.4.4.3	Booting Device List Setup .....	5208
27.4.4.4	Software Booting Configuration .....	5208
27.4.4.4.1	Public Use of SAR RAM .....	5208
27.4.4.4.2	Software Booting Configuration Structure .....	5209
27.4.5	Peripheral Booting .....	5211
27.4.5.1	Description .....	5211
27.4.5.2	UART .....	5216
27.4.5.3	USB .....	5216
27.4.5.3.1	Overview .....	5216
27.4.5.3.2	PrevStartOk Status Flag Check .....	5218
27.4.5.3.3	Charger Detection .....	5219
27.4.5.3.4	SUSPEND and RESUME Handling .....	5219
27.4.5.3.5	USB Driver Descriptors .....	5219
27.4.5.3.6	USB Customized Descriptors .....	5223
27.4.5.3.7	System Interconnection – Integrated Transceiver .....	5224
27.4.5.3.8	System Interconnection - External Transceiver .....	5224
27.4.5.3.9	USB Driver Functionality .....	5224
27.4.6	Fast External Booting .....	5225
27.4.6.1	Overview .....	5225
27.4.6.2	Fast External Booting Procedure .....	5225
27.4.7	Memory Booting .....	5226
27.4.7.1	Overview .....	5226
27.4.7.2	Non-XIP Memory .....	5228
27.4.7.3	XIP Memory .....	5229
27.4.7.3.1	GPMC Initialization .....	5230
27.4.7.4	NAND .....	5230
27.4.7.4.1	Initialization and NAND Detection .....	5231
27.4.7.4.2	Read Sector Procedure .....	5235
27.4.7.5	OneNAND/Flex-OneNAND .....	5237
27.4.7.5.1	Initialization and OneNAND/Flex-OneNAND Detection .....	5238
27.4.7.5.2	OneNAND/Flex-OneNAND Read Sector Procedure .....	5238
27.4.7.5.3	OneNAND/Flex-OneNAND Support Limitations .....	5239
27.4.7.6	eMMC/eSD Embedded Memories and MMC/SD Cards .....	5239
27.4.7.6.1	eMMC/eSD Overview .....	5241
27.4.7.6.2	MMC/SD Overview .....	5243
27.4.7.6.3	Initialization and Detection .....	5244
27.4.7.6.4	Read Sector Procedure .....	5245
27.4.7.6.5	File System Handling .....	5246

27.4.7.7	LPDDR2-NVM .....	5252
27.4.7.7.1	Initialization and Detection .....	5252
27.4.8	Image Format .....	5254
27.4.8.1	Overview .....	5254
27.4.8.2	Configuration Header .....	5255
27.4.8.2.1	CHSETTINGS Item .....	5256
27.4.8.2.2	CHRAM Item .....	5257
27.4.8.2.3	CHFLASH Item .....	5258
27.4.8.2.4	CHMMCSO Item .....	5259
27.4.8.3	GP Header .....	5259
27.4.8.4	Image Execution .....	5260
27.4.9	Tracing .....	5260
27.5	Services for HLOS Support .....	5264
27.5.1	L2 Cache Maintenance .....	5264
27.5.2	Multicore Infrastructure Maintenance .....	5265
27.5.3	Snoop Control Unit (SCU) .....	5266
27.5.4	Lockdown Translation Lookaside Buffers (TLBs) .....	5267
<b>28</b>	<b>On-Chip Debug Support .....</b>	<b>5269</b>
28.1	Introduction .....	5270
28.2	Debug Ports .....	5271
28.2.1	IEEE1149.1 .....	5271
28.2.2	IEEE1149.7 .....	5271
28.2.3	Trace Port .....	5273
28.2.4	End Product .....	5274
28.3	Debugger Connection .....	5275
28.3.1	ICEPick Module .....	5275
28.3.2	Boot Modes .....	5275
28.3.2.1	Default Boot Mode .....	5276
28.3.2.2	Wait-In-Reset .....	5276
28.3.3	Dynamic TAP Insertion .....	5276
28.3.3.1	ICEPick Secondary TAPs .....	5276
28.4	Primary Debug Support .....	5278
28.4.1	Processor Native Debug Support .....	5278
28.4.1.1	Cortex-A9 MPU .....	5278
28.4.1.2	Cortex-M3 MPU .....	5278
28.4.1.3	DSP .....	5278
28.4.1.4	IVA-HD ARM968 .....	5278
28.4.1.5	IVA-HD Hardware Accelerators .....	5278
28.4.2	Cross-Triggering .....	5279
28.4.2.1	SOC Level Cross-Triggering .....	5280
28.4.3	Suspend .....	5281
28.4.3.1	Debug Aware Peripherals and Host Processors .....	5281
28.5	Power, Reset, and Clock Management Debug Support .....	5283
28.6	Performance Monitoring .....	5284
28.6.1	Cortex-A9 MPU Subsystem Performance Monitoring .....	5284
28.6.1.1	Performance Monitoring Unit .....	5284
28.6.1.2	Snoop Control Unit .....	5284
28.6.1.3	L2 Cache (PL310) .....	5285
28.6.2	Cortex-M3 MPU Subsystem Performance Monitoring .....	5285
28.6.2.1	Subsystem Counter Timer Module .....	5285
28.6.2.2	Cache Events .....	5285
28.6.3	DSP Subsystem Performance Monitoring .....	5286
28.6.3.1	Subsystem Counter Timer Module .....	5286

	28.6.3.2 L1 and L2 Events .....	5286
28.7	Processor Trace .....	5288
28.8	System Instrumentation .....	5289
	28.8.1 MIPI STM .....	5289
	28.8.2 Software Instrumentation .....	5290
	28.8.3 OCP Watch-Point .....	5290
	28.8.4 IVA-HD Pipeline .....	5291
	28.8.5 NoC Statistics Collector .....	5291
	28.8.5.1 OCP Target Load Monitoring .....	5293
	28.8.5.2 OCP Master Latency Monitoring .....	5293
	28.8.5.2.1 STATCOLL_LAT0 Configuration .....	5294
	28.8.5.2.2 STATCOLL_LAT1 Configuration .....	5295
	28.8.6 PM Instrumentation .....	5295
	28.8.7 CM Instrumentation .....	5296
28.9	Concurrent Debug Modes .....	5297
28.10	Memory Mapping .....	5299
<b>A</b>	<b>Glossary .....</b>	<b>5301</b>

## List of Figures

1-1.	OMAP4430 High-Tier Environment.....	257
1-2.	OMAP4430 Block Diagram .....	259
1-3.	POP Concept .....	268
1-4.	Stacked Memory Package on the POP Device .....	268
2-1.	Interconnect Overview.....	273
3-1.	Functional and Interface Clocks .....	293
3-2.	Generic Clock Domain .....	298
3-3.	Clock Domain State Transitions .....	299
3-4.	Clock Domain/Slave Module Clock-Management Interaction Sequence 1.....	301
3-5.	Clock Domain/Slave Module Clock-Management Interaction Sequence 2.....	302
3-6.	Clock Domain/Slave Module Clock-Management Interaction Sequence 3.....	303
3-7.	Sliding window for Dynamic Dependency .....	305
3-8.	Generic Power Domain .....	307
3-9.	Power Domain Transitions.....	309
3-10.	Generic Voltage Domain .....	311
3-11.	Generic Logic Voltage Management .....	312
3-12.	Generic Memory Voltage Management .....	313
3-13.	SmartReflex Static Voltage Adjustment.....	314
3-14.	SmartReflex Voltage Control Functional Overview.....	314
3-15.	Comparison of Energy Consumed With/Without DVFS.....	316
3-16.	Comparison of Energy Consumed With/Without DPS .....	317
3-17.	Performance Level and Applied Power-Management Techniques.....	319
3-18.	PMFW Overview .....	321
3-19.	SCRM Power-On Reset Sequence.....	347
3-20.	PRCM Module Power-On Reset Sequence .....	348
3-21.	Cortex-A9 MPU Power-On Reset Sequence.....	350
3-22.	Cortex-A9 MPU Warm Reset Sequence.....	351
3-23.	Cortex-A9 MPU Reset Sequence on Sleep and Wake-Up Transition .....	351
3-24.	IVAHD Power-On Reset Sequence .....	352
3-25.	IVAHD Software Warm Reset Sequence .....	353
3-26.	DSP Subsystem Power-On Reset Sequence .....	354
3-27.	DSP Subsystem Software Warm Reset Sequence .....	355
3-28.	DSP Subsystem Reset Sequence On Wake-Up Transitions From OSWR RETENTION State.....	356
3-29.	Cortex-M3 MPU Power-On Reset Sequence .....	356
3-30.	Cortex-M3 MPU Subsystem Software Warm Reset Sequence.....	357
3-31.	Global Warm Reset Sequence .....	359
3-32.	PRCM Module Clock Manager Overview .....	362
3-33.	PRM Clock Manager Overview (a) .....	364
3-34.	PRM Clock Manager Overview (b) .....	365
3-35.	CM1 Overview .....	366
3-36.	CM1_USB Clock Manager Overview.....	368
3-37.	CM1_ABE Clock Manager Overview .....	369
3-38.	CM2 Overview (a).....	371
3-39.	CM2 Overview (b).....	372
3-40.	Generic DPLL Functional Diagram .....	374
3-41.	DPLL_PER Overview.....	379
3-42.	DPLL_CORE Overview .....	382

3-43.	DPLL_ABE Overview .....	385
3-44.	DPLL_MPU Overview .....	387
3-45.	DPLL_IVA Overview .....	389
3-46.	DPLL_USB Overview .....	391
3-47.	DPLLs Cascading Overview .....	393
3-48.	CD_WKUP Overview .....	394
3-49.	CD_DSP Overview .....	397
3-50.	CD_STD_EFUSE Overview .....	399
3-51.	CD_CORTEXA9 Overview .....	400
3-52.	CD_L4_PER Overview .....	403
3-53.	CD_L3_INIT Overview .....	415
3-54.	CD_IVAHD Overview .....	421
3-55.	CD_SGX Overview .....	423
3-56.	CD_EMU Overview .....	425
3-57.	CD_DSS Overview .....	427
3-58.	CD_L4_CFG Overview .....	430
3-59.	CD_L3_INSTR Overview .....	433
3-60.	CD_L3_2 Overview .....	435
3-61.	CD_L3_1 Overview .....	437
3-62.	CD_EMIF Overview .....	439
3-63.	CD_CORTEXM3 Overview .....	441
3-64.	CD_DMA Overview .....	443
3-65.	CD_C2C Overview .....	445
3-66.	CD_CAM Overview .....	447
3-67.	CD_ABE Overview .....	449
3-68.	CD_L4_ALWON_CORE Overview .....	455
3-69.	PRM Voltage Control Architecture .....	481
3-70.	SmartReflex Integration .....	483
3-71.	SmartReflex Module Functional Overview .....	484
3-72.	Voltage Processor Functional Overview .....	488
3-73.	SmartReflex – SMPS Communication for Automatic Voltage Adjustments .....	491
3-74.	I/O Pads Daisy-Chain Configuration .....	505
3-75.	DPLL Output-Frequency Change .....	510
4-1.	Cortex-A9 MPU Subsystem Overview .....	1031
4-2.	Cortex-A9 MPU Subsystem Integration .....	1033
4-3.	Cortex-A9 MPU Subsystem Clocking Scheme .....	1035
4-4.	Cortex-A9 MPU Subsystem Reset Scheme .....	1036
4-5.	Cortex-A9 MPU Block Diagram .....	1039
4-6.	Cortex-A9 MPU Subsystem Power Domain Overview .....	1041
4-7.	Cortex-A9 MPU Subsystem Wake-up Generator .....	1044
6-1.	IVA-HD in the Device .....	1072
6-2.	IVA-HD Integration .....	1074
6-3.	IVA-HD Block Diagram .....	1076
7-1.	Dual Cortex-M3 MPU Subsystem Overview .....	1096
7-2.	Dual Cortex-M3 MPU Subsystem Integration Overview .....	1098
7-3.	Dual Cortex-M3 MPU Subsystem Clocking Scheme .....	1100
7-4.	Dual Cortex-M3 MPU Subsystem Reset Scheme .....	1101
7-5.	Dual Cortex-M3 MPU Block Diagram .....	1103
7-6.	WUGEN Overview .....	1107

7-7.	Event Communication Connection in Dual Cortex-M3 MPU Subsystem .....	1110
8-1.	ISS Overview.....	1120
8-2.	ISS Integration .....	1123
8-3.	ISS Interrupt Merger .....	1127
8-4.	ISS Power Management.....	1138
8-5.	ISS Interfaces and Interconnects Highlight .....	1157
8-6.	ISS CSI2-A, CSI2-B/CCP2 Serial Interface Configuration .....	1162
8-7.	ISS Interfaces CSI2-A PHY Diagram Four D-PHY Data Lane Configuration .....	1164
8-8.	ISS CSI2 Complex I/O Power FSM .....	1165
8-9.	ISS CSI2 RxMode and StopState FSM .....	1166
8-10.	ISS CCP2 YUV4:2:2 Big Endian .....	1177
8-11.	ISS CCP2 YUV4:2:2 Little Endian.....	1178
8-12.	ISS CCP2 YUV4:2:0 .....	1179
8-13.	ISS CCP2 RGB888 .....	1180
8-14.	ISS CCP2 RGB565 .....	1181
8-15.	ISS CCP2 RGB444 .....	1181
8-16.	ISS CCP2 RAW6 .....	1182
8-17.	ISS CCP2 RAW7 .....	1183
8-18.	ISS CCP2 RAW8 .....	1184
8-19.	ISS CCP2 RAW10 .....	1185
8-20.	ISS CCP2 RAW12 .....	1186
8-21.	ISS CCP2 JPEG8 and JPEG8 FSP.....	1187
8-22.	ISS CCP2 Integration .....	1188
8-23.	ISS CCP2 Receiver Block Diagram.....	1189
8-24.	ISS CCP2 VP_PCLK Gating During Blanking Periods .....	1191
8-25.	ISS CCP2 Data Organization in Memory.....	1196
8-26.	ISS CCP2 Data Organization Packing in Memory .....	1197
8-27.	ISS CCP2 Data Organization in Memory.....	1198
8-28.	CCP2_CTRL.VP_CLK_POL Settings.....	1204
8-29.	ISS CCP2 SOF and EOF Region Settings.....	1206
8-30.	ISS CCP2 Pixel Data Region Settings.....	1207
8-31.	ISS CCP2 Pixel Data Destination Settings.....	1208
8-32.	ISS CSI2 One Data-Lane Configuration.....	1246
8-33.	ISS CSI2 Two Data-Lane Merger Configuration .....	1246
8-34.	ISS CSI2 Three Data-Lane Merger Configuration .....	1247
8-35.	ISS CSI2 Four Data-Lane Merger Configuration .....	1248
8-36.	ISS CSI2 Protocol Layer With Short and Long Packets.....	1249
8-37.	ISS CSI2 Short Packet Structure .....	1249
8-38.	ISS CSI2 Long Packet Structure .....	1250
8-39.	ISS CSI2 Data Identifier Structure .....	1251
8-40.	ISS CSI2 Virtual Channel .....	1251
8-41.	ISS CSI2 General Frame Structure (Informative) .....	1253
8-42.	ISS CSI2 Digital Interlaced Video Frame (Informative).....	1254
8-43.	ISS CSI2 YUV4:2:0 8-Bit .....	1255
8-44.	ISS CSI2 YUV4:2:0 10-Bit.....	1256
8-45.	ISS CSI2 YUV4:2:0 8-Bit Legacy.....	1257
8-46.	ISS CSI2 YUV4:2:0 8-Bit + CSPS.....	1258
8-47.	ISS CSI2 YUV4:2:0 10-Bit + CSPS .....	1259
8-48.	ISS CSI2 YUV4:2:2 8-Bit .....	1260

8-49.	ISS CSI2 YUV4:2:2 10-Bit.....	1261
8-50.	ISS CSI2 RGB565.....	1261
8-51.	ISS CSI2 RGB888.....	1262
8-52.	ISS CSI2 RGB666.....	1263
8-53.	ISS CSI2 RGB444.....	1264
8-54.	ISS CSI2 RGB555.....	1264
8-55.	ISS CSI2 RAW6 .....	1266
8-56.	ISS CSI2 RAW7 .....	1268
8-57.	ISS CSI2 RAW8 .....	1270
8-58.	ISS CSI2 RAW10.....	1271
8-59.	ISS CSI2 RAW12.....	1272
8-60.	ISS CSI2 RAW14.....	1273
8-61.	ISS CSI2 JPEG8 .....	1274
8-62.	ISS CSI2 Generic Format .....	1274
8-63.	ISS CSI2-A/CSI2-B Integration .....	1279
8-64.	ISS CSI2-A Receiver Block Diagram .....	1280
8-65.	ISS CSI2 SHORT_PACKET Bit Field Format .....	1282
8-66.	ISS CSI2 Virtual Channel to Context .....	1283
8-67.	ISS CSI2 Pixel Data Destination Setting in Progressive and Interlaced Mode.....	1285
8-68.	ISS CSI2 Frame Processing .....	1286
8-69.	ISS CSI2 Frame Cropping .....	1287
8-70.	ISS CSI2 Horizontal Scaler .....	1288
8-71.	ISS CSI2 Receiver Global Reset Flow Chart .....	1290
8-72.	ISS TCTRL Integration.....	1326
8-73.	TCTRL Control-Signal Generation .....	1327
8-74.	ISS TCTRL Use of cam_global_reset With Global Reset Release Camera Modules.....	1329
8-75.	cam_strobe Signal-Generation for Red-Eye Removal.....	1333
8-76.	ISS BTE Integration .....	1344
8-77.	ISS BTE Burst Translation Principle .....	1346
8-78.	ISS BTE Context Mapping .....	1347
8-79.	ISS BTE Expected Access Locations in the Virtual Space .....	1348
8-80.	ISS BTE Context Alignment Constraints .....	1349
8-81.	ISS BTE TILER Mode Addressing in 90- or 270-Degree Orientation .....	1350
8-82.	BTE Bandwidth Limiter Example .....	1351
8-83.	ISS BTE Buffer Fill-Level Padding .....	1352
8-84.	ISS BTE 2D Burst Generation .....	1354
8-85.	ISS BTE Tiled Mode Addressing in 90- or 270-Degree Orientation (S = 1) .....	1357
8-86.	ISS CBUFF Integration .....	1381
8-87.	ISS CBUFF Top-Level Diagram .....	1384
8-88.	ISS CBUFF Write Mode CPU Interaction Example .....	1385
8-89.	ISS CBUFF CPU Writes Data Faster Than it Is Read by the OCPI Initiator .....	1386
8-90.	CBUFF Read Mode CPU Interaction Example .....	1386
8-91.	ISS CBUFF Read/Write Mode Example.....	1387
8-92.	ISS CBUFF Write Mode CPU Interaction Example BCF Used.....	1389
8-93.	ISS CBUFF Read Mode CPU Interaction Example (1).....	1389
8-94.	ISS CBUFF Read Mode CPU Interaction Example (2).....	1390
8-95.	ISS CBUFF BCF Pretrigger Example: Write Mode .....	1391
8-96.	ISS CBUFF BCF Pretrigger Example: Read Mode .....	1391
8-97.	ISS CBUFF Single-Slice Buffer (Write Mode) .....	1395



8-98. ISS CBUFF Single-Slice Buffer Example (Write Mode) .....	1396
8-99. ISS CBUFF Extended-Slice Buffer Example.....	1396
8-100. ISS CBUFF FIFO Mode .....	1397
8-101. ISS CBUFF FIFO Use Example .....	1397
8-102. ISS ISP Block Diagram .....	1422
8-103. ISS ISP High-Level Diagram.....	1429
8-104. ISS ISP VP High-Level Diagram .....	1430
8-105. ISS ISP VP VD Pulse .....	1432
8-106. ISS ISP PG High-Level Diagram .....	1435
8-107. ISS ISP IPIPEIF High-Level Diagram.....	1437
8-108. ISS ISP IPIPEIF Top-Level Block Diagram .....	1438
8-109. ISS ISP IPIPEIF Global Frame Definition in SDRAM Input Modes (Except Dark Frame) .....	1439
8-110. ISS ISP IPIPEIF Global Frame Definition in Dark Frame Subtract Mode.....	1440
8-111. ISS ISP IPIPEIF Double-Buffer Functionality .....	1441
8-112. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 0 Data Path.....	1443
8-113. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 1 Data Path.....	1444
8-114. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 2 Data Paths .....	1445
8-115. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 3 Data Paths: First Case .....	1446
8-116. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 3 Data Paths: Second Case .....	1447
8-117. ISS ISP IPIPEIF INPSRC1= 1 and INPSRC2 = 0 Data Path.....	1448
8-118. ISS ISP IPIPEIF INPSRC1= 2 and INPSRC2 = 0 Data Path.....	1449
8-119. ISS ISP IPIPEIF INPSRC1= 3 and INPSRC2 = 0 Data Path.....	1450
8-120. ISS ISP IPIPEIF Timing Generator Submodule Description.....	1450
8-121. ISS ISP IPIPEIF DCPM Subblock.....	1451
8-122. ISS ISP IPIPEIF Dark-Frame Subtraction Subblock .....	1453
8-123. ISS ISP IPIPEIF Resizer Offset Definition .....	1455
8-124. ISS ISP IPIPEIF YUV8P Settings .....	1457
8-125. ISS ISP IPIPE High-Level Diagram .....	1459
8-126. ISS ISP IPIPE Supported CFA Format .....	1460
8-127. ISS ISP IPIPE Module Block Diagram.....	1460
8-128. ISS ISP IPIPE Module Input Format .....	1461
8-129. ISS ISP IPIPE Module Processing Window Settings .....	1462
8-130. ISS ISP IPIPE Defect Pixel Correction .....	1462
8-131. ISS ISP IPIPE Pixel Numbering in Defect Correction Algorithm .....	1463
8-132. ISS ISP IPIPE Mirroring in Defect Correction and Noise Filter.....	1464
8-133. ISS ISP IPIPE White Balance .....	1464
8-134. ISS ISP IPIPE RGB2RGB Conversion Formula.....	1465
8-135. ISS ISP IPIPE Gamma Correction Module Block Diagram .....	1465
8-136. ISS ISP IPIPE Gamma Curve Example .....	1466
8-137. ISS ISP IPIPE Gamma Table Offset/Slope Packing.....	1466
8-138. ISS ISP IPIPE RGB2RGB 2nd Conversion Formula .....	1466
8-139. ISS ISP IPIPE RGB2RGB 2nd Conversion Formula .....	1467
8-140. ISS ISP IPIPE RGB2YCbCr Module Block Diagram .....	1467
8-141. ISS ISP IPIPE Chroma Subsampling Position.....	1467
8-142. ISS ISP IPIPE 4:2:2 Conversion Module Block Diagram .....	1468
8-143. ISS ISP IPIPE 2D Edge-Enhancer Indexing .....	1468
8-144. ISS ISP IPIPE 2D Edge Intensity LUT Formula .....	1468
8-145. ISS ISP IPIPE 2D Edge-Enhancer LUT Packing.....	1469
8-146. ISS ISP IPIPE 2D Edge-Enhancer Block Diagram.....	1470

8-147. ISS ISP IPIPE Edge Sharpener Details .....	1470
8-148. ISS ISP IPIPE 2D Edge-Intensity Clipping Formula .....	1470
8-149. ISS ISP IPIPE 2D Edge Enhancer and Sharpener Merger Formula .....	1471
8-150. ISS ISP IPIPE 2D Edge Chroma-Suppression Coefficient Sets .....	1471
8-151. ISS ISP IPIPE 2D Edge-Brightness and Contrast Adjustments Formula .....	1471
8-152. ISS ISP IPIPE Boxcar Operation (8 × 8 Block) .....	1472
8-153. ISS ISP IPIPE Boxcar Operation (16 × 16 Block) .....	1473
8-154. ISS ISP IPIPE Boxcar Data Packing in SDRAM .....	1473
8-155. ISS ISP RSZ High-Level Diagram .....	1475
8-156. ISS ISP RSZ Top-Level Block Diagram .....	1476
8-157. ISS ISP RSZ MTC DMA Bandwidth Control .....	1477
8-158. ISS ISP RSZ MTC Image Data Storage Pixel Order .....	1478
8-159. ISS ISP RSZ Typical Module Integration: High-Level Summary .....	1479
8-160. ISS ISP RSZ Operating Modes .....	1482
8-161. ISS ISP RSZ Input Data Cropper Block Diagram .....	1483
8-162. ISS ISP RSZ Input Data Cropping .....	1484
8-163. ISS ISP RSZ Averager Memory Utilization .....	1485
8-164. ISS ISP RSZ Averager Border Conditions .....	1485
8-165. ISS ISP RSZ Basic Interpolation Method .....	1486
8-166. ISS ISP RSZ Interpolation Filtering .....	1487
8-167. ISS ISP RSZ-A/RSZ-B Phase Averager Effect .....	1489
8-168. ISS ISP RSZ Chroma Position and Upsampling .....	1491
8-169. ISS ISP RSZ and Circular Buffer Settings .....	1493
8-170. ISS ISP RSZ and Circular Buffer Settings – Example 1 .....	1493
8-171. ISS ISP RSZ and Circular Buffer Settings – Example 2 .....	1494
8-172. ISS ISP H3A High-Level Diagram .....	1496
8-173. ISS ISP H3A Top-Level Block Diagram .....	1497
8-174. ISS ISP H3A Frame Format Settings .....	1497
8-175. ISS ISP H3A Red, Green, and Blue Pixel Extraction Examples .....	1499
8-176. ISS ISP H3A Horizontal/Vertical FV Pixel Configuration .....	1500
8-177. ISS ISP H3A IIR Filter Model .....	1501
8-178. ISS ISP H3A AE/AWB Window Configurations .....	1503
8-179. ISS ISP H3A Black Row of Windows Before Regular Rows of Windows .....	1504
8-180. ISS ISP H3A AE/AWB Window and Subsample Definition .....	1507
8-181. ISS ISP ISIF High-Level Diagram .....	1514
8-182. ISS ISP ISIF Top-Level Block Diagram .....	1515
8-183. ISS ISP ISIF Interface Block Diagram .....	1516
8-184. ISS ISP ISIF Sensor Linearization Block Diagram .....	1517
8-185. ISS ISP ISIF Linearization LUT Memories .....	1518
8-186. ISS ISP ISIF Linearization Block Diagram .....	1519
8-187. ISS ISP ISIF Input Data Formatter Block Diagram .....	1519
8-188. ISS ISP ISIF Splits an Input Line Into Three Output Lines .....	1520
8-189. ISS ISP ISIF Input Data Formatter Area Settings .....	1521
8-190. ISS ISP ISIF Data Formatter Output Control Example .....	1522
8-191. ISS ISP ISIF Conventional Read-Out Pattern .....	1523
8-192. ISS ISP ISIF Conventional Read-Out Pattern With 2-tap AFE .....	1524
8-193. ISS ISP ISIF Combine Three Input Lines Into Single Line .....	1524
8-194. ISS ISP ISIF Example of Combining Three Input Lines Into a Single Line .....	1525
8-195. ISS ISP ISIF Color Space Converter Block Diagram .....	1527

8-196. ISS ISP ISIF Color Space Converter Operation.....	1527
8-197. ISS ISP ISIF Color Space Converter Operation: CMYG to RGBG .....	1528
8-198. ISS ISP ISIF Color Space Conversion Example .....	1528
8-199. ISS ISP ISIF 1st Pixel/1st Line Generation .....	1528
8-200. ISS ISP ISIF 2nd Pixel/1st Line Generation .....	1528
8-201. ISS ISP ISIF 2nd Last Pixel/1st Line Generation .....	1529
8-202. ISS ISP ISIF Last Pixel/1st Line Generation.....	1529
8-203. ISS ISP ISIF 1st Pixel/Last Line Generation.....	1529
8-204. ISS ISP ISIF 2nd Pixel/Last Line Generation.....	1529
8-205. ISS ISP ISIF 2nd Last Pixel/Last Line Generation.....	1530
8-206. ISS ISP ISIF Last Pixel/Last Line Generation .....	1530
8-207. ISS ISP ISIF Black Clamp Block Diagram .....	1530
8-208. ISS ISP ISIF Clamp Value for Horizontal Direction .....	1532
8-209. ISS ISP ISIF Clamp Value for Vertical Direction Calculation .....	1533
8-210. ISS ISP ISIF Clamp Value for Vertical Direction With OB Region at the Left.....	1534
8-211. ISS ISP ISIF Clamp Value for Vertical Direction With OB Region at the Right .....	1534
8-212. ISS ISP ISIF Vertical Line Defect Correction Block Diagram.....	1535
8-213. ISS ISP ISIF Vertical Line Defects .....	1535
8-214. ISS ISP ISIF Vertical Line Defects .....	1536
8-215. ISS ISP ISIF 2D-LSC Block Diagram .....	1537
8-216. ISS ISP ISIF 2D-LSC Active Region for ISIF Input Frame.....	1538
8-217. ISS ISP ISIF Gain and Offset Mask Upsampling via Bilinear Interpolation.....	1539
8-218. ISS ISP ISIF White Balance Block Diagram .....	1542
8-219. ISS ISP ISIF Low-Pass Filter Block Diagram.....	1543
8-220. ISS ISP ISIF A-Law Compression Block Diagram .....	1543
8-221. ISS ISP ISIF A-Law Table Diagram.....	1544
8-222. ISS ISP ISIF A-Law Table Values.....	1545
8-223. ISS ISP ISIF Culling Block Diagram .....	1547
8-224. ISS ISP ISIF Example for Decimation Pattern.....	1547
8-225. ISS ISP ISIF Storage Formatter Block Diagram.....	1548
8-226. ISS ISP ISIF Frame Image Format Conversion .....	1550
8-227. ISS ISP ISIF VDINT0/VDINT1/VDINT2 Interrupt Behavior When VDPOL = 0.....	1552
8-228. ISS ISP ISIF VDINT0/VDINT1/VDINT2 Interrupt Behavior When VDPOL = 1 .....	1553
8-229. ISS ISP BL High-Level Diagram.....	1554
8-230. ISS ISP BL Block Diagram .....	1556
8-231. ISS ISP BL Address Alignment.....	1557
8-232. ISS ISP ISIF Initialization Flow Chart – Part One .....	1562
8-233. ISS ISP ISIF Initialization Flow Chart – Part Two .....	1563
8-234. ISS ISP RSZ Global Image Processing Settings – Subsequence 1.....	1583
8-235. ISS ISP RSZ Global Image Processing Settings – Subsequence 2.....	1584
8-236. ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1.....	1586
8-237. ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2.....	1587
8-238. SIMCOP Subsystem Overview .....	1875
8-239. SIMCOP Integration.....	1876
8-240. SIMCOP Interrupt Merger Overview .....	1880
8-241. JPEG Encode Pipeline.....	1882
8-242. Hardware Sequencer and Buffers in the SIMCOP Subsystem.....	1904
8-243. Hardware Sequencer and Buffer Integration.....	1905
8-244. SIMCOP Static Crossbar Overview .....	1907

8-245. Image Buffer Width Translation.....	1910
8-246. Coefficient Memory Width Translation .....	1911
8-247. SIMCOP Hardware Sequencer Overview .....	1914
8-248. Hardware Sequencing Example .....	1915
8-249. Hardware Sequencer Operation Example .....	1916
8-250. Typical Hardware Sequencer Controlled Sequencer With Pipe-Up and Pipe-Down .....	1918
8-251. Pipe-Up and Pipe-Down .....	1919
8-252. Rotation Macroblock Pipeline .....	1920
8-253. LDC Transformation Pipeline .....	1921
8-254. DMA in the SIMCOP Subsystem .....	1941
8-255. DMA Engine Integration .....	1942
8-256. DMA Block Diagram.....	1944
8-257. DMA Logical Channel States .....	1946
8-258. DMA Addressing.....	1948
8-259. 2-Block Large Transfer Example .....	1948
8-260. YUV4:2:0-NV12 Read/Write Chain.....	1951
8-261. Temporal Channel Sequence.....	1951
8-262. LDC in the SIMCOP Subsystem .....	1972
8-263. DCT in the SIMCOP Subsystem .....	1974
8-264. DCT Engine Integration.....	1975
8-265. DCT Block Diagram .....	1977
8-266. DCT I/O Address Configuration .....	1978
8-267. YUV4:2:0 Format Data Order.....	1979
8-268. YUV4:2:2 Format Data Order.....	1980
8-269. VLCDJ in the SIMCOP Subsystem .....	1989
8-270. VLCDJ Engine Integration.....	1990
8-271. VLCDJ Block Diagram .....	1992
8-272. JPEG Encoding Flow .....	1993
8-273. JPEG Decoding Flow .....	1994
8-274. JPEG Zigzag Order .....	1995
8-275. VLCDJ Ring Buffer .....	1995
8-276. Huffman Encode Table Organization .....	1996
8-277. Concurrence Schedule for JPEG Encoding and Decoding .....	2003
8-278. Concurrence Schedule for JPEG Transcoding .....	2004
8-279. ISS SIMCOP ROT Module Overview .....	2021
8-280. ISS SIMCOP ROT Engine Integration .....	2022
8-281. ISS SIMCOP ROT Module Block Diagram.....	2024
8-282. ISS SIMCOP ROT 90-Degree Rotation of YUV4:2:2 Data .....	2025
8-283. ISS SIMCOP ROT 270-Degree Rotation of YUV4:2:2 Data.....	2026
8-284. ISS SIMCOP ROT 180-Degree Rotation of YUV4:2:2 Data.....	2026
8-285. ISS SIMCOP ROT Block Representation Before Rotation.....	2027
8-286. ISS SIMCOP ROT 90-Degree Block Rotation .....	2027
8-287. ISS SIMCOP ROT 180-Degree Block Rotation .....	2027
8-288. ISS SIMCOP ROT 270-Degree Block Rotation .....	2027
8-289. ISS SIMCOP ROT Data-Block Shifting Example.....	2028
8-290. ISS SIMCOP ROT Horizontal Circular Shifting Example .....	2029
8-291. ISS SIMCOP ROT Using Horizontal Circular Shifting to Manage Misaligned Data .....	2030
9-1. Face Detect Highlight .....	2042
9-2. FDIF Integration .....	2044

9-3.	FDIF Block Diagram.....	2046
9-4.	FDIF Power-Management Transitions .....	2048
10-1.	DSS Overview .....	2071
10-2.	DSS Environment .....	2072
10-3.	DSS Integration.....	2075
10-4.	DSS Clock Tree .....	2076
10-5.	DSS Reset Scheme.....	2078
10-6.	DSS SidleAck/MStandby Generation .....	2080
10-7.	DSS Wake-Up Generation .....	2081
10-8.	DSS Interrupt Generation .....	2082
10-9.	DSS DMA Request Generation.....	2083
10-10.	Display Subsystem Highlight.....	2091
10-11.	LCD Support Parallel Interface .....	2095
10-12.	LCD Pixel Data Monochrome4 Passive Matrix .....	2097
10-13.	LCD Pixel Data Monochrome8 Passive Matrix .....	2097
10-14.	LCD Pixel Data 8-Bit Color Passive Matrix .....	2098
10-15.	LCD Pixel Data Color12 Active Matrix .....	2099
10-16.	LCD Pixel Data Color16 Active Matrix.....	2100
10-17.	LCD Pixel Data Color18 Active Matrix.....	2100
10-18.	LCD Pixel Data Color24 Active Matrix.....	2101
10-19.	Active Matrix Timing Diagram of Configuration 1 (Start of Frame) .....	2102
10-20.	Active Matrix Timing Diagram of Configuration 1 (Between Lines) .....	2102
10-21.	Active Matrix Timing Diagram of Configuration 1 (Between Frames) .....	2102
10-22.	Active Matrix Timing Diagram of Configuration 1 (End of Frame) .....	2102
10-23.	Active Matrix Timing Diagram of Configuration 2 (Start of Frame) .....	2103
10-24.	Active Matrix Timing Diagram of Configuration 2 (Between Lines) .....	2103
10-25.	Active Matrix Timing Diagram of Configuration 2 (Between Frames) .....	2103
10-26.	Active Matrix Timing Diagram of Configuration 2 (End of Frame) .....	2103
10-27.	Active Matrix Timing Diagram of Configuration 3 (Start of Frame) .....	2104
10-28.	Active Matrix Timing Diagram of Configuration 3 (Between Lines) .....	2104
10-29.	Active Matrix Timing Diagram of Configuration 3 (Between Frames) .....	2104
10-30.	Active Matrix Timing Diagram of Configuration 3 (End of Frame) .....	2105
10-31.	Passive Matrix Timing Diagram (Start of Frame) .....	2105
10-32.	Passive Matrix Timing Diagram (Between Lines) .....	2105
10-33.	Passive Matrix Timing Diagram (Between Frames) .....	2106
10-34.	Passive Matrix Timing Diagram (End of Frame) .....	2106
10-35.	TV Output Pixel Data .....	2107
10-36.	Display Subsystem Integration .....	2108
10-37.	DISPC Architecture Overview .....	2111
10-38.	DISPC Clock Tree Overview.....	2112
10-39.	DMA Engine Overview.....	2117
10-40.	90° Rotation with Mirroring .....	2122
10-41.	Graphics Pipeline.....	2127
10-42.	Palette/Gamma Correction Architecture.....	2129
10-43.	Data Memory Organization: Color Mode .....	2129
10-44.	Data Memory Organization: Gray Mode.....	2129
10-45.	Expansion to ARGB40.....	2130
10-46.	Configuration 1: Video Pipeline.....	2131
10-47.	Configuration 2: Video Pipeline.....	2131

10-48. YCbCr to RGB Registers (FULLRANGE=0), 8 Bits Outputs .....	2133
10-49. YCbCr to RGB Registers (FULLRANGE=1), 8 Bits Outputs .....	2133
10-50. YCbCr to RGB Registers (FULLRANGE=0), 10 Bits Outputs .....	2134
10-51. YCbCr to RGB Registers (FULLRANGE=1), 10 Bits Outputs .....	2134
10-52. Averaging of the Chrominance Formula.....	2134
10-53. Averaging of the Chrominance Representation .....	2134
10-54. YUV4:2:2 to RGB30 Using Averaging of the Chrominance.....	2135
10-55. YUV4:2:0 to RGB30 Using Scaler Unit for Resampling Chrominance .....	2135
10-56. YUV4:2:2 to RGB30 Using Scaler Unit for Resampling Chrominance .....	2135
10-57. Video Upsampling .....	2136
10-58. Macro-architecture of the Vertical Scaling for A, R, G, B and Y Component.....	2138
10-59. Macro-architecture of the Horizontal Scaling for A, R, G, B and Y Component .....	2138
10-60. Macro-architecture of the Vertical Scaling for Cr and Cb Component .....	2138
10-61. Macro-architecture of the Horizontal Scaling for Cr and Cb Component .....	2139
10-62. Vertical Up/Downsampling Algorithm .....	2142
10-63. Horizontal Up/Downsampling Algorithm .....	2143
10-64. Write-back Pipeline.....	2145
10-65. RGB to YCbCr (FULLRANGE=0) .....	2145
10-66. RGB to YCbCr (FULLRANGE=1) .....	2146
10-67. Macro-architecture of the Vertical Scaling for A, R, G, B and Y Component.....	2146
10-68. Macro-architecture of the Horizontal Scaling for A, R, G, B and Y Component .....	2146
10-69. Macro-architecture of the Vertical Scaling for Cr and Cb Component .....	2147
10-70. Macro-architecture of the Horizontal Scaling for Cr and Cb Component .....	2147
10-71. LCD Output Architecture .....	2151
10-72. Priority Rule Architecture .....	2153
10-73. Example of Priority Rule: From Lower to Higher VID1, VID2, VID3, GFX .....	2154
10-74. Alpha Blending Architecture with Premultiplied Alpha Support.....	2155
10-75. Source Transparency Color Key Example .....	2157
10-76. Destination Transparency Color Key Example .....	2158
10-77. Data Memory Organization for Gamma Mode in LCD output .....	2159
10-78. Color Phase Rotation Matrix .....	2159
10-79. Color Phase Rotation Macro Architecture.....	2160
10-80. 8-bit Interface Settings .....	2162
10-81. 9-bit Interface Settings .....	2163
10-82. 12-bit Interface Settings .....	2164
10-83. 16-bit Interface Settings .....	2165
10-84. Timing Values (Active Matrix Display) .....	2166
10-85. RFBI Data Stall Signal Diagram .....	2167
10-86. RFBI Data Stall Signal Diagram With Handcheck .....	2167
10-87. STALL signal .....	2168
10-88. TV Output Architecture .....	2169
10-89. Data Memory Organization for Gamma Mode in TV output .....	2169
10-90. TV Formats.....	2170
10-91. Video Pipeline Processing Configuration.....	2181
10-92. Video Pipeline Processing Configuration.....	2184
10-93. Scaler Unit Programming Flow .....	2186
10-94. DSI Overview .....	2330
10-95. DSI Environment.....	2332
10-96. DSI Transmitter/Receiver Data Flow.....	2333

10-97. Protocol Layer With Short and Long Packets.....	2336
10-98. Short Packet Structure.....	2337
10-99. Long Packet Structure.....	2337
10-100. Data Identifier Structure.....	2339
10-101. VC Controller.....	2339
10-102. DSI Video Mode: Nonburst Transfer With VE and HE.....	2342
10-103. DSI Video Mode: Nonburst Transfer Without VE and HE.....	2342
10-104. DSI Video Mode: Burst Transfer Without VE and HE.....	2343
10-105. DSI General Frame Structure.....	2344
10-106. DSI General Frame Structure Using Burst Mode.....	2345
10-107. DSI General Frame Structure Using Burst Mode and Interleaving.....	2346
10-108. Four-Data Lane Configuration.....	2348
10-109. Three-Data Lane Configuration.....	2349
10-110. Two-Data Lane Configuration.....	2350
10-111. One-Data Lane Configuration.....	2350
10-112. Two Packets Using Two-Data Lane Configuration (Example).....	2350
10-113. 24-bpp RGB Color Format, Long Packet (Video Mode).....	2352
10-114. 24-bpp RGB Color Format, Long Packet (Command Mode).....	2353
10-115. 18-bpp (Loosely Packed) RGB Color Format, Long Packet (Transparent for DSI Protocol Engine) (Video Mode).....	2354
10-116. 18-bpp (Loosely Packed) RGB Color Format, Long Packet (Command Mode).....	2355
10-117. 18-bpp (Packed) RGB Color Format, Long Packet (Video Mode).....	2356
10-118. 16-bpp RGB Color Format, Long Packet (Video Mode).....	2357
10-119. 16-bpp RGB Color Format, Long Packet (Command Mode).....	2358
10-120. DSI Integration.....	2359
10-121. DSI Schematic.....	2362
10-122. DSI Clock Tree.....	2363
10-123. DSI Protocol Engine.....	2365
10-124. Interface Data Configuration.....	2368
10-125. HS Command Mode Interleaving.....	2374
10-126. LP Command Mode Interleaving.....	2377
10-127. LP to HS Timing.....	2379
10-128. HS to LP Timing.....	2380
10-129. DSI Video Mode Without Burst (No-Line Buffer).....	2385
10-130. DSI Video Mode Without Burst (One-Line Buffer).....	2386
10-131. DSI Video Mode With Burst (Two-Line Buffers).....	2387
10-132. Stall Timing With Pixel on Rising Edge.....	2389
10-133. Stall Timing With Pixel on Falling Edge.....	2389
10-134. Data Flow in Command Mode Using the Video Port.....	2390
10-135. Ping-Pong Mode.....	2391
10-136. ForceTxStopMode FSM.....	2393
10-137. TurnRequest FSM.....	2394
10-138. HS TX Timer FSM.....	2395
10-139. LP RX Timer FSM.....	2396
10-140. 64-Bit ECC Generation on TX Side.....	2400
10-141. Checksum Transmission.....	2401
10-142. 16 Bit CRC Generation Using a Shift Register.....	2401
10-143. Complex I/O Power FSM.....	2403
10-144. DSI PLL Power FSM.....	2404

10-145. DSI PLL HS Clock FSM .....	2405
10-146. DSI PLL Controller Overview .....	2408
10-147. DSI PLL Reference Diagram .....	2409
10-148. ADPLL M Functional Block Diagram .....	2409
10-149. Power State Diagram .....	2412
10-150. Gated Mode Sequence .....	2413
10-151. DSI PLL Programming Sequence .....	2415
10-152. DSI PLL Go Sequence (Manual Mode) .....	2416
10-153. DSI PLL Go Sequence (Automatic Mode) .....	2417
10-154. HS Clock Transmission .....	2420
10-155. HS Data Transmission .....	2422
10-156. Turnaround Request in Transmit Mode .....	2423
10-157. Turnaround Request in Receive Mode .....	2424
10-158. Power State Diagram .....	2425
10-159. HDMI Overview .....	2509
10-160. RFBI Highlight .....	2512
10-161. RFBI LCD Support Interface .....	2514
10-162. External Generation of TE Signal Based on Logical OR Operation Between HSYNC and VSYNC (Active-High) .....	2516
10-163. Command Data Write .....	2517
10-164. Display Data Read .....	2517
10-165. Read to Write and Write to Read .....	2518
10-166. RFBI Integration .....	2519
10-167. RFBI Architecture Overview .....	2521
10-168. RFBI Data Stall Signal Diagram With PCD=3 .....	2525
10-169. RFBI Data Stall Signal Diagram With Handshake With PCD=3 .....	2525
10-170. 8-Bit Interface Settings .....	2527
10-171. 9-Bit Interface Settings .....	2528
10-172. 16-Bit Interface Settings .....	2529
10-173. DMA Mode Configuration .....	2537
10-174. Trigger Mode Settings .....	2538
10-175. RFBI Output Enable Flow Chart .....	2541
10-176. FRAMEDONE Event Servicing Flow Chart .....	2543
10-177. Video Encoder Overview .....	2558
10-178. Video Encoder Environment, Normal Mode DC-Coupling .....	2560
10-179. Video Encoder Environment, Normal Mode AC-Coupling .....	2561
10-180. Video Encoder Environment, Bypass Mode .....	2561
10-181. Video Encoder Integration .....	2563
10-182. Video Encoder Architecture Overview .....	2565
10-183. Closed Captioning Timing .....	2568
10-184. WSS Timing .....	2570
10-185. 10-Bit Video DAC Architecture .....	2571
10-186. DC-Coupling TV Detect Waveforms for TV Connected and Disconnected .....	2572
10-187. AC-Coupling TV Detect Waveforms for TV Connected and Disconnected .....	2573
10-188. GPIO Signal Waveform Proposal for TV Detection/Disconnection in DC-Coupling Mode .....	2574
10-189. GPIO Signal Waveform Proposal for TV Detection/Disconnection in AC-Coupling Mode .....	2574
10-190. DAC Test Mode in Composite Video Mode .....	2575
11-1. SGX Overview .....	2610
11-2. SGX Integration .....	2613



11-3.	SGX Block Diagram .....	2615
12-1.	Audio Back End.....	2637
13-1.	Interconnect Overview .....	2646
13-2.	L3 Connectivity Matrix .....	2651
13-3.	L3 Interconnect Overview .....	2653
13-4.	L3 Connectivity Matrix .....	2658
13-5.	L3 Interconnect Region Overlay and Priority Level Overview .....	2665
13-6.	L3 Global Error-Routing Scheme.....	2671
13-7.	Typical Error Analysis Sequence.....	2675
13-8.	L4 Interconnect Overview .....	2750
13-9.	L4 Initiator-Target Connectivity .....	2755
13-10.	Example of CONNID_BIT_VECTOR L4_AP_PROT_GROUP_MEMBERS_k .....	2757
13-11.	L4 Segmentation.....	2760
13-12.	L4 Error Reporting.....	2768
13-13.	Protection Violation Out-of-Band Error Reporting.....	2769
13-14.	Typical Error Analysis Sequence.....	2771
13-15.	Typical Error Analysis Sequence.....	2772
14-1.	C2C Implementation Mirror Mode Example .....	2816
14-2.	PRCM Overview .....	2818
14-3.	CD_C2C Overview .....	2827
14-4.	C2C L3 Interface .....	2845
14-5.	C2C Global Error Routing Scheme .....	2848
14-6.	C2C SSCM Block Diagram.....	2849
14-7.	ICR Block Diagram .....	2851
15-1.	Memory Subsystem Functional Diagram.....	2890
15-2.	DMM and Tiler Overview .....	2895
15-3.	DMM Integration .....	2896
15-4.	DMM Block Diagram .....	2899
15-5.	DMM Sections and Memory Mapping .....	2902
15-6.	PAT Direct Access Translation .....	2904
15-7.	PAT Indirect Access Translation .....	2905
15-8.	Physical Address Translation Table .....	2905
15-9.	PAT Descriptors .....	2908
15-10.	TILER Address Space Structure for Tiled Modes.....	2911
15-11.	TILER Object Containers and Views.....	2913
15-12.	TILER Memory Footprint with PAT and Shared Physical Address Translation LUT .....	2914
15-13.	Object Container Geometry with 4-kiB Pages .....	2915
15-14.	TILER Page Mapping when Using 4-kiB Pages.....	2915
15-15.	Isometric Transforms in the TILER Container .....	2916
15-16.	Page Mode Virtual Addressing .....	2917
15-17.	Tiled Mode Addressing in 0° or 180° Orientations (S=0) .....	2918
15-18.	Tiled Mode Addressing in 90° or 270° Orientations (S=1).....	2918
15-19.	Tiled Mode Ordering of Elements in Natural View .....	2919
15-20.	Page Mode Ordering of Elements in Natural View.....	2919
15-21.	Tiled Mode Ordering of Elements in 0° View with Vertical Mirror .....	2920
15-22.	Page Mode Ordering of Elements in 0° View with Vertical Mirror.....	2920
15-23.	Tiled Mode Ordering of Elements in 0° View with Horizontal Mirror .....	2921
15-24.	Page Mode Ordering of Elements in 0° View with Horizontal Mirror.....	2921
15-25.	Tiled Mode Ordering of Elements in 180° View .....	2922

15-26. Page Mode Ordering of Elements in 180° View .....	2922
15-27. Tiled Mode Ordering of Elements in 90° View with Vertical Mirror.....	2923
15-28. Page Mode Ordering of Elements in 90° View with Vertical Mirror .....	2923
15-29. Tiled Mode Ordering of Elements in 270° View .....	2924
15-30. Page Mode Ordering of Elements in 270° View .....	2924
15-31. Tiled Mode Ordering of Elements in 90° View .....	2925
15-32. Page Mode Ordering of Elements in 90° View .....	2925
15-33. Tiled Mode Ordering of Elements in 90° View with Horizontal Mirror.....	2926
15-34. Page Mode Ordering of Elements in 90° View with Horizontal Mirror .....	2926
15-35. TILER Port Address Map .....	2927
15-36. TILER Aliased View Orientation .....	2928
15-37. Simple Manual Area Refill Scheme .....	2931
15-38. Single Auto-Configured Area Refill Scheme .....	2932
15-39. Chained Auto-Configured Area Refill Scheme .....	2933
15-40. Synchronised Auto-Configured Area Refill Scheme .....	2934
15-41. Cyclic Synchronized Auto-Configured Area Refill Scheme .....	2935
15-42. Example of 8-Bit Frame-Buffer Addressing in any Orientation .....	2942
15-43. EMIF Module Overview .....	2977
15-44. EMIF Overall Architecture .....	2978
15-45. EMIF Generic Configuration .....	2979
15-46. EMIF1 Integration .....	2981
15-47. EMIF2 Integration .....	2981
15-48. EMIF Block Diagram .....	2983
15-49. FIFO Block Diagram .....	2984
15-50. GPMC to 16-Bit Address/Data-Multiplexed Memory .....	3045
15-51. GPMC to 16-Bit Nonmultiplexed Memory .....	3045
15-52. GPMC to 8-Bit NAND Device .....	3046
15-53. GPMC Integration .....	3049
15-54. GPMC Block Diagram .....	3052
15-55. Chip-Select Address Mapping and Decoding Mask .....	3058
15-56. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1) .....	3061
15-57. Wait Behavior During a Synchronous Read Burst Access.....	3063
15-58. Read to Read for an Address-Data Multiplexed Device, on Different CS, without Bus Turnaround (nCS0 attached to a fast device).....	3065
15-59. Read to Read / Write for an Address-Data Multiplexed Device, on Different CS, with Bus Turnaround ...	3065
15-60. Read to Read / Write for a Address-Data or AAD-multiplexed device, on the Same CS, with Bus Turnaround .....	3066
15-61. Asynchronous Single Read on an Address/Data-Multiplexed Device .....	3075
15-62. Two Asynchronous Single Read Accesses on an Address/Data Multiplexed Device (32-bit read split into 2 x 16-bit read).....	3076
15-63. Asynchronous Single Write on an Address/Data-Multiplexed Device .....	3077
15-64. Asynchronous Single Read on an AAD-Multiplexed Device.....	3079
15-65. Asynchronous Single Write on an AAD-Multiplexed Device .....	3080
15-66. Synchronous Single Read (GPMCFCLKDIVIDER = 0) .....	3082
15-67. Synchronous Single Read (GPMCFCLKDIVIDER = 1) .....	3083
15-68. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0).....	3085
15-69. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1).....	3086
15-70. Synchronous Single Write on an Address/Data-Multiplexed Device .....	3087
15-71. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode .....	3088
15-72. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode .....	3089

15-73. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device .....	3091
15-74. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device .....	3092
15-75. Asynchronous Multiple (Page Mode) Read .....	3093
15-76. NAND Command Latch Cycle .....	3098
15-77. NAND Address Latch Cycle.....	3099
15-78. NAND Data Read Cycle .....	3100
15-79. NAND Data Write Cycle .....	3100
15-80. Hamming Code Accumulation Algorithm (1/2).....	3105
15-81. Hamming Code Accumulation Algorithm (2/2).....	3105
15-82. ECC Computation for a 256-Byte Data Stream (Read or Write) .....	3106
15-83. ECC Computation for a 512-Byte Data Stream (Read or Write) .....	3106
15-84. 128 Word16 ECC Computation.....	3107
15-85. 256 Word16 ECC Computation.....	3108
15-86. Manual Mode Sequence and Mapping .....	3113
15-87. NAND Page Mapping and ECC: Per-Sector Schemes .....	3117
15-88. NAND Page Mapping and ECC: Pooled Spare Schemes .....	3118
15-89. NAND Page Mapping and ECC: Per-Sector Schemes, with Separate ECC .....	3119
15-90. NAND Read Cycle Optimization Timing Description .....	3125
15-91. Programming Model Top Level Diagram .....	3127
15-92. NOR Interfacing Timing Parameters Diagram .....	3133
15-93. NAND Command Latch Cycle Timing Simplified Example .....	3137
15-94. Synchronous NOR Single Read Simplified Example .....	3141
15-95. Asynchronous NOR Single Write Simplified Example .....	3143
15-96. GPMC Connection to an External NOR Flash Memory .....	3145
15-97. Synchronous Burst Read Access (Timing Parameters in Clock Cycles).....	3147
15-98. Asynchronous Single Read Access (Timing Parameters in Clock Cycles) .....	3148
15-99. Asynchronous Single Write Access (Timing Parameters in Clock Cycles) .....	3149
15-100. ELM Integration .....	3188
15-101. OCM Subsystem Integration to the Device .....	3216
16-1. sDMA Overview.....	3221
16-2. Example of External DMA Requests Use .....	3222
16-3. Edge-Sensitive DMA Request Scheme .....	3223
16-4. Transition-Sensitive DMA Request Scheme .....	3223
16-5. sDMA Controller Integration.....	3224
16-6. sDMA Controller Top-Level Block Diagram .....	3229
16-7. Example Showing Double-Index Addressing, Elements, Frames, and Strides .....	3235
16-8. Addressing Mode Example (a) .....	3235
16-9. Addressing Mode Example (b) .....	3235
16-10. Addressing Mode Example (c).....	3236
16-11. Example of a 90-Degree Clockwise Image Rotation .....	3237
16-12. 2-D Graphic Transparent Color Block Diagram .....	3244
16-13. Save and Restore Mechanism Overview Using SAR RAM and SAR ROM .....	3252
16-14. Type-3 SAR ROM Code Example.....	3253
16-15. Type-3 SAR ROM Last transfer Example .....	3254
17-1. Interrupt Controllers Overview .....	3301
17-2. Interrupts From External Devices .....	3302
18-1. Control Module Overview.....	3313
18-2. Control Module Environment.....	3314
18-3. Control Module Integration .....	3316

18-4.	Control Module Block Diagram .....	3319
18-5.	Control Module Internal Clock Paths.....	3320
18-6.	Pad Configuration Register Functionality .....	3323
18-7.	Pad Configuration Diagram .....	3325
18-8.	Off Mode Pad control Overview .....	3346
18-9.	Wake-Up Event Detection Overview.....	3347
18-10.	I/O Power Optimization Flow Chart .....	3348
18-11.	Functional Block Diagram .....	3350
18-12.	Extended-Drain I/O.....	3353
18-13.	VDDS Ramps Up Before VDD2 .....	3357
18-14.	Functional Block Diagram .....	3358
18-15.	Single Conversion Mode (CONTCONV = 0).....	3359
18-16.	Continuous Conversion Mode (CONTCONV = 1).....	3360
18-17.	Overview of the Hardware Observability Functionality .....	3362
18-18.	Configuring PBIAS Cell to Operate in Different Modes, Interrupt Method .....	3426
19-1.	Mailbox Overview.....	3876
19-2.	System Mailbox Integration .....	3878
19-3.	IVA-HD Mailbox Integration .....	3880
19-4.	Mailbox Block Diagram .....	3883
19-5.	Example of Communication.....	3887
20-1.	Device MMU Instances .....	3908
20-2.	MMU Integration .....	3909
20-3.	DSP MMU Integration.....	3910
20-4.	MMU Block Diagram .....	3912
20-5.	Translation Process .....	3913
20-6.	Translation Hierarchy .....	3914
20-7.	First-level Descriptor Address Calculation .....	3914
20-8.	Detailed First-level Descriptor Address Calculation.....	3915
20-9.	Section Translation Summary .....	3916
20-10.	Supersection Translation Summary.....	3917
20-11.	Two-Level Translation .....	3918
20-12.	Small Page Translation Summary.....	3919
20-13.	Large Page Translation Summary .....	3920
20-14.	TLB Entry Lock Mechanism.....	3921
20-15.	TLB Entry Structure .....	3922
20-16.	MMU Global Initialization .....	3926
21-1.	Spinlock Module .....	3946
21-2.	Spinlock Integration .....	3947
21-3.	Lock Register State Diagram .....	3950
21-4.	Take and Release Spinlock.....	3952
22-1.	Timers.....	3960
22-2.	GP Timers Overview .....	3961
22-3.	GP Timers External System Interface .....	3963
22-4.	GP Timer Integration.....	3965
22-5.	Block Diagram of GPTIMER3 Through GPTIMER9, and GPTIMER11 .....	3969
22-6.	Block Diagram of GPTIMER1, GPTIMER2, and GPTIMER10 .....	3970
22-7.	Wake-Up Request Generation.....	3972
22-8.	Wake-Up Request Generation.....	3974
22-9.	GPT_TCCR Timing Value.....	3975

22-10. Block Diagram of the 1-ms Tick Module.....	3976
22-11. Capture Wave Example for GPT_TCLR[13] CAPT_MODE = 0.....	3978
22-12. Capture Wave Example for GPT_TCLR[13] CAPT_MODE = 1.....	3978
22-13. Timing Diagram of PWM With GPT_TCLR[7] SCPWM Bit = 0.....	3980
22-14. Timing Diagram of PWM With GPT_TCLR[7] SCPWM Bit = 1.....	3980
22-15. Watchdog Timers Block Diagram.....	4011
22-16. Watchdog Timers Integration.....	4013
22-17. 32-Bit Watchdog Timer Functional Block Diagram.....	4016
22-18. Watchdog Timers General Functional View.....	4017
22-19. 32-kHz Synchronized Timer Block Diagram.....	4037
22-20. Reset Resynchronization Timing Diagram.....	4038
22-21. SYNCTIMER32K Block Diagram.....	4039
23-1. HS I <sup>2</sup> C Controllers.....	4045
23-2. HS I <sup>2</sup> C and Typical Connections to I <sup>2</sup> C Devices.....	4047
23-3. HS I <sup>2</sup> C Interface Signals in I <sup>2</sup> C Mode.....	4047
23-4. HS I <sup>2</sup> C Data Transfer.....	4048
23-5. HS I <sup>2</sup> C Bit Transfer on the I <sup>2</sup> C Bus.....	4048
23-6. HS I <sup>2</sup> C S and P Condition Events.....	4049
23-7. HS I <sup>2</sup> C Data Transfer Formats in F/S Mode.....	4049
23-8. HS I <sup>2</sup> C Data Transfer in HS Mode.....	4050
23-9. HS I <sup>2</sup> C Arbitration Between Master Transmitters.....	4051
23-10. HS I <sup>2</sup> C Clock Generators Synchronization.....	4051
23-11. HS I <sup>2</sup> C and Typical Connections to SCCB Devices.....	4052
23-12. HS I <sup>2</sup> C Interface Signals in SCCB Mode.....	4053
23-13. HS I <sup>2</sup> C 3-wire SCCB Transmission Timing Diagram.....	4054
23-14. HS I <sup>2</sup> C SCCB Transmission Data Formats.....	4054
23-15. HS I <sup>2</sup> C and Typical Connection Between the HS I <sup>2</sup> C and Power Chip(s).....	4056
23-16. HS I <sup>2</sup> C I2C5 Interface Signals.....	4056
23-17. HS I <sup>2</sup> C Data Transfer Format in F/S Mode for I2C5.....	4058
23-18. HS I <sup>2</sup> C Data Transfer Format in HS Mode for the I2C5 Module.....	4059
23-19. HS I <sup>2</sup> C Integration.....	4060
23-20. HS I <sup>2</sup> C I2C5 Integration.....	4061
23-21. HS I <sup>2</sup> C Block Diagram.....	4063
23-22. HS I <sup>2</sup> C Clock Generation.....	4064
23-23. HS I <sup>2</sup> C Receive FIFO Interrupt Request Generation.....	4071
23-24. HS I <sup>2</sup> C Transmit FIFO Interrupt Request Generation.....	4072
23-25. HS I <sup>2</sup> C Receive FIFO DMA Request Generation.....	4073
23-26. HS I <sup>2</sup> C Transmit FIFO Request Generation (High Threshold).....	4074
23-27. HS I <sup>2</sup> C Transmit FIFO Request Generation (Low Threshold).....	4074
23-28. HS I <sup>2</sup> C Setup Procedure.....	4080
23-29. HS I <sup>2</sup> C Master Transmitter Mode, Polling Method, in F/S and HS Modes.....	4081
23-30. HS I <sup>2</sup> C Master Receiver Mode, Polling Method, in F/S and HS Modes.....	4083
23-31. HS I <sup>2</sup> C Master Transmitter Mode, Interrupt Method, in F/S and HS Modes.....	4085
23-32. HS I <sup>2</sup> C Master Receiver Mode, Interrupt Method, in F/S and HS Modes.....	4087
23-33. HS I <sup>2</sup> C Master Transmitter Mode, DMA Method in F/S and HS Modes.....	4089
23-34. HS I <sup>2</sup> C Master Receiver Mode, DMA Method in F/S and HS Modes.....	4091
23-35. HS I <sup>2</sup> C Slave Transmitter/Receiver Mode, Polling.....	4092
23-36. HS I <sup>2</sup> C Slave Transmitter/Receiver Mode, Interrupt.....	4093
23-37. HS I <sup>2</sup> C SCCB Setup Procedure.....	4096

23-38. HS I <sup>2</sup> C SCCB Master Transmitter Mode, Polling .....	4097
23-39. HS I <sup>2</sup> C SCCB Master Receiver Mode, Polling .....	4098
23-40. HS I <sup>2</sup> C SCCB Master Transmitter Mode, Interrupt .....	4100
23-41. HS I <sup>2</sup> C SCCB Master Receiver Mode, Interrupt .....	4102
23-42. HDQ/1-Wire Overview .....	4135
23-43. HDQ/1-Wire Typical Application System Overview .....	4136
23-44. HDQ Break-Pulse Timing Diagram.....	4137
23-45. 1-Wire (SDQ) Reset Timing Diagram .....	4137
23-46. HDQ/1-Wire Transmitted Bit Timing .....	4138
23-47. HDQ/1-Wire Communication Sequence.....	4138
23-48. HDQ/1-Wire Block Diagram .....	4139
23-49. Protocol Registers Description.....	4140
23-50. HDQ/1-Wire Integration.....	4145
23-51. UART Module .....	4155
23-52. UART Mode Bus System Overview .....	4158
23-53. IrDA System Overview .....	4158
23-54. CIR System Overview .....	4159
23-55. UART Frame Data Format .....	4160
23-56. IrDA SIR Frame Format .....	4162
23-57. IrDA SIR Encoding Mechanism.....	4163
23-58. IrDA SIR Decoding Mechanism .....	4163
23-59. SIR FF Mode.....	4164
23-60. MIR Transmit Frame Format.....	4165
23-61. MIR Baud Rate Adjustment Mechanism .....	4166
23-62. SIP.....	4167
23-63. FIR Transmit Frame Format .....	4167
23-64. CIR Pulse Modulation.....	4169
23-65. CIR Modulation Duty Cycle .....	4169
23-66. RC-5 Bit Encoding.....	4170
23-67. SIRC Bit Encoding .....	4171
23-68. RC-5 Standard Packet Format .....	4171
23-69. SIRC Packet Format .....	4171
23-70. SIRC Bit Transmission Example .....	4172
23-71. UART/IrDA/CIR Integration.....	4173
23-72. UART/IrDA/CIR Functional Specification Block Diagram .....	4176
23-73. FIFO Management Registers .....	4182
23-74. Receive FIFO Interrupt Request Generation .....	4184
23-75. Transmit FIFO Interrupt Request Generation.....	4184
23-76. Receive FIFO DMA Request Generation (32 Characters).....	4186
23-77. Transmit FIFO DMA Request Generation (56 Spaces) .....	4187
23-78. Transmit FIFO DMA Request Generation (8 Spaces).....	4188
23-79. Transmit FIFO DMA Request Generation (1 Space) .....	4188
23-80. DMA Transmission .....	4189
23-81. DMA Reception .....	4189
23-82. Baud Rate Generation .....	4196
23-83. Baud Rate Generator .....	4202
23-84. CIR Mode Block Components .....	4206
23-85. Multichannel Modules SPI1, SPI2, SPI3, and SPI4.....	4265
23-86. Typical Application Using the MCSPI.....	4267

23-87. MCSPI Interface Signals in Master Mode .....	4268
23-88. MCSPI Interface Signals in Slave Mode .....	4268
23-89. Phase and Polarity Combinations .....	4270
23-90. Full-Duplex Transfer Format With PHA = 0 .....	4271
23-91. Extended SPI Transfer With a Start-Bit (SBE = 1) .....	4272
23-92. MCSPI Master Mode (Full-Duplex) .....	4272
23-93. MCSPI Master Single Mode (Receive-Only) .....	4273
23-94. MCSPI Slave Mode (Full Duplex) .....	4273
23-95. MCSPI Slave Single Mode (Transmit Only) .....	4274
23-96. MCSPI Integration .....	4274
23-97. MCSPI Block Diagram .....	4277
23-98. SPI Full-Duplex Transmission (Example) .....	4279
23-99. Continuous Transfers With mcsnim_csx Maintained Active (Single-Data-Pin Interface Mode) .....	4281
23-100. Continuous Transfers With mcsnim_csx Maintained Active (Dual-Data-Pin Interface Mode) .....	4281
23-101. CS SPIEN Timing Controls .....	4282
23-102. Example of MCSPI Slave With One Master and Multiple Slave Devices on Channel 0 .....	4285
23-103. SPI Half-Duplex Transmission (Transmit-Only Slave) .....	4287
23-104. SPI Half-Duplex Transmission (Receive-Only Slave) .....	4288
23-105. Buffer Used in Transmit Direction Only .....	4288
23-106. Buffer Used in Receive Direction Only .....	4289
23-107. Buffer Used for Transmit and Receive Directions .....	4289
23-108. Buffer Almost Full Level (AFL) .....	4290
23-109. Buffer Almost Empty Level (AEL) .....	4290
23-110. McBSP Highlight .....	4327
23-111. Mode Overview of McBSP4 Module .....	4331
23-112. Mode Overview of McBSPi Module .....	4332
23-113. Midi Data Application .....	4332
23-114. Audio Data Application .....	4333
23-115. Voice Data Application .....	4333
23-116. McBSP Reception/Transmission Signal Activity .....	4335
23-117. Serial Data Formats .....	4335
23-118. TDM Data Format; Word Width: 32 Bits; Data Length: 24 Bits .....	4336
23-119. I2S Data Format; Word Width: 32 Bits; Data Length: 24 Bits .....	4337
23-120. Left Justified Data Format; Word Width: 32 Bits; Data Length: 24 Bits .....	4337
23-121. Right Justified Data Format; Word Width: 32 Bits; Data Length: 24 Bits .....	4337
23-122. PCM Protocol - Mode 1 Data Format .....	4338
23-123. PCM Protocol - Mode 2 Data Format .....	4338
23-124. McBSP1 Integration .....	4340
23-125. McBSP2 Integration .....	4341
23-126. McBSP3 Integration .....	4342
23-127. McBSP4 Integration .....	4343
23-128. McBSP iBlock Diagram .....	4346
23-129. McBSP Data Transfer Paths .....	4356
23-130. Conceptual Block Diagram for Clock and Frame Generation .....	4357
23-131. Clock Signal Control of Bit Transfer Timing .....	4358
23-132. McBSP Operating at Maximum Packet Frequency .....	4360
23-133. Single-Phase Frame for a McBSP Data Transfer .....	4362
23-134. Dual-Phase Frame for a McBSP Data Transfer .....	4363
23-135. McBSP Reception Physical Data Path .....	4363

23-136. McBSP Reception Signal Activity .....	4363
23-137. McBSP Transmission Physical Data Path .....	4364
23-138. McBSP Transmission Signal Activity .....	4364
23-139. Transmit Full Cycle Timing Diagram .....	4366
23-140. Transmit Half Cycle Timing Diagram .....	4366
23-141. Receive Full Cycle Timing Diagram .....	4366
23-142. Receive Half Cycle Timing Diagram .....	4367
23-143. Conceptual Block Diagram of the Sample Rate Generator .....	4367
23-144. CLKG Synchronization and FSG Generation (GSYNC = 1 and CLKGDV = 0x1) .....	4371
23-145. CLKG Synchronization and FSG Generation (GSYNC = 1 and CLKGDV = 0x3) .....	4372
23-146. Overrun in the McBSP Receiver .....	4373
23-147. Unexpected Frame-sync Pulse During a McBSP Reception .....	4374
23-148. Proper Positioning of Receive Frame-sync Pulses .....	4374
23-149. Unexpected Frame-sync Pulse During a McBSP Transmission .....	4376
23-150. Proper Positioning of Transmit Frame-sync Pulses .....	4376
23-151. McBSP Data Transfer in the 8-partition Mode .....	4379
23-152. Alternating Between Partitions A and B Channels .....	4380
23-153. Activity on McBSP Pins When XMCM=0b00 .....	4382
23-154. Activity on McBSP Pins When XMCM=0b01 .....	4382
23-155. Activity on McBSP Pins When XMCM=0b10 .....	4382
23-156. Activity on McBSP Pins When XMCM=0b11 .....	4383
23-157. Flow Diagram of McBSP Initialization Procedure for Master Mode .....	4385
23-158. Flow Diagram of McBSP Initialization Procedure for Slave Mode .....	4386
23-159. Flow Diagram for the SRG Registers Programming .....	4388
23-160. McPDM Overview .....	4433
23-161. McPDM Generic Configuration .....	4435
23-162. Protocols and Data Formats in Normal Mode .....	4435
23-163. Protocols and Data Formats in Command Mode .....	4436
23-164. Protocols and Data Formats in Status Mode .....	4438
23-165. McPDM Integration .....	4440
23-166. McPDM Block Diagram .....	4442
23-167. Interrupt Tree .....	4445
23-168. Uplink Path Detailed Block Diagram .....	4447
23-169. FIFO Uplink Overview .....	4449
23-170. Downlink Path Detailed Block Diagram .....	4451
23-171. FIFO Downlink Overview .....	4452
23-172. McPDM Watchdog Logic .....	4455
23-173. McPDM Global Initialization Flow Chart .....	4457
23-174. DMIC .....	4478
23-175. DMIC With Three Stereo DMICs .....	4479
23-176. DMIC With Six Mono DMICs .....	4480
23-177. DMIC Interface Diagram With External DMICs .....	4481
23-178. DMIC Integration .....	4482
23-179. DMIC Block Diagram .....	4484
23-180. DMIC Clock Selection .....	4484
23-181. Interrupt Tree .....	4488
23-182. DMIC FIFO .....	4490
23-183. Polling Mode .....	4493
23-184. Interrupt Handler .....	4494



23-185. McASP Module .....	4510
23-186. Definition of Bit, Word, and Slot .....	4513
23-187. Bit Order and Word Alignment Within a Slot Examples .....	4513
23-188. Definition of Frame and Frame-Sync Width .....	4514
23-189. Biphas-Mark Code .....	4515
23-190. S/PDIF Subframe Format .....	4516
23-191. S/PDIF Frame Format .....	4516
23-192. McASP Integration .....	4517
23-193. McASP Block Diagram .....	4519
23-194. Transmit Clock Generator Block Diagram .....	4520
23-195. Individual Serializer and Connections Within McASP .....	4521
23-196. Transmit Format Unit .....	4522
23-197. MPU Service Time Upon Transmit DMA Event (AXEVT) .....	4527
23-198. DMA Transmit Event in an Audio Example – One Event .....	4528
23-199. Audio Mute (AMUTE) Block Diagram .....	4530
23-200. Transmit Clock Failure Detection Circuit Block Diagram .....	4532
23-201. McASP DIT Transmission Polling Method .....	4537
23-202. Subsequence – DIT Transmission Startup Procedure .....	4538
23-203. Subsequence – DIT DMA Transmission Procedure .....	4540
23-204. McASP Interrupt Events Servicing .....	4541
23-205. McASP Error Handling .....	4542
23-206. SLIMbus1 Highlight .....	4567
23-207. SLIMbus2 Highlight .....	4568
23-208. SLIMbus1 Typical Application System Overview .....	4570
23-209. SLIMbus2 Typical Application System Overview .....	4570
23-210. SLIMbus Frame Structure Overview .....	4572
23-211. Segment Organization .....	4573
23-212. SLIMBUS1 Integration .....	4576
23-213. SLIMBUS2 Integration .....	4577
23-214. SLIMbus Block Diagram .....	4581
23-215. Example of FIFO Mapping in Shared Buffer .....	4589
23-216. Transmitting Message, Polling Mode Procedure .....	4620
23-217. Receiving Message, Polling Mode Procedure .....	4622
23-218. Transmitting message, interrupt mode procedure .....	4624
23-219. Receiving message, interrupt mode procedure .....	4625
23-220. HSI Overview .....	4690
23-221. HSI Typical Application .....	4692
23-222. HSI Integration .....	4693
23-223. HSI Controller Architecture Overview .....	4696
23-224. Receive FSM .....	4704
23-225. HSR_SWAPBUFFER_Pp_CHN_i Register .....	4706
23-226. Transmit FSM .....	4708
23-227. MAPPING Register .....	4710
23-228. HSI_Pp_WAKE Control in Case of Managed Wake .....	4711
23-229. Overall FIFO Architecture .....	4713
23-230. Transmission FIFO Architecture .....	4714
23-231. Reception FIFO Architecture .....	4715
23-232. HSI Transmit in Polling Method .....	4721
23-233. Event Servicing in HSI Transmission Mode .....	4723

23-234. Event Servicing in HSI Reception Mode .....	4725
23-235. HS USB Host Controller.....	4766
23-236. HS Multiport USB Host Subsystem Environment.....	4768
23-237. External USB Transceiver ULPs .....	4769
23-238. ULPI TLL Interfaces .....	4770
23-239. ULPs - 12-Pin/8-Bit Data SDR Version.....	4771
23-240. ULPI TLL Interfaces - 12-Pin/8-Bit Data SDR Version .....	4773
23-241. Serial Interface Sideband Integration - Transceiver Configuration .....	4778
23-242. Serial Interface Sideband Integration - TLL Configuration.....	4778
23-243. 6-Pin Unidirectional Using DAT/SE0 Signaling .....	4779
23-244. 6-Pin Unidirectional Using DP/DM Signaling .....	4780
23-245. 3-Pin Bidirectional Using DAT/SE0 Signaling .....	4781
23-246. 4-Pin Bidirectional Using DP/DM Signaling.....	4782
23-247. 6-Pin Unidirectional TLL Using DAT/SE0 Signaling .....	4783
23-248. 6-Pin Unidirectional TLL Using DP/DM Signaling.....	4784
23-249. 3-Pin Bidirectional TLL Using DAT/SE0 Signaling.....	4785
23-250. 4-Pin Bidirectional TLL Using DP/DM Signaling .....	4786
23-251. 2-Pin Bidirectional TLL Using DP/DM Encoding, With 4-Pin Bidirectional USB Device .....	4787
23-252. 2-Pin Bidirectional TLL Using DAT/SE0 Encoding, With 3-Pin Bidirectional USB Device.....	4788
23-253. HSIC Interface.....	4790
23-254. HS USB Subsystem Integration.....	4792
23-255. HS USB Host Controller Architecture .....	4795
23-256. Per-Configuration Data Path Through USBTLL.....	4801
23-257. Selecting and Configuring HS USB Host Subsystem Connectivity.....	4807
23-258. High-Speed USB OTG Controller Highlight.....	4889
23-259. High-Speed USB Controller Typical Application System .....	4891
23-260. High-Speed USB Controller Integration .....	4893
23-261. High-Speed USB OTG Controller.....	4898
23-262. High-Speed USB OTG Controller Endianness .....	4900
23-263. FSUSB Host Controller Overview.....	4930
23-264. Overview of FSUSB Host Controller Environment.....	4932
23-265. FSUSB Host Controller Integration .....	4933
23-266. 2-Pin Bidirectional Interface Using DP/DM Encoding With a Standard 4-Pin Bidirectional USB Device ..	4936
23-267. 2-Pin Bidirectional Interface Using DP/DM Encoding With a Specific 2-Pin Bidirectional USB Device ...	4937
24-1. MMC/SD/SDIO1 Overview (where i = 1) .....	4959
24-2. MMC/SD/SDIO2 Overview (where i = 2) .....	4960
24-3. MMC/SD/SDIO3 Through MMC/SD/SDIO5 Overview (where i = 3, 4, or 5) .....	4961
24-4. MMC/SD/SDIOi Controller Connected to an MMC, an SD, or an SDIO Card Without External Transceiver (where i = 1 through 5) .....	4964
24-5. MMC/SD/SDIOi Controller Connected to an MMC, an SD, or an SDIO Card with External Transceiver (where i = 2).....	4965
24-6. Sequential Read Operation (MMC Cards Only) .....	4967
24-7. Sequential Write Operation (MMC Cards Only) .....	4967
24-8. Multiple Block Read Operation .....	4967
24-9. Multiple Block Write Operation With Card Busy Signal.....	4968
24-10. Command Token Format .....	4968
24-11. Response Token Format (R1, R3, R4, R5, R6, R7) .....	4969
24-12. Response Token Format (R2) .....	4969
24-13. Data Token Format for 1-Bit Transfers .....	4970

24-14. Data Token Format for 4-Bit Transfers .....	4970
24-15. Data Token Format for 8-Bit Transfers .....	4971
24-16. MMC/SD/SDIO Controllers Integration .....	4972
24-17. MMC/SD/SDIO Diagram.....	4975
24-18. ADMA Block Diagram Overview.....	4984
24-19. ADMA Finite State-Machine.....	4986
24-20. DMA Receive Mode .....	4988
24-21. DMA Transmit Mode .....	4989
24-22. Buffer Management for a Write .....	4991
24-23. Buffer Management for a Read.....	4992
24-24. Busy Time-Out for R1b, R5b Response Type .....	4995
24-25. Busy Time-Out After Write CRC Status .....	4995
24-26. Write CRC Status Time-Out .....	4996
24-27. Read Data Time-Out.....	4996
24-28. Boot Acknowledge Time-Out When Using CMD0 .....	4997
24-29. Boot Acknowledge Time-Out When CMD Line Tied to 0 .....	4997
24-30. ACMD12 Timings During Write Transfer .....	4998
24-31. ACMD12 Timings During Read Transfer .....	4998
24-32. Output Driven on Falling Edge.....	5000
24-33. Output Driven on Rising Edge .....	5001
24-34. Boot Mode Using the CMD0 Timing Diagram .....	5001
24-35. Boot Mode With CMD Line Tied to 0 Timing Diagram .....	5002
24-36. MMC/SD/SDIO Controller Software Reset Flow .....	5005
24-37. MMC/SD/SDIO Controller Bus Configuration.....	5006
24-38. MMC/SD/SDIO Controller Card Identification and Selection – Part 1 .....	5008
24-39. MMC/SD/SDIO Controller Card Identification and Selection – Part 2 .....	5009
24-40. MMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Slave Mode With interrupt.....	5010
24-41. MMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Mode With Polling .....	5011
24-42. MMC/SD/SDIO Controller Read/Write Transfer Flow Without DMA With Polling.....	5013
24-43. MMC/SD/SDIO Controller Read/Write in CE-ATA Mode.....	5014
24-44. MMC/SD/SDIO Controller Suspend Flow .....	5017
24-45. MMC/SD/SDIO Controller Resume Flow.....	5018
24-46. MMC/SD/SDIO Controller Command Transfer Flow With Polling .....	5020
24-47. MMC/SD/SDIO Controller Command Transfer Flow With interrupts .....	5021
24-48. MMC/SD/SDIO Controller Clock Frequency Change Flow .....	5022
24-49. MMC/SD/SDIO Controller Bus Width Configuration Flow.....	5023
24-50. MMC/SD/SDIO Power Switching Procedure.....	5025
24-51. MMC/SD/SDIO Controller Boot Using CMD0.....	5026
24-52. MMC/SD/SDIO Controller Boot With CMD Line Tied to 0 .....	5028
25-1. General-Purpose Interface Overview .....	5089
25-2. General-Purpose Interface Typical Application.....	5091
25-3. General-Purpose Interface Used as a Keyboard Interface .....	5092
25-4. GPIO1 Signal Connections.....	5093
25-5. GPIO2 through GPIO6 Signal Connections.....	5094
25-6. GPIO Integration.....	5096
25-7. General-Purpose Interface Description .....	5099
25-8. Synchronous Path.....	5099
25-9. Asynchronous Path.....	5100
25-10. Interrupt Request Generation .....	5101

25-11. Wake-Up Request Generation .....	5102
25-12. Wake-Up Event Conditions .....	5103
25-13. GPIO_CLEARDATAOUT Register Example .....	5113
25-14. Write in GPIO_IRQSTATUS_SET_0 Register Example .....	5114
26-1. Keyboard Controller Overview .....	5142
26-2. Typical Keyboard Environment .....	5144
26-3. Multikey Limitation Example .....	5145
26-4. Keyboard Controller Integration .....	5147
26-5. Keyboard Controller Block Diagram .....	5149
26-6. Functional Modes and Related Interrupt Events .....	5155
26-7. Key Coding Registers .....	5156
27-1. Initialization Process .....	5181
27-2. Power Supply Connections .....	5183
27-3. Clock, Reset and Control Environment .....	5188
27-4. ROM Code Architecture .....	5199
27-5. ROM Memory Map .....	5200
27-6. RAM Memory Map .....	5201
27-7. Overall Booting Sequence .....	5204
27-8. ROM Code Multiprocessor Start-Up Sequence .....	5206
27-9. Device List Setup .....	5208
27-10. Synchronization Phase for UART .....	5212
27-11. Synchronization Phase for USB .....	5212
27-12. Peripheral Booting Procedure .....	5215
27-13. USB Initialization Procedure .....	5217
27-14. PrevStartOk Mechanism .....	5218
27-15. Charger Detection .....	5219
27-16. Customer USB Descriptor Selection .....	5223
27-17. USB System Using Internal Transceiver .....	5224
27-18. Fast External Boot Procedure .....	5226
27-19. Memory Booting Procedure .....	5228
27-20. Image Shadowing on GP Device .....	5229
27-21. NAND Device Detection .....	5234
27-22. Bad NAND Invalid Block Detection .....	5235
27-23. ECC Data Mapping for 2-KB Page and 8b BCH Encoding .....	5236
27-24. ECC Data Mapping for 4-KB Page and 16b BCH Encoding .....	5237
27-25. OneNAND/Flex-OneNAND Read Sector .....	5239
27-26. eMMC/eSD and MMC/SD Booting .....	5241
27-27. eMMC/eSD Connection .....	5243
27-28. MMC/SD Card Connection .....	5244
27-29. MMC/SD Detection Procedure .....	5245
27-30. SD/MMC Get Booting File .....	5247
27-31. MBR Detection Procedure .....	5249
27-32. MBR, Get Partition .....	5250
27-33. EMIF LPDDR2-NVM Detection and Initialization .....	5253
27-34. Image Formats .....	5255
27-35. CH Format .....	5256
28-1. ICEPick Scan Chain .....	5276
28-2. Processor Trace Flow .....	5288
28-3. System Instrumentation Framework .....	5289

## List of Tables

1-1.	Device Identification Registers .....	269
1-2.	STD_FUSE_DIE_ID .....	269
1-3.	ID_CODE .....	269
1-4.	Hawkeye Number Values .....	269
1-5.	Revision Number Values .....	269
1-6.	ID_CODE Values .....	269
1-7.	STD_FUSE_PROD_ID_0 .....	270
1-8.	Register Call Summary for Register STD_FUSE_PROD_ID_0 .....	270
1-9.	STD_FUSE_PROD_ID_1 .....	270
1-10.	Register Call Summary for Register STD_FUSE_PROD_ID_1 .....	270
2-1.	Global Memory Space Mapping .....	275
2-2.	L3_EMU Memory Space Mapping .....	276
2-3.	L4_CFG Memory Space Mapping .....	277
2-4.	L4_WKUP Peripheral Space Mapping .....	279
2-5.	L4_PER Memory Space Mapping .....	280
2-6.	ABE Cortex-A9 MPU Memory Space Mapping .....	283
2-7.	ABE L3 Memory Space Mapping .....	284
2-8.	ABE DSP Memory Space Mapping .....	285
2-9.	Dual Cortex-M3 Memory Space Mapping .....	286
2-10.	DSP Subsystem Memory Space Mapping .....	287
2-11.	L3 Access – Display Subsystem Space Mapping .....	289
2-12.	L4_PER Access – Display Subsystem Space Mapping .....	289
3-1.	Master Module Standby Mode Settings .....	294
3-2.	Master Module Standby Status .....	294
3-3.	Master Module Clock Enabling Conditions .....	294
3-4.	Module Idle Mode Settings .....	295
3-5.	Slave Module Idle Status .....	295
3-6.	Slave Module Clock Activity Settings .....	296
3-7.	Slave Module Mode Settings in PRCM .....	297
3-8.	Slave Module Interface Clock Enabling Conditions .....	297
3-9.	Slave Module Functional Clock Enabling Conditions .....	297
3-10.	Clock Domain Functional Clock States .....	299
3-11.	Clock Domain Clock States .....	299
3-12.	Clock Domain Clock Transition Mode Settings .....	300
3-13.	Clock Domain Wake-Up Conditions .....	303
3-14.	Clock Domain Sleep Conditions .....	304
3-15.	States of a Logic Area in a Power Domain .....	307
3-16.	States of a Memory Area in a Power Domain .....	308
3-17.	Power Domain Wake-Up Conditions .....	310
3-18.	Power Domain Sleep Conditions .....	310
3-19.	Power Domain Control and Status Registers .....	310
3-20.	States of Logic Voltage Domain .....	311
3-21.	External Clock Signals .....	322
3-22.	External Reset Signals .....	323
3-23.	Power Control Interface .....	323
3-24.	Voltage Sources .....	324
3-25.	PMFW Device-Level Layout .....	325

3-26.	PMFW Module Power Domains .....	328
3-27.	PMFW Module Reset Signals .....	328
3-28.	PMFW Module Interrupts .....	328
3-29.	Global Reset Sources .....	331
3-30.	Local Reset Sources .....	332
3-31.	Modules, Power Domains and Reset Domains Association .....	332
3-32.	Reset Sources For the Reset Domains .....	335
3-33.	Reset Domains Attributes .....	345
3-34.	PRM Clock Division and Muxing Control .....	365
3-35.	CM1 Clock Division and Muxing Control.....	367
3-36.	CM1_USB Clock Division and Muxing Control.....	368
3-37.	CM1_ABE Clock Division and Muxing Control.....	369
3-38.	CM2 Clock Division and Muxing Control.....	372
3-39.	CLKOUT_M2, CLKOUTX2_M2, and CLKOUTX2_M3Frequencies With DPLL State .....	374
3-40.	CLKOUTX2_Mn Frequencies With DPLL State .....	375
3-41.	DPLL Power Modes .....	376
3-42.	DPLL Recalibration Control Parameters .....	377
3-43.	DPLL SSC Control Parameters .....	378
3-44.	DPLL Power-Down Control Parameters .....	379
3-45.	DPLL_PER Clock Synthesis Parameters .....	379
3-46.	DPLL_PER Clock Output Parameters .....	379
3-47.	DPLL_PER Modes.....	381
3-48.	DPLL_PER Mode Control Parameters .....	381
3-49.	DPLL_PER Recalibration Feature Parameters .....	381
3-50.	DPLL_PER Spread-Spectrum Clocking Feature Parameters.....	381
3-51.	DPLL_PER Clock Output Power-down Parameters.....	381
3-52.	DPLL_CORE Clock Synthesis Parameters.....	382
3-53.	DPLL_CORE Clock Output Parameters .....	382
3-54.	DPLL_CORE Modes.....	383
3-55.	DPLL_CORE Mode Control Parameters.....	384
3-56.	DPLL_CORE Recalibration Feature Parameters .....	384
3-57.	DPLL_CORE Spread-Spectrum Clocking Feature Parameters .....	384
3-58.	DPLL_CORE Clock Output Power-Down Parameters .....	385
3-59.	DPLL_ABE Clock Synthesis Parameters .....	385
3-60.	DPLL_ABE Clock Output Parameters.....	386
3-61.	DPLL_ABE Modes .....	386
3-62.	DPLL_ABE Mode Control Parameters .....	386
3-63.	DPLL_ABE Recalibration Feature Parameters .....	386
3-64.	DPLL_ABE Spread-Spectrum Clocking Feature Parameters.....	387
3-65.	DPLL_MPU Clock Synthesis Parameters .....	387
3-66.	DPLL_MPU Clock Output Parameters .....	387
3-67.	DPLL_MPU Modes .....	388
3-68.	DPLL_MPU Mode Control Parameters .....	388
3-69.	DPLL_MPU Recalibration Feature Parameters.....	388
3-70.	DPLL_MPU Spread-Spectrum Clocking Feature Parameters .....	388
3-71.	DPLL_IVA Clock Synthesis Parameters .....	389
3-72.	DPLL_IVA Clock Output Parameters.....	389
3-73.	DPLL_IVA Modes .....	390
3-74.	DPLL_IVA Mode Control Parameters .....	390

3-75.	DPLL_IVA Recalibration Feature Parameters .....	390
3-76.	DPLL_IVA Spread-Spectrum Clocking Feature Parameters .....	390
3-77.	DPLL_IVA Clock Output Power-down Parameters .....	391
3-78.	DPLL_USB Clock Synthesis Parameters .....	391
3-79.	DPLL_USB Clock Output Parameters .....	391
3-80.	DPLL_USB Modes .....	392
3-81.	DPLL_USB Mode Control Parameters .....	392
3-82.	DPLL_USB Spread-Spectrum Clocking Feature Parameters .....	392
3-83.	CD_WKUP Clock Domain Modes .....	394
3-84.	CD_WKUP Control and Status Parameters .....	394
3-85.	CD_WKUP Wake-Up Dependency Association Parameters .....	395
3-86.	CD_WKUP Modules Clocks Association .....	395
3-87.	CD_WKUP Modules Wake-Up Request .....	396
3-88.	CD_WKUP Modules Clock-Management Modes and Control .....	396
3-89.	CD_WKUP Modules Slave Clock-Management Modes and Control .....	396
3-90.	CD_DSP Clock Domain Modes .....	397
3-91.	CD_DSP Control and Status Parameters .....	397
3-92.	CD_DSP Static Dependency Association Parameters .....	398
3-93.	CD_DSP Dynamic Dependency Association Parameters .....	398
3-94.	CD_DSP Modules Clocks Association .....	398
3-95.	CD_DSP Modules Wake-Up Request .....	399
3-96.	CD_DSP Modules Clock-Management Modes and Control .....	399
3-97.	CD_DSP Modules Slave Clock-Management Modes and Control .....	399
3-98.	CD_STD_EFUSE Modules Clocks Association .....	400
3-99.	CD_STD_EFUSE Modules Wake-Up Request .....	400
3-100.	CD_CORTEXA9 Clock Domain Modes .....	400
3-101.	CD_CORTEXA9 Control and Status Parameters .....	401
3-102.	CD_CORTEXA9 Static Dependency Association Parameters .....	401
3-103.	CD_CORTEXA9 Dynamic Dependency Association Parameters .....	402
3-104.	CD_CORTEXA9 Modules Clocks Association .....	402
3-105.	CD_CORTEXA9 Modules Wake-Up Request .....	402
3-106.	CD_CORTEXA9 Modules Clock-Management Modes and Control .....	402
3-107.	CD_CORTEXA9 Modules Slave Clock-Management Modes and Control .....	402
3-108.	CD_L4_PER Clock Domain Modes .....	403
3-109.	CD_L4_PER Control and Status Parameters .....	403
3-110.	CD_L4_PER Dynamic Dependency Association Parameters .....	404
3-111.	CD_L4_PER Wake-Up Dependency Association Parameters .....	404
3-112.	CD_L4_PER Modules Clocks Association .....	409
3-113.	CD_L4_PER Modules Wake-Up Request .....	410
3-114.	CD_L4_PER Modules Clock-Management Modes and Control .....	411
3-115.	CD_L4_PER Modules Slave Clock-Management Modes and Control .....	412
3-116.	CD_L3_INIT Clock Domain Modes .....	415
3-117.	CD_L3_INIT Control and Status Parameters .....	415
3-118.	CD_L3_INIT Static Dependency Association Parameters .....	416
3-119.	CD_L3_INIT Dynamic Dependency Association Parameters .....	416
3-120.	CD_L3_INIT Wake-Up Dependency Association Parameters .....	417
3-121.	CD_L3_INIT Modules Clocks Association .....	418
3-122.	CD_L3_INIT Modules Wake-Up Request .....	419
3-123.	CD_L3_INIT Modules Clock-Management Modes and Control .....	419

3-124. CD_L3_INIT Modules Slave Clock-Management Modes and Control .....	420
3-125. CD_IVAHD Clock Domain Modes .....	421
3-126. CD_IVAHD Control and Status Parameters .....	421
3-127. CD_IVAHD Static Dependency Association Parameters .....	421
3-128. CD_IVAHD Dynamic Dependency Association Parameters .....	422
3-129. CD_IVAHD Modules Clocks Association .....	422
3-130. CD_IVAHD Modules Wake-Up Request .....	422
3-131. CD_IVAHD Modules Clock-Management Modes and Control .....	422
3-132. CD_IVAHD Modules Slave Clock-Management Modes and Control .....	422
3-133. CD_SGX Clock Domain Modes .....	423
3-134. CD_SGX Control and Status Parameters .....	423
3-135. CD_SGX Static Dependency Association Parameters .....	423
3-136. CD_SGX Dynamic Dependency Association Parameters .....	424
3-137. CD_SGX Modules Clocks Association .....	424
3-138. CD_SGX Modules Wake-Up Request .....	424
3-139. CD_SGX Modules Clock-Management Modes and Control .....	424
3-140. CD_SGX Modules Slave Clock-Management Modes and Control .....	424
3-141. CD_EMU Clock Division and Muxing Control .....	425
3-142. CD_EMU Clock Domain Modes .....	426
3-143. CD_EMU Control and Status Parameters .....	426
3-144. CD_EMU Dynamic Dependency Association Parameters .....	426
3-145. CD_EMU Modules Clocks Association .....	426
3-146. CD_EMU Modules Wake-Up Request .....	426
3-147. CD_DSS Clock Domain Modes .....	427
3-148. CD_DSS Control and Status Parameters .....	427
3-149. CD_DSS Static Dependency Association Parameters .....	427
3-150. CD_DSS Dynamic Dependency Association Parameters .....	428
3-151. CD_DSS Wake-Up Dependency Association Parameters .....	428
3-152. CD_DSS Modules Clocks Association .....	429
3-153. CD_DSS Modules Wake-Up Request .....	430
3-154. CD_DSS Modules Clock-Management Modes and Control .....	430
3-155. CD_DSS Modules Slave Clock-Management Modes and Control .....	430
3-156. CD_L4_CFG Clock Domain Modes .....	430
3-157. CD_L4_CFG Control and Status Parameters .....	431
3-158. CD_L4_CFG Dynamic Dependency Association Parameters .....	431
3-159. CD_L4_CFG Modules Clocks Association .....	431
3-160. CD_L4_CFG Modules Wake-Up Request .....	432
3-161. CD_L4_CFG Modules Clock-Management Modes and Control .....	432
3-162. CD_L4_CFG Modules Slave Clock-Management Modes and Control .....	432
3-163. CD_L3_INSTR Clock Domain Modes .....	433
3-164. CD_L3_INSTR Control and Status Parameters .....	433
3-165. CD_L3_INSTR Modules Clocks Association .....	433
3-166. CD_L3_INSTR Modules Wake-Up Request .....	434
3-167. CD_L3_INSTR Modules Clock-Management Modes and Control .....	434
3-168. CD_L3_INSTR Modules Slave Clock-Management Modes and Control .....	434
3-169. CD_L3_2 Clock Domain Modes .....	435
3-170. CD_L3_2 Control and Status Parameters .....	435
3-171. CD_L3_2 Dynamic Dependency Association Parameters .....	435
3-172. CD_L3_2 Modules Clocks Association .....	436



3-173. CD_L3_2 Modules Wake-Up Request .....	436
3-174. CD_L3_2 Modules Clock-Management Modes and Control .....	436
3-175. CD_L3_2 Modules Slave Clock Management Modes and Control.....	436
3-176. CD_L3_1 Clock Domain Modes .....	437
3-177. CD_L3_1 Control and Status Parameters .....	437
3-178. CD_L3_1 Dynamic Dependency Association Parameters .....	437
3-179. CD_L3_1 Modules Clocks Association.....	438
3-180. CD_L3_1 Modules Wake-Up Request .....	438
3-181. CD_L3_1 Modules Clock Management Modes and Control .....	438
3-182. CD_L3_1 Modules Slave Clock Management Modes and Control.....	438
3-183. CD_EMIF Clock Domain Modes .....	439
3-184. CD_EMIF Control and Status Parameters.....	439
3-185. CD_EMIF Modules Clocks Association .....	439
3-186. CD_EMIF Modules Wake-Up Request.....	440
3-187. CD_EMIF Modules Clock-Management Modes and Control.....	440
3-188. CD_EMIF Modules Slave Clock-Management Modes and Control .....	440
3-189. CD_CORTEXM3 Clock Domain Modes.....	441
3-190. CD_CORTEXM3 Control and Status Parameters .....	441
3-191. CD_CORTEXM3 Static Dependency Association Parameters .....	441
3-192. CD_CORTEXM3 Dynamic Dependency Association Parameters .....	442
3-193. CD_CORTEXM3 Modules Clocks Association.....	442
3-194. CD_CORTEXM3 Modules Wake-Up Request .....	442
3-195. CD_CORTEXM3 Modules Clock-Management Modes and Control .....	443
3-196. CD_CORTEXM3 Modules Slave Clock-Management Modes and Control.....	443
3-197. CD_DMA Clock Domain Modes.....	443
3-198. CD_DMA Control and Status Parameters .....	444
3-199. CD_DMA Modules Clocks Association.....	444
3-200. CD_DMA Modules Wake-Up Request .....	444
3-201. CD_DMA Modules Clock-Management Modes and Control .....	444
3-202. CD_DMA Modules Slave Clock Management Modes and Control.....	444
3-203. CD_C2C Clock Domain Modes .....	445
3-204. CD_C2C Control and Status Parameters.....	445
3-205. CD_C2C Static Dependency Association Parameters.....	445
3-206. CD_C2C Dynamic Dependency Association Parameters .....	446
3-207. CD_C2C Modules Clocks Association .....	446
3-208. CD_C2C Modules Wake-Up Request.....	446
3-209. CD_C2C Modules Clock-Management Modes and Control .....	446
3-210. CD_C2C Modules Slave Clock-Management Modes and Control .....	447
3-211. CD_CAM Clock Domain Modes.....	447
3-212. CD_CAM Control and Status Parameters .....	447
3-213. CD_CAM Static Dependency Association Parameters .....	448
3-214. CD_CAM Dynamic Dependency Association Parameters .....	448
3-215. CD_CAM Modules Clocks Association.....	448
3-216. CD_CAM Modules Wake-Up Request .....	448
3-217. CD_CAM Modules Clock-Management Modes and Control .....	449
3-218. CD_CAM Modules Slave Clock-Management Modes and Control.....	449
3-219. CD_ABE Clock Domain Modes .....	450
3-220. CD_ABE Control and Status Parameters.....	450
3-221. CD_ABE Wake-Up Dependency Association Parameters .....	450

3-222. CD_ABE Modules Clocks Association .....	452
3-223. CD_ABE Modules Wake-Up Request.....	453
3-224. CD_ABE Modules Clock-Management Modes and Control .....	454
3-225. CD_ABE Modules Slave Clock-Management Modes and Control .....	454
3-226. CD_L4_ALWON_CORE Wake-Up Dependency Association Parameters .....	456
3-227. CD_L4_ALWON_CORE Modules Clocks Association .....	456
3-228. CD_L4_ALWON_CORE Modules Wake-Up Request .....	456
3-229. CD_L4_ALWON_CORE Modules Clock-Management Modes and Control .....	457
3-230. CD_L4_ALWON_CORE Modules Slave Clock-Management Modes and Control .....	457
3-231. PD_WKUP Modules Power Attributes .....	457
3-232. PD_DSP Modules Power Attributes .....	458
3-233. PD_DSP Logic Area Power Modes.....	458
3-234. PD_DSP Memory Area Power Modes .....	459
3-235. PD_DSP Power Modes Control Parameters .....	459
3-236. PD_DSP Power Modes Status Parameters .....	459
3-237. PD_STD_EFUSE Modules Power Attributes.....	460
3-238. PD_MPU Modules Power Attributes.....	460
3-239. PD_MPU Logic Area Power Modes .....	460
3-240. PD_MPU Memory Area Power Modes .....	460
3-241. PD_MPU Power Modes Control Parameters.....	461
3-242. PD_MPU Power Modes Status Parameters.....	461
3-243. PD_L4_PER Modules Power Attributes .....	462
3-244. PD_L4_PER Logic Area Power Modes .....	463
3-245. PD_L4_PER Memory Area Power Modes.....	463
3-246. PD_L4_PER Power Modes Control Parameters.....	463
3-247. PD_L4_PER Power Modes Status Parameters.....	464
3-248. PD_L3_INIT Modules Power Attributes .....	464
3-249. PD_L3_INIT Logic Area Power Modes .....	465
3-250. PD_L3_INIT Memory Area Power Modes .....	465
3-251. PD_L3_INIT Power Modes Control Parameters .....	465
3-252. PD_L3_INIT Power Modes Status Parameters .....	466
3-253. PD_IVAHD Modules Power Attributes .....	466
3-254. PD_IVAHD Logic Area Power Modes .....	466
3-255. PD_IVAHD Memory Area Power Modes.....	467
3-256. PD_IVAHD Power Modes Control Parameters .....	467
3-257. PD_IVAHD Power Modes Status Parameters .....	467
3-258. PD_SGX Modules Power Attributes .....	468
3-259. PD_SGX Logic Area Power Modes .....	468
3-260. PD_SGX Memory Area Power Modes .....	468
3-261. PD_SGX Power Modes Control Parameters .....	469
3-262. PD_SGX Power Modes Status Parameters .....	469
3-263. PD_EMU Modules Power Attributes.....	469
3-264. PD_EMU Logic Area Power Modes .....	469
3-265. PD_EMU Memory Area Power Modes.....	470
3-266. PD_EMU Power Modes Control Parameters.....	470
3-267. PD_EMU Power Modes Status Parameters.....	470
3-268. PD_DSS Modules Power Attributes .....	470
3-269. PD_DSS Logic Area Power Modes.....	470
3-270. PD_DSS Memory Area Power Modes .....	471

3-271. PD_DSS Power Modes Control Parameters .....	471
3-272. PD_DSS Power Modes Status Parameters .....	471
3-273. PD_CORE Modules Power Attributes .....	472
3-274. PD_CORE Logic Area Power Modes .....	473
3-275. PD_CORE Memory Area Power Modes .....	473
3-276. PD_CORE Power Modes Control Parameters .....	474
3-277. PD_CORE Power Modes Status Parameters .....	474
3-278. PD_CAM Modules Power Attributes .....	475
3-279. PD_CAM Logic Area Power Modes .....	475
3-280. PD_CAM Memory Area Power Modes .....	475
3-281. PD_CAM Power Modes Control Parameters .....	475
3-282. PD_CAM Power Modes Status Parameters .....	476
3-283. PD_AUDIO Modules Power Attributes .....	476
3-284. PD_AUDIO Logic Area Power Modes .....	477
3-285. PD_AUDIO Memory Area Power Modes .....	477
3-286. PD_AUDIO Power Modes Control Parameters .....	477
3-287. PD_AUDIO Power Modes Status Parameters .....	478
3-288. PD_ALWON_MPU Modules Power Attributes .....	478
3-289. PD_ALWON_DSP Modules Power Attributes .....	478
3-290. PD_ALWON_CORE Modules Power Attributes .....	479
3-291. SmartReflex Interrupt Events .....	486
3-292. SmartReflex Interrupt Events Enable and Status Bits .....	486
3-293. Voltage Processor Interrupt Events .....	488
3-294. Voltage Processor Interrupt Event Enable and Status Bits .....	489
3-295. Device Off Wake-Up and Hardware Restore Phases .....	502
3-296. Global Initialization of Surrounding Modules .....	507
3-297. DPLL Global Initialization .....	507
3-298. DPLL SSC Parameter Configuration .....	507
3-299. DPLL Recalibration Parameter Configuration .....	508
3-300. DPLL Synthesized Clock Parameter Configuration .....	508
3-301. DPLL Output Clock Parameter Configuration .....	508
3-302. Register Call Summary for Sequence – DPLL Output Frequency Change .....	511
3-303. Subprocess Call Summary for Sequence – DPLL Output Frequency Change .....	511
3-304. Global Initialization of Surrounding Modules .....	511
3-305. Clock Domain Global Initialization .....	511
3-306. Slave Module Clock-Management Parameter Configuration .....	512
3-307. Clock Domain Sleep Transition and Troubleshooting .....	512
3-308. Enable/Disable Software-Programmable Static Dependency .....	512
3-309. Power Domain Global Initialization .....	513
3-310. Forced Memory Area State Change With Power Domain ON .....	513
3-311. Forced Memory Area State Change With Power Domain On .....	514
3-312. Power Domain Global Initialization .....	514
3-313. SmartReflex Module Configuration .....	514
3-314. SmartReflex Module Clock Configuration .....	515
3-315. SmartReflex Module Sensor Configuration .....	515
3-316. Accumulator and Min/Max/Avg Configuration .....	515
3-317. Error Generator Configuration .....	515
3-318. Interrupt Generator Configuration .....	516
3-319. Voltage Processor Module Initialization .....	516

3-320. Error-to-Voltage Converter Configuration .....	516
3-321. FSM Configuration .....	517
3-322. Voltage Controller Module Initialization .....	517
3-323. VDD Channel Pointer Configuration .....	517
3-324. I <sup>2</sup> C Configuration .....	518
3-325. Changing OPP Using Voltage-Processor Module .....	518
3-326. PRM Instance Summary .....	519
3-327. INTRCONN_SOCKET_PRM Registers Mapping Summary .....	519
3-328. REVISION_PRM .....	520
3-329. Register Call Summary for Register REVISION_PRM.....	520
3-330. PRM_IRQSTATUS_MPU_A9 .....	520
3-331. Register Call Summary for Register PRM_IRQSTATUS_MPU_A9.....	523
3-332. PRM_IRQSTATUS_MPU_A9_2 .....	523
3-333. Register Call Summary for Register PRM_IRQSTATUS_MPU_A9_2.....	524
3-334. PRM_IRQENABLE_MPU_A9 .....	524
3-335. Register Call Summary for Register PRM_IRQENABLE_MPU_A9.....	527
3-336. PRM_IRQENABLE_MPU_A9_2 .....	527
3-337. Register Call Summary for Register PRM_IRQENABLE_MPU_A9_2.....	528
3-338. PRM_IRQSTATUS_MPU_M3.....	528
3-339. Register Call Summary for Register PRM_IRQSTATUS_MPU_M3 .....	531
3-340. PRM_IRQENABLE_MPU_M3.....	531
3-341. Register Call Summary for Register PRM_IRQENABLE_MPU_M3 .....	533
3-342. PRM_IRQSTATUS_DSP .....	534
3-343. Register Call Summary for Register PRM_IRQSTATUS_DSP .....	534
3-344. PRM_IRQENABLE_DSP .....	534
3-345. Register Call Summary for Register PRM_IRQENABLE_DSP .....	535
3-346. CM_PRM_PROFILING_CLKCTRL .....	535
3-347. Register Call Summary for Register CM_PRM_PROFILING_CLKCTRL .....	536
3-348. PRM_DEBUG_CFG .....	536
3-349. Register Call Summary for Register PRM_DEBUG_CFG.....	536
3-350. CKGEN_PRM Registers Mapping Summary.....	537
3-351. CM_ABE_DSS_SYS_CLKSEL.....	537
3-352. Register Call Summary for Register CM_ABE_DSS_SYS_CLKSEL .....	537
3-353. CM_L4_WKUP_CLKSEL .....	537
3-354. Register Call Summary for Register CM_L4_WKUP_CLKSEL.....	538
3-355. CM_ABE_PLL_REF_CLKSEL .....	538
3-356. Register Call Summary for Register CM_ABE_PLL_REF_CLKSEL.....	538
3-357. CM_SYS_CLKSEL.....	538
3-358. Register Call Summary for Register CM_SYS_CLKSEL .....	539
3-359. MPU_PRM Registers Mapping Summary .....	539
3-360. PM_MPU_PWRSTCTRL.....	539
3-361. Register Call Summary for Register PM_MPU_PWRSTCTRL .....	540
3-362. PM_MPU_PWRSTST .....	541
3-363. Register Call Summary for Register PM_MPU_PWRSTST .....	542
3-364. RM_MPU_RSTST .....	542
3-365. Register Call Summary for Register RM_MPU_RSTST .....	542
3-366. RM_MPU_MPU_CONTEXT .....	542
3-367. Register Call Summary for Register RM_MPU_MPU_CONTEXT .....	543
3-368. DSP_PRM Registers Mapping Summary.....	543

3-369. PM_DSP_PWRSTCTRL .....	544
3-370. Register Call Summary for Register PM_DSP_PWRSTCTRL.....	545
3-371. PM_DSP_PWRSTST .....	545
3-372. Register Call Summary for Register PM_DSP_PWRSTST .....	546
3-373. RM_DSP_RSTCTRL .....	546
3-374. Register Call Summary for Register RM_DSP_RSTCTRL.....	547
3-375. RM_DSP_RSTST .....	547
3-376. Register Call Summary for Register RM_DSP_RSTST .....	548
3-377. RM_DSP_DSP_CONTEXT .....	548
3-378. Register Call Summary for Register RM_DSP_DSP_CONTEXT .....	549
3-379. ABE_PRM Registers Mapping Summary.....	549
3-380. PM_ABE_PWRSTCTRL .....	550
3-381. Register Call Summary for Register PM_ABE_PWRSTCTRL.....	551
3-382. PM_ABE_PWRSTST .....	551
3-383. Register Call Summary for Register PM_ABE_PWRSTST .....	552
3-384. RM_ABE_AESS_CONTEXT .....	552
3-385. <b>Register Call Summary for Register RM_ABE_AESS_CONTEXT</b> .....	552
3-386. PM_ABE_PDM_WKDEP.....	553
3-387. Register Call Summary for Register PM_ABE_PDM_WKDEP.....	553
3-388. RM_ABE_PDM_CONTEXT .....	554
3-389. Register Call Summary for Register RM_ABE_PDM_CONTEXT .....	554
3-390. PM_ABE_DMIC_WKDEP .....	554
3-391. Register Call Summary for Register PM_ABE_DMIC_WKDEP .....	555
3-392. RM_ABE_DMIC_CONTEXT.....	555
3-393. Register Call Summary for Register RM_ABE_DMIC_CONTEXT .....	556
3-394. PM_ABE_MCASP_WKDEP .....	556
3-395. Register Call Summary for Register PM_ABE_MCASP_WKDEP.....	557
3-396. RM_ABE_MCASP_CONTEXT .....	557
3-397. Register Call Summary for Register RM_ABE_MCASP_CONTEXT .....	558
3-398. PM_ABE_MCBSP1_WKDEP.....	558
3-399. Register Call Summary for Register PM_ABE_MCBSP1_WKDEP .....	558
3-400. RM_ABE_MCBSP1_CONTEXT .....	559
3-401. Register Call Summary for Register RM_ABE_MCBSP1_CONTEXT .....	559
3-402. PM_ABE_MCBSP2_WKDEP.....	559
3-403. Register Call Summary for Register PM_ABE_MCBSP2_WKDEP .....	560
3-404. RM_ABE_MCBSP2_CONTEXT .....	560
3-405. Register Call Summary for Register RM_ABE_MCBSP2_CONTEXT .....	561
3-406. PM_ABE_MCBSP3_WKDEP.....	561
3-407. Register Call Summary for Register PM_ABE_MCBSP3_WKDEP .....	562
3-408. RM_ABE_MCBSP3_CONTEXT .....	562
3-409. Register Call Summary for Register RM_ABE_MCBSP3_CONTEXT .....	562
3-410. PM_ABE_SLIMBUS_WKDEP.....	563
3-411. Register Call Summary for Register PM_ABE_SLIMBUS_WKDEP .....	563
3-412. RM_ABE_SLIMBUS_CONTEXT.....	564
3-413. Register Call Summary for Register RM_ABE_SLIMBUS_CONTEXT .....	564
3-414. PM_ABE_GPTIMER5_WKDEP.....	564
3-415. Register Call Summary for Register PM_ABE_GPTIMER5_WKDEP .....	565
3-416. RM_ABE_GPTIMER5_CONTEXT .....	565
3-417. Register Call Summary for Register RM_ABE_GPTIMER5_CONTEXT .....	565

3-418. PM_ABE_GPTIMER6_WKDEP .....	566
3-419. Register Call Summary for Register PM_ABE_GPTIMER6_WKDEP .....	566
3-420. RM_ABE_GPTIMER6_CONTEXT .....	566
3-421. Register Call Summary for Register RM_ABE_GPTIMER6_CONTEXT .....	567
3-422. PM_ABE_GPTIMER7_WKDEP .....	567
3-423. Register Call Summary for Register PM_ABE_GPTIMER7_WKDEP .....	567
3-424. RM_ABE_GPTIMER7_CONTEXT .....	568
3-425. Register Call Summary for Register RM_ABE_GPTIMER7_CONTEXT .....	568
3-426. PM_ABE_GPTIMER8_WKDEP .....	568
3-427. Register Call Summary for Register PM_ABE_GPTIMER8_WKDEP .....	569
3-428. RM_ABE_GPTIMER8_CONTEXT .....	569
3-429. Register Call Summary for Register RM_ABE_GPTIMER8_CONTEXT .....	569
3-430. PM_ABE_WDTIMER3_WKDEP .....	570
3-431. Register Call Summary for Register PM_ABE_WDTIMER3_WKDEP .....	570
3-432. RM_ABE_WDTIMER3_CONTEXT .....	570
3-433. Register Call Summary for Register RM_ABE_WDTIMER3_CONTEXT .....	571
3-434. ALWAYS_ON_PRM Registers Mapping Summary .....	571
3-435. PM_ALWON_SR_MPU_WKDEP .....	571
3-436. Register Call Summary for Register PM_ALWON_SR_MPU_WKDEP .....	572
3-437. RM_ALWON_SR_MPU_CONTEXT .....	572
3-438. Register Call Summary for Register RM_ALWON_SR_MPU_CONTEXT .....	572
3-439. PM_ALWON_SR_IVA_WKDEP .....	572
3-440. Register Call Summary for Register PM_ALWON_SR_IVA_WKDEP .....	573
3-441. RM_ALWON_SR_IVA_CONTEXT .....	573
3-442. Register Call Summary for Register RM_ALWON_SR_IVA_CONTEXT .....	574
3-443. PM_ALWON_SR_CORE_WKDEP .....	574
3-444. Register Call Summary for Register PM_ALWON_SR_CORE_WKDEP .....	574
3-445. RM_ALWON_SR_CORE_CONTEXT .....	575
3-446. Register Call Summary for Register RM_ALWON_SR_CORE_CONTEXT .....	575
3-447. CORE_PRM Registers Mapping Summary .....	575
3-448. PM_CORE_PWRSTCTRL .....	576
3-449. Register Call Summary for Register PM_CORE_PWRSTCTRL .....	578
3-450. PM_CORE_PWRSTST .....	578
3-451. Register Call Summary for Register PM_CORE_PWRSTST .....	579
3-452. RM_L3_1_L3_1_CONTEXT .....	579
3-453. Register Call Summary for Register RM_L3_1_L3_1_CONTEXT .....	580
3-454. RM_L3_2_L3_2_CONTEXT .....	580
3-455. Register Call Summary for Register RM_L3_2_L3_2_CONTEXT .....	580
3-456. RM_L3_2_GPMC_CONTEXT .....	581
3-457. Register Call Summary for Register RM_L3_2_GPMC_CONTEXT .....	581
3-458. RM_L3_2_OCMC_RAM_CONTEXT .....	581
3-459. Register Call Summary for Register RM_L3_2_OCMC_RAM_CONTEXT .....	582
3-460. RM_MPU_M3_RSTCTRL .....	582
3-461. Register Call Summary for Register RM_MPU_M3_RSTCTRL .....	582
3-462. RM_MPU_M3_RSTST .....	583
3-463. Register Call Summary for Register RM_MPU_M3_RSTST .....	584
3-464. RM_MPU_M3_MPU_M3_CONTEXT .....	584
3-465. Register Call Summary for Register RM_MPU_M3_MPU_M3_CONTEXT .....	584
3-466. RM_SDMA_SDMA_CONTEXT .....	585

3-467. Register Call Summary for Register RM_SDMA_SDMA_CONTEXT .....	585
3-468. RM_MEMIF_DMM_CONTEXT .....	585
3-469. Register Call Summary for Register RM_MEMIF_DMM_CONTEXT .....	586
3-470. RM_MEMIF_EMIF_FW_CONTEXT .....	586
3-471. Register Call Summary for Register RM_MEMIF_EMIF_FW_CONTEXT .....	587
3-472. RM_MEMIF_EMIF_1_CONTEXT .....	587
3-473. Register Call Summary for Register RM_MEMIF_EMIF_1_CONTEXT .....	588
3-474. RM_MEMIF_EMIF_2_CONTEXT .....	588
3-475. Register Call Summary for Register RM_MEMIF_EMIF_2_CONTEXT .....	588
3-476. RM_MEMIF_DLL_CONTEXT .....	589
3-477. Register Call Summary for Register RM_MEMIF_DLL_CONTEXT .....	589
3-478. RM_C2C_C2C_CONTEXT .....	589
3-479. Register Call Summary for Register RM_C2C_C2C_CONTEXT .....	590
3-480. RM_C2C_C2C_FW_CONTEXT .....	590
3-481. Register Call Summary for Register RM_C2C_C2C_FW_CONTEXT .....	590
3-482. RM_L4CFG_L4_CFG_CONTEXT .....	591
3-483. Register Call Summary for Register RM_L4CFG_L4_CFG_CONTEXT .....	591
3-484. RM_L4CFG_SPINLOCK_CONTEXT .....	591
3-485. Register Call Summary for Register RM_L4CFG_SPINLOCK_CONTEXT .....	592
3-486. RM_L4CFG_MAILBOX_CONTEXT .....	592
3-487. Register Call Summary for Register RM_L4CFG_MAILBOX_CONTEXT .....	592
3-488. RM_L4CFG_SAR_ROM_CONTEXT .....	593
3-489. Register Call Summary for Register RM_L4CFG_SAR_ROM_CONTEXT .....	593
3-490. RM_L3INSTR_L3_3_CONTEXT .....	593
3-491. Register Call Summary for Register RM_L3INSTR_L3_3_CONTEXT .....	594
3-492. RM_L3INSTR_L3_INSTR_CONTEXT .....	594
3-493. Register Call Summary for Register RM_L3INSTR_L3_INSTR_CONTEXT .....	594
3-494. RM_L3INSTR_INTRCONN_WP1_CONTEXT .....	595
3-495. Register Call Summary for Register RM_L3INSTR_INTRCONN_WP1_CONTEXT .....	595
3-496. IVAHD_PRM Registers Mapping Summary .....	595
3-497. PM_IVAHD_PWRSTCTRL .....	596
3-498. Register Call Summary for Register PM_IVAHD_PWRSTCTRL .....	597
3-499. PM_IVAHD_PWRSTST .....	597
3-500. Register Call Summary for Register PM_IVAHD_PWRSTST .....	598
3-501. RM_IVAHD_RSTCTRL .....	599
3-502. Register Call Summary for Register RM_IVAHD_RSTCTRL .....	599
3-503. RM_IVAHD_RSTST .....	599
3-504. Register Call Summary for Register RM_IVAHD_RSTST .....	600
3-505. RM_IVAHD_IVAHD_CONTEXT .....	600
3-506. Register Call Summary for Register RM_IVAHD_IVAHD_CONTEXT .....	601
3-507. RM_IVAHD_SL2_CONTEXT .....	601
3-508. Register Call Summary for Register RM_IVAHD_SL2_CONTEXT .....	602
3-509. CAM_PRM Registers Mapping Summary .....	602
3-510. PM_CAM_PWRSTCTRL .....	602
3-511. Register Call Summary for Register PM_CAM_PWRSTCTRL .....	603
3-512. PM_CAM_PWRSTST .....	603
3-513. Register Call Summary for Register PM_CAM_PWRSTST .....	604
3-514. RM_CAM_ISS_CONTEXT .....	604
3-515. Register Call Summary for Register RM_CAM_ISS_CONTEXT .....	604

3-516. RM_CAM_FDIF_CONTEXT .....	605
3-517. Register Call Summary for Register RM_CAM_FDIF_CONTEXT .....	605
3-518. DSS_PRM Registers Mapping Summary.....	605
3-519. PM_DSS_PWRSTCTRL .....	606
3-520. Register Call Summary for Register PM_DSS_PWRSTCTRL.....	606
3-521. PM_DSS_PWRSTST.....	607
3-522. Register Call Summary for Register PM_DSS_PWRSTST .....	607
3-523. PM_DSS_DSS_WKDEP .....	608
3-524. Register Call Summary for Register PM_DSS_DSS_WKDEP .....	610
3-525. RM_DSS_DSS_CONTEXT .....	610
3-526. Register Call Summary for Register RM_DSS_DSS_CONTEXT .....	610
3-527. SGX_PRM Registers Mapping Summary.....	611
3-528. PM_SGX_PWRSTCTRL .....	611
3-529. Register Call Summary for Register PM_SGX_PWRSTCTRL .....	611
3-530. PM_SGX_PWRSTST.....	612
3-531. Register Call Summary for Register PM_SGX_PWRSTST .....	612
3-532. RM_SGX_SGX_CONTEXT .....	613
3-533. Register Call Summary for Register RM_SGX_SGX_CONTEXT .....	613
3-534. L3INIT_PRM Registers Mapping Summary .....	613
3-535. PM_L3INIT_PWRSTCTRL.....	614
3-536. Register Call Summary for Register PM_L3INIT_PWRSTCTRL .....	615
3-537. PM_L3INIT_PWRSTST .....	615
3-538. Register Call Summary for Register PM_L3INIT_PWRSTST.....	616
3-539. PM_L3INIT_MMC1_WKDEP .....	616
3-540. Register Call Summary for Register PM_L3INIT_MMC1_WKDEP.....	617
3-541. RM_L3INIT_MMC1_CONTEXT .....	617
3-542. Register Call Summary for Register RM_L3INIT_MMC1_CONTEXT .....	618
3-543. PM_L3INIT_MMC2_WKDEP .....	618
3-544. Register Call Summary for Register PM_L3INIT_MMC2_WKDEP.....	619
3-545. RM_L3INIT_MMC2_CONTEXT .....	619
3-546. Register Call Summary for Register RM_L3INIT_MMC2_CONTEXT .....	619
3-547. PM_L3INIT_HSI_WKDEP .....	620
3-548. Register Call Summary for Register PM_L3INIT_HSI_WKDEP .....	620
3-549. RM_L3INIT_HSI_CONTEXT .....	621
3-550. Register Call Summary for Register RM_L3INIT_HSI_CONTEXT .....	621
3-551. PM_L3INIT_HSUSBHOST_WKDEP .....	621
3-552. Register Call Summary for Register PM_L3INIT_HSUSBHOST_WKDEP.....	622
3-553. RM_L3INIT_HSUSBHOST_CONTEXT .....	622
3-554. Register Call Summary for Register RM_L3INIT_HSUSBHOST_CONTEXT.....	623
3-555. PM_L3INIT_HSUSBOTG_WKDEP .....	623
3-556. Register Call Summary for Register PM_L3INIT_HSUSBOTG_WKDEP .....	623
3-557. RM_L3INIT_HSUSBOTG_CONTEXT.....	624
3-558. Register Call Summary for Register RM_L3INIT_HSUSBOTG_CONTEXT .....	624
3-559. PM_L3INIT_HSUSBTLL_WKDEP .....	624
3-560. Register Call Summary for Register PM_L3INIT_HSUSBTLL_WKDEP.....	625
3-561. RM_L3INIT_HSUSBTLL_CONTEXT .....	625
3-562. Register Call Summary for Register RM_L3INIT_HSUSBTLL_CONTEXT .....	626
3-563. RM_L3INIT_P1500_CONTEXT .....	626
3-564. Register Call Summary for Register RM_L3INIT_P1500_CONTEXT.....	626



3-565. PM_L3INIT_FSUSB_WKDEP .....	627
3-566. Register Call Summary for Register PM_L3INIT_FSUSB_WKDEP .....	627
3-567. RM_L3INIT_FSUSB_CONTEXT .....	627
3-568. Register Call Summary for Register RM_L3INIT_FSUSB_CONTEXT .....	628
3-569. RM_L3INIT_USBPHY_CONTEXT .....	628
3-570. Register Call Summary for Register RM_L3INIT_USBPHY_CONTEXT .....	628
3-571. L4PER_PRM Registers Mapping Summary .....	628
3-572. PM_L4PER_PWRSTCTRL .....	631
3-573. Register Call Summary for Register PM_L4PER_PWRSTCTRL .....	632
3-574. PM_L4PER_PWRSTST .....	632
3-575. Register Call Summary for Register PM_L4PER_PWRSTST .....	633
3-576. PM_L4PER_GPTIMER10_WKDEP .....	633
3-577. Register Call Summary for Register PM_L4PER_GPTIMER10_WKDEP .....	633
3-578. RM_L4PER_GPTIMER10_CONTEXT .....	634
3-579. Register Call Summary for Register RM_L4PER_GPTIMER10_CONTEXT .....	634
3-580. PM_L4PER_GPTIMER11_WKDEP .....	634
3-581. Register Call Summary for Register PM_L4PER_GPTIMER11_WKDEP .....	635
3-582. RM_L4PER_GPTIMER11_CONTEXT .....	635
3-583. Register Call Summary for Register RM_L4PER_GPTIMER11_CONTEXT .....	635
3-584. PM_L4PER_GPTIMER2_WKDEP .....	636
3-585. Register Call Summary for Register PM_L4PER_GPTIMER2_WKDEP .....	636
3-586. RM_L4PER_GPTIMER2_CONTEXT .....	636
3-587. Register Call Summary for Register RM_L4PER_GPTIMER2_CONTEXT .....	637
3-588. PM_L4PER_GPTIMER3_WKDEP .....	637
3-589. Register Call Summary for Register PM_L4PER_GPTIMER3_WKDEP .....	637
3-590. RM_L4PER_GPTIMER3_CONTEXT .....	638
3-591. Register Call Summary for Register RM_L4PER_GPTIMER3_CONTEXT .....	638
3-592. PM_L4PER_GPTIMER4_WKDEP .....	638
3-593. Register Call Summary for Register PM_L4PER_GPTIMER4_WKDEP .....	639
3-594. RM_L4PER_GPTIMER4_CONTEXT .....	639
3-595. Register Call Summary for Register RM_L4PER_GPTIMER4_CONTEXT .....	639
3-596. PM_L4PER_GPTIMER9_WKDEP .....	640
3-597. Register Call Summary for Register PM_L4PER_GPTIMER9_WKDEP .....	640
3-598. RM_L4PER_GPTIMER9_CONTEXT .....	640
3-599. Register Call Summary for Register RM_L4PER_GPTIMER9_CONTEXT .....	641
3-600. RM_L4PER_ELM_CONTEXT .....	641
3-601. Register Call Summary for Register RM_L4PER_ELM_CONTEXT .....	641
3-602. PM_L4PER_GPIO2_WKDEP .....	642
3-603. Register Call Summary for Register PM_L4PER_GPIO2_WKDEP .....	642
3-604. RM_L4PER_GPIO2_CONTEXT .....	643
3-605. Register Call Summary for Register RM_L4PER_GPIO2_CONTEXT .....	643
3-606. PM_L4PER_GPIO3_WKDEP .....	643
3-607. Register Call Summary for Register PM_L4PER_GPIO3_WKDEP .....	644
3-608. RM_L4PER_GPIO3_CONTEXT .....	644
3-609. Register Call Summary for Register RM_L4PER_GPIO3_CONTEXT .....	644
3-610. PM_L4PER_GPIO4_WKDEP .....	645
3-611. Register Call Summary for Register PM_L4PER_GPIO4_WKDEP .....	645
3-612. RM_L4PER_GPIO4_CONTEXT .....	645
3-613. Register Call Summary for Register RM_L4PER_GPIO4_CONTEXT .....	646

3-614. PM_L4PER_GPIO5_WKDEP .....	646
3-615. Register Call Summary for Register PM_L4PER_GPIO5_WKDEP .....	646
3-616. RM_L4PER_GPIO5_CONTEXT .....	647
3-617. Register Call Summary for Register RM_L4PER_GPIO5_CONTEXT .....	647
3-618. PM_L4PER_GPIO6_WKDEP .....	647
3-619. Register Call Summary for Register PM_L4PER_GPIO6_WKDEP .....	648
3-620. RM_L4PER_GPIO6_CONTEXT .....	648
3-621. Register Call Summary for Register RM_L4PER_GPIO6_CONTEXT .....	648
3-622. RM_L4PER_HDQ1W_CONTEXT .....	649
3-623. Register Call Summary for Register RM_L4PER_HDQ1W_CONTEXT .....	649
3-624. PM_L4PER_I2C1_WKDEP .....	649
3-625. Register Call Summary for Register PM_L4PER_I2C1_WKDEP .....	650
3-626. RM_L4PER_I2C1_CONTEXT .....	650
3-627. Register Call Summary for Register RM_L4PER_I2C1_CONTEXT .....	650
3-628. PM_L4PER_I2C2_WKDEP .....	651
3-629. Register Call Summary for Register PM_L4PER_I2C2_WKDEP .....	651
3-630. RM_L4PER_I2C2_CONTEXT .....	652
3-631. Register Call Summary for Register RM_L4PER_I2C2_CONTEXT .....	652
3-632. PM_L4PER_I2C3_WKDEP .....	652
3-633. Register Call Summary for Register PM_L4PER_I2C3_WKDEP .....	653
3-634. RM_L4PER_I2C3_CONTEXT .....	653
3-635. Register Call Summary for Register RM_L4PER_I2C3_CONTEXT .....	653
3-636. PM_L4PER_I2C4_WKDEP .....	654
3-637. Register Call Summary for Register PM_L4PER_I2C4_WKDEP .....	654
3-638. RM_L4PER_I2C4_CONTEXT .....	655
3-639. Register Call Summary for Register RM_L4PER_I2C4_CONTEXT .....	655
3-640. RM_L4PER_L4_PER_CONTEXT .....	655
3-641. Register Call Summary for Register RM_L4PER_L4_PER_CONTEXT .....	656
3-642. PM_L4PER_MCBSP4_WKDEP .....	656
3-643. Register Call Summary for Register PM_L4PER_MCBSP4_WKDEP .....	657
3-644. RM_L4PER_MCBSP4_CONTEXT .....	657
3-645. Register Call Summary for Register RM_L4PER_MCBSP4_CONTEXT .....	657
3-646. PM_L4PER_MCSP11_WKDEP .....	658
3-647. Register Call Summary for Register PM_L4PER_MCSP11_WKDEP .....	658
3-648. RM_L4PER_MCSP11_CONTEXT .....	659
3-649. Register Call Summary for Register RM_L4PER_MCSP11_CONTEXT .....	659
3-650. PM_L4PER_MCSP12_WKDEP .....	659
3-651. Register Call Summary for Register PM_L4PER_MCSP12_WKDEP .....	660
3-652. RM_L4PER_MCSP12_CONTEXT .....	660
3-653. Register Call Summary for Register RM_L4PER_MCSP12_CONTEXT .....	660
3-654. PM_L4PER_MCSP13_WKDEP .....	661
3-655. Register Call Summary for Register PM_L4PER_MCSP13_WKDEP .....	661
3-656. RM_L4PER_MCSP13_CONTEXT .....	661
3-657. Register Call Summary for Register RM_L4PER_MCSP13_CONTEXT .....	662
3-658. PM_L4PER_MCSP14_WKDEP .....	662
3-659. Register Call Summary for Register PM_L4PER_MCSP14_WKDEP .....	662
3-660. RM_L4PER_MCSP14_CONTEXT .....	663
3-661. Register Call Summary for Register RM_L4PER_MCSP14_CONTEXT .....	663
3-662. PM_L4PER_MMCS3_WKDEP .....	663

3-663. Register Call Summary for Register PM_L4PER_MMCS3D3_WKDEP .....	664
3-664. RM_L4PER_MMCS3D3_CONTEXT.....	664
3-665. Register Call Summary for Register RM_L4PER_MMCS3D3_CONTEXT .....	665
3-666. PM_L4PER_MMCS4_WKDEP .....	665
3-667. Register Call Summary for Register PM_L4PER_MMCS4_WKDEP .....	666
3-668. RM_L4PER_MMCS4_CONTEXT.....	666
3-669. Register Call Summary for Register RM_L4PER_MMCS4_CONTEXT .....	666
3-670. PM_L4PER_SLIMBUS2_WKDEP .....	667
3-671. Register Call Summary for Register PM_L4PER_SLIMBUS2_WKDEP.....	667
3-672. RM_L4PER_SLIMBUS2_CONTEXT .....	668
3-673. Register Call Summary for Register RM_L4PER_SLIMBUS2_CONTEXT.....	668
3-674. PM_L4PER_UART1_WKDEP.....	668
3-675. Register Call Summary for Register PM_L4PER_UART1_WKDEP .....	669
3-676. RM_L4PER_UART1_CONTEXT .....	669
3-677. Register Call Summary for Register RM_L4PER_UART1_CONTEXT .....	670
3-678. PM_L4PER_UART2_WKDEP.....	670
3-679. Register Call Summary for Register PM_L4PER_UART2_WKDEP .....	671
3-680. RM_L4PER_UART2_CONTEXT .....	671
3-681. Register Call Summary for Register RM_L4PER_UART2_CONTEXT .....	671
3-682. PM_L4PER_UART3_WKDEP.....	672
3-683. Register Call Summary for Register PM_L4PER_UART3_WKDEP .....	672
3-684. RM_L4PER_UART3_CONTEXT .....	673
3-685. Register Call Summary for Register RM_L4PER_UART3_CONTEXT .....	673
3-686. PM_L4PER_UART4_WKDEP.....	673
3-687. Register Call Summary for Register PM_L4PER_UART4_WKDEP .....	674
3-688. RM_L4PER_UART4_CONTEXT .....	674
3-689. Register Call Summary for Register RM_L4PER_UART4_CONTEXT .....	675
3-690. PM_L4PER_MMCS5_WKDEP .....	675
3-691. Register Call Summary for Register PM_L4PER_MMCS5_WKDEP .....	676
3-692. RM_L4PER_MMCS5_CONTEXT.....	676
3-693. Register Call Summary for Register RM_L4PER_MMCS5_CONTEXT .....	676
3-694. WKUP_PRM Registers Mapping Summary .....	677
3-695. RM_WKUP_L4WKUP_CONTEXT .....	677
3-696. Register Call Summary for Register RM_WKUP_L4WKUP_CONTEXT .....	678
3-697. PM_WKUP_WDTIMER2_WKDEP.....	678
3-698. Register Call Summary for Register PM_WKUP_WDTIMER2_WKDEP .....	678
3-699. RM_WKUP_WDTIMER2_CONTEXT .....	679
3-700. Register Call Summary for Register RM_WKUP_WDTIMER2_CONTEXT .....	679
3-701. PM_WKUP_GPIO1_WKDEP.....	679
3-702. Register Call Summary for Register PM_WKUP_GPIO1_WKDEP .....	680
3-703. RM_WKUP_GPIO1_CONTEXT.....	680
3-704. Register Call Summary for Register RM_WKUP_GPIO1_CONTEXT .....	680
3-705. PM_WKUP_GPTIMER1_WKDEP .....	681
3-706. Register Call Summary for Register PM_WKUP_GPTIMER1_WKDEP.....	681
3-707. RM_WKUP_GPTIMER1_CONTEXT .....	681
3-708. Register Call Summary for Register RM_WKUP_GPTIMER1_CONTEXT.....	682
3-709. RM_WKUP_32KTIMER_CONTEXT.....	682
3-710. Register Call Summary for Register RM_WKUP_32KTIMER_CONTEXT .....	682
3-711. RM_WKUP_SARRAM_CONTEXT .....	682

3-712. Register Call Summary for Register RM_WKUP_SARRAM_CONTEXT .....	683
3-713. PM_WKUP_KEYBOARD_WKDEP .....	683
3-714. Register Call Summary for Register PM_WKUP_KEYBOARD_WKDEP .....	683
3-715. RM_WKUP_KEYBOARD_CONTEXT .....	684
3-716. Register Call Summary for Register RM_WKUP_KEYBOARD_CONTEXT .....	684
3-717. WKUP_CM Registers Mapping Summary .....	684
3-718. CM_WKUP_CLKSTCTRL .....	685
3-719. Register Call Summary for Register CM_WKUP_CLKSTCTRL .....	686
3-720. CM_WKUP_L4WKUP_CLKCTRL .....	686
3-721. Register Call Summary for Register CM_WKUP_L4WKUP_CLKCTRL .....	687
3-722. CM_WKUP_WDTIMER2_CLKCTRL .....	687
3-723. Register Call Summary for Register CM_WKUP_WDTIMER2_CLKCTRL .....	687
3-724. CM_WKUP_GPIO1_CLKCTRL .....	688
3-725. Register Call Summary for Register CM_WKUP_GPIO1_CLKCTRL .....	688
3-726. CM_WKUP_GPTIMER1_CLKCTRL .....	689
3-727. Register Call Summary for Register CM_WKUP_GPTIMER1_CLKCTRL .....	689
3-728. CM_WKUP_32KTIMER_CLKCTRL .....	690
3-729. Register Call Summary for Register CM_WKUP_32KTIMER_CLKCTRL .....	690
3-730. CM_WKUP_SARRAM_CLKCTRL .....	690
3-731. Register Call Summary for Register CM_WKUP_SARRAM_CLKCTRL .....	691
3-732. CM_WKUP_KEYBOARD_CLKCTRL .....	691
3-733. Register Call Summary for Register CM_WKUP_KEYBOARD_CLKCTRL .....	692
3-734. CM_WKUP_BANDGAP_CLKCTRL .....	692
3-735. Register Call Summary for Register CM_WKUP_BANDGAP_CLKCTRL .....	692
3-736. EMU_PRM Registers Mapping Summary .....	693
3-737. PM_EMU_PWRSTCTRL .....	693
3-738. Register Call Summary for Register PM_EMU_PWRSTCTRL .....	693
3-739. PM_EMU_PWRSTST .....	693
3-740. Register Call Summary for Register PM_EMU_PWRSTST .....	694
3-741. RM_EMU_DEBUGSS_CONTEXT .....	694
3-742. Register Call Summary for Register RM_EMU_DEBUGSS_CONTEXT .....	695
3-743. EMU_CM Registers Mapping Summary .....	695
3-744. CM_EMU_CLKSTCTRL .....	695
3-745. Register Call Summary for Register CM_EMU_CLKSTCTRL .....	696
3-746. CM_EMU_DYNAMICDEP .....	696
3-747. Register Call Summary for Register CM_EMU_DYNAMICDEP .....	697
3-748. CM_EMU_DEBUGSS_CLKCTRL .....	697
3-749. Register Call Summary for Register CM_EMU_DEBUGSS_CLKCTRL .....	698
3-750. DEVICE_PRM Registers Mapping Summary .....	698
3-751. PRM_RSTCTRL .....	700
3-752. Register Call Summary for Register PRM_RSTCTRL .....	700
3-753. PRM_RSTST .....	701
3-754. Register Call Summary for Register PRM_RSTST .....	702
3-755. PRM_RSTTIME .....	702
3-756. Register Call Summary for Register PRM_RSTTIME .....	702
3-757. PRM_CLKREQCTRL .....	703
3-758. Register Call Summary for Register PRM_CLKREQCTRL .....	703
3-759. PRM_VOLTCTRL .....	703
3-760. Register Call Summary for Register PRM_VOLTCTRL .....	705

3-761. PRM_PWRREQCTRL.....	705
3-762. Register Call Summary for Register PRM_PWRREQCTRL .....	706
3-763. PRM_PSCON_COUNT .....	706
3-764. Register Call Summary for Register PRM_PSCON_COUNT .....	706
3-765. PRM_IO_COUNT .....	706
3-766. Register Call Summary for Register PRM_IO_COUNT.....	706
3-767. PRM_IO_PMCTRL.....	707
3-768. Register Call Summary for Register PRM_IO_PMCTRL .....	708
3-769. PRM_VOLTSETUP_WARMRESET .....	708
3-770. Register Call Summary for Register PRM_VOLTSETUP_WARMRESET .....	708
3-771. PRM_VOLTSETUP_CORE_OFF.....	709
3-772. Register Call Summary for Register PRM_VOLTSETUP_CORE_OFF .....	709
3-773. PRM_VOLTSETUP_MPU_OFF.....	710
3-774. Register Call Summary for Register PRM_VOLTSETUP_MPU_OFF .....	710
3-775. PRM_VOLTSETUP_IVA_OFF .....	711
3-776. Register Call Summary for Register PRM_VOLTSETUP_IVA_OFF.....	711
3-777. PRM_VOLTSETUP_CORE_RET_SLEEP.....	712
3-778. Register Call Summary for Register PRM_VOLTSETUP_CORE_RET_SLEEP .....	712
3-779. PRM_VOLTSETUP_MPU_RET_SLEEP .....	713
3-780. Register Call Summary for Register PRM_VOLTSETUP_MPU_RET_SLEEP.....	713
3-781. PRM_VOLTSETUP_IVA_RET_SLEEP .....	714
3-782. Register Call Summary for Register PRM_VOLTSETUP_IVA_RET_SLEEP.....	714
3-783. PRM_VP_CORE_CONFIG .....	715
3-784. Register Call Summary for Register PRM_VP_CORE_CONFIG.....	715
3-785. PRM_VP_CORE_STATUS .....	716
3-786. Register Call Summary for Register PRM_VP_CORE_STATUS .....	716
3-787. PRM_VP_CORE_VLIMITTO .....	716
3-788. Register Call Summary for Register PRM_VP_CORE_VLIMITTO.....	716
3-789. PRM_VP_CORE_VOLTAGE.....	717
3-790. Register Call Summary for Register PRM_VP_CORE_VOLTAGE .....	717
3-791. PRM_VP_CORE_VSTEPMAX .....	717
3-792. Register Call Summary for Register PRM_VP_CORE_VSTEPMAX .....	717
3-793. PRM_VP_CORE_VSTEPMIN.....	718
3-794. Register Call Summary for Register PRM_VP_CORE_VSTEPMIN .....	718
3-795. PRM_VP_MPU_CONFIG .....	718
3-796. Register Call Summary for Register PRM_VP_MPU_CONFIG .....	719
3-797. PRM_VP_MPU_STATUS.....	719
3-798. Register Call Summary for Register PRM_VP_MPU_STATUS .....	719
3-799. PRM_VP_MPU_VLIMITTO .....	719
3-800. Register Call Summary for Register PRM_VP_MPU_VLIMITTO .....	720
3-801. PRM_VP_MPU_VOLTAGE.....	720
3-802. Register Call Summary for Register PRM_VP_MPU_VOLTAGE .....	720
3-803. PRM_VP_MPU_VSTEPMAX.....	720
3-804. Register Call Summary for Register PRM_VP_MPU_VSTEPMAX .....	720
3-805. PRM_VP_MPU_VSTEPMIN.....	721
3-806. Register Call Summary for Register PRM_VP_MPU_VSTEPMIN .....	721
3-807. PRM_VP_IVA_CONFIG.....	721
3-808. Register Call Summary for Register PRM_VP_IVA_CONFIG .....	722
3-809. PRM_VP_IVA_STATUS .....	722

3-810. Register Call Summary for Register PRM_VP_IVA_STATUS .....	722
3-811. PRM_VP_IVA_VLIMITTO.....	722
3-812. Register Call Summary for Register PRM_VP_IVA_VLIMITTO .....	723
3-813. PRM_VP_IVA_VOLTAGE .....	723
3-814. Register Call Summary for Register PRM_VP_IVA_VOLTAGE.....	723
3-815. PRM_VP_IVA_VSTEPMAX .....	723
3-816. Register Call Summary for Register PRM_VP_IVA_VSTEPMAX.....	723
3-817. PRM_VP_IVA_VSTEPMIN .....	724
3-818. Register Call Summary for Register PRM_VP_IVA_VSTEPMIN.....	724
3-819. PRM_VC_SMPS_SA.....	724
3-820. Register Call Summary for Register PRM_VC_SMPS_SA .....	724
3-821. PRM_VC_VAL_SMPS_RA_VOL .....	725
3-822. Register Call Summary for Register PRM_VC_VAL_SMPS_RA_VOL.....	725
3-823. PRM_VC_VAL_SMPS_RA_CMD.....	725
3-824. Register Call Summary for Register PRM_VC_VAL_SMPS_RA_CMD .....	726
3-825. PRM_VC_VAL_CMD_VDD_CORE_L .....	726
3-826. Register Call Summary for Register PRM_VC_VAL_CMD_VDD_CORE_L.....	726
3-827. PRM_VC_VAL_CMD_VDD_MPU_L .....	726
3-828. Register Call Summary for Register PRM_VC_VAL_CMD_VDD_MPU_L.....	726
3-829. PRM_VC_VAL_CMD_VDD_IVA_L.....	727
3-830. Register Call Summary for Register PRM_VC_VAL_CMD_VDD_IVA_L .....	727
3-831. PRM_VC_VAL_BYPASS .....	727
3-832. Register Call Summary for Register PRM_VC_VAL_BYPASS.....	727
3-833. PRM_VC_CFG_CHANNEL.....	728
3-834. Register Call Summary for Register PRM_VC_CFG_CHANNEL .....	729
3-835. PRM_VC_CFG_I2C_MODE .....	730
3-836. Register Call Summary for Register PRM_VC_CFG_I2C_MODE .....	730
3-837. PRM_VC_CFG_I2C_CLK.....	730
3-838. Register Call Summary for Register PRM_VC_CFG_I2C_CLK .....	731
3-839. PRM_SRAM_COUNT .....	731
3-840. Register Call Summary for Register PRM_SRAM_COUNT.....	731
3-841. PRM_SRAM_WKUP_SETUP .....	732
3-842. Register Call Summary for Register PRM_SRAM_WKUP_SETUP.....	732
3-843. PRM_LDO_SRAM_CORE_SETUP .....	732
3-844. Register Call Summary for Register PRM_LDO_SRAM_CORE_SETUP.....	733
3-845. PRM_LDO_SRAM_CORE_CTRL .....	734
3-846. Register Call Summary for Register PRM_LDO_SRAM_CORE_CTRL.....	734
3-847. PRM_LDO_SRAM_MPU_SETUP .....	734
3-848. Register Call Summary for Register PRM_LDO_SRAM_MPU_SETUP.....	735
3-849. PRM_LDO_SRAM_MPU_CTRL .....	736
3-850. Register Call Summary for Register PRM_LDO_SRAM_MPU_CTRL.....	736
3-851. PRM_LDO_SRAM_IVA_SETUP.....	736
3-852. Register Call Summary for Register PRM_LDO_SRAM_IVA_SETUP .....	737
3-853. PRM_LDO_SRAM_IVA_CTRL.....	738
3-854. Register Call Summary for Register PRM_LDO_SRAM_IVA_CTRL .....	738
3-855. PRM_LDO_ABB_MPU_SETUP.....	738
3-856. Register Call Summary for Register PRM_LDO_ABB_MPU_SETUP .....	739
3-857. PRM_LDO_ABB_MPU_CTRL .....	739
3-858. Register Call Summary for Register PRM_LDO_ABB_MPU_CTRL.....	740

3-859. PRM_LDO_ABB_IVA_SETUP .....	740
3-860. Register Call Summary for Register PRM_LDO_ABB_IVA_SETUP.....	741
3-861. PRM_LDO_ABB_IVA_CTRL .....	741
3-862. Register Call Summary for Register PRM_LDO_ABB_IVA_CTRL.....	742
3-863. PRM_LDO_BANDGAP_SETUP .....	742
3-864. Register Call Summary for Register PRM_LDO_BANDGAP_SETUP.....	742
3-865. PRM_DEVICE_OFF_CTRL .....	742
3-866. Register Call Summary for Register PRM_DEVICE_OFF_CTRL .....	743
3-867. PRM_PHASE1_CNDP .....	743
3-868. Register Call Summary for Register PRM_PHASE1_CNDP.....	744
3-869. PRM_PHASE2A_CNDP .....	744
3-870. Register Call Summary for Register PRM_PHASE2A_CNDP .....	744
3-871. PRM_PHASE2B_CNDP .....	744
3-872. Register Call Summary for Register PRM_PHASE2B_CNDP .....	744
3-873. PRM_VC_ERRST.....	744
3-874. Register Call Summary for Register PRM_VC_ERRST .....	746
3-875. INSTR_PRM Registers Mapping Summary .....	747
3-876. PMI_IDENTICATION .....	747
3-877. Register Call Summary for Register PMI_IDENTICATION.....	747
3-878. PMI_SYS_CONFIG.....	747
3-879. Register Call Summary for Register PMI_SYS_CONFIG .....	748
3-880. PMI_STATUS.....	748
3-881. Register Call Summary for Register PMI_STATUS .....	748
3-882. PMI_CONFIGURATION.....	748
3-883. Register Call Summary for Register PMI_CONFIGURATION .....	749
3-884. PMI_CLASS_FILTERING.....	749
3-885. Register Call Summary for Register PMI_CLASS_FILTERING .....	749
3-886. PMI_TRIGGERING .....	749
3-887. Register Call Summary for Register PMI_TRIGGERING.....	750
3-888. PMI_SAMPLING .....	750
3-889. Register Call Summary for Register PMI_SAMPLING.....	750
3-890. CM1 Instance Summary .....	750
3-891. INTRCONN_SOCKET_CM1 Registers Mapping Summary.....	751
3-892. REVISION_CM1 .....	751
3-893. Register Call Summary for Register REVISION_CM1 .....	751
3-894. CM_CM1_PROFILING_CLKCTRL .....	751
3-895. Register Call Summary for Register CM_CM1_PROFILING_CLKCTRL.....	752
3-896. CM1_DEBUG_CFG .....	752
3-897. Register Call Summary for Register CM1_DEBUG_CFG .....	753
3-898. CKGEN_CM1 Registers Mapping Summary .....	753
3-899. CM_CLKSEL_CORE .....	754
3-900. Register Call Summary for Register CM_CLKSEL_CORE.....	755
3-901. CM_CLKSEL_ABE.....	755
3-902. Register Call Summary for Register CM_CLKSEL_ABE .....	755
3-903. CM_DLL_CTRL .....	756
3-904. Register Call Summary for Register CM_DLL_CTRL.....	756
3-905. CM_CLKMODE_DPLL_CORE .....	756
3-906. Register Call Summary for Register CM_CLKMODE_DPLL_CORE .....	757
3-907. CM_IDLEST_DPLL_CORE.....	758

3-908. Register Call Summary for Register CM_IDLEST_DPLL_CORE .....	758
3-909. CM_AUTOIDLE_DPLL_CORE .....	758
3-910. Register Call Summary for Register CM_AUTOIDLE_DPLL_CORE .....	759
3-911. CM_CLKSEL_DPLL_CORE .....	759
3-912. Register Call Summary for Register CM_CLKSEL_DPLL_CORE .....	760
3-913. CM_DIV_M2_DPLL_CORE .....	760
3-914. Register Call Summary for Register CM_DIV_M2_DPLL_CORE .....	761
3-915. CM_DIV_M3_DPLL_CORE .....	761
3-916. Register Call Summary for Register CM_DIV_M3_DPLL_CORE .....	762
3-917. CM_DIV_M4_DPLL_CORE .....	762
3-918. Register Call Summary for Register CM_DIV_M4_DPLL_CORE .....	763
3-919. CM_DIV_M5_DPLL_CORE .....	763
3-920. Register Call Summary for Register CM_DIV_M5_DPLL_CORE .....	764
3-921. CM_DIV_M6_DPLL_CORE .....	764
3-922. Register Call Summary for Register CM_DIV_M6_DPLL_CORE .....	765
3-923. CM_DIV_M7_DPLL_CORE .....	765
3-924. Register Call Summary for Register CM_DIV_M7_DPLL_CORE .....	766
3-925. CM_SSC_DELTAMSTEP_DPLL_CORE .....	766
3-926. Register Call Summary for Register CM_SSC_DELTAMSTEP_DPLL_CORE .....	766
3-927. CM_SSC_MODFREQDIV_DPLL_CORE .....	766
3-928. Register Call Summary for Register CM_SSC_MODFREQDIV_DPLL_CORE .....	767
3-929. CM_EMU_OVERRIDE_DPLL_CORE .....	767
3-930. Register Call Summary for Register CM_EMU_OVERRIDE_DPLL_CORE .....	767
3-931. CM_CLKMODE_DPLL_MPU .....	768
3-932. Register Call Summary for Register CM_CLKMODE_DPLL_MPU .....	769
3-933. CM_IDLEST_DPLL_MPU .....	769
3-934. Register Call Summary for Register CM_IDLEST_DPLL_MPU .....	769
3-935. CM_AUTOIDLE_DPLL_MPU .....	770
3-936. Register Call Summary for Register CM_AUTOIDLE_DPLL_MPU .....	770
3-937. CM_CLKSEL_DPLL_MPU .....	770
3-938. Register Call Summary for Register CM_CLKSEL_DPLL_MPU .....	771
3-939. CM_DIV_M2_DPLL_MPU .....	771
3-940. Register Call Summary for Register CM_DIV_M2_DPLL_MPU .....	772
3-941. CM_SSC_DELTAMSTEP_DPLL_MPU .....	772
3-942. Register Call Summary for Register CM_SSC_DELTAMSTEP_DPLL_MPU .....	773
3-943. CM_SSC_MODFREQDIV_DPLL_MPU .....	773
3-944. Register Call Summary for Register CM_SSC_MODFREQDIV_DPLL_MPU .....	773
3-945. CM_BYPCLK_DPLL_MPU .....	773
3-946. Register Call Summary for Register CM_BYPCLK_DPLL_MPU .....	774
3-947. CM_CLKMODE_DPLL_IVA .....	774
3-948. Register Call Summary for Register CM_CLKMODE_DPLL_IVA .....	775
3-949. CM_IDLEST_DPLL_IVA .....	775
3-950. Register Call Summary for Register CM_IDLEST_DPLL_IVA .....	776
3-951. CM_AUTOIDLE_DPLL_IVA .....	776
3-952. Register Call Summary for Register CM_AUTOIDLE_DPLL_IVA .....	776
3-953. CM_CLKSEL_DPLL_IVA .....	777
3-954. Register Call Summary for Register CM_CLKSEL_DPLL_IVA .....	777
3-955. CM_DIV_M4_DPLL_IVA .....	778
3-956. Register Call Summary for Register CM_DIV_M4_DPLL_IVA .....	778



3-957. CM_DIV_M5_DPLL_IVA .....	779
3-958. Register Call Summary for Register CM_DIV_M5_DPLL_IVA .....	779
3-959. CM_SSC_DELTAMSTEP_DPLL_IVA.....	780
3-960. Register Call Summary for Register CM_SSC_DELTAMSTEP_DPLL_IVA .....	780
3-961. CM_SSC_MODFREQDIV_DPLL_IVA .....	780
3-962. Register Call Summary for Register CM_SSC_MODFREQDIV_DPLL_IVA.....	780
3-963. CM_BYPCCLK_DPLL_IVA .....	781
3-964. Register Call Summary for Register CM_BYPCCLK_DPLL_IVA.....	781
3-965. CM_CLKMODE_DPLL_ABE .....	781
3-966. Register Call Summary for Register CM_CLKMODE_DPLL_ABE.....	782
3-967. CM_IDLEST_DPLL_ABE .....	783
3-968. Register Call Summary for Register CM_IDLEST_DPLL_ABE.....	783
3-969. CM_AUTOIDLE_DPLL_ABE .....	783
3-970. Register Call Summary for Register CM_AUTOIDLE_DPLL_ABE.....	784
3-971. CM_CLKSEL_DPLL_ABE .....	784
3-972. Register Call Summary for Register CM_CLKSEL_DPLL_ABE.....	785
3-973. CM_DIV_M2_DPLL_ABE .....	785
3-974. Register Call Summary for Register CM_DIV_M2_DPLL_ABE .....	786
3-975. CM_DIV_M3_DPLL_ABE .....	786
3-976. Register Call Summary for Register CM_DIV_M3_DPLL_ABE .....	786
3-977. CM_SSC_DELTAMSTEP_DPLL_ABE.....	787
3-978. Register Call Summary for Register CM_SSC_DELTAMSTEP_DPLL_ABE .....	787
3-979. CM_SSC_MODFREQDIV_DPLL_ABE .....	787
3-980. Register Call Summary for Register CM_SSC_MODFREQDIV_DPLL_ABE.....	787
3-981. CM_SHADOW_FREQ_CONFIG1 .....	788
3-982. Register Call Summary for Register CM_SHADOW_FREQ_CONFIG1.....	789
3-983. CM_SHADOW_FREQ_CONFIG2 .....	789
3-984. Register Call Summary for Register CM_SHADOW_FREQ_CONFIG2.....	790
3-985. CM_DYN_DEP_PRESCAL .....	790
3-986. Register Call Summary for Register CM_DYN_DEP_PRESCAL .....	790
3-987. CM_RESTORE_ST.....	791
3-988. Register Call Summary for Register CM_RESTORE_ST .....	791
3-989. MPU_CM1 Registers Mapping Summary.....	791
3-990. CM_MPU_CLKSTCTRL.....	792
3-991. Register Call Summary for Register CM_MPU_CLKSTCTRL .....	792
3-992. CM_MPU_STATICDEP .....	793
3-993. Register Call Summary for Register CM_MPU_STATICDEP.....	794
3-994. CM_MPU_DYNAMICDEP .....	794
3-995. Register Call Summary for Register CM_MPU_DYNAMICDEP .....	795
3-996. CM_MPU_MPU_CLKCTRL .....	795
3-997. Register Call Summary for Register CM_MPU_MPU_CLKCTRL .....	796
3-998. DSP_CM1 Registers Mapping Summary .....	796
3-999. CM_DSP_CLKSTCTRL .....	796
3-1000. Register Call Summary for Register CM_DSP_CLKSTCTRL .....	797
3-1001. CM_DSP_STATICDEP .....	797
3-1002. Register Call Summary for Register CM_DSP_STATICDEP.....	798
3-1003. CM_DSP_DYNAMICDEP .....	798
3-1004. Register Call Summary for Register CM_DSP_DYNAMICDEP .....	799
3-1005. CM_DSP_DSP_CLKCTRL .....	799

3-1006. Register Call Summary for Register CM_DSP_DSP_CLKCTRL.....	800
3-1007. ABE_CM1 Registers Mapping Summary .....	800
3-1008. CM1_ABE_CLKSTCTRL .....	801
3-1009. Register Call Summary for Register CM1_ABE_CLKSTCTRL.....	802
3-1010. CM1_ABE_L4ABE_CLKCTRL.....	802
3-1011. Register Call Summary for Register CM1_ABE_L4ABE_CLKCTRL .....	803
3-1012. CM1_ABE_AESS_CLKCTRL .....	803
3-1013. Register Call Summary for Register CM1_ABE_AESS_CLKCTRL.....	804
3-1014. CM1_ABE_PDM_CLKCTRL .....	804
3-1015. Register Call Summary for Register CM1_ABE_PDM_CLKCTRL.....	804
3-1016. CM1_ABE_DMIC_CLKCTRL .....	805
3-1017. Register Call Summary for Register CM1_ABE_DMIC_CLKCTRL.....	806
3-1018. CM1_ABE_MCASP_CLKCTRL.....	806
3-1019. Register Call Summary for Register CM1_ABE_MCASP_CLKCTRL .....	807
3-1020. CM1_ABE_MCBSP1_CLKCTRL .....	807
3-1021. Register Call Summary for Register CM1_ABE_MCBSP1_CLKCTRL.....	808
3-1022. CM1_ABE_MCBSP2_CLKCTRL .....	809
3-1023. Register Call Summary for Register CM1_ABE_MCBSP2_CLKCTRL.....	810
3-1024. CM1_ABE_MCBSP3_CLKCTRL .....	810
3-1025. Register Call Summary for Register CM1_ABE_MCBSP3_CLKCTRL.....	811
3-1026. CM1_ABE_SLIMBUS_CLKCTRL .....	811
3-1027. Register Call Summary for Register CM1_ABE_SLIMBUS_CLKCTRL.....	812
3-1028. CM1_ABE_GPTIMER5_CLKCTRL .....	812
3-1029. Register Call Summary for Register CM1_ABE_GPTIMER5_CLKCTRL.....	813
3-1030. CM1_ABE_GPTIMER6_CLKCTRL .....	813
3-1031. Register Call Summary for Register CM1_ABE_GPTIMER6_CLKCTRL.....	814
3-1032. CM1_ABE_GPTIMER7_CLKCTRL .....	814
3-1033. Register Call Summary for Register CM1_ABE_GPTIMER7_CLKCTRL.....	815
3-1034. CM1_ABE_GPTIMER8_CLKCTRL .....	815
3-1035. Register Call Summary for Register CM1_ABE_GPTIMER8_CLKCTRL.....	816
3-1036. CM1_ABE_WDTIMER3_CLKCTRL.....	816
3-1037. Register Call Summary for Register CM1_ABE_WDTIMER3_CLKCTRL .....	817
3-1038. RESTORE_CM1 Registers Mapping Summary .....	817
3-1039. CM_CLKSEL_CORE_RESTORE .....	818
3-1040. Register Call Summary for Register CM_CLKSEL_CORE_RESTORE.....	819
3-1041. CM_DIV_M2_DPLL_CORE_RESTORE.....	819
3-1042. Register Call Summary for Register CM_DIV_M2_DPLL_CORE_RESTORE .....	819
3-1043. CM_DIV_M3_DPLL_CORE_RESTORE.....	820
3-1044. Register Call Summary for Register CM_DIV_M3_DPLL_CORE_RESTORE .....	820
3-1045. CM_DIV_M4_DPLL_CORE_RESTORE.....	821
3-1046. Register Call Summary for Register CM_DIV_M4_DPLL_CORE_RESTORE .....	821
3-1047. CM_DIV_M5_DPLL_CORE_RESTORE.....	822
3-1048. Register Call Summary for Register CM_DIV_M5_DPLL_CORE_RESTORE .....	822
3-1049. CM_DIV_M6_DPLL_CORE_RESTORE.....	823
3-1050. Register Call Summary for Register CM_DIV_M6_DPLL_CORE_RESTORE .....	823
3-1051. CM_DIV_M7_DPLL_CORE_RESTORE.....	824
3-1052. Register Call Summary for Register CM_DIV_M7_DPLL_CORE_RESTORE .....	824
3-1053. CM_CLKSEL_DPLL_CORE_RESTORE .....	825
3-1054. Register Call Summary for Register CM_CLKSEL_DPLL_CORE_RESTORE .....	825

3-1055. CM_SSC_DELTAMSTEP_DPLL_CORE_RESTORE .....	826
3-1056. Register Call Summary for Register CM_SSC_DELTAMSTEP_DPLL_CORE_RESTORE .....	826
3-1057. CM_SSC_MODFREQDIV_DPLL_CORE_RESTORE .....	826
3-1058. Register Call Summary for Register CM_SSC_MODFREQDIV_DPLL_CORE_RESTORE .....	827
3-1059. CM_CLKMODE_DPLL_CORE_RESTORE .....	827
3-1060. Register Call Summary for Register CM_CLKMODE_DPLL_CORE_RESTORE .....	828
3-1061. CM_SHADOW_FREQ_CONFIG2_RESTORE .....	828
3-1062. Register Call Summary for Register CM_SHADOW_FREQ_CONFIG2_RESTORE .....	829
3-1063. CM_SHADOW_FREQ_CONFIG1_RESTORE .....	829
3-1064. Register Call Summary for Register CM_SHADOW_FREQ_CONFIG1_RESTORE .....	830
3-1065. CM_AUTOIDLE_DPLL_CORE_RESTORE .....	830
3-1066. Register Call Summary for Register CM_AUTOIDLE_DPLL_CORE_RESTORE .....	831
3-1067. CM_MPU_CLKSTCTRL_RESTORE .....	832
3-1068. Register Call Summary for Register CM_MPU_CLKSTCTRL_RESTORE .....	832
3-1069. CM_CM1_PROFILING_CLKCTRL_RESTORE .....	833
3-1070. Register Call Summary for Register CM_CM1_PROFILING_CLKCTRL_RESTORE .....	833
3-1071. CM_DYN_DEP_PRESCAL_RESTORE .....	833
3-1072. Register Call Summary for Register CM_DYN_DEP_PRESCAL_RESTORE .....	834
3-1073. INSTR_CM1 Registers Mapping Summary .....	834
3-1074. CMI_IDENTICATION .....	834
3-1075. Register Call Summary for Register CMI_IDENTICATION .....	834
3-1076. CMI_SYS_CONFIG .....	834
3-1077. Register Call Summary for Register CMI_SYS_CONFIG .....	835
3-1078. CMI_STATUS .....	835
3-1079. Register Call Summary for Register CMI_STATUS .....	835
3-1080. CMI_CONFIGURATION .....	835
3-1081. Register Call Summary for Register CMI_CONFIGURATION .....	836
3-1082. CMI_CLASS_FILTERING .....	836
3-1083. Register Call Summary for Register CMI_CLASS_FILTERING .....	836
3-1084. CMI_TRIGGERING .....	837
3-1085. Register Call Summary for Register CMI_TRIGGERING .....	837
3-1086. CMI_SAMPLING .....	837
3-1087. Register Call Summary for Register CMI_SAMPLING .....	837
3-1088. CM2 Instance Summary .....	838
3-1089. INTRCONN_SOCKET_CM2 Registers Mapping Summary .....	838
3-1090. REVISION_CM2 .....	838
3-1091. Register Call Summary for Register REVISION_CM2 .....	838
3-1092. CM_CM2_PROFILING_CLKCTRL .....	839
3-1093. Register Call Summary for Register CM_CM2_PROFILING_CLKCTRL .....	839
3-1094. CM2_DEBUG_CFG .....	839
3-1095. Register Call Summary for Register CM2_DEBUG_CFG .....	840
3-1096. CKGEN_CM2 Registers Mapping Summary .....	840
3-1097. CM_CLKSEL_MPU_M3_ISS_ROOT .....	841
3-1098. Register Call Summary for Register CM_CLKSEL_MPU_M3_ISS_ROOT .....	841
3-1099. CM_CLKSEL_USB_60MHZ .....	841
3-1100. Register Call Summary for Register CM_CLKSEL_USB_60MHZ .....	842
3-1101. CM_SCALE_FCLK .....	842
3-1102. Register Call Summary for Register CM_SCALE_FCLK .....	842
3-1103. CM_CORE_DVFS_PERF1 .....	842

3-1104. Register Call Summary for Register CM_CORE_DVFS_PERF1 .....	842
3-1105. CM_CORE_DVFS_PERF2.....	843
3-1106. Register Call Summary for Register CM_CORE_DVFS_PERF2 .....	843
3-1107. CM_CORE_DVFS_PERF3.....	843
3-1108. Register Call Summary for Register CM_CORE_DVFS_PERF3 .....	843
3-1109. CM_CORE_DVFS_PERF4.....	843
3-1110. Register Call Summary for Register CM_CORE_DVFS_PERF4 .....	844
3-1111. CM_CORE_DVFS_CURRENT .....	844
3-1112. Register Call Summary for Register CM_CORE_DVFS_CURRENT.....	844
3-1113. CM_IVA_DVFS_PERF_DSP.....	844
3-1114. Register Call Summary for Register CM_IVA_DVFS_PERF_DSP .....	844
3-1115. CM_IVA_DVFS_PERF_IVAHD.....	845
3-1116. Register Call Summary for Register CM_IVA_DVFS_PERF_IVAHD .....	845
3-1117. CM_IVA_DVFS_PERF_ABE.....	845
3-1118. Register Call Summary for Register CM_IVA_DVFS_PERF_ABE .....	845
3-1119. CM_IVA_DVFS_CURRENT.....	845
3-1120. Register Call Summary for Register CM_IVA_DVFS_CURRENT .....	846
3-1121. CM_CLKMODE_DPLL_PER.....	846
3-1122. Register Call Summary for Register CM_CLKMODE_DPLL_PER .....	847
3-1123. CM_IDLEST_DPLL_PER.....	847
3-1124. Register Call Summary for Register CM_IDLEST_DPLL_PER .....	848
3-1125. CM_AUTOIDLE_DPLL_PER.....	848
3-1126. Register Call Summary for Register CM_AUTOIDLE_DPLL_PER .....	848
3-1127. CM_CLKSEL_DPLL_PER.....	849
3-1128. Register Call Summary for Register CM_CLKSEL_DPLL_PER .....	849
3-1129. CM_DIV_M2_DPLL_PER .....	849
3-1130. Register Call Summary for Register CM_DIV_M2_DPLL_PER.....	850
3-1131. CM_DIV_M3_DPLL_PER .....	851
3-1132. Register Call Summary for Register CM_DIV_M3_DPLL_PER.....	851
3-1133. CM_DIV_M4_DPLL_PER .....	852
3-1134. Register Call Summary for Register CM_DIV_M4_DPLL_PER.....	852
3-1135. CM_DIV_M5_DPLL_PER .....	853
3-1136. Register Call Summary for Register CM_DIV_M5_DPLL_PER.....	853
3-1137. CM_DIV_M6_DPLL_PER .....	854
3-1138. Register Call Summary for Register CM_DIV_M6_DPLL_PER.....	854
3-1139. CM_DIV_M7_DPLL_PER .....	855
3-1140. Register Call Summary for Register CM_DIV_M7_DPLL_PER.....	855
3-1141. CM_SSC_DELTAMSTEP_DPLL_PER .....	856
3-1142. Register Call Summary for Register CM_SSC_DELTAMSTEP_DPLL_PER.....	856
3-1143. CM_SSC_MODFREQDIV_DPLL_PER.....	856
3-1144. Register Call Summary for Register CM_SSC_MODFREQDIV_DPLL_PER .....	856
3-1145. CM_CLKMODE_DPLL_USB.....	857
3-1146. Register Call Summary for Register CM_CLKMODE_DPLL_USB .....	857
3-1147. CM_IDLEST_DPLL_USB.....	858
3-1148. Register Call Summary for Register CM_IDLEST_DPLL_USB .....	858
3-1149. CM_AUTOIDLE_DPLL_USB.....	858
3-1150. Register Call Summary for Register CM_AUTOIDLE_DPLL_USB .....	859
3-1151. CM_CLKSEL_DPLL_USB.....	859
3-1152. Register Call Summary for Register CM_CLKSEL_DPLL_USB .....	860

3-1153. CM_DIV_M2_DPLL_USB .....	860
3-1154. Register Call Summary for Register CM_DIV_M2_DPLL_USB .....	861
3-1155. CM_SSC_DELTAMSTEP_DPLL_USB .....	861
3-1156. Register Call Summary for Register CM_SSC_DELTAMSTEP_DPLL_USB.....	861
3-1157. CM_SSC_MODFREQDIV_DPLL_USB.....	861
3-1158. Register Call Summary for Register CM_SSC_MODFREQDIV_DPLL_USB .....	862
3-1159. CM_CLKDCOLDO_DPLL_USB .....	862
3-1160. Register Call Summary for Register CM_CLKDCOLDO_DPLL_USB.....	862
3-1161. ALWAYS_ON_CM2 Registers Mapping Summary .....	862
3-1162. CM_ALWON_CLKSTCTRL .....	863
3-1163. Register Call Summary for Register CM_ALWON_CLKSTCTRL.....	864
3-1164. CM_ALWON_SR_MPU_CLKCTRL .....	864
3-1165. Register Call Summary for Register CM_ALWON_SR_MPU_CLKCTRL .....	865
3-1166. CM_ALWON_SR_IVA_CLKCTRL.....	865
3-1167. Register Call Summary for Register CM_ALWON_SR_IVA_CLKCTRL .....	865
3-1168. CM_ALWON_SR_CORE_CLKCTRL .....	866
3-1169. Register Call Summary for Register CM_ALWON_SR_CORE_CLKCTRL.....	866
3-1170. CM_ALWON_USBPHY_CLKCTRL .....	866
3-1171. Register Call Summary for Register CM_ALWON_USBPHY_CLKCTRL.....	867
3-1172. CORE_CM2 Registers Mapping Summary .....	867
3-1173. CM_L3_1_CLKSTCTRL .....	868
3-1174. Register Call Summary for Register CM_L3_1_CLKSTCTRL.....	869
3-1175. CM_L3_1_DYNAMICDEP .....	869
3-1176. Register Call Summary for Register CM_L3_1_DYNAMICDEP .....	870
3-1177. CM_L3_1_L3_1_CLKCTRL .....	870
3-1178. Register Call Summary for Register CM_L3_1_L3_1_CLKCTRL.....	870
3-1179. CM_L3_2_CLKSTCTRL .....	871
3-1180. Register Call Summary for Register CM_L3_2_CLKSTCTRL.....	871
3-1181. CM_L3_2_DYNAMICDEP .....	872
3-1182. Register Call Summary for Register CM_L3_2_DYNAMICDEP .....	872
3-1183. CM_L3_2_L3_2_CLKCTRL .....	873
3-1184. Register Call Summary for Register CM_L3_2_L3_2_CLKCTRL.....	873
3-1185. CM_L3_2_GPMC_CLKCTRL .....	873
3-1186. Register Call Summary for Register CM_L3_2_GPMC_CLKCTRL.....	874
3-1187. CM_L3_2_OCMC_RAM_CLKCTRL .....	874
3-1188. Register Call Summary for Register CM_L3_2_OCMC_RAM_CLKCTRL.....	875
3-1189. CM_MPU_M3_CLKSTCTRL .....	875
3-1190. Register Call Summary for Register CM_MPU_M3_CLKSTCTRL .....	876
3-1191. CM_MPU_M3_STATICDEP .....	876
3-1192. Register Call Summary for Register CM_MPU_M3_STATICDEP .....	877
3-1193. CM_MPU_M3_DYNAMICDEP.....	877
3-1194. Register Call Summary for Register CM_MPU_M3_DYNAMICDEP .....	878
3-1195. CM_MPU_M3_MPU_M3_CLKCTRL.....	878
3-1196. Register Call Summary for Register CM_MPU_M3_MPU_M3_CLKCTRL .....	879
3-1197. CM_SDMA_CLKSTCTRL .....	879
3-1198. Register Call Summary for Register CM_SDMA_CLKSTCTRL.....	879
3-1199. CM_SDMA_STATICDEP .....	880
3-1200. Register Call Summary for Register CM_SDMA_STATICDEP .....	881
3-1201. CM_SDMA_DYNAMICDEP .....	881

3-1202. Register Call Summary for Register CM_SDMA_DYNAMICDEP .....	881
3-1203. CM_SDMA_SDMA_CLKCTRL .....	882
3-1204. Register Call Summary for Register CM_SDMA_SDMA_CLKCTRL .....	882
3-1205. CM_MEMIF_CLKSTCTRL .....	883
3-1206. Register Call Summary for Register CM_MEMIF_CLKSTCTRL .....	884
3-1207. CM_MEMIF_DMM_CLKCTRL .....	884
3-1208. Register Call Summary for Register CM_MEMIF_DMM_CLKCTRL .....	884
3-1209. CM_MEMIF_EMIF_FW_CLKCTRL .....	885
3-1210. Register Call Summary for Register CM_MEMIF_EMIF_FW_CLKCTRL .....	885
3-1211. CM_MEMIF_EMIF_1_CLKCTRL .....	885
3-1212. Register Call Summary for Register CM_MEMIF_EMIF_1_CLKCTRL .....	886
3-1213. CM_MEMIF_EMIF_2_CLKCTRL .....	886
3-1214. Register Call Summary for Register CM_MEMIF_EMIF_2_CLKCTRL .....	887
3-1215. CM_MEMIF_DLL_CLKCTRL .....	887
3-1216. Register Call Summary for Register CM_MEMIF_DLL_CLKCTRL .....	887
3-1217. CM_C2C_CLKSTCTRL .....	888
3-1218. Register Call Summary for Register CM_C2C_CLKSTCTRL .....	888
3-1219. CM_C2C_STATICDEP .....	889
3-1220. Register Call Summary for Register CM_C2C_STATICDEP .....	889
3-1221. CM_C2C_DYNAMICDEP .....	890
3-1222. Register Call Summary for Register CM_C2C_DYNAMICDEP .....	890
3-1223. CM_C2C_C2C_CLKCTRL .....	890
3-1224. Register Call Summary for Register CM_C2C_C2C_CLKCTRL .....	891
3-1225. CM_C2C_C2C_FW_CLKCTRL .....	891
3-1226. Register Call Summary for Register CM_C2C_C2C_FW_CLKCTRL .....	892
3-1227. CM_L4CFG_CLKSTCTRL .....	892
3-1228. Register Call Summary for Register CM_L4CFG_CLKSTCTRL .....	893
3-1229. CM_L4CFG_DYNAMICDEP .....	893
3-1230. Register Call Summary for Register CM_L4CFG_DYNAMICDEP .....	894
3-1231. CM_L4CFG_L4_CFG_CLKCTRL .....	894
3-1232. Register Call Summary for Register CM_L4CFG_L4_CFG_CLKCTRL .....	895
3-1233. CM_L4CFG_SPINLOCK_CLKCTRL .....	895
3-1234. Register Call Summary for Register CM_L4CFG_SPINLOCK_CLKCTRL .....	895
3-1235. CM_L4CFG_MAILBOX_CLKCTRL .....	896
3-1236. Register Call Summary for Register CM_L4CFG_MAILBOX_CLKCTRL .....	896
3-1237. CM_L4CFG_SAR_ROM_CLKCTRL .....	896
3-1238. Register Call Summary for Register CM_L4CFG_SAR_ROM_CLKCTRL .....	897
3-1239. CM_L3INSTR_CLKSTCTRL .....	897
3-1240. Register Call Summary for Register CM_L3INSTR_CLKSTCTRL .....	898
3-1241. CM_L3INSTR_L3_3_CLKCTRL .....	898
3-1242. Register Call Summary for Register CM_L3INSTR_L3_3_CLKCTRL .....	899
3-1243. CM_L3INSTR_L3_INSTR_CLKCTRL .....	899
3-1244. Register Call Summary for Register CM_L3INSTR_L3_INSTR_CLKCTRL .....	899
3-1245. CM_L3INSTR_INTRCONN_WP1_CLKCTRL .....	900
3-1246. Register Call Summary for Register CM_L3INSTR_INTRCONN_WP1_CLKCTRL .....	900
3-1247. IVAHD_CM2 Registers Mapping Summary .....	900
3-1248. CM_IVAHD_CLKSTCTRL .....	901
3-1249. Register Call Summary for Register CM_IVAHD_CLKSTCTRL .....	901
3-1250. CM_IVAHD_STATICDEP .....	902

3-1251. Register Call Summary for Register CM_IVAHD_STATICDEP .....	902
3-1252. CM_IVAHD_DYNAMICDEP .....	902
3-1253. Register Call Summary for Register CM_IVAHD_DYNAMICDEP .....	903
3-1254. CM_IVAHD_IVAHD_CLKCTRL .....	903
3-1255. Register Call Summary for Register CM_IVAHD_IVAHD_CLKCTRL .....	904
3-1256. CM_IVAHD_SL2_CLKCTRL .....	904
3-1257. Register Call Summary for Register CM_IVAHD_SL2_CLKCTRL .....	904
3-1258. CAM_CM2 Registers Mapping Summary .....	905
3-1259. CM_CAM_CLKSTCTRL .....	905
3-1260. Register Call Summary for Register CM_CAM_CLKSTCTRL .....	906
3-1261. CM_CAM_STATICDEP .....	906
3-1262. Register Call Summary for Register CM_CAM_STATICDEP .....	907
3-1263. CM_CAM_DYNAMICDEP .....	907
3-1264. Register Call Summary for Register CM_CAM_DYNAMICDEP .....	907
3-1265. CM_CAM_ISS_CLKCTRL .....	907
3-1266. Register Call Summary for Register CM_CAM_ISS_CLKCTRL .....	908
3-1267. CM_CAM_FDIF_CLKCTRL .....	909
3-1268. Register Call Summary for Register CM_CAM_FDIF_CLKCTRL .....	909
3-1269. DSS_CM2 Registers Mapping Summary .....	910
3-1270. CM_DSS_CLKSTCTRL .....	910
3-1271. Register Call Summary for Register CM_DSS_CLKSTCTRL .....	911
3-1272. CM_DSS_STATICDEP .....	911
3-1273. Register Call Summary for Register CM_DSS_STATICDEP .....	912
3-1274. CM_DSS_DYNAMICDEP .....	912
3-1275. Register Call Summary for Register CM_DSS_DYNAMICDEP .....	912
3-1276. CM_DSS_DSS_CLKCTRL .....	913
3-1277. Register Call Summary for Register CM_DSS_DSS_CLKCTRL .....	914
3-1278. SGX_CM2 Registers Mapping Summary .....	914
3-1279. CM_SGX_CLKSTCTRL .....	914
3-1280. Register Call Summary for Register CM_SGX_CLKSTCTRL .....	915
3-1281. CM_SGX_STATICDEP .....	915
3-1282. Register Call Summary for Register CM_SGX_STATICDEP .....	916
3-1283. CM_SGX_DYNAMICDEP .....	916
3-1284. Register Call Summary for Register CM_SGX_DYNAMICDEP .....	916
3-1285. CM_SGX_SGX_CLKCTRL .....	917
3-1286. Register Call Summary for Register CM_SGX_SGX_CLKCTRL .....	917
3-1287. L3INIT_CM2 Registers Mapping Summary .....	918
3-1288. CM_L3INIT_CLKSTCTRL .....	918
3-1289. Register Call Summary for Register CM_L3INIT_CLKSTCTRL .....	921
3-1290. CM_L3INIT_STATICDEP .....	921
3-1291. Register Call Summary for Register CM_L3INIT_STATICDEP .....	922
3-1292. CM_L3INIT_DYNAMICDEP .....	922
3-1293. Register Call Summary for Register CM_L3INIT_DYNAMICDEP .....	922
3-1294. CM_L3INIT_HSMMC1_CLKCTRL .....	923
3-1295. Register Call Summary for Register CM_L3INIT_HSMMC1_CLKCTRL .....	923
3-1296. CM_L3INIT_HSMMC2_CLKCTRL .....	924
3-1297. Register Call Summary for Register CM_L3INIT_HSMMC2_CLKCTRL .....	924
3-1298. CM_L3INIT_HSI_CLKCTRL .....	925
3-1299. Register Call Summary for Register CM_L3INIT_HSI_CLKCTRL .....	925

3-1300. CM_L3INIT_HSUSBHOST_CLKCTRL .....	926
3-1301. Register Call Summary for Register CM_L3INIT_HSUSBHOST_CLKCTRL.....	927
3-1302. CM_L3INIT_HSUSBOTG_CLKCTRL.....	928
3-1303. Register Call Summary for Register CM_L3INIT_HSUSBOTG_CLKCTRL .....	929
3-1304. CM_L3INIT_HSUSBTLL_CLKCTRL .....	929
3-1305. Register Call Summary for Register CM_L3INIT_HSUSBTLL_CLKCTRL.....	930
3-1306. CM_L3INIT_P1500_CLKCTRL .....	930
3-1307. Register Call Summary for Register CM_L3INIT_P1500_CLKCTRL.....	931
3-1308. CM_L3INIT_FSUSB_CLKCTRL .....	931
3-1309. Register Call Summary for Register CM_L3INIT_FSUSB_CLKCTRL.....	932
3-1310. CM_L3INIT_USBPHY_CLKCTRL .....	932
3-1311. Register Call Summary for Register CM_L3INIT_USBPHY_CLKCTRL .....	933
3-1312. L4PER_CM2 Registers Mapping Summary.....	933
3-1313. CM_L4PER_CLKSTCTRL.....	934
3-1314. Register Call Summary for Register CM_L4PER_CLKSTCTRL .....	936
3-1315. CM_L4PER_DYNAMICDEP .....	937
3-1316. Register Call Summary for Register CM_L4PER_DYNAMICDEP .....	937
3-1317. CM_L4PER_GPTIMER10_CLKCTRL .....	937
3-1318. Register Call Summary for Register CM_L4PER_GPTIMER10_CLKCTRL.....	938
3-1319. CM_L4PER_GPTIMER11_CLKCTRL .....	938
3-1320. Register Call Summary for Register CM_L4PER_GPTIMER11_CLKCTRL.....	939
3-1321. CM_L4PER_GPTIMER2_CLKCTRL.....	939
3-1322. Register Call Summary for Register CM_L4PER_GPTIMER2_CLKCTRL .....	940
3-1323. CM_L4PER_GPTIMER3_CLKCTRL.....	940
3-1324. Register Call Summary for Register CM_L4PER_GPTIMER3_CLKCTRL .....	941
3-1325. CM_L4PER_GPTIMER4_CLKCTRL.....	941
3-1326. Register Call Summary for Register CM_L4PER_GPTIMER4_CLKCTRL .....	942
3-1327. CM_L4PER_GPTIMER9_CLKCTRL.....	942
3-1328. Register Call Summary for Register CM_L4PER_GPTIMER9_CLKCTRL .....	943
3-1329. CM_L4PER_ELM_CLKCTRL .....	943
3-1330. Register Call Summary for Register CM_L4PER_ELM_CLKCTRL.....	944
3-1331. CM_L4PER_GPIO2_CLKCTRL .....	944
3-1332. Register Call Summary for Register CM_L4PER_GPIO2_CLKCTRL.....	945
3-1333. CM_L4PER_GPIO3_CLKCTRL .....	945
3-1334. Register Call Summary for Register CM_L4PER_GPIO3_CLKCTRL.....	946
3-1335. CM_L4PER_GPIO4_CLKCTRL .....	946
3-1336. Register Call Summary for Register CM_L4PER_GPIO4_CLKCTRL.....	947
3-1337. CM_L4PER_GPIO5_CLKCTRL .....	947
3-1338. Register Call Summary for Register CM_L4PER_GPIO5_CLKCTRL.....	948
3-1339. CM_L4PER_GPIO6_CLKCTRL .....	948
3-1340. Register Call Summary for Register CM_L4PER_GPIO6_CLKCTRL.....	949
3-1341. CM_L4PER_HDQ1W_CLKCTRL.....	949
3-1342. Register Call Summary for Register CM_L4PER_HDQ1W_CLKCTRL .....	950
3-1343. CM_L4PER_I2C1_CLKCTRL.....	950
3-1344. Register Call Summary for Register CM_L4PER_I2C1_CLKCTRL .....	951
3-1345. CM_L4PER_I2C2_CLKCTRL.....	951
3-1346. Register Call Summary for Register CM_L4PER_I2C2_CLKCTRL .....	951
3-1347. CM_L4PER_I2C3_CLKCTRL.....	952
3-1348. Register Call Summary for Register CM_L4PER_I2C3_CLKCTRL .....	952



3-1349. CM_L4PER_I2C4_CLKCTRL .....	952
3-1350. Register Call Summary for Register CM_L4PER_I2C4_CLKCTRL .....	953
3-1351. CM_L4PER_L4PER_CLKCTRL .....	953
3-1352. Register Call Summary for Register CM_L4PER_L4PER_CLKCTRL .....	954
3-1353. CM_L4PER_MCBSP4_CLKCTRL .....	954
3-1354. Register Call Summary for Register CM_L4PER_MCBSP4_CLKCTRL .....	955
3-1355. CM_L4PER_MCSP11_CLKCTRL .....	955
3-1356. Register Call Summary for Register CM_L4PER_MCSP11_CLKCTRL .....	956
3-1357. CM_L4PER_MCSP12_CLKCTRL .....	956
3-1358. Register Call Summary for Register CM_L4PER_MCSP12_CLKCTRL .....	957
3-1359. CM_L4PER_MCSP13_CLKCTRL .....	957
3-1360. Register Call Summary for Register CM_L4PER_MCSP13_CLKCTRL .....	958
3-1361. CM_L4PER_MCSP14_CLKCTRL .....	958
3-1362. Register Call Summary for Register CM_L4PER_MCSP14_CLKCTRL .....	958
3-1363. CM_L4PER_MMCS3_CLKCTRL .....	959
3-1364. Register Call Summary for Register CM_L4PER_MMCS3_CLKCTRL .....	959
3-1365. CM_L4PER_MMCS4_CLKCTRL .....	959
3-1366. Register Call Summary for Register CM_L4PER_MMCS4_CLKCTRL .....	960
3-1367. CM_L4PER_SLIMBUS2_CLKCTRL .....	960
3-1368. Register Call Summary for Register CM_L4PER_SLIMBUS2_CLKCTRL .....	961
3-1369. CM_L4PER_UART1_CLKCTRL .....	962
3-1370. Register Call Summary for Register CM_L4PER_UART1_CLKCTRL .....	962
3-1371. CM_L4PER_UART2_CLKCTRL .....	962
3-1372. Register Call Summary for Register CM_L4PER_UART2_CLKCTRL .....	963
3-1373. CM_L4PER_UART3_CLKCTRL .....	963
3-1374. Register Call Summary for Register CM_L4PER_UART3_CLKCTRL .....	964
3-1375. CM_L4PER_UART4_CLKCTRL .....	964
3-1376. Register Call Summary for Register CM_L4PER_UART4_CLKCTRL .....	965
3-1377. CM_L4PER_MMCS5_CLKCTRL .....	965
3-1378. Register Call Summary for Register CM_L4PER_MMCS5_CLKCTRL .....	966
3-1379. RESTORE_CM2 Registers Mapping Summary .....	966
3-1380. CM_L3_1_CLKSTCTRL_RESTORE .....	967
3-1381. Register Call Summary for Register CM_L3_1_CLKSTCTRL_RESTORE .....	967
3-1382. CM_L3_2_CLKSTCTRL_RESTORE .....	967
3-1383. Register Call Summary for Register CM_L3_2_CLKSTCTRL_RESTORE .....	968
3-1384. CM_L4CFG_CLKSTCTRL_RESTORE .....	968
3-1385. Register Call Summary for Register CM_L4CFG_CLKSTCTRL_RESTORE .....	969
3-1386. CM_MEMIF_CLKSTCTRL_RESTORE .....	969
3-1387. Register Call Summary for Register CM_MEMIF_CLKSTCTRL_RESTORE .....	970
3-1388. CM_L4PER_CLKSTCTRL_RESTORE .....	970
3-1389. Register Call Summary for Register CM_L4PER_CLKSTCTRL_RESTORE .....	972
3-1390. CM_L3INIT_CLKSTCTRL_RESTORE .....	973
3-1391. Register Call Summary for Register CM_L3INIT_CLKSTCTRL_RESTORE .....	975
3-1392. CM_L3INSTR_L3_3_CLKCTRL_RESTORE .....	975
3-1393. Register Call Summary for Register CM_L3INSTR_L3_3_CLKCTRL_RESTORE .....	976
3-1394. CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE .....	976
3-1395. Register Call Summary for Register CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE .....	977
3-1396. CM_L3INSTR_INTRCONN_WP1_CLKCTRL_RESTORE .....	977
3-1397. Register Call Summary for Register CM_L3INSTR_INTRCONN_WP1_CLKCTRL_RESTORE .....	978

3-1398. CM_CM2_PROFILING_CLKCTRL_RESTORE .....	978
3-1399. Register Call Summary for Register CM_CM2_PROFILING_CLKCTRL_RESTORE .....	979
3-1400. CM_C2C_STATICDEP_RESTORE .....	979
3-1401. Register Call Summary for Register CM_C2C_STATICDEP_RESTORE .....	980
3-1402. CM_L3_1_DYNAMICDEP_RESTORE .....	980
3-1403. Register Call Summary for Register CM_L3_1_DYNAMICDEP_RESTORE .....	980
3-1404. CM_L3_2_DYNAMICDEP_RESTORE .....	981
3-1405. Register Call Summary for Register CM_L3_2_DYNAMICDEP_RESTORE .....	982
3-1406. CM_C2C_DYNAMICDEP_RESTORE .....	982
3-1407. Register Call Summary for Register CM_C2C_DYNAMICDEP_RESTORE .....	982
3-1408. CM_L4CFG_DYNAMICDEP_RESTORE .....	982
3-1409. Register Call Summary for Register CM_L4CFG_DYNAMICDEP_RESTORE .....	983
3-1410. CM_L4PER_DYNAMICDEP_RESTORE .....	984
3-1411. Register Call Summary for Register CM_L4PER_DYNAMICDEP_RESTORE .....	984
3-1412. CM_L4PER_GPIO2_CLKCTRL_RESTORE .....	984
3-1413. Register Call Summary for Register CM_L4PER_GPIO2_CLKCTRL_RESTORE .....	985
3-1414. CM_L4PER_GPIO3_CLKCTRL_RESTORE .....	985
3-1415. Register Call Summary for Register CM_L4PER_GPIO3_CLKCTRL_RESTORE .....	986
3-1416. CM_L4PER_GPIO4_CLKCTRL_RESTORE .....	986
3-1417. Register Call Summary for Register CM_L4PER_GPIO4_CLKCTRL_RESTORE .....	987
3-1418. CM_L4PER_GPIO5_CLKCTRL_RESTORE .....	987
3-1419. Register Call Summary for Register CM_L4PER_GPIO5_CLKCTRL_RESTORE .....	988
3-1420. CM_L4PER_GPIO6_CLKCTRL_RESTORE .....	988
3-1421. Register Call Summary for Register CM_L4PER_GPIO6_CLKCTRL_RESTORE .....	989
3-1422. CM_L3INIT_HSUSBHOST_CLKCTRL_RESTORE .....	989
3-1423. Register Call Summary for Register CM_L3INIT_HSUSBHOST_CLKCTRL_RESTORE .....	991
3-1424. CM_L3INIT_HSUSBTLL_CLKCTRL_RESTORE .....	991
3-1425. Register Call Summary for Register CM_L3INIT_HSUSBTLL_CLKCTRL_RESTORE .....	992
3-1426. CM_SDMA_STATICDEP_RESTORE .....	992
3-1427. Register Call Summary for Register CM_SDMA_STATICDEP_RESTORE .....	993
3-1428. SCRM Instance Summary .....	994
3-1429. SCRM Registers Mapping Summary .....	994
3-1430. REVISION_SCRM .....	995
3-1431. Register Call Summary for Register REVISION_SCRM .....	995
3-1432. CLKSETUPTIME .....	995
3-1433. Register Call Summary for Register CLKSETUPTIME .....	995
3-1434. PMICSETUPTIME .....	996
3-1435. Register Call Summary for Register PMICSETUPTIME .....	996
3-1436. ALTCLKSRC .....	996
3-1437. Register Call Summary for Register ALTCLKSRC .....	997
3-1438. C2CCLKM .....	997
3-1439. Register Call Summary for Register C2CCLKM .....	997
3-1440. EXTCLKREQ .....	998
3-1441. Register Call Summary for Register EXTCLKREQ .....	998
3-1442. ACCCLKREQ .....	998
3-1443. Register Call Summary for Register ACCCLKREQ .....	998
3-1444. PWRREQ .....	999
3-1445. Register Call Summary for Register PWRREQ .....	999
3-1446. AUXCLKREQ0 .....	999

3-1447. Register Call Summary for Register AUXCLKREQ0 .....	1000
3-1448. AUXCLKREQ1 .....	1000
3-1449. Register Call Summary for Register AUXCLKREQ1 .....	1001
3-1450. AUXCLKREQ2 .....	1001
3-1451. Register Call Summary for Register AUXCLKREQ2 .....	1002
3-1452. AUXCLKREQ3 .....	1002
3-1453. Register Call Summary for Register AUXCLKREQ3 .....	1003
3-1454. AUXCLKREQ4 .....	1003
3-1455. Register Call Summary for Register AUXCLKREQ4 .....	1004
3-1456. AUXCLKREQ5 .....	1004
3-1457. Register Call Summary for Register AUXCLKREQ5 .....	1005
3-1458. C2CCLKREQ .....	1005
3-1459. Register Call Summary for Register C2CCLKREQ.....	1006
3-1460. AUXCLK0 .....	1006
3-1461. Register Call Summary for Register AUXCLK0.....	1007
3-1462. AUXCLK1 .....	1007
3-1463. Register Call Summary for Register AUXCLK1 .....	1008
3-1464. AUXCLK2 .....	1009
3-1465. Register Call Summary for Register AUXCLK2.....	1010
3-1466. AUXCLK3 .....	1010
3-1467. Register Call Summary for Register AUXCLK3.....	1011
3-1468. AUXCLK4 .....	1011
3-1469. Register Call Summary for Register AUXCLK4.....	1012
3-1470. AUXCLK5 .....	1012
3-1471. Register Call Summary for Register AUXCLK5.....	1013
3-1472. RSTTIME_REG .....	1013
3-1473. Register Call Summary for Register RSTTIME_REG .....	1013
3-1474. C2CRSTCTRL.....	1014
3-1475. Register Call Summary for Register C2CRSTCTRL .....	1014
3-1476. EXTPWRONRSTCTRL.....	1014
3-1477. Register Call Summary for Register EXTPWRONRSTCTRL .....	1015
3-1478. EXTWARMRSTST_REG.....	1015
3-1479. Register Call Summary for Register EXTWARMRSTST_REG .....	1015
3-1480. APEWARMRSTST_REG .....	1016
3-1481. Register Call Summary for Register APEWARMRSTST_REG .....	1016
3-1482. C2CWARMRSTST_REG .....	1016
3-1483. Register Call Summary for Register C2CWARMRSTST_REG .....	1017
3-1484. SR Instance Summary .....	1018
3-1485. SR Registers Mapping Summary .....	1018
3-1486. SRCONFIG .....	1019
3-1487. Register Call Summary for Register SRCONFIG .....	1019
3-1488. SRSTATUS .....	1020
3-1489. Register Call Summary for Register SRSTATUS.....	1020
3-1490. SENVAL .....	1020
3-1491. Register Call Summary for Register SENVAL .....	1021
3-1492. SENMIN .....	1021
3-1493. Register Call Summary for Register SENMIN.....	1021
3-1494. SENMAX .....	1021
3-1495. Register Call Summary for Register SENMAX.....	1021

3-1496. SENAVG .....	1022
3-1497. Register Call Summary for Register SENAVG .....	1022
3-1498. AVGWEIGHT .....	1022
3-1499. Register Call Summary for Register AVGWEIGHT .....	1022
3-1500. NVALUERECIPROCAL .....	1023
3-1501. Register Call Summary for Register NVALUERECIPROCAL .....	1023
3-1502. IRQSTATUS_RAW .....	1023
3-1503. Register Call Summary for Register IRQSTATUS_RAW .....	1024
3-1504. IRQSTATUS .....	1024
3-1505. Register Call Summary for Register IRQSTATUS .....	1024
3-1506. IRQENABLE_SET .....	1025
3-1507. Register Call Summary for Register IRQENABLE_SET .....	1025
3-1508. IRQENABLE_CLR .....	1026
3-1509. Register Call Summary for Register IRQENABLE_CLR .....	1026
3-1510. SENERROR .....	1026
3-1511. Register Call Summary for Register SENERROR .....	1027
3-1512. ERRCONFIG .....	1027
3-1513. Register Call Summary for Register ERRCONFIG .....	1027
4-1. Integration Attributes .....	1034
4-2. Clocks and Resets .....	1034
4-3. ARM Core Key Features .....	1039
4-4. CPUx Supported Power States .....	1042
4-5. MPU Power States Handled by the Local PRCM and Global (Device) PRCM .....	1043
4-6. PD_CPUx Modules Power Attributes .....	1044
4-7. PD_CPUx Logic Area Power Modes .....	1045
4-8. PD_CPUx Memory Area Power Modes .....	1045
4-9. PD_CPUx Power Modes Control Parameters .....	1045
4-10. PD_CPUx Power Modes Status Parameters .....	1045
4-11. Modules Clock Management Modes and Control .....	1046
4-12. Modules Slave Clock Management Modes and Control .....	1046
4-13. Cortex-A9 MPU Instance Summary .....	1047
4-14. Local PRCM Revision Register Mapping Summary .....	1047
4-15. CORTEXA9_PRCM_REVISION .....	1048
4-16. Register Call Summary for Register CORTEXA9_PRCM_REVISION .....	1048
4-17. Local PRCM Registers Mapping Summary .....	1048
4-18. PRM_RSTST .....	1048
4-19. Register Call Summary for Register PRM_RSTST .....	1049
4-20. PRM_PSCON_COUNT .....	1049
4-21. Register Call Summary for Register PRM_PSCON_COUNT .....	1049
4-22. Local PRCM CPU0 and CPU1 Register Mapping Summary .....	1049
4-23. PM_PDA_CPUi_PWRSTCTRL .....	1050
4-24. Register Call Summary for Register PM_PDA_CPUi_PWRSTCTRL .....	1051
4-25. PM_PDA_CPUi_PWRSTST .....	1051
4-26. Register Call Summary for Register PM_PDA_CPUi_PWRSTST .....	1052
4-27. RM_PDA_CPUi_CONTEXT .....	1052
4-28. Register Call Summary for Register RM_PDA_CPUi_CONTEXT .....	1052
4-29. RM_PDA_CPUi_RSTCTRL .....	1053
4-30. Register Call Summary for Register RM_PDA_CPUi_RSTCTRL .....	1053
4-31. CM_PDA_CPUi_CLKCTRL .....	1053

4-32.	Register Call Summary for Register CM_PDA_CPUi_CLKCTRL.....	1053
4-33.	CM_PDA_CPUi_CLKSTCTRL.....	1054
4-34.	Register Call Summary for Register CM_PDA_CPUi_CLKSTCTRL .....	1054
4-35.	CORTEXA9_WUGEN Registers Mapping Summary .....	1054
4-36.	WKG_CONTROL_0.....	1055
4-37.	Register Call Summary for Register WKG_CONTROL_0 .....	1056
4-38.	WKG_ENB_A_0 .....	1056
4-39.	Register Call Summary for Register WKG_ENB_A_0.....	1057
4-40.	WKG_ENB_B_0 .....	1058
4-41.	Register Call Summary for Register WKG_ENB_B_0.....	1059
4-42.	WKG_ENB_C_0 .....	1059
4-43.	Register Call Summary for Register WKG_ENB_C_0.....	1060
4-44.	WKG_ENB_D_0 .....	1060
4-45.	Register Call Summary for Register WKG_ENB_D_0.....	1061
4-46.	WKG_CONTROL_1 .....	1061
4-47.	Register Call Summary for Register WKG_CONTROL_1 .....	1062
4-48.	WKG_ENB_A_1 .....	1062
4-49.	Register Call Summary for Register WKG_ENB_A_1 .....	1063
4-50.	WKG_ENB_B_1 .....	1064
4-51.	Register Call Summary for Register WKG_ENB_B_1.....	1065
4-52.	WKG_ENB_C_1 .....	1065
4-53.	Register Call Summary for Register WKG_ENB_C_1.....	1066
4-54.	WKG_ENB_D_1 .....	1066
4-55.	Register Call Summary for Register WKG_ENB_D_1.....	1067
6-1.	IVA-HD Integration Attributes .....	1074
6-2.	IVA-HD Clocks and Resets .....	1074
6-3.	Hardware Requests .....	1075
6-4.	Local Power-Management Features .....	1078
6-5.	ICONT Memory Mapping Regions .....	1079
6-6.	ICONT1 Memory Mapping .....	1079
6-7.	ICONT2 Memory Mapping .....	1080
6-8.	DSP and L3 Interconnect Memory Mapping.....	1082
6-9.	IVA-HD Instance Summary .....	1083
6-10.	SYSCTRL Registers Mapping Summary.....	1083
6-11.	IVAHD_REVISION .....	1084
6-12.	Register Call Summary for Register IVAHD_REVISION.....	1084
6-13.	IVAHD_HWINFO .....	1084
6-14.	Register Call Summary for Register IVAHD_HWINFO.....	1085
6-15.	IVAHD_SYSCONFIG .....	1085
6-16.	Register Call Summary for Register IVAHD_SYSCONFIG .....	1086
6-17.	IVAHD_IRQSTATUS_RAW .....	1086
6-18.	Register Call Summary for Register IVAHD_IRQSTATUS_RAW.....	1087
6-19.	IVAHD_IRQSTATUS.....	1087
6-20.	Register Call Summary for Register IVAHD_IRQSTATUS .....	1087
6-21.	IVAHD_IRQENABLE_SET .....	1087
6-22.	Register Call Summary for Register IVAHD_IRQENABLE_SET.....	1088
6-23.	IVAHD_IRQENABLE_CLR.....	1088
6-24.	Register Call Summary for Register IVAHD_IRQENABLE_CLR .....	1088
6-25.	IVAHD_SYNC_IRQSTATUS_RAW .....	1089

6-26.	Register Call Summary for Register IVAHD_SYNC_IRQSTATUS_RAW.....	1089
6-27.	IVAHD_SYNC_IRQSTATUS.....	1089
6-28.	Register Call Summary for Register IVAHD_SYNC_IRQSTATUS .....	1089
6-29.	IVAHD_SYNC_IRQENABLE_SET .....	1090
6-30.	Register Call Summary for Register IVAHD_SYNC_IRQENABLE_SET.....	1090
6-31.	IVAHD_SYNC_IRQENABLE_CLR.....	1090
6-32.	Register Call Summary for Register IVAHD_SYNC_IRQENABLE_CLR .....	1090
6-33.	IVAHD_CLKCTRL .....	1091
6-34.	Register Call Summary for Register IVAHD_CLKCTRL .....	1091
6-35.	IVAHD_CLKST.....	1092
6-36.	Register Call Summary for Register IVAHD_CLKST .....	1092
6-37.	IVAHD_STDBYST.....	1093
6-38.	Register Call Summary for Register IVAHD_STDBYST .....	1093
7-1.	Integration Attributes .....	1098
7-2.	Hardware Requests .....	1098
7-3.	Clocks and Resets .....	1099
7-4.	Local Clock Gating .....	1104
7-5.	Dual Cortex-M3 MPU Power Modes.....	1105
7-6.	Power Mode Transitions.....	1105
7-7.	Shared Cache Configuration.....	1107
7-8.	Shared cache MMU Configuration .....	1108
7-9.	L2 MMU Behavior on Page-Fault .....	1109
7-10.	Dual Cortex-M3 MPU Subsystem Instance Summary.....	1112
7-11.	CORTEXM3_WKUP Register Mapping Summary .....	1113
7-12.	CORTEXM3_CTRL_REG .....	1113
7-13.	Register Call Summary for Register CORTEXM3_CTRL_REG.....	1113
7-14.	STANDBY_CORE_SYSCONFIG.....	1114
7-15.	Register Call Summary for Register STANDBY_CORE_SYSCONFIG .....	1114
7-16.	IDLE_CORE_SYSCONFIG .....	1114
7-17.	Register Call Summary for Register IDLE_CORE_SYSCONFIG.....	1115
7-18.	WUGEN_MEVT0 .....	1115
7-19.	Register Call Summary for Register WUGEN_MEVT0.....	1116
7-20.	WUGEN_MEVT1 .....	1116
7-21.	Register Call Summary for Register WUGEN_MEVT1.....	1117
7-22.	CM3_RW_Table Register Summary.....	1117
7-23.	CORTEXM3_RW_PID1 .....	1117
7-24.	Register Call Summary for Register CORTEXM3_RW_PID1 .....	1117
7-25.	CORTEXM3_RW_PID2 .....	1118
7-26.	Register Call Summary for Register CORTEXM3_RW_PID2 .....	1118
8-1.	ISS Integration Attributes.....	1124
8-2.	ISS Clocks and Resets .....	1124
8-3.	ISS Hardware Requests .....	1124
8-4.	ISS Local Clock Domains .....	1125
8-5.	ISS Interrupts .....	1126
8-6.	ISS ISP Interrupts .....	1128
8-7.	ISS CSI2-A and CSI2-B Interrupts .....	1129
8-8.	ISS CSI2-A and CSI2-B Receivers Complex I/O Interrupts .....	1129
8-9.	ISS CSI2-A/CSI2-B Receivers CONTEXT Interrupts .....	1130
8-10.	ISS CCP2 Receiver Interrupts .....	1130

8-11.	ISS CBUFF Interrupts.....	1132
8-12.	ISS BTE Interrupts .....	1133
8-13.	ISS SIMCOP High-Level Interrupts .....	1136
8-14.	ISS Submodule Clock Gating.....	1137
8-15.	ISS Instance Summary .....	1141
8-16.	ISS TOP Register Mapping Summary .....	1141
8-17.	ISS_HL_REVISION .....	1142
8-18.	Register Call Summary for Register ISS_HL_REVISION.....	1142
8-19.	ISS_HL_SYSCONFIG .....	1142
8-20.	Register Call Summary for Register ISS_HL_SYSCONFIG.....	1143
8-21.	ISS_HL_IRQSTATUS_RAW_i.....	1143
8-22.	Register Call Summary for Register ISS_HL_IRQSTATUS_RAW_i .....	1144
8-23.	ISS_HL_IRQSTATUS_i.....	1145
8-24.	Register Call Summary for Register ISS_HL_IRQSTATUS_i .....	1146
8-25.	ISS_HL_IRQENABLE_SET_i.....	1146
8-26.	Register Call Summary for Register ISS_HL_IRQENABLE_SET_i .....	1149
8-27.	ISS_HL_IRQENABLE_CLR_i.....	1149
8-28.	Register Call Summary for Register ISS_HL_IRQENABLE_CLR_i .....	1151
8-29.	ISS_CTRL .....	1151
8-30.	Register Call Summary for Register ISS_CTRL.....	1152
8-31.	ISS_CLKCTRL.....	1152
8-32.	Register Call Summary for Register ISS_CLKCTRL .....	1153
8-33.	ISS_CLKSTAT .....	1153
8-34.	Register Call Summary for Register ISS_CLKSTAT .....	1154
8-35.	ISS_PM_STATUS.....	1154
8-36.	Register Call Summary for Register ISS_PM_STATUS .....	1155
8-37.	ISS I/O Description .....	1160
8-38.	ISS Interfaces Connectivity Scheme Example Scenarios.....	1163
8-39.	ISS CSI2 Possible Time-Out Value for RxMode Counter.....	1166
8-40.	ISS CSI2 PHY Instance Summary .....	1171
8-41.	ISS CSI2 PHY Registers Mapping Summary.....	1171
8-42.	REGISTER0.....	1171
8-43.	Register Call Summary for Register REGISTER0 .....	1172
8-44.	REGISTER1.....	1172
8-45.	Register Call Summary for Register REGISTER1 .....	1173
8-46.	REGISTER2.....	1173
8-47.	Register Call Summary for Register REGISTER2 .....	1174
8-48.	I/O Description for Serial Interface CCP2 .....	1174
8-49.	ISS CCP2 Image Data Operating Modes and Alignment Constraints.....	1175
8-50.	ISS CCP2 Video interface Signals .....	1190
8-51.	ISS CCP2 Video Port Data Mapping .....	1191
8-52.	ISS CCP2 Memory-to-Memory Supported Operations .....	1193
8-53.	ISS CCP2 Memory-to-Video Port Supported Formats .....	1194
8-54.	ISS CCP2 Data Packing Benefit and Constraints.....	1197
8-55.	ISS CCP2 Output Width Restrictions in Memory-to-Memory Operation .....	1199
8-56.	ISS CCP2 Interface Enable .....	1201
8-57.	ISS CCP2 Interface Disable .....	1201
8-58.	ISS CCP2 Select the Signaling Scheme .....	1202
8-59.	ISS CCP2 Select MIPI CSI1 or CCP2 Mode .....	1202

8-60.	ISS CCP2 Configure Burst Settings .....	1202
8-61.	ISS CCP2 Enable Debug Mode .....	1202
8-62.	ISS CCP2 Configure Video Port.....	1203
8-63.	ISS CCP2 Enable Logical Channels.....	1204
8-64.	ISS CCP2 Enable Region-of-Interest .....	1204
8-65.	ISS CCP2 Enable the CRC .....	1204
8-66.	ISS CCP2 Frame Acquisition .....	1205
8-67.	ISS CCP2 Configure Read Data From Memory.....	1208
8-68.	ISS CCP2 Instance Summary .....	1209
8-69.	ISS CCP2 Registers Mapping Summary .....	1210
8-70.	CCP2_REVISION .....	1211
8-71.	Register Call Summary for Register CCP2_REVISION .....	1211
8-72.	CCP2_SYSCONFIG .....	1211
8-73.	Register Call Summary for Register CCP2_SYSCONFIG .....	1212
8-74.	CCP2_SYSSTATUS .....	1212
8-75.	Register Call Summary for Register CCP2_SYSSTATUS.....	1212
8-76.	CCP2_LC01_IRQENABLE .....	1213
8-77.	Register Call Summary for Register CCP2_LC01_IRQENABLE .....	1214
8-78.	CCP2_LC01_IRQSTATUS .....	1215
8-79.	Register Call Summary for Register CCP2_LC01_IRQSTATUS .....	1217
8-80.	CCP2_LC23_IRQENABLE .....	1217
8-81.	Register Call Summary for Register CCP2_LC23_IRQENABLE .....	1219
8-82.	CCP2_LC23_IRQSTATUS .....	1220
8-83.	Register Call Summary for Register CCP2_LC23_IRQSTATUS .....	1222
8-84.	CCP2_LCM_IRQENABLE.....	1222
8-85.	Register Call Summary for Register CCP2_LCM_IRQENABLE .....	1223
8-86.	CCP2_LCM_IRQSTATUS.....	1223
8-87.	Register Call Summary for Register CCP2_LCM_IRQSTATUS .....	1224
8-88.	CCP2_CTRL .....	1224
8-89.	Register Call Summary for Register CCP2_CTRL.....	1225
8-90.	CCP2_DBG .....	1226
8-91.	Register Call Summary for Register CCP2_DBG .....	1226
8-92.	CCP2_GNQ .....	1226
8-93.	Register Call Summary for Register CCP2_GNQ.....	1227
8-94.	CCP2_CTRL1.....	1227
8-95.	Register Call Summary for Register CCP2_CTRL1 .....	1229
8-96.	CCP2_LCx_CTRL .....	1229
8-97.	Register Call Summary for Register CCP2_LCx_CTRL .....	1231
8-98.	CCP2_LCx_CODE .....	1231
8-99.	Register Call Summary for Register CCP2_LCx_CODE.....	1231
8-100.	CCP2_LCx_STAT_START .....	1232
8-101.	Register Call Summary for Register CCP2_LCx_STAT_START .....	1232
8-102.	CCP2_LCx_STAT_SIZE.....	1232
8-103.	Register Call Summary for Register CCP2_LCx_STAT_SIZE .....	1232
8-104.	CCP2_LCx_SOF_ADDR .....	1233
8-105.	Register Call Summary for Register CCP2_LCx_SOF_ADDR.....	1233
8-106.	CCP2_LCx_EOF_ADDR .....	1233
8-107.	Register Call Summary for Register CCP2_LCx_EOF_ADDR .....	1233
8-108.	CCP2_LCx_DAT_START .....	1234



8-109. Register Call Summary for Register CCP2_LCx_DAT_START .....	1234
8-110. CCP2_LCx_DAT_SIZE .....	1234
8-111. Register Call Summary for Register CCP2_LCx_DAT_SIZE .....	1234
8-112. CCP2_LCx_DAT_PING_ADDR .....	1235
8-113. Register Call Summary for Register CCP2_LCx_DAT_PING_ADDR .....	1235
8-114. CCP2_LCx_DAT_PONG_ADDR .....	1235
8-115. Register Call Summary for Register CCP2_LCx_DAT_PONG_ADDR .....	1235
8-116. CCP2_LCx_DAT_OFST .....	1236
8-117. Register Call Summary for Register CCP2_LCx_DAT_OFST .....	1236
8-118. CCP2_LCM_CTRL .....	1236
8-119. Register Call Summary for Register CCP2_LCM_CTRL .....	1239
8-120. CCP2_LCM_VSIZE .....	1239
8-121. Register Call Summary for Register CCP2_LCM_VSIZE .....	1239
8-122. CCP2_LCM_HSIZE .....	1239
8-123. Register Call Summary for Register CCP2_LCM_HSIZE .....	1240
8-124. CCP2_LCM_PREFETCH .....	1240
8-125. Register Call Summary for Register CCP2_LCM_PREFETCH .....	1240
8-126. CCP2_LCM_SRC_ADDR .....	1241
8-127. Register Call Summary for Register CCP2_LCM_SRC_ADDR .....	1241
8-128. CCP2_LCM_SRC_OFST .....	1241
8-129. Register Call Summary for Register CCP2_LCM_SRC_OFST .....	1241
8-130. CCP2_LCM_DST_ADDR .....	1242
8-131. Register Call Summary for Register CCP2_LCM_DST_ADDR .....	1242
8-132. CCP2_LCM_DST_OFST .....	1242
8-133. Register Call Summary for Register CCP2_LCM_DST_OFST .....	1242
8-134. CCP2_LCM_HISTORY .....	1243
8-135. Register Call Summary for Register CCP2_LCM_HISTORY .....	1243
8-136. ISS CSI2 Pixel Format Modes .....	1244
8-137. ISS CSI2-A I/O Description .....	1245
8-138. ISS CSI2-B I/O Description .....	1245
8-139. ISS CSI2 Long Packet Structure Description .....	1250
8-140. ISS CSI2 Synchronization Codes .....	1251
8-141. ISS CSI2 MIPI Format Supported by the Protocol Engine .....	1275
8-142. ISS CSI2 ECC Event Logging .....	1282
8-143. ISS CSI2 Supported Transcoding Input Formats .....	1286
8-144. ISS CSI2 Transcode Alignment Constraints .....	1287
8-145. ISS CSI2-Supported Transcoding Output Formats .....	1288
8-146. ISS CSI2 Global Initialization .....	1291
8-147. ISS CSI2 Capture a Finite Number of Frames .....	1292
8-148. ISS CSI2 Enable Debug Mode .....	1293
8-149. ISS CSI2 Instance Summary .....	1294
8-150. ISS CSI2 REGS1 Registers Mapping Summary .....	1294
8-151. CSI2_REVISION .....	1295
8-152. Register Call Summary for Register CSI2_REVISION .....	1295
8-153. CSI2_SYSCONFIG .....	1295
8-154. Register Call Summary for Register CSI2_SYSCONFIG .....	1296
8-155. CSI2_SYSSTATUS .....	1296
8-156. Register Call Summary for Register CSI2_SYSSTATUS .....	1296
8-157. CSI2_IRQSTATUS .....	1297

8-158. Register Call Summary for Register CSI2_IRQSTATUS.....	1298
8-159. CSI2_IRQENABLE .....	1298
8-160. Register Call Summary for Register CSI2_IRQENABLE.....	1300
8-161. CSI2_CTRL .....	1300
8-162. Register Call Summary for Register CSI2_CTRL .....	1302
8-163. CSI2_DBG_H .....	1302
8-164. Register Call Summary for Register CSI2_DBG_H.....	1302
8-165. CSI2_COMPLEXIO_CFG .....	1303
8-166. Register Call Summary for Register CSI2_COMPLEXIO_CFG.....	1304
8-167. CSI2_COMPLEXIO_IRQSTATUS .....	1305
8-168. Register Call Summary for Register CSI2_COMPLEXIO_IRQSTATUS .....	1307
8-169. CSI2_SHORT_PACKET.....	1308
8-170. Register Call Summary for Register CSI2_SHORT_PACKET .....	1308
8-171. CSI2_COMPLEXIO_IRQENABLE .....	1308
8-172. Register Call Summary for Register CSI2_COMPLEXIO_IRQENABLE .....	1310
8-173. CSI2_DBG_P .....	1310
8-174. Register Call Summary for Register CSI2_DBG_P.....	1310
8-175. CSI2_TIMING .....	1311
8-176. Register Call Summary for Register CSI2_TIMING.....	1311
8-177. CSI2_CTX_CTRL1_i.....	1312
8-178. Register Call Summary for Register CSI2_CTX_CTRL1_i .....	1314
8-179. CSI2_CTX_CTRL2_i.....	1315
8-180. Register Call Summary for Register CSI2_CTX_CTRL2_i .....	1318
8-181. CSI2_CTX_DAT_OFST_i .....	1318
8-182. Register Call Summary for Register CSI2_CTX_DAT_OFST_i.....	1318
8-183. CSI2_CTX_DAT_PING_ADDR_i.....	1319
8-184. Register Call Summary for Register CSI2_CTX_DAT_PING_ADDR_i .....	1319
8-185. CSI2_CTX_DAT_PONG_ADDR_i .....	1319
8-186. Register Call Summary for Register CSI2_CTX_DAT_PONG_ADDR_i.....	1319
8-187. CSI2_CTX_IRQENABLE_i .....	1320
8-188. Register Call Summary for Register CSI2_CTX_IRQENABLE_i.....	1320
8-189. CSI2_CTX_IRQSTATUS_i.....	1321
8-190. Register Call Summary for Register CSI2_CTX_IRQSTATUS_i.....	1322
8-191. CSI2_CTX_CTRL3_i.....	1322
8-192. Register Call Summary for Register CSI2_CTX_CTRL3_i .....	1322
8-193. ISS CSI2 REGS2 Registers Mapping Summary .....	1323
8-194. CSI2_CTX_TRANSCODEH_i.....	1323
8-195. Register Call Summary for Register CSI2_CTX_TRANSCODEH_i.....	1323
8-196. CSI2_CTX_TRANSCODEV_i.....	1323
8-197. Register Call Summary for Register CSI2_CTX_TRANSCODEV_i .....	1324
8-198. ISS TCTRL Control-Signal Generator: CNTCLK Frequencies .....	1328
8-199. ISS TCTRL Enabling the Control-Signal Generation in First Configuration .....	1330
8-200. ISS TCTRL Enabling the Control-Signal Generation in Second Configuration .....	1331
8-201. ISS TCTRL Instance Summary .....	1333
8-202. ISS TCTRL Registers Mapping Summary .....	1333
8-203. TCTRL_REVISION.....	1334
8-204. Register Call Summary for Register TCTRL_REVISION .....	1334
8-205. TCTRL_SYSCONFIG .....	1334
8-206. Register Call Summary for Register TCTRL_SYSCONFIG .....	1335

8-207. TCTRL_SYSSTATUS.....	1335
8-208. Register Call Summary for Register TCTRL_SYSSTATUS .....	1335
8-209. TCTRL_STRB_LENGTH .....	1335
8-210. Register Call Summary for Register TCTRL_STRB_LENGTH.....	1336
8-211. TCTRL_PSTRB_LENGTH .....	1336
8-212. Register Call Summary for Register TCTRL_PSTRB_LENGTH.....	1336
8-213. TCTRL_SHUT_LENGTH .....	1336
8-214. Register Call Summary for Register TCTRL_SHUT_LENGTH.....	1337
8-215. TCTRL_GRESET_LENGTH .....	1337
8-216. Register Call Summary for Register TCTRL_GRESET_LENGTH.....	1337
8-217. TCTRL_STRB_DELAY .....	1337
8-218. Register Call Summary for Register TCTRL_STRB_DELAY .....	1338
8-219. TCTRL_PSTRB_DELAY .....	1338
8-220. Register Call Summary for Register TCTRL_PSTRB_DELAY .....	1338
8-221. TCTRL_SHUT_DELAY .....	1338
8-222. Register Call Summary for Register TCTRL_SHUT_DELAY.....	1339
8-223. TCTRL_CTRL.....	1339
8-224. Register Call Summary for Register TCTRL_CTRL .....	1340
8-225. TCTRL_PSTRB_REPLAY.....	1341
8-226. Register Call Summary for Register TCTRL_PSTRB_REPLAY .....	1341
8-227. TCTRL_FRAME .....	1341
8-228. Register Call Summary for Register TCTRL_FRAME.....	1342
8-229. ISS BTE TILER Context Configuration Example.....	1349
8-230. ISS BTE Supported TILER Formats and Views.....	1353
8-231. ISS BTE Interrupt Events.....	1356
8-232. ISS BTE Configure/Manage Interrupts .....	1356
8-233. TILER Address Format .....	1357
8-234. ISS BTE Change Context Configuration .....	1357
8-235. ISS BTE Instance Summary .....	1358
8-236. ISS BTE Registers Mapping Summary .....	1358
8-237. BTE_HL_REVISION .....	1359
8-238. Register Call Summary for Register BTE_HL_REVISION.....	1359
8-239. BTE_HL_HWINFO .....	1360
8-240. Register Call Summary for Register BTE_HL_HWINFO.....	1360
8-241. BTE_HL_SYSCONFIG .....	1360
8-242. Register Call Summary for Register BTE_HL_SYSCONFIG .....	1361
8-243. BTE_HL_IRQSTATUS_RAW .....	1361
8-244. Register Call Summary for Register BTE_HL_IRQSTATUS_RAW.....	1364
8-245. BTE_HL_IRQSTATUS.....	1365
8-246. Register Call Summary for Register BTE_HL_IRQSTATUS .....	1368
8-247. BTE_HL_IRQENABLE_SET .....	1368
8-248. Register Call Summary for Register BTE_HL_IRQENABLE_SET.....	1371
8-249. BTE_HL_IRQENABLE_CLR.....	1372
8-250. Register Call Summary for Register BTE_HL_IRQENABLE_CLR .....	1375
8-251. BTE_CTRL .....	1375
8-252. Register Call Summary for Register BTE_CTRL.....	1376
8-253. BTE_CTRL1.....	1376
8-254. Register Call Summary for Register BTE_CTRL1 .....	1376
8-255. BTE_CONTEXT_CTRL_i.....	1376

8-256. Register Call Summary for Register BTE_CONTEXT_CTRL_i .....	1378
8-257. BTE_CONTEXT_BASE_i.....	1378
8-258. Register Call Summary for Register BTE_CONTEXT_BASE_i .....	1378
8-259. BTE_CONTEXT_START_i.....	1378
8-260. Register Call Summary for Register BTE_CONTEXT_START_i .....	1378
8-261. BTE_CONTEXT_END_i .....	1379
8-262. Register Call Summary for Register BTE_CONTEXT_END_i.....	1379
8-263. ISS CBUFF Interrupt Management .....	1382
8-264. ISS CBUFF-Generated Events .....	1382
8-265. ISS CBUFF Functional Modes.....	1384
8-266. ISS CBUFF Internal Variables .....	1392
8-267. ISS CBUFF Internal State After Reset.....	1392
8-268. ISS CBUFF Address identification .....	1392
8-269. ISS CBUFF Address Translation .....	1393
8-270. ISS CBUFF Window-Level Increment .....	1394
8-271. ISS CBUFF Window-Level Comparison .....	1394
8-272. ISS CBUFF Setup Register .....	1398
8-273. ISS CBUFF Instance Summary .....	1399
8-274. ISS CBUFF Registers Mapping Summary .....	1399
8-275. CBUFF_HL_REVISION.....	1400
8-276. Register Call Summary for Register CBUFF_HL_REVISION .....	1400
8-277. CBUFF_HL_HWINFO.....	1400
8-278. Register Call Summary for Register CBUFF_HL_HWINFO .....	1400
8-279. CBUFF_HL_SYSCONFIG.....	1401
8-280. Register Call Summary for Register CBUFF_HL_SYSCONFIG .....	1401
8-281. CBUFF_HL_IRQSTATUS_RAW .....	1402
8-282. Register Call Summary for Register CBUFF_HL_IRQSTATUS_RAW .....	1405
8-283. CBUFF_HL_IRQSTATUS .....	1405
8-284. Register Call Summary for Register CBUFF_HL_IRQSTATUS.....	1408
8-285. CBUFF_HL_IRQENABLE_SET .....	1408
8-286. Register Call Summary for Register CBUFF_HL_IRQENABLE_SET.....	1412
8-287. CBUFF_HL_IRQENABLE_CLR .....	1412
8-288. Register Call Summary for Register CBUFF_HL_IRQENABLE_CLR.....	1415
8-289. CBUFF_FRAG_ADDR_j.....	1415
8-290. Register Call Summary for Register CBUFF_FRAG_ADDR_j .....	1415
8-291. CBUFF_CTX_CTRL_i .....	1416
8-292. Register Call Summary for Register CBUFF_CTX_CTRL_i.....	1417
8-293. CBUFF_CTX_START_i.....	1417
8-294. Register Call Summary for Register CBUFF_CTX_START_i .....	1417
8-295. CBUFF_CTX_END_i.....	1417
8-296. Register Call Summary for Register CBUFF_CTX_END_j .....	1418
8-297. CBUFF_CTX_WINDOWSIZE_i.....	1418
8-298. Register Call Summary for Register CBUFF_CTX_WINDOWSIZE_i .....	1418
8-299. CBUFF_CTX_THRESHOLD_F_i.....	1418
8-300. Register Call Summary for Register CBUFF_CTX_THRESHOLD_F_j .....	1418
8-301. CBUFF_CTX_THRESHOLD_S_i.....	1419
8-302. Register Call Summary for Register CBUFF_CTX_THRESHOLD_S_i .....	1419
8-303. CBUFF_CTX_STATUS_i.....	1419
8-304. Register Call Summary for Register CBUFF_CTX_STATUS_i .....	1419

8-305. CBUFF_CTX_PHY_i .....	1420
8-306. Register Call Summary for Register CBUFF_CTX_PHY_i .....	1420
8-307. ISS ISP Interrupt Tree Table .....	1423
8-308. ISS ISP VP Format Mapping .....	1431
8-309. ISS ISP VP GCK_MMR to PCLK Clock Resynchronization .....	1434
8-310. ISS ISP IPIPEIF IPIPEIF_CFG1[15:14] INPSRC1 and IPIPEIF_CFG1[3:2] INPSRC2 Possible Combinations .....	1441
8-311. ISS ISP IPIPEIF DCPM Block Possible Configuration .....	1451
8-312. ISS ISP IPIPEIF DFS Modes Supported .....	1454
8-313. ISS ISP IPIPEIF Averaging Filter Conditions for YUV4:2:2 Data .....	1454
8-314. ISS ISP IPIPE Input and Output Selections .....	1461
8-315. ISS ISP IPIPE Defect Information Packing .....	1463
8-316. ISS ISP IPIPE Correction Method Description .....	1463
8-317. ISS ISP IPIPE Edge-Enhancer LUT Mapping .....	1469
8-318. ISS ISP RSZ VP Supported Formats .....	1477
8-319. ISS ISP RSZ Data Flow vs. Input Data Format Constraints .....	1479
8-320. ISS ISP RSZ Module Modes: Register Settings .....	1480
8-321. ISS ISP RSZ Module Input Control: Register Settings .....	1481
8-322. ISS ISP RSZ-A/RSZ-B Output Format Selection .....	1481
8-323. ISS ISP RSZ Output Interface: Data Formats .....	1491
8-324. ISS ISP H3A Poxel Register Field Descriptions .....	1500
8-325. ISS ISP H3A AE/AWB Window Register Field Descriptions .....	1503
8-326. ISS ISP H3A AE/AWB Window with Additional Black Row Register Field Descriptions .....	1504
8-327. ISS ISP H3A AF Packet Format With Vertical AF Disabled .....	1505
8-328. ISS ISP H3A AF Packet Format With Vertical AF Enabled .....	1505
8-329. ISS ISP H3A AE/AWB Packet Format for Sum of Square Mode .....	1507
8-330. ISS ISP H3A AE/AWB Packet Format for Minimum-Maximum Mode .....	1509
8-331. ISS ISP H3A AE/AWB Packet Format for Sum-Only Mode .....	1511
8-332. ISS ISP ISIF Input Interface Signals .....	1516
8-333. ISS ISP ISIF Data Input Formats .....	1516
8-334. ISS ISP ISIF Raw Data Connection: Selects MSB Position of Input Data .....	1517
8-335. ISS ISP ISIF Linearization LUT .....	1518
8-336. ISS ISP ISIF LUT Memory Region .....	1518
8-337. ISS ISP ISIF Input Data Formatter Area Setting Registers .....	1521
8-338. ISS ISP ISIF Output Data Formatter Area Setting Registers .....	1521
8-339. ISS ISP ISIF Example of Combining Three Input Lines Into a Single Line: Register Setting Example .....	1525
8-340. ISS ISP ISIF Vertical Line Defect Table in Memory .....	1536
8-341. ISS ISP ISIF Supported On-the-Fly LSC Configurations .....	1541
8-342. ISS ISP ISIF RAW Data Shifting .....	1549
8-343. ISS ISP ISIF SDRAM Data Format .....	1549
8-344. ISS ISP ISIF SDRAM Data Format for 12-bit Packed .....	1549
8-345. ISS ISP ISIF Memory Output Format for YUV Data .....	1551
8-346. ISS ISP ISIF Module: Write Port Bandwidth .....	1551
8-347. ISS ISP ISIF Read Port Bandwidth .....	1552
8-348. ISS ISP BL cpriority to MFlag With ISP5_CTRL[21] MFLAG = 1 .....	1558
8-349. ISS ISP BL MFlag Write Low- and High-Level Priority Thresholds .....	1558
8-350. ISS ISP BL MFlag Read Low- and High-Level Priority Thresholds .....	1559
8-351. ISS ISP Memory Mapping .....	1559
8-352. ISS ISP PG Settings .....	1561

8-353. ISS ISP ISIF Required Configuration Parameters .....	1563
8-354. ISS ISP ISIF Maximum Line Width Versus M Value.....	1570
8-355. ISS ISP IPIPEIF Required Configuration Parameters .....	1570
8-356. ISS Global Initialization .....	1573
8-357. ISS ISP IPIPE LUT Defect Pixel Correction (LUT DPC) .....	1574
8-358. ISS ISP IPIPE White Balance Initialization.....	1574
8-359. ISS ISP IPIPE RGB2RGB Blending Module Initialization .....	1574
8-360. ISS ISP IPIPE Gamma Correction Module Initialization.....	1575
8-361. ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization .....	1575
8-362. ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization .....	1575
8-363. ISS ISP IPIPE 4:2:2 Conversion Module Initialization .....	1575
8-364. ISS ISP IPIPE 2D Edge Enhancer Initialization .....	1576
8-365. ISS ISP IPIPE Histogram Initialization.....	1576
8-366. ISS ISP IPIPE Boxcar Initialization.....	1577
8-367. ISS ISP IPIPE Processing Path: Case 1 Configuration .....	1577
8-368. ISS ISP IPIPE Processing Path: Case 2 Configuration .....	1578
8-369. ISS ISP IPIPE Processing Path: Case 3 Configuration .....	1578
8-370. ISS ISP IPIPE Processing Path: Case 4 Configuration .....	1578
8-371. ISS ISP RSZ Surrounding Modules Global Initialization .....	1579
8-372. ISS ISP RSZ Initial Register Setup .....	1579
8-373. ISS ISP RSZ Reset Behavior .....	1580
8-374. ISS ISP RSZ Global Image Processing Settings.....	1581
8-375. Register Call Summary for ISS ISP RSZ Global Image Processing Settings – Subsequence 1 .....	1584
8-376. Register Call Summary for ISS ISP RSZ Global Image Processing Settings – Subsequence 2 .....	1584
8-377. ISS ISP RSZ Engines Interframe Image Processing Settings.....	1585
8-378. Register Call Summary for ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1 .....	1587
8-379. Register Call Summary for ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2 .....	1587
8-380. ISS ISP RSZ Nonshadowed Registers .....	1588
8-381. ISS ISP H3A AF Engine Required Configuration Parameters.....	1589
8-382. ISS ISP H3A AEW Engine Required Configuration Parameters .....	1590
8-383. ISS ISP H3A Constraints When PCLK < ISP_FCLK/2 and Vertical Focus Enabled .....	1592
8-384. ISS ISP H3A Constraints When PCLK < ISP_FCLK/2 and Vertical Focus Disabled .....	1592
8-385. ISS ISP H3A Constraints When PCLK >= ISP_FCLK/2 and Vertical Focus Enabled .....	1592
8-386. ISS ISP H3A Constraints When PCLK >= ISP_FCLK/2 and Vertical Focus Disabled .....	1592
8-387. ISS ISP BL Settings.....	1593
8-388. ISS ISP Instance Summary .....	1594
8-389. ISS ISP5 SYS1 Registers Mapping Summary.....	1594
8-390. ISP5_REVISION.....	1595
8-391. Register Call Summary for Register ISP5_REVISION .....	1595
8-392. ISP5_HWINFO1 .....	1595
8-393. Register Call Summary for Register ISP5_HWINFO1.....	1595
8-394. ISP5_HWINFO2 .....	1596
8-395. Register Call Summary for Register ISP5_HWINFO2.....	1596
8-396. ISP5_SYSCONFIG.....	1596
8-397. Register Call Summary for Register ISP5_SYSCONFIG .....	1597
8-398. ISP5_IRQSTATUS_RAW_i .....	1597
8-399. Register Call Summary for Register ISP5_IRQSTATUS_RAW_i.....	1601

8-400. ISP5_IRQSTATUS_i .....	1602
8-401. Register Call Summary for Register ISP5_IRQSTATUS_i.....	1606
8-402. ISP5_IRQENABLE_SET_i .....	1607
8-403. Register Call Summary for Register ISP5_IRQENABLE_SET_i.....	1611
8-404. ISP5_IRQENABLE_CLR_i .....	1612
8-405. Register Call Summary for Register ISP5_IRQENABLE_CLR_i.....	1616
8-406. ISP5_DMAENABLE_SET .....	1616
8-407. Register Call Summary for Register ISP5_DMAENABLE_SET .....	1618
8-408. ISP5_DMAENABLE_CLR .....	1618
8-409. Register Call Summary for Register ISP5_DMAENABLE_CLR.....	1619
8-410. ISP5_CTRL.....	1619
8-411. Register Call Summary for Register ISP5_CTRL .....	1623
8-412. ISP5_PG.....	1623
8-413. Register Call Summary for Register ISP5_PG .....	1624
8-414. ISP5_PG_PULSE_CTRL .....	1624
8-415. Register Call Summary for Register ISP5_PG_PULSE_CTRL .....	1624
8-416. ISP5_PG_FRAME_SIZE .....	1624
8-417. Register Call Summary for Register ISP5_PG_FRAME_SIZE .....	1624
8-418. ISP5_MPSR .....	1625
8-419. Register Call Summary for Register ISP5_MPSR .....	1628
8-420. ISP5_BL_MTC_1 .....	1628
8-421. Register Call Summary for Register ISP5_BL_MTC_1 .....	1628
8-422. ISP5_BL_MTC_2 .....	1628
8-423. Register Call Summary for Register ISP5_BL_MTC_2 .....	1628
8-424. ISP5_BL_VBUSM .....	1629
8-425. Register Call Summary for Register ISP5_BL_VBUSM.....	1629
8-426. ISS ISP5 SYS2 Registers Mapping Summary.....	1630
8-427. ISP5_KEY_EN1 .....	1630
8-428. Register Call Summary for Register ISP5_KEY_EN1 .....	1630
8-429. ISP5_KEY_EN2 .....	1631
8-430. Register Call Summary for Register ISP5_KEY_EN2 .....	1631
8-431. ISP5_KEY_EN3 .....	1631
8-432. Register Call Summary for Register ISP5_KEY_EN3 .....	1631
8-433. ISP5_KEY_EN4 .....	1632
8-434. Register Call Summary for Register ISP5_KEY_EN4 .....	1632
8-435. ISP5_KEY_EN5 .....	1632
8-436. Register Call Summary for Register ISP5_KEY_EN5 .....	1632
8-437. ISP5_KEY_EN6 .....	1633
8-438. Register Call Summary for Register ISP5_KEY_EN6 .....	1633
8-439. ISP5_IRQSTATUS_RAW2_i.....	1633
8-440. Register Call Summary for Register ISP5_IRQSTATUS_RAW2_i .....	1634
8-441. ISP5_IRQSTATUS2_i.....	1634
8-442. Register Call Summary for Register ISP5_IRQSTATUS2_i .....	1635
8-443. ISP5_IRQENABLE_SET2_i.....	1636
8-444. Register Call Summary for Register ISP5_IRQENABLE_SET2_i .....	1636
8-445. ISP5_IRQENABLE_CLR2_i.....	1637
8-446. Register Call Summary for Register ISP5_IRQENABLE_CLR2_i .....	1638
8-447. ISS RESIZER Registers Mapping Summary .....	1638
8-448. RSZ_REVISION .....	1640

8-449. Register Call Summary for Register RSZ_REVISION.....	1640
8-450. RSZ_SYSCONFIG .....	1640
8-451. Register Call Summary for Register RSZ_SYSCONFIG.....	1641
8-452. RSZ_IN_FIFO_CTRL .....	1642
8-453. Register Call Summary for Register RSZ_IN_FIFO_CTRL.....	1642
8-454. RSZ_GNC .....	1642
8-455. Register Call Summary for Register RSZ_GNC.....	1643
8-456. RSZ_FRACDIV .....	1643
8-457. Register Call Summary for Register RSZ_FRACDIV.....	1643
8-458. RSZ_SRC_EN .....	1643
8-459. Register Call Summary for Register RSZ_SRC_EN.....	1644
8-460. RSZ_SRC_MODE.....	1644
8-461. Register Call Summary for Register RSZ_SRC_MODE .....	1645
8-462. RSZ_SRC_FMT0.....	1645
8-463. Register Call Summary for Register RSZ_SRC_FMT0 .....	1645
8-464. RSZ_SRC_FMT1 .....	1646
8-465. Register Call Summary for Register RSZ_SRC_FMT1 .....	1646
8-466. RSZ_SRC_VPS .....	1646
8-467. Register Call Summary for Register RSZ_SRC_VPS .....	1647
8-468. RSZ_SRC_VSZ.....	1647
8-469. Register Call Summary for Register RSZ_SRC_VSZ .....	1647
8-470. RSZ_SRC_HPS .....	1647
8-471. Register Call Summary for Register RSZ_SRC_HPS.....	1648
8-472. RSZ_SRC_HSZ .....	1648
8-473. Register Call Summary for Register RSZ_SRC_HSZ .....	1648
8-474. RSZ_DMA_RZA .....	1649
8-475. Register Call Summary for Register RSZ_DMA_RZA.....	1649
8-476. RSZ_DMA_RZB .....	1649
8-477. Register Call Summary for Register RSZ_DMA_RZB.....	1649
8-478. RSZ_DMA_STA .....	1650
8-479. Register Call Summary for Register RSZ_DMA_STA.....	1650
8-480. RSZ_GCK_MMR .....	1650
8-481. Register Call Summary for Register RSZ_GCK_MMR.....	1650
8-482. RSZ_GCK_SDR .....	1651
8-483. Register Call Summary for Register RSZ_GCK_SDR.....	1651
8-484. RSZ_IRQ_RZA .....	1651
8-485. Register Call Summary for Register RSZ_IRQ_RZA .....	1652
8-486. RSZ_IRQ_RZB .....	1652
8-487. Register Call Summary for Register RSZ_IRQ_RZB.....	1652
8-488. RSZ_YUV_Y_MIN.....	1652
8-489. Register Call Summary for Register RSZ_YUV_Y_MIN .....	1653
8-490. RSZ_YUV_Y_MAX.....	1653
8-491. Register Call Summary for Register RSZ_YUV_Y_MAX .....	1653
8-492. RSZ_YUV_C_MIN.....	1653
8-493. Register Call Summary for Register RSZ_YUV_C_MIN .....	1653
8-494. RSZ_YUV_C_MAX.....	1654
8-495. Register Call Summary for Register RSZ_YUV_C_MAX .....	1654
8-496. RSZ_YUV_PHS .....	1654
8-497. Register Call Summary for Register RSZ_YUV_PHS .....	1654



8-498. RSZ_SEQ .....	1655
8-499. Register Call Summary for Register RSZ_SEQ .....	1655
8-500. RZA_EN .....	1656
8-501. Register Call Summary for Register RZA_EN .....	1656
8-502. RZA_MODE .....	1656
8-503. Register Call Summary for Register RZA_MODE .....	1656
8-504. RZA_420.....	1657
8-505. Register Call Summary for Register RZA_420 .....	1657
8-506. RZA_I_VPS .....	1657
8-507. Register Call Summary for Register RZA_I_VPS .....	1658
8-508. RZA_I_HPS .....	1658
8-509. Register Call Summary for Register RZA_I_HPS.....	1658
8-510. RZA_O_VSZ .....	1658
8-511. Register Call Summary for Register RZA_O_VSZ.....	1658
8-512. RZA_O_HSZ .....	1659
8-513. Register Call Summary for Register RZA_O_HSZ.....	1659
8-514. RZA_V_PHS_Y .....	1659
8-515. Register Call Summary for Register RZA_V_PHS_Y .....	1660
8-516. RZA_V_PHS_C.....	1660
8-517. Register Call Summary for Register RZA_V_PHS_C .....	1660
8-518. RZA_V_DIF .....	1660
8-519. Register Call Summary for Register RZA_V_DIF .....	1661
8-520. RZA_V_TYP.....	1661
8-521. Register Call Summary for Register RZA_V_TYP .....	1661
8-522. RZA_V_LPF.....	1661
8-523. Register Call Summary for Register RZA_V_LPF .....	1661
8-524. RZA_H_PHS .....	1662
8-525. Register Call Summary for Register RZA_H_PHS.....	1662
8-526. RZA_H_PHS_ADJ .....	1662
8-527. Register Call Summary for Register RZA_H_PHS_ADJ .....	1662
8-528. RZA_H_DIF .....	1663
8-529. Register Call Summary for Register RZA_H_DIF.....	1663
8-530. RZA_H_TYP .....	1663
8-531. Register Call Summary for Register RZA_H_TYP .....	1663
8-532. RZA_H_LPF.....	1664
8-533. Register Call Summary for Register RZA_H_LPF .....	1664
8-534. RZA_DWN_EN .....	1664
8-535. Register Call Summary for Register RZA_DWN_EN .....	1664
8-536. RZA_DWN_AV.....	1665
8-537. Register Call Summary for Register RZA_DWN_AV .....	1665
8-538. RZA_RGB_EN .....	1665
8-539. Register Call Summary for Register RZA_RGB_EN .....	1666
8-540. RZA_RGB_TYP .....	1666
8-541. Register Call Summary for Register RZA_RGB_TYP .....	1666
8-542. RZA_RGB_BLD .....	1667
8-543. Register Call Summary for Register RZA_RGB_BLD .....	1667
8-544. RZA_SDR_Y_BAD_H.....	1667
8-545. Register Call Summary for Register RZA_SDR_Y_BAD_H .....	1667
8-546. RZA_SDR_Y_BAD_L .....	1668

8-547. Register Call Summary for Register RZA_SDR_Y_BAD_L.....	1668
8-548. RZA_SDR_Y_SAD_H.....	1669
8-549. Register Call Summary for Register RZA_SDR_Y_SAD_H.....	1669
8-550. RZA_SDR_Y_SAD_L.....	1669
8-551. Register Call Summary for Register RZA_SDR_Y_SAD_L.....	1670
8-552. RZA_SDR_Y_OFT.....	1670
8-553. Register Call Summary for Register RZA_SDR_Y_OFT.....	1671
8-554. RZA_SDR_Y_PTR_S.....	1671
8-555. Register Call Summary for Register RZA_SDR_Y_PTR_S.....	1671
8-556. RZA_SDR_Y_PTR_E.....	1671
8-557. Register Call Summary for Register RZA_SDR_Y_PTR_E.....	1672
8-558. RZA_SDR_C_BAD_H.....	1672
8-559. Register Call Summary for Register RZA_SDR_C_BAD_H.....	1672
8-560. RZA_SDR_C_BAD_L.....	1672
8-561. Register Call Summary for Register RZA_SDR_C_BAD_L.....	1673
8-562. RZA_SDR_C_SAD_H.....	1673
8-563. Register Call Summary for Register RZA_SDR_C_SAD_H.....	1673
8-564. RZA_SDR_C_SAD_L.....	1673
8-565. Register Call Summary for Register RZA_SDR_C_SAD_L.....	1674
8-566. RZA_SDR_C_OFT.....	1674
8-567. Register Call Summary for Register RZA_SDR_C_OFT.....	1674
8-568. RZA_SDR_C_PTR_S.....	1674
8-569. Register Call Summary for Register RZA_SDR_C_PTR_S.....	1675
8-570. RZA_SDR_C_PTR_E.....	1675
8-571. Register Call Summary for Register RZA_SDR_C_PTR_E.....	1675
8-572. RZB_EN.....	1675
8-573. Register Call Summary for Register RZB_EN.....	1676
8-574. RZB_MODE.....	1676
8-575. Register Call Summary for Register RZB_MODE.....	1676
8-576. RZB_420.....	1676
8-577. Register Call Summary for Register RZB_420.....	1677
8-578. RZB_I_VPS.....	1677
8-579. Register Call Summary for Register RZB_I_VPS.....	1677
8-580. RZB_I_HPS.....	1678
8-581. Register Call Summary for Register RZB_I_HPS.....	1678
8-582. RZB_O_VSZ.....	1678
8-583. Register Call Summary for Register RZB_O_VSZ.....	1678
8-584. RZB_O_HSZ.....	1679
8-585. Register Call Summary for Register RZB_O_HSZ.....	1679
8-586. RZB_V_PHS_Y.....	1679
8-587. Register Call Summary for Register RZB_V_PHS_Y.....	1679
8-588. RZB_V_PHS_C.....	1680
8-589. Register Call Summary for Register RZB_V_PHS_C.....	1680
8-590. RZB_V_DIF.....	1680
8-591. Register Call Summary for Register RZB_V_DIF.....	1680
8-592. RZB_V_TYP.....	1681
8-593. Register Call Summary for Register RZB_V_TYP.....	1681
8-594. RZB_V_LPF.....	1681
8-595. Register Call Summary for Register RZB_V_LPF.....	1681

8-596. RZB_H_PHS .....	1682
8-597. Register Call Summary for Register RZB_H_PHS.....	1682
8-598. RZB_H_PHS_ADJ .....	1682
8-599. Register Call Summary for Register RZB_H_PHS_ADJ .....	1682
8-600. RZB_H_DIF .....	1683
8-601. Register Call Summary for Register RZB_H_DIF.....	1683
8-602. RZB_H_TYP .....	1683
8-603. Register Call Summary for Register RZB_H_TYP .....	1683
8-604. RZB_H_LPF.....	1684
8-605. Register Call Summary for Register RZB_H_LPF .....	1684
8-606. RZB_DWN_EN .....	1684
8-607. Register Call Summary for Register RZB_DWN_EN.....	1684
8-608. RZB_DWN_AV.....	1685
8-609. Register Call Summary for Register RZB_DWN_AV .....	1685
8-610. RZB_RGB_EN .....	1685
8-611. Register Call Summary for Register RZB_RGB_EN .....	1686
8-612. RZB_RGB_TYP .....	1686
8-613. Register Call Summary for Register RZB_RGB_TYP .....	1686
8-614. RZB_RGB_BLD .....	1687
8-615. Register Call Summary for Register RZB_RGB_BLD .....	1687
8-616. RZB_SDR_Y_BAD_H.....	1687
8-617. Register Call Summary for Register RZB_SDR_Y_BAD_H.....	1687
8-618. RZB_SDR_Y_BAD_L .....	1688
8-619. Register Call Summary for Register RZB_SDR_Y_BAD_L.....	1688
8-620. RZB_SDR_Y_SAD_H.....	1689
8-621. Register Call Summary for Register RZB_SDR_Y_SAD_H.....	1689
8-622. RZB_SDR_Y_SAD_L .....	1689
8-623. Register Call Summary for Register RZB_SDR_Y_SAD_L.....	1690
8-624. RZB_SDR_Y_OFT .....	1690
8-625. Register Call Summary for Register RZB_SDR_Y_OFT.....	1691
8-626. RZB_SDR_Y_PTR_S.....	1691
8-627. Register Call Summary for Register RZB_SDR_Y_PTR_S .....	1691
8-628. RZB_SDR_Y_PTR_E .....	1691
8-629. Register Call Summary for Register RZB_SDR_Y_PTR_E .....	1692
8-630. RZB_SDR_C_BAD_H .....	1692
8-631. Register Call Summary for Register RZB_SDR_C_BAD_H.....	1692
8-632. RZB_SDR_C_BAD_L .....	1692
8-633. Register Call Summary for Register RZB_SDR_C_BAD_L .....	1693
8-634. RZB_SDR_C_SAD_H .....	1693
8-635. Register Call Summary for Register RZB_SDR_C_SAD_H.....	1693
8-636. RZB_SDR_C_SAD_L .....	1693
8-637. Register Call Summary for Register RZB_SDR_C_SAD_L .....	1694
8-638. RZB_SDR_C_OFT .....	1694
8-639. Register Call Summary for Register RZB_SDR_C_OFT.....	1694
8-640. RZB_SDR_C_PTR_S.....	1694
8-641. Register Call Summary for Register RZB_SDR_C_PTR_S .....	1695
8-642. RZB_SDR_C_PTR_E.....	1695
8-643. Register Call Summary for Register RZB_SDR_C_PTR_E .....	1695
8-644. ISS IPIPE Registers Mapping Summary .....	1695

8-645. IPIPE_SRC_EN.....	1699
8-646. Register Call Summary for Register IPIPE_SRC_EN .....	1700
8-647. IPIPE_SRC_MODE .....	1700
8-648. Register Call Summary for Register IPIPE_SRC_MODE.....	1700
8-649. IPIPE_SRC_FMT .....	1701
8-650. Register Call Summary for Register IPIPE_SRC_FMT .....	1701
8-651. IPIPE_SRC_COL .....	1701
8-652. Register Call Summary for Register IPIPE_SRC_COL .....	1702
8-653. IPIPE_SRC_VPS .....	1702
8-654. Register Call Summary for Register IPIPE_SRC_VPS .....	1703
8-655. IPIPE_SRC_VSZ .....	1703
8-656. Register Call Summary for Register IPIPE_SRC_VSZ.....	1703
8-657. IPIPE_SRC_HPS.....	1703
8-658. Register Call Summary for Register IPIPE_SRC_HPS .....	1704
8-659. IPIPE_SRC_HSZ .....	1704
8-660. Register Call Summary for Register IPIPE_SRC_HSZ .....	1704
8-661. IPIPE_SEL_SBU .....	1704
8-662. Register Call Summary for Register IPIPE_SEL_SBU .....	1705
8-663. IPIPE_SRC_STA .....	1705
8-664. Register Call Summary for Register IPIPE_SRC_STA.....	1705
8-665. IPIPE_GCK_MMR.....	1705
8-666. Register Call Summary for Register IPIPE_GCK_MMR .....	1706
8-667. IPIPE_GCK_PIX.....	1706
8-668. Register Call Summary for Register IPIPE_GCK_PIX .....	1706
8-669. IPIPE_DPC_LUT_EN .....	1707
8-670. Register Call Summary for Register IPIPE_DPC_LUT_EN.....	1707
8-671. IPIPE_DPC_LUT_SEL.....	1707
8-672. Register Call Summary for Register IPIPE_DPC_LUT_SEL .....	1707
8-673. IPIPE_DPC_LUT_ADR .....	1708
8-674. Register Call Summary for Register IPIPE_DPC_LUT_ADR.....	1708
8-675. IPIPE_DPC_LUT_SIZ.....	1708
8-676. Register Call Summary for Register IPIPE_DPC_LUT_SIZ .....	1708
8-677. IPIPE_LSC_VOFT.....	1709
8-678. Register Call Summary for Register IPIPE_LSC_VOFT .....	1709
8-679. IPIPE_LSC_VA2.....	1709
8-680. Register Call Summary for Register IPIPE_LSC_VA2 .....	1709
8-681. IPIPE_LSC_VA1.....	1710
8-682. Register Call Summary for Register IPIPE_LSC_VA1 .....	1710
8-683. IPIPE_LSC_VS .....	1710
8-684. Register Call Summary for Register IPIPE_LSC_VS.....	1710
8-685. IPIPE_LSC_HOFT .....	1711
8-686. Register Call Summary for Register IPIPE_LSC_HOFT .....	1711
8-687. IPIPE_LSC_HA2.....	1711
8-688. Register Call Summary for Register IPIPE_LSC_HA2 .....	1711
8-689. IPIPE_LSC_HA1.....	1712
8-690. Register Call Summary for Register IPIPE_LSC_HA1 .....	1712
8-691. IPIPE_LSC_HS .....	1712
8-692. Register Call Summary for Register IPIPE_LSC_HS.....	1712
8-693. IPIPE_LSC_GAN_R.....	1713

8-694. Register Call Summary for Register IPIPE_LSC_GAN_R .....	1713
8-695. IPIPE_LSC_GAN_GR .....	1713
8-696. Register Call Summary for Register IPIPE_LSC_GAN_GR .....	1713
8-697. IPIPE_LSC_GAN_GB.....	1713
8-698. Register Call Summary for Register IPIPE_LSC_GAN_GB .....	1714
8-699. IPIPE_LSC_GAN_B.....	1714
8-700. Register Call Summary for Register IPIPE_LSC_GAN_B .....	1714
8-701. IPIPE_LSC_OFT_R .....	1714
8-702. Register Call Summary for Register IPIPE_LSC_OFT_R .....	1714
8-703. IPIPE_LSC_OFT_GR .....	1715
8-704. Register Call Summary for Register IPIPE_LSC_OFT_GR .....	1715
8-705. IPIPE_LSC_OFT_GB .....	1715
8-706. Register Call Summary for Register IPIPE_LSC_OFT_GB.....	1715
8-707. IPIPE_LSC_OFT_B .....	1715
8-708. Register Call Summary for Register IPIPE_LSC_OFT_B.....	1716
8-709. IPIPE_LSC_SHF .....	1716
8-710. Register Call Summary for Register IPIPE_LSC_SHF .....	1716
8-711. IPIPE_LSC_MAX .....	1716
8-712. Register Call Summary for Register IPIPE_LSC_MAX .....	1716
8-713. IPIPE_WB2_OFT_R .....	1717
8-714. Register Call Summary for Register IPIPE_WB2_OFT_R.....	1717
8-715. IPIPE_WB2_OFT_GR .....	1717
8-716. Register Call Summary for Register IPIPE_WB2_OFT_GR.....	1717
8-717. IPIPE_WB2_OFT_GB .....	1718
8-718. Register Call Summary for Register IPIPE_WB2_OFT_GB .....	1718
8-719. IPIPE_WB2_OFT_B.....	1718
8-720. Register Call Summary for Register IPIPE_WB2_OFT_B .....	1718
8-721. IPIPE_WB2_WGN_R .....	1719
8-722. Register Call Summary for Register IPIPE_WB2_WGN_R.....	1719
8-723. IPIPE_WB2_WGN_GR .....	1719
8-724. Register Call Summary for Register IPIPE_WB2_WGN_GR.....	1719
8-725. IPIPE_WB2_WGN_GB .....	1720
8-726. Register Call Summary for Register IPIPE_WB2_WGN_GB.....	1720
8-727. IPIPE_WB2_WGN_B .....	1720
8-728. Register Call Summary for Register IPIPE_WB2_WGN_B.....	1720
8-729. IPIPE_RGB1_MUL_RR.....	1721
8-730. Register Call Summary for Register IPIPE_RGB1_MUL_RR .....	1721
8-731. IPIPE_RGB1_MUL_GR .....	1721
8-732. Register Call Summary for Register IPIPE_RGB1_MUL_GR .....	1721
8-733. IPIPE_RGB1_MUL_BR.....	1722
8-734. Register Call Summary for Register IPIPE_RGB1_MUL_BR .....	1722
8-735. IPIPE_RGB1_MUL_RG .....	1722
8-736. Register Call Summary for Register IPIPE_RGB1_MUL_RG .....	1722
8-737. IPIPE_RGB1_MUL_GG .....	1722
8-738. Register Call Summary for Register IPIPE_RGB1_MUL_GG .....	1723
8-739. IPIPE_RGB1_MUL_BG.....	1723
8-740. Register Call Summary for Register IPIPE_RGB1_MUL_BG .....	1723
8-741. IPIPE_RGB1_MUL_RB.....	1723
8-742. Register Call Summary for Register IPIPE_RGB1_MUL_RB .....	1723

8-743. IPIPE_RGB1_MUL_GB.....	1724
8-744. Register Call Summary for Register IPIPE_RGB1_MUL_GB .....	1724
8-745. IPIPE_RGB1_MUL_BB .....	1724
8-746. Register Call Summary for Register IPIPE_RGB1_MUL_BB .....	1724
8-747. IPIPE_RGB1_OFT_OR.....	1725
8-748. Register Call Summary for Register IPIPE_RGB1_OFT_OR .....	1725
8-749. IPIPE_RGB1_OFT_OG.....	1725
8-750. Register Call Summary for Register IPIPE_RGB1_OFT_OG .....	1725
8-751. IPIPE_RGB1_OFT_OB .....	1726
8-752. Register Call Summary for Register IPIPE_RGB1_OFT_OB .....	1726
8-753. IPIPE_GMM_CFG.....	1726
8-754. Register Call Summary for Register IPIPE_GMM_CFG .....	1727
8-755. IPIPE_RGB2_MUL_RR.....	1727
8-756. Register Call Summary for Register IPIPE_RGB2_MUL_RR .....	1727
8-757. IPIPE_RGB2_MUL_GR .....	1728
8-758. Register Call Summary for Register IPIPE_RGB2_MUL_GR .....	1728
8-759. IPIPE_RGB2_MUL_BR.....	1728
8-760. Register Call Summary for Register IPIPE_RGB2_MUL_BR .....	1728
8-761. IPIPE_RGB2_MUL_RG .....	1728
8-762. Register Call Summary for Register IPIPE_RGB2_MUL_RG .....	1729
8-763. IPIPE_RGB2_MUL_GG .....	1729
8-764. Register Call Summary for Register IPIPE_RGB2_MUL_GG .....	1729
8-765. IPIPE_RGB2_MUL_BG.....	1729
8-766. Register Call Summary for Register IPIPE_RGB2_MUL_BG .....	1729
8-767. IPIPE_RGB2_MUL_RB.....	1730
8-768. Register Call Summary for Register IPIPE_RGB2_MUL_RB .....	1730
8-769. IPIPE_RGB2_MUL_GB.....	1730
8-770. Register Call Summary for Register IPIPE_RGB2_MUL_GB .....	1730
8-771. IPIPE_RGB2_MUL_BB .....	1730
8-772. Register Call Summary for Register IPIPE_RGB2_MUL_BB .....	1731
8-773. IPIPE_RGB2_OFT_OR.....	1731
8-774. Register Call Summary for Register IPIPE_RGB2_OFT_OR .....	1731
8-775. IPIPE_RGB2_OFT_OG.....	1731
8-776. Register Call Summary for Register IPIPE_RGB2_OFT_OG .....	1732
8-777. IPIPE_RGB2_OFT_OB .....	1732
8-778. Register Call Summary for Register IPIPE_RGB2_OFT_OB .....	1732
8-779. IPIPE_YUV_ADJ .....	1732
8-780. Register Call Summary for Register IPIPE_YUV_ADJ .....	1732
8-781. IPIPE_YUV_MUL_RY.....	1733
8-782. Register Call Summary for Register IPIPE_YUV_MUL_RY .....	1733
8-783. IPIPE_YUV_MUL_GY .....	1733
8-784. Register Call Summary for Register IPIPE_YUV_MUL_GY .....	1733
8-785. IPIPE_YUV_MUL_BY.....	1734
8-786. Register Call Summary for Register IPIPE_YUV_MUL_BY .....	1734
8-787. IPIPE_YUV_MUL_RCB.....	1734
8-788. Register Call Summary for Register IPIPE_YUV_MUL_RCB .....	1734
8-789. IPIPE_YUV_MUL_GCB .....	1735
8-790. Register Call Summary for Register IPIPE_YUV_MUL_GCB .....	1735
8-791. IPIPE_YUV_MUL_BCB.....	1735

8-792. Register Call Summary for Register IPIPE_YUV_MUL_BCB .....	1735
8-793. IPIPE_YUV_MUL_RCR .....	1735
8-794. Register Call Summary for Register IPIPE_YUV_MUL_RCR .....	1736
8-795. IPIPE_YUV_MUL_GCR .....	1736
8-796. Register Call Summary for Register IPIPE_YUV_MUL_GCR .....	1736
8-797. IPIPE_YUV_MUL_BCR .....	1736
8-798. Register Call Summary for Register IPIPE_YUV_MUL_BCR .....	1736
8-799. IPIPE_YUV_OFT_Y .....	1737
8-800. Register Call Summary for Register IPIPE_YUV_OFT_Y .....	1737
8-801. IPIPE_YUV_OFT_CB .....	1737
8-802. Register Call Summary for Register IPIPE_YUV_OFT_CB .....	1737
8-803. IPIPE_YUV_OFT_CR .....	1738
8-804. Register Call Summary for Register IPIPE_YUV_OFT_CR .....	1738
8-805. IPIPE_YUV_PHS .....	1738
8-806. Register Call Summary for Register IPIPE_YUV_PHS .....	1739
8-807. IPIPE_YEE_EN .....	1739
8-808. Register Call Summary for Register IPIPE_YEE_EN .....	1739
8-809. IPIPE_YEE_TYP .....	1740
8-810. Register Call Summary for Register IPIPE_YEE_TYP .....	1740
8-811. IPIPE_YEE_SHF .....	1740
8-812. Register Call Summary for Register IPIPE_YEE_SHF .....	1740
8-813. IPIPE_YEE_MUL_00 .....	1741
8-814. Register Call Summary for Register IPIPE_YEE_MUL_00 .....	1741
8-815. IPIPE_YEE_MUL_01 .....	1741
8-816. Register Call Summary for Register IPIPE_YEE_MUL_01 .....	1741
8-817. IPIPE_YEE_MUL_02 .....	1742
8-818. Register Call Summary for Register IPIPE_YEE_MUL_02 .....	1742
8-819. IPIPE_YEE_MUL_10 .....	1742
8-820. Register Call Summary for Register IPIPE_YEE_MUL_10 .....	1742
8-821. IPIPE_YEE_MUL_11 .....	1742
8-822. Register Call Summary for Register IPIPE_YEE_MUL_11 .....	1743
8-823. IPIPE_YEE_MUL_12 .....	1743
8-824. Register Call Summary for Register IPIPE_YEE_MUL_12 .....	1743
8-825. IPIPE_YEE_MUL_20 .....	1743
8-826. Register Call Summary for Register IPIPE_YEE_MUL_20 .....	1743
8-827. IPIPE_YEE_MUL_21 .....	1744
8-828. Register Call Summary for Register IPIPE_YEE_MUL_21 .....	1744
8-829. IPIPE_YEE_MUL_22 .....	1744
8-830. Register Call Summary for Register IPIPE_YEE_MUL_22 .....	1744
8-831. IPIPE_YEE_THR .....	1744
8-832. Register Call Summary for Register IPIPE_YEE_THR .....	1745
8-833. IPIPE_YEE_E_GAN .....	1745
8-834. Register Call Summary for Register IPIPE_YEE_E_GAN .....	1745
8-835. IPIPE_YEE_E_THR_1 .....	1745
8-836. Register Call Summary for Register IPIPE_YEE_E_THR_1 .....	1746
8-837. IPIPE_YEE_E_THR_2 .....	1746
8-838. Register Call Summary for Register IPIPE_YEE_E_THR_2 .....	1746
8-839. IPIPE_YEE_G_GAN .....	1746
8-840. Register Call Summary for Register IPIPE_YEE_G_GAN .....	1746

8-841. IPIPE_YEE_G_OFT .....	1747
8-842. Register Call Summary for Register IPIPE_YEE_G_OFT .....	1747
8-843. IPIPE_BOX_EN .....	1747
8-844. Register Call Summary for Register IPIPE_BOX_EN .....	1748
8-845. IPIPE_BOX_MODE .....	1748
8-846. Register Call Summary for Register IPIPE_BOX_MODE .....	1748
8-847. IPIPE_BOX_TYP .....	1748
8-848. Register Call Summary for Register IPIPE_BOX_TYP .....	1749
8-849. IPIPE_BOX_SHF .....	1749
8-850. Register Call Summary for Register IPIPE_BOX_SHF .....	1749
8-851. IPIPE_BOX_SDR_SAD_H .....	1749
8-852. Register Call Summary for Register IPIPE_BOX_SDR_SAD_H .....	1749
8-853. IPIPE_BOX_SDR_SAD_L .....	1750
8-854. Register Call Summary for Register IPIPE_BOX_SDR_SAD_L .....	1750
8-855. IPIPE_HST_EN .....	1750
8-856. Register Call Summary for Register IPIPE_HST_EN .....	1750
8-857. IPIPE_HST_MODE .....	1751
8-858. Register Call Summary for Register IPIPE_HST_MODE .....	1751
8-859. IPIPE_HST_SEL .....	1751
8-860. Register Call Summary for Register IPIPE_HST_SEL .....	1752
8-861. IPIPE_HST_PARA .....	1752
8-862. Register Call Summary for Register IPIPE_HST_PARA .....	1753
8-863. IPIPE_HST_0_VPS .....	1753
8-864. Register Call Summary for Register IPIPE_HST_0_VPS .....	1753
8-865. IPIPE_HST_0_VSZ .....	1754
8-866. Register Call Summary for Register IPIPE_HST_0_VSZ .....	1754
8-867. IPIPE_HST_0_HPS .....	1754
8-868. Register Call Summary for Register IPIPE_HST_0_HPS .....	1755
8-869. IPIPE_HST_0_HSZ .....	1755
8-870. Register Call Summary for Register IPIPE_HST_0_HSZ .....	1755
8-871. IPIPE_HST_1_VPS .....	1755
8-872. Register Call Summary for Register IPIPE_HST_1_VPS .....	1756
8-873. IPIPE_HST_1_VSZ .....	1756
8-874. Register Call Summary for Register IPIPE_HST_1_VSZ .....	1756
8-875. IPIPE_HST_1_HPS .....	1757
8-876. Register Call Summary for Register IPIPE_HST_1_HPS .....	1757
8-877. IPIPE_HST_1_HSZ .....	1757
8-878. Register Call Summary for Register IPIPE_HST_1_HSZ .....	1758
8-879. IPIPE_HST_2_VPS .....	1758
8-880. Register Call Summary for Register IPIPE_HST_2_VPS .....	1758
8-881. IPIPE_HST_2_VSZ .....	1758
8-882. Register Call Summary for Register IPIPE_HST_2_VSZ .....	1759
8-883. IPIPE_HST_2_HPS .....	1759
8-884. Register Call Summary for Register IPIPE_HST_2_HPS .....	1759
8-885. IPIPE_HST_2_HSZ .....	1760
8-886. Register Call Summary for Register IPIPE_HST_2_HSZ .....	1760
8-887. IPIPE_HST_3_VPS .....	1760
8-888. Register Call Summary for Register IPIPE_HST_3_VPS .....	1761
8-889. IPIPE_HST_3_VSZ .....	1761



8-890. Register Call Summary for Register IPIPE_HST_3_VSZ .....	1761
8-891. IPIPE_HST_3_HPS .....	1761
8-892. Register Call Summary for Register IPIPE_HST_3_HPS.....	1762
8-893. IPIPE_HST_3_HSZ .....	1762
8-894. Register Call Summary for Register IPIPE_HST_3_HSZ.....	1762
8-895. IPIPE_HST_TBL.....	1763
8-896. Register Call Summary for Register IPIPE_HST_TBL .....	1763
8-897. IPIPE_HST_MUL_R.....	1763
8-898. Register Call Summary for Register IPIPE_HST_MUL_R .....	1764
8-899. IPIPE_HST_MUL_GR .....	1764
8-900. Register Call Summary for Register IPIPE_HST_MUL_GR .....	1764
8-901. IPIPE_HST_MUL_GB.....	1764
8-902. Register Call Summary for Register IPIPE_HST_MUL_GB .....	1764
8-903. IPIPE_HST_MUL_B.....	1765
8-904. Register Call Summary for Register IPIPE_HST_MUL_B .....	1765
8-905. ISS ISIF Registers Mapping Summary .....	1765
8-906. ISIF_SYNCEN .....	1768
8-907. Register Call Summary for Register ISIF_SYNCEN.....	1768
8-908. ISIF_MODESET .....	1768
8-909. Register Call Summary for Register ISIF_MODESET.....	1770
8-910. ISIF_HDW .....	1770
8-911. Register Call Summary for Register ISIF_HDW.....	1771
8-912. ISIF_VDW .....	1771
8-913. Register Call Summary for Register ISIF_VDW.....	1771
8-914. ISIF_PPLN .....	1771
8-915. Register Call Summary for Register ISIF_PPLN .....	1772
8-916. ISIF_LPFR.....	1772
8-917. Register Call Summary for Register ISIF_LPFR .....	1772
8-918. ISIF_SPH .....	1772
8-919. Register Call Summary for Register ISIF_SPH .....	1773
8-920. ISIF_LNH .....	1773
8-921. Register Call Summary for Register ISIF_LNH.....	1773
8-922. ISIF_LNV .....	1773
8-923. Register Call Summary for Register ISIF_LNV .....	1774
8-924. ISIF_CULH .....	1774
8-925. Register Call Summary for Register ISIF_CULH.....	1774
8-926. ISIF_CULV .....	1775
8-927. Register Call Summary for Register ISIF_CULV .....	1775
8-928. ISIF_HSIZE.....	1775
8-929. Register Call Summary for Register ISIF_HSIZE .....	1776
8-930. ISIF_CADU .....	1776
8-931. Register Call Summary for Register ISIF_CADU .....	1776
8-932. ISIF_CADL .....	1776
8-933. Register Call Summary for Register ISIF_CADL .....	1777
8-934. ISIF_LINCFG0 .....	1777
8-935. Register Call Summary for Register ISIF_LINCFG0.....	1777
8-936. ISIF_LINCFG1 .....	1778
8-937. Register Call Summary for Register ISIF_LINCFG1.....	1778
8-938. ISIF_CCOLP .....	1778

8-939. Register Call Summary for Register ISIF_CCOLP .....	1780
8-940. ISIF_CRGAIN .....	1780
8-941. Register Call Summary for Register ISIF_CRGAIN.....	1780
8-942. ISIF_CGRGAIN .....	1780
8-943. Register Call Summary for Register ISIF_CGRGAIN .....	1781
8-944. ISIF_CBGAIN .....	1781
8-945. Register Call Summary for Register ISIF_CBGAIN.....	1781
8-946. ISIF_CBGAIN .....	1781
8-947. Register Call Summary for Register ISIF_CBGAIN.....	1782
8-948. ISIF_COFSTA.....	1782
8-949. Register Call Summary for Register ISIF_COFSTA .....	1782
8-950. ISIF_VDINT0 .....	1782
8-951. Register Call Summary for Register ISIF_VDINT0 .....	1783
8-952. ISIF_VDINT1 .....	1783
8-953. Register Call Summary for Register ISIF_VDINT1 .....	1783
8-954. ISIF_VDINT2.....	1783
8-955. Register Call Summary for Register ISIF_VDINT2 .....	1783
8-956. ISIF_MISC.....	1784
8-957. Register Call Summary for Register ISIF_MISC .....	1784
8-958. ISIF_CGAMMAWD .....	1784
8-959. Register Call Summary for Register ISIF_CGAMMAWD .....	1786
8-960. ISIF_REC656IF .....	1786
8-961. Register Call Summary for Register ISIF_REC656IF .....	1786
8-962. ISIF_CCDCFG .....	1786
8-963. Register Call Summary for Register ISIF_CCDCFG .....	1788
8-964. ISIF_DFCCTL .....	1788
8-965. Register Call Summary for Register ISIF_DFCCTL .....	1789
8-966. ISIF_VDFSATLV.....	1789
8-967. Register Call Summary for Register ISIF_VDFSATLV .....	1789
8-968. ISIF_DFCMEMCTL.....	1790
8-969. Register Call Summary for Register ISIF_DFCMEMCTL .....	1790
8-970. ISIF_DFCMEM0 .....	1790
8-971. Register Call Summary for Register ISIF_DFCMEM0.....	1791
8-972. ISIF_DFCMEM1 .....	1791
8-973. Register Call Summary for Register ISIF_DFCMEM1.....	1791
8-974. ISIF_DFCMEM2 .....	1792
8-975. Register Call Summary for Register ISIF_DFCMEM2.....	1792
8-976. ISIF_DFCMEM3 .....	1792
8-977. Register Call Summary for Register ISIF_DFCMEM3.....	1792
8-978. ISIF_DFCMEM4 .....	1793
8-979. Register Call Summary for Register ISIF_DFCMEM4.....	1793
8-980. ISIF_CLAMPFCFG.....	1793
8-981. Register Call Summary for Register ISIF_CLAMPFCFG .....	1794
8-982. ISIF_CLDCOFST .....	1794
8-983. Register Call Summary for Register ISIF_CLDCOFST .....	1794
8-984. ISIF_CLSV .....	1795
8-985. Register Call Summary for Register ISIF_CLSV .....	1795
8-986. ISIF_CLHWIN0 .....	1795
8-987. Register Call Summary for Register ISIF_CLHWIN0 .....	1796

8-988. ISIF_CLHWIN1 .....	1796
8-989. Register Call Summary for Register ISIF_CLHWIN1 .....	1797
8-990. ISIF_CLHWIN2 .....	1797
8-991. Register Call Summary for Register ISIF_CLHWIN2 .....	1797
8-992. ISIF_CLVRV.....	1797
8-993. Register Call Summary for Register ISIF_CLVRV .....	1798
8-994. ISIF_CLVWIN0.....	1798
8-995. Register Call Summary for Register ISIF_CLVWIN0 .....	1798
8-996. ISIF_CLVWIN1.....	1799
8-997. Register Call Summary for Register ISIF_CLVWIN1 .....	1799
8-998. ISIF_CLVWIN2.....	1799
8-999. Register Call Summary for Register ISIF_CLVWIN2 .....	1799
8-1000. ISIF_CLVWIN3 .....	1800
8-1001. Register Call Summary for Register ISIF_CLVWIN3.....	1800
8-1002. ISIF_LSCHOFST.....	1800
8-1003. Register Call Summary for Register ISIF_LSCHOFST .....	1800
8-1004. ISIF_LSCVOFST.....	1801
8-1005. Register Call Summary for Register ISIF_LSCVOFST .....	1801
8-1006. ISIF_LSCHVAL.....	1801
8-1007. Register Call Summary for Register ISIF_LSCHVAL .....	1801
8-1008. ISIF_LSCVVAL .....	1802
8-1009. Register Call Summary for Register ISIF_LSCVVAL.....	1802
8-1010. ISIF_2DLSCCFG.....	1802
8-1011. Register Call Summary for Register ISIF_2DLSCCFG .....	1803
8-1012. ISIF_2DLSCOFST .....	1804
8-1013. Register Call Summary for Register ISIF_2DLSCOFST .....	1804
8-1014. ISIF_2DLSCINI .....	1804
8-1015. Register Call Summary for Register ISIF_2DLSCINI.....	1805
8-1016. ISIF_2DLSCGRBU.....	1805
8-1017. Register Call Summary for Register ISIF_2DLSCGRBU .....	1805
8-1018. ISIF_2DLSCGRBL .....	1805
8-1019. Register Call Summary for Register ISIF_2DLSCGRBL.....	1806
8-1020. ISIF_2DLSCGROF.....	1806
8-1021. Register Call Summary for Register ISIF_2DLSCGROF .....	1806
8-1022. ISIF_2DLSCORBU.....	1806
8-1023. Register Call Summary for Register ISIF_2DLSCORBU .....	1807
8-1024. ISIF_2DLSCORBL .....	1807
8-1025. Register Call Summary for Register ISIF_2DLSCORBL.....	1807
8-1026. ISIF_2DLSCOROF.....	1807
8-1027. Register Call Summary for Register ISIF_2DLSCOROF .....	1807
8-1028. ISIF_2DLSCIRQEN .....	1808
8-1029. Register Call Summary for Register ISIF_2DLSCIRQEN .....	1808
8-1030. ISIF_2DLSCIRQST .....	1809
8-1031. Register Call Summary for Register ISIF_2DLSCIRQST .....	1810
8-1032. ISIF_FMTCFG .....	1810
8-1033. Register Call Summary for Register ISIF_FMTCFG .....	1811
8-1034. ISIF_FMTPLEN.....	1811
8-1035. Register Call Summary for Register ISIF_FMTPLEN .....	1811
8-1036. ISIF_FMTSPH .....	1812

8-1037. Register Call Summary for Register ISIF_FMTSPH.....	1812
8-1038. ISIF_FMTLNH .....	1812
8-1039. Register Call Summary for Register ISIF_FMTLNH.....	1812
8-1040. ISIF_FMTLSV .....	1813
8-1041. Register Call Summary for Register ISIF_FMTLSV .....	1813
8-1042. ISIF_FMTLNV .....	1813
8-1043. Register Call Summary for Register ISIF_FMTLNV .....	1813
8-1044. ISIF_FMTRLEN .....	1814
8-1045. Register Call Summary for Register ISIF_FMTRLEN .....	1814
8-1046. ISIF_FMTHCNT .....	1814
8-1047. Register Call Summary for Register ISIF_FMTHCNT.....	1814
8-1048. ISIF_FMTAPTR0.....	1815
8-1049. Register Call Summary for Register ISIF_FMTAPTR0 .....	1815
8-1050. ISIF_FMTAPTR1 .....	1815
8-1051. Register Call Summary for Register ISIF_FMTAPTR1 .....	1815
8-1052. ISIF_FMTAPTR2.....	1816
8-1053. Register Call Summary for Register ISIF_FMTAPTR2 .....	1816
8-1054. ISIF_FMTAPTR3.....	1816
8-1055. Register Call Summary for Register ISIF_FMTAPTR3 .....	1816
8-1056. ISIF_FMTAPTR4.....	1817
8-1057. Register Call Summary for Register ISIF_FMTAPTR4 .....	1817
8-1058. ISIF_FMTAPTR5.....	1817
8-1059. Register Call Summary for Register ISIF_FMTAPTR5 .....	1817
8-1060. ISIF_FMTAPTR6.....	1818
8-1061. Register Call Summary for Register ISIF_FMTAPTR6 .....	1818
8-1062. ISIF_FMTAPTR7 .....	1818
8-1063. Register Call Summary for Register ISIF_FMTAPTR7 .....	1818
8-1064. ISIF_FMTAPTR8.....	1819
8-1065. Register Call Summary for Register ISIF_FMTAPTR8 .....	1819
8-1066. ISIF_FMTAPTR9.....	1819
8-1067. Register Call Summary for Register ISIF_FMTAPTR9 .....	1819
8-1068. ISIF_FMTAPTR10 .....	1820
8-1069. Register Call Summary for Register ISIF_FMTAPTR10 .....	1820
8-1070. ISIF_FMTAPTR11 .....	1820
8-1071. Register Call Summary for Register ISIF_FMTAPTR11 .....	1820
8-1072. ISIF_FMTAPTR12 .....	1821
8-1073. Register Call Summary for Register ISIF_FMTAPTR12.....	1821
8-1074. ISIF_FMTAPTR13 .....	1821
8-1075. Register Call Summary for Register ISIF_FMTAPTR13.....	1821
8-1076. ISIF_FMTAPTR14 .....	1822
8-1077. Register Call Summary for Register ISIF_FMTAPTR14 .....	1822
8-1078. ISIF_FMTAPTR15 .....	1822
8-1079. Register Call Summary for Register ISIF_FMTAPTR15.....	1822
8-1080. ISIF_FMTPGMVFO .....	1823
8-1081. Register Call Summary for Register ISIF_FMTPGMVFO .....	1824
8-1082. ISIF_FMTPGMVF1 .....	1824
8-1083. Register Call Summary for Register ISIF_FMTPGMVF1 .....	1825
8-1084. ISIF_FMTPGMAPU0 .....	1826
8-1085. Register Call Summary for Register ISIF_FMTPGMAPU0.....	1827

8-1086. ISIF_FMTPGMAPU1 .....	1827
8-1087. Register Call Summary for Register ISIF_FMTPGMAPU1 .....	1828
8-1088. ISIF_FMTPGMAPS0.....	1829
8-1089. Register Call Summary for Register ISIF_FMTPGMAPS0 .....	1829
8-1090. ISIF_FMTPGMAPS1.....	1829
8-1091. Register Call Summary for Register ISIF_FMTPGMAPS1 .....	1829
8-1092. ISIF_FMTPGMAPS2.....	1830
8-1093. Register Call Summary for Register ISIF_FMTPGMAPS2 .....	1830
8-1094. ISIF_FMTPGMAPS3.....	1830
8-1095. Register Call Summary for Register ISIF_FMTPGMAPS3 .....	1830
8-1096. ISIF_FMTPGMAPS4.....	1831
8-1097. Register Call Summary for Register ISIF_FMTPGMAPS4 .....	1831
8-1098. ISIF_FMTPGMAPS5.....	1831
8-1099. Register Call Summary for Register ISIF_FMTPGMAPS5 .....	1831
8-1100. ISIF_FMTPGMAPS6.....	1832
8-1101. Register Call Summary for Register ISIF_FMTPGMAPS6 .....	1832
8-1102. ISIF_FMTPGMAPS7.....	1832
8-1103. Register Call Summary for Register ISIF_FMTPGMAPS7 .....	1832
8-1104. ISIF_CSCCTL .....	1833
8-1105. Register Call Summary for Register ISIF_CSCCTL .....	1833
8-1106. ISIF_CSCM0.....	1833
8-1107. Register Call Summary for Register ISIF_CSCM0 .....	1833
8-1108. ISIF_CSCM1 .....	1834
8-1109. Register Call Summary for Register ISIF_CSCM1 .....	1834
8-1110. ISIF_CSCM2.....	1834
8-1111. Register Call Summary for Register ISIF_CSCM2 .....	1834
8-1112. ISIF_CSCM3.....	1835
8-1113. Register Call Summary for Register ISIF_CSCM3 .....	1835
8-1114. ISIF_CSCM4.....	1835
8-1115. Register Call Summary for Register ISIF_CSCM4 .....	1835
8-1116. ISIF_CSCM5.....	1836
8-1117. Register Call Summary for Register ISIF_CSCM5 .....	1836
8-1118. ISIF_CSCM6.....	1836
8-1119. Register Call Summary for Register ISIF_CSCM6 .....	1836
8-1120. ISIF_CSCM7.....	1837
8-1121. Register Call Summary for Register ISIF_CSCM7 .....	1837
8-1122. ISIF_CLKCTL.....	1837
8-1123. Register Call Summary for Register ISIF_CLKCTL .....	1837
8-1124. ISS IPIPEIF Registers Mapping Summary .....	1838
8-1125. IPIPEIF_ENABLE .....	1838
8-1126. Register Call Summary for Register IPIPEIF_ENABLE .....	1839
8-1127. IPIPEIF_CFG1.....	1839
8-1128. Register Call Summary for Register IPIPEIF_CFG1 .....	1841
8-1129. IPIPEIF_PPLN.....	1841
8-1130. Register Call Summary for Register IPIPEIF_PPLN .....	1842
8-1131. IPIPEIF_LPFR .....	1842
8-1132. Register Call Summary for Register IPIPEIF_LPFR .....	1842
8-1133. IPIPEIF_HNUM.....	1842
8-1134. Register Call Summary for Register IPIPEIF_HNUM .....	1843

8-1135. IPIPEIF_VNUM .....	1843
8-1136. Register Call Summary for Register IPIPEIF_VNUM .....	1843
8-1137. IPIPEIF_ADDRU .....	1843
8-1138. Register Call Summary for Register IPIPEIF_ADDRU .....	1844
8-1139. IPIPEIF_ADDRL .....	1844
8-1140. Register Call Summary for Register IPIPEIF_ADDRL .....	1844
8-1141. IPIPEIF_ADOFS .....	1844
8-1142. Register Call Summary for Register IPIPEIF_ADOFS .....	1845
8-1143. IPIPEIF_RSZ .....	1845
8-1144. Register Call Summary for Register IPIPEIF_RSZ .....	1845
8-1145. IPIPEIF_GAIN .....	1845
8-1146. Register Call Summary for Register IPIPEIF_GAIN .....	1846
8-1147. IPIPEIF_DPCM .....	1846
8-1148. Register Call Summary for Register IPIPEIF_DPCM .....	1846
8-1149. IPIPEIF_CFG2 .....	1847
8-1150. Register Call Summary for Register IPIPEIF_CFG2 .....	1848
8-1151. IPIPEIF_INIRSZ .....	1848
8-1152. Register Call Summary for Register IPIPEIF_INIRSZ .....	1849
8-1153. IPIPEIF_OCLIP .....	1849
8-1154. Register Call Summary for Register IPIPEIF_OCLIP .....	1849
8-1155. IPIPEIF_DTUDF .....	1849
8-1156. Register Call Summary for Register IPIPEIF_DTUDF .....	1850
8-1157. IPIPEIF_CLKDIV .....	1850
8-1158. Register Call Summary for Register IPIPEIF_CLKDIV .....	1850
8-1159. IPIPEIF_DPC1 .....	1850
8-1160. Register Call Summary for Register IPIPEIF_DPC1 .....	1851
8-1161. IPIPEIF_DPC2 .....	1851
8-1162. Register Call Summary for Register IPIPEIF_DPC2 .....	1851
8-1163. IPIPEIF_RSZ3A .....	1851
8-1164. Register Call Summary for Register IPIPEIF_RSZ3A .....	1852
8-1165. IPIPEIF_INIRSZ3A .....	1852
8-1166. Register Call Summary for Register IPIPEIF_INIRSZ3A .....	1852
8-1167. ISS H3A Registers Mapping Summary .....	1853
8-1168. H3A_PID .....	1853
8-1169. Register Call Summary for Register H3A_PID .....	1854
8-1170. H3A_PCR .....	1854
8-1171. Register Call Summary for Register H3A_PCR .....	1855
8-1172. H3A_AFPAX1 .....	1856
8-1173. Register Call Summary for Register H3A_AFPAX1 .....	1856
8-1174. H3A_AFPAX2 .....	1856
8-1175. Register Call Summary for Register H3A_AFPAX2 .....	1857
8-1176. H3A_AFPAXSTART .....	1857
8-1177. Register Call Summary for Register H3A_AFPAXSTART .....	1858
8-1178. H3A_AFIIRSH .....	1858
8-1179. Register Call Summary for Register H3A_AFIIRSH .....	1858
8-1180. H3A_AFBUFST .....	1859
8-1181. Register Call Summary for Register H3A_AFBUFST .....	1859
8-1182. H3A_AFCOEF010 .....	1859
8-1183. Register Call Summary for Register H3A_AFCOEF010 .....	1859

8-1184. H3A_AFCOEF032 .....	1860
8-1185. Register Call Summary for Register H3A_AFCOEF032 .....	1860
8-1186. H3A_AFCOEF054 .....	1860
8-1187. Register Call Summary for Register H3A_AFCOEF054 .....	1860
8-1188. H3A_AFCOEF076 .....	1861
8-1189. Register Call Summary for Register H3A_AFCOEF076 .....	1861
8-1190. H3A_AFCOEF098 .....	1861
8-1191. Register Call Summary for Register H3A_AFCOEF098 .....	1861
8-1192. H3A_AFCOEF0010 .....	1862
8-1193. Register Call Summary for Register H3A_AFCOEF0010 .....	1862
8-1194. H3A_AFCOEF110 .....	1862
8-1195. Register Call Summary for Register H3A_AFCOEF110 .....	1862
8-1196. H3A_AFCOEF132 .....	1863
8-1197. Register Call Summary for Register H3A_AFCOEF132 .....	1863
8-1198. H3A_AFCOEF154 .....	1863
8-1199. Register Call Summary for Register H3A_AFCOEF154 .....	1863
8-1200. H3A_AFCOEF176 .....	1864
8-1201. Register Call Summary for Register H3A_AFCOEF176 .....	1864
8-1202. H3A_AFCOEF198 .....	1864
8-1203. Register Call Summary for Register H3A_AFCOEF198 .....	1864
8-1204. H3A_AFCOEF1010 .....	1865
8-1205. Register Call Summary for Register H3A_AFCOEF1010 .....	1865
8-1206. H3A_AEWWIN1 .....	1865
8-1207. Register Call Summary for Register H3A_AEWWIN1 .....	1866
8-1208. H3A_AEWINSTART .....	1866
8-1209. Register Call Summary for Register H3A_AEWINSTART .....	1866
8-1210. H3A_AEWINBLK .....	1867
8-1211. Register Call Summary for Register H3A_AEWINBLK .....	1867
8-1212. H3A_AEWSUBWIN .....	1867
8-1213. Register Call Summary for Register H3A_AEWSUBWIN .....	1868
8-1214. H3A_AEWBUFST .....	1868
8-1215. Register Call Summary for Register H3A_AEWBUFST .....	1868
8-1216. H3A_AEWCFG .....	1869
8-1217. Register Call Summary for Register H3A_AEWCFG .....	1869
8-1218. H3A_LINE_START .....	1869
8-1219. Register Call Summary for Register H3A_LINE_START .....	1870
8-1220. H3A_VFV_CFG1 .....	1870
8-1221. Register Call Summary for Register H3A_VFV_CFG1 .....	1870
8-1222. H3A_VFV_CFG2 .....	1870
8-1223. Register Call Summary for Register H3A_VFV_CFG2 .....	1871
8-1224. H3A_VFV_CFG3 .....	1871
8-1225. Register Call Summary for Register H3A_VFV_CFG3 .....	1871
8-1226. H3A_VFV_CFG4 .....	1871
8-1227. Register Call Summary for Register H3A_VFV_CFG4 .....	1872
8-1228. H3A_HVF_THR .....	1872
8-1229. Register Call Summary for Register H3A_HVF_THR .....	1872
8-1230. H3A_ADVANCED .....	1872
8-1231. Register Call Summary for Register H3A_ADVANCED .....	1873
8-1232. Integration Attributes .....	1876

8-1233. Clocks and Resets .....	1877
8-1234. Hardware Requests.....	1877
8-1235. ISS SIMCOP Clock Control Register Settings .....	1878
8-1236. Autoclock Gating Bit Control.....	1878
8-1237. SIMCOP Interrupts.....	1880
8-1238. Global Initialization of Surrounding Modules .....	1881
8-1239. ISS SIMCOP Global Initialization .....	1881
8-1240. JPEG Encode Pipeline .....	1883
8-1241. SIMCOP Instance Summary.....	1884
8-1242. SIMCOP_CONTROL Register Mapping Summary .....	1884
8-1243. SIMCOP_HL_REVISION.....	1884
8-1244. Register Call Summary for Register SIMCOP_HL_REVISION .....	1885
8-1245. SIMCOP_HL_HWINFO.....	1885
8-1246. Register Call Summary for Register SIMCOP_HL_HWINFO .....	1886
8-1247. SIMCOP_HL_SYSCONFIG.....	1886
8-1248. Register Call Summary for Register SIMCOP_HL_SYSCONFIG .....	1887
8-1249. SIMCOP_HL_IRQSTATUS_RAW_i .....	1887
8-1250. Register Call Summary for Register SIMCOP_HL_IRQSTATUS_RAW_i.....	1889
8-1251. SIMCOP_HL_IRQSTATUS_i .....	1890
8-1252. Register Call Summary for Register SIMCOP_HL_IRQSTATUS_i .....	1892
8-1253. SIMCOP_HL_IRQENABLE_SET_i .....	1893
8-1254. Register Call Summary for Register SIMCOP_HL_IRQENABLE_SET_i.....	1895
8-1255. SIMCOP_HL_IRQENABLE_CLR_i .....	1896
8-1256. Register Call Summary for Register SIMCOP_HL_IRQENABLE_CLR_i.....	1898
8-1257. SIMCOP_CTRL .....	1899
8-1258. Register Call Summary for Register SIMCOP_CTRL .....	1900
8-1259. SIMCOP_CLKCTRL .....	1901
8-1260. Register Call Summary for Register SIMCOP_CLKCTRL.....	1902
8-1261. SIMCOP Memory Mapping .....	1903
8-1262. Integration Attributes.....	1905
8-1263. Clocks and Resets .....	1905
8-1264. Hardware Requests.....	1906
8-1265. Static Crossbar Control Summary .....	1908
8-1266. Static Crossbar Control Summary (2/2).....	1908
8-1267. Image Buffer Address Map .....	1909
8-1268. Coefficient Buffer Organization.....	1911
8-1269. Quantization Table Storage in QUANT Memory .....	1911
8-1270. QUANT Memory Organization .....	1912
8-1271. HUFF Memory Organization.....	1912
8-1272. Bitstream Buffer Mapping to SIMCOP DMA.....	1912
8-1273. LDC_LUT Memory Organization .....	1913
8-1274. LUT Memory Format on 32-Bit Words .....	1913
8-1275. Rotation Macroblock Pipeline.....	1920
8-1276. LDC Transformation .....	1921
8-1277. Concurrent Software and Hardware Sequencing.....	1922
8-1278. Hardware Sequencer Instance Summary.....	1923
8-1279. HWSEQ Registers Mapping Summary.....	1923
8-1280. SIMCOP_HWSEQ_CTRL.....	1924
8-1281. Register Call Summary for Register SIMCOP_HWSEQ_CTRL .....	1925



8-1282. SIMCOP_HWSEQ_STATUS .....	1926
8-1283. Register Call Summary for Register SIMCOP_HWSEQ_STATUS.....	1926
8-1284. SIMCOP_HWSEQ_OVERRIDE .....	1926
8-1285. Register Call Summary for Register SIMCOP_HWSEQ_OVERRIDE .....	1927
8-1286. SIMCOP_HWSEQ_STEP_CTRL_OVERRIDE.....	1927
8-1287. Register Call Summary for Register SIMCOP_HWSEQ_STEP_CTRL_OVERRIDE .....	1930
8-1288. SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE .....	1930
8-1289. Register Call Summary for Register SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE .....	1932
8-1290. SIMCOP_HWSEQ_STEP_CTRL2_OVERRIDE .....	1932
8-1291. Register Call Summary for Register SIMCOP_HWSEQ_STEP_CTRL2_OVERRIDE.....	1933
8-1292. SIMCOP_HWSEQ_STEP_CTRL_i .....	1934
8-1293. Register Call Summary for Register SIMCOP_HWSEQ_STEP_CTRL_i.....	1936
8-1294. SIMCOP_HWSEQ_STEP_SWITCH_i .....	1936
8-1295. Register Call Summary for Register SIMCOP_HWSEQ_STEP_SWITCH_i.....	1938
8-1296. SIMCOP_HWSEQ_STEP_IMX_CTRL_i.....	1938
8-1297. Register Call Summary for Register SIMCOP_HWSEQ_STEP_IMX_CTRL_i .....	1939
8-1298. SIMCOP_HWSEQ_STEP_CTRL2_i.....	1939
8-1299. Register Call Summary for Register SIMCOP_HWSEQ_STEP_CTRL2_i .....	1940
8-1300. Integration Attributes.....	1942
8-1301. Clocks and Resets .....	1942
8-1302. Hardware Requests.....	1943
8-1303. Local Power-Management Features.....	1944
8-1304. SIMCOP DMA Events List .....	1945
8-1305. Allowed Maximum Burst Size .....	1949
8-1306. Initialization of Surrounding Modules .....	1949
8-1307. SIMCOP DMA Instance Summary .....	1953
8-1308. DMA Register Mapping Summary .....	1953
8-1309. SIMCOP_DMA_REVISION .....	1954
8-1310. Register Call Summary for Register SIMCOP_DMA_REVISION.....	1954
8-1311. SIMCOP_DMA_HWINFO .....	1954
8-1312. Register Call Summary for Register SIMCOP_DMA_HWINFO.....	1954
8-1313. SIMCOP_DMA_SYSCONFIG .....	1955
8-1314. Register Call Summary for Register SIMCOP_DMA_SYSCONFIG.....	1955
8-1315. SIMCOP_DMA_CTRL .....	1955
8-1316. Register Call Summary for Register SIMCOP_DMA_CTRL .....	1956
8-1317. SIMCOP_DMA_IRQSTATUS_RAW_j.....	1956
8-1318. Register Call Summary for Register SIMCOP_DMA_IRQSTATUS_RAW_j .....	1958
8-1319. SIMCOP_DMA_IRQSTATUS_j .....	1959
8-1320. Register Call Summary for Register SIMCOP_DMA_IRQSTATUS_j.....	1961
8-1321. SIMCOP_DMA_IRQENABLE_SET_j.....	1961
8-1322. Register Call Summary for Register SIMCOP_DMA_IRQENABLE_SET_j .....	1963
8-1323. SIMCOP_DMA_IRQENABLE_CLR_j.....	1963
8-1324. Register Call Summary for Register SIMCOP_DMA_IRQENABLE_CLR_j .....	1966
8-1325. SIMCOP_DMA_CHAN_CTRL_i.....	1966
8-1326. Register Call Summary for Register SIMCOP_DMA_CHAN_CTRL_i .....	1968
8-1327. SIMCOP_DMA_CHAN_SMEM_ADDR_i .....	1968
8-1328. Register Call Summary for Register SIMCOP_DMA_CHAN_SMEM_ADDR_i.....	1968
8-1329. SIMCOP_DMA_CHAN_SMEM_OFST_i.....	1968
8-1330. Register Call Summary for Register SIMCOP_DMA_CHAN_SMEM_OFST_i .....	1968

8-1331. SIMCOP_DMA_CHAN_BUF_OFST_i.....	1969
8-1332. Register Call Summary for Register SIMCOP_DMA_CHAN_BUF_OFST_i.....	1969
8-1333. SIMCOP_DMA_CHAN_BUF_ADDR_i.....	1969
8-1334. Register Call Summary for Register SIMCOP_DMA_CHAN_BUF_ADDR_i.....	1969
8-1335. SIMCOP_DMA_CHAN_BLOCK_SIZE_i.....	1970
8-1336. Register Call Summary for Register SIMCOP_DMA_CHAN_BLOCK_SIZE_i.....	1970
8-1337. SIMCOP_DMA_CHAN_FRAME_i.....	1970
8-1338. Register Call Summary for Register SIMCOP_DMA_CHAN_FRAME_i.....	1970
8-1339. SIMCOP_DMA_CHAN_CURRENT_BLOCK_i.....	1971
8-1340. Register Call Summary for Register SIMCOP_DMA_CHAN_CURRENT_BLOCK_i.....	1971
8-1341. SIMCOP_DMA_CHAN_BLOCK_STEP_i.....	1971
8-1342. Register Call Summary for Register SIMCOP_DMA_CHAN_BLOCK_STEP_i.....	1971
8-1343. Integration Attributes.....	1975
8-1344. Clocks and Resets.....	1976
8-1345. Hardware Requests.....	1976
8-1346. Local Power-Management Features.....	1977
8-1347. Events.....	1978
8-1348. Initialization of Surrounding Modules.....	1982
8-1349. DCT Processing Procedure.....	1982
8-1350. IDCT Processing Procedure.....	1982
8-1351. ISS SIMCOP DCT Instance Summary.....	1984
8-1352. DCT Register Mapping Summary.....	1984
8-1353. DCT_VERSION.....	1985
8-1354. Register Call Summary for Register DCT_VERSION.....	1985
8-1355. DCT_CTRL.....	1985
8-1356. Register Call Summary for Register DCT_CTRL.....	1985
8-1357. DCT_CFG.....	1986
8-1358. Register Call Summary for Register DCT_CFG.....	1986
8-1359. DCT_SPTR.....	1987
8-1360. Register Call Summary for Register DCT_SPTR.....	1987
8-1361. DCT_FPTR.....	1987
8-1362. Register Call Summary for Register DCT_FPTR.....	1987
8-1363. Integration Attributes.....	1990
8-1364. Clocks and Resets.....	1990
8-1365. Hardware Requests.....	1991
8-1366. Local Power-Management Feature.....	1992
8-1367. Events.....	1992
8-1368. Image Buffer Block Format.....	1994
8-1369. Recommended QMEM Allocation for YUV4:2:0 and YUV4:2:2.....	1995
8-1370. Initialization of Surrounding Modules.....	1999
8-1371. JPEG Encoding Procedure.....	1999
8-1372. JPEG Decoding Procedure.....	2000
8-1373. Concurrent JPEG Encoding and Decoding.....	2002
8-1374. Sample Luma DC Huffman Code Table.....	2005
8-1375. ISS SIMCOP VLCDJ Instance Summary.....	2007
8-1376. VLCDJ Register Mapping Summary.....	2007
8-1377. VLCDJ_REVISION.....	2008
8-1378. Register Call Summary for Register VLCDJ_REVISION.....	2008
8-1379. VLCDJ_CTRL.....	2008

8-1380. Register Call Summary for Register VLCDJ_CTRL .....	2009
8-1381. VLCDJE_CFG .....	2009
8-1382. Register Call Summary for Register VLCDJE_CFG.....	2010
8-1383. VLCDJE_DCPREDY.....	2010
8-1384. Register Call Summary for Register VLCDJE_DCPREDY .....	2010
8-1385. VLCDJE_DCPREDUV.....	2010
8-1386. Register Call Summary for Register VLCDJE_DCPREDUV .....	2010
8-1387. VLCDJE_BSPTR.....	2011
8-1388. Register Call Summary for Register VLCDJE_BSPTR .....	2011
8-1389. VLCDJE_CBUF .....	2011
8-1390. Register Call Summary for Register VLCDJE_CBUF .....	2012
8-1391. VLCDJE_RSTCFG.....	2012
8-1392. Register Call Summary for Register VLCDJE_RSTCFG .....	2012
8-1393. VLCDJE_DCTQM.....	2012
8-1394. Register Call Summary for Register VLCDJE_DCTQM .....	2013
8-1395. VLCDJE_VLCTBL.....	2013
8-1396. Register Call Summary for Register VLCDJE_VLCTBL .....	2013
8-1397. VLCDJE_RSTPTR .....	2013
8-1398. Register Call Summary for Register VLCDJE_RSTPTR .....	2014
8-1399. VLCDJE_RSTOFST .....	2014
8-1400. Register Call Summary for Register VLCDJE_RSTOFST.....	2014
8-1401. VLCDJD_CFG .....	2014
8-1402. Register Call Summary for Register VLCDJD_CFG .....	2015
8-1403. VLCDJD_DCPREDY .....	2015
8-1404. Register Call Summary for Register VLCDJD_DCPREDY .....	2015
8-1405. VLCDJD_DCPREDUV .....	2015
8-1406. Register Call Summary for Register VLCDJD_DCPREDUV .....	2015
8-1407. VLCDJD_BSPTR.....	2016
8-1408. Register Call Summary for Register VLCDJD_BSPTR .....	2016
8-1409. VLCDJD_CBUF .....	2016
8-1410. Register Call Summary for Register VLCDJD_CBUF.....	2017
8-1411. VLCDJD_DCTQM.....	2017
8-1412. Register Call Summary for Register VLCDJD_DCTQM .....	2017
8-1413. VLCDJD_CTRLTBL.....	2017
8-1414. Register Call Summary for Register VLCDJD_CTRLTBL .....	2018
8-1415. VLCDJD_DCDTBL01.....	2018
8-1416. Register Call Summary for Register VLCDJD_DCDTBL01 .....	2018
8-1417. VLCDJD_DCDTBL23.....	2018
8-1418. Register Call Summary for Register VLCDJD_DCDTBL23 .....	2019
8-1419. VLCDJD_DCTERR .....	2019
8-1420. Register Call Summary for Register VLCDJD_DCTERR.....	2019
8-1421. ISS SIMCOP ROT Integration Attributes.....	2022
8-1422. ISS SIMCOP ROT Clocks and Resets.....	2022
8-1423. ISS SIMCOP ROT Hardware Requests.....	2023
8-1424. ISS SIMCOP ROT Global Initialization of Surrounding Modules .....	2031
8-1425. ISS SIMCOP ROT Global Initialization of the ROT.....	2031
8-1426. ISS SIMCOP ROT Trigger Source Bit Value .....	2031
8-1427. ISS SIMCOP ROT Data-Block Shifting .....	2032
8-1428. ISS SIMCOP ROT Instance Summary .....	2033

8-1429. ROT Register Mapping Summary.....	2033
8-1430. ROT_REVISION .....	2033
8-1431. Register Call Summary for Register ROT_REVISION .....	2034
8-1432. ROT_CTRL .....	2034
8-1433. Register Call Summary for Register ROT_CTRL.....	2034
8-1434. ROT_CFG.....	2034
8-1435. Register Call Summary for Register ROT_CFG .....	2035
8-1436. ROT_BLKSZ.....	2035
8-1437. Register Call Summary for Register ROT_BLKSZ .....	2036
8-1438. ROT_SRC_START1 .....	2036
8-1439. Register Call Summary for Register ROT_SRC_START1 .....	2036
8-1440. ROT_SRC_LOFST .....	2036
8-1441. Register Call Summary for Register ROT_SRC_LOFST .....	2037
8-1442. ROT_DST_START1 .....	2037
8-1443. Register Call Summary for Register ROT_DST_START1.....	2037
8-1444. ROT_DST_LOFST.....	2037
8-1445. Register Call Summary for Register ROT_DST_LOFST .....	2038
8-1446. ROT_SRC_START2.....	2038
8-1447. Register Call Summary for Register ROT_SRC_START2 .....	2038
8-1448. ROT_DST_START2 .....	2038
8-1449. Register Call Summary for Register ROT_DST_START2.....	2038
9-1. 3rd Party IP Features .....	2042
9-2. Integration Attributes .....	2044
9-3. Clocks and Resets .....	2044
9-4. Hardware Requests .....	2045
9-5. Local Power-Management Features .....	2047
9-6. Interrupts and Events Description .....	2048
9-7. Performance Parameters.....	2049
9-8. Global Initialization of Surrounding Modules.....	2050
9-9. FDIF Global Initialization .....	2050
9-10. FDIF Instance Summary.....	2053
9-11. FDIF Registers Mapping Summary .....	2053
9-12. FDIF_REVISION.....	2054
9-13. Register Call Summary for Register FDIF_REVISION .....	2054
9-14. FDIF_HWINFO.....	2054
9-15. Register Call Summary for Register FDIF_HWINFO .....	2054
9-16. FDIF_SYSCONFIG.....	2055
9-17. Register Call Summary for Register FDIF_SYSCONFIG .....	2055
9-18. FDIF_IRQSTATUS_RAW_j.....	2056
9-19. Register Call Summary for Register FDIF_IRQSTATUS_RAW_j.....	2056
9-20. FDIF_IRQSTATUS_j.....	2057
9-21. Register Call Summary for Register FDIF_IRQSTATUS_j .....	2057
9-22. FDIF_IRQENABLE_SET_j .....	2057
9-23. Register Call Summary for Register FDIF_IRQENABLE_SET_j.....	2058
9-24. FDIF_IRQENABLE_CLR_j.....	2058
9-25. Register Call Summary for Register FDIF_IRQENABLE_CLR_j.....	2059
9-26. FDIF_PICADDR .....	2059
9-27. Register Call Summary for Register FDIF_PICADDR .....	2059
9-28. FDIF_CTRL .....	2059

9-29. Register Call Summary for Register FDIF_CTRL .....	2060
9-30. FDIF_WKADDR .....	2061
9-31. Register Call Summary for Register FDIF_WKADDR .....	2061
9-32. FD_CTRL .....	2061
9-33. Register Call Summary for Register FD_CTRL .....	2062
9-34. FD_DNUM .....	2062
9-35. Register Call Summary for Register FD_DNUM .....	2062
9-36. FD_DCOND .....	2062
9-37. Register Call Summary for Register FD_DCOND .....	2063
9-38. FD_STARTX .....	2063
9-39. Register Call Summary for Register FD_STARTX .....	2063
9-40. FD_STARTY .....	2063
9-41. Register Call Summary for Register FD_STARTY .....	2064
9-42. FD_SIZEX .....	2064
9-43. Register Call Summary for Register FD_SIZEX .....	2064
9-44. FD_SIZEY .....	2064
9-45. Register Call Summary for Register FD_SIZEY .....	2065
9-46. FD_LHIT .....	2065
9-47. Register Call Summary for Register FD_LHIT .....	2065
9-48. FD_CENTERX_i .....	2065
9-49. Register Call Summary for Register FD_CENTERX_i .....	2065
9-50. FD_CENTERY_i .....	2066
9-51. Register Call Summary for Register FD_CENTERY_i .....	2066
9-52. FD_CONFSIZE_i .....	2066
9-53. Register Call Summary for Register FD_CONFSIZE_i .....	2066
9-54. FD_ANGLE_i .....	2067
9-55. Register Call Summary for Register FD_ANGLE_i .....	2067
10-1. Display Parallel Interface Signals Mapping .....	2073
10-2. DSS Clocks .....	2077
10-3. DSS Modules Clock Sources .....	2077
10-4. Possible Digital Clock Division for the Video Encoder .....	2077
10-5. DSS Resets .....	2079
10-6. DSS Power Domains .....	2079
10-7. DSS Interrupts .....	2082
10-8. DSS DMA Requests .....	2083
10-9. DSS Instance Summary .....	2085
10-10. DSS Registers Mapping Summary .....	2085
10-11. DSS_REVISION .....	2085
10-12. Register Call Summary for Register DSS_REVISION .....	2085
10-13. DSS_SYSSTATUS .....	2086
10-14. Register Call Summary for Register DSS_SYSSTATUS .....	2086
10-15. DSS_CTRL .....	2086
10-16. Register Call Summary for Register DSS_CTRL .....	2088
10-17. DSS_STATUS .....	2088
10-18. Register Call Summary for Register DSS_STATUS .....	2090
10-19. Parallel Interface Signals .....	2095
10-20. Number of Displayed Pixels per Pixel Clock Cycle Based on Display Type .....	2096
10-21. Programmable Fields in Bypass Mode .....	2101
10-22. Integration Attributes .....	2108

10-23. Clocks and Resets .....	2109
10-24. Hardware Requests .....	2109
10-25. DISPC Interrupts.....	2114
10-26. DMA Buffer Size .....	2116
10-27. Register Settings for Accessing Image in Internal Memory.....	2118
10-28. Register settings for rotation using TILER .....	2122
10-29. Mode Definition .....	2123
10-30. Orientation Definition.....	2123
10-31. Memory Formats Supported .....	2124
10-32. Replication Enabled: RGB Pixel Formats Remapping into ARGB40-10.10.10.10 .....	2127
10-33. Replication Disabled: RGB Pixel Formats Remapping into ARGB40-10.10.10.10 .....	2127
10-34. Replication Enabled: RGB Pixel Formats Remapping into ARGB32-8888 .....	2132
10-35. Replication Disabled: RGB Pixel Formats Remapping into ARGB32-8888 .....	2132
10-36. Color Space Conversion YUV to RGB Bit Field Setting.....	2133
10-37. Line Buffer Width for Scaler Unit .....	2137
10-38. Register Bifield Associated to Coefficient for ARGB and Y Configuration in VIDp Scaler .....	2139
10-39. Register Bifield Associated to Coefficient for Cb and Cr Configuration in VIDp Scaler .....	2139
10-40. Vertical/Horizontal Accumulator Phase .....	2141
10-41. Pixel Clock Frequency Limitations - Any pixel format - Active Matrix Display.....	2143
10-42. Pixel Clock Frequency Limitations - Any pixel format - Passive Matrix Display - Mono4.....	2144
10-43. Pixel Clock Frequency Limitations - Any pixel format - Passive Matrix Display - Mono8.....	2144
10-44. Pixel Clock Frequency Limitations - Any pixel format - Passive Matrix Display - Color.....	2144
10-45. Color Space Conversion RGB to YUV Bit Field Setting.....	2145
10-46. Register Bifield Associated to Coefficient for ARGB and Y Configuration in WB Scaler.....	2147
10-47. Register Bifield Associated to Coefficient for Cb and Cr Configuration in WB Scaler .....	2148
10-48. Vertical/Horizontal Accumulator Phase .....	2149
10-49. Truncation Logic .....	2150
10-50. Pipeline connection to LCD, TV or WB output.....	2151
10-51. Z-Order Register Settings and Default Configuration.....	2152
10-52. Alpha Blending - ARGB.....	2156
10-53. CPR or RGB to YUV Coefficients with Bit Fields Associated .....	2160
10-54. Programming Rules .....	2166
10-55. PPL and LLP value for TV standard .....	2170
10-56. PPL and LLP value for HD standard.....	2170
10-57. Shadow Registers .....	2171
10-58. Global Initialization of Surrounding Modules.....	2177
10-59. DISPC Configuration.....	2178
10-60. Configure the GFX DMA Channel.....	2178
10-61. Configure the Video DMA Channel .....	2178
10-62. Configure the WB DMA Channel .....	2179
10-63. Configure the GFX Pipeline .....	2180
10-64. Configure the GFX Window .....	2180
10-65. Configure the GFX Pipeline Processing.....	2180
10-66. Configure the Palette Table .....	2181
10-67. Configure the GFX Pipeline Layer Output .....	2182
10-68. Configure the Video Pipeline.....	2182
10-69. Configure the Video Window.....	2182
10-70. Configure the Video Pipeline Processing .....	2183
10-71. Configure the VC-1 Range Mapping.....	2184

10-72. Configure the Video Color Space Conversion .....	2184
10-73. Configure the Video Scaler Unit for RGB Pixel Formats or Y Component .....	2185
10-74. Configure the Video Scaler Unit for Cb and Cr Components.....	2185
10-75. Configure the Video Pipeline Layer Output .....	2186
10-76. Configure the WB Pipeline .....	2186
10-77. Configure the Capture Window .....	2187
10-78. Configure the WB Scaler Unit for RGB Pixel Formats or Y Component .....	2187
10-79. Configure the WB Scaler Unit for Cb and Cr Components .....	2187
10-80. Configure the WB Color Space Conversion Unit.....	2188
10-81. Configure the LCD Output.....	2188
10-82. Configure the LCD Overlay Manager .....	2188
10-83. Configure the Gamma Table for LCD1 .....	2189
10-84. Configure the Gamma Table for LCD2 .....	2189
10-85. Configure the Color Phase Rotation .....	2189
10-86. Configure the LCD Panel Timings and Parameters .....	2190
10-87. Configure the TV Output.....	2190
10-88. Configure the TV Overlay Manager.....	2191
10-89. Configure the Gamma Table for TV Output.....	2191
10-90. Configure the TV Panel Timings and Parameters .....	2191
10-91. DISPC_SIZE_TV values NTSC/PAL and HD Standard .....	2191
10-92. DISPC Instance Summary .....	2192
10-93. DISPC_VIDp_BA_j Logical Register Mapping.....	2192
10-94. DISPC_VIDp_BA_UV_j Logical Register Mapping .....	2193
10-95. DISPC_VIDp_POSITION Logical Register Mapping .....	2193
10-96. DISPC_VIDp_SIZE Logical Register Mapping .....	2193
10-97. DISPC_VIDp_ATTRIBUTES Logical Register Mapping.....	2193
10-98. DISPC_VIDp_ATTRIBUTES2 Logical Register Mapping .....	2193
10-99. DISPC_VIDp_BUF_THRESHOLD Logical Register Mapping.....	2193
10-100. DISPC_VIDp_ROW_INC Logical Register Mapping .....	2193
10-101. DISPC_VIDp_PIXEL_INC Logical Register Mapping .....	2193
10-102. DISPC_VIDp_FIR Logical Register Mapping.....	2194
10-103. DISPC_VIDp_PICTURE_SIZE Logical Register Mapping.....	2194
10-104. DISPC_VIDp_ACCU_j Logical Register Mapping .....	2194
10-105. DISPC_VIDp_FIR_COEF_H_i Logical Register Mapping .....	2194
10-106. DISPC_VIDp_FIR_COEF_HV_i Logical Register Mapping .....	2194
10-107. DISPC_VIDp_FIR_COEF_V_i Logical Register Mapping .....	2194
10-108. DISPC_VIDp_FIR_COEF_H2_i Logical Register Mapping.....	2194
10-109. DISPC_VIDp_FIR_COEF_HV2_i Logical Register Mapping.....	2195
10-110. DISPC_VIDp_FIR_COEF_V2_i Logical Register Mapping.....	2195
10-111. DISPC_VIDp_CONV_COEF0 Logical Register Mapping.....	2195
10-112. DISPC_VIDp_CONV_COEF1 Logical Register Mapping.....	2195
10-113. DISPC_VIDp_CONV_COEF2 Logical Register Mapping.....	2195
10-114. DISPC_VIDp_CONV_COEF3 Logical Register Mapping.....	2195
10-115. DISPC_VIDp_CONV_COEF4 Logical Register Mapping.....	2196
10-116. DISPC_CONTROLo Logical Register Mapping .....	2196
10-117. DISPC_CONFIGo Logical Register Mapping .....	2196
10-118. DISPC_DEFAULT_COLORo Logical Register Mapping.....	2196
10-119. DISPC_TRANS_COLORo Logical Register Mapping .....	2196
10-120. DISPC_GAMMA_TABLEo Logical Register Mapping.....	2196

10-121. DISPC_TIMING_Ho Logical Register Mapping .....	2196
10-122. DISPC_TIMING_Vo Logical Register Mapping .....	2196
10-123. DISPC_POL_FREQo Logical Register Mapping.....	2196
10-124. DISPC_DIVISORo Logical Register Mapping .....	2197
10-125. DISPC_SIZE_LCDo Logical Register Mapping .....	2197
10-126. DISPC_SIZE Logical Register Mapping .....	2197
10-127. DISPC_DATAo_CYCLE1 Logical Register Mapping.....	2197
10-128. DISPC_DATAo_CYCLE2 Logical Register Mapping.....	2197
10-129. DISPC_DATAo_CYCLE3 Logical Register Mapping.....	2197
10-130. DISPC_CPRo_COEF_R Logical Register Mapping.....	2197
10-131. DISPC_CPRo_COEF_G Logical Register Mapping.....	2197
10-132. DISPC_CPRo_COEF_B Logical Register Mapping.....	2197
10-133. Display Controller Registers Mapping Summary.....	2198
10-134. DISPC_REVISION .....	2203
10-135. Register Call Summary for Register DISPC_REVISION .....	2203
10-136. DISPC_SYSCONFIG .....	2203
10-137. Register Call Summary for Register DISPC_SYSCONFIG .....	2204
10-138. DISPC_SYSSTATUS.....	2204
10-139. Register Call Summary for Register DISPC_SYSSTATUS .....	2205
10-140. DISPC_IRQSTATUS .....	2205
10-141. Register Call Summary for Register DISPC_IRQSTATUS .....	2207
10-142. DISPC_IRQENABLE .....	2208
10-143. Register Call Summary for Register DISPC_IRQENABLE .....	2210
10-144. DISPC_CONTROL1 .....	2210
10-145. Register Call Summary for Register DISPC_CONTROL1.....	2213
10-146. DISPC_CONFIG1 .....	2213
10-147. Register Call Summary for Register DISPC_CONFIG1 .....	2216
10-148. DISPC_DEFAULT_COLOR0 .....	2217
10-149. Register Call Summary for Register DISPC_DEFAULT_COLOR0 .....	2217
10-150. DISPC_DEFAULT_COLOR1 .....	2217
10-151. Register Call Summary for Register DISPC_DEFAULT_COLOR1 .....	2217
10-152. DISPC_TRANS_COLOR0 .....	2218
10-153. Register Call Summary for Register DISPC_TRANS_COLOR0.....	2218
10-154. DISPC_TRANS_COLOR1 .....	2218
10-155. Register Call Summary for Register DISPC_TRANS_COLOR1.....	2218
10-156. DISPC_LINE_STATUS.....	2219
10-157. Register Call Summary for Register DISPC_LINE_STATUS .....	2219
10-158. DISPC_LINE_NUMBER.....	2219
10-159. Register Call Summary for Register DISPC_LINE_NUMBER .....	2219
10-160. DISPC_TIMING_H1 .....	2220
10-161. Register Call Summary for Register DISPC_TIMING_H1 .....	2220
10-162. DISPC_TIMING_V1.....	2220
10-163. Register Call Summary for Register DISPC_TIMING_V1 .....	2221
10-164. DISPC_POL_FREQ1 .....	2221
10-165. Register Call Summary for Register DISPC_POL_FREQ1 .....	2222
10-166. DISPC_DIVISOR1 .....	2222
10-167. Register Call Summary for Register DISPC_DIVISOR1.....	2222
10-168. DISPC_GLOBAL_ALPHA.....	2222
10-169. Register Call Summary for Register DISPC_GLOBAL_ALPHA .....	2223



10-170. DISPC_SIZE_TV .....	2223
10-171. Register Call Summary for Register DISPC_SIZE_TV .....	2223
10-172. DISPC_SIZE_LCD1 .....	2224
10-173. Register Call Summary for Register DISPC_SIZE_LCD1 .....	2224
10-174. DISPC_GFX_BA_j .....	2224
10-175. Register Call Summary for Register DISPC_GFX_BA_j.....	2225
10-176. DISPC_GFX_POSITION .....	2225
10-177. Register Call Summary for Register DISPC_GFX_POSITION .....	2225
10-178. DISPC_GFX_SIZE .....	2226
10-179. Register Call Summary for Register DISPC_GFX_SIZE .....	2226
10-180. DISPC_GFX_ATTRIBUTES .....	2226
10-181. Register Call Summary for Register DISPC_GFX_ATTRIBUTES.....	2229
10-182. DISPC_GFX_BUF_THRESHOLD .....	2229
10-183. Register Call Summary for Register DISPC_GFX_BUF_THRESHOLD .....	2230
10-184. DISPC_GFX_BUF_SIZE_STATUS.....	2230
10-185. Register Call Summary for Register DISPC_GFX_BUF_SIZE_STATUS .....	2230
10-186. DISPC_GFX_ROW_INC .....	2230
10-187. Register Call Summary for Register DISPC_GFX_ROW_INC.....	2231
10-188. DISPC_GFX_PIXEL_INC .....	2231
10-189. Register Call Summary for Register DISPC_GFX_PIXEL_INC.....	2231
10-190. DISPC_GFX_WINDOW_SKIP .....	2231
10-191. Register Call Summary for Register DISPC_GFX_WINDOW_SKIP .....	2232
10-192. DISPC_GFX_TABLE_BA .....	2232
10-193. Register Call Summary for Register DISPC_GFX_TABLE_BA.....	2232
10-194. DISPC_VID1_BA_j.....	2232
10-195. Register Call Summary for Register DISPC_VID1_BA_j .....	2233
10-196. DISPC_VID1_POSITION .....	2233
10-197. Register Call Summary for Register DISPC_VID1_POSITION .....	2233
10-198. DISPC_VID1_SIZE .....	2234
10-199. Register Call Summary for Register DISPC_VID1_SIZE .....	2234
10-200. DISPC_VID1_ATTRIBUTES.....	2234
10-201. Register Call Summary for Register DISPC_VID1_ATTRIBUTES .....	2237
10-202. DISPC_VID1_BUF_THRESHOLD.....	2238
10-203. Register Call Summary for Register DISPC_VID1_BUF_THRESHOLD .....	2238
10-204. DISPC_VID1_BUF_SIZE_STATUS .....	2238
10-205. Register Call Summary for Register DISPC_VID1_BUF_SIZE_STATUS.....	2238
10-206. DISPC_VID1_ROW_INC.....	2239
10-207. Register Call Summary for Register DISPC_VID1_ROW_INC .....	2239
10-208. DISPC_VID1_PIXEL_INC.....	2239
10-209. Register Call Summary for Register DISPC_VID1_PIXEL_INC .....	2240
10-210. DISPC_VID1_FIR .....	2240
10-211. Register Call Summary for Register DISPC_VID1_FIR.....	2240
10-212. DISPC_VID1_PICTURE_SIZE .....	2240
10-213. Register Call Summary for Register DISPC_VID1_PICTURE_SIZE.....	2241
10-214. DISPC_VID1_ACCU_j.....	2241
10-215. Register Call Summary for Register DISPC_VID1_ACCU_j .....	2241
10-216. DISPC_VID1_FIR_COEF_H_i .....	2242
10-217. Register Call Summary for Register DISPC_VID1_FIR_COEF_H_i.....	2242
10-218. DISPC_VID1_FIR_COEF_HV_i.....	2242

10-219. Register Call Summary for Register DISPC_VID1_FIR_COEF_HV_i .....	2243
10-220. DISPC_VID1_CONV_COEF0 .....	2243
10-221. Register Call Summary for Register DISPC_VID1_CONV_COEF0.....	2243
10-222. DISPC_VID1_CONV_COEF1 .....	2243
10-223. Register Call Summary for Register DISPC_VID1_CONV_COEF1.....	2244
10-224. DISPC_VID1_CONV_COEF2 .....	2244
10-225. Register Call Summary for Register DISPC_VID1_CONV_COEF2.....	2244
10-226. DISPC_VID1_CONV_COEF3 .....	2244
10-227. Register Call Summary for Register DISPC_VID1_CONV_COEF3.....	2245
10-228. DISPC_VID1_CONV_COEF4 .....	2245
10-229. Register Call Summary for Register DISPC_VID1_CONV_COEF4.....	2245
10-230. DISPC_VID2_BA_j.....	2245
10-231. Register Call Summary for Register DISPC_VID2_BA_j .....	2246
10-232. DISPC_VID2_POSITION .....	2246
10-233. Register Call Summary for Register DISPC_VID2_POSITION .....	2246
10-234. DISPC_VID2_SIZE .....	2247
10-235. Register Call Summary for Register DISPC_VID2_SIZE .....	2247
10-236. DISPC_VID2_ATTRIBUTES.....	2247
10-237. Register Call Summary for Register DISPC_VID2_ATTRIBUTES .....	2250
10-238. DISPC_VID2_BUF_THRESHOLD.....	2251
10-239. Register Call Summary for Register DISPC_VID2_BUF_THRESHOLD .....	2251
10-240. DISPC_VID2_BUF_SIZE_STATUS .....	2251
10-241. Register Call Summary for Register DISPC_VID2_BUF_SIZE_STATUS.....	2251
10-242. DISPC_VID2_ROW_INC.....	2252
10-243. Register Call Summary for Register DISPC_VID2_ROW_INC .....	2252
10-244. DISPC_VID2_PIXEL_INC.....	2252
10-245. Register Call Summary for Register DISPC_VID2_PIXEL_INC .....	2253
10-246. DISPC_VID2_FIR.....	2253
10-247. Register Call Summary for Register DISPC_VID2_FIR.....	2253
10-248. DISPC_VID2_PICTURE_SIZE .....	2253
10-249. Register Call Summary for Register DISPC_VID2_PICTURE_SIZE.....	2254
10-250. DISPC_VID2_ACCU_j.....	2254
10-251. Register Call Summary for Register DISPC_VID2_ACCU_j .....	2254
10-252. DISPC_VID2_FIR_COEF_H_i .....	2255
10-253. Register Call Summary for Register DISPC_VID2_FIR_COEF_H_i.....	2255
10-254. DISPC_VID2_FIR_COEF_HV_i.....	2255
10-255. Register Call Summary for Register DISPC_VID2_FIR_COEF_HV_i .....	2256
10-256. DISPC_VID2_CONV_COEF0 .....	2256
10-257. Register Call Summary for Register DISPC_VID2_CONV_COEF0.....	2256
10-258. DISPC_VID2_CONV_COEF1 .....	2256
10-259. Register Call Summary for Register DISPC_VID2_CONV_COEF1.....	2257
10-260. DISPC_VID2_CONV_COEF2 .....	2257
10-261. Register Call Summary for Register DISPC_VID2_CONV_COEF2.....	2257
10-262. DISPC_VID2_CONV_COEF3 .....	2257
10-263. Register Call Summary for Register DISPC_VID2_CONV_COEF3.....	2258
10-264. DISPC_VID2_CONV_COEF4 .....	2258
10-265. Register Call Summary for Register DISPC_VID2_CONV_COEF4.....	2258
10-266. DISPC_DATA1_CYCLE1 .....	2258
10-267. Register Call Summary for Register DISPC_DATA1_CYCLE1.....	2259

10-268. DISPC_DATA1_CYCLE2 .....	2259
10-269. Register Call Summary for Register DISPC_DATA1_CYCLE2.....	2259
10-270. DISPC_DATA1_CYCLE3 .....	2260
10-271. Register Call Summary for Register DISPC_DATA1_CYCLE3.....	2260
10-272. DISPC_VID1_FIR_COEF_V_i.....	2260
10-273. Register Call Summary for Register DISPC_VID1_FIR_COEF_V_i .....	2261
10-274. DISPC_VID2_FIR_COEF_V_i.....	2261
10-275. Register Call Summary for Register DISPC_VID2_FIR_COEF_V_i .....	2261
10-276. DISPC_CPR1_COEF_R .....	2261
10-277. Register Call Summary for Register DISPC_CPR1_COEF_R.....	2262
10-278. DISPC_CPR1_COEF_G .....	2262
10-279. Register Call Summary for Register DISPC_CPR1_COEF_G.....	2262
10-280. DISPC_CPR1_COEF_B .....	2262
10-281. Register Call Summary for Register DISPC_CPR1_COEF_B.....	2263
10-282. DISPC_GFX_PRELOAD .....	2263
10-283. Register Call Summary for Register DISPC_GFX_PRELOAD .....	2263
10-284. DISPC_VID1_PRELOAD .....	2264
10-285. Register Call Summary for Register DISPC_VID1_PRELOAD .....	2264
10-286. DISPC_VID2_PRELOAD .....	2264
10-287. Register Call Summary for Register DISPC_VID2_PRELOAD .....	2264
10-288. DISPC_CONTROL2 .....	2265
10-289. Register Call Summary for Register DISPC_CONTROL2.....	2267
10-290. DISPC_VID3_ACCU_j.....	2267
10-291. Register Call Summary for Register DISPC_VID3_ACCU_j.....	2268
10-292. DISPC_VID3_BA_j.....	2268
10-293. Register Call Summary for Register DISPC_VID3_BA_j .....	2268
10-294. DISPC_VID3_FIR_COEF_H_i .....	2268
10-295. Register Call Summary for Register DISPC_VID3_FIR_COEF_H_i.....	2269
10-296. DISPC_VID3_FIR_COEF_HV_i.....	2269
10-297. Register Call Summary for Register DISPC_VID3_FIR_COEF_HV_j .....	2269
10-298. DISPC_VID3_FIR_COEF_V_i.....	2270
10-299. Register Call Summary for Register DISPC_VID3_FIR_COEF_V_i .....	2270
10-300. DISPC_VID3_ATTRIBUTES.....	2270
10-301. Register Call Summary for Register DISPC_VID3_ATTRIBUTES .....	2273
10-302. DISPC_VID3_CONV_COEF0 .....	2273
10-303. Register Call Summary for Register DISPC_VID3_CONV_COEF0.....	2274
10-304. DISPC_VID3_CONV_COEF1 .....	2274
10-305. Register Call Summary for Register DISPC_VID3_CONV_COEF1.....	2274
10-306. DISPC_VID3_CONV_COEF2 .....	2274
10-307. Register Call Summary for Register DISPC_VID3_CONV_COEF2.....	2275
10-308. DISPC_VID3_CONV_COEF3 .....	2275
10-309. Register Call Summary for Register DISPC_VID3_CONV_COEF3.....	2275
10-310. DISPC_VID3_CONV_COEF4 .....	2275
10-311. Register Call Summary for Register DISPC_VID3_CONV_COEF4.....	2276
10-312. DISPC_VID3_BUF_SIZE_STATUS .....	2276
10-313. Register Call Summary for Register DISPC_VID3_BUF_SIZE_STATUS.....	2276
10-314. DISPC_VID3_BUF_THRESHOLD .....	2276
10-315. Register Call Summary for Register DISPC_VID3_BUF_THRESHOLD .....	2276
10-316. DISPC_VID3_FIR .....	2277

10-317. Register Call Summary for Register DISPC_VID3_FIR.....	2277
10-318. DISPC_VID3_PICTURE_SIZE .....	2277
10-319. Register Call Summary for Register DISPC_VID3_PICTURE_SIZE.....	2278
10-320. DISPC_VID3_PIXEL_INC.....	2278
10-321. Register Call Summary for Register DISPC_VID3_PIXEL_INC .....	2278
10-322. DISPC_VID3_POSITION .....	2279
10-323. Register Call Summary for Register DISPC_VID3_POSITION .....	2279
10-324. DISPC_VID3_PRELOAD .....	2279
10-325. Register Call Summary for Register DISPC_VID3_PRELOAD .....	2279
10-326. DISPC_VID3_ROW_INC.....	2280
10-327. Register Call Summary for Register DISPC_VID3_ROW_INC .....	2280
10-328. DISPC_VID3_SIZE .....	2280
10-329. Register Call Summary for Register DISPC_VID3_SIZE .....	2280
10-330. DISPC_DEFAULT_COLOR2 .....	2281
10-331. Register Call Summary for Register DISPC_DEFAULT_COLOR2 .....	2281
10-332. DISPC_TRANS_COLOR2 .....	2281
10-333. Register Call Summary for Register DISPC_TRANS_COLOR2.....	2281
10-334. DISPC_CPR2_COEF_B .....	2282
10-335. Register Call Summary for Register DISPC_CPR2_COEF_B.....	2282
10-336. DISPC_CPR2_COEF_G .....	2282
10-337. Register Call Summary for Register DISPC_CPR2_COEF_G.....	2283
10-338. DISPC_CPR2_COEF_R .....	2283
10-339. Register Call Summary for Register DISPC_CPR2_COEF_R.....	2283
10-340. DISPC_DATA2_CYCLE1 .....	2283
10-341. Register Call Summary for Register DISPC_DATA2_CYCLE1.....	2284
10-342. DISPC_DATA2_CYCLE2 .....	2284
10-343. Register Call Summary for Register DISPC_DATA2_CYCLE2.....	2285
10-344. DISPC_DATA2_CYCLE3 .....	2285
10-345. Register Call Summary for Register DISPC_DATA2_CYCLE3.....	2285
10-346. DISPC_SIZE_LCD2 .....	2286
10-347. Register Call Summary for Register DISPC_SIZE_LCD2.....	2286
10-348. DISPC_TIMING_H2 .....	2286
10-349. Register Call Summary for Register DISPC_TIMING_H2.....	2287
10-350. DISPC_TIMING_V2.....	2287
10-351. Register Call Summary for Register DISPC_TIMING_V2 .....	2287
10-352. DISPC_POL_FREQ2 .....	2287
10-353. Register Call Summary for Register DISPC_POL_FREQ2 .....	2288
10-354. DISPC_DIVISOR2 .....	2288
10-355. Register Call Summary for Register DISPC_DIVISOR2.....	2289
10-356. DISPC_WB_ACCU_j .....	2289
10-357. Register Call Summary for Register DISPC_WB_ACCU_j.....	2289
10-358. DISPC_WB_BA_j .....	2290
10-359. Register Call Summary for Register DISPC_WB_BA_j.....	2290
10-360. DISPC_WB_FIR_COEF_H_j .....	2290
10-361. Register Call Summary for Register DISPC_WB_FIR_COEF_H_j.....	2291
10-362. DISPC_WB_FIR_COEF_HV_j .....	2291
10-363. Register Call Summary for Register DISPC_WB_FIR_COEF_HV_j.....	2291
10-364. DISPC_WB_FIR_COEF_V_j .....	2292
10-365. Register Call Summary for Register DISPC_WB_FIR_COEF_V_j.....	2292

10-366. DISPC_WB_ATTRIBUTES .....	2292
10-367. Register Call Summary for Register DISPC_WB_ATTRIBUTES.....	2295
10-368. DISPC_WB_CONV_COEF0.....	2296
10-369. Register Call Summary for Register DISPC_WB_CONV_COEF0 .....	2296
10-370. DISPC_WB_CONV_COEF1.....	2296
10-371. Register Call Summary for Register DISPC_WB_CONV_COEF1 .....	2296
10-372. DISPC_WB_CONV_COEF2.....	2297
10-373. Register Call Summary for Register DISPC_WB_CONV_COEF2 .....	2297
10-374. DISPC_WB_CONV_COEF3.....	2297
10-375. Register Call Summary for Register DISPC_WB_CONV_COEF3 .....	2297
10-376. DISPC_WB_CONV_COEF4.....	2298
10-377. Register Call Summary for Register DISPC_WB_CONV_COEF4 .....	2298
10-378. DISPC_WB_BUF_SIZE_STATUS .....	2298
10-379. Register Call Summary for Register DISPC_WB_BUF_SIZE_STATUS .....	2298
10-380. DISPC_WB_BUF_THRESHOLD .....	2299
10-381. Register Call Summary for Register DISPC_WB_BUF_THRESHOLD .....	2299
10-382. DISPC_WB_FIR.....	2299
10-383. Register Call Summary for Register DISPC_WB_FIR .....	2300
10-384. DISPC_WB_PICTURE_SIZE.....	2300
10-385. Register Call Summary for Register DISPC_WB_PICTURE_SIZE .....	2300
10-386. DISPC_WB_PIXEL_INC .....	2301
10-387. Register Call Summary for Register DISPC_WB_PIXEL_INC.....	2301
10-388. DISPC_WB_ROW_INC .....	2301
10-389. Register Call Summary for Register DISPC_WB_ROW_INC.....	2302
10-390. DISPC_WB_SIZE .....	2302
10-391. Register Call Summary for Register DISPC_WB_SIZE.....	2302
10-392. DISPC_VID1_BA_UV_j .....	2303
10-393. Register Call Summary for Register DISPC_VID1_BA_UV_j.....	2303
10-394. DISPC_VID2_BA_UV_j .....	2303
10-395. Register Call Summary for Register DISPC_VID2_BA_UV_j.....	2304
10-396. DISPC_VID3_BA_UV_j .....	2304
10-397. Register Call Summary for Register DISPC_VID3_BA_UV_j.....	2304
10-398. DISPC_WB_BA_UV_j.....	2305
10-399. Register Call Summary for Register DISPC_WB_BA_UV_j.....	2305
10-400. DISPC_CONFIG2.....	2305
10-401. Register Call Summary for Register DISPC_CONFIG2 .....	2307
10-402. DISPC_VID1_ATTRIBUTES2 .....	2307
10-403. Register Call Summary for Register DISPC_VID1_ATTRIBUTES2.....	2308
10-404. DISPC_VID2_ATTRIBUTES2 .....	2308
10-405. Register Call Summary for Register DISPC_VID2_ATTRIBUTES2.....	2309
10-406. DISPC_VID3_ATTRIBUTES2 .....	2309
10-407. Register Call Summary for Register DISPC_VID3_ATTRIBUTES2.....	2310
10-408. DISPC_GAMMA_TABLE0 .....	2310
10-409. Register Call Summary for Register DISPC_GAMMA_TABLE0.....	2310
10-410. DISPC_GAMMA_TABLE1 .....	2311
10-411. Register Call Summary for Register DISPC_GAMMA_TABLE1.....	2311
10-412. DISPC_GAMMA_TABLE2 .....	2311
10-413. Register Call Summary for Register DISPC_GAMMA_TABLE2.....	2312
10-414. DISPC_VID1_FIR2 .....	2312

10-415. Register Call Summary for Register DISPC_VID1_FIR2 .....	2312
10-416. DISPC_VID1_ACCU2_j .....	2313
10-417. Register Call Summary for Register DISPC_VID1_ACCU2_j .....	2313
10-418. DISPC_VID1_FIR_COEF_H2_i .....	2313
10-419. Register Call Summary for Register DISPC_VID1_FIR_COEF_H2_i .....	2314
10-420. DISPC_VID1_FIR_COEF_HV2_i .....	2314
10-421. Register Call Summary for Register DISPC_VID1_FIR_COEF_HV2_i .....	2314
10-422. DISPC_VID1_FIR_COEF_V2_i .....	2315
10-423. Register Call Summary for Register DISPC_VID1_FIR_COEF_V2_i .....	2315
10-424. DISPC_VID2_FIR2 .....	2315
10-425. Register Call Summary for Register DISPC_VID2_FIR2 .....	2316
10-426. DISPC_VID2_ACCU2_j .....	2316
10-427. Register Call Summary for Register DISPC_VID2_ACCU2_j .....	2316
10-428. DISPC_VID2_FIR_COEF_H2_i .....	2317
10-429. Register Call Summary for Register DISPC_VID2_FIR_COEF_H2_i .....	2317
10-430. DISPC_VID2_FIR_COEF_HV2_i .....	2317
10-431. Register Call Summary for Register DISPC_VID2_FIR_COEF_HV2_i .....	2318
10-432. DISPC_VID2_FIR_COEF_V2_i .....	2318
10-433. Register Call Summary for Register DISPC_VID2_FIR_COEF_V2_i .....	2318
10-434. DISPC_VID3_FIR2 .....	2318
10-435. Register Call Summary for Register DISPC_VID3_FIR2 .....	2319
10-436. DISPC_VID3_ACCU2_j .....	2319
10-437. Register Call Summary for Register DISPC_VID3_ACCU2_j .....	2319
10-438. DISPC_VID3_FIR_COEF_H2_i .....	2320
10-439. Register Call Summary for Register DISPC_VID3_FIR_COEF_H2_i .....	2320
10-440. DISPC_VID3_FIR_COEF_HV2_i .....	2320
10-441. Register Call Summary for Register DISPC_VID3_FIR_COEF_HV2_i .....	2321
10-442. DISPC_VID3_FIR_COEF_V2_i .....	2321
10-443. Register Call Summary for Register DISPC_VID3_FIR_COEF_V2_i .....	2321
10-444. DISPC_WB_FIR2 .....	2321
10-445. Register Call Summary for Register DISPC_WB_FIR2 .....	2322
10-446. DISPC_WB_ACCU2_j .....	2322
10-447. Register Call Summary for Register DISPC_WB_ACCU2_j .....	2322
10-448. DISPC_WB_FIR_COEF_H2_i .....	2323
10-449. Register Call Summary for Register DISPC_WB_FIR_COEF_H2_i .....	2323
10-450. DISPC_WB_FIR_COEF_HV2_i .....	2323
10-451. Register Call Summary for Register DISPC_WB_FIR_COEF_HV2_i .....	2324
10-452. DISPC_WB_FIR_COEF_V2_i .....	2324
10-453. Register Call Summary for Register DISPC_WB_FIR_COEF_V2_i .....	2324
10-454. DISPC_GLOBAL_BUFFER .....	2324
10-455. Register Call Summary for Register DISPC_GLOBAL_BUFFER .....	2326
10-456. DISPC_DIVISOR .....	2327
10-457. Register Call Summary for Register DISPC_DIVISOR .....	2327
10-458. DISPC_WB_ATTRIBUTES2 .....	2327
10-459. Register Call Summary for Register DISPC_WB_ATTRIBUTES2 .....	2328
10-460. I/O Description of DSI1 .....	2333
10-461. I/O Description of DSI2 .....	2334
10-462. DSI Lane Configuration .....	2335
10-463. Synchronization Codes .....	2340

10-464. Sync Short Packet Values .....	2340
10-465. VC Values.....	2347
10-466. Pixel Data Format in Video Mode.....	2351
10-467. Integration Attributes.....	2360
10-468. Clocks and Resets .....	2360
10-469. Interrupt Requests .....	2360
10-470. DMA Requests .....	2360
10-471. DSI Multiplexer Controls .....	2363
10-472. DSI Global Interrupts .....	2364
10-473. Register Access Width Limitations.....	2366
10-474. VC TX FIFO Size Values.....	2369
10-475. VC TX FIFO Start Address .....	2370
10-476. VC RX FIFO Size Values .....	2371
10-477. VC RX FIFO Start Address .....	2371
10-478. LP to HS Timing Parameters .....	2379
10-479. HS to LP Timing Parameters .....	2380
10-480. Extra NULL PH .....	2381
10-481. Extra NULL Packet Payload .....	2382
10-482. Video Port Signals for DSI1 and DSI2 Protocol Engines .....	2382
10-483. Video Port Signals for DSI1 Protocol Engine.....	2382
10-484. Video Port 1 for DSI1 and DSI2 in the Context of Video Mode.....	2383
10-485. Video Port 2 for DSI1 in the Context of Video Mode .....	2383
10-486. Video Port 1 for DSI1 and DSI2 in the Context of Command Mode .....	2388
10-487. Video Port 2 for DSI1 in the Context of Command Mode .....	2388
10-488. DSI VC Interrupts .....	2408
10-489. DSI PLL Operation Modes When Not Locked.....	2410
10-490. Register Call Summary for Gated Mode Sequence .....	2413
10-491. Register Call Summary for DSI PLL Programming Sequence .....	2415
10-492. Register Call Summary for DSI PLL Go Sequence (Automatic Mode) .....	2417
10-493. Recommended Programming Values .....	2417
10-494. DSI Complex I/O Interrupts .....	2426
10-495. Global Initialization of Surrounding Modules .....	2428
10-496. DSI Global Initialization.....	2428
10-497. DSI PLL Configuration Registers .....	2429
10-498. DSI Control Registers .....	2429
10-499. DSI Complex I/O Registers .....	2430
10-500. DSI Timing Registers .....	2430
10-501. Calculate DSI_PHY Timing .....	2431
10-502. Drive STOP State .....	2431
10-503. Enable Video Mode .....	2432
10-504. Configure DSI PLL .....	2432
10-505. Switch to DSI PLL Clock Source.....	2433
10-506. DSI Control Registers .....	2433
10-507. DSI Complex I/O Registers .....	2433
10-508. DSI Timing Registers .....	2434
10-509. Configure DSI_PHY Timing.....	2435
10-510. Drive STOP State .....	2435
10-511. Enable Command Mode and Automatic TE .....	2435
10-512. Send Frame Data to LCD Panel Using Automatic TE .....	2436

10-513. DSI Instance Summary .....	2437
10-514. DSI1_PROTOCOL_ENGINE Registers Mapping Summary .....	2437
10-515. DSI2_PROTOCOL_ENGINE Registers Mapping Summary .....	2438
10-516. DSI_REVISION.....	2440
10-517. Register Call Summary for Register DSI_REVISION .....	2440
10-518. DSI_SYSCONFIG.....	2440
10-519. Register Call Summary for Register DSI_SYSCONFIG .....	2441
10-520. DSI_SYSSTATUS.....	2441
10-521. Register Call Summary for Register DSI_SYSSTATUS .....	2442
10-522. DSI_IRQSTATUS .....	2442
10-523. Register Call Summary for Register DSI_IRQSTATUS.....	2444
10-524. DSI_IRQENABLE .....	2445
10-525. Register Call Summary for Register DSI_IRQENABLE.....	2446
10-526. DSI_CTRL.....	2446
10-527. Register Call Summary for Register DSI_CTRL .....	2450
10-528. DSI_GNQ .....	2450
10-529. Register Call Summary for Register DSI_GNQ .....	2452
10-530. DSI_COMPLEXIO_CFG1 .....	2452
10-531. Register Call Summary for Register DSI_COMPLEXIO_CFG1 .....	2455
10-532. DSI_COMPLEXIO_IRQSTATUS.....	2455
10-533. Register Call Summary for Register DSI_COMPLEXIO_IRQSTATUS .....	2459
10-534. DSI_COMPLEXIO_IRQENABLE.....	2459
10-535. Register Call Summary for Register DSI_COMPLEXIO_IRQENABLE .....	2462
10-536. DSI_CLK_CTRL.....	2462
10-537. Register Call Summary for Register DSI_CLK_CTRL .....	2464
10-538. DSI_TIMING1.....	2464
10-539. Register Call Summary for Register DSI_TIMING1 .....	2465
10-540. DSI_TIMING2.....	2465
10-541. Register Call Summary for Register DSI_TIMING2 .....	2466
10-542. DSI_VM_TIMING1 .....	2467
10-543. Register Call Summary for Register DSI_VM_TIMING1.....	2467
10-544. DSI_VM_TIMING2 .....	2467
10-545. Register Call Summary for Register DSI_VM_TIMING2.....	2468
10-546. DSI_VM_TIMING3 .....	2468
10-547. Register Call Summary for Register DSI_VM_TIMING3.....	2468
10-548. DSI_CLK_TIMING .....	2469
10-549. Register Call Summary for Register DSI_CLK_TIMING .....	2469
10-550. DSI_TX_FIFO_VC_SIZE.....	2469
10-551. Register Call Summary for Register DSI_TX_FIFO_VC_SIZE .....	2470
10-552. DSI_RX_FIFO_VC_SIZE .....	2470
10-553. Register Call Summary for Register DSI_RX_FIFO_VC_SIZE .....	2471
10-554. DSI_COMPLEXIO_CFG2.....	2471
10-555. Register Call Summary for Register DSI_COMPLEXIO_CFG2 .....	2475
10-556. DSI_RX_FIFO_VC_FULLNESS .....	2475
10-557. Register Call Summary for Register DSI_RX_FIFO_VC_FULLNESS .....	2475
10-558. DSI_VM_TIMING4 .....	2475
10-559. Register Call Summary for Register DSI_VM_TIMING4.....	2476
10-560. DSI_TX_FIFO_VC_EMPTYNESS .....	2476
10-561. Register Call Summary for Register DSI_TX_FIFO_VC_EMPTYNESS.....	2476



10-562. DSI_VM_TIMING5 .....	2477
10-563. Register Call Summary for Register DSI_VM_TIMING5.....	2477
10-564. DSI_VM_TIMING6 .....	2477
10-565. Register Call Summary for Register DSI_VM_TIMING6.....	2478
10-566. DSI_VM_TIMING7 .....	2478
10-567. Register Call Summary for Register DSI_VM_TIMING7.....	2478
10-568. DSI_STOPCLK_TIMING .....	2478
10-569. Register Call Summary for Register DSI_STOPCLK_TIMING.....	2479
10-570. DSI_CTRL2 .....	2479
10-571. Register Call Summary for Register DSI_CTRL2.....	2480
10-572. DSI_VM_TIMING8 .....	2480
10-573. Register Call Summary for Register DSI_VM_TIMING8.....	2480
10-574. DSI_TE_HSYNC_WIDTH_j.....	2481
10-575. Register Call Summary for Register DSI_TE_HSYNC_WIDTH_j .....	2481
10-576. DSI_TE_VSYNC_WIDTH_j.....	2481
10-577. Register Call Summary for Register DSI_TE_VSYNC_WIDTH_j .....	2481
10-578. DSI_TE_HSYNC_NUMBER_j.....	2482
10-579. Register Call Summary for Register DSI_TE_HSYNC_NUMBER_j .....	2482
10-580. DSI_VC_CTRL_i .....	2482
10-581. Register Call Summary for Register DSI_VC_CTRL_i.....	2486
10-582. DSI_VC_TE_i.....	2486
10-583. Register Call Summary for Register DSI_VC_TE_i .....	2487
10-584. DSI_VC_LONG_PACKET_HEADER_i.....	2488
10-585. Register Call Summary for Register DSI_VC_LONG_PACKET_HEADER_i .....	2488
10-586. DSI_VC_LONG_PACKET_PAYLOAD_i .....	2489
10-587. Register Call Summary for Register DSI_VC_LONG_PACKET_PAYLOAD_i.....	2489
10-588. DSI_VC_SHORT_PACKET_HEADER_i.....	2489
10-589. Register Call Summary for Register DSI_VC_SHORT_PACKET_HEADER_i .....	2490
10-590. DSI_VC_IRQSTATUS_i.....	2490
10-591. Register Call Summary for Register DSI_VC_IRQSTATUS_i .....	2491
10-592. DSI_VC_IRQENABLE_i.....	2492
10-593. Register Call Summary for Register DSI_VC_IRQENABLE_i .....	2493
10-594. DSI1_PHY Registers Mapping Summary.....	2493
10-595. DSI2_PHY Registers Mapping Summary.....	2493
10-596. DSI_PHY_REGISTER0 .....	2494
10-597. Register Call Summary for Register DSI_PHY_REGISTER0.....	2495
10-598. DSI_PHY_REGISTER1 .....	2495
10-599. Register Call Summary for Register DSI_PHY_REGISTER1 .....	2497
10-600. DSI_PHY_REGISTER2 .....	2497
10-601. Register Call Summary for Register DSI_PHY_REGISTER2.....	2497
10-602. DSI_PHY_REGISTER3 .....	2498
10-603. Register Call Summary for Register DSI_PHY_REGISTER3.....	2498
10-604. DSI_PHY_REGISTER4 .....	2498
10-605. Register Call Summary for Register DSI_PHY_REGISTER4.....	2498
10-606. DSI_PHY_REGISTER5 .....	2499
10-607. Register Call Summary for Register DSI_PHY_REGISTER5.....	2499
10-608. DSI1_PLLCTRL Registers Mapping Summary.....	2500
10-609. DSI2_PLLCTRL Registers Mapping Summary.....	2500
10-610. DSI_PLL_CONTROL .....	2501

10-611. Register Call Summary for Register DSI_PLL_CONTROL.....	2501
10-612. DSI_PLL_STATUS.....	2502
10-613. Register Call Summary for Register DSI_PLL_STATUS .....	2503
10-614. DSI_PLL_GO .....	2503
10-615. Register Call Summary for Register DSI_PLL_GO.....	2504
10-616. DSI_PLL_CONFIGURATION1 .....	2504
10-617. Register Call Summary for Register DSI_PLL_CONFIGURATION1.....	2504
10-618. DSI_PLL_CONFIGURATION2 .....	2505
10-619. Register Call Summary for Register DSI_PLL_CONFIGURATION2.....	2506
10-620. DSI_PLL_CONFIGURATION3 .....	2507
10-621. Register Call Summary for Register DSI_PLL_CONFIGURATION3.....	2507
10-622. DSI_PLL_SSC_CONFIGURATION1 .....	2507
10-623. Register Call Summary for Register DSI_PLL_SSC_CONFIGURATION1.....	2507
10-624. DSI_PLL_SSC_CONFIGURATION2 .....	2508
10-625. Register Call Summary for Register DSI_PLL_SSC_CONFIGURATION2.....	2508
10-626. DSI_PLL_CONFIGURATION4 .....	2508
10-627. Register Call Summary for Register DSI_PLL_CONFIGURATION4.....	2508
10-628. HDMI Video Timings (CEA-861-D) .....	2510
10-629. HDMI Video timings (VESA DMT) .....	2511
10-630. I/O Pad Mode Selection .....	2514
10-631. LCD Interface Signals .....	2514
10-632. Programmable Timing Fields in RFBI Mode.....	2516
10-633. Integration Attributes.....	2519
10-634. Clocks and Resets .....	2520
10-635. Hardware Requests .....	2520
10-636. RFBI DMA Request Description .....	2520
10-637. Local Power Management Features.....	2522
10-638. DMA Requests Events .....	2523
10-639. Read/Write Function Description.....	2530
10-640. Minimum Cycle Time for CS0/WE Always Asserted .....	2530
10-641. RFBI Timings Configuration .....	2533
10-642. Minimum Pulse Width (HSYNC/VSYNC) .....	2534
10-643. Global Initialization of Surrounding Modules .....	2535
10-644. RFBI Global Initialization .....	2536
10-645. DISPC Configuration in RFBI Mode .....	2536
10-646. RFBI Configuration – Main Sequence.....	2536
10-647. Register Call Summary for Subsequence: DMA Mode Configuration .....	2537
10-648. Timing and Polarity Signals Configuration .....	2538
10-649. Register Call Summary for Subsequence: Trigger Mode Settings .....	2539
10-650. RFBI Command Interface: Writing Command to the LCD Panel .....	2539
10-651. RFBI Command Interface: Reading Command From the LCD Panel .....	2539
10-652. RFBI Start Transfer With Polling Method .....	2539
10-653. RFBI Start Transfer With Interrupts Method .....	2539
10-654. DISPC Interrupts Enable .....	2540
10-655. DISPC Output Enable .....	2540
10-656. Register Call Summary for Subsequence: RFBI Output Enable.....	2541
10-657. RFBI Behavior .....	2541
10-658. Detect Event .....	2542
10-659. Register Call Summary for RFBI Events Servicing: FRAMEDONE Interrupt Servicing .....	2543

10-660. RFBI Hardware Status Features .....	2544
10-661. RFBI Instance Summary .....	2545
10-662. RFBI Registers Mapping Summary .....	2545
10-663. RFBI_REVISION .....	2546
10-664. Register Call Summary for Register RFBI_REVISION.....	2546
10-665. RFBI_SYSCONFIG .....	2546
10-666. Register Call Summary for Register RFBI_SYSCONFIG.....	2547
10-667. RFBI_SYSSTATUS .....	2547
10-668. Register Call Summary for Register RFBI_SYSSTATUS .....	2548
10-669. RFBI_CONTROL.....	2548
10-670. Register Call Summary for Register RFBI_CONTROL .....	2549
10-671. RFBI_PIXEL_CNT .....	2549
10-672. Register Call Summary for Register RFBI_PIXEL_CNT.....	2549
10-673. RFBI_LINE_NUMBER.....	2550
10-674. Register Call Summary for Register RFBI_LINE_NUMBER .....	2550
10-675. RFBI_CMD.....	2550
10-676. Register Call Summary for Register RFBI_CMD .....	2550
10-677. RFBI_PARAM .....	2551
10-678. Register Call Summary for Register RFBI_PARAM.....	2551
10-679. RFBI_DATA.....	2551
10-680. Register Call Summary for Register RFBI_DATA .....	2551
10-681. RFBI_READ .....	2552
10-682. Register Call Summary for Register RFBI_READ .....	2552
10-683. RFBI_STATUS .....	2552
10-684. Register Call Summary for Register RFBI_STATUS.....	2552
10-685. RFBI_CONFIG .....	2553
10-686. Register Call Summary for Register RFBI_CONFIG .....	2554
10-687. RFBI_ONOFF_TIME.....	2554
10-688. Register Call Summary for Register RFBI_ONOFF_TIME .....	2555
10-689. RFBI_CYCLE_TIME .....	2555
10-690. Register Call Summary for Register RFBI_CYCLE_TIME.....	2556
10-691. RFBI_DATA_CYCLEi.....	2556
10-692. Register Call Summary for Register RFBI_DATA_CYCLEi .....	2556
10-693. RFBI_VSYNC_WIDTH .....	2557
10-694. Register Call Summary for Register RFBI_VSYNC_WIDTH.....	2557
10-695. RFBI_HSYNC_WIDTH .....	2557
10-696. Register Call Summary for Register RFBI_HSYNC_WIDTH.....	2557
10-697. TV Display Interface Pins .....	2561
10-698. Typical Values for Rout, Rset, Rload, and Cout .....	2562
10-699. Clocks .....	2563
10-700. Digital Clock Division for the Video Encoder .....	2564
10-701. 100/100 Color Bar Table .....	2565
10-702. VENC_S_CARR Register Recommended Values.....	2566
10-703. Closed-Caption Data Format .....	2567
10-704. Closed-Caption RunClock Frequency Settings .....	2568
10-705. Closed-Caption Standard Timing Values .....	2568
10-706. Wide-Screen Signaling RunClock Frequency Settings .....	2569
10-707. Video DAC Power-Management Configurations .....	2576
10-708. Global Initialization of Surrounding Modules .....	2577

10-709. Start Video Encoder .....	2577
10-710. Start Video Encoder .....	2577
10-711. Reset Video Encoder .....	2578
10-712. TV Connection Detection Subsequence .....	2578
10-713. TV Disconnection Detection Subsequence .....	2578
10-714. Enable Video Composite Signal Output.....	2579
10-715. Disable Video Composite Signal Output .....	2579
10-716. Video Encoder Register Programming Values .....	2579
10-717. Video Encoder Instance Summary .....	2581
10-718. Video Encoder Registers Mapping Summary .....	2581
10-719. VENC_REV_ID .....	2583
10-720. Register Call Summary for Register VENC_REV_ID .....	2583
10-721. VENC_STATUS .....	2583
10-722. Register Call Summary for Register VENC_STATUS .....	2583
10-723. VENC_F_CONTROL .....	2584
10-724. Register Call Summary for Register VENC_F_CONTROL.....	2584
10-725. VENC_VIDOUT_CTRL.....	2585
10-726. Register Call Summary for Register VENC_VIDOUT_CTRL .....	2585
10-727. VENC_SYNC_CTRL.....	2585
10-728. Register Call Summary for Register VENC_SYNC_CTRL .....	2586
10-729. VENC_LLEN .....	2587
10-730. Register Call Summary for Register VENC_LLEN .....	2587
10-731. VENC_FLENS .....	2587
10-732. Register Call Summary for Register VENC_FLENS .....	2587
10-733. VENC_HFLTR_CTRL .....	2588
10-734. Register Call Summary for Register VENC_HFLTR_CTRL .....	2588
10-735. VENC_CC_CARR_WSS_CARR .....	2588
10-736. Register Call Summary for Register VENC_CC_CARR_WSS_CARR .....	2589
10-737. VENC_C_PHASE .....	2589
10-738. Register Call Summary for Register VENC_C_PHASE.....	2589
10-739. VENC_GAIN_U.....	2589
10-740. Register Call Summary for Register VENC_GAIN_U .....	2589
10-741. VENC_GAIN_V.....	2590
10-742. Register Call Summary for Register VENC_GAIN_V .....	2590
10-743. VENC_GAIN_Y .....	2590
10-744. Register Call Summary for Register VENC_GAIN_Y .....	2590
10-745. VENC_BLACK_LEVEL.....	2591
10-746. Register Call Summary for Register VENC_BLACK_LEVEL .....	2591
10-747. VENC_BLANK_LEVEL.....	2591
10-748. Register Call Summary for Register VENC_BLANK_LEVEL .....	2591
10-749. VENC_X_COLOR.....	2592
10-750. Register Call Summary for Register VENC_X_COLOR .....	2592
10-751. VENC_M_CONTROL.....	2593
10-752. Register Call Summary for Register VENC_M_CONTROL .....	2593
10-753. VENC_BSTAMP_WSS_DATA .....	2594
10-754. Register Call Summary for Register VENC_BSTAMP_WSS_DATA.....	2594
10-755. VENC_S_CARR.....	2594
10-756. Register Call Summary for Register VENC_S_CARR .....	2595
10-757. VENC_LINE21.....	2595

10-758. Register Call Summary for Register VENC_LINE21 .....	2595
10-759. VENC_LN_SEL.....	2595
10-760. Register Call Summary for Register VENC_LN_SEL .....	2596
10-761. VENC_L21_WC_CTL .....	2596
10-762. Register Call Summary for Register VENC_L21_WC_CTL .....	2597
10-763. VENC_HTRIGGER_VTRIGGER.....	2597
10-764. Register Call Summary for Register VENC_HTRIGGER_VTRIGGER .....	2597
10-765. VENC_SAVID_EAVID .....	2597
10-766. Register Call Summary for Register VENC_SAVID_EAVID .....	2598
10-767. VENC_FLEN_FAL .....	2598
10-768. Register Call Summary for Register VENC_FLEN_FAL.....	2598
10-769. VENC_LAL_PHASE_RESET.....	2598
10-770. Register Call Summary for Register VENC_LAL_PHASE_RESET .....	2599
10-771. VENC_HS_INT_START_STOP_X .....	2599
10-772. Register Call Summary for Register VENC_HS_INT_START_STOP_X.....	2599
10-773. VENC_HS_EXT_START_STOP_X.....	2599
10-774. Register Call Summary for Register VENC_HS_EXT_START_STOP_X.....	2600
10-775. VENC_VS_INT_START_X .....	2600
10-776. Register Call Summary for Register VENC_VS_INT_START_X.....	2600
10-777. VENC_VS_INT_STOP_X_VS_INT_START_Y .....	2600
10-778. Register Call Summary for Register VENC_VS_INT_STOP_X_VS_INT_START_Y.....	2600
10-779. VENC_VS_INT_STOP_Y_VS_EXT_START_X.....	2601
10-780. Register Call Summary for Register VENC_VS_INT_STOP_Y_VS_EXT_START_X.....	2601
10-781. VENC_VS_EXT_STOP_X_VS_EXT_START_Y.....	2601
10-782. Register Call Summary for Register VENC_VS_EXT_STOP_X_VS_EXT_START_Y.....	2601
10-783. VENC_VS_EXT_STOP_Y .....	2602
10-784. Register Call Summary for Register VENC_VS_EXT_STOP_Y.....	2602
10-785. VENC_AVID_START_STOP_X.....	2602
10-786. Register Call Summary for Register VENC_AVID_START_STOP_X .....	2602
10-787. VENC_AVID_START_STOP_Y.....	2603
10-788. Register Call Summary for Register VENC_AVID_START_STOP_Y .....	2603
10-789. VENC_FID_INT_START_X_FID_INT_START_Y .....	2603
10-790. Register Call Summary for Register VENC_FID_INT_START_X_FID_INT_START_Y.....	2603
10-791. VENC_FID_INT_OFFSET_Y_FID_EXT_START_X.....	2604
10-792. Register Call Summary for Register VENC_FID_INT_OFFSET_Y_FID_EXT_START_X.....	2604
10-793. VENC_FID_EXT_START_Y_FID_EXT_OFFSET_Y.....	2604
10-794. Register Call Summary for Register VENC_FID_EXT_START_Y_FID_EXT_OFFSET_Y .....	2604
10-795. VENC_TVDETGP_INT_START_STOP_X.....	2605
10-796. Register Call Summary for Register VENC_TVDETGP_INT_START_STOP_X.....	2605
10-797. VENC_TVDETGP_INT_START_STOP_Y.....	2605
10-798. Register Call Summary for Register VENC_TVDETGP_INT_START_STOP_Y.....	2605
10-799. VENC_GEN_CTRL .....	2606
10-800. Register Call Summary for Register VENC_GEN_CTRL.....	2607
10-801. VENC_OUTPUT_CONTROL.....	2607
10-802. Register Call Summary for Register VENC_OUTPUT_CONTROL .....	2608
10-803. VENC_OUTPUT_TEST .....	2608
10-804. Register Call Summary for Register VENC_OUTPUT_TEST.....	2608
11-1. Integration Attributes .....	2613
11-2. Clocks and Resets .....	2613

11-3.	Hardware Requests .....	2614
11-4.	SGX Instance Summary .....	2618
11-5.	SGX Registers Mapping Summary.....	2618
11-6.	OCP_REVISION.....	2619
11-7.	Register Call Summary for Register OCP_REVISION .....	2619
11-8.	OCP_HWINFO.....	2619
11-9.	Register Call Summary for Register OCP_HWINFO .....	2619
11-10.	OCP_SYSCONFIG.....	2620
11-11.	Register Call Summary for Register OCP_SYSCONFIG .....	2620
11-12.	OCP_IRQSTATUS_RAW_0 .....	2620
11-13.	Register Call Summary for Register OCP_IRQSTATUS_RAW_0.....	2621
11-14.	OCP_IRQSTATUS_RAW_1 .....	2621
11-15.	Register Call Summary for Register OCP_IRQSTATUS_RAW_1.....	2621
11-16.	OCP_IRQSTATUS_RAW_2 .....	2622
11-17.	Register Call Summary for Register OCP_IRQSTATUS_RAW_2.....	2622
11-18.	OCP_IRQSTATUS_0 .....	2622
11-19.	Register Call Summary for Register OCP_IRQSTATUS_0.....	2623
11-20.	OCP_IRQSTATUS_1 .....	2623
11-21.	Register Call Summary for Register OCP_IRQSTATUS_1.....	2623
11-22.	OCP_IRQSTATUS_2 .....	2623
11-23.	Register Call Summary for Register OCP_IRQSTATUS_2.....	2624
11-24.	OCP_IRQENABLE_SET_0 .....	2624
11-25.	Register Call Summary for Register OCP_IRQENABLE_SET_0.....	2624
11-26.	OCP_IRQENABLE_SET_1 .....	2625
11-27.	Register Call Summary for Register OCP_IRQENABLE_SET_1.....	2625
11-28.	OCP_IRQENABLE_SET_2 .....	2625
11-29.	Register Call Summary for Register OCP_IRQENABLE_SET_2.....	2626
11-30.	OCP_IRQENABLE_CLR_0 .....	2626
11-31.	Register Call Summary for Register OCP_IRQENABLE_CLR_0.....	2626
11-32.	OCP_IRQENABLE_CLR_1 .....	2627
11-33.	Register Call Summary for Register OCP_IRQENABLE_CLR_1.....	2627
11-34.	OCP_IRQENABLE_CLR_2 .....	2627
11-35.	Register Call Summary for Register OCP_IRQENABLE_CLR_2.....	2628
11-36.	OCP_PAGE_CONFIG .....	2628
11-37.	Register Call Summary for Register OCP_PAGE_CONFIG.....	2628
11-38.	OCP_INTERRUPT_EVENT.....	2629
11-39.	Register Call Summary for Register OCP_INTERRUPT_EVENT .....	2630
11-40.	OCP_DEBUG_CONFIG .....	2630
11-41.	Register Call Summary for Register OCP_DEBUG_CONFIG.....	2631
11-42.	OCP_DEBUG_STATUS.....	2631
11-43.	Register Call Summary for Register OCP_DEBUG_STATUS .....	2633
13-1.	MCmd Qualifier Description .....	2644
13-2.	MReqInfo Qualifier Description .....	2645
13-3.	SResp Qualifier Description.....	2645
13-4.	Master NIUs .....	2647
13-5.	Slave NIUs .....	2647
13-6.	L4_PER TAs .....	2648
13-7.	L4_PER IAs .....	2649
13-8.	L4_CFG TAs .....	2649

13-9. L4_CFG IAs .....	2650
13-10. L4_WKUP TAs .....	2650
13-11. L4_WKUP IAs .....	2650
13-12. Integration Attributes .....	2654
13-13. Clocks and Resets .....	2654
13-14. Hardware Requests .....	2654
13-15. Master NIUs .....	2655
13-16. Slave NIUs .....	2656
13-17. L3 Clock Domains and Elements .....	2656
13-18. ConnID Values .....	2659
13-19. L3 Firewall Exported Reset Values .....	2661
13-20. L3 Firewall Exported Values Mapping .....	2661
13-21. MReqInfo Values .....	2662
13-22. L3 ReqInfo Mapping .....	2663
13-23. Slave NIU Firewall and Region Configuration .....	2664
13-24. L3 Firewall Read/Write Permission-Setting Register .....	2666
13-25. L3 Firewall Permission-Setting Register .....	2667
13-26. L3 Firewall Error-Logging Registers .....	2668
13-27. L3 Firewalls Default Configurations .....	2669
13-28. Control Module Register – Factorization .....	2669
13-29. L3 Connectivity and Holes Error Routing .....	2672
13-30. Interconnect Flag Mapping .....	2673
13-31. Global Initialization of Surrounding Modules .....	2674
13-32. Register Call Summary: Main Sequence – Error Analysis Mode .....	2676
13-33. Subprocess Call Summary for Main Sequence – Error Analysis Mode .....	2676
13-34. Protection Violation Error Identification .....	2676
13-35. L3 Standard Error Identification .....	2676
13-36. FLAGMUX Configuration .....	2677
13-37. L3 Firewall Instance Summary .....	2678
13-38. L3 Firewall Registers Summary .....	2678
13-39. L3 Firewall Registers Mapping Summary .....	2679
13-40. L3 Firewall Registers Mapping Summary .....	2679
13-41. L3 Firewall Registers Mapping Summary .....	2680
13-42. L3 Firewall Registers Mapping Summary .....	2680
13-43. ERROR_LOG_k .....	2681
13-44. Register Call Summary for Register ERROR_LOG_k .....	2681
13-45. LOGICAL_ADDR_ERRLOG_k .....	2682
13-46. Register Call Summary for Register LOGICAL_ADDR_ERRLOG_k .....	2682
13-47. REGUPDATE_CONTROL .....	2682
13-48. Register Call Summary for Register REGUPDATE_CONTROL .....	2682
13-49. START_REGION_i .....	2683
13-50. Register Call Summary for Register START_REGION_i .....	2683
13-51. Size of START_REGION_i[] START_REGION Bit Field .....	2683
13-52. END_REGION_i .....	2683
13-53. Register Call Summary for Register END_REGION_i .....	2684
13-54. Size of END_REGION_i[] END_REGION Bit Field .....	2684
13-55. MRM_PERMISSION_REGION_HIGH_j .....	2685
13-56. Register Call Summary for Register MRM_PERMISSION_REGION_HIGH_j .....	2685
13-57. MRM_PERMISSION_REGION_LOW_j .....	2686

13-58. Register Call Summary for Register MRM_PERMISSION_REGION_LOW_j.....	2687
13-59. Reset Value for MRM_PERMISSION_REGION_LOW_j.....	2687
13-60. HOST Instance Summary .....	2687
13-61. HOST Registers Summary .....	2687
13-62. L3_HOST_STDHOSHTDR_COREREG.....	2688
13-63. Register Call Summary for Register L3_HOST_STDHOSHTDR_COREREG .....	2688
13-64. L3_HOST_STDHOSHTDR_VERSIONREG .....	2688
13-65. Register Call Summary for Register L3_HOST_STDHOSHTDR_VERSIONREG.....	2689
13-66. L3_HOST_STDHOSHTDR_MAINCTLREG.....	2689
13-67. Register Call Summary for Register L3_HOST_STDHOSHTDR_MAINCTLREG .....	2690
13-68. L3_HOST_STDERRLOG_SVRTSTDLVL.....	2690
13-69. Register Call Summary for Register L3_HOST_STDERRLOG_SVRTSTDLVL .....	2690
13-70. L3_HOST_STDERRLOG_SVRTCUSTOMLVL .....	2691
13-71. Register Call Summary for Register L3_HOST_STDERRLOG_SVRTCUSTOMLVL .....	2691
13-72. L3_HOST_STDERRLOG_MAIN .....	2691
13-73. Register Call Summary for Register L3_HOST_STDERRLOG_MAIN .....	2692
13-74. L3_HOST_STDERRLOG_HDR .....	2692
13-75. Register Call Summary for Register L3_HOST_STDERRLOG_HDR.....	2693
13-76. L3_HOST_STDERRLOG_MSTADDR.....	2693
13-77. Register Call Summary for Register L3_HOST_STDERRLOG_MSTADDR .....	2693
13-78. L3_HOST_STDERRLOG_SLVADDR .....	2693
13-79. Register Call Summary for Register L3_HOST_STDERRLOG_SLVADDR.....	2694
13-80. L3_HOST_STDERRLOG_INFO.....	2694
13-81. Register Call Summary for Register L3_HOST_STDERRLOG_INFO .....	2694
13-82. L3_HOST_STDERRLOG_SLVOFSLSB .....	2694
13-83. Register Call Summary for Register L3_HOST_STDERRLOG_SLVOFSLSB .....	2694
13-84. L3_HOST_STDERRLOG_SLVOFSMSB.....	2695
13-85. Register Call Summary for Register L3_HOST_STDERRLOG_SLVOFSMSB .....	2695
13-86. L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR .....	2695
13-87. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR.....	2696
13-88. L3_HOST_STDERRLOG_CUSTOMINFO_INFO .....	2696
13-89. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_INFO.....	2696
13-90. L3_HOST_STDERRLOG_CUSTOMINFO_WR .....	2696
13-91. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_WR.....	2697
13-92. L3_HOST_STDERRLOG_CUSTOMINFO_ADDR.....	2697
13-93. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_ADDR .....	2697
13-94. L3_HOST_STDERRLOG_CUSTOMINFO_DECERR .....	2697
13-95. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_DECERR.....	2698
13-96. TARG Instance Summary .....	2698
13-97. TARG Registers Summary .....	2699
13-98. TARG Registers Summary .....	2699
13-99. TARG Registers Summary .....	2700
13-100. TARG Registers Summary.....	2700
13-101. TARG Registers Summary.....	2701
13-102. TARG Registers Summary.....	2701
13-103. TARG Registers Summary.....	2702
13-104. L3_TARG_STDHOSHTDR_COREREG .....	2702
13-105. Register Call Summary for Register L3_TARG_STDHOSHTDR_COREREG.....	2703
13-106. L3_TARG_STDHOSHTDR_VERSIONREG .....	2703



13-107. Register Call Summary for Register L3_TARG_STDHOSTHDR_VERSIONREG .....	2703
13-108. L3_TARG_STDHOSTHDR_MAINCTLREG .....	2704
13-109. Register Call Summary for Register L3_TARG_STDHOSTHDR_MAINCTLREG.....	2704
13-110. L3_TARG_STDHOSTHDR_NTTPADDR_0 .....	2704
13-111. Register Call Summary for Register L3_TARG_STDHOSTHDR_NTTPADDR_0.....	2705
13-112. L3_TARG_STDERRLOG_SVRTSTDLVL .....	2705
13-113. Register Call Summary for Register L3_TARG_STDERRLOG_SVRTSTDLVL.....	2705
13-114. L3_TARG_STDERRLOG_SVRTCUSTOMLVL .....	2706
13-115. Register Call Summary for Register L3_TARG_STDERRLOG_SVRTCUSTOMLVL.....	2706
13-116. L3_TARG_STDERRLOG_MAIN .....	2706
13-117. Register Call Summary for Register L3_TARG_STDERRLOG_MAIN.....	2707
13-118. L3_TARG_STDERRLOG_HDR .....	2707
13-119. Register Call Summary for Register L3_TARG_STDERRLOG_HDR .....	2708
13-120. L3_TARG_STDERRLOG_MSTADDR .....	2708
13-121. Register Call Summary for Register L3_TARG_STDERRLOG_MSTADDR.....	2708
13-122. L3_TARG_STDERRLOG_SLVADDR .....	2708
13-123. Register Call Summary for Register L3_TARG_STDERRLOG_SLVADDR.....	2709
13-124. L3_TARG_STDERRLOG_INFO .....	2709
13-125. Register Call Summary for Register L3_TARG_STDERRLOG_INFO.....	2709
13-126. L3_TARG_STDERRLOG_SLVOFSLSB .....	2709
13-127. Register Call Summary for Register L3_TARG_STDERRLOG_SLVOFSLSB.....	2709
13-128. L3_TARG_STDERRLOG_SLVOFSMSB .....	2710
13-129. Register Call Summary for Register L3_TARG_STDERRLOG_SLVOFSMSB.....	2710
13-130. L3_TARG_STDERRLOG_CUSTOMINFO_INFO.....	2710
13-131. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_INFO .....	2711
13-132. L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR.....	2711
13-133. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR .....	2711
13-134. L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE .....	2711
13-135. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE .....	2712
13-136. L3_TARG_ADDRSPACESIZELOG.....	2712
13-137. Register Call Summary for Register L3_TARG_ADDRSPACESIZELOG .....	2712
13-138. PWR_DISC Instance Summary .....	2713
13-139. PWR_DISC Registers Summary .....	2713
13-140. L3_PWR_DISC_STDHOSTHDR_COREREG .....	2713
13-141. Register Call Summary for Register L3_PWR_DISC_STDHOSTHDR_COREREG .....	2714
13-142. L3_PWR_DISC_STDHOSTHDR_VERSIONREG .....	2714
13-143. Register Call Summary for Register L3_PWR_DISC_STDHOSTHDR_VERSIONREG.....	2715
13-144. L3_PWR_DISC_STDHOSTHDR_MAINCTLREG .....	2715
13-145. Register Call Summary for Register L3_PWR_DISC_STDHOSTHDR_MAINCTLREG.....	2715
13-146. L3_PWR_DISC_STDERRLOG_SVRTSTDLVL.....	2716
13-147. Register Call Summary for Register L3_PWR_DISC_STDERRLOG_SVRTSTDLVL .....	2716
13-148. L3_PWR_DISC_STDERRLOG_MAIN .....	2716
13-149. Register Call Summary for Register L3_PWR_DISC_STDERRLOG_MAIN.....	2717
13-150. L3_PWR_DISC_STDERRLOG_HDR .....	2717
13-151. Register Call Summary for Register L3_PWR_DISC_STDERRLOG_HDR.....	2718
13-152. L3_PWR_DISC_STDERRLOG_MSTADDR.....	2718
13-153. Register Call Summary for Register L3_PWR_DISC_STDERRLOG_MSTADDR .....	2718
13-154. L3_PWR_DISC_STDERRLOG_SLVADDR .....	2718
13-155. Register Call Summary for Register L3_PWR_DISC_STDERRLOG_SLVADDR.....	2719

13-156. L3_PWR_DISC_STDERRLOG_INFO.....	2719
13-157. Register Call Summary for Register L3_PWR_DISC_STDERRLOG_INFO .....	2719
13-158. L3_PWR_DISC_STDERRLOG_SLVOFSLSB .....	2719
13-159. Register Call Summary for Register L3_PWR_DISC_STDERRLOG_SLVOFSLSB .....	2719
13-160. L3_PWR_DISC_STDERRLOG_SLVOFSMSB.....	2720
13-161. Register Call Summary for Register L3_PWR_DISC_STDERRLOG_SLVOFSMSB .....	2720
13-162. FLAGMUX Instance Summary .....	2720
13-163. FLAGMUX Registers Summary .....	2720
13-164. FLAGMUX Registers Summary .....	2721
13-165. L3_FLAGMUX_STDHOSHTDR_COREREG.....	2721
13-166. Register Call Summary for Register L3_FLAGMUX_STDHOSHTDR_COREREG .....	2722
13-167. L3_FLAGMUX_STDHOSHTDR_VERSIONREG .....	2722
13-168. Register Call Summary for Register L3_FLAGMUX_STDHOSHTDR_VERSIONREG .....	2722
13-169. L3_FLAGMUX_MASK0 .....	2722
13-170. Register Call Summary for Register L3_FLAGMUX_MASK0 .....	2723
13-171. L3_FLAGMUX_REGERR0 .....	2723
13-172. Register Call Summary for Register L3_FLAGMUX_REGERR0 .....	2723
13-173. L3_FLAGMUX_MASK1 .....	2723
13-174. Register Call Summary for Register L3_FLAGMUX_MASK1 .....	2723
13-175. L3_FLAGMUX_REGERR1 .....	2724
13-176. Register Call Summary for Register L3_FLAGMUX_REGERR1 .....	2724
13-177. BW_REGULATOR Instance Summary.....	2724
13-178. BW_REGULATOR Registers Summary .....	2724
13-179. BW_REGULATOR Registers Summary .....	2724
13-180. L3_BW_R_STDHOSHTDR_COREREG .....	2725
13-181. Register Call Summary for Register L3_BW_R_STDHOSHTDR_COREREG .....	2725
13-182. L3_BW_R_STDHOSHTDR_VERSIONREG.....	2726
13-183. Register Call Summary for Register L3_BW_R_STDHOSHTDR_VERSIONREG .....	2726
13-184. L3_BW_R_BANDWIDTH .....	2726
13-185. Register Call Summary for Register L3_BW_R_BANDWIDTH .....	2726
13-186. L3_BW_R_WATERMARK .....	2727
13-187. Register Call Summary for Register L3_BW_R_WATERMARK.....	2727
13-188. L3_BW_R_PRESS.....	2727
13-189. Register Call Summary for Register L3_BW_R_PRESS .....	2727
13-190. L3_BW_R_CLEARHISTORY.....	2728
13-191. Register Call Summary for Register L3_BW_R_CLEARHISTORY .....	2728
13-192. Rate Adapt Instance Summary.....	2728
13-193. Rate Adapt Register Summary .....	2728
13-194. L3_RA_STDHOSHTDR_COREREG .....	2729
13-195. Register Call Summary for Register L3_RA_STDHOSHTDR_COREREG .....	2729
13-196. L3_RA_STDHOSHTDR_VERSIONREG.....	2729
13-197. Register Call Summary for Register L3_RA_STDHOSHTDR_VERSIONREG .....	2730
13-198. L3_RA_CNF .....	2730
13-199. Register Call Summary for Register L3_RA_CNF.....	2731
13-200. STATCOLL Instance Summary .....	2731
13-201. STATCOLL Register Summary .....	2731
13-202. L3_STCOL_STDHOSHTDR_COREREG.....	2732
13-203. Register Call Summary for Register L3_STCOL_STDHOSHTDR_COREREG .....	2733
13-204. L3_STCOL_STDHOSHTDR_VERSIONREG .....	2733

13-205. Register Call Summary for Register L3_STCOL_STDHOSTHDR_VERSIONREG .....	2733
13-206. L3_STCOL_EN .....	2734
13-207. Register Call Summary for Register L3_STCOL_EN .....	2734
13-208. L3_STCOL_SOFTEN .....	2734
13-209. Register Call Summary for Register L3_STCOL_SOFTEN .....	2734
13-210. L3_STCOL_TRIGEN .....	2735
13-211. Register Call Summary for Register L3_STCOL_TRIGEN .....	2735
13-212. L3_STCOL_REQEVT .....	2735
13-213. Register Call Summary for Register L3_STCOL_REQEVT .....	2735
13-214. L3_STCOL_RSPEVT .....	2736
13-215. Register Call Summary for Register L3_STCOL_RSPEVT .....	2736
13-216. L3_STCOL_EVTMUX_SEL0 .....	2736
13-217. Register Call Summary for Register L3_STCOL_EVTMUX_SEL0 .....	2737
13-218. L3_STCOL_EVTMUX_SEL1 .....	2737
13-219. Register Call Summary for Register L3_STCOL_EVTMUX_SEL1 .....	2737
13-220. L3_STCOL_EVTMUX_SEL2 .....	2737
13-221. Register Call Summary for Register L3_STCOL_EVTMUX_SEL2 .....	2737
13-222. L3_STCOL_EVTMUX_SEL3 .....	2738
13-223. Register Call Summary for Register L3_STCOL_EVTMUX_SEL3 .....	2738
13-224. L3_STCOL_EVTMUX_SEL4 .....	2738
13-225. Register Call Summary for Register L3_STCOL_EVTMUX_SEL4 .....	2738
13-226. L3_STCOL_DUMP_IDENTIFIER .....	2739
13-227. Register Call Summary for Register L3_STCOL_DUMP_IDENTIFIER .....	2739
13-228. L3_STCOL_DUMP_COLLECTTIME .....	2739
13-229. Register Call Summary for Register L3_STCOL_DUMP_COLLECTTIME .....	2739
13-230. L3_STCOL_DUMP_SLVADDR .....	2740
13-231. Register Call Summary for Register L3_STCOL_DUMP_SLVADDR .....	2740
13-232. L3_STCOL_DUMP_MSTADDR .....	2740
13-233. Register Call Summary for Register L3_STCOL_DUMP_MSTADDR .....	2740
13-234. L3_STCOL_DUMP_SLVOFS .....	2740
13-235. Register Call Summary for Register L3_STCOL_DUMP_SLVOFS .....	2741
13-236. L3_STCOL_DUMP_MANUAL .....	2741
13-237. Register Call Summary for Register L3_STCOL_DUMP_MANUAL .....	2741
13-238. L3_STCOL_DUMP_SEND .....	2741
13-239. Register Call Summary for Register L3_STCOL_DUMP_SEND .....	2741
13-240. L3_STCOL_FILTER_i_GLOBALEN .....	2742
13-241. Register Call Summary for Register L3_STCOL_FILTER_i_GLOBALEN .....	2742
13-242. L3_STCOL_FILTER_i_EN_k .....	2742
13-243. Register Call Summary for Register L3_STCOL_FILTER_i_EN_k .....	2742
13-244. L3_STCOL_FILTER_i_MASK_m_MSTADDR .....	2743
13-245. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_MSTADDR .....	2743
13-246. L3_STCOL_FILTER_i_MASK_m_RD .....	2743
13-247. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_RD .....	2743
13-248. L3_STCOL_FILTER_i_MASK_m_WR .....	2744
13-249. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_WR .....	2744
13-250. L3_STCOL_FILTER_i_MASK_m_ERR .....	2744
13-251. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_ERR .....	2744
13-252. L3_STCOL_FILTER_i_MASK_m_USERINFO .....	2745
13-253. Register Call Summary for Register L3_STCOL_FILTER_i_MASK_m_USERINFO .....	2745

13-254. L3_STCOL_FILTER_i_MATCH_m_MSTADDR.....	2745
13-255. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_MSTADDR .....	2745
13-256. L3_STCOL_FILTER_i_MATCH_m_RD .....	2746
13-257. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_RD .....	2746
13-258. L3_STCOL_FILTER_i_MATCH_m_WR .....	2746
13-259. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_WR.....	2746
13-260. L3_STCOL_FILTER_i_MATCH_m_ERR .....	2747
13-261. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_ERR.....	2747
13-262. L3_STCOL_FILTER_i_MATCH_m_USERINFO.....	2747
13-263. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH_m_USERINFO .....	2747
13-264. L3_STCOL_OP_i_THRESHOLD_MINVAL.....	2748
13-265. Register Call Summary for Register L3_STCOL_OP_i_THRESHOLD_MINVAL .....	2748
13-266. L3_STCOL_OP_i_THRESHOLD_MAXVAL .....	2748
13-267. Register Call Summary for Register L3_STCOL_OP_i_THRESHOLD_MAXVAL.....	2748
13-268. L3_STCOL_OP_i_EVTINFOSEL .....	2748
13-269. Register Call Summary for Register L3_STCOL_OP_i_EVTINFOSEL.....	2749
13-270. L3_STCOL_OP_i_SEL .....	2749
13-271. Register Call Summary for Register L3_STCOL_OP_i_SEL.....	2749
13-272. Integration Attributes.....	2751
13-273. Clocks and Resets .....	2751
13-274. L4_PER TAs.....	2752
13-275. L4_PER IAs.....	2753
13-276. L4_CFG TAs.....	2753
13-277. L4_CFG IAs.....	2754
13-278. L4_WKUP TAs .....	2754
13-279. L4_WKUP IAs .....	2754
13-280. L4 ConnID Definition.....	2756
13-281. L4_PER Firewall Default Configuration .....	2758
13-282. L4_CFG Firewall Default Configuration .....	2758
13-283. Region Allocations for L4_PER Interconnect.....	2760
13-284. Region Allocations for L4_CFG Interconnect.....	2762
13-285. Region Allocations for L4_WKUP Interconnect .....	2764
13-286. L4 Firewall Register Description Overview .....	2765
13-287. L4 CODE Bit Field Definition .....	2765
13-288. L4 Time-Out Link and TA Programming .....	2766
13-289. Global Initialization of Surrounding Modules .....	2770
13-290. Main Sequence – Error Analysis Mode .....	2772
13-291. Subprocess Call Summary for Main Sequence – Error Analysis Mode .....	2772
13-292. Protection Violation Error Identification.....	2773
13-293. Unsupported Command/Address Hole Error Identification .....	2773
13-294. Reset TA and Module .....	2773
13-295. Time-Out Configuration .....	2774
13-296. Firewall Configuration .....	2774
13-297. L4_PER Instance Summary .....	2775
13-298. L4_CFG Instance Summary .....	2775
13-299. L4_WKUP Instance Summary.....	2776
13-300. IA Registers Mapping Summary .....	2777
13-301. L4_IA_COMPONENT_L .....	2777
13-302. Register Call Summary for Register L4_IA_COMPONENT_L.....	2778

13-303. L4_IA_COMPONENT_H .....	2778
13-304. Register Call Summary for Register L4_IA_COMPONENT_H.....	2778
13-305. L4_IA_CORE_L .....	2778
13-306. Register Call Summary for Register L4_IA_CORE_L.....	2778
13-307. L4_IA_CORE_H.....	2779
13-308. Register Call Summary for Register L4_IA_CORE_H .....	2779
13-309. L4_IA_AGENT_CONTROL_L .....	2779
13-310. Register Call Summary for Register L4_IA_AGENT_CONTROL_L.....	2780
13-311. L4_IA_AGENT_CONTROL_H.....	2780
13-312. Register Call Summary for Register L4_IA_AGENT_CONTROL_H .....	2780
13-313. L4_IA_AGENT_STATUS_L.....	2780
13-314. Register Call Summary for Register L4_IA_AGENT_STATUS_L .....	2781
13-315. L4_IA_AGENT_STATUS_H .....	2781
13-316. Register Call Summary for Register L4_IA_AGENT_STATUS_H.....	2781
13-317. L4_IA_ERROR_LOG_L .....	2781
13-318. Register Call Summary for Register L4_IA_ERROR_LOG_L.....	2782
13-319. L4_IA_ERROR_LOG_H.....	2782
13-320. Register Call Summary for Register L4_IA_ERROR_LOG_H .....	2782
13-321. L4_IA_ERROR_LOG_ADDR_L .....	2782
13-322. Register Call Summary for Register L4_IA_ERROR_LOG_ADDR_L .....	2782
13-323. L4_IA_ERROR_LOG_ADDR_H .....	2783
13-324. Register Call Summary for Register L4_IA_ERROR_LOG_ADDR_H .....	2783
13-325. CFG_TA Register Mapping Summary 1 .....	2783
13-326. CFG_TA Register Mapping Summary 2 .....	2783
13-327. CFG_TA Register Mapping Summary 3 .....	2784
13-328. CFG_TA Register Mapping Summary 4 .....	2784
13-329. CFG_TA Register Mapping Summary 5 .....	2785
13-330. CFG_TA Register Mapping Summary 6 .....	2785
13-331. CFG_TA Register Mapping Summary 7 .....	2785
13-332. CFG_TA Register Mapping Summary 8 .....	2786
13-333. CFG_TA Register Mapping Summary 9 .....	2786
13-334. CFG_TA Register Mapping Summary 10.....	2786
13-335. CFG_TA Register Mapping Summary 11.....	2787
13-336. CFG_TA Register Mapping Summary 12.....	2787
13-337. CFG_TA Register Mapping Summary 13.....	2787
13-338. PER_TA Register Mapping Summary 1 .....	2788
13-339. PER_TA Register Mapping Summary 2 .....	2788
13-340. PER_TA Register Mapping Summary 3 .....	2788
13-341. PER_TA Register Mapping Summary 4 .....	2789
13-342. PER_TA Register Mapping Summary 5 .....	2789
13-343. PER_TA Register Mapping Summary 6 .....	2789
13-344. PER_TA Register Mapping Summary 7 .....	2790
13-345. PER_TA Register Mapping Summary 8 .....	2790
13-346. PER_TA Register Mapping Summary 9 .....	2790
13-347. PER_TA Register Mapping Summary 10.....	2791
13-348. PER_TA Register Mapping Summary 11.....	2791
13-349. PER_TA Register Mapping Summary 12.....	2791
13-350. PER_TA Register Mapping Summary 13.....	2792
13-351. WKUP_TA Register Mapping Summary 1 .....	2792

13-352. WKUP_TA Register Mapping Summary 2 .....	2792
13-353. WKUP_TA Register Mapping Summary 3 .....	2793
13-354. WKUP_TA Register Mapping Summary 4 .....	2793
13-355. WKUP_TA Register Mapping Summary 5 .....	2793
13-356. L4_TA_COMPONENT_H .....	2794
13-357. Register Call Summary for Register L4_TA_COMPONENT_H.....	2794
13-358. L4_TA_COMPONENT_L.....	2794
13-359. Register Call Summary for Register L4_TA_COMPONENT_L .....	2794
13-360. L4_TA_CORE_L .....	2794
13-361. Register Call Summary for Register L4_TA_CORE_L.....	2795
13-362. L4_TA_CORE_H.....	2795
13-363. Register Call Summary for Register L4_TA_CORE_H .....	2795
13-364. L4_TA_AGENT_CONTROL_L .....	2795
13-365. Register Call Summary for Register L4_TA_AGENT_CONTROL_L.....	2796
13-366. L4_TA_AGENT_CONTROL_H.....	2796
13-367. Register Call Summary for Register L4_TA_AGENT_CONTROL_H .....	2796
13-368. L4_TA_AGENT_STATUS_L.....	2797
13-369. Register Call Summary for Register L4_TA_AGENT_STATUS_L .....	2797
13-370. L4_TA_AGENT_STATUS_H .....	2797
13-371. Register Call Summary for Register L4_TA_AGENT_STATUS_H.....	2797
13-372. LA Register Mapping Summary .....	2798
13-373. L4_LA_COMPONENT_L.....	2798
13-374. Register Call Summary for Register L4_LA_COMPONENT_L .....	2798
13-375. L4_LA_COMPONENT_H .....	2799
13-376. Register Call Summary for Register L4_LA_COMPONENT_H.....	2799
13-377. L4_LA_NETWORK_L .....	2799
13-378. Register Call Summary for Register L4_LA_NETWORK_L .....	2799
13-379. L4_LA_NETWORK_H .....	2799
13-380. Register Call Summary for Register L4_LA_NETWORK_H.....	2799
13-381. L4_LA_INITIATOR_INFO_L .....	2800
13-382. Register Call Summary for Register L4_LA_INITIATOR_INFO_L.....	2800
13-383. Reset value for L4_LA_INITIATOR_INFO_L.....	2800
13-384. L4_LA_INITIATOR_INFO_H.....	2801
13-385. Register Call Summary for Register L4_LA_INITIATOR_INFO_H .....	2801
13-386. Reset value for L4_LA_INITIATOR_INFO_H .....	2801
13-387. L4_LA_NETWORK_CONTROL_L.....	2802
13-388. Register Call Summary for Register L4_LA_NETWORK_CONTROL_L .....	2802
13-389. L4_LA_NETWORK_CONTROL_H .....	2802
13-390. Register Call Summary for Register L4_LA_NETWORK_CONTROL_H.....	2803
13-391. L4 AP Register Summary .....	2803
13-392. L4_AP_COMPONENT_L .....	2804
13-393. Register Call Summary for Register L4_AP_COMPONENT_L .....	2804
13-394. L4_AP_COMPONENT_H.....	2804
13-395. Register Call Summary for Register L4_AP_COMPONENT_H.....	2804
13-396. L4_AP_SEGMENT_i_L.....	2804
13-397. Register Call Summary for Register L4_AP_SEGMENT_i_L .....	2805
13-398. L4_AP_SEGMENT_i_H .....	2805
13-399. Register Call Summary for Register L4_AP_SEGMENT_i_H.....	2805
13-400. Reset Value for L4_AP_SEGMENT_i .....	2805

13-401. L4_AP_PROT_GROUP_MEMBERS_k_L .....	2806
13-402. Register Call Summary for Register L4_AP_PROT_GROUP_MEMBERS_k_L.....	2807
13-403. Reset Value for L4_AP_PROT_GROUP_MEMBERS_k.....	2807
13-404. L4_AP_PROT_GROUP_MEMBERS_k_H.....	2807
13-405. Register Call Summary for Register L4_AP_PROT_GROUP_MEMBERS_k_H .....	2807
13-406. L4_AP_PROT_GROUP_ROLES_k_L.....	2808
13-407. Register Call Summary for Register L4_AP_PROT_GROUP_ROLES_k_L .....	2808
13-408. Reset Value for L4_AP_PROT_GROUP_ROLES_k .....	2808
13-409. L4_AP_PROT_GROUP_ROLES_k_H .....	2808
13-410. Register Call Summary for Register L4_AP_PROT_GROUP_ROLES_k_H.....	2808
13-411. L4_AP_REGION_I_L .....	2809
13-412. Register Call Summary for Register L4_AP_REGION_I_L.....	2809
13-413. L4_AP_REGION_I_H.....	2809
13-414. Register Call Summary for Register L4_AP_REGION_I_H .....	2810
13-415. L4_AP_REGION_I Reset Value for L4 PER.....	2810
13-416. L4_AP_REGION_I Reset Value for L4 CFG .....	2812
14-1. Integration Attributes .....	2817
14-2. Clocks and Resets .....	2817
14-3. Hardware Requests .....	2817
14-4. Global Reset Sources.....	2818
14-5. C2C Power and Reset Domain Association .....	2819
14-6. Reset Sources For the Reset Domains .....	2819
14-7. CD_C2C Clock Domain Modes.....	2827
14-8. CD_C2C Control and Status Parameters .....	2827
14-9. CD_C2C Static Dependency Association Parameters .....	2828
14-10. CD_C2C Dynamic Dependency Association Parameters.....	2828
14-11. CD_C2C Module Clocks Association .....	2828
14-12. CD_C2C Module Wake-Up Request.....	2828
14-13. CD_C2C Module Clock Management Modes and Control.....	2828
14-14. CD_C2C Modules Slave Clock Management Modes and Control.....	2829
14-15. CD_CORTEXA9 Static Dependency Association Parameters .....	2829
14-16. CD_L4_CFG Dynamic Dependency Association Parameters.....	2829
14-17. CD_L3_2 Dynamic Dependency Association Parameters .....	2830
14-18. PD_CORE C2C Power Attributes .....	2830
14-19. PRM Instance Summary.....	2830
14-20. CORE_PRM Registers Mapping Summary .....	2831
14-21. RM_C2C_C2C_CONTEXT.....	2831
14-22. Register Call Summary for Register RM_C2C_C2C_CONTEXT .....	2832
14-23. RM_C2C_C2C_FW_CONTEXT .....	2832
14-24. Register Call Summary for Register RM_C2C_C2C_FW_CONTEXT .....	2832
14-25. CORE_CM2 Registers Mapping Summary .....	2832
14-26. CM_C2C_CLKSTCTRL.....	2833
14-27. Register Call Summary for Register CM_C2C_CLKSTCTRL .....	2834
14-28. CM_C2C_STATICDEP .....	2834
14-29. Register Call Summary for Register CM_C2C_STATICDEP .....	2835
14-30. CM_C2C_DYNAMICDEP .....	2835
14-31. Register Call Summary for Register CM_C2C_DYNAMICDEP .....	2835
14-32. CM_C2C_C2C_CLKCTRL .....	2836
14-33. Register Call Summary for Register CM_C2C_C2C_CLKCTRL.....	2836

14-34. CM_C2C_C2C_FW_CLKCTRL .....	2836
14-35. Register Call Summary for Register CM_C2C_C2C_FW_CLKCTRL .....	2837
14-36. CM_L4CFG_DYNAMICDEP .....	2837
14-37. Register Call Summary for Register CM_L4CFG_DYNAMICDEP.....	2838
14-38. SCRM Instance Summary.....	2838
14-39. SCRM Registers Mapping Summary .....	2838
14-40. C2CCLKM .....	2839
14-41. Register Call Summary for Register C2CCLKM.....	2839
14-42. C2CCLKREQ .....	2840
14-43. Register Call Summary for Register C2CCLKREQ .....	2840
14-44. C2CRSTCTRL .....	2840
14-45. Register Call Summary for Register C2CRSTCTRL.....	2841
14-46. EXTWARMRSTCTRL.....	2841
14-47. Register Call Summary for Register EXTWARMRSTCTRL .....	2841
14-48. APEWARMRSTCTRL.....	2841
14-49. Register Call Summary for Register APEWARMRSTCTRL .....	2842
14-50. C2CWARMRSTCTRL.....	2842
14-51. Register Call Summary for Register C2CWARMRSTCTRL .....	2842
14-52. C2CWARMRSTST_REG .....	2843
14-53. Register Call Summary for Register C2CWARMRSTST_REG .....	2843
14-54. Clocks .....	2846
14-55. Resets.....	2846
14-56. C2C Connectivity .....	2847
14-57. C2C ReqInfo Mapping .....	2847
14-58. SSCM GPI Interrupt Mapping.....	2850
14-59. C2C Instance Summary .....	2852
14-60. ICR Registers Mapping Summary.....	2853
14-61. ICR_REVISION .....	2853
14-62. Register Call Summary for Register ICR_REVISION .....	2853
14-63. SYS_CONFIG.....	2853
14-64. Register Call Summary for Register SYS_CONFIG .....	2854
14-65. SYS_STATUS .....	2854
14-66. Register Call Summary for Register SYS_STATUS.....	2854
14-67. M_IRQ_STATUS .....	2855
14-68. Register Call Summary for Register M_IRQ_STATUS.....	2855
14-69. G_IRQ_STATUS .....	2855
14-70. Register Call Summary for Register G_IRQ_STATUS .....	2856
14-71. M_ICR.....	2856
14-72. Register Call Summary for Register M_ICR .....	2856
14-73. G_ICR.....	2857
14-74. Register Call Summary for Register G_ICR .....	2857
14-75. M_CTL .....	2857
14-76. Register Call Summary for Register M_CTL.....	2857
14-77. G_CTL .....	2858
14-78. Register Call Summary for Register G_CTL.....	2858
14-79. C2C Instance Summary .....	2859
14-80. C2C Registers Mapping Summary.....	2859
14-81. C2C_REVISION .....	2860
14-82. Register Call Summary for Register C2C_REVISION.....	2860



14-83. C2C_SYSCONFIG .....	2860
14-84. Register Call Summary for Register C2C_SYSCONFIG.....	2860
14-85. C2C_SYSSTATUS .....	2860
14-86. Register Call Summary for Register C2C_SYSSTATUS.....	2860
14-87. C2C_PORTCONFIG .....	2861
14-88. Register Call Summary for Register C2C_PORTCONFIG.....	2861
14-89. C2C_MIRRORMODE .....	2861
14-90. Register Call Summary for Register C2C_MIRRORMODE.....	2861
14-91. C2C_IRQ_RAW_STATUS_0 .....	2862
14-92. Register Call Summary for Register C2C_IRQ_RAW_STATUS_0.....	2862
14-93. C2C_IRQ_RAW_STATUS_1 .....	2863
14-94. Register Call Summary for Register C2C_IRQ_RAW_STATUS_1.....	2866
14-95. C2C_IRQ_ENABLE_STATUS_0 .....	2866
14-96. Register Call Summary for Register C2C_IRQ_ENABLE_STATUS_0.....	2866
14-97. C2C_IRQ_ENABLE_STATUS_1 .....	2867
14-98. Register Call Summary for Register C2C_IRQ_ENABLE_STATUS_1.....	2869
14-99. C2C_IRQ_ENABLE_SET_0 .....	2870
14-100. Register Call Summary for Register C2C_IRQ_ENABLE_SET_0.....	2870
14-101. C2C_IRQ_ENABLE_SET_1 .....	2870
14-102. Register Call Summary for Register C2C_IRQ_ENABLE_SET_1.....	2872
14-103. C2C_IRQ_ENABLE_CLEAR_0 .....	2872
14-104. Register Call Summary for Register C2C_IRQ_ENABLE_CLEAR_0.....	2872
14-105. C2C_IRQ_ENABLE_CLEAR_1 .....	2872
14-106. Register Call Summary for Register C2C_IRQ_ENABLE_CLEAR_1.....	2874
14-107. C2C_FCLK_FREQ.....	2874
14-108. Register Call Summary for Register C2C_FCLK_FREQ .....	2874
14-109. C2C_RX_MAX_FREQ.....	2874
14-110. Register Call Summary for Register C2C_RX_MAX_FREQ .....	2875
14-111. C2C_TX_MAX_FREQ .....	2875
14-112. Register Call Summary for Register C2C_TX_MAX_FREQ .....	2875
14-113. C2C_RX_MAX_FREQ_ACK.....	2875
14-114. Register Call Summary for Register C2C_RX_MAX_FREQ_ACK .....	2875
14-115. C2C_WAKE_REQ .....	2876
14-116. Register Call Summary for Register C2C_WAKE_REQ .....	2876
14-117. C2C_WAKE_ACK.....	2876
14-118. Register Call Summary for Register C2C_WAKE_ACK .....	2876
14-119. C2C_STANDBY .....	2877
14-120. Register Call Summary for Register C2C_STANDBY .....	2877
14-121. C2C_STANDBY_IN.....	2877
14-122. Register Call Summary for Register C2C_STANDBY_IN .....	2877
14-123. C2C_WAIT .....	2877
14-124. Register Call Summary for Register C2C_WAIT .....	2878
14-125. C2C_GENI_CONTROL .....	2878
14-126. Register Call Summary for Register C2C_GENI_CONTROL .....	2880
14-127. C2C_GENI_MASK .....	2881
14-128. Register Call Summary for Register C2C_GENI_MASK .....	2882
14-129. C2C_GENO_STATUS.....	2882
14-130. Register Call Summary for Register C2C_GENO_STATUS .....	2883
14-131. C2C_GENO_INTERRUPT .....	2883

14-132. Register Call Summary for Register C2C_GENO_INTERRUPT .....	2886
14-133. C2C_GENO_LEVEL .....	2886
14-134. Register Call Summary for Register C2C_GENO_LEVEL .....	2888
15-1. DMM Integration Attributes .....	2896
15-2. DMM Clocks and Resets .....	2896
15-3. Hardware Requests .....	2897
15-4. DMM TILER Container Geometry .....	2897
15-5. DMM Clocks .....	2899
15-6. DMM Local Power Management Features .....	2900
15-7. DMM Hardware Status Features .....	2900
15-8. Events .....	2900
15-9. Well Formed Tiled Mode 2D Block Requests .....	2913
15-10. Coding and Description of TILER Modes .....	2917
15-11. Coding and Description of TILER Orientations .....	2917
15-12. Tiled Mode Container Characteristics .....	2918
15-13. TILER Aliased View in the L3 Interconnect Mapping .....	2927
15-14. Address Format in the TILER-Specific Address Map .....	2927
15-15. Address Format of the TILER Aliased View in the System Address Map .....	2928
15-16. Minimum Buffer Size to Efficiently Handle Lines of up to N Elements .....	2929
15-17. Memory Data Payload for Buffered Raster-Based Initiators on up-to-32-bit DDR, lpDDR, or lpDDR2 .....	2930
15-18. Memory Data Payload for Buffered Raster-Based Initiators on 32-bit DDR2 or DDR3 .....	2930
15-19. Address Definition .....	2935
15-20. Configuration .....	2935
15-21. Address Definition .....	2936
15-22. Configuration .....	2936
15-23. Address Upper Bits Shifting .....	2938
15-24. Global Initialization of Surrounding Modules .....	2939
15-25. DMM Global Initialization .....	2939
15-26. Coding and Description of TILER Modes .....	2939
15-27. Configuration Settings and LUT Refill .....	2939
15-28. Interleaving Settings .....	2940
15-29. Aliased Tiled View Orientation Settings and LUT Refill .....	2940
15-30. Priority Settings .....	2940
15-31. Error Handling .....	2941
15-32. 29-Bit View Address Offset and 33-Bit Full TILER Address for an 8-Bit Frame-Buffer .....	2942
15-33. 29-Bit View Address Offset and 33-Bit Full TILER Address for a 1D Object .....	2944
15-34. DMM Instance Summary .....	2945
15-35. DMM Registers Mapping Summary .....	2945
15-36. DMM_REVISION .....	2946
15-37. Register Call Summary for Register DMM_REVISION .....	2946
15-38. DMM_HWINFO .....	2946
15-39. Register Call Summary for Register DMM_HWINFO .....	2946
15-40. DMM_LISA_HWINFO .....	2947
15-41. Register Call Summary for Register DMM_LISA_HWINFO .....	2947
15-42. DMM_SYSCONFIG .....	2947
15-43. Register Call Summary for Register DMM_SYSCONFIG .....	2948
15-44. DMM_LISA_LOCK .....	2948
15-45. Register Call Summary for Register DMM_LISA_LOCK .....	2948
15-46. DMM_LISA_MAP_i .....	2949

15-47. Register Call Summary for Register DMM_LISA_MAP_i .....	2949
15-48. DMM_TILER_HWINFO .....	2950
15-49. Register Call Summary for Register DMM_TILER_HWINFO .....	2950
15-50. DMM_TILER_OR0 .....	2950
15-51. Register Call Summary for Register DMM_TILER_OR0 .....	2951
15-52. DMM_TILER_OR1 .....	2951
15-53. Register Call Summary for Register DMM_TILER_OR1 .....	2952
15-54. DMM_PAT_HWINFO .....	2953
15-55. Register Call Summary for Register DMM_PAT_HWINFO .....	2953
15-56. DMM_PAT_GEOMETRY .....	2954
15-57. Register Call Summary for Register DMM_PAT_GEOMETRY .....	2954
15-58. DMM_PAT_CONFIG .....	2955
15-59. Register Call Summary for Register DMM_PAT_CONFIG .....	2955
15-60. DMM_PAT_VIEW0 .....	2955
15-61. Register Call Summary for Register DMM_PAT_VIEW0 .....	2957
15-62. DMM_PAT_VIEW1 .....	2957
15-63. Register Call Summary for Register DMM_PAT_VIEW1 .....	2958
15-64. DMM_PAT_VIEW_MAP_i .....	2958
15-65. Register Call Summary for Register DMM_PAT_VIEW_MAP_i .....	2959
15-66. DMM_PAT_VIEW_MAP_BASE .....	2960
15-67. Register Call Summary for Register DMM_PAT_VIEW_MAP_BASE .....	2960
15-68. DMM_PAT_IRQSTATUS_RAW .....	2960
15-69. Register Call Summary for Register DMM_PAT_IRQSTATUS_RAW .....	2962
15-70. DMM_PAT_IRQSTATUS .....	2963
15-71. Register Call Summary for Register DMM_PAT_IRQSTATUS .....	2965
15-72. DMM_PAT_IRQENABLE_SET .....	2965
15-73. Register Call Summary for Register DMM_PAT_IRQENABLE_SET .....	2967
15-74. DMM_PAT_IRQENABLE_CLR .....	2968
15-75. Register Call Summary for Register DMM_PAT_IRQENABLE_CLR .....	2970
15-76. DMM_PAT_STATUS_i .....	2970
15-77. Register Call Summary for Register DMM_PAT_STATUS_i .....	2971
15-78. DMM_PAT_DESCR_i .....	2971
15-79. Register Call Summary for Register DMM_PAT_DESCR_i .....	2972
15-80. DMM_PAT_AREA_i .....	2972
15-81. Register Call Summary for Register DMM_PAT_AREA_i .....	2972
15-82. DMM_PAT_CTRL_i .....	2973
15-83. Register Call Summary for Register DMM_PAT_CTRL_i .....	2973
15-84. DMM_PAT_DATA_i .....	2974
15-85. Register Call Summary for Register DMM_PAT_DATA_i .....	2974
15-86. DMM_PEG_HWINFO .....	2974
15-87. Register Call Summary for Register DMM_PEG_HWINFO .....	2975
15-88. DMM_PEG_PRIO_k .....	2975
15-89. Register Call Summary for Register DMM_PEG_PRIO_k .....	2976
15-90. DMM_PEG_PRIO_PAT .....	2976
15-91. Register Call Summary for Register DMM_PEG_PRIO_PAT .....	2976
15-92. I/O Signals .....	2979
15-93. Integration Attributes .....	2982
15-94. Clocks and Resets .....	2982
15-95. Hardware Requests .....	2982

15-96. MAddrSpace Mapping to Chip Selects .....	2983
15-97. FIFO Allocation .....	2984
15-98. Events.....	2989
15-99. 64-Byte Linear Read Starting at Address 0x0 .....	2991
15-100. 64-Byte Linear Read Starting at Address 0x8 (LPDDR2-S2) .....	2991
15-101. 64-Byte Linear Read Starting at Address 0x8 (LPDDR2-S4) .....	2992
15-102. 64-Byte Linear Read Starting at Address 0x10 .....	2992
15-103. 64-Byte Linear Read Starting at Address 0x18 .....	2992
15-104. Turn Around Time.....	2992
15-105. SDRAM Addressing Space .....	2993
15-106. L3 Address to SDRAM Address Mapping for 16-Bit SDRAM (EMIF_SDRAM_CONFIG[28:27] REG_IBANK_POS=0) .....	2994
15-107. L3 Address to SDRAM Address Mapping for 32-Bit SDRAM (EMIF_SDRAM_CONFIG[28:27] REG_IBANK_POS=0) .....	2995
15-108. L3 Address to SDRAM Address Mapping for EMIF_SDRAM_CONFIG[28:27] REG_IBANK_POS=1 ....	2997
15-109. L3 Address to SDRAM Address Mapping for EMIF_SDRAM_CONFIG[28:27] REG_IBANK_POS=2 ....	2998
15-110. L3 Address to SDRAM Address Mapping for EMIF_SDRAM_CONFIG[28:27] REG_IBANK_POS=3 ....	2999
15-111. Global Initialization of Surrounding Modules .....	3001
15-112. EMIF Global Initialization.....	3001
15-113. Sub-process call summary for Main Sequence - EMIF Global Initialization.....	3002
15-114. Sub-sequence - EMIF Timing Initialization .....	3002
15-115. EMIF Output Impedance Calibration Mode.....	3003
15-116. EMIF Read.....	3003
15-117. EMIF Write .....	3003
15-118. EMIF SDRAM Refresh .....	3004
15-119. EMIF SDRAM Self-Refresh.....	3004
15-120. EMIF SDRAM Power-Down Mode.....	3004
15-121. EMIF LPDDR2-SDRAM Deep Power-Down Mode .....	3005
15-122. EMIF Global Warm Reset (Software reset).....	3005
15-123. EMIF Temperature Monitoring Mode .....	3005
15-124. EMIF Event Servicing.....	3007
15-125. EMIF Instance Summary .....	3009
15-126. EMIF Registers Mapping Summary .....	3009
15-127. EMIF_MOD_ID_REV .....	3011
15-128. Register Call Summary for Register EMIF_MOD_ID_REV.....	3011
15-129. EMIF_STATUS .....	3011
15-130. Register Call Summary for Register EMIF_STATUS.....	3012
15-131. EMIF_SDRAM_CONFIG .....	3012
15-132. Register Call Summary for Register EMIF_SDRAM_CONFIG .....	3013
15-133. EMIF_SDRAM_CONFIG_2 .....	3013
15-134. Register Call Summary for Register EMIF_SDRAM_CONFIG_2 .....	3014
15-135. EMIF_SDRAM_REF_CTRL .....	3014
15-136. Register Call Summary for Register EMIF_SDRAM_REF_CTRL .....	3015
15-137. EMIF_SDRAM_REF_CTRL_SHDW.....	3015
15-138. Register Call Summary for Register EMIF_SDRAM_REF_CTRL_SHDW .....	3015
15-139. EMIF_SDRAM_TIM_1 .....	3015
15-140. Register Call Summary for Register EMIF_SDRAM_TIM_1 .....	3016
15-141. EMIF_SDRAM_TIM_1_SHDW .....	3016
15-142. Register Call Summary for Register EMIF_SDRAM_TIM_1_SHDW.....	3017

15-143. EMIF_SDRAM_TIM_2.....	3017
15-144. Register Call Summary for Register EMIF_SDRAM_TIM_2 .....	3017
15-145. EMIF_SDRAM_TIM_2_SHDW .....	3018
15-146. Register Call Summary for Register EMIF_SDRAM_TIM_2_SHDW.....	3018
15-147. EMIF_SDRAM_TIM_3.....	3019
15-148. Register Call Summary for Register EMIF_SDRAM_TIM_3 .....	3019
15-149. EMIF_SDRAM_TIM_3_SHDW .....	3020
15-150. Register Call Summary for Register EMIF_SDRAM_TIM_3_SHDW.....	3020
15-151. EMIF_LPDDR2_NVM_TIM .....	3021
15-152. Register Call Summary for Register EMIF_LPDDR2_NVM_TIM.....	3021
15-153. EMIF_LPDDR2_NVM_TIM_SHDW.....	3021
15-154. Register Call Summary for Register EMIF_LPDDR2_NVM_TIM_SHDW .....	3022
15-155. EMIF_PWR_MGMT_CTRL.....	3022
15-156. Register Call Summary for Register EMIF_PWR_MGMT_CTRL.....	3023
15-157. EMIF_PWR_MGMT_CTRL_SHDW .....	3024
15-158. Register Call Summary for Register EMIF_PWR_MGMT_CTRL_SHDW.....	3024
15-159. EMIF_LPDDR2_MODE_REG_DATA .....	3024
15-160. Register Call Summary for Register EMIF_LPDDR2_MODE_REG_DATA.....	3025
15-161. EMIF_LPDDR2_MODE_REG_CFG .....	3025
15-162. Register Call Summary for Register EMIF_LPDDR2_MODE_REG_CFG .....	3025
15-163. EMIF_L3_CONFIG.....	3026
15-164. Register Call Summary for Register EMIF_L3_CONFIG .....	3026
15-165. EMIF_L3_CFG_VAL_1 .....	3027
15-166. Register Call Summary for Register EMIF_L3_CFG_VAL_1 .....	3027
15-167. EMIF_L3_CFG_VAL_2.....	3027
15-168. Register Call Summary for Register EMIF_L3_CFG_VAL_2 .....	3028
15-169. EMIF_PERF_CNT_1 .....	3028
15-170. Register Call Summary for Register EMIF_PERF_CNT_1.....	3028
15-171. EMIF_PERF_CNT_2 .....	3028
15-172. Register Call Summary for Register EMIF_PERF_CNT_2.....	3028
15-173. EMIF_PERF_CNT_CFG .....	3029
15-174. Register Call Summary for Register EMIF_PERF_CNT_CFG.....	3029
15-175. Performance Counter Filter Configuration .....	3029
15-176. EMIF_PERF_CNT_SEL.....	3030
15-177. Register Call Summary for Register EMIF_PERF_CNT_SEL .....	3030
15-178. EMIF_PERF_CNT_TIM .....	3031
15-179. Register Call Summary for Register EMIF_PERF_CNT_TIM.....	3031
15-180. EMIF_READ_IDLE_CTRL .....	3031
15-181. Register Call Summary for Register EMIF_READ_IDLE_CTRL.....	3031
15-182. EMIF_READ_IDLE_CTRL_SHDW .....	3032
15-183. Register Call Summary for Register EMIF_READ_IDLE_CTRL_SHDW.....	3032
15-184. EMIF_IRQSTATUS_RAW_SYS .....	3033
15-185. Register Call Summary for Register EMIF_IRQSTATUS_RAW_SYS .....	3033
15-186. EMIF_IRQSTATUS_RAW_LL .....	3033
15-187. Register Call Summary for Register EMIF_IRQSTATUS_RAW_LL .....	3034
15-188. EMIF_IRQSTATUS_SYS .....	3034
15-189. Register Call Summary for Register EMIF_IRQSTATUS_SYS.....	3034
15-190. EMIF_IRQSTATUS_LL.....	3035
15-191. Register Call Summary for Register EMIF_IRQSTATUS_LL .....	3035

15-192. EMIF_IRQENABLE_SET_SYS.....	3035
15-193. Register Call Summary for Register EMIF_IRQENABLE_SET_SYS .....	3036
15-194. EMIF_IRQENABLE_SET_LL .....	3036
15-195. Register Call Summary for Register EMIF_IRQENABLE_SET_LL .....	3037
15-196. EMIF_IRQENABLE_CLR_SYS .....	3037
15-197. Register Call Summary for Register EMIF_IRQENABLE_CLR_SYS .....	3037
15-198. EMIF_IRQENABLE_CLR_LL.....	3038
15-199. Register Call Summary for Register EMIF_IRQENABLE_CLR_LL .....	3038
15-200. EMIF_ZQ_CONFIG .....	3038
15-201. Register Call Summary for Register EMIF_ZQ_CONFIG .....	3039
15-202. EMIF_TEMP_ALERT_CONFIG .....	3039
15-203. Register Call Summary for Register EMIF_TEMP_ALERT_CONFIG .....	3040
15-204. EMIF_L3_ERR_LOG .....	3040
15-205. Register Call Summary for Register EMIF_L3_ERR_LOG.....	3041
15-206. EMIF_DDR_PHY_CTRL_1 .....	3041
15-207. Register Call Summary for Register EMIF_DDR_PHY_CTRL_1.....	3041
15-208. EMIF_DDR_PHY_CTRL_1_SHDW .....	3042
15-209. Register Call Summary for Register EMIF_DDR_PHY_CTRL_1_SHDW .....	3042
15-210. EMIF_DDR_PHY_CTRL_2 .....	3043
15-211. Register Call Summary for Register EMIF_DDR_PHY_CTRL_2.....	3043
15-212. GPMC I/O Description.....	3046
15-213. GPMC Pin Multiplexing Options.....	3047
15-214. GPMC Integration Attributes.....	3050
15-215. GPMC Clocks and Resets .....	3050
15-216. GPMC Hardware Requests.....	3050
15-217. GPMC Clocks.....	3053
15-218. gpmc_clk Configuration .....	3053
15-219. GPMC Local Power Management Features.....	3053
15-220. GPMC Interrupt Events.....	3054
15-221. Boot Control Interface Input Signals Description .....	3056
15-222. Idle Cycle Insertion Configuration.....	3067
15-223. Chip-Select Configuration for NAND Interfacing .....	3096
15-224. ECC Enable Settings .....	3104
15-225. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits) .....	3109
15-226. Aligned Message Byte Mapping in 8-bit NAND .....	3109
15-227. Aligned Message Byte Mapping in 16-bit NAND.....	3110
15-228. Aligned Nibble Mapping of Message in 8-bit NAND .....	3110
15-229. Misaligned Nibble Mapping of Message in 8-bit NAND .....	3111
15-230. Aligned Nibble Mapping of Message in 16-bit NAND .....	3111
15-231. Misaligned Nibble Mapping of Message in 16-bit NAND (1 Unused Nibble).....	3111
15-232. Misaligned Nibble Mapping of Message in 16-bit NAND (2 Unused Nibbles) .....	3111
15-233. Misaligned Nibble Mapping of Message in 16-bit NAND (3 Unused Nibbles) .....	3112
15-234. Prefetch Mode Configuration .....	3121
15-235. Write-Posting Mode Configuration .....	3123
15-236. GPMC Initialization.....	3128
15-237. GPMC Configuration in NOR mode .....	3128
15-238. GPMC Configuration in NAND mode.....	3128
15-239. Enable GPMC Clocks .....	3128
15-240. Enable GPMC Pads .....	3128

15-241. Reset GPMC.....	3129
15-242. NOR Memory Type .....	3129
15-243. NOR Chip-Select Configuration .....	3130
15-244. NOR Timings Configuration .....	3130
15-245. Wait Pin Configuration.....	3130
15-246. Enable Chip-Select .....	3130
15-247. NAND Memory Type.....	3130
15-248. NAND Chip-Select Configuration .....	3130
15-249. Asynchronous Read and Write Operations.....	3130
15-250. ECC Engine.....	3130
15-251. Prefetch and Write-Posting Engine .....	3131
15-252. Wait Pin Configuration.....	3131
15-253. Enable Chip-Select .....	3131
15-254. Mode Parameters Check List Table .....	3132
15-255. Access Type Parameters Check List Table .....	3132
15-256. Timing Parameters.....	3134
15-257. NAND Formulas Description Table .....	3136
15-258. Synchronous NOR Formulas Description Table .....	3137
15-259. Asynchronous NOR Formulas Description Table.....	3141
15-260. GPMC Signals.....	3144
15-261. Useful Timing Parameters on the Memory Side .....	3145
15-262. Calculating GPMC Timing Parameters.....	3146
15-263. AC Characteristics for Asynchronous Read Access.....	3147
15-264. GPMC Timing Parameters for Asynchronous Read Access .....	3148
15-265. AC Characteristics for Asynchronous Single Write ( Memory Side).....	3148
15-266. GPMC Timing parameters for Asynchronous Single Write.....	3149
15-267. Supported memories interfaces .....	3150
15-268. NAND Interface Bus Operations Summary.....	3151
15-269. NOR Interface Bus Operations Summary .....	3152
15-270. GPMC Instance Summary .....	3153
15-271. GPMC Registers Mapping Summary.....	3153
15-272. GPMC_REVISION .....	3154
15-273. Register Call Summary for Register GPMC_REVISION.....	3154
15-274. GPMC_SYSCONFIG .....	3154
15-275. Register Call Summary for Register GPMC_SYSCONFIG.....	3155
15-276. GPMC_SYSSTATUS .....	3155
15-277. Register Call Summary for Register GPMC_SYSSTATUS .....	3155
15-278. GPMC_IRQSTATUS.....	3156
15-279. Register Call Summary for Register GPMC_IRQSTATUS .....	3157
15-280. GPMC_IRQENABLE.....	3157
15-281. Register Call Summary for Register GPMC_IRQENABLE .....	3158
15-282. GPMC_TIMEOUT_CONTROL .....	3158
15-283. Register Call Summary for Register GPMC_TIMEOUT_CONTROL.....	3159
15-284. GPMC_ERR_ADDRESS.....	3159
15-285. Register Call Summary for Register GPMC_ERR_ADDRESS .....	3159
15-286. GPMC_ERR_TYPE.....	3159
15-287. Register Call Summary for Register GPMC_ERR_TYPE .....	3160
15-288. GPMC_CONFIG .....	3160
15-289. Register Call Summary for Register GPMC_CONFIG .....	3161

15-290. GPMC_STATUS .....	3161
15-291. Register Call Summary for Register GPMC_STATUS .....	3162
15-292. GPMC_CONFIG1_i .....	3162
15-293. Register Call Summary for Register GPMC_CONFIG1_i .....	3165
15-294. GPMC_CONFIG2_i .....	3165
15-295. Register Call Summary for Register GPMC_CONFIG2_i .....	3166
15-296. GPMC_CONFIG3_i .....	3166
15-297. Register Call Summary for Register GPMC_CONFIG3_i .....	3168
15-298. GPMC_CONFIG4_i .....	3168
15-299. Register Call Summary for Register GPMC_CONFIG4_i .....	3169
15-300. GPMC_CONFIG5_i .....	3169
15-301. Register Call Summary for Register GPMC_CONFIG5_i .....	3170
15-302. GPMC_CONFIG6_i .....	3170
15-303. Register Call Summary for Register GPMC_CONFIG6_i .....	3172
15-304. GPMC_CONFIG7_i .....	3172
15-305. Register Call Summary for Register GPMC_CONFIG7_i .....	3172
15-306. GPMC_NAND_COMMAND_i .....	3173
15-307. Register Call Summary for Register GPMC_NAND_COMMAND_i .....	3173
15-308. GPMC_NAND_ADDRESS_i .....	3173
15-309. Register Call Summary for Register GPMC_NAND_ADDRESS_i .....	3174
15-310. GPMC_NAND_DATA_i .....	3174
15-311. Register Call Summary for Register GPMC_NAND_DATA_i .....	3174
15-312. GPMC_PREFETCH_CONFIG1 .....	3174
15-313. Register Call Summary for Register GPMC_PREFETCH_CONFIG1 .....	3176
15-314. GPMC_PREFETCH_CONFIG2 .....	3176
15-315. Register Call Summary for Register GPMC_PREFETCH_CONFIG2 .....	3177
15-316. GPMC_PREFETCH_CONTROL .....	3177
15-317. Register Call Summary for Register GPMC_PREFETCH_CONTROL .....	3177
15-318. GPMC_PREFETCH_STATUS .....	3178
15-319. Register Call Summary for Register GPMC_PREFETCH_STATUS .....	3178
15-320. GPMC_ECC_CONFIG .....	3179
15-321. Register Call Summary for Register GPMC_ECC_CONFIG .....	3179
15-322. GPMC_ECC_CONTROL .....	3180
15-323. Register Call Summary for Register GPMC_ECC_CONTROL .....	3180
15-324. GPMC_ECC_SIZE_CONFIG .....	3181
15-325. Register Call Summary for Register GPMC_ECC_SIZE_CONFIG .....	3182
15-326. GPMC_ECCj_RESULT .....	3182
15-327. Register Call Summary for Register GPMC_ECCj_RESULT .....	3183
15-328. GPMC_BCH_RESULT0_i .....	3183
15-329. Register Call Summary for Register GPMC_BCH_RESULT0_i .....	3183
15-330. GPMC_BCH_RESULT1_i .....	3183
15-331. Register Call Summary for Register GPMC_BCH_RESULT1_i .....	3183
15-332. GPMC_BCH_RESULT2_i .....	3184
15-333. Register Call Summary for Register GPMC_BCH_RESULT2_i .....	3184
15-334. GPMC_BCH_RESULT3_i .....	3184
15-335. Register Call Summary for Register GPMC_BCH_RESULT3_i .....	3184
15-336. GPMC_BCH_RESULT4_i .....	3184
15-337. Register Call Summary for Register GPMC_BCH_RESULT4_i .....	3185
15-338. GPMC_BCH_RESULT5_i .....	3185



15-339. Register Call Summary for Register GPMC_BCH_RESULT5_i .....	3185
15-340. GPMC_BCH_RESULT6_i.....	3185
15-341. Register Call Summary for Register GPMC_BCH_RESULT6_i .....	3185
15-342. GPMC_BCH_SWDATA .....	3185
15-343. Register Call Summary for Register GPMC_BCH_SWDATA.....	3186
15-344. Integration Attributes.....	3188
15-345. Clocks and Resets .....	3189
15-346. Hardware Requests.....	3189
15-347. Local Power Management Features.....	3190
15-348. Events .....	3190
15-349. ELM_LOCATION_STATUS_i Value Decoding Table .....	3191
15-350. ELM Processing Initialization .....	3192
15-351. ELM Processing Completion for Continuous Mode .....	3193
15-352. ELM Processing Completion for Page Mode.....	3193
15-353. Use Case: Continuous Mode .....	3194
15-354. 16-bit NAND Sector Buffer Address Map .....	3195
15-355. Use Case: Page Mode .....	3195
15-356. ELM Instance Summary.....	3197
15-357. ELM Registers Mapping Summary .....	3197
15-358. ELM_REVISION.....	3198
15-359. Register Call Summary for Register ELM_REVISION .....	3199
15-360. ELM_SYSCONFIG.....	3199
15-361. Register Call Summary for Register ELM_SYSCONFIG .....	3200
15-362. ELM_SYSSTATUS .....	3200
15-363. Register Call Summary for Register ELM_SYSSTATUS .....	3200
15-364. ELM_IRQSTATUS .....	3201
15-365. Register Call Summary for Register ELM_IRQSTATUS.....	3201
15-366. ELM_IRQENABLE .....	3202
15-367. Register Call Summary for Register ELM_IRQENABLE.....	3202
15-368. ELM_LOCATION_CONFIG.....	3203
15-369. Register Call Summary for Register ELM_LOCATION_CONFIG .....	3203
15-370. ELM_PAGE_CTRL .....	3203
15-371. Register Call Summary for Register ELM_PAGE_CTRL .....	3204
15-372. ELM_SYNDROME_FRAGMENT_0_i .....	3204
15-373. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_0_i.....	3204
15-374. ELM_SYNDROME_FRAGMENT_1_i .....	3205
15-375. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_1_i.....	3205
15-376. ELM_SYNDROME_FRAGMENT_2_i .....	3205
15-377. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_2_i.....	3205
15-378. ELM_SYNDROME_FRAGMENT_3_i .....	3205
15-379. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_3_i.....	3206
15-380. ELM_SYNDROME_FRAGMENT_4_i .....	3206
15-381. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_4_i.....	3206
15-382. ELM_SYNDROME_FRAGMENT_5_i .....	3206
15-383. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_5_i.....	3206
15-384. ELM_SYNDROME_FRAGMENT_6_i .....	3207
15-385. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_6_i.....	3207
15-386. ELM_LOCATION_STATUS_i .....	3207
15-387. Register Call Summary for Register ELM_LOCATION_STATUS_i .....	3208

15-388. ELM_ERROR_LOCATION_0_i .....	3208
15-389. Register Call Summary for Register ELM_ERROR_LOCATION_0_i .....	3208
15-390. ELM_ERROR_LOCATION_1_i .....	3209
15-391. Register Call Summary for Register ELM_ERROR_LOCATION_1_i .....	3209
15-392. ELM_ERROR_LOCATION_2_i .....	3209
15-393. Register Call Summary for Register ELM_ERROR_LOCATION_2_i .....	3209
15-394. ELM_ERROR_LOCATION_3_i .....	3210
15-395. Register Call Summary for Register ELM_ERROR_LOCATION_3_i .....	3210
15-396. ELM_ERROR_LOCATION_4_i .....	3210
15-397. Register Call Summary for Register ELM_ERROR_LOCATION_4_i .....	3210
15-398. ELM_ERROR_LOCATION_5_i .....	3210
15-399. Register Call Summary for Register ELM_ERROR_LOCATION_5_i .....	3211
15-400. ELM_ERROR_LOCATION_6_i .....	3211
15-401. Register Call Summary for Register ELM_ERROR_LOCATION_6_i .....	3211
15-402. ELM_ERROR_LOCATION_7_i .....	3211
15-403. Register Call Summary for Register ELM_ERROR_LOCATION_7_i .....	3211
15-404. ELM_ERROR_LOCATION_8_i .....	3212
15-405. Register Call Summary for Register ELM_ERROR_LOCATION_8_i .....	3212
15-406. ELM_ERROR_LOCATION_9_i .....	3212
15-407. Register Call Summary for Register ELM_ERROR_LOCATION_9_i .....	3212
15-408. ELM_ERROR_LOCATION_10_i .....	3212
15-409. Register Call Summary for Register ELM_ERROR_LOCATION_10_i .....	3213
15-410. ELM_ERROR_LOCATION_11_i .....	3213
15-411. Register Call Summary for Register ELM_ERROR_LOCATION_11_i .....	3213
15-412. ELM_ERROR_LOCATION_12_i .....	3213
15-413. Register Call Summary for Register ELM_ERROR_LOCATION_12_i .....	3213
15-414. ELM_ERROR_LOCATION_13_i .....	3213
15-415. Register Call Summary for Register ELM_ERROR_LOCATION_13_i .....	3214
15-416. ELM_ERROR_LOCATION_14_i .....	3214
15-417. Register Call Summary for Register ELM_ERROR_LOCATION_14_i .....	3214
15-418. ELM_ERROR_LOCATION_15_i .....	3214
15-419. Register Call Summary for Register ELM_ERROR_LOCATION_15_i .....	3214
15-420. Integration Attributes .....	3217
15-421. Clocks and Resets .....	3217
16-1. External sDMA Request Signals .....	3222
16-2. Integration Attributes .....	3224
16-3. Clocks and Resets .....	3225
16-4. Hardware Requests .....	3225
16-5. sDMA Controller Request Mapping .....	3226
16-6. Local Power Management Features .....	3230
16-7. Clock Activity Settings .....	3230
16-8. Logical DMA Channel Events .....	3231
16-9. Parameter Values for Addressing Mode Examples (a), (b), and (c) .....	3236
16-10. Equations for Rotation .....	3236
16-11. Example Parameter Values for a 90-Degree Clockwise Image Rotation .....	3237
16-12. Buffering Disable .....	3240
16-13. Type 1 .....	3247
16-14. Type 2 With Source and Destination Address Updates .....	3247
16-15. Type 2 With Source or Destination Address Update .....	3248

16-16. Type 3 With Source and Destination Address Updates .....	3248
16-17. Type 3 With Source or Destination Address Update .....	3248
16-18. SAR RAM Allocation .....	3254
16-19. SAR_RAM_1 Memory Mapping .....	3255
16-20. SAR_RAM_2 Memory Mapping .....	3259
16-21. SAR_RAM_3 memory mapping .....	3260
16-22. sDMA Instance Summary .....	3269
16-23. sDMA Registers Mapping Summary .....	3269
16-24. DMA4_REVISION .....	3270
16-25. Register Call Summary for Register DMA4_REVISION.....	3271
16-26. DMA4_IRQSTATUS_Lj .....	3271
16-27. Register Call Summary for Register DMA4_IRQSTATUS_Lj .....	3271
16-28. DMA4_IRQENABLE_Lj .....	3271
16-29. Register Call Summary for Register DMA4_IRQENABLE_Lj .....	3271
16-30. DMA4_SYSSTATUS .....	3272
16-31. Register Call Summary for Register DMA4_SYSSTATUS .....	3272
16-32. DMA4_OCP_SYSCONFIG.....	3272
16-33. Register Call Summary for Register DMA4_OCP_SYSCONFIG .....	3273
16-34. DMA4_CAPS_0.....	3274
16-35. Register Call Summary for Register DMA4_CAPS_0 .....	3274
16-36. DMA4_CAPS_2.....	3274
16-37. Register Call Summary for Register DMA4_CAPS_2 .....	3276
16-38. DMA4_CAPS_3.....	3276
16-39. Register Call Summary for Register DMA4_CAPS_3 .....	3277
16-40. DMA4_CAPS_4.....	3277
16-41. Register Call Summary for Register DMA4_CAPS_4 .....	3278
16-42. DMA4_GCR .....	3279
16-43. Register Call Summary for Register DMA4_GCR .....	3280
16-44. DMA4_CCRi.....	3280
16-45. Register Call Summary for Register DMA4_CCRi .....	3283
16-46. DMA4_CLNK_CTRLi.....	3283
16-47. Register Call Summary for Register DMA4_CLNK_CTRLi .....	3284
16-48. DMA4_CICRi.....	3284
16-49. Register Call Summary for Register DMA4_CICRi .....	3285
16-50. DMA4_CSRi.....	3285
16-51. Register Call Summary for Register DMA4_CSRi .....	3287
16-52. DMA4_CSDPi .....	3287
16-53. Register Call Summary for Register DMA4_CSDPi .....	3289
16-54. DMA4_CENi.....	3289
16-55. Register Call Summary for Register DMA4_CENi .....	3289
16-56. DMA4_CFNi.....	3289
16-57. Register Call Summary for Register DMA4_CFNi .....	3290
16-58. DMA4_CSSAi .....	3290
16-59. Register Call Summary for Register DMA4_CSSAi.....	3290
16-60. DMA4_CDSAi .....	3290
16-61. Register Call Summary for Register DMA4_CDSAi .....	3291
16-62. DMA4_CSEli .....	3291
16-63. Register Call Summary for Register DMA4_CSEli.....	3291
16-64. DMA4_CSFli .....	3291

16-65. Register Call Summary for Register DMA4_CSFLi .....	3292
16-66. DMA4_CDELi .....	3292
16-67. Register Call Summary for Register DMA4_CDELi .....	3292
16-68. DMA4_CDFLi .....	3292
16-69. Register Call Summary for Register DMA4_CDFLi .....	3293
16-70. DMA4_CSACi .....	3293
16-71. Register Call Summary for Register DMA4_CSACi .....	3293
16-72. DMA4_CDACi .....	3293
16-73. Register Call Summary for Register DMA4_CDACi .....	3293
16-74. DMA4_CCENi .....	3294
16-75. Register Call Summary for Register DMA4_CCENi .....	3294
16-76. DMA4_CCFNi .....	3294
16-77. Register Call Summary for Register DMA4_CCFNi .....	3294
16-78. DMA4_COLORi .....	3295
16-79. Register Call Summary for Register DMA4_COLORi .....	3295
16-80. DMA4_CDPi .....	3295
16-81. Register Call Summary for Register DMA4_CDPi .....	3296
16-82. DMA4_CNDPi .....	3297
16-83. Register Call Summary for Register DMA4_CNDPi .....	3297
16-84. DMA4_CCDNi .....	3297
16-85. Register Call Summary for Register DMA4_CCDNi .....	3297
17-1. Interrupt Controllers I/O Signals .....	3302
17-2. Cortex-A9 MPU INTC Interrupt Mapping .....	3303
17-3. Cortex-M3 MPU INTC Interrupt Mapping .....	3306
18-1. Control Module I/O Description .....	3314
18-2. Clocks and Resets .....	3315
18-3. Integration Attributes .....	3317
18-4. Internal Clock Gating Options .....	3321
18-5. Bit Meanings for CONTROL_PADCONF_x Registers .....	3325
18-6. Mode Selection .....	3326
18-7. Pull Selection .....	3327
18-8. Device Wakeup Control Module Pad Configuration Register Fields .....	3328
18-9. Device Core Control Module Pad Configuration Register Fields .....	3330
18-10. Pin Types .....	3348
18-11. PBIAS Cell and Extended-Drain I/O Pin CONTROL_PBIAS_LITE Bit control .....	3350
18-12. Power Supplies .....	3351
18-13. PBIAS/Extended-Drain I/O Cells Control Signals .....	3354
18-14. Voltage Configuration .....	3355
18-15. PBIAS Error Signal Truth Table .....	3356
18-16. Band Gap Voltage and Temperature Sensor Signal Description .....	3358
18-17. ADC Code Versus Temperature .....	3360
18-18. CORE Power Domain Observability Signals Multiplexing .....	3364
18-19. DPLL Signal Observability Controls .....	3369
18-20. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 0 .....	3372
18-21. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 1 .....	3372
18-22. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 2 .....	3373
18-23. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 3 .....	3374
18-24. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 4 .....	3374
18-25. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 5 .....	3375

18-26. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 6 .....	3376
18-27. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 7 .....	3377
18-28. Observability Signals at IVAHD Subsystem Level .....	3377
18-29. Observability Signals at DSS Subsystem Level .....	3378
18-30. Observability Signals at ABE Subsystem Level .....	3379
18-31. CM2 Observable Signals Specific to Initiators .....	3380
18-32. CM2 Observable Signals Specific to Targets .....	3381
18-33. CM2 - Other Observable Signals .....	3386
18-34. CM1 Observable Signals Specific to Initiators .....	3387
18-35. CM1 Observable Signals Specific to Targets .....	3387
18-36. CM1 - Other Observable Signals .....	3388
18-37. PRM - Observable Signals Specific to Targets .....	3389
18-38. Power State Controller PGOODOUT/RET Signals .....	3390
18-39. Reset Signals in PRM Module .....	3390
18-40. PRM - Other Observable Signals .....	3392
18-41. Different DPLL Observable Clock output signals .....	3393
18-42. Other CORE DPLL Observability Signals .....	3393
18-43. Other ABE DPLL Observability Signals .....	3394
18-44. Other PERIPH DPLL Observability Signals .....	3394
18-45. Other DSI1 DPLL Observability Signals .....	3395
18-46. Other DSI2 DPLL Observability Signals .....	3395
18-47. Other USB DPLL Observability Signals .....	3396
18-48. PBIAS Control Register .....	3396
18-49. Temperature Sensor Register .....	3396
18-50. CSI Receiver control Register .....	3396
18-51. Protection Status Registers .....	3397
18-52. Signal Integrity Parameter Control Registers .....	3398
18-53. Driver Impedance Output Settings .....	3400
18-54. Driver Slew Rate Settings .....	3400
18-55. Weak Driver PUPD /Latch mode .....	3401
18-56. Internal reference configuration .....	3401
18-57. Vref cell decoupling cap selection table .....	3402
18-58. Slew control mode (coarse) settings .....	3402
18-59. Signal performance for slew control mode 0 - SC[1:0]=00 .....	3402
18-60. Load bit (fine) settings .....	3403
18-61. Signal performance for slew control mode 1 - SC[1:0]=01 .....	3403
18-62. Signal performance for slew control mode 2 - SC[1:0]=10 .....	3403
18-63. Software controls for I/Os with combined Mode and Load settings .....	3404
18-64. I/O cell Features in Different Modes .....	3404
18-65. Software controls for 50-Ω output buffer I/O cells with combined Mode and Load settings .....	3405
18-66. Different Mode features .....	3405
18-67. Driver Impedance Control .....	3405
18-68. I/O cell performance for different impedances .....	3406
18-69. I/O cell performance for C2C I/F at 1.2/1.8 V .....	3406
18-70. I/O cell performance for GPMC I/F at 1.2/1.8 V .....	3406
18-71. I/O cell performance for SDMMC2 I/F at 1.2/1.8 V .....	3407
18-72. Vref cell decoupling cap selection table .....	3407
18-73. SDMMC1 Pullup Strength Controls .....	3408
18-74. Example I2Cx Pullup vs. I2C LB Parameter Settings in Different Modes .....	3408

18-75. Internal Pullup Resistor in Fast/HS Mode .....	3408
18-76. EMIF (LPDDR2IO) Signal Group Parameter Controls to Different Interface I/O Pads Mapping.....	3409
18-77. GPMC I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping.....	3411
18-78. CAMERA I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3412
18-79. USBB1 I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3412
18-80. USBB2 I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3413
18-81. USBC1 I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping.....	3414
18-82. KEYPAD I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping.....	3414
18-83. HDQ I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping.....	3415
18-84. GPIO Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3415
18-85. SDMMC1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping.....	3415
18-86. SDMMC5 Signal Group Parameter Controls to Different Interface I/O Pads Mapping.....	3416
18-87. ABE McBSP1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3416
18-88. ABE McBSP2 Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3417
18-89. ABE McPDM Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3417
18-90. ABE DMIC Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3417
18-91. ABE Shared Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3417
18-92. UART2 Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3418
18-93. UART3 Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3418
18-94. UART4 Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3418
18-95. McSPI1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping.....	3418
18-96. McSPI4 Signal Group Parameter Controls to Different Interface I/O Pads Mapping.....	3419
18-97. I2C1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3419
18-98. I2C2 Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3420
18-99. I2C3 Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3420
18-100. I2C4 Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3421
18-101. SR Signal Group Parameter Controls to Different Interface I/O Pads Mapping .....	3421
18-102. System Signals Group Parameter Controls to Different Interface I/O Pads Mapping .....	3422
18-103. DPM Signals Group Parameter Controls to Different Interface I/O Pads Mapping .....	3423
18-104. Global Initialization of Surrounding Modules .....	3425
18-105. Configure Clock Gating for Different Control Module Instances .....	3425
18-106. Control Module AVDAC Configuration .....	3426
18-107. Register Settings for Configuring the MMC1_PBIAS and MMC1 I/O Cells .....	3427
18-108. Register Settings for Configuring the PBIAS1 Cell, USBC1_ICUSB and gpio_wk0gpio_wk2 Associated I/O Cells.....	3428
18-109. Configure Hardware Observability Features.....	3429
18-110. Setting PADS Off Mode Through Padconfiguration Registers .....	3430
18-111. ACTIVE PADS Configuration Points.....	3431
18-112. Control Module Instance Summary .....	3432
18-113. SYSCTRL_GENERAL_CORE Register Mapping Summary .....	3432
18-114. CONTROL_GEN_CORE_REVISION .....	3435
18-115. Register Call Summary for Register CONTROL_GEN_CORE_REVISION .....	3435
18-116. CONTROL_GEN_CORE_HWINFO .....	3435
18-117. Register Call Summary for Register CONTROL_GEN_CORE_HWINFO .....	3436
18-118. CONTROL_GEN_CORE_SYSCONFIG .....	3436
18-119. Register Call Summary for Register CONTROL_GEN_CORE_SYSCONFIG .....	3436
18-120. CONTROL_STD_FUSE_DIE_ID_0.....	3436
18-121. Register Call Summary for Register CONTROL_STD_FUSE_DIE_ID_0 .....	3437
18-122. CONTROL_ID_CODE .....	3437

18-123. Register Call Summary for Register CONTROL_ID_CODE .....	3437
18-124. CONTROL_STD_FUSE_DIE_ID_1.....	3437
18-125. Register Call Summary for Register CONTROL_STD_FUSE_DIE_ID_1 .....	3437
18-126. CONTROL_STD_FUSE_DIE_ID_2.....	3438
18-127. Register Call Summary for Register CONTROL_STD_FUSE_DIE_ID_2 .....	3438
18-128. CONTROL_STD_FUSE_DIE_ID_3.....	3438
18-129. Register Call Summary for Register CONTROL_STD_FUSE_DIE_ID_3 .....	3438
18-130. CONTROL_STD_FUSE_PROD_ID_0 .....	3438
18-131. Register Call Summary for Register CONTROL_STD_FUSE_PROD_ID_0.....	3439
18-132. CONTROL_STD_FUSE_PROD_ID_1 .....	3439
18-133. Register Call Summary for Register CONTROL_STD_FUSE_PROD_ID_1 .....	3439
18-134. CONTROL_STD_FUSE_USB_CONF.....	3439
18-135. Register Call Summary for Register CONTROL_STD_FUSE_USB_CONF .....	3439
18-136. CONTROL_STD_FUSE_OPP_VDD_WKUP .....	3440
18-137. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_WKUP .....	3440
18-138. CONTROL_STD_FUSE_OPP_VDD_IVA_0.....	3440
18-139. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_IVA_0 .....	3440
18-140. CONTROL_STD_FUSE_OPP_VDD_IVA_1.....	3441
18-141. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_IVA_1 .....	3441
18-142. CONTROL_STD_FUSE_OPP_VDD_IVA_2.....	3441
18-143. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_IVA_2 .....	3442
18-144. CONTROL_STD_FUSE_OPP_VDD_IVA_3.....	3442
18-145. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_IVA_3 .....	3443
18-146. CONTROL_STD_FUSE_OPP_VDD_IVA_4.....	3443
18-147. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_IVA_4 .....	3443
18-148. CONTROL_STD_FUSE_OPP_VDD_MPU_0.....	3443
18-149. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_MPU_0 .....	3444
18-150. CONTROL_STD_FUSE_OPP_VDD_MPU_1 .....	3444
18-151. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_MPU_1 .....	3444
18-152. CONTROL_STD_FUSE_OPP_VDD_MPU_2.....	3444
18-153. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_MPU_2 .....	3445
18-154. CONTROL_STD_FUSE_OPP_VDD_MPU_3.....	3445
18-155. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_MPU_3 .....	3446
18-156. CONTROL_STD_FUSE_OPP_VDD_MPU_4.....	3446
18-157. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_MPU_4 .....	3446
18-158. CONTROL_STD_FUSE_OPP_VDD_CORE_0 .....	3446
18-159. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_CORE_0.....	3447
18-160. CONTROL_STD_FUSE_OPP_VDD_CORE_1 .....	3447
18-161. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_CORE_1.....	3448
18-162. CONTROL_STD_FUSE_OPP_VDD_CORE_2 .....	3448
18-163. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_CORE_2.....	3448
18-164. CONTROL_STD_FUSE_OPP_BGAP.....	3449
18-165. Register Call Summary for Register CONTROL_STD_FUSE_OPP_BGAP .....	3449
18-166. CONTROL_STD_FUSE_OPP_DPLL_0 .....	3449
18-167. Register Call Summary for Register CONTROL_STD_FUSE_OPP_DPLL_0.....	3449
18-168. CONTROL_STD_FUSE_OPP_DPLL_1 .....	3449
18-169. Register Call Summary for Register CONTROL_STD_FUSE_OPP_DPLL_1 .....	3450
18-170. CONTROL_STATUS .....	3450
18-171. Register Call Summary for Register CONTROL_STATUS.....	3450

18-172. CONTROL_SEC_ERR_STATUS_FUNC .....	3451
18-173. Register Call Summary for Register CONTROL_SEC_ERR_STATUS_FUNC.....	3452
18-174. CONTROL_SEC_ERR_STATUS_DEBUG .....	3452
18-175. Register Call Summary for Register CONTROL_SEC_ERR_STATUS_DEBUG .....	3454
18-176. CONTROL_STD_FUSE_OPP_VDD_MPU_5 .....	3454
18-177. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_MPU_5 .....	3454
18-178. CONTROL_STD_FUSE_OPP_VDD_IVA_5.....	3455
18-179. Register Call Summary for Register CONTROL_STD_FUSE_OPP_VDD_IVA_5 .....	3455
18-180. CONTROL_DEV_CONF .....	3455
18-181. Register Call Summary for Register CONTROL_DEV_CONF.....	3456
18-182. CONTROL_LDOVBB_IVA_VOLTAGE_CTRL .....	3456
18-183. Register Call Summary for Register CONTROL_LDOVBB_IVA_VOLTAGE_CTRL.....	3456
18-184. CONTROL_LDOVBB_MPU_VOLTAGE_CTRL.....	3457
18-185. Register Call Summary for Register CONTROL_LDOVBB_MPU_VOLTAGE_CTRL .....	3457
18-186. CONTROL_LDOSRAM_IVA_VOLTAGE_CTRL .....	3457
18-187. Register Call Summary for Register CONTROL_LDOSRAM_IVA_VOLTAGE_CTRL .....	3458
18-188. CONTROL_LDOSRAM_MPU_VOLTAGE_CTRL .....	3458
18-189. Register Call Summary for Register CONTROL_LDOSRAM_MPU_VOLTAGE_CTRL.....	3459
18-190. CONTROL_LDOSRAM_CORE_VOLTAGE_CTRL .....	3459
18-191. Register Call Summary for Register CONTROL_LDOSRAM_CORE_VOLTAGE_CTRL .....	3460
18-192. CONTROL_TEMP_SENSOR .....	3460
18-193. Register Call Summary for Register CONTROL_TEMP_SENSOR .....	3461
18-194. CONTROL_DPLL_NWELL_TRIM_0 .....	3462
18-195. Register Call Summary for Register CONTROL_DPLL_NWELL_TRIM_0.....	3462
18-196. CONTROL_DPLL_NWELL_TRIM_1 .....	3463
18-197. Register Call Summary for Register CONTROL_DPLL_NWELL_TRIM_1.....	3463
18-198. CONTROL_USBOTGHS_CONTROL .....	3464
18-199. Register Call Summary for Register CONTROL_USBOTGHS_CONTROL.....	3465
18-200. CONTROL_DSS_CONTROL.....	3465
18-201. Register Call Summary for Register CONTROL_DSS_CONTROL .....	3465
18-202. CONTROL_CORTEX_M3_MMUADDRTRANSLTR.....	3465
18-203. Register Call Summary for Register CONTROL_CORTEX_M3_MMUADDRTRANSLTR .....	3466
18-204. CONTROL_CORTEX_M3_MMUADDRLOGICTR .....	3466
18-205. Register Call Summary for Register CONTROL_CORTEX_M3_MMUADDRLOGICTR .....	3466
18-206. CONTROL_HWOBS_CONTROL .....	3466
18-207. Register Call Summary for Register CONTROL_HWOBS_CONTROL.....	3467
18-208. CONTROL_DEBOBS_FINAL_MUX_SEL .....	3468
18-209. Register Call Summary for Register CONTROL_DEBOBS_FINAL_MUX_SEL .....	3468
18-210. CONTROL_DEBOBS_MMR_MPU .....	3468
18-211. Register Call Summary for Register CONTROL_DEBOBS_MMR_MPU.....	3468
18-212. CONTROL_CONF_SDMA_REQ_SEL0.....	3469
18-213. Register Call Summary for Register CONTROL_CONF_SDMA_REQ_SEL0 .....	3471
18-214. CONTROL_CONF_SDMA_REQ_SEL1.....	3472
18-215. Register Call Summary for Register CONTROL_CONF_SDMA_REQ_SEL1 .....	3474
18-216. CONTROL_CONF_SDMA_REQ_SEL2.....	3475
18-217. Register Call Summary for Register CONTROL_CONF_SDMA_REQ_SEL2 .....	3477
18-218. CONTROL_CONF_SDMA_REQ_SEL3.....	3478
18-219. Register Call Summary for Register CONTROL_CONF_SDMA_REQ_SEL3 .....	3480
18-220. CONTROL_CONF_CLK_SEL0 .....	3481



18-221. Register Call Summary for Register CONTROL_CONF_CLK_SEL0 .....	3481
18-222. CONTROL_CONF_CLK_SEL1 .....	3481
18-223. Register Call Summary for Register CONTROL_CONF_CLK_SEL1 .....	3482
18-224. CONTROL_CONF_CLK_SEL2 .....	3482
18-225. Register Call Summary for Register CONTROL_CONF_CLK_SEL2 .....	3482
18-226. CONTROL_CONF_DPLL_FREQLOCK_SEL .....	3482
18-227. Register Call Summary for Register CONTROL_CONF_DPLL_FREQLOCK_SEL .....	3483
18-228. CONTROL_CONF_DPLL_TINITZ_SEL .....	3483
18-229. Register Call Summary for Register CONTROL_CONF_DPLL_TINITZ_SEL .....	3483
18-230. CONTROL_CONF_DPLL_PHASELOCK_SEL .....	3484
18-231. Register Call Summary for Register CONTROL_CONF_DPLL_PHASELOCK_SEL .....	3484
18-232. CONTROL_CONF_DPLL_TENABLE_SEL .....	3484
18-233. Register Call Summary for Register CONTROL_CONF_DPLL_TENABLE_SEL .....	3485
18-234. CONTROL_CONF_DPLL_TENABLEDIV_SEL .....	3485
18-235. Register Call Summary for Register CONTROL_CONF_DPLL_TENABLEDIV_SEL .....	3485
18-236. CONTROL_CONF_DPLL_BYPASSACK_SEL .....	3486
18-237. Register Call Summary for Register CONTROL_CONF_DPLL_BYPASSACK_SEL .....	3486
18-238. CONTROL_CONF_DPLL_IDLE_SEL .....	3486
18-239. Register Call Summary for Register CONTROL_CONF_DPLL_IDLE_SEL .....	3487
18-240. CONTROL_CORE_CONF_DEBUG_SEL_TST_0 .....	3487
18-241. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_0 .....	3487
18-242. CONTROL_CORE_CONF_DEBUG_SEL_TST_1 .....	3488
18-243. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_1 .....	3488
18-244. CONTROL_CORE_CONF_DEBUG_SEL_TST_2 .....	3488
18-245. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_2 .....	3489
18-246. CONTROL_CORE_CONF_DEBUG_SEL_TST_3 .....	3489
18-247. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_3 .....	3489
18-248. CONTROL_CORE_CONF_DEBUG_SEL_TST_4 .....	3490
18-249. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_4 .....	3490
18-250. CONTROL_CORE_CONF_DEBUG_SEL_TST_5 .....	3490
18-251. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_5 .....	3491
18-252. CONTROL_CORE_CONF_DEBUG_SEL_TST_6 .....	3491
18-253. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_6 .....	3491
18-254. CONTROL_CORE_CONF_DEBUG_SEL_TST_7 .....	3492
18-255. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_7 .....	3492
18-256. CONTROL_CORE_CONF_DEBUG_SEL_TST_8 .....	3492
18-257. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_8 .....	3493
18-258. CONTROL_CORE_CONF_DEBUG_SEL_TST_9 .....	3493
18-259. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_9 .....	3494
18-260. CONTROL_CORE_CONF_DEBUG_SEL_TST_10 .....	3494
18-261. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_10 .....	3494
18-262. CONTROL_CORE_CONF_DEBUG_SEL_TST_11 .....	3494
18-263. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_11 .....	3495
18-264. CONTROL_CORE_CONF_DEBUG_SEL_TST_12 .....	3495
18-265. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_12 .....	3496
18-266. CONTROL_CORE_CONF_DEBUG_SEL_TST_13 .....	3496
18-267. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_13 .....	3496
18-268. CONTROL_CORE_CONF_DEBUG_SEL_TST_14 .....	3496
18-269. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_14 .....	3497

18-270. CONTROL_CORE_CONF_DEBUG_SEL_TST_15 .....	3497
18-271. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_15.....	3498
18-272. CONTROL_CORE_CONF_DEBUG_SEL_TST_16 .....	3498
18-273. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_16.....	3498
18-274. CONTROL_CORE_CONF_DEBUG_SEL_TST_17 .....	3498
18-275. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_17.....	3499
18-276. CONTROL_CORE_CONF_DEBUG_SEL_TST_18 .....	3499
18-277. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_18.....	3500
18-278. CONTROL_CORE_CONF_DEBUG_SEL_TST_19 .....	3500
18-279. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_19.....	3500
18-280. CONTROL_CORE_CONF_DEBUG_SEL_TST_20 .....	3500
18-281. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_20.....	3501
18-282. CONTROL_CORE_CONF_DEBUG_SEL_TST_21 .....	3501
18-283. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_21.....	3502
18-284. CONTROL_CORE_CONF_DEBUG_SEL_TST_22 .....	3502
18-285. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_22.....	3502
18-286. CONTROL_CORE_CONF_DEBUG_SEL_TST_23 .....	3502
18-287. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_23.....	3503
18-288. CONTROL_CORE_CONF_DEBUG_SEL_TST_24 .....	3503
18-289. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_24.....	3504
18-290. CONTROL_CORE_CONF_DEBUG_SEL_TST_25 .....	3504
18-291. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_25.....	3504
18-292. CONTROL_CORE_CONF_DEBUG_SEL_TST_26 .....	3504
18-293. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_26.....	3505
18-294. CONTROL_CORE_CONF_DEBUG_SEL_TST_27 .....	3505
18-295. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_27.....	3506
18-296. CONTROL_CORE_CONF_DEBUG_SEL_TST_28 .....	3506
18-297. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_28.....	3506
18-298. CONTROL_CORE_CONF_DEBUG_SEL_TST_29 .....	3507
18-299. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_29.....	3507
18-300. CONTROL_CORE_CONF_DEBUG_SEL_TST_30 .....	3507
18-301. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_30.....	3508
18-302. CONTROL_CORE_CONF_DEBUG_SEL_TST_31 .....	3508
18-303. Register Call Summary for Register CONTROL_CORE_CONF_DEBUG_SEL_TST_31.....	3509
18-304. SYSCTRL_GENERAL_WKUP Register Mapping Summary .....	3509
18-305. CONTROL_GEN_WKUP_REVISION .....	3509
18-306. Register Call Summary for Register CONTROL_GEN_WKUP_REVISION.....	3510
18-307. CONTROL_GEN_WKUP_HWINFO .....	3510
18-308. Register Call Summary for Register CONTROL_GEN_WKUP_HWINFO.....	3510
18-309. CONTROL_GEN_WKUP_SYSCONFIG .....	3510
18-310. Register Call Summary for Register CONTROL_GEN_WKUP_SYSCONFIG.....	3511
18-311. CONTROL_OCPREG_SPARE.....	3511
18-312. Register Call Summary for Register CONTROL_OCPREG_SPARE .....	3512
18-313. CONTROL_WKUP_CONF_DEBUG_SEL_TST_i .....	3512
18-314. Register Call Summary for Register CONTROL_WKUP_CONF_DEBUG_SEL_TST_i.....	3512
18-315. SYSCTRL_PADCONF_CORE Register Mapping Summary .....	3512
18-316. CONTROL_PADCONF_CORE_REVISION .....	3518
18-317. Register Call Summary for Register CONTROL_PADCONF_CORE_REVISION.....	3519
18-318. CONTROL_PADCONF_CORE_HWINFO .....	3519

18-319. Register Call Summary for Register CONTROL_PADCONF_CORE_HWINFO .....	3519
18-320. CONTROL_PADCONF_CORE_SYSCONFIG .....	3519
18-321. Register Call Summary for Register CONTROL_PADCONF_CORE_SYSCONFIG.....	3520
18-322. CONTROL_CORE_PAD0_GPMC_AD0_PAD1_GPMC_AD1 .....	3520
18-323. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_AD0_PAD1_GPMC_AD1 .....	3522
18-324. CONTROL_CORE_PAD0_GPMC_AD2_PAD1_GPMC_AD3 .....	3522
18-325. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_AD2_PAD1_GPMC_AD3.....	3524
18-326. CONTROL_CORE_PAD0_GPMC_AD4_PAD1_GPMC_AD5 .....	3524
18-327. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_AD4_PAD1_GPMC_AD5.....	3526
18-328. CONTROL_CORE_PAD0_GPMC_AD6_PAD1_GPMC_AD7 .....	3526
18-329. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_AD6_PAD1_GPMC_AD7.....	3528
18-330. CONTROL_CORE_PAD0_GPMC_AD8_PAD1_GPMC_AD9 .....	3529
18-331. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_AD8_PAD1_GPMC_AD9.....	3531
18-332. CONTROL_CORE_PAD0_GPMC_AD10_PAD1_GPMC_AD11 .....	3531
18-333. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_AD10_PAD1_GPMC_AD11 ....	3533
18-334. CONTROL_CORE_PAD0_GPMC_AD12_PAD1_GPMC_AD13 .....	3533
18-335. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_AD12_PAD1_GPMC_AD13....	3535
18-336. CONTROL_CORE_PAD0_GPMC_AD14_PAD1_GPMC_AD15 .....	3535
18-337. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_AD14_PAD1_GPMC_AD15....	3538
18-338. CONTROL_CORE_PAD0_GPMC_A16_PAD1_GPMC_A17 .....	3538
18-339. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_A16_PAD1_GPMC_A17.....	3540
18-340. CONTROL_CORE_PAD0_GPMC_A18_PAD1_GPMC_A19 .....	3540
18-341. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_A18_PAD1_GPMC_A19.....	3542
18-342. CONTROL_CORE_PAD0_GPMC_A20_PAD1_GPMC_A21 .....	3543
18-343. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_A20_PAD1_GPMC_A21 .....	3545
18-344. CONTROL_CORE_PAD0_GPMC_A22_PAD1_GPMC_A23 .....	3545
18-345. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_A22_PAD1_GPMC_A23.....	3547
18-346. CONTROL_CORE_PAD0_GPMC_A24_PAD1_GPMC_A25 .....	3547
18-347. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_A24_PAD1_GPMC_A25.....	3550
18-348. CONTROL_CORE_PAD0_GPMC_NCS0_PAD1_GPMC_NCS1 .....	3550
18-349. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_NCS0_PAD1_GPMC_NCS1 ...	3552
18-350. CONTROL_CORE_PAD0_GPMC_NCS2_PAD1_GPMC_NCS3 .....	3552
18-351. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_NCS2_PAD1_GPMC_NCS3...	3554
18-352. CONTROL_CORE_PAD0_GPMC_NWP_PAD1_GPMC_CLK .....	3554
18-353. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_NWP_PAD1_GPMC_CLK.....	3556
18-354. CONTROL_CORE_PAD0_GPMC_NADV_ALE_PAD1_GPMC_NOE .....	3557
18-355. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_NADV_ALE_PAD1_GPMC_NOE .....	3559
18-356. CONTROL_CORE_PAD0_GPMC_NWE_PAD1_GPMC_NBE0_CLE.....	3559
18-357. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_NWE_PAD1_GPMC_NBE0_CLE .....	3561
18-358. CONTROL_CORE_PAD0_GPMC_NBE1_PAD1_GPMC_WAIT0.....	3561
18-359. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_NBE1_PAD1_GPMC_WAIT0 ..	3563
18-360. CONTROL_CORE_PAD0_GPMC_WAIT1_PAD1_GPMC_WAIT2 .....	3564
18-361. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_WAIT1_PAD1_GPMC_WAIT2.	3566
18-362. CONTROL_CORE_PAD0_GPMC_NCS4_PAD1_GPMC_NCS5 .....	3566
18-363. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_NCS4_PAD1_GPMC_NCS5 ...	3568
18-364. CONTROL_CORE_PAD0_GPMC_NCS6_PAD1_GPMC_NCS7 .....	3569
18-365. Register Call Summary for Register CONTROL_CORE_PAD0_GPMC_NCS6_PAD1_GPMC_NCS7 ...	3571

18-366. CONTROL_CORE_PAD0_GPIO63_PAD1_GPIO64.....	3571
18-367. Register Call Summary for Register CONTROL_CORE_PAD0_GPIO63_PAD1_GPIO64 .....	3573
18-368. CONTROL_CORE_PAD0_GPIO65_PAD1_GPIO66.....	3573
18-369. Register Call Summary for Register CONTROL_CORE_PAD0_GPIO65_PAD1_GPIO66 .....	3575
18-370. CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0.....	3576
18-371. Register Call Summary for Register CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0 .....	3577
18-372. CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1.....	3577
18-373. Register Call Summary for Register CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1 .....	3578
18-374. CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2.....	3579
18-375. Register Call Summary for Register CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2 .....	3580
18-376. CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3.....	3580
18-377. Register Call Summary for Register CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3 .....	3581
18-378. CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4.....	3582
18-379. Register Call Summary for Register CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4 .....	3583
18-380. CONTROL_CORE_PAD0_CSI22_DX0_PAD1_CSI22_DY0.....	3583
18-381. Register Call Summary for Register CONTROL_CORE_PAD0_CSI22_DX0_PAD1_CSI22_DY0 .....	3584
18-382. CONTROL_CORE_PAD0_CSI22_DX1_PAD1_CSI22_DY1.....	3585
18-383. Register Call Summary for Register CONTROL_CORE_PAD0_CSI22_DX1_PAD1_CSI22_DY1 .....	3586
18-384. CONTROL_CORE_PAD0_CAM_SHUTTER_PAD1_CAM_STROBE .....	3586
18-385. Register Call Summary for Register CONTROL_CORE_PAD0_CAM_SHUTTER_PAD1_CAM_STROBE.....	3588
18-386. CONTROL_CORE_PAD0_CAM_GLOBALRESET_PAD1_USBB1_ULPITLL_CLK .....	3588
18-387. Register Call Summary for Register CONTROL_CORE_PAD0_CAM_GLOBALRESET_PAD1_USBB1_ULPITLL_CLK .....	3591
18-388. CONTROL_CORE_PAD0_USBB1_ULPITLL_STP_PAD1_USBB1_ULPITLL_DIR .....	3591
18-389. Register Call Summary for Register CONTROL_CORE_PAD0_USBB1_ULPITLL_STP_PAD1_USBB1_ULPITLL_DIR.....	3593
18-390. CONTROL_CORE_PAD0_USBB1_ULPITLL_NXT_PAD1_USBB1_ULPITLL_DAT0.....	3593
18-391. Register Call Summary for Register CONTROL_CORE_PAD0_USBB1_ULPITLL_NXT_PAD1_USBB1_ULPITLL_DAT0 .....	3596
18-392. CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT1_PAD1_USBB1_ULPITLL_DAT2 .....	3596
18-393. Register Call Summary for Register CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT1_PAD1_USBB1_ULPITLL_DAT2.....	3598
18-394. CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT3_PAD1_USBB1_ULPITLL_DAT4 .....	3599
18-395. Register Call Summary for Register CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT3_PAD1_USBB1_ULPITLL_DAT4.....	3601
18-396. CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT5_PAD1_USBB1_ULPITLL_DAT6 .....	3601
18-397. Register Call Summary for Register CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT5_PAD1_USBB1_ULPITLL_DAT6.....	3604
18-398. CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT7_PAD1_USBB1_HSIC_DATA .....	3604
18-399. Register Call Summary for Register CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT7_PAD1_USBB1_HSIC_DATA.....	3606
18-400. CONTROL_CORE_PAD0_USBB1_HSIC_STROBE_PAD1_USBC1_ICUSB_DP.....	3606
18-401. Register Call Summary for Register CONTROL_CORE_PAD0_USBB1_HSIC_STROBE_PAD1_USBC1_ICUSB_DP .....	3608
18-402. CONTROL_CORE_PAD0_USBC1_ICUSB_DM_PAD1_SDMMC1_CLK .....	3609
18-403. Register Call Summary for Register CONTROL_CORE_PAD0_USBC1_ICUSB_DM_PAD1_SDMMC1_CLK.....	3611
18-404. CONTROL_CORE_PAD0_SDMMC1_CMD_PAD1_SDMMC1_DAT0.....	3611
18-405. Register Call Summary for Register CONTROL_CORE_PAD0_SDMMC1_CMD_PAD1_SDMMC1_DAT0 .....	3613
18-406. CONTROL_CORE_PAD0_SDMMC1_DAT1_PAD1_SDMMC1_DAT2.....	3613

18-407. Register Call Summary for Register CONTROL_CORE_PAD0_SDMMC1_DAT1_PAD1_SDMMC1_DAT2 .....	3615
18-408. CONTROL_CORE_PAD0_SDMMC1_DAT3_PAD1_SDMMC1_DAT4 .....	3616
18-409. Register Call Summary for Register CONTROL_CORE_PAD0_SDMMC1_DAT3_PAD1_SDMMC1_DAT4 .....	3618
18-410. CONTROL_CORE_PAD0_SDMMC1_DAT5_PAD1_SDMMC1_DAT6 .....	3618
18-411. Register Call Summary for Register CONTROL_CORE_PAD0_SDMMC1_DAT5_PAD1_SDMMC1_DAT6 .....	3620
18-412. CONTROL_CORE_PAD0_SDMMC1_DAT7_PAD1_ABE_MCBSP2_CLKX .....	3620
18-413. Register Call Summary for Register CONTROL_CORE_PAD0_SDMMC1_DAT7_PAD1_ABE_MCBSP2_CLKX .....	3622
18-414. CONTROL_CORE_PAD0_ABE_MCBSP2_DR_PAD1_ABE_MCBSP2_DX .....	3623
18-415. Register Call Summary for Register CONTROL_CORE_PAD0_ABE_MCBSP2_DR_PAD1_ABE_MCBSP2_DX .....	3625
18-416. CONTROL_CORE_PAD0_ABE_MCBSP2_FSX_PAD1_ABE_MCBSP1_CLKX .....	3625
18-417. Register Call Summary for Register CONTROL_CORE_PAD0_ABE_MCBSP2_FSX_PAD1_ABE_MCBSP1_CLKX .....	3627
18-418. CONTROL_CORE_PAD0_ABE_MCBSP1_DR_PAD1_ABE_MCBSP1_DX .....	3628
18-419. Register Call Summary for Register CONTROL_CORE_PAD0_ABE_MCBSP1_DR_PAD1_ABE_MCBSP1_DX .....	3630
18-420. CONTROL_CORE_PAD0_ABE_MCBSP1_FSX_PAD1_ABE_PDM_UL_DATA .....	3630
18-421. Register Call Summary for Register CONTROL_CORE_PAD0_ABE_MCBSP1_FSX_PAD1_ABE_PDM_UL_DATA .....	3632
18-422. CONTROL_CORE_PAD0_ABE_PDM_DL_DATA_PAD1_ABE_PDM_FRAME .....	3632
18-423. Register Call Summary for Register CONTROL_CORE_PAD0_ABE_PDM_DL_DATA_PAD1_ABE_PDM_FRAME .....	3634
18-424. CONTROL_CORE_PAD0_ABE_PDM_LB_CLK_PAD1_ABE_CLKS .....	3634
18-425. Register Call Summary for Register CONTROL_CORE_PAD0_ABE_PDM_LB_CLK_PAD1_ABE_CLKS .....	3637
18-426. CONTROL_CORE_PAD0_ABE_DMIC_CLK1_PAD1_ABE_DMIC_DIN1 .....	3637
18-427. Register Call Summary for Register CONTROL_CORE_PAD0_ABE_DMIC_CLK1_PAD1_ABE_DMIC_DIN1 .....	3639
18-428. CONTROL_CORE_PAD0_ABE_DMIC_DIN2_PAD1_ABE_DMIC_DIN3 .....	3639
18-429. Register Call Summary for Register CONTROL_CORE_PAD0_ABE_DMIC_DIN2_PAD1_ABE_DMIC_DIN3 .....	3641
18-430. CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS .....	3642
18-431. Register Call Summary for Register CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS .....	3644
18-432. CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX .....	3644
18-433. Register Call Summary for Register CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX .....	3646
18-434. CONTROL_CORE_PAD0_HDQ_SIO_PAD1_I2C1_SCL .....	3646
18-435. Register Call Summary for Register CONTROL_CORE_PAD0_HDQ_SIO_PAD1_I2C1_SCL .....	3648
18-436. CONTROL_CORE_PAD0_I2C1_SDA_PAD1_I2C2_SCL .....	3648
18-437. Register Call Summary for Register CONTROL_CORE_PAD0_I2C1_SDA_PAD1_I2C2_SCL .....	3650
18-438. CONTROL_CORE_PAD0_I2C2_SDA_PAD1_I2C3_SCL .....	3650
18-439. Register Call Summary for Register CONTROL_CORE_PAD0_I2C2_SDA_PAD1_I2C3_SCL .....	3652
18-440. CONTROL_CORE_PAD0_I2C3_SDA_PAD1_I2C4_SCL .....	3652
18-441. Register Call Summary for Register CONTROL_CORE_PAD0_I2C3_SDA_PAD1_I2C4_SCL .....	3654
18-442. CONTROL_CORE_PAD0_I2C4_SDA_PAD1_MCSP11_CLK .....	3654
18-443. Register Call Summary for Register CONTROL_CORE_PAD0_I2C4_SDA_PAD1_MCSP11_CLK .....	3656
18-444. CONTROL_CORE_PAD0_MCSP11_SOMI_PAD1_MCSP11_SIMO .....	3656
18-445. Register Call Summary for Register CONTROL_CORE_PAD0_MCSP11_SOMI_PAD1_MCSP11_SIMO .....	3659

18-446. CONTROL_CORE_PAD0_MCSP1_CS0_PAD1_MCSP1_CS1 .....	3659
18-447. Register Call Summary for Register CONTROL_CORE_PAD0_MCSP1_CS0_PAD1_MCSP1_CS1 ...	3661
18-448. CONTROL_CORE_PAD0_MCSP1_CS2_PAD1_MCSP1_CS3 .....	3661
18-449. Register Call Summary for Register CONTROL_CORE_PAD0_MCSP1_CS2_PAD1_MCSP1_CS3 ...	3663
18-450. CONTROL_CORE_PAD0_UART3_CTS_RCTX_PAD1_UART3_RTS_SD .....	3663
18-451. Register Call Summary for Register CONTROL_CORE_PAD0_UART3_CTS_RCTX_PAD1_UART3_RTS_SD .....	3666
18-452. CONTROL_CORE_PAD0_UART3_RX_IRRX_PAD1_UART3_TX_IRTX.....	3666
18-453. Register Call Summary for Register CONTROL_CORE_PAD0_UART3_RX_IRRX_PAD1_UART3_TX_IRTX.....	3668
18-454. CONTROL_CORE_PAD0_SDMC5_CLK_PAD1_SDMC5_CMD .....	3668
18-455. Register Call Summary for Register CONTROL_CORE_PAD0_SDMC5_CLK_PAD1_SDMC5_CMD.....	3670
18-456. CONTROL_CORE_PAD0_SDMC5_DAT0_PAD1_SDMC5_DAT1 .....	3671
18-457. Register Call Summary for Register CONTROL_CORE_PAD0_SDMC5_DAT0_PAD1_SDMC5_DAT1 .....	3673
18-458. CONTROL_CORE_PAD0_SDMC5_DAT2_PAD1_SDMC5_DAT3.....	3673
18-459. Register Call Summary for Register CONTROL_CORE_PAD0_SDMC5_DAT2_PAD1_SDMC5_DAT3 .....	3675
18-460. CONTROL_CORE_PAD0_MCSP4_CLK_PAD1_MCSP4_SIMO .....	3675
18-461. Register Call Summary for Register CONTROL_CORE_PAD0_MCSP4_CLK_PAD1_MCSP4_SIMO .	3678
18-462. CONTROL_CORE_PAD0_MCSP4_SOMI_PAD1_MCSP4_CS0.....	3678
18-463. Register Call Summary for Register CONTROL_CORE_PAD0_MCSP4_SOMI_PAD1_MCSP4_CS0 .	3680
18-464. CONTROL_CORE_PAD0_UART4_RX_PAD1_UART4_TX .....	3680
18-465. Register Call Summary for Register CONTROL_CORE_PAD0_UART4_RX_PAD1_UART4_TX.....	3682
18-466. CONTROL_CORE_PAD0_USBB2_ULPITLL_CLK_PAD1_USBB2_ULPITLL_STP.....	3682
18-467. Register Call Summary for Register CONTROL_CORE_PAD0_USBB2_ULPITLL_CLK_PAD1_USBB2_ULPITLL_STP .....	3685
18-468. CONTROL_CORE_PAD0_USBB2_ULPITLL_DIR_PAD1_USBB2_ULPITLL_NXT .....	3685
18-469. Register Call Summary for Register CONTROL_CORE_PAD0_USBB2_ULPITLL_DIR_PAD1_USBB2_ULPITLL_NXT .....	3687
18-470. CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT0_PAD1_USBB2_ULPITLL_DAT1 .....	3688
18-471. Register Call Summary for Register CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT0_PAD1_USBB2_ULPITLL_DAT1.....	3690
18-472. CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT2_PAD1_USBB2_ULPITLL_DAT3 .....	3690
18-473. Register Call Summary for Register CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT2_PAD1_USBB2_ULPITLL_DAT3.....	3693
18-474. CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT4_PAD1_USBB2_ULPITLL_DAT5 .....	3693
18-475. Register Call Summary for Register CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT4_PAD1_USBB2_ULPITLL_DAT5.....	3695
18-476. CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT6_PAD1_USBB2_ULPITLL_DAT7 .....	3696
18-477. Register Call Summary for Register CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT6_PAD1_USBB2_ULPITLL_DAT7.....	3698
18-478. CONTROL_CORE_PAD0_USBB2_HSIC_DATA_PAD1_USBB2_HSIC_STROBE .....	3698
18-479. Register Call Summary for Register CONTROL_CORE_PAD0_USBB2_HSIC_DATA_PAD1_USBB2_HSIC_STROBE.....	3700
18-480. CONTROL_CORE_PAD0_KPD_COL3_PAD1_KPD_COL4.....	3700
18-481. Register Call Summary for Register CONTROL_CORE_PAD0_KPD_COL3_PAD1_KPD_COL4 .....	3702
18-482. CONTROL_CORE_PAD0_KPD_COL5_PAD1_KPD_COL0.....	3703
18-483. Register Call Summary for Register CONTROL_CORE_PAD0_KPD_COL5_PAD1_KPD_COL0 .....	3705
18-484. CONTROL_CORE_PAD0_KPD_COL1_PAD1_KPD_COL2.....	3705
18-485. Register Call Summary for Register CONTROL_CORE_PAD0_KPD_COL1_PAD1_KPD_COL2 .....	3707

18-486. CONTROL_CORE_PAD0_KPD_ROW3_PAD1_KPD_ROW4.....	3708
18-487. Register Call Summary for Register CONTROL_CORE_PAD0_KPD_ROW3_PAD1_KPD_ROW4 .....	3709
18-488. CONTROL_CORE_PAD0_KPD_ROW5_PAD1_KPD_ROW0.....	3709
18-489. Register Call Summary for Register CONTROL_CORE_PAD0_KPD_ROW5_PAD1_KPD_ROW0 .....	3711
18-490. CONTROL_CORE_PAD0_KPD_ROW1_PAD1_KPD_ROW2.....	3711
18-491. Register Call Summary for Register CONTROL_CORE_PAD0_KPD_ROW1_PAD1_KPD_ROW2 .....	3712
18-492. CONTROL_CORE_PAD0_USBA0_OTG_CE_PAD1_USBA0_OTG_DP .....	3713
18-493. Register Call Summary for Register CONTROL_CORE_PAD0_USBA0_OTG_CE_PAD1_USBA0_OTG_DP .....	3714
18-494. CONTROL_CORE_PAD0_USBA0_OTG_DM_PAD1_FREF_CLK1_OUT.....	3714
18-495. Register Call Summary for Register CONTROL_CORE_PAD0_USBA0_OTG_DM_PAD1_FREF_CLK1_OUT .....	3715
18-496. CONTROL_CORE_PAD0_FREF_CLK2_OUT_PAD1_SYS_NIRQ1 .....	3716
18-497. Register Call Summary for Register CONTROL_CORE_PAD0_FREF_CLK2_OUT_PAD1_SYS_NIRQ1.....	3717
18-498. CONTROL_CORE_PAD0_SYS_NIRQ2_PAD1_SYS_BOOT0.....	3718
18-499. Register Call Summary for Register CONTROL_CORE_PAD0_SYS_NIRQ2_PAD1_SYS_BOOT0 .....	3720
18-500. CONTROL_CORE_PAD0_SYS_BOOT1_PAD1_SYS_BOOT2.....	3720
18-501. Register Call Summary for Register CONTROL_CORE_PAD0_SYS_BOOT1_PAD1_SYS_BOOT2 ....	3722
18-502. CONTROL_CORE_PAD0_SYS_BOOT3_PAD1_SYS_BOOT4.....	3722
18-503. Register Call Summary for Register CONTROL_CORE_PAD0_SYS_BOOT3_PAD1_SYS_BOOT4 ....	3724
18-504. CONTROL_CORE_PAD0_SYS_BOOT5_PAD1_DPM_EMU0 .....	3724
18-505. Register Call Summary for Register CONTROL_CORE_PAD0_SYS_BOOT5_PAD1_DPM_EMU0 .....	3726
18-506. CONTROL_CORE_PAD0_DPM_EMU1_PAD1_DPM_EMU2 .....	3726
18-507. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU1_PAD1_DPM_EMU2.....	3729
18-508. CONTROL_CORE_PAD0_DPM_EMU3_PAD1_DPM_EMU4 .....	3729
18-509. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU3_PAD1_DPM_EMU4.....	3731
18-510. CONTROL_CORE_PAD0_DPM_EMU5_PAD1_DPM_EMU6 .....	3731
18-511. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU5_PAD1_DPM_EMU6.....	3733
18-512. CONTROL_CORE_PAD0_DPM_EMU7_PAD1_DPM_EMU8 .....	3734
18-513. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU7_PAD1_DPM_EMU8.....	3736
18-514. CONTROL_CORE_PAD0_DPM_EMU9_PAD1_DPM_EMU10.....	3736
18-515. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU9_PAD1_DPM_EMU10 .....	3738
18-516. CONTROL_CORE_PAD0_DPM_EMU11_PAD1_DPM_EMU12 .....	3739
18-517. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU11_PAD1_DPM_EMU12....	3741
18-518. CONTROL_CORE_PAD0_DPM_EMU13_PAD1_DPM_EMU14 .....	3741
18-519. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU13_PAD1_DPM_EMU14....	3743
18-520. CONTROL_CORE_PAD0_DPM_EMU15_PAD1_DPM_EMU16 .....	3744
18-521. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU15_PAD1_DPM_EMU16....	3746
18-522. CONTROL_CORE_PAD0_DPM_EMU17_PAD1_DPM_EMU18 .....	3746
18-523. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU17_PAD1_DPM_EMU18....	3748
18-524. CONTROL_CORE_PAD0_DPM_EMU19.....	3749
18-525. Register Call Summary for Register CONTROL_CORE_PAD0_DPM_EMU19 .....	3750
18-526. CONTROL_PADCONF_WAKEUPEVENT_0.....	3750
18-527. Register Call Summary for Register CONTROL_PADCONF_WAKEUPEVENT_0 .....	3753
18-528. CONTROL_PADCONF_WAKEUPEVENT_1.....	3753
18-529. Register Call Summary for Register CONTROL_PADCONF_WAKEUPEVENT_1 .....	3756
18-530. CONTROL_PADCONF_WAKEUPEVENT_2.....	3757
18-531. Register Call Summary for Register CONTROL_PADCONF_WAKEUPEVENT_2 .....	3760
18-532. CONTROL_PADCONF_WAKEUPEVENT_3.....	3760

18-533. Register Call Summary for Register CONTROL_PADCONF_WAKEUPEVENT_3 .....	3762
18-534. CONTROL_PADCONF_WAKEUPEVENT_4.....	3763
18-535. Register Call Summary for Register CONTROL_PADCONF_WAKEUPEVENT_4 .....	3765
18-536. CONTROL_PADCONF_WAKEUPEVENT_5.....	3766
18-537. Register Call Summary for Register CONTROL_PADCONF_WAKEUPEVENT_5 .....	3768
18-538. CONTROL_PADCONF_WAKEUPEVENT_6.....	3769
18-539. Register Call Summary for Register CONTROL_PADCONF_WAKEUPEVENT_6 .....	3770
18-540. CONTROL_PADCONF_GLOBAL .....	3770
18-541. Register Call Summary for Register CONTROL_PADCONF_GLOBAL .....	3770
18-542. CONTROL_CORE_PADCONF_MODE .....	3771
18-543. Register Call Summary for Register CONTROL_CORE_PADCONF_MODE .....	3772
18-544. CONTROL_SMART1IO_PADCONF_0 .....	3772
18-545. Register Call Summary for Register CONTROL_SMART1IO_PADCONF_0 .....	3773
18-546. CONTROL_SMART1IO_PADCONF_1 .....	3773
18-547. Register Call Summary for Register CONTROL_SMART1IO_PADCONF_1 .....	3776
18-548. CONTROL_SMART2IO_PADCONF_0 .....	3776
18-549. Register Call Summary for Register CONTROL_SMART2IO_PADCONF_0 .....	3776
18-550. CONTROL_SMART2IO_PADCONF_1 .....	3776
18-551. Register Call Summary for Register CONTROL_SMART2IO_PADCONF_1 .....	3776
18-552. CONTROL_SMART3IO_PADCONF_0 .....	3777
18-553. Register Call Summary for Register CONTROL_SMART3IO_PADCONF_0 .....	3778
18-554. CONTROL_SMART3IO_PADCONF_1 .....	3778
18-555. Register Call Summary for Register CONTROL_SMART3IO_PADCONF_1 .....	3779
18-556. CONTROL_SMART3IO_PADCONF_2 .....	3779
18-557. Register Call Summary for Register CONTROL_SMART3IO_PADCONF_2 .....	3781
18-558. CONTROL_USBB_HSIC.....	3781
18-559. Register Call Summary for Register CONTROL_USBB_HSIC .....	3784
18-560. CONTROL_SMART3IO_PADCONF_3 .....	3784
18-561. Register Call Summary for Register CONTROL_SMART3IO_PADCONF_3 .....	3785
18-562. CONTROL_SMART2IO_PADCONF_2 .....	3785
18-563. Register Call Summary for Register CONTROL_SMART2IO_PADCONF_2 .....	3787
18-564. CONTROL_SMART1IO_PADCONF_2 .....	3787
18-565. Register Call Summary for Register CONTROL_SMART1IO_PADCONF_2 .....	3788
18-566. CONTROL_SMART1IO_PADCONF_3 .....	3788
18-567. Register Call Summary for Register CONTROL_SMART1IO_PADCONF_3 .....	3789
18-568. CONTROL_C2CIO_PADCONF_0 .....	3789
18-569. Register Call Summary for Register CONTROL_C2CIO_PADCONF_0 .....	3791
18-570. CONTROL_PBIASLITE .....	3792
18-571. Register Call Summary for Register CONTROL_PBIASLITE .....	3793
18-572. CONTROL_I2C_0.....	3793
18-573. Register Call Summary for Register CONTROL_I2C_0 .....	3796
18-574. CONTROL_CAMERA_RX .....	3796
18-575. Register Call Summary for Register CONTROL_CAMERA_RX.....	3797
18-576. CONTROL_AVDAC.....	3797
18-577. Register Call Summary for Register CONTROL_AVDAC .....	3798
18-578. CONTROL_MMC2 .....	3798
18-579. Register Call Summary for Register CONTROL_MMC2 .....	3798
18-580. CONTROL_DSIPHY .....	3798
18-581. Register Call Summary for Register CONTROL_DSIPHY .....	3799



18-582. CONTROL_MCBSPLP .....	3799
18-583. Register Call Summary for Register CONTROL_MCBSPLP.....	3799
18-584. CONTROL_USB2PHYCORE .....	3800
18-585. Register Call Summary for Register CONTROL_USB2PHYCORE .....	3802
18-586. CONTROL_MMC1 .....	3802
18-587. Register Call Summary for Register CONTROL_MMC1 .....	3803
18-588. CONTROL_HSI .....	3803
18-589. Register Call Summary for Register CONTROL_HSI .....	3804
18-590. CONTROL_USB .....	3804
18-591. Register Call Summary for Register CONTROL_USB.....	3805
18-592. CONTROL_HDQ .....	3805
18-593. Register Call Summary for Register CONTROL_HDQ .....	3805
18-594. CONTROL_LPDDR2IO1_0.....	3805
18-595. Register Call Summary for Register CONTROL_LPDDR2IO1_0 .....	3807
18-596. CONTROL_LPDDR2IO1_1.....	3807
18-597. Register Call Summary for Register CONTROL_LPDDR2IO1_1 .....	3809
18-598. CONTROL_LPDDR2IO1_2.....	3810
18-599. Register Call Summary for Register CONTROL_LPDDR2IO1_2 .....	3811
18-600. CONTROL_LPDDR2IO1_3.....	3812
18-601. Register Call Summary for Register CONTROL_LPDDR2IO1_3 .....	3814
18-602. CONTROL_LPDDR2IO2_0.....	3814
18-603. Register Call Summary for Register CONTROL_LPDDR2IO2_0 .....	3816
18-604. CONTROL_LPDDR2IO2_1.....	3817
18-605. Register Call Summary for Register CONTROL_LPDDR2IO2_1 .....	3819
18-606. CONTROL_LPDDR2IO2_2.....	3819
18-607. Register Call Summary for Register CONTROL_LPDDR2IO2_2 .....	3821
18-608. CONTROL_LPDDR2IO2_3.....	3821
18-609. Register Call Summary for Register CONTROL_LPDDR2IO2_3 .....	3823
18-610. CONTROL_BUS_HOLD .....	3824
18-611. Register Call Summary for Register CONTROL_BUS_HOLD .....	3824
18-612. CONTROL_C2C .....	3824
18-613. Register Call Summary for Register CONTROL_C2C .....	3824
18-614. CONTROL_CORE_CONTROL_SPARE_RW.....	3824
18-615. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW .....	3825
18-616. CONTROL_CORE_CONTROL_SPARE_R .....	3825
18-617. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_R.....	3825
18-618. CONTROL_CORE_CONTROL_SPARE_R_C0 .....	3825
18-619. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_R_C0.....	3826
18-620. CONTROL_CORE_CONTROL_SPARE_RW1 .....	3826
18-621. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW1.....	3826
18-622. CONTROL_CORE_CONTROL_SPARE_RW2 .....	3826
18-623. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW2.....	3827
18-624. CONTROL_CORE_CONTROL_SPARE_RW3 .....	3827
18-625. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW3.....	3827
18-626. CONTROL_CORE_CONTROL_SPARE_RW4 .....	3827
18-627. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW4.....	3827
18-628. CONTROL_CORE_CONTROL_SPARE_RW5 .....	3827
18-629. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW5.....	3828
18-630. CONTROL_CORE_CONTROL_SPARE_RW6 .....	3828

18-631. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW6 .....	3828
18-632. CONTROL_CORE_CONTROL_SPARE_RW7 .....	3828
18-633. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW7 .....	3828
18-634. CONTROL_CORE_CONTROL_SPARE_RW8 .....	3829
18-635. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW8 .....	3829
18-636. CONTROL_CORE_CONTROL_SPARE_RW9 .....	3829
18-637. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_RW9 .....	3829
18-638. CONTROL_CORE_CONTROL_SPARE_R1 .....	3829
18-639. Register Call Summary for Register CONTROL_CORE_CONTROL_SPARE_R1 .....	3830
18-640. CONTROL_EFUSE_1 .....	3830
18-641. Register Call Summary for Register CONTROL_EFUSE_1 .....	3830
18-642. CONTROL_EFUSE_2 .....	3830
18-643. Register Call Summary for Register CONTROL_EFUSE_2 .....	3832
18-644. CONTROL_EFUSE_3 .....	3832
18-645. Register Call Summary for Register CONTROL_EFUSE_3 .....	3832
18-646. CONTROL_EFUSE_4 .....	3832
18-647. Register Call Summary for Register CONTROL_EFUSE_4 .....	3833
18-648. SYSCONF_PADCONF_WKUP Register Mapping Summary .....	3833
18-649. CONTROL_PADCONF_WKUP_REVISION .....	3835
18-650. Register Call Summary for Register CONTROL_PADCONF_WKUP_REVISION .....	3835
18-651. CONTROL_PADCONF_WKUP_HWINFO .....	3835
18-652. Register Call Summary for Register CONTROL_PADCONF_WKUP_HWINFO .....	3835
18-653. CONTROL_PADCONF_WKUP_SYSCONFIG .....	3835
18-654. Register Call Summary for Register CONTROL_PADCONF_WKUP_SYSCONFIG .....	3836
18-655. CONTROL_WKUP_PAD0_GPIO_WK0_PAD1_GPIO_WK1 .....	3836
18-656. Register Call Summary for Register CONTROL_WKUP_PAD0_GPIO_WK0_PAD1_GPIO_WK1 .....	3838
18-657. CONTROL_WKUP_PAD0_GPIO_WK2_PAD1_GPIO_WK3 .....	3838
18-658. Register Call Summary for Register CONTROL_WKUP_PAD0_GPIO_WK2_PAD1_GPIO_WK3 .....	3839
18-659. CONTROL_WKUP_PAD0_GPIO_WK4_PAD1_SR_SCL .....	3839
18-660. Register Call Summary for Register CONTROL_WKUP_PAD0_GPIO_WK4_PAD1_SR_SCL .....	3840
18-661. CONTROL_WKUP_PAD0_SR_SDA_PAD1_FREF_XTAL_IN .....	3841
18-662. Register Call Summary for Register CONTROL_WKUP_PAD0_SR_SDA_PAD1_FREF_XTAL_IN .....	3841
18-663. CONTROL_WKUP_PAD0_FREF_SLICER_IN_PAD1_FREF_CLK_IOREQ .....	3842
18-664. Register Call Summary for Register CONTROL_WKUP_PAD0_FREF_SLICER_IN_PAD1_FREF_CLK_IOREQ .....	3843
18-665. CONTROL_WKUP_PAD0_FREF_CLK0_OUT_PAD1_FREF_CLK3_REQ .....	3843
18-666. Register Call Summary for Register CONTROL_WKUP_PAD0_FREF_CLK0_OUT_PAD1_FREF_CLK3_REQ .....	3844
18-667. CONTROL_WKUP_PAD0_FREF_CLK3_OUT_PAD1_FREF_CLK4_REQ .....	3845
18-668. Register Call Summary for Register CONTROL_WKUP_PAD0_FREF_CLK3_OUT_PAD1_FREF_CLK4_REQ .....	3846
18-669. CONTROL_WKUP_PAD0_FREF_CLK4_OUT_PAD1_SYS_32K .....	3846
18-670. Register Call Summary for Register CONTROL_WKUP_PAD0_FREF_CLK4_OUT_PAD1_SYS_32K ..	3848
18-671. CONTROL_WKUP_PAD0_SYS_NRESPWRON_PAD1_SYS_NRESWARM .....	3848
18-672. Register Call Summary for Register CONTROL_WKUP_PAD0_SYS_NRESPWRON_PAD1_SYS_NRESWARM .....	3849
18-673. CONTROL_WKUP_PAD0_SYS_PWR_REQ_PAD1_SYS_PWRON_RESET_OUT .....	3849
18-674. Register Call Summary for Register CONTROL_WKUP_PAD0_SYS_PWR_REQ_PAD1_SYS_PWRON_RESET_OUT .....	3850
18-675. CONTROL_WKUP_PAD0_SYS_BOOT6_PAD1_SYS_BOOT7 .....	3850

18-676. Register Call Summary for Register CONTROL_WKUP_PAD0_SYS_BOOT6_PAD1_SYS_BOOT7 ....	3852
18-677. CONTROL_WKUP_PAD0_JTAG_NTRST_PAD1_JTAG_TCK.....	3852
18-678. Register Call Summary for Register CONTROL_WKUP_PAD0_JTAG_NTRST_PAD1_JTAG_TCK .....	3853
18-679. CONTROL_WKUP_PAD0_JTAG_RTCK_PAD1_JTAG_TMS_TMSC.....	3854
18-680. Register Call Summary for Register CONTROL_WKUP_PAD0_JTAG_RTCK_PAD1_JTAG_TMS_TMSC .....	3855
18-681. CONTROL_WKUP_PAD0_JTAG_TDI_PAD1_JTAG_TDO .....	3855
18-682. Register Call Summary for Register CONTROL_WKUP_PAD0_JTAG_TDI_PAD1_JTAG_TDO .....	3856
18-683. CONTROL_WKUP_PADCONF_WAKEUPEVENT_0 .....	3856
18-684. Register Call Summary for Register CONTROL_WKUP_PADCONF_WAKEUPEVENT_0 .....	3859
18-685. CONTROL_SMART1NOPMIO_PADCONF_0 .....	3859
18-686. Register Call Summary for Register CONTROL_SMART1NOPMIO_PADCONF_0.....	3860
18-687. CONTROL_SMART1NOPMIO_PADCONF_1 .....	3860
18-688. Register Call Summary for Register CONTROL_SMART1NOPMIO_PADCONF_1 .....	3861
18-689. CONTROL_WKUP_PADCONF_MODE.....	3861
18-690. Register Call Summary for Register CONTROL_WKUP_PADCONF_MODE .....	3861
18-691. CONTROL_XTAL_OSCILLATOR.....	3862
18-692. Register Call Summary for Register CONTROL_XTAL_OSCILLATOR .....	3862
18-693. CONTROL_SMART3NOPMIO_PADCONF_0 .....	3862
18-694. Register Call Summary for Register CONTROL_SMART3NOPMIO_PADCONF_0.....	3863
18-695. CONTROL_SMART3NOPMIO_PADCONF_1 .....	3863
18-696. Register Call Summary for Register CONTROL_SMART3NOPMIO_PADCONF_1 .....	3864
18-697. CONTROL_GPIOWK.....	3864
18-698. Register Call Summary for Register CONTROL_GPIOWK .....	3865
18-699. CONTROL_I2C_2.....	3865
18-700. Register Call Summary for Register CONTROL_I2C_2 .....	3866
18-701. CONTROL_JTAG .....	3866
18-702. Register Call Summary for Register CONTROL_JTAG.....	3867
18-703. CONTROL_SYS .....	3867
18-704. Register Call Summary for Register CONTROL_SYS .....	3867
18-705. CONTROL_WKUP_CONTROL_SPARE_RW .....	3867
18-706. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW .....	3867
18-707. CONTROL_WKUP_CONTROL_SPARE_R .....	3868
18-708. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_R.....	3868
18-709. CONTROL_WKUP_CONTROL_SPARE_R_C0 .....	3868
18-710. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_R_C0.....	3869
18-711. CONTROL_WKUP_CONTROL_SPARE_RW1 .....	3869
18-712. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW1 .....	3869
18-713. CONTROL_WKUP_CONTROL_SPARE_RW2 .....	3869
18-714. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW2 .....	3869
18-715. CONTROL_WKUP_CONTROL_SPARE_RW3 .....	3870
18-716. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW3 .....	3870
18-717. CONTROL_WKUP_CONTROL_SPARE_RW4 .....	3870
18-718. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW4 .....	3870
18-719. CONTROL_WKUP_CONTROL_SPARE_RW5 .....	3870
18-720. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW5 .....	3871
18-721. CONTROL_WKUP_CONTROL_SPARE_RW6 .....	3871
18-722. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW6 .....	3871
18-723. CONTROL_WKUP_CONTROL_SPARE_RW7 .....	3871

18-724. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW7 .....	3871
18-725. CONTROL_WKUP_CONTROL_SPARE_RW8 .....	3872
18-726. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW8 .....	3872
18-727. CONTROL_WKUP_CONTROL_SPARE_RW9 .....	3872
18-728. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_RW9 .....	3872
18-729. CONTROL_WKUP_CONTROL_SPARE_R1 .....	3872
18-730. Register Call Summary for Register CONTROL_WKUP_CONTROL_SPARE_R1 .....	3873
19-1. Integration Attributes .....	3878
19-2. Clocks and Resets .....	3879
19-3. Hardware Requests .....	3879
19-4. Integration Attributes .....	3880
19-5. Clocks and Resets .....	3880
19-6. Hardware Requests .....	3880
19-7. Local Power Management Features .....	3884
19-8. Interrupt Events .....	3884
19-9. Global Initialization of Surrounding Modules for System Mailbox .....	3888
19-10. Global Initialization of Surrounding Modules for IVA-HD Mailbox.....	3888
19-11. Mailbox Global Initialization .....	3888
19-12. Sending a Message (Polling Method) .....	3889
19-13. Sending a Message (Interrupt Method) .....	3889
19-14. Receiving a Message (Polling Method).....	3889
19-15. Receiving a Message (Interrupt Method) .....	3889
19-16. Events Servicing in Sending Mode.....	3890
19-17. Events Servicing in Receiving Mode.....	3890
19-18. System Mailbox Instance Summary .....	3891
19-19. IVAHD Mailbox Instance Summary .....	3891
19-20. System Mailbox Registers Mapping Summary .....	3891
19-21. IVAHD Mailbox Registers Mapping Summary .....	3891
19-22. MAILBOX_REVISION.....	3892
19-23. Register Call Summary for Register MAILBOX_REVISION .....	3892
19-24. MAILBOX_SYSCONFIG.....	3892
19-25. Register Call Summary for Register MAILBOX_SYSCONFIG .....	3893
19-26. MAILBOX_MESSAGE_m .....	3893
19-27. Register Call Summary for Register MAILBOX_MESSAGE_m .....	3893
19-28. MAILBOX_FIFOSTATUS_m .....	3894
19-29. Register Call Summary for Register MAILBOX_FIFOSTATUS_m .....	3894
19-30. MAILBOX_MSGSTATUS_m .....	3894
19-31. Register Call Summary for Register MAILBOX_MSGSTATUS_m .....	3895
19-32. MAILBOX_IRQSTATUS_RAW_u .....	3895
19-33. Register Call Summary for Register MAILBOX_IRQSTATUS_RAW_u.....	3897
19-34. MAILBOX_IRQSTATUS_CLR_u .....	3897
19-35. Register Call Summary for Register MAILBOX_IRQSTATUS_CLR_u.....	3900
19-36. MAILBOX_IRQENABLE_SET_u .....	3900
19-37. Register Call Summary for Register MAILBOX_IRQENABLE_SET_u.....	3902
19-38. MAILBOX_IRQENABLE_CLR_u .....	3902
19-39. Register Call Summary for Register MAILBOX_IRQENABLE_CLR_u.....	3905
20-1. Integration Attributes .....	3910
20-2. Clocks and Resets .....	3910
20-3. Hardware Requests .....	3911

20-4.	First-Level Descriptor Format .....	3915
20-5.	Second-Level Descriptor Format .....	3918
20-6.	MMU Local Power Management Features .....	3923
20-7.	Events .....	3923
20-8.	Error Handling .....	3923
20-9.	Global Initialization of Surrounding Modules .....	3925
20-10.	Register Call Summary for Main Sequence—MMU Global Initialization .....	3926
20-11.	Configure a TLB Entry .....	3926
20-12.	MMU Writing TLB Entries Statically .....	3927
20-13.	Protecting TLB Entries .....	3927
20-14.	Deleting TLB Entries .....	3927
20-15.	Read TLB Entries .....	3927
20-16.	MMU Instance Summary .....	3928
20-17.	CORTEXM3_L2MMU and DSP MMU Registers Mapping Summary .....	3928
20-18.	MMU_REVISION .....	3929
20-19.	Register Call Summary for Register MMU_REVISION .....	3929
20-20.	MMU_SYSCONFIG .....	3929
20-21.	Register Call Summary for Register MMU_SYSCONFIG .....	3930
20-22.	MMU_SYSSTATUS .....	3930
20-23.	Register Call Summary for Register MMU_SYSSTATUS .....	3930
20-24.	MMU_IRQSTATUS .....	3931
20-25.	Register Call Summary for Register MMU_IRQSTATUS .....	3931
20-26.	MMU_IRQENABLE .....	3932
20-27.	Register Call Summary for Register MMU_IRQENABLE .....	3932
20-28.	MMU_WALKING_ST .....	3933
20-29.	Register Call Summary for Register MMU_WALKING_ST .....	3933
20-30.	MMU_CNTL .....	3933
20-31.	Register Call Summary for Register MMU_CNTL .....	3934
20-32.	MMU_FAULT_AD .....	3934
20-33.	Register Call Summary for Register MMU_FAULT_AD .....	3934
20-34.	MMU_TTB .....	3934
20-35.	Register Call Summary for Register MMU_TTB .....	3934
20-36.	MMU_LOCK .....	3935
20-37.	Register Call Summary for Register MMU_LOCK .....	3935
20-38.	MMU_LD_TLB .....	3935
20-39.	Register Call Summary for Register MMU_LD_TLB .....	3936
20-40.	MMU_CAM .....	3936
20-41.	Register Call Summary for Register MMU_CAM .....	3936
20-42.	MMU_RAM .....	3937
20-43.	Register Call Summary for Register MMU_RAM .....	3937
20-44.	MMU_GFLUSH .....	3937
20-45.	Register Call Summary for Register MMU_GFLUSH .....	3938
20-46.	MMU_FLUSH_ENTRY .....	3938
20-47.	Register Call Summary for Register MMU_FLUSH_ENTRY .....	3939
20-48.	MMU_READ_CAM .....	3939
20-49.	Register Call Summary for Register MMU_READ_CAM .....	3939
20-50.	MMU_READ_RAM .....	3940
20-51.	Register Call Summary for Register MMU_READ_RAM .....	3940
20-52.	MMU_EMU_FAULT_AD .....	3940

20-53. Register Call Summary for Register MMU_EMU_FAULT_AD .....	3941
20-54. MMU_FAULT_PC .....	3941
20-55. Register Call Summary for Register MMU_FAULT_PC.....	3941
20-56. MMU_FAULT_STATUS .....	3941
20-57. Register Call Summary for Register MMU_FAULT_STATUS.....	3942
20-58. MMU_GP_REG .....	3942
20-59. Register Call Summary for Register MMU_GP_REG .....	3943
21-1. Integration Attributes .....	3947
21-2. Clocks and Resets .....	3947
21-3. Local Power Management Features .....	3948
21-4. Global Initialization of Surrounding Modules.....	3951
21-5. Spinlock System Bug Recovery .....	3951
21-6. Register Call Summary .....	3952
21-7. Subprocess Call Summary .....	3952
21-8. Spinlock Instance Summary .....	3954
21-9. Spinlock Registers Mapping Summary .....	3954
21-10. SPINLOCK_REVISION .....	3954
21-11. Register Call Summary for Register SPINLOCK_REVISION .....	3954
21-12. SPINLOCK_SYSCONFIG .....	3955
21-13. Register Call Summary for Register SPINLOCK_SYSCONFIG .....	3955
21-14. SPINLOCK_SYSTATUS.....	3956
21-15. Register Call Summary for Register SPINLOCK_SYSTATUS .....	3956
21-16. SPINLOCK_LOCK_REG_i .....	3957
21-17. Register Call Summary for Register SPINLOCK_LOCK_REG_i.....	3957
22-1. Input/Output Description.....	3963
22-2. Integration Attributes .....	3965
22-3. Clocks and Resets .....	3965
22-4. Hardware Requests .....	3966
22-5. IDLEMODE Settings .....	3971
22-6. CLOCKACTIVITY Settings .....	3971
22-7. IDLEMODE Settings .....	3973
22-8. Value Loaded in GPT_TCRR to Generate 1-ms Tick .....	3976
22-9. Prescaler/Timer Reload Values Versus Contexts.....	3979
22-10. Prescaler Clock Ratio Values .....	3980
22-11. Value and Corresponding Interrupt Period.....	3981
22-12. Global Initialization of Surrounding Modules.....	3984
22-13. GP Timer Module Global Initialization .....	3985
22-14. GP Timer Mode Configuration .....	3985
22-15. GP Timer Compare Mode Configuration .....	3985
22-16. GP Timer Capture Mode Configuration.....	3986
22-17. Initialize Capture Mode .....	3986
22-18. Detect Event .....	3986
22-19. GP Timer PWM Mode Configuration.....	3986
22-20. GP Timer Instance Summary .....	3988
22-21. GPTIMER1, GPTIMER2, and GPTIMER10 Register Mapping Summary .....	3989
22-22. GPTIMER3 and GPTIMER4 Register Mapping Summary .....	3989
22-23. GPTIMER5 Register Mapping Summary .....	3990
22-24. GPTIMER6 Register Mapping Summary .....	3990
22-25. GPTIMER7 Register Mapping Summary .....	3991

22-26. GPTIMER8 Register Mapping Summary .....	3991
22-27. GPTIMER9 and GPTIMER11 Register Mapping Summary .....	3992
22-28. GPT_TIDR.....	3992
22-29. Register Call Summary for Register GPT_TIDR .....	3992
22-30. GPT_TIOCP_CFG .....	3993
22-31. Register Call Summary for Register GPT_TIOCP_CFG .....	3993
22-32. GPT1MS_TIOCP_CFG .....	3994
22-33. Register Call Summary for Register GPT1MS_TIOCP_CFG .....	3995
22-34. GPT_IRQSTATUS_RAW.....	3995
22-35. Register Call Summary for Register GPT_IRQSTATUS_RAW .....	3995
22-36. GPT_IRQSTATUS .....	3996
22-37. Register Call Summary for Register GPT_IRQSTATUS .....	3996
22-38. GPT_IRQENABLE_SET.....	3996
22-39. Register Call Summary for Register GPT_IRQENABLE_SET .....	3997
22-40. GPT_IRQENABLE_CLR.....	3997
22-41. Register Call Summary for Register GPT_IRQENABLE_CLR .....	3998
22-42. GPT_IRQWAKEEN .....	3998
22-43. Register Call Summary for Register GPT_IRQWAKEEN .....	3999
22-44. GPT_TCLR .....	3999
22-45. Register Call Summary for Register GPT_TCLR .....	4000
22-46. GPT_TCRR .....	4000
22-47. Register Call Summary for Register GPT_TCRR .....	4001
22-48. GPT_TLDR.....	4001
22-49. Register Call Summary for Register GPT_TLDR .....	4001
22-50. GPT_TTGR.....	4002
22-51. Register Call Summary for Register GPT_TTGR .....	4002
22-52. GPT_TWPS .....	4002
22-53. Register Call Summary for Register GPT_TWPS.....	4003
22-54. GPT_TMAR .....	4003
22-55. Register Call Summary for Register GPT_TMAR .....	4003
22-56. GPT_TCAR1 .....	4003
22-57. Register Call Summary for Register GPT_TCAR1.....	4003
22-58. GPT_TSICR .....	4004
22-59. Register Call Summary for Register GPT_TSICR .....	4004
22-60. GPT_TCAR2 .....	4004
22-61. Register Call Summary for Register GPT_TCAR2.....	4004
22-62. GPT_TISTAT.....	4005
22-63. Register Call Summary for Register GPT_TISTAT .....	4005
22-64. GPT_TISR.....	4005
22-65. Register Call Summary for Register GPT_TISR .....	4006
22-66. GPT_TIER.....	4006
22-67. Register Call Summary for Register GPT_TIER .....	4007
22-68. GPT_TWER .....	4007
22-69. Register Call Summary for Register GPT_TWER .....	4007
22-70. GPT_TPIR.....	4008
22-71. Register Call Summary for Register GPT_TPIR .....	4008
22-72. GPT_TNIR.....	4008
22-73. Register Call Summary for Register GPT_TNIR .....	4008
22-74. GPT_TCVR.....	4009

22-75. Register Call Summary for Register GPT_TCVR .....	4009
22-76. GPT_TOCR .....	4009
22-77. Register Call Summary for Register GPT_TOCR .....	4009
22-78. GPT_TOWR .....	4010
22-79. Register Call Summary for Register GPT_TOWR .....	4010
22-80. Watchdog Timers Default State .....	4011
22-81. Integration Attributes .....	4013
22-82. Clocks and Resets .....	4013
22-83. Hardware Requests .....	4014
22-84. IDLEMODE Settings .....	4015
22-85. Watchdog Timer Events .....	4016
22-86. Count and Prescaler Default Reset Values .....	4016
22-87. Prescaler Clock Ratio Values .....	4017
22-88. Reset Period Examples .....	4018
22-89. Default Watchdog Timer Reset Periods .....	4018
22-90. Global Initialization of Surrounding Modules .....	4021
22-91. Watchdog Timer Module Global Initialization .....	4022
22-92. Watchdog Timer Basic Configuration .....	4022
22-93. Disable the Watchdog Timer .....	4022
22-94. Enable the Watchdog Timer .....	4022
22-95. Watchdog Timer Instance Summary .....	4023
22-96. WDTIMER2 Register Mapping Summary .....	4023
22-97. WDTIMER3 Register Mapping Summary .....	4024
22-98. WDT_WIDR .....	4025
22-99. Register Call Summary for Register WDT_WIDR .....	4025
22-100. WDT_WDSC .....	4025
22-101. Register Call Summary for Register WDT_WDSC .....	4026
22-102. WDT_WDST .....	4026
22-103. Register Call Summary for Register WDT_WDST .....	4027
22-104. WDT_WISR .....	4027
22-105. Register Call Summary for Register WDT_WISR .....	4027
22-106. WDT_WIER .....	4028
22-107. Register Call Summary for Register WDT_WIER .....	4028
22-108. WDT_WWER .....	4028
22-109. Register Call Summary for Register WDT_WWER .....	4029
22-110. WDT_WCLR .....	4029
22-111. Register Call Summary for Register WDT_WCLR .....	4029
22-112. WDT_WCRR .....	4029
22-113. Register Call Summary for Register WDT_WCRR .....	4030
22-114. WDT_WLDR .....	4030
22-115. Register Call Summary for Register WDT_WLDR .....	4030
22-116. WDT_WTGR .....	4030
22-117. Register Call Summary for Register WDT_WTGR .....	4031
22-118. WDT_WWPS .....	4031
22-119. Register Call Summary for Register WDT_WWPS .....	4032
22-120. WDT_WDLY .....	4032
22-121. Register Call Summary for Register WDT_WDLY .....	4032
22-122. WDT_WSPR .....	4032
22-123. Register Call Summary for Register WDT_WSPR .....	4032



22-124. WDT_WIRQSTATRAW .....	4033
22-125. Register Call Summary for Register WDT_WIRQSTATRAW .....	4033
22-126. WDT_WIRQSTAT .....	4033
22-127. Register Call Summary for Register WDT_WIRQSTAT .....	4034
22-128. WDT_WIRQENSET .....	4034
22-129. Register Call Summary for Register WDT_WIRQENSET .....	4035
22-130. WDT_WIRQENCLR .....	4035
22-131. Register Call Summary for Register WDT_WIRQENCLR .....	4035
22-132. WDT_WIRQWAKEEN .....	4036
22-133. Register Call Summary for Register WDT_WIRQWAKEEN .....	4036
22-134. Integration Attributes .....	4038
22-135. Clocks and Resets .....	4038
22-136. Hardware Requests .....	4038
22-137. 32KTimer Instance Summary .....	4040
22-138. 32KTimer Register Summary .....	4040
22-139. 32KSYNCNT_REV .....	4040
22-140. Register Call Summary for Register 32KSYNCNT_REV .....	4040
22-141. 32KSYNCNT_SYSCONFIG .....	4041
22-142. Register Call Summary for Register 32KSYNCNT_SYSCONFIG .....	4041
22-143. 32KSYNCNT_CR .....	4041
22-144. Register Call Summary for Register 32KSYNCNT_CR .....	4041
23-1. HS I <sup>2</sup> C Input/Output .....	4047
23-2. HS I <sup>2</sup> C Input/Output .....	4053
23-3. HS I <sup>2</sup> C Input/Output Description .....	4056
23-4. HS I <sup>2</sup> C Integration Attributes .....	4061
23-5. HS I <sup>2</sup> C Clocks and Resets .....	4061
23-6. HS I <sup>2</sup> C Hardware Requests .....	4062
23-7. HS I <sup>2</sup> C Operation Mode Selection .....	4063
23-8. HS I <sup>2</sup> C $t_{Low}$ and $t_{high}$ Values of the I <sup>2</sup> C Clock .....	4064
23-9. HS I <sup>2</sup> C Register Values for Maximum I <sup>2</sup> C Bit Rates in I <sup>2</sup> C F/S, I <sup>2</sup> C HS, and SCCB Modes .....	4066
23-10. HS I <sup>2</sup> C Local Power-Management Features .....	4068
23-11. HS I <sup>2</sup> C Clock Activity Settings .....	4069
23-12. HS I <sup>2</sup> C Events .....	4069
23-13. HS I <sup>2</sup> C DMA Requests .....	4070
23-14. HS I <sup>2</sup> C List of Tests .....	4076
23-15. HS I <sup>2</sup> C Subprocess call summary for Sequence - Setup Procedure .....	4080
23-16. HS I <sup>2</sup> C Register call summary for Sequence - Setup Procedure .....	4081
23-17. HS I <sup>2</sup> C Register call summary for Sequence - Master Transmitter Mode, Polling Method, in F/S and HS Modes .....	4082
23-18. HS I <sup>2</sup> C Register call summary for Sequence - Master Receiver Mode, Polling Method, in F/S and HS Modes .....	4084
23-19. HS I <sup>2</sup> C Register call summary for Sequence - Master Transmitter Mode, Interrupt Method, in F/S and HS Modes .....	4086
23-20. HS I <sup>2</sup> C Register call summary for Sequence - Master Receiver Mode, Interrupt Method, in F/S and HS Modes .....	4088
23-21. HS I <sup>2</sup> C Register call summary for Sequence - Master Transmitter Mode, DMA Method in F/S and HS Modes .....	4090
23-22. HS I <sup>2</sup> C Register call summary for Sequence - Master Receiver Mode, DMA Method in F/S and HS Modes .....	4092
23-23. HS I <sup>2</sup> C Register call summary for Sequence - Slave Transmitter/Receiver Mode, Polling .....	4092

23-24. HS I <sup>2</sup> C Register call summary for Sequence - Slave Transmitter/Receiver Mode, Interrupt .....	4093
23-25. HS I <sup>2</sup> C Register call summary for Sequence - SCCB Setup Procedure .....	4096
23-26. HS I <sup>2</sup> C Register call summary for Sequence - SCCB Master Transmitter Mode, Polling.....	4097
23-27. HS I <sup>2</sup> C Register call summary for Sequence - SCCB Master Receiver Mode, Polling.....	4099
23-28. HS I <sup>2</sup> C Register call summary for Sequence - SCCB Master Transmitter Mode, Interrupt.....	4101
23-29. HS I <sup>2</sup> C Register call summary for Sequence - SCCB Master Receiver Mode, Interrupt.....	4103
23-30. HS I <sup>2</sup> C Instance Summary.....	4103
23-31. HS I <sup>2</sup> C Registers Mapping Summary 1 .....	4103
23-32. HS I <sup>2</sup> C Registers Mapping Summary 2 .....	4104
23-33. I2C_REVNB_LO .....	4105
23-34. Register Call Summary for Register I2C_REVNB_LO .....	4105
23-35. I2C_REVNB_HI .....	4105
23-36. Register Call Summary for Register I2C_REVNB_HI .....	4106
23-37. I2C_SYSC .....	4106
23-38. Register Call Summary for Register I2C_SYSC .....	4106
23-39. I2C_IRQSTATUS_RAW .....	4107
23-40. Register Call Summary for Register I2C_IRQSTATUS_RAW.....	4108
23-41. I2C_IRQSTATUS .....	4108
23-42. Register Call Summary for Register I2C_IRQSTATUS .....	4110
23-43. I2C_IRQENABLE_SET .....	4110
23-44. Register Call Summary for Register I2C_IRQENABLE_SET.....	4111
23-45. I2C_IRQENABLE_CLR .....	4112
23-46. Register Call Summary for Register I2C_IRQENABLE_CLR .....	4113
23-47. I2C_WE .....	4113
23-48. Register Call Summary for Register I2C_WE .....	4114
23-49. I2C_DMARXENABLE_SET .....	4114
23-50. Register Call Summary for Register I2C_DMARXENABLE_SET.....	4115
23-51. I2C_DMATXENABLE_SET .....	4115
23-52. Register Call Summary for Register I2C_DMATXENABLE_SET .....	4115
23-53. I2C_DMARXENABLE_CLR .....	4116
23-54. Register Call Summary for Register I2C_DMARXENABLE_CLR.....	4116
23-55. I2C_DMATXENABLE_CLR .....	4116
23-56. Register Call Summary for Register I2C_DMATXENABLE_CLR.....	4116
23-57. I2C_DMARXWAKE_EN .....	4117
23-58. Register Call Summary for Register I2C_DMARXWAKE_EN.....	4118
23-59. I2C_DMATXWAKE_EN.....	4118
23-60. Register Call Summary for Register I2C_DMATXWAKE_EN .....	4119
23-61. I2C_IE .....	4119
23-62. Register Call Summary for Register I2C_IE .....	4120
23-63. I2C_STAT .....	4121
23-64. Register Call Summary for Register I2C_STAT .....	4122
23-65. I2C_SYSS .....	4123
23-66. Register Call Summary for Register I2C_SYSS.....	4123
23-67. I2C_BUF .....	4123
23-68. Register Call Summary for Register I2C_BUF .....	4124
23-69. I2C_CNT.....	4124
23-70. Register Call Summary for Register I2C_CNT .....	4124
23-71. I2C_DATA .....	4125
23-72. Register Call Summary for Register I2C_DATA.....	4125

23-73. I2C_CON .....	4125
23-74. Register Call Summary for Register I2C_CON .....	4126
23-75. I2C_OA .....	4127
23-76. Register Call Summary for Register I2C_OA .....	4127
23-77. I2C_SA .....	4127
23-78. Register Call Summary for Register I2C_SA .....	4127
23-79. I2C_PSC .....	4128
23-80. Register Call Summary for Register I2C_PSC .....	4128
23-81. I2C_SCLL .....	4128
23-82. Register Call Summary for Register I2C_SCLL .....	4128
23-83. I2C_SCLH .....	4129
23-84. Register Call Summary for Register I2C_SCLH .....	4129
23-85. I2C_SYSTEST .....	4129
23-86. Register Call Summary for Register I2C_SYSTEST .....	4131
23-87. I2C_BUFSTAT .....	4131
23-88. Register Call Summary for Register I2C_BUFSTAT .....	4131
23-89. I2C_OA1 .....	4131
23-90. Register Call Summary for Register I2C_OA1 .....	4131
23-91. I2C_OA2 .....	4132
23-92. Register Call Summary for Register I2C_OA2 .....	4132
23-93. I2C_OA3 .....	4132
23-94. Register Call Summary for Register I2C_OA3 .....	4132
23-95. I2C_ACTOA .....	4132
23-96. Register Call Summary for Register I2C_ACTOA .....	4133
23-97. I2C_SBLOCK .....	4133
23-98. Register Call Summary for Register I2C_SBLOCK .....	4134
23-99. Functional Modes .....	4136
23-100. I/O Description .....	4136
23-101. HDQ/1-Wire Command Byte .....	4138
23-102. Local Power-Management Features .....	4141
23-103. Events .....	4142
23-104. Integration Attributes .....	4146
23-105. Clocks and Resets .....	4147
23-106. Hardware Requests .....	4147
23-107. Global Initialization of Surrounding Modules .....	4147
23-108. HDQ/1-Wire Module Global Initialization .....	4147
23-109. HDQ Mode Selection .....	4148
23-110. HDQ Write Operation Mode .....	4148
23-111. HDQ Read Operation Mode .....	4148
23-112. Initialize HDQ Slave .....	4148
23-113. 1-Wire Mode Selection .....	4149
23-114. 1-Wire Write Operation Mode .....	4149
23-115. 1-Wire Read Operation Mode .....	4149
23-116. Initialize 1-Wire Slave .....	4149
23-117. HDQ/1-Wire Instance Summary .....	4150
23-118. HDQ/1-Wire Registers Mapping Summary .....	4150
23-119. HDQ_REVISION .....	4150
23-120. Register Call Summary for Register HDQ_REVISION .....	4151
23-121. HDQ_TX_DATA .....	4151

23-122. Register Call Summary for Register HDQ_TX_DATA .....	4151
23-123. HDQ_RX_DATA.....	4151
23-124. Register Call Summary for Register HDQ_RX_DATA .....	4151
23-125. HDQ_CTRL_STATUS .....	4152
23-126. Register Call Summary for Register HDQ_CTRL_STATUS .....	4153
23-127. HDQ_INT_STATUS.....	4153
23-128. Register Call Summary for Register HDQ_INT_STATUS .....	4153
23-129. HDQ_SYSCONFIG .....	4154
23-130. Register Call Summary for Register HDQ_SYSCONFIG.....	4154
23-131. HDQ_SYSSTATUS .....	4154
23-132. Register Call Summary for Register HDQ_SYSSTATUS.....	4155
23-133. UART Mode Baud Rates, Divisor Values, and Error Rates .....	4156
23-134. UART IrDA Mode Baud Rates, Divisor Values, and Error Rates.....	4157
23-135. UART I/O Pins.....	4159
23-136. UART3 I/O Signals.....	4160
23-137. UART_EFR[1:0] IR Address Checking Options.....	4164
23-138. 4-PPM Format .....	4167
23-139. FIR Preamble, Start Flag, and Stop Flag .....	4167
23-140. FIR Data Byte Transmission Order Example.....	4168
23-141. CIR I/O Signals .....	4168
23-142. Integration Attributes.....	4173
23-143. Clocks and Resets .....	4174
23-144. Hardware Requests.....	4174
23-145. Local Power Management features.....	4178
23-146. UART Mode Interrupts .....	4178
23-147. IrDA Mode Interrupts .....	4179
23-148. CIR Mode Interrupts .....	4180
23-149. TX FIFO Trigger Level Setting Summary .....	4183
23-150. RX FIFO Trigger Level Setting Summary.....	4183
23-151. UART/IrDA/CIR Register Access Mode Programming (Using UART_LCR) .....	4190
23-152. Subconfiguration Mode A Summary.....	4190
23-153. Subconfiguration Mode B Summary.....	4190
23-154. Suboperational Mode Summary.....	4190
23-155. UART/IrDA/CIR Register Access Mode Overview.....	4191
23-156. UART Mode Selection.....	4192
23-157. UART Mode Register Overview .....	4192
23-158. IrDA Mode Register Overview .....	4193
23-159. CIR Mode Register Overview .....	4194
23-160. UART Baud Rate Settings (48-MHz Clock) .....	4196
23-161. UART Parity Bit Encoding.....	4197
23-162. UART_EFR[0:3] Software Flow Control Options.....	4198
23-163. IrDA Baud Rate Settings .....	4202
23-164. Duty Cycle .....	4207
23-165. UART/IrDA/CIR Instance Summary .....	4216
23-166. UART/IrDA/CIR Register Mapping Summary (UART1 and UART2).....	4216
23-167. UART/IrDA/CIR Register Mapping Summary (UART3 and UART4).....	4217
23-168. UART_THR .....	4218
23-169. Register Call Summary for Register UART_THR.....	4219
23-170. UART_RHR.....	4219

23-171. Register Call Summary for Register UART_RHR .....	4219
23-172. UART_DLL.....	4220
23-173. Register Call Summary for Register UART_DLL .....	4220
23-174. UART_IER .....	4220
23-175. Register Call Summary for Register UART_IER.....	4221
23-176. UART_IIR .....	4223
23-177. Register Call Summary for Register UART_IIR.....	4223
23-178. UART_DLH .....	4225
23-179. Register Call Summary for Register UART_DLH.....	4225
23-180. UART_FCR .....	4226
23-181. Register Call Summary for Register UART_FCR.....	4227
23-182. UART_EFR .....	4227
23-183. Register Call Summary for Register UART_EFR.....	4228
23-184. UART_LCR .....	4228
23-185. Register Call Summary for Register UART_LCR.....	4230
23-186. UART_XON1_ADDR1.....	4230
23-187. Register Call Summary for Register UART_XON1_ADDR1 .....	4230
23-188. UART_MCR.....	4231
23-189. Register Call Summary for Register UART_MCR .....	4232
23-190. UART_LSR .....	4232
23-191. Register Call Summary for Register UART_LSR .....	4233
23-192. UART_XON2_ADDR2.....	4235
23-193. Register Call Summary for Register UART_XON2_ADDR2 .....	4235
23-194. UART_TCR .....	4235
23-195. Register Call Summary for Register UART_TCR.....	4236
23-196. UART_XOFF1 .....	4236
23-197. Register Call Summary for Register UART_XOFF1.....	4236
23-198. UART_MSR.....	4237
23-199. Register Call Summary for Register UART_MSR .....	4237
23-200. UART_SPR .....	4238
23-201. Register Call Summary for Register UART_SPR.....	4238
23-202. UART_TLR.....	4238
23-203. Register Call Summary for Register UART_TLR .....	4239
23-204. UART_XOFF2 .....	4239
23-205. Register Call Summary for Register UART_XOFF2.....	4239
23-206. UART_MDR1 .....	4239
23-207. Register Call Summary for Register UART_MDR1.....	4241
23-208. UART_MDR2 .....	4241
23-209. Register Call Summary for Register UART_MDR2.....	4242
23-210. UART_SFLSR .....	4243
23-211. Register Call Summary for Register UART_SFLSR.....	4243
23-212. UART_TXFLL.....	4244
23-213. Register Call Summary for Register UART_TXFLL .....	4244
23-214. UART_RESUME .....	4244
23-215. Register Call Summary for Register UART_RESUME.....	4245
23-216. UART_TXFLH .....	4245
23-217. Register Call Summary for Register UART_TXFLH.....	4245
23-218. UART_SFREGL .....	4246
23-219. Register Call Summary for Register UART_SFREGL .....	4246

23-220. UART_RXFLL .....	4246
23-221. Register Call Summary for Register UART_RXFLL .....	4247
23-222. UART_SFREGH .....	4247
23-223. Register Call Summary for Register UART_SFREGH .....	4247
23-224. UART_RXFLH .....	4248
23-225. Register Call Summary for Register UART_RXFLH .....	4248
23-226. UART_BLR .....	4248
23-227. Register Call Summary for Register UART_BLR .....	4249
23-228. UART_UASR .....	4249
23-229. Register Call Summary for Register UART_UASR .....	4250
23-230. UART_ACREG .....	4250
23-231. Register Call Summary for Register UART_ACREG .....	4251
23-232. UART_SCR .....	4251
23-233. Register Call Summary for Register UART_SCR .....	4252
23-234. UART_SSR .....	4253
23-235. Register Call Summary for Register UART_SSR .....	4253
23-236. UART_EBLR .....	4254
23-237. Register Call Summary for Register UART_EBLR .....	4254
23-238. UART_MVR .....	4255
23-239. Register Call Summary for Register UART_MVR .....	4255
23-240. UART_SYSC .....	4255
23-241. Register Call Summary for Register UART_SYSC .....	4256
23-242. UART_SYSS .....	4256
23-243. Register Call Summary for Register UART_SYSS .....	4257
23-244. UART_WER .....	4257
23-245. Register Call Summary for Register UART_WER .....	4258
23-246. UART_CFPS .....	4258
23-247. Register Call Summary for Register UART_CFPS .....	4259
23-248. UART_RXFIFO_LVL .....	4259
23-249. Register Call Summary for Register UART_RXFIFO_LVL .....	4259
23-250. UART_TXFIFO_LVL .....	4259
23-251. Register Call Summary for Register UART_TXFIFO_LVL .....	4259
23-252. UART_IER2 .....	4260
23-253. Register Call Summary for Register UART_IER2 .....	4260
23-254. UART_ISR2 .....	4260
23-255. Register Call Summary for Register UART_ISR2 .....	4261
23-256. UART_FREQ_SEL .....	4261
23-257. Register Call Summary for Register UART_FREQ_SEL .....	4261
23-258. UART_MDR3 .....	4262
23-259. Register Call Summary for Register UART_MDR3 .....	4262
23-260. UART_TX_DMA_THRESHOLD .....	4262
23-261. Register Call Summary for Register UART_TX_DMA_THRESHOLD .....	4263
23-262. MCSPI I/O Description (Master Mode) .....	4268
23-263. MCSPI I/O Description (Slave Mode) .....	4268
23-264. SPI Master Clock Rates .....	4269
23-265. Phase and Polarity Combinations .....	4270
23-266. Integration Attributes .....	4275
23-267. Clocks and Resets .....	4275
23-268. Hardware Requests .....	4275

23-269. Wake-Up Requests .....	4276
23-270. SPI Master Clock Rates.....	4283
23-271. CLKSPPIO High/Low Time Computation .....	4283
23-272. Clock Granularity Examples .....	4284
23-273. FIFO Writes, Word Length Relationship .....	4289
23-274. Smart-Idle Mode and Wake-Up Capabilities .....	4296
23-275. Global Initialization of Surrounding Modules .....	4298
23-276. MCSPI Global Initialization.....	4298
23-277. MCSPI Receive Mode initialization .....	4298
23-278. MCSPI Transmit Mode initialization .....	4299
23-279. MCSPI Transmit and Receive Mode initialization.....	4299
23-280. Receive-Only Procedure – Polling Method .....	4299
23-281. Receive-Only Procedure – Interrupt Method .....	4299
23-282. Transmit-Only Procedure – Polling Method .....	4300
23-283. Transmit and Receive Procedure – Polling Method .....	4300
23-284. Receive-Only Procedure With Word Count – Polling Method.....	4300
23-285. Transmit-Only Procedure Without Word Count – Polling Method .....	4301
23-286. Transmit-Only Procedure With Word Count – Interrupt Method .....	4301
23-287. Transmit and Receive Procedure With Word Count – Polling Method .....	4301
23-288. Transmit and Receive Procedure With Word Count – Interrupt Method.....	4301
23-289. Transmit and Receive Procedure Without Word Count – Polling Method.....	4302
23-290. MCSPI Instance Summary.....	4303
23-291. MCSPI Register Summary .....	4303
23-292. MCSPI_HL_REV .....	4304
23-293. Register Call Summary for Register MCSPI_HL_REV.....	4304
23-294. MCSPI_HL_HWINFO.....	4304
23-295. Register Call Summary for Register MCSPI_HL_HWINFO .....	4305
23-296. MCSPI_HL_SYSCONFIG.....	4305
23-297. Register Call Summary for Register MCSPI_HL_SYSCONFIG .....	4306
23-298. MCSPI_REVISION.....	4306
23-299. Register Call Summary for Register MCSPI_REVISION .....	4306
23-300. MCSPI_SYSCONFIG.....	4307
23-301. Register Call Summary for Register MCSPI_SYSCONFIG .....	4308
23-302. MCSPI_SYSSTATUS .....	4308
23-303. Register Call Summary for Register MCSPI_SYSSTATUS .....	4308
23-304. MCSPI_IRQSTATUS .....	4308
23-305. Register Call Summary for Register MCSPI_IRQSTATUS.....	4311
23-306. MCSPI_IRQENABLE .....	4311
23-307. Register Call Summary for Register MCSPI_IRQENABLE.....	4313
23-308. MCSPI_WAKEUPENABLE .....	4313
23-309. Register Call Summary for Register MCSPI_WAKEUPENABLE .....	4314
23-310. MCSPI_SYST .....	4314
23-311. Register Call Summary for Register MCSPI_SYST .....	4315
23-312. MCSPI_MODULCTRL.....	4315
23-313. Register Call Summary for Register MCSPI_MODULCTRL .....	4317
23-314. MCSPI_CHxCONF.....	4317
23-315. Register Call Summary for Register MCSPI_CHxCONF .....	4321
23-316. MCSPI_CHxSTAT .....	4321
23-317. Register Call Summary for Register MCSPI_CHxSTAT .....	4322

23-318. MCSPI_CHxCTRL .....	4323
23-319. Register Call Summary for Register MCSPI_CHxCTRL.....	4323
23-320. MCSPI_TXx.....	4324
23-321. Register Call Summary for Register MCSPI_TXx .....	4324
23-322. MCSPI_RXx .....	4324
23-323. Register Call Summary for Register MCSPI_RXx .....	4325
23-324. MCSPI_XFERLEVEL .....	4325
23-325. Register Call Summary for Register MCSPI_XFERLEVEL .....	4326
23-326. Functions Description.....	4329
23-327. Input/Output Description .....	4330
23-328. Integration Attributes.....	4344
23-329. Clocks and Resets .....	4344
23-330. Hardware Requests.....	4344
23-331. Software Reset Signals to All McBSP Modules.....	4350
23-332. State of Clocks When the Module is in Idle State .....	4351
23-333. McBSP Smart Idle Mode Configuration Behavior.....	4354
23-334. Receiver Clock Mode .....	4357
23-335. Phases, Words and Bits per Frame Control Bit.....	4361
23-336. Assumptions for the Single-Phase Frame Example .....	4361
23-337. Assumptions for the Dual-Phase Frame Example .....	4362
23-338. Effects of DLB and ALB Bits on Clock Modes .....	4369
23-339. McBSP Channels .....	4377
23-340. Eight Partitions – Receive Channel Assignment and Control .....	4378
23-341. Eight Partitions – Transmit Channel Assignment and Control .....	4379
23-342. Selecting a Transmit Multi-Channel Selection Mode with the XMCM Bit Field .....	4380
23-343. McBSP Channel Control Options .....	4381
23-344. Register Call Summary for Flow Diagram of McBSP Initialization Procedure .....	4387
23-345. Register Call Summary for Flow Diagram for the SRG Registers Programmation.....	4388
23-346. Data Transfer DMA Request Configuration.....	4389
23-347. Interruption Configuration .....	4389
23-348. Receiver Reset .....	4389
23-349. Global Configuration .....	4390
23-350. Data Configuration .....	4390
23-351. Frame Sync Configuration .....	4390
23-352. Clock Configuration .....	4391
23-353. Take the Receiver Out of Reset .....	4391
23-354. Transmitter Reset .....	4391
23-355. Global Configuration .....	4392
23-356. Data Configuration .....	4392
23-357. Frame-Synchronization Configuration .....	4392
23-358. Clock Configuration .....	4393
23-359. Take the Receiver Out of Reset .....	4393
23-360. Use Serial Receive Pins as GPIO Pins .....	4393
23-361. Use Serial Transmit Pins as GPIO Pins .....	4393
23-362. Use External Clock Pin as GPIO Pin.....	4393
23-363. McBSP Instance Summary.....	4394
23-364. McBSP1 Register Summary .....	4394
23-365. McBSP2 Register Summary .....	4395
23-366. McBSP3 Register Summary .....	4396



23-367. MCBSP4 Register Summary .....	4397
23-368. MCBSP4_DRR_REG .....	4399
23-369. Register Call Summary for Register MCBSP4_DRR_REG.....	4399
23-370. MCBSP4_DXR_REG .....	4399
23-371. Register Call Summary for Register MCBSP4_DXR_REG.....	4399
23-372. MCBSP4_SPCR2_REG .....	4399
23-373. Register Call Summary for Register MCBSP4_SPCR2_REG .....	4401
23-374. MCBSP4_SPCR1_REG .....	4401
23-375. Register Call Summary for Register MCBSP4_SPCR1_REG .....	4402
23-376. MCBSP4_RCR2_REG .....	4402
23-377. Register Call Summary for Register MCBSP4_RCR2_REG .....	4403
23-378. MCBSP4_RCR1_REG .....	4403
23-379. Register Call Summary for Register MCBSP4_RCR1_REG .....	4404
23-380. MCBSP4_XCR2_REG .....	4404
23-381. Register Call Summary for Register MCBSP4_XCR2_REG .....	4405
23-382. MCBSP4_XCR1_REG .....	4405
23-383. Register Call Summary for Register MCBSP4_XCR1_REG .....	4406
23-384. MCBSP4_SRGR2_REG.....	4406
23-385. Register Call Summary for Register MCBSP4_SRGR2_REG .....	4407
23-386. MCBSP4_SRGR1_REG.....	4407
23-387. Register Call Summary for Register MCBSP4_SRGR1_REG .....	4407
23-388. MCBSP4_MCR2_REG .....	4408
23-389. Register Call Summary for Register MCBSP4_MCR2_REG.....	4409
23-390. MCBSP4_MCR1_REG .....	4409
23-391. Register Call Summary for Register MCBSP4_MCR1_REG.....	4410
23-392. MCBSP4_RCERA_REG .....	4410
23-393. Register Call Summary for Register MCBSP4_RCERA_REG .....	4410
23-394. MCBSP4_RCERB_REG .....	4411
23-395. Register Call Summary for Register MCBSP4_RCERB_REG .....	4411
23-396. MCBSP4_XCERA_REG.....	4411
23-397. Register Call Summary for Register MCBSP4_XCERA_REG .....	4411
23-398. MCBSP4_XCERB_REG.....	4412
23-399. Register Call Summary for Register MCBSP4_XCERB_REG .....	4412
23-400. MCBSP4_PCR_REG .....	4412
23-401. Register Call Summary for Register MCBSP4_PCR_REG.....	4414
23-402. MCBSP4_RCERC_REG .....	4414
23-403. Register Call Summary for Register MCBSP4_RCERC_REG.....	4414
23-404. MCBSP4_RCERD_REG .....	4415
23-405. Register Call Summary for Register MCBSP4_RCERD_REG.....	4415
23-406. MCBSP4_XCERC_REG .....	4415
23-407. Register Call Summary for Register MCBSP4_XCERC_REG .....	4415
23-408. MCBSP4_XCERD_REG .....	4416
23-409. Register Call Summary for Register MCBSP4_XCERD_REG .....	4416
23-410. MCBSP4_RCERE_REG .....	4416
23-411. Register Call Summary for Register MCBSP4_RCERE_REG .....	4416
23-412. MCBSP4_RCERF_REG.....	4417
23-413. Register Call Summary for Register MCBSP4_RCERF_REG .....	4417
23-414. MCBSP4_XCERE_REG.....	4417
23-415. Register Call Summary for Register MCBSP4_XCERE_REG .....	4417

23-416. MCBSP_LP_XCERF_REG .....	4418
23-417. Register Call Summary for Register MCBSP_LP_XCERF_REG .....	4418
23-418. MCBSP_LP_RCERF_REG .....	4418
23-419. Register Call Summary for Register MCBSP_LP_RCERF_REG .....	4418
23-420. MCBSP_LP_RCERH_REG .....	4419
23-421. Register Call Summary for Register MCBSP_LP_RCERH_REG .....	4419
23-422. MCBSP_LP_XCERH_REG .....	4419
23-423. Register Call Summary for Register MCBSP_LP_XCERH_REG .....	4419
23-424. MCBSP_LP_XCERH_REG .....	4420
23-425. Register Call Summary for Register MCBSP_LP_XCERH_REG .....	4420
23-426. MCBSP_LP_REV_REG .....	4420
23-427. Register Call Summary for Register MCBSP_LP_REV_REG .....	4420
23-428. MCBSP_LP_RINTCLR_REG .....	4420
23-429. Register Call Summary for Register MCBSP_LP_RINTCLR_REG .....	4421
23-430. MCBSP_LP_XINTCLR_REG .....	4421
23-431. Register Call Summary for Register MCBSP_LP_XINTCLR_REG .....	4421
23-432. MCBSP_LP_ROVFLCLR_REG .....	4421
23-433. Register Call Summary for Register MCBSP_LP_ROVFLCLR_REG .....	4421
23-434. MCBSP_LP_SYSCONFIG_REG .....	4422
23-435. Register Call Summary for Register MCBSP_LP_SYSCONFIG_REG .....	4422
23-436. MCBSP_LP_THRSH2_REG .....	4423
23-437. Register Call Summary for Register MCBSP_LP_THRSH2_REG .....	4423
23-438. MCBSP_LP_THRSH1_REG .....	4423
23-439. Register Call Summary for Register MCBSP_LP_THRSH1_REG .....	4424
23-440. MCBSP_LP_IRQSTATUS_REG .....	4424
23-441. Register Call Summary for Register MCBSP_LP_IRQSTATUS_REG .....	4426
23-442. MCBSP_LP_IRQENABLE_REG .....	4426
23-443. Register Call Summary for Register MCBSP_LP_IRQENABLE_REG .....	4427
23-444. MCBSP_LP_WAKEUPEN_REG .....	4428
23-445. Register Call Summary for Register MCBSP_LP_WAKEUPEN_REG .....	4429
23-446. MCBSP_LP_XCCR_REG .....	4429
23-447. Register Call Summary for Register MCBSP_LP_XCCR_REG .....	4430
23-448. MCBSP_LP_RCCR_REG .....	4431
23-449. Register Call Summary for Register MCBSP_LP_RCCR_REG .....	4431
23-450. MCBSP_LP_XBUFFSTAT_REG .....	4432
23-451. Register Call Summary for Register MCBSP_LP_XBUFFSTAT_REG .....	4432
23-452. MCBSP_LP_RBUFFSTAT_REG .....	4432
23-453. Register Call Summary for Register MCBSP_LP_RBUFFSTAT_REG .....	4432
23-454. Modes .....	4434
23-455. I/O .....	4434
23-456. Integration Attributes .....	4440
23-457. Clocks and Resets .....	4440
23-458. Hardware Requests .....	4440
23-459. Local Power-Management Features .....	4443
23-460. Interrupt Events .....	4444
23-461. McPDM DMA Requests .....	4446
23-462. Global Initialization of Surrounding Modules .....	4456
23-463. McPDM Downlink Path Global Initialization .....	4458
23-464. McPDM Uplink Path Global Initialization .....	4458

23-465. FIFO-Downlink-Full Event Servicing .....	4459
23-466. FIFO-Downlink-Empty Event Servicing .....	4459
23-467. FIFO-Downlink-Almost-Empty Event Servicing .....	4460
23-468. FIFO Downlink Write-Request Event Servicing .....	4460
23-469. FIFO-Uplink-Full Event Servicing .....	4460
23-470. FIFO-Uplink-Almost-Full Event Servicing .....	4460
23-471. FIFO-Uplink-Empty Event Servicing .....	4461
23-472. FIFO-Uplink-Read-Request Event Servicing .....	4461
23-473. McPDM Instance Summary.....	4462
23-474. McPDM Register Mapping Summary.....	4462
23-475. MCPDM_REVISION .....	4463
23-476. Register Call Summary for Register MCPDM_REVISION.....	4463
23-477. MCPDM_SYSCONFIG .....	4463
23-478. Register Call Summary for Register MCPDM_SYSCONFIG.....	4464
23-479. MCPDM_IRQSTATUS_RAW.....	4464
23-480. Register Call Summary for Register MCPDM_IRQSTATUS_RAW .....	4465
23-481. MCPDM_IRQSTATUS .....	4465
23-482. Register Call Summary for Register MCPDM_IRQSTATUS .....	4467
23-483. MCPDM_IRQENABLE_SET .....	4467
23-484. Register Call Summary for Register MCPDM_IRQENABLE_SET .....	4468
23-485. MCPDM_IRQENABLE_CLR.....	4468
23-486. Register Call Summary for Register MCPDM_IRQENABLE_CLR .....	4469
23-487. MCPDM_IRQWAKEEN .....	4470
23-488. Register Call Summary for Register MCPDM_IRQWAKEEN .....	4470
23-489. MCPDM_DMAENABLE_SET .....	4470
23-490. Register Call Summary for Register MCPDM_DMAENABLE_SET .....	4471
23-491. MCPDM_DMAENABLE_CLR .....	4471
23-492. Register Call Summary for Register MCPDM_DMAENABLE_CLR .....	4471
23-493. MCPDM_DMAWAKEEN .....	4472
23-494. Register Call Summary for Register MCPDM_DMAWAKEEN.....	4472
23-495. MCPDM_CTRL.....	4472
23-496. Register Call Summary for Register MCPDM_CTRL .....	4473
23-497. MCPDM_DN_DATA .....	4474
23-498. Register Call Summary for Register MCPDM_DN_DATA .....	4474
23-499. MCPDM_UP_DATA .....	4474
23-500. Register Call Summary for Register MCPDM_UP_DATA .....	4474
23-501. MCPDM_FIFO_CTRL_DN.....	4474
23-502. Register Call Summary for Register MCPDM_FIFO_CTRL_DN .....	4475
23-503. MCPDM_FIFO_CTRL_UP .....	4475
23-504. Register Call Summary for Register MCPDM_FIFO_CTRL_UP .....	4475
23-505. MCPDM_DN_OFFSET .....	4475
23-506. Register Call Summary for Register MCPDM_DN_OFFSET .....	4476
23-507. Input/Output.....	4480
23-508. Integration Attributes.....	4482
23-509. Clocks and Resets .....	4483
23-510. Hardware Requests.....	4483
23-511. abe_dmic_clkx Generation.....	4485
23-512. Reset Description .....	4485
23-513. Local Power-Management Features.....	4485

23-514. Interrupt Events .....	4487
23-515. DMIC DMA Request .....	4489
23-516. Audio Data Path Registers.....	4491
23-517. Global Initialization of Surrounding Modules .....	4492
23-518. DMIC Global Initialization .....	4492
23-519. Register Call Summary for Polling Mode .....	4493
23-520. Interrupt Mode .....	4493
23-521. DMA Mode .....	4493
23-522. Register Call Summary for the Interrupt Handler .....	4494
23-523. DMIC Instance Summary .....	4496
23-524. DMIC Register Mapping Summary .....	4496
23-525. DMIC_REVISION .....	4497
23-526. Register Call Summary for Register DMIC_REVISION.....	4497
23-527. DMIC_SYSCONFIG .....	4497
23-528. Register Call Summary for Register DMIC_SYSCONFIG.....	4498
23-529. DMIC_IRQSTATUS_RAW .....	4498
23-530. Register Call Summary for Register DMIC_IRQSTATUS_RAW .....	4499
23-531. DMIC_IRQSTATUS.....	4499
23-532. Register Call Summary for Register DMIC_IRQSTATUS .....	4500
23-533. DMIC_IRQENABLE_SET .....	4500
23-534. Register Call Summary for Register DMIC_IRQENABLE_SET.....	4501
23-535. DMIC_IRQENABLE_CLR .....	4501
23-536. Register Call Summary for Register DMIC_IRQENABLE_CLR .....	4502
23-537. DMIC_IRQWAKEEN.....	4502
23-538. Register Call Summary for Register DMIC_IRQWAKEEN .....	4502
23-539. DMIC_DMAENABLE_SET .....	4502
23-540. Register Call Summary for Register DMIC_DMAENABLE_SET .....	4503
23-541. DMIC_DMAENABLE_CLR.....	4503
23-542. Register Call Summary for Register DMIC_DMAENABLE_CLR .....	4503
23-543. DMIC_DMAWAKEEN .....	4504
23-544. Register Call Summary for Register DMIC_DMAWAKEEN .....	4504
23-545. DMIC_CTRL .....	4504
23-546. Register Call Summary for Register DMIC_CTRL.....	4505
23-547. DMIC_DATA_REG.....	4505
23-548. Register Call Summary for Register DMIC_DATA_REG .....	4505
23-549. DMIC_FIFO_CTRL .....	4506
23-550. Register Call Summary for Register DMIC_FIFO_CTRL .....	4506
23-551. DMIC_FIFO_DMIC1R_DATA.....	4506
23-552. Register Call Summary for Register DMIC_FIFO_DMIC1R_DATA .....	4506
23-553. DMIC_FIFO_DMIC1L_DATA .....	4507
23-554. Register Call Summary for Register DMIC_FIFO_DMIC1L_DATA .....	4507
23-555. DMIC_FIFO_DMIC2R_DATA.....	4507
23-556. Register Call Summary for Register DMIC_FIFO_DMIC2R_DATA .....	4507
23-557. DMIC_FIFO_DMIC2L_DATA .....	4507
23-558. Register Call Summary for Register DMIC_FIFO_DMIC2L_DATA .....	4508
23-559. DMIC_FIFO_DMIC3R_DATA.....	4508
23-560. Register Call Summary for Register DMIC_FIFO_DMIC3R_DATA .....	4508
23-561. DMIC_FIFO_DMIC3L_DATA .....	4508
23-562. Register Call Summary for Register DMIC_FIFO_DMIC3L_DATA .....	4508

23-563. I/O .....	4511
23-564. Biphas-Mark Encoder .....	4515
23-565. Preamble Codes .....	4516
23-566. Integration Attributes.....	4518
23-567. Clocks and Resets .....	4518
23-568. Hardware Requests.....	4518
23-569. Local Power-Management Features.....	4523
23-570. Channel Status and User Data for Each DIT Block .....	4525
23-571. TX Events .....	4528
23-572. Global Initialization of Surrounding Modules .....	4533
23-573. McASP Global Initialization .....	4533
23-574. Transmit Format Unit Configuration .....	4534
23-575. Transmit Frame-Synchronization Generator Configuration.....	4534
23-576. Transmit Clock Generator Configuration.....	4535
23-577. McASP Pins Functional Configuration .....	4535
23-578. McASP Mute Input/Output Trigger Condition Settings .....	4535
23-579. DIT-Specific Subframe Fields Configuration.....	4536
23-580. Register Call Summary for Main Sequence – McASP DIT Transmission Polling Method .....	4537
23-581. Subprocess Call Summary for Main Sequence – McASP DIT Transmission Polling Method .....	4537
23-582. McASP Interrupt Transmission Model.....	4539
23-583. Register Call Summary for Subsequence – McASP DIT Transmission Startup Procedure.....	4539
23-584. McASP DMA Transmission Model with Interrupt Events Servicing .....	4539
23-585. Register Call Summary for Subsequence – McASP DMA Transmission Procedure .....	4540
23-586. Register Call Summary for McASP Interrupt Events Servicing .....	4541
23-587. Subprocess Call Summary for Interrupt Events Servicing.....	4541
23-588. Register Call Summary for McASP Error Handling.....	4542
23-589. McASP Instance Summary .....	4543
23-590. McASP Register Mapping Summary .....	4543
23-591. MCASP_PID .....	4545
23-592. Register Call Summary for Register MCASP_PID.....	4545
23-593. MCASP_SYSCONFIG.....	4545
23-594. Register Call Summary for Register MCASP_SYSCONFIG .....	4545
23-595. MCASP_PFUNC .....	4546
23-596. Register Call Summary for Register MCASP_PFUNC.....	4546
23-597. MCASP_PDIR .....	4546
23-598. Register Call Summary for Register MCASP_PDIR.....	4547
23-599. MCASP_PDOUT .....	4548
23-600. Register Call Summary for Register MCASP_PDOUT.....	4548
23-601. MCASP_PDIN .....	4549
23-602. Register Call Summary for Register MCASP_PDIN.....	4549
23-603. MCASP_PDSET .....	4549
23-604. Register Call Summary for Register MCASP_PDSET .....	4550
23-605. MCASP_PDCLR .....	4550
23-606. Register Call Summary for Register MCASP_PDCLR.....	4551
23-607. MCASP_GBLCTL .....	4551
23-608. Register Call Summary for Register MCASP_GBLCTL.....	4552
23-609. MCASP_AMUTE .....	4552
23-610. Register Call Summary for Register MCASP_AMUTE.....	4554
23-611. MCASP_TXDITCTL.....	4554

23-612. Register Call Summary for Register MCASP_TXDITCTL .....	4554
23-613. MCASP_TXMASK .....	4554
23-614. Register Call Summary for Register MCASP_TXMASK .....	4555
23-615. MCASP_TXFMT .....	4555
23-616. Register Call Summary for Register MCASP_TXFMT .....	4556
23-617. MCASP_TXFMCTL .....	4556
23-618. Register Call Summary for Register MCASP_TXFMCTL.....	4557
23-619. MCASP_ACLKXCTL.....	4557
23-620. Register Call Summary for Register MCASP_ACLKXCTL .....	4557
23-621. MCASP_AHCLKXCTL.....	4557
23-622. Register Call Summary for Register MCASP_AHCLKXCTL .....	4558
23-623. MCASP_TXTDM .....	4558
23-624. Register Call Summary for Register MCASP_TXTDM.....	4558
23-625. MCASP_EVTCTLX .....	4559
23-626. Register Call Summary for Register MCASP_EVTCTLX.....	4560
23-627. MCASP_TXSTAT .....	4560
23-628. Register Call Summary for Register MCASP_TXSTAT.....	4561
23-629. MCASP_TXTDMSLOT .....	4561
23-630. Register Call Summary for Register MCASP_TXTDMSLOT.....	4561
23-631. MCASP_TXCLKCHK .....	4562
23-632. Register Call Summary for Register MCASP_TXCLKCHK.....	4562
23-633. MCASP_TXEVTCTL.....	4562
23-634. Register Call Summary for Register MCASP_TXEVTCTL .....	4563
23-635. MCASP_DITCSRAi .....	4563
23-636. Register Call Summary for Register MCASP_DITCSRAi.....	4563
23-637. MCASP_DITCSRBi .....	4563
23-638. Register Call Summary for Register MCASP_DITCSRBi.....	4564
23-639. MCASP_DITUDRAi .....	4564
23-640. Register Call Summary for Register MCASP_DITUDRAi .....	4564
23-641. MCASP_DITUDRBi .....	4564
23-642. Register Call Summary for Register MCASP_DITUDRBi .....	4564
23-643. MCASP_XRSRCTL0 .....	4565
23-644. Register Call Summary for Register MCASP_XRSRCTL0.....	4565
23-645. MCASP_TXBUF0 .....	4565
23-646. Register Call Summary for Register MCASP_TXBUF0.....	4566
23-647. SLIMbus1 Signals Description .....	4571
23-648. SLIMbus2 Signals Description .....	4571
23-649. SLIMbus Transfer Protocols .....	4573
23-650. Integration Attributes.....	4577
23-651. Clocks and Resets .....	4578
23-652. Hardware Requests.....	4578
23-653. Local Power-Management Features.....	4582
23-654. Reset and Boot Control Summary .....	4583
23-655. Examples of AUX + DATA Mapping in 32-bit Read/Write Accesses.....	4587
23-656. HILEVEL/Empty Status of a 4-Word Receive FIFO, for all DMA Threshold Values .....	4590
23-657. LOLEVEL/Full Status of a 4-Word Transmit FIFO, for all DMA Threshold Values .....	4590
23-658. Segment Composition .....	4591
23-659. Register Fields Used to Configure Transfer Protocols .....	4593
23-660. Register Fields Used for Isochronous Transfer Protocol .....	4593

23-661. Register Fields Used for Pushed Transfer Protocol .....	4594
23-662. Register fields used for Pulled Transfer Protocol .....	4595
23-663. "Plain" Asynchronous Endpoint Configurations .....	4596
23-664. Register fields used for "Plain" Asynchronous Transfer Protocol .....	4597
23-665. Extended Asynchronous Endpoint Configurations .....	4598
23-666. Register fields used for Extended Asynchronous Transfer Protocol .....	4599
23-667. Enumeration Address (EA) makeup .....	4602
23-668. Events .....	4605
23-669. Interrupts summary .....	4605
23-670. Global Initialization of Surrounding Modules .....	4616
23-671. SLIMbus component reset .....	4616
23-672. SLIMbus idle mode configuration .....	4616
23-673. SLIMbus FIFOs mapping and setup .....	4617
23-674. SLIMbus transfer protocol configuration sequence .....	4617
23-675. SLIMbus transfer protocol configuration .....	4618
23-676. SLIMbus interrupts enabling .....	4618
23-677. SLIMbus DMA configuration .....	4618
23-678. Clock Receiver Component Boot .....	4619
23-679. Clock Source Component Boot .....	4619
23-680. Register Settings for the SLIMbus Message Transmit Procedure, Polling Method .....	4620
23-681. Register Settings for the SLIMbus message receive procedure, Polling Method .....	4622
23-682. Register Settings for the SLIMbus message transmit procedure, Interrupt Method .....	4624
23-683. Register Settings for the SLIMbus message receive procedure, Transmit Method .....	4625
23-684. Data word receiving .....	4626
23-685. Data word transmitting .....	4626
23-686. SLIMbus Instance Summary .....	4628
23-687. SLIMBUS1 Register Mapping Summary .....	4628
23-688. SLIMBUS2 Register Mapping Summary .....	4630
23-689. SLIMBUS_CMP_REVISION .....	4632
23-690. Register Call Summary for Register SLIMBUS_CMP_REVISION .....	4632
23-691. SLIMBUS_CMP_HWINFO .....	4632
23-692. Register Call Summary for Register SLIMBUS_CMP_HWINFO .....	4633
23-693. Reset values for SLIMBUS_CMP_HWINFO .....	4633
23-694. SLIMBUS_CMP_SYSCONFIG .....	4633
23-695. Register Call Summary for Register SLIMBUS_CMP_SYSCONFIG .....	4634
23-696. SLIMBUS_CMP_IRQSTATUS_RAW .....	4634
23-697. Register Call Summary for Register SLIMBUS_CMP_IRQSTATUS_RAW .....	4637
23-698. SLIMBUS_CMP_IRQSTATUS .....	4637
23-699. Register Call Summary for Register SLIMBUS_CMP_IRQSTATUS .....	4640
23-700. SLIMBUS_CMP_IRQENABLE_SET .....	4640
23-701. Register Call Summary for Register SLIMBUS_CMP_IRQENABLE_SET .....	4643
23-702. SLIMBUS_CMP_IRQENABLE_CLR .....	4643
23-703. Register Call Summary for Register SLIMBUS_CMP_IRQENABLE_CLR .....	4645
23-704. SLIMBUS_CMP_DMAENABLE_SET .....	4646
23-705. Register Call Summary for Register SLIMBUS_CMP_DMAENABLE_SET .....	4648
23-706. SLIMBUS_CMP_DMAENABLE_CLR .....	4648
23-707. Register Call Summary for Register SLIMBUS_CMP_DMAENABLE_CLR .....	4650
23-708. SLIMBUS_CMP_IV .....	4650
23-709. Register Call Summary for Register SLIMBUS_CMP_IV .....	4650

23-710. SLIMBUS_CMP_MI_PC .....	4651
23-711. Register Call Summary for Register SLIMBUS_CMP_MI_PC .....	4651
23-712. SLIMBUS_SMT_INFO.....	4651
23-713. Register Call Summary for Register SLIMBUS_SMT_INFO .....	4652
23-714. SLIMBUS_SMT_MESSAGE.....	4653
23-715. Register Call Summary for Register SLIMBUS_SMT_MESSAGE .....	4653
23-716. SLIMBUS_SMT_CONTROL .....	4653
23-717. Register Call Summary for Register SLIMBUS_SMT_CONTROL .....	4654
23-718. SLIMBUS_SMT_FIFO_SETUP .....	4654
23-719. Register Call Summary for Register SLIMBUS_SMT_FIFO_SETUP .....	4654
23-720. SLIMBUS_SMR_INFO .....	4655
23-721. Register Call Summary for Register SLIMBUS_SMR_INFO .....	4656
23-722. SLIMBUS_SMR_MESSAGE .....	4656
23-723. Register Call Summary for Register SLIMBUS_SMR_MESSAGE .....	4656
23-724. SLIMBUS_SMR_CONTROL.....	4657
23-725. Register Call Summary for Register SLIMBUS_SMR_CONTROL .....	4657
23-726. SLIMBUS_SMR_FIFO_SETUP .....	4657
23-727. Register Call Summary for Register SLIMBUS_SMR_FIFO_SETUP .....	4658
23-728. SLIMBUS_FL_INFO .....	4658
23-729. Register Call Summary for Register SLIMBUS_FL_INFO.....	4660
23-730. SLIMBUS_FL_CONTROL .....	4660
23-731. Register Call Summary for Register SLIMBUS_FL_CONTROL.....	4662
23-732. SLIMBUS_FL_SM.....	4662
23-733. Register Call Summary for Register SLIMBUS_FL_SM .....	4663
23-734. SLIMBUS_FL_CG.....	4664
23-735. Register Call Summary for Register SLIMBUS_FL_CG .....	4664
23-736. SLIMBUS_FL_RF .....	4665
23-737. Register Call Summary for Register SLIMBUS_FL_RF.....	4665
23-738. SLIMBUS_FR_INFO.....	4665
23-739. Register Call Summary for Register SLIMBUS_FR_INFO .....	4666
23-740. SLIMBUS_FR_CLOCK_SOURCE .....	4667
23-741. Register Call Summary for Register SLIMBUS_FR_CLOCK_SOURCE .....	4667
23-742. SLIMBUS_FR_CONTROL .....	4668
23-743. Register Call Summary for Register SLIMBUS_FR_CONTROL .....	4668
23-744. SLIMBUS_FR_FRAMER_HANDOVER .....	4669
23-745. Register Call Summary for Register SLIMBUS_FR_FRAMER_HANDOVER .....	4669
23-746. SLIMBUS_FR_CLOCK_PAUSE .....	4669
23-747. Register Call Summary for Register SLIMBUS_FR_CLOCK_PAUSE.....	4670
23-748. SLIMBUS_DEV_LA_i.....	4670
23-749. Register Call Summary for Register SLIMBUS_DEV_LA_i .....	4671
23-750. SLIMBUS_DEV_EA_LO_j .....	4671
23-751. Register Call Summary for Register SLIMBUS_DEV_EA_LO_j.....	4671
23-752. SLIMBUS_DEV_EA_HI_j .....	4671
23-753. Register Call Summary for Register SLIMBUS_DEV_EA_HI_j.....	4672
23-754. SLIMBUS_DCT_INFO_j.....	4672
23-755. Register Call Summary for Register SLIMBUS_DCT_INFO_j .....	4673
23-756. SLIMBUS_DCT_FIFO_SETUP1_j.....	4673
23-757. Register Call Summary for Register SLIMBUS_DCT_FIFO_SETUP1_j .....	4674
23-758. SLIMBUS_DCT_FIFO_SETUP2_j.....	4674



23-759. Register Call Summary for Register SLIMBUS_DCT_FIFO_SETUP2_j .....	4675
23-760. SLIMBUS_DCT_FIFO_STATUS_j.....	4675
23-761. Register Call Summary for Register SLIMBUS_DCT_FIFO_STATUS_j .....	4676
23-762. SLIMBUS_DCT_MAP_j .....	4677
23-763. Register Call Summary for Register SLIMBUS_DCT_MAP_j.....	4677
23-764. SLIMBUS_DCT_CONFIG1_j .....	4677
23-765. Register Call Summary for Register SLIMBUS_DCT_CONFIG1_j.....	4679
23-766. SLIMBUS_DCT_CONFIG2_j .....	4679
23-767. Register Call Summary for Register SLIMBUS_DCT_CONFIG2_j.....	4680
23-768. SLIMBUS_DCT_DATA_j.....	4680
23-769. Register Call Summary for Register SLIMBUS_DCT_DATA_j .....	4680
23-770. SLIMBUS_DCR_INFO_j .....	4680
23-771. Register Call Summary for Register SLIMBUS_DCR_INFO_j.....	4681
23-772. SLIMBUS_DCR_FIFO_SETUP1_j.....	4681
23-773. Register Call Summary for Register SLIMBUS_DCR_FIFO_SETUP1_j .....	4682
23-774. SLIMBUS_DCR_FIFO_SETUP2_j.....	4682
23-775. Register Call Summary for Register SLIMBUS_DCR_FIFO_SETUP2_j .....	4683
23-776. SLIMBUS_DCR_FIFO_STATUS_j.....	4683
23-777. Register Call Summary for Register SLIMBUS_DCR_FIFO_STATUS_j.....	4685
23-778. SLIMBUS_DCR_MAP_j .....	4685
23-779. Register Call Summary for Register SLIMBUS_DCR_MAP_j .....	4685
23-780. SLIMBUS_DCR_CONFIG1_j.....	4686
23-781. Register Call Summary for Register SLIMBUS_DCR_CONFIG1_j .....	4688
23-782. SLIMBUS_DCR_CONFIG2_j.....	4688
23-783. Register Call Summary for Register SLIMBUS_DCR_CONFIG2_j .....	4688
23-784. SLIMBUS_DCR_DATA_j.....	4689
23-785. Register Call Summary for Register SLIMBUS_DCR_DATA_j .....	4689
23-786. HSI I/O Signals .....	4691
23-787. Integration Attributes.....	4693
23-788. Clocks and Resets .....	4694
23-789. Hardware Requests.....	4694
23-790. Local Power-Management Features.....	4698
23-791. Events .....	4699
23-792. Threshold for Transmit FIFO Interrupt/DMA Request.....	4714
23-793. Global Initialization of Surrounding Modules for HSI .....	4719
23-794. HSI Global Initialization.....	4719
23-795. HSI Configure HSI Transmitter .....	4719
23-796. HSI Configure HSI Receiver .....	4720
23-797. Register call summary for Sub-sequence - HSI Transmit in Polling Method.....	4721
23-798. Event Servicing in HSI Receive in Polling Method .....	4721
23-799. HSI Interrupt Mode.....	4722
23-800. Enable HSI Transmit Interrupts .....	4722
23-801. Register call summary for Sub-sequence - Event Servicing in HSI Transmission Mode.....	4723
23-802. Enable HSI Receive Interrupts .....	4724
23-803. Register call summary for Sub-sequence - Event Servicing in HSI Reception Mode.....	4725
23-804. HSI DMA Configuration .....	4726
23-805. Configure DMA Receive Mode .....	4726
23-806. Configure DMA Transmit Mode .....	4726
23-807. Enable DMA Interrupts .....	4727

23-808. Event Servicing in DMA Transmit/Receive Mode .....	4727
23-809. Enable HSI Error Interrupts .....	4727
23-810. Event Servicing Error .....	4727
23-811. HSI Instance Summary .....	4728
23-812. HSI_TOP Registers Mapping Summary .....	4728
23-813. HSI_REVISION .....	4729
23-814. Register Call Summary for Register HSI_REVISION .....	4729
23-815. HSI_HWINFO .....	4729
23-816. Register Call Summary for Register HSI_HWINFO .....	4730
23-817. HSI_SYSCONFIG .....	4730
23-818. Register Call Summary for Register HSI_SYSCONFIG .....	4730
23-819. HSI_SYSSTATUS .....	4731
23-820. Register Call Summary for Register HSI_SYSSTATUS .....	4731
23-821. HSI_Pp_M_IRQrU_STATUS .....	4731
23-822. Register Call Summary for Register HSI_Pp_M_IRQrU_STATUS .....	4732
23-823. HSI_Pp_M_IRQrU_ENABLE .....	4732
23-824. Register Call Summary for Register HSI_Pp_M_IRQrU_ENABLE .....	4732
23-825. HSI_Pp_D_IRQrU_STATUS .....	4733
23-826. Register Call Summary for Register HSI_Pp_D_IRQrU_STATUS .....	4733
23-827. HSI_Pp_D_IRQrU_ENABLE .....	4733
23-828. Register Call Summary for Register HSI_Pp_D_IRQrU_ENABLE .....	4734
23-829. HSI_DMA_M_IRQSTATUS .....	4734
23-830. Register Call Summary for Register HSI_DMA_M_IRQSTATUS .....	4734
23-831. HSI_DMA_M_IRQENABLE .....	4735
23-832. Register Call Summary for Register HSI_DMA_M_IRQENABLE .....	4735
23-833. HSI_Pp_M_IRQr_STATUS .....	4735
23-834. Register Call Summary for Register HSI_Pp_M_IRQr_STATUS .....	4736
23-835. HSI_Pp_M_IRQr_ENABLE .....	4736
23-836. Register Call Summary for Register HSI_Pp_M_IRQr_ENABLE .....	4736
23-837. HSI_DMA_D_IRQSTATUS .....	4736
23-838. Register Call Summary for Register HSI_DMA_D_IRQSTATUS .....	4737
23-839. HSI_DMA_D_IRQENABLE .....	4737
23-840. Register Call Summary for Register HSI_DMA_D_IRQENABLE .....	4737
23-841. HSI_Pp_D_IRQr_STATUS .....	4737
23-842. Register Call Summary for Register HSI_Pp_D_IRQr_STATUS .....	4738
23-843. HSI_Pp_D_IRQr_ENABLE .....	4738
23-844. Register Call Summary for Register HSI_Pp_D_IRQr_ENABLE .....	4738
23-845. HSI_Pp_WAKE .....	4739
23-846. Register Call Summary for Register HSI_Pp_WAKE .....	4739
23-847. HSI_Pp_CLEAR_WAKE .....	4739
23-848. Register Call Summary for Register HSI_Pp_CLEAR_WAKE .....	4739
23-849. HSI_Pp_SET_WAKE .....	4740
23-850. Register Call Summary for Register HSI_Pp_SET_WAKE .....	4740
23-851. DMA_GCR .....	4740
23-852. Register Call Summary for Register DMA_GCR .....	4741
23-853. DMA_GRST .....	4741
23-854. Register Call Summary for Register DMA_GRST .....	4741
23-855. HSI_DMA_CHANNELS Registers Mapping Summary .....	4741
23-856. DMA_CCR_CSDP_i .....	4742

23-857. Register Call Summary for Register DMA_CCR_CSDP_i.....	4743
23-858. DMA_CSR_CCIR_i .....	4743
23-859. Register Call Summary for Register DMA_CSR_CCIR_i.....	4744
23-860. DMA_CSSA_i.....	4744
23-861. Register Call Summary for Register DMA_CSSA_i .....	4744
23-862. DMA_CDSA_i .....	4745
23-863. Register Call Summary for Register DMA_CDSA_i .....	4745
23-864. DMA_CEN_i .....	4745
23-865. Register Call Summary for Register DMA_CEN_i.....	4745
23-866. DMA_CDAC_CSAC_i .....	4746
23-867. Register Call Summary for Register DMA_CDAC_CSAC_i .....	4746
23-868. HSI_PORTS Registers Mapping Summary.....	4746
23-869. HST_ID_Pp .....	4747
23-870. Register Call Summary for Register HST_ID_Pp.....	4748
23-871. HST_MODE_Pp.....	4748
23-872. Register Call Summary for Register HST_MODE_Pp .....	4748
23-873. HST_FRAMESIZE_Pp .....	4749
23-874. Register Call Summary for Register HST_FRAMESIZE_Pp.....	4749
23-875. HST_TXSTATE_Pp.....	4749
23-876. Register Call Summary for Register HST_TXSTATE_Pp .....	4750
23-877. HST_BUFSTATE_Pp.....	4750
23-878. Register Call Summary for Register HST_BUFSTATE_Pp .....	4750
23-879. HST_DIVISOR_Pp.....	4750
23-880. Register Call Summary for Register HST_DIVISOR_Pp .....	4751
23-881. HST_BREAK_Pp.....	4751
23-882. Register Call Summary for Register HST_BREAK_Pp .....	4751
23-883. HST_CHANNELS_Pp .....	4751
23-884. Register Call Summary for Register HST_CHANNELS_Pp.....	4752
23-885. HST_ARBMODE_Pp .....	4752
23-886. Register Call Summary for Register HST_ARBMODE_Pp.....	4752
23-887. HST_BUFFER_Pp_CHN_i.....	4753
23-888. Register Call Summary for Register HST_BUFFER_Pp_CHN_i .....	4753
23-889. HST_SWAPBUFFER_Pp_CHN_i.....	4753
23-890. Register Call Summary for Register HST_SWAPBUFFER_Pp_CHN_i .....	4753
23-891. HST_MAPPINGf .....	4754
23-892. Register Call Summary for Register HST_MAPPINGf .....	4754
23-893. HSR_ID_Pp.....	4755
23-894. Register Call Summary for Register HSR_ID_Pp .....	4755
23-895. HSR_MODE_Pp .....	4755
23-896. Register Call Summary for Register HSR_MODE_Pp .....	4756
23-897. HSR_FRAMESIZE_Pp .....	4756
23-898. Register Call Summary for Register HSR_FRAMESIZE_Pp.....	4756
23-899. HSR_RXSTATE_Pp .....	4756
23-900. Register Call Summary for Register HSR_RXSTATE_Pp.....	4757
23-901. HSR_BUFSTATE_Pp .....	4757
23-902. Register Call Summary for Register HSR_BUFSTATE_Pp .....	4757
23-903. HSR_BREAK_Pp .....	4757
23-904. Register Call Summary for Register HSR_BREAK_Pp .....	4758
23-905. HSR_ERROR_Pp.....	4758

23-906. Register Call Summary for Register HSR_ERROR_Pp .....	4758
23-907. HSR_ERRORACK_Pp .....	4759
23-908. Register Call Summary for Register HSR_ERRORACK_Pp.....	4759
23-909. HSR_CHANNELS_Pp .....	4759
23-910. Register Call Summary for Register HSR_CHANNELS_Pp .....	4760
23-911. HSR_OVERRUN_Pp .....	4760
23-912. Register Call Summary for Register HSR_OVERRUN_Pp.....	4760
23-913. HSR_OVERRUNACK_Pp.....	4761
23-914. Register Call Summary for Register HSR_OVERRUNACK_Pp .....	4761
23-915. HSR_COUNTERS_Pp .....	4761
23-916. Register Call Summary for Register HSR_COUNTERS_Pp .....	4761
23-917. HSR_BUFFER_Pp_CHN_i .....	4762
23-918. Register Call Summary for Register HSR_BUFFER_Pp_CHN_i.....	4762
23-919. HSR_SWAPBUFFER_Pp_CHN_i .....	4762
23-920. Register Call Summary for Register HSR_SWAPBUFFER_Pp_CHN_i.....	4762
23-921. HSR_MAPPINGf .....	4763
23-922. Register Call Summary for Register HSR_MAPPINGf.....	4763
23-923. HSR_DLL .....	4763
23-924. Register Call Summary for Register HSR_DLL .....	4764
23-925. HSR_DIVISOR_Pp .....	4764
23-926. Register Call Summary for Register HSR_DIVISOR_Pp.....	4764
23-927. ULPI - 12-Pin/8-Bit Data SDR Version I/O Description .....	4772
23-928. ULPI TLL Interfaces - 12-Pin/8-Bit Data SDR Version I/O Description .....	4774
23-929. Signaling Between HS USB Host Subsystem and 6-Pin Unidirectional USB Transceiver (DAT/SE0 Signaling) .....	4775
23-930. Signaling Between HS USB Host Subsystem and 6-Pin Unidirectional USB Transceiver (DP/DM Signaling) .....	4775
23-931. Signaling Between HS USB Host Subsystem and 3-Pin Bidirectional USB Transceiver Using DAT/SE0 Signaling .....	4776
23-932. Signaling Between HS USB Host Subsystem and 4-Pin Bidirectional USB Transceiver Using DP/DM Signaling .....	4776
23-933. Pullup/Pulldown Configuration for DP/DM Encoding .....	4787
23-934. Pullup/Pulldown Configuration for DAT/SE0 Encoding .....	4788
23-935. I/O Description.....	4789
23-936. HSIC Interface .....	4790
23-937. Integration Attributes.....	4793
23-938. Clocks and Resets .....	4793
23-939. Hardware Requests .....	4793
23-940. HS USB Host Controller Standby Mode Settings.....	4796
23-941. HS USB Host Controller SIDLEMODE Settings .....	4797
23-942. HS USBTLL Module SIDLEMODE Settings .....	4799
23-943. USBTLL Registers Affected by the SAR Context.....	4799
23-944. USBTLL Channel Configuration.....	4800
23-945. VBUS Level Software Reporting for Serial Transceiver Configuration.....	4802
23-946. Emulation of VBUS Levels for UTMI-to-ULPI TLL Mode .....	4802
23-947. Serial Mode Description, Signal Functionality .....	4804
23-948. Pullup Enable Emulation in Serial TLL Modes .....	4805
23-949. Global Initialization of Surrounding Modules .....	4806
23-950. HS Multiport USB Host Subsystem Global Initialization .....	4806
23-951. Register Call Summary for Selecting and Configuring High-Speed USB Host Subsystem Connectivity..	4807

23-952. USB_Host_HS Instance Summary .....	4809
23-953. USBTLLHS_config Register Mapping Summary.....	4809
23-954. USBTLL_REVISION .....	4810
23-955. Register Call Summary for Register USBTLL_REVISION.....	4810
23-956. USBTLL_HWINFO .....	4810
23-957. Register Call Summary for Register USBTLL_HWINFO .....	4810
23-958. USBTLL_SYSCONFIG .....	4810
23-959. Register Call Summary for Register USBTLL_SYSCONFIG.....	4811
23-960. USBTLL_SYSSTATUS .....	4811
23-961. Register Call Summary for Register USBTLL_SYSSTATUS .....	4812
23-962. USBTLL_IRQSTATUS .....	4812
23-963. Register Call Summary for Register USBTLL_IRQSTATUS .....	4812
23-964. USBTLL_IRQENABLE .....	4812
23-965. Register Call Summary for Register USBTLL_IRQENABLE .....	4813
23-966. TLL_SHARED_CONF .....	4813
23-967. Register Call Summary for Register TLL_SHARED_CONF.....	4814
23-968. TLL_CHANNEL_CONF_i .....	4814
23-969. Register Call Summary for Register TLL_CHANNEL_CONF_i.....	4817
23-970. USBTLL_SAR_CNTX_j .....	4817
23-971. Register Call Summary for Register USBTLL_SAR_CNTX_j.....	4817
23-972. USBTLLHS_ULPI Register Mapping Summary.....	4817
23-973. VENDOR_ID_LO_i.....	4818
23-974. Register Call Summary for Register VENDOR_ID_LO_i .....	4819
23-975. VENDOR_ID_HI_i.....	4819
23-976. Register Call Summary for Register VENDOR_ID_HI_i .....	4819
23-977. PRODUCT_ID_LO_i .....	4819
23-978. Register Call Summary for Register PRODUCT_ID_LO_i .....	4819
23-979. PRODUCT_ID_HI_i.....	4820
23-980. Register Call Summary for Register PRODUCT_ID_HI_i .....	4820
23-981. FUNCTION_CTRL_i .....	4820
23-982. Register Call Summary for Register FUNCTION_CTRL_i.....	4821
23-983. FUNCTION_CTRL_SET_i .....	4821
23-984. Register Call Summary for Register FUNCTION_CTRL_SET_i.....	4821
23-985. FUNCTION_CTRL_CLR_i .....	4822
23-986. Register Call Summary for Register FUNCTION_CTRL_CLR_i.....	4822
23-987. INTERFACE_CTRL_i.....	4822
23-988. Register Call Summary for Register INTERFACE_CTRL_i .....	4823
23-989. INTERFACE_CTRL_SET_i.....	4824
23-990. Register Call Summary for Register INTERFACE_CTRL_SET_i .....	4824
23-991. INTERFACE_CTRL_CLR_i.....	4825
23-992. Register Call Summary for Register INTERFACE_CTRL_CLR_i .....	4825
23-993. OTG_CTRL_i .....	4826
23-994. Register Call Summary for Register OTG_CTRL_i.....	4826
23-995. OTG_CTRL_SET_i .....	4827
23-996. Register Call Summary for Register OTG_CTRL_SET_i.....	4827
23-997. OTG_CTRL_CLR_i .....	4828
23-998. Register Call Summary for Register OTG_CTRL_CLR_i.....	4828
23-999. USB_INT_EN_RISE_i .....	4829
23-1000. Register Call Summary for Register USB_INT_EN_RISE_i .....	4829

23-1001. USB_INT_EN_RISE_SET_i.....	4830
23-1002. Register Call Summary for Register USB_INT_EN_RISE_SET_i .....	4830
23-1003. USB_INT_EN_RISE_CLR_i.....	4831
23-1004. Register Call Summary for Register USB_INT_EN_RISE_CLR_i .....	4831
23-1005. USB_INT_EN_FALL_i .....	4832
23-1006. Register Call Summary for Register USB_INT_EN_FALL_i.....	4832
23-1007. USB_INT_EN_FALL_SET_i.....	4832
23-1008. Register Call Summary for Register USB_INT_EN_FALL_SET_i .....	4833
23-1009. USB_INT_EN_FALL_CLR_i .....	4834
23-1010. Register Call Summary for Register USB_INT_EN_FALL_CLR_i .....	4834
23-1011. USB_INT_STATUS_i .....	4835
23-1012. Register Call Summary for Register USB_INT_STATUS_i.....	4835
23-1013. USB_INT_LATCH_i .....	4836
23-1014. Register Call Summary for Register USB_INT_LATCH_i.....	4836
23-1015. DEBUG_i .....	4836
23-1016. Register Call Summary for Register DEBUG_i.....	4837
23-1017. SCRATCH_REGISTER_i.....	4837
23-1018. Register Call Summary for Register SCRATCH_REGISTER_i .....	4837
23-1019. SCRATCH_REGISTER_SET_i.....	4837
23-1020. Register Call Summary for Register SCRATCH_REGISTER_SET_i .....	4838
23-1021. SCRATCH_REGISTER_CLR_i.....	4838
23-1022. Register Call Summary for Register SCRATCH_REGISTER_CLR_i .....	4838
23-1023. EXTENDED_SET_ACCESS_i.....	4838
23-1024. Register Call Summary for Register EXTENDED_SET_ACCESS_i .....	4838
23-1025. UTMI_VCONTROL_EN_i.....	4839
23-1026. Register Call Summary for Register UTMI_VCONTROL_EN_i .....	4839
23-1027. UTMI_VCONTROL_EN_SET_i.....	4839
23-1028. Register Call Summary for Register UTMI_VCONTROL_EN_SET_i .....	4840
23-1029. UTMI_VCONTROL_EN_CLR_i.....	4840
23-1030. Register Call Summary for Register UTMI_VCONTROL_EN_CLR_i .....	4840
23-1031. UTMI_VCONTROL_STATUS_i.....	4841
23-1032. Register Call Summary for Register UTMI_VCONTROL_STATUS_i .....	4841
23-1033. UTMI_VCONTROL_LATCH_i.....	4841
23-1034. Register Call Summary for Register UTMI_VCONTROL_LATCH_i .....	4841
23-1035. UTMI_VSTATUS_i .....	4842
23-1036. Register Call Summary for Register UTMI_VSTATUS_i.....	4842
23-1037. UTMI_VSTATUS_SET_i.....	4842
23-1038. Register Call Summary for Register UTMI_VSTATUS_SET_i .....	4842
23-1039. UTMI_VSTATUS_CLR_i.....	4843
23-1040. Register Call Summary for Register UTMI_VSTATUS_CLR_i .....	4843
23-1041. USB_INT_LATCH_NOCLR_i .....	4843
23-1042. Register Call Summary for Register USB_INT_LATCH_NOCLR_i.....	4844
23-1043. VENDOR_INT_EN_i .....	4844
23-1044. Register Call Summary for Register VENDOR_INT_EN_i.....	4844
23-1045. VENDOR_INT_EN_SET_i.....	4844
23-1046. Register Call Summary for Register VENDOR_INT_EN_SET_i .....	4844
23-1047. VENDOR_INT_EN_CLR_i .....	4845
23-1048. Register Call Summary for Register VENDOR_INT_EN_CLR_i.....	4845
23-1049. VENDOR_INT_STATUS_i .....	4845

23-1050. Register Call Summary for Register VENDOR_INT_STATUS_i .....	4845
23-1051. VENDOR_INT_LATCH_i .....	4846
23-1052. Register Call Summary for Register VENDOR_INT_LATCH_i.....	4846
23-1053. HSUSBHOST Register Mapping Summary .....	4846
23-1054. UHH_REVISION .....	4847
23-1055. Register Call Summary for Register UHH_REVISION .....	4847
23-1056. UHH_HWINFO .....	4847
23-1057. Register Call Summary for Register UHH_HWINFO .....	4847
23-1058. UHH_SYSCONFIG .....	4847
23-1059. Register Call Summary for Register UHH_SYSCONFIG .....	4848
23-1060. UHH_SYSSTATUS.....	4848
23-1061. Register Call Summary for Register UHH_SYSSTATUS .....	4849
23-1062. UHH_HOSTCONFIG .....	4849
23-1063. Register Call Summary for Register UHH_HOSTCONFIG .....	4850
23-1064. UHH_DEBUG_CSR .....	4851
23-1065. Register Call Summary for Register UHH_DEBUG_CSR .....	4851
23-1066. UHH_SAR_CNTX_i .....	4852
23-1067. Register Call Summary for Register UHH_SAR_CNTX_i.....	4852
23-1068. OHCI Register Mapping Summary .....	4852
23-1069. HCREVISION .....	4853
23-1070. Register Call Summary for Register HCREVISION .....	4853
23-1071. HCCONTROL .....	4853
23-1072. Register Call Summary for Register HCCONTROL.....	4854
23-1073. HCCOMMANDSTATUS .....	4854
23-1074. Register Call Summary for Register HCCOMMANDSTATUS .....	4855
23-1075. HCINTERRUPTSTATUS .....	4855
23-1076. Register Call Summary for Register HCINTERRUPTSTATUS.....	4856
23-1077. HCINTERRUPTENABLE .....	4856
23-1078. Register Call Summary for Register HCINTERRUPTENABLE.....	4857
23-1079. HCINTERRUPTDISABLE .....	4858
23-1080. Register Call Summary for Register HCINTERRUPTDISABLE .....	4858
23-1081. HCHCCA.....	4859
23-1082. Register Call Summary for Register HCHCCA .....	4859
23-1083. HCPERIODCURRENTED .....	4859
23-1084. Register Call Summary for Register HCPERIODCURRENTED .....	4859
23-1085. HCCONTROLHEADED .....	4859
23-1086. Register Call Summary for Register HCCONTROLHEADED .....	4860
23-1087. HCCONTROLCURRENTED .....	4860
23-1088. Register Call Summary for Register HCCONTROLCURRENTED.....	4860
23-1089. HCBULKHEADED .....	4860
23-1090. Register Call Summary for Register HCBULKHEADED .....	4860
23-1091. HCBULKCURRENTED .....	4860
23-1092. Register Call Summary for Register HCBULKCURRENTED.....	4861
23-1093. HCDONEHEAD .....	4861
23-1094. Register Call Summary for Register HCDONEHEAD .....	4861
23-1095. HCFMINTERVAL .....	4861
23-1096. Register Call Summary for Register HCFMINTERVAL.....	4862
23-1097. HCFMREMAINING .....	4862
23-1098. Register Call Summary for Register HCFMREMAINING.....	4862

23-1099. HCFMNUMBER.....	4862
23-1100. Register Call Summary for Register HCFMNUMBER .....	4863
23-1101. HCPERIODICSTART .....	4863
23-1102. Register Call Summary for Register HCPERIODICSTART .....	4863
23-1103. HCLSTHRESHOLD .....	4863
23-1104. Register Call Summary for Register HCLSTHRESHOLD .....	4863
23-1105. HCRHDESCRIPTORA .....	4864
23-1106. Register Call Summary for Register HCRHDESCRIPTORA .....	4864
23-1107. HCRHDESCRIPTORB .....	4865
23-1108. Register Call Summary for Register HCRHDESCRIPTORB .....	4865
23-1109. HCRHSTATUS.....	4865
23-1110. Register Call Summary for Register HCRHSTATUS .....	4866
23-1111. HCRHPORTSTATUS_1 .....	4866
23-1112. Register Call Summary for Register HCRHPORTSTATUS_1 .....	4868
23-1113. HCRHPORTSTATUS_2 .....	4868
23-1114. Register Call Summary for Register HCRHPORTSTATUS_2 .....	4869
23-1115. EHCI Register Mapping Summary .....	4869
23-1116. HCCAPBASE .....	4870
23-1117. Register Call Summary for Register HCCAPBASE .....	4871
23-1118. HCSPARAMS .....	4871
23-1119. Register Call Summary for Register HCSPARAMS.....	4872
23-1120. HCCPARAMS.....	4872
23-1121. Register Call Summary for Register HCCPARAMS .....	4873
23-1122. USBCMD .....	4873
23-1123. Register Call Summary for Register USBCMD.....	4875
23-1124. USBSTS .....	4875
23-1125. Register Call Summary for Register USBSTS.....	4876
23-1126. USBINTR .....	4876
23-1127. Register Call Summary for Register USBINTR.....	4877
23-1128. FRINDEX .....	4877
23-1129. Register Call Summary for Register FRINDEX.....	4878
23-1130. CTRLDSSEGMENT .....	4878
23-1131. Register Call Summary for Register CTRLDSSEGMENT .....	4878
23-1132. PERIODICLISTBASE .....	4878
23-1133. Register Call Summary for Register PERIODICLISTBASE.....	4878
23-1134. ASYNCLISTADDR .....	4879
23-1135. Register Call Summary for Register ASYNCLISTADDR .....	4879
23-1136. CONFIGFLAG.....	4879
23-1137. Register Call Summary for Register CONFIGFLAG .....	4879
23-1138. PORTSC_i.....	4880
23-1139. Register Call Summary for Register PORTSC_i .....	4882
23-1140. INSNREG00.....	4882
23-1141. Register Call Summary for Register INSNREG00 .....	4882
23-1142. INSNREG01 .....	4883
23-1143. Register Call Summary for Register INSNREG01 .....	4883
23-1144. INSNREG02.....	4883
23-1145. Register Call Summary for Register INSNREG02 .....	4883
23-1146. INSNREG03.....	4883
23-1147. Register Call Summary for Register INSNREG03 .....	4884



23-1148. INSNREG04 .....	4884
23-1149. Register Call Summary for Register INSNREG04 .....	4884
23-1150. INSNREG05_UTMI .....	4885
23-1151. Register Call Summary for Register INSNREG05_UTMI .....	4885
23-1152. INSNREG05_ULPI .....	4885
23-1153. Register Call Summary for Register INSNREG05_ULPI .....	4886
23-1154. INSNREG06 .....	4886
23-1155. Register Call Summary for Register INSNREG06 .....	4887
23-1156. INSNREG07 .....	4887
23-1157. Register Call Summary for Register INSNREG07 .....	4887
23-1158. INSNREG08 .....	4887
23-1159. Register Call Summary for Register INSNREG08 .....	4887
23-1160. Input/Output Description .....	4892
23-1161. Integration Attributes .....	4894
23-1162. Clocks and Resets .....	4894
23-1163. Hardware Requests .....	4894
23-1164. HSUSBOTG Host Controller STANDBYMODE Settings .....	4896
23-1165. HSUSBOTG Host Controller SIDLEMODE Settings .....	4897
23-1166. Interconnect Data and MByteEn in Little- and Big-Endian Modes .....	4901
23-1167. Global Initialization of Surrounding Modules .....	4904
23-1168. High-Speed USB OTG Software Reset .....	4904
23-1169. Power Management Configuration .....	4904
23-1170. High-Speed USB Controller Interface Selection .....	4905
23-1171. Power Management Configuration - No Application Mode .....	4905
23-1172. Power Management Configuration - Host or Peripheral Mode .....	4905
23-1173. Power Management Configuration - Switching Between Host and Peripheral Mode .....	4905
23-1174. High-Speed USB OTG Controller Instance Summary .....	4906
23-1175. HSUSBOTG Registers Mapping Summary .....	4906
23-1176. OTG_REVISION .....	4906
23-1177. Register Call Summary for Register OTG_REVISION .....	4906
23-1178. OTG_SYSCONFIG .....	4907
23-1179. Register Call Summary for Register OTG_SYSCONFIG .....	4908
23-1180. OTG_SYSSTATUS .....	4908
23-1181. Register Call Summary for Register OTG_SYSSTATUS .....	4908
23-1182. OTG_INTERFSEL .....	4908
23-1183. Register Call Summary for Register OTG_INTERFSEL .....	4909
23-1184. OTG_SIMENABLE .....	4909
23-1185. Register Call Summary for Register OTG_SIMENABLE .....	4909
23-1186. OTG_FORCESTDBY .....	4909
23-1187. Register Call Summary for Register OTG_FORCESTDBY .....	4910
23-1188. OTG_BIGENDIAN .....	4910
23-1189. Register Call Summary for Register OTG_BIGENDIAN .....	4910
23-1190. USBPHY Registers Mapping Summary .....	4910
23-1191. USBPHY_TERMINATION_CONTROL .....	4911
23-1192. Register Call Summary for Register USBPHY_TERMINATION_CONTROL .....	4912
23-1193. USBPHY_RX_CALIB .....	4912
23-1194. Register Call Summary for Register USBPHY_RX_CALIB .....	4913
23-1195. USBPHY_DLLHS_2 .....	4913
23-1196. Register Call Summary for Register USBPHY_DLLHS_2 .....	4913

23-1197. USBPHY_RX_TEST_2 .....	4914
23-1198. Register Call Summary for Register USBPHY_RX_TEST_2 .....	4914
23-1199. USBPHY_TX_TEST_CHRG_DET .....	4914
23-1200. Register Call Summary for Register USBPHY_TX_TEST_CHRG_DET .....	4915
23-1201. USBPHY_CHRG_DET .....	4915
23-1202. Register Call Summary for Register USBPHY_CHRG_DET .....	4916
23-1203. USBPHY_PWR_CNTL .....	4916
23-1204. Register Call Summary for Register USBPHY_PWR_CNTL .....	4917
23-1205. USBPHY_UTMI_INTERFACE_CNTL_1 .....	4917
23-1206. Register Call Summary for Register USBPHY_UTMI_INTERFACE_CNTL_1 .....	4918
23-1207. USBPHY_UTMI_INTERFACE_CNTL_2 .....	4918
23-1208. Register Call Summary for Register USBPHY_UTMI_INTERFACE_CNTL_2 .....	4920
23-1209. USBPHY_BIST .....	4920
23-1210. Register Call Summary for Register USBPHY_BIST .....	4920
23-1211. USBPHY_BIST_CRC .....	4921
23-1212. Register Call Summary for Register USBPHY_BIST_CRC .....	4921
23-1213. USBPHY_CDR_BIST2 .....	4921
23-1214. Register Call Summary for Register USBPHY_CDR_BIST2 .....	4921
23-1215. USBPHY_GPIO .....	4922
23-1216. Register Call Summary for Register USBPHY_GPIO .....	4922
23-1217. USBPHY_DLLHS .....	4922
23-1218. Register Call Summary for Register USBPHY_DLLHS .....	4923
23-1219. USBPHY_USB2PHYCM_TRIM .....	4923
23-1220. Register Call Summary for Register USBPHY_USB2PHYCM_TRIM .....	4924
23-1221. USBPHY_USB2PHYCM_CONFIG .....	4924
23-1222. Register Call Summary for Register USBPHY_USB2PHYCM_CONFIG .....	4924
23-1223. USBPHY_USBOTG .....	4924
23-1224. Register Call Summary for Register USBPHY_USBOTG .....	4925
23-1225. USBPHY_AD_INTERFACE_REG1 .....	4925
23-1226. Register Call Summary for Register USBPHY_AD_INTERFACE_REG1 .....	4926
23-1227. USBPHY_AD_INTERFACE_REG2 .....	4926
23-1228. Register Call Summary for Register USBPHY_AD_INTERFACE_REG2 .....	4927
23-1229. USBPHY_AD_INTERFACE_REG3 .....	4927
23-1230. Register Call Summary for Register USBPHY_AD_INTERFACE_REG3 .....	4927
23-1231. USBPHY_ANA_CONFIG1 .....	4928
23-1232. Register Call Summary for Register USBPHY_ANA_CONFIG1 .....	4928
23-1233. USBPHY_ANA_CONFIG2 .....	4928
23-1234. Register Call Summary for Register USBPHY_ANA_CONFIG2 .....	4929
23-1235. Pin List for FS USB-Host (USB-Interchip Mode) .....	4932
23-1236. 4-Pin Mode Delivered on Pin (for External Transceiver Connection) .....	4932
23-1237. Integration Attributes .....	4933
23-1238. Clocks and Resets .....	4933
23-1239. Hardware Requests .....	4934
23-1240. FSUSB Host Controller Idle Mode Settings .....	4935
23-1241. FSUSB Host Controller STANDBY_MODE Settings .....	4936
23-1242. Global Initialization of Surrounding Modules .....	4937
23-1243. High-Speed Multiport USB Host Subsystem Global Initialization .....	4937
23-1244. USBFSHOST Instance Summary .....	4939
23-1245. USBFSHOST Registers Mapping Summary .....	4939

23-1246. HCREVISION .....	4940
23-1247. Register Call Summary for Register HCREVISION .....	4940
23-1248. HCCONTROL .....	4940
23-1249. Register Call Summary for Register HCCONTROL .....	4941
23-1250. HCCOMMANDSTATUS .....	4941
23-1251. Register Call Summary for Register HCCOMMANDSTATUS .....	4941
23-1252. HCINTERRUPTSTATUS .....	4942
23-1253. Register Call Summary for Register HCINTERRUPTSTATUS .....	4942
23-1254. HCINTERRUPTENABLE .....	4943
23-1255. Register Call Summary for Register HCINTERRUPTENABLE .....	4944
23-1256. HCINTERRUPTDISABLE .....	4944
23-1257. Register Call Summary for Register HCINTERRUPTDISABLE .....	4944
23-1258. HCHCCA .....	4945
23-1259. Register Call Summary for Register HCHCCA .....	4945
23-1260. HCPERIODCURRENTED .....	4945
23-1261. Register Call Summary for Register HCPERIODCURRENTED .....	4945
23-1262. HCCONTROLHEADED .....	4945
23-1263. Register Call Summary for Register HCCONTROLHEADED .....	4946
23-1264. HCCONTROLCURRENTED .....	4946
23-1265. Register Call Summary for Register HCCONTROLCURRENTED .....	4946
23-1266. HCBULKHEADED .....	4946
23-1267. Register Call Summary for Register HCBULKHEADED .....	4946
23-1268. HCBULKCURRENTED .....	4946
23-1269. Register Call Summary for Register HCBULKCURRENTED .....	4947
23-1270. HCDONEHEAD .....	4947
23-1271. Register Call Summary for Register HCDONEHEAD .....	4947
23-1272. HCFMINTERVAL .....	4947
23-1273. Register Call Summary for Register HCFMINTERVAL .....	4947
23-1274. HCFMREMAINING .....	4948
23-1275. Register Call Summary for Register HCFMREMAINING .....	4948
23-1276. HCFMNUMBER .....	4948
23-1277. Register Call Summary for Register HCFMNUMBER .....	4948
23-1278. HCPERIODICSTART .....	4948
23-1279. Register Call Summary for Register HCPERIODICSTART .....	4949
23-1280. HCLSTHRESHOLD .....	4949
23-1281. Register Call Summary for Register HCLSTHRESHOLD .....	4949
23-1282. HCRHDESCRIPTORA .....	4949
23-1283. Register Call Summary for Register HCRHDESCRIPTORA .....	4950
23-1284. HCRHDESCRIPTORB .....	4950
23-1285. Register Call Summary for Register HCRHDESCRIPTORB .....	4950
23-1286. HCRHSTATUS .....	4951
23-1287. Register Call Summary for Register HCRHSTATUS .....	4951
23-1288. HCRHPORTSTATUS .....	4952
23-1289. Register Call Summary for Register HCRHPORTSTATUS .....	4953
23-1290. HCOCPREV .....	4953
23-1291. Register Call Summary for Register HCOCPREV .....	4953
23-1292. HCOCPHWI .....	4954
23-1293. Register Call Summary for Register HCOCPHWI .....	4954
23-1294. HCOCPSYS .....	4954

23-1295. Register Call Summary for Register HCOCPYSYS .....	4955
24-1. MMC/SD/SDIOi I/O Description Without External Transceiver (where i= 1 through 5) .....	4965
24-2. MMC/SD/SDIOi I/O Description With External Transceiver (where i = 2) .....	4965
24-3. Relationship Between Configuration and Name of Response Type .....	4970
24-4. Integration Attributes .....	4973
24-5. Clocks and Resets .....	4973
24-6. Hardware Requests .....	4974
24-7. Smart-Idle Mode and Wake-Up Capabilities .....	4977
24-8. Local Power-Management Features .....	4978
24-9. Clock Activity Settings .....	4979
24-10. Events .....	4980
24-11. Descriptor Line Overview .....	4984
24-12. Available Actions of a Descriptor Line .....	4985
24-13. Additional Parameters of a Descriptor Line .....	4985
24-14. ADMA2 States Description .....	4986
24-15. ADMA FSM Symbol Definition .....	4987
24-16. Memory Size, BLEN, and Buffer Relationship .....	4993
24-17. MMC, SD, SDIO Responses in the MMCHS_RSPxx Registers .....	4993
24-18. CC and TC Values Upon Error Detected .....	4994
24-19. MMC/SD/SDIOi Controller Transfer Stop Command Summary .....	4999
24-20. MMC/SD/SDIO Hardware Status Features .....	5003
24-21. Global Initialization of Surrounding Modules .....	5004
24-22. MMC/SD/SDIO Controller Meta Initialization Steps .....	5004
24-23. Register Call Summary for Main Sequence – Software Reset Flow .....	5005
24-24. MMC/SD/SDIO Controller Wake-Up Configuration .....	5005
24-25. Register Call Summary for Main Sequence – Bus Configuration .....	5007
24-26. Register Call Summary for Main Sequence – Card Identification and Selection .....	5009
24-27. Subprocess Call Summary for Main Sequence – Card Identification and Selection .....	5010
24-28. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With interrupt ....	5010
24-29. Sub-process call summary for Main Sequence - MMC/SD/SDIO Controller Read/Write Transfer Flow in DMA mode with interrupt .....	5011
24-30. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling .....	5011
24-31. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling ..	5012
24-32. Register Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling .....	5014
24-33. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling ..	5014
24-34. Register Call Summary for Main Sequence – Read/Write in CE-ATA Mode .....	5014
24-35. Register Call Summary for Main Sequence – Suspend Flow .....	5017
24-36. Subprocess Call Summary for Main Sequence – Suspend Flow .....	5018
24-37. Register Call Summary for Main Sequence - Resume Flow .....	5018
24-38. Subprocess Call Summary for Main Sequence - Resume Flow .....	5019
24-39. Register Call Summary for Main Sequence – Command Transfer Flow With Polling .....	5020
24-40. Register Call Summary for Main Sequence – Command Transfer Flow With Interrupts .....	5022
24-41. Register Call Summary for Main Sequence – Clock Frequency Change Flow .....	5022
24-42. Register Call Summary for Main Sequence – Bus Width Configuration Flow .....	5023
24-43. Subprocess Call Summary for Main Sequence – Bus Width Configuration Flow .....	5023
24-44. Register Call Summary for Main Sequence – Power Switching Procedure .....	5025
24-45. Subprocess Call Summary for Main Sequence – Power Switching Procedure .....	5026
24-46. Register Call Summary for Main Sequence – Boot Using CMD0 .....	5027
24-47. Subprocess Call Summary for Main Sequence – Boot Using CMD0 .....	5027

24-48. Register Call Summary for Main Sequence – Boot Using CMD0.....	5028
24-49. Subprocess Call Summary for Main Sequence – Boot Using CMD0.....	5029
24-50. MMC/SD/SDIO Instance Summary .....	5030
24-51. MMC/SD/SDIO Registers Mapping Summary 1.....	5030
24-52. MMC/SD/SDIO Registers Mapping Summary 2.....	5031
24-53. MMCHS_HL_REV.....	5032
24-54. Register Call Summary for Register MMCHS_HL_REV .....	5032
24-55. MMCHS_HL_HWINFO .....	5032
24-56. Register Call Summary for Register MMCHS_HL_HWINFO .....	5033
24-57. Reset value for MMCHS_HL_HWINFO .....	5033
24-58. Reset value for MADMA_EN.....	5033
24-59. MMCHS_HL_SYSCONFIG .....	5034
24-60. Register Call Summary for Register MMCHS_HL_SYSCONFIG.....	5035
24-61. MMCHS_SYSCONFIG .....	5035
24-62. Register Call Summary for Register MMCHS_SYSCONFIG.....	5036
24-63. MMCHS_SYSSTATUS .....	5037
24-64. Register Call Summary for Register MMCHS_SYSSTATUS.....	5037
24-65. MMCHS_CSRE.....	5038
24-66. Register Call Summary for Register MMCHS_CSRE .....	5038
24-67. MMCHS_SYSTEST .....	5038
24-68. Register Call Summary for Register MMCHS_SYSTEST.....	5042
24-69. MMCHS_CON .....	5042
24-70. Register Call Summary for Register MMCHS_CON.....	5047
24-71. MMCHS_PWCNT .....	5047
24-72. Register Call Summary for Register MMCHS_PWCNT.....	5048
24-73. MMCHS_SDMASA.....	5048
24-74. MMCHS_BLK .....	5048
24-75. Register Call Summary for Register MMCHS_BLK.....	5049
24-76. MMCHS_ARG .....	5050
24-77. Register Call Summary for Register MMCHS_ARG.....	5050
24-78. MMCHS_CMD .....	5050
24-79. Register Call Summary for Register MMCHS_CMD.....	5054
24-80. MMCHS_RSP10.....	5054
24-81. Register Call Summary for Register MMCHS_RSP10 .....	5054
24-82. MMCHS_RSP32.....	5055
24-83. Register Call Summary for Register MMCHS_RSP32 .....	5055
24-84. MMCHS_RSP54.....	5055
24-85. Register Call Summary for Register MMCHS_RSP54 .....	5055
24-86. MMCHS_RSP76.....	5056
24-87. Register Call Summary for Register MMCHS_RSP76 .....	5056
24-88. MMCHS_DATA .....	5056
24-89. Register Call Summary for Register MMCHS_DATA.....	5057
24-90. MMCHS_PSTATE.....	5057
24-91. Register Call Summary for Register MMCHS_PSTATE .....	5060
24-92. MMCHS_HCTL .....	5061
24-93. Register Call Summary for Register MMCHS_HCTL.....	5064
24-94. MMCHS_SYSCTL.....	5064
24-95. Register Call Summary for Register MMCHS_SYSCTL .....	5067
24-96. MMCHS_STAT .....	5067

24-97. Register Call Summary for Register MMCHS_STAT .....	5072
24-98. MMCHS_IE.....	5073
24-99. Register Call Summary for Register MMCHS_IE .....	5075
24-100. MMCHS_ISE.....	5075
24-101. Register Call Summary for Register MMCHS_ISE .....	5077
24-102. MMCHS_AC12 .....	5078
24-103. Register Call Summary for Register MMCHS_AC12.....	5079
24-104. MMCHS_CAPA.....	5079
24-105. Register Call Summary for Register MMCHS_CAPA .....	5081
24-106. MMCHS_CUR_CAPA .....	5082
24-107. Register Call Summary for Register MMCHS_CUR_CAPA.....	5082
24-108. MMCHS_FE .....	5082
24-109. Register Call Summary for Register MMCHS_FE .....	5084
24-110. MMCHS_ADMAES .....	5084
24-111. Register Call Summary for Register MMCHS_ADMAES .....	5085
24-112. MMCHS_ADMASAL .....	5085
24-113. Register Call Summary for Register MMCHS_ADMASAL.....	5085
24-114. MMCHS_REV .....	5086
24-115. Register Call Summary for Register MMCHS_REV .....	5086
25-1. I/O .....	5092
25-2. Integration Attributes .....	5097
25-3. Clocks and Resets .....	5097
25-4. Hardware Requests .....	5097
25-5. Functional Clock Configuration .....	5104
25-6. Local Power-Management Features .....	5107
25-7. Clock Activity Settings .....	5108
25-8. Events.....	5108
25-9. Wake-Up Signals .....	5110
25-10. GPIO Channel Description .....	5110
25-11. Global Initialization of Surrounding Modules.....	5115
25-12. General-Purpose Interface Global Initialization.....	5115
25-13. General-Purpose Interface Read Input Register .....	5116
25-14. General-Purpose Interface Write Output Register .....	5116
25-15. General-Purpose Interface Set Bit Function .....	5116
25-16. General-Purpose Interface Clear Bit Function.....	5116
25-17. Instance Summary .....	5117
25-18. General-Purpose Interface GPIO1 to GPIO3 Registers Summary .....	5117
25-19. General-Purpose Interface GPIO4 to GPIO6 Registers Summary .....	5118
25-20. GPIO_REVISION.....	5119
25-21. Register Call Summary for Register GPIO_REVISION .....	5119
25-22. GPIO_SYSCONFIG.....	5119
25-23. Register Call Summary for Register GPIO_SYSCONFIG .....	5120
25-24. GPIO_IRQSTATUS_RAW_0 .....	5120
25-25. Register Call Summary for Register GPIO_IRQSTATUS_RAW_0.....	5120
25-26. GPIO_IRQSTATUS_RAW_1 .....	5121
25-27. Register Call Summary for Register GPIO_IRQSTATUS_RAW_1 .....	5121
25-28. GPIO_IRQSTATUS_0 .....	5121
25-29. Register Call Summary for Register GPIO_IRQSTATUS_0.....	5121
25-30. GPIO_IRQSTATUS_1 .....	5122

25-31. Register Call Summary for Register GPIO_IRQSTATUS_1 .....	5122
25-32. GPIO_IRQSTATUS_SET_0.....	5122
25-33. Register Call Summary for Register GPIO_IRQSTATUS_SET_0 .....	5122
25-34. GPIO_IRQSTATUS_SET_1.....	5123
25-35. Register Call Summary for Register GPIO_IRQSTATUS_SET_1 .....	5123
25-36. GPIO_IRQSTATUS_CLR_0 .....	5124
25-37. Register Call Summary for Register GPIO_IRQSTATUS_CLR_0 .....	5124
25-38. GPIO_IRQSTATUS_CLR_1 .....	5124
25-39. Register Call Summary for Register GPIO_IRQSTATUS_CLR_1 .....	5125
25-40. GPIO_IRQWAKEN_0 .....	5125
25-41. Register Call Summary for Register GPIO_IRQWAKEN_0.....	5125
25-42. GPIO_IRQWAKEN_1 .....	5126
25-43. Register Call Summary for Register GPIO_IRQWAKEN_1.....	5126
25-44. GPIO_SYSSTATUS.....	5126
25-45. Register Call Summary for Register GPIO_SYSSTATUS .....	5127
25-46. GPIO_IRQSTATUS1.....	5127
25-47. Register Call Summary for Register GPIO_IRQSTATUS1 .....	5127
25-48. GPIO_IRQENABLE1.....	5128
25-49. Register Call Summary for Register GPIO_IRQENABLE1 .....	5128
25-50. GPIO_WAKEUPENABLE .....	5128
25-51. Register Call Summary for Register GPIO_WAKEUPENABLE .....	5128
25-52. GPIO_IRQSTATUS2.....	5129
25-53. Register Call Summary for Register GPIO_IRQSTATUS2 .....	5129
25-54. GPIO_IRQENABLE2.....	5129
25-55. Register Call Summary for Register GPIO_IRQENABLE2 .....	5129
25-56. GPIO_CTRL.....	5130
25-57. Register Call Summary for Register GPIO_CTRL .....	5130
25-58. GPIO_OE.....	5131
25-59. Register Call Summary for Register GPIO_OE .....	5131
25-60. GPIO_DATAIN.....	5131
25-61. Register Call Summary for Register GPIO_DATAIN .....	5131
25-62. GPIO_DATAOUT .....	5132
25-63. Register Call Summary for Register GPIO_DATAOUT .....	5132
25-64. GPIO_LEVELDETECT0 .....	5132
25-65. Register Call Summary for Register GPIO_LEVELDETECT0.....	5133
25-66. GPIO_LEVELDETECT1 .....	5133
25-67. Register Call Summary for Register GPIO_LEVELDETECT1.....	5133
25-68. GPIO_RISINGDETECT.....	5133
25-69. Register Call Summary for Register GPIO_RISINGDETECT .....	5134
25-70. GPIO_FALLINGDETECT.....	5134
25-71. Register Call Summary for Register GPIO_FALLINGDETECT .....	5134
25-72. GPIO_DEBOUNCENABLE.....	5134
25-73. Register Call Summary for Register GPIO_DEBOUNCENABLE .....	5135
25-74. GPIO_DEBOUNCINGTIME .....	5135
25-75. Register Call Summary for Register GPIO_DEBOUNCINGTIME.....	5135
25-76. GPIO_CLEARIRQENABLE1.....	5135
25-77. Register Call Summary for Register GPIO_CLEARIRQENABLE1 .....	5136
25-78. GPIO_SETIRQENABLE1.....	5136
25-79. Register Call Summary for Register GPIO_SETIRQENABLE1 .....	5136

25-80. GPIO_CLEARIRQENABLE2.....	5136
25-81. Register Call Summary for Register GPIO_CLEARIRQENABLE2 .....	5137
25-82. GPIO_SETIRQENABLE2.....	5137
25-83. Register Call Summary for Register GPIO_SETIRQENABLE2 .....	5137
25-84. GPIO_CLEARWKUPENA .....	5137
25-85. Register Call Summary for Register GPIO_CLEARWKUPENA.....	5137
25-86. GPIO_SETWKUENA.....	5138
25-87. Register Call Summary for Register GPIO_SETWKUENA .....	5138
25-88. GPIO_CLEARDATAOUT .....	5138
25-89. Register Call Summary for Register GPIO_CLEARDATAOUT .....	5138
25-90. GPIO_SETDATAOUT.....	5139
25-91. Register Call Summary for Register GPIO_SETDATAOUT .....	5139
26-1. I/O External Keyboard Signals.....	5145
26-2. Integration Attributes .....	5147
26-3. Clocks and Resets .....	5147
26-4. Hardware Requests .....	5148
26-5. Local Power-Management Features .....	5150
26-6. Clock Activity Settings .....	5150
26-7. Events.....	5151
26-8. Keyboard Controller Functional Modes.....	5151
26-9. Keyboard Controller Timer Values .....	5152
26-10. Timer Prescale Values.....	5153
26-11. State-Machine Values.....	5153
26-12. Global Initialization of Surrounding Modules.....	5159
26-13. Keyboard Controller Global Initialization .....	5159
26-14. Keyboard Controller Hardware Mode.....	5160
26-15. Keyboard Controller Software Mode .....	5161
26-16. Keyboard Controller Event Servicing .....	5161
26-17. Keyboard Controller Instance Summary.....	5163
26-18. Keyboard Controller Register Mapping Summary.....	5163
26-19. KBD_REVISION .....	5164
26-20. Register Call Summary for Register KBD_REVISION.....	5164
26-21. KBD_SYSCONFIG .....	5164
26-22. Register Call Summary for Register KBD_SYSCONFIG.....	5165
26-23. KBD_SYSSTATUS .....	5165
26-24. Register Call Summary for Register KBD_SYSSTATUS .....	5166
26-25. KBD_IRQSTATUS .....	5166
26-26. Register Call Summary for Register KBD_IRQSTATUS .....	5166
26-27. KBD_IRQENABLE .....	5167
26-28. Register Call Summary for Register KBD_IRQENABLE .....	5167
26-29. KBD_WAKEUPENABLE.....	5168
26-30. Register Call Summary for Register KBD_WAKEUPENABLE .....	5168
26-31. KBD_PENDING.....	5168
26-32. Register Call Summary for Register KBD_PENDING .....	5169
26-33. KBD_CTRL .....	5169
26-34. Register Call Summary for Register KBD_CTRL .....	5170
26-35. KBD_DEBOUNCINGTIME .....	5170
26-36. Register Call Summary for Register KBD_DEBOUNCINGTIME.....	5171
26-37. KBD_KEYLONGTIME .....	5171



26-38. Register Call Summary for Register KBD_KEYLONGTIME .....	5171
26-39. KBD_TIMEOUT .....	5171
26-40. Register Call Summary for Register KBD_TIMEOUT .....	5172
26-41. KBD_STATEMACHINE .....	5172
26-42. Register Call Summary for Register KBD_STATEMACHINE .....	5172
26-43. KBD_ROWINPUTS .....	5172
26-44. Register Call Summary for Register KBD_ROWINPUTS .....	5173
26-45. KBD_COLUMNOUTPUTS .....	5173
26-46. Register Call Summary for Register KBD_COLUMNOUTPUTS .....	5173
26-47. KBD_FULLCODE31_0.....	5173
26-48. Register Call Summary for Register KBD_FULLCODE31_0 .....	5173
26-49. KBD_FULLCODE63_32 .....	5174
26-50. Register Call Summary for Register KBD_FULLCODE63_32.....	5174
26-51. KBD_FULLCODE17_0.....	5174
26-52. Register Call Summary for Register KBD_FULLCODE17_0 .....	5174
26-53. KBD_FULLCODE35_18 .....	5175
26-54. Register Call Summary for Register KBD_FULLCODE35_18.....	5175
26-55. KBD_FULLCODE53_36 .....	5175
26-56. Register Call Summary for Register KBD_FULLCODE53_36.....	5176
26-57. KBD_FULLCODE71_54 .....	5176
26-58. Register Call Summary for Register KBD_FULLCODE71_54.....	5176
26-59. KBD_FULLCODE80_72 .....	5176
26-60. Register Call Summary for Register KBD_FULLCODE80_72.....	5176
27-1. OMAP Power Pins .....	5184
27-2. Mapping for Input Sources .....	5189
27-3. TWL6030 Clock Requirements .....	5190
27-4. Memory Preferred Booting .....	5194
27-5. Peripheral Preferred Booting.....	5195
27-6. Pin Multiplexing According to Boot Peripheral.....	5195
27-7. ROM Exception Vectors .....	5200
27-8. Dead Loops .....	5201
27-9. RAM Exception Vectors .....	5202
27-10. Tracing Data .....	5202
27-11. ROM Code Default Clock Settings.....	5207
27-12. Public Use of SAR RAM.....	5209
27-13. Software Booting Configuration Structure.....	5210
27-14. ASIC ID Structure .....	5213
27-15. Items .....	5213
27-16. ID Subblock .....	5213
27-17. Checksum Subblock .....	5213
27-18. Booting Messages.....	5214
27-19. Device Descriptor.....	5220
27-20. Device-Qualifier Descriptor.....	5220
27-21. Configuration Descriptor.....	5220
27-22. Other Speed Configuration Descriptor.....	5221
27-23. Interface Descriptor .....	5221
27-24. BULK IN Endpoint Descriptor.....	5221
27-25. BULK OUT Endpoint Descriptor.....	5222
27-26. Language ID String Descriptor .....	5222

27-27. Manufacturer ID String Descriptor.....	5222
27-28. Product ID String Descriptor .....	5222
27-29. Configuration String Descriptor .....	5222
27-30. Interface String Descriptor.....	5222
27-31. Customized Descriptor Parameters .....	5223
27-32. Standard Device Requests Supported .....	5224
27-33. Blocks and Sectors Searched on Non-XIP Memories .....	5229
27-34. XIP Timing Parameters .....	5230
27-35. NAND Timing Parameters.....	5231
27-36. ONFI Parameters Page Description .....	5232
27-37. Supported NAND Devices.....	5232
27-38. Fourth NAND ID Data Byte .....	5233
27-39. Master Boot Record Structure .....	5248
27-40. Partition Table Entry .....	5248
27-41. FAT Directory Entry .....	5251
27-42. FAT Entry Description .....	5252
27-43. Summary of Mode Register Commands .....	5254
27-44. CH TOC Item .....	5256
27-45. CHSETTINGS Item.....	5256
27-46. CHRAM Item .....	5257
27-47. CHFLASH Item .....	5258
27-48. CHMMCS D Item .....	5259
27-49. GP Header Format .....	5259
27-50. Booting Parameter Structure.....	5260
27-51. Tracing Vector 1 .....	5261
27-52. Tracing Vector 2 .....	5261
27-53. Tracing Vector 3 .....	5262
27-54. L2 Cache Write Debug Register.....	5264
27-55. L2 Cache Clean and Invalidate Range of Physical Address .....	5264
27-56. Enable/Disable L2 Cache .....	5264
27-57. Write Auxiliary Control Register .....	5265
27-58. Write Tag and Data RAM Latency Control Register .....	5265
27-59. Read AuxCoreBoot 0 and AuxCoreBoot 1 Register .....	5265
27-60. Modify AuxCoreBoot 0 .....	5265
27-61. Write AuxCoreBoot 1 .....	5266
27-62. Read Control and Status Register .....	5266
27-63. Clear Control and Status Register .....	5266
27-64. Set Power Status Register .....	5266
27-65. Select the TLB Entry for Read.....	5267
27-66. Select the TLB Entry for Write .....	5267
27-67. Read the TLB VA Entry.....	5267
27-68. Write the TLB VA Entry .....	5267
27-69. Read the TLB PA Entry.....	5268
27-70. Write the TLB PA Entry .....	5268
27-71. Read the TLB Attributes Entry.....	5268
27-72. Write the TLB Attributes Entry .....	5268
28-1. IEEE1149.1 Signals .....	5271
28-2. OMAP4430 JTAG ID Code.....	5271
28-3. IEEE1149.7 Signals .....	5272

28-4. IEEE1149.7 Features Subset .....	5272
28-5. Trace Port Signals.....	5273
28-6. NIDnT Port Reusing MMC Interface Pins .....	5274
28-7. ICEPick Boot Modes Upon POR .....	5275
28-8. ICEPick Secondary Debug TAP Mapping.....	5276
28-9. ICEPick Debug Core Mapping.....	5277
28-10. Hardware Accelerator Debug Capability Options .....	5279
28-11. XTRIGGER.....	5280
28-12. Debug Subsystem Suspend Input Lines .....	5281
28-13. Debug Subsystem Suspend Output Lines .....	5281
28-14. PMU Events.....	5284
28-15. PL310 L2 Cache Events.....	5285
28-16. Cortex-M3 SCTM Counters Configuration .....	5285
28-17. SCTM Events For Cortex-M3 MPU Subsystem .....	5285
28-18. DSP SCTM Counters Configuration .....	5286
28-19. SCTM Events For DSP Subsystem .....	5286
28-20. L3 Interconnect Functional Probe Mapping.....	5291
28-21. Performance Monitoring Events Detection.....	5292
28-22. Performance Filtering Options .....	5292
28-23. Aggregation Modes.....	5292
28-24. STATCOLL_SDRAM Configuration.....	5293
28-25. STATCOLL_LAT0 Configuration .....	5294
28-26. STATCOLL_LAT1 Configuration .....	5295
28-27. Trace Port Configuration .....	5297
28-28. Concurrent Debug and Trace.....	5298
28-29. Debug Modules Memory Mapping .....	5299

PRELIMINARY



## Read This First

---

---

---

### Survey

**Help us meet your expectations:**

We are always looking at ways to develop our service and improve our quality to fit your needs. Please take a few minutes to complete our short questionnaire by:

- Providing general suggestions  
or
- Rating a document and describing its critical points

<http://www.ti.com/csdocsurvey>

(password: 123survey)

Thank you.

If You Need Assistance. . .

www.ti.com

---

## If You Need Assistance. . .

---

<b>If you want to . . .</b>	<b>Do this . . .</b>
Request more information about Texas Instruments Digital Signal Processing (DSP) products	Call the CRC <sup>(1)</sup> hotline: <b>(800) 336-5236</b> Or write to: Texas Instruments Incorporated Market Communications Manager, MS 736 P.O. Box 1443 Houston, Texas 77251-1443
Order Texas Instruments documentation	Call the CRC <sup>(1)</sup> hotline: <b>(800) 336-5236</b>

---

<sup>(1)</sup> Texas Instruments Customer Response Center

## About This Manual

### FCC Warning

This equipment is intended for use in a laboratory test environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

### Information About Cautions and Warnings

This book may contain cautions and warnings.

#### **CAUTION**

This is an example of a caution statement.

A caution statement describes a situation that could potentially damage your software or equipment.

#### **WARNING**

**This is an example of a warning statement.**

**A warning statement describes a situation that could potentially cause harm to you.**

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

## Register, Field, and Bit Calls

The naming convention applied for a call consists of:

- For a register call: *<Module name>.<Register name>*; for example: UART.UASR
- For a bit field call:
  - *<Module name>.<Register name>[End:Start] <Field name> field*; for example, UART.UASR[4:0] SPEED bit field
  - *<Field name> field <Module name>.<Register name>[End:Start]*; for example, SPEED bit field UART.UASR[4:0]
- For a bit call:
  - *<Module name>.<Register name>[pos] <Bit name> bit*, for example, UART.UASR[5] BIT\_BY\_CHAR bit
  - *<Bit name> bit <Module name>.<Register name>[pos]*; for example, BIT\_BY\_CHAR bit UART.UASR[5]

To help the reader navigate the document, each register call is hyperlinked to its register description in the register manual section. After each register description, a table summarizes all hyperlinked register calls.

To navigate in the PDF documents, see [Acrobat Reader Tips](#).















## Coding Rules

The programming models or code listings follow the rules:

Type	Definition	Example
File	Starts with the module name	PRCM_test1.c MCBSP1_init.h
Variable	Global variables are prefixed by "g_" Pointers are prefixed by "p" Global pointers are prefixed by "g_p"	g_SDMA_LogicalChan pAddrCounter g_pSDMA_LogicalChan
Function	Starts with the module name	PRCM_SetupClocks() ArmIntC_MaskInterrupts()
Typedef	Ends with "_t"	PRCM_Struct_t
Definition	Starts with the module name and is followed by the register name	#define SMS_ERR_TYPE *((volatile Uint32*)0x680080F4) #define MCBSP2_RCR1_REG *((volatile Uint32*)0x4807401C)
Enumeration	Starts with the module name	Typedef enum DMA_Mode_Label { INPUT_MODE OUTPUT_MODE } DMA_Mode_t;

## Flow Chart Rules

Flow charts follow the following rules:

Shape	Name	Definition
	Process	Any computational steps or processing function of a program; defined operation(s) causing change in value, form, or location of information
	Decision	A decision or switching type operation that determines which of a number of alternate paths is followed
	Predefined process or sub-process	One or more named operations or program steps specified in a subroutine or another set of flow charts
	Data or I/O	General I/O function; information available for processing (input) or recording of processed information (output)
	Terminator	Terminal point in a flow chart: start, stop, halt, delay, or interrupt; may show exit from a closed subroutine
 <small>The height of the text box and its associated line increases or decreases as you add text. To change the width of the comment, drag the side handle.</small>	Annotation	Additional descriptive clarification, comment
	On page connector (reference)	Exit to, or entry from, another part of chart in the same page
	Off page connector (reference)	The flow continues on a different page.
	Summing Junction	Logical AND
	Or	Logical OR
	Parallel mode (ISO)	Beginning or end of two or more simultaneous operations
	Flow Line	Lines indicate the sequence of steps and the direction of flow.

## Acrobat Reader Tips

Acrobat includes two methods to search for words in a PDF:

- The Find toolbar provides a basic set of options to locate a word in the current PDF.
- The Search window lists words or partial words that match your text in the current PDF.

These guidelines apply to Acrobat Reader 5.x, 6.0, and 7.0.

For more information on Acrobat Reader search features, see the Adobe Reader Help.

### To search for words in a document using the Find dialog box:

1. Open the document.
2. To display the Find toolbar, right-click in the toolbar area and select Find.
3. In the Find box, type the word, words, or partial words for which you want to search.
4. From the Find Options menu, select options as desired.
5. To view each search result, click the Find toolbar, the Find Previous button, or the Find Next button to go backward or forward through the document.

### To search for words in a document using the Search PDF window:

1. Open the document.
2. Click the Search button on the File toolbar or right-click on your document and select Search.
3. Type the word, words, or part of a word for which you want to search.
4. Click Search.
5. The results appear in page order and, if applicable, show a few words of context. Each result displays an icon to identify the type of occurrence. All other searchable areas display the Search Result icon.
6. To display the page that contains a search result, click an item in the Results list. The occurrence is highlighted.
7. To navigate to the next result, choose Edit > Search Results > Next Result (or Ctrl+G).
8. To navigate to the previous result, choose Edit > Search Results > Previous Result (or Shift+Ctrl+G).

### Navigate through your previous view

To retrace your path within an Adobe PDF document:

- For the previous view: Choose View > Go To > Previous View or Alt+Left Arrow.
- For the next view: Choose View > Go To > Next View or Alt+Right Arrow. The Next View command is available only if you have chosen Previous View.

If you view the PDF document in a browser, use options on the Navigation toolbar to move between views.

- Right-click the toolbar area, and then choose *Navigation*.
- Click the Go To Previous View button or the Go To Next View button.

---

**NOTE:** This navigation tip is useful to return to your previous view after clicking on a register call hyperlink.

---

**OMAP4430 MIPI® Disclaimer**

The material contained herein is not a license, either expressly or impliedly, to any IPR owned or controlled by any of the authors or developers of this material or MIPI. The material contained herein is provided on an "AS IS" basis and to the maximum extent permitted by applicable law, this material is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material and MIPI hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence.

ALSO, THERE IS NO WARRANTY OF CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THIS MATERIAL OR THE CONTENTS OF THIS DOCUMENT. IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR THE CONTENTS OF THIS DOCUMENT OR MIPI BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT, SPECIFICATION OR DOCUMENT RELATING TO THIS MATERIAL, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Without limiting the generality of this Disclaimer stated above, the user of the contents of this Document is further notified that MIPI: (a) does not evaluate, test or verify the accuracy, soundness or credibility of the contents of this Document; (b) does not monitor or enforce compliance with the contents of this Document; and (c) does not certify, test, or in any manner investigate products or services or any claims of compliance with the contents of this Document. The use or implementation of the contents of this Document may involve or require the use of intellectual property rights ("IPR") including (but not limited to) patents, patent applications, or copyrights owned by one or more parties, whether or not Members of MIPI. MIPI does not make any search or investigation for IPR, nor does MIPI require or request the disclosure of any IPR or claims of IPR as respects the contents of this Document or otherwise. Questions pertaining to this document, or the terms or conditions of its provision, should be addressed to:

MIPI Alliance, Inc. c/o IEEE-ISTO 445 Hoes Lane Piscataway, NJ 08854 Attn: Board Secretary

## Trademarks

OMAP, TMS320DMC64x, C64x, ICECrusher, IcePick and SmartReflex are trademarks of Texas Instruments Incorporated. PicoDLP is a registered trademark of Texas Instruments Incorporated.

ARM, Cortex, Jazelle, and Thumb are registered trademarks of ARM Limited. ETM, ETB, ARM9, CoreSight, ISA, Java and Neon are trademarks of ARM Limited.

Bluetooth is a registered trademark of Bluetooth SIG, Inc. and is licensed to Texas Instruments.

HDQ is a trademark of Benchmark.

1-Wire is a registered trademark of Dallas Semiconductor.

Windows, WinMobile and Direct3D are trademarks of Microsoft Corporation in the United States and other countries.

USSE and POWERVR are trademarks or registered trademarks of Imagination Technologies Ltd.

Mentor Graphics is a registered trademark of Mentor Graphics Corporation or its affiliated companies in the United States and other countries.

OpenGL is a trademark of Silicon Graphics, Inc.

OpenVG and OpenMAX are trademarks of Khronous Group, Inc.

Arteris is a registered trademark of Arteris, Inc.

RealVideo is a registered trademark of RealNetworks, Inc.

SD is a registered trademark of Toshiba Corporation.

eSD is a trademark of SD Association.

MMC and eMMC are trademarks of MultiMediaCard Association.

SonicsMX, Sonics3220 are trademarks or registered trademarks of Sonics, Inc.

JTAG is a registered trademark of JTAG Technologies, Inc.

Linux is a registered trademark of Linus Torvalds.

On2 is a registered trademark of On2 Technologies.

Symbian and all Symbian based trademarks and logos are trademarks of Symbian Software Limited.

Synopsys is a registered trademarks of Synopsys, Inc.

I2S is a trademark of Phillips Electronics.

I<sup>2</sup>C is a trademark of Phillips Semiconductor Corp.

SLIMbus and MIPI are registered trademarks of the Mobile Industry Processor Interface (MIPI) Alliance.

Flex-OneNAND and OneNAND are trademarks of SAMSUNG Electronics, Corporation.

Palm OS is a registered trademark of Palm Inc.

All other trademarks are the property of their respective owners.

## History

The following table summarizes the OMAP4430 Public TRM versions.

Version	Literature Number	Date	Notes
I	SWPU231	July 2010	See <sup>(1)</sup>
J	SWPU231	August 2010	See <sup>(2)</sup>
K	SWPU231	September 2010	See <sup>(3)</sup>

<sup>(1)</sup> *OMAP4430 Multimedia Device Silicon Revision 2.0 Technical Reference Manual*, public version I (SWPU231I) - Not released to the public domain.

<sup>(2)</sup> *OMAP4430 Multimedia Device Silicon Revision 2.0 Technical Reference Manual*, public version J (SWPU231J). - Initial release to the public domain.

<sup>(3)</sup> *OMAP4430 Multimedia Device Silicon Revision 2.x Technical Reference Manual*, public version K (SWPU231K) - Chapters impacted by changes between version J and version K:

- Chapter 1: Introduction
- Chapter 3: Power, Reset, and Clock Management
- Chapter 4: Dual Cortex-A9 MPU Subsystem
- Chapter 6: IVA-HD Subsystem
- Chapter 10: Display Subsystem
- Chapter 18: Control Module
- Chapter 23: Serial Communication Interfaces
- Chapter 26: Keyboard Controller
- Chapter 27: Initialization
- Chapter 28: On-Chip Debug Support

## Introduction

---

---

---

This chapter introduces the features, subsystems, and architecture of the OMAP4430 high-performance multimedia device.

Topic	Page
1.1 Overview .....	256
1.2 Environment .....	257
1.3 Description .....	258
1.4 Package-On-Package Concept .....	268
1.5 Device Identification .....	269

## 1.1 Overview

The OMAP4430 high-performance multimedia application device is based on enhanced OMAP™ architecture and uses 45-nm technology.

- The architecture is designed to provide best-in-class video, image, and graphics processing for 2.5/3G wireless terminals, high-performance personal digital assistants (PDAs). For that purpose, the device supports the following functions:
  - Streaming video up to full high definition (HD) (1920 × 1080 p, 30 fps)
  - 2-dimensional (2D)/3-dimensional (3D) mobile gaming
  - Video conferencing
  - High-resolution still image (up to 16 Mp)
- The device supports high-level operating systems (OSs) such as:
  - Windows™ CE, WinMobile™
  - Symbian OS™
  - Linux®
  - Palm OS™
- The device is composed of the following subsystems:
  - Cortex™-A9 microprocessor unit (MPU) subsystem, including two ARM® Cortex-A9 cores
  - Digital signal processor (DSP) subsystem
  - Image and video accelerator high-definition (IVA-HD) subsystem
  - Cortex™-M3 MPU subsystem, including two ARM Cortex-M3 microprocessors
  - Display subsystem
  - Audio back-end (ABE) subsystem
  - Imaging subsystem (ISS), consisting of image signal processor (ISP) and still image coprocessor (SIMCOP) block
  - 2D/3D graphic accelerator (SGX) subsystem
  - Emulation (EMU) subsystem
- The device includes state-of-art power-management techniques required for high-performance mobile products.
- Comprehensive power management is integrated into the device.
- The device also integrates:
  - On-chip memory
  - External memory interfaces
  - Memory management
  - Level 3 (L3) and level 4 (L4) interconnects
  - System and connecting peripherals

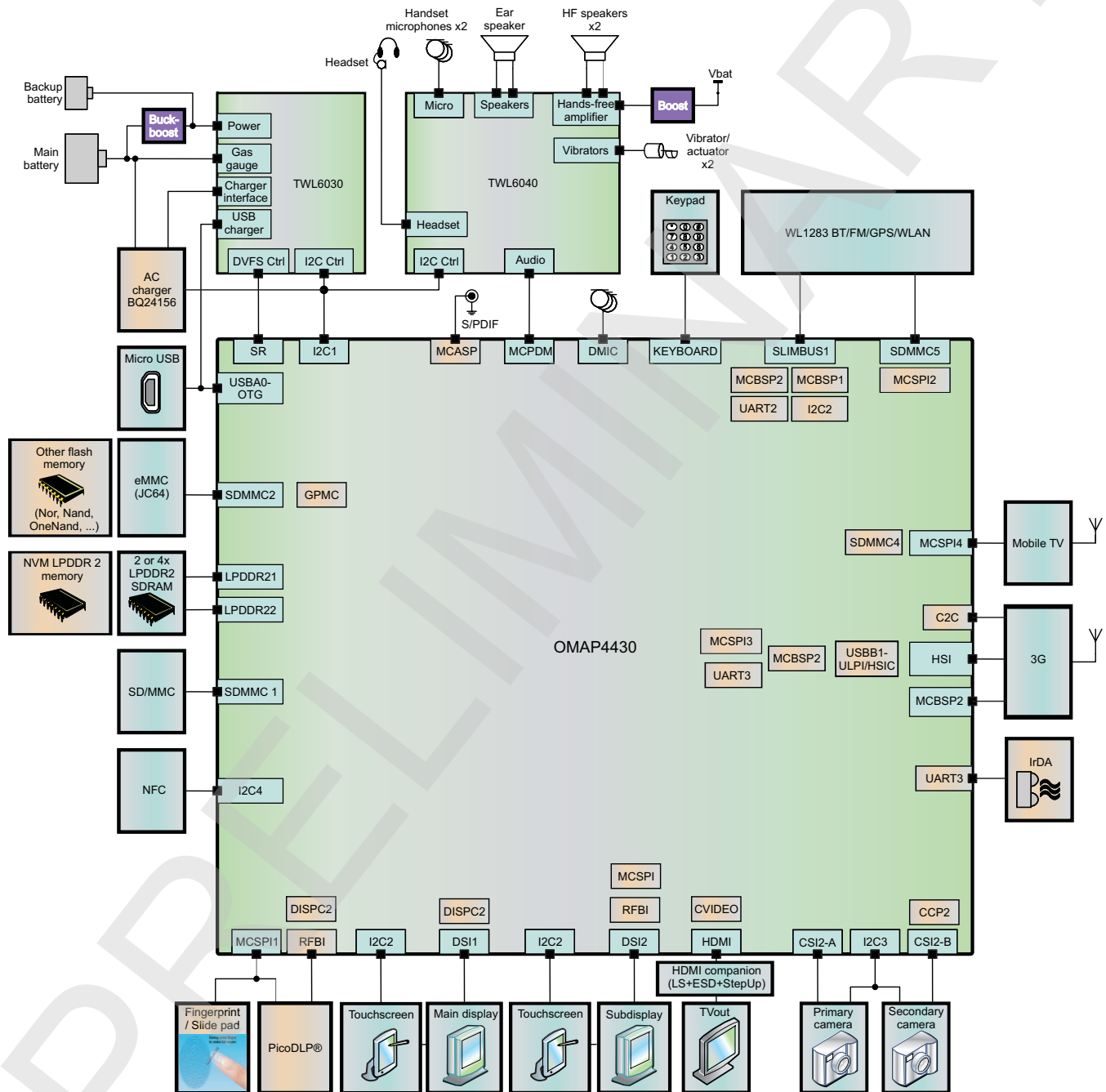


## 1.2 Environment

This section provides an overview of the OMAP environment. The device is associated with power and audio integrated circuits (ICs). TI provides a global solution with the TWL6030/TWL6040 devices.

Figure 1-1 is an overview of a nonexhaustive environment for the high-tier OMAP4430 device.

Figure 1-1. OMAP4430 High-Tier Environment



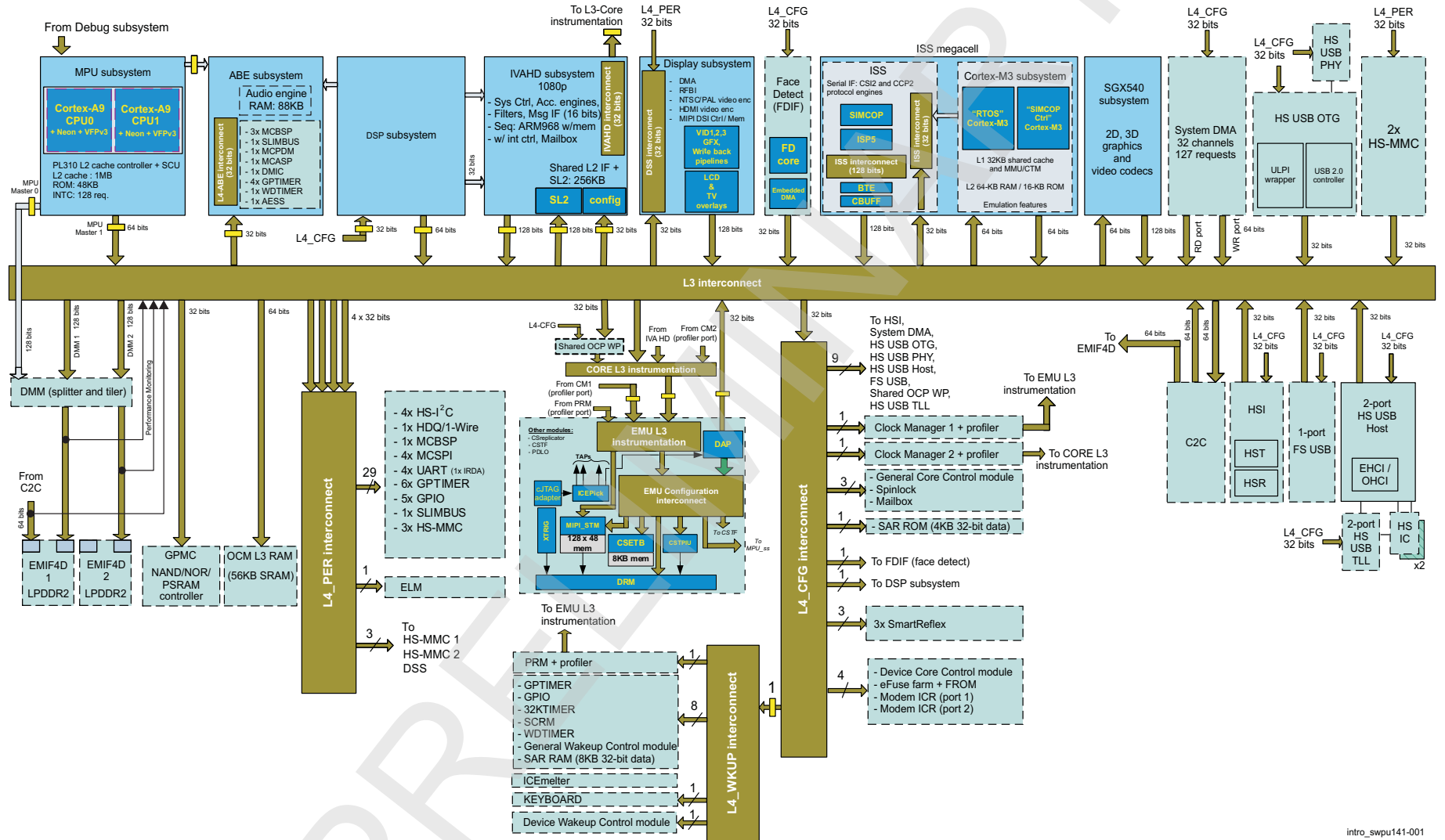
intro-002

### 1.3 Description

The device is offered in a 547-ball, 12 × 12 × 0.73-mm, 0.4-mm (bottom) ball pitch package. Some balls are available at the top of the device to allow memory stacking.

[Figure 1-2](#) is the block diagram of the OMAP4430 device.

Figure 1-2. OMAP4430 Block Diagram



intro\_swpu141-001

### 1.3.1 Cortex-A9 MPU Subsystem Description

The Cortex-A9 MPU subsystem integrates the following submodules:

- ARM Cortex-A9 MPCore
  - Two ARM Cortex-A9 central processing units (CPUs)
  - ARM Version 7 ISA™: Standard ARM instruction set plus Thumb®-2, Jazelle® RCT and Jazelle DBX Java™ accelerators
  - Neon™ SIMD coprocessor and VFPv3 per CPU
  - Interrupt controller (Cortex-A9 MPU INTC) with up to 128 interrupt requests
  - One general-purpose timer and one watchdog timer per CPU
  - Debug and trace features
  - 32-KB instruction and 32-KB data level 1 (L1) caches per CPU
- Shared 1-MB level 2 (L2) cache
- 48 KB bootable ROM
- Local power, reset, and clock management (PRCM) module
- Emulation features
- Digital phase-locked loop (DPLL)

### 1.3.2 IVA-HD Subsystem Description

The IVA-HD subsystem is a set of video encoder/decoder hardware accelerators. It supports up to 1080p x 30 fps, slow-motion camcorder, triple play (HD and SD capture and JPEG capture), real-time transcoding of up to 720p, and video conferencing up to 720p.

The IVA-HD subsystem is composed of:

- Improved motion estimation acceleration engine (iME3), which is used in encoding processing
- Improved loop filter acceleration engine (iLF3), which performs deblocking filtering
- Improved sequencer (ICONT1) based on the ARM968E-S™ microcontroller. It includes memory and INTC and is used as a primary sequencer.
- Intraprediction estimation engine (iPE3). It is used in encoding processing.
- Calculation engine (CALC3), which performs transform and quantization calculations
- Motion compensation engine (MC3), which creates an interprediction macroblock with given motion vectors and modes from the reference data
- Entropy coder/decoder (ECD3), which uses Huffman and arithmetic codes during the process of encoding and decoding the stream
- Video DMA processor (ICONT2), which is also based on the ARM968E-S microcontroller and can be used as secondary sequencer
- Video DMA engine (vDMA), which is a DMA engine for data transmission between external memories and shared L2 memory
- Synchronization box (SyncBox) embedded in each hardware accelerator and in both ICONTs
- Mailbox for communication between IVA-HD and external to it processors (DSP, Cortex-A9, and Cortex-M3)
- Shared L2 interface and memory
- Video local interconnect for connection between the submodules of the IVA-HD, and between the IVA-HD and DSP subsystems
- IVA-HD system control module (SYSCTRL), which controls the clocks in the subsystem and PRCM handshaking

The IVA-HD subsystem can process three data formats for internal data: picture or slice, macroblock header, and residual data.

The IVA-HD supports the following formats:

- MPEG-1/-2/-4 such as MPEG-2 MP, ML, and MPEG-4 as SP/ASP
- Divx 5.02 and above
- Sorenson Spark (decode)

- H.263 P0 (encode and decode) and P3 (decode)
- H.264 Annex G (scalable baseline profile up to 720p)
- H.264 BL/MP/HP
- H.264 Annex H (partial)
- Stereoscopic video
- JPEG (encode/decode)
- VC-1 SP/MP/AP
- AVS-1.0
- RealVideo® 8/9/10 (decode only)
- On2® VP6.2/VP7 (decode only)

### 1.3.3 Cortex-M3 MPU Subsystem Description

The Cortex-M3 MPU subsystem includes the following components:

- Two Cortex-M3 CPUs: One for SIMCOP control, and the other for RTOS, ISP, and display subsystem control
- ARMv7-M and Thumb-2 instruction set architecture
- Dedicated INTC with up to 64 physical interrupt events
- Two-level memory subsystem hierarchy
  - L1
    - 32-KB shared cache memory
  - L2 ROM + RAM
    - 64-KB RAM
    - 16-KB bootable ROM
- Cortex-M3 system bus directly connected to the ISS interconnect
- MMU for address translation
- Integrated power management
- Emulation feature embedded in the Cortex-M3

### 1.3.4 Display Subsystem Description

The display subsystem provides the control signals required to interface the OMAP system memory frame buffer (SDRAM) directly to the displays. It supports hardware cursor, independent gamma curve on all interfaces, multiple-buffer, and programmable color phase rotation. The display subsystem allows low-power display refresh and arbitration between normal and low-priority pipelines.

The display subsystem consists of the following sections:

- Display controller: It can read and display the encoded pixel data stored in memory and write the output of one of the overlays or one of the pipelines into the system memory. It supports the following components:
  - Three video pipelines, one graphic pipeline, and one write-back pipeline. The graphic pipeline supports pixel formats such as: ARGB16-4444, RGB16-565, ARGB16-1555, ARGB32-8888, RGBA32-8888, RGB24-888, and BITMAP (1, 2, 4, or 8 bits per pixel). It allows selection of the color-depth expansion.
  - Write-back pipeline: it uses poly-phase filtering for independent horizontal and vertical resampling (upsampling and downsampling). It allows programmable color space conversion of RGB24 into YUV4:2:2-UYYV, YUV4:2:2-YUV2, or YUV4:2:0-NV12, and selection of color-depth reduction from RGB24 to RGB16.
  - Two LCD outputs, each one with dedicated overlay manager, for support of passive matrix color and monochrome displays (up to 8-bit interface) and active matrix color displays (up to 24-bit interface). Secondary LCD output is available through parallel CMOS interface for MIPI®-DPI 1.0 support.
  - One TV output with dedicated overlay manager
  - Own direct memory access (DMA) engine

- Remote frame buffer interface (RFBI) module.
  - Support for MIPI-DBI protocol
  - 8-/9-/16-bit parallel interface
  - Programmable pixel modes and output formats
- Two MIPI display serial interfaces (DSIs) with the following main features:
  - Support for MIPI-DSI (four data-lane complex inputs/outputs (I/Os) for DSI1 and two data-lane complex I/Os for DSI2)
  - Support for video mode and command mode
  - Data interleaving support for synchronous and asynchronous streams
  - Bidirectional data link support
- High-definition multimedia interface (HDMI) encoder with the following main features:
  - HDMI 1.3, HDCP 1.2, and DVI 1.0 compliant
  - Deep-color mode support (10-bit for up to 1080p and up to 12-bit for 1080i/720p)
  - Support for uncompressed multichannel audio
  - Integrated high-bandwidth digital content protection (HDCP) encryption engine for transmitting protected audio and video content
  - Integrated transition minimized differential signaling (TMDS) and TER4 encoders for data island support
- NTSC/PAL video encoder with the following main features:
  - Output to on-chip video digital-to-analog converter (VDAC) providing composite analog output signal: NTSC-J, M; PAL-B, D, G, H, I; PAL-M
  - Support for square pixel sampling
  - Programmable horizontal synchronization, vertical timing, and waveforms

### 1.3.5 ABE Subsystem Description

The ABE subsystem handles audio processing for the application. It manages the audio and voice streams between the Cortex-A9 MPU subsystem and/or DSP, and the physical interfaces.

The ABE subsystem allows:

- Buffering of audio samples
- Mixing audio with voice downstream and/or microphone upstream (sidetone)
- Postprocessing of equalization, 3D effects, bass-boost

The ABE subsystem consists of:

- Audio engine (AE) subsystem, which performs real-time signal processing such as:
  - Muxing and mixing voice and data streams
  - Postprocessing operations such as sampling rate conversion, volume control, 3D effects
  - Execution of whole data transfers in the ABE subsystem using audio traffic controller (ATC)

The AE subsystem includes an AE and has the following on-chip memories available: 64-KB data memory (DMEM); 6-KB coefficient memory (CMEM); and 18-KB sample memory (SMEM).  
The ATC manages the data movement in the ABE subsystem and is in charge of interrupt generation to the DSP and Cortex-A9 MPU subsystems.
- Four general-purpose timers (GPTIMERS) and one watchdog timer (WDTIMER)
- Peripheral interfaces:
  - Three multichannel buffered serial ports (McBSPs) for inter-IC sound (I<sup>2</sup>S™) external connectivity
  - One multichannel audio serial port (McASP) supporting Sony/Philips digital interconnect format (S/PDIF) output
  - One MIPI SLIMbus interface to support new generations of MIPI-compliant components
  - One digital microphone (DMIC) for three stereo digital microphones support
  - One multichannel pulse-density modulation (McPDM) interface, which ensures communication with the TWL6040 audio companion chip
- Internal interfaces for connection with the DSP and Cortex-A9 MPU subsystems and other modules in

the device

- Dedicated power domain (ABE power domain)

### 1.3.6 ISS Description

The ISS processes data coming from the image sensor, memory, and IVA-HD subsystem. The ISS is responsible for multimedia applications such as: camera viewfinder; video record with up to 1080 p at 30 fps with digital zoom and still image processing, such as image capture up to 16 Mp with digital zoom and rotation. The ISS supports a pixel throughput of up to 200 Mp/s. It assures good performance with sensors up to 16 Mp and more (higher resolution can be achieved through multiple passes). The ISS can implement third-party algorithms for further flexibility when working with image sensors.

The ISS consists of:

- The ISP, which deals with on-the-fly or memory-to-memory data processing. It allows data collection for autoexposure, autowhite balance, autofocus, resizing, and histogram generation. The ISP consists of:
  - Image pipe interface (IPIPEIF) for synchronization signals (HD, VD) for the ISIF, IPIPE, RSZ, and hardware 3A (H3A) modules, and data transfer from video port, SDRAM, ISIF. Various pixel data manipulation functions.
  - Image pipe (IPIPE) front-end and back-end modules for raw data processing and RGB and YUV data processing, respectively. They support:
    - Sensor data linearization for dynamic range extension
    - Programmable 2D lens shading compensation correction
    - Black-level compensation
    - Gamma correction
    - RGB color correction
    - RGB to YUV4:2:2 color conversion
    - 3D look up table (LUT) for color correction
    - 2D edge enhancement
    - False chroma suppression
  - H3A for autowhite balance, autoexposure, and autofocus
  - Pattern generator (PG) for internal data generation for test purposes. It provides the ability to test some of the ISP submodules without the use of an external image sensor.
  - Two independent resizers, which allow YUV4:2:2 to YUV4:2:0 planar Chroma filtering and downsampling. The resizers support input and output flows with up to 200 Mp/s, and memory-to-memory rescaling in the range  $\times 1/4096$  scale down, and  $\times 20$  scale up.
  - Image sensor interface (ISIF) can process the incoming data and supports the following main functions:
    - Sensor data linearization
    - Supports VGA read out mode
    - Color space conversion
    - Digital clamp with horizontal/vertical offset drift compensation
    - Vertical line defect correction
    - Programmable 2D-matrix lens shading correction
    - 10-to-8 bits A-Law compression table inside
  - Buffer logic (BL), which processes and manages the requests to the module and memory subsystem
- Peripheral serial interfaces for connection with sensors and memories:
  - Two PHYs, CSIPHY1 and CSIPHY2, for physical connection to external sensors
  - Peripheral serial interfaces CSI2-A and CSI2-B/CCP2 for image data transfer from sensors to memory or ISP
- SIMCOP module for memory-to-memory operation; JPEG encode/decode hardware acceleration; high-ISO filtering; block-based rotation; warping and fusion; and general-purpose imaging acceleration. The SIMCOP includes the following main submodules:
  - Two imaging extension (iMX) modules - programmable image and video processing engines

- Noise filter 2 (NSF2) – for advanced noise filtering and edge-enhancement
- Variable-length coder/decoder for JPEG (VLCDJ) module
- Discrete cosine transform (DCT) module
- Lens distortion correction (LDC) module
- Rotation accelerator (ROT) engine
- Hardware sequencer, which offloads sequencing tasks from the MPU
- Shared buffers/memories
- DMA controller
- Timing control module for CAM global reset control, CAM flash strobe, and CAM shutter
- System interfaces and interconnects comprising:
  - Two configuration interfaces
  - One 128-bit master data interface
  - Internal ISS interconnects for image data and configuration
  - On-chip RAM interface
  - Circular buffer (CBUFF) and burst-translation engine (BTE) for efficient communication with external memory (SDRAM/TILER support)

### 1.3.7 2D/3D Graphics Accelerator Description

The 2D/3D graphics accelerator subsystem is based on POWERVR® SGX540 core from Imagination Technologies. It supports phone/PDA and handheld gaming applications. The SGX can process different data types simultaneously, such as: pixel data, vertex data, video data, and general-purpose data processing.

The SGX subsystem has the following features:

- Universal scalable shader engine ( USSE™), multithreaded engine incorporating pixel and vertex shader functionality to reduce die area
- Advanced shader feature set in excess of Microsoft VS3.0, PS3.0, and OGL2.0
- Industry-standard API supports Direct3D™ Mobile, OGL-ES 1.1 and 2.0, OpenVG™ 1.1, and OpenMAX™
- Fine-grained task switching, load balancing, and power management
- Programmable high-quality image antialiasing
- Advanced geometry DMA driven operation for minimum CPU interaction
- Fully virtualized memory addressing for OS operation in a unified memory architecture
- Advanced and standard 2D operations, such as vector graphics, BLTs, ROPs, etc.
- Programmable video encode and decode support for H.264, H.263, MPEG-4 (SP), WMV9, and JPEG

### 1.3.8 Face Detect Module Description

The face detect module is a stand-alone module that performs face detection and tracking on a picture stored in the SDRAM memory. It communicates with the Cortex-A9 MPU, DSP, and Cortex-M3 MPU subsystems.

Face detect is typically used on:

- Video encoding
- Face-based priority auto-focusing
- Red-eye removal

The face detect module comprises:

- Face detection core with embedded DMA engine for data memory access
- RAM and ROM memories
- L3 and L4 port interfaces

### 1.3.9 On-Chip Debug Support Description

The on-chip debug support has the following features:



- Multiprocessor debugging lets users control multiple CPU cores embedded in the device, such as:
  - Global starting and stopping of individual or multiple processors
  - Each processor can generate triggers that can be used to alter the execution flow of other processors
  - System clocking and power down
  - Interconnection of multiple devices
  - Channel triggering
- Target debugging, using JTAG (IEEE1149.1) or cJTAG (IEEE1149.7) port
- Reduction of power consumption in normal operating mode
- Real-time software trace allows the OMAP software masters to transmit trace data from OS processes or tasks on 256 different channels.

The debug subsystem includes:

- IEEE1149.7 adapter
- Generic TAP for emulation and test control (ICEPick-D™)
- Debug access port (DAP)
- Processor trace subsystem
- System trace subsystem
- EMU configuration interconnect
- Cross-triggering unit (XTRIGGER)
- Debug resource manager (DRM)

ICEMelter:

- Controls the wake-up and power-down of the emulation power domain

CORE instrumentation interconnect:

- Initiator ports:
  - L3 interconnect (for software instrumentation and performance probes)
  - OCP-WP
  - IVA-HD instrumentation (HWA profiling)
  - CM2 instrumentation
- Target port:
  - EMU instrumentation interconnect

OCP watch-point (OCP-WP):

- Monitors L3 interconnect transaction when target transaction attributes match the user-defined attributes or trigger on external debug event
- Only one instance, shared among the following L3 targets:
  - GPMC
  - L4\_PER
  - L4\_CFG
 Power Management Events Profiler (PM instrumentation),  
 Clock Management Events Profiler (CM instrumentation),  
 Statistics Collector (performance probes).

### **1.3.10 Power, Reset, and Clock Management Description**

The PRCM module allows efficient control of clocks and power according to the required performance, and reduction of power consumption.

The PRCM module is divided into:

- Power and reset management (PRM), based on the SmartReflex™ framework with the following features:
  - Dynamic clock gating

- Dynamic voltage and frequency scaling (DVFS)
- Dynamic power switching (DPS)
- Static leakage management (SLM)
- Adaptive body bias (ABB)
- Retention-till-access (RTA) for memories
- Clock management 1 (CM1) for clock generation, distribution, and management for the Cortex-A9 MPU, ABE, and CORE always-on power domains. The clock management allows reduction of dynamic consumption.
- Clock management 2 (CM2) for clock generation, distribution, and management for other modules

### 1.3.11 On-Chip Memory Description

The on-chip memory is divided into L3 OCM RAM, SAR ROM, SAR RAM, and memories in the subsystems (Cortex-A9, Cortex-M3, ABE, and IVA-HD).

- The L3 OCM RAM consists of 56KB of on-chip SRAM.
- The save-and-restore (SAR) ROM consists of 4KB and contains a linked list of descriptors used by the system DMA (sDMA).
- The SAR RAM consists of 8KB divided into four blocks. It is used as context-saving memory when the device goes into off mode.

### 1.3.12 Memory Management Description

The memory management is performed from:

- sDMA controller with up to 127 requests, 32 prioritizable logical channels, and 256 × 64-bit FIFO
- Dynamic memory management (DMM) module, which performs global address translation, address rotation (tiling), and access interleaving

### 1.3.13 External Memory Interface Description

There are two main interfaces for connection to external memories: general-purpose memory controller (GPMC) and dual-channel SDRAM controller (SDRC).

The GPMC supports:

- Asynchronous SRAM memories
- Asynchronous/synchronous NOR flash memories
- NAND flash memories
- Pseudo-SRAM devices

The SDRC/EMIF allows:

- Connection between the device and LPDDR2-type memory. It supports double-data rate (DDR) and single-data rate (SDR) protocols. The EMIF is the interface between LPDDR2 SDRAM/NVM and the Cortex-A9 MPU subsystem, ISS, IVA-HD subsystem, SGX, and DMA controllers.
- PHY is the DDR physical interface, which implements data-rate conversion in compliance with LPDDR2 JEDEC requirements.

### 1.3.14 System and Connection Peripherals

The OMAP device supports a comprehensive set of peripherals to provide flexible and high-speed (HS) interfacing and on-chip programming resources.

#### 1.3.14.1 System Peripherals

- Seven GPTIMERS
- WDTIMER
- One 32-kHz synchronization timer (32KTIMER)
- System control module, which contains registers for the following functions:
  - Static device configuration

- Debug and observability
- Status
- Pad configuration
- I/O configuration
- eFuse logic
- Analog function control
- System boot decoding logic
- System mailbox with eight mailbox message queues
- One SPINLOCK module with 32 hardware semaphores, which can service tasks between the Cortex-A9 MPU, DSP, and Cortex-M3 MPU subsystems
- One chip-to-chip (C2C) interface, which services the communication between the OMAP device and external devices

#### **1.3.14.2 Connection Peripherals**

- Three universal asynchronous receiver/transmitter (UART) modules as serial-communication interfaces
- One UART + IrDA SIR up to FIR + TV remote control interface (CIR)
- McBSP module to provide full-duplex serial communication between the OMAP and other applications chips and codecs
- Five HS I<sup>2</sup>C™ controller modules; four of them are general-purpose modules with rates up to 3.4 Mbps, and the fifth one, in the PRCM module, performs dynamic voltage control and power sequencing with an external power IC.
- HDQ™/ 1-Wire® – Benchmarq HDQ and Dallas Semiconductor 1-Wire protocols interface
- Two HS MMC/SD/SDIO modules with 8-bit data bus interface, that can act as an initiator on L3 interconnect thanks to an embedded DMA
- Three HS MMC/SD/SDIO modules with 4-bit data bus interface
- Six general-purpose input/output (GPIO) modules with 32 I/Os each
- One keyboard controller, which supports up to 9 × 9 matrix keypads
- One MIPI SLIMbus interface
- Four multichannel serial peripheral interface (MCSPi) modules
- One HS universal serial bus (USB) On-The-Go (OTG) module with embedded PHY, compliant with the USB2.0 (up to 480 Mbps) standard for HS functions and with the OTG supplement
- One HS multiport USB host module, which can be used for interchip connection or with an off-chip transceiver. It is compliant with the USB2.0 standard. The USB host module allows communication with USB peripherals with data rates up to 480 Mbps for HS, up to 12 Mbps for full-speed, and up to 1.5 Mbps for low-speed.
- One full-speed USB module compliant with the USB1.1 standard for full-speed functions
- One MIPI high-speed synchronous serial interface (HSI) module with two full-duplex serial communication interfaces. It is used for communication between the OMAP device and an external device, with data rates up to 192 Mbps for transmission, and up to 225 Mbps for reception. The MIPI HSI supports 16 logical channels on each destination (RX/TX).

## 1.4 Package-On-Package Concept

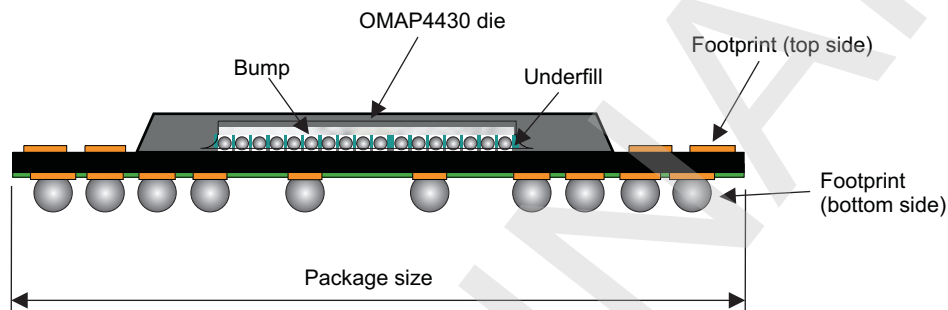
The OMAP4430 die uses flip-chip technology. The OMAP4430 package-on-page (POP) device supports memory stacking using a POP implementation.

The OMAP4430 die provides two LPDDR2 interfaces. Each interface supports up to two chip-selects, so up to four LPDDR2 memory dies are supported. Those interfaces are available only on device top ball out.

The two stacked memory packages are directly connected to the two LPDDR2 EMIF4D interfaces of the OMAP4430 die.

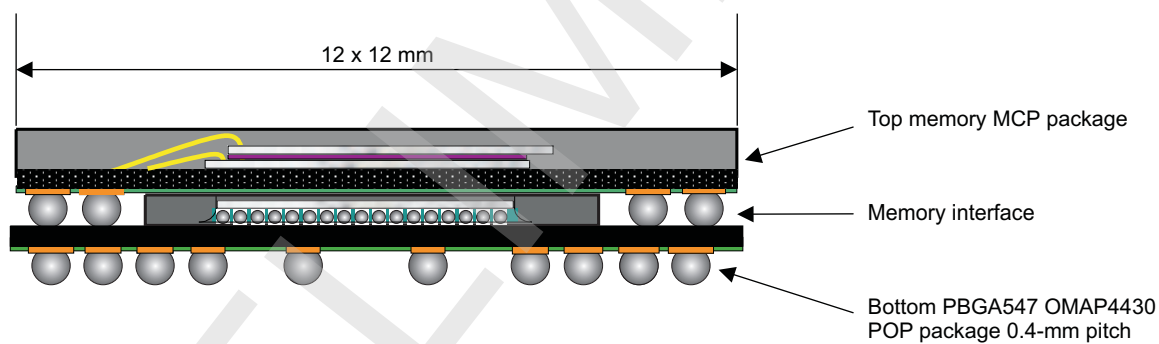
Figure 1-3 shows the concept of the POP solution, and Figure 1-4 shows stacked memory package on the POP device.

**Figure 1-3. POP Concept**



intro-003

**Figure 1-4. Stacked Memory Package on the POP Device**



intro-004

Two types of LPDDR - SDRAM memories are supported in POP package: S4 and S2 with size up to 2GB and 32-bit data width.

The POP device includes feedthroughs. The feedthroughs are defined from the bottom ball-grid array (BGA) to the stacked memory. The purpose of some of the feedthroughs is to provide power supply to the stacked memories.

## 1.5 Device Identification

[Table 1-1](#) describes the identification registers.

The identification registers include the data registers listed in [Table 1-2](#) and [Table 1-3](#). These registers are read-only accessed ports that are programmed into eFuses FARM FROM.

**Table 1-1. Device Identification Registers**

Register Name	Alias Name	Physical Address	Address Offset	Size
STD_FUSE_DIE_ID_0[31:0]	DIE_ID[31:0]	0x4A00 2200	0x200	32
ID_CODE[31:0]	ID_CODE[31:0]	0x4A00 2204	0x204	32
STD_FUSE_DIE_ID_1[31:0]	DIE_ID[63:32]	0x4A00 2208	0x208	32
STD_FUSE_DIE_ID_2[31:0]	DIE_ID[95:64]	0x4A00 220C	0x20C	32
STD_FUSE_DIE_ID_3[31:0]	DIE_ID[127:96]	0x4A00 2210	0x210	32
<a href="#">STD_FUSE_PROD_ID_0[31:0]</a>	PROD_ID[31:0]	0x4A00 2214	0x214	32
<a href="#">STD_FUSE_PROD_ID_1[31:0]</a>	PROD_ID[63:32]	0x4A00 2218	0x218	32

**Table 1-2. STD\_FUSE\_DIE\_ID**

Field	Alias Name	Bits	Value
STD_FUSE_DIE_ID_0[31:0]	DIE_ID[31:0]	–	Single identifier
STD_FUSE_DIE_ID_1[31:0]	DIE_ID[63:32]	–	Single identifier
STD_FUSE_DIE_ID_2[31:0]	DIE_ID[95:64]	–	Single identifier
STD_FUSE_DIE_ID_3[31:0]	DIE_ID[127:96]	–	Single identifier

The product type can be read in the HAWKEYE bit field value of the [Table 1-3](#) register. The silicon revision can be read in the VERSION bit field value of the [Table 1-3](#) register.

**Table 1-3. ID\_CODE**

Field	Bits	Value	Comment
ID_CODE[31:28]	VERSION	See <a href="#">Table 1-5</a> .	Revision number
ID_CODE[27:12]	HAWKEYE	See <a href="#">Table 1-4</a> .	Hawkeye number
ID_CODE[11:1]	TI_IDM	0x17	Manufacturer identity (TI)
ID_CODE[0]	–	0x1	Always set to 1

The Hawkeye number is hardcoded in the design. [Table 1-4](#) lists the Hawkeye number values, and [Table 1-5](#) lists the revision number values.

**Table 1-4. Hawkeye Number Values**

Silicon Type	Field	Value
OMAP4430 ES1.0 and ES2.0	ID_CODE[27:12]	0xB852
OMAP4430 ES2.1	ID_CODE[27:12]	0xB95C

**Table 1-5. Revision Number Values**

Silicon Type	Field	Value
OMAP4430 ES1.0	ID_CODE[31:28]	0x0
OMAP4430 ES2.0	ID_CODE[31:28]	0x1
OMAP4430 ES2.1	ID_CODE[31:28]	0x3

**Table 1-6. ID\_CODE Values**

Silicon Type	Field	Value
OMAP4430 ES1.0	ID_CODE[31:0]	0x0B85 202F
OMAP4430 ES2.0	ID_CODE[31:0]	0x1B85 202F

**Table 1-6. ID\_CODE Values (continued)**

Silicon Type	Field	Value
OMAP4430 ES2.1	ID_CODE[31:0]	0x3B95 C02F

The device type can be read in the [Table 1-7](#) register.

**Table 1-7. STD\_FUSE\_PROD\_ID\_0**

<b>Address Offset</b>	0x0000 0214	<b>Instance</b>	GENERAL
<b>Physical Address</b>	0x4A00 2214		
<b>Description</b>	This register shows the device type.		
<b>Type</b>	R		

31 ... ..	... ..	... ..	8	7	6	5	4	3	2	1	0
RESERVED								DEVICE_TYPE			

Bits	Field Name	Description	Type	Reset Value
31:8	RESERVED	Reserved	R	0x-
7:0	DEVICE_TYPE	Define the device type 0xF0 = GP device Other values = Reserved	R	0x-

**Table 1-8. Register Call Summary for Register STD\_FUSE\_PROD\_ID\_0**

Device Identification

- [Device Identification: \[0\]](#)

**Table 1-9. STD\_FUSE\_PROD\_ID\_1**

<b>Address Offset</b>	0x0000 0218	<b>Instance</b>	GENERAL
<b>Physical Address</b>	0x4A00 2218		
<b>Description</b>	This register shows the device type.		
<b>Type</b>	R		

31 ... ..	... ..	... ..	8	7	6	5	4	3	2	1	0
RESERVED											

Bits	Field Name	Description	Type	Reset Value
31:0	RESERVED	Reserved	R	0x-

**Table 1-10. Register Call Summary for Register STD\_FUSE\_PROD\_ID\_1**

Device Identification

- [Device Identification: \[0\]](#)

## Memory Mapping

This chapter describes the memory mapping in the device.

Topic	Page
<b>2.1 Introduction</b> .....	<b>272</b>
<b>2.2 L3 Memory Space Mapping</b> .....	<b>274</b>
<b>2.3 L4 Memory Space Mapping</b> .....	<b>277</b>
<b>2.4 Dual Cortex-M3 Subsystem Memory Space Mapping</b> .....	<b>286</b>
<b>2.5 DSP Subsystem Memory Space Mapping</b> .....	<b>287</b>
<b>2.6 Display Subsystem Memory Space Mapping</b> .....	<b>289</b>

## 2.1 Introduction

The Cortex™-A9 microprocessor unit (MPU) has a 32-bit address port, which allows it to handle a 4-GB space divided into several regions, depending on the target type.

The memory map comprises the following features that are shared among the initiators (for example, the Cortex-A9 MPU subsystem or the image and video accelerator [IVA-HD] subsystem).

- Memory space: General-purpose memory controller (GPMC)
- Dynamic memory management controller (DMM)
- Register spaces: Level 3 (L3) and level 4 (L4) interconnects
- Dedicated spaces: IVA-HD subsystem, graphic accelerator (SGX), etc.

The GPMC and DMM are dedicated to memory connection. The GPMC is used for NOR/NAND flash and static RAM (SRAM) memories. The DMM is used for synchronous dynamic random access memory (SDRAM) memories, such as regular SDR (single-data rate)-SDRAM or mobile DDR (double-data rate)-SDRAM. For more information, see [Section 15.2, Dynamic Memory Manager](#), and [Section 15.3, EMIF Controller](#).

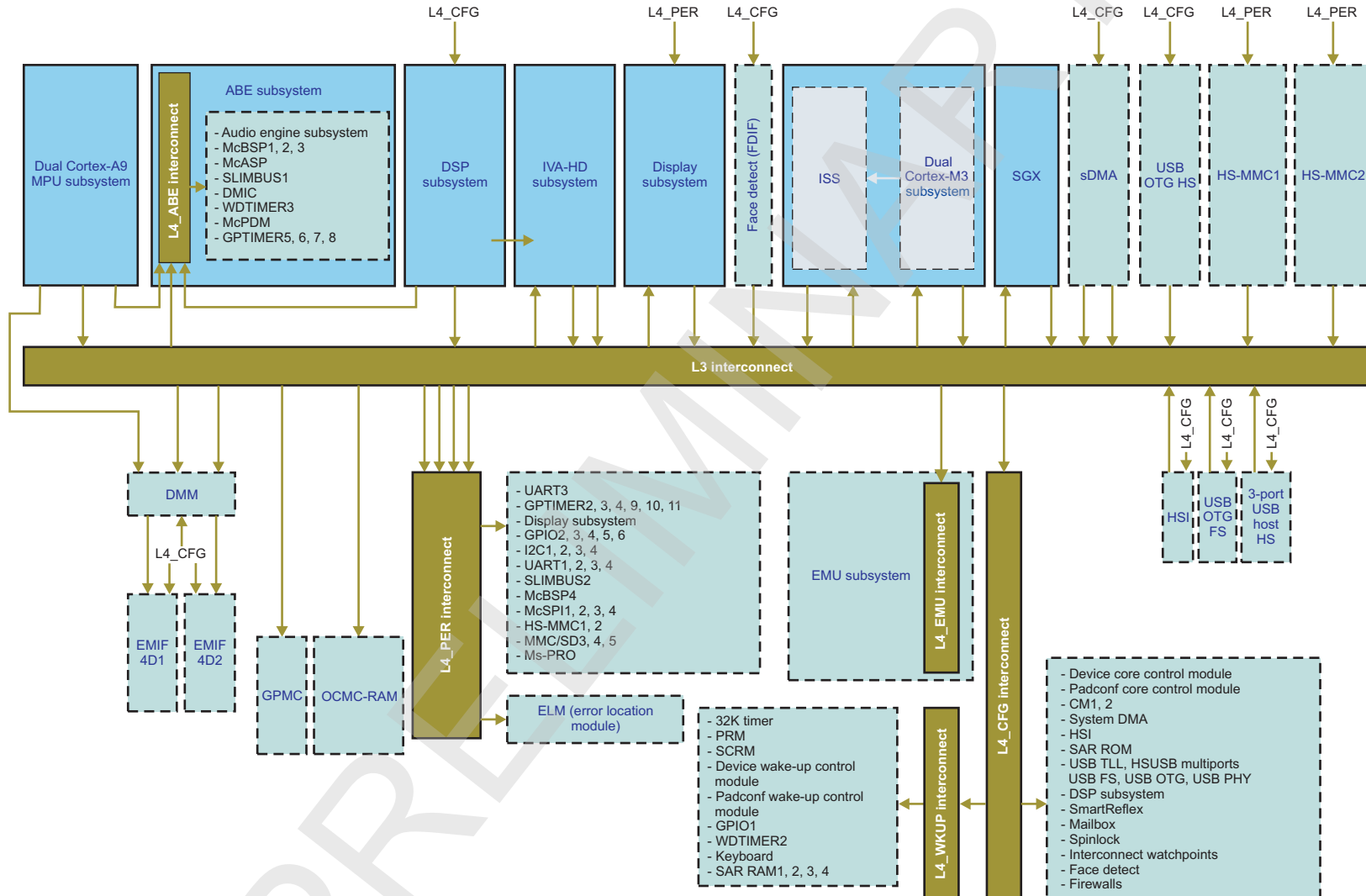
The L3 interconnect allows the sharing of resources, such as peripherals and external or on-chip memories, among all the initiators of the platform. The L4 interconnects control access to the peripherals.

Transfers across the platform between initiators and targets are physically conditioned by the chip interconnect and can be logically conditioned by firewalls. For more information about the intercommunication (L3 and L4 interconnects) and protection mechanisms implemented in the device, see [Section 13.2, L3 Interconnect](#), and [Section 13.3, L4 Interconnects](#).

[Figure 2-1](#) shows the interconnect of the device and the main modules and subsystems in the platform.



Figure 2-1. Interconnect Overview



memmap-001

## 2.2 L3 Memory Space Mapping

The memory space system is hierarchical: level 1 (L1), level 2 (L2), L3, and L4. L1 and L2 are memories in the Cortex-A9 MPU, Cortex™-M3 MPU, and the digital signal processor (DSP) subsystems. L3 handles many types of data transfers, including data exchange with system on-chip/external memories. The chip-level interconnect, which consists of one L3 and four L4s, enables communication among all modules and subsystems.

This section provides a global view of the memory mapping of the device at the L3 interconnect and describes the boot, GPMC, and SDRAM controller (SDRC) (EMIF/DMM) spaces.

The system memory mapping is flexible, with two levels of granularity for target address space allocation:

- L1: Four quarters are labeled Q0, Q1, Q2, and Q3. Each quarter corresponds to a 1-GB address space (total address space is 4GB).
- L2: Each quarter is divided into eight blocks of 32MB, with target spaces mapped in the blocks.

This organization allows all target spaces to be decoded based on the 7 most-significant bits (MSBs) of the 32-bit address ([31:25]).

- Boot space:

The system has a 48-KB boot space in the on-chip boot ROM or on the GPMC space.

When booting from the on-chip ROM with the appropriate external sys\_boot pin configuration, the 48-KB memory space is redirected to the on-chip boot ROM address space [0x4002 0000 – 0x4002 BFFF].

When booting from the GPMC with the appropriate external sys\_boot pins configuration, the memory space is part of the GPMC space. At reset, the 0x0000 0000 address is available on chip-select 0(CS0) for a memory size of 16MB.

For more information about sys\_boot pins configuration, see [Section 15.4, General-Purpose Memory Controller](#), and [Chapter 27, Initialization](#).

- GPMC space:

Eight independent GPMC chip-selects (gpmc\_ncs0 to gpmc\_ncs7) are available in the first quarter (Q0) of the addressing space to access NOR/NAND flash and SRAM. The chip-selects have a programmable start address and programmable size (16MB, 32MB, 64MB, or 128MB) in a total memory space of 1GB.

- EMIF1/EMIF2 CS0 space:

Q2 addressing space is interleaved on two LPDDR-memory controllers (EMIF1|EMIF2), each activating its CS0 line (lpddr21\_ncs0 and lpddr22\_ncs0). These chip-selects can address 64, 128, 256, 512, 1024, and 2048MBs. Interleaving occurs from 128-byte granularity up to 512-byte granularity.

Both memories must be of the same kind (SDRAM or nonvolatile memory [NVM]) and have the same density memory device.

The EMIF1-CS0 base address is always 0x8000 0000 at reset, and occupies a 1-GB address space at reset (interleaving disabled at reset).

- EMIF1/EMIF2 CS1 space:

Q3 addressing space is interleaved on two LPDDR-memory controllers (EMIF1|EMIF2), each activating its CS1 line (lpddr21\_ncs1 and lpddr22\_ncs1). These chip-selects can be programmed to 64, 128, 256, 512, and 1024MBs. Interleaving occurs at 128-MB granularity.

EMIF1-CS1 and EMIF2-CS1 are disabled at reset. Their base address is programmable to achieve a continuous address space with the respective CS0, regardless of the address range programmed.

EMIF1-CS1 is disabled if EMIF1-CS0 memory density is set to 2048MB (2GBs) when interleaving is disabled, or if EMIF1-CS0 + EMIF2-CS0 memory density are set to 1024MB (1GB) when interleaving is enabled.

- Tiler space:

Q3 addressing space is also used to access the Tiler system. This space is only visible for the imaging subsystem (ISS) and display subsystem.

[Table 2-1](#) describes the global memory space mapping.

**Table 2-1. Global Memory Space Mapping**

Quarter	Module Name	Start Address (hex)	End Address (hex)	Size	Description
Q0 (1GB)	Boot space GPMC <sup>(1)</sup>	0x0000 0000	0x3FFF FFFF	1GB	
	GPMC	0x0000 0000	0x3FFF FFFF	1GB	8/16 Ex <sup>(2)</sup> /R/W
Q1 (1GB)	On-chip memory	0x4000 0000	0x7FFF FFFF	1GB	ROM/SRAM address space
	Reserved	0x4000 0000	0x4002 7FFF	160KB	Reserved
	Boot ROM internal	0x4002 8000	0x4003 3FFF	48KB	32-bit Ex <sup>(2)</sup> /R Boot ROM
	Reserved	0x4003 4000	0x400F FFFF	816KB	Reserved
	Audio back-end (ABE)	0x4010 0000	0x401F FFFF	1MB	ABE domain (direct Cortex-A9 MPU access). See <a href="#">Table 2-6</a> .
	Reserved	0x4020 0000	0x402F FFFF	1MB	Reserved
	L3 OCM_RAM	0x4030 0000	0x4030 DFFF	56KB	32-bit Ex/R/W
	Reserved	0x4030 E000	0x43FF FFFF	63MB	Reserved
	L3 configuration registers	0x4400 0000	0x44FF FFFF	16MB	L3 configuration registers
	Reserved	0x4500 0000	0x47FF FFFF	48MB	Reserved
	L4_PER domain	0x4800 0000	0x48FF FFFF	16MB	Peripheral domain (see <a href="#">Table 2-5</a> )
	L4_ABE domain	0x4900 0000	0x49FF FFFF	16MB	ABE domain (double-mapped for Cortex-A9 MPU)
	L4_CFG domain	0x4A00 0000	0x4AFF FFFF	16MB	Configuration register domain (see <a href="#">Section 2.2, L3 Memory Space Mapping</a> ). Also Includes L4_WKUP.
	Reserved	0x4B00 0000	0x4BFF FFFF	16MB	Reserved
	EMIF1 registers	0x4C00 0000	0x4CFF FFFF	16MB	Configuration registers
	EMIF2 registers	0x4D00 0000	0x4DFF FFFF	16MB	Configuration registers
	DMM registers	0x4E00 0000	0x4FFF FFFF	32MB	Configuration registers
	GPMC	0x5000 0000	0x51FF FFFF	32MB	Configuration registers
	ISS	0x5200 0000	0x53FF FFFF	32MB	ISP5 (128KB) + SIMCOP (128KB)
	L3_EMU domain	0x5400 0000	0x54FF FFFF	16MB	Emulation domain (see <a href="#">Table 2-2</a> )
	Dual Cortex-M3 subsystem target	0x5500 0000	0x55FF FFFF	16MB	Dual Cortex-M3 subsystem target
	SGX	0x5600 0000	0x57FF FFFF	32MB	3D SGX540 domain
	Display subsystem	0x5800 0000	0x58FF FFFF	16MB	Display subsystem domain
	Reserved			16MB	Reserved
	IVA-HD configuration	0x5A00 0000	0x5AFF FFFF	16MB	IVA-HD domain
	IVA-HD SL2	0x5B00 0000	0x5BFF FFFF	16MB	IVA-HD SL2 domain
	C2C	0x5C00 0000	0x5FFF FFFF	64MB	Chip-to-chip domain
Tiler	0x6000 0000	0x7FFF FFFF	512MB	Tiler address mapping	
Q2 (1GB)	DRAM address space	0x8000 0000	0xBFFF FFFF	1GB	DDR-SDRAM CS0 address space
	EMIF1-CS0 SDRAM	0x8000 0000	0xBFFF FFFF	1GB	DDR Ctrl1 chip-select 0
	EMIF2-CS0 SDRAM	0x8000 0000	0xBFFF FFFF	1GB	DDR Ctrl2 chip-select 0
Q3 (1GB)	DRAM address space	0xC000 0000	0xFFFF FFFF	1 GB	DDR-SDRAM CS1 address space
	EMIF1-CS1 SDRAM	0xC000 0000	0xFFFF FFFF	1GB	DDR Ctrl1 chip-select 1
	EMIF2-CS1 SDRAM	0xC000 0000	0xFFFF FFFF	1GB	DDR Ctrl2 chip-select 1
	Tiler view (visible only for ISS and display subsystem)				
	Tiler view 0	0x1 0000 0000	0x1 1FFF FFFF	512MB	Natural view
	Tiler view 1	0x1 2000 0000	0x1 3FFF FFFF	512MB	0-degree view with vertical mirror
	Tiler view 2	0x1 4000 0000	0x1 5FFF FFFF	512MB	0-degree view with horizontal mirror
Tiler view 3	0x1 6000 0000	0x1 7FFF FFFF	512MB	180-degree view	

<sup>(1)</sup> Boot space location depends on the external sys\_boot[5:0] pins.

<sup>(2)</sup> Ex = Executable

**Table 2-1. Global Memory Space Mapping (continued)**

Quarter	Module Name	Start Address (hex)	End Address (hex)	Size	Description
	Tiler view 4	0x1 8000 0000	0x1 9FFF FFFF	512MB	90-degree view with vertical mirror
	Tiler view 5	0x1 A000 0000	0x1 BFFF FFFF	512MB	270-degree view
	Tiler view 6	0x1 C000 0000	0x1 DFFF FFFF	512MB	90-degree view
	Tiler view 7	0x1 E000 0000	0x1 FFFF FFFF	512MB	90-degree view with horizontal mirror

### 2.2.1 L3\_EMU Memory Space Mapping

The L3\_EMU interconnect is a 2-MB space composed of the L3\_EMU interconnect configuration registers and module registers.

Table 2-2 describes the mapping of the registers for the L3\_EMU interconnect.

**Table 2-2. L3\_EMU Memory Space Mapping**

Region Name	Start Address (hex)	End Address (hex)	Size
MIPI_STM [256 × 4K channels] (address space 0)	0x5400 0000	0x540F FFFF	1MB
MIPI_STM [256 × 1K channels] (address space 1)	0x5410 0000	0x5413 FFFF	256KB
Cortex-A9 CPU0 debug/PMU	0x5414 0000	0x5414 1FFF	8KB
Cortex-A9 CPU1 debug/PMU	0x5414 2000	0x5414 3FFF	8KB
Reserved	0x5414 4000	0x5414 7FFF	16KB
CTI0 component	0x5414 8000	0x5414 8FFF	4KB
CTI1 component	0x5414 9000	0x5414 9FFF	4KB
Reserved	0x5414 A000	0x5414 BFFF	8KB
PTM0 component	0x5414 C000	0x5414 CFFF	4KB
PTM1 component	0x5414 D000	0x5414 DFFF	4KB
Reserved	0x5414 E000	0x5415 7FFF	8KB
Trace funnel	0x5415 8000	0x5415 8FFF	4KB
DAP_PC	0x5415 9000	0x5415 9FFF	4KB
Reserved	0x5415 A000	0x5415 EFFF	20KB
APB bridge control and time-out register	0x5415 F000	0x5415 FFFF	4KB
DRM (interconnect)	0x5416 0000	0x5416 0FFF	4KB
MIPI_STM (interconnect)	0x5416 1000	0x5416 1FFF	4KB
ETB (APBv3)	0x5416 2000	0x5416 2FFF	4KB
CS_TPIU (APBv3)	0x5416 3000	0x5416 3FFF	4KB
CS_TF0 (APBv3)	0x5416 4000	0x5416 4FFF	4KB
Reserved	0x5416 5000	0x5416 6FFF	8KB
Technology-specific interconnect registers	0x5416 7000	0x5416 7FFF	4KB
Reserved	0x5416 8000	0x5417 FFFF	96KB
Technology-specific interconnect registers	0x5418 0000	0x5418 0FFF	4KB
Reserved	0x5418 1000	0x541F FFFF	508KB

## 2.3 L4 Memory Space Mapping

Four L4 interconnects handle transfers with peripherals. Each interconnect is in a distinct power domain:

- L4\_CFG: CORE power domain
- L4\_WKUP: WKUP power domain
- L4\_PER: PER power domain
- L4\_ABE: ABE power domain

As with the L3 interconnect, the L4 interconnect can be configured to tune the access according to the characteristics of each module.

The following sections describe the register mapping of the L4 interconnect. Software configures these registers.

### 2.3.1 L4\_CFG Memory Space Mapping

The L4\_CFG interconnect is a 16-MB space composed of the L4\_CFG interconnect configuration registers and the module registers.

Table 2-3 describes the mapping of the registers for the L4\_CFG interconnect.

**NOTE:** All memory spaces described as modules provide direct access to module registers outside the L4\_CFG interconnect. All other accesses are internal to the L4\_CFG interconnect.

**Table 2-3. L4\_CFG Memory Space Mapping**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
L4_CFG configuration	0x4A00 0000	0x4A00 07FF	2KB	Address/protection (AP)
	0x4A00 0800	0x4A00 0FFF	2KB	Link agent (LA)
	0x4A00 1000	0x4A00 1FFF	4KB	Initiator port (IP0)
SYSCTRL_GENERAL_CORE	0x4A00 2000	0x4A00 2FFF	4KB	Module
	0x4A00 3000	0x4A00 3FFF	4KB	L4 interconnect
CM1 - Manages Cortex-A9 MPU, DSP, IVA, ABE, memory interface, digital phase-locked loops (DPLLs) for Cortex-A9 MPU, DSP, ABE, CORE	0x4A00 4000	0x4A00 4FFF	4KB	Module region A
	0x4A00 5000	0x4A00 5FFF	4KB	L4 interconnect
Reserved	0x4A00 6000	0x4A00 7FFF	8KB	Reserved
CM2 (manages all others)	0x4A00 8000	0x4A00 9FFF	8KB	Module
	0x4A00 A000	0x4A00 AFFF	4KB	L4 interconnect
Reserved	0x4A00 B000	0x4A05 5FFF	300KB	Reserved
System DMA (sDMA)	0x4A05 6000	0x4A05 6FFF	4KB	Module
	0x4A05 7000	0x4A05 7FFF	4KB	L4 interconnect
HSI	0x4A05 8000	0x4A05 8FFF	4KB	HSI top
	0x4A05 9000	0x4A05 9FFF	4KB	HSI DMA
	0x4A05 A000	0x4A05 AFFF	4KB	HSI port 1
	0x4A05 B000	0x4A05 BFFF	4KB	HSI port 2
	0x4A05 C000	0x4A05 CFFF	4KB	L4 interconnect
Reserved	0x4A05 D000	0x4A05 DFFF	4KB	Reserved
SAR_ROM (save and restore)	0x4A05 E000	0x4A05 FFFF	8KB	Module
	0x4A06 0000	0x4A06 0FFF	4KB	L4 interconnect
Reserved	0x4A06 1000	0x4A06 1FFF	4KB	Reserved
HSUSBTLL	0x4A06 2000	0x4A06 2FFF	4KB	Module
	0x4A06 3000	0x4A06 3FFF	4KB	L4 interconnect

**Table 2-3. L4\_CFG Memory Space Mapping (continued)**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
HSUSBHOST	0x4A06 4000	0x4A06 4FFF	4KB	Module
	0x4A06 5000	0x4A06 5FFF	4KB	L4 interconnect
DSP subsystem	0x4A06 6000	0x4A06 6FFF	4KB	Module
	0x4A06 7000	0x4A06 7FFF	4KB	L4 interconnect
Reserved	0x4A06 8000	0x4A0A 8FFF	266KB	Reserved
FSUSB	0x4A0A 9000	0x4A0A 9FFF	4KB	Module
	0x4A0A A000	0x4A0A AFFF	4KB	L4 interconnect
HSUSBOTG	0x4A0A B000	0x4A0A BFFF	4KB	Module
	0x4A0A C000	0x4A0A CFFF	4KB	L4 interconnect
USBPHY	0x4A0A D000	0x4A0A DFFF	4KB	Module
	0x4A0A E000	0x4A0A EFFF	4KB	L4 interconnect
Reserved	0x4A0A F000	0x4A0D 8FFF	144KB	Reserved
SR_MPU (SmartReflex™ MPU)	0x4A0D 9000	0x4A0D 9FFF	4KB	Port SR MPU
	0x4A0D A000	0x4A0D AFFF	4KB	L4 interconnect
SR_IVA (SmartReflex IVA)	0x4A0D B000	0x4A0D BFFF	4KB	Port SR IVA
	0x4A0D C000	0x4A0D CFFF	4KB	L4 interconnect
SR_CORE (SmartReflex core)	0x4A0D D000	0x4A0D DFFF	4KB	Port SR core
	0x4A0D E000	0x4A0D EFFF	4KB	L4 interconnect
Reserved	0x4A0D F000	0x4A0F 3FFF	86KB	Reserved
Mailbox	0x4A0F 4000	0x4A0F 4FFF	4KB	Module
	0x4A0F 5000	0x4A0F 5FFF	4KB	L4 interconnect
Spinlock	0x4A0F 6000	0x4A0F 6FFF	4KB	Module
	0x4A0F 7000	0x4A0F 7FFF	4KB	L4 interconnect
Reserved	0x4A0F 8000	0x4A0F FFFF	32KB	Reserved
SYSCTRL_PADCONF_CORE	0x4A10 0000	0x4A10 0FFF	4KB	Module
	0x4A10 1000	0x4A10 1FFF	4KB	L4 interconnect
OCP-WP – Shared	0x4A10 2000	0x4A10 2FFF	4KB	Module
	0x4A10 3000	0x4A10 3FFF	4KB	L4 interconnect
Reserved	0x4A10 4000	0x4A10 9FFF	24KB	Reserved
Face detect	0x4A10 A000	0x4A10 AFFF	4KB	Module
	0x4A10 B000	0x4A10 BFFF	4KB	L4 interconnect
Reserved	0x4A10 C000	0x4A20 3FFF	992KB	Reserved
C2C-Init firewall	0x4A20 4000	0x4A20 4FFF	4KB	Module
	0x4A20 5000	0x4A20 5FFF	4KB	L4 interconnect
C2C-Target firewall	0x4A20 6000	0x4A20 6FFF	4KB	Module
	0x4A20 7000	0x4A20 7FFF	4KB	L4 interconnect
Reserved	0x4A20 8000	0x4A20 BFFF	16KB	Reserved
EMIF firewall	0x4A20 C000	0x4A20 CFFF	4KB	Configuration registers
	0x4A20 D000	0x4A20 DFFF	4KB	L4 interconnect
Reserved	0x4A20 E000	0x4A20 FFFF	8KB	Reserved
GPMC firewall	0x4A21 0000	0x4A21 0FFF	4KB	Configuration registers
	0x4A21 1000	0x4A21 1FFF	4KB	L4 interconnect
OCMC RAM firewall	0x4A21 2000	0x4A21 2FFF	4KB	Module
	0x4A21 3000	0x4A21 3FFF	4KB	L4 interconnect
GFX-T firewall	0x4A21 4000	0x4A21 4FFF	4KB	Module
	0x4A21 5000	0x4A21 5FFF	4KB	L4 interconnect

**Table 2-3. L4\_CFG Memory Space Mapping (continued)**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
ISS-T firewall	0x4A21 6000	0x4A21 6FFF	4KB	Module
	0x4A21 7000	0x4A21 7FFF	4KB	L4 interconnect
Dual Cortex-M3-T firewall	0x4A21 8000	0x4A21 8FFF	4KB	Module
	0x4A21 9000	0x4A21 9FFF	4KB	L4 interconnect
Reserved	0x4A21 A000	0x4A21 BFFF	8KB	Reserved
DSS-T firewall	0x4A21 C000	0x4A21 CFFF	4KB	Module
	0x4A21 D000	0x4A21 DFFF	4KB	L4 interconnect
SL2-T firewall	0x4A21 E000	0x4A21 EFFF	4KB	Module
	0x4A21 F000	0x4A21 FFFF	4KB	L4 interconnect
IVAHD-CFG-T firewall	0x4A22 0000	0x4A22 0FFF	4KB	Module
	0x4A22 1000	0x4A22 1FFF	4KB	L4 interconnect
Reserved	0x4A22 2000	0x4A22 5FFF	16KB	Reserved
L4-EMU firewall	0x4A22 6000	0x4A22 6FFF	4KB	Module
	0x4A22 7000	0x4A22 7FFF	4KB	L4 interconnect
L4_ABE firewall	0x4A22 8000	0x4A22 8FFF	4KB	Module
	0x4A22 9000	0x4A22 9FFF	4KB	L4 interconnect
Reserved	0x4A22 A000	0x4A2F FFFF	856KB	Reserved
L4_WKUP	0x4A30 0000	0x4A33 FFFF	256KB	WKUP domain (see <a href="#">Section 2.3.2, L4_WKUP Memory Space Mapping</a> )
	0x4A34 0000	0x4A34 0FFF	4KB	L4 interconnect
Reserved	0x4A34 1000	0x4AFF FFFF	13052KB	Reserved

### 2.3.2 L4\_WKUP Memory Space Mapping

The L4\_WKUP interconnect is a 256-KB space composed of the L4\_WKUP interconnect configuration registers and the module registers.

[Table 2-4](#) describes the mapping of the registers for the L4\_WKUP interconnect.

**NOTE:** All memory spaces described as modules provide direct access to module registers outside the L4\_WKUP interconnect. All other accesses are internal to the L4\_WKUP interconnect.

**Table 2-4. L4\_WKUP Peripheral Space Mapping**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
L4_WKUP configuration	0x4A30 0000	0x4A30 07FF	2KB	Address/protection (AP)
	0x4A30 0800	0x4A30 0FFF	2KB	Link agent (LA)
	0x4A30 1000	0x4A30 1FFF	4KB	Initiator port (IP0)
Reserved	0x4A30 2000	0x4A30 3FFF	8KB	Reserved
32KTIMER	0x4A30 4000	0x4A30 4FFF	4KB	Module
	0x4A30 5000	0x4A30 5FFF	4KB	L4 interconnect
PRM (power and reset manager)	0x4A30 6000	0x4A30 7FFF	8KB	Module
	0x4A30 8000	0x4A30 8FFF	4KB	L4 interconnect
Reserved	0x4A30 9000	0x4A30 9FFF	4KB	Reserved
SCRM	0x4A30 A000	0x4A30 AFFF	4KB	Module
	0x4A30 B000	0x4A30 BFFF	4KB	L4 interconnect
SYSCTRL_GENERAL_WKUP	0x4A30 C000	0x4A30 CFFF	4KB	Module
	0x4A30 D000	0x4A30 DFFF	4KB	L4 interconnect
Reserved	0x4A30 E000	0x4A30 FFFF	8KB	Reserved

**Table 2-4. L4\_WKUP Peripheral Space Mapping (continued)**

Module Name	Start Address (hex)	End Address (hex)	Size	Description	
GPIO1	0x4A31 0000	0x4A31 0FFF	4KB	Module	
	0x4A31 1000	0x4A31 1FFF	4KB	L4 interconnect	
Reserved	0x4A31 2000	0x4A31 3FFF	8KB	Reserved	
WDTIMER2	0x4A31 4000	0x4A31 4FFF	4KB	Module	
	0x4A31 5000	0x4A31 5FFF	4KB	L4 interconnect	
Reserved	0x4A31 6000	0x4A31 7FFF	8KB	Reserved	
GPTIMER1	0x4A31 8000	0x4A31 8FFF	4KB	Module	
	0x4A31 9000	0x4A31 9FFF	4KB	L4 interconnect	
Reserved	0x4A31 A000	0x4A31 BFFF	8KB	Module – Address space 0	
Keyboard	0x4A31 C000	0x4A31 CFFF	4KB	Module	
	0x4A31 D000	0x4A31 DFFF	4KB	L4 interconnect	
SYSCTRL_PADCONF_WKUP	0x4A31 E000	0x4A31 EFFF	4KB	Module	
	0x4A31 F000	0x4A31 FFFF	4KB	L4 interconnect	
SAR_RAM	Reserved	0x4A32 0000	0x4A32 5FFF	24KB	Reserved
		0x4A32 6000	0x4A32 6FFF	4KB	Module – SAR space 1
		0x4A32 7000	0x4A32 73FF	1KB	Module – SAR space 2
		0x4A32 7400	0x4A32 7FFF	3KB	Reserved
		0x4A32 8000	0x4A32 87FF	2KB	Module – SAR space 3
		0x4A32 8800	0x4A32 8FFF	2KB	Reserved
		0x4A32 9000	0x4A32 93FF	1KB	Module – SAR space 4
		0x4A32 9400	0x4A32 9FFF	3KB	Reserved
Reserved	0x4A32 A000	0x4A32 AFFF	4KB	L4 interconnect	
Reserved	0x4A32 B000	0x4A33 FFFF	84KB	Reserved	

**NOTE:** 8-bit and 16-bit peripherals are aligned on 32-bit address boundaries.

### 2.3.3 L4\_PER Memory Space Mapping

The L4\_PER interconnect is a 16-MB space composed of the L4\_PER interconnect configuration registers and the module registers.

Table 2-5 describes the mapping of the registers for the L4\_PER interconnect.

**NOTE:** All memory spaces described as modules provide direct access to the module registers outside the L4\_PER interconnect. All other accesses are internal to the L4\_PER interconnect.

**Table 2-5. L4\_PER Memory Space Mapping**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
L4_PER	0x4800 0000	0x4800 07FF	2KB	Address protection (AP)
	0x4800 0800	0x4800 0FFF	2KB	Link agent (LA)
	0x4800 1000	0x4800 13FF	1KB	Initiator port 0 (IP0)
	0x4800 1400	0x4800 17FF	1KB	Initiator port 1 (IP1)
	0x4800 1800	0x4800 1BFF	1KB	Initiator port 2 (IP2)
	0x4800 1C00	0x4800 1FFF	1KB	Initiator port 3 (IP3)
Reserved	0x4800 2000	0x4801 FFFF	120KB	Reserved



**Table 2-5. L4\_PER Memory Space Mapping (continued)**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
UART3	0x4802 0000	0x4802 0FFF	4KB	Module
	0x4802 1000	0x4802 1FFF	4KB	L4 interconnect
Reserved	0x4802 2000	0x4803 1FFF	64KB	Reserved
GPTIMER2	0x4803 2000	0x4803 2FFF	4KB	Module
	0x4803 3000	0x4803 3FFF	4KB	L4 interconnect
GPTIMER3	0x4803 4000	0x4803 4FFF	4KB	Module
	0x4803 5000	0x4803 5FFF	4KB	L4 interconnect
GPTIMER4	0x4803 6000	0x4803 6FFF	4KB	Module
	0x4803 7000	0x4803 7FFF	4KB	L4 interconnect
Reserved	0x4803 8000	0x4803 DFFF	24KB	Reserved
GPTIMER9	0x4803 E000	0x4803 EFFF	4KB	Module
	0x4803 F000	0x4803 FFFF	4KB	L4 interconnect
Display subsystem	0x4804 0000	0x4804 FFFF	64KB	Display subsystem – Configuration + data
	0x4805 0000	0x4805 0FFF	4KB	L4 interconnect
Reserved	0x4805 1000	0x4805 4FFF	16KB	Reserved
GPIO2	0x4805 5000	0x4805 5FFF	4KB	Module
	0x4805 6000	0x4805 6FFF	4KB	L4 interconnect
GPIO3	0x4805 7000	0x4805 7FFF	4KB	Module
	0x4805 8000	0x4805 8FFF	4KB	L4 interconnect
GPIO4	0x4805 9000	0x4805 9FFF	4KB	Module
	0x4805 A000	0x4805 AFFF	4KB	L4 interconnect
GPIO5	0x4805 B000	0x4805 BFFF	4KB	Module
	0x4805 C000	0x4805 CFFF	4KB	L4 interconnect
GPIO6	0x4805 D000	0x4805 DFFF	4KB	Module
	0x4805 E000	0x4805 EFFF	4KB	L4 interconnect
Reserved	0x4805 F000	0x4805 FFFF	4KB	Reserved
I2C3	0x4806 0000	0x4806 0FFF	4KB	Module
	0x4806 1000	0x4806 1FFF	4KB	L4 interconnect
Reserved	0x4806 2000	0x4806 9FFF	32KB	Reserved
UART1	0x4806 A000	0x4806 AFFF	4KB	Module
	0x4806 B000	0x4806 BFFF	4KB	L4 interconnect
UART2	0x4806 C000	0x4806 CFFF	4KB	Module
	0x4806 D000	0x4806 DFFF	4KB	L4 interconnect
UART4	0x4806 E000	0x4806 EFFF	4KB	Module
	0x4806 F000	0x4806 FFFF	4KB	L4 interconnect
I2C1	0x4807 0000	0x4807 0FFF	4KB	Module
	0x4807 1000	0x4807 1FFF	4KB	L4 interconnect
I2C2	0x4807 2000	0x4807 2FFF	4KB	Module
	0x4807 3000	0x4807 3FFF	4KB	L4 interconnect
Reserved	0x4807 4000	0x4807 5FFF	8KB	Reserved
SLIMBUS2	0x4807 6000	0x4807 6FFF	4KB	Module
	0x4807 7000	0x4807 7FFF	4KB	L4 interconnect
ELM (error locator module)	0x4807 8000	0x4807 8FFF	4KB	Module
	0x4807 9000	0x4807 9FFF	4KB	L4 interconnect
Reserved	0x4807 A000	0x4808 5FFF	48KB	Reserved
GPTIMER10	0x4808 6000	0x4808 6FFF	4KB	Module
	0x4808 7000	0x4808 7FFF	4KB	L4 interconnect

**Table 2-5. L4\_PER Memory Space Mapping (continued)**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
GPTIMER11	0x4808 8000	0x4808 8FFF	4KB	Module
	0x4808 9000	0x4808 9FFF	4KB	L4 interconnect
Reserved	0x4808 A000	0x4809 5FFF	48KB	Reserved
McBSP4	0x4809 6000	0x4809 6FFF	4KB	Module
	0x4809 7000	0x4809 7FFF	4KB	L4 interconnect
McSPI1	0x4809 8000	0x4809 8FFF	4KB	Module
	0x4809 9000	0x4809 9FFF	4KB	L4 interconnect
McSPI2	0x4809 A000	0x4809 AFFF	4KB	Module
	0x4809 B000	0x4809 BFFF	4KB	L4 interconnect
HSMMC1	0x4809 C000	0x4809 CFFF	4KB	Module
	0x4809 D000	0x4809 DFFF	4KB	L4 interconnect
Reserved	0x4809 E000	0x480A CFFF	60KB	Reserved
MMC/SD3	0x480A D000	0x480A DFFF	4KB	Module
	0x480A E000	0x480A EFFF	4KB	L4 interconnect
Reserved	0x480A F000	0x480B 1FFF	12KB	Reserved
HDQ™ (1-Wire®)	0x480B 2000	0x480B 2FFF	4KB	Module
	0x480B 3000	0x480B 3FFF	4KB	L4 interconnect
HSMMC2	0x480B 4000	0x480B 4FFF	4KB	Module
	0x480B 5000	0x480B 5FFF	4KB	L4 interconnect
Reserved	0x480B 6000	0x480B 6FFF	8KB	Reserved
MCSPI3	0x480B 8000	0x480B 8FFF	4KB	Module
	0x480B 9000	0x480B 9FFF	4KB	L4 interconnect
MCSPI4	0x480B A000	0x480B AFFF	4KB	Module
	0x480B B000	0x480B BFFF	4KB	L4 interconnect
Reserved	0x480B C000	0x480D 0FFF	84KB	Reserved
MMC/SD4	0x480D 1000	0x480D 1FFF	4KB	Module
	0x480D 2000	0x480D 2FFF	4KB	L4 interconnect
Reserved	0x480D 3000	0x480D 4FFF	8KB	Reserved
MMC/SD5	0x480D 5000	0x480D 5FFF	4KB	Module
	0x480D 6000	0x480D 6FFF	4KB	L4 interconnect
Reserved	0x480D 7000	0x4834 FFFF	2532KB	Reserved
I2C4	0x4835 0000	0x4835 0FFF	4KB	Module
	0x4835 1000	0x4835 1FFF	4KB	L4 interconnect
Reserved	0x4835 2000	0x48FF FFFF	13MB	Reserved

### 2.3.4 L4\_ABE Memory Space Mapping

The L4\_ABE interconnect is a 16-MB space composed of the L4\_ABE interconnect configuration registers and the module registers.

ABE modules are dual-mapped inside the Cortex-A9 MPU/DSP address space:

- Mapped in L3 as the L4 ABE space
- Mapped in Cortex-A9 MPU and DSP nonshared device map

---

**NOTE:** All memory spaces described as modules provide direct access to the module registers outside the L4\_ABE interconnect. All other accesses are internal to the L4\_ABE interconnect.

---

Table 2-6 describes the mapping of the registers for the L4\_ABE interconnect in the Cortex-A9 MPU nonshared (private) device address range (L4\_ABEP).

**Table 2-6. ABE Cortex-A9 MPU Memory Space Mapping**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
L3_ABE	0x4010 0000	0x4010 3FFF	16KB	ABE domain (direct MPU access)
Reserved	0x4010 4000	0x4012 1FFF	120KB	Reserved
McBSP1	0x4012 2000	0x4012 2FFF	4KB	Module
	0x4012 3000	0x4012 3FFF	4KB	L4 interconnect
McBSP2	0x4012 4000	0x4012 4FFF	4KB	Module
	0x4012 5000	0x4012 5FFF	4KB	L4 interconnect
McBSP3	0x4012 6000	0x4012 6FFF	4KB	Module
	0x4012 7000	0x4012 7FFF	4KB	L4 interconnect
McASP	0x4012 8000	0x4012 8FFF	4KB	McASP CFG port
	0x4012 9000	0x4012 9FFF	4KB	L4 interconnect
	0x4012 A000	0x4012 AFFF	4KB	McASP data port
	0x4012 B000	0x4012 BFFF	4KB	L4 interconnect
SLIMBUS1	0x4012 C000	0x4012 CFFF	4KB	Module
	0x4012 D000	0x4012 DFFF	4KB	L4 interconnect
DMIC	0x4012 E000	0x4012 EFFF	4KB	Module
	0x4012 F000	0x4012 FFFF	4KB	L4 interconnect
WDTIMER3	0x4013 0000	0x4013 0FFF	4KB	Module
	0x4013 1000	0x4013 1FFF	4KB	L4 interconnect
McPDM	0x4013 2000	0x4013 2FFF	4KB	Module
	0x4013 3000	0x4013 3FFF	4KB	L4 interconnect
Reserved	0x4013 4000	0x4013 7FFF	16KB	Reserved
GPTIMER5	0x4013 8000	0x4013 8FFF	4KB	Module
	0x4013 9000	0x4013 9FFF	4KB	L4 interconnect
GPTIMER6	0x4013 A000	0x4013 AFFF	4KB	Module
	0x4013 B000	0x4013 BFFF	4KB	L4 interconnect
GPTIMER7	0x4013 C000	0x4013 CFFF	4KB	Module
	0x4013 D000	0x4013 DFFF	4KB	L4 interconnect
GPTIMER8	0x4013 E000	0x4013 EFFF	4KB	Module
	0x4013 F000	0x4013 FFFF	4KB	L4 interconnect
Reserved	0x4014 0000	0x4017 FFFF	256KB	Reserved
DMEM (64KB)	0x4018 0000	0x4018 FFFF	64KB	Data memory
	0x4019 0000	0x4019 0FFF	4KB	L4 interconnect
Reserved	0x4019 1000	0x4019 FFFF	60KB	Reserved
CMEM (4KB)	0x401A 0000	0x401A FFFF	64KB	Coefficient memory
	0x401B 0000	0x401B 0FFF	4KB	L4 interconnect
Reserved	0x401B 1000	0x401B FFFF	60KB	Reserved
SMEM (32KB)	0x401C 0000	0x401C FFFF	64KB	Module
	0x401D 0000	0x401D 0FFF	4KB	L4 interconnect
Reserved	0x401D 1000	0x401F 0FFF	128KB	Reserved
AESS configuration	0x401F 1000	0x401F 1FFF	4KB	Module
	0x401F 2000	0x401F 2FFF	4KB	L4 interconnect
Reserved	0x401F 3000	0x401F FFFF	52KB	Reserved

Table 2-7 describes the mapping of the registers for the L3 interconnect.

**Table 2-7. ABE L3 Memory Space Mapping**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
L4 ABE	0x4900 0000	0x4900 3FFF	16KB	ABE domain (double-mapped for MPU)
Reserved	0x4900 4000	0x4902 1FFF	120KB	Reserved
McBSP1	0x4902 2000	0x4902 2FFF	4KB	Module
	0x4902 3000	0x4902 3FFF	4KB	L4 interconnect
McBSP2	0x4902 4000	0x4902 4FFF	4KB	Module
	0x4902 5000	0x4902 5FFF	4KB	L4 interconnect
McBSP3	0x4902 6000	0x4902 6FFF	4KB	Module
	0x4902 7000	0x4902 7FFF	4KB	L4 interconnect
McASP	0x4902 8000	0x4902 8FFF	4KB	McASP CFG port
	0x4902 9000	0x4902 9FFF	4KB	L4 interconnect
	0x4902 A000	0x4902 AFFF	4KB	McASP data port
	0x4902 B000	0x4902 BFFF	4KB	L4 interconnect
SLIMBUS1	0x4902 C000	0x4902 CFFF	4KB	Module
	0x4902 D000	0x4902 DFFF	4KB	L4 interconnect
DMIC	0x4902 E000	0x4902 EFFF	4KB	Module
	0x4902 F000	0x4902 FFFF	4KB	L4 interconnect
WDTIMER3	0x4903 0000	0x4903 0FFF	4KB	Module
	0x4903 1000	0x4903 1FFF	4KB	L4 interconnect
McPDM	0x4903 2000	0x4903 2FFF	4KB	Module
	0x4903 3000	0x4903 3FFF	4KB	L4 interconnect
Reserved	0x4903 4000	0x4903 7FFF	16KB	Reserved
GPTIMER5	0x4903 8000	0x4903 8FFF	4KB	Module
	0x4903 9000	0x4903 9FFF	4KB	L4 interconnect
GPTIMER6	0x4903 A000	0x4903 AFFF	4KB	Module
	0x4903 B000	0x4903 BFFF	4KB	L4 interconnect
GPTIMER7	0x4903 C000	0x4903 CFFF	4KB	Module
	0x4903 D000	0x4903 DFFF	4KB	L4 interconnect
GPTIMER8	0x4903 E000	0x4903 EFFF	4KB	Module
	0x4903 F000	0x4903 FFFF	4KB	L4 interconnect
Reserved	0x4904 0000	0x4907 FFFF	256KB	Reserved
DMEM (64KB)	0x4908 0000	0x4908 FFFF	64KB	Data memory
	0x4909 0000	0x4909 0FFF	4KB	L4 interconnect
Reserved	0x4909 1000	0x4909 FFFF	60KB	Reserved
CMEM (4KB)	0x490A 0000	0x490A FFFF	64KB	Coefficient memory
	0x490B 0000	0x490B 0FFF	4KB	L4 interconnect
Reserved	0x490B 1000	0x490B FFFF	60KB	Reserved
SMEM (32KB)	0x490C 0000	0x490C FFFF	64KB	Module
	0x490D 0000	0x490D 0FFF	4KB	L4 interconnect
Reserved	0x490D 1000	0x490F 0FFF	128KB	Reserved
AESS configuration	0x490F 1000	0x490F 1FFF	4KB	Module
	0x490F 2000	0x490F 2FFF	4KB	L4 interconnect
Reserved	0x490F 3000	0x490F FFFF	52KB	Reserved

Table 2-8 describes the mapping of the registers for the DSP subsystem.

**Table 2-8. ABE DSP Memory Space Mapping**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
L4 ABE	0x0000 0000	0x0000 3FFF	16KB	Boot space
Reserved	0x0000 4000	0x0002 1FFF	120KB	Reserved
McBSP1	0x0002 2000	0x0002 2FFF	4KB	Module
	0x0002 3000	0x0002 3FFF	4KB	L4 interconnect
McBSP2	0x0002 4000	0x0002 4FFF	4KB	Module
	0x0002 5000	0x0002 5FFF	4KB	L4 interconnect
McBSP3	0x0002 6000	0x0002 6FFF	4KB	Module
	0x0002 7000	0x0002 7FFF	4KB	L4 interconnect
McASP	0x0002 8000	0x0002 8FFF	4KB	McASP CFG port
	0x0002 9000	0x0002 9FFF	4KB	L4 interconnect
	0x0002 A000	0x0002 AFFF	4KB	McASP data port
	0x0002 B000	0x0002 BFFF	4KB	L4 interconnect
SLIMBUS1	0x0002 C000	0x0002 CFFF	4KB	Module
	0x0002 D000	0x0002 DFFF	4KB	L4 interconnect
DMIC	0x0002 E000	0x0002 EFFF	4KB	Module
	0x0002 F000	0x0002 FFFF	4KB	L4 interconnect
WDTIMER3	0x0003 0000	0x0003 0FFF	4KB	Module
	0x0003 1000	0x0003 1FFF	4KB	L4 interconnect
McPDM	0x0003 2000	0x0003 2FFF	4KB	Module
	0x0003 3000	0x0003 3FFF	4KB	L4 interconnect
Reserved	0x0003 4000	0x0003 7FFF	16KB	Reserved
GPTIMER5	0x0003 8000	0x0003 8FFF	4KB	Module
	0x0003 9000	0x0003 9FFF	4KB	L4 interconnect
GPTIMER6	0x0003 A000	0x0003 AFFF	4KB	Module
	0x0003 B000	0x0003 BFFF	4KB	L4 interconnect
GPTIMER7	0x0003 C000	0x0003 CFFF	4KB	Module
	0x0003 D000	0x0003 DFFF	4KB	L4 interconnect
GPTIMER8	0x0003 E000	0x0003 EFFF	4KB	Module
	0x0003 F000	0x0003 FFFF	4KB	L4 interconnect
Reserved	0x0004 0000	0x0007 FFFF	256KB	Reserved
DMEM (64KB)	0x0008 0000	0x0008 FFFF	64KB	Data memory
	0x0009 0000	0x0009 0FFF	4KB	L4 interconnect
Reserved	0x0009 1000	0x0009 FFFF	60KB	Reserved
CMEM (4KB)	0x000A 0000	0x000A FFFF	64KB	Coefficient memory
	0x000B 0000	0x000B 0FFF	4KB	L4 interconnect
Reserved	0x490B 1000	0x490B FFFF	60KB	Reserved
SMEM (32KB)	0x000C 0000	0x000C FFFF	64KB	Module
	0x000D 0000	0x000D 0FFF	4KB	L4 interconnect
Reserved	0x000D 1000	0x000F 0FFF	128KB	Reserved
AESS configuration	0x000F 1000	0x000F 1FFF	4KB	Module
	0x000F 2000	0x000F 2FFF	4KB	L4 interconnect
Reserved	0x000F 3000	0x000F FFFF	52KB	Reserved

## 2.4 Dual Cortex-M3 Subsystem Memory Space Mapping

Table 2-5 describes the mapping of the registers for the dual Cortex-M3.

**Table 2-9. Dual Cortex-M3 Memory Space Mapping**

Module Name	Start Address (hex)	End Address (hex)	Size
<b>Private Memory Map (can also be accessed by the slave port)</b>			
ROM (16KB)	0x5500 0000	0x5500 3FFF	16KB
RAM (64KB)	0x5502 0000	0x5502 FFFF	64KB
Shared cache/MMU configuration registers	0x5508 0000	0x5508 0FFF	4KB
WKUGEN masks	0x5508 1000	0x5508 1FFF	4KB
Reserved	0x5503 0000	0x5503 FFFF	64KB
<b>32-Bit OCP Port (master port to ISS)</b>			
ISP5	0x5504 0000	0x5505 FFFF	128KB
SIMCOP	0x5506 0000	0x5507 FFFF	128KB
Reserved	0x5508 2000	0x55FF FFFF	16MB – 384KB
<b>Bit-Band Mapping (supported at the dual Cortex-M3 interface)</b>			
Bit band region 1	0x2000 0000	0x200F FFFF	1MB
Bit band alias 1	0x2200 0000	0x23FF FFFF	32MB
Bit band region 2	0x4000 0000	0x400F FFFF	1MB
Bit band alias 2	0x4200 0000	0x43FF FFFF	32MB
<b>Private Peripheral Map (accessible only from the dual Cortex-M3)</b>			
Reserved	0xE000 0000	0xE000 0FFF	4KB
DWT	0xE000 1000	0xE000 1FFF	4KB
FPB	0xE000 2000	0xE000 2FFF	4KB
Reserved	0xE000 3000	0xE000 DFFF	44KB
NVIC	0xE000 E000	0xE000 EFFF	4KB
Reserved	0xE000 F000	0xE004 1FFF	204KB
External private peripheral bus	0xE0042000	0xE00F EFFF	756KB
ROM table (registers)	0xE00F F000	0xE00F FFFF	4KB
<b>64-Bit Interconnect Master Port (to L3)</b>			
Boot space <sup>(1)</sup>	0x0000 0000	0x0000 3FFF	–
L3	0x0000 0000	0x5FFF FFFF	1.5GB – 1MB
Tiler	0x6000 0000	0x7FFF FFFF	512MB

<sup>(1)</sup> At reset, the MMU is loaded with page 0, which forces the L2 ROM or L2 RAM to be at address 0x0. The reset page also includes the L2 ROM, L2 RAM, shared cache MMU register, and WUGEN registers.

## 2.5 DSP Subsystem Memory Space Mapping

The DSP subsystem can access the ABE and IVA-HD blocks.

ABE has three different address spaces:

- **ABE NC:** This region is for accesses to registers in ABE. For more information about ABE peripheral memory mapping, see [Table 2-6](#).
- **ABE:** This region is for accesses to memory in ABE. Memory in the ABE subsystem can be cached in DSP L1 and L2. When enhanced direct memory access (EDMA) tries to access this space, L1/L2 is snooped and if data is present in DSP subsystem caches, data is provided to EDMA from DSP caches. However, if data is not present in DSP caches, the shared cache sends the access to ABE and data returned is not allocated into DSP cache; instead, it is sent to EDMA.
- **ABE locked region:** This region is similar to the ABE region, except that data from ABE is allocated into DSP caches and returned to EDMA.

The IVA-HD region is for accesses to IVA-HD accelerator memories and registers in IVA-HD. SL2 is the L2 cache in IVA-HD. There are three ways to access the IVA-HD accelerator memories:

- **SL2 -> L1 region:** This region is for accesses to SL2 memory that can be mapped to DSP L1. When EDMA tries to access this space, L1/L2 is snooped and if data is present in DSP caches, the data is provided to EDMA from DSP caches. However, if data is not present in DSP caches, the shared cache sends the access to SL2 and data returned is not allocated into DSP L1 cache; instead, it is sent to EDMA.
- **SL2 -> L2 region:** This region is for accesses to SL2 memory that can be mapped to DSP L2. When EDMA tries to access this space, L1/L2 is snooped and if data is present in DSP caches, the data is provided to EDMA from DSP caches. However, if data is not present in DSP caches, the shared cache sends the access to SL2 and data returned is not allocated into DSP L2 cache; instead, it is sent to EDMA.
- **SL2 locked region:** This region is similar to the previous regions, except that data is allocated into DSP caches and returned to EDMA.

[Table 2-10](#) describes the mapping of the registers for the DSP subsystem.

**Table 2-10. DSP Subsystem Memory Space Mapping**

Region Name	Start Address (hex)	End Address (hex)	Size
DSP EDM registers	0x01BC 0000	0x01BC 0FFF	4KB
3PCC_CONFIG	0x01C0 0000	0x01C0 FFFF	64KB
3PTC0_CONFIG	0x01C1 0000	0x01C1 03FF	1KB
3PTC1_CONFIG	0x01C1 0400	0x01C1 07FF	1KB
SYSC_CONFIG	0x01C2 0000	0x01C2 0FFF	4KB
WUGEN	0x01C2 1000	0x01C2 1FFF	4KB
L1 configuration registers	0x01C3 0000	0x01C3 00FF	256B
L2 configuration registers	0x01C3 0200	0x01C3 02FF	256B
Shared cache debug	0x01C3 0400	0x01C3 05FF	512B
Attr MMU	0x01C3 0800	0x01C3 0FFF	2KB
DSP CPU configuration registers	0x01C4 0000	0x01C4 0FFF	4KB
Interrupt controller	0x01C5 0000	0x01C5 FFFF	64KB
DSP unique ID	0x01C6 1000	0x01C6 1FFF	4KB
DSP revision ID	0x01C6 2000	0x01C6 2FFF	4KB
ABE NC region	0x01D0 0000	0x01DF FFFF	1MB
IVA-HD (accelerator memories, NC region)	0x01E0 0000	0x01EF FFFF	1MB
Interconnect registers	0x01F0 0000	0x01FF FFFF	1MB
SL2 -> L1	0x1080 0000	0x1081 FFFF	128KB
SL2 -> L2	0x1082 0000	0x1083 FFFF	128KB
SL2 locked region -> L2/ L1 locked	0x1090 0000	0x1093 FFFF	256KB
ABE	0x10D0 0000	0x10DF FFFF	1MB

**Table 2-10. DSP Subsystem Memory Space Mapping (continued)**

Region Name	Start Address (hex)	End Address (hex)	Size
ABE locked region	0x10E0 0000	0x10EF FFFF	1MB
External SOC memories and peripherals	0x2000 0000	0xFFFF FFFF	3584MB



## 2.6 Display Subsystem Memory Space Mapping

The display subsystem integrates a display controller (DISPC), a remote frame buffer interface (RFBI), two MIPI® display serial interfaces (DSI1 with four data lanes, and DSI2 with two data lanes), an HDMI link with its PHY (high-definition multimedia interface, digital part), and a video PAL/NTSC encoder (VENC) with video DAC (and TV out buffer).

The DISPC is connected through an interconnect master port to the L3 interconnect and has its own DMAs to fetch the data from the system memory.

There are two views (from the L3\_PER and L4\_PER interconnects) of the display subsystem memory space mapping.

### 2.6.1 L3 Interconnect View of the Display Memory Space

[Table 2-11](#) lists the display subsystem memory space mapping from the perspective of the Cortex-A9 MPU subsystem through the L3 interconnect.

**Table 2-11. L3 Access – Display Subsystem Space Mapping**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
Display subsystem registers	0x5800 0000	0x5800 0FFF	4KB	Display subsystem and DSI registers
DISPC	0x5800 1000	0x5800 1FFF	4KB	Display controller
RFBI	0x5800 2000	0x5800 2FFF	4KB	Remote frame buffer interface
VENC	0x5800 3000	0x5800 3FFF	4KB	Video encoder
DSI1	0x5800 4000	0x5800 4FFF	4KB	Display serial interface 1
DSI2	0x5800 5000	0x5800 5FFF	4KB	Display serial interface 2
HDMI	0x5800 6000	0x5800 6FFF	4KB	HDMI configuration registers
HDCP	0x5800 7000	0x5800 7FFF	4KB	HDCP registers
Reserved	0x5800 8000	0x58FF FFFF	16MB	Reserved

### 2.6.2 L4 Interconnect View of the Display Memory Space

[Table 2-12](#) lists the display subsystem memory space mapping internally from the perspective of the Cortex-A9 MPU subsystem through the L4 interconnect.

**Table 2-12. L4\_PER Access – Display Subsystem Space Mapping**

Module Name	Start Address (hex)	End Address (hex)	Size	Description
Display subsystem registers	0x4804 0000	0x4804 0FFF	4KB	Display subsystem and DSI registers
DISPC	0x4804 1000	0x4804 1FFF	4KB	Display controller
RFBI	0x4804 2000	0x4804 2FFF	4KB	Remote frame buffer interface
VENC	0x4804 3000	0x4804 3FFF	4KB	Video encoder
DSI1	0x4804 4000	0x4804 4FFF	4KB	Display serial interface 1
DSI2	0x4804 5000	0x4804 5FFF	4KB	Display serial interface 2
HDMI	0x4804 6000	0x4804 6FFF	4KB	HDMI configuration registers
HDCP	0x4804 7000	0x4804 7FFF	4KB	HDCP registers
Reserved	0x4804 8000	0x4804 8FFF	32KB	Reserved

PRELIMINARY

## Power, Reset and Clock Management

This chapter describes the power, reset and clock management in the device.

Topic	Page
3.1 Device Power Management Introduction .....	292
3.2 PRCM Subsystem Overview .....	320
3.3 PRCM Subsystem Environment .....	322
3.4 PRCM Subsystem Integration .....	325
3.5 Reset Management Functional Description .....	329
3.6 Clock Management Functional Description .....	361
3.7 Power Management Functional Description .....	457
3.8 Voltage Management Functional Description .....	480
3.9 Device Low-Power States .....	498
3.10 PRCM Module Programming Guide .....	507
3.11 PRCM Register Manual .....	519
3.12 SCRM Register Manual .....	994
3.13 SR Register Manual .....	1018

## 3.1 Device Power Management Introduction

Power management (efficient use of the limited battery resources on a mobile device) is one of the most important design aspects of any mobile system. It imposes strong control over limited available power resources to ensure they function for the longest possible length of time.

The device power-management architecture ensures maximum performance and operation time for user satisfaction (audio/video support) while offering versatile power-management techniques for maximum design flexibility, depending on application requirements.

This introduction contains the following information:

- Power-management architecture building blocks for the device
- State-of-the-art power-management techniques supported by the power-management architecture of the device

### 3.1.1 Device Power-Management Architecture Building Blocks

To provide a versatile architecture supporting multiple power-management techniques, the power-management framework is built with three levels of resource management: clock, power, and voltage management.

These management levels are enforced by defining the managed entities or building blocks of the power-management architecture, called the clock, power, and voltage domains.

A domain is a group of modules or subsections of the device that share a common entity (for example, common clock source, common voltage source, or a common power switch). The group forming the domain is managed by a policy manager. For example, a clock for a clock domain is managed by a dedicated clock manager within the power, reset, and clock management (PRCM) module. The clock manager takes into consideration the joint clocking constraints of all the modules belonging to that clock domain (and, hence, receiving that clock).

---

**NOTE:** In the following sections, the term *module* is used to represent the device IPs (that is, modules or subsystems), other than the PRCM module, that receive clock, reset, or power signals from the PRCM module.

---

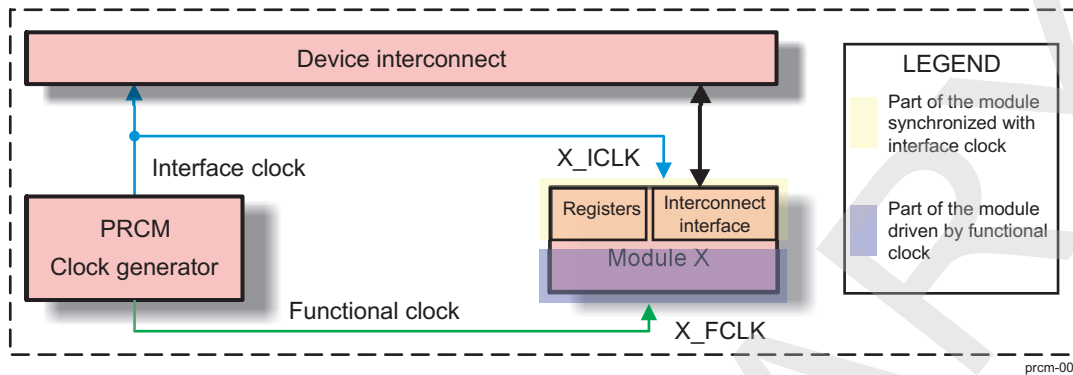
#### 3.1.1.1 Clock Management

The PRCM module manages the gating (that is, switching off) and enabling of the clocks to the device modules. The clocks are managed based on the requirement constraints of the associated modules. The following sections identify the module clock characteristics, management policy, clock domains, and clock domain management.

##### 3.1.1.1.1 Module Interface and Functional Clocks

Each module within the device has specific clock input characteristics requirements. Based on the characteristics of the clocks delivered to the modules, the clocks are divided into two categories: interface clocks and functional clocks (see [Figure 3-1](#)).

Figure 3-1. Functional and Interface Clocks



The interface clocks have the following characteristics:

- They ensure proper communication between any module/subsystem and the interconnect.
- In most cases, they supply the system interconnect interface and registers of the module.
- A typical module has one interface clock, but modules with multiple interface clocks may also exist (that is, when connected to multiple interconnect buses).
- Interface clock management is done at the device level.
- From the standpoint of the PRCM module, an interface clock is identified with an `_ICLK` suffix.

Functional clocks have the following characteristics:

- They supply the functional part of a module or a subsystem.
- A module can have one or more functional clocks. Some functional clocks are mandatory, while others are optional for its functioning. A module needs its mandatory clock(s) to be operational. The optional clocks are used for specific features and can be shut down without stopping the module activity (for example, the clock for the camera).
- From the standpoint of the PRCM module, a functional clock is directly distributed to the related modules through a dedicated clock tree. It is identified with an `_FCLK` suffix.

**NOTE:** At the module level, the interface clocks are always fed by the interface clock outputs of the PRCM module. The functional clocks are fed either by a PRCM module functional clock output, or by a PRCM module interface clock output. In the latter case, the functional and interface module clocks inherit the clock-management features (autoidle features) of the PRCM module interface clock.

### 3.1.1.1.2 Module-Level Clock Management

Each module in the device may also have specific clock requirements. Certain module clocks must be *active* when operating in specific modes, or may be gated otherwise. Globally, the activation and gating of the module clocks are managed by the PRCM module. Hence, the PRCM module must be aware of when to activate and when to gate the module clocks.

The PRCM module differentiates the clock-management behavior for device modules based on whether the module can initiate transactions on the device interconnect (called master module) or it cannot initiate transactions and only responds to the transactions initiated by the master (called slave module). Thus, two hardware-based power-management protocols are used:

- Master standby protocol: Clock-management protocol between the PRCM and master modules
- Slave idle protocol: Clock-management protocol between the PRCM and slave modules

#### Master standby protocol:

This protocol is used to indicate that a master module must initiate a transaction on the device interconnect and requests specific (functional and interface) clocks for the purpose. The PRCM module ensures that the required clocks are active when the master module requests the PRCM module to enable them. This is called a module wake-up transition and the module is said to be functional after this transition completes.

Similarly, when the master module no longer requires the clocks, it informs the PRCM module, and the PRCM module can then gate the clocks to the module. The master module is then said to be in standby mode.

Although the protocol is completely hardware-controlled, the software must configure the clock management behavior for the module. This is done by setting the module register bit field **<Module>\_SYSCONFIG.MIDDLEMODE** or **<Module>\_SYSCONFIG.STANDBYMODE**, as presented in [Table 3-1](#). The behavior, identified by standby mode values, must be configured.

**Table 3-1. Master Module Standby Mode Settings**

Standby Mode Value	Selected Mode	Description
0x0	Force-standby	The module unconditionally asserts the standby request to the PRCM module, regardless of its internal operations. The PRCM module may gate the functional and interface clocks to the module. This mode must be used carefully because it does not prevent any loss of data at the time the clocks are gated.
0x1	No-standby	The module never asserts the standby request to the PRCM module. This mode is safe from a module point of view as it ensures that the clocks remain active; however, it is not efficient from a power-saving perspective because it never allows the PRCM module output clocks to be gated.
0x2	Smart-standby	The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idled. The PRCM module can then gate the clocks to the module.
0x3	Smart-standby wakeup-capable mode	The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idle. The PRCM module can then gate the clocks to the module. The module may generate (master-related) wake-up events when in standby state. The mode is relevant only if the appropriate module mwakeup output is implemented.

**NOTE:**

- Smart-standby mode is the preferred mode of operation, while forced-standby and no-standby modes are intended for debugging purposes.
- A master module may support all or some of the standby modes listed in [Table 3-1](#). See the power management section in the module chapter to identify the supported standby mode.

The standby status of a master module is indicated by the **CM\_<Power domain>\_<Module>\_CLKCTRL[x].STBYST** bit in the PRCM module. [Table 3-2](#) describes the master module standby status.

**Table 3-2. Master Module Standby Status**

STBYST Bit Value	Description
0x0	The module is functional.
0x1	The module is in standby mode.

[Table 3-3](#) lists the enabling conditions for the master module clocks managed by the standby protocol.

**Table 3-3. Master Module Clock Enabling Conditions**

Relation	Condition
AND	Clock domain is ready.
OR	Master module standby request is deasserted.

**Table 3-3. Master Module Clock Enabling Conditions (continued)**

Relation	Condition
	Master module wake-up request is asserted.

**Slave Idle protocol:**

This hardware protocol allows the PRCM module to control the state of a slave module. The PRCM module informs the slave module, through assertion of an idle request, when its clocks (interface and functional) can be gated. The slave can then acknowledge the request from the PRCM module and the PRCM module is then allowed to gate the clocks to the module. A slave module is said to be in IDLE state when its clocks are gated by the PRCM module.

Similarly, an idled slave module may need to be woken up because of a service request from a master module or as a result of an event (called a wake-up event, for example, interrupt or DMA request) received by the slave module. In this situation the PRCM module enables the clocks to the module and then deasserts the idle request to signal the module to wake up.

Although the protocol is completely hardware-controlled, software must configure the clock management behavior for the slave module. This is done by setting the module register bit field **<Module>\_SYSCONFIG.SIDLEMODE** or **<Module>\_SYSCONFIG.IDLEMODE**, as presented in [Table 3-4](#). The behavior, identified by the Idle Mode Value column, must be configured by software.

**Table 3-4. Module Idle Mode Settings**

Idle Mode Value	Selected Mode	Description
0x0	Force-idle	The module unconditionally acknowledges the idle request from the PRCM module, regardless of its internal operations. This mode must be used carefully because it does not prevent any loss of data at the time the clock is switched off.
0x1	No-idle	The module never acknowledges any idle request from the PRCM module. This mode is safe from a module point of view as it ensures the clocks remain active; however it is not efficient from a power-saving perspective because it does not allow the PRCM module output clock to be shut off, and thus the power domain to be set to a lower power state.
0x2	Smart-idle	The module acknowledges the idle request basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or direct memory access (DMA) requests are processed. This is the best approach to an efficient system power management.
0x3	Smart-idle wakeup-capable mode	The module acknowledges the idle request basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or DMA requests are processed. This is the best approach to efficient system power management. The module may generate (IRQ- or DMA-request-related) wake-up events when in idle state. The mode is relevant only if the appropriate module wakeup output(s) is implemented.

**NOTE:**

- Smart-idle mode is the preferred mode of operation, while forced-idle and no-idle modes are intended for debugging purposes.
- A slave module may support all or some of the idle modes listed in [Table 3-4](#). See the power-management section in the module chapter to identify the supported idle modes.

The idle status of a slave module is indicated by the **CM\_<Power domain>\_<Module>\_CLKCTRL[x] IDLEST** bit field in the PRCM module. [Table 3-5](#) lists the possible idle statuses for a slave module.

**Table 3-5. Slave Module Idle Status**

IDLEST Bit Value	Idle Status	Description
0x0	Functional	The module is fully functional. The interface and functional clocks are active.
0x1	In transition	The module is performing a wake-up or a sleep transition.
0x2	Interface idle	The module interface clock is idled. The module may remain functional if using a separate functional clock.

**Table 3-5. Slave Module Idle Status (continued)**

IDLEST Bit Value	Idle Status	Description
0x3	Full idle	The module is fully idle. the interface and functional clocks are gated.

When configured in smart-idle mode, the slave module may acknowledge the idle request of the PRCM module based on the activity of its interface and/or functional clocks. To define which module clocks (that is, interface and/or functional) should be considered when responding to the PRCM module request, the software must configure the **<Module>\_SYSCONFIG[x] CLOCKACTIVITY** bit field.

The CLOCKACTIVITY setting is used internally by the module to determine on which part of the module the conditions to acknowledge the PRCM module idle request is tested. As an example, if the functional clock must remain active when the module is in idle mode, then the module must acknowledge PRCM module idle request by only considering the interface clock gating conditions (that is, there is no pending activity on the interconnect).

**NOTE:** See the power-management section in the module chapter to identify whether this feature is configurable.

Using the CLOCKACTIVITY setting along with the smart-idle mode ensures that the clock remains active for the module features that must remain available during the module idle mode. [Table 3-6](#) describes the possible CLOCKACTIVITY settings for a module:

**Table 3-6. Slave Module Clock Activity Settings**

CLOCKACTIVITY Bit Value	Module Interface Clock	Module Functional Clock	Description
0x0	Gated	Gated	The interface and functional clocks are considered when generating the acknowledgement. This setting also means both clocks may be gated upon a PRCM module idle request.
0x1	Active	Gated	The interface clock is not shut down and is not considered when generating the acknowledgement to the PRCM module idle request. Only the functional clock is considered.
0x2	Gated	Active	The functional clock is not shut down and is not considered when generating the acknowledgement to the PRCM module idle request. Only the interface clock is considered.
0x3	Active	Active	The interface and functional clocks are not shut down. The module can acknowledge the idle request without checking the internal functionalities linked to its clocks.

(1) Optional if it concerns all features.

**NOTE:**

- The software configuration of the CLOCKACTIVITY settings may not be available for a given module. For some modules, the CLOCKACTIVITY settings can be hardwired.
- A slave module may support all or some of the CLOCKACTIVITY settings listed in [Table 3-6](#).

See the power-management section in the specific module chapter to identify the supported idle feature and settings.

**CAUTION**

The PRCM module does not have any hardware means to read the CLOCKACTIVITY settings of the module. Software must ensure a consistent programming between the CLOCKACTIVITY settings of the module and the PRCM module clock gating control bits. The PRCM module must be configured (where software control is available) not to gate the module clock, which should remain active according to the CLOCKACTIVITY settings of the module.



For the idle protocol management on the PRCM module side, the behavior of PRCM module is configured in the CM\_<Power domain>\_<module>\_CLKCTRL[x] MODULEMODE bit field. Based on the configured behavior, the PRCM module asserts the idle request to the module either unconditionally (that is, immediately when the software requests), or through hardware control when the module idle conditions are satisfied. Table 3-7 describes the configurable behavior of MODULEMODE.

**Table 3-7. Slave Module Mode Settings in PRCM**

MODULEMODE Bit Value	Selected Mode	Description
0x0	Disabled	The PRCM module unconditionally asserts the module idle request. This request applies to the gating of the functional and interface clocks to the module. If acknowledged by the module, the PRCM module can gate all clocks to the module (that is, the module is completely disabled). It can only react to an asynchronous wake-up event (that is, a wake-up event that does not require the module functional clock to be active).
0x1	Auto	This mode applies to a module when the PRCM module only manages its interface clock and not its functional clock. The PRCM module automatically asserts/deasserts the module idle request based on the clock-domain transitions. If acknowledged by the module, the PRCM module can gate the interface clock to the module.
0x2	Enabled	This mode applies to a module when the PRCM module manages its interface and functional clocks. The functional clock to the module remains active unconditionally, while the PRCM module automatically asserts/deasserts the module idle request based on the clock-domain transitions. If acknowledged by the module, the PRCM module can only gate the interface clock to the module.
0x3	Reserved	Not available.

**NOTE:** The PRCM module may support all or some of the MODULEMODE module settings listed in Table 3-7. See the CM\_<Power domain>\_<module>\_CLKCTRL[x] MODULEMODE bit field description for the module to identify the supported settings.

Table 3-8 and Table 3-9 list the enabling conditions for the slave module clocks managed by the Idle protocol.

**Table 3-8. Slave Module Interface Clock Enabling Conditions**

Relation	Condition
AND	Clock domain is ready.
	OR
	Slave module idle status is 0x0 (fully functional).
	Slave module idle status is 0x1 (in transition).
	Slave module wake-up request is asserted.

**Table 3-9. Slave Module Functional Clock Enabling Conditions**

Relation	Condition
AND	Clock domain is ready.
	OR
	Slave module idle status is 0x0 (fully functional).
	Slave module idle status is 0x1 (in transition).
	Slave module idle status is 0x2 (interface clock is idled).
	Slave module wake-up request is asserted.

The module clock domain must be ready for the optional clocks to the module, and any associated clock-enable control is asserted.

**NOTE:** A given clock can be used by more than one module. Clock enabling conditions are then ORed together (that is, the clock is provided as soon as one of the enabling conditions is true). As a consequence, the clock is disabled only when all related enabling conditions are false.

### Module wake-up request:

In IDLE state, a slave module may have to wake-up to generate an interrupt or a DMA request. This may be the result of an external request (for example, to the I/O port of a general-purpose input/output [GPIO] module) or an internally generated event (for example, WDTIMER time-up). The slave module, with wake-up capability, sends a wake-up request to the PRCM module. The PRCM module then activates the module clocks and acknowledges the module wake-up request.

In IDLE state, some slave modules may require functional clock(s) to generate a wake-up event. Such requests are called synchronous wake-up events on the PRCM module side, while the events generated when the functional or interface module clocks are gated are called asynchronous wake-up events.

**NOTE:** See the power-management section in the module chapter to identify whether its wake-up event is synchronous or asynchronous.

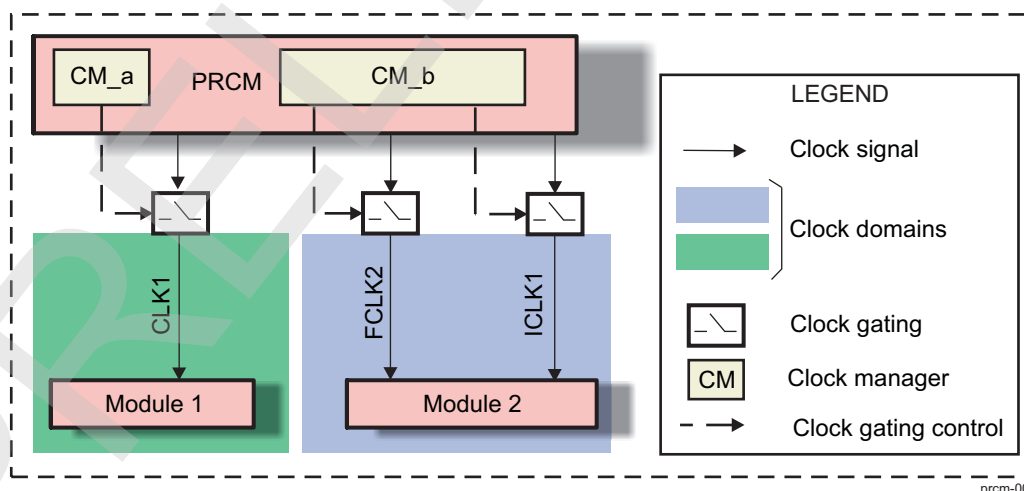
The standby and idle clock-management protocols allow the configuration of the module-level clock-management interaction between the PRCM module and individual modules of the device. However, the PRCM module may not necessarily gate the clock to the module immediately after the module switches to standby or idle mode at the end of this interaction. This is because the same clock can be shared by other modules that are active and need this shared clock to complete their activity. As a result, the PRCM module provides a second level of clock management called the clock domain level, as explained in [Section 3.1.1.1.3](#).

#### 3.1.1.1.3 Clock Domain

A clock domain is a group of modules fed by clock signals controlled by the same clock manager in the PRCM module (see [Figure 3-2](#)). By gating the clocks in a clock domain, the clocks to all the modules belonging to that clock domain can be cut to lower their active power consumption (that is, the device is on and the clocks to the modules are dynamically switched to ACTIVE or INACTIVE (Gated) states). Thus, a clock domain allows control of the dynamic power consumption of the device.

The device is partitioned into multiple clock domains and each clock domain is controlled by an associated clock manager within the PRCM module. This allows the PRCM module to individually activate and gate each clock domain of the device.

**Figure 3-2. Generic Clock Domain**



[Figure 3-2](#) presents an example of two clock managers: CM\_a and CM\_b. Each clock manager manages a clock domain. The clock domain of CM\_b is composed of two clocks: a functional clock (FCLK2) and an interface clock (ICLK1), while that of CM\_a consists of a clock (CLK1) that is used by the module as functional and interface clock. The clocks to Module 2 can be gated independently of the clock to Module 1, thus ensuring power savings when Module 2 is not in use.

The PRCM module lets software check the status of the clock domain functional clocks. The CM\_<Clock domain>\_CLKSTCTRL[x] CLKACTIVITY\_<FCLK/Clock name\_FCLK> bit in the PRCM module identifies the state of the functional clock(s) within the clock domain. Table 3-10 lists the two possible states of the functional clock.

**Table 3-10. Clock Domain Functional Clock States**

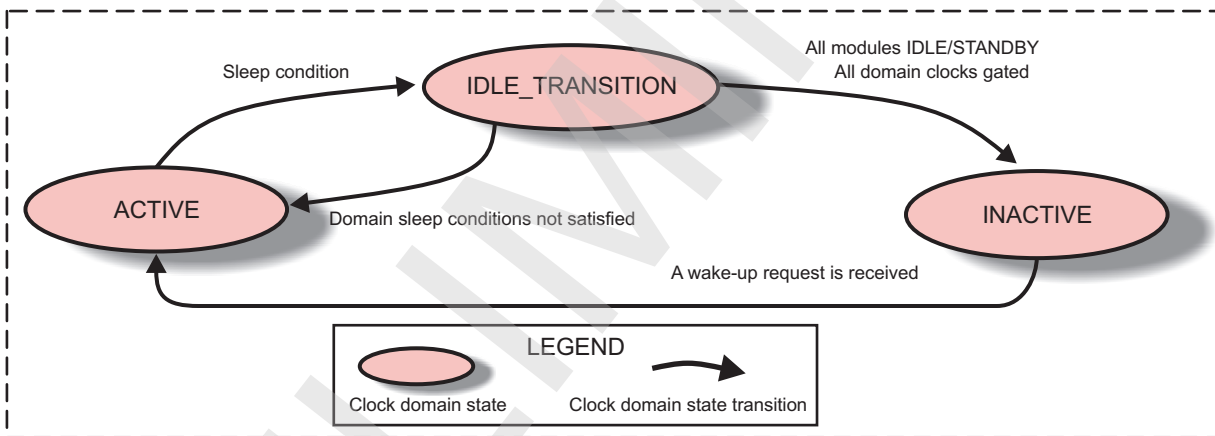
CLKACTIVITY Bit Value	Status	Description
0x0	Gated	The functional clock of the clock domain is inactive.
0x1	Active	The functional clock of the clock domain is running.

**3.1.1.1.4 Clock Domain-Level Clock Management**

The domain clock manager can automatically (that is, based on hardware conditions) and jointly manage the interface clocks within the clock domain. The functional clocks within the clock domain are managed through software settings.

A clock domain can switch between three possible states: ACTIVE, IDLE\_TRANSITION, or INACTIVE. Figure 3-3 identifies the sleep and wake-up transitions of the clock domain between the ACTIVE and INACTIVE states.

**Figure 3-3. Clock Domain State Transitions**



prcm-003

Table 3-11 defines the clock domain states.

**Table 3-11. Clock Domain Clock States**

State	Description
ACTIVE	<ul style="list-style-type: none"> <li>Every nondisabled slave module (that is, those whose MODULEMODE value is not set to disabled) is put out of IDLE state.</li> <li>All functional clocks to the active slave modules (that is, not idled) of the clock domain are provided.</li> <li>All interface clocks to the nondisabled slave modules in the clock domain are provided.</li> <li>All functional and interface clocks to the active master modules (that is, not in STANDBY state) in the clock domain are provided.</li> <li>Every enabled optional clocks to the modules in the clock domain are provided.</li> </ul>
IDLE_TRANSITION	<p>This is a transitory state.</p> <ul style="list-style-type: none"> <li>Every master module in the clock domain is in STANDBY state.</li> <li>Every idle request to all the slave modules in the clock domain is asserted.</li> <li>The functional clocks to the slave module in enabled state (that is, those whose MODULEMODE values are set to enabled) remain active.</li> <li>Every enabled optional clocks to the modules in the clock domain are provided.</li> </ul>

**Table 3-11. Clock Domain Clock States (continued)**

State	Description
INACTIVE	<p>All clocks within the clock domain are gated.</p> <ul style="list-style-type: none"> <li>• Every slave module in the clock domain (that is, those whose MODULEMODE is set to disabled or auto) is in IDLE state and set to disabled or auto mode.</li> <li>• Every master module in the clock domain is in STANDBY state.</li> <li>• Every optional functional clock in the clock domain is gated.</li> </ul>

Each clock domain transition behavior is managed by an associated register bit field in the CM\_<Clock domain>\_CLKSTCTRL[x] CLKTRCTRL PRCM module.

[Table 3-12](#) describes the clock domain clock transition mode settings.

**Table 3-12. Clock Domain Clock Transition Mode Settings**

CLKTRCTRL Bit Value	Selected Mode	Description
0x0	NO_SLEEP	A clock domain sleep transition is never initiated, irrespective of the hardware conditions.
0x1	SW_SLEEP	A software-forced sleep transition. The transition is initiated when the associated hardware conditions are satisfied (see <a href="#">Table 3-14</a> ).
0x2	SW_WKUP	A software-forced clock domain wake-up transition is initiated, irrespective of the hardware conditions identified in <a href="#">Table 3-13</a> .
0x3	HW_AUTO	Hardware-controlled automatic sleep and wake-up transition is initiated by the PRCM module when the associated hardware conditions are satisfied (see <a href="#">Table 3-14</a> and <a href="#">Table 3-13</a> ).

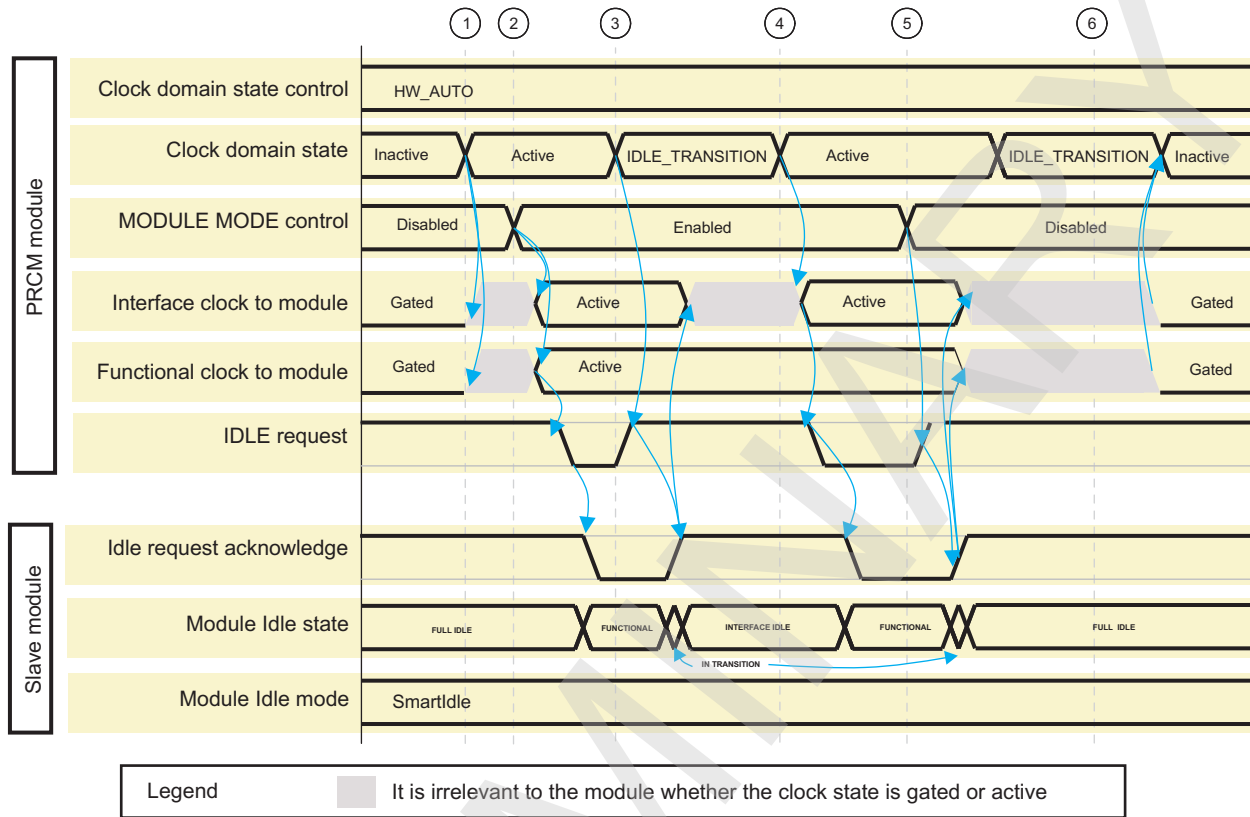
**NOTE:** Depending on the characteristics of a clock domain, it may or may not support all the clock transition mode settings described in [Table 3-12](#). See the clock domain clock management section of the specific clock domain to identify the supported clock transition mode settings.

### 3.1.1.1.5 Clock Domain HW\_AUTO Mode Sequences

The sequence diagrams in [Figure 3-4](#) through [Figure 3-6](#) identify the PRCM module hardware-controlled enabling and gating of the functional and interface clocks to the module. They highlight the changes in the module state based on the changes to the clock domain state and module mode settings.

[Figure 3-4](#) highlights the behavior of a slave module receiving the interface and functional clocks and having two configurable module modes: Disabled and Enabled.

**Figure 3-4. Clock Domain/Slave Module Clock-Management Interaction Sequence 1**

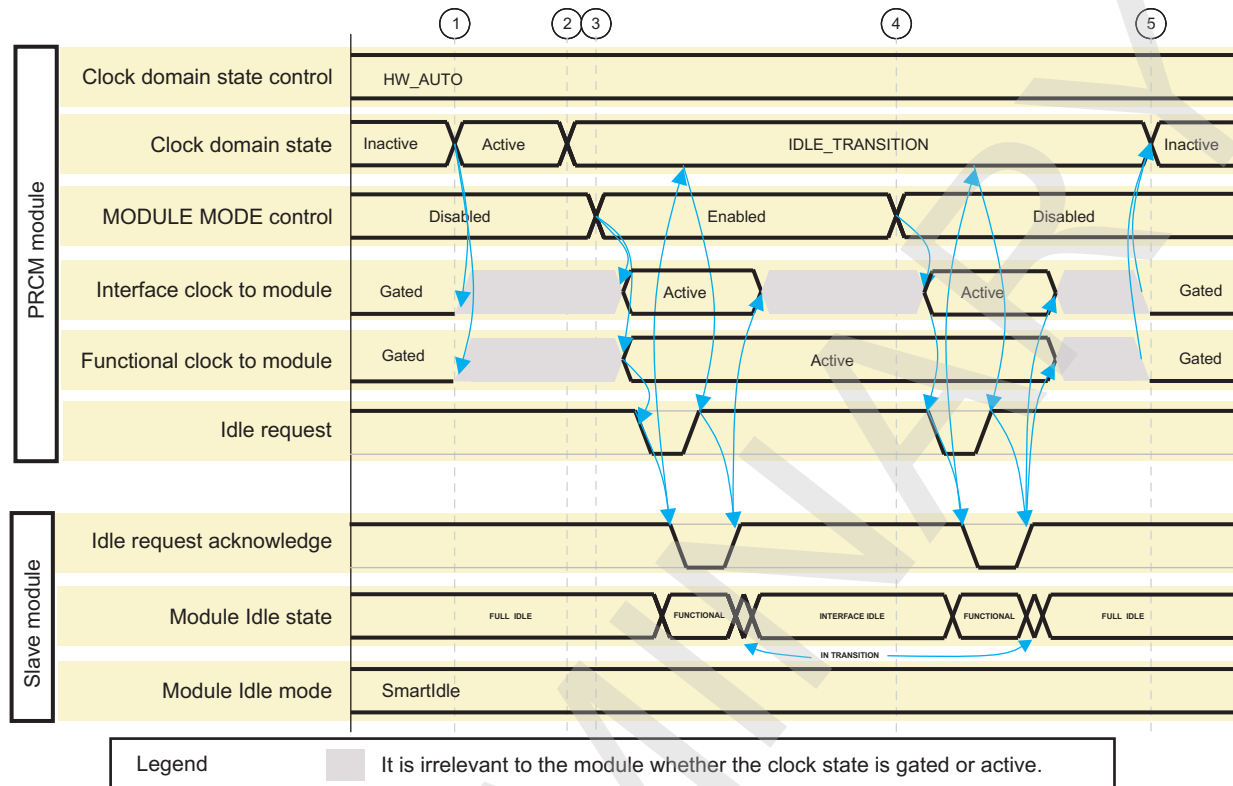


prcm-004

Initially, the clock domain, which includes the slave module, is inactive. The module is in FULL-IDLE state and its functional and interface clocks are gated.

1. The clock domain wakes up and changes state to Active. As the Module Mode control is still Disabled, this event has no effect on the module state. The functional and interface clocks may still be restarted automatically, based on the requirements of other modules sharing these clocks.
2. Software changes the module mode to Enabled. The clocks to the module are restarted automatically. The PRCM module then deasserts a hardware idle request signal to the module. The module sends an idle request acknowledge to the PRCM module. The module is now effectively awake. The module Idle state is functional. If the module is enabled before the clock domain wakes up, Steps 1 and 2 are simultaneous.
3. The clock domain switches to IDLE\_TRANSITION state. In turn, the PRCM module requests the module to go into IDLE state by asserting the idle request signal. When acknowledged, the clock to the module can be gated, depending on other modules sharing the same clock. The functional clock of the module remains enabled as the module is in Enabled mode.
4. The clock domain does not have all conditions to complete the sleep transition, and wakes up again. In turn, the interface clock to the module is automatically restarted, if applicable, then the module is put out of IDLE state.
5. Software disables the module. The PRCM module requests the module to go to IDLE state by asserting the idle request signal. When acknowledged, the clock to the module can be gated, depending on other modules sharing the same clock.
6. The clock domain switches to the IDLE\_TRANSITION state. When the clock domain sleep transition conditions are satisfied, the clocks (functional and interface) are gated. The clock domain then switches to INACTIVE state. This has no effect on the module, which is already in FULL-IDLE state.

Figure 3-5 highlights the behavior of the same slave module, receiving the interface and functional clocks, when the module mode is changed while the clock domain state is Idle\_Transition.

**Figure 3-5. Clock Domain/Slave Module Clock-Management Interaction Sequence 2**

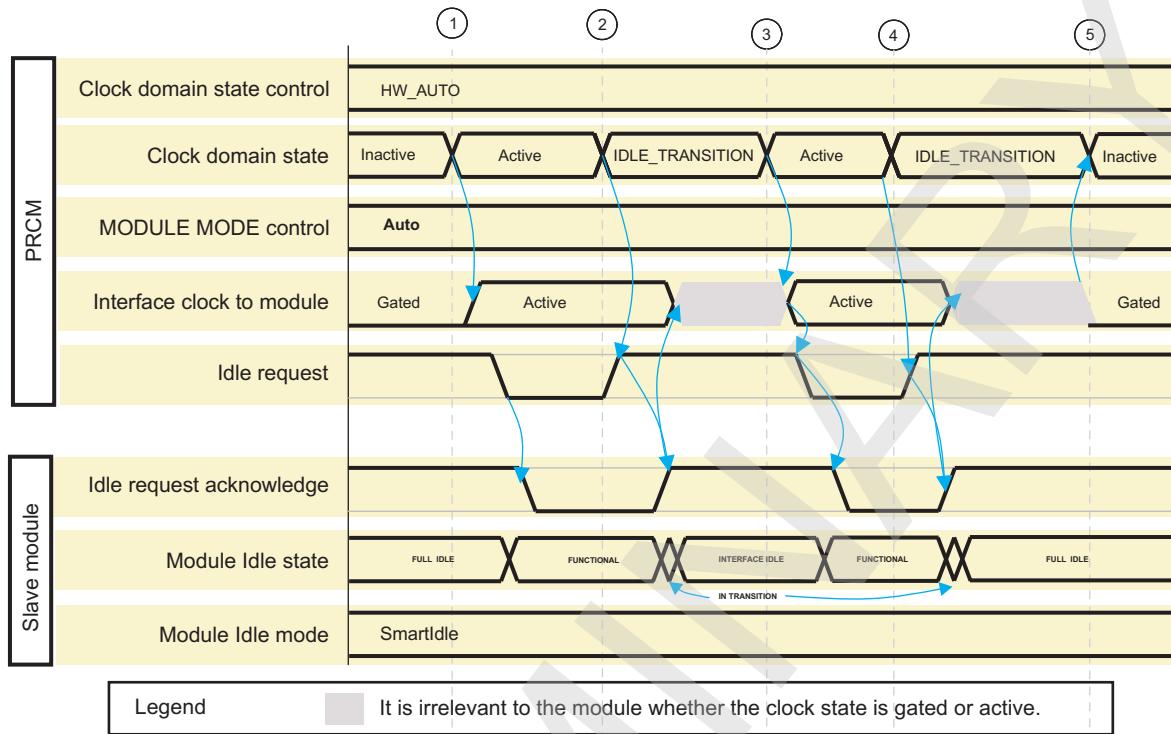
prcm-005

Initially, the clock domain, which includes the slave module, is inactive. The module is in FULL-IDLE state and its functional and interface clocks are gated.

1. The clock domain wakes up and changes state to Active. As the Module Mode control is Disabled, this event has no effect on the module state. The functional and interface clocks may still be restarted automatically, based on the requirements of other modules sharing these clocks.
2. The clock domain goes into the IDLE\_TRANSITION state. As the Module Mode control is Disabled, this event has no effect on the module state.
3. Software changes the module mode to Enabled. The clocks to the module are restarted automatically and then the module is put out of IDLE state. As soon as acknowledged, the module is requested to go back to IDLE state with gating of the interface clock only (that is, the INTERFACE IDLE state). The interface clock to the module can be gated, depending on other modules sharing the same clock.
4. Software disables the module. The interface clock to the module is restarted automatically. The PRCM module requests the module to go to IDLE state by asserting the idle request signal. When acknowledged, the interface and the functional clocks to the module can be gated, depending on other modules sharing the same clock.
5. When the clock domain sleep transition conditions are satisfied and the functional and interface clocks are gated, the clock domain switches to INACTIVE state. This has no effect on the module, which is already in FULL-IDLE state.

Figure 3-6 highlights the behavior of a slave module receiving only interface clock and supporting the configurable Auto module mode.

**Figure 3-6. Clock Domain/Slave Module Clock-Management Interaction Sequence 3**



prcm-006

Initially, the clock domain, which includes the slave module, is inactive. The module is in FULL-IDLE state and its interface clock is gated.

1. The clock domain wakes up and changes state to Active. In turn, the interface clock to the module is automatically restarted. The PRCM module then deasserts a hardware idle request signal to the module. The module sends an idle request acknowledge to the PRCM module. The module is now effectively awake, that is, the module Idle state is functional.
2. The clock domain switches to the IDLE\_TRANSITION state. The PRCM module requests the module to go to IDLE state by asserting the idle request signal. When acknowledged, the interface clock to the module can be gated, depending on other modules sharing the same clock.
3. The clock domain does not have all conditions to complete the sleep transition, and wakes up again. In turn, the interface clock to the module is automatically restarted if applicable, then the module is put out of IDLE state.
4. Same as Step 2.
5. The clock domain has all conditions to complete the sleep transition. The module is under IDLE state and its clock is gated.

### 3.1.1.1.6 Clock Domain Sleep/Wakeup

The clock domain manager initiates a domain wake-up transition when the conditions listed in [Table 3-13](#) are satisfied.

**Table 3-13. Clock Domain Wake-Up Conditions**

Relation	Condition
OR	The SW_WKUP clock transition mode for the clock domain is set. (CLKTRCTRL = 0x2)
	At least one wake-up request asserted by one of the modules of the clock domain.
	At least one dynamic dependency <sup>(1)</sup> from another clock domain is active.
	At least one static dependency <sup>(1)</sup> from another clock domain is active.
	At least one wake-up dependency <sup>(1)</sup> from a module in another clock domain is active.

(1) The dynamic, static dependency and the wake-up dependency are explained in [Section 3.1.1.1.7](#), *Clock Domain Dependency*.

The clock domain manager initiates a domain sleep transition when the conditions listed in [Table 3-14](#) are satisfied.

**Table 3-14. Clock Domain Sleep Conditions**

Relation		Condition
AND		All master modules in the clock domain are in STANDBY state.
		No wake-up request is asserted by any module of the clock domain.
		No dynamic domain dependency <sup>(1)</sup> from any other domain is active.
		No wake-up dependency <sup>(1)</sup> from any module in another domain is active.
OR		The SW_SLEEP clock transition mode is set for the clock domain (CLKTRCTRL = 0x1).
	AND	The HW_AUTO clock transition mode is set for the clock domain (CLKTRCTRL = 0x3).
		No static domain dependency <sup>(1)</sup> from any other domain is active

(1) The dynamic, static dependency and the wake-up dependency are explained in [Section 3.1.1.1.7](#), *Clock Domain Dependency*.

### 3.1.1.1.7 Clock Domain Dependency

A domain dependency is a binary relationship between two clock domains. A clock domain A is said to depend on a clock domain B when a module in clock domain B provides services to a module in clock domain A. As a result, clock domain B must be active when clock domain A is active so that the module in clock domain B is accessible by the module in clock domain A.

The dependency between two clock domains may also exist if one clock domain serves to ensure communication between two modules (for example, the clock domain of the device interconnect).

Thus, a clock domain may support the types of clock domain dependencies described in the following subsections.

#### 3.1.1.1.7.1 Static Dependency

If clock domain A has a master module that can access a slave module in clock domain B, then clock domain A can have a static dependency with clock domain B. Similarly, a static dependency can also exist between domain A and B if domain B conveys the transactions from domain A module toward a module in any other domain. For example, CD\_DSP can have a static dependency with CD\_L3\_1 because this domain has the L3 interconnect to carry the transactions from the DSP module.

This static dependency consists of forcing clock domain B to stay active, as long as there is at least one master module of clock domain A that is not in STANDBY state. If clock domains A and B are initially in GATED state, then clock domain B becomes active as soon as clock domain A becomes active when a wake-up request from the master module is received by the PRCM module.

Similarly, as a result of the static dependency, clock domain B can only be gated if all the master modules of clock domain A that can access the slave modules in clock domain B are in STANDBY state.

The static dependency between a source clock domain and a destination clock domain is configured in the PRCM module by setting CM\_<Source Clock domain>\_STATICDEP[x] <Destination Clock domain>\_STATDEP bit. As a result, the source clock domain forces the destination clock domain to become active and stay active as long as the source clock domain is active.

#### 3.1.1.1.7.2 Dynamic Dependency

When clock domains A and B contain modules directly linked to a common device interconnect, these clock domains can have a dynamic dependency.



A dynamic dependency consists of forcing clock domain B to stay active as long as a module from clock domain A is communicating with the module in clock domain B through the interconnect. Clock domain B becomes active as soon as the communication is initiated. This is automatically managed by the PRCM module by monitoring the communication on the interconnect between the modules of the two clock domains.

Similarly, the inverse condition of this dependency can be stated: Clock domain B can be inactive only if all modules between clock domains A and B are quiet for a given sampling delay, identified as a sliding window duration on the interconnect activity status.

The size of the sliding window is based on the number of cycles of a prescaled L4 clock whose frequency is configured by setting the `CM_DYN_DEP_PRESCAL[5:0]` PRESCAL bit field. The prescaled clock frequency is given as:

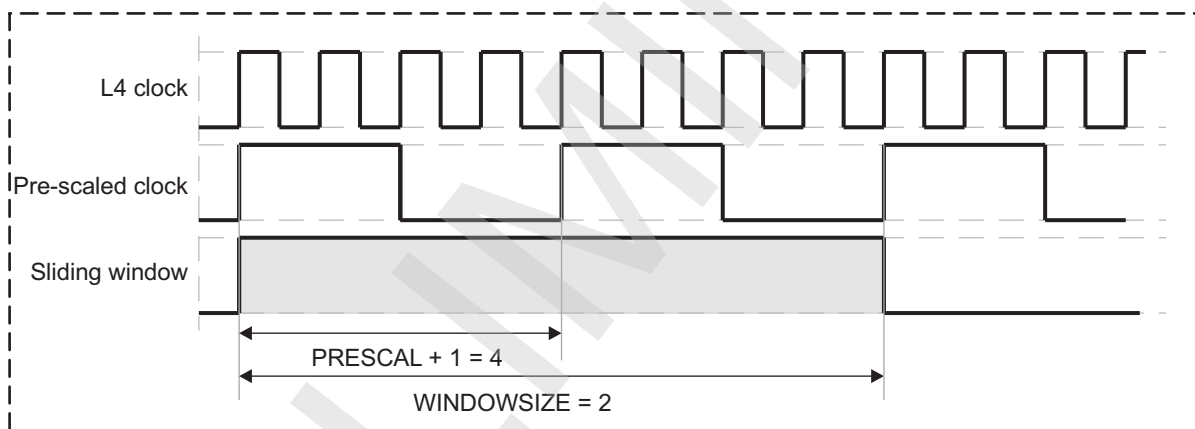
$$\text{Prescaled clock frequency} = \text{L4 interface clock frequency} / (\text{PRESCAL} + 1)$$

The size of the sliding window is fixed by setting the `CM_<Clock domain>_DYNAMICDEP[27:24]` WINDOWSIZE bit field. It is given as:

$$\text{Sliding window duration} = \text{WINDOWSIZE} \times \text{Period of Prescaled clock cycle}$$

Figure 3-7 presents an example of the sliding window duration equal to eight clock cycles of L4 clock when PRESCAL is set to 3 and WINDOWSIZE is set to 2.

**Figure 3-7. Sliding window for Dynamic Dependency**



prcm-007

This dynamic dependency is also referred to as the autosleep/autowake-up feature.

**NOTE:**

- The static dependency between two clock domains can be configured by software (PRCM module registers) or hardwired in the PRCM module.
- The dynamic dependency between two clock domains is hardwired in the PRCM module.

A dynamic dependency is said to be active when:

- At least one master module or a slave interconnect module from clock domain A to domain B is active (that is, the master module is not in STANDBY state and/or the interconnect slave module is not in IDLE state), and
- There has been one or more transaction on the interconnect within the sliding window duration.

It is said to be inactive otherwise.

The dynamic dependency between a source clock domain and a destination clock domain can be read in the PRCM module from the corresponding read-only `CM_<Source Clock domain>_DYNAMICDEP[x]` <Destination Clock domain>\_DYNDEP bit.

### 3.1.1.1.7.3 Wake-Up Dependency

A wake-up dependency is a dependency between the clock domain of a module that owns one or several wake-up signals, toward the clock domain of another module needed to service the associated wake-up event. As a result of this dependency, the wake-up event to a module not only activates its clock domain, but also the clock domain of the servicing module.

---

**NOTE:** The wake-up signal that triggers a wake-up dependency stays active as long as the source of the event is not serviced, to ensure that clock domain of the servicing module remains active.

---

Wake-up dependencies allow acceleration of the wake-up transition of multiple domains needed to service the wake-up event by initiating their transition in parallel. The static and dynamic dependencies can allow the wake-up of related domains but the complete wake-up transition of all the associated domains is slower because of the sequential cascading of their wake-up transitions.

In the device, the source event of the wake-up signal to a slave module can be one of following types:

- Interrupt request to microprocessor (MPU) or digital signal processor (DSP) interrupt controller (INTC)
- DMA request to a DMA controller

Upon wakeup by these types of wake-up events, and for as long as they remain asserted, the PRCM module takes the following actions:

- The power domain of the servicing module (for example, MPU, DSP, or DMA) is forced to POWER ON state and the clock domain becomes active.
- The power domain of the device interconnect between servicing module and the module originator of the wake-up event is forced to POWER ON state and the clock domain becomes active.
- The power domain of the slave module originator of the wake-up event is forced to POWER ON state and the clock domain becomes active.
- The slave module originator of the wake-up event is switched from IDLE to ACTIVE state.

On assertion of a wake-up event of a standalone master module, and as long as it remains asserted, the PRCM module takes the following action:

- The power domain of the master module originator of the wake-up event is forced to POWER ON state and the clock domain becomes active

---

**NOTE:** For slave modules, the static and dynamic dependencies of a clock domain are not impacted by its wake-up dependency settings. For master modules, the static dependencies are not impacted.

Hence, in addition to the activation of the clock domains identified in [Section 3.1.1.1.7.3](#) above, all clock domains associated by static dependencies are also activated.

However, the clock domains associated with the waking up clock domain through dynamic dependencies are only activated if a transaction is initiated to these clock domains.

---

For each wake-up signal coming from a slave module, the type of the corresponding event can be configured in the PM\_<Power domain>\_<Originator Module>\_WKDEP[x] WKUPDEP\_<Originator Module>\_<Servicing Module> bit of the PRCM module. Where <Power domain> is the name of the power domain of the originator module of the wake-up event identified as <Originator Module>. <Servicing Module> refers to the module servicing the wake-up event.

---

**NOTE:** When only one event type is associated with the wake-up signal of a slave module, the WKUPDEP wake-up dependency for the module clock domain is not be configurable and may be hardwired in the PRCM module.

In case of the master modules, there is no configurable wake-up dependency. Their power domain is switched on and their clock domain is activated by the PRCM module when they assert their wake-up signal.

---

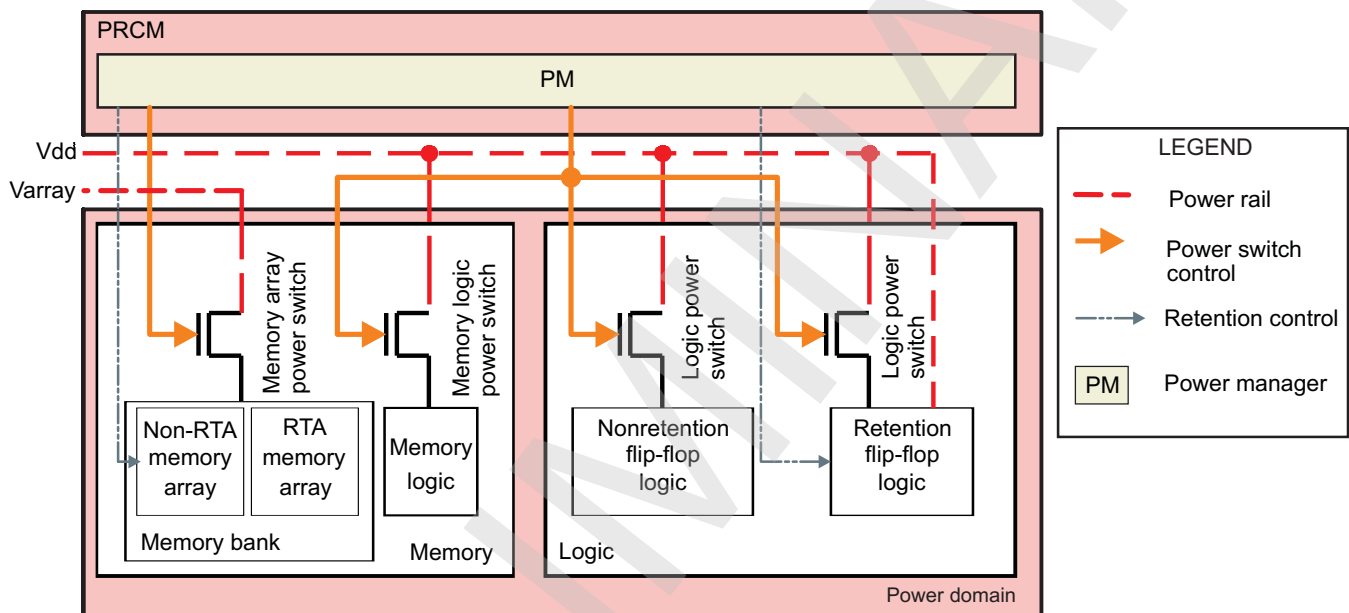
### 3.1.1.2 Power Management

The PRCM module manages the switching on and off of the power supply to the device modules. The power to the modules can be switched off when they are not in use to minimize device power consumption. Independent power control of sections of the device allow the PRCM module to turn on and off specific sections of the device without affecting the others.

#### 3.1.1.2.1 Power Domain

A power domain is a section (that is, a group of modules) of the device with independent and dedicated power manager (see Figure 3-8). A power domain can be turned on and off without affecting the other parts of the device.

Figure 3-8. Generic Power Domain



prcm-008

To minimize device power consumption, the modules are grouped into power domains. A power domain can be split into some logic area and some memory area.

The memory area contains two entities:

- Memory bank: The memory bank is composed of memory arrays. It is powered by a dedicated voltage rail and an associated power switch (for example, Varray and memory array power switches). The memory array within the memory bank can be of RTA (retention-till-access), see Section 3.1.1.2.3, *Module Logic and Memory Context* or nonRTA type.
- Memory logic: The memory logic is powered by the same voltage source as the logic area of the power domain, but has its dedicated power switch (for example, Vdd and memory logic power switches).

The logic area in the power domain can also be split between retention flip-flops (RFFs) or nonretention flip-flops (DFFs).

Table 3-15 lists the possible states and substates of the logic area in a power domain.

Table 3-15. States of a Logic Area in a Power Domain

State	Substate	Description
ON	ON-ACTIVE	Logic is fully powered and at least one enclosed clock domain is active.
	ON-INACTIVE	Logic is fully powered and all enclosed clock domains are idled.
RETENTIO N	CSWR (close switch retention)	Logic is fully powered and all enclosed clock domains are idled.

**Table 3-15. States of a Logic Area in a Power Domain (continued)**

State	Substate	Description
	OSWR (open switch retention)	Logic is powered down, except the RFF-based logic retains its value. The DFF is lost and reset. Vdd can be lowered to its retention voltage level if all power domains associated are in this or lower states.
OFF		Logic power switches are off. All the logic (DFF and RFF) is lost except for the context, which has been saved in the scratchpad memory of an always-on power domain. Vdd can be set to 0 V if all associated power domains are in this state.

**NOTE:** A power domain that does not have RFF logic may still support the open-switch retention power state. In this case, all its logic context is lost, yet its memory area context may remain valid if programmed to retention mode (see [Table 3-16](#) and [Table 3-19](#)).

The RETENTION state is useful for quick switching to low-power idle mode (in which the domain clocks are gated and the domain voltage is less than the on-voltage level) without losing the context, and then quickly switching back to ON-ACTIVE state when necessary. In RETENTION state power consumption is less than in On Power state.

Similarly, in RETENTION state the CSWR state has the advantage of faster turn-on times over the OSWR state, because in the latter case the entire context is maintained. However, CSWR state has the disadvantage of higher leakage currents over OSWR state.

The behavior of the memory array power switch and memory logic power switch can be selected through software settings in the PRCM module, or can be hardwired. Once the behavior is selected, the PRCM module hardware automatically handles these elements to ensure correct power transition sequencing between the power domain states.

Software can also initiate the memory array power state changes when the associated power domain is in On Power state. This allows the memory array to be turned off and on as needed.

The memory area can be configured to any of the power states identified in [Table 3-16](#).

**Table 3-16. States of a Memory Area in a Power Domain**

State	Description
ON	The memory array is powered and fully functional. Additionally, the RTA feature may put part or all of the memory array automatically into retention when not accessed, if this feature exists and is enabled on a given memory instance.
RETENTION	The memory array is fully powered, but memory is not accessible. The array can be put into retention either through applicable direct retention control signal, or automatically if the RTA feature exists and is enabled on a given memory instance. Data in memory are always retained.
OFF	The memory array is powered down. Data in memory are lost.

### 3.1.1.2.2 Module Logic and Memory Context

In case of a power state transition in the logic area or the memory area, the context of the module may no longer be valid. This can also be the case when the domain resets are asserted by the device. A specific RM\_<Clock Domain Name>\_<Module Name>\_CONTEXT register provides the status of the device logic and memory context.

- The module logic context is made of simple flip-flops (DFFs) if module has no logic RFFs.
- If module has logic retention (full or partial), it is assumed that the context is made only of RFFs.

**NOTE:** The display subsystem (DSS) is an exception where the status of DFF and RFF context is presented, because only HMDI keys are retained, while most of the DSS is nonretained.

These context status bits must be cleared by software.

### 3.1.1.2.3 Retention-Till-Access Memory Feature

Most high-density memories within the device support the RTA feature. It consists of automatically putting part of or the entire memory array into retention when no access is made to the corresponding locations. This is managed within the memory bank by hardware and is completely transparent to the user software. As a consequence, RTA memories do not have any retention mode control signal coming from the PRCM module.

The RTA feature is controlled (enabled and disabled) from:

- eFuse bits configuration in the system control module, automatically loaded to the PRCM module.
- Software-controlled override capability through a bit field per voltage domain in the PRCM module (that is, PRM\_LDO\_SRAM\_<Memory Voltage Domain>\_SETUP[0] DISABLE\_RT\_A\_EXPORT).
- Hardware override capability controlled by the PRCM module through control of embedded static random-access memory (SRAM) low-dropout linear regulator (LDO) controller.

### 3.1.1.2.4 Power Domain Management

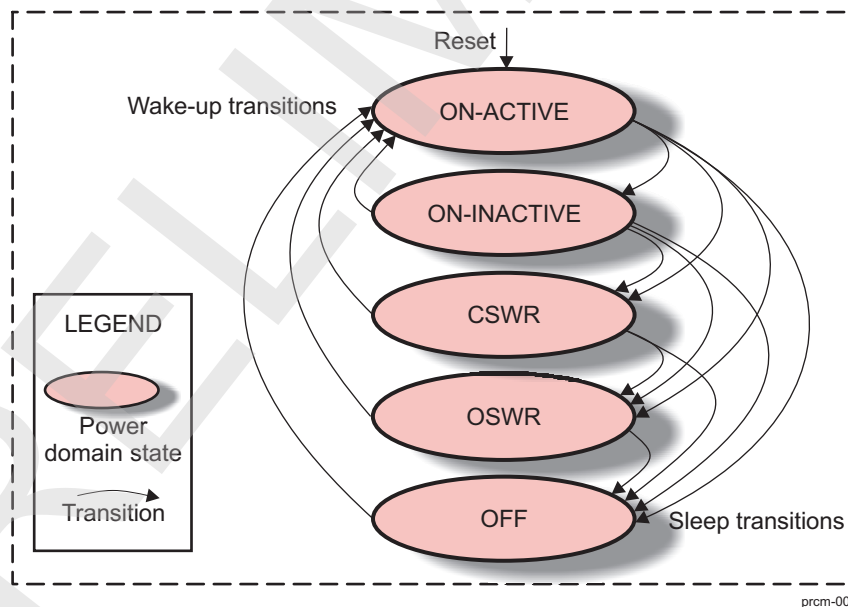
The power manager associated with each power domain is assigned the task of managing the domain power transitions. It ensures that all hardware conditions are satisfied before it can initiate a power domain transition from a source to a target power state (for example, from ON-ACTIVE state to OSWR RETENTION state).

The hardware condition for power domain transition from ON-ACTIVE to any other transition state is:

- All clock domain managers in the power domain are in INACTIVE state: All modules in the domain are in IDLE or STANDBY state, and there is no constraining dependency (static, dynamic, or wake-up dependency) between the clock domains blocking the clock domain sleep transition.

Figure 3-9 shows all possible power domain state transitions.

Figure 3-9. Power Domain Transitions



Successive power-down transitions can be performed by lowering the power state from ON-ACTIVE to ON-INACTIVE to RETENTION, and then to OFF, as long as the hardware condition is satisfied.

However, the power domain wake-up transition from any low-power state (ON-INACTIVE, CSWR, OSWR, or OFF) to ON-ACTIVE is always direct. For example, user software cannot request a power domain transition from OSWR state to CSWR state, but only to ON-ACTIVE state.

The power domain manager initiates a power domain wake-up transition when the conditions listed in Table 3-17 are satisfied.

**Table 3-17. Power Domain Wake-Up Conditions**

Relation	Condition
AND	Voltage domain is on.
	OR
	There is at least a wake-up condition for one enclosed functional clock domain.
	There is a request for clock generation or distribution enclosed in the power domain.
	There is a PRCM module service request (applicable only to power domains, including PRCM module logic).

The power domain manager initiates a domain sleep transition when the conditions listed in [Table 3-18](#) are satisfied.

**Table 3-18. Power Domain Sleep Conditions**

Relation	Condition
AND	All functional clock domains enclosed in the power domain are idled.
	All clock generation or distribution enclosed in the power domain is quiet, and corresponding input clocks are gated. For example, DPLL, if present, must be in stop mode.
	There is no PRCM module service request (applicable only to power domains, including PRCM module logic).

[Table 3-19](#) lists the PRCM module power domain control and status features.

**Table 3-19. Power Domain Control and Status Registers**

Register/Bit Field	Type	Description
PM_<Power domain>_PWRSTCTRL[1:0] POWERSTATE	Control	Selects the target power state of the power domain among OFF, ON-ACTIVE, ON-INACTIVE, and RETENTION.
PM_<Power domain>_PWRSTCTRL[x] LOWPOWERSTATECHANGE	Control	Power state change request when domain has already performed a sleep transition. Allows going into deeper LOW-POWER state without waking up the power domain.
PM_<Power domain>_PWRSTCTRL[2] LOGICRETSTATE	Control	Selects whether the Power domain logic is in CSWR or OSWR RETENTION state when the domain transitions to RETENTION state.
PM_<Power domain>_PWRSTCTRL[x] <memory bank>_RETSTATE	Control	Selects whether the memory bank in the power domain is in ON or RETENTION state when the power domain is in RETENTION state. The memory bank can not be in ON state when the power domain is in RETENTION state.
PM_<Power domain>_PWRSTCTRL[x] <memory bank>_ONSTATE	Control	Selects whether the memory bank is in ON, RETENTION, or OFF state when the power domain is in ON state.
PM_<Power domain>_PWRSTST[1:0] POWERSTATEST	Status	Identifies the current state of the power domain. It can be OFF, RETENTION, ON-INACTIVE, or ON-ACTIVE.
PM_<Power domain>_PWRSTST[2] LOGICSTATEST	Status	Identifies the current state of the logic area in the power domain. It can be OFF or ON.
PM_<Power domain>_PWRSTST[20] INTRANSITION	Status	Identifies whether a power domain power state transition is in progress or there is no ongoing transition.
PM_<Power domain>_PWRSTST[x] <memory bank>_STATEST	Status	Identifies the current power state of the memory bank in the power domain. It can be OFF, RETENTION, or ON. For memory with RTA feature, the ON and RETENTION memory states are equivalent.

### 3.1.1.3 Voltage Management

The PRCM module controls the voltage scaling (that is, switching the voltage in discrete steps or in a continuum within a range of possible values) of the power sources of the device. This allows control of the device power consumption according to the performance criteria defined. Higher performance is ensured with higher voltage and clock frequencies (and hence higher power consumption), while lower performance can be supported with lowered power consumption by reducing or completely gating the power supply to specific areas of the device and gating the associated clocks.

### 3.1.1.3.1 Voltage Domain

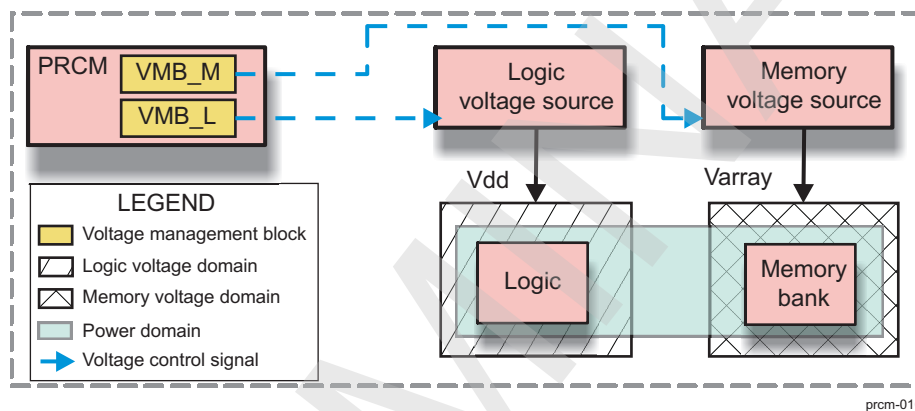
A voltage domain is a section of the device supplied by a dedicated voltage source (that is, an internal LDO or external switch mode power supply [SMPS]). A voltage domain may or may not be controlled by the PRCM module. When a voltage domain is controlled by the PRCM module, a dedicated voltage manager is associated. The voltage manager allows regulation of the voltage level of the source for the voltage domain, independently of other voltage domains of the device.

The voltage managers in the PRCM module can be of two types:

- Dynamically configurable by software to scale the domain voltage level to specific values within the device operational voltage range. This is called adaptive voltage scaling (AVS).
- Hardware controlled to automatically switch the domain voltage levels according to the associated hardware conditions.

Figure 3-10 shows a voltage domain.

Figure 3-10. Generic Voltage Domain



By partitioning the device into independent voltage domains, different operating voltages can be assigned to the different sections of the device (that is, group of modules or memory banks). The independent voltage control allows voltage scaling of device subsections to ensure that each module or memory bank operates at the optimized operating voltage level based on the application performance requirements. When all modules within a voltage domain are idle (that is, have no ongoing transaction), the domain voltage can be lowered to reduce power consumption and then switched back to normal operating voltage level when a wake-up event is received by one of the modules of the voltage domain. Similarly, when a memory bank is not in use, it may be switched to retention or off voltage levels to ensure power savings.

Table 3-20 describes the different states of the logic voltage domain.

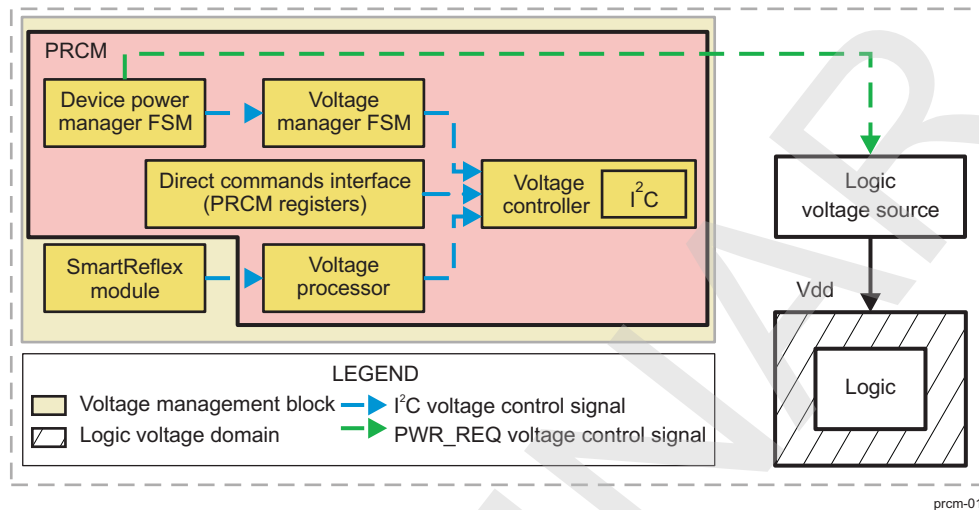
Table 3-20. States of Logic Voltage Domain

State	Description
ON	The voltage regulator source for the voltage domain is active and operating in its normal voltage range for the current operating performance point (OPP) and can deliver up to the maximum load current defined for the current OPP.
SLEEP	The voltage regulator source is operating in its normal voltage range but it can only deliver a limited load current (no logic activity on this voltage domain). In sleep mode, regulator power consumption is reduced.
RETENTION	The voltage regulator source is operating at retention voltage level. It delivers the minimal load current required to maintain the voltage domain logic context.
OFF	The voltage regulator source is inactive and its output voltage is 0 volts. No supply current is provided in this mode.

### 3.1.1.3.2 Voltage Domain Management

Figure 3-11 highlights the different voltage control paths available within a generic logic voltage management block to control the voltage supply to the device logic voltage domains.

**Figure 3-11. Generic Logic Voltage Management**



#### 3.1.1.3.2.1 Logic Voltage PWR\_REQ Voltage Control

A PWR\_REQ signal is deasserted when the device enters the OFF state. The external power integrated circuit (IC) can use it to properly manage turn-off or turn-on of the logic voltage source.

#### 3.1.1.3.2.2 Logic Voltage I<sup>2</sup>C Voltage Control

The PRCM module allows three voltage control paths to a dedicated inter-integrated circuit (I<sup>2</sup>C™) interface for the logic voltage domains managed by it.

1. The PRCM module hardware-managed voltage changes from device power manager voltage finite state-machine (FSM) through voltage manager FSM based on hardware conditions. It sends the voltage commands when the device enters in RETENTION or OFF state and upon device wakeup.
2. Direct control by user software through the PRCM module registers for voltage scaling.
3. Adaptive voltage scaling (see [Section 3.1.1.3.3, AVS Overview](#)) with SmartReflex™ voltage control commands through the voltage processor.

---

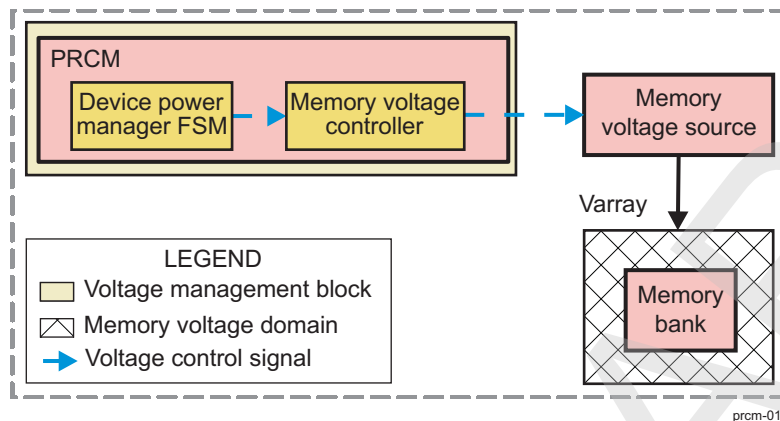
**NOTE:** An I<sup>2</sup>C host module (not controlled by the PRCM module) can be used to program the external power IC. Because this procedure completely bypasses PRCM module, see [Section 23.1 I<sup>2</sup>C](#).

---

Figure 3-12 highlights the voltage control path available within a generic memory voltage management block to control the voltage supply to the device memory voltage domains.



**Figure 3-12. Generic Memory Voltage Management**



The PRCM hardware supports automatic scaling down of the memory array supply whenever they transition to RETENTION or OFF power state. The Device power manager FSM manages the voltage scaling of memory voltage domains through memory voltage controller (or LDO).

### 3.1.1.3.3 AVS Overview

SmartReflex is a power-management technique used for controlling the operating voltage of a device to reduce its active power consumption.

With SmartReflex, the power-supply voltage is adapted to the silicon performance in two ways:

- Statically adapted to the manufacturing process of a given device
- Dynamically adapted to the temperature-induced current performance of the device

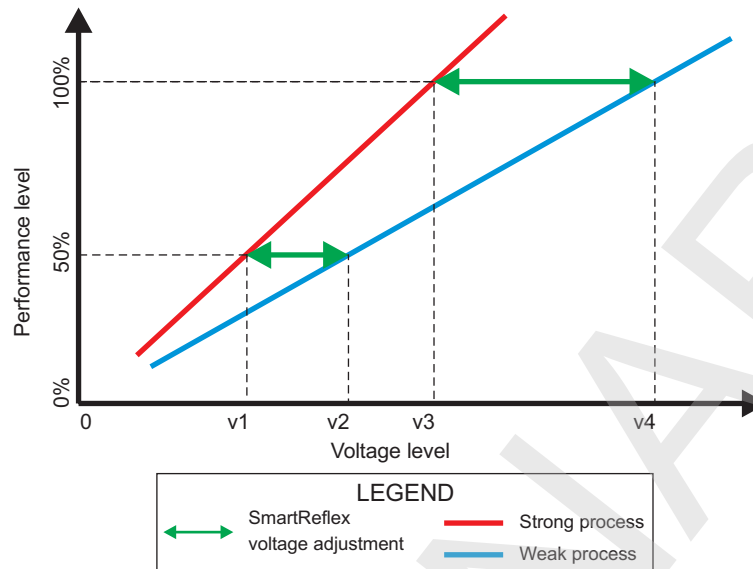
SmartReflex achieves optimal performance/power trade-off for all devices across the technology process spectrum and across temperature variations.

The static correction of the device voltage level (see [Figure 3-13](#)) is based on the desired performance level and silicon performance characteristics of the device. Due to process dispersion, each die has its specific silicon performance. The range of the process distribution defines the weak devices (low-performance silicon) and the strong devices (high-performance silicon).

A weak device is a device with the lowest performance tolerated for a process distribution; that is, at the typical voltage, the inherent maximum frequency is the lowest frequency of the chip distribution. Considered as the worst case, weak devices are used to constrain the target frequency of all the chips (OPP definition).

A strong device is a device with the highest performance tolerated for a process distribution. The inherent maximum frequency at the typical voltage is greater than the targeted frequency.

[Figure 3-13](#) identifies the fact that, with SmartReflex voltage-control architecture, it is possible to compensate for the device silicon characteristics and obtain optimal performance characteristics. Based on the device characteristics, the device voltage level can be adjusted for specific performance level.

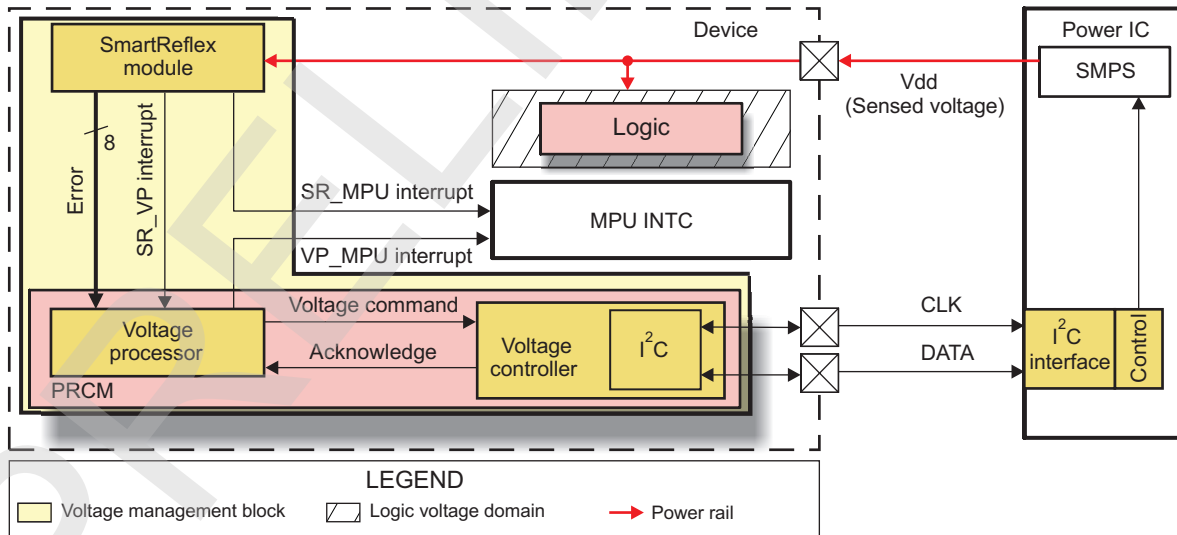
**Figure 3-13. SmartReflex Static Voltage Adjustment**

prcm-013

The dynamic correction of the device voltage is based on the real-time comparison of predefined performance points to the on-chip measured performance level. The device voltage level is raised or lowered to compensate for temperature-induced performance variations.

Without SmartReflex, the voltage variation is around the nominal voltage level and not controlled through any active dynamic compensation. This leads to power waste.

Figure 3-14 is a functional overview of the SmartReflex voltage-control architecture of the device connected to an external power IC.

**Figure 3-14. SmartReflex Voltage Control Functional Overview**

prcm-015

SmartReflex voltage control consists of the following modules:

- SmartReflex
- MPU INTC
- Voltage processor
- Voltage controller
- I<sup>2</sup>C interface

- SMPS

The SmartReflex module senses input voltage (V<sub>dd</sub>) and generates an error value that identifies the difference between the desired optimal voltage and the actual value of the SMPS. This error value is set in an internal register (for software read, if necessary) and is also passed to the voltage processor.

The voltage processor converts the error value to a voltage command that defines the change in the output voltage of the SMPS required to bring it to the desired voltage level.

The voltage command is sent to the voltage controller, which passes it to the power IC through the dedicated I<sup>2</sup>C interface. The power IC then adjusts the output voltage of the SMPS according to the command.

In this way, the SmartReflex module dynamically adjusts the SMPS voltage to compensate for voltage variations.

The device supports two operational modes for SmartReflex voltage control:

- Class-2 or software-controlled
- Class-3 or hardware-controlled

### 3.1.1.3.3.1 Class-2 SmartReflex Voltage Control

In software voltage control, the SmartReflex module interrupts the MPU when the voltage level crosses the defined voltage limits (minimum/maximum). The MPU reads the error value and determines the new voltage command to be sent to the SMPS to return the voltage to within the limits. The new voltage command is sent to the voltage controller, which passes the command to the SMPS and acknowledges its reception.

In class-2 control, the voltage processor is bypassed.

### 3.1.1.3.3.2 Class-3 SmartReflex Voltage Control

In hardware voltage control, the SmartReflex module interrupts the voltage processor when the voltage level crosses the defined voltage limits (minimum/maximum). The voltage processor reads the error value and determines the new voltage command to be sent to the SMPS to return the voltage to within the limits. The new voltage command is sent to the voltage controller, which passes the command to the SMPS and acknowledges its reception.

In class-3 mode, the software does not intervene in voltage control; the entire loop is handled by the hardware modules.

## 3.1.2 Power-Management Techniques

The following sections describe the state-of-the-art power-management techniques supported by the device.

---

**NOTE:** The values in [Figure 3-15](#) through [Figure 3-17](#), which show power-management techniques, are hypothetical. They do not represent valid test results on the device.

---

### 3.1.2.1 Standby Leakage Management

Standby leakage management (SLM) is a power-management technique that reduces standby power consumption by reducing power leakage.

With SLM, the device switches into low-power system modes automatically or in response to user requests during system standby (that is, in situations when no application is started and system activity is negligible or limited).

When applying SLM, the system remains in the lowest static power mode compatible with the system response time requirement.

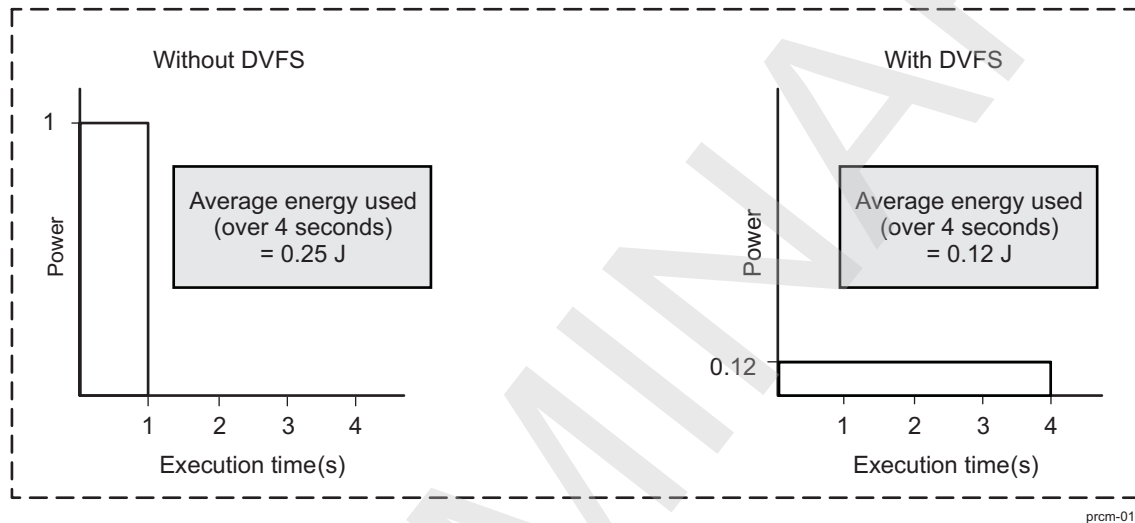
This technique trades static power consumption for wake-up latency.

### 3.1.2.2 Dynamic Voltage and Frequency Scaling

Dynamic voltage and frequency scaling (DVFS) consists of minimizing the idle time of the system. The DVFS technique uses dynamic selection of the optimal operating frequency and voltage to allow a task to be performed in the required length of time. This reduces the active power consumption (power consumed while executing a task) of the device while still meeting task requirements.

**NOTE:** The values in [Figure 3-15](#) are hypothetical. They are meant to clarify the concept only and do not represent valid test results on the device.

**Figure 3-15. Comparison of Energy Consumed With/Without DVFS**



[Figure 3-15](#) shows the DVFS technique by comparing a process executed at maximum frequency and operating voltage without applying DVFS to the same process executed at optimal frequency and voltage using DVFS, based on the task requirements. If a task that must terminate in 4 seconds is performed at maximum operating frequency (left side of the figure), it terminates in 1 second, and the remaining 3 seconds are spent in idle mode. With DVFS (right side of the figure), the operating frequency is reduced to an optimal level; the task takes the full 4 seconds to complete, but power consumption is reduced. In addition, the voltage can be reduced further to save power so the dynamic and leakage power consumptions are reduced.

DVFS requires control over the clock frequency and the operating voltage of the device elements. By intelligently switching the individual elements of the device to their OPPs, the power consumption of the device for a given task can be minimized.

For practical reasons related to device making (flow, tools), DVFS can be used only for a few discrete steps, not over a continuum of voltage and frequency values. Each step, or OPP, is composed of a voltage (V) and frequency (F) pair. For an OPP, the frequency corresponds to the maximum frequency allowed at a voltage, or reciprocally; the voltage corresponds to the minimum voltage allowed for a frequency.

When applying DVFS, a processor or system always runs at the lowest OPP that meets the performance requirement at a given time. The user determines the optimal OPP for a given task and then switches to that OPP to save power.

### 3.1.2.3 Dynamic Power Switching

Like DVFS, dynamic power switching (DPS) is a power-management technique aimed at reducing active power consumption of a device. However, whereas DVFS reduces the dynamic and leakage power consumptions, DPS reduces only leakage power consumption, at the expense of a slight overhead in dynamic power consumption.

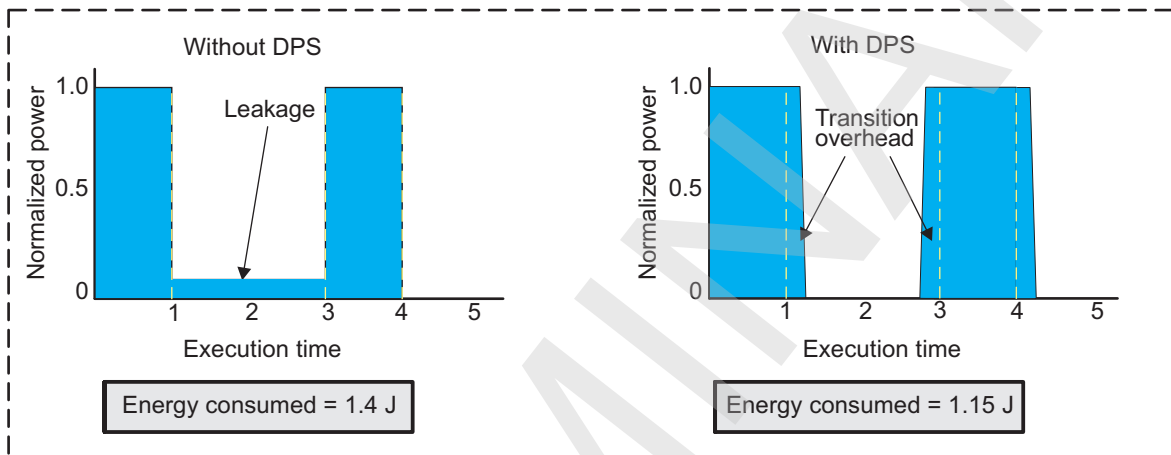
With DPS, the system switches dynamically between high- and low-consumption system power modes

during system active time. When DPS is applied, a processor or system runs at the highest OPP (maximum frequency and voltage) to complete its tasks quickly, followed by an automatic switch to a low-power mode, for minimum power consumption. DPS is useful when a real-time application is waiting for an event. The system can switch into a low-power system mode if the wake-up latency conditions allow it.

This technique consists of maximizing the idle period of the system to reduce its power consumption.

**NOTE:** The values in Figure 3-16 are hypothetical. They are meant to clarify the concept only and do not represent valid test results on the device.

**Figure 3-16. Comparison of Energy Consumed With/Without DPS**



prcm-017

Figure 3-16 compares the behavior of power consumption for the same device operation without DPS (left side of the figure) and with DPS (right side of the figure). When operating without DPS, the device has a constant leakage current in idle mode. By using DPS, the system reduces the leakage current to zero. However, the transitions between system power modes may require storing the information before entering a low-power inactive state and restoring the information after a wake-up event (see Figure 3-16). This results in additional dynamic power consumption, referred to as the transition overhead (see Figure 3-16). Transition overhead must be considered for a DPS operation.

For efficient deployment of DPS techniques, it is necessary to predict dynamically the performance requirement of the applications running on the processor. The DPS controller must account for the overhead of wake-up latencies related to domain switching and ensure that they do not significantly affect the performance of the device. Even with transition overhead, however, the user can identify an optimal idle-time limit, after which the DPS is useful for dynamic power-saving.

### 3.1.2.4 Adaptive Voltage Scaling

SmartReflex-based AVS is a power-management technique for automatic control of the operating voltages of the device to reduce active power consumption.

With SmartReflex, power-supply voltage is adapted to silicon performance, either statically (based on performance points predefined in the manufacturing process of a given device) or dynamically (based on the temperature-induced real-time performance of the device). A comparison of these predefined performance points to the real-time on-chip measured performance determines whether to raise or lower the power-supply voltage.

AVS achieves the optimal performance/power trade-off for all devices across the technology process spectrum and across temperature variation. The device voltage is automatically adapted to maintain performance of the device. This ensures optimal power consumption for a given OPP.

With AVS, the frequency steps are identified and the voltage is adapted according to the silicon performance of the device. In this case, instead of a voltage step for each frequency step, there is a corresponding range of voltages. The range depends on the fabrication process of the device and its real-time operating state (temperature) at a given frequency.

### 3.1.2.5 Adaptive Body Bias

The device implements transistor body bias techniques for two situations:

- Forward body-bias (FBB) for operating clock frequency boost for operation at higher OPPs.
- Reverse body-bias (RBB) for leakage current reduction for operation at lowest OPPs.

Adaptive body bias (ABB) is based on the process corner and the current OPP. This is configured in the efuse at the device characterization and is not continuously updated. A dedicated LDO, that is, VBBLDO, is used to produce the voltage bias.

### 3.1.2.6 Combining Power-Management Techniques

The power-management techniques previously described have specific features and are most effective when used under the specific operating conditions of the device. Hence, the best active power savings are obtained by combining the DVFS, DPS, SLM, and AVS techniques. For a given operating state, one or more of the power-saving techniques can be applied to ensure optimal operation with maximum power saving.

AVS can be used at boot time to adapt the voltage to device process characteristics (strong/weak) and then used continuously to compensate temperature variations. AVS can also ensure maximum available application performance of the device at a given OPP.

When medium application performance is required, or when application performance requirements vary, DVFS can be applied. The voltage and frequency can be scaled to match the closest OPP that meets the performance requirement.

When application performance requirements fall between two OPPs, or when a low application performance is required that is below the lowest performance OPP, DPS can be applied to switch to low-power mode.

When combining DVFS and DPS, the operating frequency must not be scaled to match the performance requirement without scaling the voltage. Lower operating frequency increases task completion time and reduces idle time. This prevents DPS or reduces its efficiency (DPS becomes more effective as idle time increases). Unless DPS cannot be applied for other reasons, for a given operating point of DVFS the operating frequency must always be set to the maximum allowed at a given voltage. This ensures optimal process completion time and application of DPS.

If DPS cannot be applied in a given context, scaling the frequency while keeping the voltage constant does not save energy; it does, however, reduce peak power consumption. This can have a positive effect on temperature dissipation and battery life.

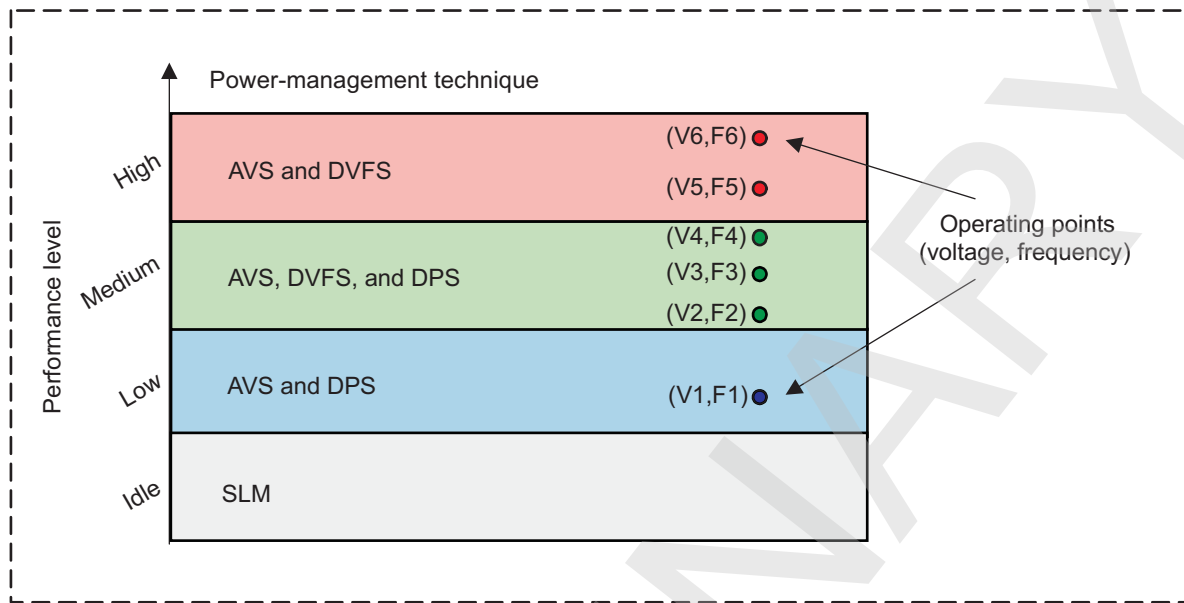
SLM must be used when no applications are running and performance requirement drops to zero.

---

**NOTE:** The OPPs shown in [Figure 3-17](#) are only for indication and clarity of text. They do not correspond to validated OPPs of the device.

---

Figure 3-17. Performance Level and Applied Power-Management Techniques



prcm-018

### 3.1.2.6.1 DPS Versus SLM

DPS and SLM are similar concepts: they consist of switching the system between high- and low-power consumption modes. However, their operating timescales differ, principally in the latency allowed for mode transitions.

DPS is generally used in an applicative context (tasks are started). Therefore, mode transitions are related to system performance requirements or the processor load. DPS transition latencies must be small (typically between 10  $\mu$ s and 100  $\mu$ s) compared to the time constraints or deadlines of the application so that they do not degrade application performance. DPS requires performance prediction to ensure that transition latencies do not deteriorate device performance to the point that real-time application deadlines are missed or the user experience degrades too much for an interactive application.

SLM is not used in an applicative context (no task started). Mode transitions are related more to system responsiveness, and the transition latencies must be small compared to user sensitivity so that they do not degrade the user experience. For SLM, transition latencies are typically 1 ms to 10 ms or more.

DPS and SLM also differ in the type of wake-up event used to exit low-power idle mode. For DPS, wake-up events are application-related (timer, DMA request, peripheral interrupt, etc.); for SLM, wake-up events are user-related (touch screen, key pressed, peripheral connections, etc.).

## 3.2 PRCM Subsystem Overview

### 3.2.1 Introduction

The power-management framework of the device significantly reduces dynamic power consumption and static leakage current to extend the life of the battery in the end product. This framework incorporates support for state-of-the-art power-management techniques. It ensures optimal device operation with significantly reduced power consumption. The power-management framework (PMFW) of the device is split over following modules:

- SCRM: Handles system-level (with one or more devices) clock and reset distribution and management.
- PRCM: A logical module composed of following three physical submodules:
  - PRM: Handles device-level power and reset management. It also handles some clocks in the device. This module always remains on, unless no power is supplied to the device pads.
  - CM1: Handles device-level clock management 1. This module remains always on unless either the device is switched to off mode or no power is supplied to the device pads.
  - CM2: Handles device-level clock management 2. This module does not remain always on. It is switched to full-retention mode when its power domain (PD\_CORE) is in RETENTION state. It is off when either the device is switched to off mode or no power is supplied to the device pads.
- SR\_MPU, SR\_IVA, and SR\_CORE (SmartReflex): Handle AVS features

Together, these modules provide enhanced power-management features with the centralized control for the clock, reset, and power-management signals in the device.

The device supports the power-management techniques with the following features:

- Partitioning of the device into voltage, power, clock, and reset domains
- Domain isolation that allows any combination of domain on/off states
- Clock tree with selective clock-gating conditions
- Hardware-controlled reset sequencing management
- Power, reset, and clock control hardware mechanism to manage sleep and wake-up dependencies of power domains
- Support for hardware-controlled autogating of module clocks
- Memory retention capability for preserving memory contents in low-power sleep mode
- DVFS support for the processor and peripherals
- AVS support for the processor and peripherals, for real-time performance adjustments
- Support for low-power device off mode input/output (I/O) pad configuration for minimum power consumption when in OFF power state

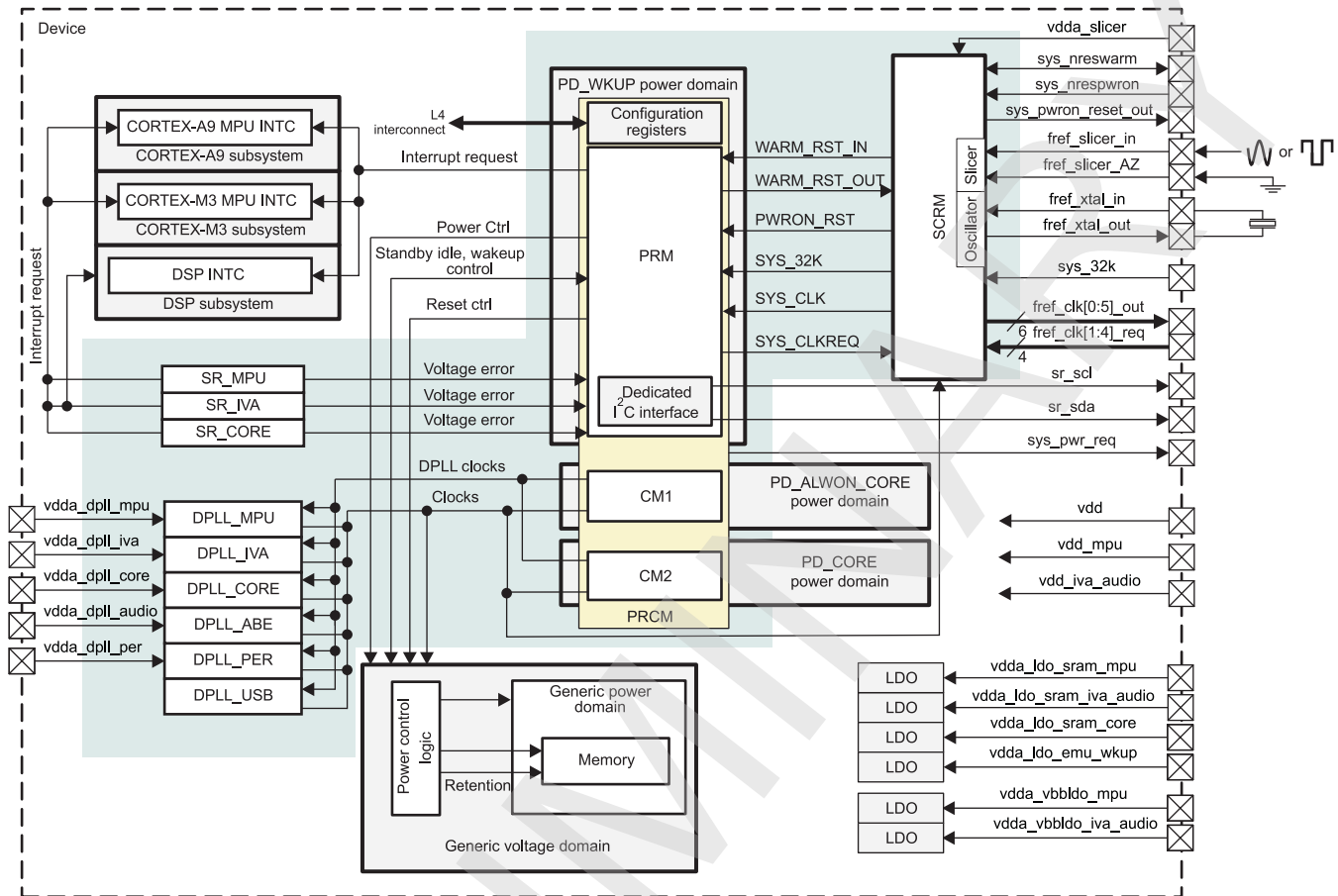
The PMFW interfaces with all the components of the device for power, clock, and reset management through associated control signals. It integrates enhanced features to let the device adapt energy consumption dynamically according to changing application and performance requirements. The innovative hardware architecture allows a substantial reduction in leakage current.

The power-management modules are fully configurable through their level 4 (L4) interface ports.

[Figure 3-18](#) is an overview of the power-management modules and their internal connections with a generic power domain.



Figure 3-18. PMFW Overview



prcm-019

### 3.2.2 Power-Management Framework Features

The power-management modules have the following features:

- Manage independent power domains
- Control scalable logic voltage domains and selectable voltage modes for memory voltage domains
- Handle standby, idle, and wake-up procedures for the modules of the device
- Allow software and partial hardware control
- Monitor and handle wake-up events
- Control system clock and reset input sources
- Manage and distribute clocks and resets with high control granularity
- Handle power-up sequences
- Debug and emulation features
- Allow adaptive and dynamic voltage scaling feature through dedicated high-speed (HS) master I<sup>2</sup>C interface
- Control of RFF of device modules to support DPS

### 3.3 PRCM Subsystem Environment

The modules of the PMFW receive the external reset, clock, and power signals. See [Figure 3-18](#).

**NOTE:** In the remainder of this chapter, power IC refers to a peripheral power source IC that is interfaced with the device. It receives power control commands (voltage scaling and power switching) from the device and provides the necessary voltages and reset signals.

The following sections describe the interfaces for external clock, reset, and power sources.

#### 3.3.1 External Clock Signals

The SCRM module receives a low-frequency clock input, sys\_32k, at 32-kHz frequency. It can receive high-frequency system clock through two possible inputs: fref\_slicer\_in or fref\_xtal\_in. The fref\_slicer\_in supplies the input clock to an on-chip slicer, while fref\_xtal\_in can be used as external crystal connection for internal high-frequency clock generation through an on-chip oscillator.

The SCRM can output six alternate clocks, fref\_clk[0:5]\_out. Output clocks 1 to 4 can be gated or enabled through associated clock request signals, fref\_clk[1:4]\_req.

[Table 3-21](#) lists the external clock pins, signal names, their direction, associated modules, description and reset values of the signals. If the signal is input to the device, then the module is the destination module for the signal; if the signal is an output from the device then the module is the source module of the signal.

**Table 3-21. External Clock Signals**

Pin	Signal	I/O <sup>(1)</sup>	PMFW Module	Description	Module Reset Value
sys_32k	SYS_32K	I	SCRM	32-kHz clock input	Z
fref_clk_ioreq	SYSClk_REQ	I/O	SCRM	System clock request. Input when pins sysboot6=0 and sysboot7=0. Otherwise output.	
fref_slicer_in	SLICER_IN	I	SCRM	System clock input to the slicer. It can be sinusoidal or square input signal. Otherwise, it may serve as alternate clock input to internal peripheral (for example, Video DAC).	Z
fref_slicer_AZ	SLICER_AZ	I	SCRM	Connected to ground at package level	Z
fref_xtal_in	XTAL_IN	I/O	SCRM	Oscillator drive input from crystal or alternate square clock input	Unknown
fref_xtal_out	XTAL_OUT	O	SCRM	Oscillator drive output to crystal	Unknown
fref_clk[1:4]_req	CLK[1:4]_REQ	I	SCRM	Auxillary clocks request	Z
fref_clk[0:5]_out	CLK[0:5]_OUT	O	SCRM	Auxillary clocks outputs 0 to 5	0

(1) I = Input; O = Output; I/O = Input and Output (bidirectional)

The auxillary clocks are controlled by the configuration of the SCRM.AUXCLK<clock id> (where clock id can be 0 to 5). It controls the following features:

- Select the clock off output polarity
- Source clock input:
  - System clock
  - Clock from DPLL\_CORE
  - Clock from DPLL\_PER
- Initiate software generated request
- Software-controlled clock gating
- Divide factor configuration between 1 and 16

The bidirectional fref\_clk\_ioreq pad can be used to manage the external system clock request and auxiliary clock request 0. With this implementation, it is not expected to externally drive the system clock request when the SOC is clock requester. This implementation requires programming the SCRM accordingly in case the SOC needs the external system clock request. The programming consists in setting `AUXCLK0[9] DISABLECLK = 1` and `AUXCLKREQ0[4:2] MAPPING = 0`. Without these settings the SCRM would wrongly interpret an external clock request as an auxiliary clock request #0.

### 3.3.2 External Reset Signals

The SCRM manages three reset signals:

- power-on reset input on sys\_nrespwron pin
- power-on reset output on sys\_pwron\_reset\_out pin
- warm reset on sys\_nreswarm pin

sys\_nrespwron is asserted at device power up to reset the full logic in the device. The SCRM asserts the sys\_pwron\_reset\_out signal to reset any associated external peripherals.

sys\_nreswarm is a bidirectional pin. It can be activated at any time by an external device or an external reset push button action to cause a global warm reset event. The SCRM then holds sys\_nreswarm to be driven out and maintained for a limited length of time at the boundary of the device. In this way, the device and its related peripherals are reset together.

The sys\_reswarm and sys\_nrespwron signals are active low. The sys\_pwron\_reset\_out signal is also active low, and active means that the peripheral is under reset.

Table 3-22 lists the external reset pins, signal names, their direction, associated modules, description and reset values of the signals. If the signal is input to the device, then the module is the destination module for the signal; if the signal is an output from the device, then the module is the source module of the signal.

**Table 3-22. External Reset Signals**

Pin	Signal	I/O <sup>(1)</sup>	PMFW Module	Description	Module Reset Value
sys_nreswarm	SYS_WARM_RST	I/O	SCRM	Warm reset input and output	Z
sys_nrespwron	SYS_PWRON_RST_IN	I	SCRM	Power-on reset input	Z
sys_pwron_reset_out	SYS_PWRON_RST_OUT	O	SCRM	Power-on reset to external peripherals	0

(1) I = Input; O = Output

### 3.3.3 External Power Control Signals

The power control signals let the PMFW modules control the device voltage levels.

The voltage level of the scalable voltage sources in the external power IC can be scaled by sending commands to the power IC through a PRCM module dedicated I<sup>2</sup>C interface. The sr\_scl and sr\_sda are the clock and data pins, respectively, for this I<sup>2</sup>C interface.

The PRCM module can also command the external power IC to switch the device voltages to off level when the device is in off mode and reactivate them when it wakes up. This is managed through the signal on the sys\_pwr\_req pin.

Table 3-23 lists the signals, I/Os, and module reset values for the signals to the external power IC.

**Table 3-23. Power Control Interface**

Pin	Signal	I/O <sup>(1)</sup>	PMFW Module	Description	Module Reset Value
sr_scl	SR_SCL	I/O	PRM	The SmartReflex dedicated I <sup>2</sup> C interface clock line	1
sr_sda	SR_SDA	I/O	PRM	The SmartReflex dedicated I <sup>2</sup> C interface data line	1
sys_pwr_req	PWR_REQ	O	PRM	Power request to external power IC, to exit device off mode.	Unknown

(1) I = Input; O = Output

### 3.3.4 External Voltage Inputs

Table 3-24 lists the external voltage sources related to the PRCM module.

**NOTE:** Table 3-24 highlights only the voltage sources that are either directly managed by the PRCM module or are received by parts of the PRCM module (for example, DPLLs, LDOs, etc.). It does not give the device voltage sources not directly associated with the PRCM module.

**Table 3-24. Voltage Sources**

Pin	Signal	Managed by the PRCM	Description
vdd	VDD_CORE_L	Yes (AVS and DVFS)	Supplies VDD_CORE_L logic voltage domain
vdd_mpu	VDD_MPU_L	Yes (AVS and DVFS)	Supplies VDD_MPU_L logic voltage domain
vdd_iva_audio	VDD_IVA_L	Yes (AVS and DVFS)	Supplies VDD_IVA_L logic voltage domain (that is, DSP and IVAHD)
vdda_dpll_core	VDDA_DPLL_CORE	No (fixed)	DPLL_CORE analog power supply
vdda_dpll_mpu	VDDA_DPLL_MPU	No (fixed)	DPLL_MPU analog power supply
vdda_dpll_iva	VDDA_DPLL_IVA	No (fixed)	DPLL_IVA analog power supply
vdda_dpll_per	VDDA_DPLL_PER	No (fixed)	DPLL_PER analog power supply
vdda_dpll_audio	VDDA_DPLL_AUDIO	No (fixed)	DPLL_AUDIO analog power supply
vdda_slicer	VDDA_SLICER	No (fixed)	Clock slicer power supply
vdd_osc	VDD_OSC	No (fixed)	Clock oscillator core power supply
vdda_ldo_sram_mpu	VDDA_LDO_SRAM_MPU	No	Supplies LDO_MPU_M for MPU Cortex-A9 SRAM
vdda_ldo_sram_iva_audio	VDDA_LDO_SRAM_IVA_AUDIO	No	Supplies LDO_IVA_M for IVA SRAM
vdda_ldo_sram_core	VDDA_LDO_SRAM_CORE	No	Supplies LDO_CORE_M for CORE SRAM
vdda_ldo_emu_wkup	VDDA_LDO_EMU_WKUP	No	Supplies LDO_WAKEUP for WAKEUP/EMU logic
vdda_bdgp	VDDA_BDGP	No (fixed)	Bandgap supply input
vdda_vbldo_mpu	VDDA_VBBLDO_MPU	No	MPU VBB LDO supply input
vdda_vbldo_iva_audio	VDDA_VBBLDO_IVA	No	IVA VBB LDO supply input

### 3.4 PRCM Subsystem Integration

The internal configuration registers of the power-management modules can be accessed for configuration and control through their respective WKUP\_L4 interconnect. In addition to the L4 interconnect, the internal modules interface contains the following interfaces and signals:

- A set of signals for idle/wake-up control for each module
- Clocks and reset signals for the modules
- Power control signals (switches and memories) to the power domains
- Interrupts to the MPU Cortex-A9, MPU Cortex-M3, and DSP INTCs
- Voltage error commands from the SmartReflex modules to the external power IC
- Phase-locked loop (PLL) control commands for recalibration and bypass of the digital phase-locked loops (DPLLs)

Figure 3-18 shows details of the control interface to a generic power domain.

#### 3.4.1 Device Power-Management Layout

The PMFW sees the device split into voltage domains, power domain, and clock domains. Table 3-25 lists the device-level view with modules association to clock domains, power domains, and voltage domains.

**Table 3-25. PMFW Device-Level Layout**

Voltage Domain	Power Domain	Clock Domain	Module	
LDO_WAKEUP	PD_EMU	CD_EMU	DEBUG	
		N/A (the PRCM module itself)	CM1_EMU	
	PD_WKUP		CD_WKUP	SYSCTRL_PADCONF_WKUP
			BANDGAP	
			SYSCTRL_GENERAL_WKUP	
			GPIO1	
			Keyboard	
			SAR_RAM	
			32KTIMER	
			GPTIMER1	
			WDTIMER2	
			L4_WKUP interconnect	
			N/A (the PRCM module itself)	PRM
			N/A (the PRCM module itself)	SCRM
VDD_CORE_L	PD_ALWON_CORE	CD_L4_ALWON_CORE	SR_CORE	
			SR_IVA	
			SR_MPU	
		N/A (the PRCM module itself)	CM1	
		N/A (clock generator)	DPLL_ABE	
		N/A (clock generator)	DPLL_CORE	
		N/A (clock generator)	DPLL_PER	
		N/A (not clocked by the PRCM)	CORTEXM3_WKUPGEN	
		N/A (not clocked by the PRCM)	SDMA_WKUPGEN	
		N/A (not clocked by the PRCM)	SPINNER	
		PD_CAM	CD_CAM	Face detect
				ISS
		PD_CORE	CD_L4_CFG	SYSCTRL_PADCONF_CORE
				SYSCTRL_GENERAL_CORE
		SPINLOCK		
		L4_CFG interconnect		
		Mailbox		

**Table 3-25. PMFW Device-Level Layout (continued)**

Voltage Domain	Power Domain	Clock Domain	Module
			SAR_ROM
		CD_EMIF	DDRPHY
			DLL
			DMM
			EMIF1
			EMIF2
			EMIF_FW
		CD_CORTEXM3	CORTEXM3
		CD_L3_2	GPMC
			L3_2 interconnect
			OCMC_RAM
		CD_L3_INSTR	L3_3 interconnect
			L3_INSTR interconnect
			INTERCONN_WP1
		CD_L3_1	L3_1 interconnect
		CD_C2C	C2C
			C2C_FW
		CD_DMA	sDMA
		N/A (the PRCM module itself)	CM2
	PD_DSS	CD_DSS	DSS
	PD_SGX	CD_SGX	SGX
	PD_L3_INIT	CD_L3_INIT	HSI
			P1500
			HSMMC1
			HSMMC2
			USBPHY
			FSUSB
			HSUSBHOST
			HSUSBOTG
			HSUSBTLL
		N/A (the PRCM module itself)	CM1_USB
		N/A (clock generator)	DPLL_USB
	PD_L4_PER	CD_L4_PER	GPTIMER10
			GPTIMER11
			GPTIMER2
			GPTIMER3
			GPTIMER4
			GPTIMER9
			ELM
			GPIO2
			GPIO3
			GPIO4
			GPIO5
			GPIO6
			HDQ
			I2C1
			I2C2

**Table 3-25. PMFW Device-Level Layout (continued)**

Voltage Domain	Power Domain	Clock Domain	Module
			I2C3
			I2C4
			L4_PER interconnect
			MCBSP4
			MCSP1
			MCSP2
			MCSP3
			MCSP4
			HSMMC3
			HSMMC4
			HSMMC5
			SLIMBUS2
			UART1
			UART2
			UART3
			UART4
	PD_STD_EFUSE	CD_STD_EFUSE	STD_EFUSE
VDD_IVA_L	PD_ALWON_DSP	N/A (clock generator)	DPLL_IVA
		N/A (not clocked by the PRCM)	DSP_WKUPGEN
	PD_AUDIO	CD_ABE	Audio engine
			DMIC
			L4_ABE interconnect
			MCASP1
			MCBSP1
			MCBSP2
			MCBSP3
			MCPDM
			SLIMBUS1
			GPTIMER5
			GPTIMER6
			GPTIMER7
			GPTIMER8
			WDTIMER3
		N/A (the PRCM module itself)	CM1_ABE
	PD_IVAHD	CD_IVAHD	IVAHD
			SL2
	PD_DSP	CD_DSP	DSP
VDD_MPU_L	PD_ALWON_MPU	N/A (clock generator)	DPLL_MPU
		N/A (not clocked by the PRCM)	CORTEXA9_MPU_INTC
	PD_MPU	CD_CORTEXA9	CORTEXA9

### 3.4.2 Power-Management Scheme, Reset, and Interrupt Requests

#### 3.4.2.1 Power Domain

Table 3-26 lists the PMFW modules and their associated power domains.

**Table 3-26. PMFW Module Power Domains**

PMFW Module	Power Domain
SCRM	PD_WKUP
PRM	PD_WKUP
CM1	PD_ALWON_CORE
CM2	PD_CORE
SR_MPU	PD_ALWON_CORE
SR_IVA	PD_ALWON_CORE
SR_CORE	PD_ALWON_CORE

The PRM part of the PRCM module is in the PD\_WKUP power domain, which is continuously active. It is composed of the logic that must be permanently supplied to manage domain power-state transitions and detect wake-up events.

The CM1 part of the PRCM module is in the PD\_ALWON\_CORE power domain, which is an always-on power domain, while the CM2 part of the PRCM module is in the PD\_CORE power domain, which can be activated and deactivated according to the requirements of the executing applications.

### 3.4.2.2 Resets

The PMFW modules are reset by independent reset signals (see [Table 3-27](#)).

**Table 3-27. PMFW Module Reset Signals**

PMFW Module	Reset Signal
SCRM	SYS_PWRON_RST_IN
PRM	SYS_PWRON_RST_IN
CM1	ALWON_CM1_PWRON_RST ALWON_CM1_RST
CM2	CM2_PWRON_RET_RST CM2_RET_RST
SR_MPU	ALWON_CORE_RST
SR_IVA	ALWON_CORE_RST
SR_CORE	ALWON_CORE_RST

**NOTE:** See [Section 3.5.3, Reset Sources](#) for details on the reset trigger sources and assertion conditions.

### 3.4.2.3 Interrupt Requests

The PMFW modules can generate following interrupts listed in [Table 3-28](#).

**Table 3-28. PMFW Module Interrupts**

Source PMFW Module	Source Signal Name	Destination Interrupt Controller	Destination Signal Name
PRCM	PRCM_MPU_IRQ	CORTEX-A9_MPU_INTC	MA_IRQ_11
PRCM	PRCM_M3_IRQ	CORTEX-M3_MPU_INTC	MM_IRQ_31
PRCM	PRCM_DSP_IRQ	DSP_INTC	D_IRQ_57
SR_IVA	SR_IVA_IRQ	CORTEX-A9_MPU_INTC	MA_IRQ_102
SR_MPU	SR_MPU_IRQ	CORTEX-A9_MPU_INTC	MA_IRQ_18
SR_CORE	SR_CORE_IRQ	CORTEX-A9_MPU_INTC	MA_IRQ_19



## 3.5 Reset Management Functional Description

### 3.5.1 Overview

In the device the reset scheme is managed by the following modules:

- SCRM: System-level reset management. It provides correct reset routing and sequencing when one or more devices are stacked together in the same package.
- PRM: Device-level reset management

#### 3.5.1.1 SCRM Reset Management Functional Description

The SCRM handles the device power-on and warm reset pads: `sys_nrespwron` and `sys_nreswarm` resets. Its tasks basically consist of:

- Extending reset duration beyond the pad reset release
- Routing the device pad resets to the reset manager in the PRM module

The SCRM is functionally sensitive to the device power-on reset. However, some of the register bits of SCRM registers are reset when a device warm reset occurs.

##### 3.5.1.1.1 Power-On Reset

The SCRM receives the device power-on reset on the `sys_nrespwron` reset pad of the device. It extends the power-on reset duration to the device PRM module at least until a stable 32-kHz clock can be provided to it. The 32-kHz clock version of the device is active once the SCRM releases the internal power-on reset. The SCRM automatically deasserts the reset.

The SCRM automatically requests the system clock under a power-on reset condition. This mechanism is functionally active on the 32-kHz clock.

The power-on reset can also be asserted through software control. Software must enable the software reset assertion feature by setting the `SCRM.EXTPWRONRSTCTRL[0]` ENABLE bit and then by setting the `SCRM.EXTPWRONRSTCTRL[1]` PWRONRST bit to assert the reset.

##### 3.5.1.1.2 Warm Reset

The device warm reset can be received from an external source through warm reset pad `sys_nreswarm` or from the device PRM module as a result of internal event.

The SCRM extends the device pad warm reset duration to the device PRM module.

The SCRM routes a warm reset source (that is, `sys_nreswarm` pad/PRM module) to a warm reset destination (that is, PRM module).

This routing of reset from the device pad to the PRM is enabled and disabled using the `SCRM.EXTWARMRSTCTRL` register. It is enabled by default.

The source of a warm reset, for each destination is logged in the corresponding `SCRM.EXTWARMRSTST_REG` register. The status of the reset source is logged when the destination reset is released. This is done at 32-kHz clock.

Software must clear the reset status bit.

#### 3.5.1.2 PRM Reset Management Functional Description

The PRM module manages the resets to all power domains inside the device.

---

**NOTE:** The PRM module has no knowledge or control over resets generated locally within a module (for example, through configuration register bit `<Module name>_SYSCONFIG[x] SOFTRESET`). Software reset has the same effect on the module logic as a hardware reset.

---

All PRM reset outputs are asynchronously asserted, while the deassertion is synchronous to the SYS\_CLK clock. The reset managers in PRM use this clock for internal stall (delay) counter to delay deassertion of reset when the input reset source is deasserted.

In each power domain, one or more reset domains are defined. A reset domain is defined by a unique reset signal that originates from the reset manager and is connected to one or more modules of the device. All the connected modules of the reset domain are reset simultaneously when the reset signal is asserted. Independent control of these reset domains allows sequencing of the release of resets and ensures a safe reset of the entire power domain.

---

**NOTE:** All internal reset signals are active low, except for the DPLL reset signals, which are active high.

---

### 3.5.2 General Characteristics of Reset Signals

Reset signals can be categorized based on four criteria:

- Scope: Global or local reset
- Occurrence: Cold or warm reset
- Source type: Software or hardware triggered reset
- Retention type: Retention reset or nonretention reset

#### 3.5.2.1 Scope

A reset signal can be categorized according to its scope (the area of the device affected by the reset):

- Global reset: Affects the entire logic of the device; all modules are reset. Generally, occurs when the device powers up or an abnormal operation is detected (the eFuse bad device is detected, etc.).
- Local reset: Affects one power domain, reset domain, or module. Generally, when a power domain transitions from off mode to active mode, or when a software-reset control bit for a domain is set, only the group of modules within that domain is affected.

#### 3.5.2.2 Occurrence

A reset signal can be categorized depending on when the reset occurs:

- Cold reset: Occurs only on device power up or in certain emulation modes. The cold reset is a global reset that affects every module in the device. It usually corresponds to the initial power-on reset.
- Warm reset: Occurs when the device is in normal operating state. A module may use a warm reset to reset a subset of its logic. This is often done to speed up reset recovery time; that is, the time to transition to a safe operating state, compared to the time required upon receipt of a cold reset. Warm reset events include software-triggered reset per power domain, watchdog time-out, externally triggered and emulation initiated.

Modules that behave differently in cold reset and warm reset have two reset signals: RST and PWRON\_RST. These reset signals reconstruct warm reset and cold reset in modules that require them.

The following modules are reset upon global cold reset events and not upon global warm reset events:

- All DPLLs associated with the PRCM
- All eFuses
- EMIF
- Control module – All the module instances in the PD\_WKUP power domain

---

**NOTE:** For information about the PRCM module registers affected by the global warm reset, see the register description in [Section 3.11, PRCM Register Manual](#).

---

#### 3.5.2.3 Source Type

A reset can be categorized depending on whether it is software-controlled or hardware-triggered:

- Software reset: Triggered by setting a bit in a configuration register of the PRCM module
- Hardware reset: Triggered by a signal from a hardware module inside or outside the PRCM module

### 3.5.2.4 Retention Type

The Power Domain Manager in PRM controls the assertion of two types of local cold reset sources, retention and nonretention reset. They are identified by the naming convention <PowerDomain>\_DOM\_RET\_RST and <PowerDomain>\_DOM\_RST, respectively.

Upon transitioning a power domain to the ON-ACTIVE power state from the OFF power state:

- Retention type reset source is always asserted.
- Nonretention type reset source is always asserted.

Upon transitioning a power domain to the ON-ACTIVE power state from the RETENTION power state:

- Retention type reset source is never asserted.
- Nonretention type reset source is optionally asserted. The software-selectable option enables the PRM to support two retention mechanisms: CSWRET and OSWRET. Nonretention type reset source has following behavior:
  - It is asserted in-case of power domain state transition from OSWR-RETENTION state.
  - It is not asserted in-case of power domain state transition from CSWR-RETENTION state.

## 3.5.3 Reset Sources

The reset sources triggering the reset managers in the PRM are listed in this section.

### 3.5.3.1 Global Reset Sources

Table 3-29 lists the global reset sources of the device. The global reset source signals received by the reset manager trigger the reset of all the device modules. For all hardware reset signals, the source of the reset is identified; for the software reset signals, the reset triggering bit is identified.

**Table 3-29. Global Reset Sources**

Type <sup>(1)</sup>	Name	Source/Control	Description
H/C	SYS_PWRON_RST	Received from SCRM through input pin or through software control	The entire device is reset on power up.
H/C	BAD_DEVICE_RST	PRCM	Asserted when during the power-up sequence the device is identified as bad, after reading eFuses
H/W	SYS_WARMIN_RST	Received from SCRM through input pin or through software control	External hardware warm reset
S/C	GLOBAL_COLD_SW_RST	PRCM module.PRM_RSTCTRL[1] RST_GLOBAL_COLD_SW	Global software cold reset
S/W	GLOBAL_WARM_SW_RST	PRCM module.PRM_RSTCTRL[0] RST_GLOBAL_WARM_SW	Global software warm reset
H/W	VDD_CORE_VOLT_MGR_RST	PRCM	Asserted by the voltage manager FSMs when no response from the power IC is received during wake-up transition from retention or off mode
H/W	VDD_IVA_VOLT_MGR_RST	PRCM	
H/W	VDD_MPU_VOLT_MGR_RST	PRCM	
H/W	ICEPICK_RST	ICEPick module	It is used in emulation mode only.
H/W	ICEPICKPOR_RST	ICEPick module	It is used in emulation mode only.
H/W	CORTEXA9_WDT_RST	Either WDTIMER2 or MPU_A9 subsystem	It is triggered by a time-out event.
H/W	C2C_WARM_RST	C2C module	Global hardware warm reset.

(1) H = Hardware reset, S = Software reset, C = Cold reset, W = Warm reset

### 3.5.3.2 Local Reset Sources

In addition to the global reset sources the device can have a number of local reset sources for each power domain. The local reset sources can be cold or warm reset sources. They can be software or hardware

triggered. [Table 3-30](#) identifies the possible types of hardware-triggered local cold reset sources. Some power domains may support one or both of these local cold reset sources. The table does not highlight the software-triggered local warm reset sources which are listed in the reset management section of the respective power domains. A local reset source signal received by the reset manager resets only a specific part of the device (for example, some modules/subsystems within the power domain).

**Table 3-30. Local Reset Sources**

Type <sup>(1)</sup>	Name	Source/Control	Description
H/C	<Power domain>_RET_RST	PRCM	Asserted only for a power domain state transition from OFF to ON-ACTIVE state
H/C	<Power domain>_RST	PRCM	Asserted for any power domain transition from OFF or OSWR-RETENTION state to ON-ACTIVE state

(1) H = Hardware reset, C = Cold reset

### 3.5.4 Reset Domains

A power domain may receive power-on reset (PWRON\_RST) and/or normal reset (RST) signals. These signals reset nonretention logic and behave as follows:

- On any global or local cold reset, RST and PWRON\_RST are asserted.
- On any global or local warm reset, only RST is asserted.

A power domain may receive two additional retention logic reset signals: power-on retention reset (PWRON\_RET\_RST) and/or retention reset (RET\_RST). These signals behave as follows:

- On any global cold reset or wakeup from OFF state to ON-ACTIVE state, RET\_RST and PWRON\_RET\_RST are asserted.
- On any global warm reset, only RET\_RST is asserted.
- On wakeup from RETENTION state, these signals are not asserted.

This section presents the trigger sources and attributes for all reset domains of the device. See [Section 3.5.3, Reset Sources](#), for an explanation on each reset trigger source of the device.

[Table 3-31](#) identifies the associated power and reset domains for each module.

**Table 3-31. Modules, Power Domains and Reset Domains Association**

Module	Power Domain	Reset Domains
CM1	PD_ALWON_CORE	ALWON_CM1_PWRON_RST, ALWON_CM1_RST
DPLL_ABE	PD_ALWON_CORE	ALWON_CORE_PWRON_RST
DPLL_CORE	PD_ALWON_CORE	ALWON_CORE_PWRON_RST
DPLL_PER	PD_ALWON_CORE	ALWON_CORE_PWRON_RST
CORTEXM3_WKUPGEN	PD_ALWON_CORE	None
SDMA_WKUPGEN	PD_ALWON_CORE	None
SR_CORE	PD_ALWON_CORE	ALWON_CORE_RST
SR_IVA	PD_ALWON_CORE	ALWON_CORE_RST
SR_MPU	PD_ALWON_CORE	ALWON_CORE_RST
SPINNER	PD_ALWON_CORE	None
DPLL_IVA	PD_ALWON_DSP	DPLL_IVA_PWRON_RST
DSP_WKUPGEN	PD_ALWON_DSP	ALWON_IVA_RST
DPLL_MPU	PD_ALWON_MPU	DPLL_MPU_PWRON_RST
CORTEXA9_MPU_INTC	PD_ALWON_MPU	ALWON_MPU_RST
AUDIO ENGINE	PD_AUDIO	AUDIO_RST
CM1_ABE	PD_AUDIO	AUDIO_PWRON_RST
DMIC	PD_AUDIO	AUDIO_RST
L4_ABE interconnect	PD_AUDIO	AUDIO_RST
MCASP1	PD_AUDIO	AUDIO_RST

**Table 3-31. Modules, Power Domains and Reset Domains Association (continued)**

Module	Power Domain	Reset Domains
MCBSP1	PD_AUDIO	AUDIO_RST
MCBSP2	PD_AUDIO	AUDIO_RST
MCBSP3	PD_AUDIO	AUDIO_RST
MCPDM	PD_AUDIO	AUDIO_RST
SLIMBUS1	PD_AUDIO	AUDIO_RST
GPTIMER5	PD_AUDIO	AUDIO_RST
GPTIMER6	PD_AUDIO	AUDIO_RST
GPTIMER7	PD_AUDIO	AUDIO_RST
GPTIMER8	PD_AUDIO	AUDIO_RST
WDTIMER3	PD_AUDIO	AUDIO_RST
FACE DETECT	PD_CAM	CAM_RST
ISS	PD_CAM	CAM_RST
SYSCTRL_PADCONF_CORE	PD_CORE	CORE_PWRON_RET_RST
CM2	PD_CORE	CM2_PWRON_RET_RST, CM2_RET_RST
DDRPHY	PD_CORE	EMIF_PWRON_RST
SYSCTRL_GENERAL_CORE	PD_CORE	CORE_PWRON_RET_RST
DLL	PD_CORE	DLL_RST
DMM	PD_CORE	CORE_RET_RST, CORE_RST
CORTEXM3	PD_CORE	CORTEXM3_PWRON_RST, CORTEXM3_RET_RST, CORTEXM3_RST1, CORTEXM3_RST2, CORTEXM3_RST3
EMIF1	PD_CORE	CORE_PWRON_RET_RST
EMIF2	PD_CORE	CORE_PWRON_RET_RST
EMIF_FW	PD_CORE	CORE_PWRON_RET_RST, CORE_RST
GPMC	PD_CORE	CORE_RET_RST
SPINLOCK	PD_CORE	CORE_RET_RST
L3_2 interconnect	PD_CORE	CORE_PWRON_RET_RST, CORE_RST
L3_3 interconnect	PD_CORE	CORE_PWRON_RET_RST, CORE_RST
L3_1 interconnect	PD_CORE	CORE_PWRON_RET_RST, CORE_RST
L3_INSTR interconnect	PD_CORE	CORE_RST
INTERCONN_WP1	PD_CORE	CORE_PWRON_RET_RST, CORE_RST
L4_CFG interconnect	PD_CORE	CORE_PWRON_RET_RST, CORE_RST
Mailbox	PD_CORE	CORE_RET_RST
OCMC_RAM	PD_CORE	CORE_RST
C2C	PD_CORE	CORE_RET_RST, CORE_RST
C2C_FW	PD_CORE	CORE_PWRON_RET_RST, CORE_RST
SAR_ROM	PD_CORE	CORE_RST
sDMA	PD_CORE	SDMA_RET_RST
DSS	PD_DSS	DSS_RET_RST, DSS_RST
CM1_EMU	PD_EMU	EMU_PWRON_RST
DEBUG	PD_EMU	EMU_EARLY_PWRON_RST, EMU_PWRON_RST, EMU_RST
SGX	PD_SGX	SGX_RST
IVAHD	PD_IVAHD	IVAHD_PWRON_RST, IVAHD_RST, IVAHD_SEQ1_RST, IVAHD_SEQ2_RST
SL2	PD_IVAHD	IVAHD_RST
CM1_USB	PD_L3_INIT	L3_INIT_PWRON_RST

**Table 3-31. Modules, Power Domains and Reset Domains Association (continued)**

Module	Power Domain	Reset Domains
DPLL_USB	PD_L3_INIT	DPLL_L3_INIT_PWRON_RET_RST
HSI	PD_L3_INIT	L3_INIT_RET_RST
P1500	PD_L3_INIT	L3_INIT_RST
HSMMC1	PD_L3_INIT	L3_INIT_RET_RST
HSMMC2	PD_L3_INIT	L3_INIT_RET_RST
USBPHY	PD_L3_INIT	L3_INIT_RST
FSUSB	PD_L3_INIT	L3_INIT_RET_RST
HSUSBHOST	PD_L3_INIT	L3_INIT_RET_RST
HSUSBOTG	PD_L3_INIT	L3_INIT_RET_RST
HSUSBTLL	PD_L3_INIT	L3_INIT_RET_RST
GPTIMER10	PD_L4_PER	L4_PER_RST
GPTIMER11	PD_L4_PER	L4_PER_RST
GPTIMER2	PD_L4_PER	L4_PER_RST
GPTIMER3	PD_L4_PER	L4_PER_RST
GPTIMER4	PD_L4_PER	L4_PER_RST
GPTIMER9	PD_L4_PER	L4_PER_RST
ELM	PD_L4_PER	L4_PER_RST
GPIO2	PD_L4_PER	L4_PER_RET_RST
GPIO3	PD_L4_PER	L4_PER_RET_RST
GPIO4	PD_L4_PER	L4_PER_RET_RST
GPIO5	PD_L4_PER	L4_PER_RET_RST
GPIO6	PD_L4_PER	L4_PER_RET_RST
HDQ	PD_L4_PER	L4_PER_RST
I2C1	PD_L4_PER	L4_PER_RET_RST
I2C2	PD_L4_PER	L4_PER_RST
I2C3	PD_L4_PER	L4_PER_RST
I2C4	PD_L4_PER	L4_PER_RST
L4_PER interconnect	PD_L4_PER	L4_PER_PWRON_RET_RST, L4_PER_RST
MCBSP4	PD_L4_PER	L4_PER_RST
MCSPI1	PD_L4_PER	L4_PER_RST
MCSPI2	PD_L4_PER	L4_PER_RST
MCSPI3	PD_L4_PER	L4_PER_RST
MCSPI4	PD_L4_PER	L4_PER_RST
HSMMC3	PD_L4_PER	L4_PER_RST
HSMMC4	PD_L4_PER	L4_PER_RST
HSMMC5	PD_L4_PER	L4_PER_RST
SLIMBUS2	PD_L4_PER	L4_PER_RST
UART1	PD_L4_PER	L4_PER_RET_RST
UART2	PD_L4_PER	L4_PER_RET_RST
UART3	PD_L4_PER	L4_PER_RET_RST
UART4	PD_L4_PER	L4_PER_RET_RST
CORTEXA9	PD_MPU	CORTEXA9_PWRON_RST, CORTEXA9_RST
STD_EFUSE	PD_STD_EFUSE	STD_EFUSE_PWRON_RST
DSP	PD_DSP	DSP_RST, DSP_PWRON_RST, DSP_RET_RST, DSP_SYS_RST
SYSCTRL_PADCONF_WKUP	PD_WKUP	WKUP_PWRON_RST

**Table 3-31. Modules, Power Domains and Reset Domains Association (continued)**

Module	Power Domain	Reset Domains
BANDGAP	PD_WKUP	WKUP_PWRON_RST
SYSCTRL_GENERAL_WKUP	PD_WKUP	WKUP_PWRON_RST
GPIO1	PD_WKUP	WKUP_RST
Keyboard	PD_WKUP	WKUP_RST
SAR_RAM	PD_WKUP	WKUP_RST
32KTIMER	PD_WKUP	WKUP_RST, WKUP_SYS_PWRON_RST
GPTIMER1	PD_WKUP	WKUP_RST
WDTIMER2	PD_WKUP	WKUP_RST
L4_WKUP interconnect	PD_WKUP	WKUP_RST

Table 3-32 lists the reset sources that trigger the reset domains of the device.

**Table 3-32. Reset Sources For the Reset Domains**

Reset Domain	Reset Source	Reset Source Type
AUDIO_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
AUDIO_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
ICEPICK_RST	Global warm	
ALWON_CM1_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
ALWON_CM1_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
ICEPICK_RST	Global warm	
ALWON_CORE_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold

**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
ALWON_CORE_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
ALWON_IVA_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
ALWON_MPU_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
CAM_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
VDD_IVA_VOLT_MGR_RST	Global warm	



**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
CM2_PWRON_RET_RST	BAD_DEVICE_RST	Global cold	
	GLOBAL_COLD_SW_RST	Global cold	
	ICEPICKPOR_RST	Global cold	
	SYS_PWRON_RST	Global cold	
CM2_RET_RST	BAD_DEVICE_RST	Global cold	
	GLOBAL_COLD_SW_RST	Global cold	
	GLOBAL_WARM_SW_RST	Global warm	
	ICEPICKPOR_RST	Global cold	
	CORTEXA9_WDT_RST	Global warm	
	SYS_PWRON_RST	Global cold	
	SYS_WARMIN_RST	Global warm	
	VDD_CORE_VOLT_MGR_RST	Global warm	
	VDD_IVA_VOLT_MGR_RST	Global warm	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
	CORE_PWRON_RET_RST	BAD_DEVICE_RST	Global cold
		GLOBAL_COLD_SW_RST	Global cold
ICEPICKPOR_RST		Global cold	
ICEPICK_RST		Global warm	
CORE_RET_RST	BAD_DEVICE_RST	Global cold	
	GLOBAL_COLD_SW_RST	Global cold	
	GLOBAL_WARM_SW_RST	Global warm	
	ICEPICKPOR_RST	Global cold	
	CORTEXA9_WDT_RST	Global warm	
	SYS_PWRON_RST	Global cold	
	SYS_WARMIN_RST	Global warm	
	VDD_CORE_VOLT_MGR_RST	Global warm	
	VDD_IVA_VOLT_MGR_RST	Global warm	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
	ICEPICK_RST	Global warm	
	CORE_RST	BAD_DEVICE_RST	Global cold
GLOBAL_COLD_SW_RST		Global cold	
GLOBAL_WARM_SW_RST		Global warm	
ICEPICKPOR_RST		Global cold	
CORTEXA9_WDT_RST		Global warm	
SYS_PWRON_RST		Global cold	
SYS_WARMIN_RST		Global warm	
VDD_CORE_VOLT_MGR_RST		Global warm	
VDD_IVA_VOLT_MGR_RST		Global warm	
VDD_MPU_VOLT_MGR_RST		Global warm	
C2C_WARM_RST		Global warm	
ICEPICK_RST		Global warm	
DLL_RST		BAD_DEVICE_RST	Global cold
	DLL_FREQCHANGE_RST	Local Warm	

**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
DPLL_IVA_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DPLL_L3_INIT_PWRON_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DPLL_MPU_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
DSS_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
ICEPICK_RST	Global warm	
DSS_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
ICEPICK_RST	Global warm	

**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
CORTEXM3_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
CORTEXM3_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	RM_CORTEXM3_RSTCTRL_RST3	Local Warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
	CORTEXM3_RST1	BAD_DEVICE_RST
CORTEXM3_ICECRUSHER1_RST		Local Warm
GLOBAL_COLD_SW_RST		Global cold
GLOBAL_WARM_SW_RST		Global warm
ICEPICKPOR_RST		Global cold
CORTEXA9_WDT_RST		Global warm
RM_CORTEXM3_RSTCTRL_RST1		Local Warm
SYS_PWRON_RST		Global cold
SYS_WARMIN_RST		Global warm
VDD_CORE_VOLT_MGR_RST		Global warm
VDD_IVA_VOLT_MGR_RST		Global warm
VDD_MPU_VOLT_MGR_RST		Global warm
C2C_WARM_RST		Global warm
ICEPICK_RST		Global warm
CORTEXM3_RST2	BAD_DEVICE_RST	Global cold
	CORTEXM3_ICECRUSHER2_RST	Local Warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	RM_CORTEXM3_RSTCTRL_RST2	Local Warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
CORTEXM3_RST3	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold

**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	RM_CORTEXM3_RSTCTRL_RST3	Local Warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
EMIF_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
EMU_EARLY_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	SYS_PWRON_RST	Global cold
EMU_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
EMU_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
ICEPICK_RST	Global warm	
SGX_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
ICEPICK_RST	Global warm	
IVAHD_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold

**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
IVAHD_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	RM_IVAHD_RSTCTRL_RST3	Local Warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
IVAHD_SEQ1_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	IVAHD_ICECRUSHER1_RST	Local Warm
	CORTEXA9_WDT_RST	Global warm
	RM_IVAHD_RSTCTRL_RST1	Local Warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
ICEPICK_RST	Global warm	
IVAHD_SEQ2_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	IVAHD_ICECRUSHER2_RST	Local Warm
	CORTEXA9_WDT_RST	Global warm
	RM_IVAHD_RSTCTRL_RST2	Local Warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
ICEPICK_RST	Global warm	
L3_INIT_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold

**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
L3_INIT_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
L3_INIT_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
L4_PER_PWRON_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
L4_PER_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
L4_PER_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
VDD_CORE_VOLT_MGR_RST	Global warm	

**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type	
	VDD_IVA_VOLT_MGR_RST	Global warm	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
	ICEPICK_RST	Global warm	
CORTEXA9_PWRON_RST	BAD_DEVICE_RST	Global cold	
	GLOBAL_COLD_SW_RST	Global cold	
	ICEPICKPOR_RST	Global cold	
	SYS_PWRON_RST	Global cold	
CORTEXA9_RST	BAD_DEVICE_RST	Global cold	
	GLOBAL_COLD_SW_RST	Global cold	
	GLOBAL_WARM_SW_RST	Global warm	
	ICEPICKPOR_RST	Global cold	
	CORTEXA9_ICECRUSHER_RST	Local Warm	
	CORTEXA9_WDT_RST	Global warm	
	SYS_PWRON_RST	Global cold	
	SYS_WARMIN_RST	Global warm	
	VDD_CORE_VOLT_MGR_RST	Global warm	
	VDD_IVA_VOLT_MGR_RST	Global warm	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
	ICEPICK_RST	Global warm	
	SDMA_RET_RST	BAD_DEVICE_RST	Global cold
		GLOBAL_COLD_SW_RST	Global cold
GLOBAL_WARM_SW_RST		Global warm	
ICEPICKPOR_RST		Global cold	
CORTEXA9_WDT_RST		Global warm	
SDMA_RESTORE_RST		Local Warm	
SYS_PWRON_RST		Global cold	
SYS_WARMIN_RST		Global warm	
VDD_CORE_VOLT_MGR_RST		Global warm	
VDD_IVA_VOLT_MGR_RST		Global warm	
VDD_MPU_VOLT_MGR_RST		Global warm	
C2C_WARM_RST		Global warm	
ICEPICK_RST		Global warm	
STD_EFUSE_PWRON_RST	SYS_PWRON_RST	Global cold	
DSP_RST	BAD_DEVICE_RST	Global cold	
	GLOBAL_COLD_SW_RST	Global cold	
	GLOBAL_WARM_SW_RST	Global warm	
	ICEPICKPOR_RST	Global cold	
	CORTEXA9_WDT_RST	Global warm	
	RM_DSP_RSTCTRL_RST1	Local Warm	
	SYS_PWRON_RST	Global cold	
	SYS_WARMIN_RST	Global warm	
	DSP_EMU_RESET_REQ_TR	Local Warm	
	VDD_CORE_VOLT_MGR_RST	Global warm	
	VDD_IVA_VOLT_MGR_RST	Global warm	
VDD_MPU_VOLT_MGR_RST	Global warm		

**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
DSP_PWRON_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
	ICEPICK_RST	Global warm
DSP_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	RM_DSP_RSTCTRL_RST2	Local Warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
	DSP_SYS_RST	BAD_DEVICE_RST
GLOBAL_COLD_SW_RST		Global cold
GLOBAL_WARM_SW_RST		Global warm
ICEPICKPOR_RST		Global cold
CORTEXA9_WDT_RST		Global warm
RM_DSP_RSTCTRL_RST2		Local Warm
SYS_PWRON_RST		Global cold
SYS_WARMIN_RST		Global warm
VDD_CORE_VOLT_MGR_RST		Global warm
VDD_IVA_VOLT_MGR_RST		Global warm
VDD_MPU_VOLT_MGR_RST		Global warm
C2C_WARM_RST		Global warm
ICEPICK_RST		Global warm
WKUP_PWRON_RST		BAD_DEVICE_RST
	GLOBAL_COLD_SW_RST	Global cold
	ICEPICKPOR_RST	Global cold
	SYS_PWRON_RST	Global cold
WKUP_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
VDD_MPU_VOLT_MGR_RST	Global warm	



**Table 3-32. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
	C2C_WARM_RST	Global warm
	ICEPICK_RST	Global warm
WKUP_SYS_PWRON_RST	SYS_PWRON_RST	Global cold

Table 3-33 lists the attributes of the reset manager associated with the reset domains. The clock to the reset manager, the delay count before release of reset, and the reset release stall conditions for the reset domains are listed. Apart from the identified reset release stall conditions, a universal release stall condition (that is, applicable to all reset signals) is the eFuse scan completion.

**Table 3-33. Reset Domains Attributes**

Reset Domain	RM Clock	RM Clock Count	Release Stall Conditions
AUDIO_PWRON_RST	SYS_CLK	0x0	None
AUDIO_RST	SYS_CLK	0x0	None
ALWON_CM1_PWRON_RST	SYS_CLK	0x2	None
ALWON_CM1_RST	SYS_CLK	0x2	L4_ROOT_CLK clock is not active.
ALWON_CORE_PWRON_RST	SYS_CLK	0x0	None
ALWON_CORE_RST	SYS_CLK	0x0	None
ALWON_IVA_RST	SYS_CLK	0x0	None
ALWON_MPU_RST	SYS_CLK	0x0	MPU_CLK is not active.
CAM_RST	SYS_CLK	0x0	None
CM2_PWRON_RET_RST	SYS_CLK	0x2	None
CM2_RET_RST	SYS_CLK	0x2	L4_ICLK clock is not active.
CORE_PWRON_RET_RST	SYS_CLK	0x0	None
CORE_RET_RST	SYS_CLK	0x0	None
CORE_RST	SYS_CLK	0x0	None
DLL_RST	SYS_CLK	0x3	None
DPLL_IVA_PWRON_RST	SYS_CLK	0x0	None
DPLL_L3_INIT_PWRON_RET_RST	SYS_CLK	0x0	None
DPLL_MPU_PWRON_RST	SYS_CLK	0x0	None
DSS_RET_RST	SYS_CLK	0x0	None
DSS_RST	SYS_CLK	0x0	None
CORTEXM3_PWRON_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	MPU_M3_CLK clock is not active, RM_MPU_M3_RSTCTRL[2] RST3 is cleared, and automatic restore is complete.
CORTEXM3_RET_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	MPU_M3_CLK clock is not active and subsystem is reset.
CORTEXM3_RST1	SYS_CLK	ResetTime2 <sup>(1)</sup>	MPU_M3_CLK clock is not active and subsystem is reset.
CORTEXM3_RST2	SYS_CLK	ResetTime2 <sup>(1)</sup>	MPU_M3_CLK clock is not active.
CORTEXM3_RST3	SYS_CLK	ResetTime2 <sup>(1)</sup>	MPU_M3_CLK clock is not active and subsystem is reset.
EMIF_PWRON_RST	SYS_CLK	0x0	None
EMU_EARLY_PWRON_RST	SYS_CLK	0x2	None
EMU_PWRON_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	EMU_SYS_CLK clock is not active.
EMU_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	EMU_SYS_CLK clock is not active.

**Table 3-33. Reset Domains Attributes (continued)**

Reset Domain	RM Clock	RM Clock Count	Release Stall Conditions
SGX_RST	SYS_CLK	0x0	None
IVAHD_PWRON_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	IVAHD_ROOT_CLK clock is not active and <a href="#">RM_IVAHD_RSTCTRL[2]</a> RST3 is cleared.
IVAHD_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	IVAHD_ROOT_CLK clock is not active.
IVAHD_SEQ1_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	IVAHD_ROOT_CLK clock is not active and IVAHD and SL2 are not idle.
IVAHD_SEQ2_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	IVAHD_ROOT_CLK clock is not active and IVAHD and SL2 are not idle.
L3_INIT_PWRON_RST	SYS_CLK	0x0	None
L3_INIT_RET_RST	SYS_CLK	0x0	None
L3_INIT_RST	SYS_CLK	0x0	None
L4_PER_PWRON_RET_RST	SYS_CLK	0x0	None
L4_PER_RET_RST	SYS_CLK	0x0	None
L4_PER_RST	SYS_CLK	0x0	None
CORTEXA9_PWRON_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	MPU_DPLL_CLK clock is not active.
CORTEXA9_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	MPU_DPLL_CLK clock is not active, subsystem is reset, and automatic restore is complete.
SDMA_RET_RST	SYS_CLK	0x0	None
STD_EFUSE_PWRON_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	STDEFUSE_SYS_CLK clock is not active and logic voltages are in ON state.
DSP_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	DSP_ROOT_CLK clock is not active and subsystem is reset.
DSP_PWRON_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	DSP_ROOT_CLK clock is not active, <a href="#">RM_DSP_RSTCTRL[1]</a> RST2 is cleared, and automatic restore is complete.
DSP_RET_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	DSP_ROOT_CLK clock is not active and subsystem is reset.
DSP_SYS_RST	SYS_CLK	ResetTime2 <sup>(1)</sup>	DSP_ROOT_CLK clock is not active and subsystem is reset.
WKUP_PWRON_RST	SYS_CLK	0x0	None
WKUP_RST	SYS_CLK	0x0	None
WKUP_SYS_PWRON_RST	None	N/A	None

(1) ResetTime2 is set in the [PRM\\_RSTTIME\[14:10\]](#) RSTTIME2 bit field.

**NOTE:** WKUP\_SYS\_PWRON\_RST is directly connected to SYS\_PWRON\_RST source reset.

### 3.5.5 Reset Logging

A reset of the device is logged in two ways. First, dedicated registers in the PRM module (the [RM\\_<Power domain>\\_RSTST](#)) log the reset sources to power domain. Second, the Control Module logs the device reset activity in dedicated registers.

### 3.5.5.1 PRCM Module Reset Logging Mechanism

The reset status registers RM\_<power domain>\_RSTST and RM\_RSTST are reset asynchronously on assertion of a global cold reset. However, a reset status bit is always logged when the reset is released to the domain.

For this reason, after the assertion of a global cold reset, the reset status register is cleared to 0. When the domain reset is released, the register bit to log the global cold reset (the RM\_<Power domain>\_RSTST[0] GLOBALCOLD\_RST bit) is updated to 1. For the same reason, the reset status register of domains released from reset by software is updated only when software releases the domain reset.

The assertion of a global cold reset prevents logging any other source of reset until after the release of the domain reset. This is valid in the following situations:

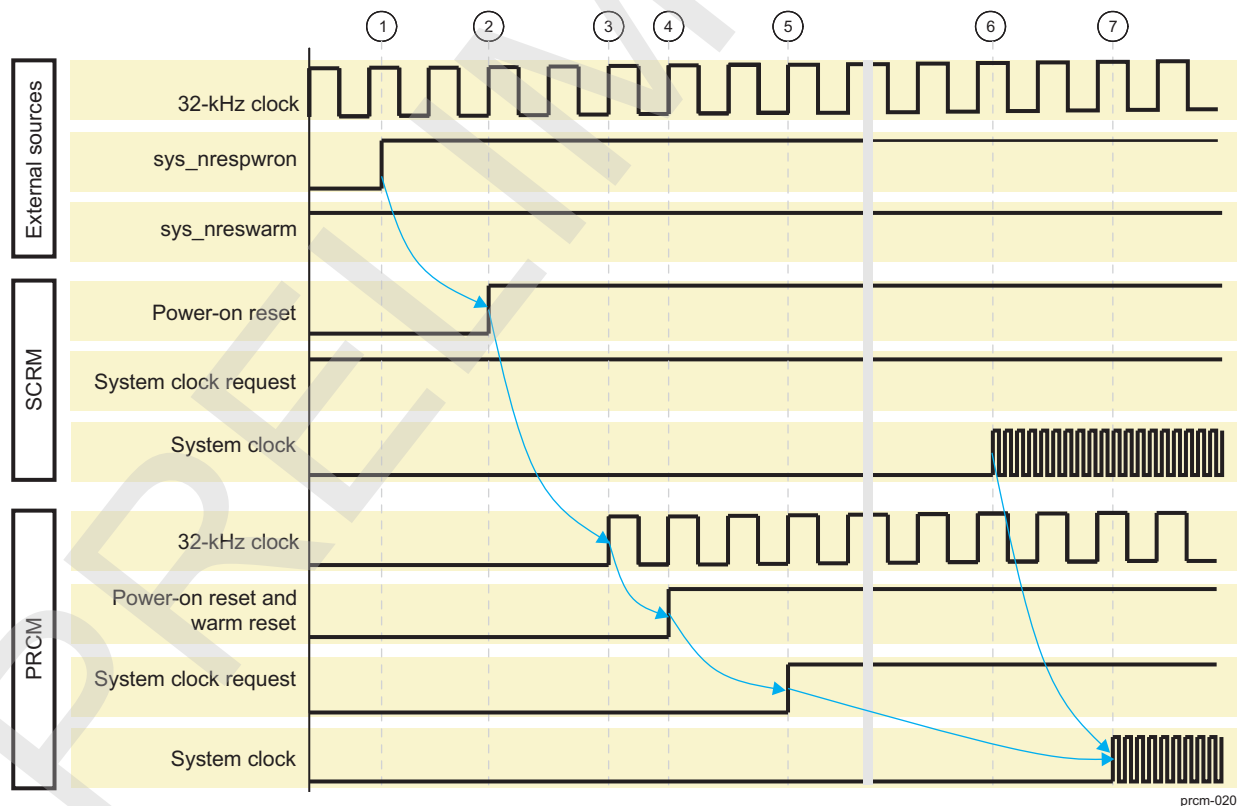
- A source of reset other than global cold reset is asserted before, during, or after the active period of a global cold source of reset and before the release of the domain reset signal.
- A source of reset other than global cold reset was asserted and then released, but a global cold reset source was asserted before the release of the domain reset signal.

### 3.5.6 Reset Sequences

#### 3.5.6.1 SCRM Power-on Reset Sequence

Figure 3-19 shows the power-on reset sequence as the reset is received by the SCRM and resulting interaction between SCRM and PRCM modules. Figure 3-20 shows the power-on reset sequence within the PRCM module.

Figure 3-19. SCRM Power-On Reset Sequence



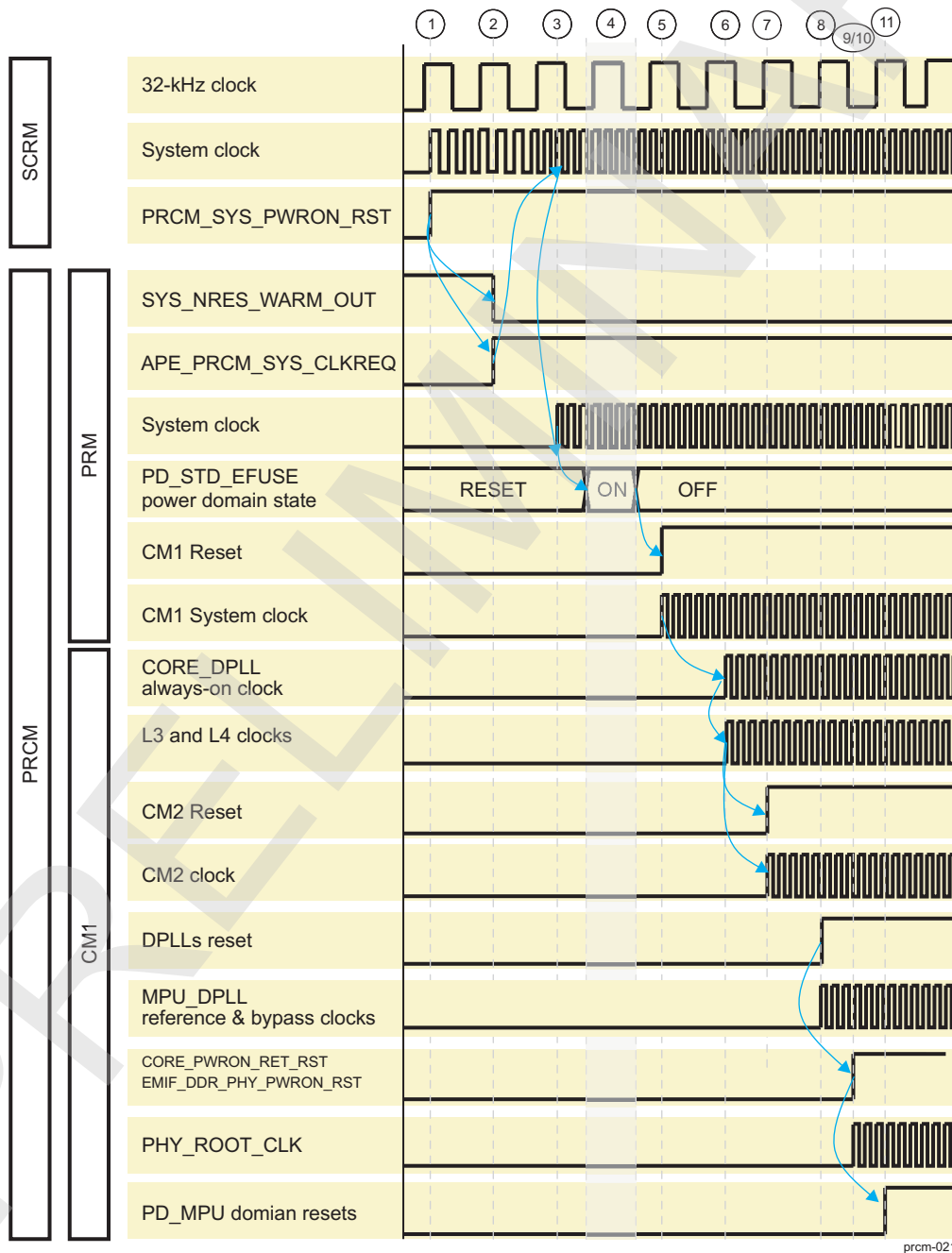
1. The power-on reset pad is released from reset while the 32 kHz is already active.
2. SCRM releases its internal power-on reset clocked at the 32-kHz clock.
3. SCRM delays the release of PRCM module resets to account for the stabilization time of the system clock source. It routes the 32-kHz clock to the PRCM module.

4. SCRM releases from reset (both power-on and warm reset) the PRCM module.
5. The PRCM module asserts a system clock request to the SCRM.
6. Once the system clock setup time counter overflows, the internal system clock version of SCRM is stable.
7. SCRM ungates the system clock to the PRCM module.

### 3.5.6.2 PRCM Module Power-On Reset Sequence

Figure 3-20 shows the power-on reset sequence.

**Figure 3-20. PRCM Module Power-On Reset Sequence**



The assumptions are:

- All the logic and memory voltage sources are at their nominal voltage levels.

- The SCRM module is supplying the 32-kHz clock to the PRCM module and the clock is stable.
- The power-on reset SYS\_PWRON\_RST\_IN has been released by the external power IC.

The power-up sequence is:

1. The SCRM releases global power-on reset to the PRCM module (that is, PRCM\_SYS\_PWRON\_RST). As a result, the PRCM module starts the power-on reset sequence.
2. The PRCM module releases the SYS\_NRES\_WARM\_OUT to the SCRM (only after all voltage domains have ramped-up) after some 32-kHz clock cycles, and it also asserts the APE\_PRCM\_SYS\_CLKREQ to the SCRM module to request the system clock.
3. SCRM provides the system clock to the PRCM module when the system clock source is stable.
4. Once the Standard eFuse controller clock has been enabled by the PRCM module, the PRM releases the standard eFuse controller domain from reset. This starts the eFuse sensing sequence. Upon completion of the eFuse auto load sequence, the PRM turns off the Standard eFuse controller clock. It also powers down the standard efuse controller power domain PD\_STD\_EFUSE.
5. The PRM logic is then released from reset. The PRM provides the CM1 with a version of the system clock and releases the CM1 from reset.
6. The CM1 starts the reference clock (and also bypass clock) of the CORE DPLL. In turn, the CORE DPLL, which is configured in bypass mode, generates the L3 clock (running at the system clock frequency at this stage). Now the CM1 logic is fully operational: CM1 logic is released from reset and L3 clock is running.
7. The CM1 starts the L4 clock version for the CM2 and the PRM deasserts the reset of the CM2.
8. The DPLLs are released from reset. The DPLLs can be released from reset before the L3 clock is running. The DPLL being under reset does not prevent the output clock from running. The CM1 starts the reference and bypass clocks to the DPLL\_MPU.
9. The CORE is released from reset, provided the L3 clock is running. The EMIF comes out of reset as soon as CORE power domain power on retention reset (CORE\_PWRON\_RET\_RST, see [Table 3-33](#)) is released by the PRCM module. The PRCM module also releases the EMIF\_DDR\_PHY\_PWRON\_RST. The EMIF DDR PHY X2 clock (that is, PHY\_ROOT\_CLK) is not running at this point, hence all the PHYs are asynchronously reset at the same time. The EMIF PHY clock is started later.
10. The PRCM module releases the idle request signal to the EMIF module once the EMIF functional clock is enabled. The EMIF then acknowledges the deassertion of idle request and becomes active.
11. The Cortex-A9 MPU subsystem is released from reset provided its clock is active and the L3 interconnect is operational.

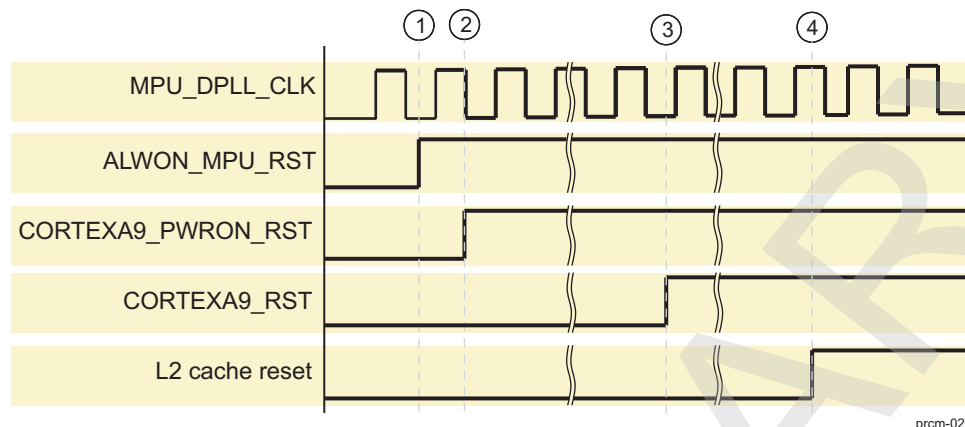
---

**NOTE:**

- PD\_DSP and PD\_IVAHD power domains are held under reset after power up by the assertion of software source of reset.
  - PD\_DSS, PD\_SGX, PD\_AUDIO, and PD\_CAM are held under reset after power up until the PRCM module enables their interface clock.
- 

### 3.5.6.3 Cortex-A9 MPU Subsystem Power-On Reset Sequence

[Figure 3-21](#) shows the power-on reset sequence of the Cortex-A9 MPU subsystem.

**Figure 3-21. Cortex-A9 MPU Power-On Reset Sequence**

prcm-022

The assumptions after power-on reset assertion are:

- The PRCM module provides the DPLL\_MPU reference clock and the bypass clock.
- The PRCM module has released DPLL\_MPU reset and the DPLL\_MPU is in bypass mode providing the clock (that is, bypass clock) to all the modules inside the Cortex-A9 MPU subsystem.

The power-on reset sequence is:

1. The PRCM module releases asynchronously in PD\_ALWON\_MPU the ALWON\_MPU\_RST reset to the CORTEXA9\_MPU\_INTC module inside the Cortex-A9 MPU subsystem.
2. The PRCM module releases in PD\_MPU the CORTEXA9\_PWRON\_RST (only after the MPU\_DPLL\_CLK is active) to the Cortex-A9 MPU subsystem and waits until the subsystem completes its internal reset sequence.
3. Once the Cortex-A9 MPU subsystem internal reset sequence completes, the PRCM module releases in PD\_MPU the CORTEXA9\_RST signal and the Cortex-A9 MPU starts booting.
4. The PRCM module releases the reset to L2 cache memory in the Cortex-A9 MPU subsystem 16 MPU\_DPLL\_CLK cycles after the release of CORTEXA9\_PWRON\_RST and CORTEXA9\_RST.

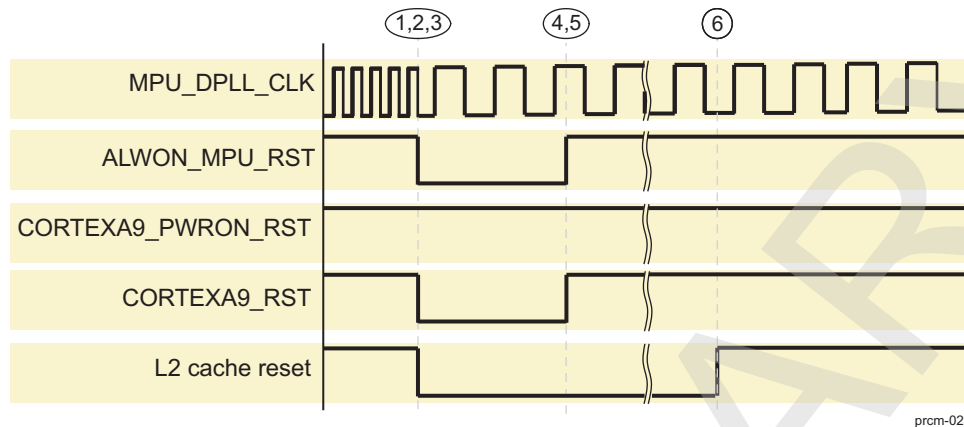
**NOTE:**

- The reset to the L2 cache memory in the Cortex-A9 MPU subsystem is asserted during initial power-on reset (that is, when the PD\_MPU wakes-up from OFF state). It is also asserted during local or global warm reset. However, it is not asserted when the PD\_MPU wakes-up from RETENTION state, that is (when the logic is off and L2 memory is in RETENTION state). This ensures that the L2 cache is retained on wake-up.
- The L1 cache memory in the Cortex-A9 MPU subsystem is not retained on PD\_MPU wake-up.

**3.5.6.4 Cortex-A9 MPU Subsystem Warm Reset Sequence**

Figure 3-22 shows the warm reset sequence of the Cortex-A9 MPU subsystem.

**Figure 3-22. Cortex-A9 MPU Warm Reset Sequence**



The assumptions are:

- The DPLL\_MPU is locked and providing the clock to the Cortex-A9 MPU subsystem.
- A global or local warm reset to the Cortex-A9 MPU subsystem is asserted.

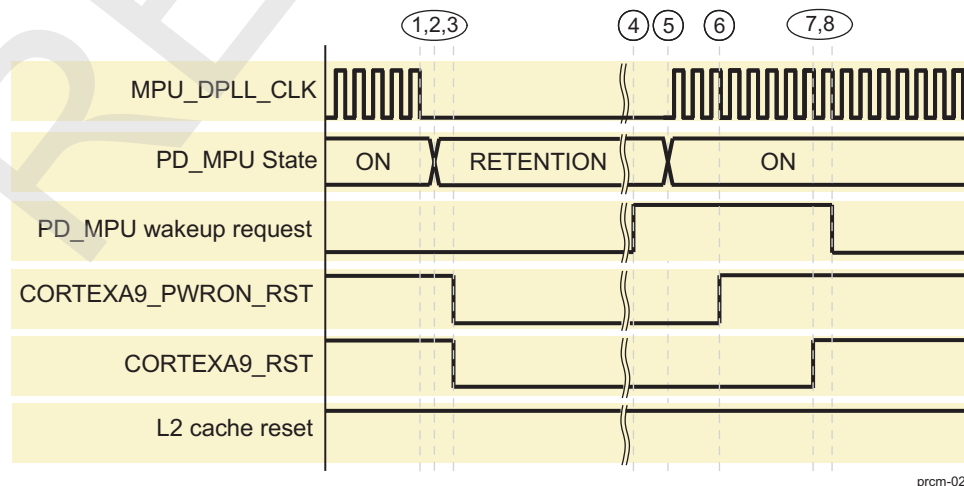
The warm reset sequence is:

1. The PRCM module asserts in PD\_ALWON\_MPU the ALWON\_MPU\_RST reset to the CORTEXA9\_MPU\_INTC module inside Cortex-A9 MPU subsystem.
2. The PRCM module asserts in PD\_MPU the CORTEXA9\_RST reset to the Cortex-A9 MPU subsystem.
3. The PRCM module resets the L2 cache memory in the Cortex-A9 MPU subsystem by asserting its reset.
4. The PRCM module releases the ALWON\_MPU\_RST reset to the CORTEXA9\_MPU\_INTC module.
5. The PRCM module releases the CORTEXA9\_RST, only after the DPLL\_MPU is in bypass mode and MPU\_DPLL\_CLK is stable and active.
6. The PRCM module releases the reset to L2 cache memory in the Cortex-A9 MPU subsystem 16 MPU\_DPLL\_CLK cycles after the release of CORTEXA9\_RST.

### 3.5.6.5 Cortex-A9 MPU Subsystem Reset Sequence On Sleep and Wake-Up Transitions From RETENTION or Off STATE

Figure 3-22 shows the sleep and wake-up transitions reset sequence from RETENTION or OFF state of the Cortex-A9 MPU subsystem.

**Figure 3-23. Cortex-A9 MPU Reset Sequence on Sleep and Wake-Up Transition**



The assumption is:

- The DPLL\_MPU is locked and providing the clock to the Cortex-A9 MPU subsystem.

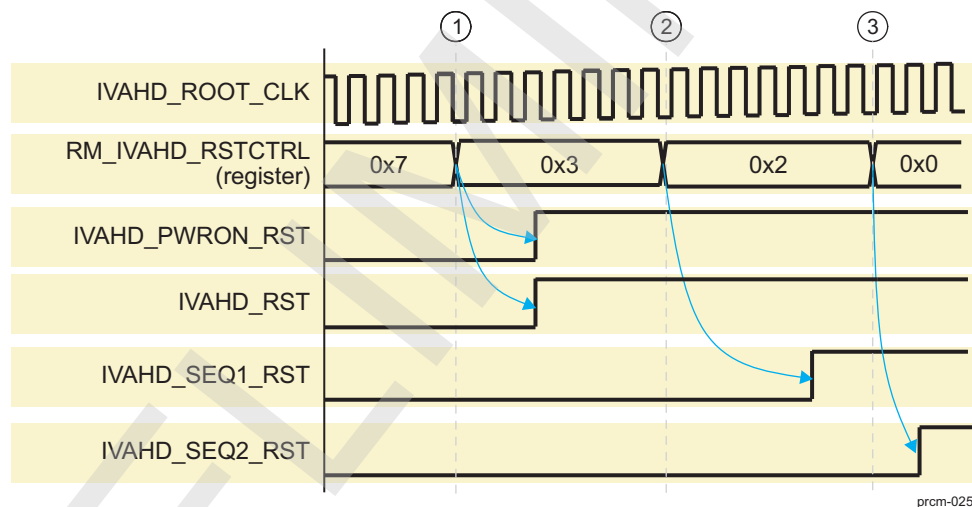
The sleep and wake-up transitions reset sequence is:

1. The PRCM module gates the MPU\_DPLL\_CLK to Cortex-A9 MPU subsystem.
2. The PRCM module switches the PD\_MPU to RETENTION state.
3. The PRCM module asserts CORTEXA9\_PWRON\_RST and CORTEXA9\_RST resets to Cortex-A9 MPU subsystem. The entire logic in the PD\_MPU is held in reset. The reset to the L2 cache memory in the Cortex-A9 MPU subsystem is not asserted if the logic in the PD\_MPU is held in reset.
4. The PRCM module resets the L2 cache memory in the Cortex-A9 MPU subsystem by asserting its reset.
5. The PRCM module releases the ALWON\_MPU\_RST reset to the CORTEXA9\_MPU\_INTC module.
6. The PRCM module releases the CORTEXA9\_RST, only after the DPLL\_MPU is in bypass mode and MPU\_DPLL\_CLK is stable and active.
7. The PRCM module releases the reset to L2 cache memory in the Cortex-A9 MPU subsystem 16 MPU\_DPLL\_CLK cycles after the release of CORTEXA9\_RST.

### 3.5.6.6 IVAHD Subsystem Power-On Reset Sequence

Figure 3-24 shows the power-on reset sequence of the IVAHD subsystem.

**Figure 3-24. IVAHD Power-On Reset Sequence**



The power-on reset to IVAHD is applied when PD\_IVAHD is powered. The assumptions after power-on reset assertion are:

- The PRCM module provides the functional clock IVAHD\_ROOT\_CLK to the IVAHD subsystem and it has been enabled by the software control.

The power-on reset sequence is:

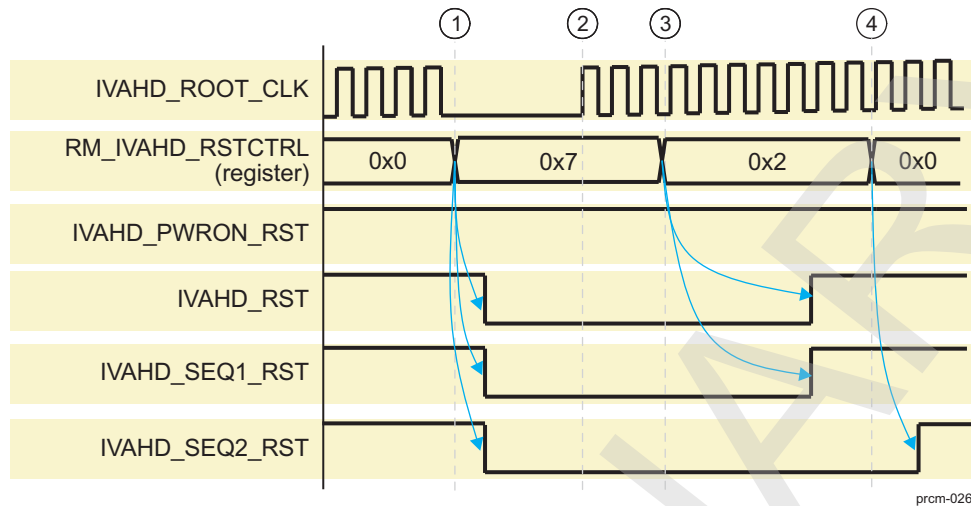
1. Software clears the [RM\\_IVAHD\\_RSTCTRL\[2\]](#) RST3 bit. This causes the PRCM module to release the IVAHD\_PWRON\_RST reset used inside IVAHD to reset mainly the emulation logic and the IVAHD\_RST reset used to reset all logic inside IVAHD. Then software can download data into TCM memory while keeping the sequencer CPUs under reset.
2. Software clears the [RM\\_IVAHD\\_RSTCTRL\[0\]](#) RST1 bit. This releases the IVAHD\_SEQ1\_RST reset to the Sequencer1 CPU.
3. Similarly, software can clear the [RM\\_IVAHD\\_RSTCTRL\[1\]](#) RST2 bit. This releases the IVAHD\_SEQ2\_RST reset to the Sequencer2 CPU.

### 3.5.6.7 IVAHD Subsystem Software Warm Reset Sequence

Figure 3-25 shows the software warm reset sequence of the IVAHD Subsystem.



Figure 3-25. IVAHD Software Warm Reset Sequence



Before asserting the software reset to the IVAHD subsystem the MPU software must ensure that:

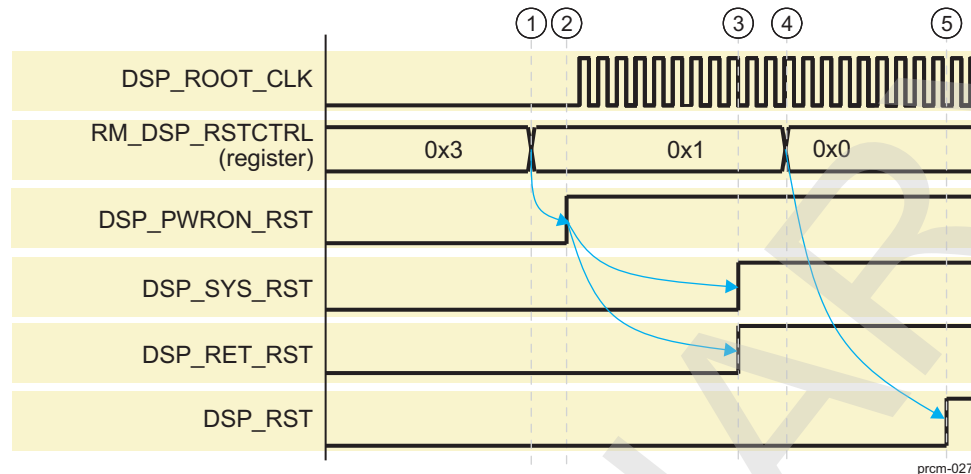
- IVAHD sequencer CPUs are in IDLE state ([CM\\_IVAHD\\_IVAHD\\_CLKCTRL\[17:16\]](#) IDLEST).
- The IVA HD subsystem is in STANDBY state ([CM\\_IVAHD\\_IVAHD\\_CLKCTRL\[18\]](#) STBYST).
- The functional clock to the IVAHD subsystem has been gated by the PRCM module ([CM\\_IVAHD\\_CLKSTCTRL\[8\]](#) CLKACTIVITY\_IVAHD\_CLK).

The software reset sequence is:

1. The MPU software sets the [RM\\_IVAHD\\_RSTCTRL\[2\]](#) RST3, [RM\\_IVAHD\\_RSTCTRL\[1\]](#) RST2, and [RM\\_IVAHD\\_RSTCTRL\[0\]](#) RST1 bits. This causes the PRCM module to assert the IVAHD\_RST, IVAHD\_SEQ1\_RST, and IVAHD\_SEQ2\_RST resets to the IVAHD subsystem. The IVAHD\_PWRON\_RST remains deasserted.
2. The MPU software enables the functional clock to the IVAHD subsystem.
3. The MPU software clears the [RM\\_IVAHD\\_RSTCTRL\[2\]](#) RST3 and [RM\\_IVAHD\\_RSTCTRL\[0\]](#) RST1 bits. This causes the PRCM module to release the IVAHD\_RST and IVAHD\_SEQ1\_RST resets to the IVAHD subsystem.
4. The MPU software clears the [RM\\_IVAHD\\_RSTCTRL\[1\]](#) RST2 bit. This releases the IVAHD\_SEQ2\_RST reset to the Sequencer2 CPU.

### 3.5.6.8 DSP Subsystem Power-On Reset Sequence

Figure 3-26 shows the power-on reset sequence of the DSP subsystem.

**Figure 3-26. DSP Subsystem Power-On Reset Sequence**

The assumptions on power-on reset assertion are:

- The PD\_DSP is on.
- The MPU software sets the [CM\\_DSP\\_DSP\\_CLKCTRL\[1:0\]](#) MODULEMODE bit field to Auto.
- The MPU software sets the [CM\\_DSP\\_CLKSTCTRL\[1:0\]](#) CLKTRCTRL bit field to SW\_WKUP or HW\_AUTO.
- The DPLL\_IVA is active and providing the DSP\_ROOT\_CLK. It can be locked prior to this sequence.

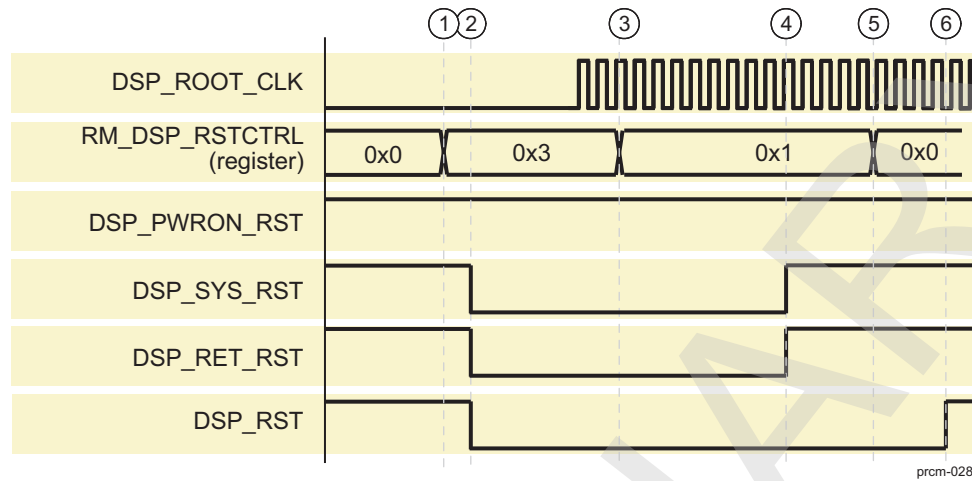
The power-on reset sequence is:

1. The MPU software clears the [RM\\_DSP\\_RSTCTRL\[1\]](#) RST2 bit in the PRCM module register to release from reset the DSP, MMU, and CACHE interface.
2. The PRCM module releases DSP\_PWRON\_RST once the reset manager counter ([PRM\\_RSTTIME\[14:10\]](#) RSTTIME2) has reached its limit. Upon the deassertion of power-on reset signal, the DSP subsystem starts the initialization sequence. During this initialization sequence all the internal registers inside DSP are properly reset and it also completes the reset for DSP, MMU, and CACHE interface.
3. Once the reset sequence of Step 2 completes and the reset manager counter ([PRM\\_RSTTIME\[14:10\]](#) RSTTIME2) has expired, the PRCM module releases DSP\_SYS\_RST and DSP\_RET\_RST signals.
4. The MPU software must configure the MMU once the MMU is out of reset. After MMU configuration is done, the MPU software clears the [RM\\_DSP\\_RSTCTRL\[0\]](#) RST1 bit in the PRCM module register.
5. The PRCM module releases DSP\_RST, which causes the DSP to start booting.

### 3.5.6.9 DSP Subsystem Software Warm Reset Sequence

[Figure 3-27](#) shows the software warm reset sequence of the DSP subsystem.

**Figure 3-27. DSP Subsystem Software Warm Reset Sequence**



The assumptions on power-on reset assertion are:

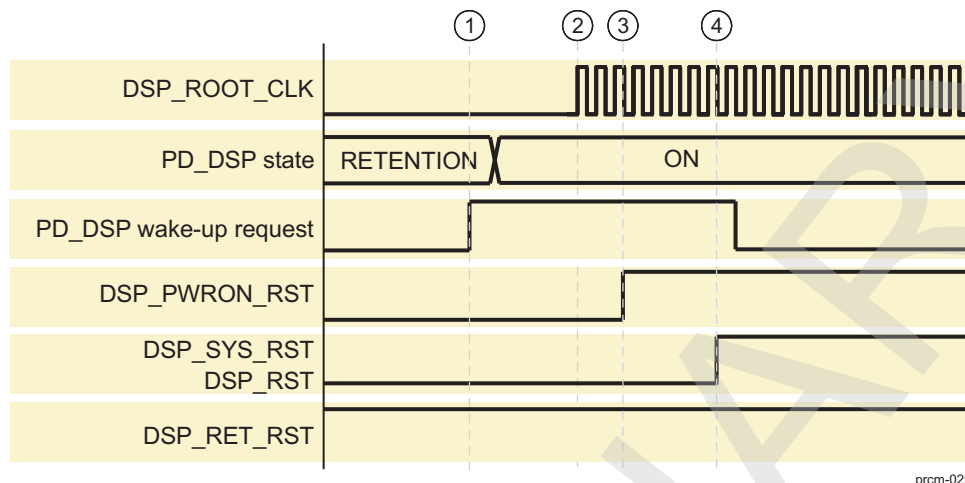
- The DSP is in IDLE state.
- The DSP\_ROOT\_CLK is gated.

The power-on reset sequence is:

1. The MPU software sets the [RM\\_DSP\\_RSTCTRL\[1\]](#) RST2 and [RM\\_DSP\\_RSTCTRL\[0\]](#) RST1 bits.
2. The PRCM module asserts the DSP\_SYS\_RST, DSP\_RET\_RST, and DSP\_RST reset signals. DSP\_PWRON\_RST remains deasserted in this case.
3. The MPU software re-enables the DSP\_ROOT\_CLK and clears the [RM\\_DSP\\_RSTCTRL\[1\]](#) RST2 bit in the PRCM module register to reset DSP, MMU, and CACHE interface. The DSP subsystem starts the partial initialization sequence for the warm reset.
4. Once the reset sequence of Step 3 completes and the reset manager counter ([PRM\\_RSTTIME\[14:10\]](#) RSTTIME2) has expired, the PRCM module releases the DSP\_SYS\_RST and DSP\_RET\_RST signals.
5. The MPU software clears the [RM\\_DSP\\_RSTCTRL\[0\]](#) RST1 bit in the PRCM module register.
6. The PRCM module releases DSP\_RST, which causes the DSP to start booting.

### 3.5.6.10 DSP Subsystem Reset Sequence On Wake-Up Transitions From OSWR RETENTION State

Figure 3-28 shows the reset sequence of the DSP subsystem on wake-up transitions from OSWR RETENTION state.

**Figure 3-28. DSP Subsystem Reset Sequence On Wake-Up Transitions From OSWR RETENTION State**

The assumptions on power-on reset assertion are:

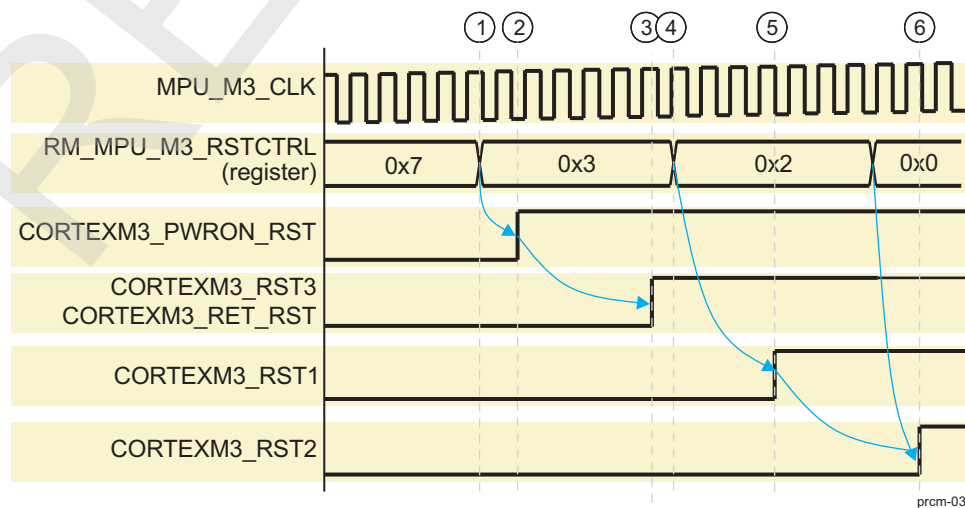
- The DSP is in INACTIVE state.
- The DSP\_ROOT\_CLK is gated.
- The PD\_DSP is in OSWR RETENTION state.
- The DSP\_PWRON\_RST, DSP\_SYS\_RST, and DSP\_RST resets are held asserted, while DSP\_RET\_RST is not asserted by the PRCM module.

The power-on reset sequence is:

1. A wake-up request is received by the PRCM module and it switches the PD\_DSP to ON state
2. The PRCM module then enables the DSP\_ROOT\_CLK and switches the CD\_DSP to ACTIVE state.
3. The PRCM module releases DSP\_PWRON\_RST once the reset manager counter ([PRM\\_RSTTIME\[14:10\] RSTTIME2](#)) has reached its limit. Upon the deassertion of the power-on reset signal, the DSP subsystem starts the initialization sequence.
4. Once the reset sequence of Step 3 completes the PRCM module releases the DSP\_SYS\_RST and DSP\_RST signals.

### 3.5.6.11 Cortex-M3 MPU Subsystem Power-On Reset Sequence

Figure 3-29 shows the power-on reset sequence of the Cortex-M3 MPU subsystem.

**Figure 3-29. Cortex-M3 MPU Power-On Reset Sequence**

The assumptions on power-on reset assertion are:

- The Cortex-M3 MPU subsystem is held in reset by the PRCM module and the CORTEXM3\_PWRON\_RST, CORTEXM3\_RET\_RST, CORTEXM3\_RST1, CORTEXM3\_RST2, and CORTEXM3\_RST3 are asserted.

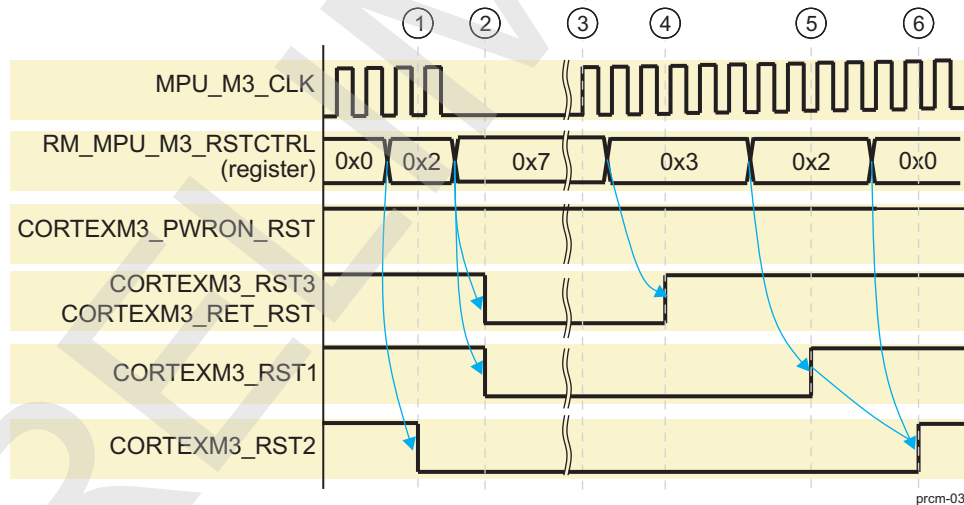
The power-on reset sequence is:

1. Software clears the `RM_MPU_M3_RSTCTRL[2]` RST3 bit in the PRCM module register to release from reset the MPU M3 Cache and MMU.
2. The PRCM module releases `CORTEXM3_PWRON_RST` once the reset manager counter (`PRM_RSTTIME[14:10]` RSTTIME2) has reached its limit and the `MPU_M3_CLK` is running. Upon the deassertion of the power-on reset signal, the Cortex-M3 MPU subsystem starts the CPU and MMU initialization sequence.
3. Once the reset sequence of Step 2 completes and the reset manager counter (`PRM_RSTTIME[14:10]` RSTTIME2) has expired, the PRCM module releases the `CORTEXM3_RST3` and `CORTEXM3_RET_RST` signals.
4. The MPU software must configure the MMU once the MMU is out of reset. After MMU configuration and cache initialization is done, the MPU software clears the `RM_MPU_M3_RSTCTRL[0]` RST1 bit in the PRCM module register.
5. The PRCM module releases `CORTEXM3_RST1`, which causes the Cortex-M3 MPU Core 1 to start booting.
6. The MPU software can clear the `RM_MPU_M3_RSTCTRL[1]` RST2 bit in the PRCM module register so that the PRCM module releases the `CORTEXM3_RST2` to the Cortex-M3 MPU Core 2.

### 3.5.6.12 Cortex-M3 MPU Subsystem Software Warm Reset Sequence

Figure 3-30 shows the software warm reset sequence of the Cortex-M3 MPU subsystem.

Figure 3-30. Cortex-M3 MPU Subsystem Software Warm Reset Sequence



The assumptions on power-on reset assertion are:

- The Cortex-M3 MPU Core 2 is in Idle state.

The power-on reset sequence is:

1. The Cortex-M3 MPU Core 1 software or Cortex-A9 MPU software sets the `RM_MPU_M3_RSTCTRL[1]` RST2 bit. The PRCM module asserts the `CORTEXM3_RST2` reset signal to the Cortex-M3 Core 2.
2. The Cortex-A9 MPU cuts the `MPU_M3_CLK` and software sets the `RM_MPU_M3_RSTCTRL[2]` RST3 and `RM_MPU_M3_RSTCTRL[0]` RST1 bits.
3. The PRCM module asserts the `CORTEXM3_RST3`, `CORTEXM3_RET_RST`, and `CORTEXM3_RST1` reset signals. The `CORTEXM3_PWRON_RST` remains deasserted in this case.
4. The MPU software re-enables the `MPU_M3_CLK` and the initialization sequence starts inside Cortex-M3 subsystem. Software clears the `RM_MPU_M3_RSTCTRL[2]` RST3 bit in the PRCM module

register.

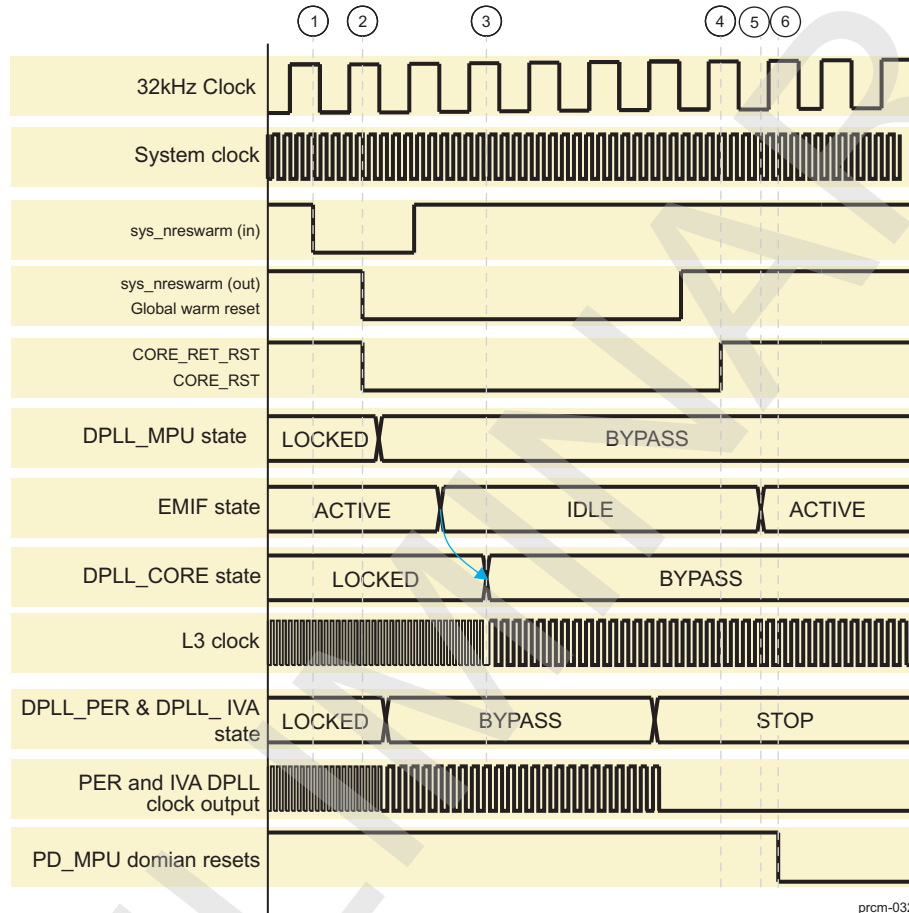
5. Once the reset sequence of Step 4 completes and the reset manager counter ([PRM\\_RSTTIME\[14:10\]](#) RSTTIME2) has expired, the PRCM module releases CORTEXM3\_RST3, CORTEXM3\_RET\_RST, and CORTEXM3\_RST1 reset signals. The Cortex-M3 MPU Core 1 starts rebooting.
6. Software can then clear the [RM\\_MPU\\_M3\\_RSTCTRL\[1\]](#) RST2 bit in the PRCM module register.. The PRCM module releases the CORTEXM3\_RST2 reset signal to the Cortex-M3 MPU Core 2 to start booting.

### 3.5.6.13 Global Warm Reset Sequence

This section describes the global warm reset sequence.

Figure 3-31 shows the global warm reset sequence.

Figure 3-31. Global Warm Reset Sequence



The assumptions are:

- All the logic and memory voltage sources are at their nominal voltage levels.
- The device is active:
  - All resets are released.
  - MPU, CORE, and DSP DPLLs are locked.

The steps of a global warm reset sequence are:

1. On assertion of any global warm reset source the PRM signals the EMIF that a global warm reset event has occurred. The EMIF initiates IDLE state transition. The PRCM module delays global warm reset to the device for minimum 16 L3 clock cycles so that the EMIF switches to IDLE state and switches the external SDRAM memory to self-refresh mode.
2. The reset managers in PRM assert the following resets:
  - The external warm reset SYS\_WARM\_RST (output) to the SCRM.
  - All power domain warm resets are asserted.
  - DPLL resets are not asserted.
    - DPLL\_MPU transitions to bypass mode.
    - DPLL\_CORE transitions to bypass mode once the EMIF has switched to IDLE state.
    - DPLL\_IVA, DPLL\_PER, and DPLL\_ABE transition to low-power stop mode.
  - The system clock remains active.

- The PRM and CM registers, sensitive to warm reset, are asynchronously reset.
  - CM gates the clocks not needed, as per their default reset setting in the associated registers.
3. After the warm reset source is released, the global warm reset and extended until the following conditions are met:
- Device reset manager counter overflows. It is set up by the `PRM_RSTTIME[9:0]` RSTTIME1 bit field.
  - All logic voltage sources (VDD\_IVA and VDD\_CORE) are stable.

---

**NOTE:** Voltage stabilization is an additional condition if voltage scaling was performed before the assertion of the warm reset.

---

4. The CORE power domain is released from reset (that is, warm reset sensitive modules in the CORE power domain).
5. The PRCM module switches the EMIF back to ACTIVE state from IDLE state.
6. PD\_MPU is released from reset when the clocks to the Cortex-A9 MPU subsystem are active. The Cortex-A9 MPU reboots.

---

**NOTE:**

- PD\_DSP and PD\_IVAHD power domains are held under reset after global warm reset by assertion of the software source of the reset.
  - PD\_L4\_PER, PD\_L3\_INIT, PD\_DSS, PD\_SGX, and PD\_CAM are held under reset after global warm reset until the PRCM module enables their interface clock.
  - The reset to the standard eFuse power domain, PD\_STD\_EFUSE, is not asserted on a global warm reset.
-



## 3.6 Clock Management Functional Description

### 3.6.1 Overview

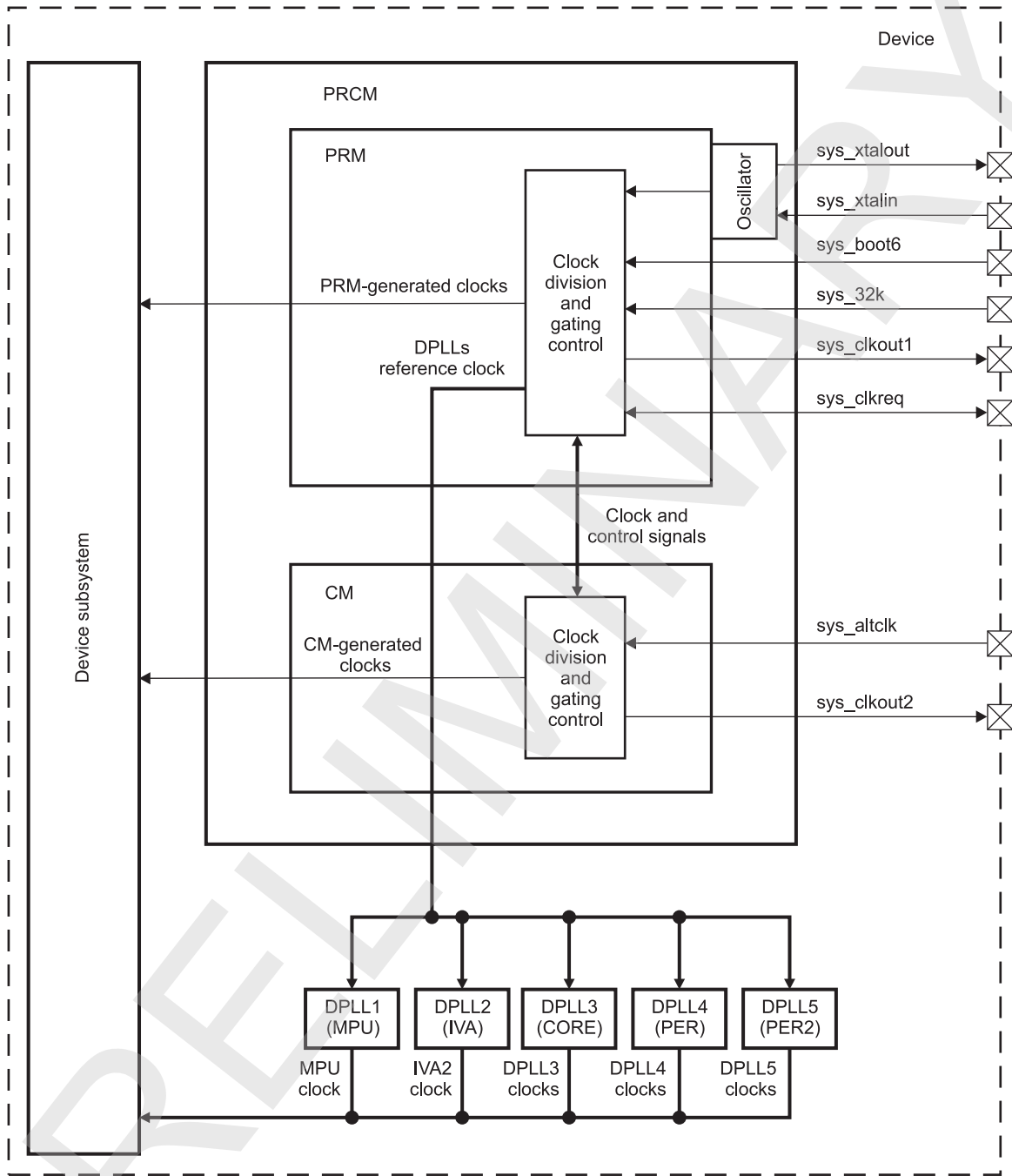
The PRCM module provides control for clock generation, division, distribution, synchronization, and gating. It distributes the clock sources to all modules in the device. See [Section 3.1.1.1, \*Clock Management\*](#), for information on the clock-management functional architecture of the device.

The high-frequency system clock is received by the device from an external source or generated within the device (internal oscillator) by the SCRM. The SCRM also receives the 32-kHz low-frequency clock from the external source.

The stable versions of the system and 32-kHz clock are provided by the SCRM to the PRCM module. The PRCM module provides clocks to the internal DPLLs for internal high-frequency clocks generation. Clock division and gating are handled by the PRM, the CM1 and CM2 sections of the PRCM module.

[Figure 3-32](#) shows the high-level clock-management scheme in the device.

Figure 3-32. PRCM Module Clock Manager Overview



prcm-078

### **3.6.2 External Clock Inputs**

#### **3.6.2.1 sys\_32k Clock Input**

The 32-kHz frequency is used for low-frequency operation (timer, debouncing, etc.). It supplies the wake-up domain for operation in lowest power mode and as clock source to the DPLL\_ABE module (for audio back-end [ABE]).

#### **3.6.2.2 High-Frequency System Clock Input**

The system clock, SYS\_CLK, is the main source clock of the device. SYS\_CLK is received directly from the SCRM by the PRM subsection of the PRCM module. It is supplied as the reference clock to the DPLLs as well as functional clock to several modules.

### **3.6.3 Internal Clock Sources/Generators**

The PRCM module clock sources/generators are split into the following parts:

- PRM clock source that receives the 32-kHz and system clocks from SCRM
- CM1 and CM2 clock sources that distribute high-frequency clocks
- DPLL clock generators that synthesize high-frequency clocks for the device

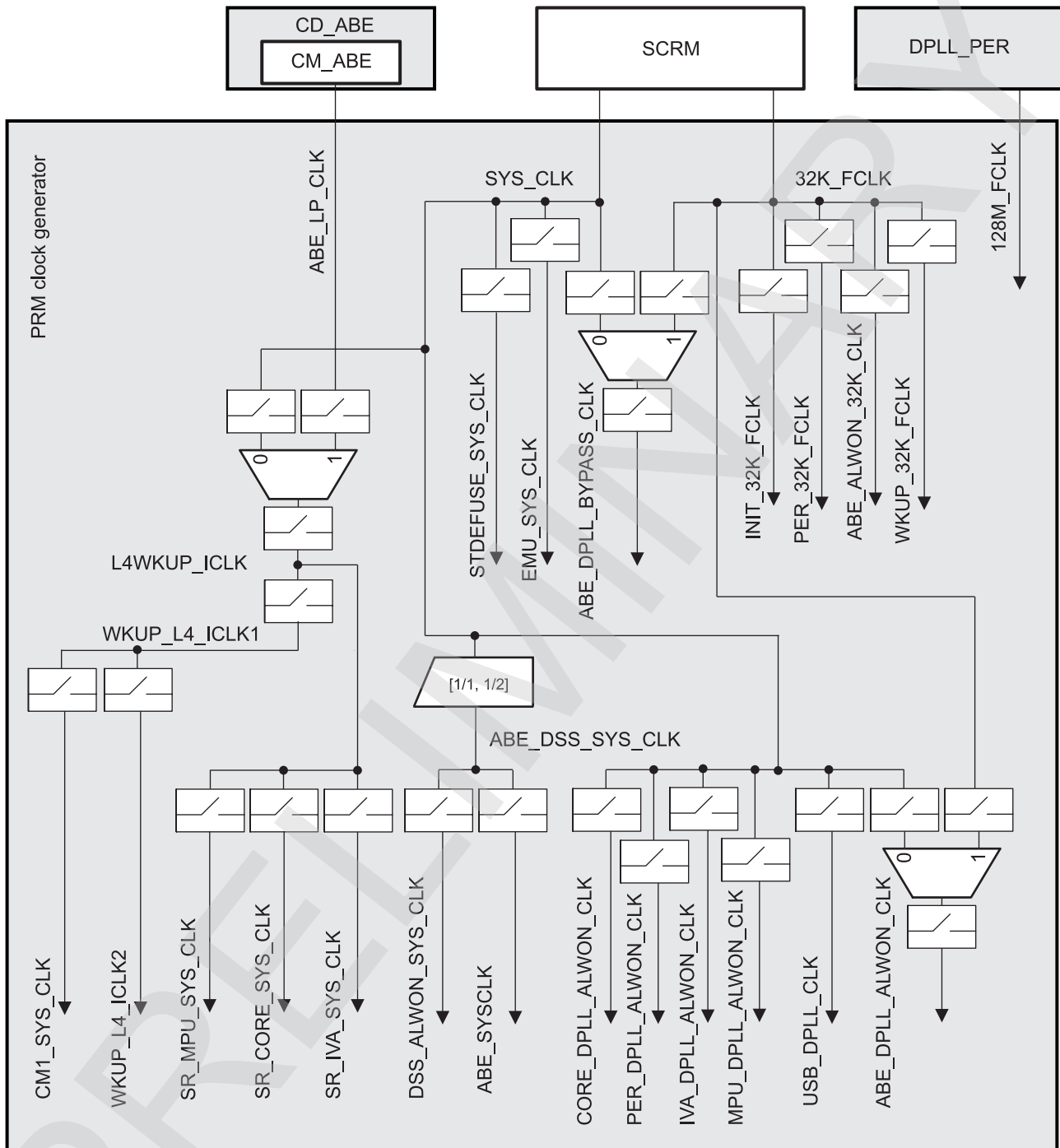
#### **3.6.3.1 PRM Clock Source**

The PRM clock source receives the SYS\_32K and SYS\_CLK clocks from the SCRM. It sources various versions (through gating controls) of these clocks to supply:

- PRCM-managed DPLLs with a reference clock, which is permanently supplied with always-on buffers
- The SR modules with a reference clock, which is permanently supplied with always-on buffers
- The DSS with a reference clock, which is permanently supplied with always-on buffers
- eFuse controller clocks (gated SYS\_CLK clock), which are permanently supplied with always-on buffers
- The clocks for CM clock generator and the CORE power domain
- A version of SYS\_CLK to ABE supplied by always-on buffers

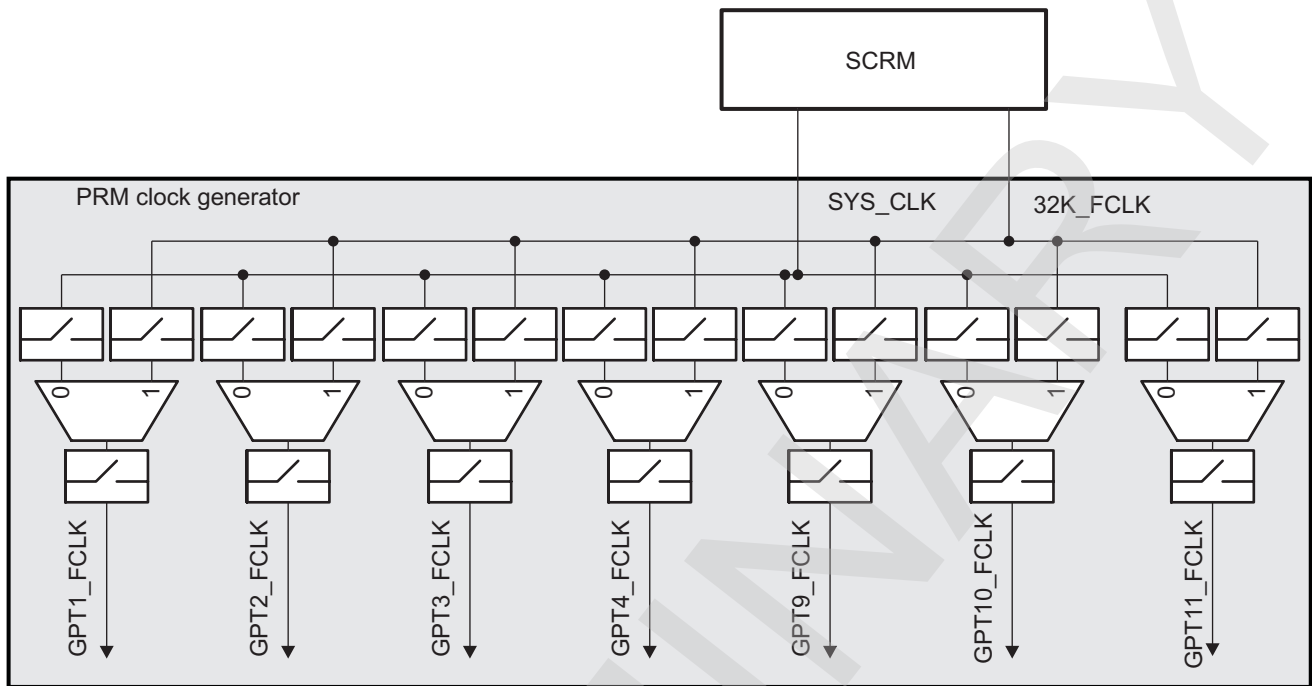
[Figure 3-33](#) and [Figure 3-34](#) present a logical representation of the PRM clock source.

Figure 3-33. PRM Clock Manager Overview (a)



prcm-034

Figure 3-34. PRM Clock Manager Overview (b)



prcm-035

Table 3-34 identifies controls for clock dividers or muxes in the PRM.

Table 3-34. PRM Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux ABE_DPLL_ALWON_CLK	CM_ABE_PLL_REF_CLKSEL[0] CLKSEL
Mux GPT1_FCLK	CM_WKUP_GPTIMER1_CLKCTRL[24] CLKSEL
Mux GPT2_FCLK	CM_L4PER_GPTIMER2_CLKCTRL[24] CLKSEL
Mux GPT3_FCLK	CM_L4PER_GPTIMER3_CLKCTRL[24] CLKSEL
Mux GPT4_FCLK	CM_L4PER_GPTIMER4_CLKCTRL[24] CLKSEL
Mux GPT9_FCLK	CM_L4PER_GPTIMER9_CLKCTRL[24] CLKSEL
Mux GPT10_FCLK	CM_L4PER_GPTIMER10_CLKCTRL[24] CLKSEL
Mux GPT11_FCLK	CM_L4PER_GPTIMER11_CLKCTRL[24] CLKSEL
Mux L4WKUP_ICLK	CM_L4_WKUP_CLKSEL[0] CLKSEL
Mux ABE_DPLL_BYPASS_CLK	CM_L4_WKUP_CLKSEL[0] CLKSEL

PRM provides 32-kHz gated and ungated clock for use by portions of PD\_WKUP power domain and some peripherals outside the PD\_WKUP power domain.

It also provides the system clock to the display subsystem (DSS), and gated and buffered version to L4WKUP\_ICLK interconnect clock, and the DPLLs controlled by the PRCM module. The gated version of the SYS\_CLK is also provided to the SmartReflex modules of the device.

### 3.6.3.2 CM Clock Source

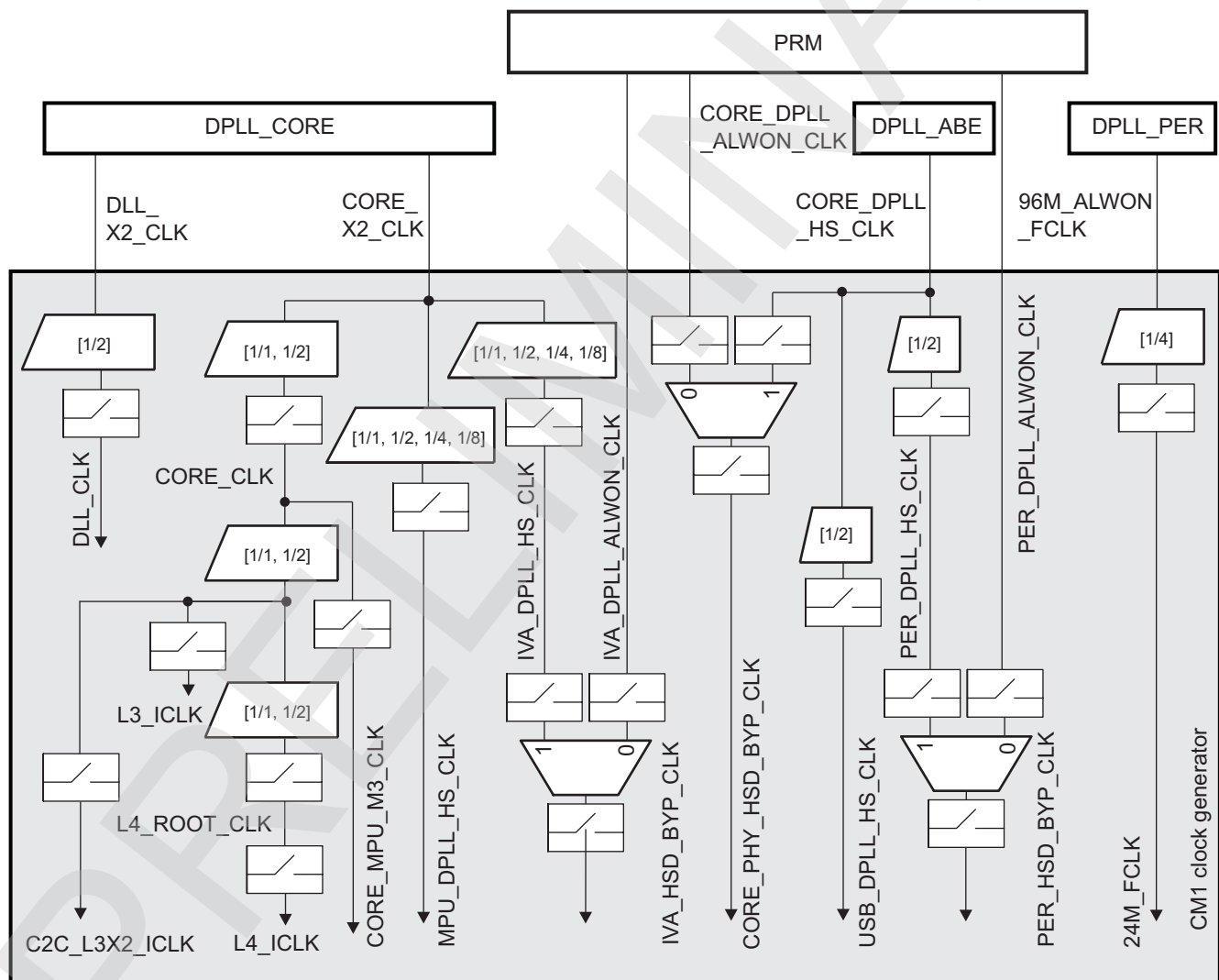
The clock manager (CM) is primarily responsible for generating interface and functional clocks from the internal clocks provided by DPLL\_CORE and DPLL\_PER. The CM is physically divided into two independent entities, CM1 placed in PD\_ALWON\_CORE always-on power domain and CM2 placed in PD\_CORE switchable power domain. The split is done to provide control over various entities like modules, DPLLs and clocks during low power use case scenarios when the PD\_CORE power domain can be switched to RETENTION state.

#### 3.6.3.2.1 CM1 Clock Generator

CM1 receives system clock from the PRM, which serves as its functional clock. CM1 provides gated clock to PD\_AUDIO and some DPLLs. It also provides clock to the L3 and L4 interconnects in the device.

Figure 3-35 shows the various functional and interface clocks generated by CM1:

Figure 3-35. CM1 Overview



prcm-036

Table 3-35 identifies controls for clock dividers or muxes in the CM1 clock source.

**Table 3-35. CM1 Clock Division and Muxing Control**

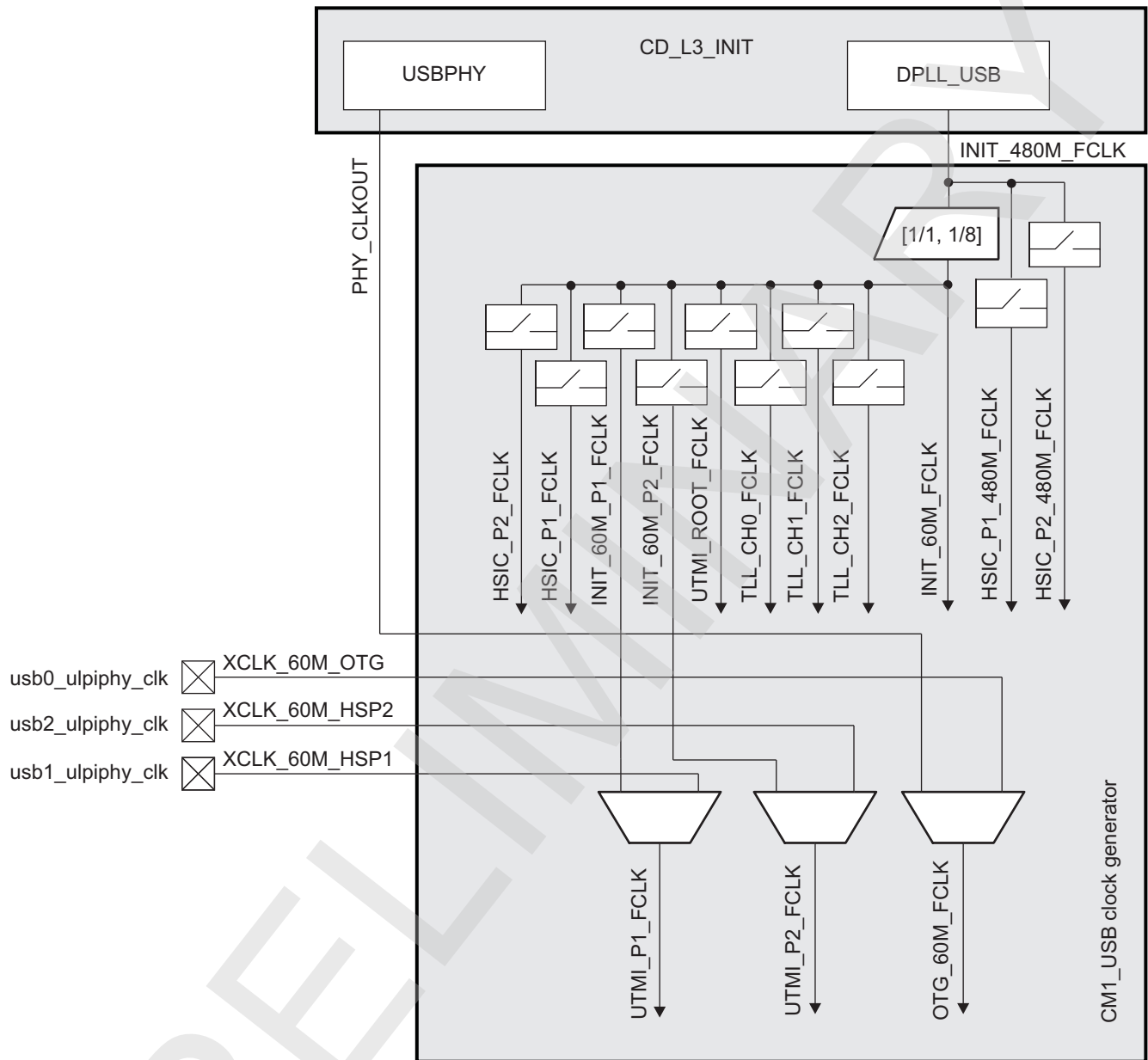
Divider/Mux	Control Bit Field
Divider CORE_CLK	CM_CLKSEL_CORE[0] CLKSEL_CORE
Divider L3_ICLK	CM_CLKSEL_CORE[4] CLKSEL_L3
Divider L4_ROOT_CLK	CM_CLKSEL_CORE[8] CLKSEL_L4
Divider MPU_DPLL_HS_CLK	CM_BYPCLK_DPLL_MPU[1:0] CLKSEL
Divider IVA_DPLL_HS_CLK	CM_BYPCLK_DPLL_IVA[1:0] CLKSEL
Mux IVA_HSD_BYP_CLK	CM_CLKSEL_DPLL_IVA[23] DPLL_BYP_CLKSEL
Mux PER_HSD_BYP_CLK	CM_CLKSEL_DPLL_PER[23] DPLL_BYP_CLKSEL
Mux CORE_PHY_HSD_BYP_CLK	CM_CLKSEL_DPLL_CORE[23] DPLL_BYP_CLKSEL

**3.6.3.2.2 CM1\_USB Clock Generator**

CM\_USB receives system clock from USBPHY and DPLL\_USB. It provides gated clock to CD\_L3\_INIT.

Figure 3-36 shows the various functional and interface clocks generated by CM1\_USB:

**Figure 3-36. CM1\_USB Clock Manager Overview**



prcm-037

Table 3-36 identifies controls for clock dividers or muxes in the CM1\_USB clock source.

**Table 3-36. CM1\_USB Clock Division and Muxing Control**

Divider/Mux	Control Bit Field
Divider INIT_60M_FCLK	CM_CLKSEL_USB_60MHZ[0] CLKSEL
Mux UTMI_P1_FCLK	CM_L3INIT_HSUSBHOST_CLKCTRL[24] CLKSEL_U TMI_ P1
Mux UTMI_P2_FCLK	CM_L3INIT_HSUSBHOST_CLKCTRL[25] CLKSEL_U TMI_ P2
Mux OTG_60M_FCLK	CM_L3INIT_HSUSBOTG_CLKCTRL[24] CLKSEL_6 0M



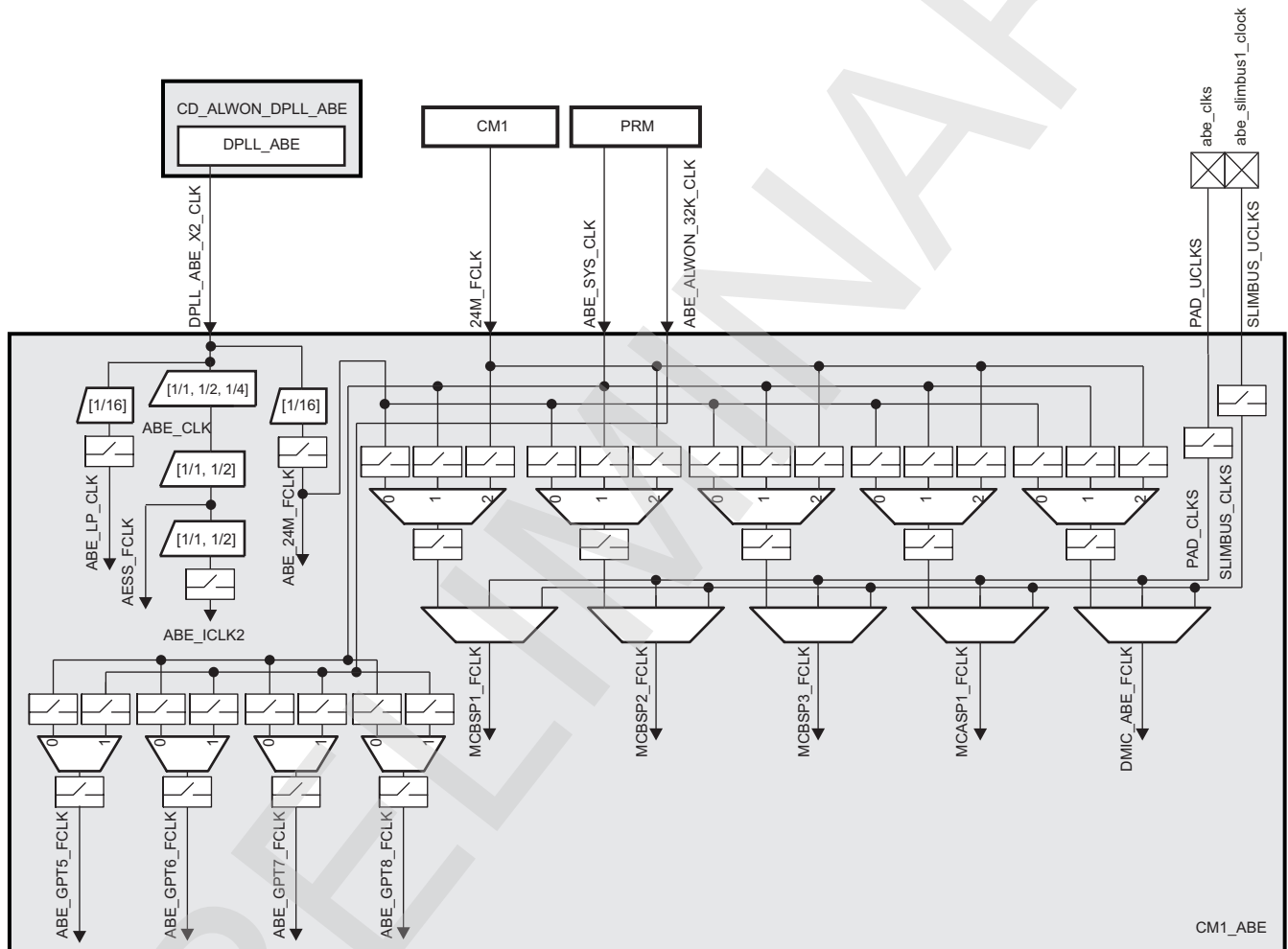
**NOTE:** For the clock status bits see [Table 3-117](#)

### 3.6.3.2.3 CM1\_ABE Clock Generator

CM1\_ABE receives system clock from PRM. It provides gated clock to CD\_L4\_PER.

Figure 3-37 show the various functional and interface clocks generated by CM1\_ABE clock source.

**Figure 3-37. CM1\_ABE Clock Manager Overview**



prcm-038

Table 3-37 identifies controls for clock dividers or muxes in the CM1\_ABE.

**Table 3-37. CM1\_ABE Clock Division and Muxing Control**

Divider/Mux	Control Bit Field
Clock switch PAD_CLKS	CM_CLKSEL_ABE[8] PAD_CLKS_GAT E
Clock switch SLIMBUS_CLKS	CM_CLKSEL_ABE[10] SLIMBUS_CLK_GATE
Divider ABE_CLK	CM_CLKSEL_ABE[1:0] CLKSEL_OPP
Divider AESS_FCLK and ABE_ICLK2 <sup>(1)</sup>	CM1_ABE_AESS_CLKCTRL[24] CLKSEL_AESS_FCLK
Mux ABE_GPT5_FCLK	CM1_ABE_GPTIMER5_CLKCTRL[24] CLKSEL
Mux ABE_GPT6_FCLK	CM1_ABE_GPTIMER6_CLKCTRL[24] CLKSEL
Mux ABE_GPT7_FCLK	CM1_ABE_GPTIMER7_CLKCTRL[24] CLKSEL

**Table 3-37. CM1\_ABE Clock Division and Muxing Control (continued)**

Divider/Mux	Control Bit Field
Mux ABE_GPT8_FCLK	CM1_ABE_GPTIMER8_CLKCTRL[24] CLKSEL
Mux DMIC_FCLK	CM1_ABE_DMIC_CLKCTRL[25:24] CLKSEL_SOURCE
Mux DMIC_INT_FCLK	CM1_ABE_DMIC_CLKCTRL[27:26] CLKSEL_INTERNAL_SOURCE
Mux MCASP_INT_FCLK	CM1_ABE_MCASP_CLKCTRL[27:26] CLKSEL_INTERNAL_SOURCE
Mux MCASP_FCLK	CM1_ABE_MCASP_CLKCTRL[25:24] CLKSEL_SOURCE
Mux MCBSP1_FCLK	CM1_ABE_MCBSP1_CLKCTRL[25:24] CLKSEL_SOURCE
Mux MCBSP1_INT_FCLK	CM1_ABE_MCBSP1_CLKCTRL[27:26] CLKSEL_INTERNAL_SOURCE
Mux MCBSP2_FCLK	CM1_ABE_MCBSP2_CLKCTRL[25:24] CLKSEL_SOURCE
Mux MCBSP2_INT_FCLK	CM1_ABE_MCBSP2_CLKCTRL[27:26] CLKSEL_INTERNAL_SOURCE
Mux MCBSP3_FCLK	CM1_ABE_MCBSP3_CLKCTRL[25:24] CLKSEL_SOURCE
Mux MCBSP3_INT_FCLK	CM1_ABE_MCBSP3_CLKCTRL[27:26] CLKSEL_INTERNAL_SOURCE

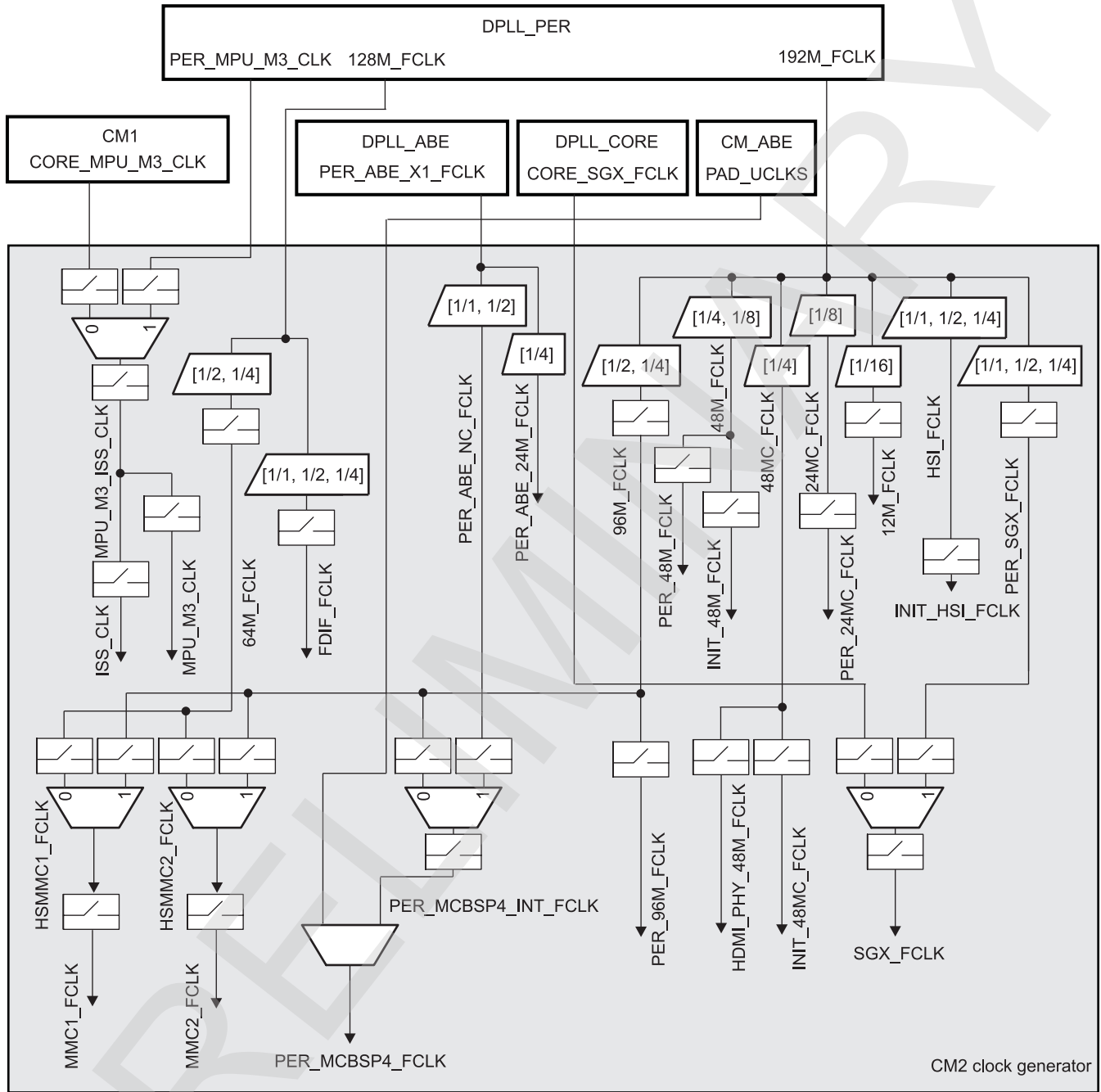
(1) ABE\_ICLK2 depends on the divider settings of the AESS\_FCLK and is always divided by 2 of the ABE\_CLK. So when AESS\_FCLK divider is set to 1, then ABE\_ICLK2 divider is set to 2 and vice versa.

#### 3.6.3.2.4 CM2 Clock Generator

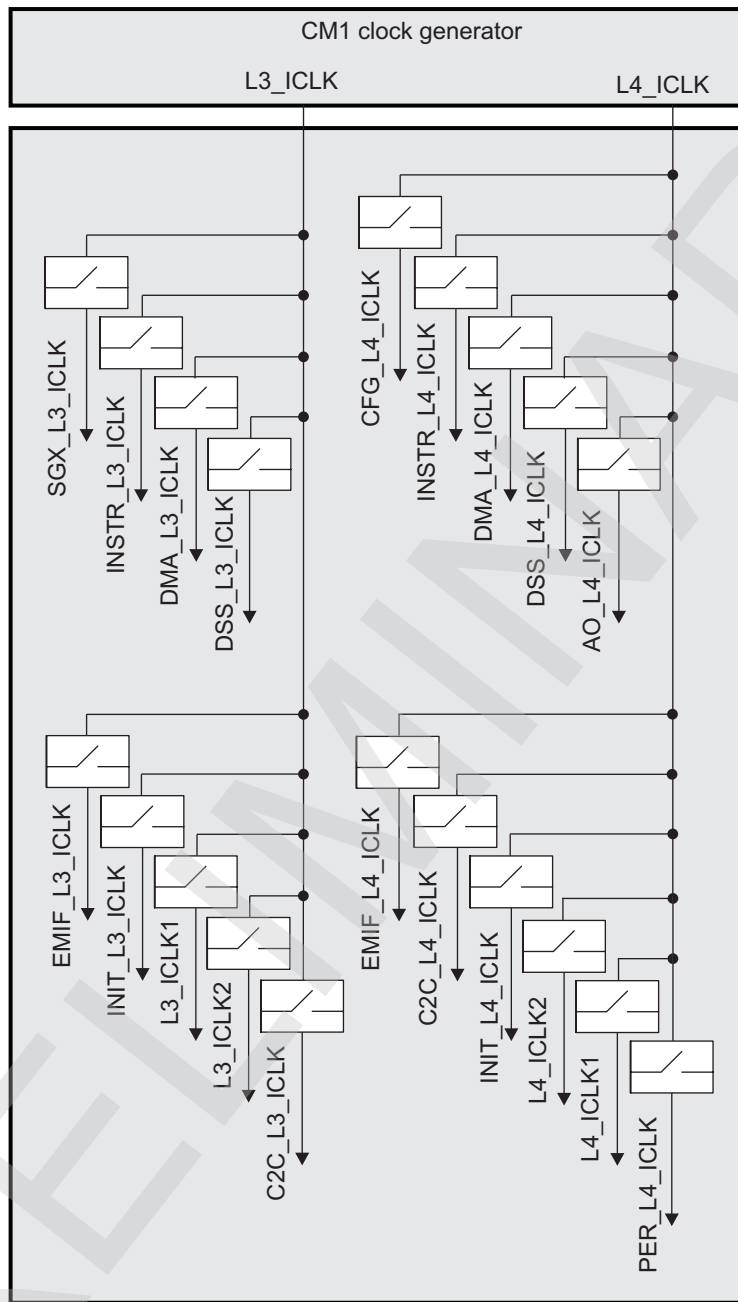
CM2 provides gated functional clocks to various modules of the device.

Figure 3-38 and Figure 3-39 show the various functional and interface clocks generated by CM2:

Figure 3-38. CM2 Overview (a)



prcm-039

**Figure 3-39. CM2 Overview (b)**

prcm-040

Table 3-38 identifies controls for clock dividers or muxes in the CM2 clock source.

**Table 3-38. CM2 Clock Division and Muxing Control**

Divider/Mux	Control Bit Field
Divider FDIF_FCLK	CM_CAM_FDIF_CLKCTRL[25:24] CLKSEL_FCLK
Divider HSI_FCLK	CM_L3INIT_HSI_CLKCTRL[25:24] CLKSEL
Mux HSMMC1_FCLK	CM_L3INIT_HSMMC1_CLKCTRL[24] CLKSEL
Mux HSMMC2_FCLK	CM_L3INIT_HSMMC2_CLKCTRL[24] CLKSEL
Mux MPU_M3_ISS_CLK	CM_CLKSEL_MPU_M3_ISS_ROOT[0] CLKSEL
Mux PER_MCBSP4_FCLK	CM_L4PER_MCBSP4_CLKCTRL[24] CLKSEL_SOURCE

**Table 3-38. CM2 Clock Division and Muxing Control (continued)**

Divider/Mux	Control Bit Field
Mux PER_MCBSP4_INT_FCLK	CM_L4PER_MCBSP4_CLKCTRL[25] CLKSEL_INTERNAL_SOURCE
Mux SGX_FCLK	CM_SGX_SGX_CLKCTRL[24] CLKSEL_SGX_FCLK
Divider PER_ABE_NC_FCLK, 96M_FCLK, 64M_FCLK and 48M_FCLK	CM_SCALE_FCLK[0] SCALE_FCLK

### 3.6.3.3 Generic DPLL Overview

To generate high-frequency clocks, the device supports multiple on-chip DPLLs controlled directly by the PRCM module. They are of two types, identified as type A and type B DPLLs.

Following DPLLs belong to type A:

- DPLL\_MPU
- DPLL\_IVA
- DPLL\_CORE
- DPLL\_PER
- DPLL\_ABE

Following DPLLs belong to type B:

- DPLL\_USB

All DPLLs support the features presented in the following sections, unless identified otherwise.

---

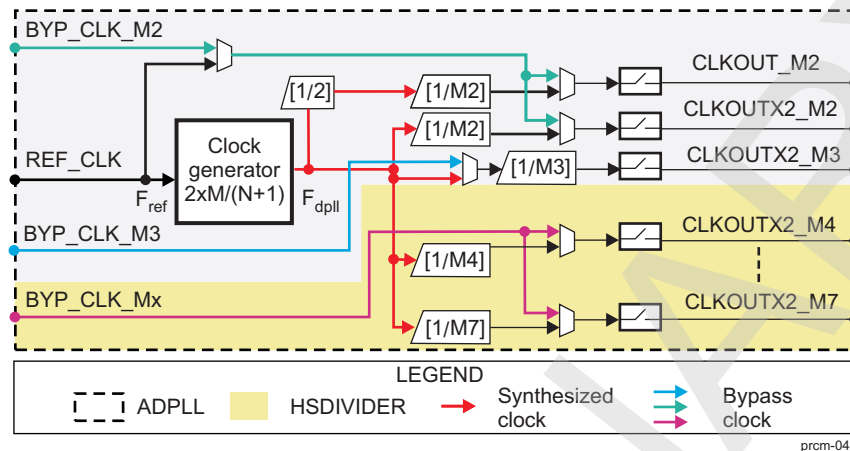
**NOTE:** This chapter discusses only the DPLLs that are directly controlled by the PRCM module. The other DPLLs embedded in and managed by other subsystems are covered in the description of their respective subsystems. For example, [Chapter 10, Display Subsystem](#), discusses the DPLL in the display subsystem.

---

### 3.6.3.3.1 DPLLs Output Clocks Parameters

Figure 3-40 shows the functional architecture of a generic DPLL.

**Figure 3-40. Generic DPLL Functional Diagram**



The DPLL has three input clocks:

- REF\_CLK: used to generate the synthesized clock but can also be used as the bypass clock for two outputs of the DPLL, whenever the DPLL enters a bypass mode. It is mandatory for the DPLL clock synthesis.
- BYP\_CLK\_M2: Selectable bypass clock for the output of an M2 post-divider (optional).
- BYP\_CLK\_M3: Selectable bypass clock for the output of an M3 post-divider (optional).

The DPLL can be programmed to be locked at any frequency given by the following equation:

- $F_{dppll} = F_{ref} \times 2 \times M / (N + 1)$  or
- $F_{dppll} = F_{ref} \times 2 \times (4 \times M / (N + 1))$  in case REGM4XEN bit field is set (only applicable to DPLL\_ABE).

Where:

- $F_{dppll}$  is the DPLL lock frequency.
- $F_{ref}$  is the REF\_CLK frequency.
- M is the software-configured multiplication ratio binary value.
- N is the software-configured division ratio binary value.

It internally generates three main clocks: CLKOUT\_M2, CLKOUTX2\_M2, and CLKOUTX2\_M3 as presented in Table 3-39.

**Table 3-39. CLKOUT\_M2, CLKOUTX2\_M2, and CLKOUTX2\_M3 Frequencies With DPLL State**

Output	Equation	DPLL Mode
CLKOUT_M2	$F_{dppll} / (2 \times M2)$	Locked
	$F_{dppll}$ or BYP_CLK_M2	Before lock or during relock
CLKOUTX2_M2	$F_{dppll} / M2$	Locked
	$F_{dppll}$ or BYP_CLK_M2	Before lock or during relock
CLKOUTX2_M3	$F_{dppll} / M3$ or BYP_CLK_M3 / M3	Locked
	0	Before lock or during relock

Where:

- M2 is the software-configured division ratio binary value.
- M3 is the software-configured division ratio binary value.
- CLKOUT\_M2 and CLKOUTX2\_M2 bypass clock input can be switched when the DPLL is not in locked state, using the M2 Bypass Clock Select Control bit.
- CLKOUTX2\_M3 output clock can be switched when the DPLL is in locked state, using the M3 Clock

Select Control bit.

**NOTE:**

- Zero-value for M2 and M3 division ratios is not allowed. They are set to 1 after reset.
- CLKOUT\_M2 is generated based on a fix divide-by-2 ratio, except in bypass mode.

The DPLL may contain an HSDIVIDER module to produce more clocks with divided ratio based on the DPLL synthesized clock frequency. HSDIVIDER provides four extra post-dividers from M4 to M7. The HSDIVIDER output clocks frequency is given by the equations in [Table 3-40](#):

**Table 3-40. CLKOUTX2\_Mn Frequencies With DPLL State**

Equation	DPLL Mode
$\text{CLKOUTX2\_Mn} = \text{Fdpll}/\text{Mn}$	Locked
$\text{CLKOUTX2\_Mn} = \text{BYP\_CLK\_Mx}$	Before lock or during relock

Where:

- Fdpll is the DPLL lock frequency.
- Mn is the software configured division ratio binary value.
- n is in the range from 4 to 7.

**NOTE:**

- Mn division ratio is set to 1 after reset and 31 (0x1f) is an illegal value.

All clock outputs of the DPLL can be gated. The PRCM module provides the DPLL with a clock gating control signal to enable or disable the clock, and the DPLL provides the PRCM module with a clock activity status signal to let the PRCM module hardware know when the clock is effectively running or effectively gated.

The type B DPLL (only DPLL\_USB) has two outputs: CLKOUT and CLKDCOLDO. The DPLL can be programmed to be locked with the output clock frequencies given by the following equation:

- $\text{Fclkout} = \text{Fref} \times (\text{M}/(\text{N}+1)) \times (1/\text{M}2)$
- $\text{Fclkdcoldo} = \text{Fref} \times (\text{M}/(\text{N}+1))$

Where:

- Fclkout is the frequency of the clock at output CLKOUT.
  - Fclkdcoldo is the frequency of the clock at output CLKDCOLDO.
  - Fref is the REF\_CLK frequency.
  - M is the software-configured multiplication ratio binary value.
  - N is the software-configured division ratio binary value.
  - M2 is the software-configured division ratio binary value.
- CLKOUT supports the same bypass modes as identified earlier. CLKDCOLDO does not support bypass mode.

### 3.6.3.3.2 Enable Control, Status, and Low-Power Operation Mode

The DPLL has a Manual Mode Control bit field, which allows the setting of the different operating modes of the DPLL. When the DPLL is switched to lock mode the current values of the multiplication ratio (M) and the division ratio (N) are latched in the DPLL. The DPLL then starts the lock or relock sequence, to synthesize the corresponding output frequency clock.

The status of the synthesized clock output of the DPLL is represented by the CLKOUT status bit. It can be gated or active.

The DPLL can be switched to low-power operation mode (also called LPMODE) to optimize DPLL power consumption when the input and output clock frequencies are low. This mode can be software enabled using the low-power mode control bit of the DPLL.

It must only be enabled if both of the following operating conditions are satisfied:

- $F_{ref}/(N+1)$  is less than or equal to 1 MHz.
- $F_{ref} \times M/(N+1)$  is less than or equal to 100 MHz.

With:

- $F_{ref}$  is the REF\_CLK frequency.
- M is the software-configured multiplication ratio binary value.
- N is the software-configured division ratio binary value.

### 3.6.3.3.3 DPLL Power Modes

DPLL may supports several power modes. Each mode results in a tradeoff between power savings and relock time.

Table 3-41 shows the DPLL power modes.

**Table 3-41. DPLL Power Modes**

Power Mode	CLKOUT State	Logic Current (mA)	Analog Current (mA)	Freq Lock Time	Phase Lock Time
Low-power stop	Clock stopped	0.05 (leakage)	0.001 (leakage)	$1.5 \mu\text{s} + (20 \times 1/F_{ref})$	$1.5 \mu\text{s} + (100 \times 1/F_{ref})$
Fast-relock stop	Clock stopped	0.05 (leakage)	0.25 (leakage)	$0.05 \mu\text{s} + (20 \times 1/F_{ref})$	$0.05 \mu\text{s} + (100 \times 1/F_{ref})$
MN bypass	Bypass clock	0.05 (leakage)	0.001 (leakage)	$1.5 \mu\text{s} + (350 \times 1/F_{ref})$	$1.5 \mu\text{s} + (500 \times 1/F_{ref})$
Low-power bypass	Bypass clock/Clock stopped	0.05 (leakage)	0.001 (leakage)	$1.5 \mu\text{s} + (20 \times 1/F_{ref})$	$1.5 \mu\text{s} + (100 \times 1/F_{ref})$
Fast-relock bypass	Bypass clock/Clock stopped	0.05 (leakage)	0.25 (leakage)	$0.05 \mu\text{s} + (20 \times 1/F_{ref})$	$0.05 \mu\text{s} + (100 \times 1/F_{ref})$
Lock	Synthesized clock	0.55 (active)	1.95 (active)	N/A	N/A

Where:

- $F_{ref}$  is the REF\_CLK frequency.

A DPLL power mode can be achieved on a software request (manual) and/or automatically (automatic), depending on the specific hardware conditions.

A DPLL can switch from one mode to the other as a result of the following:

- Software-programmed transition (manual): The software configures the dedicated DPLL Manual Mode Control feature for the next desired DPLL mode. It must ensure that the transition can be performed based on the clock activity on the device.
- Combined software-programmed and hardware-conditions-based transition (auto): This mode allows the DPLL to automatically transition to a low-power state (that is, any state other than MN BYPASS and the LOCK state) when the output clocks are gated or the destination clock domain is inactive. And to switch back to the LOCK state when the output clock is needed (that is, the clock is ungated or the clock domain becomes active). The desired low-power state for the automatic transition is configured in dedicated Auto Mode Control parameter of the DPLL.

**NOTE:** With REF\_CLK period =  $1/F_{ref} = (N + 1)/CLKINP$ ; ( $F_{ref} = CLKINP/(N + 1)$ ):

This formula indicates that a smaller N divider value provides a smaller time for switching the clock after an M2 post-divider change.

A compromise is necessary between the clock switching latency and power consumption. Having a smaller N value:

- Requires a higher M2 post-divider value to obtain the same target frequency.
- Results in a higher DPLL lock frequency, and then higher power consumption.



**NOTE:**

- A manual transition can be performed from any power mode to any other power mode.
- An automatic transition can be performed from lock mode to any low-power mode, except for the MN bypass mode.

**3.6.3.3.4 DPLL Recalibration**

Each time the DPLL is reset or performs a lock sequence (following a change in multiplier M or divider N value), it performs a recalibration of the output frequency, based on voltage and temperature conditions. In the locked mode the DPLL maintains a steady lock frequency output by compensating for voltage and temperature changes within a certain range. However, if the voltage or temperature drifts outside the range or shows a significant or fast change, the DPLL may not be able to track and compensate it. It would need a recalibration, which is signaled by assertion of a recalibration flag.

**NOTE:**

- The recalibration mechanism is active only while the DPLL is in locked mode. When the DPLL is in off or bypass mode (low-power or fast-relock), it does not assert the recalibration flag.
- If the DPLL drifts out of the operating range limits while not locked, then when it tries to relock it fails to lock within the normal delay and recalibrates automatically before eventually locking. The only difference between this case and a standard relock is the recalibration delay.

During recalibration, the DPLL loses lock and output clock switches to the bypass clock.

The DPLL can automatically start recalibration when the recalibration flag is asserted, or recalibration can be triggered by software control. The trigger setting of the recalibration can be configured by the corresponding registers of the DPLL in the PRCM module. The software-controlled recalibration mode is selected by default.

Software-controlled recalibration: The DPLL continues its tracking mechanism as long as the recalibration is not triggered by software (that is, by enabling the Recalibration-Enable control parameter). If the DPLL reaches upper or lower bounds of the DCO control code and software has still not triggered recalibration, the DPLL stops its tracking mechanism. The output clock remains active, but frequency and jitter are not ensured to meet the requirement.

Automatic recalibration: The DPLL immediately starts the recalibration as soon as the recalibration flag is asserted.

**NOTE:**

Automatic recalibration of the DPLL can start at any time. While relocking, the DPLL switches to bypass mode, which introduces a frequency change. For modules that are sensitive to frequency change while operating, this can introduce operational instability. For example, the external memory (EMIF) controller is sensitive to a frequency change on the DPLL because its embedded DLL relocks on a frequency change. Any EMIF access during this DLL relock period can be corrupted. It is, therefore, important to stall EMIF access during DPLL recalibration.

To allow the software to recalibrate the DPLL at the correct time depending on the device activity, the PRCM module can generate a wake-up event on processor power domain followed by an interrupt on the processor subsystem when the DPLL recalibration flag is asserted.

Table 3-42 shows the DPLL recalibration and control parameters.

**Table 3-42. DPLL Recalibration Control Parameters**

Parameter	Description
Recalibration-Enable control	Enable/disable the DPLL automatic recalibration feature.

**Table 3-42. DPLL Recalibration Control Parameters (continued)**

Parameter	Description
Recalibration-Interrupt mask control	Mask/unmask the DPLL recalibration interrupt to processor.
Recalibration-Interrupt status	Status of the DPLL recalibration interrupt to processor

### 3.6.3.3.5 DPLL Spread Spectrum Clocking

The synthesized clock output of the DPLL has a stable frequency. This periodicity generates a significant power peak at the selected frequency, which in turn causes an electromagnetic interference (EMI) disturbance to the environment. In communication devices, this clock signal generates unwanted spurious signals that interfere with the decoding of information received. It results in performance degradation, for example, in the form of high bit-error rates in case of a wireless communication interface.

To reduce the power peaks (and the electromagnetic noise generated at a specific frequency), an internal frequency modulation of the DPLL is used to distribute the energy to many different frequencies, thus reducing the power peaks. This is called spread spectrum clocking (SSC).

SSC in the DPLL is performed by changing the feedback divider (M) in a triangular pattern. This varies the frequency of the output clock in a triangular pattern. The frequency of this pattern is the modulation frequency (fm). It is programmed as a ratio of the REF\_CLK/4. It is given as:

$$f_m = (\text{REF\_CLK}/4) * (1/\text{ModFreqDivider})$$

Where: ModFreqDivider is the value of the modulation frequency divider and is given as:

$$\text{ModFreqDivider} = \text{Mantissa} * 2^{\text{Exponent}}$$

The Mantissa and Exponent are configurable parameters of the DPLL.

The ratio of output frequency variation (that is, frequency spread Df) with respect to the programmed synthesized frequency Fout of the DPLL is equal to the ratio of peak (M) or the amplitude of the SSC modulation with feedback divider (M).

$$M/M = f/F_{out}$$

The frequency spread (Df), can thus be controlled by configuring the amplitude of modulation (M). The M is controlled by setting the step size of variation of M (that is, DeltaMStep) for a given REF\_CLK.

$$M = \text{ModFreqDivider} * \text{DeltaMStep}$$

Where: DeltaMStep is split into integer part and fractional part.

Table 3-43 shows the DPLL SSC control parameters.

**Table 3-43. DPLL SSC Control Parameters**

Parameter	Description
SSC enable control	Enable/disable the DPLL SSC feature. When set to disable, SSC is disabled only after completion of one full cycle of the modulation pattern to maintain the average frequency.
SSC acknowledge	Notifies the exact start and end of SSC
Modulation Frequency Divider setting (Mantissa and Exponent)	Set the ratio of the modulation frequency with respect to the REF_CLK/4 frequency.
M step size setting (DeltaMStepInteger and DeltaMStepFraction)	Set the feedback divider variation step-size.

### 3.6.3.3.6 DPLL Output Power Down

The DCO clock LDO (DCOCLKLDO) of the DPLL can be powered down if all output dividers of the DPLL are powered down. The PRCM module automatically re-enables the power to the LDO when an output divider is powered up or the DPLL switches to bypass mode.

Similarly, all the output dividers (M2, M3, and the Mn) can be powered down when their output clocks are gated and powered up when the clocks are needed again.

**Table 3-44. DPLL Power-Down Control Parameters**

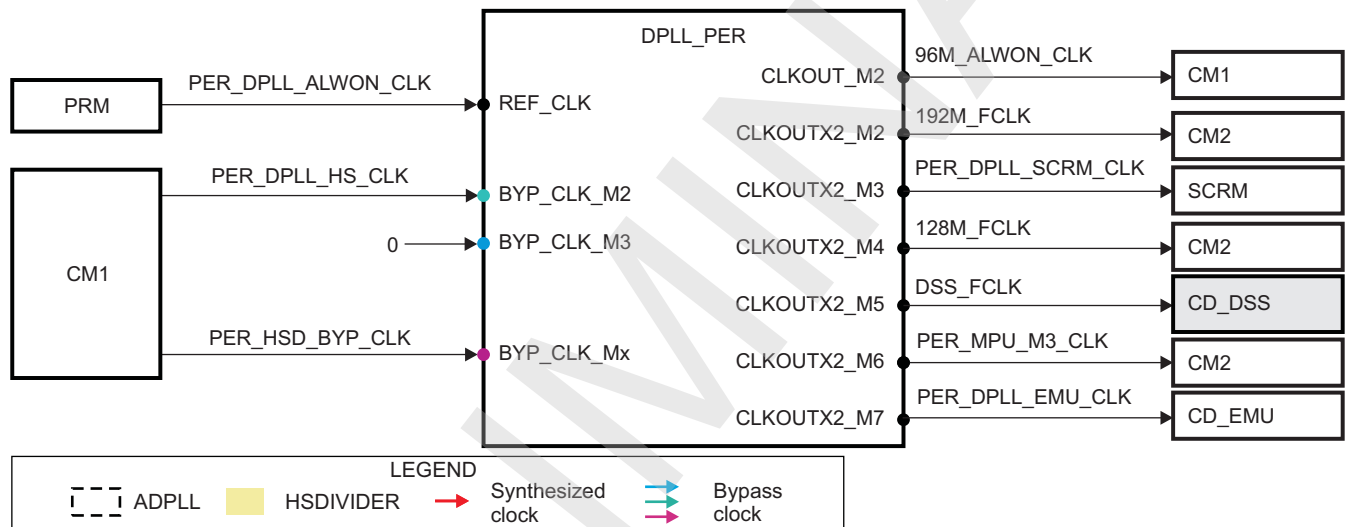
Parameter	Description
DCO clock LDO power down control	Enable/disable automatic power-down feature if the all the output dividers are power-down.
M2, M3, and Mn (where n = 4 to 7)	Enables/disables automatic power-down feature of the corresponding output divider if the output clocks of the divider are gated.

### 3.6.3.4 DPLL\_PER Description

#### 3.6.3.4.1 Overview

Figure 3-41 shows the overview of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for a functional overview of a generic DPLL module.

**Figure 3-41. DPLL\_PER Overview**



prcm-042

#### 3.6.3.4.2 Synthesized Clock Parameters

This section presents the clock synthesis and clock out divider parameters of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for an explanation of the clock synthesis and output divider parameters of the DPLL module.

Table 3-45 lists the clock synthesis parameters of the DPLL.

**Table 3-45. DPLL\_PER Clock Synthesis Parameters**

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_PER[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_PER[6:0] DPLL_DIV

Table 3-46 lists the clock output divider parameters of the DPLL.

**Table 3-46. DPLL\_PER Clock Output Parameters**

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_PER[9] ST_DPLL_CLKOUT
CLKOUT_M2	Gating Control	CM_DIV_M2_DPLL_PER[8] DPLL_CLKOUT_GATE_CTRL

**Table 3-46. DPLL\_PER Clock Output Parameters (continued)**

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Divider Acknowledge	CM_DIV_M2_DPLL_PER[5] DPLL_CLKOUT_DIVCHACK
CLKOUTX2_M2	Status	CM_DIV_M2_DPLL_PER[11] ST_DPLL_CLKOUTX2
CLKOUTX2_M2	Gating Control	CM_DIV_M2_DPLL_PER[10] DPLL_CLKOUTX2_GATE_CTRL
CLKOUT_M2	Divider Control	CM_DIV_M2_DPLL_PER[4:0] DPLL_CLKOUT_DIV
CLKOUTX2_M3	Status	CM_DIV_M3_DPLL_PER[9] ST_DPLL_CLKOUTHIF
CLKOUTX2_M3	Gating Control	CM_DIV_M3_DPLL_PER[8] DPLL_CLKOUTHIF_GATE_CTRL
CLKOUTX2_M3	Divider Acknowledge	CM_DIV_M3_DPLL_PER[5] DPLL_CLKOUTHIF_DIVCHACK
CLKOUTX2_M3	Divider Control	CM_DIV_M3_DPLL_PER[4:0] DPLL_CLKOUTHIF_DIV
CLKOUTX2_M4	Status	CM_DIV_M4_DPLL_PER[9] ST_HSDIVIDER_CLKOUT1
CLKOUTX2_M4	Gating Control	CM_DIV_M4_DPLL_PER[8] HSDIVIDER_CLKOUT1_GATE_CTRL
CLKOUTX2_M4	Divider Acknowledge	CM_DIV_M4_DPLL_PER[5] HSDIVIDER_CLKOUT1_DIVCHACK
CLKOUTX2_M4	Divider Control	CM_DIV_M4_DPLL_PER[4:0] HSDIVIDER_CLKOUT1_DIV
CLKOUTX2_M5	Status	CM_DIV_M5_DPLL_PER[9] ST_HSDIVIDER_CLKOUT2
CLKOUTX2_M5	Gating Control	CM_DIV_M5_DPLL_PER[8] HSDIVIDER_CLKOUT2_GATE_CTRL
CLKOUTX2_M5	Divider Acknowledge	CM_DIV_M5_DPLL_PER[5] HSDIVIDER_CLKOUT2_DIVCHACK
CLKOUTX2_M5	Divider Control	CM_DIV_M5_DPLL_PER[4:0] HSDIVIDER_CLKOUT2_DIV
CLKOUTX2_M6	Status	CM_DIV_M6_DPLL_PER[9] ST_HSDIVIDER_CLKOUT3
CLKOUTX2_M6	Gating Control	CM_DIV_M6_DPLL_PER[8] HSDIVIDER_CLKOUT3_GATE_CTRL
CLKOUTX2_M6	Divider Acknowledge	CM_DIV_M6_DPLL_PER[5] HSDIVIDER_CLKOUT3_DIVCHACK
CLKOUTX2_M6	Divider Control	CM_DIV_M6_DPLL_PER[4:0] HSDIVIDER_CLKOUT3_DIV
CLKOUTX2_M7	Status	CM_DIV_M7_DPLL_PER[9] ST_HSDIVIDER_CLKOUT4
CLKOUTX2_M7	Gating Control	CM_DIV_M7_DPLL_PER[8] HSDIVIDER_CLKOUT4_GATE_CTRL
CLKOUTX2_M7	Divider Acknowledge	CM_DIV_M7_DPLL_PER[5] HSDIVIDER_CLKOUT4_DIVCHACK
CLKOUTX2_M7	Divider Control	CM_DIV_M7_DPLL_PER[4:0] HSDIVIDER_CLKOUT4_DIV

### 3.6.3.4.3 Power Modes

This section identifies the operating modes supported by the DPLL and also identify the control bit fields to set its operating modes. See [Section 3.6.3.3.2, Enable Control, Status and Low-Power Operation Mode](#), and [Section 3.6.3.3.3, DPLL Power Modes](#), for an explanation of the DPLL operating modes, and associated control and status features.

[Table 3-47](#) lists the operating modes supported by the DPLL.

**Table 3-47. DPLL\_PER Modes**

Low-Power Stop	Fast-Relock Stop	MN Bypass	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available (default)	Available	Available	Available

Table 3-48 lists the control bit fields for the operating mode control of the DPLL.

**Table 3-48. DPLL\_PER Mode Control Parameters**

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_PER[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_PER[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_PER[2:0] AUTO_DPLL_MODE

#### 3.6.3.4.4 Recalibration

Table 3-49 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. See Section 3.6.3.3.4, *DPLL Recalibration*, for an explanation of the DPLL recalibration feature.

**Table 3-49. DPLL\_PER Recalibration Feature Parameters**

Parameter Name	Control/Status Bit Field
Recalibration - Enable Control	CM_CLKMODE_DPLL_PER[8] DPLL_DRIFTGUARD_EN
Recalibration - Interrupt Status	PRM_IRQSTATUS_MPU_A9[3] DPLL_PER_RECAL_ST
Recalibration - Interrupt MASK Control	PRM_IRQENABLE_MPU_A9[3] DPLL_PER_RECAL_EN
Recalibration - Interrupt Status	PRM_IRQSTATUS_MPU_M3[3] DPLL_PER_RECAL_ST
Recalibration - Interrupt MASK Control	PRM_IRQENABLE_MPU_M3[3] DPLL_PER_RECAL_EN

#### 3.6.3.4.5 Spread Spectrum Clocking

Table 3-50 lists the control bit fields associated with the SSC features of the DPLL. See Section 3.6.3.3.5, *DPLL Spread Spectrum Clocking*, for an explanation of the DPLL SSC feature.

**Table 3-50. DPLL\_PER Spread-Spectrum Clocking Feature Parameters**

Parameter Name	Control Bit Field
SSC Downspread	CM_CLKMODE_DPLL_PER[14] DPLL_SSC_DOWNSPREAD
SSC Acknowledge	CM_CLKMODE_DPLL_PER[13] DPLL_SSC_ACK
SSC Enable Control	CM_CLKMODE_DPLL_PER[12] DPLL_SSC_EN
Modulation Step Size Setting	CM_SSC_DELTAMSTEP_DPLL_PER[19:0] DELTAMSTEP
Modulation Frequency Divider - Exponent	CM_SSC_MODFREQDIV_DPLL_PER[10:8] MODFREQDIV_EXPONENT
Modulation Frequency Divider - Mantissa	CM_SSC_MODFREQDIV_DPLL_PER[6:0] MODFREQDIV_MANTISSA

#### 3.6.3.4.6 Output Power Down

Table 3-51 lists the control bit fields for the output power down features of the DPLL. See Section 3.6.3.3.6, *DPLL Output Power-Down*, for an explanation of the DPLL output power down feature.

**Table 3-51. DPLL\_PER Clock Output Power-down Parameters**

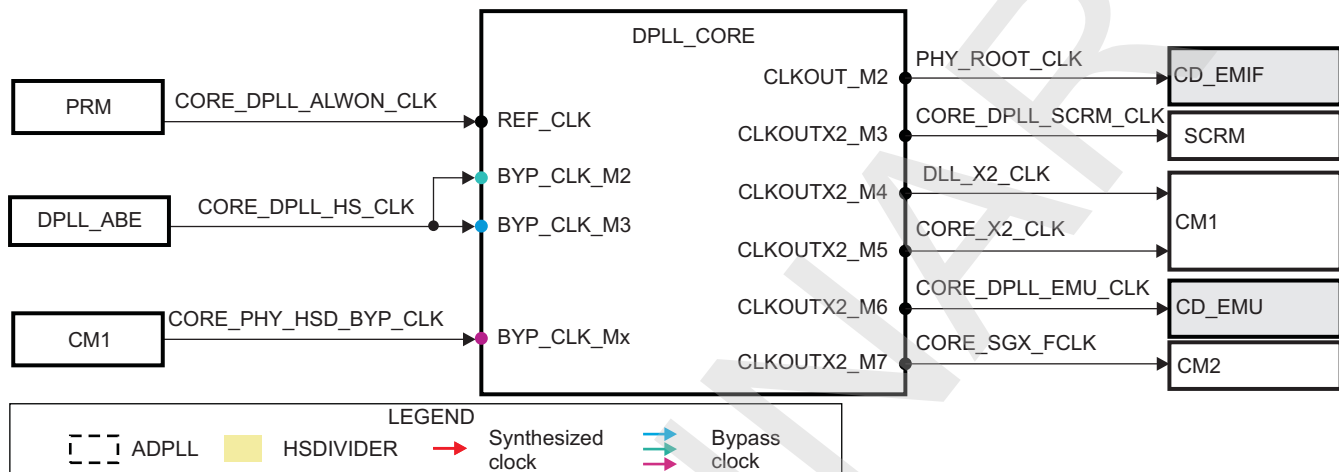
Parameter Name	Control/Status Bit Field
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M4_DPLL_PER[12] HSDIVIDER_CLKOUT1_PWDN
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M5_DPLL_PER[12] HSDIVIDER_CLKOUT2_PWDN
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M6_DPLL_PER[12] HSDIVIDER_CLKOUT3_PWDN
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M7_DPLL_PER[12] HSDIVIDER_CLKOUT4_PWDN

### 3.6.3.5 DPLL\_CORE Description

#### 3.6.3.5.1 Overview

Figure 3-42 shows the overview of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for a functional overview of a generic DPLL module.

Figure 3-42. DPLL\_CORE Overview



prcm-043

#### 3.6.3.5.2 Synthesized Clock Parameters

This section presents the clock synthesis and clock out divider parameters of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for an explanation of the clock synthesis and output divider parameters of the DPLL module.

Table 3-52 lists the clock synthesis parameters of the DPLL.

Table 3-52. DPLL\_CORE Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_CORE[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_CORE[6:0] DPLL_DIV
M (Restore)	CM_CLKSEL_DPLL_CORE_RESTORE[18:8] DPLL_MULT
N (Restore)	CM_CLKSEL_DPLL_CORE_RESTORE[6:0] DPLL_DIV

Table 3-53 lists the clock output divider parameters of the DPLL.

Table 3-53. DPLL\_CORE Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_CORE[9] ST_DPLL_CLKOUT
CLKOUT_M2	Gating Control	CM_DIV_M2_DPLL_CORE[8] DPLL_CLKOUT_GATE_CTRL
CLKOUT_M2	Divider Acknowledge	CM_DIV_M2_DPLL_CORE[5] DPLL_CLKOUT_DIVCHACK
CLKOUT_M2	Divider Control	CM_DIV_M2_DPLL_CORE[4:0] DPLL_CLKOUT_DIV
CLKOUTX2_M3	Status	CM_DIV_M3_DPLL_CORE[9] ST_DPLL_CLKOUTHIF
CLKOUTX2_M3	Gating Control	CM_DIV_M3_DPLL_CORE[8] DPLL_CLKOUTHIF_GATE_CTRL

**Table 3-53. DPLL\_CORE Clock Output Parameters (continued)**

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUTX2_M3	Divider Acknowledge	CM_DIV_M3_DPLL_CORE[5] DPLL_CLKOUTHIF_DIVCHACK
CLKOUTX2_M3	Divider Control	CM_DIV_M3_DPLL_CORE[4:0] DPLL_CLKOUTHIF_DIV
CLKOUTX2_M4	Status	CM_DIV_M4_DPLL_CORE[9] ST_HSDIVIDER_CLKOUT1
CLKOUTX2_M4	Gating Control	CM_DIV_M4_DPLL_CORE[8] HSDIVIDER_CLKOUT1_GATE_CTRL
CLKOUTX2_M4	Divider Acknowledge	CM_DIV_M4_DPLL_CORE[5] HSDIVIDER_CLKOUT1_DIVCHACK
CLKOUTX2_M4	Divider Control	CM_DIV_M4_DPLL_CORE[4:0] HSDIVIDER_CLKOUT1_DIV
CLKOUTX2_M5	Status	CM_DIV_M5_DPLL_CORE[9] ST_HSDIVIDER_CLKOUT2
CLKOUTX2_M5	Gating Control	CM_DIV_M5_DPLL_CORE[8] HSDIVIDER_CLKOUT2_GATE_CTRL
CLKOUTX2_M5	Divider Acknowledge	CM_DIV_M5_DPLL_CORE[5] HSDIVIDER_CLKOUT2_DIVCHACK
CLKOUTX2_M5	Divider Control	CM_DIV_M5_DPLL_CORE[4:0] HSDIVIDER_CLKOUT2_DIV
CLKOUTX2_M6	Status	CM_DIV_M6_DPLL_CORE[9] ST_HSDIVIDER_CLKOUT3
CLKOUTX2_M6	Gating Control	CM_DIV_M6_DPLL_CORE[8] HSDIVIDER_CLKOUT3_GATE_CTRL
CLKOUTX2_M6	Divider Acknowledge	CM_DIV_M6_DPLL_CORE[5] HSDIVIDER_CLKOUT3_DIVCHACK
CLKOUTX2_M6	Divider Control	CM_DIV_M6_DPLL_CORE[4:0] HSDIVIDER_CLKOUT3_DIV
CLKOUTX2_M7	Status	CM_DIV_M7_DPLL_CORE[9] ST_HSDIVIDER_CLKOUT4
CLKOUTX2_M7	Gating Control	CM_DIV_M7_DPLL_CORE[8] HSDIVIDER_CLKOUT4_GATE_CTRL
CLKOUTX2_M7	Divider Acknowledge	CM_DIV_M7_DPLL_CORE[5] HSDIVIDER_CLKOUT4_DIVCHACK
CLKOUTX2_M7	Divider Control	CM_DIV_M7_DPLL_CORE[4:0] HSDIVIDER_CLKOUT4_DIV

### 3.6.3.5.3 Power Modes

This section identifies the operating modes supported by the DPLL and also identifies the control bit fields to set its operating modes. See [Section 3.6.3.3.2, Enable Control, Status and Low-Power Operation Mode](#), and [Section 3.6.3.3.3, DPLL Power Modes](#), for an explanation of the DPLL operating modes, and associated control and status features.

[Table 3-54](#) lists the operating modes supported by the DPLL.

**Table 3-54. DPLL\_CORE Modes**

Low-Power Stop	Fast-Relock Stop	MN Bypass	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available (default)	Available	Available	Available

[Table 3-55](#) lists the control bit fields for the operating mode control of the DPLL.

**Table 3-55. DPLL\_CORE Mode Control Parameters**

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_CORE[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_CORE[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_CORE[2:0] AUTO_DPLL_MODE
Low-Power Mode Control (Restore)	CM_CLKMODE_DPLL_CORE_RESTORE[10] DPLL_LPMODE_EN
Manual Mode Control (Restore)	CM_CLKMODE_DPLL_CORE_RESTORE[2:0] DPLL_EN
Auto Mode Control (Restore)	CM_AUTOIDLE_DPLL_CORE_RESTORE[2:0] AUTO_DPLL_MODE

### 3.6.3.5.4 Recalibration

Table 3-56 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. See Section 3.6.3.3.4, *DPLL Recalibration*, for an explanation of the DPLL recalibration feature.

**Table 3-56. DPLL\_CORE Recalibration Feature Parameters**

Parameter Name	Control/Status Bit Field
Recalibration - Enable Control	CM_CLKMODE_DPLL_CORE[8] DPLL_DRIFTGUARD_EN
Recalibration - Enable Control	CM_CLKMODE_DPLL_CORE_RESTORE[8] DPLL_DRIFTGUARD_EN
Recalibration - Interrupt Status	PRM_IRQSTATUS_MPU_A9[0] DPLL_CORE_RECAL_ST
Recalibration - Interrupt mask Control	PRM_IRQENABLE_MPU_A9[0] DPLL_CORE_RECAL_EN
Recalibration - Interrupt Status	PRM_IRQSTATUS_MPU_M3[0] DPLL_CORE_RECAL_ST
Recalibration - Interrupt Mask Control	PRM_IRQENABLE_MPU_M3[0] DPLL_CORE_RECAL_EN

### 3.6.3.5.5 Spread Spectrum Clocking

Table 3-57 lists the control bit fields associated with the SSC features of the DPLL. See Section 3.6.3.3.5, *DPLL Spread Spectrum Clocking*, for an explanation of the DPLL SSC feature.

**Table 3-57. DPLL\_CORE Spread-Spectrum Clocking Feature Parameters**

Parameter Name	Control Bit Field
SSC Downspread	CM_CLKMODE_DPLL_CORE[14] DPLL_SSC_DOWNSPREAD
SSC Acknowledge	CM_CLKMODE_DPLL_CORE[13] DPLL_SSC_ACK
SSC Enable Control	CM_CLKMODE_DPLL_CORE[12] DPLL_SSC_EN
Modulation Step Size Setting	CM_SSC_DELTAMSTEP_DPLL_CORE[19:0] DELTAMSTEP
Modulation Frequency Divider - Exponent	CM_SSC_MODFREQDIV_DPLL_CORE[10:8] MODFREQDIV_EXPONENT
Modulation Frequency Divider - Mantissa	CM_SSC_MODFREQDIV_DPLL_CORE[6:0] MODFREQDIV_MANTISSA
Restore Modulation Step Size setting	CM_SSC_DELTAMSTEP_DPLL_CORE_RESTORE[19:0] DELTAMSTEP
Restore Modulation Frequency Divider - Exponent	CM_SSC_MODFREQDIV_DPLL_CORE_RESTORE[10:8] MODFREQDIV_EXPONENT
Restore Modulation Frequency Divider - Mantissa	CM_SSC_MODFREQDIV_DPLL_CORE_RESTORE[6:0] MODFREQDIV_MANTISSA
Restore SSC Downspread	CM_CLKMODE_DPLL_CORE_RESTORE[14] DPLL_SSC_DOWNSPREAD
Restore SSC Acknowledge	CM_CLKMODE_DPLL_CORE_RESTORE[13] DPLL_SSC_ACK
Restore SSC Enable Control	CM_CLKMODE_DPLL_CORE_RESTORE[12] DPLL_SSC_EN



### 3.6.3.5.6 Output Power-Down

Table 3-58 lists the control bit fields for the output power down features of the DPLL. See Section 3.6.3.3.6, *DPLL Output Power Down*, for an explanation of the DPLL output power-down feature.

**Table 3-58. DPLL\_CORE Clock Output Power-Down Parameters**

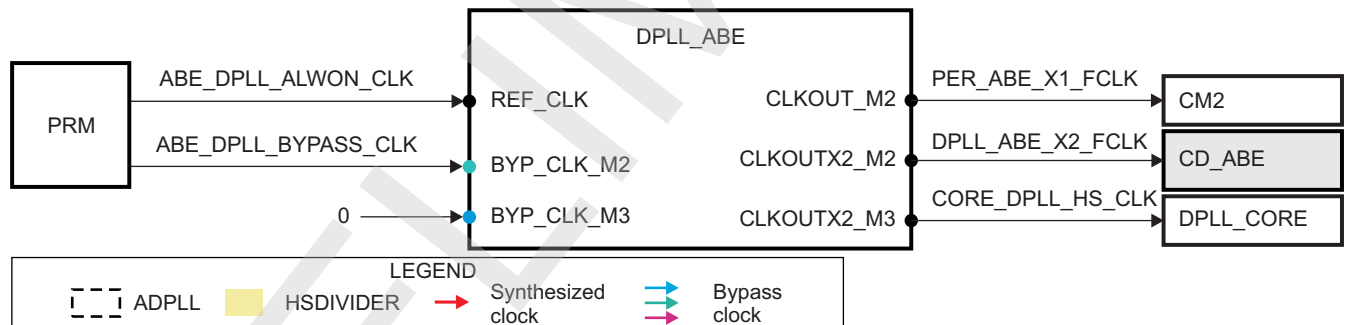
Parameter Name	Control/Status Bit Field
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M4_DPLL_CORE[12] HSDIVIDER_CLKOUT1_PWDN
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M5_DPLL_CORE[12] HSDIVIDER_CLKOUT2_PWDN
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M6_DPLL_CORE[12] HSDIVIDER_CLKOUT3_PWDN
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M7_DPLL_CORE[12] HSDIVIDER_CLKOUT4_PWDN
Restore Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M4_DPLL_CORE_RESTORE[12] HSDIVIDER_CLKOUT1_PWDN
Restore Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M5_DPLL_CORE_RESTORE[12] HSDIVIDER_CLKOUT2_PWDN
Restore Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M6_DPLL_CORE_RESTORE[12] HSDIVIDER_CLKOUT3_PWDN
Restore Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M7_DPLL_CORE_RESTORE[12] HSDIVIDER_CLKOUT4_PWDN

### 3.6.3.6 DPLL\_ABE Description

#### 3.6.3.6.1 Overview

Figure 3-43 shows the overview of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for a functional overview of a generic DPLL module.

**Figure 3-43. DPLL\_ABE Overview**



prcm-044

#### 3.6.3.6.2 Synthesized Clock Parameters

This section presents the clock synthesis and clock out divider parameters of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for an explanation of the clock synthesis and output divider parameters of the DPLL module.

Table 3-59 lists the clock synthesis parameters of the DPLL.

**Table 3-59. DPLL\_ABE Clock Synthesis Parameters**

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_ABE[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_ABE[6:0] DPLL_DIV

Table 3-60 lists the clock output divider parameters of the DPLL.

**Table 3-60. DPLL\_ABE Clock Output Parameters**

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_ABE[9] ST_DPLL_CLKOUT
CLKOUT_M2	Gating Control	CM_DIV_M2_DPLL_ABE[8] DPLL_CLKOUT_GATE_CTRL
CLKOUT_M2	Divider Acknowledge	CM_DIV_M2_DPLL_ABE[5] DPLL_CLKOUT_DIVCHACK
CLKOUTX2_M2	Status	CM_DIV_M2_DPLL_ABE[11] ST_DPLL_CLKOUTX2
CLKOUTX2_M2	Gating Control	CM_DIV_M2_DPLL_ABE[10] DPLL_CLKOUTX2_GATE_CTRL
CLKOUT_M2	Divider Control	CM_DIV_M2_DPLL_ABE[4:0] DPLL_CLKOUT_DIV
CLKOUTX2_M3	Status	CM_DIV_M3_DPLL_ABE[9] ST_DPLL_CLKOUTHIF
CLKOUTX2_M3	Gating Control	CM_DIV_M3_DPLL_ABE[8] DPLL_CLKOUTHIF_GATE_CTRL
CLKOUTX2_M3	Divider Acknowledge	CM_DIV_M3_DPLL_ABE[5] DPLL_CLKOUTHIF_DIVCHACK
CLKOUTX2_M3	Divider Control	CM_DIV_M3_DPLL_ABE[4:0] DPLL_CLKOUTHIF_DIV

### 3.6.3.6.3 Power Modes

This section identifies the operating modes supported by the DPLL and also identify the control bit fields to set its operating modes. See Section 3.6.3.3.2, *Enable Control, Status and Low-Power Operation Mode*, and Section 3.6.3.3.3, *DPLL Power Mode*, for an explanation of the DPLL operating modes, and associated control and status features.

Table 3-61 lists the operating modes supported by the DPLL.

**Table 3-61. DPLL\_ABE Modes**

Low-Power Stop	Fast-Relock Stop	MN Bypass	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available (Default)	Available	Available	Available

Table 3-62 lists the control bit fields for the operating mode control of the DPLL.

**Table 3-62. DPLL\_ABE Mode Control Parameters**

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_ABE[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_ABE[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_ABE[2:0] AUTO_DPLL_MODE

### 3.6.3.6.4 Recalibration

Table 3-63 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. See Section 3.6.3.3.4, *DPLL Recalibration*, for an explanation of the DPLL recalibration feature.

**Table 3-63. DPLL\_ABE Recalibration Feature Parameters**

Parameter Name	Control/Status Bit Field
Recalibration - Enable Control	CM_CLKMODE_DPLL_ABE[8] DPLL_DRIFTGUARD_EN
Recalibration - Interrupt Status	PRM_IRQSTATUS_MPU_A9[4] DPLL_ABE_RECAL_ST
Recalibration - Interrupt Mask Control	PRM_IRQENABLE_MPU_A9[4] DPLL_ABE_RECAL_EN

### 3.6.3.6.5 Spread Spectrum Clocking

Table 3-64 lists the control bit fields associated with the SSC features of the DPLL. See Section 3.6.3.3.5, *DPLL Spread Spectrum Clocking*, for an explanation of the DPLL SSC feature.

**Table 3-64. DPLL\_ABE Spread-Spectrum Clocking Feature Parameters**

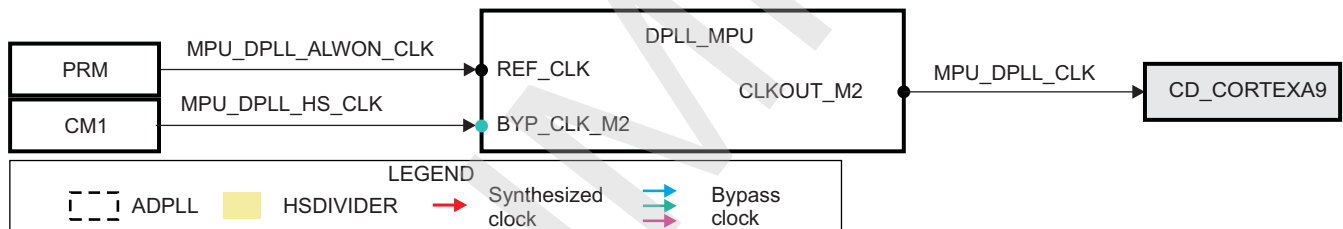
Parameter Name	Control Bit Field
SSC Downspread	CM_CLKMODE_DPLL_ABE[14] DPLL_SSC_DOWNSPREAD
SSC Acknowledge	CM_CLKMODE_DPLL_ABE[13] DPLL_SSC_ACK
SSC Enable Control	CM_CLKMODE_DPLL_ABE[12] DPLL_SSC_EN
Modulation Step Size Setting	CM_SSC_DELTAMSTEP_DPLL_ABE[19:0] DELTAMSTEP
Modulation Frequency Divider - Exponent	CM_SSC_MODFREQDIV_DPLL_ABE[10:8] MODFREQDIV_EXPONENT
Modulation Frequency Divider - Mantissa	CM_SSC_MODFREQDIV_DPLL_ABE[6:0] MODFREQDIV_MANTISSA

### 3.6.3.7 DPLL\_MPU Description

#### 3.6.3.7.1 Overview

Figure 3-44 shows the overview of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for a functional overview of a generic DPLL module.

**Figure 3-44. DPLL\_MPU Overview**



prcm-045

#### 3.6.3.7.2 Synthesized Clock Parameters

This section presents the clock synthesis and clock out divider parameters of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for an explanation of the clock synthesis and output divider parameters of the DPLL module.

Table 3-65 lists the clock synthesis parameters of the DPLL.

**Table 3-65. DPLL\_MPU Clock Synthesis Parameters**

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_MPU[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_MPU[6:0] DPLL_DIV

Table 3-66 lists the clock output divider parameters of the DPLL.

**Table 3-66. DPLL\_MPU Clock Output Parameters**

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_MPU[9] ST_DPLL_CLKOUT
CLKOUT_M2	Gating Control	CM_DIV_M2_DPLL_MPU[8] DPLL_CLKOUT_GATE_CTRL
CLKOUT_M2	Divider Acknowledge	CM_DIV_M2_DPLL_MPU[5] DPLL_CLKOUT_DIVCHACK

**Table 3-66. DPLL\_MPU Clock Output Parameters (continued)**

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Divider Control	CM_DIV_M2_DPLL_MPU[4:0] DPLL_CLKOUT_DIV

### 3.6.3.7.3 Power Modes

This section identifies the operating modes supported by the DPLL and also identify the control bit fields to set its operating modes. See [Section 3.6.3.3.2, Enable Control, Status and Low-Power Operation Mode](#), and [Section 3.6.3.3.3, DPLL Power Modes](#), for an explanation of the DPLL operating modes, and associated control and status features.

[Table 3-67](#) lists the operating modes supported by the DPLL.

**Table 3-67. DPLL\_MPU Modes**

Low-Power Stop	Fast-Relock Stop	MN Bypass	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available (Default)	Available	Available	Available

[Table 3-68](#) lists the control bit fields for the operating mode control of the DPLL.

**Table 3-68. DPLL\_MPU Mode Control Parameters**

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_MPU[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_MPU[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_MPU[2:0] AUTO_DPLL_MODE

### 3.6.3.7.4 Recalibration

[Table 3-69](#) lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. See [Section 3.6.3.3.4, DPLL Recalibration](#), for an explanation of the DPLL recalibration feature.

**Table 3-69. DPLL\_MPU Recalibration Feature Parameters**

Parameter Name	Control/Status Bit Field
Recalibration - Enable Control	CM_CLKMODE_DPLL_MPU[8] DPLL_DRIFTGUARD_EN
Recalibration - Interrupt Status	PRM_IRQSTATUS_MPU_A9[1] DPLL_MPU_RECAL_ST
Recalibration - Interrupt Mask Control	PRM_IRQENABLE_MPU_A9[1] DPLL_MPU_RECAL_EN

### 3.6.3.7.5 Spread Spectrum Clocking

[Table 3-70](#) lists the control bit fields associated with the SSC features of the DPLL. See [Section 3.6.3.3.5, DPLL Spread Spectrum Clocking](#), for an explanation of the DPLL SSC feature.

**Table 3-70. DPLL\_MPU Spread-Spectrum Clocking Feature Parameters**

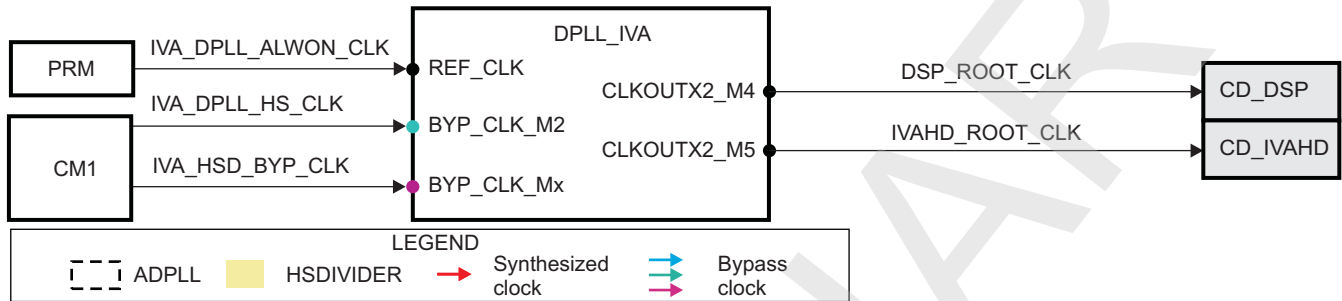
Parameter Name	Control Bit Field
SSC Downspread	CM_CLKMODE_DPLL_MPU[14] DPLL_SSC_DOWNSPREAD
SSC Acknowledge	CM_CLKMODE_DPLL_MPU[13] DPLL_SSC_ACK
SSC Enable Control	CM_CLKMODE_DPLL_MPU[12] DPLL_SSC_EN
Modulation Step Size Setting	CM_SSC_DELTAMSTEP_DPLL_MPU[19:0] DELTAMSTEP
Modulation Frequency Divider - Exponent	CM_SSC_MODFREQDIV_DPLL_MPU[10:8] MODFREQDIV_EXPONENT
Modulation Frequency Divider - Mantissa	CM_SSC_MODFREQDIV_DPLL_MPU[6:0] MODFREQDIV_MANTISSA

### 3.6.3.8 DPLL\_IVA Description

#### 3.6.3.8.1 Overview

Figure 3-45 shows the overview of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for a functional overview of a generic DPLL module.

Figure 3-45. DPLL\_IVA Overview



prcm-046

#### 3.6.3.8.2 Synthesized Clock Parameters

This section presents the clock synthesis and clock out divider parameters of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for an explanation of the clock synthesis and output divider parameters of the DPLL module.

Table 3-71 lists the clock synthesis parameters of the DPLL.

Table 3-71. DPLL\_IVA Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_IVA[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_IVA[6:0] DPLL_DIV

Table 3-72 lists the clock output divider parameters of the DPLL.

Table 3-72. DPLL\_IVA Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUTX2_M4	Status	CM_DIV_M4_DPLL_IVA[9] ST_HSDIVIDER_CLKOUT1
CLKOUTX2_M4	Gating Control	CM_DIV_M4_DPLL_IVA[8] HSDIVIDER_CLKOUT1_GATE_CTRL
CLKOUTX2_M4	Divider Acknowledge	CM_DIV_M4_DPLL_IVA[5] HSDIVIDER_CLKOUT1_DIVCHACK
CLKOUTX2_M4	Divider Control	CM_DIV_M4_DPLL_IVA[4:0] HSDIVIDER_CLKOUT1_DIV
CLKOUTX2_M5	Status	CM_DIV_M5_DPLL_IVA[9] ST_HSDIVIDER_CLKOUT2
CLKOUTX2_M5	Gating Control	CM_DIV_M5_DPLL_IVA[8] HSDIVIDER_CLKOUT2_GATE_CTRL
CLKOUTX2_M5	Divider Acknowledge	CM_DIV_M5_DPLL_IVA[5] HSDIVIDER_CLKOUT2_DIVCHACK
CLKOUTX2_M5	Divider Control	CM_DIV_M5_DPLL_IVA[4:0] HSDIVIDER_CLKOUT2_DIV

### 3.6.3.8.3 Power Modes

This section identifies the operating modes supported by the DPLL and also identify the control bit fields to set its operating modes. See [Section 3.6.3.3.2, Enable Control, Status and Low-Power Operation Mode](#) and [Section 3.6.3.3.3, DPLL Power Modes](#) for explanation on the DPLL operating modes, and associated control and status features.

[Table 3-73](#) presents the operating modes supported by the DPLL.

**Table 3-73. DPLL\_IVA Modes**

Low-Power Stop	Fast-Relock Stop	MN Bypass	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available (Default)	Available	Available	Available

[Table 3-74](#) presents the control bit fields for the operating mode control of the DPLL.

**Table 3-74. DPLL\_IVA Mode Control Parameters**

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_IVA[10] DPLL_LP_MODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_IVA[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_IVA[2:0] AUTO_DPLL_MODE

### 3.6.3.8.4 Recalibration

[Table 3-75](#) lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. See [Section 3.6.3.3.4, DPLL Recalibration](#), for an explanation of the DPLL recalibration feature.

**Table 3-75. DPLL\_IVA Recalibration Feature Parameters**

Parameter Name	Control/Status Bit Field
Recalibration - Enable Control	CM_CLKMODE_DPLL_IVA[8] DPLL_DRIFTGUARD_EN
Recalibration - Interrupt Status	PRM_IRQSTATUS_MPU_A9[2] DPLL_IVA_RECAL_ST
Recalibration - Interrupt Mask Control	PRM_IRQENABLE_MPU_A9[2] DPLL_IVA_RECAL_EN
Recalibration - Interrupt Status	PRM_IRQSTATUS_MPU_M3[2] DPLL_IVA_RECAL_ST
Recalibration - Interrupt Mask Control	PRM_IRQENABLE_MPU_M3[2] DPLL_IVA_RECAL_EN

### 3.6.3.8.5 Spread Spectrum Clocking

[Table 3-76](#) lists the control bit fields associated with the SSC features of the DPLL. See [Section 3.6.3.3.5, DPLL Spread Spectrum Clocking](#), for an explanation of the DPLL SSC feature.

**Table 3-76. DPLL\_IVA Spread-Spectrum Clocking Feature Parameters**

Parameter Name	Control Bit Field
SSC Downspread	CM_CLKMODE_DPLL_IVA[14] DPLL_SSC_DOWNSPREAD
SSC Acknowledge	CM_CLKMODE_DPLL_IVA[13] DPLL_SSC_ACK
SSC Enable Control	CM_CLKMODE_DPLL_IVA[12] DPLL_SSC_EN
Modulation Step Size Setting	CM_SSC_DELTAMSTEP_DPLL_IVA[19:0] DELTAMSTEP
Modulation Frequency Divider - Exponent	CM_SSC_MODFREQDIV_DPLL_IVA[10:8] MODFREQDIV_EXPONENT
Modulation Frequency Divider - Mantissa	CM_SSC_MODFREQDIV_DPLL_IVA[6:0] MODFREQDIV_MANTISSA

### 3.6.3.8.6 Output Power Down

[Table 3-77](#) lists the control bit fields for the output power down features of the DPLL. See [Section 3.6.3.3.6, DPLL Output Power-Down](#), for an explanation of the DPLL output power-down feature.

**Table 3-77. DPLL\_IVA Clock Output Power-down Parameters**

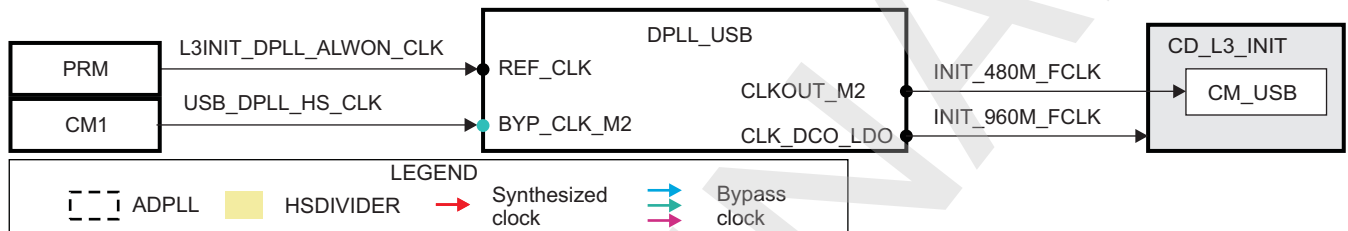
Parameter Name	Control/Status Bit Field
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M4_DPLL_IVA[12] HSDIVIDER_CLKOUT1_PWDN
Clock Divider - HSDIVIDER-Power Down Control	CM_DIV_M5_DPLL_IVA[12] HSDIVIDER_CLKOUT2_PWDN

### 3.6.3.9 DPLL\_USB Description

#### 3.6.3.9.1 Overview

Figure 3-46 shows the overview of the DPLL. See Section 3.6.3.3, *Generic DPLL Overview*, for a functional overview of a generic DPLL module.

**Figure 3-46. DPLL\_USB Overview**



prom-047

#### 3.6.3.9.2 Synthesized Clock Parameters

This section presents the clock synthesis and clock out divider parameters of the DPLL. See Section 3.6.3.3, *Generic PDLL Overview*, for an explanation of the clock synthesis and output divider parameters of the DPLL module.

Table 3-78 lists the clock synthesis parameters of the DPLL.

**Table 3-78. DPLL\_USB Clock Synthesis Parameters**

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_USB[19:8] DPLL_MULT
N	CM_CLKSEL_DPLL_USB[7:0] DPLL_DIV

Table 3-79 lists the clock output divider parameters of the DPLL.

**Table 3-79. DPLL\_USB Clock Output Parameters**

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_USB[9] ST_DPLL_CLKOUT
CLKOUT_M2	Gating Control	CM_DIV_M2_DPLL_USB[8] DPLL_CLKOUT_GATE_CTRL
CLKOUT_M2	Divider Acknowledge	CM_DIV_M2_DPLL_USB[7] DPLL_CLKOUT_DIVCHACK
CLKOUT_M2	Divider Control	CM_DIV_M2_DPLL_USB[6:0] DPLL_CLKOUT_DIV

#### 3.6.3.9.3 Power Modes

This section identifies the operating modes supported by the DPLL and also identifies the control bit fields to set its operating modes. See Section 3.6.3.3.2, *Enable Control, Status and Low-Power Operation Mode*, and Section 3.6.3.3.3, *DPLL Power Modes*, for an explanation of the DPLL operating modes, and associated control and status features.

Table 3-80 lists the operating modes supported by the DPLL.

**Table 3-80. DPLL\_USB Modes**

Low-Power Stop	Fast-Relock Stop	MN Bypass	Low-Power Bypass	Fast-Relock Bypass	Lock
Available	Not available	Available (Default)	Available	Not available	Available

Table 3-81 lists the control bit fields for the operating mode control of the DPLL.

**Table 3-81. DPLL\_USB Mode Control Parameters**

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_USB[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_USB[2:0] AUTO_DPLL_MODE

### 3.6.3.9.4 Spread Spectrum Clocking

Table 3-82 lists the control bit fields associated with the SSC features of the DPLL. See Section 3.6.3.3.5, *DPLL Spread Spectrum Clocking*, for an explanation of the DPLL SSC feature.

**Table 3-82. DPLL\_USB Spread-Spectrum Clocking Feature Parameters**

Parameter Name	Control Bit Field
SSC Downspread	CM_CLKMODE_DPLL_USB[14] DPLL_SSC_DOWNSPREAD
SSC Acknowledge	CM_CLKMODE_DPLL_USB[13] DPLL_SSC_ACK
SSC Enable Control	CM_CLKMODE_DPLL_USB[12] DPLL_SSC_EN
Modulation Step Size Setting	CM_SSC_DELTAMSTEP_DPLL_USB[19:0] DELTAMSTEP
Modulation Frequency Divider - Exponent	CM_SSC_MODFREQDIV_DPLL_USB[10:8] MODFREQDIV_EXPONENT
Modulation Frequency Divider - Mantissa	CM_SSC_MODFREQDIV_DPLL_USB[6:0] MODFREQDIV_MANTISSA

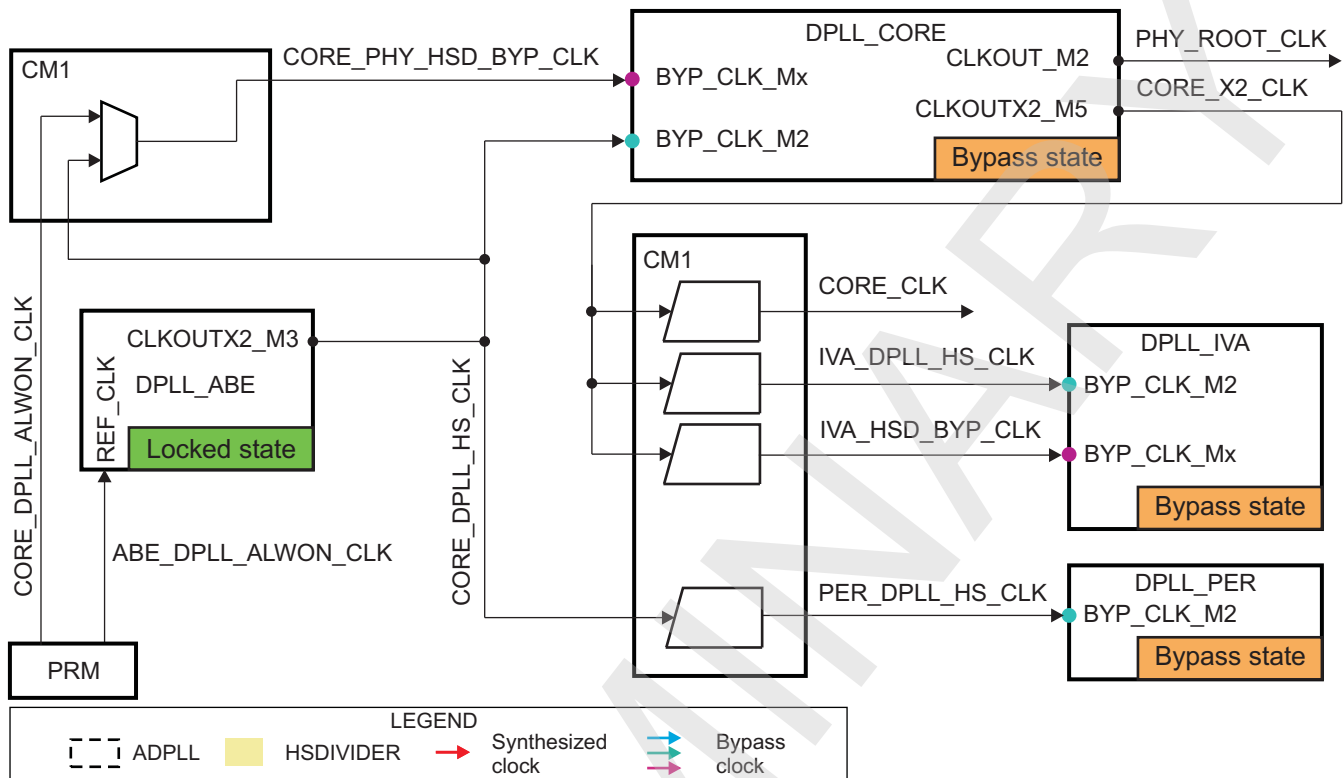
### 3.6.3.10 DPLLs Cascading

To reduce the device power consumption in certain low-power use cases of the device, the DPLLs in the device can be cascaded so that functional clock to a power domain is provided by a DPLL upstream without locking the DPLL associated with the domain. In such a case, the unlocked DPLLs can be kept in the low-power bypass mode shows the scheme for cascading DPLLs.

Figure 3-47 shows a generic scheme for cascading DPLLs.



Figure 3-47. DPLLs Cascading Overview



prom-049

The sequence of enabling the DPLL cascading is given below:

1. DPLL\_ABE is locked by software request. Other DPLLs down stream must not be moved to bypass mode before DPLL\_ABE is locked. This ensures that the output clocks of these DPLLs are not stopped when they switch to the DPLL\_ABE output clock.
2. Internal EMIF shadow registers are set to take PHY clock and DLL reference clock derived from DPLL\_ABE.
3. Bypass paths for DPLL\_IVA is set up by configuring the programmable dividers in CM1 responsible for providing the bypass clock. The setting is based upon the OPP requirements.
4. DPLL\_IVA is put into bypass mode by software (any of the bypass mode is allowed) and the output of the DPLL, if already enabled, switches from locked clock to bypass clock from DPLL\_CORE
5. DPLL\_PER is put into bypass mode as well by software, if required. The outputs of the DPLL, if enabled, switch to the divided clock of DPLL\_ABE CLKOUTX2\_M3.
6. DPLL\_CORE is set in bypass mode by configuring the bypass clocks input mux and the [CM\\_SHADOW\\_FREQ\\_CONFIG1\[10:8\]](#) DPLL\_CORE\_DPLL\_EN bit field to a bypass mode in the DVFS shadow register in CM1 and setting [CM\\_SHADOW\\_FREQ\\_CONFIG1\[0\]](#) FREQ\_UPDATE = 1 by software. This step triggers the handshake with EMIF to safely unlock the DPLL and move the output clock of DPLL\_CORE to CORE\_DPLL\_HS\_CLK. Ensure that the CORE\_CLK divider in CM1 is correctly programmed to handle the OPP needs of the CORE power domain before this step.
7. Software can then switch the L4\_WKUP interface clock from SYS\_CLK to ABE\_LP\_CLK to allow for deasserting of the SYS\_CLK request line to SCRM, if SYS\_CLK is not requested by any other dependency.

The sequence of disabling the DPLL cascading is given below:

1. Software switches the L4\_WKUP interface clock from ABE\_LP\_CLK to SYS\_CLK requesting the restart of SYS\_CLK by the SCRM, if not already present.
2. Software programs the EMIF internal shadow registers to prepare the switchback of EMIF interface from using DPLL\_ABE clock to DPLL\_CORE clock.
3. DPLL\_IVA bypass path is correctly set for the targeted DPLL\_CORE frequency by programming the

bypass dividers for the DPLL in CM1.

4. DPLL\_CORE is relocked using the DVFS shadow register in CM1. The PRCM module ensures that this step is executed with proper handshake with EMIF to allow for a frequency update on the EMIF interface.
5. Other DPLLs are locked as well if required, to fully exit from the DPLL cascading mode.
6. DPLL\_ABE might be forced into STOPMODE or bypass mode by software or may enter one of the auto modes if so programmed.

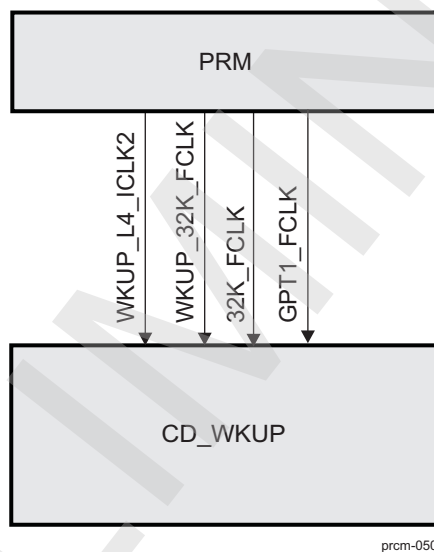
### 3.6.4 CD\_WKUP Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.4.1 Overview

Figure 3-48 shows the overview of the clock domain.

**Figure 3-48. CD\_WKUP Overview**



#### 3.6.4.2 Clock Domain Modes

Table 3-83 lists the clock domain modes supported by the clock domain.

**Table 3-83. CD\_WKUP Clock Domain Modes**

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Not available	Available

Table 3-84 lists the clock domain state transition control and status bits for the clock in this clock domain.

**Table 3-84. CD\_WKUP Control and Status Parameters**

Parameter Name	Control/Status Bit Field
ABE_LP_CLK clock status	CM_WKUP_CLKSTCTRL[9] CLKACTIVITY_ABE_LP_CLK
WKUP_L4_ICLK2 clock status	CM_WKUP_CLKSTCTRL[12] CLKACTIVITY_WKUP_L4_ICLK2
SYS_CLK clock status	CM_WKUP_CLKSTCTRL[8] CLKACTIVITY_SYS_CLK
WKUP_32K_FCLK clock status	CM_WKUP_CLKSTCTRL[11] CLKACTIVITY_WKUP_32K_FCLK
Clock Domain State Transition Control	CM_WKUP_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.4.3 Clock Domain Dependency

CD\_WKUP has no static dependency or dynamic dependency with any other clock domain of the device.

#### 3.6.4.3.1 Wake-Up Dependency

Table 3-85 lists the wake-up dependency settings for the modules of this clock domain.

**Table 3-85. CD\_WKUP Wake-Up Dependency Association Parameters**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO1	CD_WKUP	CD_DSP	Always enabled	PM_WKUP_GPIO1_WKDEP[6] WKUPDEP_GPIO1_IRQ2_DSP	Read only
GPIO1	CD_WKUP	CD_CORTEXM3	Disabled	PM_WKUP_GPIO1_WKDEP[1] WKUPDEP_GPIO1_IRQ1_MPU_M3	Read/Write
GPIO1	CD_WKUP	CD_CORTEXA9	Disabled	PM_WKUP_GPIO1_WKDEP[0] WKUPDEP_GPIO1_IRQ1_MPU	Read/Write
KEYBOARD	CD_WKUP	CD_CORTEXA9	Always enabled	PM_WKUP_KEYBOARD_WKDEP[0] WKUPDEP_KEYBOARD_MPU	Read only
GPTIMER1	CD_WKUP	CD_CORTEXA9	Always enabled	PM_WKUP_GPTIMER1_WKDEP[0] WKUPDEP_TIMER1_MPU	Read only
WDTIMER2	CD_WKUP	CD_CORTEXM3	Disabled	PM_WKUP_WDTIMER2_WKDEP[1] WKUPDEP_WDT2_MPU_M3	Read/Write
WDTIMER2	CD_WKUP	CD_CORTEXA9	Disabled	PM_WKUP_WDTIMER2_WKDEP[0] WKUPDEP_WDT2_MPU	Read/Write

#### 3.6.4.4 Clock Domain Module Attributes

Table 3-86 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-86. CD\_WKUP Modules Clocks Association**

Module	Clock	Clock Type
SYSCONFIG_PADCONF_WKUP	WKUP_L4_ICLK2	Functional
BANDGAP	WKUP_32K_FCLK	Functional
SYSCONFIG_GENERAL_WKUP	WKUP_L4_ICLK2	Functional
GPIO1	WKUP_L4_ICLK2	Interface
	WKUP_32K_FCLK	Functional
Keyboard	WKUP_L4_ICLK2	Interface
	WKUP_32K_FCLK	Functional
SAR_RAM	WKUP_L4_ICLK2	Interface
32KTIMER	32K_FCLK	Functional
	WKUP_L4_ICLK2	Interface
GPTIMER1	GPT1_FCLK	Functional
	WKUP_L4_ICLK2	Interface

**Table 3-86. CD\_WKUP Modules Clocks Association (continued)**

Module	Clock	Clock Type
WDTIMER2	WKUP_L4_ICLK2	Interface
	WKUP_32K_FCLK	Functional
L4_WKUP interconnect	WKUP_L4_ICLK2	Interface

Table 3-87 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-87. CD\_WKUP Modules Wake-Up Request**

Module	Wake-Up Feature
SYSCTRL_PADCONF_WKUP	None
BANDGAP	None
SYSCTRL_GENERAL_WKUP	None
GPIO1	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, DSP -IRQ)
Keyboard	Slave wake-up request (MPU_A9 -IRQ)
SAR_RAM	None
32KTIMER	None
GPTIMER1	Slave wake-up request (MPU_A9 -IRQ)
WDTIMER2	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)
L4_WKUP interconnect	None

Table 3-88 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-88. CD\_WKUP Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
GPIO1	Slave	CM_WKUP_GPIO1_CLKCTRL [17:16] IDLEST	Idle status
KEYBOARD	Slave	CM_WKUP_KEYBOARD_CLKCTRL [17:16] IDLEST	Idle status
SAR_RAM	Slave	CM_WKUP_SARRAM_CLKCTRL [17:16] IDLEST	Idle status
32KTIMER	Slave	CM_WKUP_32KTIMER_CLKCTRL [17:16] IDLEST	Idle status
GPTIMER1	Slave	CM_WKUP_GPTIMER1_CLKCTRL [17:16] IDLEST	Idle status
WDTIMER2	Slave	CM_WKUP_WDTIMER2_CLKCTRL [17:16] IDLEST	Idle status
L4_WKUP interconnect	Slave	CM_WKUP_L4WKUP_CLKCTRL [17:16] IDLEST	Idle status

Table 3-89 lists the supported clock management modes and associated software control bit fields for each module of the power domain.

**Table 3-89. CD\_WKUP Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
GPIO1	Available	Available	N/A	CM_WKUP_GPIO1_CLKCTRL [1:0] MODULEMODE	Read/Write
KEYBOARD	Available	N/A	Available	CM_WKUP_KEYBOARD_CLKCTRL [1:0] MODULEMODE	Read/Write
SAR_RAM	N/A	Available	N/A	CM_WKUP_SARRAM_CLKCTRL [1:0] MODULEMODE	Read only

**Table 3-89. CD\_WKUP Modules Slave Clock-Management Modes and Control (continued)**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
32KTIMER	N/A	Available	N/A	CM_WKUP_32KTIMER_CLKCTRL[1:0] MODULEMODE	Read only
GPTIMER1	Available	N/A	Available	CM_WKUP_GPTIMER1_CLKCTRL[1:0] MODULEMODE	Read/Write
WDTIMER2	Available	N/A	Available	CM_WKUP_WDTIMER2_CLKCTRL[1:0] MODULEMODE	Read/Write
L4_WKUP interconnect	N/A	Available	N/A	CM_WKUP_L4WKUP_CLKCTRL[1:0] MODULEMODE	Read only

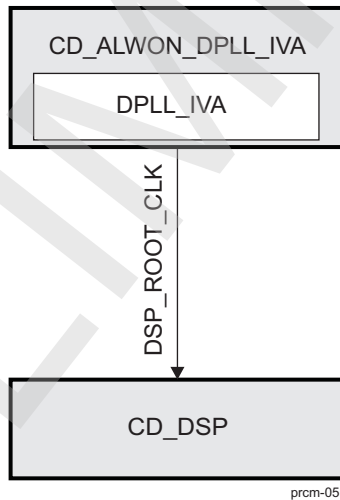
### 3.6.5 CD\_DSP Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.5.1 Overview

Figure 3-49 shows the overview of the clock domain.

**Figure 3-49. CD\_DSP Overview**



#### 3.6.5.2 Clock Domain Modes

Table 3-90 lists the clock domain modes supported by the clock domain.

**Table 3-90. CD\_DSP Clock Domain Modes**

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-91 lists the clock domain state transition control and status bits for the clock in this clock domain.

**Table 3-91. CD\_DSP Control and Status Parameters**

Parameter Name	Control/Status Bit Field
DSP_ROOT_CLK clock status	CM_DSP_CLKSTCTRL[8] CLKACTIVITY_DSP_ROOT_CLK
Clock Domain State Transition Control	CM_DSP_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.5.3 Clock Domain Dependency

CD\_DSP has no module wake-up dependency with any other clock domain of the device.

#### 3.6.5.3.1 Static Dependency

Table 3-92 lists the static dependency of the clock domain with respect to other clock domains of the device.

**Table 3-92. CD\_DSP Static Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_ABE	Disabled	CM_DSP_STATICDEP[3] ABE_STATDEP	Read/Write
CD_CAM	Always disabled	CM_DSP_STATICDEP[9] ISS_STATDEP	Read only
CD_IVAHD	Disabled	CM_DSP_STATICDEP[2] IVAHD_STATDEP	Read/Write
CD_L3_1	Disabled	CM_DSP_STATICDEP[5] L3_1_STATDEP	Read/Write
CD_L3_2	Disabled	CM_DSP_STATICDEP[6] L3_2_STATDEP	Read/Write
CD_L3_INIT	Disabled	CM_DSP_STATICDEP[7] L3INIT_STATDEP	Read/Write
CD_L4_CFG	Disabled	CM_DSP_STATICDEP[12] L4CFG_STATDEP	Read/Write
CD_L4_PER	Disabled	CM_DSP_STATICDEP[13] L4PER_STATDEP	Read/Write
CD_EMIF	Disabled	CM_DSP_STATICDEP[4] MEMIF_STATDEP	Read/Write

#### 3.6.5.3.2 Dynamic Dependency

Table 3-93 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-93. CD\_DSP Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_ABE	Always enabled	CM_DSP_DYNAMICDEP[3] ABE_DYNDEP	Read only
CD_IVAHD	Always enabled	CM_DSP_DYNAMICDEP[2] IVAHD_DYNDEP	Read only
CD_L3_1	Always enabled	CM_DSP_DYNAMICDEP[5] L3_1_DYNDEP	Read only

### 3.6.5.4 Clock Domain Module Attributes

Table 3-94 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-94. CD\_DSP Modules Clocks Association**

Module	Clock	Clock Type
DSP	DSP_ROOT_CLK	Interface and functional

Table 3-95 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-95. CD\_DSP Modules Wake-Up Request**

Module	Wake-Up Feature
DSP	Master wake-up request

Table 3-96 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-96. CD\_DSP Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
DSP	Master/Slave	CM_DSP_DSP_CLKCTRL[18] STBYST	Standby status
		CM_DSP_DSP_CLKCTRL[17:16] IDLEST	Idle status

Table 3-97 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-97. CD\_DSP Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DSP	Available	Available	N/A	CM_DSP_DSP_CLKCTRL[1:0] MODULEMODE	Read/Write

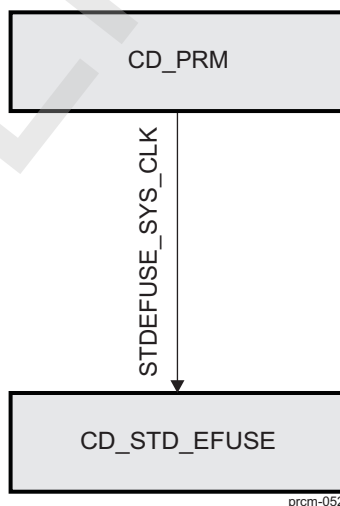
### 3.6.6 CD\_STD\_EFUSE Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.6.1 Overview

Figure 3-50 shows the overview of the clock domain.

**Figure 3-50. CD\_STD\_EFUSE Overview**



CD\_STD\_EFUSE is hardware-controlled by the PRCM module and has no software control or status bit fields. Similarly, it has no static, dynamic and wake-up dependency with respect to other clock domains of the device

### 3.6.6.2 Clock Domain Module Attributes

The STD\_EFUSE module of the clock domain is automatically controlled by the PRCM module hardware and has no software configurable module mode control or status bit fields.

Table 3-98 identifies for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-98. CD\_STD\_EFUSE Modules Clocks Association**

Module	Clock	Clock Type
STD_EFUSE	STDEFUSE_SYS_CLK	Functional

Table 3-99 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-99. CD\_STD\_EFUSE Modules Wake-Up Request**

Module	Wake-Up Feature
STD_EFUSE	None

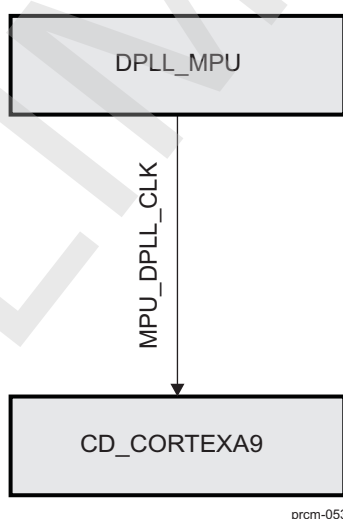
### 3.6.7 CD\_CORTEXA9 Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.7.1 Overview

Figure 3-51 shows the overview of the clock domain.

**Figure 3-51. CD\_CORTEXA9 Overview**



#### 3.6.7.2 Clock Domain Modes

Table 3-100 lists the clock domain modes supported by the clock domain.

**Table 3-100. CD\_CORTEXA9 Clock Domain Modes**

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available



Table 3-101 lists the clock domain state transition control and status bits for the clock in this clock domain.

**Table 3-101. CD\_CORTEXA9 Control and Status Parameters**

Parameter Name	Control/Status Bit Field
MPU_DPLL_CLK clock status	CM_MPU_CLKSTCTRL[9] CLKACTIVITY_MPU_DPLL_CLK
Clock Domain State Transition Control	CM_MPU_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.7.3 Clock Domain Dependency

CD\_CORTEXA9 has no module wake-up dependency with any other clock domain of the device.

#### 3.6.7.3.1 Static Dependency

Table 3-102 lists the static dependency of the clock domain with respect to other clock domains of the device.

**Table 3-102. CD\_CORTEXA9 Static Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_ABE	Disabled	CM_MPU_STATICDEP[3] ABE_STATDEP	Read/Write
CD_C2C	Always disabled	CM_MPU_STATICDEP[18] C2C_STATDEP	Read only
CD_DSS	Disabled	CM_MPU_STATICDEP[8] DSS_STATDEP	Read/Write
CD_CORTEXM3	Disabled	CM_MPU_STATICDEP[0] MPU_M3_STATDEP	Read/Write
CD_SGX	Disabled	CM_MPU_STATICDEP[10] SGX_STATDEP	Read/Write
CD_CAM	Always disabled	CM_MPU_STATICDEP[9] ISS_STATDEP	Read only
CD_IVAHD	Disabled	CM_MPU_STATICDEP[2] IVAHD_STATDEP	Read/Write
CD_L3_1	Enabled	CM_MPU_STATICDEP[5] L3_1_STATDEP	Read/Write
CD_L3_2	Enabled	CM_MPU_STATICDEP[6] L3_2_STATDEP	Read/Write
CD_L3_INIT	Enabled	CM_MPU_STATICDEP[7] L3INIT_STATDEP	Read/Write
CD_L4_CFG	Enabled	CM_MPU_STATICDEP[12] L4CFG_STATDEP	Read/Write
CD_L4_PER	Enabled	CM_MPU_STATICDEP[13] L4PER_STATDEP	Read/Write
CD_EMIF	Enabled	CM_MPU_STATICDEP[4] MEMIF_STATDEP	Read/Write
CD_DMA	Always disabled	CM_MPU_STATICDEP[11] SDMA_STATDEP	Read only
CD_DSP	Disabled	CM_MPU_STATICDEP[1] DSP_STATDEP	Read/Write

#### 3.6.7.3.2 Dynamic Dependency

Table 3-103 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-103. CD\_CORTEXA9 Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_ABE	Always enabled	CM_MPU_DYNAMICDEP[3] ABE_DYNDEP	Read only
CD_L3_1	Always enabled	CM_MPU_DYNAMICDEP[5] L3_1_DYNDEP	Read only
CD_EMIF	Always enabled	CM_MPU_DYNAMICDEP[4] MEMIF_DYNDEP	Read only

### 3.6.7.4 Clock Domain Module Attributes

Table 3-104 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-104. CD\_CORTEXA9 Modules Clocks Association**

Module	Clock	Clock Type
CORTEXA9	MPU_DPLL_CLK	Interface and Functional

Table 3-105 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-105. CD\_CORTEXA9 Modules Wake-Up Request**

Module	Wake-Up Feature
CORTEXA9	Master wake-up request

Table 3-106 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-106. CD\_CORTEXA9 Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
CORTEXA9	Master/Slave	CM_MPU_MPU_CLKCTRL[18] STBYST	Standby status
		CM_MPU_MPU_CLKCTRL[17: 16] IDLEST	Idle status

Table 3-107 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-107. CD\_CORTEXA9 Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
CORTEXA9	N/A	Available	N/A	CM_MPU_MPU_CL KCTRL[1:0] MODULEMODE	Read only

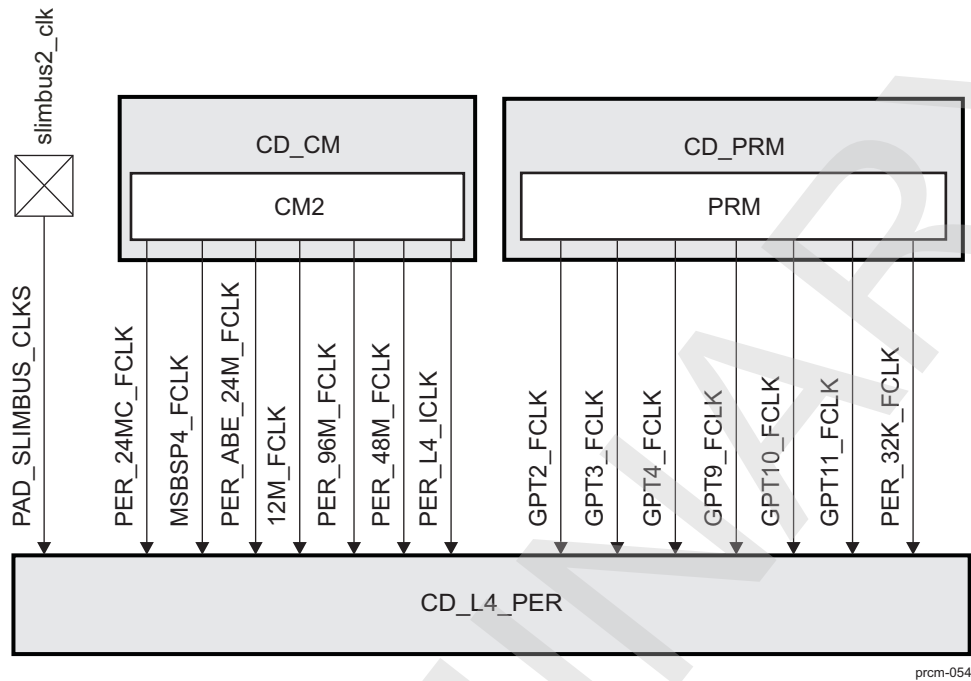
## 3.6.8 CD\_L4\_PER Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

### 3.6.8.1 Overview

Figure 3-52 shows the overview of the clock domain.

Figure 3-52. CD\_L4\_PER Overview



3.6.8.2 Clock Domain Modes

Table 3-108 lists the clock domain modes supported by the clock domain.

Table 3-108. CD\_L4\_PER Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-109 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-109. CD\_L4\_PER Control and Status Parameters

Parameter Name	Control/Status Bit Field
GPT10_FCLK clock status	CM_L4PER_CLKSTCTRL[9] CLKACTIVITY_GPT10_FCLK
GPT11_FCLK clock status	CM_L4PER_CLKSTCTRL[10] CLKACTIVITY_GPT11_FCLK
GPT2_FCLK clock status	CM_L4PER_CLKSTCTRL[11] CLKACTIVITY_GPT2_FCLK
GPT3_FCLK clock status	CM_L4PER_CLKSTCTRL[12] CLKACTIVITY_GPT3_FCLK
GPT4_FCLK clock status	CM_L4PER_CLKSTCTRL[13] CLKACTIVITY_GPT4_FCLK
GPT9_FCLK clock status	CM_L4PER_CLKSTCTRL[14] CLKACTIVITY_GPT9_FCLK
12M_FCLK clock status	CM_L4PER_CLKSTCTRL[15] CLKACTIVITY_12M_FCLK_FCLK
PER_L4_ICLK clock status	CM_L4PER_CLKSTCTRL[8] CLKACTIVITY_L4_PER_ICLK
PER_24MC_FCLK clock status	CM_L4PER_CLKSTCTRL[16] CLKACTIVITY_PER_24MC_FCLK
PER_32K_FCLK clock status	CM_L4PER_CLKSTCTRL[17] CLKACTIVITY_PER_32K_FCLK
PER_48M_FCLK clock status	CM_L4PER_CLKSTCTRL[18] CLKACTIVITY_PER_48M_FCLK
PER_96M_FCLK clock status	CM_L4PER_CLKSTCTRL[19] CLKACTIVITY_PER_96M_FCLK
PER_ABE_24M_FCLK clock status	CM_L4PER_CLKSTCTRL[25] CLKACTIVITY_PER_ABE_24M_FCLK
MCBSP4_FCLK clock status	CM_L4PER_CLKSTCTRL[22] CLKACTIVITY_PER_MCBSP4_FCLK
Clock Domain State Transition Control	CM_L4PER_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.8.3 Clock Domain Dependency

CD\_L4\_PER has no static dependency with any other clock domain of the device.

#### 3.6.8.3.1 Dynamic Dependency

Table 3-110 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-110. CD\_L4\_PER Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_DSS	Always enabled	CM_L4PER_DYNAMICDEP[8] DSS_DYNDEP	Read only
CD_L3_INIT	Always enabled	CM_L4PER_DYNAMICDEP[7] L3_INIT_DYNDEP	Read only

#### 3.6.8.3.2 Wake-Up Dependency

Table 3-111 lists the wake-up dependency settings for the modules of this clock domain.

**Table 3-111. CD\_L4\_PER Wake-Up Dependency Association Parameters**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPTIMER10	CD_L4_PER	CD_CORTEXA9	Always enabled	PM_L4PER_GPTIM ER10_WKDEP[0] WKUPDEP_DMTIM ER10_MPU	Read only
GPTIMER11	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_GPTIM ER11_WKDEP[1] WKUPDEP_DMTIM ER11_MPU_M3	Read/Write
GPTIMER11	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_GPTIM ER11_WKDEP[0] WKUPDEP_DMTIM ER11_MPU	Read/Write
GPTIMER2	CD_L4_PER	CD_CORTEXA9	Always enabled	PM_L4PER_GPTIM ER2_WKDEP[0] WKUPDEP_DMTIM ER2_MPU	Read only
GPTIMER3	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_GPTIM ER3_WKDEP[1] WKUPDEP_DMTIM ER3_MPU_M3	Read/Write
GPTIMER3	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_GPTIM ER3_WKDEP[0] WKUPDEP_DMTIM ER3_MPU	Read/Write
GPTIMER4	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_GPTIM ER4_WKDEP[1] WKUPDEP_DMTIM ER4_MPU_M3	Read/Write
GPTIMER4	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_GPTIM ER4_WKDEP[0] WKUPDEP_DMTIM ER4_MPU	Read/Write
GPTIMER9	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_GPTIM ER9_WKDEP[1] WKUPDEP_DMTIM ER9_MPU_M3	Read/Write
GPTIMER9	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_GPTIM ER9_WKDEP[0] WKUPDEP_DMTIM ER9_MPU	Read/Write

**Table 3-111. CD\_L4\_PER Wake-Up Dependency Association Parameters (continued)**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO2	CD_L4_PER	CD_DSP	Always enabled	PM_L4PER_GPIO2_WKDEP[6] WKUPDEP_GPIO2_IRQ2_DSP	Read only
GPIO2	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_GPIO2_WKDEP[1] WKUPDEP_GPIO2_IRQ1_MPU_M3	Read/Write
GPIO2	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_GPIO2_WKDEP[0] WKUPDEP_GPIO2_IRQ1_MPU	Read/Write
GPIO3	CD_L4_PER	CD_DSP	Always enabled	PM_L4PER_GPIO3_WKDEP[6] WKUPDEP_GPIO3_IRQ2_DSP	Read only
GPIO3	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_GPIO3_WKDEP[0] WKUPDEP_GPIO3_IRQ1_MPU	Read only
GPIO4	CD_L4_PER	CD_DSP	Always enabled	PM_L4PER_GPIO4_WKDEP[6] WKUPDEP_GPIO4_IRQ2_DSP	Read only
GPIO4	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_GPIO4_WKDEP[0] WKUPDEP_GPIO4_IRQ1_MPU	Read only
GPIO5	CD_L4_PER	CD_DSP	Always enabled	PM_L4PER_GPIO5_WKDEP[6] WKUPDEP_GPIO5_IRQ2_DSP	Read only
GPIO5	CD_L4_PER	CD_CORTEXA9	Always enabled	PM_L4PER_GPIO5_WKDEP[0] WKUPDEP_GPIO5_IRQ1_MPU	Read only
GPIO6	CD_L4_PER	CD_DSP	Always enabled	PM_L4PER_GPIO6_WKDEP[6] WKUPDEP_GPIO6_IRQ2_DSP	Read only
GPIO6	CD_L4_PER	CD_CORTEXA9	Always enabled	PM_L4PER_GPIO6_WKDEP[0] WKUPDEP_GPIO6_IRQ1_MPU	Read only
I2C1	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Always enabled	PM_L4PER_I2C1_WKDEP[7] WKUPDEP_I2C1_DMA_SDMA	Read only
I2C1	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_I2C1_WKDEP[1] WKUPDEP_I2C1_IRQ_MPU_M3	Read/Write
I2C1	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_I2C1_WKDEP[0] WKUPDEP_I2C1_IRQ_MPU	Read/Write
I2C2	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Always enabled	PM_L4PER_I2C2_WKDEP[7] WKUPDEP_I2C2_DMA_SDMA	Read only

**Table 3-111. CD\_L4\_PER Wake-Up Dependency Association Parameters (continued)**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
I2C2	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_I2C2_WKDEP[1] WKUPDEP_I2C2_I RQ_MPU_M3	Read/Write
I2C2	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_I2C2_WKDEP[0] WKUPDEP_I2C2_I RQ_MPU	Read/Write
I2C3	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Always enabled	PM_L4PER_I2C3_WKDEP[7] WKUPDEP_I2C3_D MA_SDMA	Read only
I2C3	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_I2C3_WKDEP[1] WKUPDEP_I2C3_I RQ_MPU_M3	Read/Write
I2C3	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_I2C3_WKDEP[0] WKUPDEP_I2C3_I RQ_MPU	Read/Write
I2C4	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Always enabled	PM_L4PER_I2C4_WKDEP[7] WKUPDEP_I2C4_D MA_SDMA	Read only
I2C4	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_I2C4_WKDEP[1] WKUPDEP_I2C4_I RQ_MPU_M3	Read/Write
I2C4	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_I2C4_WKDEP[0] WKUPDEP_I2C4_I RQ_MPU	Read/Write
MCBSP4	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_MCBSP4_WKDEP[3] WKUPDEP_MCBSP4_SDMA	Read/Write
MCBSP4	CD_L4_PER	CD_DSP	Disabled	PM_L4PER_MCBSP4_WKDEP[2] WKUPDEP_MCBSP4_DSP	Read/Write
MCBSP4	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_MCBSP4_WKDEP[0] WKUPDEP_MCBSP4_MPU	Read/Write
MCSP11	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_MCSP11_WKDEP[3] WKUPDEP_MCSP11_SDMA	Read/Write
MCSP11	CD_L4_PER	CD_DSP	Disabled	PM_L4PER_MCSP11_WKDEP[2] WKUPDEP_MCSP11_DSP	Read/Write
MCSP11	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_MCSP11_WKDEP[1] WKUPDEP_MCSP11_MPU_M3	Read/Write
MCSP11	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_MCSP11_WKDEP[0] WKUPDEP_MCSP11_MPU	Read/Write

**Table 3-111. CD\_L4\_PER Wake-Up Dependency Association Parameters (continued)**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
MCSP12	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_MCSP12_WKDEP[3] WKUPDEP_MCSP12_SDMA	Read/Write
MCSP12	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_MCSP12_WKDEP[1] WKUPDEP_MCSP12_MPU_M3	Read/Write
MCSP12	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_MCSP12_WKDEP[0] WKUPDEP_MCSP12_MPU	Read/Write
MCSP13	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_MCSP13_WKDEP[3] WKUPDEP_MCSP13_SDMA	Read/Write
MCSP13	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_MCSP13_WKDEP[0] WKUPDEP_MCSP13_MPU	Read/Write
MCSP14	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_MCSP14_WKDEP[3] WKUPDEP_MCSP14_SDMA	Read/Write
MCSP14	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_MCSP14_WKDEP[0] WKUPDEP_MCSP14_MPU	Read/Write
HSMMC3	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_MMCS D3_WKDEP[3] WKUPDEP_MMCS D3_SDMA	Read/Write
HSMMC3	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_MMCS D3_WKDEP[1] WKUPDEP_MMCS D3_MPU_M3	Read/Write
HSMMC3	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_MMCS D3_WKDEP[0] WKUPDEP_MMCS D3_MPU	Read/Write
HSMMC4	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_MMCS D4_WKDEP[3] WKUPDEP_MMCS D4_SDMA	Read/Write
HSMMC4	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_MMCS D4_WKDEP[1] WKUPDEP_MMCS D4_MPU_M3	Read/Write
HSMMC4	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_MMCS D4_WKDEP[0] WKUPDEP_MMCS D4_MPU	Read/Write
HSMMC5	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_MMCS D5_WKDEP[3] WKUPDEP_MMCS D5_SDMA	Read/Write
HSMMC5	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_MMCS D5_WKDEP[1] WKUPDEP_MMCS D5_MPU_M3	Read/Write

**Table 3-111. CD\_L4\_PER Wake-Up Dependency Association Parameters (continued)**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
HSMMC5	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_MMCS D5_WKDEP[0] WKUPDEP_MMCS D5_MPU	Read/Write
SLIMBUS2	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_SLIMB US2_WKDEP[7] WKUPDEP_SLIMB US2_DMA_SDMA	Read/Write
SLIMBUS2	CD_L4_PER	CD_DSP	Disabled	PM_L4PER_SLIMB US2_WKDEP[6] WKUPDEP_SLIMB US2_DMA_DSP	Read/Write
SLIMBUS2	CD_L4_PER	CD_DSP	Disabled	PM_L4PER_SLIMB US2_WKDEP[2] WKUPDEP_SLIMB US2_IRQ_DSP	Read/Write
SLIMBUS2	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_SLIMB US2_WKDEP[0] WKUPDEP_SLIMB US2_IRQ_MPU	Read/Write
UART1	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_UART1 _WKDEP[3] WKUPDEP_UART1 _SDMA	Read/Write
UART1	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_UART1 _WKDEP[0] WKUPDEP_UART1 _MPU	Read/Write
UART2	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_UART2 _WKDEP[3] WKUPDEP_UART2 _SDMA	Read/Write
UART2	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_UART2 _WKDEP[0] WKUPDEP_UART2 _MPU	Read/Write
UART3	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_UART3 _WKDEP[3] WKUPDEP_UART3 _SDMA	Read/Write
UART3	CD_L4_PER	CD_DSP	Disabled	PM_L4PER_UART3 _WKDEP[2] WKUPDEP_UART3 _DSP	Read/Write
UART3	CD_L4_PER	CD_CORTEXM3	Disabled	PM_L4PER_UART3 _WKDEP[1] WKUPDEP_UART3 _MPU_M3	Read/Write
UART3	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_UART3 _WKDEP[0] WKUPDEP_UART3 _MPU	Read/Write
UART4	CD_L4_PER	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L4PER_UART4 _WKDEP[3] WKUPDEP_UART4 _SDMA	Read/Write
UART4	CD_L4_PER	CD_CORTEXA9	Disabled	PM_L4PER_UART4 _WKDEP[0] WKUPDEP_UART4 _MPU	Read/Write



### 3.6.8.4 Clock Domain Module Attributes

Table 3-112 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-112. CD\_L4\_PER Modules Clocks Association**

Module	Clock	Clock Type
GPTIMER10	GPT10_FCLK	Functional
	PER_L4_ICLK	Interface
GPTIMER11	GPT11_FCLK	Functional
	PER_L4_ICLK	Interface
GPTIMER2	GPT2_FCLK	Functional
	PER_L4_ICLK	Interface
GPTIMER3	GPT3_FCLK	Functional
	PER_L4_ICLK	Interface
GPTIMER4	GPT4_FCLK	Functional
	PER_L4_ICLK	Interface
GPTIMER9	GPT9_FCLK	Functional
	PER_L4_ICLK	Interface
ELM	PER_L4_ICLK	Interface
GPIO2	PER_L4_ICLK	Interface
	PER_32K_FCLK	Functional
GPIO3	PER_L4_ICLK	Interface
	PER_32K_FCLK	Functional
GPIO4	PER_L4_ICLK	Interface
	PER_32K_FCLK	Functional
GPIO5	PER_L4_ICLK	Interface
	PER_32K_FCLK	Functional
GPIO6	PER_L4_ICLK	Interface
	PER_32K_FCLK	Functional
HDQ	12M_FCLK	Functional
	PER_L4_ICLK	Interface and functional
I2C1	PER_L4_ICLK	Interface
	PER_96M_FCLK	Functional
I2C2	PER_L4_ICLK	Interface
	PER_96M_FCLK	Functional
I2C3	PER_L4_ICLK	Interface
	PER_96M_FCLK	Functional
I2C4	PER_L4_ICLK	Interface
	PER_96M_FCLK	Functional
L4_PER interconnect	PER_L4_ICLK	Interface
MCBSP4	PER_L4_ICLK	Interface
	MCBSP4_FCLK	Functional
MCSPI1	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional
MCSPI2	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional
MCSPI3	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional
MCSPI4	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional

**Table 3-112. CD\_L4\_PER Modules Clocks Association (continued)**

Module	Clock	Clock Type
HSMMC3	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional
HSMMC4	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional
HSMMC5	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional
SLIMBUS2	PER_L4_ICLK	Interface
	PAD_SLIMBUS_CLKS	Functional
	PER_24MC_FCLK	Functional
	PER_ABE_24M_FCLK	Functional
UART1	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional
UART2	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional
UART3	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional
UART4	PER_L4_ICLK	Interface
	PER_48M_FCLK	Functional

Table 3-113 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-113. CD\_L4\_PER Modules Wake-Up Request**

Module	Wake-Up Feature
GPTIMER10	Slave wake-up request (MPU_A9 -IRQ)
GPTIMER11	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)
GPTIMER2	Slave wake-up request (MPU_A9 -IRQ)
GPTIMER3	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)
GPTIMER4	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)
GPTIMER9	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)
ELM	None
GPIO2	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, DSP -IRQ)
GPIO3	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ)
GPIO4	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ)
GPIO5	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ)
GPIO6	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ)
HDQ	None
I2C1	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, sDMA -DMA)
I2C2	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, sDMA -DMA)
I2C3	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, sDMA -DMA)
I2C4	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, sDMA -DMA)
L4_PER interconnect	None
MCBSP4	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ and DMA, sDMA -DMA)

**Table 3-113. CD\_L4\_PER Modules Wake-Up Request (continued)**

Module	Wake-Up Feature
MCSP11	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, DSP -IRQ, sDMA -DMA)
MCSP12	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, sDMA -DMA)
MCSP13	Slave wake-up request (MPU_A9 -IRQ, sDMA -DMA)
MCSP14	Slave wake-up request (MPU_A9 -IRQ, sDMA -DMA)
HSMMC3	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, sDMA -DMA)
HSMMC4	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, sDMA -DMA)
HSMMC5	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, sDMA -DMA)
SLIMBUS2	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ and DMA, sDMA -DMA)
UART1	Slave wake-up request (MPU_A9 -IRQ, sDMA -DMA)
UART2	Slave wake-up request (MPU_A9 -IRQ, sDMA -DMA)
UART3	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, DSP -IRQ and DMA, sDMA -DMA)
UART4	Slave wake-up request (MPU_A9 -IRQ, sDMA -DMA)

Table 3-114 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-114. CD\_L4\_PER Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
GPTIMER10	Slave	CM_L4PER_GPTIMER10_CLK_CTRL[17:16] IDLEST	Idle status
GPTIMER11	Slave	CM_L4PER_GPTIMER11_CLK_CTRL[17:16] IDLEST	Idle status
GPTIMER2	Slave	CM_L4PER_GPTIMER2_CLK_CTRL[17:16] IDLEST	Idle status
GPTIMER3	Slave	CM_L4PER_GPTIMER3_CLK_CTRL[17:16] IDLEST	Idle status
GPTIMER4	Slave	CM_L4PER_GPTIMER4_CLK_CTRL[17:16] IDLEST	Idle status
GPTIMER9	Slave	CM_L4PER_GPTIMER9_CLK_CTRL[17:16] IDLEST	Idle status
ELM	Slave	CM_L4PER_ELM_CLKCTRL[17:16] IDLEST	Idle status
GPIO2	Slave	CM_L4PER_GPIO2_CLKCTRL[17:16] IDLEST	Idle status
GPIO3	Slave	CM_L4PER_GPIO3_CLKCTRL[17:16] IDLEST	Idle status
GPIO4	Slave	CM_L4PER_GPIO4_CLKCTRL[17:16] IDLEST	Idle status
GPIO5	Slave	CM_L4PER_GPIO5_CLKCTRL[17:16] IDLEST	Idle status
GPIO6	Slave	CM_L4PER_GPIO6_CLKCTRL[17:16] IDLEST	Idle status
HDQ	Slave	CM_L4PER_HDQ1W_CLKCTRL[17:16] IDLEST	Idle status
I2C1	Slave	CM_L4PER_I2C1_CLKCTRL[17:16] IDLEST	Idle status
I2C2	Slave	CM_L4PER_I2C2_CLKCTRL[17:16] IDLEST	Idle status

**Table 3-114. CD\_L4\_PER Modules Clock-Management Modes and Control (continued)**

Module	Clock-Management Protocol	Status Bit Field	Role
I2C3	Slave	CM_L4PER_I2C3_CLKCTRL[17:16] IDLEST	Idle status
I2C4	Slave	CM_L4PER_I2C4_CLKCTRL[17:16] IDLEST	Idle status
L4_PER interconnect	Slave	CM_L4PER_L4PER_CLKCTRL[17:16] IDLEST	Idle status
MCBSP4	Slave	CM_L4PER_MCBSP4_CLKCTRL[17:16] IDLEST	Idle status
MCSP1	Slave	CM_L4PER_MCSP1_CLKCTRL[17:16] IDLEST	Idle status
MCSP2	Slave	CM_L4PER_MCSP2_CLKCTRL[17:16] IDLEST	Idle status
MCSP3	Slave	CM_L4PER_MCSP3_CLKCTRL[17:16] IDLEST	Idle status
MCSP4	Slave	CM_L4PER_MCSP4_CLKCTRL[17:16] IDLEST	Idle status
HSMMC3	Slave	CM_L4PER_MMCS3_CLKCTRL[17:16] IDLEST	Idle status
HSMMC4	Slave	CM_L4PER_MMCS4_CLKCTRL[17:16] IDLEST	Idle status
HSMMC5	Slave	CM_L4PER_MMCS5_CLKCTRL[17:16] IDLEST	Idle status
SLIMBUS2	Slave	CM_L4PER_SLIMBUS2_CLKCTRL[17:16] IDLEST	Idle status
UART1	Slave	CM_L4PER_UART1_CLKCTRL[17:16] IDLEST	Idle status
UART2	Slave	CM_L4PER_UART2_CLKCTRL[17:16] IDLEST	Idle status
UART3	Slave	CM_L4PER_UART3_CLKCTRL[17:16] IDLEST	Idle status
UART4	Slave	CM_L4PER_UART4_CLKCTRL[17:16] IDLEST	Idle status

Table 3-115 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-115. CD\_L4\_PER Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
GPTIMER10	Available	N/A	Available	CM_L4PER_GPTIMER10_CLKCTRL[1:0] MODULEMODE	Read/Write
GPTIMER11	Available	N/A	Available	CM_L4PER_GPTIMER11_CLKCTRL[1:0] MODULEMODE	Read/Write
GPTIMER2	Available	N/A	Available	CM_L4PER_GPTIMER2_CLKCTRL[1:0] MODULEMODE	Read/Write
GPTIMER3	Available	N/A	Available	CM_L4PER_GPTIMER3_CLKCTRL[1:0] MODULEMODE	Read/Write
GPTIMER4	Available	N/A	Available	CM_L4PER_GPTIMER4_CLKCTRL[1:0] MODULEMODE	Read/Write
GPTIMER9	Available	N/A	Available	CM_L4PER_GPTIMER9_CLKCTRL[1:0] MODULEMODE	Read/Write

**Table 3-115. CD\_L4\_PER Modules Slave Clock-Management Modes and Control (continued)**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
ELM	N/A	Available	N/A	CM_L4PER_ELM_CLKCTRL[1:0]MODULEMODE	Read only
GPIO2	Available	Available	N/A	CM_L4PER_GPIO2_CLKCTRL[1:0]MODULEMODE	Read/Write
GPIO2	Available	Available	Available	CM_L4PER_GPIO2_CLKCTRL_RESTORE[1:0]MODULEMODE	Read/Write
GPIO3	Available	Available	N/A	CM_L4PER_GPIO3_CLKCTRL[1:0]MODULEMODE	Read/Write
GPIO3	Available	Available	Available	CM_L4PER_GPIO3_CLKCTRL_RESTORE[1:0]MODULEMODE	Read/Write
GPIO4	Available	Available	N/A	CM_L4PER_GPIO4_CLKCTRL[1:0]MODULEMODE	Read/Write
GPIO4	Available	Available	Available	CM_L4PER_GPIO4_CLKCTRL_RESTORE[1:0]MODULEMODE	Read/Write
GPIO5	Available	Available	N/A	CM_L4PER_GPIO5_CLKCTRL[1:0]MODULEMODE	Read/Write
GPIO5	Available	Available	Available	CM_L4PER_GPIO5_CLKCTRL_RESTORE[1:0]MODULEMODE	Read/Write
GPIO6	Available	Available	N/A	CM_L4PER_GPIO6_CLKCTRL[1:0]MODULEMODE	Read/Write
GPIO6	Available	Available	Available	CM_L4PER_GPIO6_CLKCTRL_RESTORE[1:0]MODULEMODE	Read/Write
HDQ	Available	N/A	Available	CM_L4PER_HDQ1W_CLKCTRL[1:0]MODULEMODE	Read/Write
I2C1	Available	N/A	Available	CM_L4PER_I2C1_CLKCTRL[1:0]MODULEMODE	Read/Write
I2C2	Available	N/A	Available	CM_L4PER_I2C2_CLKCTRL[1:0]MODULEMODE	Read/Write
I2C3	Available	N/A	Available	CM_L4PER_I2C3_CLKCTRL[1:0]MODULEMODE	Read/Write
I2C4	Available	N/A	Available	CM_L4PER_I2C4_CLKCTRL[1:0]MODULEMODE	Read/Write
L4_PER interconnect	N/A	Available	N/A	CM_L4PER_L4PER_CLKCTRL[1:0]MODULEMODE	Read only
MCBSP4	Available	N/A	Available	CM_L4PER_MCBSP4_CLKCTRL[1:0]MODULEMODE	Read/Write

**Table 3-115. CD\_L4\_PER Modules Slave Clock-Management Modes and Control (continued)**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
MCSPI1	Available	N/A	Available	CM_L4PER_MCSPI1_CLKCTRL[1:0] MODULEMODE	Read/Write
MCSPI2	Available	N/A	Available	CM_L4PER_MCSPI2_CLKCTRL[1:0] MODULEMODE	Read/Write
MCSPI3	Available	N/A	Available	CM_L4PER_MCSPI3_CLKCTRL[1:0] MODULEMODE	Read/Write
MCSPI4	Available	N/A	Available	CM_L4PER_MCSPI4_CLKCTRL[1:0] MODULEMODE	Read/Write
HSMMC3	Available	N/A	Available	CM_L4PER_MMCS D3_CLKCTRL[1:0] MODULEMODE	Read/Write
HSMMC4	Available	N/A	Available	CM_L4PER_MMCS D4_CLKCTRL[1:0] MODULEMODE	Read/Write
HSMMC5	Available	N/A	Available	CM_L4PER_MMCS D5_CLKCTRL[1:0] MODULEMODE	Read/Write
SLIMBUS2	Available	N/A	Available	CM_L4PER_SLIMBUS2_CLKCTRL[1:0] MODULEMODE	Read/Write
UART1	Available	N/A	Available	CM_L4PER_UART1_CLKCTRL[1:0] MODULEMODE	Read/Write
UART2	Available	N/A	Available	CM_L4PER_UART2_CLKCTRL[1:0] MODULEMODE	Read/Write
UART3	Available	N/A	Available	CM_L4PER_UART3_CLKCTRL[1:0] MODULEMODE	Read/Write
UART4	Available	N/A	Available	CM_L4PER_UART4_CLKCTRL[1:0] MODULEMODE	Read/Write

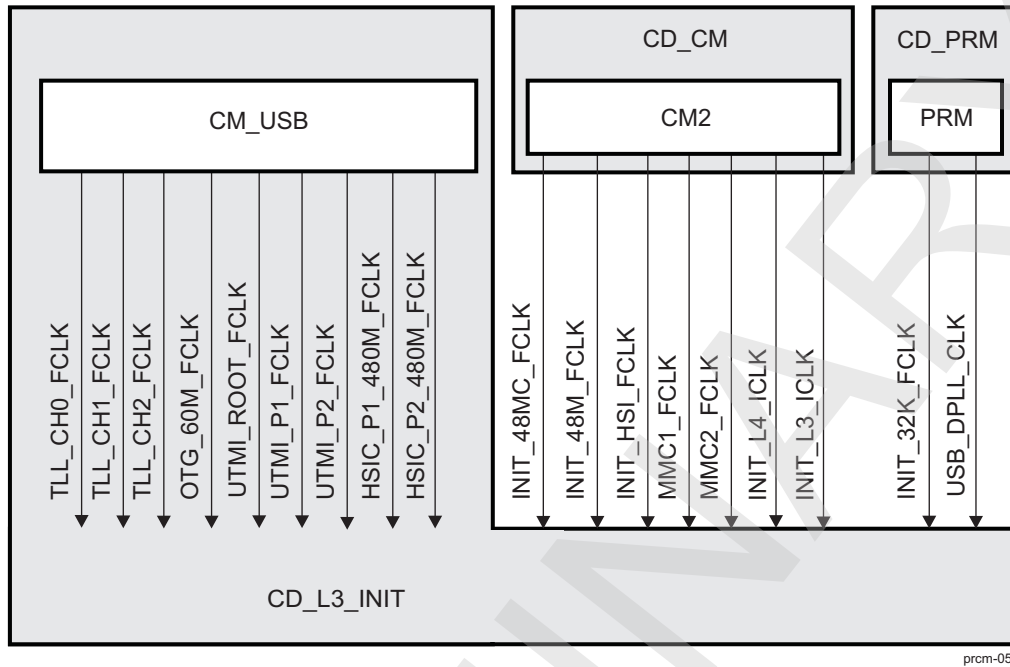
### 3.6.9 CD\_L3\_INIT Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.9.1 Overview

Figure 3-53 shows the overview of the clock domain.

Figure 3-53. CD\_L3\_INIT Overview



### 3.6.9.2 Clock Domain Modes

Table 3-116 lists the clock domain modes supported by the clock domain.

Table 3-116. CD\_L3\_INIT Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-117 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-117. CD\_L3\_INIT Control and Status Parameters

Parameter Name	Control/Status Bit Field
HSIC_P1_480M_FCLK clock status	CM_L3INIT_CLKSTCTRL[20] CLKACTIVITY_HSIC_P1_480M_FCLK
HSIC_P1_FCLK clock status	CM_L3INIT_CLKSTCTRL[26] CLKACTIVITY_HSIC_P1_FCLK
HSIC_P2_480M_FCLK clock status	CM_L3INIT_CLKSTCTRL[21] CLKACTIVITY_HSIC_P2_480M_FCLK
HSIC_P2_FCLK clock status	CM_L3INIT_CLKSTCTRL[27] CLKACTIVITY_HSIC_P2_FCLK
INIT_32K_FCLK clock status	CM_L3INIT_CLKSTCTRL[31] CLKACTIVITY_INIT_32K_FCLK
INIT_48MC_FCLK clock status	CM_L3INIT_CLKSTCTRL[12] CLKACTIVITY_INIT_48MC_FCLK
INIT_48M_FCLK clock status	CM_L3INIT_CLKSTCTRL[13] CLKACTIVITY_INIT_48M_FCLK
INIT_60M_P1_FCLK clock status	CM_L3INIT_CLKSTCTRL[28] CLKACTIVITY_INIT_60M_P1_FCLK
INIT_60M_P2_FCLK clock status	CM_L3INIT_CLKSTCTRL[29] CLKACTIVITY_INIT_60M_P2_FCLK
INIT_HSI_FCLK clock status	CM_L3INIT_CLKSTCTRL[16] CLKACTIVITY_INIT_HSI_FCLK
MMC1_FCLK clock status	CM_L3INIT_CLKSTCTRL[17] CLKACTIVITY_INIT_HSMMC1_FCLK
MMC2_FCLK clock status	CM_L3INIT_CLKSTCTRL[18] CLKACTIVITY_INIT_HSMMC2_FCLK
INIT_L3_ICLK clock status	CM_L3INIT_CLKSTCTRL[8] CLKACTIVITY_INIT_L3_ICLK

**Table 3-117. CD\_L3\_INIT Control and Status Parameters (continued)**

Parameter Name	Control/Status Bit Field
INIT_L4_ICLK clock status	CM_L3INIT_CLKSTCTRL[9] CLKACTIVITY_INIT_L4_ICLK
TLL_CH0_FCLK clock status	CM_L3INIT_CLKSTCTRL[22] CLKACTIVITY_TLL_CH0_FCLK
TLL_CH1_FCLK clock status	CM_L3INIT_CLKSTCTRL[23] CLKACTIVITY_TLL_CH1_FCLK
TLL_CH2_FCLK clock status	CM_L3INIT_CLKSTCTRL[24] CLKACTIVITY_TLL_CH2_FCLK
UTMI_ROOT_FCLK clock status	CM_L3INIT_CLKSTCTRL[25] CLKACTIVITY_UTMI_ROOT_FCLK
Clock Domain State Transition Control	CM_L3INIT_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.9.3 Clock Domain Dependency

#### 3.6.9.3.1 Static Dependency

Table 3-118 lists the static dependency of the clock domain with respect to other clock domains of the device.

**Table 3-118. CD\_L3\_INIT Static Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_ABE	Disabled	CM_L3INIT_STATICDEP[3] ABE_STATDEP	Read/Write
CD_IVAHD	Disabled	CM_L3INIT_STATICDEP[2] IVAHD_STATDEP	Read/Write
CD_L3_1	Always enabled	CM_L3INIT_STATICDEP[5] L3_1_STATDEP	Read only
CD_L3_2	Always enabled	CM_L3INIT_STATICDEP[6] L3_2_STATDEP	Read only
CD_L4_CFG	Disabled	CM_L3INIT_STATICDEP[12] L4CFG_STATDEP	Read/Write
CD_L4_PER	Disabled	CM_L3INIT_STATICDEP[13] L4PER_STATDEP	Read/Write
CD_EMIF	Disabled	CM_L3INIT_STATICDEP[4] MEMIF_STATDEP	Read/Write

#### 3.6.9.3.2 Dynamic Dependency

Table 3-119 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-119. CD\_L3\_INIT Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_1	Always disabled	CM_L3INIT_DYNAMICDEP[5] L3_1_DYNDEP	Read only
CD_L3_2	Always disabled	CM_L3INIT_DYNAMICDEP[6] L3_2_DYNDEP	Read only

#### 3.6.9.3.3 Wake-Up Dependency

Table 3-120 lists the wake-up dependency settings for the modules of this clock domain.



**Table 3-120. CD\_L3\_INIT Wake-Up Dependency Association Parameters**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
HSI	CD_L3_INIT	CD_CORTEXA9	Disabled	PM_L3INIT_HSI_WKDEP[8] WKUPDEP_WGM_HSI_WAKE_MPU	Read/Write
HSI	CD_L3_INIT	CD_DSP	Always enabled	PM_L3INIT_HSI_WKDEP[6] WKUPDEP_HSI_DSP_DSP	Read only
HSI	CD_L3_INIT	CD_CORTEXM3	Disabled	PM_L3INIT_HSI_WKDEP[1] WKUPDEP_HSI_MCU_MPU_M3	Read/Write
HSI	CD_L3_INIT	CD_CORTEXA9	Disabled	PM_L3INIT_HSI_WKDEP[0] WKUPDEP_HSI_MCU_MPU	Read/Write
HSMMC1	CD_L3_INIT	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L3INIT_MMC1_WKDEP[3] WKUPDEP_MMC1_SDMA	Read/Write
HSMMC1	CD_L3_INIT	CD_DSP	Disabled	PM_L3INIT_MMC1_WKDEP[2] WKUPDEP_MMC1_DSP	Read/Write
HSMMC1	CD_L3_INIT	CD_CORTEXM3	Disabled	PM_L3INIT_MMC1_WKDEP[1] WKUPDEP_MMC1_MPU_M3	Read/Write
HSMMC1	CD_L3_INIT	CD_CORTEXA9	Disabled	PM_L3INIT_MMC1_WKDEP[0] WKUPDEP_MMC1_MPU	Read/Write
HSMMC2	CD_L3_INIT	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_L3INIT_MMC2_WKDEP[3] WKUPDEP_MMC2_SDMA	Read/Write
HSMMC2	CD_L3_INIT	CD_DSP	Disabled	PM_L3INIT_MMC2_WKDEP[2] WKUPDEP_MMC2_DSP	Read/Write
HSMMC2	CD_L3_INIT	CD_CORTEXM3	Disabled	PM_L3INIT_MMC2_WKDEP[1] WKUPDEP_MMC2_MPU_M3	Read/Write
HSMMC2	CD_L3_INIT	CD_CORTEXA9	Disabled	PM_L3INIT_MMC2_WKDEP[0] WKUPDEP_MMC2_MPU	Read/Write
FSUSB	CD_L3_INIT	CD_CORTEXM3	Disabled	PM_L3INIT_FSUSB_WKDEP[1] WKUPDEP_FSUSB_MPU_M3	Read/Write
FSUSB	CD_L3_INIT	CD_CORTEXA9	Disabled	PM_L3INIT_FSUSB_WKDEP[0] WKUPDEP_FSUSB_MPU	Read/Write
HSUSBHOST	CD_L3_INIT	CD_CORTEXM3	Disabled	PM_L3INIT_HSUSB_HOST_WKDEP[1] WKUPDEP_HSUSB_HOST_MPU_M3	Read/Write

**Table 3-120. CD\_L3\_INIT Wake-Up Dependency Association Parameters (continued)**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
HSUSBHOST	CD_L3_INIT	CD_CORTEXA9	Disabled	PM_L3INIT_HSUSB HOST_WKDEP[0] WKUPDEP_HSUSB HOST_MPU	Read/Write
HSUSBOTG	CD_L3_INIT	CD_CORTEXM3	Disabled	PM_L3INIT_HSUSB OTG_WKDEP[1] WKUPDEP_HSUSB OTG_MPU_M3	Read/Write
HSUSBOTG	CD_L3_INIT	CD_CORTEXA9	Disabled	PM_L3INIT_HSUSB OTG_WKDEP[0] WKUPDEP_HSUSB OTG_MPU	Read/Write
HSUSBTLL	CD_L3_INIT	CD_CORTEXM3	Disabled	PM_L3INIT_HSUSB TLL_WKDEP[1] WKUPDEP_HSUSB TLL_MPU_M3	Read/Write
HSUSBTLL	CD_L3_INIT	CD_CORTEXA9	Disabled	PM_L3INIT_HSUSB TLL_WKDEP[0] WKUPDEP_HSUSB TLL_MPU	Read/Write

### 3.6.9.4 Clock Domain Module Attributes

Table 3-121 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-121. CD\_L3\_INIT Modules Clocks Association**

Module	Clock	Clock Type
HSI	INIT_HSI_FCLK	Interface and functional
	INIT_L3_ICLK	Interface
	INIT_L4_ICLK	Interface
P1500	INIT_L3_ICLK	Interface and functional
	INIT_L4_ICLK	Interface
HSMC1	MMC1_FCLK	Functional
	INIT_L3_ICLK	Interface
	INIT_L4_ICLK	Interface
HSMC2	MMC2_FCLK	Functional
	INIT_L3_ICLK	Interface
	INIT_L4_ICLK	Interface
USBPHY	INIT_32K_FCLK	Functional
	INIT_48M_FCLK	Functional
	INIT_L4_ICLK	Interface
	INIT_960M_FCLK	Functional
FSUSB	INIT_48MC_FCLK	Functional
	INIT_L3_ICLK	Interface
	INIT_L4_ICLK	Interface
HSUSBHOST	HSIC_P1_480M_FCLK	Functional
	HSIC_P1_FCLK	Functional
	HSIC_P2_480M_FCLK	Functional
	HSIC_P2_FCLK	Functional
	INIT_48MC_FCLK	Functional (mandatory)
	INIT_48MC_FCLK	Functional (optional)

**Table 3-121. CD\_L3\_INIT Modules Clocks Association (continued)**

Module	Clock	Clock Type
	INIT_L3_ICLK	Interface
	INIT_L4_ICLK	Interface
	UTMI_P1_FCLK	Functional
	UTMI_P2_FCLK	Functional
	UTMI_ROOT_FCLK	Functional
HSUSBOTG	INIT_L3_ICLK	Interface
	INIT_L4_ICLK	Interface
	OTG_60M_FCLK	Functional
HSUSBTLL	INIT_L4_ICLK	Interface
	TLL_CH0_FCLK	Functional
	TLL_CH1_FCLK	Functional
	TLL_CH2_FCLK	Functional

Table 3-122 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-122. CD\_L3\_INIT Modules Wake-Up Request**

Module	Wake-Up Feature
HSI	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, DSP -IRQ)/Master wake-up request
HSMMC1	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, DSP -IRQ, sDMA -DMA)
HSMMC2	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, DSP -IRQ, sDMA -DMA)
USBPHY	None
FSUSB	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)/Master wake-up request
HSUSBHOST	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)
HSUSBOTG	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)
HSUSBTLL	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)

Table 3-123 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-123. CD\_L3\_INIT Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
HSI	Slave/Master	CM_L3INIT_HSI_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_HSI_CLKCTRL[17:16] IDLEST	Idle status
P1500	Slave/Master	CM_L3INIT_P1500_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_HSI_CLKCTRL[17:16] IDLEST	Idle status
HSMMC1	Master/Slave	CM_L3INIT_HSMMC1_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_HSMMC1_CLKCTRL[17:16] IDLEST	Idle status
HSMMC2	Master/Slave	CM_L3INIT_HSMMC2_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_HSMMC2_CLKCTRL[17:16] IDLEST	Idle status

**Table 3-123. CD\_L3\_INIT Modules Clock-Management Modes and Control (continued)**

Module	Clock-Management Protocol	Status Bit Field	Role
USBPHY	Slave	CM_L3INIT_USBPHY_CLKCTRL[17:16] IDLEST	Idle status
FSUSB	Slave/Master	CM_L3INIT_FSUSB_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_FSUSB_CLKCTRL[17:16] IDLEST	Idle status
HSUSBHOST	Slave/Master	CM_L3INIT_HSUSBHOST_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_HSUSBHOST_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INIT_HSUSBHOST_CLKCTRL_RESTORE[18] STBYST	Standby status
		CM_L3INIT_HSUSBHOST_CLKCTRL_RESTORE[17:16] IDLEST	Idle status
HSUSBOTG	Slave/Master	CM_L3INIT_HSUSBOTG_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_HSUSBOTG_CLKCTRL[17:16] IDLEST	Idle status
HSUSBTLL	Slave	CM_L3INIT_HSUSBTLL_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INIT_HSUSBTLL_CLKCTRL_RESTORE[17:16] IDLEST	Idle status

Table 3-124 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-124. CD\_L3\_INIT Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
HSI	Available	Available	N/A	CM_L3INIT_HSI_CLKCTRL[1:0] MODULEMODE	Read/Write
P1500	N/A	Available	N/A	CM_L3INIT_P1500_CLKCTRL[1:0] MODULEMODE	Read
HSMMC1	Available	N/A	Available	CM_L3INIT_HSMMC1_CLKCTRL[1:0] MODULEMODE	Read/Write
HSMMC2	Available	N/A	Available	CM_L3INIT_HSMMC2_CLKCTRL[1:0] MODULEMODE	Read/Write
USBPHY	Available	Available	N/A	CM_L3INIT_USBPHY_CLKCTRL[1:0] MODULEMODE	Read/Write
FSUSB	Available	N/A	Available	CM_L3INIT_FSUSB_CLKCTRL[1:0] MODULEMODE	Read/Write
HSUSBHOST	Available	N/A	Available	CM_L3INIT_HSUSBHOST_CLKCTRL[1:0] MODULEMODE	Read/Write
HSUSBOTG	Available	Available	N/A	CM_L3INIT_HSUSBOTG_CLKCTRL[1:0] MODULEMODE	Read/Write
HSUSBTLL	Available	Available	N/A	CM_L3INIT_HSUSBTLL_CLKCTRL[1:0] MODULEMODE	Read/Write

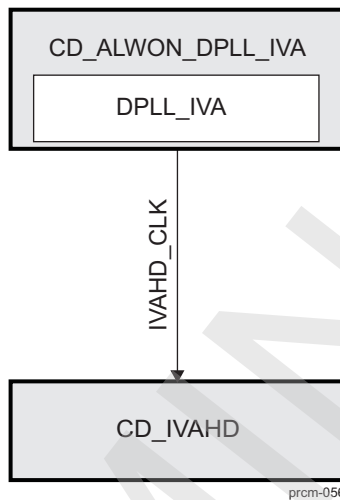
### 3.6.10 CD\_IVAHD Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.10.1 Overview

Figure 3-54 shows the overview of the clock domain.

Figure 3-54. CD\_IVAHD Overview



#### 3.6.10.2 Clock Domain Modes

Table 3-125 lists the clock domain modes supported by the clock domain.

Table 3-125. CD\_IVAHD Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-126 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-126. CD\_IVAHD Control and Status Parameters

Parameter Name	Control/Status Bit Field
IVAHD_ROOT_CLK clock status	CM_IVAHD_CLKSTCTRL[8] CLKACTIVITY_IVAHD_CLK
Clock Domain State Transition Control	CM_IVAHD_CLKSTCTRL[1:0] CLKTRCTRL

#### 3.6.10.3 Clock Domain Dependency

CD\_IVAHD has no module wake-up dependency with any other clock domain of the device

##### 3.6.10.3.1 Static Dependency

Table 3-127 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-127. CD\_IVAHD Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_1	Disabled	CM_IVAHD_STATICDEP[5] L3_1_STATDEP	Read/Write
CD_L3_2	Always enabled	CM_IVAHD_STATICDEP[6] L3_2_STATDEP	Read only

**Table 3-127. CD\_IVAHD Static Dependency Association Parameters (continued)**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_EMIF	Disabled	CM_IVAHD_STATICDEP[4] MEMIF_STATDEP	Read/Write

### 3.6.10.3.2 Dynamic Dependency

Table 3-128 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-128. CD\_IVAHD Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_2	Always disabled	CM_IVAHD_DYNAMICDEP[6] L3_2_DYNDEP	Read only

### 3.6.10.4 Clock Domain Module Attributes

Table 3-129 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-129. CD\_IVAHD Modules Clocks Association**

Module	Clock	Clock Type
IVAHD	IVAHD_ROOT_CLK	Interface and Functional
SL2	IVAHD_ROOT_CLK	Interface

Table 3-130 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-130. CD\_IVAHD Modules Wake-Up Request**

Module	Wake-Up Feature
IVAHD	None
SL2	None

Table 3-131 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-131. CD\_IVAHD Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
IVAHD	Master/Slave	CM_IVAHD_IVAHD_CLKCTRL [18] STBYST	Standby status
		CM_IVAHD_IVAHD_CLKCTRL [17:16] IDLEST	Idle status
SL2	Slave	CM_IVAHD_SL2_CLKCTRL[17 :16] IDLEST	Idle status

Table 3-132 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-132. CD\_IVAHD Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
IVAHD	Available	Available	N/A	CM_IVAHD_IVAHD_CLKCTRL[1:0] MODULEMODE	Read/Write
SL2	Available	Available	N/A	CM_IVAHD_SL2_CLKCTRL[1:0] MODULEMODE	Read/Write

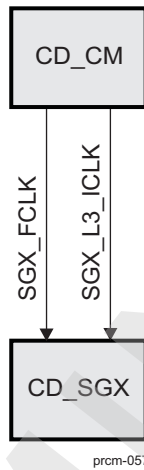
### 3.6.11 CD\_SGX Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.11.1 Overview

Figure 3-55 shows the overview of the clock domain.

Figure 3-55. CD\_SGX Overview



#### 3.6.11.2 Clock Domain Modes

Table 3-133 lists the clock domain modes supported by the clock domain.

Table 3-133. CD\_SGX Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-134 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-134. CD\_SGX Control and Status Parameters

Parameter Name	Control/Status Bit Field
SGX_L3_ICLK clock status	CM_SGX_CLKSTCTRL[8] CLKACTIVITY_SGX_L3_ICLK
SGX_FCLK clock status	CM_SGX_CLKSTCTRL[9] CLKACTIVITY_SGX_FCLK
Clock Domain State Transition Control	CM_SGX_CLKSTCTRL[1:0] CLKTRCTRL

#### 3.6.11.3 Clock Domain Dependency

CD\_SGX has no module wake-up dependency with any other clock domain of the device.

##### 3.6.11.3.1 Static Dependency

Table 3-135 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-135. CD\_SGX Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_IVAHD	Disabled	CM_SGX_STATICDEP[2] IVAHD_STATDEP	Read/Write

**Table 3-135. CD\_SGX Static Dependency Association Parameters (continued)**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_1	Disabled	CM_SGX_STATICDEP[5] L3_1_STATDEP	Read/Write
CD_L3_2	Always enabled	CM_SGX_STATICDEP[6] L3_2_STATDEP	Read only
CD_EMIF	Disabled	CM_SGX_STATICDEP[4] MEMIF_STATDEP	Read/Write

### 3.6.11.3.2 Dynamic Dependency

Table 3-136 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-136. CD\_SGX Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_2	Always disabled	CM_SGX_DYNAMICDEP[6] L3_2_DYNDEP	Read only

### 3.6.11.4 Clock Domain Module Attributes

Table 3-137 identifies for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-137. CD\_SGX Modules Clocks Association**

Module	Clock	Clock Type
SGX	SGX_L3_ICLK	Interface
	SGX_FCLK	Functional

Table 3-138 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-138. CD\_SGX Modules Wake-Up Request**

Module	Wake-Up Feature
SGX	None

Table 3-139 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-139. CD\_SGX Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
SGX	Master/Slave	CM_SGX_SGX_CLKCTRL[18] STBYST	Standby status
		CM_SGX_SGX_CLKCTRL[17: 16] IDLEST	Idle status

Table 3-140 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-140. CD\_SGX Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
SGX	Available	N/A	Available	CM_SGX_SGX_CL KCTRL[1:0] MODULEMODE	Read/Write



### 3.6.12 CD\_EMU Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.12.1 Overview

Figure 3-56 shows the overview of the clock domain.

Figure 3-56. CD\_EMU Overview

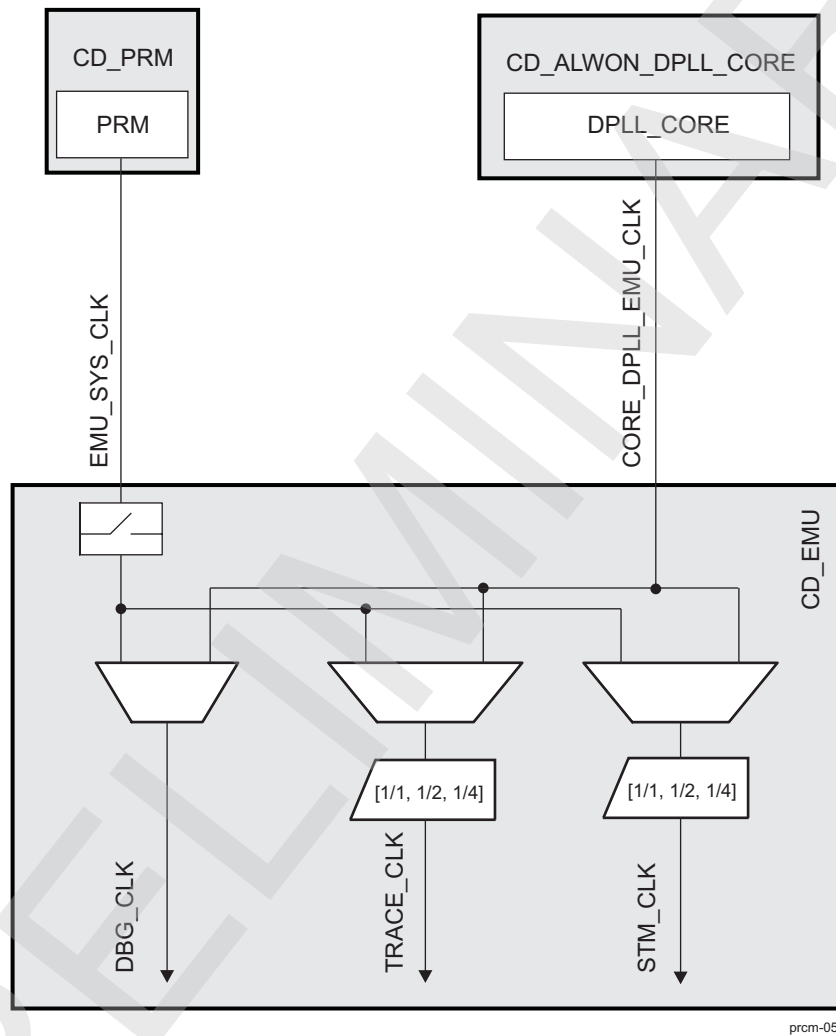


Table 3-141 lists controls for clock dividers or muxes in the clock domain.

Table 3-141. CD\_EMU Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Divider STM_CLK	CM_EMU_DEBUGSS_CLKCTRL[29:27] CLKSEL_PMD_STM_CLK
Divider TRACE_CLK	CM_EMU_DEBUGSS_CLKCTRL[26:24] CLKSEL_PMD_TRACE_CLK

#### 3.6.12.2 Clock Domain Modes

Table 3-142 lists the clock domain modes supported by the clock domain.

**Table 3-142. CD\_EMU Clock Domain Modes**

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Not available	Not available	Available	Available

Table 3-143 lists the clock domain state transition control and status bits for the clock in this clock domain.

**Table 3-143. CD\_EMU Control and Status Parameters**

Parameter Name	Control/Status Bit Field
CORE_DPLL_EMU_CLK clock status	CM_EMU_CLKSTCTRL[9] CLKACTIVITY_CORE_DPLL_EMU_CLK
EMU_SYS_CLK clock status	CM_EMU_CLKSTCTRL[8] CLKACTIVITY_EMU_SYS_CLK
Clock Domain State Transition Control	CM_EMU_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.12.3 Clock Domain Dependency

CD\_EMU has no static or module wake-up dependency with any other clock domain of the device.

#### 3.6.12.3.1 Dynamic Dependency

Table 3-144 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-144. CD\_EMU Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_2	Always enabled	CM_EMU_DYNAMICDEP[6] L3_2_DYNDEP	Read only

### 3.6.12.4 Clock Domain Module Attributes

Table 3-145 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-145. CD\_EMU Modules Clocks Association**

Module	Clock	Clock Type
DEBUG	DBG_CLK	Interface
	EMU_SYS_CLK	Interface and Functional
	STM_CLK	Interface and Functional
	TRACE_CLK	Interface and Functional

Table 3-146 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-146. CD\_EMU Modules Wake-Up Request**

Module	Wake-Up Feature
DEBUG	Master wake-up request

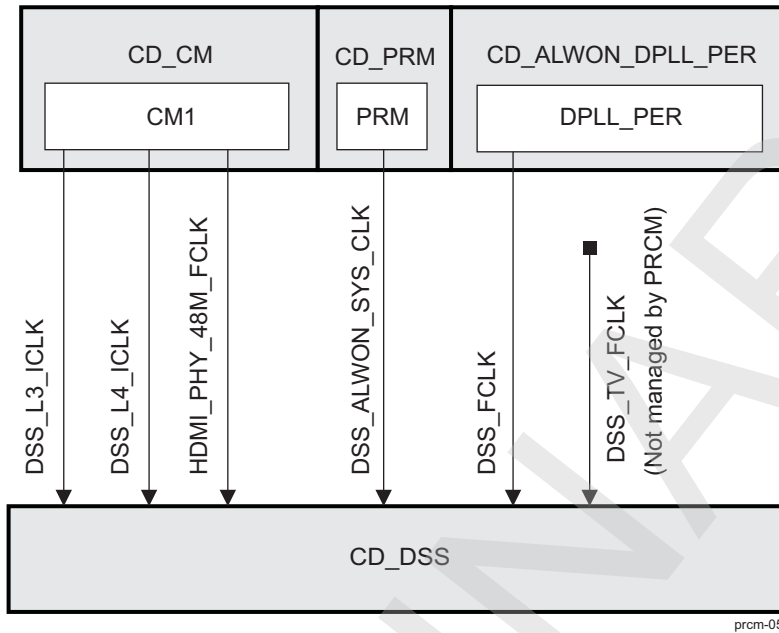
### 3.6.13 CD\_DSS Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.13.1 Overview

Figure 3-57 shows the overview of the clock domain.

Figure 3-57. CD\_DSS Overview



### 3.6.13.2 Clock Domain Modes

Table 3-147 lists the clock domain modes supported by the clock domain.

Table 3-147. CD\_DSS Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-148 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-148. CD\_DSS Control and Status Parameters

Parameter Name	Control/Status Bit Field
DSS_ALWON_SYS_CLK clock status	CM_DSS_CLKSTCTRL[10] CLKACTIVITY_DSS_ALWON_SYS_CLK
DSS_FCLK clock status	CM_DSS_CLKSTCTRL[9] CLKACTIVITY_DSS_FCLK
HDMI_PHY_48M_FCLK clock status	CM_DSS_CLKSTCTRL[11] CLKACTIVITY_HDMI_PHY_48M_FCLK
DSS_L3_ICLK clock status	CM_DSS_CLKSTCTRL[8] CLKACTIVITY_DSS_L3_ICLK
Clock Domain State Transition Control	CM_DSS_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.13.3 Clock Domain Dependency

#### 3.6.13.3.1 Static Dependency

Table 3-149 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-149. CD\_DSS Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_IVAHD	Disabled	CM_DSS_STATICDEP[2] IVAHD_STATDEP	Read/Write

**Table 3-149. CD\_DSS Static Dependency Association Parameters (continued)**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_1	Always enabled	CM_DSS_STATICDEP[5] L3_1_STATDEP	Read only
CD_L3_2	Disabled	CM_DSS_STATICDEP[6] L3_2_STATDEP	Read/Write
CD_EMIF	Disabled	CM_DSS_STATICDEP[4] MEMIF_STATDEP	Read/Write

**3.6.13.3.2 Dynamic Dependency**

Table 3-150 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-150. CD\_DSS Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_1	Always disabled	CM_DSS_DYNAMICDEP[5] L3_1_DYNDEP	Read only

**3.6.13.3.3 Wake-Up Dependency**

Table 3-151 lists the wake-up dependency settings for the modules of this clock domain.

**Table 3-151. CD\_DSS Wake-Up Dependency Association Parameters**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DSS-DSI1	CD_DSS	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_DSS_DSS_WK DEP[7] WKUPDEP_DSI1_S DMA	Read/Write
DSS-DSI1	CD_DSS	CD_DSP	Disabled	PM_DSS_DSS_WK DEP[6] WKUPDEP_DSI1_D SP	Read/Write
DSS-DSI1	CD_DSS	CD_CORTEXM3	Disabled	PM_DSS_DSS_WK DEP[5] WKUPDEP_DSI1_ MPU_M3	Read/Write
DSS-DSI1	CD_DSS	CD_CORTEXA9	Disabled	PM_DSS_DSS_WK DEP[4] WKUPDEP_DSI1_ MPU	Read/Write
DSS-DSI2	CD_DSS	CD_CORTEXM3	Disabled	PM_DSS_DSS_WK DEP[9] WKUPDEP_DSI2_ MPU_M3	Read/Write
DSS-DSI2	CD_DSS	CD_CORTEXA9	Disabled	PM_DSS_DSS_WK DEP[8] WKUPDEP_DSI2_ MPU	Read/Write
DSS-DSI2	CD_DSS	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_DSS_DSS_WK DEP[11] WKUPDEP_DSI2_S DMA	Read/Write
DSS-DSI2	CD_DSS	CD_DSP	Disabled	PM_DSS_DSS_WK DEP[10] WKUPDEP_DSI2_D SP	Read/Write

**Table 3-151. CD\_DSS Wake-Up Dependency Association Parameters (continued)**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DSS-HDMI	CD_DSS	CD_DMA, CD_L3_1, CD_L3_2	Enabled	PM_DSS_DSS_WK DEP[19] WKUPDEP_HDMID MA_SDMA	Read/Write
DSS-HDMI	CD_DSS	CD_DSP	Disabled	PM_DSS_DSS_WK DEP[14] WKUPDEP_HDMII RQ_DSP	Read/Write
DSS-HDMI	CD_DSS	CD_CORTEXM3	Disabled	PM_DSS_DSS_WK DEP[13] WKUPDEP_HDMII RQ_MPU_M3	Read/Write
DSS-HDMI	CD_DSS	CD_CORTEXA9	Disabled	PM_DSS_DSS_WK DEP[12] WKUPDEP_HDMII RQ_MPU	Read/Write
DSS-DISPC	CD_DSS	CD_CORTEXM3	Disabled	PM_DSS_DSS_WK DEP[1] WKUPDEP_DISPC _MPU_M3	Read/Write
DSS-DISPC	CD_DSS	CD_CORTEXA9	Disabled	PM_DSS_DSS_WK DEP[0] WKUPDEP_DISPC _MPU	Read/Write
DSS-DISPC	CD_DSS	CD_DSP	Disabled	PM_DSS_DSS_WK DEP[2] WKUPDEP_DISPC _DSP	Read/Write
DSS-DISPC	CD_DSS	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_DSS_DSS_WK DEP[3] WKUPDEP_DISPC _SDMA	Read/Write

**3.6.13.4 Clock Domain Module Attributes**

Table 3-152 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-152. CD\_DSS Modules Clocks Association**

Module	Clock	Clock Type
DSS	DSS_ALWON_SYS_CLK	Functional
	DSS_FCLK	Functional
	DSS_TV_FCLK	Functional
	HDMI_PHY_48M_FCLK	Functional
	DSS_L3_ICLK	Interface
	DSS_L4_ICLK	Interface

**NOTE:** DSS\_TV\_FCLK is sourced from `fref_slicer_in` input pin. To enable DSS\_TV\_FCLK it must be ungated in `ALTCLKSRC[3] ENABLE_EXT` and must be enabled in `CM_DSS_DSS_CLKCTRL[11] OPTFCLKEN_TV_FCLK`.

Table 3-153 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-153. CD\_DSS Modules Wake-Up Request**

Module	Wake-Up Feature
DSS	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ, DSP -IRQ, sDMA -DMA)

Table 3-154 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-154. CD\_DSS Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
DSS	Master/Slave	CM_DSS_DSS_CLKCTRL[18] STBYST	Standby status
		CM_DSS_DSS_CLKCTRL[17:16] IDLEST	Idle status

Table 3-155 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-155. CD\_DSS Modules Slave Clock-Management Modes and Control**

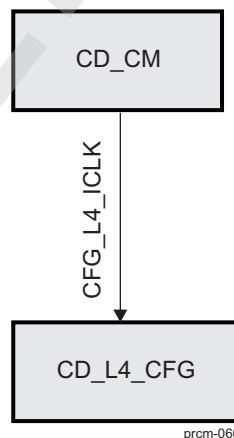
Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DSS	Available	N/A	Available	CM_DSS_DSS_CLKCTRL[1:0] MODULEMODE	Read/Write

### 3.6.14 CD\_L4\_CFG Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.14.1 Overview

Figure 3-58 shows the overview of the clock domain.

**Figure 3-58. CD\_L4\_CFG Overview**

#### 3.6.14.2 Clock Domain Modes

Table 3-156 lists the clock domain modes supported by the clock domain.

**Table 3-156. CD\_L4\_CFG Clock Domain Modes**

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Not available	Available

Table 3-157 lists the clock domain state transition control and status bits for the clock in this clock domain.

**Table 3-157. CD\_L4\_CFG Control and Status Parameters**

Parameter Name	Control/Status Bit Field
CFG_L4_ICLK clock status	CM_L4CFG_CLKSTCTRL[8] CLKACTIVITY_CFG_L4_ICLK
Clock Domain State Transition Control	CM_L4CFG_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.14.3 Clock Domain Dependency

CD\_L4\_CFG has no static or module wake-up dependency with any other clock domain of the device.

#### 3.6.14.3.1 Dynamic Dependency

Table 3-158 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-158. CD\_L4\_CFG Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_C2C	Always enabled	CM_L4CFG_DYNAMICDEP[18] C2C_DYNDEP	Read only
CD_DSS	Always enabled	CM_L4CFG_DYNAMICDEP[8] DSS_DYNDEP	Read only
CD_CAM	Always disabled	CM_L4CFG_DYNAMICDEP[9] ISS_DYNDEP	Read only
CD_L3_1	Always enabled	CM_L4CFG_DYNAMICDEP[5] L3_1_DYNDEP	Read only
CD_L3_2	Always enabled	CM_L4CFG_DYNAMICDEP[6] L3_2_DYNDEP	Read only
CD_L3_INIT	Always enabled	CM_L4CFG_DYNAMICDEP[7] L3INIT_DYNDEP	Read only
CD_EMIF	Always enabled	CM_L4CFG_DYNAMICDEP[4] MEMIF_DYNDEP	Read only
CD_DMA	Always enabled	CM_L4CFG_DYNAMICDEP[11] SDMA_DYNDEP	Read only
CD_DSP	Always enabled	CM_L4CFG_DYNAMICDEP[1] DSP_DYNDEP	Read only

### 3.6.14.4 Clock Domain Module Attributes

Table 3-159 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-159. CD\_L4\_CFG Modules Clocks Association**

Module	Clock	Clock Type
SYSCTRL_PADCONF_CORE	CFG_L4_ICLK	Functional
SYSCTRL_GENERAL_CORE	CFG_L4_ICLK	Functional
SPINLOCK	CFG_L4_ICLK	Interface
L4_CFG interconnect	CFG_L4_ICLK	Interface
MAILBOX	CFG_L4_ICLK	Interface
SAR_ROM	CFG_L4_ICLK	Interface

Table 3-160 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-160. CD\_L4\_CFG Modules Wake-Up Request**

Module	Wake-Up Feature
SYSCTRL_PADCONF_CORE	None
SYSCTRL_GENERAL_CORE	None
SPINLOCK	None
L4_CFG interconnect	None
MAILBOX	None
SAR_ROM	None

Table 3-161 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-161. CD\_L4\_CFG Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
SPINLOCK	Slave	CM_L4CFG_SPINLOCK_CLKCTRL[17:16] IDLEST	Idle status
L4_CFG interconnect	Slave	CM_L4CFG_L4_CFG_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX	Slave	CM_L4CFG_MAILBOX_CLKCTRL[17:16] IDLEST	Idle status
SAR_ROM	Slave	CM_L4CFG_SAR_ROM_CLKCTRL[17:16] IDLEST	Idle status

Table 3-162 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-162. CD\_L4\_CFG Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
SPINLOCK	N/A	Available	N/A	CM_L4CFG_SPINLOCK_CLKCTRL[1:0] MODULEMODE	Read only
L4_CFG interconnect	N/A	Available	N/A	CM_L4CFG_L4_CFG_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX	N/A	Available	N/A	CM_L4CFG_MAILBOX_CLKCTRL[1:0] MODULEMODE	Read only
SAR_ROM	N/A	Available	N/A	CM_L4CFG_SAR_ROM_CLKCTRL[1:0] MODULEMODE	Read only

### 3.6.15 CD\_L3\_INSTR Clock Domain

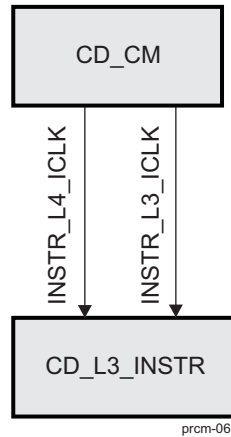
This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.15.1 Overview

Figure 3-59 shows the overview of the clock domain.



Figure 3-59. CD\_L3\_INSTR Overview



### 3.6.15.2 Clock Domain Modes

Table 3-163 lists the clock domain modes supported by the clock domain.

Table 3-163. CD\_L3\_INSTR Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Not available	Not available	Not available	Available

Table 3-164 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-164. CD\_L3\_INSTR Control and Status Parameters

Parameter Name	Control/Status Bit Field
INSTR_L3_ICLK clock status	CM_L3INSTR_CLKSTCTRL[8] CLKACTIVITY_L3_INSTR_ICLK
Clock Domain State Transition Control	CM_L3INSTR_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.15.3 Clock Domain Dependency

CD\_L3\_INSTR has no static, dynamic, or module wake-up dependency with any other clock domain of the device

### 3.6.15.4 Clock Domain Module Attributes

Table 3-165 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

Table 3-165. CD\_L3\_INSTR Modules Clocks Association

Module	Clock	Clock Type
L3_3 interconnect	INSTR_L3_ICLK	Interface
	INSTR_L4_ICLK	Interface
L3_INSTR interconnect	INSTR_L3_ICLK	Interface
INTERCONN_WP1	INSTR_L3_ICLK	Interface
	INSTR_L4_ICLK	Interface

Table 3-166 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-166. CD\_L3\_INSTR Modules Wake-Up Request**

Module	Wake-Up Feature
L3_3 interconnect	None
L3_INSTR interconnect	None

Table 3-167 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-167. CD\_L3\_INSTR Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
L3_3 interconnect	Slave	CM_L3INSTR_L3_3_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INSTR_L3_3_CLKCTRL_RESTORE[17:16] IDLEST	Idle status
L3_INSTR interconnect	Slave	CM_L3INSTR_L3_INSTR_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE[17:16] IDLEST	Idle status

Table 3-168 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-168. CD\_L3\_INSTR Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L3_3 interconnect	Available	Available	N/A	CM_L3INSTR_L3_3_CLKCTRL[1:0] MODULEMODE	Read/Write
L3_3 interconnect	Available	Available	N/A	CM_L3INSTR_L3_3_CLKCTRL_RESTORE[1:0] MODULEMODE	Read/Write
L3_INSTR interconnect	Available	Available	N/A	CM_L3INSTR_L3_INSTR_CLKCTRL[1:0] MODULEMODE	Read/Write
L3_INSTR interconnect	Available	Available	N/A	CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE[1:0] MODULEMODE	Read/Write

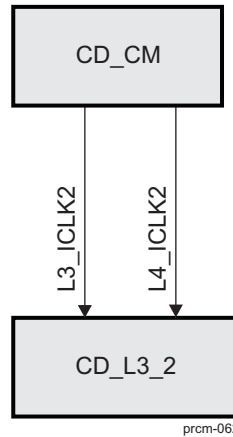
### 3.6.16 CD\_L3\_2 Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.16.1 Overview

Figure 3-60 shows the overview of the clock domain.

Figure 3-60. CD\_L3\_2 Overview



### 3.6.16.2 Clock Domain Modes

Table 3-169 lists the clock domain modes supported by the clock domain.

Table 3-169. CD\_L3\_2 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Not available	Available

Table 3-170 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-170. CD\_L3\_2 Control and Status Parameters

Parameter Name	Control/Status Bit Field
L3_ICLK2 clock status	CM_L3_2_CLKSTCTRL[8] CLKACTIVITY_L3_2_ICLK
Clock Domain State Transition Control	CM_L3_2_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.16.3 Clock Domain Dependency

CD\_L3\_2 has no static or module wake-up dependency with any other clock domain of the device.

#### 3.6.16.3.1 Dynamic Dependency

Table 3-171 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-171. CD\_L3\_2 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_C2C	Always enabled	CM_L3_2_DYNAMICDEP[18] C2C_DYNDEP	Read only
CD_DSS	Always enabled	CM_L3_2_DYNAMICDEP[8] DSS_DYNDEP	Read only
CD_CORTEXM3	Always enabled	CM_L3_2_DYNAMICDEP[0] MPU_M3_DYNDEP	Read only
CD_SGX	Always enabled	CM_L3_2_DYNAMICDEP[10] SGX_DYNDEP	Read only
CD_CAM	Always disabled	CM_L3_2_DYNAMICDEP[9] ISS_DYNDEP	Read only
CD_IVAHD	Always enabled	CM_L3_2_DYNAMICDEP[2] IVAHD_DYNDEP	Read only

**Table 3-171. CD\_L3\_2 Dynamic Dependency Association Parameters (continued)**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_1	Always enabled	<a href="#">CM_L3_2_DYNAMICDEP</a> [5] L3_1_DYNDEP	Read only
CD_L3_INIT	Always enabled	<a href="#">CM_L3_2_DYNAMICDEP</a> [7] L3INIT_DYNDEP	Read only
CD_L4_PER	Always enabled	<a href="#">CM_L3_2_DYNAMICDEP</a> [13] L4PER_DYNDEP	Read only

### 3.6.16.4 Clock Domain Module Attributes

[Table 3-172](#) lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-172. CD\_L3\_2 Modules Clocks Association**

Module	Clock	Clock Type
GPMC	L3_ICLK2	Interface
L3_2 interconnect	L3_ICLK2	Interface
	L4_ICLK2	Interface
OCMC_RAM	L3_ICLK2	Interface

[Table 3-173](#) lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-173. CD\_L3\_2 Modules Wake-Up Request**

Module	Wake-Up Feature
GPMC	None
L3_2 interconnect	None
OCMC_RAM	None

[Table 3-174](#) lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-174. CD\_L3\_2 Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
GPMC	Slave	<a href="#">CM_L3_2_GPMC_CLKCTRL</a> [17:16] IDLEST	Idle status
L3_2 interconnect	Slave	<a href="#">CM_L3_2_L3_2_CLKCTRL</a> [17:16] IDLEST	Idle status
OCMC_RAM	Slave	<a href="#">CM_L3_2_OCMC_RAM_CLKCTRL</a> [17:16] IDLEST	Idle status

[Table 3-175](#) lists the supported clock management modes and associated software control bit fields for each module of the power domain.

**Table 3-175. CD\_L3\_2 Modules Slave Clock Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
GPMC	Available	Available	N/A	<a href="#">CM_L3_2_GPMC_CLKCTRL</a> [1:0] MODULEMODE	Read/Write
L3_2 interconnect	N/A	Available	N/A	<a href="#">CM_L3_2_L3_2_CLKCTRL</a> [1:0] MODULEMODE	Read only
OCMC_RAM	N/A	Available	N/A	<a href="#">CM_L3_2_OCMC_RAM_CLKCTRL</a> [1:0] MODULEMODE	Read only

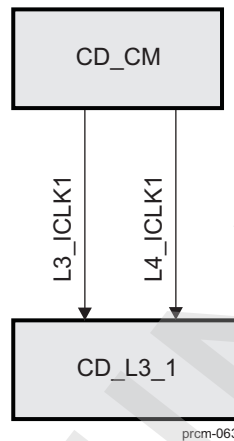
### 3.6.17 CD\_L3\_1 Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.17.1 Overview

Figure 3-61 shows the overview of the clock domain.

Figure 3-61. CD\_L3\_1 Overview



#### 3.6.17.2 Clock Domain Modes

Table 3-176 lists the clock domain modes supported by the clock domain.

Table 3-176. CD\_L3\_1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Not available	Available

Table 3-177 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-177. CD\_L3\_1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
L3_ICLK1 clock status	CM_L3_1_CLKSTCTRL[8] CLKACTIVITY_L3_1_ICLK
Clock Domain State Transition Control	CM_L3_1_CLKSTCTRL[1:0] CLKTRCTRL

#### 3.6.17.3 Clock Domain Dependency

CD\_L3\_1 has no static or module wake-up dependency with any other clock domain of the device.

##### 3.6.17.3.1 Dynamic Dependency

Table 3-178 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-178. CD\_L3\_1 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_ABE	Always enabled	CM_L3_1_DYNAMICDEP[3] ABE_DYNDEP	Read only
CD_L3_2	Always enabled	CM_L3_1_DYNAMICDEP[6] L3_2_DYNDEP	Read only
CD_L4_CFG	Always enabled	CM_L3_1_DYNAMICDEP[12] L4CFG_DYNDEP	Read only

**Table 3-178. CD\_L3\_1 Dynamic Dependency Association Parameters (continued)**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_EMIF	Always enabled	CM_L3_1_DYNAMICDEP[4] MEMIF_DYNDEP	Read only

### 3.6.17.4 Clock Domain Module Attributes

Table 3-179 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-179. CD\_L3\_1 Modules Clocks Association**

Module	Clock	Clock Type
L3_1 interconnect	L3_ICLK1	Interface
	L4_ICLK1	Interface

Table 3-180 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-180. CD\_L3\_1 Modules Wake-Up Request**

Module	Wake-Up Feature
L3_1 interconnect	None

Table 3-181 lists the supported clock management modes and associated software control bit fields for each module of the power domain.

**Table 3-181. CD\_L3\_1 Modules Clock Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
L3_1 interconnect	Slave	CM_L3_1_L3_1_CLKCTRL[17: 16] IDLEST	Idle status

Table 3-182 lists the supported clock management modes and associated software control bit fields for each module of the power domain.

**Table 3-182. CD\_L3\_1 Modules Slave Clock Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L3_1 interconnect	N/A	Available	N/A	CM_L3_1_L3_1_CLKCTRL[1:0] MODULEMODE	Read only

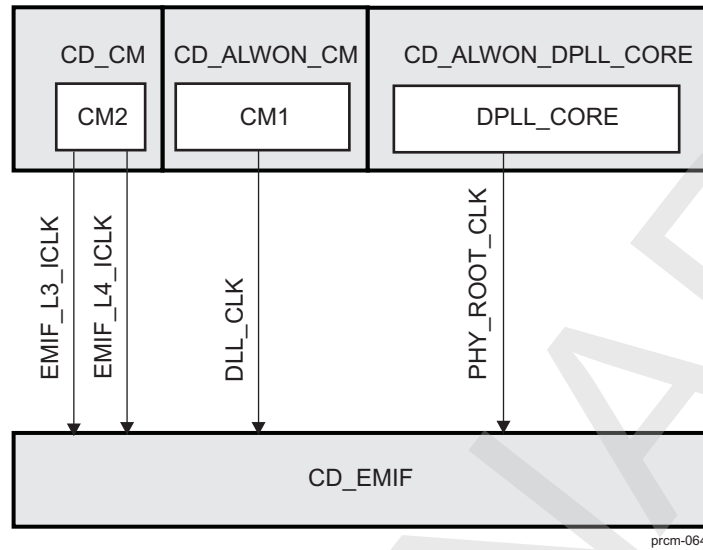
## 3.6.18 CD\_EMIF Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

### 3.6.18.1 Overview

Figure 3-62 shows the overview of the clock domain.

Figure 3-62. CD\_EMIF Overview



3.6.18.2 Clock Domain Modes

Table 3-183 lists the clock domain modes supported by the clock domain.

Table 3-183. CD\_EMIF Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-184 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-184. CD\_EMIF Control and Status Parameters

Parameter Name	Control/Status Bit Field
DLL_CLK clock status	CM_MEMIF_CLKSTCTRL[9] CLKACTIVITY_DLL_CLK
EMIF_L3_ICLK clock status	CM_MEMIF_CLKSTCTRL[8] CLKACTIVITY_L3_EMIF_ICLK
PHY_ROOT_CLK clock status	CM_MEMIF_CLKSTCTRL[10] CLKACTIVITY_PHY_ROOT_CLK
Clock Domain State Transition Control	CM_MEMIF_CLKSTCTRL[1:0] CLKTRCTRL

3.6.18.3 Clock Domain Dependency

CD\_EMIF has no static, dynamic, or module wake-up dependency with any other clock domain of the device.

3.6.18.4 Clock Domain Module Attributes

Table 3-185 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

Table 3-185. CD\_EMIF Modules Clocks Association

Module	Clock	Clock Type
DLL	DLL_CLK	Functional
DMM	EMIF_L3_ICLK	Interface
EMIF1	DLL_CLK	Interface
	EMIF_L3_ICLK	Interface

**Table 3-185. CD\_EMIF Modules Clocks Association (continued)**

Module	Clock	Clock Type
EMIF2	PHY_ROOT_CLK <sup>(1)</sup>	Interface
	PHY_ROOT_CLK <sup>(1)</sup>	Interface
	DLL_CLK	Interface
	EMIF_L3_ICLK	Interface
EMIF_FW	PHY_ROOT_CLK <sup>(1)</sup>	Interface
	PHY_ROOT_CLK <sup>(1)</sup>	Interface
	EMIF_L3_ICLK	Interface
	EMIF_L4_ICLK	Interface

<sup>(1)</sup> One PHY\_ROOT\_CLK goes directly into the module. The second PHY\_ROOT\_CLK passes through DDRPHY before going into the module.

Table 3-186 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-186. CD\_EMIF Modules Wake-Up Request**

Module	Wake-Up Feature
DDRPHY	None
DLL	None
DMM	None
EMIF1	None
EMIF2	None
EMIF_FW	None

Table 3-187 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-187. CD\_EMIF Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
DMM	Slave	CM_MEMIF_DMM_CLKCTRL[17:16] IDLEST	Idle status
EMIF1	Slave	CM_MEMIF_EMIF_1_CLKCTRL[17:16] IDLEST	Idle status
EMIF2	Slave	CM_MEMIF_EMIF_2_CLKCTRL[17:16] IDLEST	Idle status
EMIF_FW	Slave	CM_MEMIF_EMIF_FW_CLKCTRL[17:16] IDLEST	Idle status

Table 3-188 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-188. CD\_EMIF Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DMM	N/A	Available	N/A	CM_MEMIF_DMM_CLKCTRL[1:0] MODULEMODE	Read only
EMIF1	Available	Available	N/A	CM_MEMIF_EMIF_1_CLKCTRL[1:0] MODULEMODE	Read/Write
EMIF2	Available	Available	N/A	CM_MEMIF_EMIF_2_CLKCTRL[1:0] MODULEMODE	Read/Write
EMIF_FW	N/A	Available	N/A	CM_MEMIF_EMIF_FW_CLKCTRL[1:0] MODULEMODE	Read only



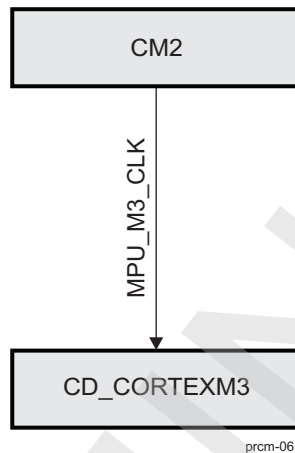
### 3.6.19 CD\_CORTEXM3 Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.19.1 Overview

Figure 3-63 shows the overview of the clock domain.

Figure 3-63. CD\_CORTEXM3 Overview



#### 3.6.19.2 Clock Domain Modes

Table 3-189 lists the clock domain modes supported by the clock domain.

Table 3-189. CD\_CORTEXM3 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-190 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-190. CD\_CORTEXM3 Control and Status Parameters

Parameter Name	Control/Status Bit Field
MPU_M3_CLK clock status	CM_MPU_M3_CLKSTCTRL[8] CLKACTIVITY_MPU_M3_CLK
Clock Domain State Transition Control	CM_MPU_M3_CLKSTCTRL[1:0] CLKTRCTRL

#### 3.6.19.3 Clock Domain Dependency

CD\_CORTEXM3 has no module wake-up dependency with any other clock domain of the device.

##### 3.6.19.3.1 Static Dependency

Table 3-191 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-191. CD\_CORTEXM3 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_ABE	Disabled	CM_MPU_M3_STATICDEP[3] ABE_STATDEP	Read/Write
CD_DSS	Disabled	CM_MPU_M3_STATICDEP[8] DSS_STATDEP	Read/Write

**Table 3-191. CD\_CORTEXM3 Static Dependency Association Parameters (continued)**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_SGX	Disabled	CM_MPU_M3_STATICDEP[10] ] SGX_STATDEP	Read/Write
CD_CAM	Always disabled	CM_MPU_M3_STATICDEP[9] ISS_STATDEP	Read only
CD_IVAHD	Disabled	CM_MPU_M3_STATICDEP[2] IVAHD_STATDEP	Read/Write
CD_L3_1	Enabled	CM_MPU_M3_STATICDEP[5] L3_1_STATDEP	Read/Write
CD_L3_2	Enabled	CM_MPU_M3_STATICDEP[6] L3_2_STATDEP	Read/Write
CD_L3_INIT	Disabled	CM_MPU_M3_STATICDEP[7] L3INIT_STATDEP	Read/Write
CD_L4_CFG	Enabled	CM_MPU_M3_STATICDEP[12] ] L4CFG_STATDEP	Read/Write
CD_L4_PER	Disabled	CM_MPU_M3_STATICDEP[13] ] L4PER_STATDEP	Read/Write
CD_EMIF	Enabled	CM_MPU_M3_STATICDEP[4] MEMIF_STATDEP	Read/Write
CD_DMA	Always disabled	CM_MPU_M3_STATICDEP[11] ] SDMA_STATDEP	Read only
CD_DSP	Disabled	CM_MPU_M3_STATICDEP[1] DSP_STATDEP	Read/Write

### 3.6.19.3.2 Dynamic Dependency

Table 3-192 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-192. CD\_CORTEXM3 Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_CAM	Always disabled	CM_MPU_M3_DYNAMICDEP[9] ISS_DYNDEP	Read only
CD_L3_2	Always enabled	CM_MPU_M3_DYNAMICDEP[6] L3_2_DYNDEP	Read only

### 3.6.19.4 Clock Domain Module Attributes

Table 3-193 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-193. CD\_CORTEXM3 Modules Clocks Association**

Module	Clock	Clock Type
CORTEXM3	MPU_M3_CLK	Interface and functional

Table 3-194 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-194. CD\_CORTEXM3 Modules Wake-Up Request**

Module	Wake-Up Feature
CORTEXM3	Master wake-up request

Table 3-195 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-195. CD\_CORTEXM3 Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
CORTEXM3	Master/Slave	CM_MPU_M3_MPU_M3_CLK_CTRL[18] STBYST	Standby status
		CM_MPU_M3_MPU_M3_CLK_CTRL[17:16] IDLEST	Idle status

Table 3-196 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-196. CD\_CORTEXM3 Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
CORTEXM3	Available	Available	N/A	CM_MPU_M3_MPU_M3_CLKCTRL[1:0] MODULEMODE	Read/Write

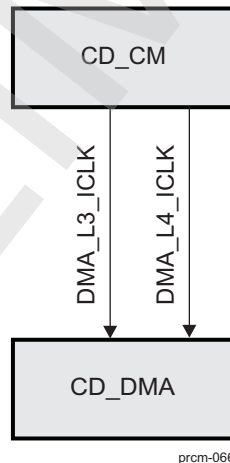
### 3.6.20 CD\_DMA Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.20.1 Overview

Figure 3-64 shows the overview of the clock domain.

**Figure 3-64. CD\_DMA Overview**



#### 3.6.20.2 Clock Domain Modes

Table 3-197 lists the clock domain modes supported by the clock domain.

**Table 3-197. CD\_DMA Clock Domain Modes**

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-198 lists the clock domain state transition control and status bits for the clock in this clock domain.

**Table 3-198. CD\_DMA Control and Status Parameters**

Parameter Name	Control/Status Bit Field
DMA_L3_ICLK clock status	CM_SDMA_CLKSTCTRL[8] CLKACTIVITY_DMA_L3_ICLK
Clock Domain State Transition Control	CM_SDMA_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.20.3 Clock Domain Dependency

CD\_DMA has a permanently enabled static dependency to the CD\_L3\_2 clock domain. It has no dynamic or module wake-up dependency with any other clock domain of the device.

### 3.6.20.4 Clock Domain Module Attributes

Table 3-199 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-199. CD\_DMA Modules Clocks Association**

Module	Clock	Clock Type
sDMA	DMA_L3_ICLK	Interface and Functional
	DMA_L4_ICLK	Interface

Table 3-200 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-200. CD\_DMA Modules Wake-Up Request**

Module	Wake-Up Feature
sDMA	Master wake-up request

Table 3-201 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-201. CD\_DMA Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
sDMA	Master/Slave	CM_SDMA_SDMA_CLKCTRL[18] STBYST	Standby status
		CM_SDMA_SDMA_CLKCTRL[17:16] IDLEST	Idle status

Table 3-202 lists the supported clock management modes and associated software control bit fields for each module of the power domain.

**Table 3-202. CD\_DMA Modules Slave Clock Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
sDMA	N/A	Available	N/A	CM_SDMA_SDMA_CLKCTRL[1:0] MODULEMODE	Read only

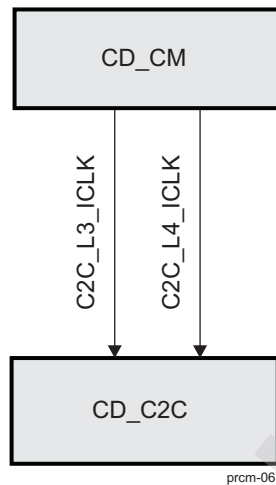
## 3.6.21 CD\_C2C Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

### 3.6.21.1 Overview

Figure 3-65 shows the overview of the clock domain.

Figure 3-65. CD\_C2C Overview



### 3.6.21.2 Clock Domain Modes

Table 3-203 lists the clock domain modes supported by the clock domain.

Table 3-203. CD\_C2C Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-204 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-204. CD\_C2C Control and Status Parameters

Parameter Name	Control/Status Bit Field
C2C_L3_ICLK clock status	CM_C2C_CLKSTCTRL[8] CLKACTIVITY_L3_C2C_ICLK
C2C_L3X2_ICLK clock status	CM_C2C_CLKSTCTRL[10] CLKACTIVITY_L3X2_C2C_ICLK
C2C_L4_ICLK clock status	CM_C2C_CLKSTCTRL[9] CLKACTIVITY_L4_C2C_ICLK
Clock Domain State Transition Control	CM_C2C_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.21.3 Clock Domain Dependency

CD\_C2C has no module wake-up dependency with any other clock domain of the device.

#### 3.6.21.3.1 Static Dependency

Table 3-205 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-205. CD\_C2C Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_ABE	Disabled	CM_C2C_STATICDEP[3] ABE_STATDEP	Read/Write
CD_IVAHD	Disabled	CM_C2C_STATICDEP[2] IVAHD_STATDEP	Read/Write
CD_L3_1	Disabled	CM_C2C_STATICDEP[5] L3_1_STATDEP	Read/Write
CD_L3_2	Disabled	CM_C2C_STATICDEP[6] L3_2_STATDEP	Read/Write

**Table 3-205. CD\_C2C Static Dependency Association Parameters (continued)**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_INIT	Disabled	CM_C2C_STATICDEP[7] L3INIT_STATDEP	Read/Write
CD_L4_CFG	Disabled	CM_C2C_STATICDEP[12] L4CFG_STATDEP	Read/Write
CD_L4_PER	Disabled	CM_C2C_STATICDEP[13] L4PER_STATDEP	Read/Write
CD_EMIF	Enabled	CM_C2C_STATICDEP[4] MEMIF_STATDEP	Read/Write

### 3.6.21.3.2 Dynamic Dependency

Table 3-206 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-206. CD\_C2C Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_2	Always enabled	CM_C2C_DYNAMICDEP[6] L3_2_DYNDEP	Read only
CD_EMIF	Always enabled	CM_C2C_DYNAMICDEP[4] MEMIF_DYNDEP	Read only

### 3.6.21.4 Clock Domain Module Attributes

Table 3-207 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-207. CD\_C2C Modules Clocks Association**

Module	Clock	Clock Type
C2C	C2C_L3X2_ICLK	Interface
	C2C_L3_ICLK	Interface
C2C_FW	C2C_L3_ICLK	Interface
	C2C_L4_ICLK	Interface

Table 3-208 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-208. CD\_C2C Modules Wake-Up Request**

Module	Wake-Up Feature
C2C	Master wake-up request
C2C_FW	None

Table 3-209 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-209. CD\_C2C Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
C2C	Master/Slave	CM_C2C_C2C_CLKCTRL[18] STBYST	Standby status
		CM_C2C_C2C_CLKCTRL[17:16] IDLEST	Idle status
C2C_FW	Slave	CM_C2C_C2C_FW_CLKCTRL [17:16] IDLEST	Idle status

Table 3-210 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-210. CD\_C2C Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
C2C	N/A	Available	N/A	CM_C2C_C2C_CLKCTRL[1:0]MODULEMODE	Read only
C2C_FW	N/A	Available	N/A	CM_C2C_C2C_FW_CLKCTRL[1:0]MODULEMODE	Read only

### 3.6.22 CD\_CAM Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.22.1 Overview

Figure 3-66 shows the overview of the clock domain.

**Figure 3-66. CD\_CAM Overview**



#### 3.6.22.2 Clock Domain Modes

Table 3-211 lists the clock domain modes supported by the clock domain.

**Table 3-211. CD\_CAM Clock Domain Modes**

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-212 lists the clock domain state transition control and status bits for the clock in this clock domain.

**Table 3-212. CD\_CAM Control and Status Parameters**

Parameter Name	Control/Status Bit Field
CAM_PHY_CTRL_CLK clock status	CM_CAM_CLKSTCTRL[9] CLKACTIVITY_CAM_PHY_CTRL_CLK
FDIF_FCLK clock status	CM_CAM_CLKSTCTRL[10] CLKACTIVITY_FDIF_FCLK
ISS_CLK clock status	CM_CAM_CLKSTCTRL[8] CLKACTIVITY_ISS_CLK

**Table 3-212. CD\_CAM Control and Status Parameters (continued)**

Parameter Name	Control/Status Bit Field
Clock Domain State Transition Control	CM_CAM_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.22.3 Clock Domain Dependency

CD\_CAM has no module wake-up dependency with any other clock domain of the device.

#### 3.6.22.3.1 Static Dependency

Table 3-213 lists the static dependency of the clock domain with respect to other clock domains of the device.

**Table 3-213. CD\_CAM Static Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_IVAHD	Disabled	CM_CAM_STATICDEP[2] IVAHD_STATDEP	Read/Write
CD_L3_1	Disabled	CM_CAM_STATICDEP[5] L3_1_STATDEP	Read/Write
CD_L3_2	Always enabled	CM_CAM_STATICDEP[6] L3_2_STATDEP	Read only
CD_EMIF	Disabled	CM_CAM_STATICDEP[4] MEMIF_STATDEP	Read/Write

#### 3.6.22.3.2 Dynamic Dependency

Table 3-214 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

**Table 3-214. CD\_CAM Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_2	Always disabled	CM_CAM_DYNAMICDEP[6] L3_2_DYNDEP	Read only

### 3.6.22.4 Clock Domain Module Attributes

Table 3-215 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-215. CD\_CAM Modules Clocks Association**

Module	Clock	Clock Type
FACE DETECT	FDIF_FCLK	Interface and Functional
ISS	CAM_PHY_CTRL_CLK	Functional
	ISS_CLK	Interface and Functional

Table 3-216 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-216. CD\_CAM Modules Wake-Up Request**

Module	Wake-Up Feature
FACE DETECT	None
ISS	None



Table 3-217 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-217. CD\_CAM Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
FACE DETECT	Master/Slave	CM_CAM_FDIF_CLKCTRL[18] STBYST	Standby status
		CM_CAM_FDIF_CLKCTRL[17:16] IDLEST	Idle status
ISS	Master/Slave	CM_CAM_ISS_CLKCTRL[18] STBYST	Standby status
		CM_CAM_ISS_CLKCTRL[17:16] IDLEST	Idle status

Table 3-218 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-218. CD\_CAM Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
FACE DETECT	Available	N/A	Available	CM_CAM_FDIF_CLKCTRL[1:0] MODULEMODE	Read/Write
ISS	Available	N/A	Available	CM_CAM_ISS_CLKCTRL[1:0] MODULEMODE	Read/Write

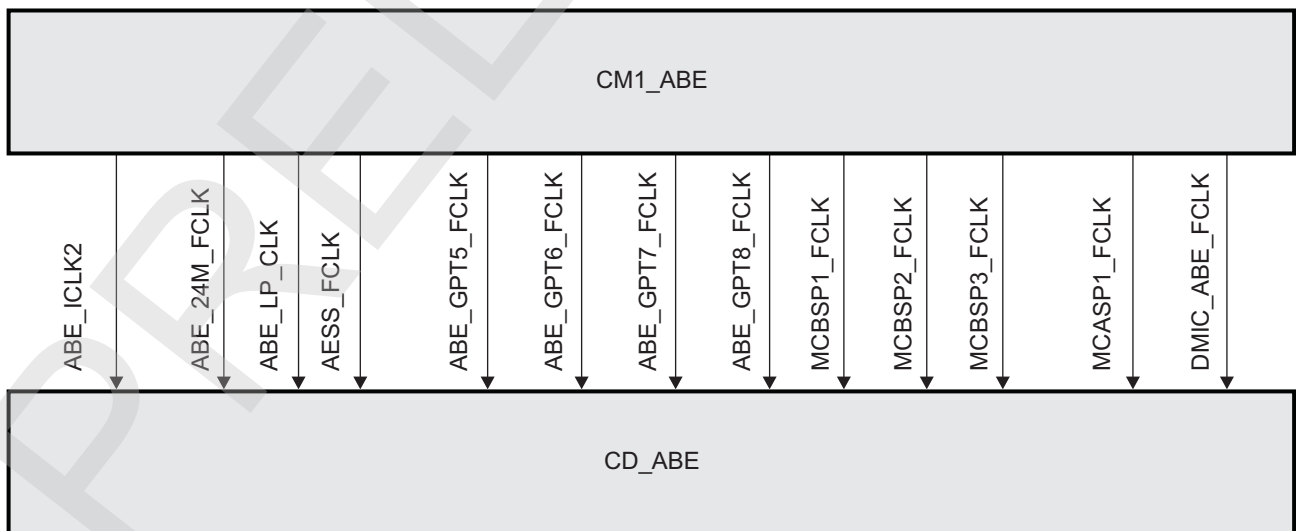
### 3.6.23 CD\_ABE Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.23.1 Overview

Figure 3-67 shows the overview of the clock domain.

**Figure 3-67. CD\_ABE Overview**



prcm-069

#### 3.6.23.2 Clock Domain Modes

Table 3-219 lists the clock domain modes supported by the clock domain.

**Table 3-219. CD\_ABE Clock Domain Modes**

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-220 lists the clock domain state transition control and status bits for the clock in this clock domain.

**Table 3-220. CD\_ABE Control and Status Parameters**

Parameter Name	Control/Status Bit Field
ABE_24M_FCLK clock status	CM1_ABE_CLKSTCTRL[13] CLKACTIVITY_ABE_24M_FCLK
ABE_ALWON_32K_CLK clock status	CM1_ABE_CLKSTCTRL[12] CLKACTIVITY_ABE_ALWON_32K_CLK
ABE_SYSCCLK clock status	CM1_ABE_CLKSTCTRL[11] CLKACTIVITY_ABE_SYSCCLK
ABE_X2_CLK clock status	CM1_ABE_CLKSTCTRL[8] CLKACTIVITY_ABE_X2_CLK
24M_FCLK clock status	CM1_ABE_CLKSTCTRL[10] CLKACTIVITY_24M_FCLK
ABE_ICLK2 clock status	CM1_ABE_CLKSTCTRL[9] CLKACTIVITY_ABE_ICLK2
Clock Domain State Transition Control	CM1_ABE_CLKSTCTRL[1:0] CLKTRCTRL

### 3.6.23.3 Clock Domain Dependency

CD\_ABE has no static or dynamic dependency with any other clock domain of the device.

#### 3.6.23.3.1 Wake-Up Dependency

Table 3-221 lists the wake-up dependency settings for the modules of this clock domain.

**Table 3-221. CD\_ABE Wake-Up Dependency Association Parameters**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DMIC	CD_ABE	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_ABE_DMIC_WKDEP[7] WKUPDEP_DMIC_DMA_SDMA	Read/Write
DMIC	CD_ABE	CD_DSP	Disabled	PM_ABE_DMIC_WKDEP[6] WKUPDEP_DMIC_DMA_DSP	Read/Write
DMIC	CD_ABE	CD_DSP	Disabled	PM_ABE_DMIC_WKDEP[2] WKUPDEP_DMIC_IRQ_DSP	Read/Write
DMIC	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_DMIC_WKDEP[0] WKUPDEP_DMIC_IRQ_MPU	Read/Write
MCASP1	CD_ABE	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_ABE_MCASP_WKDEP[7] WKUPDEP_MCASP1_DMA_SDMA	Read/Write
MCASP1	CD_ABE	CD_DSP	Disabled	PM_ABE_MCASP_WKDEP[6] WKUPDEP_MCASP1_DMA_DSP	Read/Write
MCASP1	CD_ABE	CD_DSP	Disabled	PM_ABE_MCASP_WKDEP[2] WKUPDEP_MCASP1_IRQ_DSP	Read/Write
MCASP1	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_MCASP_WKDEP[0] WKUPDEP_MCASP1_IRQ_MPU	Read/Write

**Table 3-221. CD\_ABE Wake-Up Dependency Association Parameters (continued)**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
MCBSP1	CD_ABE	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_ABE_MCBSP1_WKDEP[3] WKUPDEP_MCBS P1_SDMA	Read/Write
MCBSP1	CD_ABE	CD_DSP	Disabled	PM_ABE_MCBSP1_WKDEP[2] WKUPDEP_MCBS P1_DSP	Read/Write
MCBSP1	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_MCBSP1_WKDEP[0] WKUPDEP_MCBS P1_MPU	Read/Write
MCBSP2	CD_ABE	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_ABE_MCBSP2_WKDEP[3] WKUPDEP_MCBS P2_SDMA	Read/Write
MCBSP2	CD_ABE	CD_DSP	Disabled	PM_ABE_MCBSP2_WKDEP[2] WKUPDEP_MCBS P2_DSP	Read/Write
MCBSP2	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_MCBSP2_WKDEP[0] WKUPDEP_MCBS P2_MPU	Read/Write
MCBSP3	CD_ABE	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_ABE_MCBSP3_WKDEP[3] WKUPDEP_MCBS P3_SDMA	Read/Write
MCBSP3	CD_ABE	CD_DSP	Disabled	PM_ABE_MCBSP3_WKDEP[2] WKUPDEP_MCBS P3_DSP	Read/Write
MCBSP3	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_MCBSP3_WKDEP[0] WKUPDEP_MCBS P3_MPU	Read/Write
MCPDM	CD_ABE	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_ABE_PDM_WKDEP[7] WKUPDEP_PDM_D MA_SDMA	Read/Write
MCPDM	CD_ABE	CD_DSP	Disabled	PM_ABE_PDM_WKDEP[6] WKUPDEP_PDM_D MA_DSP	Read/Write
MCPDM	CD_ABE	CD_DSP	Disabled	PM_ABE_PDM_WKDEP[2] WKUPDEP_PDM_I RQ_DSP	Read/Write
MCPDM	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_PDM_WKDEP[0] WKUPDEP_PDM_I RQ_MPU	Read/Write
SLIMBUS1	CD_ABE	CD_DMA, CD_L3_1, CD_L3_2	Disabled	PM_ABE_SLIMBUS_WKDEP[7] WKUPDEP_SLIMBUS1_DMA_SDMA	Read/Write
SLIMBUS1	CD_ABE	CD_DSP	Disabled	PM_ABE_SLIMBUS_WKDEP[6] WKUPDEP_SLIMBUS1_DMA_DSP	Read/Write

**Table 3-221. CD\_ABE Wake-Up Dependency Association Parameters (continued)**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
SLIMBUS1	CD_ABE	CD_DSP	Disabled	PM_ABE_SLIMBUS_WKDEP[2] WKUPDEP_SLIMBUS1_IRQ_DSP	Read/Write
SLIMBUS1	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_SLIMBUS_WKDEP[0] WKUPDEP_SLIMBUS1_IRQ_MPU	Read/Write
GPTIMER5	CD_ABE	CD_DSP	Disabled	PM_ABE_GPTIME_R5_WKDEP[2] WKUPDEP_TIMER5_DSP	Read/Write
GPTIMER5	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_GPTIME_R5_WKDEP[0] WKUPDEP_TIMER5_MPU	Read/Write
GPTIMER6	CD_ABE	CD_DSP	Disabled	PM_ABE_GPTIME_R6_WKDEP[2] WKUPDEP_TIMER6_DSP	Read/Write
GPTIMER6	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_GPTIME_R6_WKDEP[0] WKUPDEP_TIMER6_MPU	Read/Write
GPTIMER7	CD_ABE	CD_DSP	Disabled	PM_ABE_GPTIME_R7_WKDEP[2] WKUPDEP_TIMER7_DSP	Read/Write
GPTIMER7	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_GPTIME_R7_WKDEP[0] WKUPDEP_TIMER7_MPU	Read/Write
GPTIMER8	CD_ABE	CD_DSP	Disabled	PM_ABE_GPTIME_R8_WKDEP[2] WKUPDEP_TIMER8_DSP	Read/Write
GPTIMER8	CD_ABE	CD_CORTEXA9	Disabled	PM_ABE_GPTIME_R8_WKDEP[0] WKUPDEP_TIMER8_MPU	Read/Write
WDTIMER3	CD_ABE	CD_CORTEXA9	Always enabled	PM_ABE_WDTIME_R3_WKDEP[0] WKUPDEP_WDT3_MPU	Read only

### 3.6.23.4 Clock Domain Module Attributes

Table 3-222 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-222. CD\_ABE Modules Clocks Association**

Module	Clock	Clock Type
AUDIO ENGINE	AESS_FCLK	Functional
	ABE_ICLK2	Interface
DMIC	DMIC_ABE_FCLK	Functional
	ABE_ICLK2	Interface
L4_ABE interconnect	ABE_ICLK2	Interface
MCASP1	MCASP1_FCLK	Functional

**Table 3-222. CD\_ABE Modules Clocks Association (continued)**

Module	Clock	Clock Type
	ABE_ICLK2	Interface and functional
MCBSP1	MCBSP1_FCLK	Functional
	ABE_ICLK2	Interface
MCBSP2	MCBSP2_FCLK	Functional
	ABE_ICLK2	Interface
MCBSP3	MCBSP3_FCLK	Functional
	ABE_ICLK2	Interface
MCPDM	ABE_ICLK2	Interface
	PAD_CLKS	Functional
SLIMBUS1	ABE_24M_FCLK	Functional
	24M_FCLK	Functional
	ABE_ICLK2	Interface
	PAD_CLKS	Functional
	SLIMBUS_UCLKS	Functional
GPTIMER5	ABE_GPT5_FCLK	Functional
	ABE_ICLK2	Interface
GPTIMER6	ABE_GPT6_FCLK	Functional
	ABE_ICLK2	Interface
GPTIMER7	ABE_GPT7_FCLK	Functional
	ABE_ICLK2	Interface
GPTIMER8	ABE_GPT8_FCLK	Functional
	ABE_ICLK2	Interface
WDTIMER3	ABE_ALWON_32K_CLK	Functional
	ABE_ICLK2	Interface

Table 3-223 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-223. CD\_ABE Modules Wake-Up Request**

Module	Wake-Up Feature
AUDIO ENGINE	None
DMIC	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ and DMA, sDMA -DMA)
L4_ABE interconnect	None
MCASP1	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ and DMA, sDMA -DMA)
MCBSP1	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ and DMA, sDMA -DMA)
MCBSP2	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ and DMA, sDMA -DMA)
MCBSP3	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ and DMA, sDMA -DMA)
MCPDM	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ and DMA, sDMA -DMA)
SLIMBUS1	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ and DMA, sDMA -DMA)
GPTIMER5	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ)
GPTIMER6	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ)
GPTIMER7	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ)
GPTIMER8	Slave wake-up request (MPU_A9 -IRQ, DSP -IRQ)

**Table 3-223. CD\_ABE Modules Wake-Up Request (continued)**

Module	Wake-Up Feature
WDTIMER3	Slave wake-up request (MPU_A9 -IRQ)

Table 3-224 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-224. CD\_ABE Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
AUDIO ENGINE	Slave/Master	CM1_ABE_AESS_CLKCTRL[18] STBYST	Standby status
		CM1_ABE_AESS_CLKCTRL[17:16] IDLEST	Idle status
DMIC	Slave	CM1_ABE_DMIC_CLKCTRL[17:16] IDLEST	Idle status
L4_ABE interconnect	Slave	CM1_ABE_L4ABE_CLKCTRL[17:16] IDLEST	Idle status
MCASP1	Slave	CM1_ABE_MCASP_CLKCTRL[17:16] IDLEST	Idle status
MCBSP1	Slave	CM1_ABE_MCBSP1_CLKCTRL[17:16] IDLEST	Idle status
MCBSP2	Slave	CM1_ABE_MCBSP2_CLKCTRL[17:16] IDLEST	Idle status
MCBSP3	Slave	CM1_ABE_MCBSP3_CLKCTRL[17:16] IDLEST	Idle status
MCPDM	Slave	CM1_ABE_PDM_CLKCTRL[17:16] IDLEST	Idle status
SLIMBUS1	Slave	CM1_ABE_SLIMBUS_CLKCTRL[17:16] IDLEST	Idle status
GPTIMER5	Slave	CM1_ABE_GPTIMER5_CLKCTRL[17:16] IDLEST	Idle status
GPTIMER6	Slave	CM1_ABE_GPTIMER6_CLKCTRL[17:16] IDLEST	Idle status
GPTIMER7	Slave	CM1_ABE_GPTIMER7_CLKCTRL[17:16] IDLEST	Idle status
GPTIMER8	Slave	CM1_ABE_GPTIMER8_CLKCTRL[17:16] IDLEST	Idle status
WDTIMER3	Slave	CM1_ABE_WDTIMER3_CLKCTRL[17:16] IDLEST	Idle status

Table 3-225 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-225. CD\_ABE Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
AUDIO ENGINE	Available	N/A	Available	CM1_ABE_AESS_CLKCTRL[1:0] MODULEMODE	Read/Write
DMIC	Available	N/A	Available	CM1_ABE_DMIC_CLKCTRL[1:0] MODULEMODE	Read/Write
L4_ABE interconnect	N/A	Available	N/A	CM1_ABE_L4ABE_CLKCTRL[1:0] MODULEMODE	Read only
MCASP1	Available	N/A	Available	CM1_ABE_MCASP_CLKCTRL[1:0] MODULEMODE	Read/Write

**Table 3-225. CD\_ABE Modules Slave Clock-Management Modes and Control (continued)**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
MCBSP1	Available	N/A	Available	CM1_ABE_MCBSP1_CLKCTRL[1:0] MODULEMODE	Read/Write
MCBSP2	Available	N/A	Available	CM1_ABE_MCBSP2_CLKCTRL[1:0] MODULEMODE	Read/Write
MCBSP3	Available	N/A	Available	CM1_ABE_MCBSP3_CLKCTRL[1:0] MODULEMODE	Read/Write
MCPDM	Available	N/A	Available	CM1_ABE_PDM_CLKCTRL[1:0] MODULEMODE	Read/Write
SLIMBUS1	Available	N/A	Available	CM1_ABE_SLIMBUS_CLKCTRL[1:0] MODULEMODE	Read/Write
GPTIMER5	Available	N/A	Available	CM1_ABE_GPTIMER5_CLKCTRL[1:0] MODULEMODE	Read/Write
GPTIMER6	Available	N/A	Available	CM1_ABE_GPTIMER6_CLKCTRL[1:0] MODULEMODE	Read/Write
GPTIMER7	Available	N/A	Available	CM1_ABE_GPTIMER7_CLKCTRL[1:0] MODULEMODE	Read/Write
GPTIMER8	Available	N/A	Available	CM1_ABE_GPTIMER8_CLKCTRL[1:0] MODULEMODE	Read/Write
WDTIMER3	Available	N/A	Available	CM1_ABE_WDTIMER3_CLKCTRL[1:0] MODULEMODE	Read/Write

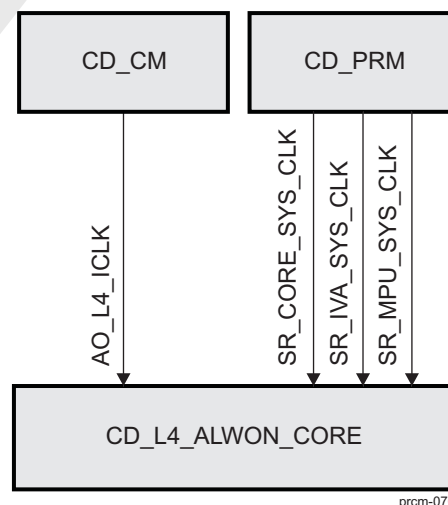
### 3.6.24 CD\_L4\_ALWON\_CORE Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

#### 3.6.24.1 Overview

Figure 3-68 shows the overview of the clock domain.

**Figure 3-68. CD\_L4\_ALWON\_CORE Overview**



prom-070

### 3.6.24.2 Clock Domain Modes

CD\_L4\_ALWON\_CORE is an always-on clock domain and has no software-controlled clock domain modes management associated.

### 3.6.24.3 Clock Domain Dependency

CD\_L4\_ALWON\_CORE has no static or dynamic dependency with any other clock domain of the device.

#### 3.6.24.3.1 Wake-Up Dependency

Table 3-226 lists the wake-up dependency settings for the modules of this clock domain.

**Table 3-226. CD\_L4\_ALWON\_CORE Wake-Up Dependency Association Parameters**

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
SR_CORE	CD_L4_ALWON_CORE	CD_CORTEXM3	Disabled	PM_ALWON_SR_CORE_WKDEP[1] WKUPDEP_SR_CORE_MPU_M3	Read/Write
	CD_L4_ALWON_CORE	CD_CORTEXA9	Disabled	PM_ALWON_SR_CORE_WKDEP[0] WKUPDEP_SR_CORE_MPU	Read/Write
SR_IVA	CD_L4_ALWON_CORE	CD_CORTEXM3	Disabled	PM_ALWON_SR_IVA_WKDEP[1] WKUPDEP_SR_IVA_MPU_M3	Read/Write
	CD_L4_ALWON_CORE	CD_CORTEXA9	Disabled	PM_ALWON_SR_IVA_WKDEP[0] WKUPDEP_SR_IVA_MPU	Read/Write
SR_MPU	CD_L4_ALWON_CORE	CD_CORTEXA9	Always enabled	PM_ALWON_SR_MPU_WKDEP[0] WKUPDEP_SR_MPU_MPU	Read only

### 3.6.24.4 Clock Domain Module Attributes

Table 3-227 lists for each module of the clock domain the clocks received by it and the role (that is, functional or interface clock).

**Table 3-227. CD\_L4\_ALWON\_CORE Modules Clocks Association**

Module	Clock	Clock Type
SR_CORE	AO_L4_ICLK	Interface
	SR_CORE_SYS_CLK	Functional
SR_IVA	AO_L4_ICLK	Interface
	SR_IVA_SYS_CLK	Functional
SR_MPU	AO_L4_ICLK	Interface
	SR_MPU_SYS_CLK	Functional

Table 3-228 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 3-228. CD\_L4\_ALWON\_CORE Modules Wake-Up Request**

Module	Wake-Up Feature
SR_CORE	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)
SR_IVA	Slave wake-up request (MPU_A9 -IRQ, MPU_M3 -IRQ)
SR_MPU	Slave wake-up request (MPU_A9 -IRQ)



Table 3-229 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-229. CD\_L4\_ALWON\_CORE Modules Clock-Management Modes and Control**

Module	Clock-Management Protocol	Status Bit Field	Role
SR_CORE	Slave	CM_ALWON_SR_CORE_CLKCTRL[17:16] IDLEST	Idle status
SR_IVA	Slave	CM_ALWON_SR_IVA_CLKCTRL[17:16] IDLEST	Idle status
SR_MPU	Slave	CM_ALWON_SR_MPU_CLKCTRL[17:16] IDLEST	Idle status

Table 3-230 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 3-230. CD\_L4\_ALWON\_CORE Modules Slave Clock-Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
SR_CORE	Available	N/A	Available	CM_ALWON_SR_CORE_CLKCTRL[1:0] MODULEMODE	Read/Write
SR_IVA	Available	N/A	Available	CM_ALWON_SR_IVA_CLKCTRL[1:0] MODULEMODE	Read/Write
SR_MPU	Available	N/A	Available	CM_ALWON_SR_MPU_CLKCTRL[1:0] MODULEMODE	Read/Write

### 3.7 Power Management Functional Description

This section highlights the functional concepts of power domain level power management in the device.

The following power domains support the DPS with switching times of less than 5  $\mu$ s.

- PD\_CORE
- PD\_DSP
- PD\_MPU

#### 3.7.1 PD\_WKUP Description

PD\_WKUP contains the following reset domains:

- WKUP\_PWRON\_RST
- WKUP\_RST
- WKUP\_SYS\_PWRON\_RST

PD\_WKUP contains CD\_WKUP clock domain.

Table 3-231 lists the logic retention capability for each module of the power domain.

**Table 3-231. PD\_WKUP Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
SYSCTRL_PADCONF_WKUP	No	None	None
BANDGAP	No	None	None
SYSCTRL_GENERAL_WKUP	No	None	None
GPIO1	No	RM_WKUP_GPIO1_CONTEXT[0] LOSTCONTEXT_DFF	None
KEYBOARD	No	RM_WKUP_KEYBOARD_CONTEXT[0] LOSTCONTEXT_DFF	None
PRM	No	None	None

**Table 3-231. PD\_WKUP Modules Power Attributes (continued)**

Module	Logic Retention	DFF Context Status	RFF Context Status
SAR_RAM	No	<a href="#">RM_WKUP_SARRAM_CONTEXT[0]</a> LOSTCONTEXT_DFF	None
SCRM	No	None	None
32KTIMER	No	<a href="#">RM_WKUP_32KTIMER_CONTEXT[0]</a> LOSTCONTEXT_DFF	None
GPTIMER1	No	<a href="#">RM_WKUP_GPTIMER1_CONTEXT[0]</a> LOSTCONTEXT_DFF	None
WDTIMER2	No	<a href="#">RM_WKUP_WDTIMER2_CONTEXT[0]</a> LOSTCONTEXT_DFF	None
L4_WKUP interconnect	No	<a href="#">RM_WKUP_L4WKUP_CONTEXT[0]</a> LOSTCONTEXT_DFF	None

### 3.7.1.1 Power Domain Modes

The PD\_WKUP power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power state control or status bit field for this power domain.

The PD\_WKUP power domain has no memory banks.

### 3.7.2 PD\_DSP Description

PD\_DSP contains the following reset domains:

- DSP\_RST
- DSP\_PWRON\_RST
- DSP\_RET\_RST
- DSP\_SYS\_RST

PD\_DSP contains the CD\_DSP clock domain.

[Table 3-232](#) lists the logic retention capability for each module of the power domain.

**Table 3-232. PD\_DSP Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
DSP	Partial	<a href="#">RM_DSP_DSP_CONTEXT[0]</a> LOSTCONTEXT_DFF	<a href="#">RM_DSP_DSP_CONTEXT[1]</a> LOSTCONTEXT_RFF

### 3.7.2.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See [Section 3.1.1.2.1](#), for a functional description of the power-management architecture of a generic power domain.

#### 3.7.2.1.1 Logic and Memory Area Power Modes

[Table 3-233](#) lists the power modes supported by the logic area of the power domain.

**Table 3-233. PD\_DSP Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Available	Available	Available	Available

[Table 3-234](#) presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-234. PD\_DSP Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
DSP_EDMA		ON	RETENTION	OFF	
	DSP - DSP_EDMA	always_on	always_retention		always_enabled
DSP_L1		ON	OFF, RETENTION	OFF	
	DSP - DSP_L1	always_on	software_control		None
DSP_L2		ON	OFF, RETENTION	OFF	
	DSP - DSP_L2	always_on	software_control		always_enabled

### 3.7.2.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-235 lists the power modes controls for the power domain.

**Table 3-235. PD\_DSP Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in Retention state)	DSP_L2	PM_DSP_PWRSTCTRL[9] DSP_L2_RETSTATE	Read/Write
Memory Area - State Control (Logic in Retention state)	DSP_L1	PM_DSP_PWRSTCTRL[8] DSP_L1_RETSTATE	Read/Write
Power Domain - Low Power State Change Control		PM_DSP_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/Write
Memory Area - State Control (Logic in On state)	DSP_EDMA	PM_DSP_PWRSTCTRL[21:20] DSP_EDMA_ONSTATE	Read only
Logic Area - Retention State Control		PM_DSP_PWRSTCTRL[2] LOGICRETSTATE	Read/Write
Memory Area - State Control (Logic in On state)	DSP_L2	PM_DSP_PWRSTCTRL[19:18] DSP_L2_ONSTATE	Read only
Memory Area - State Control (Logic in On state)	DSP_L1	PM_DSP_PWRSTCTRL[17:16] DSP_L1_ONSTATE	Read only
Memory Area - State Control (Logic in Retention state)	DSP_EDMA	PM_DSP_PWRSTCTRL[10] DSP_EDMA_RETSTATE	Read only
Power Domain - State Transition Control		PM_DSP_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-236 lists the power modes status for the power domain.

**Table 3-236. PD\_DSP Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Power Domain - Last Power State Entered Status		PM_DSP_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area - State Status	DSP_EDMA	PM_DSP_PWRSTST[9:8] DSP_EDMA_STATEST
Memory Area - State Status	DSP_L2	PM_DSP_PWRSTST[7:6] DSP_L2_STATEST
Memory Area - State Status	DSP_L1	PM_DSP_PWRSTST[5:4] DSP_L1_STATEST
Power Domain - State Transition Status		PM_DSP_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_DSP_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_DSP_PWRSTST[1:0] POWERSTATEST

### 3.7.3 PD\_STD\_EFUSE Description

PD\_STD\_EFUSE contains the STD\_EFUSE\_PWRON\_RST reset domain.

PD\_STD\_EFUSE contains the CD\_STD\_EFUSE clock domain.

Table 3-237 lists the logic retention capability for each module of the power domain.

**Table 3-237. PD\_STD\_EFUSE Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
STD_EFUSE	No	None	None

### 3.7.3.1 Power Domain Modes

The PD\_STD\_EFUSE power domain is hardware-controlled by the PRCM module and has no software control associated. It is switched to ON power state when the device is powered on and switched to OFF power state after the power-up sequence of the device.

The PD\_STD\_EFUSE power domain has no memory banks.

### 3.7.4 PD\_MPU Description

PD\_MPU contains the following reset domains:

- CORTEXA9\_PWRON\_RST
- CORTEXA9\_RST

PD\_MPU contains the CD\_CORTEXA9 clock domain.

Table 3-238 lists the logic retention capability for each module of the power domain.

**Table 3-238. PD\_MPU Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
CORTEXA9	No	RM_MPU_MPU_CONTEXT[0] LOSTCONTEXT_DFF	None

### 3.7.4.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See Section 3.1.1.2.1 for a functional description of the power-management architecture of a generic power domain.

#### 3.7.4.1.1 Logic and Memory Area Power Modes

Table 3-239 lists the power modes supported by the logic area of the power domain.

**Table 3-239. PD\_MPU Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Available	Available	Available	Available

Table 3-240 presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-240. PD\_MPU Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
MPU_L1		ON	OFF, RETENTION	OFF	
	CORTEXA9 - MPU_L1	always_on	software_control		None
MPU_L2		ON	OFF, RETENTION	OFF	
	CORTEXA9 - MPU_L2	always_on	software_control		always_enabled

**Table 3-240. PD\_MPU Memory Area Power Modes (continued)**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
MPU_RAM		ON	RETENTION	OFF	
	CORTEXA9 - MPU_RAM	always_on	always_retention		always_enabled

### 3.7.4.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-241 lists the power modes controls for the power domain.

**Table 3-241. PD\_MPU Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in Retention state)	MPU_L2	PM_MPU_PWRSTCTRL[9] MPU_L2_RETSTATE	Read/Write
Memory Area - State Control (Logic in Retention state)	MPU_L1	PM_MPU_PWRSTCTRL[8] MPU_L1_RETSTATE	Read/Write
Power Domain - Low Power State Change Control		PM_MPU_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read only
Memory Area - State Control (Logic in On state)	MPU_RAM	PM_MPU_PWRSTCTRL[21:20] ] MPU_RAM_ONSTATE	Read only
Logic Area - Retention State Control		PM_MPU_PWRSTCTRL[2] LOGICRETSTATE	Read/Write
Memory Area - State Control (Logic in On state)	MPU_L2	PM_MPU_PWRSTCTRL[19:18] ] MPU_L2_ONSTATE	Read only
Memory Area - State Control (Logic in On state)	MPU_L1	PM_MPU_PWRSTCTRL[17:16] ] MPU_L1_ONSTATE	Read only
Memory Area - State Control (Logic in Retention state)	MPU_RAM	PM_MPU_PWRSTCTRL[10] MPU_RAM_RETSTATE	Read only
Power Domain - State Transition Control		PM_MPU_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-242 lists the power modes status for the power domain.

**Table 3-242. PD\_MPU Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Power Domain - Last Power State Entered Status		PM_MPU_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area - State Status	MPU_RAM	PM_MPU_PWRSTST[9:8] MPU_RAM_STATEST
Memory Area - State Status	MPU_L2	PM_MPU_PWRSTST[7:6] MPU_L2_STATEST
Memory Area - State Status	MPU_L1	PM_MPU_PWRSTST[5:4] MPU_L1_STATEST
Power Domain - State Transition Status		PM_MPU_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_MPU_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_MPU_PWRSTST[1:0] POWERSTATEST

### 3.7.5 PD\_L4\_PER Description

PD\_L4\_PER contains the following reset domains:

- L4\_PER\_PWRON\_RET\_RST
- L4\_PER\_RET\_RST
- L4\_PER\_RST

PD\_L4\_PER contains the CD\_L4\_PER clock domain.

Table 3-243 lists the logic retention capability for each module of the power domain.

**Table 3-243. PD\_L4\_PER Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
GPTIMER10	No	RM_L4PER_GPTIMER10_CONTEXT[0] LOSTCONTEXT_DFF	None
GPTIMER11	No	RM_L4PER_GPTIMER11_CONTEXT[0] LOSTCONTEXT_DFF	None
GPTIMER2	No	RM_L4PER_GPTIMER2_CONTEXT[0] LOSTCONTEXT_DFF	None
GPTIMER3	No	RM_L4PER_GPTIMER3_CONTEXT[0] LOSTCONTEXT_DFF	None
GPTIMER4	No	RM_L4PER_GPTIMER4_CONTEXT[0] LOSTCONTEXT_DFF	None
GPTIMER9	No	RM_L4PER_GPTIMER9_CONTEXT[0] LOSTCONTEXT_DFF	None
ELM	No	RM_L4PER_ELM_CONTEXT[0] LOSTCONTEXT_DFF	None
GPIO2	Full	None	RM_L4PER_GPIO2_CONTEXT[1] LOSTCONTEXT_RFF
GPIO3	Full	None	RM_L4PER_GPIO3_CONTEXT[1] LOSTCONTEXT_RFF
GPIO4	Full	None	RM_L4PER_GPIO4_CONTEXT[1] LOSTCONTEXT_RFF
GPIO5	Full	None	RM_L4PER_GPIO5_CONTEXT[1] LOSTCONTEXT_RFF
GPIO6	Full	None	RM_L4PER_GPIO6_CONTEXT[1] LOSTCONTEXT_RFF
HDQ	No	RM_L4PER_HDQ1W_CONTEXT[0] LOSTCONTEXT_DFF	None
I2C1	Full	None	RM_L4PER_I2C1_CONTEXT[1] LOSTCONTEXT_RFF
I2C2	No	RM_L4PER_I2C2_CONTEXT[0] LOSTCONTEXT_DFF	None
I2C3	No	RM_L4PER_I2C3_CONTEXT[0] LOSTCONTEXT_DFF	None
I2C4	No	RM_L4PER_I2C4_CONTEXT[0] LOSTCONTEXT_DFF	None
L4_PER interconnect	Partial	RM_L4PER_L4_PER_CONTEXT[0] LOSTCONTEXT_DFF	RM_L4PER_L4_PER_CONTEXT[1] LOSTCONTEXT_RFF
MCBSP4	No	RM_L4PER_MCBSP4_CONTEXT[0] LOSTCONTEXT_DFF	None
MCSP1	No	RM_L4PER_MCSP1_CONTEXT[0] LOSTCONTEXT_DFF	None
MCSP2	No	RM_L4PER_MCSP2_CONTEXT[0] LOSTCONTEXT_DFF	None
MCSP3	No	RM_L4PER_MCSP3_CONTEXT[0] LOSTCONTEXT_DFF	None
MCSP4	No	RM_L4PER_MCSP4_CONTEXT[0] LOSTCONTEXT_DFF	None
HSMMC3	No	RM_L4PER_MMCS3_CONTEXT[0] LOSTCONTEXT_DFF	None
HSMMC4	No	RM_L4PER_MMCS4_CONTEXT[0] LOSTCONTEXT_DFF	None
HSMMC5	No	RM_L4PER_MMCS5_CONTEXT[0] LOSTCONTEXT_DFF	None

**Table 3-243. PD\_L4\_PER Modules Power Attributes (continued)**

Module	Logic Retention	DFF Context Status	RFF Context Status
SLIMBUS2	No	<a href="#">RM_L4PER_SLIMBUS2_CONTEXT[0] LOSTCONTEXT_DFF</a>	None
UART1	Full	None	<a href="#">RM_L4PER_UART1_CONTEXT[1] LOSTCONTEXT_RFF</a>
UART2	Full	None	<a href="#">RM_L4PER_UART2_CONTEXT[1] LOSTCONTEXT_RFF</a>
UART3	Full	None	<a href="#">RM_L4PER_UART3_CONTEXT[1] LOSTCONTEXT_RFF</a>
UART4	Full	None	<a href="#">RM_L4PER_UART4_CONTEXT[1] LOSTCONTEXT_RFF</a>

### 3.7.5.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See [Section 3.1.1.2.1](#) for a functional description of the power-management architecture of a generic power domain.

#### 3.7.5.1.1 Logic and Memory Area Power Modes

[Table 3-244](#) lists the power modes supported by the logic area of the power domain.

**Table 3-244. PD\_L4\_PER Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Available	Available	Available	Available

[Table 3-245](#) presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-245. PD\_L4\_PER Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
NONRETAINED_BANK		ON	OFF	OFF	
	MCBSP4 - MCBSP_BANK	always_on	always_off		None
	HSMC3 - MMCHSSRAM	always_on	always_off		None
	SLIMBUS2 - SLIMBUS_BANK	always_on	always_off		None
RETAINED_BANK		ON	RETENTION	OFF	
	UART1 - UART_MEM	always_on	always_retention		None

#### 3.7.5.1.2 Logic and Memory Area Power Modes Control and Status

[Table 3-246](#) lists the power modes controls for the power domain.

**Table 3-246. PD\_L4\_PER Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in Retention state)	NONRETAINED_BANK	<a href="#">PM_L4PER_PWRSTCTRL[9] NONRETAINED_BANK_RETSTATE</a>	Read only

**Table 3-246. PD\_L4\_PER Power Modes Control Parameters (continued)**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in Retention state)	RETAINED_BANK	PM_L4PER_PWRSTCTRL[8] RETAINED_BANK_RETSTATE	Read only
Power Domain - Low Power State Change Control		PM_L4PER_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/Write
Logic Area - Retention State Control		PM_L4PER_PWRSTCTRL[2] LOGICRETSTATE	Read/Write
Memory Area - State Control (Logic in On state)	NONRETAINED_BANK	PM_L4PER_PWRSTCTRL[19: 18] NONRETAINED_BANK_ONSTATE	Read only
Memory Area - State Control (Logic in On state)	RETAINED_BANK	PM_L4PER_PWRSTCTRL[17: 16] RETAINED_BANK_ONSTATE	Read only
Power Domain - State Transition Control		PM_L4PER_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-247 lists the power modes status for the power domain.

**Table 3-247. PD\_L4\_PER Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Power Domain - Last Power State Entered Status		PM_L4PER_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area - State Status	NONRETAINED_BANK	PM_L4PER_PWRSTST[7:6] NONRETAINED_BANK_STATEST
Memory Area - State Status	RETAINED_BANK	PM_L4PER_PWRSTST[5:4] RETAINED_BANK_STATEST
Power Domain - State Transition Status		PM_L4PER_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_L4PER_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_L4PER_PWRSTST[1:0] POWERSTATEST

### 3.7.6 PD\_L3\_INIT Description

PD\_L3\_INIT contains the following reset domains:

- DPLL\_L3\_INIT\_PWRON\_RET\_RST
- L3\_INIT\_PWRON\_RST
- L3\_INIT\_RET\_RST
- L3\_INIT\_RST

PD\_L3\_INIT contains the CD\_L3\_INIT clock domain.

Table 3-248 lists the logic retention capability for each module of the power domain.

**Table 3-248. PD\_L3\_INIT Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
CM1_USB	No	None	None
DPLL_USB	Full	None	None
HSI	Full	None	RM_L3INIT_HSI_CONTEXT[1] LOSTCONTEXT_RFF
P1500	No	RM_L3INIT_P1500_CONTEXT [0] LOSTCONTEXT_DFF	None
HSMMC1	Full	None	RM_L3INIT_MMC1_CONTEXT [1] LOSTCONTEXT_RFF



**Table 3-248. PD\_L3\_INIT Modules Power Attributes (continued)**

Module	Logic Retention	DFP Context Status	RFF Context Status
HSMMC2	Full	None	<a href="#">RM_L3INIT_MMC2_CONTEXT[1] LOSTCONTEXT_RFF</a>
USBPHY	No	<a href="#">RM_L3INIT_USBPHY_CONTEXT[0] LOSTCONTEXT_DFF</a>	None
FSUSB	Full	None	<a href="#">RM_L3INIT_USB_HOST_FS_CONTEXT[1] LOSTCONTEXT_RFF</a>
HSUSBHOST	Full	None	<a href="#">RM_L3INIT_USB_HOST_CONTEXT[1] LOSTCONTEXT_RFF</a>
HSUSBOTG	Full	None	<a href="#">RM_L3INIT_USB_OTG_CONTEXT[1] LOSTCONTEXT_RFF</a>
HSUSBTLL	Full	None	<a href="#">RM_L3INIT_HSUSBTLL_CONTEXT[1] LOSTCONTEXT_RFF</a>

### 3.7.6.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See [Section 3.1.1.2.1](#) for a functional description of the power-management architecture of a generic power domain.

#### 3.7.6.1.1 Logic and Memory Area Power Modes

[Table 3-249](#) lists the power modes supported by the logic area of the power domain.

**Table 3-249. PD\_L3\_INIT Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Available	Available	Available	Available

[Table 3-250](#) presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-250. PD\_L3\_INIT Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
L3INIT_BANK1		ON	OFF	OFF	
	HSI - HSI_DMAFIFO	always_on	always_off		None
	HSMMC1 - HSMMCSRAM	always_on	always_off		None
	HSUSBOTG - USBMEM	always_on	always_off		always_enabled

#### 3.7.6.1.2 Logic and Memory Area Power Modes Control and Status

[Table 3-251](#) lists the power modes controls for the power domain.

**Table 3-251. PD\_L3\_INIT Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in Retention state)	L3INIT_BANK1	<a href="#">PM_L3INIT_PWRSTCTRL[8] L3INIT_BANK1_RETSTATE</a>	Read only
Power Domain - Low Power State Change Control		<a href="#">PM_L3INIT_PWRSTCTRL[4] LOWPOWERSTATECHANGE</a>	Read/Write

**Table 3-251. PD\_L3\_INIT Power Modes Control Parameters (continued)**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Logic Area - Retention State Control		PM_L3INIT_PWRSTCTRL[2] LOGICRETSTATE	Read/Write
Memory Area - State Control (Logic in On state)	L3INIT_BANK1	PM_L3INIT_PWRSTCTRL[17:16] L3INIT_BANK1_ONSTATE	Read only
Power Domain - State Transition Control		PM_L3INIT_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-252 lists the power modes status for the power domain.

**Table 3-252. PD\_L3\_INIT Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Power Domain - Last Power State Entered Status		PM_L3INIT_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area - State Status	L3INIT_BANK1	PM_L3INIT_PWRSTST[5:4] L3INIT_BANK1_STATEST
Power Domain - State Transition Status		PM_L3INIT_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_L3INIT_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_L3INIT_PWRSTST[1:0] POWERSTATEST

### 3.7.7 PD\_IVAHD Description

PD\_IVAHD contains the following reset domains:

- IVAHD\_PWRON\_RST
- IVAHD\_RST
- IVAHD\_SEQ1\_RST
- IVAHD\_SEQ2\_RST

PD\_IVAHD contains the CD\_IVAHD clock domain.

Table 3-253 lists the logic retention capability for each module of the power domain.

**Table 3-253. PD\_IVAHD Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
IVAHD	No	RM_IVAHD_IVAHD_CONTEX T[0] LOSTCONTEXT_DFF	None
SL2	No	RM_IVAHD_SL2_CONTEXT[0] LOSTCONTEXT_DFF	None

#### 3.7.7.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See Section 3.1.1.2.1 for a functional description of the power-management architecture of a generic power domain.

##### 3.7.7.1.1 Logic and Memory Area Power Modes

Table 3-254 lists the power modes supported by the logic area of the power domain.

**Table 3-254. PD\_IVAHD Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Available	Not available	Available	Available

Table 3-255 presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-255. PD\_IVAHD Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
HWA_MEM		ON	OFF	OFF	
	IVAHD - HWA_MEM	always_on	always_off		always_enabled
SL2_MEM		ON	OFF, RETENTION	OFF	
	SL2 - SL2MEM	always_on	software_control		always_enabled
TCM1_MEM		ON	OFF, RETENTION	OFF	
	IVAHD - TCM_1	always_on	software_control		always_enabled
TCM2_MEM		ON	OFF, RETENTION	OFF	
	IVAHD - TCM_2	always_on	software_control		always_enabled

### 3.7.7.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-256 lists the power modes controls for the power domain.

**Table 3-256. PD\_IVAHD Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in Retention state)	SL2_MEM	PM_IVAHD_PWRSTCTRL[9] SL2_MEM_RETSTATE	Read/Write
Memory Area - State Control (Logic in Retention state)	HWA_MEM	PM_IVAHD_PWRSTCTRL[8] HWA_MEM_RETSTATE	Read only
Power Domain - Low Power State Change Control		PM_IVAHD_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/Write
Memory Area - State Control (Logic in On state)	TCM2_MEM	PM_IVAHD_PWRSTCTRL[23: 22] TCM2_MEM_ONSTATE	Read only
Memory Area - State Control (Logic in On state)	TCM1_MEM	PM_IVAHD_PWRSTCTRL[21: 20] TCM1_MEM_ONSTATE	Read only
Logic Area - Retention State Control		PM_IVAHD_PWRSTCTRL[2] LOGICRETSTATE	Read only
Memory Area - State Control (Logic in On state)	SL2_MEM	PM_IVAHD_PWRSTCTRL[19: 18] SL2_MEM_ONSTATE	Read only
Memory Area - State Control (Logic in On state)	HWA_MEM	PM_IVAHD_PWRSTCTRL[17: 16] HWA_MEM_ONSTATE	Read only
Memory Area - State Control (Logic in Retention state)	TCM2_MEM	PM_IVAHD_PWRSTCTRL[11] TCM2_MEM_RETSTATE	Read/Write
Memory Area - State Control (Logic in Retention state)	TCM1_MEM	PM_IVAHD_PWRSTCTRL[10] TCM1_MEM_RETSTATE	Read/Write
Power Domain - State Transition Control		PM_IVAHD_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-257 lists the power modes status for the power domain.

**Table 3-257. PD\_IVAHD Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Power Domain - Last Power State Entered Status		PM_IVAHD_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area - State Status	TCM1_MEM	PM_IVAHD_PWRSTST[9:8] TCM1_MEM_STATEST

**Table 3-257. PD\_IVAHD Power Modes Status Parameters (continued)**

Parameter Name	Memory Bank	Status Bit Field
Memory Area - State Status	SL2_MEM	PM_IVAHD_PWRSTST[7:6] SL2_MEM_STATEST
Memory Area - State Status	HWA_MEM	PM_IVAHD_PWRSTST[5:4] HWA_MEM_STATEST
Power Domain - State Transition Status		PM_IVAHD_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_IVAHD_PWRSTST[2] LOGICSTATEST
Memory Area - State Status	TCM2_MEM	PM_IVAHD_PWRSTST[11:10] TCM2_MEM_STATEST
Power Domain - State Status		PM_IVAHD_PWRSTST[1:0] POWERSTATEST

### 3.7.8 PD\_SGX Description

PD\_SGX contains the SGX\_RST reset domain.

PD\_SGX contains the CD\_SGX clock domain.

Table 3-258 lists the logic retention capability for each module of the power domain.

**Table 3-258. PD\_SGX Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
SGX	No	RM_SGX_SGX_CONTEXT[0] LOSTCONTEXT_DFF	None

#### 3.7.8.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See Section 3.1.1.2.1 for a functional description of the power-management architecture of a generic power domain.

##### 3.7.8.1.1 Logic and Memory Area Power Modes

Table 3-259 lists the power modes supported by the logic area of the power domain.

**Table 3-259. PD\_SGX Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Not available	Not available	Available	Available

Table 3-260 presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-260. PD\_SGX Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
	SGX - SGX_MEM	always_on	always_off		always_enabled

##### 3.7.8.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-261 lists the power modes controls for the power domain.

**Table 3-261. PD\_SGX Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain - Low Power State Change Control		PM_SGX_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/Write
Memory Area - State Control (Logic in On state)	SGX_MEM	PM_SGX_PWRSTCTRL[17:16] SGX_MEM_ONSTATE	Read only
Power Domain - State Transition Control		PM_SGX_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-262 lists the power modes status for the power domain.

**Table 3-262. PD\_SGX Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Memory Area - State Status	SGX_MEM	PM_SGX_PWRSTST[5:4] SGX_MEM_STATEST
Power Domain - State Transition Status		PM_SGX_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_SGX_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_SGX_PWRSTST[1:0] POWERSTATEST

### 3.7.9 PD\_EMU Description

PD\_EMU contains the following reset domains:

- EMU\_EARLY\_PWRON\_RST
- EMU\_PWRON\_RST
- EMU\_RST

PD\_EMU contains the CD\_EMU clock domain.

Table 3-263 lists the logic retention capability for each module of the power domain.

**Table 3-263. PD\_EMU Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
CM1_EMU	No	None	None
DEBUG	No	None	None

### 3.7.9.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See Section 3.1.1.2.1 for a functional description of the power-management architecture of a generic power domain.

#### 3.7.9.1.1 Logic and Memory Area Power Modes

Table 3-264 lists the power modes supported by the logic area of the power domain.

**Table 3-264. PD\_EMU Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Not available	Not available	Not available	Available

Table 3-265 presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-265. PD\_EMU Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
	DEBUG - DebugSS_MEM	always_on	always_off		always_enabled

### 3.7.9.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-266 lists the power modes controls for the power domain.

**Table 3-266. PD\_EMU Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in On state)	EMU_BANK	PM_EMU_PWRSTCTRL[17:16] ] EMU_BANK_ONSTATE	Read only
Power Domain - State Transition Control		PM_EMU_PWRSTCTRL[1:0] POWERSTATE	Read only

Table 3-267 lists the power modes status for the power domain.

**Table 3-267. PD\_EMU Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Memory Area - State Status	EMU_BANK	PM_EMU_PWRSTST[5:4] EMU_BANK_STATEST
Power Domain - State Transition Status		PM_EMU_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_EMU_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_EMU_PWRSTST[1:0] POWERSTATEST

## 3.7.10 PD\_DSS Description

PD\_DSS contains the following reset domains:

- DSS\_RET\_RST
- DSS\_RST

PD\_DSS contains the CD\_DSS clock domain.

Table 3-268 lists the logic retention capability for each module of the power domain.

**Table 3-268. PD\_DSS Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
DSS	Partial	RM_DSS_DSS_CONTEXT[0] LOSTCONTEXT_DFF	RM_DSS_DSS_CONTEXT[1] LOSTCONTEXT_RFF

### 3.7.10.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See Section 3.1.1.2.1 for a functional description of the power-management architecture of a generic power domain.

#### 3.7.10.1.1 Logic and Memory Area Power Modes

Table 3-269 lists the power modes supported by the logic area of the power domain.

**Table 3-269. PD\_DSS Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Available	Available	Available	Available

Table 3-270 presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-270. PD\_DSS Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
DSS_MEM		ON	OFF	OFF	
	DSS - DSSMEM	always_on	always_off		always_enabled

### 3.7.10.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-271 lists the power modes controls for the power domain.

**Table 3-271. PD\_DSS Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in Retention state)	DSS_MEM	PM_DSS_PWRSTCTRL[8] DSS_MEM_RETSTATE	Read only
Power Domain - Low Power State Change Control		PM_DSS_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/Write
Logic Area - Retention State Control		PM_DSS_PWRSTCTRL[2] LOGICRETSTATE	Read/Write
Memory Area - State Control (Logic in On state)	DSS_MEM	PM_DSS_PWRSTCTRL[17:16] DSS_MEM_ONSTATE	Read only
Power Domain - State Transition Control		PM_DSS_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-272 lists the power modes status for the power domain.

**Table 3-272. PD\_DSS Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Memory Area - State Status	DSS_MEM	PM_DSS_PWRSTST[5:4] DSS_MEM_STATEST
Power Domain - State Transition Status		PM_DSS_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_DSS_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_DSS_PWRSTST[1:0] POWERSTATEST

### 3.7.11 PD\_CORE Description

PD\_CORE contains the following reset domains:

- CM2\_PWRON\_RET\_RST
- CM2\_RET\_RST
- CORE\_PWRON\_RET\_RST
- CORE\_RET\_RST
- CORE\_RST
- DLL\_RST
- CORTEXM3\_PWRON\_RST
- CORTEXM3\_RET\_RST
- CORTEXM3\_RST1
- CORTEXM3\_RST2
- CORTEXM3\_RST3
- EMIF\_PWRON\_RST

- SDMA\_RET\_RST

PD\_CORE contains the following clock domains:

- CD\_C2C
- CD\_DMA
- CD\_CORTEXM3
- CD\_EMIF
- CD\_L3\_1
- CD\_L3\_2
- CD\_L3\_INSTR
- CD\_L4\_CFG

Table 3-273 lists the logic retention capability for each module of the power domain.

**Table 3-273. PD\_CORE Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
SYSCTRL_PADCONF_CORE	Full	None	None
CM2	Full	None	None
DDRPHY	No	None	None
SYSCTRL_GENERAL_CORE	Full	None	None
DLL	No	None	None
DMM	Partial	RM_MEMIF_DMM_CONTEXT[0] LOSTCONTEXT_DFF	RM_MEMIF_DMM_CONTEXT[1] LOSTCONTEXT_RFF
CORTEXM3	Partial	RM_MPU_M3_MPU_M3_CONTEXT[0] LOSTCONTEXT_DFF	RM_MPU_M3_MPU_M3_CONTEXT[1] LOSTCONTEXT_RFF
EMIF1	Full	RM_MEMIF_EMIF_1_CONTEXT[0] LOSTCONTEXT_DFF	RM_MEMIF_EMIF_1_CONTEXT[1] LOSTCONTEXT_RFF
EMIF2	Full	RM_MEMIF_EMIF_2_CONTEXT[0] LOSTCONTEXT_DFF	RM_MEMIF_EMIF_2_CONTEXT[1] LOSTCONTEXT_RFF
EMIF_FW	Partial	RM_MEMIF_EMIF_FW_CONTEXT[0] LOSTCONTEXT_DFF	RM_MEMIF_EMIF_FW_CONTEXT[1] LOSTCONTEXT_RFF
GPMC	Full	None	RM_L3_2_GPMC_CONTEXT[1] LOSTCONTEXT_RFF
SPINLOCK	Full	None	RM_L4CFG_SPINLOCK_CONTEXT[1] LOSTCONTEXT_RFF
L3_2 interconnect	Partial	RM_L3_2_L3_2_CONTEXT[0] LOSTCONTEXT_DFF	RM_L3_2_L3_2_CONTEXT[1] LOSTCONTEXT_RFF
L3_3 interconnect	Partial	RM_L3INSTR_L3_3_CONTEXT[0] LOSTCONTEXT_DFF	RM_L3INSTR_L3_3_CONTEXT[1] LOSTCONTEXT_RFF
L3_1 interconnect	Partial	RM_L3_1_L3_1_CONTEXT[0] LOSTCONTEXT_DFF	RM_L3_1_L3_1_CONTEXT[1] LOSTCONTEXT_RFF
L3_INSTR interconnect	No	RM_L3INSTR_L3_INSTR_CONTEXT[0] LOSTCONTEXT_DFF	None
L4_CFG interconnect	Partial	RM_L4CFG_L4_CFG_CONTEXT[0] LOSTCONTEXT_DFF	RM_L4CFG_L4_CFG_CONTEXT[1] LOSTCONTEXT_RFF
Mailbox	Full	None	RM_L4CFG_MAILBOX_CONTEXT[1] LOSTCONTEXT_RFF
OCMC_RAM	No	RM_L3_2_OCMC_RAM_CONTEXT[0] LOSTCONTEXT_DFF	None
C2C	Partial	RM_C2C_C2C_CONTEXT[0] LOSTCONTEXT_DFF	RM_C2C_C2C_CONTEXT[1] LOSTCONTEXT_RFF
C2C_FW	Partial	RM_C2C_C2C_FW_CONTEXT[0] LOSTCONTEXT_DFF	RM_C2C_C2C_FW_CONTEXT[1] LOSTCONTEXT_RFF
SAR_ROM	No	RM_L4CFG_SAR_ROM_CONTEXT[0] LOSTCONTEXT_DFF	None



**Table 3-273. PD\_CORE Modules Power Attributes (continued)**

Module	Logic Retention	DFF Context Status	RFF Context Status
sDMA	Full	None	RM_SDMA_SDMA_CONTEXT[1] LOSTCONTEXT_RFF

### 3.7.11.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See [Section 3.1.1.2.1](#) for a functional description of the power-management architecture of a generic power domain.

#### 3.7.11.1.1 Logic and Memory Area Power Modes

[Table 3-274](#) lists the power modes supported by the logic area of the power domain.

**Table 3-274. PD\_CORE Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Available	Available	Available	Available

[Table 3-275](#) presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-275. PD\_CORE Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	logic off	RTA Feature
CORE_NRET_BANK		ON	OFF	OFF	
	DMM - DMM_MEMBANK2	always_on	always_off		None
CORE_OCMRAM		OFF, RETENTION, ON	OFF, RETENTION	OFF	
	OCMC_RAM - OCMC_RAM_Bank 1	software_control	software_control		always_enabled
CORE_OTHER_BANK		ON	RETENTION	OFF	
	sDMA - DMA_MEM	always_on	always_retention		None
	DMM - DMM_MEMBANK1	always_on	always_retention		always_enabled
MPU_M3_L2RAM		ON	OFF, RETENTION	OFF	
	CORTEXM3 - MPU_M3_L2RAM_MEM	always_on	software_control		always_enabled
MPU_M3_UNICACHE		ON	OFF, RETENTION	OFF	
	CORTEXM3 - MPU_M3_Unicode_mem	always_on	software_control		None

#### 3.7.11.1.2 Logic and Memory Area Power Modes Control and Status

[Table 3-276](#) lists the power modes controls for the power domain.

**Table 3-276. PD\_CORE Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in Retention state)	CORE_OCMRAM	PM_CORE_PWRSTCTRL[9] CORE_OCMRAM_RETSTATE	Read/Write
Memory Area - State Control (Logic in Retention state)	CORE_OTHER_BANK	PM_CORE_PWRSTCTRL[8] CORE_OTHER_BANK_RETSTATE	Read only
Memory Area - State Control (Logic in On state)	CORE_NRET_BANK	PM_CORE_PWRSTCTRL[25:24] INTRCONN_NRET_BANK_ONSTATE	Read only
Memory Area - State Control (Logic in On state)	MPU_M3_UNICACHE	PM_CORE_PWRSTCTRL[23:22] MPU_M3_UNICACHE_ONSTATE	Read only
Memory Area - State Control (Logic in On state)	MPU_M3_L2RAM	PM_CORE_PWRSTCTRL[21:20] MPU_M3_L2RAM_ONSTATE	Read only
Memory Area - State Control (Logic in On state)	CORE_OCMRAM	PM_CORE_PWRSTCTRL[19:18] CORE_OCMRAM_ONSTATE	Read/Write
Memory Area - State Control (Logic in On state)	CORE_OTHER_BANK	PM_CORE_PWRSTCTRL[17:16] CORE_OTHER_BANK_ONSTATE	Read only
Memory Area - State Control (Logic in Retention state)	CORE_NRET_BANK	PM_CORE_PWRSTCTRL[12] INTRCONN_NRET_BANK_RETSTATE	Read only
Memory Area - State Control (Logic in Retention state)	MPU_M3_UNICACHE	PM_CORE_PWRSTCTRL[11] MPU_M3_UNICACHE_RETSTATE	Read/Write
Memory Area - State Control (Logic in Retention)	MPU_M3_L2RAM	PM_CORE_PWRSTCTRL[10] MPU_M3_L2RAM_RETSTATE	Read/Write
Logic Area - Retention State Control	N/A	PM_CORE_PWRSTCTRL[2] LOGICRETSTATE	Read/Write
Power Domain - Low Power State Change Control	N/A	PM_CORE_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/Write
Power Domain - State Transition Control	N/A	PM_CORE_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-277 lists the power modes status for the power domain.

**Table 3-277. PD\_CORE Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Power Domain - Last Power State Entered Status		PM_CORE_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area - State Status	MPU_M3_L2RAM	PM_CORE_PWRSTST[9:8] MPU_M3_L2RAM_STATEST
Memory Area - State Status	CORE_OCMRAM	PM_CORE_PWRSTST[7:6] CORE_OCMRAM_STATEST
Memory Area - State Status	CORE_OTHER_BANK	PM_CORE_PWRSTST[5:4] CORE_OTHER_BANK_STATEST
Memory Area - State Status	CORE_NRET_BANK	PM_CORE_PWRSTST[13:12] INTRCONN_NRET_BANK_STATEST
Memory Area - State Status	MPU_M3_UNICACHE	PM_CORE_PWRSTST[11:10] MPU_M3_UNICACHE_STATEST
Logic Area - State Status		PM_CORE_PWRSTST[2] LOGICSTATEST
Power Domain - State Transition Status		PM_CORE_PWRSTST[20] INTRANSITION

**Table 3-277. PD\_CORE Power Modes Status Parameters (continued)**

Parameter Name	Memory Bank	Status Bit Field
Power Domain - State Status		PM_CORE_PWRSTST[1:0] POWERSTATEST

**3.7.12 PD\_CAM Description**

PD\_CAM contains the CAM\_RST reset domain.

PD\_CAM contains the CD\_CAM clock domain.

Table 3-278 lists the logic retention capability for each module of the power domain.

**Table 3-278. PD\_CAM Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
FACE DETECT	No	RM_CAM_FDIF_CONTEXT[0] LOSTCONTEXT_DFF	None
ISS	No	RM_CAM_ISS_CONTEXT[0] LOSTCONTEXT_DFF	None

**3.7.12.1 Power Domain Modes**

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See Section 3.1.1.2.1 for functional description of the power-management architecture of a generic power domain.

**3.7.12.1.1 Logic and Memory Area Power Modes**

Table 3-279 lists the power modes supported by the logic area of the power domain.

**Table 3-279. PD\_CAM Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Not available	Not available	Available	Available

Table 3-280 presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-280. PD\_CAM Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
	FACE DETECT - FDIFMEM	always_on	always_off		always_enabled
	ISS - ISSMEM	always_on	always_off		always_enabled

**3.7.12.1.2 Logic and Memory Area Power Modes Control and Status**

Table 3-281 lists the power modes controls for the power domain.

**Table 3-281. PD\_CAM Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain - Low Power State Change Control		PM_CAM_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/Write
Memory Area - State Control (Logic in On state)	CAM_MEM	PM_CAM_PWRSTCTRL[17:16] ]CAM_MEM_ONSTATE	Read only

**Table 3-281. PD\_CAM Power Modes Control Parameters (continued)**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain - State Transition Control		PM_CAM_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-282 lists the power modes status for the power domain.

**Table 3-282. PD\_CAM Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Memory Area - State Status	CAM_MEM	PM_CAM_PWRSTST[5:4] CAM_MEM_STATEST
Power Domain - State Transition Status		PM_CAM_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_CAM_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_CAM_PWRSTST[1:0] POWERSTATEST

### 3.7.13 PD\_AUDIO Description

PD\_AUDIO contains the following reset domains:

- AUDIO\_PWRON\_RST
- AUDIO\_RST

PD\_AUDIO contains the CD\_ABE clock domain.

Table 3-283 lists the logic retention capability for each module of the power domain.

**Table 3-283. PD\_AUDIO Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
AUDIO ENGINE	No	RM_ABE_AESS_CONTEXT[0] LOSTCONTEXT_DFF	None
CM1_ABE	No	None	None
DMIC	No	RM_ABE_DMIC_CONTEXT[0] LOSTCONTEXT_DFF	None
L4_ABE interconnect	No	None	None
MCASP1	No	RM_ABE_MCASP_CONTEXT[0] LOSTCONTEXT_DFF	None
MCBSP1	No	RM_ABE_MCBSP1_CONTEXT[0] LOSTCONTEXT_DFF	None
MCBSP2	No	RM_ABE_MCBSP2_CONTEXT[0] LOSTCONTEXT_DFF	None
MCBSP3	No	RM_ABE_MCBSP3_CONTEXT[0] LOSTCONTEXT_DFF	None
MCPDM	No	RM_ABE_PDM_CONTEXT[0] LOSTCONTEXT_DFF	None
SLIMBUS1	No	RM_ABE_SLIMBUS_CONTEXT[0] LOSTCONTEXT_DFF	None
GPTIMER5	No	RM_ABE_GPTIMER5_CONTEXT[0] LOSTCONTEXT_DFF	None
GPTIMER6	No	RM_ABE_GPTIMER6_CONTEXT[0] LOSTCONTEXT_DFF	None
GPTIMER7	No	RM_ABE_GPTIMER7_CONTEXT[0] LOSTCONTEXT_DFF	None
GPTIMER8	No	RM_ABE_GPTIMER8_CONTEXT[0] LOSTCONTEXT_DFF	None
WDTIMER3	No	RM_ABE_WDTIMER3_CONTEXT[0] LOSTCONTEXT_DFF	None

### 3.7.13.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See [Section 3.1.1.2.1](#) for a functional description of the power-management architecture of a generic power domain.

#### 3.7.13.1.1 Logic and Memory Area Power Modes

[Table 3-284](#) lists the power modes supported by the logic area of the power domain.

**Table 3-284. PD\_AUDIO Logic Area Power Modes**

Off	Retention-OSWR	Retention-CSWR	On-Inactive	On-Active
Available	Available	Not available	Available	Available

[Table 3-285](#) presents the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column). The RTA Feature column identifies whether the memory supports the RTA feature.

**Table 3-285. PD\_AUDIO Memory Area Power Modes**

Memory Bank	Module - Memory	Logic On	Logic Retention	Logic Off	RTA Feature
AESSMEM		ON	RETENTION	OFF	
	AUDIO ENGINE - AESS_MemBank1	always_on	always_retention		always_enabled
PERIPHEM		ON	OFF	OFF	
	DMIC - DMIC_BANK	always_on	always_off		None
	MCBSP1 - MCBSP_BANK	always_on	always_off		None
	MCPDM - MCPDM_BANK	always_on	always_off		None
	SLIMBUS1 - SLIMBUS_BANK	always_on	always_off		None

#### 3.7.13.1.2 Logic and Memory Area Power Modes Control and Status

[Table 3-286](#) lists the power modes controls for the power domain.

**Table 3-286. PD\_AUDIO Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in Retention state)	AESSMEM	PM_ABE_PWRSTCTRL[8] AESSMEM_RETSTATE	Read only
Memory Area - State Control (Logic in On state)	PERIPHEM	PM_ABE_PWRSTCTRL[21:20] PERIPHEM_ONSTATE	Read only
Memory Area - State Control (Logic in On state)	AESSMEM	PM_ABE_PWRSTCTRL[17:16] AESSMEM_ONSTATE	Read only
Memory Area - State Control (Logic in Retention state)	PERIPHEM	PM_ABE_PWRSTCTRL[10] PERIPHEM_RETSTATE	Read only
Logic Area - Retention State Control		PM_ABE_PWRSTCTRL[2] LOGICRETSTATE	Read only
Power Domain - Low Power State Change Control		PM_ABE_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/Write
Power Domain - State Transition Control		PM_ABE_PWRSTCTRL[1:0] POWERSTATE	Read/Write

Table 3-287 lists the power modes status for the power domain.

**Table 3-287. PD\_AUDIO Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Power Domain - Last Power State Entered Status		PM_ABE_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area - State Status	PERIPHMEM	PM_ABE_PWRSTST[9:8] PERIPHMEM_STATEST
Memory Area - State Status	AESSMEM	PM_ABE_PWRSTST[5:4] AESSMEM_STATEST
Power Domain - State Transition Status		PM_ABE_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_ABE_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_ABE_PWRSTST[1:0] POWERSTATEST

### 3.7.14 PD\_ALWON\_MPU Description

PD\_ALWON\_MPU contains the following reset domains:

- ALWON\_MPU\_RST
- DPLL\_MPU\_PWRON\_RST

PD\_ALWON\_MPU has no associated clock domains.

Table 3-288 lists the logic retention capability for each module of the power domain.

**Table 3-288. PD\_ALWON\_MPU Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
DPLL_MPU	No	None	None
CORTEXA9_MPU_INTC	No	None	None

#### 3.7.14.1 Power Domain Modes

The PD\_ALWON\_MPU power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power state control or status bit field for this power domain.

The PD\_ALWON\_MPU power domain has no memory banks.

### 3.7.15 PD\_ALWON\_DSP Description

PD\_ALWON\_DSP contains the following reset domains:

- ALWON\_IVA\_RST
- DPLL\_IVA\_PWRON\_RST

PD\_ALWON\_DSP has no clock domains associated.

Table 3-289 lists the logic retention capability for each module of the power domain.

**Table 3-289. PD\_ALWON\_DSP Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
DPLL_IVA	No	None	None
DSP_WKUPGEN	No	None	None

#### 3.7.15.1 Power Domain Modes

The PD\_ALWON\_DSP power domain is an always-on power domain and does not switch to retention state. There is no logic power state control or status bit field for this power domain.

The PD\_ALWON\_DSP power domain has no memory banks.

### 3.7.16 PD\_ALWON\_CORE Description

PD\_ALWON\_CORE contains the following reset domains:

- ALWON\_CM1\_PWRON\_RST
- ALWON\_CM1\_RST
- ALWON\_CORE\_PWRON\_RST
- ALWON\_CORE\_RST

PD\_ALWON\_CORE contains the following clock domains:

- CD\_ALWON\_CM
- CD\_L4\_ALWON\_CORE

Table 3-290 lists the logic retention capability for each module of the power domain.

**Table 3-290. PD\_ALWON\_CORE Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
CM1	No	None	None
DPLL_ABE	No	None	None
DPLL_CORE	No	None	None
DPLL_PER	No	None	None
CORTEXM3_WKUPGEN	No	None	None
SDMA_WKUPGEN	No	None	None
SR_CORE	No	<a href="#">RM_ALWON_SR_CORE_CONTEXT[0] LOSTCONTEXT_DFF</a>	None
SR_IVA	No	<a href="#">RM_ALWON_SR_IVA_CONTEXT[0] LOSTCONTEXT_DFF</a>	None
SR_MPU	No	<a href="#">RM_ALWON_SR_MPU_CONTEXT[0] LOSTCONTEXT_DFF</a>	None
SPINNER	No	None	None

#### 3.7.16.1 Power Domain Modes

The PD\_ALWON\_CORE power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power state control or status bit field for this power domain.

The PD\_ALWON\_CORE power domain has no memory banks.

### 3.8 Voltage Management Functional Description

This section describes the voltage domains and voltage control architecture. It also explains the interactions between the device and the external power IC.

#### 3.8.1 Overview

The voltage management architecture of the device is based on voltage sources managed by the PRCM module. They define the voltage domains within the device (see [Section 3.1.1.3, Voltage Management](#)). This partition of the voltage domains ensures independent voltage control of each voltage domain through dedicated SMPS or LDO. The following voltage domains are managed by the PRCM module:

- VDD\_CORE\_L
- VDD\_MPU\_L
- VDD\_IVA\_L
- LDO\_CORE\_M
- LDO\_IVA\_M
- LDO\_MPU\_M
- LDO\_IVA\_ABB
- LDO\_MPU\_ABB
- LDO\_WAKEUP

---

**NOTE:** See [Table 3-24](#) for device power supply pin association to the power domain.

---

The PRCM module supports the AVS technique on the VDD\_MPU\_L, VDD\_CORE\_L and VDD\_IVA\_L voltage domains, through automatic monitoring and adjustments of voltages by their dedicated SmartReflex modules. The VDD\_MPU\_L and VDD\_IVA\_L voltage domains support five DVFS OPPs. The VDD\_CORE\_L voltage domain supports three DVFS OPPs.

The PRCM module also supports ABB technique on the MPU and IVA voltage domains, through the LDO\_MPU\_ABB and LDO\_IVA\_ABB biasing voltages.

The PRCM module automatically scales down the LDO\_MPU\_M, LDO\_CORE\_M, and LDO\_IVA\_M to retention voltage level whenever all memory banks on these domain have transitioned to the RETENTION or OFF power state.

#### 3.8.2 Voltage-Control Architecture

The PRM is split over several blocks that manage the different voltage sources.

- Voltage processors for converting SmartReflex data to voltage values before sending to the voltage controller
- Voltage FSMs for hardware control of voltages. They either send commands to the voltage controller (I<sup>2</sup>C mode).
- A voltage controller for gathering commands from the PRCM module register (direct access), voltage processors and voltage PRCMs. It then handles communication with the external power IC through the dedicated I<sup>2</sup>C interface.
- A device PRCM for managing the voltage PRCMs and I/O wake-up control and system clock control sequencing during device sleep and wake-up transitions.
- LDO regulator controllers for ABB management, memory arrays voltage management and WAKEUP logic.
- Bandgap reference voltage sleep control

---

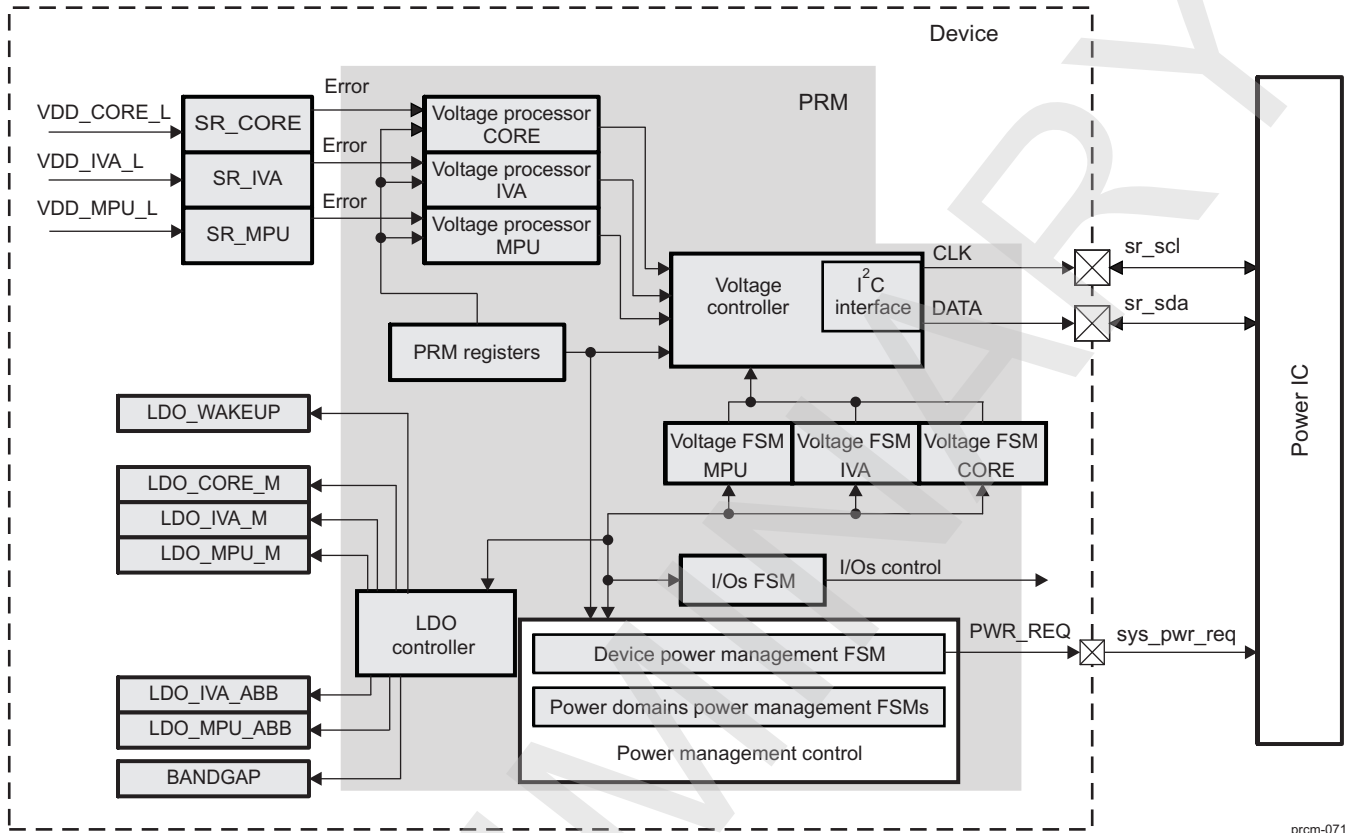
**NOTE:** The SmartReflex module allows dynamic voltage adjustments around an OPP voltage level to ensure target performance. It also allows switching from one OPP to another. However, it does not control voltage switching of the voltage domain to RETENTION or OFF level when all power domains within the power domain are in RETENTION or OFF power states. This is handled by the voltage PRCMs.

---



Figure 3-69 shows the architecture for PRM voltage control.

Figure 3-69. PRM Voltage Control Architecture



prcm-071

### 3.8.3 VDD\_MPU\_L, VDD\_CORE\_L, and VDD\_IVA\_L Control

These power supply sources can be controlled using PRM dedicated I<sup>2</sup>C control.

#### 3.8.3.1 PRM Dedicated I<sup>2</sup>C Control

The PRM I<sup>2</sup>C interface is dedicated to SmartReflex voltage control. It enables the voltage controller to send voltage commands to the external power IC. To reduce latency of voltage changes, the voltage controller is configurable to run in HS I<sup>2</sup>C mode. The PRM I<sup>2</sup>C interface supports multimaster mode.

The voltage controller receives voltage control commands from following input ports:

- Voltage processors ports: Each voltage processor sends voltage change commands to the voltage controller depending on the information received from the associated SmartReflex module.
- Voltage PRCMs ports: Each voltage PRCM sends commands to the voltage controller when the device enters RETENTION or OFF power state, and also when the device wakes up.
- PRM register port: The PRM registers give direct software control over the voltage levels of the external voltage sources.

An arbitration scheme in the voltage controller manages concurrent requests on different ports.

Each internal port has a handshake to indicate when the I<sup>2</sup>C frame resulting from the request on that port has been acknowledged by the external Power IC.

The PRM can also send external voltage source sleep commands that can be used to activate power IC sleep mode, where the voltage regulator maintains output voltage but only a small load is supported. This lets the external power IC reduce its power consumption as well.

**NOTE:** An I<sup>2</sup>C interface not controlled by the PRM can be used to program voltage levels of the external power IC. Because it is not a hardware-controlled interface its use is restricted to move the voltage level from one OPP to another.

In this case, the control software must operate voltage transition when the device activity allows one.

### 3.8.3.2 Adaptive Voltage Scaling

As explained in [Section 3.1.2.4, Adaptive Voltage Scaling](#), SmartReflex is a technology that uses adaptive power supply to achieve the goal of reducing active power consumption. With SmartReflex, the power supply voltage can be adapted to the silicon performance either statically (for example, adapted to the manufacturing process of a given device), or dynamically (for example, adapted to the temperature-induced current performance of the device).

SmartReflex voltage control in the device is based on the dedicated SmartReflex modules and the voltage processors, in addition to the voltage controller with dedicated I<sup>2</sup>C interface, which are shared with the voltage PRCM modules and voltage control registers.

The SmartReflex modules allow a continuous real-time monitoring of voltage supply and device performance to do the following:

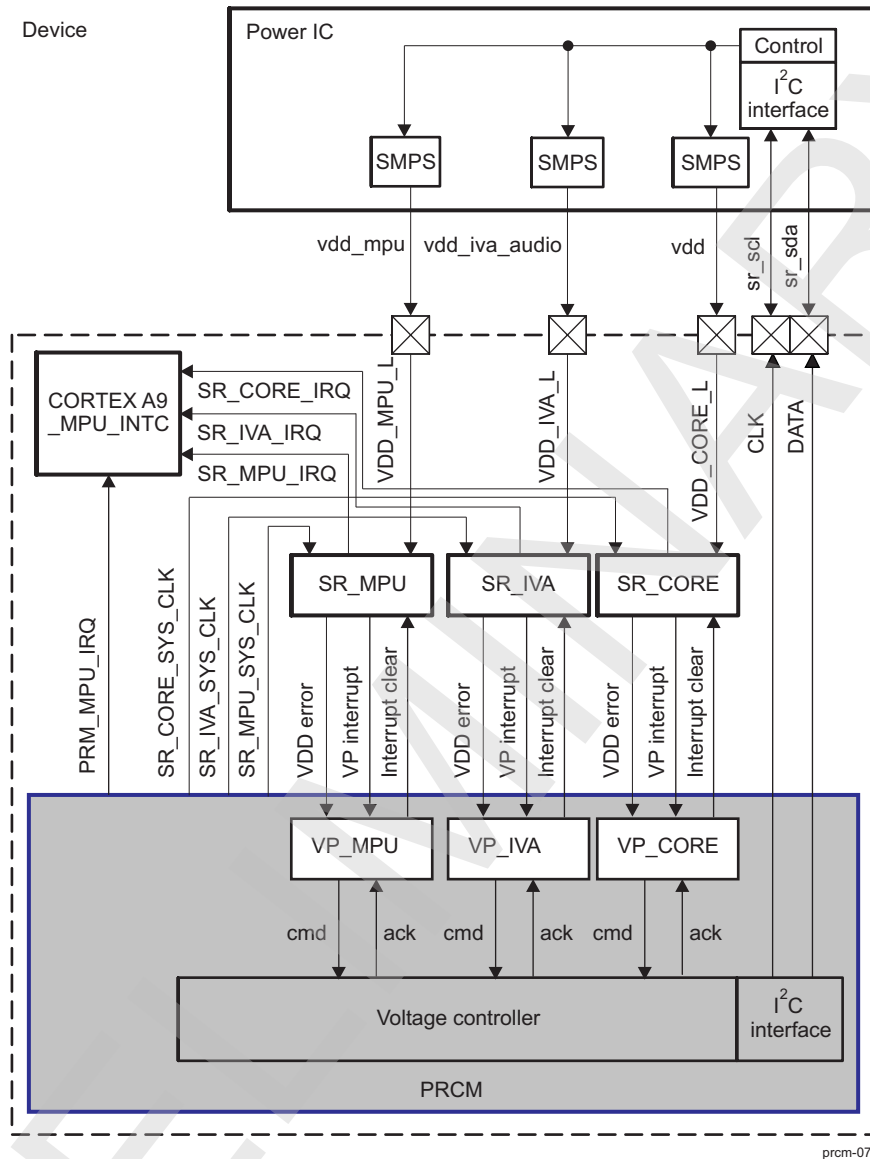
- Minimize the supply voltage to reduce the device power consumption.
- Maintain the desired device performance (by dynamically adjusting the device voltage) as the temperature of the device varies.

Because the SmartReflex modules and the voltage processors are dedicated to SmartReflex voltage control, they are described together in this section.

#### 3.8.3.2.1 SmartReflex in the Device

[Figure 3-70](#) shows the SmartReflex integration.

Figure 3-70. SmartReflex Integration

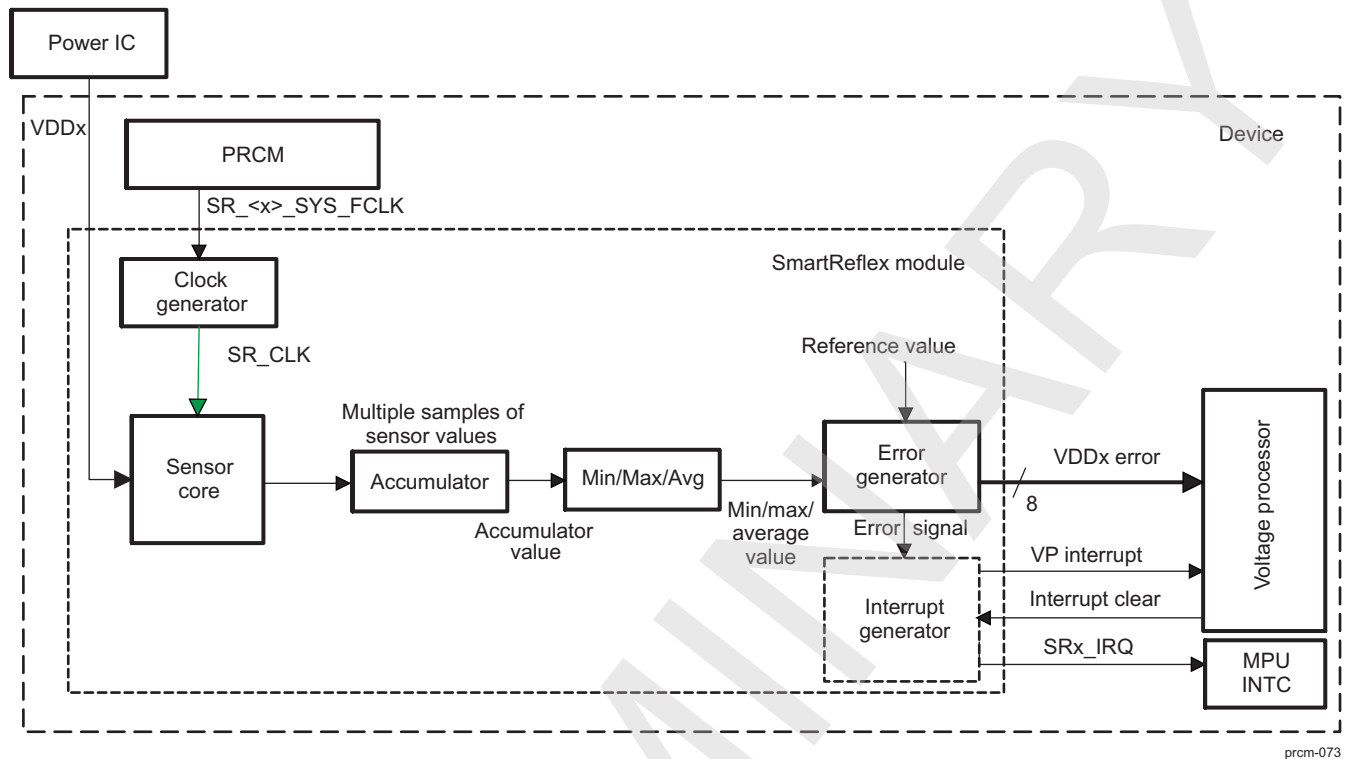


SmartReflex voltage control in the device is implemented for simultaneous control of three independent voltage sources. One SmartReflex module controls the VDD\_CORE\_L voltage, the second one is dedicated to VDD\_IVA\_L control, and the third controls the VDD\_MPU\_L voltage. Each SmartReflex module is connected to a voltage processor. Voltage commands from all voltage processors are passed to a voltage control, which sends them through a dedicated I<sup>2</sup>C master interface to the power IC.

SmartReflex modules are in the PD\_ALWON\_CORE power domain and supplied by the VDD\_CORE\_L voltage domain. Each module senses its own voltage domain (VDD\_CORE\_L, VDD\_IVA\_L, or VDD\_MPU\_L).

### 3.8.3.2.2 SmartReflex Module

Figure 3-71 is a functional overview of the SmartReflex module.

**Figure 3-71. SmartReflex Module Functional Overview**

### 3.8.3.2.3 SmartReflex Submodules

**NOTE:** In this and following section SR<sub>n</sub> refers to SR\_IVA, SR\_MPU, and SR\_CORE SmartReflex modules.

The SmartReflex module is enabled by setting the SR<sub>n</sub>.SRCONFIG[11] SR\_EN bit. The SmartReflex module is composed of six blocks:

- Clock generator
- Sensor core
- Accumulator
- Minimum/maximum/average
- Error generator
- Interrupt generator

#### Clock Generator

The clock generator provides the internal SR\_CLK sampling clock to the sensor core of the module. The SR<sub>n</sub>.SRCONFIG[21:12] SRCLKLENGTH bit field allows the setting of the frequency divider ratio between the <SRn>\_SYS\_CLK and the SR\_CLK. It is calculated using Equation 1:

$$\text{SR}_n.\text{SRCONFIG}[21:12] \text{ SRCLKLENGTH} = f_{\text{<SRn>_SYS\_CLK}} / (2^* f_{\text{SR\_CLK}}) \quad (1)$$

Where  $f_{\text{<SRn>_SYS\_CLK}}$  is the frequency of the <SRn>\_SYS\_CLK, and  $f_{\text{SR\_CLK}}$  is the desired SR\_CLK frequency.

To accurately use the target values programmed for SmartReflex modules in the device, the SRCLKLENGTH parameter must be set correctly. The target values for the SmartReflex modules are calculated with the SR\_CLK frequency set at 100 kHz. It is thus mandatory that the value of the SRCLKLENGTH parameter is calculated from Equation 1-1 with  $f_{\text{SR\_CLK}}$  set at 100 kHz.

For example, if the system clock has a frequency of 38.4 MHz, and the target SR\_CLK frequency is 100 kHz, the SRn.SRCONFIG[21:12] SRCLKLENGTH bit field is 192 (0x0C0).

### **Sensor Core**

The sensor core receives the SR\_CLK sampling clock and the voltage to be sensed. It generates sensor value samples proportional to the voltage sensed. For accuracy, a sensor core is composed of two sensors and generates two values per sample (one from each sensor).

The two sample values generated by the sensor core output can be read from the SRn.SENVAL[15:0] SENNVAL and SRn.SENVAL[31:16] SENPVAL bit fields.

The sensor core is enabled by the SRn.SRCONFIG[10] SENENABLE bit.

### **Accumulator**

The accumulator consists of two stacks that store multiple samples of the two sensor values received from the sensor core.

The SRn.SRCONFIG[31:22] ACCUMDATA bit field defines the size of the accumulator in the number of samples to be stored. The allowable range is from 2 to 1023. The value of SRn.SRCONFIG[31:22] ACCUMDATA value is related to the desired sampling time window ( $T_{\text{TimeWindow}}$ ) and the SR\_CLK frequency ( $f_{\text{SR\_CLK}}$ ). It can be calculated using [Equation 2](#):

$$\text{SRn.SRCONFIG[31:22] ACCUMDATA} = T_{\text{TimeWindow}} * f_{\text{SR\_CLK}} \quad (2)$$

For example, for an accumulator time window of 10 ms and SR\_CLK frequency of 32 kHz, the value of SRn.SRCONFIG[31:22] ACCUMDATA is 320 (0x140). The accumulation window must be large enough so that the minimum, maximum, and average counter values are accurate.

### **Minimum/Maximum/Average**

The minimum/maximum/average block reads the samples stored in the accumulator and returns the minimum, maximum, and average values of the samples. Because the accumulator contains two separate groups of samples (one from each sensor of the sensor core), the minimum/maximum/average block also generates two sets of minimum, maximum, and average values.

The minimum, maximum, and average values of the samples of the first sensor can be read from the SRn.SENMIN[15:0] SENNMIN, SRn.SENMAX[15:0] SENNMAX, and SRn.SENAVG[15:0] SENNAVG bit fields.

For the sample values of the second sensor, the minimum, maximum, and average values can be read from the SRn.SENMIN[31:16] SENPMIN, SRn.SENMAX[31:16] SENPMAX, and SRn.SENAVG[31:16] SENPAVG bit fields.

The minimum/maximum/average block is enabled by the SRn.SRCONFIG[8] MINMAXAVGENABLE bit.

### **Error Generator**

The error generator block reads the sample value generated by the sensor core and compares it with a reference value to calculate the error. This error is then passed to the voltage processor and the internal interrupt generator block.

The reference value for a given OPP of the device is configured by setting the SRn.NVALUERECIPROCAL[23:20] SENPGAIN, SRn.NVALUERECIPROCAL[19:16] SENNGAIN, SRn.NVALUERECIPROCAL[15:8] RSENPN, and SRn.NVALUERECIPROCAL[7:0] RSENENN bit fields by reading corresponding values from the eFuses (see [Section 3.8.3.2.5, SmartReflex Parameters Set After Silicon Characterization](#)).

The error generator sets the SRn.SRSTATUS[1] ERRGEN\_VALID status bit when a valid error value is set in the SRn.SENERROR register.

The SRn.SENERROR register contains the average error (SRn.SENERROR[15:8] AVGERROR) and the percentage of error (SRn.SENERROR[7:0] SENERROR).

The error generator block is enabled by the SRn.SRCONFIG[9] ERRORGENERATORENABLE bit.

### **Interrupt Generator**

The interrupt generator block generates interrupts to the MPU INTC and the voltage processor module (if the corresponding interrupts are enabled) to indicate errors.

Table 3-291 and Table 3-292 list the interrupt events in the SmartReflex module and their enable and status bits.

**Table 3-291. SmartReflex Interrupt Events**

Interrupt Type	Description
Accumulator	The minimum/maximum/average module has completed computation over the accumulator data.
Valid	The average error is less than 2 percent of the true average error.
Disable acknowledge	The SmartReflex module is disabled and has cleared all MPU INTC and VP interrupts (internal registers are reset). This interrupt indicates to the software that the SmartReflex module is available for programming.
Bounds	The frequency error has crossed the maximum limit (ERRMAXLIMIT) or the minimum limit (ERRMINLIMIT). It is mapped to the voltage processor also.

The SRn\_CORTEX\_M3\_IRQs from SR\_IVA and SR\_CORE are mapped to the MM\_IRQ\_47 interrupt line of the CORTEXM3\_MPU\_INTC.

**Table 3-292. SmartReflex Interrupt Events Enable and Status Bits**

Interrupt Type	Enable Bit	Status Bit
Accumulator	SRn.IRQENABLE_SET[3] MCUACCUMINTENSET	SRn.IRQSTATUS[3] MCUACCUMINTSTATENA
	SRn.IRQENABLE_CLR[3] MCUACCUMINTENCLR	
Valid	SRn.IRQENABLE_SET[2] MCUVALIDINTENSET	SRn.IRQSTATUS[2] MCUVALIDINTSTATENA
	SRn.IRQENABLE_CLR[2] MCUVALIDINTENCLR	
Disable acknowledge	SRn.IRQENABLE_SET[0] MCUDISABLEACKINTENSET	SRn.IRQSTATUS[0] MCUDISABLEACKINTSTATENA
	SRn.IRQENABLE_CLR[0] MCUDISABLEACKINTENCLR	
Bounds	SRn.IRQENABLE_SET[1] MCUBOUNDSINTENSET	SRn.IRQSTATUS[1] MCUBOUNDSINTSTATENA
	SRn.IRQENABLE_CLR[1] MCUBOUNDSINTENCLR	
	SRn.ERRCONFIG[22] VPBOUNDSINTENABLE	SRn.ERRCONFIG[23] VPBOUNDSINTSTATENA

The minimum and maximum error limits for the bounds interrupt are configured in the SRn.ERRCONFIG[15:8] ERRMAXLIMIT and SRn.ERRCONFIG[7:0] ERRMINLIMIT bit fields.

With automatic (also called hardware or Class-3) voltage control, the voltage processor interrupt (SRn.ERRCONFIG[31] VPBOUNDINTEN) is enabled and the SmartReflex module interrupts the voltage processor when the error crosses the error limits. In this case, the CORTEXM3 interrupts can also be enabled to allow software to monitor the behavior of the SmartReflex module.

With manual (also called software or Class-2) voltage control, the voltage processor interrupt remains disabled and only the CORTEXM3 interrupts are enabled. Thus, when an interrupt condition occurs, the CORTEXM3 is interrupted and software can then control the voltage by generating voltage commands.

Writing 1 to a bit in the IRQSTATUS register clears the interrupt pending status of each interrupt source whose corresponding bit is 1 in the value written. Other interrupt pending status bits are not affected (writing 0 to that bit does not affect the status).

### 3.8.3.2.4 Status Register

The status register (**SRSTATUS**) indicates the validity of the minimum/maximum/average and error generator output values.

**SRSTATUS** has the following bits and bit fields:

- **SRn.SRSTATUS[3] AVGERVALID**: Indicates the validity of the value in the **SRn.SENERROR\_REG[15:8] AVGERROR** bit field. When the value is 0, the average error is not valid. When the value is 1, the average error is within 2 percent of the valid average error.
- **SRn.SRSTATUS[2] MINMAXAVGVALID**: Indicates the validity of the **SRn.SENVAL**, **SRn.SENMIN**, **SRn.SENMAX**, and **SRn.SENAVG** registers. When the value is 0, the registers are not valid. When the value is 1, the registers contain valid values, although the values are not necessarily fully accumulated.
- **SRn.SRSTATUS[1] ERRGEN\_VALID**: Indicates the validity of the value in the **SRn.SENERROR\_REG[7:0] SENERROR** bit field. When the value is 0, the error value is invalid. When the value is 1, the error value is valid.
- **SRn.SRSTATUS[0] MINMAXAVGACCUMVALID**: Indicates that the **SRn.SENVAL**, **SRn.SENMIN**, **SRn.SENMAX**, and **SRn.SENAVG** registers contain their final values accumulated over the defined sample time period.

### 3.8.3.2.5 SmartReflex Parameters Set After Silicon Characterization

Certain parameters of the SmartReflex module are characterized and calibrated after silicon testing. These values are then either set in the eFuse farm of the device or provided separately. Users must configure these parameters according to their given values to ensure correct functioning of the module.

The values for the following parameters (explained in [Section 3.8.3.2.3, SmartReflex Submodules](#)) are set in the device eFuse after device silicon characterization:

- **SRn.SRCONFIG[6:5] SENNENABLE**
- **SRn.SRCONFIG[4:3] SENPENABLE**
- **SRn.NVALUERECPROCAL[23:20] SENPGAIN**
- **SRn.NVALUERECPROCAL[19:16] SENNGAIN**
- **SRn.NVALUERECPROCAL[15:8] RNSENP**
- **SRn.NVALUERECPROCAL[7:0] RNSENN**

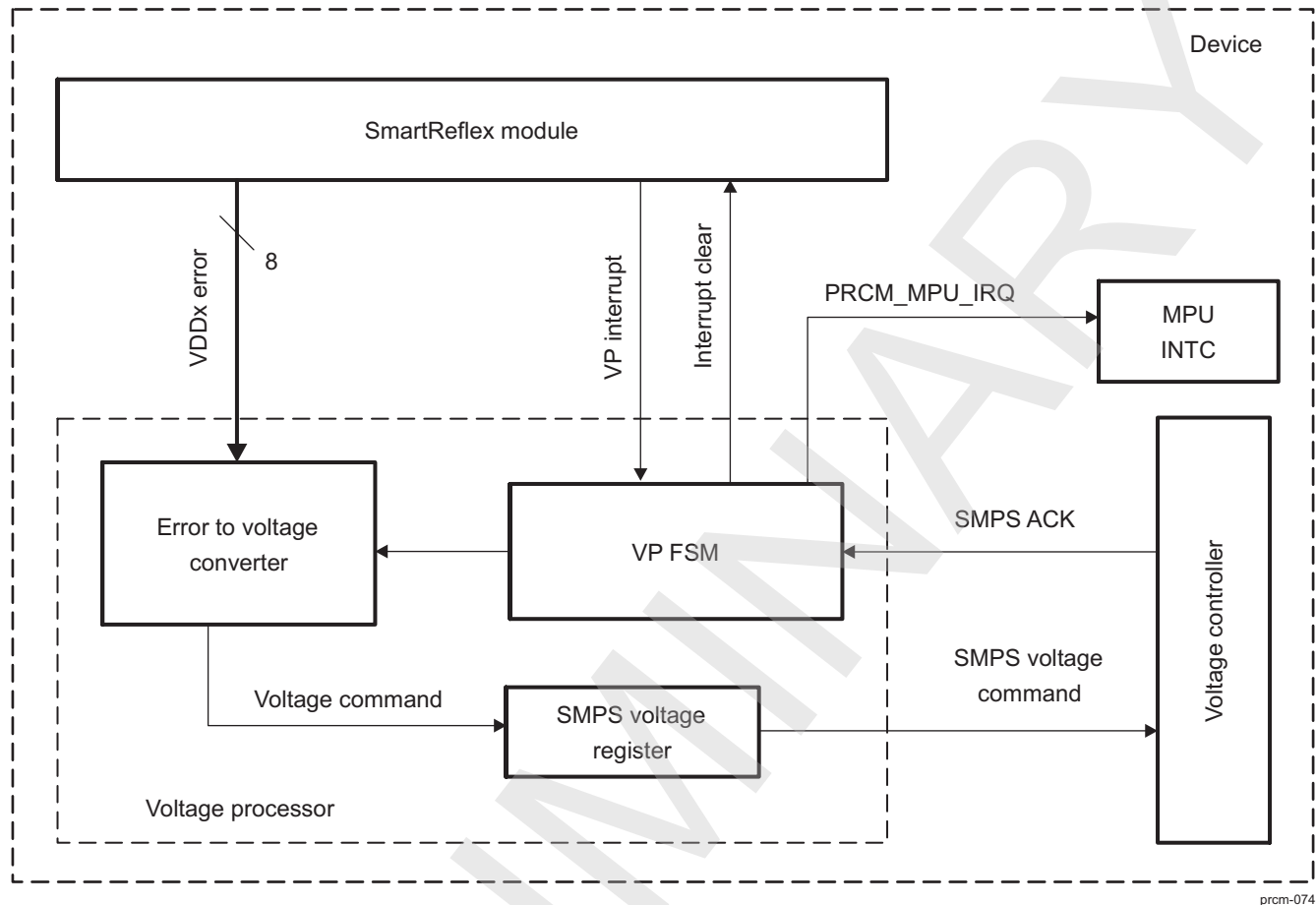
The eFuse values of these parameters must be read from the corresponding registers of the control module.

Information about the following parameters of the SmartReflex module is provided after silicon characterization:

- **SRn.AVGWEIGHT[3:2] SENPAVGWEIGHT**
- **SRn.AVGWEIGHT[1:0] SENNAVGWEIGHT**
- **SRn.ERRCONFIG[18:16] ERRWEIGHT**
- **SRn.ERRCONFIG[15:8] ERRMAXLIMIT**
- **SRn.ERRCONFIG[7:0] ERRMINLIMIT**

### 3.8.3.2.6 Voltage Processor Module

[Figure 3-72](#) is a functional overview of the voltage processor.

**Figure 3-72. Voltage Processor Functional Overview**

The voltage processor receives an error value and a VP interrupt signal from the SmartReflex module. Each time a new error value is sent, the SmartReflex module triggers an interrupt to inform the voltage processor. The voltage processor consists of an error-to-voltage command converter and a state controller. It processes an error and sends a voltage command to the voltage controller. It receives an acknowledge signal from the voltage controller when the SMPS receives the command. The voltage processor then acknowledges the SmartReflex module by clearing its interrupt.

The voltage processor is enabled by the PRM\_VPn\_CONFIG[0] VPENABLE bit.

### **Voltage Processor Interrupts**

The voltage processor uses the PRCM\_CORTEXM3\_MPU\_IRQ (MM\_IRQ\_46 of CORTEXM3\_MPU\_INTC) interrupt line of the CORTEXM3. [Table 3-293](#) and [Table 3-294](#) list the interrupt events in the voltage processor module and their enable and status bits.

**Table 3-293. Voltage Processor Interrupt Events**

Interrupt Type	Description
Transaction done	Voltage processor transaction is complete.
Equal value	Voltage requested in the new voltage command is the same as the current SMPS voltage.
No SMPS acknowledge	SMPS has not responded in a defined time interval to the transmitted voltage command.
Maximum VDD	New voltage requested in the voltage command is equal to or greater than maximum VDD.
Minimum VDD	New voltage requested in the voltage command is equal to or less than minimum VDD.
OPP change done	The average error is within the desired limit.



**NOTE:** Transaction done and OPP change done interrupt events are generated under the same conditions (after voltage change).

**Table 3-294. Voltage Processor Interrupt Event Enable and Status Bits**

Interrupt Type	Enable Bit	Status Bit
Transaction done	PRM_IRQENABLE_MPU_M3[21] VP_CORE_TRANXDONE_EN	PRM_IRQSTATUS_MPU_M3[21] VP_CORE_TRANXDONE_ST
	PRM_IRQENABLE_MPU_M3[29] VP_IVA_TRANXDONE_EN	PRM_IRQSTATUS_MPU_M3[29] VP_IVA_TRANXDONE_ST
Equal value	PRM_IRQENABLE_MPU_M3[20] VP_CORE_EQVALUE_EN	PRM_IRQSTATUS_MPU_M3[20] VP_CORE_EQVALUE_ST
	PRM_IRQENABLE_MPU_M3[28] VP_CORE_EQVALUE_EN	PRM_IRQSTATUS_MPU_M3[28] VP_IVA_EQVALUE_ST
No SMPS acknowledge	PRM_IRQENABLE_MPU_M3[19] VP_CORE_NOSMPSACK_EN	PRM_IRQSTATUS_MPU_M3[19] VP_CORE_NOSMPSACK_ST
	PRM_IRQENABLE_MPU_M3[27] VP_IVA_NOSMPSACK_EN	PRM_IRQSTATUS_MPU_M3[27] VP_IVA_NOSMPSACK_ST
Maximum VDD	PRM_IRQENABLE_MPU_M3[18] VP_CORE_MAXVDD_EN	PRM_IRQSTATUS_MPU_M3[18] VP_CORE_MAXVDD_ST
	PRM_IRQENABLE_MPU_M3[26] VP_IVA_MAXVDD_EN	PRM_IRQSTATUS_MPU_M3[26] VP_IVA_MAXVDD_ST
Minimum VDD	PRM_IRQENABLE_MPU_M3[17] VP_CORE_MINVDD_EN	PRM_IRQSTATUS_MPU_M3[17] VP_CORE_MINVDD_ST
	PRM_IRQENABLE_MPU_M3[25] VP_IVA_MINVDD_EN	PRM_IRQSTATUS_MPU_M3[25] VP_IVA_MINVDD_ST
OPP change done	PRM_IRQENABLE_MPU_M3[16] VP_CORE_OPPCHANGEDONE_EN	PRM_IRQSTATUS_MPU_M3[16] VP_CORE_OPPCHANGEDONE_ST
	PRM_IRQENABLE_MPU_M3 [24] VP_IVA_OPPCHANGEDONE_EN	PRM_IRQSTATUS_MPU_M3[24] VP_IVA_OPPCHANGEDONE_ST

### Voltage Processor Status

The status of the voltage processor is represented by the following:

Bit/Bit Field	Parameter	Description
PRCM.module.PRM_VPn_VOLTAGE[7:0] VPVOLTAGE	Current voltage for SMPS	Value of the current voltage level requested by the voltage processor
PRCM.module.PRM_VPn_STATUS[0] VPINIDLE	Voltage processor in INACTIVE state	The voltage processor is in idle mode.

### Voltage Processor Parameters

The following parameters of the voltage processor must be configured:

Bit/Bit Field	Parameter	Description
PRCM.PRM_VPn_CONFIG[31:24] ERROROFFSET	Error offset	Error in voltage offset value
PRCM.PRM_VPn_CONFIG[231:16] ERRORGAIN	Error gain	Error in voltage gain value
PRCM.PRM_VPn_CONFIG[3] TIMEOUTEN	Time-out enable	Enables/disables the time-out set by the time-out delay parameter
PRCM.PRM_VPn_VSTEPMIN[7:0] VSTEPMIN	Minimum step size	Minimum voltage step size
PRCM.PRM_VPn_VSTEPMAX[7:0] VSTEPMAX	Maximum step size	Maximum voltage step size
PRCM.PRM_VPn_VLIMITTO[31:24] VDDMAX	Maximum VDD limit	Maximum voltage limit of the VDD
PRCM.PRM_VPn_VLIMITTO[23:16] VDDMIN	Minimum VDD limit	Minimum voltage limit of the VDD
PRCM.PRM_VPn_VLIMITTO[15:0] TIMEOUT	Time-out delay	Maximum delay between a voltage change command and its acknowledge
PRCM.PRM_VPn_CONFIG[15:8] INITVDD	Initial VDD voltage	Initial voltage set in the voltage processor

Bit/Bit Field	Parameter	Description
PRCM.PRM_VPn_CONFIG[2] INITVDD	Initialize VDD	Set initial voltage given in initial VDD voltage parameter in the voltage processor
PRCM.PRM_VPn_VSTEPMIN[23:8] SMPSWAITTIMEMIN	Minimum SMPS wait time	Slew rate for negative voltage steps
PRCM.PRM_VPn_VSTEPMAX[28:8] SMPSWAITTIMEMAX	Maximum SMPS wait time	Slew rate for positive voltage steps
PRCM.PRM_VPn_CONFIG[1] FORCEUPDATE	Force update	Sends the value of the VP current voltage parameter as the new voltage command to SMPS

### 3.8.3.2.7 SMPS-Dependent Parameter Configuration

The values of the following parameters of the voltage processor module depend on the characteristics of the SMPS used:

- PRCM.PRM\_VPn\_CONFIG[31:24] ERROROFFSET
- PRCM.PRM\_VPn\_CONFIG[23:16] ERRORGAIN
- PRCM.PRM\_VPn\_VSTEPMIN[7:0] VSTEPMIN
- PRCM.PRM\_VPn\_VSTEPMAX[7:0] VSTEPMAX
- PRCM.PRM\_VPn\_VLIMITTO[31:24] VDDMAX
- PRCM.PRM\_VPn\_VLIMITTO[23:16] VDDMIN
- PRCM.PRM\_VPn\_VLIMITTO[15:0] TIMEOUT
- PRCM.PRM\_VPn\_VSTEPMIN[23:8] SMPSWAITTIMEMIN
- PRCM.PRM\_VPn\_VSTEPMAX[28:8] SMPSWAITTIMEMAX

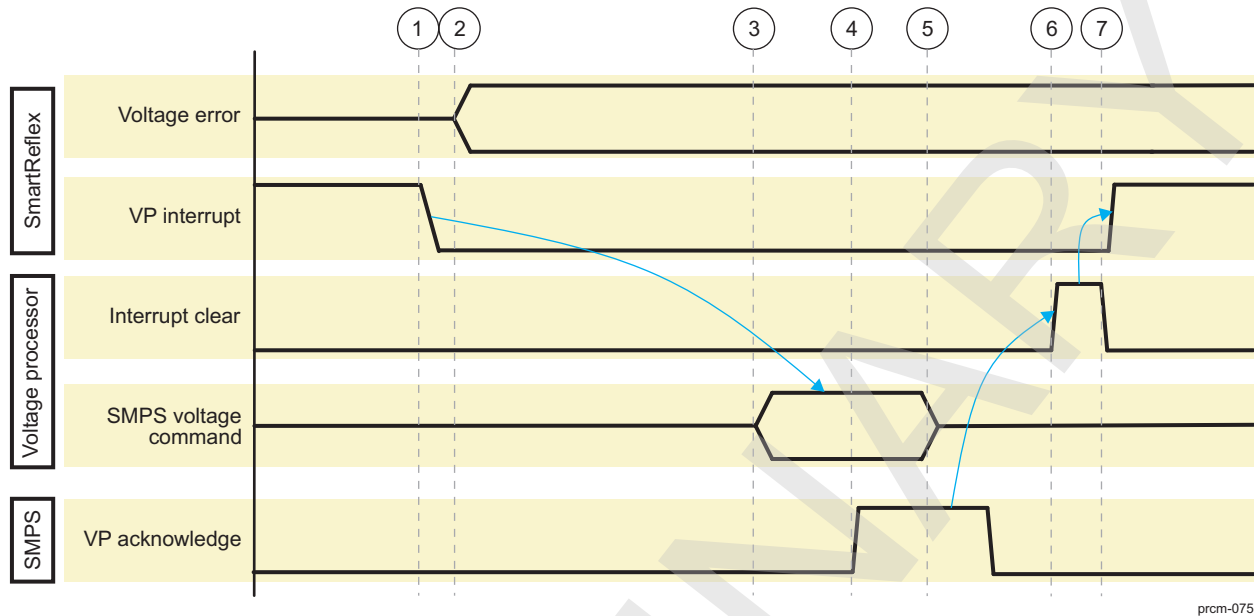
For information about the values of these parameters, see the specification document of the SMPS used.

### 3.8.3.2.8 Communication Between SmartReflex, Voltage Processor, Voltage Controller, and SMPS

The SmartReflex module, voltage processor, voltage controller, and SMPS implement a simple handshake protocol based on a request/acknowledge mechanism (see [Figure 3-73](#)):

1. The SmartReflex module generates a VP interrupt (clears to 0) when the average error crosses the minimum or maximum error bounds.
2. The average VDDx error is passed to the voltage processor when automatic voltage control is enabled.
3. The voltage processor reads the error and generates the voltage command for the SMPS. It then informs the voltage controller that a new voltage command is ready for the SMPS. The voltage controller sends the voltage command to the SMPS (external power IC) through the dedicated I<sup>2</sup>C interface.
4. The SMPS responds to the reception of the voltage command with an acknowledgement to the voltage controller.
5. The voltage controller sends the acknowledgement to the voltage processor.
6. The voltage processor sends the interrupt-clear signal to the SmartReflex module.
7. The SmartReflex module clears the VP interrupt. It is now again ready to send a new frequency-error interrupt to the voltage processor.

Figure 3-73. SmartReflex – SMPS Communication for Automatic Voltage Adjustments



prcm-075

### 3.8.4 Internal LDOs Control

#### 3.8.4.1 Memory LDOs

Embedded SRAM LDOs (that is, LDO\_CORE\_M, LDO\_IVA\_M and LDO\_MPU\_M) are used to supply power to the split-rail memory arrays. The PRM generates the controls used to select LDO operating mode: on-active, on-retention, or off.

Split-rail type SRAMs are used in the device to implement the larger memories. These SRAMs feature memory array and periphery logic which are on separate supplies to allow independent power management. Proper memory operation, however, requires the SRAM array voltage to never be operated at a level lower than the SRAM periphery logic.

Memory LDO can switch to on, retention and off mode:

- On mode: 1.2 V is the normal voltage reference used through all functional OPPs, whenever memories must be functional. When logic voltage level VDD\_x\_L (where x can be MPU, CORE or IVA) becomes higher than the associated memory voltage level VDD\_x\_M, the LDO operates in the tracking mode and follows its respective the VDD\_x\_L voltage level.
- Retention mode: 0.6 V is set when software allows and when all memories banks belonging to the LDO memory voltage are in Retention or Off state. In this mode, the output voltage is generated from the corresponding VDD\_x\_L logic voltage source.
- Off mode: 0 V is set when the device switches to off mode.

Section 3.8.5.1, *Memory LDOs Transitions*, presents the state transition conditions and sequences for the memory LDOs.

**NOTE:** The voltage levels associated with the different modes may depend on the device characteristics.

#### 3.8.4.2 LDO\_WAKEUP

LDO\_WAKEUP supports three voltage states depending on the PD\_WKUP and PD\_EMU activity:

- EMULATION overdrive: 1.2 V is voltage level and the LDO is set in bypass mode.
- ON: 1.06 V is the voltage reference (default state).

- SLEEP: 0.83 V is the voltage reference. It is set when the device enters sleep mode.

[Section 3.8.5.3, LDO\\_WAKEUP Transitions](#), presents the state transition conditions and sequences for the LDO\_WKUP.

---

**NOTE:** The voltage levels associated with the different modes may depend on the device characteristics.

---

At boot time, the LDO\_WAKEUP LDO is started in emulation mode. It is a software command (that is, from OS or ROM code) to set it to a different mode.

### 3.8.4.3 ABB LDOs Control

The ABB LDO supports three voltage modes:

- Bypass mode: In this mode the x\_ABB LDO is bypassed and outputs the VDD\_x\_L voltages (x refers to MPU and IVA). This mode is activated when FBB or RBB are not required, or when voltage domain enters low-power mode.
- RBB mode is enabled when the device is a strong process device at lowest OPP.
- FBB mode is enabled when the device is a weak process device at highest OPP.

The information regarding the type of device silicon process (weak, nominal, strong) is stored in eFuse. The eFuse bits values are copied to the PRCM module registers during power-on sequence.

The PRCM module provides [PRM\\_LDO\\_ABB\\_MPU\\_SETUP](#) and [PRM\\_LDO\\_ABB\\_IVA\\_SETUP](#) registers for configuration with the following controls:

- SR2EN – To enable or bypass the ABB power management
- ACTIVE\_RBB\_SEL – To enable or bypass RBB mode
- ACTIVE\_FBB\_SEL – To enable or bypass FBB mode
- SLEEP\_RBB\_SEL – To enable or bypass RBB mode when the voltage domain is lowered to sleep level
- SR2\_WTCNT\_VALUE – LDO settling delay on OPP change. The delay is in number of system clock cycles.

The PRCM module provides [PRM\\_LDO\\_ABB\\_MPU\\_CTRL](#) and [PRM\\_LDO\\_ABB\\_IVA\\_CTRL](#) register for control:

- OPP\_SEL – Current operational OPP
- OPP\_CHANGE – Initiate an OPP based ABB LDO setting change
- SR2\_STATUS – Current mode of operation of ABB LDO
- SR2\_IN\_TRANSITION – ABB LDO is in transition.

For the ABB LDO state change programming sequence, see [Section 3.10.4.2, Changing OPP](#).

### 3.8.4.4 Bandgap control

Bandgap provides voltage reference for internal LDOs. The PRCM module automatically controls the switching between ON and OFF states of the bandgap, based on the device power state. It is completely transparent to user software.

The [PRM\\_LDO\\_BANDGAP\\_CTRL\[0\] OFF\\_ENABLE](#) bit controls whether the bandgap remains on or is shut off when the device switches to OFF power state.

## 3.8.5 Voltage Domain State Transitions

In [Section 3.1.1.3.1, Voltage Domain](#), the supported states of a voltage domain were presented. This section describes the trigger conditions to enable and perform sleep and wake-up transitions on PRCM module-controlled voltage domains:

- VDD\_MPU\_L
- VDD\_IVA\_L
- VDD\_CORE\_L

- LDO\_MPU\_M
- LDO\_IVA\_M
- LDO\_CORE\_M

It also identifies state transition sequences for these voltage domains.

When the voltage domain state transition control is set to automatic, the PRCM module initiates the voltage domain state transition when the required hardware conditions are satisfied. These hardware conditions can be based on the power domain state conditions and/or system clock state conditions. The PRCM module prevents any transition when the associated conditions are not satisfied.

The LDO\_MPU\_M, LDO\_IVA\_M, or LDO\_CORE\_M memory voltage domain is automatically transitioned to the retention level whenever all memory arrays in the domain reach the RETENTION or OFF power state. Similarly, a transition back to the ON state is triggered upon one or more domain memory arrays requiring a transition back to the ON power state.

If a voltage command acknowledgement is not received within an allowed response time window (communications error over the I<sup>2</sup>C interface), an error status is logged, a global warm reset is generated, and the transition is skipped.

### 3.8.5.1 VDD\_x\_L Transitions

Transition trigger conditions for sleep transition from ON to SLEEP or RETENTION state are:

- Software setting: `PRM_VOLTCTRL[a:b] AUTO_CTRL_VDD_x_L = SLEEP (0x1) or RETENTION (0x2)`
- Power states conditions:
  - For voltage domain SLEEP state transition – all power domains in the voltage domain must be in ON-INACTIVE, RETENTION, or OFF state.
  - For voltage domain Retention state transition – all power domains in the voltage domain must be in RETENTION or OFF state.
- Clock states conditions: No clock activity on voltage domain

---

**NOTE:**

- In case of VDD\_CORE\_L sleep transition from ON to SLEEP state, the PD\_CORE can be either ON-INACTIVE or in RETENTION only.
  - In case of VDD\_MPU\_L and VDD\_IVAL sleep transition from ON to SLEEP state, the reference and bypass clocks to their respective DPLLs must also be gated.
- 

Transition trigger conditions for wake-up transition from SLEEP or RETENTION state to ON state is:

- Any wake-up event occurs on the voltage domain.

The State transition sequence:

1. All transition trigger conditions are met.
2. The voltage controller sends an On, SLEEP, or Retention command sequence over the I<sup>2</sup>C interface.
3. The PRCM module waits for I<sup>2</sup>C command sequence acknowledge.
4. The PRCM module starts the voltage transition counter.
5. When the counter expires, a voltage transition completion state is reached.

### 3.8.5.2 Memory LDOs Transitions

The transition trigger conditions for on to retention mode are:

- At least one memory bank associated with the LDO is in RETENTION state and the rest of the memory banks may either be in RETENTION or OFF states.

The transition trigger conditions for on/retention mode to off mode transition:

- Device is switching to OFF state.

Transition sequence:

1. Transition trigger conditions are true (that is, satisfied).

2. Switch LDO reference to:
  - Retention mode value if on-to-retention mode transition
  - Off mode value if on/retention-to-off mode transition
3. Start counter for LDO stabilization.
4. When counter expires voltage transition completion state is reached.

### 3.8.5.3 LDO\_WAKEUP Transitions

The ON-to-SLEEP transition trigger conditions are:

- Power domain states: Device is in low-power state (STANDBY or OFF state) and PD\_EMU power domain is off.
- Clock states: Only the 32K clock is active in CD\_WKUP (and system clock is gated at SCRM level)

The SLEEP-to-ON transition trigger conditions are:

- Any device wake-up event other than emulation wake-up event

The EMULATION-to-ON transition trigger conditions are:

- Power domain states: PD\_EMU is Off and device is in active mode OR
- Clock states: PD\_EMU is Off and any clock other than 32K clock is active in PD\_WKUP

The EMULATION-to-SLEEP transition trigger conditions are:

- Power domain states: Device is in low-power state (STANDBY or OFF state) and PD\_EMU power domain is off.
- Clock states: Only the 32K clock is active in CD\_WKUP (and system clock is gated at SCRM level)

The SLEEP or ON-to-EMULATION transition trigger condition is:

- Any wake-up event on PD\_EMU

Transition sequence:

1. Transition trigger conditions are met.
2. LDO\_WAKEUP initiates the transition.
3. Starts a counter for LDO\_WAKEUP transition.
4. LDO\_WAKEUP transition completes.

### 3.8.6 DVFS

Dynamic voltage and frequency scaling is a technique that can be used on the logic voltage domains (VDD\_MPU\_L, VDD\_IVA-L, and VDD\_CORE\_L), independently of each other. Upon current or predictive performance request, determined by software according to ad hoc algorithms or heuristics, software can configure the PRCM module to change the OPP of a voltage domain. For an increase of performance, voltage is first raised, and then frequency is increased. For a decrease of performance, frequency is first decreased, and then voltage is dropped.

#### 3.8.6.1 EMIF Clocks Frequency Scaling Constraints

CD\_EMIF is inside the PD\_CORE. The EMIF subsystem is getting several clocks:

- L3 interface clock
- DLL\_CLK clock (that is, DLL clock): For the DLL only
- PHY\_ROOT\_CLK clock (that is, DDR PHY clock): For external memory access and source of functional clock to EMIF

L3 interface clock frequency can be changed on the fly without affecting EMIF operations as long as DLL\_CLK and PHY\_ROOT\_CLK clocks are not affected. EMIF has internal asynchronous bridge to handle different interface and functional clocks.

When DDR PHY clock frequency is to be changed, EMIF must be put into IDLE state. This ensures that external memory is in self-refresh mode.

Similarly, DLL frequency change implies that DLL must relock. It can be done only when EMIF is in IDLE state. Relock time is 400 DLL clock cycles.

The EMIF clocking scheme allows for change of OPP without DLL relock (that is, DLL is clocked by a relatively low frequency supported in all OPPs). Then change of OPP consists only in changing the DDR PHY clock frequency (and eventually L3 interface clock frequency).

---

**NOTE:** Software must ensure that no master module tries to access the registers of the EMIF during this sequence.

---

EMIF clocks control requirements in the PRCM module:

- EMIF clock frequency setting should be done through shadow registers for DLL\_CLK and DDR\_PHY clocks.
- EMIF clock frequency update bit field should be used to copy shadow register value as current values.
- The PRCM module hardware manages automatic EMIF clock frequency update sequence:
  - Put the EMIF into IDLE state.
  - Upload shadow register values in corresponding current register and wait for new EMIF clocks to be stable.
  - Put the EMIF module back into ACTIVE state

---

**NOTE:** The PRCM module should not gate the DLL\_CLK and DDR\_PHY clocks during automatic EMIF frequency change.

---

Programming model to perform automatic EMIF frequency change as part or not of an OPP change (DVFS). The automatic EMIF frequency change must always be performed while CD\_EMIF is under software-forced wakeup (SW\_WKUP).

#### ***VDD\_CORE\_L voltage fixed and only clock frequency switching***

Changing EMIF clock frequency without voltage change (example, change of SYS\_CLK divider value while DPLL\_CORE is in bypass, change of DPLL\_CORE mode between bypass and lock)

- Software writes new EMIF timing parameters in shadow registers of the EMIF.
- It then configures the clock selection and frequency setting shadow registers in the PRCM module.
- The PRCM module sends an idle request to the EMIF (overriding the software settings of the MODULEMODE bit field for EMIF).
- EMIF stalls transactions on internal interconnect, completes all committed transaction to the external memory and then puts the external memory in self-refresh mode
- EMIF module then acknowledges the idle request of the PRCM module.
- The PRCM module uploads the values configured in the shadow register bit fields to the corresponding current register bit fields.
- The PRCM module waits for stabilization of the PHY\_ROOT\_CLK and DLL\_CLK clocks through hardware monitoring, according to new settings.
- The PRCM module requests the EMIF to exit idle (stops the overriding of the software settings of the MODULEMODE bit field for EMIF).
- The PRCM module clears the frequency update field.
- The EMIF copies the timing parameter configured in the shadow register as the current values.
- The EMIF acknowledges that the PRCM module is functional, and starts accepting the transactions on the internal interconnect.
- If DLL clock was changed, EMIF waits for DLL to be relocked and transactions on internal interconnect, if required. The PRCM module is not aware of the DLL operations.
- Accesses to SDRAM are serviced again:

Software execution can be stalled during the frequency change without harm as the critical hardware sequence is performed without software intervention. After having set the frequency update field, software should only poll for its clearing.

Accesses to external SDRAM are stalled during following duration:

- 2  $\mu$ s maximum due to EMIF entering and exiting idle
- + 400 DLL clock cycles in case DLL clock frequency has changed.
- + DPLL lock time in case DPLL has been locked.

#### **VDD\_CORE\_L voltage change for OPP switching**

It is assumed that DLL clock frequency is not changed and that the DPLL\_CORE mode is not changed and stays locked. Only the DDR\_PHY clock frequency is changed using the post divider. L3 clock is likely also changed, but it does not impact EMIF.

Accesses to external SDRAM should be stalled at maximum during 2  $\mu$ s due to this sequence.

Voltage scaling up consists of:

Software indicates to EMIF that voltage change operation has started to allow EMIF to ensure regular code updates from DLL masters.

Software disables SmartReflex for core voltage

Software performs direct VDD\_CORE setting to new noncorrected value (for weak silicon) and wait for voltage stabilization.

Software configures SmartReflex for corresponding OPP .

Software re-enable SmartReflex for core voltage.

Software indicates to EMIF that voltage change operation is complete and it need not control the code update from DLL masters directly.

#### **3.8.6.2 GPMC Clocks Frequency Scaling Constraints**

CD\_L3\_2 is inside the PD\_CORE. It supplies the L3\_ICLK2 clock to the general-purpose memory controller (GPMC) module, which is used as interface and functional clock. GPMC timing parameters are supposed be programmed at boot for the nominal OPP. Theses timing parameters can be used for lower L3 frequency, at the downside of having slower access to external memory.

External memory connected to GPMC may not support on-the-fly L3 frequency changes. Following is the sequence to perform L3 frequency change as part or not of an OPP change (DVFS).

##### **Hardware-controlled sequence**

1. Software writes new L3 clock settings in the shadow register and sets the frequency update field.
2. The PRCM module requests idle to GPMC.
3. GPMC requests disconnection of its interconnect port and completes all outstanding transactions.
4. GPMC acknowledges idle. Any access to GPMC is then stalled.
5. The PRCM module uploads the values of the shadow-register bit field in corresponding current bit field.
6. The PRCM module waits for stabilization of the L3 clocks by monitoring the DPLL\_CORE lock and/or divider acknowledge signals according to new settings.
7. The PRCM module clears the frequency update field.
8. The PRCM module requests the GPMC to exit idle.
9. GPMC re-allows interconnect connection and acknowledges it is functional.
10. Accesses to external memory are serviced again.

##### **NOTE:**

- The frequency update field is the same as the one for EMIF clock change. The same hardware in the PRCM module handles the frequency change of L3, DDR\_PHY, and DLL clocks.
- If the GPMC supports on-the-fly L3 frequency change, then the [CM\\_SHADOW\\_FREQ\\_CONFIG2\[0\]](#) GPMC\_FREQ\_UPDATE bit can allow bypassing of the sequence.

##### **Software-controlled sequence**

1. Software configures the PRCM module to disable GPMC.



2. The PRCM module requests idle to GPMC.
3. GPMC requests disconnection of its interconnect port and completes all outstanding transactions.
4. GPMC acknowledges idle. Any access to GPMC is then stalled.
5. Software programs the PRCM module to change L3 frequency and wait for L3 stabilization.
6. Software configures the PRCM module to re-enable GPMC.
7. GPMC re-allows interconnect connection and acknowledges it is functional.
8. Accesses to external memory are serviced again.

### 3.8.6.3 CORE DVFS Versus Subsystem Functionality

#### 3.8.6.3.1 Display Subsystem

The display subsystem is part of the CORE voltage domain. It has internal FIFOs that are sized to sustain a change of OPP on the CORE voltage domain with no DPLL relock and no DLL frequency change. Accesses to external SDRAM are stalled at maximum during 2  $\mu$ s in these conditions. DSS functional clock is not affected by the change of OPP on CORE voltage domain. Hence, DSS supports on the fly DVFS of the CORE voltage domain, assuming selected OPP remains compatible with the current DSS processing.

#### 3.8.6.3.2 IVAHD

IVAHD is part of the IVA voltage domain. It can sustain a temporary stall of external SDRAM accesses because it is not real-time processing. IVAHD functional clock is not affected by the change of OPP on the CORE voltage domain; therefore, IVAHD supports on-the-fly DVFS of the CORE voltage domain.

#### 3.8.6.3.3 Imaging Subsystem

The Imaging subsystem (ISS) is part of the CORE voltage domain. It is real-time processing (or data flow). It cannot support a 2- $\mu$ s stall of external SDRAM access, due to the heavy data throughput (FIFO size increase is prohibitive). As a consequence, ISS does not support on-the-fly change of OPP on CORE voltage domain. However, the CORE DVFS can be synchronized to the interframe idle window of ISS. During this window, ISS does not initiate transaction to external SDRAM. Core OPP can be changed safely assuming selected OPP remains compatible with the current ISS processing. ISS functional clock frequency can either remain unchanged, or be scaled according to the OPP change, assuming it remains compatible with the current ISS processing. This is the responsibility of software.

Example: CORE OPP change OPP50% $\rightarrow$ OPP100%, with functional clock frequency unchanged – OPP is 50%, ISS functional clock is at 50 MHz, clock coming from CORE DPLL, ISS clock divider is set to 4

Software performs OPP change: OPP50% $\rightarrow$ OPP100%, – ISS is now at 100 MHz – SW sets the ISS clock divider to 8. - ISS functional clock is back at 50 MHz

### 3.8.6.4 DVFS Voting Mechanism (VDD\_CORE\_L and VDD\_IVA\_L Voltage Domains)

#### 3.8.6.4.1 CORE OPP Voting Architecture

Core voltage domain operating point can be determined based on the following system masters:

- APE DSP processor
- APE Cortex-M3 MPU processor
- Modem

The PRCM module provides:

- Four read/write voting registers (that is, CM\_CORE\_DVFS\_PERFn, where n = 1 to 4). Software can use these, to specify the performance level of the CORE voltage domain as requested by each system master (likely EMIF bandwidth only). Software must encode the bit fields; encoding is not interpreted by hardware.
- One read/write register (that is, CM\_CORE\_DVFS\_CURRENT) to identify the current CORE voltage domain OPP setting. Software must configure this register based on current OPP setting of the register. It is not managed by hardware.

See [Section 3.8.6.4.3, Voting Mechanism](#), to understand the voting mechanism for DVFS on the voltage domain.

### 3.8.6.4.2 IVA OPP Voting Architecture

IVA voltage domain operating point can be determined based on the following subsystems performing different type of processing:

- DSP processor
- IVAHD accelerator
- ABE

The PRCM module provides:

- A read/write voting register (that is, `CM_IVA_DVFS_PERF_<Subsystem name>`) for each subsystem. Software can use these, to specify the performance level of the IVA voltage domain as requested by each subsystem. Software must encode the bit fields; encoding is not interpreted by hardware.
- One read/write register, that is, `CM_IVA_DVFS_CURRENT`, to identify the current IVA voltage domain OPP setting. It is again the software's responsibility to configure this register based on current OPP setting of the register. It is not managed by hardware.

See [Section 3.8.6.4.3, Voting Mechanism](#), to understand the voting mechanism for DVFS on the voltage domain.

### 3.8.6.4.3 Voting Mechanism

The operating point change for the CORE and IVA voltage domains can be performed by the respective masters or subsystems by executing the following steps:

1. Obtain ownership of the voltage domain DVFS semaphore using the SPINLOCK module.
2. Write new performance requirement value in its voting register.
3. Read the current OPP and the voting registers.
4. Compute the result of the voting using ad hoc algorithm (for example, sum of performance requirements).
5. If the current OPP does not match the voting result (that is, either the OPP must be decreased in case of low performance demand, or the OPP must be increased in case of high performance demand) then the system master performs voltage domain OPP change to best match the voting result.
6. When change of OPP is completed, update the current voltage domain OPP register accordingly.
7. Release the voltage domain DVFS semaphore, using the SPINLOCK module.

## 3.9 Device Low-Power States

The device low-power states are the result of any valid combination of power domain states in which all the power domains are no longer in the ACTIVE state. In such a situation the PRCM module hardware can trigger events to further lower the consumption of the device and the system.

Typically, the PRCM module can lower or even turn off the SMPS, shut down the system clock supplier, and allow the external power IC to turn off its internal resources.

These device low-power states are characterized by the system power consumption, the wake-up latency, and the required functionality.

The low-power states are:

- RETENTION: All the logic voltage domain (`VDD_IVA_L` and `VDD_CORE_L`) are in Retention state
- OFF: All the logic voltage domains are in Off state
- LOW POWER: Any combination of logic voltage domains states (ON, SLEEP, and RETENTION state), other than those identified above.

Once the PRCM module hardware detects any valid combination of power domain state and if a proper programming model of the PRCM module has been set, the PRCM module automatically triggers the transition into the device low-power mode.

### 3.9.1 Device Wake-Up Source Summary

Each power domain in the device may contain modules that can generate the wake-up events. However, when the device is in OFF power state and all the logic voltage domains are off, only the modules with wake-up capability and that are in the PD\_WKUP power domain can generate the wake-up event. Modules in the other power domains require that their voltage domain be in RETENTION, SLEEP, or ON state to generate a wake-up event. Some modules may have a further constraint that their power domain should be ON-INACTIVE or in CSWR RETENTION state to generate the wake-up event.

The wake-up events can be asynchronous or synchronous. Synchronous wake-up events require the 32-kHz clock or the system clock to be active, while asynchronous wake-up events do not require any active clock.

The Modules Attributes subsection of each clock domain in [Section 3.6, Clock Management Functional Description](#), presents the wake-up capability support for each module of the corresponding power domain.

### 3.9.2 Device RETENTION State Management

In Retention state, the voltages to the device are lowered to their minimum value to retain only the logic build with RFFs, while all DFF logic is lost. The memory area may be retained, depending on software configuration.

The device integrates enhanced device wake-up management while in Retention state. This allows:

- Triggering wake-up with limited capabilities from almost all programmed input pads all around the device.
- Triggering wake-up with full capabilities by using few GPIO inputs in the PD\_WKUP power domain.

The logic voltages VDD\_CORE\_L, VDD\_MPU\_L, and VDD\_IVA\_L can transition to their retention value independently. Once the voltage domain state transition conditions are met in compliance with the software configuration, the PRCM module hardware initiates the transition into Retention state by sending an I<sup>2</sup>C command for the corresponding SMPS.

When all the logic voltages are transitioned to their Retention state, the PRCM module hardware can, depending on a software configuration, toggle its system clock request and/or its power request to allow the SCRM to shut down the system clock source and to request the external power IC to shut down internal resources.

The PRCM module, SCRM, and the external power IC can also be programmed so that the voltage VDD\_CORE\_L is managed by the power request signal. This means the following:

- The power IC manages the ramp-down and ramp-up time of the VDD\_CORE\_L SMPS upon toggle of the power request.
- The SCRM accounts for the overall setup of the PMIC resources (internal PMIC resources and VDD\_CORE\_L SMPS).
- The PRCM module sends Retention and On commands through the I<sup>2</sup>C interface. Indeed, the retention voltage value must be equal to the current voltage value and the value of the On voltage must be set to the expected voltage value following the wakeup. Also, the setup time periods corresponding to the SMPS ramp-up and ramp-down must be set to a zero value.

### 3.9.3 Device OFF State Management

The off mode is the device mode where all power domains except for PD\_WKUP are in OFF state and the logic voltage domains (VDD\_MPU\_L, VDD\_IVA\_L and VDD\_CORE\_L) are shut down by the external power IC. Other power sources, such as the DPLL supplies, can be shut down by the external power IC during the off mode period.

The device integrates enhanced device wake-up management while in Retention state. This allows:

- Saving I/Os leakage by settings pads in the lowest power state compatible with the device environment.
- Triggering wake-up with limited capabilities from almost all programmed input pads all around the device.
- Triggering wake-up with full capabilities by using few GPIO inputs in the PD\_WKUP power domain.

This section details the way the PRM manages and sequences the control of the various off mode contributors:

- LDO\_WAKEUP
- Memory LDOs
- ABB LDOs
- Voltage controller
- Hardware save-and-restore mechanism
- Hardware power request signal

---

**NOTE:** For more information about the save-and-restore mechanism, see [Section 16.4.21](#), *Auto-Restore Feature*, in [Chapter 16](#).

---

Once the hardware conditions is met and according to the software configuration the PRCM module hardware initiates the transition into off mode by sending I<sup>2</sup>C commands for each SMPS and by respecting required sequencing between the voltage domains.

When all the logic voltages are transitioned to their Retention state, the PRCM module hardware can, depending on a software configuration, toggle its system clock request and/or its power request to allow the SCRM to shut down the system clock source and to request the external power IC to shut down internal resources.

The PRCM module, SCRM and the external power IC can also be programmed so that the voltage VDD\_CORE\_L is managed by the power request signal. This means the following:

- The power IC manages the ramp-down and ramp-up time of the VDD\_CORE\_L SMPS upon toggle of the power request.
- The SCRM accounts for the overall setup of the PMIC resources (internal PMIC resources and VDD\_CORE\_L SMPS).
- The PRCM module sends Off and On commands through the I<sup>2</sup>C interface. Indeed, the Off voltage value must be equal to the current voltage value and the value of the On voltage must be set to the expected voltage value following the wakeup. Also, the setup time periods corresponding to the SMPS ramp-up and ramp-down shall be set to a zero value.

### 3.9.3.1 Device Off Mode Sleep Sequence

This section describes the main steps of the device off mode sleep sequence. The Off command to shut down the voltages VDD\_MPU\_L, VDD\_IVA\_L, and VDD\_CORE\_L can be sent through the I<sup>2</sup>C interface. The VDD\_CORE\_L may also be controlled independently using the power request signal.

---

**NOTE:** The power request signal can be used to turn off other resources of the power-management IC, even when it is not used for the VDD\_CORE\_L management.

---



---

**NOTE:** LDOs in the power companion device (TWL60xx), which are turned off in ACTIVE state, are switched on when entering into SLEEP state with the default value.

After the first power on, all resources must be configured, especially the power resources that are not part of the power-up sequence. This must be done by sending a command through the I<sup>2</sup>C interface.

---

1. Software has completed preliminary settings such as:
  - Settings in the external power IC
  - Voltage transition settling time settings in PRCM
  - Context save for the relevant modules
  - System control module is configured with valid I/O pads off mode configuration.
  - Enable/disable the wake-up feature for each pad by programming the system control module.
  - Enable/disable the automatic gating of the system clock (PRCM).
  - Enable the I/O wake-up scheme (PRCM).
  - Program a consistent power state for each power domain (that is, Off state for domains without

hardware restored modules and OSWR state for domains with hardware restored modules build with RFF logic).

- All modules in the PD\_WKUP power domain must have their interrupts to Cortex-M3 and DSP, and their DMA requests to sDMA disabled.
- Program automatic transition of the CD\_WKUP clock domain.
- Enable the off mode scheme (PRCM).
- Configure the VDD\_CORE\_L management option:
  - Disable the option if power request signal is not used for VDD\_CORE\_L management.
  - Enable the option if power request signal is used for VDD\_CORE\_L management.

2. The PRCM module initiates the transition into off mode upon all following conditions:

- All the programmable power domains have reached their targeted Low-Power state which can either be Off or OSWRET. It does not apply to always-on power domains and to the EMU power domain.
- The CD\_WKUP clock domain is idled.
- The off mode has been enabled by software.

At this stage the I/O wake-up scheme through the I/O pad daisy-chain is already enabled. The I/O isolation scheme is managed in parallel with the following sequence (see [Section 3.9.4, I/O Management](#)).

3. The PRCM module triggers in parallel:

- Switch off all SRAM LDOs.
- Bypass all ABB LDOs.
- Isolate the analog area from digital supply removal.

4. The PRCM module then sends in a row the Off command for the voltages VDD\_IVA\_L and the VDD\_MPU\_L, once the following conditions are met:

- I/O isolation scheme is complete.
- SRAM LDOs have ramped down to 0 V.
- ABB LDOs have switched in bypass.

The order between the two Off commands is irrelevant and managed by the voltage controller.

5. Once the voltages VDD\_IVA\_L and VDD\_MPU\_L are ramped down to 0 V the PRCM module sends the Off command for the voltage VDD\_CORE\_L:

- If power request signal is not used for VDD\_CORE\_L management then the VDD\_CORE\_L is ramped down to 0 V.
- If power request signal is used for VDD\_CORE\_L management then the Off command for the voltage VDD\_CORE\_L, through the I<sup>2</sup>C interface, has no effect and the voltage VDD\_CORE\_L remains unchanged.

6. Once the Off command for VDD\_CORE\_L is sent and acknowledged, the PRCM module gates locally its system clock.

7. The PRCM module then deassert the signal system clock request to the SCRM, if so configured by the software. Similarly, it deasserts the signal power request if configured by software.

- If power request signal is used for VDD\_CORE\_L management, then VDD\_CORE\_L is ramped down to 0 V.

8. The SCRM then acknowledges the power request to the PRCM module. This acknowledge is required by the PRCM module before toggling again the signal power request.

9. The SCRM informs the PRCM module that all versions of the system clock have been gated; hence the PRCM module can transition the LDO\_WAKEUP to SLEEP state if the PD\_EMU power domain is in OFF state.

10. The DEVICE\_OFFWKUP\_CORERSTACTST signal intended for the core control module is asserted. The EMIF1\_DEVICE\_OFFWKUP\_CORERSTACTST and EMIF2\_DEVICE\_OFFWKUP\_CORERSTACTST signals intended for EMIF1 and EMIF2, respectively, are asserted according to their associated register bit fields, EMIF1\_OFFWKUP\_DISABLE and EMIF1\_OFFWKUP\_DISABLE.

### 3.9.3.2 Device Off Mode Wake-Up Sequences

The wake-up sequence includes a phase of hardware restore. Depending on the wake-up event source the restore sequence completes in one or two phases. The first phase always happens and is sufficient to serve some type of wake-up events; for other sources of event a second phase is required and occurs immediately after the first phase. Phase 2 is also split into two subphases (phase 2a and phase 2b). The device returns to off mode after the first phase completes.

Table 3-295 lists the device off wake-up and hardware restore phases.

**Table 3-295. Device Off Wake-Up and Hardware Restore Phases**

Wake-Up Event	Phase 1	Phase 2	Comment
Wake-up request from PD_WKUP	Yes	Yes	This event must wake up the PD_MPU power domain by dependency.
PRM interrupt toward MPU-A9 (includes I/O wake-up detection)	Yes	Yes	This event wakes up the PD_WKUP power domain. The PD_MPU power domain is woken up by broadcast of the wake-up event and the MPU-A9 boots up.
Global warm reset	Yes	No	After voltage stabilization, phase 1 restoration is performed. Then MPU-A9 reboots, because global warm reset is a source of PD_MPU power domain wakeup.

#### 3.9.3.2.1 Wake Up From a Wake-Up Event

This section describes the main steps of the wake-up sequence to exit the device from off mode. The On command to ramp-up the voltages VDD\_MPU\_L, VDD\_IVA\_L, and VDD\_CORE\_L can be sent through the I<sup>2</sup>C interface. VDD\_CORE\_L can also be controlled independently using the power request signal.

---

**NOTE:** The power request signal can be used to turn on other resources of the power-management IC, even when it is not used for the VDD\_CORE\_L management.

---

Additionally it is assumed in this sequence that the wake-up source is not a global warm reset.

1. The PRCM module switches the LDO\_WAKEUP from SLEEP to ON state and waits for the setup time of the LDO\_WAKEUP (that is, four clock cycles of the 32-kHz clock).
2. The PRCM module asserts the system clock request signal to SCRM, which in turn sends the request to the external power IC. The SCRM accounts for the system clock stabilization time before enabling the clock to the PRCM module.
3. The PRCM module also asserts the power request signal to the SCRM. The PRCM module waits for the acknowledge of its power request signal and activation of the system clock before sending an I<sup>2</sup>C command for the voltage ramp-up sequence.
  - If power request signal is used for VDD\_CORE\_L management, then the power-management IC ramps up the voltage VDD\_CORE\_L.
  - If power request signal is not used for VDD\_CORE\_L management, then the voltage VDD\_CORE\_L is unchanged.
4. The PRCM module sends the On command for the VDD\_CORE\_L voltage ramp up, through the I<sup>2</sup>C interface and waits for the setup time of the SMPS.
  - If power request signal is used for VDD\_CORE\_L management, then the voltage VDD\_CORE\_L is unchanged.
  - If power request signal is not used for VDD\_CORE\_L management, then the power-management IC ramps up the voltage VDD\_CORE\_L.
5. The PRCM module sends in a row the On commands for the voltages VDD\_MPU\_L and VDD\_IVA\_L and waits for the setup time of the two SMPS.
6. The PRCM module enables the system clock to the eFuse controller and releases its reset line. It holds the other asserted resets until completion of the EFUSE bits sensing.
7. The PRCM module waits for the stabilization of the bandgap, memory LDOs and ABB LDOs stabilization.
8. Hardware restore phase 1 starts:
  - (a) The PRCM module ensures that:

- CORTEXA9, CORTEXM3, and DSP are held in reset.
  - C2C is kept in STANDBY state.
  - sDMA receives the restore request.
  - Phase 1 descriptor start address is sent to the sDMA.
  - MODULEMODE bit field of the L3\_3 module is overridden to auto mode. This ensures the successful restoration of its firewalls during phase 1 of the restore.
- (b) The PRCM module generates a wake-up event on the CD\_SDMA clock domain. As a result, the PD\_CORE power domain and CD\_SDMA wake-up. This causes the CD\_L3\_1, CD\_L3\_2, CD\_L4\_CFG, CD\_MEMIF, CD\_L4\_PER, and CD\_L3\_INIT to wake up because of the wake-up broadcast due to static dependency and their respective reset are released once their clock is active. The clock domain L4\_WKUP is woken up (due to a wake-up broadcast on static dependency).
- (c) The sDMA switches from STANDBY to ACTIVE state and performs phase 1 of the hardware restore:
- Firewalls
  - System control module
  - EMIF
  - Part of the CM1 and part of the CM2 (core clock configuration, registers CLKTRCTRL of the CD\_MPU, CD\_L3\_1, CD\_L3\_2, CD\_L4\_CFG, CD\_MEMIF, CD\_L4\_PER, and CD\_L3\_INIT clock domains)
- The sDMA make the last write into the PRCM module end of phase 1 bit field. The PRCM module then switches it to STANDBY state and resets it.
- (d) The PRCM module releases I/O isolation of the EMIF. The other I/Os stay isolated.
- (e) The PRCM module checks and waits if necessary for the DPLL\_CORE to be in a stable state.
- (f) In parallel the PRCM module does the following:
- Release the condition preventing C2C to remain in STANDBY state
  - Release the override of the MODULEMODE bit field of the L3\_3 module
9. Upon a need to perform phase 2 of the hardware automatic restore (according to the wake-up source), the following sequence occurs:
- (a) Phase 2a sequence.
- (i) The PRCM module drives the phase 2a descriptor start address to the sDMA.
  - (ii) The PRCM module releases the sDMA from reset.
  - (iii) The PRCM module generates a wake-up event on the CD\_SDMA clock domain (same as in Step 2 of phase 1).
  - (iv) The sDMA performs phase 2a of the hardware restore:
    - The MODULEMODE bit field of each module to be restored during phase 2b is set to Enabled or Auto and the SARMODE bit field is enable if applicable. Only modules that have just interface clock(s) can be restored automatically.
    - Register [CM\\_SDMA\\_STATICDEP](#) is set to enable all necessary static dependency To wake up all power domains having modules to be restored during phase 2b

The sDMA makes the last write into the PRCM module at the end of phase 2a bit field. The PRCM module then switches it to Standby state and resets it. This completes phase 2a.
- (b) Phase 2b sequence.
- (i) The PRCM module drives the phase 2b descriptor start address to the sDMA.
  - (ii) The PRCM module releases the sDMA from reset.
  - (iii) The PRCM module generates a wake-up event on the CD\_SDMA clock domain (same as in Step 2 of phase 1).
  - (iv) The sDMA performs phase 2b of the hardware restore:
    - Restore of the relevant modules

The sDMA make the last write into the PRCM module at the end of phase 2b bit field. The PRCM module then switches it to STANDBY state and resets it. This completes phase 2b.
10. The PRCM module releases I/O isolation of the remaining I/O, only if the [PRM\\_IO\\_PMCTRL.ISOOVR\\_EXTEND](#) bit field is not set.
11. The PRCM module releases the reset stall conditions holding CORTEXA9, CORTEXM3, and DSP in reset.
12. The PD\_MPU power domain is woken up if required according to the wake-up source.

### 3.9.3.2.2 Wakeup Upon Global Warm Reset

When global warm reset is the source of the device wakeup, the sequence referred in [Section 3.9.3.2.1, Wake up From a Wake-Up Event](#), is modified as follows:

Steps 4 and 5 are replaced by:

- The PRCM module waits for the three voltages VDD\_MPU\_L, VDD\_IVA\_L, and VDD\_CORE\_L stabilization time. It uses the [PRM\\_VOLTSETUP\\_WARMRESET](#) register for that purpose.

Step 6 is replaced by:

- The PRCM module enables the system clock to the eFuse controller and releases its reset line. In parallel, the device reset manager counts for global reset extension (setup by the [PRM\\_RSTTIME\[9:0\] RSTTIME1](#) bit field). The PRCM module holds the other asserted resets until completion of the eFuse bits sensing and the global reset counter overflow.

Phase 2 of the hardware restore sequence is skipped.

The hardware blocks and modifies the [FREQ\\_UPDATE](#) bit field of [CM\\_SHADOW\\_FREQ\\_CONFIG1](#) register during phase 1.

### 3.9.3.3 Global Warm Reset During a Device Wake-Up Sequence

If a global warm reset occurs before the PRCM module completes voltage stabilization count, the global warm reset is applied immediately. As a consequence, the sequence described in [Section 3.9.3.2.2, Wake up Upon Global Warm Reset](#), is performed.

If the global warm reset occurs after the PRCM module completes voltage stabilization (during eFuse sensing or during phase 1 of automatic restore):

- Global warm reset is delayed and applied after of phase 1 restore completes.
- Phase 2 of restore is discarded.
- MPU boots after the warm reset sequence completes.

If the global warm reset occurs during phase 2 of automatic restore:

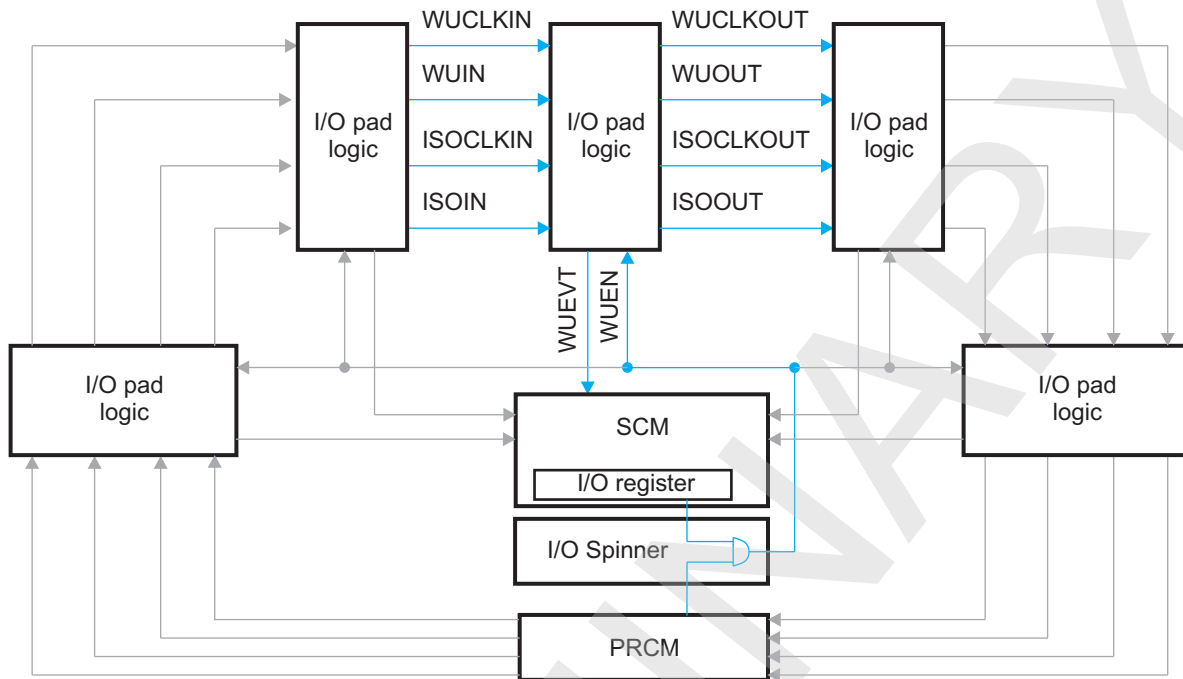
- Global warm reset is applied immediately.
- Phase 2 of restore is discarded.
- MPU boots after the warm reset sequence completes.

### 3.9.4 I/O Management

[Figure 3-74](#) is an overview of the power-management modules and their internal connections with a generic power domain.



Figure 3-74. I/O Pads Daisy-Chain Configuration



prcm-076

Any I/O pad of the device can be configured to generate a wake-up event when the device is in off mode. This is managed by the configurations in the system control module and the PRCM.

In the off mode, the I/O pads of the device form a daisy chain with the I/O pad logic output of one pad connected to the I/O pad logic input of the next one. The I/O pad logic at the two ends of the chain is connected to the PRCM.

The I/O pad wake-up scheme must be enabled (WUEN signal) globally by setting the [PRM\\_IO\\_PMCTRL\[16\]](#) GLOBAL\_WUEN bit and by also setting I/O pad wake-up enabled/disabled individually (WUEN signal) by writing to the following bit fields in the control module:

- CONTROL.CONTROL\_PADCONF\_<IOpad>[14] WAKEUPENABLE0
- CONTROL.CONTROL\_PADCONF\_<IOpad>[30] WAKEUPENABLE1

Once configured, the wake-up scheme within each I/O pad is enabled and disabled by triggering a control (WUCLKIN signal) of the I/O daisy chain. This is done thanks to a dedicated PRCM module register bit [PRM\\_IO\\_PMCTRL\[8\]](#) WUCLK\_CTRL. The software must enable the I/O wakeup prior entering a low-power mode and disable it following a wake-up event.

Writing the [PRM\\_IO\\_PMCTRL\[8\]](#) WUCLK\_CTRL bit to 1 asserts signal WUCLKIN high to reset spurious wake-up event and to latch the current pad input value. PRCM module register [PRM\\_IO\\_PMCTRL\[9\]](#) WUCLK\_STATUS logs the signal WUCLKOUT of the last pad of the I/O ring. Once this status is set to 1, the software may clear the [PRM\\_IO\\_PMCTRL\[8\]](#) WUCLK\_CTRL bit to effectively enable or disable the wake-up feature within each pad.

Additionally, it is required that the WUCLKIN is forced to 1 during global power-on reset. The I/O wake-up feature is enabled or disabled depending on a control (WUEN) for each I/O pad. This control is generated by the Spinner logic resulting of the combination of a local enable/disable bit per pad located in the control module and a global enable/disable bit located in the PRCM module, that is, [PRM\\_IO\\_PMCTRL\[16\]](#) GLOBAL\_WUEN.

When a wake-up event is detected by a wake-up enabled I/O pad in the daisy chain, it is logged within the I/O and is conveyed to the system control module (WUEVT signal). All the I/O wake-up event statuses are mapped on memory-mapped registers in the system control module.

When the device wakes up, the MPU can determine all sources of the current wake-up event logged into the corresponding

- CONTROL.CONTROL\_PADCONF\_<IOpad>[15] WAKEUPEVENT0
  - CONTROL.CONTROL\_PADCONF\_<IOpad>[31] WAKEUPEVENT1
- bit fields in the Control Module.

For information about the Control Module, see [Chapter 18, Control Module](#).

I/Os have two associated buffers, that is, 1.2 V and 1.8 V. A software configuration determines which one is used in normal operation. However, in the device off mode, the 1.2-V buffer is always disabled, and the 1.8-V buffer stays on but is powered by 1.2 V instead of 1.8 V to minimize leakage, if pad configure to drive the output.

### 3.9.4.1 Hardware-Controlled I/O Isolation Sequences

I/Os have 2 buffers: 1.2 V and 1.8 V.

The software configuration determines whether the 1.2-V buffer or the 1.8-V buffer is used. Regardless of the buffer used, in off mode the 1.2-V buffer is always disabled and the 1.8-V buffer stays on but is powered by 1.2 V instead of 1.8 V to minimize leakage (if the pad is configure to drive the output).

In addition to this, I/Os support the following features:

**wake-up override:** The PRCM module can control the I/Os that are woken-up, that is, brought out of isolation from device off mode based on the source of wakeup. The primary application of this feature is to only restore minimal context of the device for access to EMIF upon C2C request.

**Wakeup Phased Wakeup:** The I/Os support a phased wake-up approach, meaning, the I/Os can be brought out of off mode isolation in two phases without affecting the functional operation of the I/Os already out of isolation. This feature is used to wake up the I/Os still held in isolation without affecting the EMIF operations when a full device wake-up is requested.

The I/O transitions are:

- Full on to full isolation - Pad output controls are stable (A, GZ). Pad off mode configuration is activated.
- Full Isolation to EMIF on - EMIF I/O interface is brought out of isolation. Partial device context is restored.
- EMIF on to Full on - EMIF I/O interface are already out of isolation and a new wake-up event requires a full device wakeup.
- EMIF on to Full Isolation - EMIF I/O interface and some nonrestore sensitive I/Os are already out of isolation. The PRCM module puts the device back in off mode.

### 3.9.4.2 Software-Controlled I/O Isolation

In addition to the hardware-controlled I/O transitions, there is a dedicated PRCM module register bit, [PRM\\_IO\\_PMCTRL\[0\] ISOCLK\\_OVERRIDE](#), that can be use by the software to force ISOCLKIN signal to the I/O pad. It is then propagated from one pad to the other in the wake-up enabled dasy chain of I/Os. The PRCM module register [PRM\\_IO\\_PMCTR\[1\] ISOCLK\\_STATUS](#) logs the signal ISOCLKOUT of the last pad of the I/O dasy chain.

---

**NOTE:** Override should be used at boot time only when software needs to change the mode of an I/O from 1.8-V default mode to 1.2-V mode. When not overridden, this signal is controlled by hardware only.

---

[PRM\\_IO\\_PMCTRL\[4\] ISOOVR\\_EXTEND](#) bit allows extending the nonEMIF I/O isolation. This feature can be used by software to restore modules driving output such as GPIO while nonEMIF I/Os are still isolated. Once software has completed relevant module restore, it clears the bit and hardware performs the Full Isolation to EMIF on hardware-controlled I/O transition.

The [PRM\\_IO\\_PMCTRL\[5\] IO\\_ON\\_STATUS](#) status bit is available for software to check completion of the EMIF on transition.

### 3.10 PRCM Module Programming Guide

#### 3.10.1 DPLLs Low-Level Programming Models

This section covers the low-level hardware programming sequences for configuration and use of the module.

##### 3.10.1.1 Global Initialization

###### 3.10.1.1.1 Surrounding Module Global Initialization

This section identifies the requirements of initializing the surrounding modules when the module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DPLLs.

Table 3-296 shows the global initialization of the surrounding modules.

**Table 3-296. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	Ensure that the DPLL reference clock (gated version of system clock) is active.

###### 3.10.1.1.2 DPLL Global Initialization

###### 3.10.1.1.2.1 Main Sequence – DPLL Global Initialization

This procedure initializes the DPLL after a power-on or software reset and then locks it to the desired synthesized clock frequency.

**Table 3-297. DPLL Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Configure SSC parameters	See Section 3.10.1.1.2.2.	
Configure Recalibration parameters	See Section 3.10.1.1.2.3.	
Set DPLL automatic idle mode	CM_AUTOIDLE_<DPLL name>[2:0] AUTO_DPLL_MODE	xx <sup>(1)</sup>
Configure synthesized clock parameters	See Section 3.10.1.1.2.4.	
Configure output clocks parameters	See Section 3.10.1.1.2.5.	
Lock DPLL	CM_CLKMODE_<DPLL name>[0] DPLL_EN	0x7

(1) It depends on the desired auto idle mode. See Section 3.6.3.3.3, DPLL Power Modes.

###### 3.10.1.1.2.2 Subsequence – SSC Parameter Configuration

This procedure configures the SSC parameters for the DPLL and enables the SSC feature.

**Table 3-298. DPLL SSC Parameter Configuration**

Step	Register/Bit Field/Programming Model	Value
Configure M Step Size	CM_SSC_DELTAMSTEP_<DPLL name>[19:0] DELTAMSTEP	xx <sup>(1)</sup>
Configure modulation frequency divider exponent part	CM_SSC_MODFREQDIV_<DPLL name>[10:8] MODFREQDIV_EXPONENT	xx <sup>(1)</sup>
Configure modulation frequency divider mantissa part	CM_SSC_MODFREQDIV_<DPLL name>[6:0] MODFREQDIV_MANTISSA	xx <sup>(1)</sup>
Enable/disable frequency downspread	CM_CLKMODE_<DPLL name>[14] DPLL_SSC_DOWNSPREAD	xx <sup>(2)</sup>
Enable SSC feature	CM_CLKMODE_<DPLL name>[12] DPLL_SSC_EN	0x1

**Table 3-298. DPLL SSC Parameter Configuration (continued)**

Step	Register/Bit Field/Programming Model	Value
Wait until SSC start acknowledge	CM_CLKMODE_<DPLL name>[13] DPLL_SSC_ACK	0x1

(1) See SSC description in [Section 3.6.3.3.5, DPLL Spread Spectrum Clocking](#).

(2) The selected value depends on the desired SSC feature settings.

### 3.10.1.1.2.3 Subsequence – Recalibration Parameter Configuration

This procedure enables the recalibration feature and the associated processor interrupt flag.

**Table 3-299. DPLL Recalibration Parameter Configuration**

Step	Register/Bit Field/Programming Model	Value
Clear recalibration interrupt status	PRM_IRQSTATUS_<Processor name>[x] <DPLL name>_RECAL_ST	0x0
Unmask recalibration interrupt flag	PRM_IRQENABLE_<Processor name>[x] <DPLL name>_RECAL_EN	0x1
Enable recalibration feature	CM_CLKMODE_<DPLL name>[8] DPLL_DRIFTGUARD_EN	0x1

### 3.10.1.1.2.4 Subsequence – Synthesized Clock Parameter Configuration

This procedure configures the settings for the synthesized clock of the DPLL.

**Table 3-300. DPLL Synthesized Clock Parameter Configuration**

Step	Register/Bit Field/Programming Model	Value
Set DPLL clock synthesis multiplier	CM_CLKSEL_<DPLL name>[2:0] DPLL_MULT	xx <sup>(1)</sup>
Set DPLL clock synthesis divider	CM_CLKSEL_<DPLL name>[2:0] DPLL_DIV	xx <sup>(1)</sup>
<b>IF</b> : Low-power mode operation conditions satisfied?	Software test condition. See <a href="#">Section 3.6.3.3.2</a> .	
Enable DPLL low-power operation mode	CM_CLKMODE_<DPLL name>[10] DPLL_LPMODE_EN	0x1
<b>ENDIF</b>		

(1) It depends on the desired synthesized clock frequency. See [Section 3.6.3.3, Generic DPLL Overview](#).

### 3.10.1.1.2.5 Subsequence – Output Clock Parameter Configuration

This procedure configures the settings for the output clocks of the DPLL.

**Table 3-301. DPLL Output Clock Parameter Configuration**

Step	Register/Bit Field/Programming Model	Value
Set output clock dividers (that is, M2, M3 and Mn), where n is from 4 to 7. It depends on the available clock output of the DPLL.	CM_DIV_M2_<DPLL name>[4:0] DPLL_CLKOUT_DIV CM_DIV_M3_<DPLL name>[4:0] DPLL_CLKOUTHIF_DIV CM_DIV_Mn_<DPLL name>[4:0] HSDIVIDER_CLKOUTn-3_DIV	xx <sup>(1)</sup>
Wait until divider change acknowledged	CM_DIV_M2_<DPLL name>[5] DPLL_CLKOUT_DIVCHACK CM_DIV_M3_<DPLL name>[5] DPLL_CLKOUTHIF_DIVCHACK CM_DIV_Mn_<DPLL name>[5] HSDIVIDER_CLKOUTn-3_DIVCHACK	Bit toggled <sup>(2)</sup>
Enable/disable output clock dividers auto power-down feature. It applies to M2, M3, and Mn dividers, where n is from 4 to 7. It depends on the available clock output dividers of the DPLL.	CM_DIV_M2_<DPLL name>[12] DPLL_M2_PWDN CM_DIV_M3_<DPLL name>[12] DPLL_CLKOUTHIF_PWDN CM_DIV_Mn_<DPLL name>[12] HSDIVIDER_CLKOUTn-3_PWDN	xx <sup>(3)</sup>

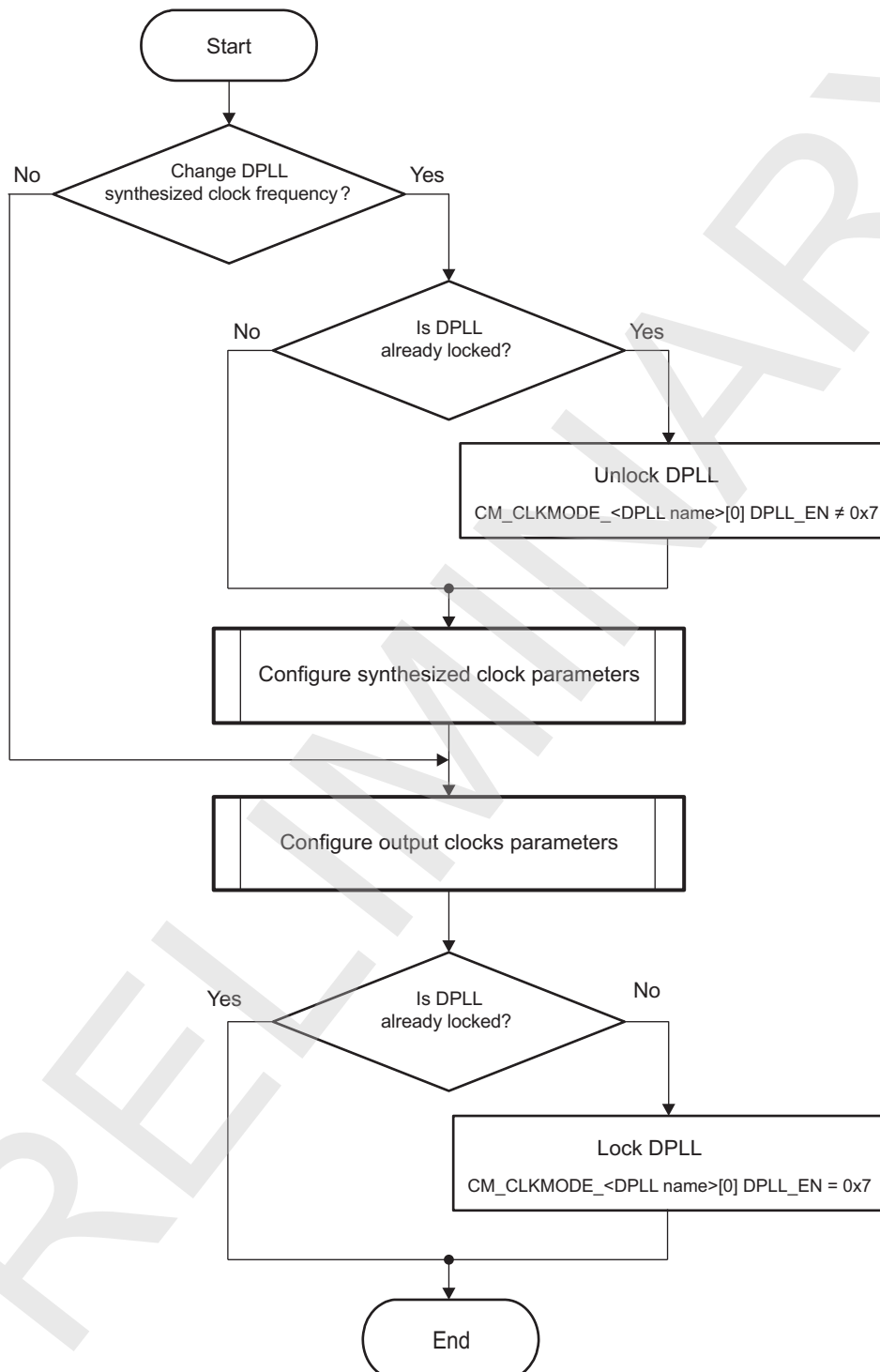
**Table 3-301. DPLL Output Clock Parameter Configuration (continued)**

Step	Register/Bit Field/Programming Model	Value
Enable/disable output clocks automatic gating feature. It applies to M2, M3, and Mn clock outputs, where n is from 4 to 7. It depends on the available clock output of the DPLL.	CM_DIV_M2_<DPLL name>[8] DPLL_CLKOUT_GATE_CTRL CM_DIV_M3_<DPLL name>[8] DPLL_CLKOUTHIF_GATE_CTRL CM_DIV_Mn_<DPLL name>[8] HSDIVIDER_CLKOUTn-3_GATE_CTRL	xx <sup>(4)</sup>

- (1) It depends on the desired output clock frequency. See [Section 3.6.3.3, Generic DPLL Overview](#).
- (2) It is toggled only if the programmed divider value differs from the previous one.
- (3) It depends on whether output clock divider automatic power down feature is to be enabled or not. See [Section 3.6.3.3.6, DPLL Output Power-Down](#).
- (4) It depends on whether output clock automatic gating feature is to be enabled or not.

### 3.10.1.2 DPLL Output Frequency Change

[Figure 3-75](#) shows the DPLL output-frequency change.

**Figure 3-75. DPLL Output-Frequency Change**

prcm-077

**Table 3-302. Register Call Summary for Sequence – DPLL Output Frequency Change**

Register Name
CM_CLKMODE_<DPLL name>

**Table 3-303. Subprocess Call Summary for Sequence – DPLL Output Frequency Change**

Subprocess Name	Cross-reference
Configure synthesized clock parameters	See <a href="#">Section 3.10.1.1.2.4.</a>
Configure output clocks parameters	See <a href="#">Section 3.10.1.1.2.5.</a>

### 3.10.2 Clock Management Low-Level Programming Models

This section covers the low-level hardware programming sequences for configuration and use of the clocks in the device.

#### 3.10.2.1 Global Initialization

##### 3.10.2.1.1 Surrounding Module Global Initialization

This section identifies the requirements of initializing the surrounding modules when the module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DPLLs.

[Table 3-304](#) shows the global initialization of the surrounding modules.

**Table 3-304. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM DPLLs	Ensure that the DPLLs managed by the PRCM module are initialized.

##### 3.10.2.1.2 Clock Management Global Initialization

###### 3.10.2.1.2.1 Main Sequence – Clock Domain Global Initialization

This procedure initializes the clock domain of the device after a power-on or software reset.

**Table 3-305. Clock Domain Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Configure module clock-management feature of master modules in the clock domain	<Module name>_SYSCONFIG[x] MIDDLEMODE <Module name>_SYSCONFIG[x] STANDBYMODE	xx <sup>(1)</sup>
Configure module clock-management feature of slave modules in the clock domain	See <a href="#">Section 3.10.3.3.</a>	
Enable/disable Static Sleep Dependency with other clock domains (that is, destination clock domains). Not all dependencies are configurable.	CM_<Clock Domain name>_STATICDEP [x] <Destination Clock Domain name>_STATDEP	0x0 to Disable 0x1 to Enable
Set Dynamic Dependency window size	CM_<Clock Domain name>_DYNAMICDEP[27:24] WINDOWSIZE	xx <sup>(2)</sup>
Enable/disable module wake-up dependency for the modules of the clock domain. It is available when the module can generate an interrupt or a DMA request to a service provider module (for example, a processor or DMA).	PM_<Clock Domain name>_<Module name>_WKDEP[x] WKUPDEP_<Module name>_<DMA/IRQ request>_<DMA/Processor name>	0x0 to disable 0x1 to enable
Set clock domain state transition feature	CM_<Clock Domain name>_CLKSTCTRL[1:0]11 CLKTRCTRL	xx <sup>(3)</sup>

- (1) See the module register for valid modes.
- (2) It depends on the desired size of the window. See [Section 3.1.1.1.7.2, Clock Domain Dependency](#).
- (3) It depends on the desired state of the clock domain. See [Table 3-11](#).

### 3.10.2.1.2.2 Subsequence – Slave Module Clock-Management Parameters Configuration

This procedure configures the SSC parameters for the DPLL and enables the SSC feature.

**Table 3-306. Slave Module Clock-Management Parameter Configuration**

Step	Register/Bit Field/Programming Model	Value
Configure module idle mode feature	<Module name>_SYSCONFIG[x] SIDLEMODE <Module name>_SYSCONFIG[x] IDLEMODE	xx <sup>(1)</sup>
<b>IF</b> : Smart-idle mode is selected	<Module name>_SYSCONFIG[x] SIDLEMODE <Module name>_SYSCONFIG[x] IDLEMODE	0x10
Configure module clock requirement feature	<Module name>_SYSCONFIG[x] CLOCKACTIVITY	x <sup>(1)</sup>
<b>ENDIF</b>		
Configure module management behavior on the PRCM module side	CM_<Clock Domain name>_<Module name>_CLKCTRL[1:0] MODULEMODE	xx <sup>(2)</sup>

- (1) See the module register for valid settings.
- (2) The selected value depends on the desired clock-management behavior.

### 3.10.2.2 Clock Domain Sleep Transition and Troubleshooting

This procedure initiates a sleep transition on a clock domain and allows to debug if the transition does not occur.

**Table 3-307. Clock Domain Sleep Transition and Troubleshooting**

Step	Register/Bit Field/Programming Model	Value
Set clock domain sleep transition state	CM_<Clock Domain name>_CLKSTCTRL[1:0]11 CLKTRCTRL	0x1: SW_SLEEP 0x3: HW_AUTO
<b>IF</b> : Clock domain sleep transition not initiated?		
Check all clock domain master modules are in standby mode	CM_<Clock Domain name>_<Module name>_CLKCTRL[18] STBYST	0x1: Module in standby
Check all clock domain slave modules are in idle mode	CM_<Clock Domain name>_<Module name>_CLKCTRL[17:16] IDLEST	0x1: In transition 0x2: Interface clock idled 0x3: Module idled
<b>ENDIF</b>		

### 3.10.2.3 Enable/Disable Software-Programmable Static Dependency

To change the setting of a software-programmable static dependency use the procedure described in [Table 3-308](#).

**Table 3-308. Enable/Disable Software-Programmable Static Dependency**

Step	Register/Bit Field/Programming Model	Value
Force destination domain to be awake (SW_WKUP)	CM_<Dest_CDname>_CLKSTCTRL[1:0] CLKTRCTRL	0x2
Wait until power domain which encloses the destination domain is ON	PM_<Dest_PDname>_PWRSTST	=0x3
Change the static dependency	CM_<Src_CDname>_STATICDEP[x] <Dest_CDname>_STATDEP	0x1
Put destination domain back to automatic transition (HW_AUTO)	CM_<Dest_CDname >_CLKSTCTRL[1:0] CLKTRCTRL	0x3



### 3.10.3 Power Management Low-Level Programming Models

This section covers the low-level hardware programming sequences for configuration and control of the power domain in the device.

#### 3.10.3.1 Global Initialization

##### 3.10.3.1.1 Surrounding Module Global Initialization

It does not require any initialization sequence of any surrounding modules within the device. The external power IC and the device clocks should be active.

##### 3.10.3.1.2 Power Management Global Initialization

###### 3.10.3.1.2.1 Main Sequence – Power Domain Global Initialization and Setting

This procedure initializes the power domain of the device after a power-on or software reset.

**Table 3-309. Power Domain Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Configure memory area power state when the power domain is on. Not all memory area states are programmable.	PM_<Power Domain name>_PWRSTCTRL[x] <Memory Bank name>_ONSTATE	0x0: Off 0x1: RETAINED 0x3: On
Configure memory area power state when the power domain transitions to RETENTION state. Not all memory area states are programmable.	PM_<Power Domain name>_PWRSTCTRL[x] <Memory Bank name>_RETSTATE	0x0: Off 0x1: RETAINED
Configure logic area RETENTION power state when the power domain transitions to RETENTION state.	PM_<Power Domain name>_PWRSTCTRL[2] LOGICRETSTATE	0x0: OSWR 0x1: CSWR
Select target power state of the power domain. Not all states are programmable.	PM_<Power Domain name>_PWRSTCTRL[1:0] POWERSTATE	0x0: OFF 0x1: RETENTION 0x2: ON-INACTIVE 0x3: ON-ACTIVE
Wait until power state change is complete	PM_<Power Domain name>_PWRSTST[1:0] POWERSTATEST	0x0: OFF 0x1: RETENTION 0x2: ON-INACTIVE 0x3: ON-ACTIVE

#### 3.10.3.2 Forced Memory Area State Change With Power Domain ON

This procedure initiates a forced memory area state change while the power domain is ON.

**Table 3-310. Forced Memory Area State Change With Power Domain ON**

Step	Register/Bit Field/Programming Model	Value
Configure memory area target power state. Not all memory area states are programmable.	PM_<Power Domain name>_PWRSTCTRL[x] <Memory Bank name>_ONSTATE	0x0: OFF 0x1: RETAINED 0x3: ON
Wait until memory area state change is complete?	PM_<Power Domain name>_PWRSTCTRL[3] MEMORYCHANGE	0x0: Change complete
Get memory area current state	PM_<Power Domain name>_PWRSTST[x] <Memory Bank name>_STATEST	0x0: OFF 0x1: RETAINED 0x3: ON

#### 3.10.3.3 Forced Power Domain Low-Power State Transition

This procedure initiates a power state transition on a power domain from current low-power state (ON-INACTIVE or RETENTION) to a lower power state (RETENTION, OFF).

**Table 3-311. Forced Memory Area State Change With Power Domain On**

Step	Register/Bit Field/Programming Model	Value
Select target low-power state of the power domain. Not all states are programmable.	PM_<Power Domain name>_PWRSTCTRL[1:0] POWERSTATE	0x0: Off 0x1: Retention
Force power domain low-power state transition.	PM_<Power Domain name>_PWRSTCTRL[4] LOWPOWERSTATECHANGE	0x1: Force change
Wait until state change is complete	PM_<Power Domain name>_PWRSTCTRL[4] LOWPOWERSTATECHANGE	0x0: Change complete
Get current power state.	PM_<Power Domain name>_PWRSTST[1:0] POWERSTATEST	0x0: Off 0x1: Retention

### 3.10.4 Voltage Management Low-Level Programming Models

This section covers the low-level hardware programming sequences for configuration and management of the voltage domains of the device.

#### 3.10.4.1 Global Initialization

##### 3.10.4.1.1 Surrounding Module Global Initialization

It does not require any initialization sequence of any surrounding modules within the device. The external power IC and the device clocks should be active.

##### 3.10.4.1.2 Voltage Management Global Initialization

###### 3.10.4.1.2.1 Main Sequence – Voltage Management Global Initialization and Setting

This procedure initializes the voltage management framework in the device.

**Table 3-312. Power Domain Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Initialize SmartReflex modules	See <a href="#">Section 3.10.4.1.3</a> .	
Initialize voltage processor modules	See <a href="#">Section 3.10.4.1.4</a> .	
Initialize voltage controller module	See <a href="#">Section 3.10.4.1.5</a> .	

##### 3.10.4.1.3 SmartReflex Module Initialization

This procedure initializes the SmartReflex module.

**Table 3-313. SmartReflex Module Configuration**

Step	Register/Bit Field/Programming Model	Value
Configure clock	See <a href="#">Section 3.10.4.1.3.1</a> .	
Configure sensor	See <a href="#">Section 3.10.4.1.3.2</a> .	
Configure accumulator and min/max/avg	See <a href="#">Section 3.10.4.1.3.3</a> .	
Configure error generator	See <a href="#">Section 3.10.4.1.3.4</a> .	
Configure interrupt generator	See <a href="#">Section 3.10.4.1.3.5</a> .	
Enable module	SRn.SRCONFIG[11] SR_EN	0x1

###### 3.10.4.1.3.1 Subsequence – Clock Configuration

This procedure configures the clock for the SmartReflex module.

**Table 3-314. SmartReflex Module Clock Configuration**

Step	Register/Bit Field/Programming Model	Value
Configure desired clock frequency	SRn.SRCONFIG[21:12] SRCLKLENGTH	xx <sup>(1)</sup>
Configure desired idle mode	SRn.ERRCONFIG[25:24] IDLEMODE	0x0: Forced-idle 0x1: No-idle 0x2: Smart-idle 0x3: Smart-idle Wakeup
Enable/disable wake-up feature	SRn.ERRCONFIG[26] WAKEUPENABLE	0x0: Disable 0x1: Enable
Enable clock	CM_ALWON_<SR name>_CLKCTRL[1:0] MODULEMODE	0x2: Enable

<sup>(1)</sup> See [Equation 1](#) for clock frequency calculation.

#### 3.10.4.1.3.2 Subsequence – Sensor Configuration

This procedure configures the sensor in the SmartReflex module.

**Table 3-315. SmartReflex Module Sensor Configuration**

Step	Register/Bit Field/Programming Model	Value
Enable sensor core A	SRn.SRCONFIG[1] SENENABLE	0x1
Enable sensor core B	SRn.SRCONFIG[0] SENPENABLE	0x1
Enable sensor core	SRn.SRCONFIG[10] SENENABLE	0x1

#### 3.10.4.1.3.3 Subsequence – Accumulator and Min/Max/Avg Configuration

This procedure configures the accumulator and min/max/avg part of the SmartReflex module.

**Table 3-316. Accumulator and Min/Max/Avg Configuration**

Step	Register/Bit Field/Programming Model	Value
Set size of the accumulator	SRn.SRCONFIG[31:22] ACCUMDATA	xxx <sup>(1)</sup>
Set average weight parameter for the sensor A	SRn.AVGWEIGHT[1:0] SENNAVGWEIGHT	x <sup>(1)</sup>
Set average weight parameter for the sensor B	SRn.AVGWEIGHT[3:2] SENPAVGWEIGHT	x <sup>(1)</sup>
Enable minimum/maximum/average	SRn.SRCONFIG[8] MINMAXAVGENABLE	0x1

<sup>(1)</sup> These parameters are calculated as given in [Section 3.8.3.2.3, SmartReflex Submodules](#).

#### 3.10.4.1.3.4 Subsequence – Error Generator Configuration

This procedure configures the error generator part of the SmartReflex module.

**Table 3-317. Error Generator Configuration**

Step	Register/Bit Field/Programming Model	Value
Configure four reference value parameters	SRn.NVALUERECIPROCAL[23:20] SENPGAIN	x <sup>(1)</sup>
	SRn.NVALUERECIPROCAL[19:16] SENNGAIN	x <sup>(1)</sup>
	SRn.NVALUERECIPROCAL[15:8] SENPRN	xx <sup>(1)</sup>
	SRn.NVALUERECIPROCAL[7:0] SENNRN	xx <sup>(1)</sup>
Enable the error generator	SRn.SRCONFIG[9] ERRORGENERATORENABLE	0x1

<sup>(1)</sup> Configured according to the values read for the current operating voltage level from the control module register in [Section 3.8.3.2.3, SmartReflex Submodules](#).

#### 3.10.4.1.3.5 Subsequence – Interrupt Generator Configuration

This procedure configures the interrupt generator part of the SmartReflex module.

**Table 3-318. Interrupt Generator Configuration**

Step	Register/Bit Field/Programming Model	Value
Set the four reference value parameters	SRn.ERRCONFIG[18:16] ERRWEIGHT	x
	SRn.ERRCONFIG[15:8] ERRMAXLIMIT	xx
	SRn.ERRCONFIG[7:0] ERRMINLIMIT	xx
<b>IF</b> : Automatic voltage control using voltage processor	Software test condition	
Clear voltage processor bounds interrupt status	SRn.ERRCONFIG[23] VPBOUNDSINTSTATENA	0x1
Unmask voltage processor bounds interrupt	SRn.ERRCONFIG[22] VPBOUNDSINTENABLE	0x1
<b>ELSE-IF</b> : Software-controlled voltage management	Software test condition	
Unmask <Processor> interrupt events	SRn.IRQENABLE_SET[3] MCUACCUMINTENASET	0x1
	SRn.IRQENABLE_SET[2] MCUVALIDINTENASET	0x1
	SRn.IRQENABLE_SET[1] MCUBOUNDSINTENASET	0x1
<b>ENDIF</b>		

#### 3.10.4.1.4 Voltage Processor Initialization

This procedure initializes the voltage processor.

**Table 3-319. Voltage Processor Module Initialization**

Step	Register/Bit Field/Programming Model	Value
Configure error-to-voltage converter	See <a href="#">Section 3.10.4.1.4.1</a> .	
Configure PRCM	See <a href="#">Section 3.10.4.1.4.2</a> .	
Mask/unmask interrupt events	PRCM.PRM_IRQENABLE_<Processor name>	
Enable module	PRCM.PRM_<Voltage Processor name>_CONFIG[0] VPENABLE	0x1

##### 3.10.4.1.4.1 Subsequence – Error-to-Voltage Converter Configuration

This procedure configures the error-to-voltage converter of the voltage processor.

**Table 3-320. Error-to-Voltage Converter Configuration**

Step	Register/Bit Field/Programming Model	Value
Set error offset	PRCM.PRM_<Voltage Processor name>_CONFIG[31:24] ERROROFFSET	xx <sup>(1)</sup>
Set error gain	PRCM.PRM_<Voltage Processor name>_CONFIG[23:16] ERRORGAIN	xx <sup>(1)</sup>
Set SMPS negative voltage step slew rate	PRCM.PRM_<Voltage Processor name>_VSTEPMIN[23:8] SMPSWAITTIMEMIN	xx <sup>(1)</sup>
Set SMPS minimum voltage step size	PRCM.PRM_<Voltage Processor name>_VSTEPMIN[7:0] VSTEPMIN	xx <sup>(1)</sup>
Set SMPS positive voltage step slew rate	PRCM.PRM_<Voltage Processor name>_VSTEPMAX[23:8] SMPSWAITTIMEMAX	xx <sup>(1)</sup>
Set SMPS maximum voltage step size	PRCM.PRM_<Voltage Processor name>_VSTEPMAX[7:0] VSTEPMAX	xx <sup>(1)</sup>
Set SMPS minimum voltage limit	PRCM.PRM_<Voltage Processor name>_VLIMITTO[23:16] VDDMIN	xx <sup>(1)</sup>
Set SMPS maximum voltage limit	PRCM.PRM_<Voltage Processor name>_VLIMITTO[31:24] VDDMAX	xx <sup>(1)</sup>

(1) Depend on the characteristics of the SMPS.

#### 3.10.4.1.4.2 Subsequence – FSM Configuration

This procedure configures the FSM of the voltage processor.

**Table 3-321. FSM Configuration**

Step	Register/Bit Field/Programming Model	Value
Set maximum length of time to wait for the command acknowledge from the SMPS	PRCM.PRM_<Voltage Processor name>_VLIMITTO[15:0] TIMEOUT	xxxx <sup>(1)</sup>
Set target initial voltage level of the SMPS	PRCM.PRM_<Voltage Processor name>_CONFIG[15:8] INITVOLTAGE	xx
Assert the initial voltage into the voltage processor FSM	PRCM.PRM_<Voltage Processor name>_CONFIG[2] INITVDD	0x1

(1) Depend on the characteristics of the SMPS.

#### 3.10.4.1.5 Voltage Controller Initialization

This procedure initializes the voltage controller.

**Table 3-322. Voltage Controller Module Initialization**

Step	Register/Bit Field/Programming Model	Value
Configure power IC slave addresses	PRCM.PRM_VC_SMPS_SA[x] SA_<Voltage Domain name>	xx <sup>(1)</sup>
Configure voltage configuration register addresses	PRCM.PRM_VC_VAL_SMPS_RA_VOL[x] VOLRA_<Voltage Domain name>	xx <sup>(1)</sup>
Configure command configuration register addresses	PRCM.PRM_VC_VAL_SMPS_RA_CMD[x] CMDRA_<Voltage Domain name>	xx <sup>(1)</sup>
Configure voltage domain command values	PRCM.PRM_VC_VAL_CMD_<Voltage Domain name>[x] <Command name>	xx <sup>(1)</sup>
Configure pointers for the VDD channels	See <a href="#">Section 3.10.4.1.5.1</a> .	
Configure I <sup>2</sup> C interface	See <a href="#">Section 3.10.4.1.5.2</a> .	

(1) Depend on the characteristics of the SMPS.

#### 3.10.4.1.5.1 Subsequence – VDD Channels Pointers Configuration

This procedure configures the FSM of the voltage processor.

**Table 3-323. VDD Channel Pointer Configuration**

Step	Register/Bit Field/Programming Model	Value
Set slave address pointer	PRCM.PRM_VC_CFG_CHANNEL[x] SA_<Voltage Domain name>	
Set voltage configuration register address pointer	PRCM.PRM_VC_CFG_CHANNEL[x] RAV_<Voltage Domain name>	
Set command configuration register address pointer	PRCM.PRM_VC_CFG_CHANNEL[x] RAC_<Voltage Domain name>	
Select voltage or command configuration register for FSM commands	PRCM.PRM_VC_CFG_CHANNEL[x] RACEN_<Voltage Domain name>	
Set command voltage level selection pointer	PRCM.PRM_VC_CFG_CHANNEL[x] CMD_<Voltage Domain name>	

(1) Depend on the connectivity characteristics of the SMPS.

#### 3.10.4.1.5.2 Subsequence – I<sup>2</sup>C Configuration

This procedure configures the I<sup>2</sup>C in the voltage controller.

**Table 3-324. I<sup>2</sup>C Configuration**

Step	Register/Bit Field/Programming Model	Value
Enable/disable I <sup>2</sup> C bus double digital filter	PRCM.PRM_VC_CFG_I2C_MODE[6] DFILTEREN	
Set master code value for I <sup>2</sup> C HS preamble	PRCM.PRM_VC_CFG_I2C_MODE[2:0] HSMCODE	
Enable/disable I <sup>2</sup> C HS mode	PRCM.PRM_VC_CFG_I2C_MODE[3] HSMODEEN	0x0: Disable 0x1: Enable
Enable/disable repeated start operation.	PRCM.PRM_VC_CFG_I2C_MODE[4] SRMODEEN	0x0: Disable 0x1: Enable

### 3.10.4.2 Changing OPP

This procedure allows changing of the voltage of the external SMPS from one OPP to the other. It also covers the programming sequence for ABB LDO if the OPP change requires a change in state of the ABB LDO.

**NOTE:** When switching from the current OPP to a target OPP, if the clock frequency of the target OPP is less than that of the current OPP, the clock frequency is first switched to the lower frequency of target OPP, and only then is the voltage scaling initiated. However, if the clock frequency of the target OPP is greater than that of the current OPP, voltage scaling is initiated before frequency scaling.

**Table 3-325. Changing OPP Using Voltage-Processor Module**

Step	Register/Bit Field/Programming Model	Value
Disable the SmartReflex module	SRn.SRCONFIG[11] SR_EN	0x0
Mask SmartReflex interrupts		
Disable the voltage processor module	PRCM.PRM_<Voltage Processor name>_CONFIG[0] VPENABLE	0x0
Configured according to the target OPP voltage value	PRCM.PRM_<Voltage Processor name>_CONFIG[15:8] INITVOLTAGE	
<b>IF:</b> ABB LDO state change is necessary	Software test (See <a href="#">Section 3.8.4.3</a> )	
Set the desired state of the ABB LDO	PRCM.PRM_LDO_ABB_<MPU or DSP>_SETUP[x] ACTIVE_RBB_SEL or ACTIVE_FBB_SEL	0x0/0x1
Enable/bypass the ABB power management of the ABB LDO	PRCM.PRM_LDO_ABB_<MPU or DSP>_SETUP[x] SR2EN	0x0/0x1
Select the next OPP for the ABB LDO	PRM_LDO_ABB_<MPU or DSP>_CTRL[1:0] OPP_SEL	
Initiate the state change for the ABB LDO	PRM_LDO_ABB_<MPU or DSP>_CTRL[2] OPP_CHANGE	0x1
Wait until ABB LDO state transition completes	PRM_LDO_ABB_<MPU or DSP>_CTRL[6] SR2_IN_TRANSITION	0x0
<b>ENDIF</b>		
Triggers a write of the value in the INITVOLTAGE into the voltage processor	PRCM.PRM_<Voltage Processor name>_CONFIG[2] INITVDD	0x1
Force update of the SMPS	PRCM.PRM_<Voltage Processor name>_CONFIG[1] FORCEUPDATE	0x1

## 3.11 PRCM Register Manual

### 3.11.1 PRM Instance Summary

**Table 3-326. PRM Instance Summary**

Module Name	L4 Base Address	Size
INTRCONN_SOCKET_PRM	0x4A30 6000	256Bytes
CKGEN_PRM	0x4A30 6100	256Bytes
MPU_PRM	0x4A30 6300	256Bytes
DSP_PRM	0x4A30 6400	256Bytes
ABE_PRM	0x4A30 6500	256Bytes
ALWAYS_ON_PRM	0x4A30 6600	256Bytes
CORE_PRM	0x4A30 6700	2KBytes
IWAHD_PRM	0x4A30 6F00	256Bytes
CAM_PRM	0x4A30 7000	256Bytes
DSS_PRM	0x4A30 7100	256Bytes
SGX_PRM	0x4A30 7200	256Bytes
L3INIT_PRM	0x4A30 7300	256Bytes
L4PER_PRM	0x4A30 7400	512Bytes
WKUP_PRM	0x4A30 7700	256Bytes
WKUP_CM	0x4A30 7800	256Bytes
EMU_PRM	0x4A30 7900	256Bytes
EMU_CM	0x4A30 7A00	256Bytes
DEVICE_PRM	0x4A30 7B00	256Bytes
INSTR_PRM	0x4A30 7F00	256Bytes

### 3.11.2 INTRCONN\_SOCKET\_PRM Registers

#### 3.11.2.1 INTRCONN\_SOCKET\_PRM Register Summary

**Table 3-327. INTRCONN\_SOCKET\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	INTRCONN_SOCKET_PRM L4 Base Address
REVISION_PRM	R	32	0x0000 0000	0x4A30 6000
PRM_IRQSTATUS_MPU_A9	RW	32	0x0000 0010	0x4A30 6010
PRM_IRQSTATUS_MPU_A9_2	RW	32	0x0000 0014	0x4A30 6014
PRM_IRQENABLE_MPU_A9	RW	32	0x0000 0018	0x4A30 6018
PRM_IRQENABLE_MPU_A9_2	RW	32	0x0000 001C	0x4A30 601C
PRM_IRQSTATUS_MPU_M3	RW	32	0x0000 0020	0x4A30 6020
PRM_IRQENABLE_MPU_M3	RW	32	0x0000 0028	0x4A30 6028
PRM_IRQSTATUS_DSP	RW	32	0x0000 0030	0x4A30 6030
PRM_IRQENABLE_DSP	RW	32	0x0000 0038	0x4A30 6038
CM_PRM_PROFILING_CLKCTRL	RW	32	0x0000 0040	0x4A30 6040
PRM_DEBUG_CFG	RW	32	0x0000 00F0	0x4A30 60F0

#### 3.11.2.2 INTRCONN\_SOCKET\_PRM Register Description

**Table 3-328. REVISION\_PRM**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 3-327</a>	<b>Instance</b> INTRCONN_SOCKET_PRM
<b>Description</b>	This register contains the IP revision code for the PRM part of the PRCM	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															

Bits	Field Name	Description	Type	Reset
31:0	REV	Revision Number	R	0x- - TI Internal data

**Table 3-329. Register Call Summary for Register REVISION\_PRM**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[0\]](#)

**Table 3-330. PRM\_IRQSTATUS\_MPU\_A9**

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Table 3-327</a>	<b>Instance</b> INTRCONN_SOCKET_PRM
<b>Description</b>	This register provides status on MPU interrupt events. Any event is logged independently of the corresponding IRQENABLE value. Software is required to clear a set bit by writing a 1 into the bit-position to be cleared.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_IVA_DONE_ST	VC_IVA_VPACK_ST	VP_IVA_TRANXDONE_ST	VP_IVA_EQVALUE_ST	VP_IVA_NOSMPSACK_ST	VP_IVA_MAXVDD_ST	VP_IVA_MINVDD_ST	VP_IVA_OPPOCHANGEDONE_ST	RESERVED	VC_CORE_VPACK_ST	VP_CORE_TRANXDONE_ST	VP_CORE_EQVALUE_ST	VP_CORE_NOSMPSACK_ST	VP_CORE_MAXVDD_ST	VP_CORE_MINVDD_ST	VP_CORE_OPPOCHANGEDONE_ST	RESERVED	VC_BYPASSACK_ST	VC_TOERR_ST	VC_RAERR_ST	VC_SAERR_ST	RESERVED	IO_ST	TRANSITION_ST	RESERVED	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST		

Bits	Field Name	Description	Type	Reset
31	ABB_IVA_DONE_ST	IVA ABB mode change done. This status is set when OPP_CHANGE bit is cleared by hardware in PRM_LDO_ABB_IVA_CRTL register. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
30	VC_IVA_VPACK_ST	Voltage Controller IVA voltage processor command acknowledge status.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0



Bits	Field Name	Description	Type	Reset
29	VP_IVA_TRANXDONE_ST	Voltage Processor IVA transaction completion status. This status is set when a transaction is completed in the voltage processor, including ABB mode change done if applicable (OPP_CHANGE bit cleared in PRM_LDO_ABB_IVA_CTRL). It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
28	VP_IVA_EQVALUE_ST	Voltage Processor IVA voltage value change event. This status is set when an update has been requested but the new voltage value is the same as the current SMPS voltage value. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
27	VP_IVA_NOSMPSACK_ST	Voltage Processor IVA timeout event status. This status is set when the timeout occurred before the SMPS acknowledge. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
26	VP_IVA_MAXVDD_ST	Voltage Processor IVA voltage higher limit event status. This status is set when the voltage higher limit is reached. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
25	VP_IVA_MINVDD_ST	Voltage Processor IVA voltage lower limit event status. This status is set when the voltage lower limit is reached. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
24	VP_IVA_OPPCHANGEDONE_ST	Voltage Processor IVA OPP change done status, including ABB mode change done if applicable (OPP_CHANGE bit cleared in PRM_LDO_ABB_IVA_CTRL). It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
23	RESERVED		R	0
22	VC_CORE_VPACK_ST	Voltage Controller CORE voltage processor command acknowledge status  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
21	VP_CORE_TRANXDONE_ST	Voltage Processor CORE transaction completion status. This status is set when a transaction is completed in the voltage processor. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
20	VP_CORE_EQVALUE_ST	Voltage Processor CORE voltage value change event. This status is set when an update has been requested but the new voltage value is the same as the current SMPS voltage value. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
19	VP_CORE_NOSMPSACK_ST	Voltage Processor CORE timeout event status. This status is set when the timeout occurred before the SMPS acknowledge. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
18	VP_CORE_MAXVDD_ST	Voltage Processor CORE voltage higher limit event status. This status is set when the voltage higher limit is reached. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
17	VP_CORE_MINVDD_ST	Voltage Processor CORE voltage lower limit event status. This status is set when the voltage lower limit is reached. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
16	VP_CORE_OPPCHANGEDONE_ST	Voltage Processor CORE OPP change done status. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
15	Reserved		R	0
14	VC_BYPASSACK_ST	Voltage Controller bypass command acknowledge status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
13	VC_TOERR_ST	Voltage Controller timeout error event status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
12	VC_RAERR_ST	Voltage Controller register address acknowledge error event status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
11	VC_SAERR_ST	Voltage Controller slave address acknowledge error event status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0x0
10	RESERVED		R	0
9	IO_ST	I/O pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
8	TRANSITION_ST	Software supervised transition completed event interrupt status (any domain). Asserted upon completion of any clock domain force wakeup transition or upon completion of any power domain sleep transition with at least one enclosed clock domain configured in forced-sleep. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
7:5	RESERVED		R	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0

**Table 3-331. Register Call Summary for Register PRM\_IRQSTATUS\_MPU\_A9**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\]](#)
- [DPLL\\_CORE Description: \[1\]](#)
- [DPLL\\_ABE Description: \[2\]](#)
- [DPLL\\_MPU Description: \[3\]](#)
- [DPLL\\_IVA Description: \[4\]](#)

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[5\]](#)

**Table 3-332. PRM\_IRQSTATUS\_MPU\_A9\_2**

<b>Address Offset</b>	0x0000 0014	
<b>Physical Address</b>	See <a href="#">Table 3-327</a>	<b>Instance</b> INTRCONN_SOCKET_PRM
<b>Description</b>	This register provides status on MPU interrupt events. Any event is logged independently of the corresponding IRQENABLE value. Software is required to clear a set bit by writing a 1 into the bit-position to be cleared.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ABB_MPU_DONE_ST	VC_MPU_VPACK_ST	VP_MPU_TRANXDONE_ST	VP_MPU_EQVALUE_ST	VP_MPU_NOSMPSACK_ST	VP_MPU_MAXVDD_ST	VP_MPU_MINVDD_ST	VP_MPU_OPPCHANGEDONE_ST								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	ABB_MPU_DONE_ST	MPU ABB mode change done. This status is set when OPP_CHANGE bit is cleared by hardware in PRM_LDO_ABB_MPU_CRTL register. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
6	VC_MPU_VPACK_ST	Voltage Controller MPU voltage processor command acknowledge status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
5	VP_MPU_TRANXDONE_ST	Voltage Processor MPU transaction completion status. This status is set when a transaction is completed in the voltage processor, including ABB mode change done if applicable (OPP_CHANGE bit cleared in PRM_LDO_ABB_MPU_CRTL). It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
4	VP_MPU_EQVALUE_ST	Voltage Processor MPU voltage value change event. This status is set when an update has been requested but the new voltage value is the same as the current SMPS voltage value. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
3	VP_MPU_NOSMPSACK_ST	Voltage Processor MPU timeout event status. This status is set when the timeout occurred before the SMPS acknowledge. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
2	VP_MPU_MAXVDD_ST	Voltage Processor MPU voltage higher limit event status. This status is set when the voltage higher limit is reached. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
1	VP_MPU_MINVDD_ST	Voltage Processor MPU voltage lower limit event status. This status is set when the voltage lower limit is reached. It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
0	VP_MPU_OPPCHANGEDONE_ST	Voltage Processor MPU OPP change done status, including ABB mode change done if applicable (OPP_CHANGE bit cleared in PRM_LDO_ABB_MPU_CRTL). It is cleared by software.  0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0

**Table 3-333. Register Call Summary for Register PRM\_IRQSTATUS\_MPU\_A9\_2**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[0\]](#)

**Table 3-334. PRM\_IRQENABLE\_MPU\_A9**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	INTRCONN_SOCKET_PRM
<b>Physical Address</b>	See <a href="#">Table 3-327</a>		
<b>Description</b>	This register is used to enable or disable MPU interrupt activation upon presence of corresponding <a href="#">IRQSTATUS</a> bit.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_IVA_DONE_EN	VC_IVA_VPACK_EN	VP_IVA_TRANXDONE_EN	VP_IVA_EQVALUE_EN	VP_IVA_NOSMPSACK_EN	VP_IVA_MAXVDD_EN	VP_IVA_MINVDD_EN	VP_IVA_OPPCHANGEDONE_EN	RESERVED	VC_CORE_VPACK_EN	VP_CORE_TRANXDONE_EN	VP_CORE_EQVALUE_EN	VP_CORE_NOSMPSACK_EN	VP_CORE_MAXVDD_EN	VP_CORE_MINVDD_EN	VP_CORE_OPPCHANGEDONE_EN	RESERVED	VC_BYPASSACK_EN	VC_TOERR_EN	VC_RAERR_EN	VC_SAERR_EN	RESERVED	IO_EN	TRANSITION_EN	RESERVED	RESERVED	RESERVED	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_IVA_DONE_EN	IIVA ABB mode change done enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
30	VC_IVA_VPACK_EN	Voltage Controller IVA voltage processor command acknowledge enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
29	VP_IVA_TRANXDONE_EN	Voltage Processor IVA transaction completion enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
28	VP_IVA_EQVALUE_EN	Voltage Processor IVA voltage value change event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
27	VP_IVA_NOSMPSACK_EN	Voltage Processor IVA timeout event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
26	VP_IVA_MAXVDD_EN	Voltage Processor IVA voltage higher limit event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
25	VP_IVA_MINVDD_EN	Voltage Processor IVA voltage lower limit event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
24	VP_IVA_OPPCHANGEDONE_EN	Voltage Processor IVA OPP change done enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
23	RESERVED		R	0
22	VC_CORE_VPACK_EN	Voltage Controller CORE voltage processor command acknowledge enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
21	VP_CORE_TRANXDONE_EN	Voltage Processor CORE transaction completion enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0

Bits	Field Name	Description	Type	Reset
20	VP_CORE_EQVALUE_EN	Voltage Processor CORE voltage value change event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
19	VP_CORE_NOSMPSACK_EN	Voltage Processor CORE timeout event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
18	VP_CORE_MAXVDD_EN	Voltage Processor CORE voltage higher limit event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
17	VP_CORE_MINVDD_EN	Voltage Processor CORE voltage lower limit event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
16	VP_CORE_OPPCHANGEDONE_EN	Voltage Processor CORE OPP change done enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
15	RESERVED		R	0
14	VC_BYPASSACK_EN	Voltage Controller bypass command acknowledge enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
13	VC_TOERR_EN	Voltage Controller timeout error event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
12	VC_RAERR_EN	Voltage Controller register address acknowledge error event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
11	VC_SAERR_EN	Voltage Controller slave address acknowledge error event enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	RESERVED		R	0
9	IO_EN	I/O pad event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
8	TRANSITION_EN	Software supervised transition completed event interrupt enable (any domain). 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
7:5	RESERVED		R	0x0
4	DPLL_ABE_RECAL_EN	ABE DPLL recalibration interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0

Bits	Field Name	Description	Type	Reset
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0

**Table 3-335. Register Call Summary for Register PRM\_IRQENABLE\_MPU\_A9**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\]](#)
- [DPLL\\_CORE Description: \[1\]](#)
- [DPLL\\_ABE Description: \[2\]](#)
- [DPLL\\_MPU Description: \[3\]](#)
- [DPLL\\_IVA Description: \[4\]](#)

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[5\]](#)

**Table 3-336. PRM\_IRQENABLE\_MPU\_A9\_2**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	INTRCONN_SOCKET_PRM
<b>Physical Address</b>	See <a href="#">Table 3-327</a>		
<b>Description</b>	This register is used to enable or disable MPU interrupt activation upon presence of corresponding <a href="#">IRQSTATUS</a> bit.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ABB_MPU_DONE_EN VC_MPU_VPACK_EB VP_MPU_TRANXDONE_EN VP_MPU_EQVALUE_EN VP_MPU_NOSMPSACK_EN VP_MPU_MAXVDD_EN VP_MPU_MINVDD_EN VP_MPU_OPPCHANGEDONE_EN															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x00000000
7	ABB_MPU_DONE_EN	MPU ABB mode change done enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
6	VC_MPU_VPACK_EB	Voltage Controller MPU voltage processor command acknowledge enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0

Bits	Field Name	Description	Type	Reset
5	VP_MPU_TRANXDONE_EN	Voltage Processor MPU transaction completion enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
4	VP_MPU_EQVALUE_EN	Voltage Processor MPU voltage value change event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
3	VP_MPU_NOSMPSACK_EN	Voltage Processor MPU timeout event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
2	VP_MPU_MAXVDD_EN	Voltage Processor MPU voltage higher limit event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
1	VP_MPU_MINVDD_EN	Voltage Processor MPU voltage lower limit event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
0	VP_MPU_OPPCHANGEDONE_EN	Voltage Processor MPU OPP change done enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0

**Table 3-337. Register Call Summary for Register PRM\_IRQENABLE\_MPU\_A9\_2**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[0\]](#)

**Table 3-338. PRM\_IRQSTATUS\_MPU\_M3**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	INTRCONN_SOCKET_PRM
<b>Physical Address</b>	See <a href="#">Table 3-327</a>		
<b>Description</b>	This register provides status on MPU_A3 interrupt events. Any event is logged independently of the corresponding IRQENABLE value. Software is required to clear a set bit by writing a 1 into the bit-position to be cleared.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_IVA_DONE_ST	VC_IVA_VPACK_ST	VP_IVA_TRANXDONE_ST	VP_IVA_EQVALUE_ST	VP_IVA_NOSMPSACK_ST	VP_IVA_MAXVDD_ST	VP_IVA_MINVDD_ST	VP_IVA_OPPCHANGEDONE_ST	RESERVED	VC_CORE_VPACK_ST	VP_CORE_TRANXDONE_ST	VP_CORE_EQVALUE_ST	VP_CORE_NOSMPSACK_ST	VP_CORE_MAXVDD_ST	VP_CORE_MINVDD_ST	VP_CORE_OPPCHANGEDONE_ST	RESERVED	VC_BYPASSACK_ST	VC_TOERR_ST	VC_RAERR_ST	VC_SAERR_ST	FORCEWKUP_ST	IO_ST	TRANSITION_ST	RESERVED	RESERVED	RESERVED	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	RESERVED	DPLL_CORE_RECAL_ST



Bits	Field Name	Description	Type	Reset
31	ABB_IVA_DONE_ST	IVA ABB mode change done. This status is set when OPP_CHANGE bit is cleared by hardware in PRM_LDO_ABB_IVA_CRTL register. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
30	VC_IVA_VPACK_ST	Voltage Controller IVA voltage processor command acknowledge status 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
29	VP_IVA_TRANXDONE_ST	Voltage Processor IVA transaction completion status. This status is set when a transaction is completed in the voltage processor, including ABB mode change done if applicable (OPP_CHANGE bit cleared in PRM_LDO_ABB_IVA_CRTL). It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
28	VP_IVA_EQVALUE_ST	Voltage Processor IVA voltage value change event. This status is set when an update has been requested but the new voltage value is the same as the current SMPS voltage value. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
27	VP_IVA_NOSMPSACK_ST	Voltage Processor IVA timeout event status. This status is set when the timeout occurred before the SMPS acknowledge. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
26	VP_IVA_MAXVDD_ST	Voltage Processor IVA voltage higher limit event status. This status is set when the voltage higher limit is reached. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
25	VP_IVA_MINVDD_ST	Voltage Processor IVA voltage lower limit event status. This status is set when the voltage lower limit is reached. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
24	VP_IVA_OPPCHANGEDONE_ST	Voltage Processor IVA OPP change done status, including ABB mode change done if applicable (OPP_CHANGE bit cleared in PRM_LDO_ABB_IVA_CRTL). It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
23	RESERVED		R	0
22	VC_CORE_VPACK_ST	Voltage Controller CORE voltage processor command acknowledge status 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
21	VP_CORE_TRANXDONE_ST	Voltage Processor CORE transaction completion status. This status is set when a transaction is completed in the voltage processor. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
20	VP_CORE_EQVALUE_ST	Voltage Processor CORE voltage value change event. This status is set when an update has been requested but the new voltage value is the same as the current SMPS voltage value. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
19	VP_CORE_NOSMPSACK_ST	Voltage Processor CORE timeout event status. This status is set when the timeout occurred before the SMPS acknowledge. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
18	VP_CORE_MAXVDD_ST	Voltage Processor CORE voltage higher limit event status. This status is set when the voltage higher limit is reached. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
17	VP_CORE_MINVDD_ST	Voltage Processor CORE voltage lower limit event status. This status is set when the voltage lower limit is reached. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
16	VP_CORE_OPPCHANGEDONE_ST	Voltage Processor CORE OPP change done status. It is cleared by software. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
15	RESERVED		R	0
14	VC_BYPASSACK_ST	Voltage Controller bypass command acknowledge status 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
13	VC_TOERR_ST	Voltage Controller timeout error event status 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
12	VC_RAERR_ST	Voltage Controller register address acknowledge error event status 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
11	VC_SAERR_ST	Voltage Controller slave address acknowledge error event status 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
10	FORCEWKUP_ST	CORTEXM3 domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
9	IO_ST	I/O pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
8	TRANSITION_ST	Software supervised transition completed event interrupt status (any domain). Asserted upon completion of any clock domain force wakeup transition or upon completion of any power domain sleep transition with at least one enclosed clock domain configured in forced-sleep. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
1	RESERVED		R	0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0

**Table 3-339. Register Call Summary for Register PRM\_IRQSTATUS\_MPU\_M3**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\]](#)
- [DPLL\\_CORE Description: \[1\]](#)
- [DPLL\\_IVA Description: \[2\]](#)

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[15\]](#)

**Table 3-340. PRM\_IRQENABLE\_MPU\_M3**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	INTRCONN_SOCKET_PRM
<b>Physical Address</b>	See <a href="#">Table 3-327</a>		
<b>Description</b>	This register is used to enable or disable MPU_A3 interrupt activation upon presence of corresponding <a href="#">IRQSTATUS</a> bit.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_IVA_DONE_EN	VC_IVA_VPACK_EN	VP_IVA_TRANXDONE_EN	VP_IVA_EQVALUE_EN	VP_IVA_NOSMPSACK_EN	VP_IVA_MAXVDD_EN	VP_IVA_MINVDD_EN	VP_IVA_OPPCHANGEDONE_EN	RESERVED	VC_CORE_VPACK_EN	VP_CORE_TRANXDONE_EN	VP_CORE_EQVALUE_EN	VP_CORE_NOSMPSACK_EN	VP_CORE_MAXVDD_EN	VP_CORE_MINVDD_EN	VP_CORE_OPPCHANGEDONE_EN	RESERVED	VC_BYPASSACK_EN	VC_TOERR_EN	VC_RAERR_EN	VC_SAERR_EN	FORCEWKUP_EN	IO_EN	TRANSITION_EN	RESERVED	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	RESERVED	DPLL_CORE_RECAL_EN		

Bits	Field Name	Description	Type	Reset
31	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0

Bits	Field Name	Description	Type	Reset
30	VC_IVA_VPACK_EN	Voltage Controller IVA voltage processor command acknowledge enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
29	VP_IVA_TRANXDONE_EN	Voltage Processor IVA transaction completion enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
28	VP_IVA_EQVALUE_EN	Voltage Processor IVA voltage value change event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
27	VP_IVA_NOSMPSACK_EN	Voltage Processor IVA timeout event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
26	VP_IVA_MAXVDD_EN	Voltage Processor IVA voltage higher limit event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
25	VP_IVA_MINVDD_EN	Voltage Processor IVA voltage lower limit event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
24	VP_IVA_OPPCHANGEDONE_EN	Voltage Processor IVA OPP change done enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
23	RESERVED		R	0
22	VC_CORE_VPACK_EN	Voltage Controller CORE voltage processor command acknowledge enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
21	VP_CORE_TRANXDONE_EN	Voltage Processor CORE transaction completion enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
20	VP_CORE_EQVALUE_EN	Voltage Processor CORE voltage value change event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
19	VP_CORE_NOSMPSACK_EN	Voltage Processor CORE timeout event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
18	VP_CORE_MAXVDD_EN	Voltage Processor CORE voltage higher limit event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
17	VP_CORE_MINVDD_EN	Voltage Processor CORE voltage lower limit event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
16	VP_CORE_OPPCHANGEDONE_EN	Voltage Processor CORE OPP change done enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
15	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
14	VC_BYPASSACK_EN	Voltage Controller bypass command acknowledge enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
13	VC_TOERR_EN	Voltage Controller timeout error event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
12	VC_RAERR_EN	Voltage Controller register address acknowledge error event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
11	VC_SAERR_EN	Voltage Controller slave address acknowledge error event enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
10	FORCEWKUP_EN	CORTEXM3 domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
9	IO_EN	I/O pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
8	TRANSITION_EN	Software supervised transition completed event interrupt enable (any domain) 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
7:5	RESERVED		R	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
1	RESERVED		R	0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0

**Table 3-341. Register Call Summary for Register PRM\_IRQENABLE\_MPU\_M3**

## Clock Management Functional Description

- [DPLL\\_PER Description: \[0\]](#)
- [DPLL\\_CORE Description: \[1\]](#)
- [DPLL\\_IVA Description: \[2\]](#)

## Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

## PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[15\]](#)

**Table 3-342. PRM\_IRQSTATUS\_DSP**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 3-327</a>	<b>Instance</b>	INTRCONN_SOCKET_PRM
<b>Description</b>	This register provides status on DSP interrupt events. Any event is logged independently of the corresponding IRQENABLE value. Software is required to clear a set bit by writing a 1 into the bit-position to be cleared.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FORCEWKUP_ST	RESERVED				DPLL_ABE_RECAL_ST	RESERVED	DPLL_IVA_RECAL_ST	RESERVED							

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10	FORCEWKUP_ST	DSP domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
9:5	RESERVED		R	0x00
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
3	RESERVED		R	0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW W1toClr	0
1:0	RESERVED		R	0x0

**Table 3-343. Register Call Summary for Register PRM\_IRQSTATUS\_DSP**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[0\]](#)

**Table 3-344. PRM\_IRQENABLE\_DSP**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-327</a>	<b>Instance</b>	INTRCONN_SOCKET_PRM
<b>Description</b>	This register is used to enable or disable DSP interrupt activation upon presence of corresponding <a href="#">IRQSTATUS</a> bit.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FORCEWKUP_EN	RESERVED								DPLL_ABE_RECAL_EN	RESERVED	DPLL_IVA_RECAL_EN	RESERVED			

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10	FORCEWKUP_EN	DSP domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
9:5	RESERVED		R	0x00
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
3	RESERVED		R	0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
1:0	RESERVED		R	0x0

**Table 3-345. Register Call Summary for Register PRM\_IRQENABLE\_DSP**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[0\]](#)

**Table 3-346. CM\_PRM\_PROFILING\_CLKCTRL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	INTRCONN_SOCKET_PRM
<b>Physical Address</b>	See <a href="#">Table 3-327</a>		
<b>Description</b>	This register manages the PRM_PROFILING clock. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED								MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status Read 0x0: Module is fully functional Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle Read 0x3: Module is disabled	R	0x3

Bits	Field Name	Description	Type	Reset
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by software. INTRCONN configuration port is not accessible. 0x1: Module is managed automatically by hardware along with EMU domain. INTRCONN configuration port is accessible only when EMU domain is on. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-347. Register Call Summary for Register CM\_PRM\_PROFILING\_CLKCTRL**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[0\]](#)

**Table 3-348. PRM\_DEBUG\_CFG**

<b>Address Offset</b>	0x0000 00F0		
<b>Physical Address</b>	See <a href="#">Table 3-327</a>	<b>Instance</b>	INTRCONN_SOCKET_PRM
<b>Description</b>	This register is used to configure the PRM's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	SEL3							RESERVED	SEL2							RESERVED	SEL1							RESERVED	SELO						

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30:24	SEL3	Internal signal block select for debug word byte-3	RW	0x03
23	RESERVED		R	0
22:16	SEL2	Internal signal block select for debug word byte-2	RW	0x02
15	RESERVED		R	0
14:8	SEL1	Internal signal block select for debug word byte-1	RW	0x01
7	RESERVED		R	0
6:0	SELO	Internal signal block select for debug word byte-0	RW	0x00

**Table 3-349. Register Call Summary for Register PRM\_DEBUG\_CFG**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Summary: \[0\]](#)



### 3.11.3 CKGEN\_PRM Registers

#### 3.11.3.1 CKGEN\_PRM Register Summary

**Table 3-350. CKGEN\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_PRM L4 Base Address
<a href="#">CM_ABE_DSS_SYS_CLKSEL</a>	RW	32	0x0000 0000	0x4A30 6100
RESERVED	RW	32	0x0000 0004	0x4A30 6104
<a href="#">CM_L4_WKUP_CLKSEL</a>	RW	32	0x0000 0008	0x4A30 6108
<a href="#">CM_ABE_PLL_REF_CLKSEL</a>	RW	32	0x0000 000C	0x4A30 610C
<a href="#">CM_SYS_CLKSEL</a>	RW	32	0x0000 0010	0x4A30 6110

#### 3.11.3.2 CKGEN\_PRM Register Description

**Table 3-351. CM\_ABE\_DSS\_SYS\_CLKSEL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-350</a>	<b>Instance</b>	CKGEN_PRM
<b>Description</b>	Select the SYS CLK for ABE and DSS subsystems. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKSEL															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK divided by 1 0x1: Select SYS_CLK divided by 2 Must be used for SYS_CLK>26MHz	RW	0

**Table 3-352. Register Call Summary for Register CM\_ABE\_DSS\_SYS\_CLKSEL**

- PRCM Register Manual
- [CKGEN\\_PRM Register Summary: \[0\]](#)

**Table 3-353. CM\_L4\_WKUP\_CLKSEL**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 3-350</a>	<b>Instance</b>	CKGEN_PRM
<b>Description</b>	Control the functional clock source of L4_WKUP, PRM and Smart Reflex functional clock.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKSEL															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CLKSEL	Select the clock source for L4WKUP_ICLK and for ABE_DPLL_BYPASS_CLK clocks.  0x0: Selects SYS_CLK for L4WKUP_ICLK Selects SYS_CLK for ABE_DPLL_BYPASS_CLK  0x1: Selects ABE_LP_CLK for L4WKUP_ICLK Selects 32K_FCLK for ABE_DPLL_BYPASS_CLK	RW	0

**Table 3-354. Register Call Summary for Register CM\_L4\_WKUP\_CLKSEL**

Clock Management Functional Description

- [PRM Clock Source: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_PRM Register Summary: \[2\]](#)

**Table 3-355. CM\_ABE\_PLL\_REF\_CLKSEL**

<b>Address Offset</b>	0x0000 000C	<b>Instance</b>	CKGEN_PRM
<b>Physical Address</b>	See <a href="#">Table 3-350</a>		
<b>Description</b>	Control the source of the reference clock for DPLL_ABE		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																CLKSEL																

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CLKSEL	Select the source for the DPLL_ABE reference clock.  0x0: Selects SYS_CLK 0x1: Selects SYS_32K	RW	1

**Table 3-356. Register Call Summary for Register CM\_ABE\_PLL\_REF\_CLKSEL**

Clock Management Functional Description

- [PRM Clock Source: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_PRM Register Summary: \[1\]](#)

**Table 3-357. CM\_SYS\_CLKSEL**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	CKGEN_PRM
<b>Physical Address</b>	See <a href="#">Table 3-350</a>		
<b>Description</b>	Software sets the SYS_CLK configuration corresponding to the frequency of SYS_CLK. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																SYS_CLKSEL																

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	SYS_CLKSEL	System clock input selection. 0x0: Uninitialized 0x1: Input clock is 12 MHz 0x2: Reserved 0x3: Input clock is 16.8 MHz 0x4: Input clock is 19.2 MHz 0x5: Input clock is 26 MHz 0x6: Reserved 0x7: Input clock is 38.4 MHz	RW	0x0

**Table 3-358. Register Call Summary for Register CM\_SYS\_CLKSEL**

PRCM Register Manual

- [CKGEN\\_PRM Register Summary: \[0\]](#)

### 3.11.4 MPU\_PRM Registers

#### 3.11.4.1 MPU\_PRM Register Summary

**Table 3-359. MPU\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	MPU_PRM L4 Base Address
<a href="#">PM_MPU_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 6300
<a href="#">PM_MPU_PWRSTST</a>	R	32	0x0000 0004	0x4A30 6304
<a href="#">RM_MPU_RSTST</a>	RW	32	0x0000 0014	0x4A30 6314
<a href="#">RM_MPU_MPU_CONTEXT</a>	RW	32	0x0000 0024	0x4A30 6324

#### 3.11.4.2 MPU\_PRM Register Description

**Table 3-360. PM\_MPU\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	MPU_PRM
<b>Physical Address</b>	See <a href="#">Table 3-359</a>		
<b>Description</b>	This register controls the MPU domain power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								MPU_RAM_ONSTATE	MPU_L2_ONSTATE	MPU_L1_ONSTATE	RESERVED					MPU_RAM_RETSTATE	MPU_L2_RETSTATE	MPU_L1_RETSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE								

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21:20	MPU_RAM_ONSTATE	MPU_RAM memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
19:18	MPU_L2_ONSTATE	MPU_L2 memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
17:16	MPU_L1_ONSTATE	MPU_L1 memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
15:11	RESERVED		R	0x00
10	MPU_RAM_RETSTATE	MPU_RAM memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1
9	MPU_L2_RETSTATE	MPU_L2 memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1
8	MPU_L1_RETSTATE	MPU_L1 memory state when domain is RETENTION. Should always be same as LogicRETState bit field. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3	RESERVED		R	0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaining logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	1
1:0	POWERSTATE	Power state control. 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

**Table 3-361. Register Call Summary for Register PM\_MPU\_PWRSTCTRL**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

PRCM Register Manual

- [MPU\\_PRM Register Summary: \[9\]](#)

**Table 3-362. PM\_MPU\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	MPU_PRM
<b>Physical Address</b>	See <a href="#">Table 3-359</a>		
<b>Description</b>	This register provides a status on the MPU domain current power state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED		INTRANSITION	RESERVED										MPU_RAM_STATEST	MPU_L2_STATEST	MPU_L1_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST					
LASTPOWERSTATEENTERED																															

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. Read 0x3: Power domain was previously ON-ACTIVE Read 0x2: Power domain was previously ON-INACTIVE Read 0x1: Power domain was previously in RETENTION Read 0x0: Power domain was previously OFF	RW W1toSet	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:10	RESERVED		R	0x000
9:8	MPU_RAM_STATEST	MPU_RAM memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
7:6	MPU_L2_STATEST	MPU_L2 memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
5:4	MPU_L1_STATEST	MPU_L1 memory state status Read 0x0: Memory is OFF Read 0x1: Reserved Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-363. Register Call Summary for Register PM\_MPU\_PWRSTST**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

PRCM Register Manual

- [MPU\\_PRM Register Summary: \[7\]](#)

**Table 3-364. RM\_MPU\_RSTST**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	MPU_PRM
<b>Physical Address</b>	See <a href="#">Table 3-359</a>		
<b>Description</b>	This register logs the different reset sources of the MPU domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												EMULATION_RST			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	EMULATION_RST	MPU domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: The domain has been reset upon emulation reset	RW W1toClr	0

**Table 3-365. Register Call Summary for Register RM\_MPU\_RSTST**

PRCM Register Manual

- [MPU\\_PRM Register Summary: \[0\]](#)

**Table 3-366. RM\_MPU\_MPU\_CONTEXT**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	MPU_PRM
<b>Physical Address</b>	See <a href="#">Table 3-359</a>		
<b>Description</b>	This register contains dedicated MPU context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																LOSTMEM_MPU_RAM			LOSTMEM_MPU_L2			LOSTMEM_MPU_L1			RESERVED								LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10	LOSTMEM_MPU_RAM	Specify if memory-based context in MPU_RAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
9	LOSTMEM_MPU_L2	Specify if memory-based context in MPU_L2 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
8	LOSTMEM_MPU_L1	Specify if memory-based context in MPU_L1 memory bank has been lost due to a previous power transition or other reset source. Not applicable to Cortex-A9 SMP. 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of MPU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-367. Register Call Summary for Register RM\_MPU\_MPU\_CONTEXT**

Power Management Functional Description

- [PD\\_MPU Description: \[0\]](#)

PRCM Register Manual

- [MPU\\_PRM Register Summary: \[1\]](#)

### 3.11.5 DSP\_PRM Registers

#### 3.11.5.1 DSP\_PRM Register Summary

**Table 3-368. DSP\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSP_PRM L4 Base Address
<a href="#">PM_DSP_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 6400
<a href="#">PM_DSP_PWRSTST</a>	R	32	0x0000 0004	0x4A30 6404
<a href="#">RM_DSP_RSTCTRL</a>	RW	32	0x0000 0010	0x4A30 6410
<a href="#">RM_DSP_RSTST</a>	RW	32	0x0000 0014	0x4A30 6414
<a href="#">RM_DSP_DSP_CONTEXT</a>	RW	32	0x0000 0024	0x4A30 6424

## 3.11.5.2 DSP\_PRM Register Description

Table 3-369. PM\_DSP\_PWRSTCTRL

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	DSP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-368</a>		
<b>Description</b>	This register controls the DSP power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
RESERVED								DSP_EDMA_ONSTATE			DSP_L2_ONSTATE			DSP_L1_ONSTATE			RESERVED					DSP_EDMA_RETSTATE			DSP_L2_RETSTATE			DSP_L1_RETSTATE			RESERVED			LOWPOWERSTATECHANGE			RESERVED			LOGICRETSTATE			POWERSTATE		

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21:20	DSP_EDMA_ONSTATE	DSP_EDMA state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
19:18	DSP_L2_ONSTATE	DSP_L2 state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
17:16	DSP_L1_ONSTATE	DSP_L1 state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
15:11	RESERVED		R	0x00
10	DSP_EDMA_RETSTATE	DSP_EDMA state when domain is RETENTION. Read 0x1: Memory bank is retained when domain is in RETENTION state.	R Rreturns 1s	1
9	DSP_L2_RETSTATE	DSP_L2 state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1
8	DSP_L1_RETSTATE	DSP_L2 state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3	RESERVED		R	0



Bits	Field Name	Description	Type	Reset
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaining logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	1
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

**Table 3-370. Register Call Summary for Register PM\_DSP\_PWRSTCTRL**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

PRCM Register Manual

- [DSP\\_PRM Register Summary: \[9\]](#)

**Table 3-371. PM\_DSP\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	DSP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-368</a>		
<b>Description</b>	This register provides a status on the DSP domain current power state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED		INTRANSITION	RESERVED											DSP_EDMA_STATEST	DSP_L2_STATEST	DSP_L1_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST				
LASTPOWERSTATEENTERED																															

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. Read 0x3: Power domain was previously ON-ACTIVE Read 0x2: Power domain was previously ON-INACTIVE Read 0x1: Power domain was previously in RETENTION Read 0x0: Power domain was previously OFF	RW W1toSet	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:10	RESERVED		R	0x000

Bits	Field Name	Description	Type	Reset
9:8	DSP_EDMA_STATEST	DSP_EDMA memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
7:6	DSP_L2_STATEST	DSP_L2 memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
5:4	DSP_L1_STATEST	DSP_L1 memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-372. Register Call Summary for Register PM\_DSP\_PWRSTST**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

PRCM Register Manual

- [DSP\\_PRM Register Summary: \[7\]](#)

**Table 3-373. RM\_DSP\_RSTCTRL**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	DSP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-368</a>		
<b>Description</b>	This register controls the release of the DSP sub-system resets.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RST2	RST1														

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	RST2	DSP - MMU, cache and slave interface reset control 0x0: Reset is cleared for the MMU, cache and slave interface 0x1: Reset is asserted for the MMU, cache and slave interface	RW	1

Bits	Field Name	Description	Type	Reset
0	RST1	DSP - DSP reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	1

**Table 3-374. Register Call Summary for Register RM\_DSP\_RSTCTRL**

Reset Management Functional Description

- [Reset Domains: \[0\]](#)
- [DSP Subsystem Power-On Reset Sequence: \[1\] \[2\]](#)
- [DSP Subsystem Software Warm Reset Sequence: \[3\] \[4\] \[5\] \[6\]](#)

PRCM Register Manual

- [DSP\\_PRM Register Summary: \[7\]](#)

**Table 3-375. RM\_DSP\_RSTST**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	DSP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-368</a>		
<b>Description</b>	This register logs the different reset sources of the DSP domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											DSP_DSP_EMU_REQ_RSTST	DSPSS_EMU_RSTST	RST2ST	RST1ST	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	DSP_DSP_EMU_REQ_RSTST	DSP DSP processor has been reset due to DSP emulation reset request driven from DSPSS 0x0: No emulation reset 0x1: DSP DSP has been reset upon emulation reset request	RW W1toClr	0
2	DSPSS_EMU_RSTST	DSP domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: DSP has been reset upon emulation reset	RW W1toClr	0
1	RST2ST	DSP MMU, cache and slave interface software reset status 0x0: No software reset occurred 0x1: MMU, cache and slave interface has been reset upon software reset	RW W1toClr	0
0	RST1ST	DSP DSP software reset 0x0: No software reset occurred 0x1: DSP has been reset upon software reset	RW W1toClr	0

**Table 3-376. Register Call Summary for Register RM\_DSP\_RSTST**

PRCM Register Manual

- [DSP\\_PRM Register Summary: \[0\]](#)

**Table 3-377. RM\_DSP\_DSP\_CONTEXT**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	DSP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-368</a>		
<b>Description</b>	This register contains dedicated DSP context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED																LOSTMEM_DSP_EDMA			LOSTMEM_DSP_L2			LOSTMEM_DSP_L1			RESERVED							LOSTCONTEXT_RFF		LOSTCONTEXT_DFF	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10	LOSTMEM_DSP_EDMA	Specify if memory-based context in DSP_EDMA memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
9	LOSTMEM_DSP_L2	Specify if memory-based context in DSP_L2 memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
8	LOSTMEM_DSP_L1	Specify if memory-based context in DSP_L1 memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSP_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSP_SYS_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-378. Register Call Summary for Register RM\_DSP\_DSP\_CONTEXT**

Power Management Functional Description

- [PD\\_DSP Description: \[0\] \[1\]](#)

PRCM Register Manual

- [DSP\\_PRM Register Summary: \[2\]](#)

### 3.11.6 ABE\_PRM Registers

#### 3.11.6.1 ABE\_PRM Register Summary

**Table 3-379. ABE\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ABE_PRM L4 Base Address
<a href="#">PM_ABE_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 6500
<a href="#">PM_ABE_PWRSTST</a>	R	32	0x0000 0004	0x4A30 6504
<a href="#">RM_ABE_AESS_CONTEXT</a>	RW	32	0x0000 002C	0x4A30 652C
<a href="#">PM_ABE_PDM_WKDEP</a>	RW	32	0x0000 0030	0x4A30 6530
<a href="#">RM_ABE_PDM_CONTEXT</a>	RW	32	0x0000 0034	0x4A30 6534
<a href="#">PM_ABE_DMIC_WKDEP</a>	RW	32	0x0000 0038	0x4A30 6538
<a href="#">RM_ABE_DMIC_CONTEXT</a>	RW	32	0x0000 003C	0x4A30 653C
<a href="#">PM_ABE_MCASP_WKDEP</a>	RW	32	0x0000 0040	0x4A30 6540
<a href="#">RM_ABE_MCASP_CONTEXT</a>	RW	32	0x0000 0044	0x4A30 6544
<a href="#">PM_ABE_MCBSP1_WKDEP</a>	RW	32	0x0000 0048	0x4A30 6548
<a href="#">RM_ABE_MCBSP1_CONTEXT</a>	RW	32	0x0000 004C	0x4A30 654C
<a href="#">PM_ABE_MCBSP2_WKDEP</a>	RW	32	0x0000 0050	0x4A30 6550
<a href="#">RM_ABE_MCBSP2_CONTEXT</a>	RW	32	0x0000 0054	0x4A30 6554
<a href="#">PM_ABE_MCBSP3_WKDEP</a>	RW	32	0x0000 0058	0x4A30 6558
<a href="#">RM_ABE_MCBSP3_CONTEXT</a>	RW	32	0x0000 005C	0x4A30 655C
<a href="#">PM_ABE_SLIMBUS_WKDEP</a>	RW	32	0x0000 0060	0x4A30 6560
<a href="#">RM_ABE_SLIMBUS_CONTEXT</a>	RW	32	0x0000 0064	0x4A30 6564
<a href="#">PM_ABE_GPTIMER5_WKDEP</a>	RW	32	0x0000 0068	0x4A30 6568
<a href="#">RM_ABE_GPTIMER5_CONTEXT</a>	RW	32	0x0000 006C	0x4A30 656C
<a href="#">PM_ABE_GPTIMER6_WKDEP</a>	RW	32	0x0000 0070	0x4A30 6570
<a href="#">RM_ABE_GPTIMER6_CONTEXT</a>	RW	32	0x0000 0074	0x4A30 6574
<a href="#">PM_ABE_GPTIMER7_WKDEP</a>	RW	32	0x0000 0078	0x4A30 6578
<a href="#">RM_ABE_GPTIMER7_CONTEXT</a>	RW	32	0x0000 007C	0x4A30 657C
<a href="#">PM_ABE_GPTIMER8_WKDEP</a>	RW	32	0x0000 0080	0x4A30 6580
<a href="#">RM_ABE_GPTIMER8_CONTEXT</a>	RW	32	0x0000 0084	0x4A30 6584
<a href="#">PM_ABE_WDTIMER3_WKDEP</a>	R	32	0x0000 0088	0x4A30 6588
<a href="#">RM_ABE_WDTIMER3_CONTEXT</a>	RW	32	0x0000 008C	0x4A30 658C

#### 3.11.6.2 ABE\_PRM Register Description

**Table 3-380. PM\_ABE\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register controls the ABE domain power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PERIPHEM_ONSTATE	RESERVED	AESSMEM_ONSTATE	RESERVED				PERIPHEM_RETSTATE	RESERVED	AESSMEM_RETSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE									

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21:20	PERIPHEM_ONSTATE	PERIPHEM memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
19:18	RESERVED		R	0x0
17:16	AESSMEM_ONSTATE	AESSMEM memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
15:11	RESERVED		R	0x00
10	PERIPHEM_RETSTATE	PERIPHEM memory state when domain is RETENTION. Read 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0
9	RESERVED		R	0
8	AESSMEM_RETSTATE	AESSMEM memory state when domain is RETENTION. Read 0x1: Memory bank is retained when domain is in RETENTION state.	R Rreturns 1s	1
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3	RESERVED		R	0
2	LOGICRETSTATE	Logic state when power domain is RETENTION Read 0x0: Whole logic is off when the domain is in RETENTION state.	R	0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x0

**Table 3-381. Register Call Summary for Register PM\_ABE\_PWRSTCTRL**

- Power Management Functional Description
- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- PRCM Register Manual
- [ABE\\_PRM Register Summary: \[7\]](#)

**Table 3-382. PM\_ABE\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register provides a status on the ABE domain current power domain state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED											PERIPHEM_STATEST	RESERVED	AESSMEM_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST				

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. Read 0x3: Power domain was previously ON-ACTIVE Read 0x2: Power domain was previously ON-INACTIVE Read 0x1: Power domain was previously in RETENTION Read 0x0: Power domain was previously OFF	RW W1toSet	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:10	RESERVED		R	0x000
9:8	PERIPHEM_STATEST	PERIPHEM memory state status Read 0x0: Memory is OFF Read 0x1: Reserved Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
7:6	RESERVED		R	0x0
5:4	AESSMEM_STATEST	AESSMEM memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-383. Register Call Summary for Register PM\_ABE\_PWRSTST**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[6\]](#)

**Table 3-384. RM\_ABE\_AESS\_CONTEXT**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register contains dedicated AESS context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_AESSMEM	RESERVED										LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_AESSMEM	Specify if memory-based context in AESSMEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-385. Register Call Summary for Register RM\_ABE\_AESS\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)



**Table 3-386. PM\_ABE\_PDM\_WKDEP**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register controls wakeup dependency based on PDM service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_PDM_DMA_SDMA		WKUPDEP_PDM_DMA_DSP		RESERVED			WKUPDEP_PDM_IRQ_DSP		RESERVED		WKUPDEP_PDM_IRQ_MPU				

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	WKUPDEP_PDM_DMA_SDMA	Wakeup dependency from PDM module (softwareakeup_dma signal) towards SDMA + L3_2 + L3_1 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	WKUPDEP_PDM_DMA_DSP	Wakeup dependency from PDM module (softwareakeup_dma signal) towards DSP domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5:3	RESERVED		R	0x0
2	WKUPDEP_PDM_IRQ_DSP	Wakeup dependency from PDM module (softwareakeup_irq signal) towards DSP domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_PDM_IRQ_MPU	Wakeup dependency from PDM module (softwareakeup_irq signal) towards MPU domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-387. Register Call Summary for Register PM\_ABE\_PDM\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[4\]](#)

**Table 3-388. RM\_ABE\_PDM\_CONTEXT**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register contains dedicated PDM context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_PERIHPMEM	RESERVED								LOSTCONTEXT_DFF						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_PERIHPMEM	Specify if memory-based context in PERIHPMEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-389. Register Call Summary for Register RM\_ABE\_PDM\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-390. PM\_ABE\_DMIC\_WKDEP**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register controls wakeup dependency based on DMIC service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_DMIC_DMA_SDMA	WKUPDEP_DMIC_DMA_DSP	RESERVED			WKUPDEP_DMIC_IRQ_DSP	RESERVED	WKUPDEP_DMIC_IRQ_MPU								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	WKUPDEP_DMIC_DMA_SDMA	Wakeup dependency from DMIC module (softwareakeup_dma signal) towards SDMA + L3_2 + L3_1 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	WKUPDEP_DMIC_DMA_DSP	Wakeup dependency from DMIC module (softwareakeup_dma signal) towards DSP domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5:3	RESERVED		R	0x0
2	WKUPDEP_DMIC_IRQ_DSP	Wakeup dependency from DMIC module (softwareakeup_irq signal) towards DSP domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_DMIC_IRQ_MPU	Wakeup dependency from DMIC module (softwareakeup_irq signal) towards MPU domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-391. Register Call Summary for Register PM\_ABE\_DMIC\_WKDEP**

- Clock Management Functional Description
- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)
- PRCM Register Manual
- [ABE\\_PRM Register Summary: \[4\]](#)

**Table 3-392. RM\_ABE\_DMIC\_CONTEXT**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register contains dedicated DMIC context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_PERIHPMEM	RESERVED								LOSTCONTEXT_DFF						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_PERIHPMEM	Specify if memory-based context in PERIHPMEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-393. Register Call Summary for Register RM\_ABE\_DMIC\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-394. PM\_ABE\_MCASP\_WKDEP**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register controls wakeup dependency based on MCASP service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP1_DMA_SDMA	WKUPDEP_MCASP1_DMA_DSP	RESERVED	WKUPDEP_MCASP1_IRQ_DSP	RESERVED	WKUPDEP_MCASP1_IRQ_MPU										

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	WKUPDEP_MCASP1_DMA_SD MA	Wakeup dependency from MCASP1 module (softwareakeup_dma signal) towards SDMA + L3_2 + L3_1 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	WKUPDEP_MCASP1_DMA_DS P	Wakeup dependency from MCASP1 module (softwareakeup_dma signal) towards DSP domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5:3	RESERVED		R	0x0
2	WKUPDEP_MCASP1_IRQ_DSP	Wakeup dependency from MCASP1 module (softwareakeup_irq signal) towards DSP domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_MCASP1_IRQ_MPU	Wakeup dependency from MCASP1 module (softwareakeup_irq signal) towards MPU domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-395. Register Call Summary for Register PM\_ABE\_MCASP\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[4\]](#)

**Table 3-396. RM\_ABE\_MCASP\_CONTEXT**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register contains dedicated MCASP context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-397. Register Call Summary for Register RM\_ABE\_MCBSP1\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-398. PM\_ABE\_MCBSP1\_WKDEP**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register controls wakeup dependency based on MCBSP1 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCBSP1_SDMA	WKUPDEP_MCBSP1_DSP	RESERVED	WKUPDEP_MCBSP1_MPU												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MCBSP1_SDMA	Wakeup dependency from MCBSP1 module (softwareakeup signal) towards SDMA + L3_2 + L3_1 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	WKUPDEP_MCBSP1_DSP	Wakeup dependency from MCBSP1 module (softwareakeup signal) towards DSP domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_MCBSP1_MPU	Wakeup dependency from MCBSP1 module (softwareakeup signal) towards MPU domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-399. Register Call Summary for Register PM\_ABE\_MCBSP1\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[3\]](#)

**Table 3-400. RM\_ABE\_MCBSP1\_CONTEXT**

<b>Address Offset</b>	0x0000 004C		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register contains dedicated MCBSP1 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_PERIHPMEM	RESERVED								LOSTCONTEXT_DFF						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_PERIHPMEM	Specify if memory-based context in PERIHPMEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-401. Register Call Summary for Register RM\_ABE\_MCBSP1\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-402. PM\_ABE\_MCBSP2\_WKDEP**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register controls wakeup dependency based on MCBSP2 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											WKUPDEP_MCBSP2_SDMA	WKUPDEP_MCBSP2_DSP	RESERVED	WKUPDEP_MCBSP2_MPU	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MCBSP2_SDMA	Wakeup dependency from MCBSP2 module (softwareakeup signal) towards SDMA + L3_2 + L3_1 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	WKUPDEP_MCBSP2_DSP	Wakeup dependency from MCBSP2 module (softwareakeup signal) towards DSP domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_MCBSP2_MPU	Wakeup dependency from MCBSP2 module (softwareakeup signal) towards MPU domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-403. Register Call Summary for Register PM\_ABE\_MCBSP2\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[3\]](#)

**Table 3-404. RM\_ABE\_MCBSP2\_CONTEXT**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register contains dedicated MCBSP2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_PERIHPMEM	RESERVED											LOSTCONTEXT_DFF			



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_PERIHPMEM	Specify if memory-based context in PERIHPMEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-405. Register Call Summary for Register RM\_ABE\_MCBSP2\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-406. PM\_ABE\_MCBSP3\_WKDEP**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register controls wakeup dependency based on MCBSP3 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCBSP3_SDMA			WKUPDEP_MCBSP3_DSP		RESERVED	WKUPDEP_MCBSP3_MPU									

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3	WKUPDEP_MCBSP3_SDMA	Wakeup dependency from MCBSP3 module (softwareakeup signal) towards SDMA + L3_2 + L3_1 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	WKUPDEP_MCBSP3_DSP	Wakeup dependency from MCBSP3 module (softwareakeup signal) towards DSP domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_MCBSP3_MPU	Wakeup dependency from MCBSP3 module (softwareakeup signal) towards MPU domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-407. Register Call Summary for Register PM\_ABE\_MCBSP3\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[3\]](#)

**Table 3-408. RM\_ABE\_MCBSP3\_CONTEXT**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register contains dedicated MCBSP3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_PERIHPMEM	RESERVED										LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_PERIHPMEM	Specify if memory-based context in PERIHPMEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-409. Register Call Summary for Register RM\_ABE\_MCBSP3\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-410. PM\_ABE\_SLIMBUS\_WKDEP**

<b>Address Offset</b>	0x0000 0060
<b>Physical Address</b>	See <a href="#">Table 3-379</a>
<b>Instance</b>	ABE_PRM
<b>Description</b>	This register controls wakeup dependency based on SLIMBUS service requests.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_SLIMBUS1_DMA_SDMA		WKUPDEP_SLIMBUS1_DMA_DSP		RESERVED			WKUPDEP_SLIMBUS1_IRQ_DSP		RESERVED		WKUPDEP_SLIMBUS1_IRQ_MPU				

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0000000
7	WKUPDEP_SLIMBUS1_DMA_S DMA	Wakeup dependency from SLIMBUS1 module (softwareakeup_dma signal) towards SDMA + L3_2 + L3_1 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	WKUPDEP_SLIMBUS1_DMA_D SP	Wakeup dependency from SLIMBUS1 module (softwareakeup_dma signal) towards DSP domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5:3	RESERVED		R	0x0
2	WKUPDEP_SLIMBUS1_IRQ_DS P	Wakeup dependency from SLIMBUS1 module (softwareakeup_irq signal) towards DSP domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_SLIMBUS1_IRQ_MP U	Wakeup dependency from SLIMBUS1 module (softwareakeup_irq signal) towards MPU domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-411. Register Call Summary for Register PM\_ABE\_SLIMBUS\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[4\]](#)

**Table 3-412. RM\_ABE\_SLIMBUS\_CONTEXT**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register contains dedicated SLIMBUS context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_PERIHPMEM	RESERVED								LOSTCONTEXT_DFF						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_PERIHPMEM	Specify if memory-based context in PERIHPMEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-413. Register Call Summary for Register RM\_ABE\_SLIMBUS\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-414. PM\_ABE\_GPTIMER5\_WKDEP**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register controls wakeup dependency based on TIMER5 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_TIMER5_DSP	RESERVED	WKUPDEP_TIMER5_MPU													

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	WKUPDEP_TIMER5_DSP	Wakeup dependency from TIMER5 module (softwareakeup signal) towards DSP domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_TIMER5_MPU	Wakeup dependency from TIMER5 module (softwareakeup signal) towards MPU domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-415. Register Call Summary for Register PM\_ABE\_GPTIMER5\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[2\]](#)

**Table 3-416. RM\_ABE\_GPTIMER5\_CONTEXT**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register contains dedicated TIMER5 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-417. Register Call Summary for Register RM\_ABE\_GPTIMER5\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-418. PM\_ABE\_GPTIMER6\_WKDEP**

<b>Address Offset</b>	0x0000 0070		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register controls wakeup dependency based on TIMER6 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_TIMER6_DSP		RESERVED		WKUPDEP_TIMER6_MPU											

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	WKUPDEP_TIMER6_DSP	Wakeup dependency from TIMER6 module (softwareakeup signal) towards DSP domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_TIMER6_MPU	Wakeup dependency from TIMER6 module (softwareakeup signal) towards MPU domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-419. Register Call Summary for Register PM\_ABE\_GPTIMER6\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[2\]](#)

**Table 3-420. RM\_ABE\_GPTIMER6\_CONTEXT**

<b>Address Offset</b>	0x0000 0074		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register contains dedicated TIMER6 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-421. Register Call Summary for Register RM\_ABE\_GPTIMER6\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-422. PM\_ABE\_GPTIMER7\_WKDEP**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register controls wakeup dependency based on TIMER7 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_TIMER7_DSP		RESERVED	WKUPDEP_TIMER7_MPU												

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	WKUPDEP_TIMER7_DSP	Wakeup dependency from TIMER7 module (softwareakeup signal) towards DSP domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_TIMER7_MPU	Wakeup dependency from TIMER7 module (softwareakeup signal) towards MPU domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-423. Register Call Summary for Register PM\_ABE\_GPTIMER7\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[2\]](#)

**Table 3-424. RM\_ABE\_GPTIMER7\_CONTEXT**

<b>Address Offset</b>	0x0000 007C		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register contains dedicated TIMER7 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-425. Register Call Summary for Register RM\_ABE\_GPTIMER7\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-426. PM\_ABE\_GPTIMER8\_WKDEP**

<b>Address Offset</b>	0x0000 0080		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register controls wakeup dependency based on TIMER8 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_TIMER8_DSP	RESERVED	WKUPDEP_TIMER8_MPU													



Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	WKUPDEP_TIMER8_DSP	Wakeup dependency from TIMER8 module (softwareakeup signal) towards DSP domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_TIMER8_MPU	Wakeup dependency from TIMER8 module (softwareakeup signal) towards MPU domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-427. Register Call Summary for Register PM\_ABE\_GPTIMER8\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[2\]](#)

**Table 3-428. RM\_ABE\_GPTIMER8\_CONTEXT**

<b>Address Offset</b>	0x0000 0084	<b>Instance</b>	ABE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-379</a>		
<b>Description</b>	This register contains dedicated TIMER8 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-429. Register Call Summary for Register RM\_ABE\_GPTIMER8\_CONTEXT**

Power Management Functional Description

- [PD\\_AUDIO Description: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-430. PM\_ABE\_WDTIMER3\_WKDEP**

<b>Address Offset</b>	0x0000 0088		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register controls wakeup dependency based on WDT3 service requests.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_WDT3_MPU			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	WKUPDEP_WDT3_MPU	Wakeup dependency from WDT3 module (softwareakeup signal) towards MPU domain Read 0x1: Dependency is enabled	R Returns 1s	1

**Table 3-431. Register Call Summary for Register PM\_ABE\_WDTIMER3\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\]](#)

PRCM Register Manual

- [ABE\\_PRM Register Summary: \[1\]](#)

**Table 3-432. RM\_ABE\_WDTIMER3\_CONTEXT**

<b>Address Offset</b>	0x0000 008C		
<b>Physical Address</b>	See <a href="#">Table 3-379</a>	<b>Instance</b>	ABE_PRM
<b>Description</b>	This register contains dedicated WDT3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of AUDIO_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-433. Register Call Summary for Register RM\_ABE\_WDTIMER3\_CONTEXT**

Power Management Functional Description
• <a href="#">PD_AUDIO Description: [0]</a>
PRCM Register Manual
• <a href="#">ABE_PRM Register Summary: [1]</a>

### 3.11.7 ALWAYS\_ON\_PRM Registers

#### 3.11.7.1 ALWAYS\_ON\_PRM Register Summary

**Table 3-434. ALWAYS\_ON\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ALWAYS_ON_PRM L4 Base Address
RESERVED	RW	32	0x0000 0024	0x4A30 6624
<a href="#">PM_ALWON_SR_MPU_WKDEP</a>	R	32	0x0000 0028	0x4A30 6628
<a href="#">RM_ALWON_SR_MPU_CONTEXT</a>	RW	32	0x0000 002C	0x4A30 662C
<a href="#">PM_ALWON_SR_IVA_WKDEP</a>	RW	32	0x0000 0030	0x4A30 6630
<a href="#">RM_ALWON_SR_IVA_CONTEXT</a>	RW	32	0x0000 0034	0x4A30 6634
<a href="#">PM_ALWON_SR_CORE_WKDEP</a>	RW	32	0x0000 0038	0x4A30 6638
<a href="#">RM_ALWON_SR_CORE_CONTEXT</a>	RW	32	0x0000 003C	0x4A30 663C

#### 3.11.7.2 ALWAYS\_ON\_PRM Register Description

**Table 3-435. PM\_ALWON\_SR\_MPU\_WKDEP**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 3-434</a>	<b>Instance</b>	ALWAYS_ON_PRM
<b>Description</b>	This register controls wakeup dependency based on SR_MPU service requests.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_SR_MPU_MPU															
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	WKUPDEP_SR_MPU_MPU	Wakeup dependency from SR_MPU module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains Read 0x1: Dependency is enabled	R Rreturns 1s	1

**Table 3-436. Register Call Summary for Register PM\_ALWON\_SR\_MPU\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\]](#)

PRCM Register Manual

- [ALWAYS\\_ON\\_PRM Register Summary: \[1\]](#)

**Table 3-437. RM\_ALWON\_SR\_MPU\_CONTEXT**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	ALWAYS_ON_PRM
<b>Physical Address</b>	See <a href="#">Table 3-434</a>		
<b>Description</b>	This register contains dedicated SR_MPU context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of Always_on_CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-438. Register Call Summary for Register RM\_ALWON\_SR\_MPU\_CONTEXT**

Power Management Functional Description

- [PD\\_ALWON\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [ALWAYS\\_ON\\_PRM Register Summary: \[1\]](#)

**Table 3-439. PM\_ALWON\_SR\_IVA\_WKDEP**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	ALWAYS_ON_PRM
<b>Physical Address</b>	See <a href="#">Table 3-434</a>		
<b>Description</b>	This register controls wakeup dependency based on SR_IVA service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_SR_IVA_MPU_M3	WKUPDEP_SR_IVA_MPU		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_SR_IVA_MPU_M3	Wakeup dependency from SR_IVA module (softwareakeup signal) towards MPU_A3 + L3_2 + L3_1 + L4_CFG domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_SR_IVA_MPU	Wakeup dependency from SR_IVA module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-440. Register Call Summary for Register PM\_ALWON\_SR\_IVA\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [ALWAYS\\_ON\\_PRM Register Summary: \[2\]](#)

**Table 3-441. RM\_ALWON\_SR\_IVA\_CONTEXT**

<b>Address Offset</b>	0x0000 0034
<b>Physical Address</b>	See <a href="#">Table 3-434</a>
<b>Instance</b>	ALWAYS_ON_PRM
<b>Description</b>	This register contains dedicated SR_IVA context statuses. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of Always_on_CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-442. Register Call Summary for Register RM\_ALWON\_SR\_IVA\_CONTEXT**

Power Management Functional Description

- [PD\\_ALWON\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [ALWAYS\\_ON\\_PRM Register Summary: \[1\]](#)

**Table 3-443. PM\_ALWON\_SR\_CORE\_WKDEP**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	ALWAYS_ON_PRM
<b>Physical Address</b>	See <a href="#">Table 3-434</a>		
<b>Description</b>	This register controls wakeup dependency based on SR_CORE service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_SR_CORE_MPU_M3		WKUPDEP_SR_CORE_MPU													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_SR_CORE_MPU_M3	Wakeup dependency from SR_CORE module (softwareakeup signal) towards MPU_A3 + L3_2 + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_SR_CORE_MPU	Wakeup dependency from SR_CORE module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-444. Register Call Summary for Register PM\_ALWON\_SR\_CORE\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [ALWAYS\\_ON\\_PRM Register Summary: \[2\]](#)

**Table 3-445. RM\_ALWON\_SR\_CORE\_CONTEXT**

<b>Address Offset</b>	0x0000 003C		
<b>Physical Address</b>	See <a href="#">Table 3-434</a>	<b>Instance</b>	ALWAYS_ON_PRM
<b>Description</b>	This register contains dedicated SR_CORE context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of Always_on_CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-446. Register Call Summary for Register RM\_ALWON\_SR\_CORE\_CONTEXT**

Power Management Functional Description

- [PD\\_ALWON\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [ALWAYS\\_ON\\_PRM Register Summary: \[1\]](#)

### 3.11.8 CORE\_PRM Registers

#### 3.11.8.1 CORE\_PRM Register Summary

**Table 3-447. CORE\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CORE_PRM L4 Base Address
<a href="#">PM_CORE_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 6700
<a href="#">PM_CORE_PWRSTST</a>	R	32	0x0000 0004	0x4A30 6704
<a href="#">RM_L3_1_L3_1_CONTEXT</a>	RW	32	0x0000 0024	0x4A30 6724
<a href="#">RM_L3_2_L3_2_CONTEXT</a>	RW	32	0x0000 0124	0x4A30 6824
<a href="#">RM_L3_2_GPMC_CONTEXT</a>	RW	32	0x0000 012C	0x4A30 682C
<a href="#">RM_L3_2_OCMC_RAM_CONTEXT</a>	RW	32	0x0000 0134	0x4A30 6834
<a href="#">RM_MPU_M3_RSTCTRL</a>	RW	32	0x0000 0210	0x4A30 6910
<a href="#">RM_MPU_M3_RSTST</a>	RW	32	0x0000 0214	0x4A30 6914
<a href="#">RM_MPU_M3_MPU_M3_CONTEXT</a>	RW	32	0x0000 0224	0x4A30 6924
<a href="#">RM_SDMA_SDMA_CONTEXT</a>	RW	32	0x0000 0324	0x4A30 6A24
<a href="#">RM_MEMIF_DMM_CONTEXT</a>	RW	32	0x0000 0424	0x4A30 6B24
<a href="#">RM_MEMIF_EMIF_FW_CONTEXT</a>	RW	32	0x0000 042C	0x4A30 6B2C
<a href="#">RM_MEMIF_EMIF_1_CONTEXT</a>	RW	32	0x0000 0434	0x4A30 6B34
<a href="#">RM_MEMIF_EMIF_2_CONTEXT</a>	RW	32	0x0000 043C	0x4A30 6B3C

**Table 3-447. CORE\_PRM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	CORE_PRM L4 Base Address
RM_MEMIF_DLL_CONTEXT	RW	32	0x0000 0444	0x4A30 6B44
RESERVED	RW	32	0x0000 0454	0x4A30 6B54
RESERVED	RW	32	0x0000 045C	0x4A30 6B5C
RESERVED	RW	32	0x0000 0464	0x4A30 6B64
RM_C2C_C2C_CONTEXT	RW	32	0x0000 0524	0x4A30 6C24
RESERVED	RW	32	0x0000 052C	0x4A30 6C2C
RM_C2C_C2C_FW_CONTEXT	RW	32	0x0000 0534	0x4A30 6C34
RM_L4CFG_L4_CFG_CONTEXT	RW	32	0x0000 0624	0x4A30 6D24
RM_L4CFG_SPINLOCK_CONTEXT	RW	32	0x0000 062C	0x4A30 6D2C
RM_L4CFG_MAILBOX_CONTEXT	RW	32	0x0000 0634	0x4A30 6D34
RM_L4CFG_SAR_ROM_CONTEXT	RW	32	0x0000 063C	0x4A30 6D3C
RM_L3INSTR_L3_3_CONTEXT	RW	32	0x0000 0724	0x4A30 6E24
RM_L3INSTR_L3_INSTR_CONTEXT	RW	32	0x0000 072C	0x4A30 6E2C
RM_L3INSTR_INTRCONN_WP1_CONTEXT	RW	32	0x0000 0744	0x4A30 6E44

**3.11.8.2 CORE\_PRM Register Description****Table 3-448. PM\_CORE\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-447</a>		
<b>Description</b>	This register controls the CORE power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED						INTRCONN_NRET_BANK_ONSTATE	MPU_M3_UNICACHE_ONSTATE	MPU_M3_L2RAM_ONSTATE	CORE_OCMRAM_ONSTATE	CORE_OTHER_BANK_ONSTATE	RESERVED	INTRCONN_NRET_BANK_RETSTATE	MPU_M3_UNICACHE_RETSTATE	MPU_M3_L2RAM_RETSTATE	CORE_OCMRAM_RETSTATE	CORE_OTHER_BANK_RETSTATE	RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE											

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	INTRCONN_NRET_BANK_ONSTATE	INTRCONN_WP bank and DMM bank2 state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Returns 1s	0x3
23:22	MPU_M3_UNICACHE_ONSTATE	MPU_A3 UNICACHE bank state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Returns 1s	0x3



Bits	Field Name	Description	Type	Reset
21:20	MPU_M3_L2RAM_ONSTATE	MPU_A3 L2 bank state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
19:18	CORE_OCMRAM_ONSTATE	OCMRAM bank state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
17:16	CORE_OTHER_BANK_ONSTATE	DMA/ICR bank and DMM bank1 state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
15:13	RESERVED		R	0x0
12	INTRCONN_NRET_BANK_RETSTATE	INTRCONN_WP bank and DMM bank2 state when domain is RETENTION. Read 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0
11	MPU_M3_UNICACHE_RETSTATE	MPU_A3 UNICACHE bank state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1
10	MPU_M3_L2RAM_RETSTATE	MPU_A3 L2 bank state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1
9	CORE_OCMRAM_RETSTATE	OCMRAM bank state when domain is RETENTION. Read 0x1: Memory bank is retained when domain is in RETENTION state.	R Rreturns 1s	1
8	CORE_OTHER_BANK_RETSTATE	DMA/ICR bank and DMM bank1 state when domain is RETENTION. Read 0x1: Memory bank is retained when domain is in RETENTION state.	R Rreturns 1s	1
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3	RESERVED		R	0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaining logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	1
1:0	POWERSTATE	Power state control Read 0x0: Reserved 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

**Table 3-449. Register Call Summary for Register PM\_CORE\_PWRSTCTRL**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[13\]](#)

**Table 3-450. PM\_CORE\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-447</a>		
<b>Description</b>	This register provides a status on the current CORE power domain state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LASTPOWERSTATEENTERED	RESERVED		INTRANSITION	RESERVED					INTRCONN_NRET_BANK_STATE	MPU_M3_UNICACHE_STATE	MPU_M3_L2RAM_STATE	CORE_OCMRAM_STATE	CORE_OTHER_BANK_STATE	RESERVED	LOGICSTATE	POWERSTATE							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only.  Read 0x3: Power domain was previously ON-ACTIVE Read 0x2: Power domain was previously ON-INACTIVE Read 0x1: Power domain was previously in RETENTION Read 0x0: Power domain was previously OFF	RW W1toSet	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status  Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:14	RESERVED		R	0x00
13:12	INTRCONN_NRET_BANK_STATE	INTRCONN_WP bank and DMM bank2 state status  Read 0x0: Memory is OFF Read 0x1: Reserved Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
11:10	MPU_M3_UNICACHE_STATES	MPU_A3 UNICACHE bank state status  Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3

Bits	Field Name	Description	Type	Reset
9:8	MPU_M3_L2RAM_STATEST	MPU_A3 L2 bank state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
7:6	CORE_OCMRAM_STATEST	OCMRAM bank state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
5:4	CORE_OTHER_BANK_STATES T	DMA/ICR bank and DMM bank1 state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1
1:0	POWERSTATEST	Current power state status Read 0x0: Reserved Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-451. Register Call Summary for Register PM\_CORE\_PWRSTST**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[9\]](#)

**Table 3-452. RM\_L3\_1\_L3\_1\_CONTEXT**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-447</a>		
<b>Description</b>	This register contains dedicated L3_1 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		LOSTCONTEXT_DFF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-453. Register Call Summary for Register RM\_L3\_1\_L3\_1\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[2\]](#)

**Table 3-454. RM\_L3\_2\_L3\_2\_CONTEXT**

<b>Address Offset</b>	0x0000 0124	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-447</a>		
<b>Description</b>	This register contains dedicated L3_2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		LOSTCONTEXT_DFF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-455. Register Call Summary for Register RM\_L3\_2\_L3\_2\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[2\]](#)

**Table 3-456. RM\_L3\_2\_GPMC\_CONTEXT**

<b>Address Offset</b>	0x0000 012C		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated GPMC context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											LOSTCONTEXT_RFF	RESERVED			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-457. Register Call Summary for Register RM\_L3\_2\_GPMC\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[1\]](#)

**Table 3-458. RM\_L3\_2\_OCMC\_RAM\_CONTEXT**

<b>Address Offset</b>	0x0000 0134		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated OCMC_RAM context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_CORE_OCMRAM	RESERVED											LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_CORE_OCMRAM	Specify if memory-based context in CORE_OCMRAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset).  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-459. Register Call Summary for Register RM\_L3\_2\_OCMC\_RAM\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[1\]](#)

**Table 3-460. RM\_MPU\_M3\_RSTCTRL**

<b>Address Offset</b>	0x0000 0210	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-447</a>		
<b>Description</b>	This register controls the release of the MPU_A3 sub-system resets.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RST3	RST2	RST1													

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	RST3	MPU_A3 MMU and CACHE interface reset control.  0x0: Reset is cleared for MPU_A3 CACHE & MMU 0x1: Reset is asserted for the MPU_A3 CACHE & MMU	RW	1
1	RST2	MPU_A3 Cortex M3 CPU2 reset control.  0x0: Reset is cleared for the MPU_A3 Cortex M3 CPU2 0x1: Reset is asserted for the MPU_A3 Cortex M3 CPU2	RW	1
0	RST1	MPU_A3 Cortex M3 CPU1 reset control.  0x0: Reset is cleared for the MPU_A3 Cortex M3 CPU1 0x1: Reset is asserted for the MPU_A3 Cortex M3 CPU1	RW	1

**Table 3-461. Register Call Summary for Register RM\_MPU\_M3\_RSTCTRL**

Reset Management Functional Description

- [Reset Domains: \[0\]](#)
- [Cortex-M3 MPU Subsystem Power-On Reset Sequence: \[1\] \[2\] \[3\]](#)
- [Cortex-M3 MPU Subsystem Software Warm Reset Sequence: \[4\] \[5\] \[6\] \[7\] \[8\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[9\]](#)

**Table 3-462. RM\_MPU\_M3\_RSTST**

<b>Address Offset</b>	0x0000 0214		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register logs the different reset sources of the MPU_A3 SS. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ICECRUSHER_RST2ST	ICECRUSHER_RST1ST	EMULATION_RST2ST	EMULATION_RST1ST	RST3ST	RST2ST	RST1ST									

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	ICECRUSHER_RST2ST	Cortex M3 CPU2 has been reset due to MPU_A3 ICECRUSHER2 reset source 0x0: No icecrusher reset 0x1: CPU2 has been reset upon icecrusher reset	RW W1toClr	0
5	ICECRUSHER_RST1ST	Cortex M3 CPU1 has been reset due to MPU_A3 ICECRUSHER1 reset source 0x0: No icecrusher reset 0x1: CPU1 has been reset upon icecrusher reset	RW W1toClr	0
4	EMULATION_RST2ST	Cortex M3 CPU2 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: CPU2 has been reset upon emulation reset	RW W1toClr	0
3	EMULATION_RST1ST	Cortex M3 CPU1 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: CPU1 has been reset upon emulation reset	RW W1toClr	0
2	RST3ST	MPU_A3 MMU and CACHE interface software reset status 0x0: No software reset occurred 0x1: MPU_A3 MMU and CACHE interface has been reset upon software reset	RW W1toClr	0
1	RST2ST	MPU_A3 Cortex-M3 CPU2 software reset status 0x0: No software reset occurred 0x1: Cortex M3 CPU2 has been reset upon software reset	RW W1toClr	0
0	RST1ST	MPU_A3 Cortex-M3 CPU1 software reset status 0x0: No software reset occurred 0x1: Cortex M3 CPU1 has been reset upon software reset	RW W1toClr	0

**Table 3-463. Register Call Summary for Register RM\_MPU\_M3\_RSTST**

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[0\]](#)

**Table 3-464. RM\_MPU\_M3\_MPU\_M3\_CONTEXT**

<b>Address Offset</b>	0x0000 0224	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-447</a>		
<b>Description</b>	This register contains dedicated MPU_A3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_MPU_M3_L2RAM		LOSTMEM_MPU_M3_UNICACHE		RESERVED						LOSTCONTEXT_RFF		LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	LOSTMEM_MPU_M3_L2RAM	Specify if memory-based context in MPU_A3_L2RAM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
8	LOSTMEM_MPU_M3_UNICACHE	Specify if memory-based context in MPU_A3_UNICACHE memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of MPU_A3_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of MPU_A3_RST3 signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-465. Register Call Summary for Register RM\_MPU\_M3\_MPU\_M3\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[2\]](#)



**Table 3-466. RM\_SDMA\_SDMA\_CONTEXT**

<b>Address Offset</b>	0x0000 0324		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated SDMA context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_CORE_OTHER_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_CORE_OTHER_BA NK	Specify if memory-based context in CORE_OTHER_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of SDMA_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-467. Register Call Summary for Register RM\_SDMA\_SDMA\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[1\]](#)

**Table 3-468. RM\_MEMIF\_DMM\_CONTEXT**

<b>Address Offset</b>	0x0000 0424		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated DMM context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_CORE_NRET_BANK		LOSTMEM_CORE_OTHER_BANK		RESERVED								LOSTCONTEXT_RFF		LOSTCONTEXT_DFF	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	LOSTMEM_CORE_NRET_BANK	Specify if memory-based context in CORE_NRET_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
8	LOSTMEM_CORE_OTHER_BANK	Specify if memory-based context in CORE_OTHER_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-469. Register Call Summary for Register RM\_MEMIF\_DMM\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[2\]](#)

**Table 3-470. RM\_MEMIF\_EMIF\_FW\_CONTEXT**

<b>Address Offset</b>	0x0000 042C	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-447</a>		
<b>Description</b>	This register contains dedicated EMIF_FW context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_RFF	LOSTCONTEXT_DFF		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-471. Register Call Summary for Register RM\_MEMIF\_EMIF\_FW\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[2\]](#)

**Table 3-472. RM\_MEMIF\_EMIF\_1\_CONTEXT**

<b>Address Offset</b>	0x0000 0434
<b>Physical Address</b>	See <a href="#">Table 3-447</a>
<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated EMIF_1 context statuses. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_RFF	LOSTCONTEXT_DFF		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-473. Register Call Summary for Register RM\_MEMIF\_EMIF\_1\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[2\]](#)

**Table 3-474. RM\_MEMIF\_EMIF\_2\_CONTEXT**

<b>Address Offset</b>	0x0000 043C		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated EMIF_2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		LOSTCONTEXT_DFF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-475. Register Call Summary for Register RM\_MEMIF\_EMIF\_2\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[2\]](#)

**Table 3-476. RM\_MEMIF\_DLL\_CONTEXT**

<b>Address Offset</b>	0x0000 0444		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated DLL context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DLL_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-477. Register Call Summary for Register RM\_MEMIF\_DLL\_CONTEXT**

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[0\]](#)

**Table 3-478. RM\_C2C\_C2C\_CONTEXT**

<b>Address Offset</b>	0x0000 0524		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated C2C context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_RFF	LOSTCONTEXT_DFF		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-479. Register Call Summary for Register RM\_C2C\_C2C\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRCM Register Summary: \[2\]](#)

**Table 3-480. RM\_C2C\_C2C\_FW\_CONTEXT**

<b>Address Offset</b>	0x0000 0534		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRCM
<b>Description</b>	This register contains dedicated C2C_FW context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		LOSTCONTEXT_DFF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-481. Register Call Summary for Register RM\_C2C\_C2C\_FW\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRCM Register Summary: \[2\]](#)

**Table 3-482. RM\_L4CFG\_L4\_CFG\_CONTEXT**

<b>Address Offset</b>	0x0000 0624		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated L4_CFG context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		LOSTCONTEXT_DFF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-483. Register Call Summary for Register RM\_L4CFG\_L4\_CFG\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[2\]](#)

**Table 3-484. RM\_L4CFG\_SPINLOCK\_CONTEXT**

<b>Address Offset</b>	0x0000 062C		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated HW_SEM context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		RESERVED													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-485. Register Call Summary for Register RM\_L4CFG\_SPINLOCK\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[1\]](#)

**Table 3-486. RM\_L4CFG\_MAILBOX\_CONTEXT**

<b>Address Offset</b>	0x0000 0634	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-447</a>		
<b>Description</b>	This register contains dedicated MAILBOX context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		RESERVED													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-487. Register Call Summary for Register RM\_L4CFG\_MAILBOX\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[1\]](#)



**Table 3-488. RM\_L4CFG\_SAR\_ROM\_CONTEXT**

<b>Address Offset</b>	0x0000 063C		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated SAR_ROM context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-489. Register Call Summary for Register RM\_L4CFG\_SAR\_ROM\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[1\]](#)

**Table 3-490. RM\_L3INSTR\_L3\_3\_CONTEXT**

<b>Address Offset</b>	0x0000 0724		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated L3_3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_RFF	LOSTCONTEXT_DFF		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-491. Register Call Summary for Register RM\_L3INSTR\_L3\_3\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[2\]](#)

**Table 3-492. RM\_L3INSTR\_L3\_INSTR\_CONTEXT**

<b>Address Offset</b>	0x0000 072C		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated L3_INSTR context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																LOSTCONTEXT_DFF																

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-493. Register Call Summary for Register RM\_L3INSTR\_L3\_INSTR\_CONTEXT**

Power Management Functional Description

- [PD\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [CORE\\_PRM Register Summary: \[1\]](#)

**Table 3-494. RM\_L3INSTR\_INTRCONN\_WP1\_CONTEXT**

<b>Address Offset</b>	0x0000 0744		
<b>Physical Address</b>	See <a href="#">Table 3-447</a>	<b>Instance</b>	CORE_PRM
<b>Description</b>	This register contains dedicated INTRCONN_WP1 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_CORE_NRET_BANK	RESERVED						LOSTCONTEXT_RFF	LOSTCONTEXT_DFF							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_CORE_NRET_BANK	Specify if memory-based context in CORE_NRET_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-495. Register Call Summary for Register RM\_L3INSTR\_INTRCONN\_WP1\_CONTEXT**

- PRCM Register Manual
- [CORE\\_PRM Register Summary: \[0\]](#)

### 3.11.9 IVAHD\_PRM Registers

#### 3.11.9.1 IVAHD\_PRM Register Summary

**Table 3-496. IVAHD\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	IVAHD_PRM L4 Base Address
<a href="#">PM_IVAHD_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 6F00
<a href="#">PM_IVAHD_PWRSTST</a>	R	32	0x0000 0004	0x4A30 6F04

**Table 3-496. IVAHD\_PRM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	IVAHD_PRM L4 Base Address
<a href="#">RM_IVAHD_RSTCTRL</a>	RW	32	0x0000 0010	0x4A30 6F10
<a href="#">RM_IVAHD_RSTST</a>	RW	32	0x0000 0014	0x4A30 6F14
<a href="#">RM_IVAHD_IVAHD_CONTEXT</a>	RW	32	0x0000 0024	0x4A30 6F24
<a href="#">RM_IVAHD_SL2_CONTEXT</a>	RW	32	0x0000 002C	0x4A30 6F2C

**3.11.9.2 IVAHD\_PRM Register Description****Table 3-497. PM\_IVAHD\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	IVAHD_PRM
<b>Physical Address</b>	See <a href="#">Table 3-496</a>		
<b>Description</b>	This register controls the IVAHD power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TCM2_MEM_ONSTATE	TCM1_MEM_ONSTATE	SL2_MEM_ONSTATE	HWA_MEM_ONSTATE	RESERVED				TCM2_MEM_RETSTATE	TCM1_MEM_RETSTATE	SL2_MEM_RETSTATE	HWA_MEM_RETSTATE	RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE					

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:22	TCM2_MEM_ONSTATE	TCM2 memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
21:20	TCM1_MEM_ONSTATE	TCM1 memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
19:18	SL2_MEM_ONSTATE	SL2 memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
17:16	HWA_MEM_ONSTATE	HWA memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
15:12	RESERVED		R	0x0
11	TCM2_MEM_RETSTATE	TCM2 memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1
10	TCM1_MEM_RETSTATE	TCM1 memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1

Bits	Field Name	Description	Type	Reset
9	SL2_MEM_RETSTATE	SL2 memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	1
8	HWA_MEM_RETSTATE	HWA memory state when domain is RETENTION. Read 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3	RESERVED		R	0
2	LOGICRETSTATE	Logic state when power domain is RETENTION Read 0x0: Whole logic is off when the domain is in RETENTION state.	R	0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

**Table 3-498. Register Call Summary for Register PM\_IVAHD\_PWRSTCTRL**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

PRCM Register Manual

- [IVAHD\\_PRM Register Summary: \[11\]](#)

**Table 3-499. PM\_IVAHD\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	IVAHD_PRM
<b>Physical Address</b>	See <a href="#">Table 3-496</a>		
<b>Description</b>	This register provides a status on the current IVAHD power domain state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LASTPOWERSTATEENTERED		RESERVED		INTRANSITION		RESERVED				TCM2_MEM_STATE		TCM1_MEM_STATE		SL2_MEM_STATE		HWA_MEM_STATE		RESERVED		LOGICSTATE		POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. Read 0x3: Power domain was previously ON-ACTIVE Read 0x2: Power domain was previously ON-INACTIVE Read 0x1: Power domain was previously in RETENTION Read 0x0: Power domain was previously OFF	RW W1toSet	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:12	RESERVED		R	0x00
11:10	TCM2_MEM_STATEST	TCM2 memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
9:8	TCM1_MEM_STATEST	TCM1 memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
7:6	SL2_MEM_STATEST	SL2 memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
5:4	HWA_MEM_STATEST	HWA memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-500. Register Call Summary for Register PM\_IVAHD\_PWRSTST**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

PRCM Register Manual

- [IVAHD\\_PRM Register Summary: \[8\]](#)

**Table 3-501. RM\_IVAHD\_RSTCTRL**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 3-496</a>	<b>Instance</b>	IVAHD_PRM
<b>Description</b>	This register controls the release of the IVAHD sub-system resets.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											RST3	RST2	RST1		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	RST3	IVAHD logic and SL2 reset control 0x0: Reset is cleared for the IVAHD logic and SL2 0x1: Reset is asserted for IVAHD logic and SL2	RW	1
1	RST2	IVAHD Sequencer2 reset control 0x0: Reset is cleared for IVAHD Sequencer CPU2 0x1: Reset is asserted for IVAHD Sequencer CPU2	RW	1
0	RST1	IVAHD sequencer1 reset control 0x0: Reset is cleared for the IVAHD Sequencer CPU1 0x1: Reset is asserted for the IVAHD sequencer CPU1	RW	1

**Table 3-502. Register Call Summary for Register RM\_IVAHD\_RSTCTRL**

Reset Management Functional Description

- [Reset Domains: \[0\]](#)
- [IVAHD Subsystem Power-On Reset Sequence: \[1\] \[2\] \[3\]](#)
- [IVAHD Subsystem Software Warm Reset Sequence: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)

PRCM Register Manual

- [IVAHD\\_PRM Register Summary: \[10\]](#)

**Table 3-503. RM\_IVAHD\_RSTST**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 3-496</a>	<b>Instance</b>	IVAHD_PRM
<b>Description</b>	This register logs the different reset sources of the IVAHD domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																											ICECRUSHER_SEQ2_RST2ST	ICECRUSHER_SEQ1_RST1ST	EMULATION_SEQ2_RST2ST	EMULATION_SEQ1_RST1ST	RST3ST	RST2ST	RST1ST

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6	ICECRUSHER_SEQ2_RST2ST	Sequencer2 CPU has been reset due to IVAHD ICECRUSHER2 reset event  0x0: No icecrusher reset 0x1: Sequencer2 has been reset upon icecrusher reset	RW W1toClr	0
5	ICECRUSHER_SEQ1_RST1ST	Sequencer1 CPU has been reset due to IVAHD ICECRUSHER1 reset event  0x0: No icecrusher reset 0x1: Sequencer1 has been reset upon icecrusher reset	RW W1toClr	0
4	EMULATION_SEQ2_RST2ST	0x0: No emulation reset 0x1: Sequencer2 has been reset upon emulation reset	RW W1toClr	0
3	EMULATION_SEQ1_RST1ST	Sequencer1 CPU has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module  0x0: No emulation reset 0x1: Sequencer1 has been reset upon emulation reset	RW W1toClr	0
2	RST3ST	IVAHD logic and SL2 software reset 0x0: No software reset occurred 0x1: IVAHD logic and SL2 has been reset upon software reset	RW W1toClr	0
1	RST2ST	IVAHD Sequencer2 CPU software reset 0x0: No software reset occurred 0x1: Sequencer2 has been reset upon software reset	RW W1toClr	0
0	RST1ST	IVAHD Sequencer1 CPU software reset 0x0: No software reset occurred 0x1: Sequencer1 has been reset upon software reset	RW W1toClr	0

**Table 3-504. Register Call Summary for Register RM\_IVAHD\_RSTST**

PRCM Register Manual

- [IVAHD\\_PRM Register Summary: \[0\]](#)

**Table 3-505. RM\_IVAHD\_IVAHD\_CONTEXT**

<b>Address Offset</b>	0x0000 0024	
<b>Physical Address</b>	See <a href="#">Table 3-496</a>	<b>Instance</b> IVAHD_PRM
<b>Description</b>	This register contains dedicated IVAHD context statuses. [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																LOSTMEM_HWA_MEM			LOSTMEM_TCM2_MEM			LOSTMEM_TCM1_MEM			RESERVED							LOSTCONTEXT_DFF



Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10	LOSTMEM_HWA_MEM	Specify if memory-based context in HWA_MEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
9	LOSTMEM_TCM2_MEM	Specify if memory-based context in TCM2_MEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
8	LOSTMEM_TCM1_MEM	Specify if memory-based context in TCM1_MEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IVAHD_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-506. Register Call Summary for Register RM\_IVAHD\_IVAHD\_CONTEXT**

Power Management Functional Description

- [PD\\_IVAHD Description: \[0\]](#)

PRCM Register Manual

- [IVAHD\\_PRM Register Summary: \[1\]](#)

**Table 3-507. RM\_IVAHD\_SL2\_CONTEXT**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	IVAHD_PRM
<b>Physical Address</b>	See <a href="#">Table 3-496</a>		
<b>Description</b>	This register contains dedicated SL2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_SL2_MEM	RESERVED										LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_SL2_MEM	Specify if memory-based context in SL2_MEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IVAHD_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-508. Register Call Summary for Register RM\_IVAHD\_SL2\_CONTEXT**

Power Management Functional Description

- [PD\\_IVAHD Description: \[0\]](#)

PRCM Register Manual

- [IVAHD\\_PRM Register Summary: \[1\]](#)

### 3.11.10 CAM\_PRM Registers

#### 3.11.10.1 CAM\_PRM Register Summary

**Table 3-509. CAM\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CAM_PRM L4 Base Address
<a href="#">PM_CAM_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 7000
<a href="#">PM_CAM_PWRSTST</a>	R	32	0x0000 0004	0x4A30 7004
<a href="#">RM_CAM_ISS_CONTEXT</a>	RW	32	0x0000 0024	0x4A30 7024
<a href="#">RM_CAM_FDIF_CONTEXT</a>	RW	32	0x0000 002C	0x4A30 702C

#### 3.11.10.2 CAM\_PRM Register Description

**Table 3-510. PM\_CAM\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	CAM_PRM
<b>Physical Address</b>	See <a href="#">Table 3-509</a>		
<b>Description</b>	This register controls the CAM power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CAM_MEM_ONSTATE	RESERVED								LOWPOWERSTATECHANGE	RESERVED	POWERSTATE												

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	CAM_MEM_ONSTATE	CAM_MEM memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Returns 1s	0x3
15:5	RESERVED		R	0x000

Bits	Field Name	Description	Type	Reset
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain.  0x0: Do not request a low power state change.  0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control  0x0: OFF state Read 0x1: Reserved 0x2: INACTIVE state 0x3: ON State	RW	0x0

**Table 3-511. Register Call Summary for Register PM\_CAM\_PWRSTCTRL**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [CAM\\_PRM Register Summary: \[3\]](#)

**Table 3-512. PM\_CAM\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	CAM_PRM
<b>Physical Address</b>	See <a href="#">Table 3-509</a>		
<b>Description</b>	This register provides a status on the current CAM power domain state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INTRANSITION	RESERVED								CAM_MEM_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST											

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20	INTRANSITION	Domain transition status  Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:6	RESERVED		R	0x0000
5:4	CAM_MEM_STATEST	CAM_MEM memory state status  Read 0x0: Memory is OFF Read 0x1: Reserved Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0
2	LOGICSTATEST	Logic state status  Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-513. Register Call Summary for Register PM\_CAM\_PWRSTST**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CAM\\_PRM Register Summary: \[4\]](#)

**Table 3-514. RM\_CAM\_ISS\_CONTEXT**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	CAM_PRM
<b>Physical Address</b>	See <a href="#">Table 3-509</a>		
<b>Description</b>	This register contains dedicated ISS context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_CAM_MEM	RESERVED										LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_CAM_MEM	Specify if memory-based context in CAM_MEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-515. Register Call Summary for Register RM\_CAM\_ISS\_CONTEXT**

Power Management Functional Description

- [PD\\_CAM Description: \[0\]](#)

PRCM Register Manual

- [CAM\\_PRM Register Summary: \[1\]](#)

**Table 3-516. RM\_CAM\_FDIF\_CONTEXT**

<b>Address Offset</b>	0x0000 002C		
<b>Physical Address</b>	See <a href="#">Table 3-509</a>	<b>Instance</b>	CAM_PRM
<b>Description</b>	This register contains dedicated FDIF context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_CAM_MEM	RESERVED										LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_CAM_MEM	Specify if memory-based context in CAM_MEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-517. Register Call Summary for Register RM\_CAM\_FDIF\_CONTEXT**

Power Management Functional Description

- [PD\\_CAM Description: \[0\]](#)

PRCM Register Manual

- [CAM\\_PRM Register Summary: \[1\]](#)

### 3.11.11 DSS\_PRM Registers

#### 3.11.11.1 DSS\_PRM Register Summary

**Table 3-518. DSS\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSS_PRM L4 Base Address
<a href="#">PM_DSS_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 7100
<a href="#">PM_DSS_PWRSTST</a>	R	32	0x0000 0004	0x4A30 7104
<a href="#">PM_DSS_DSS_WKDEP</a>	RW	32	0x0000 0020	0x4A30 7120
<a href="#">RM_DSS_DSS_CONTEXT</a>	RW	32	0x0000 0024	0x4A30 7124

## 3.11.11.2 DSS\_PRM Register Description

Table 3-519. PM\_DSS\_PWRSTCTRL

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	DSS_PRM
<b>Physical Address</b>	See <a href="#">Table 3-518</a>		
<b>Description</b>	This register controls the DSS power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED																DSS_MEM_ONSTATE		RESERVED						DSS_MEM_RETSTATE		RESERVED		LOWPOWERSTATECHANGE		RESERVED		LOGICRETSTATE		POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	DSS_MEM_ONSTATE	DSS_MEM state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
15:9	RESERVED		R	0x00
8	DSS_MEM_RETSTATE	DSS_MEM state when domain is RETENTION. Read 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3	RESERVED		R	0
2	LOGICRETSTATE	Logic state when power domain is RETENTION Read 0x0: Whole logic is off when the domain is in RETENTION state.	R	0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x0

Table 3-520. Register Call Summary for Register PM\_DSS\_PWRSTCTRL

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [DSS\\_PRM Register Summary: \[5\]](#)

**Table 3-521. PM\_DSS\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	DSS_PRM
<b>Physical Address</b>	See <a href="#">Table 3-518</a>		
<b>Description</b>	This register provides a status on the current DSS power domain state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INTRANSITION	RESERVED								DSS_MEM_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST											

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:6	RESERVED		R	0x0000
5:4	DSS_MEM_STATEST	DSS_MEM state status Read 0x0: Memory is OFF Read 0x1: Reserved Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-522. Register Call Summary for Register PM\_DSS\_PWRSTST**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [DSS\\_PRM Register Summary: \[4\]](#)

**Table 3-523. PM\_DSS\_DSS\_WKDEP**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	DSS_PRM
<b>Physical Address</b>	See <a href="#">Table 3-518</a>		
<b>Description</b>	This register controls wakeup dependency based on DSS service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								WKUPDEP_HDMIIRQ_DSP	WKUPDEP_HDMIIRQ_MPU_M3	WKUPDEP_HDMIIRQ_MPU	WKUPDEP_DSI2_SDMA	WKUPDEP_DSI2_DSP	WKUPDEP_DSI2_MPU_M3	WKUPDEP_DSI2_MPU	WKUPDEP_DSI1_SDMA	WKUPDEP_DSI1_DSP	WKUPDEP_DSI1_MPU_M3	WKUPDEP_DSI1_MPU	WKUPDEP_DISPC_SDMA	WKUPDEP_DISPC_DSP	WKUPDEP_DISPC_MPU_M3	WKUPDEP_DISPC_MPU	

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19	WKUPDEP_HDMIDMA_SDMA	Wakeup dependency from HDMI module (softwareakeup_HDMI_dma signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
18:15	RESERVED		R	0x0
14	WKUPDEP_HDMIIRQ_DSP	Wakeup dependency from HDMI module (softwareakeup_HDMI_irq signal) towards DSP + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
13	WKUPDEP_HDMIIRQ_MPU_M3	Wakeup dependency from HDMI module (softwareakeup_HDMI_irq signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
12	WKUPDEP_HDMIIRQ_MPU	Wakeup dependency from HDMI module (softwareakeup_HDMI_irq signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
11	WKUPDEP_DSI2_SDMA	Wakeup dependency from DSI2 module (softwareakeup_DSI2 signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
10	WKUPDEP_DSI2_DSP	Wakeup dependency from DSI2 module (softwareakeup_DSI2 signal) towards DSP + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0



Bits	Field Name	Description	Type	Reset
9	WKUPDEP_DSI2_MPU_M3	Wakeup dependency from DSI2 module (softwareakeup_DSI2 signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
8	WKUPDEP_DSI2_MPU	Wakeup dependency from DSI2 module (softwareakeup_DSI2 signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
7	WKUPDEP_DSI1_SDMA	Wakeup dependency from DSI1 module (softwareakeup_DSI1 signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	WKUPDEP_DSI1_DSP	Wakeup dependency from DSI1 module (softwareakeup_DSI1 signal) towards DSP + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5	WKUPDEP_DSI1_MPU_M3	Wakeup dependency from DSI1 module (softwareakeup_DSI1 signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
4	WKUPDEP_DSI1_MPU	Wakeup dependency from DSI1 module (softwareakeup_DSI1 signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
3	WKUPDEP_DISPC_SDMA	Wakeup dependency from DISPC module (softwareakeup_DISPC signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	WKUPDEP_DISPC_DSP	Wakeup dependency from DISPC module (softwareakeup_DISPC signal) towards DSP + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	WKUPDEP_DISPC_MPU_M3	Wakeup dependency from DISPC module (softwareakeup_DISPC signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_DISPC_MPU	Wakeup dependency from DISPC module (softwareakeup_DISPC signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-524. Register Call Summary for Register PM\_DSS\_DSS\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

PRCM Register Manual

- [DSS\\_PRM Register Summary: \[16\]](#)

**Table 3-525. RM\_DSS\_DSS\_CONTEXT**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	DSS_PRM
<b>Physical Address</b>	See <a href="#">Table 3-518</a>		
<b>Description</b>	This register contains dedicated DSS context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_DSS_MEM	RESERVED						LOSTCONTEXT_RFF	LOSTCONTEXT_DFF							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_DSS_MEM	Specify if memory-based context in DSS_MEM memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSS_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSS_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-526. Register Call Summary for Register RM\_DSS\_DSS\_CONTEXT**

Power Management Functional Description

- [PD\\_DSS Description: \[0\] \[1\]](#)

PRCM Register Manual

- [DSS\\_PRM Register Summary: \[2\]](#)

### 3.11.12 SGX\_PRM Registers

#### 3.11.12.1 SGX\_PRM Register Summary

**Table 3-527. SGX\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SGX_PRM L4 Base Address
<a href="#">PM_SGX_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 7200
<a href="#">PM_SGX_PWRSTST</a>	R	32	0x0000 0004	0x4A30 7204
<a href="#">RM_SGX_SGX_CONTEXT</a>	RW	32	0x0000 0024	0x4A30 7224

**3.11.12.2 SGX\_PRM Register Description**

**Table 3-528. PM\_SGX\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	SGX_PRM
<b>Physical Address</b>	See <a href="#">Table 3-527</a>		
<b>Description</b>	This register controls the SGX power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SGX_MEM_ONSTATE	RESERVED								LOWPOWERSTATECHANGE	RESERVED	POWERSTATE												

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	SGX_MEM_ONSTATE	SGX_MEM memory bank state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
15:5	RESERVED		R	0x000
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain.  0x0: Do not request a low power state change.  0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control  0x0: OFF state  0x1: Reserved  0x2: INACTIVE state  0x3: ON State	RW	0x0

**Table 3-529. Register Call Summary for Register PM\_SGX\_PWRSTCTRL**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [SGX\\_PRM Register Summary: \[3\]](#)

**Table 3-530. PM\_SGX\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	SGX_PRM
<b>Physical Address</b>	See <a href="#">Table 3-527</a>		
<b>Description</b>	This register provides a status on the current SGX power domain state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INTRANSITION	RESERVED								SGX_MEM_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST											

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:6	RESERVED		R	0x0000
5:4	SGX_MEM_STATEST	SGX_MEM memory bank state status Read 0x0: Memory is OFF Read 0x1: Reserved Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-531. Register Call Summary for Register PM\_SGX\_PWRSTST**

## Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\]](#)

## PRCM Register Manual

- [SGX\\_PRM Register Summary: \[4\]](#)

**Table 3-532. RM\_SGX\_SGX\_CONTEXT**

<b>Address Offset</b>	0x0000 0024		
<b>Physical Address</b>	See <a href="#">Table 3-527</a>	<b>Instance</b>	SGX_PRM
<b>Description</b>	This register contains dedicated SGX context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_SGX_MEM	RESERVED										LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_SGX_MEM	Specify if memory-based context in SGX_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of SGX_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-533. Register Call Summary for Register RM\_SGX\_SGX\_CONTEXT**

Power Management Functional Description

- [PD\\_SGX Description: \[0\]](#)

PRCM Register Manual

- [SGX\\_PRM Register Summary: \[1\]](#)

### 3.11.13 L3INIT\_PRM Registers

#### 3.11.13.1 L3INIT\_PRM Register Summary

**Table 3-534. L3INIT\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	L3INIT_PRM L4 Base Address
<a href="#">PM_L3INIT_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 7300
<a href="#">PM_L3INIT_PWRSTST</a>	R	32	0x0000 0004	0x4A30 7304
<a href="#">PM_L3INIT_MMC1_WKDEP</a>	RW	32	0x0000 0028	0x4A30 7328
<a href="#">RM_L3INIT_MMC1_CONTEXT</a>	RW	32	0x0000 002C	0x4A30 732C
<a href="#">PM_L3INIT_MMC2_WKDEP</a>	RW	32	0x0000 0030	0x4A30 7330
<a href="#">RM_L3INIT_MMC2_CONTEXT</a>	RW	32	0x0000 0034	0x4A30 7334
<a href="#">PM_L3INIT_HSI_WKDEP</a>	RW	32	0x0000 0038	0x4A30 7338
<a href="#">RM_L3INIT_HSI_CONTEXT</a>	RW	32	0x0000 003C	0x4A30 733C

**Table 3-534. L3INIT\_PRM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	L3INIT_PRM L4 Base Address
RESERVED	RW	32	0x0000 0040	0x4A30 7340
RESERVED	RW	32	0x0000 0044	0x4A30 7344
PM_L3INIT_HSUSBHOST_WKDEP	RW	32	0x0000 0058	0x4A30 7358
RM_L3INIT_HSUSBHOST_CONTEXT	RW	32	0x0000 005C	0x4A30 735C
PM_L3INIT_HSUSBOTG_WKDEP	RW	32	0x0000 0060	0x4A30 7360
RM_L3INIT_HSUSBOTG_CONTEXT	RW	32	0x0000 0064	0x4A30 7364
PM_L3INIT_HSUSBTLL_WKDEP	RW	32	0x0000 0068	0x4A30 7368
RM_L3INIT_HSUSBTLL_CONTEXT	RW	32	0x0000 006C	0x4A30 736C
RM_L3INIT_P1500_CONTEXT	RW	32	0x0000 007C	0x4A30 737C
RESERVED	RW	32	0x0000 0084	0x4A30 7384
RESERVED	RW	32	0x0000 0088	0x4A30 7388
RESERVED	RW	32	0x0000 008C	0x4A30 738C
RESERVED	RW	32	0x0000 0094	0x4A30 7394
RESERVED	RW	32	0x0000 0098	0x4A30 7398
RESERVED	RW	32	0x0000 009C	0x4A30 739C
RESERVED	RW	32	0x0000 00AC	0x4A30 73AC
RESERVED	R	32	0x0000 00C0	0x4A30 73C0
RESERVED	RW	32	0x0000 00C4	0x4A30 73C4
RESERVED	RW	32	0x0000 00C8	0x4A30 73C8
RESERVED	RW	32	0x0000 00CC	0x4A30 73CC
PM_L3INIT_FSUSB_WKDEP	RW	32	0x0000 00D0	0x4A30 73D0
RM_L3INIT_FSUSB_CONTEXT	RW	32	0x0000 00D4	0x4A30 73D4
RM_L3INIT_USBPHY_CONTEXT	RW	32	0x0000 00E4	0x4A30 73E4

**3.11.13.2 L3INIT\_PRM Register Description****Table 3-535. PM\_L3INIT\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	L3INIT_PRM
<b>Physical Address</b>	See <a href="#">Table 3-534</a>		
<b>Description</b>	This register controls the L3INIT power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								RESERVED								RESERVED	LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE			
								L3INIT_BANK1_ONSTATE																L3INIT_BANK1_RETSTATE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	L3INIT_BANK1_ONSTATE	L3INIT BANK state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
15:9	RESERVED		R	0x00
8	L3INIT_BANK1_RETSTATE	L3INIT BANK1 state when domain is RETENTION. Read 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3	RESERVED		R	0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remain logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	1
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x0

**Table 3-536. Register Call Summary for Register PM\_L3INIT\_PWRSTCTRL**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[5\]](#)

**Table 3-537. PM\_L3INIT\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	L3INIT_PRM
<b>Physical Address</b>	See <a href="#">Table 3-534</a>		
<b>Description</b>	This register provides a status on the current L3INIT power domain state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED		INTRANSITION		RESERVED										L3INIT_BANK1_STATE		RESERVED		LOGICSTATE		POWERSTATE			
LASTPOWERSTATEENTERED																															

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only.  Read 0x3: Power domain was previously ON-ACTIVE Read 0x2: Power domain was previously ON-INACTIVE Read 0x1: Power domain was previously in RETENTION Read 0x0: Power domain was previously OFF	RW W1toSet	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:6	RESERVED		R	0x0000
5:4	L3INIT_BANK1_STATEST	L3INIT BANK1 state status Read 0x0: Memory is OFF Read 0x1: Reserved Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-538. Register Call Summary for Register PM\_L3INIT\_PWRSTST**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[5\]](#)

**Table 3-539. PM\_L3INIT\_MMC1\_WKDEP**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	L3INIT_PRM
<b>Physical Address</b>	See <a href="#">Table 3-534</a>		
<b>Description</b>	This register controls wakeup dependency based on MMC1 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MMC1_SDMA	WKUPDEP_MMC1_DSP	WKUPDEP_MMC1_MPU_M3	WKUPDEP_MMC1_MPU												



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3	WKUPDEP_MMC1_SDMA	Wakeup dependency from MMC1 module (softwareakeup signal) towards SDMA + L3_2 + L4PER domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	WKUPDEP_MMC1_DSP	Wakeup dependency from MMC1 module (softwareakeup signal) towards DSP + L3_1 + L3_2 + L4_PER domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	WKUPDEP_MMC1_MPU_M3	Wakeup dependency from MMC1 module (softwareakeup signal) towards MPU_A3 + L3_2 + L4_PER domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_MMC1_MPU	Wakeup dependency from MMC1 module (softwareakeup signal) towards MPU + L3_1 + L3_2 + L4_PER domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-540. Register Call Summary for Register PM\_L3INIT\_MMC1\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[4\]](#)

**Table 3-541. RM\_L3INIT\_MMC1\_CONTEXT**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	L3INIT_PRM
<b>Physical Address</b>	See <a href="#">Table 3-534</a>		
<b>Description</b>	This register contains dedicated MMC1 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_L3INIT_BANK1	RESERVED							LOSTCONTEXT_RFF	RESERVED						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0000000
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-542. Register Call Summary for Register RM\_L3INIT\_MMC1\_CONTEXT**

Power Management Functional Description

- [PD\\_L3\\_INIT Description: \[0\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[1\]](#)

**Table 3-543. PM\_L3INIT\_MMC2\_WKDEP**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	L3INIT_PRM
<b>Physical Address</b>	See <a href="#">Table 3-534</a>		
<b>Description</b>	This register controls wakeup dependency based on MMC2 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MMC2_SDMA	WKUPDEP_MMC2_DSP	WKUPDEP_MMC2_MPU_M3	WKUPDEP_MMC2_MPU												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MMC2_SDMA	Wakeup dependency from MMC2 module (softwareakeup signal) towards SDMA + L3_2 + L4PER domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	WKUPDEP_MMC2_DSP	Wakeup dependency from MMC2 module (softwareakeup signal) towards DSP + L3_1 + L3_2 + L4_PER domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	WKUPDEP_MMC2_MPU_M3	Wakeup dependency from MMC2 module (softwareakeup signal) towards MPU_A3 + L3_2 + L4_PER domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_MMC2_MPU	Wakeup dependency from MMC2 module (softwareakeup signal) towards MPU + L3_1 + L3_2 + L4_PER domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-544. Register Call Summary for Register PM\_L3INIT\_MMC2\_WKDEP**

- Clock Management Functional Description
- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)
- PRCM Register Manual
- [L3INIT\\_PRM Register Summary: \[4\]](#)

**Table 3-545. RM\_L3INIT\_MMC2\_CONTEXT**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	L3INIT_PRM
<b>Physical Address</b>	See <a href="#">Table 3-534</a>		
<b>Description</b>	This register contains dedicated MMC2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_L3INIT_BANK1	RESERVED						LOSTCONTEXT_RFF	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-546. Register Call Summary for Register RM\_L3INIT\_MMC2\_CONTEXT**

- Power Management Functional Description
- [PD\\_L3\\_INIT Description: \[0\]](#)
- PRCM Register Manual
- [L3INIT\\_PRM Register Summary: \[1\]](#)

**Table 3-547. PM\_L3INIT\_HSI\_WKDEP**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-534</a>	<b>Instance</b>	L3INIT_PRM
<b>Description</b>	This register controls wakeup dependency based on HSI service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_WGM_HSI_WAKE_MPU	RESERVED	WKUPDEP_HSI_DSP_DSP	RESERVED				WKUPDEP_HSI_MCU_MPU_M3	WKUPDEP_HSI_MCU_MPU							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	WKUPDEP_WGM_HSI_WAKE_MPU	Wakeup dependency from modem HSI_WAKE signal towards MPU + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
7	RESERVED		R	0
6	WKUPDEP_HSI_DSP_DSP	Wakeup dependency from HSI module (softwareakeup_DSP signal) towards DSP + L3_1 + L4_CFG domains Read 0x1: Dependency is enabled	R Rreturns 1s	1
5:2	RESERVED		R	0x0
1	WKUPDEP_HSI_MCU_MPU_M3	Wakeup dependency from HSI module (softwareakeup_MPU signal) towards MPU_A3 + L3_2 + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_HSI_MCU_MPU	Wakeup dependency from HSI module (softwareakeup_MPU signal) towards MPU + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-548. Register Call Summary for Register PM\_L3INIT\_HSI\_WKDEP**

## Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

## PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[4\]](#)

**Table 3-549. RM\_L3INIT\_HSI\_CONTEXT**

<b>Address Offset</b>	0x0000 003C		
<b>Physical Address</b>	See <a href="#">Table 3-534</a>	<b>Instance</b>	L3INIT_PRM
<b>Description</b>	This register contains dedicated HSI context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_L3INIT_BANK1	RESERVED							LOSTCONTEXT_RFF	RESERVED						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-550. Register Call Summary for Register RM\_L3INIT\_HSI\_CONTEXT**

Power Management Functional Description

- [PD\\_L3\\_INIT Description: \[0\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[1\]](#)

**Table 3-551. PM\_L3INIT\_HSUSBHOST\_WKDEP**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Table 3-534</a>	<b>Instance</b>	L3INIT_PRM
<b>Description</b>	This register controls wakeup dependency based on USB_HOST_HS service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_HSUSBHOST_MPU_M3	WKUPDEP_HSUSBHOST_MPU		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_HSUSBHOST_MPU_M3	Wakeup dependency from USB_HOST_HS module (softwareakeup signal) towards MPU_A3 + L3_2 + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_HSUSBHOST_MPU	Wakeup dependency from USB_HOST_HS module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-552. Register Call Summary for Register PM\_L3INIT\_HSUSBHOST\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[2\]](#)

**Table 3-553. RM\_L3INIT\_HSUSBHOST\_CONTEXT**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	L3INIT_PRM
<b>Physical Address</b>	See <a href="#">Table 3-534</a>		
<b>Description</b>	This register contains dedicated USB_HOST context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_RFF	RESERVED		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-554. Register Call Summary for Register RM\_L3INIT\_HSUSBHOST\_CONTEXT**

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[0\]](#)

**Table 3-555. PM\_L3INIT\_HSUSBOTG\_WKDEP**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	L3INIT_PRM
<b>Physical Address</b>	See <a href="#">Table 3-534</a>		
<b>Description</b>	This register controls wakeup dependency based on USB_OTG_HS service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_HSUSBOTG_MPU_M3		WKUPDEP_HSUSBOTG_MPU													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_HSUSBOTG_MPU_M3	Wakeup dependency from USB_OTG_HS module (softwareakeup signal) towards MPU_A3 + L3_2 + L3_1 + L4_CFG domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_HSUSBOTG_MPU	Wakeup dependency from USB_OTG_HS module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-556. Register Call Summary for Register PM\_L3INIT\_HSUSBOTG\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[2\]](#)

**Table 3-557. RM\_L3INIT\_HSUSBOTG\_CONTEXT**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See <a href="#">Table 3-534</a>	<b>Instance</b>	L3INIT_PRM
<b>Description</b>	This register contains dedicated USB_OTG_HS context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_L3INIT_BANK1	RESERVED								LOSTCONTEXT_RFF	RESERVED					

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-558. Register Call Summary for Register RM\_L3INIT\_HSUSBOTG\_CONTEXT**

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[0\]](#)

**Table 3-559. PM\_L3INIT\_HSUSBTLL\_WKDEP**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See <a href="#">Table 3-534</a>	<b>Instance</b>	L3INIT_PRM
<b>Description</b>	This register controls wakeup dependency based on USB_TLL service requests.		
<b>Type</b>	RW		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_HSUSBTLL_MPU_M3	WKUPDEP_HSUSBTLL_MPU		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_HSUSBTLL_MPU_M3	Wakeup dependency from USB_TLL module (softwareakeup signal) towards MPU_A3 + L3_2 + L3_1 + L4_CFG domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_HSUSBTLL_MPU	Wakeup dependency from USB_TLL module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-560. Register Call Summary for Register PM\_L3INIT\_HSUSBTLL\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[2\]](#)

**Table 3-561. RM\_L3INIT\_HSUSBTLL\_CONTEXT**

<b>Address Offset</b>	0x0000 006C
<b>Physical Address</b>	See <a href="#">Table 3-534</a>
<b>Description</b>	This register contains dedicated USB_TLL context statuses. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_RFF	RESERVED		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-562. Register Call Summary for Register RM\_L3INIT\_HSUSBTLL\_CONTEXT**

Power Management Functional Description

- [PD\\_L3\\_INIT Description: \[0\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[1\]](#)

**Table 3-563. RM\_L3INIT\_P1500\_CONTEXT**

<b>Address Offset</b>	0x0000 007C	<b>Instance</b>	L3INIT_PRM
<b>Physical Address</b>	See <a href="#">Table 3-534</a>		
<b>Description</b>	This register contains dedicated P1500 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
LOSTCONTEXT_DFF																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-564. Register Call Summary for Register RM\_L3INIT\_P1500\_CONTEXT**

Power Management Functional Description

- [PD\\_L3\\_INIT Description: \[0\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[1\]](#)

**Table 3-565. PM\_L3INIT\_FSUSB\_WKDEP**

<b>Address Offset</b>	0x0000 00D0		
<b>Physical Address</b>	See <a href="#">Table 3-534</a>	<b>Instance</b>	L3INIT_PRM
<b>Description</b>	This register controls wakeup dependency based on USB_HOST_FS service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_FSUSB_MPU_M3		WKUPDEP_FSUSB_MPU													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_FSUSB_MPU_M3	Wakeup dependency from USB_HOST_FS module (softwareakeup signal) towards MPU_A3 + L3_2 + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_FSUSB_MPU	Wakeup dependency from USB_HOST_FS module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-566. Register Call Summary for Register PM\_L3INIT\_FSUSB\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[2\]](#)

**Table 3-567. RM\_L3INIT\_FSUSB\_CONTEXT**

<b>Address Offset</b>	0x0000 00D4		
<b>Physical Address</b>	See <a href="#">Table 3-534</a>	<b>Instance</b>	L3INIT_PRM
<b>Description</b>	This register contains dedicated USB_HOST_FS context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		RESERVED													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-568. Register Call Summary for Register RM\_L3INIT\_FSUSB\_CONTEXT**

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[0\]](#)

**Table 3-569. RM\_L3INIT\_USBPHY\_CONTEXT**

<b>Address Offset</b>	0x0000 00E4		
<b>Physical Address</b>	See <a href="#">Table 3-534</a>	<b>Instance</b>	L3INIT_PRM
<b>Description</b>	This register contains dedicated USBPHY context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3_INIT_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-570. Register Call Summary for Register RM\_L3INIT\_USBPHY\_CONTEXT**

Power Management Functional Description

- [PD\\_L3\\_INIT Description: \[0\]](#)

PRCM Register Manual

- [L3INIT\\_PRM Register Summary: \[1\]](#)

### 3.11.14 L4PER\_PRM Registers

#### 3.11.14.1 L4PER\_PRM Register Summary

**Table 3-571. L4PER\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM L4 Base Address
<a href="#">PM_L4PER_PWRSTCTRL</a>	RW	32	0x0000 0000	0x4A30 7400

**Table 3-571. L4PER\_PRM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM L4 Base Address
PM_L4PER_PWRSTST	R	32	0x0000 0004	0x4A30 7404
RESERVED	RW	32	0x0000 0024	0x4A30 7424
PM_L4PER_GPTIMER10_WKDEP	R	32	0x0000 0028	0x4A30 7428
RM_L4PER_GPTIMER10_CONTEXT	RW	32	0x0000 002C	0x4A30 742C
PM_L4PER_GPTIMER11_WKDEP	RW	32	0x0000 0030	0x4A30 7430
RM_L4PER_GPTIMER11_CONTEXT	RW	32	0x0000 0034	0x4A30 7434
PM_L4PER_GPTIMER2_WKDEP	R	32	0x0000 0038	0x4A30 7438
RM_L4PER_GPTIMER2_CONTEXT	RW	32	0x0000 003C	0x4A30 743C
PM_L4PER_GPTIMER3_WKDEP	RW	32	0x0000 0040	0x4A30 7440
RM_L4PER_GPTIMER3_CONTEXT	RW	32	0x0000 0044	0x4A30 7444
PM_L4PER_GPTIMER4_WKDEP	RW	32	0x0000 0048	0x4A30 7448
RM_L4PER_GPTIMER4_CONTEXT	RW	32	0x0000 004C	0x4A30 744C
PM_L4PER_GPTIMER9_WKDEP	RW	32	0x0000 0050	0x4A30 7450
RM_L4PER_GPTIMER9_CONTEXT	RW	32	0x0000 0054	0x4A30 7454
RM_L4PER_ELM_CONTEXT	RW	32	0x0000 005C	0x4A30 745C
PM_L4PER_GPIO2_WKDEP	RW	32	0x0000 0060	0x4A30 7460
RM_L4PER_GPIO2_CONTEXT	RW	32	0x0000 0064	0x4A30 7464
PM_L4PER_GPIO3_WKDEP	R	32	0x0000 0068	0x4A30 7468
RM_L4PER_GPIO3_CONTEXT	RW	32	0x0000 006C	0x4A30 746C
PM_L4PER_GPIO4_WKDEP	R	32	0x0000 0070	0x4A30 7470
RM_L4PER_GPIO4_CONTEXT	RW	32	0x0000 0074	0x4A30 7474
PM_L4PER_GPIO5_WKDEP	R	32	0x0000 0078	0x4A30 7478
RM_L4PER_GPIO5_CONTEXT	RW	32	0x0000 007C	0x4A30 747C
PM_L4PER_GPIO6_WKDEP	R	32	0x0000 0080	0x4A30 7480
RM_L4PER_GPIO6_CONTEXT	RW	32	0x0000 0084	0x4A30 7484
RM_L4PER_HDQ1W_CONTEXT	RW	32	0x0000 008C	0x4A30 748C
RESERVED	R	32	0x0000 0090	0x4A30 7490
RESERVED	RW	32	0x0000 0094	0x4A30 7494
RESERVED	R	32	0x0000 0098	0x4A30 7498
RESERVED	RW	32	0x0000 009C	0x4A30 749C
PM_L4PER_I2C1_WKDEP	RW	32	0x0000 00A0	0x4A30 74A0
RM_L4PER_I2C1_CONTEXT	RW	32	0x0000 00A4	0x4A30 74A4
PM_L4PER_I2C2_WKDEP	RW	32	0x0000 00A8	0x4A30 74A8
RM_L4PER_I2C2_CONTEXT	RW	32	0x0000 00AC	0x4A30 74AC
PM_L4PER_I2C3_WKDEP	RW	32	0x0000 00B0	0x4A30 74B0
RM_L4PER_I2C3_CONTEXT	RW	32	0x0000 00B4	0x4A30 74B4
PM_L4PER_I2C4_WKDEP	RW	32	0x0000 00B8	0x4A30 74B8
RM_L4PER_I2C4_CONTEXT	RW	32	0x0000 00BC	0x4A30 74BC
RM_L4PER_L4_PER_CONTEXT	RW	32	0x0000 00C0	0x4A30 74C0
RESERVED	RW	32	0x0000 00D0	0x4A30 74D0
RESERVED	RW	32	0x0000 00D4	0x4A30 74D4
RESERVED	RW	32	0x0000 00D8	0x4A30 74D8
RESERVED	RW	32	0x0000 00DC	0x4A30 74DC
PM_L4PER_MCBSP4_WKDEP	RW	32	0x0000 00E0	0x4A30 74E0
RM_L4PER_MCBSP4_CONTEXT	RW	32	0x0000 00E4	0x4A30 74E4
RESERVED	RW	32	0x0000 00EC	0x4A30 74EC

**Table 3-571. L4PER\_PRM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM L4 Base Address
PM_L4PER_MCSP11_WKDEP	RW	32	0x0000 00F0	0x4A30 74F0
RM_L4PER_MCSP11_CONTEXT	RW	32	0x0000 00F4	0x4A30 74F4
PM_L4PER_MCSP12_WKDEP	RW	32	0x0000 00F8	0x4A30 74F8
RM_L4PER_MCSP12_CONTEXT	RW	32	0x0000 00FC	0x4A30 74FC
PM_L4PER_MCSP13_WKDEP	RW	32	0x0000 0100	0x4A30 7500
RM_L4PER_MCSP13_CONTEXT	RW	32	0x0000 0104	0x4A30 7504
PM_L4PER_MCSP14_WKDEP	RW	32	0x0000 0108	0x4A30 7508
RM_L4PER_MCSP14_CONTEXT	RW	32	0x0000 010C	0x4A30 750C
PM_L4PER_MMCS13_WKDEP	RW	32	0x0000 0120	0x4A30 7520
RM_L4PER_MMCS13_CONTEXT	RW	32	0x0000 0124	0x4A30 7524
PM_L4PER_MMCS14_WKDEP	RW	32	0x0000 0128	0x4A30 7528
RM_L4PER_MMCS14_CONTEXT	RW	32	0x0000 012C	0x4A30 752C
RESERVED	RW	32	0x0000 0134	0x4A30 7534
PM_L4PER_SLIMBUS2_WKDEP	RW	32	0x0000 0138	0x4A30 7538
RM_L4PER_SLIMBUS2_CONTEXT	RW	32	0x0000 013C	0x4A30 753C
PM_L4PER_UART1_WKDEP	RW	32	0x0000 0140	0x4A30 7540
RM_L4PER_UART1_CONTEXT	RW	32	0x0000 0144	0x4A30 7544
PM_L4PER_UART2_WKDEP	RW	32	0x0000 0148	0x4A30 7548
RM_L4PER_UART2_CONTEXT	RW	32	0x0000 014C	0x4A30 754C
PM_L4PER_UART3_WKDEP	RW	32	0x0000 0150	0x4A30 7550
RM_L4PER_UART3_CONTEXT	RW	32	0x0000 0154	0x4A30 7554
PM_L4PER_UART4_WKDEP	RW	32	0x0000 0158	0x4A30 7558
RM_L4PER_UART4_CONTEXT	RW	32	0x0000 015C	0x4A30 755C
PM_L4PER_MMCS15_WKDEP	RW	32	0x0000 0160	0x4A30 7560
RM_L4PER_MMCS15_CONTEXT	RW	32	0x0000 0164	0x4A30 7564
RESERVED	R	32	0x0000 0168	0x4A30 7568
RESERVED	RW	32	0x0000 016C	0x4A30 756C
RESERVED	RW	32	0x0000 01A4	0x4A30 75A4
RESERVED	RW	32	0x0000 01AC	0x4A30 75AC
RESERVED	RW	32	0x0000 01B4	0x4A30 75B4
RESERVED	RW	32	0x0000 01BC	0x4A30 75BC
RESERVED	RW	32	0x0000 01C4	0x4A30 75C4
RESERVED	RW	32	0x0000 01CC	0x4A30 75CC
RESERVED	RW	32	0x0000 01DC	0x4A30 75DC

**3.11.14.2 L4PER\_PRM Register Description**

**Table 3-572. PM\_L4PER\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls the L4PER power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								NONRETAINED_BANK_ONSTATE		RETAINED_BANK_ONSTATE		RESERVED						NONRETAINED_BANK_RETSTATE		RETAINED_BANK_RETSTATE		RESERVED		LOWPOWERSTATECHANGE		RESERVED		LOGICRETSTATE		POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:18	NONRETAINED_BANK_ONSTATE	NONRETAINED_BANK state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
17:16	RETAINED_BANK_ONSTATE	RETAINED_BANK state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Rreturns 1s	0x3
15:10	RESERVED		R	0x00
9	NONRETAINED_BANK_RETSTATE	NONRETAINED_BANK state when domain is RETENTION. Read 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0
8	RETAINED_BANK_RETSTATE	RETAINED_BANK state when domain is RETENTION. Read 0x1: Memory bank is retained when domain is in RETENTION state.	R Rreturns 1s	1
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW WSpecial	0
3	RESERVED		R	0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaining logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	1
1:0	POWERSTATE	Power state control 0x0: Reserved 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

**Table 3-573. Register Call Summary for Register PM\_L4PER\_PWRSTCTRL**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[7\]](#)

**Table 3-574. PM\_L4PER\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register provides a status on the current L4PER power domain state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LASTPOWERSTATEENTERED	RESERVED		INTRANSITION	RESERVED								NONRETAINED_BANK_STATE	RETAINED_BANK_STATE		RESERVED	LOGICSTATE	POWERSTATE						

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only.  Read 0x3: Power domain was previously ON-ACTIVE Read 0x2: Power domain was previously ON-INACTIVE Read 0x1: Power domain was previously in RETENTION Read 0x0: Power domain was previously OFF	RW W1toSet	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status  Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:8	RESERVED		R	0x000
7:6	NONRETAINED_BANK_STATE	NONRETAINED_BANK state status  Read 0x0: Memory is OFF Read 0x1: Reserved Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
5:4	RETAINED_BANK_STATE	RETAINED_BANK state status  Read 0x0: Memory is OFF Read 0x1: Memory is RETENTION Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED		R	0



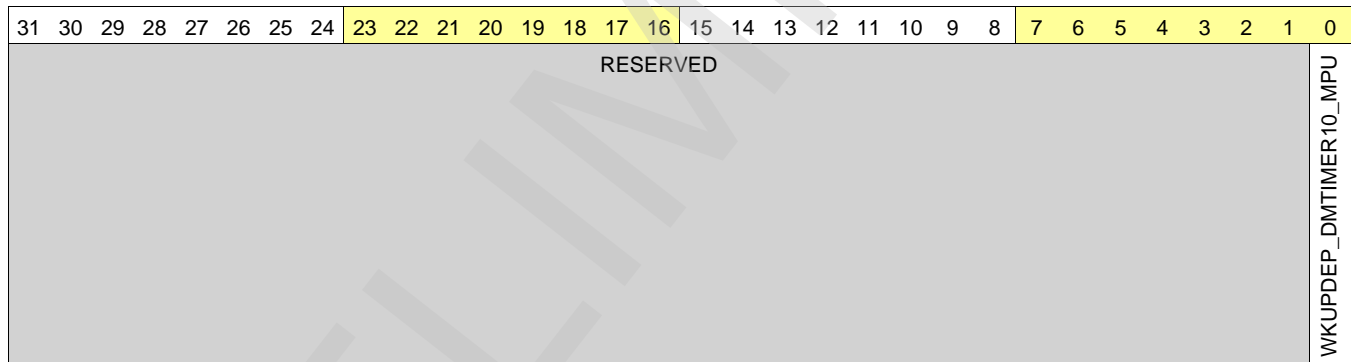
Bits	Field Name	Description	Type	Reset
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1
1:0	POWERSTATEST	Current power state status Read 0x0: Reserved Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-575. Register Call Summary for Register PM\_L4PER\_PWRSTST**

- Power Management Functional Description
- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- PRCM Register Manual
- [L4PER\\_PRM Register Summary: \[6\]](#)

**Table 3-576. PM\_L4PER\_GPTIMER10\_WKDEP**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register controls wakeup dependency based on DMTIMER10 service requests.		
<b>Type</b>	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	WKUPDEP_DMTIMER10_MPU	Wakeup dependency from DMTIMER10 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains Read 0x1: Dependency is enabled	R Rreturns 1s	1

**Table 3-577. Register Call Summary for Register PM\_L4PER\_GPTIMER10\_WKDEP**

- Clock Management Functional Description
- [Clock Domain Dependency: \[0\]](#)
- PRCM Register Manual
- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-578. RM\_L4PER\_GPTIMER10\_CONTEXT**

<b>Address Offset</b>	0x0000 002C		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated DMTIMER10 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-579. Register Call Summary for Register RM\_L4PER\_GPTIMER10\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-580. PM\_L4PER\_GPTIMER11\_WKDEP**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on DMTIMER11 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_DMTIMER11_MPU_M3 WKUPDEP_DMTIMER11_MPU															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_DMTIMER11_MPU_M3	Wakeup dependency from DMTIMER11 module (softwareakeup signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_DMTIMER11_MPU	Wakeup dependency from DMTIMER11 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-581. Register Call Summary for Register PM\_L4PER\_GPTIMER11\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-582. RM\_L4PER\_GPTIMER11\_CONTEXT**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated DMTIMER11 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-583. Register Call Summary for Register RM\_L4PER\_GPTIMER11\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-584. PM\_L4PER\_GPTIMER2\_WKDEP**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on DMTIMER2 service requests.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_DMTIMER2_MPU			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	WKUPDEP_DMTIMER2_MPU	Wakeup dependency from DMTIMER2 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains Read 0x1: Dependency is enabled	R Rreturns 1s	1

**Table 3-585. Register Call Summary for Register PM\_L4PER\_GPTIMER2\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-586. RM\_L4PER\_GPTIMER2\_CONTEXT**

<b>Address Offset</b>	0x0000 003C		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated DMTIMER2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-587. Register Call Summary for Register RM\_L4PER\_GPTIMER2\_CONTEXT**

Power Management Functional Description

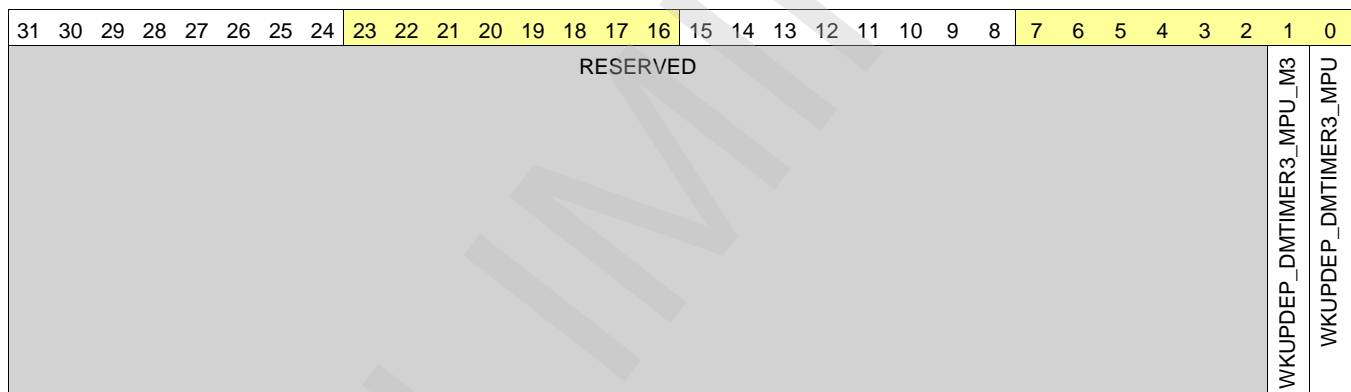
- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-588. PM\_L4PER\_GPTIMER3\_WKDEP**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register controls wakeup dependency based on DMTIMER3 service requests.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_DMTIMER3_MPU_M3	Wakeup dependency from DMTIMER3 module (softwareakeup signal) towards MPU_A3 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_DMTIMER3_MPU	Wakeup dependency from DMTIMER3 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-589. Register Call Summary for Register PM\_L4PER\_GPTIMER3\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-590. RM\_L4PER\_GPTIMER3\_CONTEXT**

<b>Address Offset</b>	0x0000 0044		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated DMTIMER3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-591. Register Call Summary for Register RM\_L4PER\_GPTIMER3\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-592. PM\_L4PER\_GPTIMER4\_WKDEP**

<b>Address Offset</b>	0x0000 0048		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on DMTIMER4 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_DMTIMER4_MPU_M3	WKUPDEP_DMTIMER4_MPU		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_DMTIMER4_MPU_M3	Wakeup dependency from DMTIMER4 module (softwareakeup signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_DMTIMER4_MPU	Wakeup dependency from DMTIMER4 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-593. Register Call Summary for Register PM\_L4PER\_GPTIMER4\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-594. RM\_L4PER\_GPTIMER4\_CONTEXT**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated DMTIMER4 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-595. Register Call Summary for Register RM\_L4PER\_GPTIMER4\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-596. PM\_L4PER\_GPTIMER9\_WKDEP**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on DMTIMER9 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_DMTIMER9_MPU_M3	WKUPDEP_DMTIMER9_MPU		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_DMTIMER9_MPU_M3	Wakeup dependency from DMTIMER9 module (softwareakeup signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_DMTIMER9_MPU	Wakeup dependency from DMTIMER9 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-597. Register Call Summary for Register PM\_L4PER\_GPTIMER9\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-598. RM\_L4PER\_GPTIMER9\_CONTEXT**

<b>Address Offset</b>	0x0000 0054		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated DMTIMER9 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-599. Register Call Summary for Register RM\_L4PER\_GPTIMER9\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-600. RM\_L4PER\_ELM\_CONTEXT**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated ELM context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-601. Register Call Summary for Register RM\_L4PER\_ELM\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-602. PM\_L4PER\_GPIO2\_WKDEP**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on GPIO2 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_GPIO2_IRQ2_DSP		RESERVED					WKUPDEP_GPIO2_IRQ1_MPU_M3	WKUPDEP_GPIO2_IRQ1_MPU							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	WKUPDEP_GPIO2_IRQ2_DSP	Wakeup dependency from GPIO2 module (POINTRsoftwareAKEUP2 signal) towards DSP + L3_1 + L3_2 domains Read 0x1: Dependency is enabled	R Returns 1s	1
5:2	RESERVED		R	0x0
1	WKUPDEP_GPIO2_IRQ1_MPU_M3	Wakeup dependency from GPIO2 module (POINTRsoftwareAKEUP1 signal) module towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_GPIO2_IRQ1_MPU	Wakeup dependency from GPIO2 module (POINTRsoftwareAKEUP1 signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-603. Register Call Summary for Register PM\_L4PER\_GPIO2\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency](#): [0] [1] [2]

PRCM Register Manual

- [L4PER\\_PRM Register Summary](#): [3]

**Table 3-604. RM\_L4PER\_GPIO2\_CONTEXT**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated GPIO2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		RESERVED													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-605. Register Call Summary for Register RM\_L4PER\_GPIO2\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-606. PM\_L4PER\_GPIO3\_WKDEP**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on GPIO3 service requests.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_GPIO3_IRQ2_DSP	RESERVED		WKUPDEP_GPIO3_IRQ1_MPU												

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	WKUPDEP_GPIO3_IRQ2_DSP	Wakeup dependency from GPIO3 module (POINTRsoftwareAKEUP2 signal) towards DSP + L3_1 + L3_2 domains Read 0x1: Dependency is enabled	R Rreturns 1s	1
5:1	RESERVED		R	0x00
0	WKUPDEP_GPIO3_IRQ1_MPU	Wakeup dependency from GPIO3 module (POINTRsoftwareAKEUP1 signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-607. Register Call Summary for Register PM\_L4PER\_GPIO3\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-608. RM\_L4PER\_GPIO3\_CONTEXT**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated GPIO3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
																LOSTCONTEXT_RFF	RESERVED														

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-609. Register Call Summary for Register RM\_L4PER\_GPIO3\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-610. PM\_L4PER\_GPIO4\_WKDEP**

<b>Address Offset</b>	0x0000 0070		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on GPIO4 service requests.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_GPIO4_IRQ2_DSP		RESERVED						WKUPDEP_GPIO4_IRQ1_MPU							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	WKUPDEP_GPIO4_IRQ2_DSP	Wakeup dependency from GPIO4 module (POINTRsoftwareAKEUP2 signal) towards DSP + L3_1 + L3_2 domains Read 0x1: Dependency is enabled	R Returns 1s	1
5:1	RESERVED		R	0x00
0	WKUPDEP_GPIO4_IRQ1_MPU	Wakeup dependency from GPIO4 module (POINTRsoftwareAKEUP1 signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-611. Register Call Summary for Register PM\_L4PER\_GPIO4\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-612. RM\_L4PER\_GPIO4\_CONTEXT**

<b>Address Offset</b>	0x0000 0074		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated GPIO4 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF	RESERVED														

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-613. Register Call Summary for Register RM\_L4PER\_GPIO4\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-614. PM\_L4PER\_GPIO5\_WKDEP**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register controls wakeup dependency based on GPIO5 service requests.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_GPIO5_IRQ2_DSP		RESERVED				WKUPDEP_GPIO5_IRQ1_MPU									

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	WKUPDEP_GPIO5_IRQ2_DSP	Wakeup dependency from GPIO5 module (POINTRsoftwareAKEUP2 signal) towards DSP + L3_1 + L3_2 domains  Read 0x1: Dependency is enabled	R Returns 1s	1
5:1	RESERVED		R	0x00
0	WKUPDEP_GPIO5_IRQ1_MPU	Wakeup dependency from GPIO5 module (POINTRsoftwareAKEUP1 signal) towards MPU + L3_1 + L3_2 domains  Read 0x1: Dependency is enabled	R Returns 1s	1

**Table 3-615. Register Call Summary for Register PM\_L4PER\_GPIO5\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-616. RM\_L4PER\_GPIO5\_CONTEXT**

<b>Address Offset</b>	0x0000 007C		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated GPIO5 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		RESERVED													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-617. Register Call Summary for Register RM\_L4PER\_GPIO5\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-618. PM\_L4PER\_GPIO6\_WKDEP**

<b>Address Offset</b>	0x0000 0080		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on GPIO6 service requests.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_GPIO6_IRQ2_DSP	RESERVED		WKUPDEP_GPIO6_IRQ1_MPU												

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	WKUPDEP_GPIO6_IRQ2_DSP	Wakeup dependency from GPIO6 module (POINTRsoftwareAKEUP2 signal) towards DSP + L3_1 + L3_2 domains Read 0x1: Dependency is enabled	R Rreturns 1s	1
5:1	RESERVED		R	0x00
0	WKUPDEP_GPIO6_IRQ1_MPU	Wakeup dependency from GPIO6 module (POINTRsoftwareAKEUP1 signal) towards MPU + L3_1 + L3_2 domains Read 0x1: Dependency is enabled	R Rreturns 1s	1

**Table 3-619. Register Call Summary for Register PM\_L4PER\_GPIO6\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-620. RM\_L4PER\_GPIO6\_CONTEXT**

<b>Address Offset</b>	0x0000 0084	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated GPIO6 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
																RESERVED															
																RESERVED															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-621. Register Call Summary for Register RM\_L4PER\_GPIO6\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)



**Table 3-622. RM\_L4PER\_HDQ1W\_CONTEXT**

<b>Address Offset</b>	0x0000 008C		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated HDQ1W context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-623. Register Call Summary for Register RM\_L4PER\_HDQ1W\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-624. PM\_L4PER\_I2C1\_WKDEP**

<b>Address Offset</b>	0x0000 00A0		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on I2C1 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_I2C1_DMA_SDMA	RESERVED				WKUPDEP_I2C1_IRQ_MPU_M3	WKUPDEP_I2C1_IRQ_MPU									
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	WKUPDEP_I2C1_DMA_SDMA	Wakeup dependency from I2C1 module (softwareakeup_dma signal) towards SDMA + L3_2 domains Read 0x1: Dependency is enabled	R Rreturns 1s	1
6:2	RESERVED		R	0x00
1	WKUPDEP_I2C1_IRQ_MPU_M3	Wakeup dependency from I2C1 module (softwareakeup_irq signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_I2C1_IRQ_MPU	Wakeup dependency from I2C1 module (softwareakeup_irq signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-625. Register Call Summary for Register PM\_L4PER\_I2C1\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[3\]](#)

**Table 3-626. RM\_L4PER\_I2C1\_CONTEXT**

<b>Address Offset</b>	0x0000 00A4	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated I2C1 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
																RESERVED															
																LOSTCONTEXT_RFF															
																RESERVED															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-627. Register Call Summary for Register RM\_L4PER\_I2C1\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-628. PM\_L4PER\_I2C2\_WKDEP**

<b>Address Offset</b>	0x0000 00A8	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register controls wakeup dependency based on I2C2 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_I2C2_DMA_SDMA	RESERVED				WKUPDEP_I2C2_IRQ_MPU_M3	WKUPDEP_I2C2_IRQ_MPU									

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	WKUPDEP_I2C2_DMA_SDMA	Wakeup dependency from I2C2 module (softwareakeup_dma signal) towards SDMA + L3_2 domains Read 0x1: Dependency is enabled	R Rreturns 1s	1
6:2	RESERVED		R	0x00
1	WKUPDEP_I2C2_IRQ_MPU_M3	Wakeup dependency from I2C2 module (softwareakeup_irq signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_I2C2_IRQ_MPU	Wakeup dependency from I2C2 module (softwareakeup_irq signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-629. Register Call Summary for Register PM\_L4PER\_I2C2\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[3\]](#)

**Table 3-630. RM\_L4PER\_I2C2\_CONTEXT**

<b>Address Offset</b>	0x0000 00AC		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated I2C2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-631. Register Call Summary for Register RM\_L4PER\_I2C2\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-632. PM\_L4PER\_I2C3\_WKDEP**

<b>Address Offset</b>	0x0000 00B0		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on I2C3 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_I2C3_DMA_SDMA	RESERVED						WKUPDEP_I2C3_IRQ_MPU_M3	WKUPDEP_I2C3_IRQ_MPU							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	WKUPDEP_I2C3_DMA_SDMA	Wakeup dependency from I2C3 module (softwareakeup_dma signal) towards SDMA + L3_2 domains Read 0x1: Dependency is enabled	R Rreturns 1s	1
6:2	RESERVED		R	0x00
1	WKUPDEP_I2C3_IRQ_MPU_M3	Wakeup dependency from I2C3 module (softwareakeup_irq signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_I2C3_IRQ_MPU	Wakeup dependency from I2C3 module (softwareakeup_irq signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-633. Register Call Summary for Register PM\_L4PER\_I2C3\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[3\]](#)

**Table 3-634. RM\_L4PER\_I2C3\_CONTEXT**

<b>Address Offset</b>	0x0000 00B4	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated I2C3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-635. Register Call Summary for Register RM\_L4PER\_I2C3\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-636. PM\_L4PER\_I2C4\_WKDEP**

<b>Address Offset</b>	0x0000 00B8		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on I2C4 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_I2C4_DMA_SDMA	RESERVED						WKUPDEP_I2C4_IRQ_MPU_M3	WKUPDEP_I2C4_IRQ_MPU							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	WKUPDEP_I2C4_DMA_SDMA	Wakeup dependency from I2C4 module (softwareakeup_dma signal) towards SDMA + L3_2 domains Read 0x1: Dependency is enabled	R Rreturns 1s	1
6:2	RESERVED		R	0x00
1	WKUPDEP_I2C4_IRQ_MPU_M3	Wakeup dependency from I2C4 module (softwareakeup_irq signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_I2C4_IRQ_MPU	Wakeup dependency from I2C4 module (softwareakeup_irq signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-637. Register Call Summary for Register PM\_L4PER\_I2C4\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[3\]](#)

**Table 3-638. RM\_L4PER\_I2C4\_CONTEXT**

<b>Address Offset</b>	0x0000 00BC		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated I2C4 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-639. Register Call Summary for Register RM\_L4PER\_I2C4\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-640. RM\_L4PER\_L4\_PER\_CONTEXT**

<b>Address Offset</b>	0x0000 00C0		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated L4_PER context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_RFF	LOSTCONTEXT_DFF		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-641. Register Call Summary for Register RM\_L4PER\_L4\_PER\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-642. PM\_L4PER\_MCBSP4\_WKDEP**

<b>Address Offset</b>	0x0000 00E0	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register controls wakeup dependency based on MCBSP4 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											WKUPDEP_MCBSP4_SDMA	WKUPDEP_MCBSP4_DSP	RESERVED	WKUPDEP_MCBSP4_MPU	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MCBSP4_SDMA	Wakeup dependency from MCBSP4 module (softwareakeup signal) towards SDMA + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	WKUPDEP_MCBSP4_DSP	Wakeup dependency from MCBSP4 module (softwareakeup signal) towards DSP + L3_1 + L3_2 domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_MCBSP4_MPU	Wakeup dependency from MCBSP4 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0



**Table 3-643. Register Call Summary for Register PM\_L4PER\_MCBSP4\_WKDEP**

- Clock Management Functional Description
- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)
- PRCM Register Manual
- [L4PER\\_PRM Register Summary: \[3\]](#)

**Table 3-644. RM\_L4PER\_MCBSP4\_CONTEXT**

<b>Address Offset</b>	0x0000 00E4	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated MCBSP4 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_NONRETAINED_BANK	RESERVED								LOSTCONTEXT_DFF						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-645. Register Call Summary for Register RM\_L4PER\_MCBSP4\_CONTEXT**

- Power Management Functional Description
- [PD\\_L4\\_PER Description: \[0\]](#)
- PRCM Register Manual
- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-646. PM\_L4PER\_MCSP11\_WKDEP**

<b>Address Offset</b>	0x0000 00F0		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on MCSP11 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCSP11_SDMA		WKUPDEP_MCSP11_DSP		WKUPDEP_MCSP11_MPU_M3		WKUPDEP_MCSP11_MPU									

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MCSP11_SDMA	Wakeup dependency from MCSP11 module (softwareakeup signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	WKUPDEP_MCSP11_DSP	Wakeup dependency from MCSP11 module (softwareakeup signal) towards DSP + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	WKUPDEP_MCSP11_MPU_M3	Wakeup dependency from MCSP11 module (softwareakeup signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_MCSP11_MPU	Wakeup dependency from MCSP11 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-647. Register Call Summary for Register PM\_L4PER\_MCSP11\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[4\]](#)

**Table 3-648. RM\_L4PER\_MCSP11\_CONTEXT**

<b>Address Offset</b>	0x0000 00F4		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated MCSP11 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-649. Register Call Summary for Register RM\_L4PER\_MCSP11\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-650. PM\_L4PER\_MCSP12\_WKDEP**

<b>Address Offset</b>	0x0000 00F8		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on MCSP12 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_MCSP12_SDMA	RESERVED	WKUPDEP_MCSP12_MPU_M3	WKUPDEP_MCSP12_MPU

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MCSPi2_SDMA	Wakeup dependency from MCSPi2 module (softwareakeup signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	RESERVED		R	0
1	WKUPDEP_MCSPi2_MPU_M3	Wakeup dependency from MCSPi2 module (softwareakeup signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_MCSPi2_MPU	Wakeup dependency from MCSPi2 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-651. Register Call Summary for Register PM\_L4PER\_MCSPi2\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[3\]](#)

**Table 3-652. RM\_L4PER\_MCSPi2\_CONTEXT**

<b>Address Offset</b>	0x0000 00FC	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated MCSPi2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
LOSTCONTEXT_DFF																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-653. Register Call Summary for Register RM\_L4PER\_MCSPi2\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-654. PM\_L4PER\_MCSPi3\_WKDEP**

<b>Address Offset</b>	0x0000 0100		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on MCSPi3 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCSPi3_SDMA			RESERVED		WKUPDEP_MCSPi3_MPU										

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MCSPi3_SDMA	Wakeup dependency from MCSPi3 module (softwareakeup signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2:1	RESERVED		R	0x0
0	WKUPDEP_MCSPi3_MPU	Wakeup dependency from MCSPi3 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-655. Register Call Summary for Register PM\_L4PER\_MCSPi3\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-656. RM\_L4PER\_MCSPi3\_CONTEXT**

<b>Address Offset</b>	0x0000 0104		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated MCSPi3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-657. Register Call Summary for Register RM\_L4PER\_MCSPi3\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-658. PM\_L4PER\_MCSPi4\_WKDEP**

<b>Address Offset</b>	0x0000 0108	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register controls wakeup dependency based on MCSPi4 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCSPi4_SDMA			RESERVED		WKUPDEP_MCSPi4_MPU										

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MCSPi4_SDMA	Wakeup dependency from MCSPi4 module (softwareakeup signal) towards SDMA + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2:1	RESERVED		R	0x0
0	WKUPDEP_MCSPi4_MPU	Wakeup dependency from MCSPi4 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-659. Register Call Summary for Register PM\_L4PER\_MCSPi4\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-660. RM\_L4PER\_MCSPi4\_CONTEXT**

<b>Address Offset</b>	0x0000 010C		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated MCSPI4 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-661. Register Call Summary for Register RM\_L4PER\_MCSPi4\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-662. PM\_L4PER\_MMCSd3\_WKDEP**

<b>Address Offset</b>	0x0000 0120		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on MMCSd3 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_MMCSd3_SDMA	RESERVED	WKUPDEP_MMCSd3_MPU_M3	WKUPDEP_MMCSd3_MPU

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3	WKUPDEP_MMCS3_SDMA	Wakeup dependency from MMCS3 module (softwareakeup signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	RESERVED		R	0
1	WKUPDEP_MMCS3_MPU_M3	Wakeup dependency from MMCS3 module (softwareakeup signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_MMCS3_MPU	Wakeup dependency from MMCS3 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-663. Register Call Summary for Register PM\_L4PER\_MMCS3\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[3\]](#)

**Table 3-664. RM\_L4PER\_MMCS3\_CONTEXT**

<b>Address Offset</b>	0x0000 0124	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated MMCS3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_NONRETAINED_BANK	RESERVED										LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00



Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-665. Register Call Summary for Register RM\_L4PER\_MMCS4\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRCM Register Summary: \[1\]](#)

**Table 3-666. PM\_L4PER\_MMCS4\_WKDEP**

<b>Address Offset</b>	0x0000 0128	<b>Instance</b>	L4PER_PRCM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register controls wakeup dependency based on MMCS4 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
																WKUPDEP_MMCS4_SDMA	RESERVED	WKUPDEP_MMCS4_MPU_M3	WKUPDEP_MMCS4_MPU												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MMCS4_SDMA	Wakeup dependency from MMCS4 module (softwareakeup signal) towards SDMA + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	RESERVED		R	0
1	WKUPDEP_MMCS4_MPU_M3	Wakeup dependency from MMCS4 module (softwareakeup signal) towards MPU_A3 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_MMCS4_MPU	Wakeup dependency from MMCS4 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-667. Register Call Summary for Register PM\_L4PER\_MMCS4\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[3\]](#)

**Table 3-668. RM\_L4PER\_MMCS4\_CONTEXT**

<b>Address Offset</b>	0x0000 012C	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated MMCS4 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_NONRETAINED_BANK	RESERVED								LOSTCONTEXT_DFF						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-669. Register Call Summary for Register RM\_L4PER\_MMCS4\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-670. PM\_L4PER\_SLIMBUS2\_WKDEP**

<b>Address Offset</b>	0x0000 0138		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on SLIMBUS2 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_SLIMBUS2_DMA_SDMA	WKUPDEP_SLIMBUS2_DMA_DSP	RESERVED				WKUPDEP_SLIMBUS2_IRQ_DSP	RESERVED	WKUPDEP_SLIMBUS2_IRQ_MPU							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0000000
7	WKUPDEP_SLIMBUS2_DMA_S DMA	Wakeup dependency from SLIMBUS2 module (softwareakeup_dma signal) towards SDMA + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	WKUPDEP_SLIMBUS2_DMA_DSP	Wakeup dependency from SLIMBUS2 module (softwareakeup_dma signal) towards DSP + L3_1 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5:3	RESERVED		R	0x0
2	WKUPDEP_SLIMBUS2_IRQ_DSP	Wakeup dependency from SLIMBUS2 module (softwareakeup_irq signal) towards DSP + L3_1 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	WKUPDEP_SLIMBUS2_IRQ_MPU	Wakeup dependency from SLIMBUS2 module (softwareakeup_irq signal) towards MPU + L3_1 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-671. Register Call Summary for Register PM\_L4PER\_SLIMBUS2\_WKDEP**

- Clock Management Functional Description
- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)
- PRCM Register Manual
- [L4PER\\_PRM Register Summary: \[4\]](#)

**Table 3-672. RM\_L4PER\_SLIMBUS2\_CONTEXT**

<b>Address Offset</b>	0x0000 013C		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated SLIMBUS2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_NONRETAINED_BANK	RESERVED								LOSTCONTEXT_DFF						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-673. Register Call Summary for Register RM\_L4PER\_SLIMBUS2\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-674. PM\_L4PER\_UART1\_WKDEP**

<b>Address Offset</b>	0x0000 0140		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on UART1 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											WKUPDEP_UART1_SDMA	RESERVED	WKUPDEP_UART1_MPU		

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3	WKUPDEP_UART1_SDMA	Wakeup dependency from UART1 module (softwareakeup signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2:1	RESERVED		R	0x0
0	WKUPDEP_UART1_MPU	Wakeup dependency from UART1 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-675. Register Call Summary for Register PM\_L4PER\_UART1\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-676. RM\_L4PER\_UART1\_CONTEXT**

<b>Address Offset</b>	0x0000 0144
<b>Physical Address</b>	See <a href="#">Table 3-571</a>
<b>Description</b>	This register contains dedicated UART1 context statuses. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_RETAINED_BANK	RESERVED											LOSTCONTEXT_RFF	RESERVED		

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-677. Register Call Summary for Register RM\_L4PER\_UART1\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-678. PM\_L4PER\_UART2\_WKDEP**

<b>Address Offset</b>	0x0000 0148	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register controls wakeup dependency based on UART2 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_UART2_SDMA			RESERVED		WKUPDEP_UART2_MPU										

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_UART2_SDMA	Wakeup dependency from UART2 module (softwareakeup signal) towards SDMA + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2:1	RESERVED		R	0x0
0	WKUPDEP_UART2_MPU	Wakeup dependency from UART2 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-679. Register Call Summary for Register PM\_L4PER\_UART2\_WKDEP**

- Clock Management Functional Description
- [Clock Domain Dependency: \[0\] \[1\]](#)
- PRCM Register Manual
- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-680. RM\_L4PER\_UART2\_CONTEXT**

<b>Address Offset</b>	0x0000 014C	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated UART2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-681. Register Call Summary for Register RM\_L4PER\_UART2\_CONTEXT**

- Power Management Functional Description
- [PD\\_L4\\_PER Description: \[0\]](#)
- PRCM Register Manual
- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-682. PM\_L4PER\_UART3\_WKDEP**

<b>Address Offset</b>	0x0000 0150		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on UART3 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_UART3_SDMA				WKUPDEP_UART3_DSP		WKUPDEP_UART3_MPU_M3		WKUPDEP_UART3_MPU							

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_UART3_SDMA	Wakeup dependency from UART3 module (softwareakeup signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	WKUPDEP_UART3_DSP	Wakeup dependency from UART3 module (softwareakeup signal) towards DSP + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	WKUPDEP_UART3_MPU_M3	Wakeup dependency from UART3 module (softwareakeup signal) towards MPU_A3 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_UART3_MPU	Wakeup dependency from UART3 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-683. Register Call Summary for Register PM\_L4PER\_UART3\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[4\]](#)



**Table 3-684. RM\_L4PER\_UART3\_CONTEXT**

<b>Address Offset</b>	0x0000 0154		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated UART3 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-685. Register Call Summary for Register RM\_L4PER\_UART3\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-686. PM\_L4PER\_UART4\_WKDEP**

<b>Address Offset</b>	0x0000 0158		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register controls wakeup dependency based on UART4 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											WKUPDEP_UART4_SDMA	RESERVED	WKUPDEP_UART4_MPU		

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3	WKUPDEP_UART4_SDMA	Wakeup dependency from UART4 module (softwareakeup signal) towards SDMA + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2:1	RESERVED		R	0x0
0	WKUPDEP_UART4_MPU	Wakeup dependency from UART4 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-687. Register Call Summary for Register PM\_L4PER\_UART4\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[2\]](#)

**Table 3-688. RM\_L4PER\_UART4\_CONTEXT**

<b>Address Offset</b>	0x0000 015C	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register contains dedicated UART4 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_RETAINED_BANK	RESERVED											LOSTCONTEXT_RFF	RESERVED		

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:2	RESERVED		R	0x00
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RET_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
0	RESERVED		R	0

**Table 3-689. Register Call Summary for Register RM\_L4PER\_UART4\_CONTEXT**

Power Management Functional Description

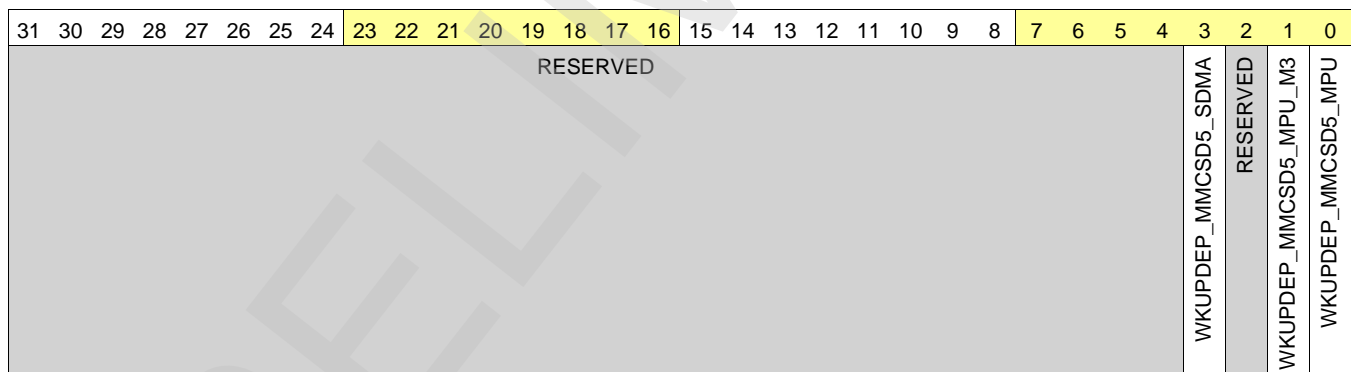
- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

**Table 3-690. PM\_L4PER\_MMCS5\_WKDEP**

<b>Address Offset</b>	0x0000 0160	<b>Instance</b>	L4PER_PRM
<b>Physical Address</b>	See <a href="#">Table 3-571</a>		
<b>Description</b>	This register controls wakeup dependency based on MMCS5 service requests.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	WKUPDEP_MMCS5_SDMA	Wakeup dependency from MMCS5 module (softwareakeup signal) towards SDMA + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	RESERVED		R	0
1	WKUPDEP_MMCS5_MPU_M3	Wakeup dependency from MMCS5 module (softwareakeup signal) towards MPU_A3 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_MMCS5 MPU	Wakeup dependency from MMCS5 module (softwareakeup signal) towards MPU + L3_1 + L3_2 domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-691. Register Call Summary for Register PM\_L4PER\_MMCS5\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[3\]](#)

**Table 3-692. RM\_L4PER\_MMCS5\_CONTEXT**

<b>Address Offset</b>	0x0000 0164		
<b>Physical Address</b>	See <a href="#">Table 3-571</a>	<b>Instance</b>	L4PER_PRM
<b>Description</b>	This register contains dedicated MMCS5 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_NONRETAINED_BANK	RESERVED							LOSTCONTEXT_DFF							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4_PER_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-693. Register Call Summary for Register RM\_L4PER\_MMCS5\_CONTEXT**

Power Management Functional Description

- [PD\\_L4\\_PER Description: \[0\]](#)

PRCM Register Manual

- [L4PER\\_PRM Register Summary: \[1\]](#)

### 3.11.15 WKUP\_PRM Registers

#### 3.11.15.1 WKUP\_PRM Register Summary

Table 3-694. WKUP\_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	WKUP_PRM L4 Base Address
RM_WKUP_L4WKUP_CONTEXT	RW	32	0x0000 0024	0x4A30 7724
RESERVED	RW	32	0x0000 002C	0x4A30 772C
PM_WKUP_WDTIMER2_WKDEP	RW	32	0x0000 0030	0x4A30 7730
RM_WKUP_WDTIMER2_CONTEXT	RW	32	0x0000 0034	0x4A30 7734
PM_WKUP_GPIO1_WKDEP	RW	32	0x0000 0038	0x4A30 7738
RM_WKUP_GPIO1_CONTEXT	RW	32	0x0000 003C	0x4A30 773C
PM_WKUP_GPTIMER1_WKDEP	R	32	0x0000 0040	0x4A30 7740
RM_WKUP_GPTIMER1_CONTEXT	RW	32	0x0000 0044	0x4A30 7744
RESERVED	R	32	0x0000 0048	0x4A30 7748
RESERVED	RW	32	0x0000 004C	0x4A30 774C
RM_WKUP_32KTIMER_CONTEXT	RW	32	0x0000 0054	0x4A30 7754
RESERVED	RW	32	0x0000 0058	0x4A30 7758
RESERVED	RW	32	0x0000 005C	0x4A30 775C
RM_WKUP_SARRAM_CONTEXT	RW	32	0x0000 0064	0x4A30 7764
PM_WKUP_KEYBOARD_WKDEP	R	32	0x0000 0078	0x4A30 7778
RM_WKUP_KEYBOARD_CONTEXT	RW	32	0x0000 007C	0x4A30 777C
RESERVED	R	32	0x0000 0080	0x4A30 7780
RESERVED	RW	32	0x0000 0084	0x4A30 7784

#### 3.11.15.2 WKUP\_PRM Register Description

Table 3-695. RM\_WKUP\_L4WKUP\_CONTEXT

<b>Address Offset</b>	0x0000 0024
<b>Physical Address</b>	See Table 3-694
<b>Instance</b>	WKUP_PRM
<b>Description</b>	This register contains dedicated L4WKUP context statuses. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-696. Register Call Summary for Register RM\_WKUP\_L4WKUP\_CONTEXT**

Power Management Functional Description

- [PD\\_WKUP Description: \[0\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[1\]](#)

**Table 3-697. PM\_WKUP\_WDTIMER2\_WKDEP**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	WKUP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-694</a>		
<b>Description</b>	This register controls wakeup dependency based on WDT2 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_WDT2_MPU_M3		WKUPDEP_WDT2_MPU													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	WKUPDEP_WDT2_MPU_M3	Wakeup dependency from WDT2 module (softwareakeup signal) towards MPU_A3 + L3_2 + L3_1 + L4_CFG domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_WDT2_MPU	Wakeup dependency from WDT2 module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-698. Register Call Summary for Register PM\_WKUP\_WDTIMER2\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[2\]](#)

**Table 3-699. RM\_WKUP\_WDTIMER2\_CONTEXT**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 3-694</a>	<b>Instance</b>	WKUP_PRM
<b>Description</b>	This register contains dedicated WDT2 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-700. Register Call Summary for Register RM\_WKUP\_WDTIMER2\_CONTEXT**

Power Management Functional Description

- [PD\\_WKUP Description: \[0\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[1\]](#)

**Table 3-701. PM\_WKUP\_GPIO1\_WKDEP**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-694</a>	<b>Instance</b>	WKUP_PRM
<b>Description</b>	This register controls wakeup dependency based on GPIO1 service requests.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_GPIO1_IRQ2_DSP	RESERVED			WKUPDEP_GPIO1_IRQ1_MPU_M3	WKUPDEP_GPIO1_IRQ1_MPU										

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	WKUPDEP_GPIO1_IRQ2_DSP	Wakeup dependency from GPIO1 module (POINTRsoftwareAKEUP2 signal) towards DSP + L3_1 + L4_CFG domains Read 0x1: Dependency is enabled	R Returns 1s	1
5:2	RESERVED		R	0x0
1	WKUPDEP_GPIO1_IRQ1_MPU_M3	Wakeup dependency from GPIO1 module (POINTRsoftwareAKEUP1 signal) module towards MPU_A3 + L3_2 + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	WKUPDEP_GPIO1_IRQ1_MPU	Wakeup dependency from GPIO1 module (POINTRsoftwareAKEUP1 signal) towards MPU + L3_1 + L4_CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-702. Register Call Summary for Register PM\_WKUP\_GPIO1\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[3\]](#)

**Table 3-703. RM\_WKUP\_GPIO1\_CONTEXT**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	WKUP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-694</a>		
<b>Description</b>	This register contains dedicated GPIO1 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-704. Register Call Summary for Register RM\_WKUP\_GPIO1\_CONTEXT**

Power Management Functional Description

- [PD\\_WKUP Description: \[0\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[1\]](#)



**Table 3-705. PM\_WKUP\_GPTIMER1\_WKDEP**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 3-694</a>	<b>Instance</b>	WKUP_PRM
<b>Description</b>	This register controls wakeup dependency based on TIMER1 service requests.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WKUPDEP_TIMER1_MPU			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	WKUPDEP_TIMER1_MPU	Wakeup dependency from TIMER1 module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains Read 0x1: Dependency is enabled	R Rreturns 1s	1

**Table 3-706. Register Call Summary for Register PM\_WKUP\_GPTIMER1\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[1\]](#)

**Table 3-707. RM\_WKUP\_GPTIMER1\_CONTEXT**

<b>Address Offset</b>	0x0000 0044		
<b>Physical Address</b>	See <a href="#">Table 3-694</a>	<b>Instance</b>	WKUP_PRM
<b>Description</b>	This register contains dedicated TIMER1 context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LOSTCONTEXT_DFF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-708. Register Call Summary for Register RM\_WKUP\_GPTIMER1\_CONTEXT**

Power Management Functional Description

- [PD\\_WKUP Description: \[0\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[1\]](#)

**Table 3-709. RM\_WKUP\_32KTIMER\_CONTEXT**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	WKUP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-694</a>		
<b>Description</b>	This register contains dedicated SYNCTIMER context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_DFF															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_SYS_PWRON_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-710. Register Call Summary for Register RM\_WKUP\_32KTIMER\_CONTEXT**

Power Management Functional Description

- [PD\\_WKUP Description: \[0\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[1\]](#)

**Table 3-711. RM\_WKUP\_SARRAM\_CONTEXT**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	WKUP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-694</a>		
<b>Description</b>	This register contains dedicated SARRAM context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_WKUP_BANK	RESERVED										LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_WKUP_BANK	Specify if memory-based context in WKUP_BANK memory bank has been lost due to a previous global cold reset.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-712. Register Call Summary for Register RM\_WKUP\_SARRAM\_CONTEXT**

Power Management Functional Description

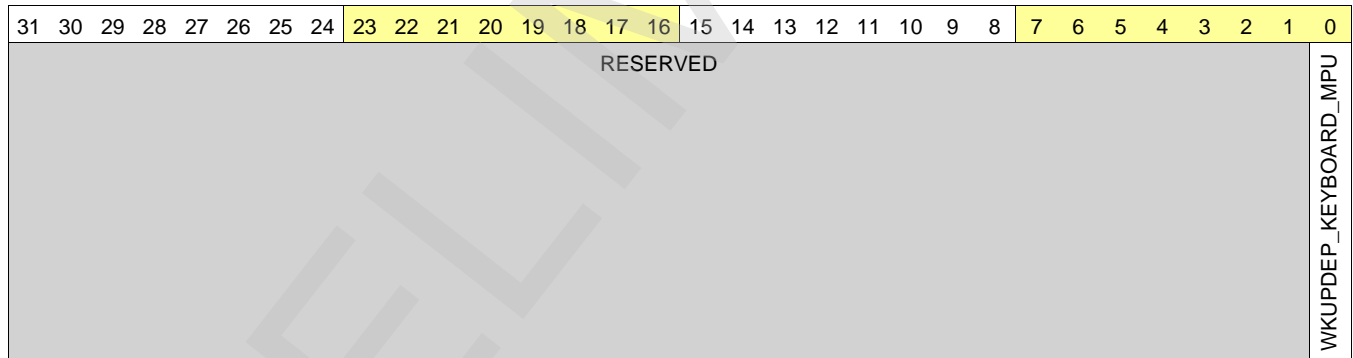
- [PD\\_WKUP Description: \[0\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[1\]](#)

**Table 3-713. PM\_WKUP\_KEYBOARD\_WKDEP**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	WKUP_PRM
<b>Physical Address</b>	See <a href="#">Table 3-694</a>		
<b>Description</b>	This register controls wakeup dependency based on KEYBOARD service requests.		
<b>Type</b>	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	WKUPDEP_KEYBOARD_MPU	Wakeup dependency from KEYBOARD module (softwareakeup signal) towards MPU + L3_1 + L4_CFG domains  Read 0x1: Dependency is enabled	R Returns 1s	1

**Table 3-714. Register Call Summary for Register PM\_WKUP\_KEYBOARD\_WKDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[1\]](#)

**Table 3-715. RM\_WKUP\_KEYBOARD\_CONTEXT**

<b>Address Offset</b>	0x0000 007C		
<b>Physical Address</b>	See <a href="#">Table 3-694</a>	<b>Instance</b>	WKUP_PRM
<b>Description</b>	This register contains dedicated KEYBOARD context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTCONTEXT_DFF								

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUP_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	1

**Table 3-716. Register Call Summary for Register RM\_WKUP\_KEYBOARD\_CONTEXT**

Power Management Functional Description

- [PD\\_WKUP Description: \[0\]](#)

PRCM Register Manual

- [WKUP\\_PRM Register Summary: \[1\]](#)

### 3.11.16 WKUP\_CM Registers

#### 3.11.16.1 WKUP\_CM Register Summary

**Table 3-717. WKUP\_CM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	WKUP_CM L4 Base Address
<a href="#">CM_WKUP_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A30 7800
<a href="#">CM_WKUP_L4WKUP_CLKCTRL</a>	R	32	0x0000 0020	0x4A30 7820
RESERVED	R	32	0x0000 0028	0x4A30 7828
<a href="#">CM_WKUP_WDTIMER2_CLKCTRL</a>	RW	32	0x0000 0030	0x4A30 7830
<a href="#">CM_WKUP_GPIO1_CLKCTRL</a>	RW	32	0x0000 0038	0x4A30 7838
<a href="#">CM_WKUP_GPTIMER1_CLKCTRL</a>	RW	32	0x0000 0040	0x4A30 7840
RESERVED	R	32	0x0000 0048	0x4A30 7848
<a href="#">CM_WKUP_32KTIMER_CLKCTRL</a>	R	32	0x0000 0050	0x4A30 7850
RESERVED	RW	32	0x0000 0058	0x4A30 7858
<a href="#">CM_WKUP_SARRAM_CLKCTRL</a>	R	32	0x0000 0060	0x4A30 7860
<a href="#">CM_WKUP_KEYBOARD_CLKCTRL</a>	RW	32	0x0000 0078	0x4A30 7878
RESERVED	RW	32	0x0000 0080	0x4A30 7880
<a href="#">CM_WKUP_BANDGAP_CLKCTRL</a>	RW	32	0x0000 0088	0x4A30 7888

3.11.16.2 WKUP\_CM Register Description

Table 3-718. CM\_WKUP\_CLKSTCTRL

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	WKUP_CM
<b>Physical Address</b>	See Table 3-717		
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_L4_WKUP_ICLK	CLKACTIVITY_WKUP_32K_FCLK	RESERVED	CLKACTIVITY_ABE_LP_CLK	CLKACTIVITY_SYS_CLK	RESERVED						CLKTRCTRL				

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	CLKACTIVITY_L4_WKUP_ICLK	This field indicates the state of the clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
11	CLKACTIVITY_WKUP_32K_FCLK	This field indicates the state of the clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
10	RESERVED		R	0
9	CLKACTIVITY_ABE_LP_CLK	This field indicates the state of the clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_SYS_CLK	This field indicates the state of the SYS_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the WKUP clock domain.  0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.  Read 0x1: Reserved  Read 0x2: Reserved  0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-719. Register Call Summary for Register CM\_WKUP\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [WKUP\\_CM Register Summary: \[5\]](#)

**Table 3-720. CM\_WKUP\_L4WKUP\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	WKUP_CM
<b>Physical Address</b>	See <a href="#">Table 3-717</a>		
<b>Description</b>	This register manages the L4WKUP clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								MODULEMODE														

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-721. Register Call Summary for Register CM\_WKUP\_L4WKUP\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- PRCM Register Manual
- [WKUP\\_CM Register Summary: \[2\]](#)

**Table 3-722. CM\_WKUP\_WDTIMER2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	WKUP_CM
<b>Physical Address</b>	See <a href="#">Table 3-717</a>		
<b>Description</b>	This register manages the WDT2 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED										MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-723. Register Call Summary for Register CM\_WKUP\_WDTIMER2\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- PRCM Register Manual
- [WKUP\\_CM Register Summary: \[2\]](#)

**Table 3-724. CM\_WKUP\_GPIO1\_CLKCTRL**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-717</a>	<b>Instance</b>	WKUP_CM
<b>Description</b>	This register manages the GPIO1 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																IDLEST	RESERVED								OPTFCLKEN_DBCLK	RESERVED							MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-725. Register Call Summary for Register CM\_WKUP\_GPIO1\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [WKUP\\_CM Register Summary: \[2\]](#)



**Table 3-726. CM\_WKUP\_GPTIMER1\_CLKCTRL**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 3-717</a>	<b>Instance</b>	WKUP_CM
<b>Description</b>	This register manages the TIMER1 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKSEL	RESERVED								IDLEST	RESERVED											MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Select the source of the functional clock 0x0: Selects the SYS_CLK as the source 0x1: Selects the 32KHz as the source	RW	0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-727. Register Call Summary for Register CM\_WKUP\_GPTIMER1\_CLKCTRL**

Clock Management Functional Description

- [PRM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [WKUP\\_CM Register Summary: \[3\]](#)

**Table 3-728. CM\_WKUP\_32KTIMER\_CLKCTRL**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 3-717</a>	<b>Instance</b>	WKUP_CM
<b>Description</b>	This register manages the SYNCTIMER clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Returns	0x1

**Table 3-729. Register Call Summary for Register CM\_WKUP\_32KTIMER\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [WKUP\\_CM Register Summary: \[2\]](#)

**Table 3-730. CM\_WKUP\_SARRAM\_CLKCTRL**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See <a href="#">Table 3-717</a>	<b>Instance</b>	WKUP_CM
<b>Description</b>	This register manages the SARRAM clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-731. Register Call Summary for Register CM\_WKUP\_SARRAM\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [WKUP\\_CM Register Summary: \[2\]](#)

**Table 3-732. CM\_WKUP\_KEYBOARD\_CLKCTRL**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	WKUP_CM
<b>Physical Address</b>	See <a href="#">Table 3-717</a>		
<b>Description</b>	This register manages the KEYBOARD clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST		RESERVED										MODULEMODE			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-733. Register Call Summary for Register CM\_WKUP\_KEYBOARD\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [WKUP\\_CM Register Summary: \[2\]](#)

**Table 3-734. CM\_WKUP\_BANDGAP\_CLKCTRL**

<b>Address Offset</b>	0x0000 0088	<b>Instance</b>	WKUP_CM
<b>Physical Address</b>	See <a href="#">Table 3-717</a>		
<b>Description</b>	This register manages the bandgap clock.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OPTFCLKEN_BGAP_32K	RESERVED														

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	OPTFCLKEN_BGAP_32K	Optional functional clock control.  0x0: Optional functional clock is disabled  0x1: Optional functional clock is enabled	RW	0
7:0	RESERVED		R	0x00

**Table 3-735. Register Call Summary for Register CM\_WKUP\_BANDGAP\_CLKCTRL**

PRCM Register Manual

- [WKUP\\_CM Register Summary: \[0\]](#)

### 3.11.17 EMU\_PRM Registers

#### 3.11.17.1 EMU\_PRM Register Summary

**Table 3-736. EMU\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	EMU_PRM L4 Base Address
<a href="#">PM_EMU_PWRSTCTRL</a>	R	32	0x0000 0000	0x4A30 7900
<a href="#">PM_EMU_PWRSTST</a>	R	32	0x0000 0004	0x4A30 7904
<a href="#">RM_EMU_DEBUGSS_CONTEXT</a>	RW	32	0x0000 0024	0x4A30 7924

**3.11.17.2 EMU\_PRM Register Description**

**Table 3-737. PM\_EMU\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-736</a>	<b>Instance</b>	EMU_PRM
<b>Description</b>	This register controls the EMU power state to reach upon a domain sleep transition		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EMU_BANK_ONSTATE								RESERVED								POWERSTATE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	EMU_BANK_ONSTATE	EMU memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R Returns 1s	0x3
15:2	RESERVED		R	0x0000
1:0	POWERSTATE	Power state control Read 0x0: OFF state	R	0x0

**Table 3-738. Register Call Summary for Register PM\_EMU\_PWRSTCTRL**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\]](#)

PRCM Register Manual

- [EMU\\_PRM Register Summary: \[2\]](#)

**Table 3-739. PM\_EMU\_PWRSTST**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 3-736</a>	<b>Instance</b>	EMU_PRM
<b>Description</b>	This register provides a status on the EMU domain current power state. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INTRANSITION	RESERVED								EMU_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST											

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:6	RESERVED		R	0x0000
5:4	EMU_BANK_STATEST	EMU memory bank state status Read 0x0: Memory is OFF Read 0x1: Reserved Read 0x2: Reserved Read 0x3: Memory is ON It is supplied by WKUP LDO	R	0x3
3	RESERVED		R	0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 3-740. Register Call Summary for Register PM\_EMU\_PWRSTST**

Power Management Functional Description

- [Power Domain Modes: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [EMU\\_PRM Register Summary: \[4\]](#)

**Table 3-741. RM\_EMU\_DEBUGSS\_CONTEXT**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	EMU_PRM
<b>Physical Address</b>	See <a href="#">Table 3-736</a>		
<b>Description</b>	This register contains dedicated DEBUGSS context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											LOSTMEM_EMU_BANK	RESERVED						LOSTCONTEXT_DFF													

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	LOSTMEM_EMU_BANK	Specify if memory-based context in EMU_BANK memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED		R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EMU_RST signal)  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 3-742. Register Call Summary for Register RM\_EMU\_DEBUGSS\_CONTEXT**

PRCM Register Manual

- [EMU\\_PRM Register Summary: \[0\]](#)

### 3.11.18 EMU\_CM Registers

#### 3.11.18.1 EMU\_CM Register Summary

**Table 3-743. EMU\_CM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	EMU_CM L4 Base Address
<a href="#">CM_EMU_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A30 7A00
<a href="#">CM_EMU_DYNAMICDEP</a>	RW	32	0x0000 0008	0x4A30 7A08
<a href="#">CM_EMU_DEBUGSS_CLKCTRL</a>	RW	32	0x0000 0020	0x4A30 7A20

#### 3.11.18.2 EMU\_CM Register Description

**Table 3-744. CM\_EMU\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	EMU_CM
<b>Physical Address</b>	See <a href="#">Table 3-743</a>		
<b>Description</b>	This register enables the EMU domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_CORE_DPLL_EMU_CLK		CLKACTIVITY_EMU_SYS_CLK		RESERVED										CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0000000
9	CLKACTIVITY_CORE_DPLL_EMU_CLK	This field indicates the state of the CORE_DPLL_EMU_CLK clock in the domain. Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_EMU_SYS_CLK	This field indicates the state of the EMU_SYS_CLK clock in the domain. Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the EMU clock domain. Read 0x0: Reserved Read 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x2

**Table 3-745. Register Call Summary for Register CM\_EMU\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [EMU\\_CM Register Summary: \[3\]](#)

**Table 3-746. CM\_EMU\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008
<b>Physical Address</b>	See <a href="#">Table 3-743</a>
<b>Description</b>	This register controls the dynamic domain dependencies from EMU domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED										L3_2_DYNDEP		RESERVED											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:7	RESERVED		R	0x00000
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Returns 1s	1
5:0	RESERVED		R	0x00



**Table 3-747. Register Call Summary for Register CM\_EMU\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\]](#)

PRCM Register Manual

- [EMU\\_CM Register Summary: \[1\]](#)

**Table 3-748. CM\_EMU\_DEBUGSS\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	EMU_CM
<b>Physical Address</b>	See <a href="#">Table 3-743</a>		
<b>Description</b>	This register manages the DEBUGSS clocks. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		CLKSEL_PMD_STM_CLK		CLKSEL_PMD_TRACE_CLK		PMD_TRACE_MUX_CTRL		PMD_STM_MUX_CTRL		RESERVED		STBYST		IDLEST		RESERVED											MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:27	CLKSEL_PMD_STM_CLK	Selection of STM clock division Read 0x0: Reserved 0x1: STM_CLK is the selected STM source clock divided by 1 0x2: STM_CLK is the selected STM source clock divided by 2 Read 0x3: Reserved 0x4: STM_CLK is the selected STM source clock divided by 4 Read 0x5: Reserved Read 0x6: Reserved Read 0x7: Reserved	RW	0x1
26:24	CLKSEL_PMD_TRACE_CLK	Selection of TRACE clock division Read 0x0: Reserved 0x1: TRACE_CLK is the selected TRACE source clock divided by 1 0x2: TRACE_CLK is the selected TRACE source clock divided by 2 Read 0x3: Reserved 0x4: TRACE_CLK is the selected TRACE source clock divided by 4 Read 0x5: Reserved Read 0x6: Reserved Read 0x7: Reserved	RW	0x1

Bits	Field Name	Description	Type	Reset
23:22	PMD_TRACE_MUX_CTRL	Selection of TRACE source clock 0x0: TRACE source clock is SYS_CLK 0x1: TRACE source clock is CORE_DPLL_EMU_CLK Read 0x2: Reserved Read 0x3: Reserved	RW	0x0
21:20	PMD_STM_MUX_CTRL	Selection of STM source clock 0x0: STM source clock is SYS_CLK 0x1: STM source clock is CORE_DPLL_EMU_CLK Read 0x2: Reserved Read 0x3: Reserved	RW	0x0
19	RESERVED		R	0
18	STBYST	Module standby status Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Returns	0x1

**Table 3-749. Register Call Summary for Register CM\_EMU\_DEBUGSS\_CLKCTRL**

Clock Management Functional Description

- [Overview: \[0\] \[1\]](#)

PRCM Register Manual

- [EMU\\_CM Register Summary: \[2\]](#)

### 3.11.19 DEVICE\_PRM Registers

#### 3.11.19.1 DEVICE\_PRM Register Summary

**Table 3-750. DEVICE\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DEVICE_PRM L4 Base Address
<a href="#">PRM_RSTCTRL</a>	RW	32	0x0000 0000	0x4A30 7B00
<a href="#">PRM_RSTST</a>	RW	32	0x0000 0004	0x4A30 7B04
<a href="#">PRM_RSTTIME</a>	RW	32	0x0000 0008	0x4A30 7B08
<a href="#">PRM_CLKREQCTRL</a>	RW	32	0x0000 000C	0x4A30 7B0C
<a href="#">PRM_VOLTCTRL</a>	RW	32	0x0000 0010	0x4A30 7B10
<a href="#">PRM_PWRREQCTRL</a>	RW	32	0x0000 0014	0x4A30 7B14
<a href="#">PRM_PSCON_COUNT</a>	RW	32	0x0000 0018	0x4A30 7B18
<a href="#">PRM_IO_COUNT</a>	RW	32	0x0000 001C	0x4A30 7B1C

**Table 3-750. DEVICE\_PRM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DEVICE_PRM L4 Base Address
PRM_IO_PMCTRL	RW	32	0x0000 0020	0x4A30 7B20
PRM_VOLTSETUP_WARMRESET	RW	32	0x0000 0024	0x4A30 7B24
PRM_VOLTSETUP_CORE_OFF	RW	32	0x0000 0028	0x4A30 7B28
PRM_VOLTSETUP_MPU_OFF	RW	32	0x0000 002C	0x4A30 7B2C
PRM_VOLTSETUP_IVA_OFF	RW	32	0x0000 0030	0x4A30 7B30
PRM_VOLTSETUP_CORE_RET_SLEEP	RW	32	0x0000 0034	0x4A30 7B34
PRM_VOLTSETUP_MPU_RET_SLEEP	RW	32	0x0000 0038	0x4A30 7B38
PRM_VOLTSETUP_IVA_RET_SLEEP	RW	32	0x0000 003C	0x4A30 7B3C
PRM_VP_CORE_CONFIG	RW	32	0x0000 0040	0x4A30 7B40
PRM_VP_CORE_STATUS	R	32	0x0000 0044	0x4A30 7B44
PRM_VP_CORE_VLIMITTO	RW	32	0x0000 0048	0x4A30 7B48
PRM_VP_CORE_VOLTAGE	R	32	0x0000 004C	0x4A30 7B4C
PRM_VP_CORE_VSTEPMAX	RW	32	0x0000 0050	0x4A30 7B50
PRM_VP_CORE_VSTEPMIN	RW	32	0x0000 0054	0x4A30 7B54
PRM_VP_MPU_CONFIG	RW	32	0x0000 0058	0x4A30 7B58
PRM_VP_MPU_STATUS	R	32	0x0000 005C	0x4A30 7B5C
PRM_VP_MPU_VLIMITTO	RW	32	0x0000 0060	0x4A30 7B60
PRM_VP_MPU_VOLTAGE	R	32	0x0000 0064	0x4A30 7B64
PRM_VP_MPU_VSTEPMAX	RW	32	0x0000 0068	0x4A30 7B68
PRM_VP_MPU_VSTEPMIN	RW	32	0x0000 006C	0x4A30 7B6C
PRM_VP_IVA_CONFIG	RW	32	0x0000 0070	0x4A30 7B70
PRM_VP_IVA_STATUS	R	32	0x0000 0074	0x4A30 7B74
PRM_VP_IVA_VLIMITTO	RW	32	0x0000 0078	0x4A30 7B78
PRM_VP_IVA_VOLTAGE	R	32	0x0000 007C	0x4A30 7B7C
PRM_VP_IVA_VSTEPMAX	RW	32	0x0000 0080	0x4A30 7B80
PRM_VP_IVA_VSTEPMIN	RW	32	0x0000 0084	0x4A30 7B84
PRM_VC_SMPS_SA	RW	32	0x0000 0088	0x4A30 7B88
PRM_VC_VAL_SMPS_RA_VOL	RW	32	0x0000 008C	0x4A30 7B8C
PRM_VC_VAL_SMPS_RA_CMD	RW	32	0x0000 0090	0x4A30 7B90
PRM_VC_VAL_CMD_VDD_CORE_L	RW	32	0x0000 0094	0x4A30 7B94
PRM_VC_VAL_CMD_VDD_MPU_L	RW	32	0x0000 0098	0x4A30 7B98
PRM_VC_VAL_CMD_VDD_IVA_L	RW	32	0x0000 009C	0x4A30 7B9C
PRM_VC_VAL_BYPASS	RW	32	0x0000 00A0	0x4A30 7BA0
PRM_VC_CFG_CHANNEL	RW	32	0x0000 00A4	0x4A30 7BA4
PRM_VC_CFG_I2C_MODE	RW	32	0x0000 00A8	0x4A30 7BA8
PRM_VC_CFG_I2C_CLK	RW	32	0x0000 00AC	0x4A30 7BAC
PRM_SRAM_COUNT	RW	32	0x0000 00B0	0x4A30 7BB0
PRM_SRAM_WKUP_SETUP	RW	32	0x0000 00B4	0x4A30 7BB4
PRM_LDO_SRAM_CORE_SETUP	RW	32	0x0000 00B8	0x4A30 7BB8
PRM_LDO_SRAM_CORE_CTRL	RW	32	0x0000 00BC	0x4A30 7BBC
PRM_LDO_SRAM_MPU_SETUP	RW	32	0x0000 00C0	0x4A30 7BC0
PRM_LDO_SRAM_MPU_CTRL	RW	32	0x0000 00C4	0x4A30 7BC4
PRM_LDO_SRAM_IVA_SETUP	RW	32	0x0000 00C8	0x4A30 7BC8
PRM_LDO_SRAM_IVA_CTRL	RW	32	0x0000 00CC	0x4A30 7BCC
PRM_LDO_ABB_MPU_SETUP	RW	32	0x0000 00D0	0x4A30 7BD0
PRM_LDO_ABB_MPU_CTRL	RW	32	0x0000 00D4	0x4A30 7BD4
PRM_LDO_ABB_IVA_SETUP	RW	32	0x0000 00D8	0x4A30 7BD8

**Table 3-750. DEVICE\_PRM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DEVICE_PRM L4 Base Address
<a href="#">PRM_LDO_ABB_IVA_CTRL</a>	RW	32	0x0000 00DC	0x4A30 7BDC
<a href="#">PRM_LDO_BANDGAP_SETUP</a>	RW	32	0x0000 00E0	0x4A30 7BE0
<a href="#">PRM_DEVICE_OFF_CTRL</a>	RW	32	0x0000 00E4	0x4A30 7BE4
<a href="#">PRM_PHASE1_CNDP</a>	R	32	0x0000 00E8	0x4A30 7BE8
<a href="#">PRM_PHASE2A_CNDP</a>	RW	32	0x0000 00EC	0x4A30 7BEC
<a href="#">PRM_PHASE2B_CNDP</a>	RW	32	0x0000 00F0	0x4A30 7BF0
RESERVED	RW	32	0x0000 00F4	0x4A30 7BF4
<a href="#">PRM_VC_ERRST</a>	RW	32	0x0000 00F8	0x4A30 7BF8

**3.11.19.2 DEVICE\_PRM Register Description****Table 3-751. PRM\_RSTCTRL**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b> DEVICE_PRM
<b>Description</b>	Global software cold and warm reset control. This register is auto-cleared. Only write 1 is possible. A read returns 0 only.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																																
																RST_GLOBAL_COLD_software																
																RST_GLOBAL_WARM_software																

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	RST_GLOBAL_COLD_software	Global COLD software reset control. This bit is reset only upon a global cold source of reset. 0x0: Global COLD software reset is cleared. 0x1: Triggers a global COLD software reset. The software must ensure the SDRAM is properly put in self-refresh mode before applying this reset.	RW	0
0	RST_GLOBAL_WARM_software	Global WARM software reset control. This bit is reset upon any global source of reset (warm and cold). 0x0: Global warm software reset is cleared. 0x1: Triggers a global warm software reset.	RW	0

**Table 3-752. Register Call Summary for Register PRM\_RSTCTRL**

Reset Management Functional Description

- [Global Reset Sources: \[0\] \[1\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[2\]](#)

**Table 3-753. PRM\_RSTST**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register logs the global reset sources. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ICEPICK_RST	VDD_CORE_VOLT_MGR_RST	VDD_IVA_VOLT_MGR_RST	VDD_MPU_VOLT_MGR_RST	EXTERNAL_WARM_RST	RESERVED	MPU_WDT_RST	RESERVED	GLOBAL_WARM_SW_RST	GLOBAL_COLD_RST						

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	ICEPICK_RST	IcePick reset event. This is a source of global warm reset initiated by the emulation. 0x0: No ICEPICK reset. 0x1: IcePick reset has occurred.	RW W1toClr	0
8	VDD_CORE_VOLT_MGR_RST	VDD_CORE voltage manager reset event This is a source of global WARM reset. 0x0: No VDD_CORE voltage manager reset. 0x1: VDD_CORE voltage manager reset has occurred.	RW W1toClr	0
7	VDD_IVA_VOLT_MGR_RST	VDD_IVA voltage manager reset event This is a source of global WARM reset. 0x0: No VDD_IVA voltage manager reset. 0x1: VDD_IVA voltage manager reset has occurred.	RW W1toClr	0
6	VDD_MPU_VOLT_MGR_RST	VDD_MPU voltage manager reset event This is a source of global WARM reset. 0x0: No VDD_MPU voltage manager reset. 0x1: VDD_MPU voltage manager reset has occurred.	RW W1toClr	0
5	EXTERNAL_WARM_RST	External warm reset event 0x0: No global warm reset. 0x1: Global external warm reset has occurred.	RW W1toClr	0
4	RESERVED		RW W1toClr	0
3	MPU_WDT_RST	MPU Watchdog timer reset event. This is a source of global WARM reset. 0x0: No MPU watchdog reset. 0x1: MPU watchdog reset has occurred.	RW W1toClr	0
2	RESERVED		RW W1toClr	0
1	GLOBAL_WARM_SW_RST	Global warm software reset event 0x0: No global warm software reset 0x1: Global warm software reset has occurred.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
0	GLOBAL_COLD_RST	Power-on (cold) reset event 0x0: No power-on reset. 0x1: Power-on reset has occurred.	RW W1toClr	1

**Table 3-754. Register Call Summary for Register PRM\_RSTST**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-755. PRM\_RSTTIME**

<b>Address Offset</b>	0x0000 0008	
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b> DEVICE_PRM
<b>Description</b>	Reset duration control. [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														RSTTIME2				RSTTIME1													

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:10	RSTTIME2	(Power domain) reset duration 2 (number of RM.SYSCLK clock cycles) 0x0: Reserved	RW	0x10
9:0	RSTTIME1	(Global) reset duration 1 (number of Func_32k.clk clock cycles) 0x0: Reserved	RW	0x06

**Table 3-756. Register Call Summary for Register PRM\_RSTTIME**

Reset Management Functional Description

- [Reset Domains: \[0\]](#)
- [DSP Subsystem Power-On Reset Sequence: \[1\] \[2\]](#)
- [DSP Subsystem Software Warm Reset Sequence: \[3\]](#)
- [DSP Subsystem Reset Sequence On Wake-Up Transitions From OSWR RETENTION State: \[4\]](#)
- [Cortex-M3 MPU Subsystem Power-On Reset Sequence: \[5\] \[6\]](#)
- [Cortex-M3 MPU Subsystem Software Warm Reset Sequence: \[7\]](#)
- [Global Warm Reset Sequence: \[8\]](#)

Device Low-Power States

- [Device Off Mode Wake-Up Sequences: \[9\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[10\]](#)

**Table 3-757. PRM\_CLKREQCTRL**

<b>Address Offset</b>	0x0000 000C	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	This register allows controlling the CLKREQ signal towards SCRM.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKREQ_COND								

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	CLKREQ_COND	Control upon which condition CLKREQ signal is de-asserted. 0x0: CLKREQ is never de-asserted 0x1: CLKREQ is de-asserted when system clock is not required by any function in the device and if all voltage domains are in OFF state. 0x2: CLKREQ is de-asserted when system clock is not required by any function in the device and if all voltage domains are in RET or OFF state. 0x3: CLKREQ is de-asserted when system clock is not required by any function in the device and if all voltage domains are in SLEEP or RET or OFF state. 0x4: CLKREQ is de-asserted when system clock is not required by any function in the device. This is designed for low-power use-cases using the DPLL cascading scheme (ex: MP3) Read 0x5: Reserved Read 0x6: Reserved Read 0x7: Reserved	RW	0x0

**Table 3-758. Register Call Summary for Register PRM\_CLKREQCTRL**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-759. PRM\_VOLTCTRL**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	This register provides voltage domain management controls.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VDD_IVA_I2C_DISABLE	VDD_MPU_I2C_DISABLE	VDD_CORE_I2C_DISABLE	RESERVED			VDD_IVA_PRESENCE	VDD_MPU_PRESENCE	RESERVED			AUTO_CTRL_VDD_IVA_L	AUTO_CTRL_VDD_MPU_L	AUTO_CTRL_VDD_CORE_L		

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x000000
14	VDD_IVA_I2C_DISABLE	This bit allows disabling I2C interface with powerIC for IVA voltage (for debug purpose only). [warm reset insensitive]  0x0: Normal mode: I2C is enabled. 0x1: Debug mode: I2C is disabled.	RW	0
13	VDD_MPU_I2C_DISABLE	This bit allows disabling I2C interface with powerIC for MPU voltage (for debug purpose only). [warm reset insensitive]  0x0: Normal mode: I2C is enabled. 0x1: Debug mode: I2C is disabled.	RW	0
12	VDD_CORE_I2C_DISABLE	This bit allows disabling I2C interface with powerIC for CORE voltage (for debug purpose only). [warm reset insensitive]  0x0: Normal mode: I2C is enabled. 0x1: Debug mode: I2C is disabled.	RW	0
11:10	RESERVED		R	0x0
9	VDD_IVA_PRESENCE	This bit control the presence of IVA voltage in device. [warm reset insensitive]  0x0: IVA voltage is not present as an individual voltage: IVA voltage is merged with MPU voltage if VDD_MPU_presence=1. IVA voltage is merged with CORE voltage if VDD_MPU_presence=0. 0x1: IVA voltage is present on the device.	RW	1
8	VDD_MPU_PRESENCE	This bit control the presence of MPU voltage in device. [warm reset insensitive]  0x0: MPU voltage is not present as an individual voltage: MPU voltage is merged with IVA voltage if VDD_IVA_presence=1. MPU voltage is merged with CORE voltage if VDD_IVA_presence=0. 0x1: MPU voltage is present on the device.	RW	1
7:6	RESERVED		R	0x0
5:4	AUTO_CTRL_VDD_IVA_L	This bit field specifies the state to which the hardware can automatically transition the VDD_IVA_L voltage domain.  0x0: Voltage domain transitions are disabled. 0x1: Voltage domain transitions to SLEEP are enabled. 0x2: Voltage domain transitions to RET are enabled. Read 0x3: reserved	RW	0x0



Bits	Field Name	Description	Type	Reset
3:2	AUTO_CTRL_VDD_MPU_L	This bit field specifies the state to which the hardware can automatically transition the VDD_MPU_L voltage domain. 0x0: Voltage domain transitions are disabled. 0x1: Voltage domain transitions to SLEEP are enabled. 0x2: Voltage domain transitions to RET are enabled. Read 0x3: reserved	RW	0x0
1:0	AUTO_CTRL_VDD_CORE_L	This bit field specifies the state to which the hardware can automatically transition the VDD_CORE_L voltage domain. 0x0: Voltage domain transitions are disabled. 0x1: Voltage domain transitions to SLEEP are enabled. 0x2: Voltage domain transitions to RET are enabled. Read 0x3: reserved	RW	0x0

**Table 3-760. Register Call Summary for Register PRM\_VOLTCTRL**

Voltage Management Functional Description

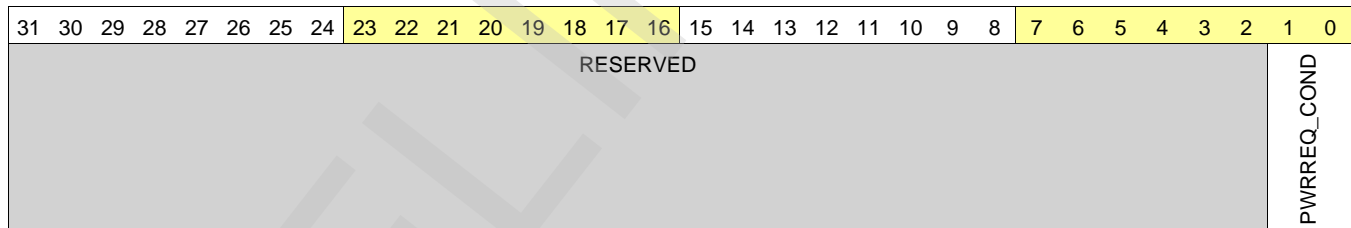
- [VDD\\_x\\_L Transitions: \[0\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[1\]](#)

**Table 3-761. PRM\_PWRREQCTRL**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	This register allows controlling the <a href="#">PWRREQ</a> signal towards power IC.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	PWRREQ_COND	Control upon which condition from MPU, IVA and CORE voltage domains <a href="#">PWRREQ</a> is de-asserted. 0x0: <a href="#">PWRREQ</a> is never de-asserted 0x1: <a href="#">PWRREQ</a> is de-asserted if all voltage domain are in SLEEP, RET or OFF state. Conversely, <a href="#">PWRREQ</a> is asserted upon any voltage domain entering or staying in ON state. 0x2: <a href="#">PWRREQ</a> is de-asserted if all voltage domain are in RET or OFF state. Conversely, <a href="#">PWRREQ</a> is asserted upon any voltage domain entering or staying in ON or SLEEP state. 0x3: <a href="#">PWRREQ</a> is de-asserted if all voltage domain are in OFF state. Conversely, <a href="#">PWRREQ</a> is asserted upon any voltage domain entering or staying in ON or SLEEP or RET state.	RW	0x0

**Table 3-762. Register Call Summary for Register PRM\_PWRREQCTRL**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-763. PRM\_PSCON\_COUNT**

<b>Address Offset</b>	0x0000 0018	
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b> DEVICE_PRM
<b>Description</b>	This register allows controlling 2 parameters for power state controller. [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PONOUT_2_PGOODIN_TIME								PCHARGE_TIME							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	PONOUT_2_PGOODIN_TIME	The value "NbCycles" set in this field determines the duration of the PONOUT to PGOODIN transition for power domain without DPS. The duration is computed as 8 x NbCycles of system clock cycles.	RW	0x9A
7:0	PCHARGE_TIME	Number of system clock cycles for the SRAM precharge duration.	RW	0x27

**Table 3-764. Register Call Summary for Register PRM\_PSCON\_COUNT**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-765. PRM\_IO\_COUNT**

<b>Address Offset</b>	0x0000 001C	
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b> DEVICE_PRM
<b>Description</b>	This register allows controlling LPDDR2 I/O isolation removal setup. [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ISO_2_ON_TIME															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	ISO_2_ON_TIME	Determines the setup time of the LPDDR2 IOs going out of isolation. Counting on the system clock. Target is 1.5us.	RW	0x3A

**Table 3-766. Register Call Summary for Register PRM\_IO\_COUNT**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-767. PRM\_IO\_PMCTRL**

<b>Address Offset</b>	0x0000 0020
<b>Physical Address</b>	See <a href="#">Table 3-750</a>
<b>Description</b>	This register allows overriding ISOCLK signal towards I/O pad ring.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								GLOBAL_WUEN	RESERVED								WUCLK_STATUS		WUCLK_CTRL		RESERVED								ISOCLK_STATUS		ISOCLK_OVERRIDE

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16	GLOBAL_WUEN	Global I/O wakeup enable. This is a gating condition to all individual I/O WUEN coming from control module. Gating is done in the Spinner logic.  0x0: All individual I/O WUEN are gated in the Spinner logic (overridden to 0).  0x1: All individual I/O WUEN from control module are going to IOs.	RW	0
15:10	RESERVED		R	0x00
9	WUCLK_STATUS	Gives value of WUCLKOUT signal coming back from I/O pad ring.	R	0
8	WUCLK_CTRL	Direct control on WUCLKIN signal to I/O pad ring.  0x0: WUCLKIN signal is driven to 0. I/O wakeup daisy chain is functional as well as I/O whose wakeup feature is enabled.  0x1: WUCLKIN signal is driven to 1. I/O wakeup daisy chain is reset and is latching current pad states and WUEN inputs.	RW	0
7:6	RESERVED		R	0x0
5	IOON_STATUS	Gives the functional status of the I/O ring.  Read 0x0: Part or all of the IOs are not in the ON state, that is are in isolation state  Read 0x1: All IOs are in the ON state	R	1
4	ISOOVR_EXTEND	Control non-EMIF I/O isolation extension upon a device wakeup from OFF mode.  0x0: Non-EMIF I/O isolation is not extended. "EMIF_ON" I/O transition happens as soon as automatic restore is completed  0x1: Non-EMIF I/O isolation is extended. "EMIF_ON" I/O transition is stalled	RW	0
3:2	RESERVED		R	0x0
1	ISOCLK_STATUS	Gives value of ISOCLKOUT signal coming back from I/O pad ring.	R	0
0	ISOCLK_OVERRIDE	Override control on ISOCLKIN signal to I/O pad ring. Override should be used at boot time only when it is needed to change the mode of an I/O from 1.8-V default mode to 1.2-V mode. When not overridden, this signal is controlled by hardware only.  0x0: ISOCLKIN signal is not overridden.  0x1: ISOCLKIN signal is overridden to active value (1).	RW	0

**Table 3-768. Register Call Summary for Register PRM\_IO\_PMCTRL**

## Device Low-Power States

- [Device Off Mode Wake-Up Sequences: \[0\]](#)
- [I/O Management: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Software-Controlled I/O Isolation: \[7\] \[8\] \[9\]](#)

## PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[10\]](#)

**Table 3-769. PRM\_VOLTSETUP\_WARMRESET**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	This register provides bit fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_CORE_L domain transitions with OFF state. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STABLE_PRESCAL		RESERVED		STABLE_COUNT											

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9:8	STABLE_PRESCAL	Determines prescaler for stabilization duration counting. 0x0: Ramp-up counter is incremented every 32 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 2048 system clock cycles 0x3: Ramp-up counter is incremented every 16384 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	STABLE_COUNT	Determines the stabilization duration of all VDD_xxx_L regulators upon a global warm reset assertion. The duration is computed according to Stable_Prescal.	RW	0x00

**Table 3-770. Register Call Summary for Register PRM\_VOLTSETUP\_WARMRESET**

## Device Low-Power States

- [Device Off Mode Wake-Up Sequences: \[0\]](#)

## PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[1\]](#)

**Table 3-771. PRM\_VOLTSETUP\_CORE\_OFF**

<b>Address Offset</b>	0x0000 0028
<b>Physical Address</b>	See <a href="#">Table 3-750</a>
<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register provides bit fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_CORE_L domain transitions with OFF state. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT						RESERVED						RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 64 system clock cycles 0x1: Ramp-down counter is incremented every 256 system clock cycles 0x2: Ramp-down counter is incremented every 512 system clock cycles 0x3: Ramp-down counter is incremented every 2048 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x00
15:10	RESERVED		R	0x00
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 64 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 512 system clock cycles 0x3: Ramp-up counter is incremented every 2048 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Up_Prescal. At cold reset, PRCM assumes that VDD_CORE_L will be at a valid ON voltage before SYS_NRESPWRON is de-asserted.	RW	0x00

**Table 3-772. Register Call Summary for Register PRM\_VOLTSETUP\_CORE\_OFF**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-773. PRM\_VOLTSETUP\_MPU\_OFF**

<b>Address Offset</b>	0x0000 002C
<b>Physical Address</b>	See <a href="#">Table 3-750</a>
<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register provides bit fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_MPU_L domain transitions to or from OFF state. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT						RESERVED						RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 64 system clock cycles 0x1: Ramp-down counter is incremented every 256 system clock cycles 0x2: Ramp-down counter is incremented every 512 system clock cycles 0x3: Ramp-down counter is incremented every 2048 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x00
15:10	RESERVED		R	0x00
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 64 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 512 system clock cycles 0x3: Ramp-up counter is incremented every 2048 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Up_Prescal. At cold reset, PRCM assumes that VDD_CORE_L will be at a valid ON voltage before SYS_NRESPWRON is de-asserted.	RW	0x00

**Table 3-774. Register Call Summary for Register PRM\_VOLTSETUP\_MPU\_OFF**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-775. PRM\_VOLTSETUP\_IVA\_OFF**

<b>Address Offset</b>	0x0000 0030
<b>Physical Address</b>	See <a href="#">Table 3-750</a>
<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register provides bit fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_IVA_L domain transitions to or from OFF state. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT								RESERVED	RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT										

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 64 system clock cycles 0x1: Ramp-down counter is incremented every 256 system clock cycles 0x2: Ramp-down counter is incremented every 512 system clock cycles 0x3: Ramp-down counter is incremented every 2048 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x00
15:10	RESERVED		R	0x00
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 64 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 512 system clock cycles 0x3: Ramp-up counter is incremented every 2048 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Up_Prescal. At cold reset, PRCM assumes that VDD_CORE_L will be at a valid ON voltage before SYS_NRESPWRON is de-asserted.	RW	0x00

**Table 3-776. Register Call Summary for Register PRM\_VOLTSETUP\_IVA\_OFF**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-777. PRM\_VOLTSETUP\_CORE\_RET\_SLEEP**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register provides bit fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_CORE_L domain transitions between ON and RET or SLEEP state. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT						RESERVED						RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 16 system clock cycles 0x1: Ramp-down counter is incremented every 64 system clock cycles 0x2: Ramp-down counter is incremented every 128 system clock cycles 0x3: Ramp-down counter is incremented every 512 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x00
15:10	RESERVED		R	0x00
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 16 system clock cycles 0x1: Ramp-up counter is incremented every 64 system clock cycles 0x2: Ramp-up counter is incremented every 128 system clock cycles 0x3: Ramp-up counter is incremented every 512 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Up_Prescal.	RW	0x00

**Table 3-778. Register Call Summary for Register PRM\_VOLTSETUP\_CORE\_RET\_SLEEP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)



**Table 3-779. PRM\_VOLTSETUP\_MPU\_RET\_SLEEP**

<b>Address Offset</b>	0x0000 0038
<b>Physical Address</b>	See <a href="#">Table 3-750</a>
<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register provides bit fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_MPU_L domain transitions between ON and RET or SLEEP state. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT						RESERVED						RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 16 system clock cycles 0x1: Ramp-down counter is incremented every 64 system clock cycles 0x2: Ramp-down counter is incremented every 128 system clock cycles 0x3: Ramp-down counter is incremented every 512 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x00
15:10	RESERVED		R	0x00
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 16 system clock cycles 0x1: Ramp-up counter is incremented every 64 system clock cycles 0x2: Ramp-up counter is incremented every 128 system clock cycles 0x3: Ramp-up counter is incremented every 512 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Up_Prescal.	RW	0x00

**Table 3-780. Register Call Summary for Register PRM\_VOLTSETUP\_MPU\_RET\_SLEEP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-781. PRM\_VOLTSETUP\_IVA\_RET\_SLEEP**

<b>Address Offset</b>	0x0000 003C		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register provides bit fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_IVA_L domain transitions between ON and RET or SLEEP state. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT						RESERVED						RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 16 system clock cycles 0x1: Ramp-down counter is incremented every 64 system clock cycles 0x2: Ramp-down counter is incremented every 128 system clock cycles 0x3: Ramp-down counter is incremented every 512 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x00
15:10	RESERVED		R	0x00
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 16 system clock cycles 0x1: Ramp-up counter is incremented every 64 system clock cycles 0x2: Ramp-up counter is incremented every 128 system clock cycles 0x3: Ramp-up counter is incremented every 512 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Up_Prescal.	RW	0x00

**Table 3-782. Register Call Summary for Register PRM\_VOLTSETUP\_IVA\_RET\_SLEEP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-783. PRM\_VP\_CORE\_CONFIG**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	This register allows the configuration of the Voltage Processor dedicated to CORE Voltage Domain (VDD_CORE_L).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERROROFFSET								ERRORGAIN								INITVOLTAGE								RESERVED				TIMEOUTEN	INITVDD	FORCEUPDATE	VPENABLE

Bits	Field Name	Description	Type	Reset
31:24	ERROROFFSET	Offset value in the Error to Voltage converter (two's complement number).	RW	0x00
23:16	ERRORGAIN	Gain value in the Error to Voltage converter (two's complement number).	RW	0x00
15:8	INITVOLTAGE	Set the initial voltage level of the SMPS. It must be reconfigured before enable the SmartReflex around a new OPP.	RW	0x00
7:4	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
3	TIMEOUTEN	Enable or disable the timeout capability of the Voltage Controller State Machine.  0x0: Timeout is disabled. Loop will wait indefinitely. 0x1: Timeout will occur when TIMEOUT cycles have elapsed.	RW	0
2	INITVDD	Initializes the voltage in the Voltage Processor.  0x0: Reset the initialization bit. 0x1: The positive edge of InitVdd triggers a write of the value in the InitVoltage into the Voltage Processor.	RW	0
1	FORCEUPDATE	Forces an update of the SMPS.  0x0: Reset the force bit. 0x1: The positive edge of ForceUpdate triggers an update of the voltage to the SMPS.	RW	0
0	VPENABLE	Enables or disables the Voltage Processor updates on SR_SInterruptz.  0x0: Disables the Voltage Processor. 0x1: Enables the Voltage Processor.	RW	0

**Table 3-784. Register Call Summary for Register PRM\_VP\_CORE\_CONFIG**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-785. PRM\_VP\_CORE\_STATUS**

<b>Address Offset</b>	0x0000 0044		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register reflects the idle state of the Voltage Processor dedicated to the CORE Voltage Domain (VDD_CORE_L. This register is read only and automatically updated.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VPINIDLE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0000 0000
0	VPINIDLE	CORE Voltage Processor idle status. Read 0x0: The Voltage Processor for CORE is processing. Warm reset sensitive Read 0x1: The Voltage Processor for CORE is in idle state.	R	1

**Table 3-786. Register Call Summary for Register PRM\_VP\_CORE\_STATUS**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-787. PRM\_VP\_CORE\_VLIMITTO**

<b>Address Offset</b>	0x0000 0048		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the configuration of the voltage limits and timeout values of the Voltage Processor dedicated to the CORE Voltage Domain (VDD_CORE_L).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDMAX								VDDMIN								TIMEOUT															

Bits	Field Name	Description	Type	Reset
31:24	VDDMAX	Defines the maximum voltage supply level.	RW	0x00
23:16	VDDMIN	Defines the minimum voltage supply level.	RW	0x00
15:0	TIMEOUT	Defines Voltage Controller maximum wait time for responses.	RW	0x0000

**Table 3-788. Register Call Summary for Register PRM\_VP\_CORE\_VLIMITTO**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-789. PRM\_VP\_CORE\_VOLTAGE**

<b>Address Offset</b>	0x0000 004C		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register indicates the current value of the SMPS voltage for the Voltage Processor dedicated to the CORE Voltage Domain (VDD_CORE_L).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FORCEUPDATEWAIT																VPVOLTAGE															

Bits	Field Name	Description	Type	Reset
31:8	FORCEUPDATEWAIT	The time voltage processor needs to wait for SMPS to be settled after receiving SMPS acknowledge. This wait only be used during force_update operation.	RW	0x000111
7:0	VPVOLTAGE	Indicates the current SMPS programmed voltage.	R	0x00

**Table 3-790. Register Call Summary for Register PRM\_VP\_CORE\_VOLTAGE**

- PRCM Register Manual
- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-791. PRM\_VP\_CORE\_VSTEPMAX**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the programming of the maximum voltage step and waiting time of the Voltage Processor dedicated to CORE Voltage Domain (VDD_CORE_L).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPsoftwareAITTIMEMAX																VSTEPMAX							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x00
23:8	SMPsoftwareAITTIMEMAX	Slew rate for positive voltage step (in number of cycles per step).	RW	0x0000
7:0	VSTEPMAX	Maximum voltage step	RW	0x00

**Table 3-792. Register Call Summary for Register PRM\_VP\_CORE\_VSTEPMAX**

- PRCM Register Manual
- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-793. PRM\_VP\_CORE\_VSTEPMIN**

<b>Address Offset</b>	0x0000 0054		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the programming of the minimum voltage step and waiting time of the Voltage Processor dedicated to the CORE Voltage Domain (VDD_CORE_L).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPsoftwareAITTIMEMIN								VSTEPMIN															

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x00
23:8	SMPsoftwareAITTIMEMIN	Slew rate for negative voltage step (in number of cycles per step).	RW	0x0000
7:0	VSTEPMIN	Minimum voltage step	RW	0x00

**Table 3-794. Register Call Summary for Register PRM\_VP\_CORE\_VSTEPMIN**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-795. PRM\_VP\_MPU\_CONFIG**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the configuration of the Voltage Processor dedicated to MPU Voltage Domain (VDD_MPU_L).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERROROFFSET								ERRORGAIN								INITVOLTAGE								RESERVED		TIMEOUTEN	INITVDD	FORCEUPDATE	VPENABLE		

Bits	Field Name	Description	Type	Reset
31:24	ERROROFFSET	Offset value in the Error to Voltage converter (two's complement number).	RW	0x00
23:16	ERRORGAIN	Gain value in the Error to Voltage converter (two's complement number).	RW	0x00
15:8	INITVOLTAGE	Set the initial voltage level of the SMPS. It must be reconfigured before enable the SmartReflex around a new OPP.	RW	0x00
7:4	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
3	TIMEOUTEN	Enable or disable the timeout capability of the Voltage Controller State Machine. 0x0: Timeout is disabled. Loop will wait indefinitely. 0x1: Timeout will occur when TIMEOUT cycles have elapsed.	RW	0
2	INITVDD	Initializes the voltage in the Voltage Processor. 0x0: Reset the initialization bit. 0x1: The positive edge of InitVdd triggers a write of the value in the InitVoltage into the Voltage Processor.	RW	0

Bits	Field Name	Description	Type	Reset
1	FORCEUPDATE	Forces an update of the SMPS. 0x0: Reset the force bit. 0x1: The positive edge of ForceUpdate triggers an update of the voltage to the SMPS.	RW	0
0	VPENABLE	Enables or disables the Voltage Processor updates on SR_SInterruptz. 0x0: Disables the Voltage Processor. 0x1: Enables the Voltage Processor.	RW	0

**Table 3-796. Register Call Summary for Register PRM\_VP\_MPU\_CONFIG**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-797. PRM\_VP\_MPU\_STATUS**

<b>Address Offset</b>	0x0000 005C		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register reflects the idle state of the Voltage Processor dedicated to the MPU Voltage Domain (VDD_MPU_L). This register is read only and automatically updated.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VPINIDLE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0000 0000
0	VPINIDLE	Voltage Processor 1 idle status. Read 0x0: The Voltage Processor 1 is processing. Read 0x1: The Voltage Processor 1 is in idle state.	R	1

**Table 3-798. Register Call Summary for Register PRM\_VP\_MPU\_STATUS**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-799. PRM\_VP\_MPU\_VLIMITTO**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the configuration of the voltage limits and timeout values of the Voltage Processor dedicated to the MPU Voltage Domain (VDD_MPU_L).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDMAX								VDDMIN								TIMEOUT															

Bits	Field Name	Description	Type	Reset
31:24	VDDMAX	Defines the maximum voltage supply level.	RW	0x00
23:16	VDDMIN	Defines the minimum voltage supply level.	RW	0x00
15:0	TIMEOUT	Defines Voltage Controller maximum wait time for responses.	RW	0x0000

**Table 3-800. Register Call Summary for Register PRM\_VP\_MPU\_VLIMITTO**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-801. PRM\_VP\_MPU\_VOLTAGE**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register indicates the current value of the SMPS voltage for the Voltage Processor dedicated to the MPU Voltage Domain (VDD_MPU_L).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FORCEUPDATEWAIT																VPVOLTAGE															

Bits	Field Name	Description	Type	Reset
31:8	FORCEUPDATEWAIT	The time voltage processor needs to wait for SMPS to be settled after receiving SMPS acknowledge. This wait only be used during force_update operation.	RW	0x000111
7:0	VPVOLTAGE	Indicates the current SMPS programmed voltage.	R	0x00

**Table 3-802. Register Call Summary for Register PRM\_VP\_MPU\_VOLTAGE**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-803. PRM\_VP\_MPU\_VSTEPMAX**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the programming of the maximum voltage step and waiting time of the Voltage Processor dedicated to MPU Voltage Domain (VDD_MPU_L).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPsoftwareAITTIMEMAX																VSTEPMAX							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x00
23:8	SMPsoftwareAITTIMEMAX	Slew rate for positive voltage step (in number of cycles per step).	RW	0x0000
7:0	VSTEPMAX	Maximum voltage step	RW	0x00

**Table 3-804. Register Call Summary for Register PRM\_VP\_MPU\_VSTEPMAX**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)



**Table 3-805. PRM\_VP\_MPU\_VSTEPMIN**

<b>Address Offset</b>	0x0000 006C		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the programming of the minimum voltage step and waiting time of the Voltage Processor dedicated to the MPU Voltage Domain (VDD_MPU_L).		
<b>Type</b>	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	SMPsoftwareAITTIMEMIN		VSTEPMIN

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x00
23:8	SMPsoftwareAITTIMEMIN	Slew rate for negative voltage step (in number of cycles per step).	RW	0x0000
7:0	VSTEPMIN	Minimum voltage step	RW	0x00

**Table 3-806. Register Call Summary for Register PRM\_VP\_MPU\_VSTEPMIN**

- PRCM Register Manual
- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-807. PRM\_VP\_IVA\_CONFIG**

<b>Address Offset</b>	0x0000 0070		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the configuration of the Voltage Processor dedicated to IVA Voltage Domain (VDD_IVA_L).		
<b>Type</b>	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
ERROROFFSET	ERRORGAIN	INITVOLTAGE	RESERVED TIMEOUTEN INITVDD FORCEUPDATE VPENABLE

Bits	Field Name	Description	Type	Reset
31:24	ERROROFFSET	Offset value in the Error to Voltage converter (two's complement number).	RW	0x00
23:16	ERRORGAIN	Gain value in the Error to Voltage converter (two's complement number).	RW	0x00
15:8	INITVOLTAGE	Set the initial voltage level of the SMPS. It must be reconfigured before enable the SmartReflex around a new OPP.	RW	0x00
7:4	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
3	TIMEOUTEN	Enable or disable the timeout capability of the Voltage Controller State Machine. 0x0: Timeout is disabled. Loop will wait indefinitely. 0x1: Timeout will occur when TIMEOUT cycles have elapsed.	RW	0
2	INITVDD	Initializes the voltage in the Voltage Processor. 0x0: Reset the initialization bit. 0x1: The positive edge of InitVdd triggers a write of the value in the InitVoltage into the Voltage Processor.	RW	0

Bits	Field Name	Description	Type	Reset
1	FORCEUPDATE	Forces an update of the SMPS. 0x0: Reset the force bit. 0x1: The positive edge of ForceUpdate triggers an update of the voltage to the SMPS.	RW	0
0	VPENABLE	Enables or disables the Voltage Processor updates on SR_SInterruptz. 0x0: Disables the Voltage Processor. 0x1: Enables the Voltage Processor.	RW	0

**Table 3-808. Register Call Summary for Register PRM\_VP\_IVA\_CONFIG**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-809. PRM\_VP\_IVA\_STATUS**

<b>Address Offset</b>	0x0000 0074		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register reflects the idle state of the Voltage Processor dedicated to the MPU Voltage Domain (VDD_IVA_L. This register is read only and automatically updated.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VPINIDLE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0000 0000
0	VPINIDLE	Voltage Processor 1 idle status. Read 0x0: The Voltage Processor 1 is processing. Read 0x1: The Voltage Processor 1 is in idle state.	R	1

**Table 3-810. Register Call Summary for Register PRM\_VP\_IVA\_STATUS**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-811. PRM\_VP\_IVA\_VLIMITTO**

<b>Address Offset</b>	0x0000 0078		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the configuration of the voltage limits and timeout values of the Voltage Processor dedicated to the IVA Voltage Domain (VDD_IVA_L).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDMAX				VDDMIN				TIMEOUT																							

Bits	Field Name	Description	Type	Reset
31:24	VDDMAX	Defines the maximum voltage supply level.	RW	0x00
23:16	VDDMIN	Defines the minimum voltage supply level.	RW	0x00
15:0	TIMEOUT	Defines Voltage Controller maximum wait time for responses.	RW	0x0000

**Table 3-812. Register Call Summary for Register PRM\_VP\_IVA\_VLIMITTO**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-813. PRM\_VP\_IVA\_VOLTAGE**

<b>Address Offset</b>	0x0000 007C		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register indicates the current value of the SMPS voltage for the Voltage Processor dedicated to the IVA Voltage Domain (VDD_IVA_L).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FORCEUPDATEWAIT																VPVOLTAGE															

Bits	Field Name	Description	Type	Reset
31:8	FORCEUPDATEWAIT	The time voltage processor needs to wait for SMPS to be settled after receiving SMPS acknowledge. This wait only be used during force_update operation.	RW	0x000111
7:0	VPVOLTAGE	Indicates the current SMPS programmed voltage.	R	0x00

**Table 3-814. Register Call Summary for Register PRM\_VP\_IVA\_VOLTAGE**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-815. PRM\_VP\_IVA\_VSTEPMAX**

<b>Address Offset</b>	0x0000 0080		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the programming of the maximum voltage step and waiting time of the Voltage Processor dedicated to IVA Voltage Domain (VDD_IVA_L).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPsoftwareAITTIMEMAX																VSTEPMAX							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x00
23:8	SMPsoftwareAITTIMEMAX	Slew rate for positive voltage step (in number of cycles per step).	RW	0x0000
7:0	VSTEPMAX	Maximum voltage step	RW	0x00

**Table 3-816. Register Call Summary for Register PRM\_VP\_IVA\_VSTEPMAX**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-817. PRM\_VP\_IVA\_VSTEPMIN**

<b>Address Offset</b>	0x0000 0084		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the programming of the minimum voltage step and waiting time of the Voltage Processor dedicated to the IVA Voltage Domain (VDD_IVA_L).		
<b>Type</b>	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	SMPsoftwareAITTIMEMIN		VSTEPMIN

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x00
23:8	SMPsoftwareAITTIMEMIN	Slew rate for negative voltage step (in number of cycles per step).	RW	0x0000
7:0	VSTEPMIN	Minimum voltage step	RW	0x00

**Table 3-818. Register Call Summary for Register PRM\_VP\_IVA\_VSTEPMIN**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-819. PRM\_VC\_SMPS\_SA**

<b>Address Offset</b>	0x0000 0088		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the setting of the I2C slave address of the Power IC device. [warm reset insensitive]		
<b>Type</b>	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	SA_VDD_MPU_L	RESERVED	SA_VDD_IVA_L
		RESERVED	SA_VDD_CORE_L

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x000
22:16	SA_VDD_MPU_L	Set the I2C slave address value for the third (if any) Power IC device.	RW	0x00
15	RESERVED		R	0
14:8	SA_VDD_IVA_L	Set the I2C slave address value for the second (if any) Power IC device.	RW	0x00
7	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0
6:0	SA_VDD_CORE_L	Set the I2C slave address value for the first Power IC device.	RW	0x00

**Table 3-820. Register Call Summary for Register PRM\_VC\_SMPS\_SA**

PRCM Module Programming Guide

- [Global Initialization: \[0\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[1\]](#)

**Table 3-821. PRM\_VC\_VAL\_SMPS\_RA\_VOL**

<b>Address Offset</b>	0x0000 008C	
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b> DEVICE_PRM
<b>Description</b>	This register allows the setting of the voltage configuration register address for the VDD channels. [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VOLRA_VDD_MPU_L								VOLRA_VDD_IVA_L								VOLRA_VDD_CORE_L							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x00
23:16	VOLRA_VDD_MPU_L	Voltage configuration register address value for VDD_MPU_L channel (if VDD_MPU_L source is placed in same chip as VDD_CORE_L source and have different voltage configuration register)	RW	0x00
15:8	VOLRA_VDD_IVA_L	Voltage configuration register address value for VDD_IVA_L channel (if VDD_IVA_L source is placed in same chip as VDD_CORE_L source and have different voltage configuration register)	RW	0x00
7:0	VOLRA_VDD_CORE_L	Set the voltage configuration register address value for the VDD_CORE_L channel.	RW	0x00

**Table 3-822. Register Call Summary for Register PRM\_VC\_VAL\_SMPS\_RA\_VOL**

PRCM Module Programming Guide

- [Global Initialization: \[0\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[1\]](#)

**Table 3-823. PRM\_VC\_VAL\_SMPS\_RA\_CMD**

<b>Address Offset</b>	0x0000 0090	
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b> DEVICE_PRM
<b>Description</b>	Command (ON/ON-Low-Power/Retention/OFF) configuration register address values for the VDD channels (if used SMPS chips have different command configuration register than voltage configuration register) [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CMDRA_VDD_MPU_L								CMDRA_VDD_IVA_L								CMDRA_VDD_CORE_L							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x00
23:16	CMDRA_VDD_MPU_L	Command (ON/ON-Low-Power /Retention/OFF) configuration register address value for VDD_MPU_L channel (if VDD_IVA_L source has different command configuration register than voltage VDD_CORE_L)	RW	0x00
15:8	CMDRA_VDD_IVA_L	Command (ON/ON-Low-Power /Retention/OFF) configuration register address value for VDD_IVA_L channel (if VDD_IVA_L source has different command configuration register than voltage VDD_CORE_L)	RW	0x00
7:0	CMDRA_VDD_CORE_L	Command (ON/ON-Low-Power /Retention/OFF) configuration register address value for VDD_CORE_L channel.	RW	0x00

**Table 3-824. Register Call Summary for Register PRM\_VC\_VAL\_SMPS\_RA\_CMD**

PRCM Module Programming Guide

- [Global Initialization: \[0\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[1\]](#)

**Table 3-825. PRM\_VC\_VAL\_CMD\_VDD\_CORE\_L**

<b>Address Offset</b>	0x0000 0094		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the setting of the ON/ON-Low-Power/Retention/OFF command values for VDD_CORE_L channel. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ON								ONLP								RET								OFF							

Bits	Field Name	Description	Type	Reset
31:24	ON	Set the ON command value.	RW	0x00
23:16	ONLP	Set the ON-Low-Power command value.	RW	0x00
15:8	RET	Set the RET command value.	RW	0x00
7:0	OFF	Set the OFF command value.	RW	0x00

**Table 3-826. Register Call Summary for Register PRM\_VC\_VAL\_CMD\_VDD\_CORE\_L**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-827. PRM\_VC\_VAL\_CMD\_VDD\_MPU\_L**

<b>Address Offset</b>	0x0000 0098		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the setting of the ON/ON-Low-Power/Retention/OFF command values for VDD_MPU_L channel. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ON								ONLP								RET								OFF							

Bits	Field Name	Description	Type	Reset
31:24	ON	Set the ON command value.	RW	0x00
23:16	ONLP	Set the ON-Low-Power command value.	RW	0x00
15:8	RET	Set the RET command value.	RW	0x00
7:0	OFF	Set the OFF command value.	RW	0x00

**Table 3-828. Register Call Summary for Register PRM\_VC\_VAL\_CMD\_VDD\_MPU\_L**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-829. PRM\_VC\_VAL\_CMD\_VDD\_IVA\_L**

<b>Address Offset</b>	0x0000 009C		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	This register allows the setting of the ON/ON-Low-Power/Retention/OFF command values for VDD_IVA_L channel. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ON								ONLP								RET								OFF							

Bits	Field Name	Description	Type	Reset
31:24	ON	Set the ON command value.	RW	0x00
23:16	ONLP	Set the ON-Low-Power command value.	RW	0x00
15:8	RET	Set the RET command value.	RW	0x00
7:0	OFF	Set the OFF command value.	RW	0x00

**Table 3-830. Register Call Summary for Register PRM\_VC\_VAL\_CMD\_VDD\_IVA\_L**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-831. PRM\_VC\_VAL\_BYPASS**

<b>Address Offset</b>	0x0000 00A0		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Bypass data values register used for bypass command channel to send other configuration information (other than voltage configuration parameters) for SMPS chips which have no other configuration interface then this I2C interface. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED								VALID	DATA								REGADDR								RESERVED	SLAVEADDR							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	VALID	This bit validates the bypass command. It is automatically cleared by hardware either after getting the acknowledge back from the SMPS or if an error occurred. Read 0x0: The last command send has been acknowledged 0x1: Pending command is being process	RW WSpecial	0
23:16	DATA	Data to send to the Power IC device.	RW	0x00
15:8	REGADDR	Set the address of Power IC device register to configure.	RW	0x00
7	RESERVED		R	0
6:0	SLAVEADDR	Set the I2C slave address value.	RW	0x00

**Table 3-832. Register Call Summary for Register PRM\_VC\_VAL\_BYPASS**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-833. PRM\_VC\_CFG\_CHANNEL**

<b>Address Offset</b>	0x0000 00A4	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	This register allows the configuration pointers for both VDD channels. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RACEN_VDD_MPU_L	RAC_VDD_MPU_L	RAV_VDD_MPU_L	CMD_VDD_MPU_L	SA_VDD_MPU_L	RESERVED	CMD_VDD_IVA_L	RACEN_VDD_IVA_L	RAC_VDD_IVA_L	RAV_VDD_IVA_L	SA_VDD_IVA_L	RESERVED	CMD_VDD_CORE_L	RACEN_VDD_CORE_L	RAC_VDD_CORE_L	RAV_VDD_CORE_L	SA_VDD_CORE_L							

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20	RACEN_VDD_MPU_L	Enable bit for usage of RAC_VDD_MPU_L  0x0: VDD_MPU_L channel uses VOLRA values for register address of VFSM-s commands. VFSM-s commands goes also to voltage configuration register.  0x1: VDD_MPU_L channel uses CMDRA values for register address of VFSM-s commands. VFSM-s commands goes to different command configuration register.	RW	0
19	RAC_VDD_MPU_L	Command (ON/ON-Low-Power/Retention/OFF) configuration register address pointer for VDD_MPU_L channel  0x0: Select CMDRA_VDD_CORE_L for VDD_MPU_L channel  0x1: Select CMDRA_VDD_MPU_L for VDD_MPU_L channel	RW	0
18	RAV_VDD_MPU_L	Voltage configuration register address pointer for VDD_MPU_L channel.  0x0: Select VOLRA_VDD_CORE_L for VDD_MPU_L channel  0x1: Select VOLRA_VDD_MPU_L for VDD_MPU_L channel	RW	0
17	CMD_VDD_MPU_L	Command values (ON/ON-Low-Power/Retention/OFF voltage values) set selection for VDD_MPU_L channel  0x0: VDD_MPU_L channel use VC_VAL_CMD_VDD_CORE_L set for command values  0x1: VDD_MPU_L channel use VC_VAL_CMD_VDD_MPU_L set for command values	RW	0
16	SA_VDD_MPU_L	Slave address pointer for VDD_MPU_L channel.  0x0: Select SA_VDD_CORE_L for VDD_MPU_L channel  0x1: Select SA_VDD_MPU_L for VDD_MPU_L channel	RW	0
15:13	RESERVED		R	0x0
12	CMD_VDD_IVA_L	Command values (ON/ON-Low-Power/Retention/OFF voltage values) set selection for VDD_IVA_L channel  0x0: VDD_IVA_L channel use VC_VAL_CMD_VDD_CORE_L set for command values  0x1: VDD_IVA_L channel use VC_VAL_CMD_VDD_IVA_L set for command values	RW	0



Bits	Field Name	Description	Type	Reset
11	RACEN_VDD_IVA_L	Enable bit for usage of RAC_VDD_IVA_L  0x0: VDD_IVA_L channel uses VOLRA values for register address of VFSM-s commands. VFSM-s commands goes also to voltage configuration register.  0x1: VDD_IVA_L channel uses CMDRA values for register address of VFSM-s commands. VFSM-s commands goes to different command configuration register.	RW	0
10	RAC_VDD_IVA_L	Command (ON/ON-Low-Power/Retention/OFF) configuration register address pointer for VDD_IVA_L channel  0x0: Select CMDRA_VDD_CORE_L for VDD_IVA_L channel  0x1: Select CMDRA_VDD_IVA_L for VDD_IVA_L channel	RW	0
9	RAV_VDD_IVA_L	Voltage configuration register address pointer for VDD_IVA_L channel.  0x0: Select VOLRA_VDD_CORE_L for VDD_IVA_L channel  0x1: Select VOLRA_VDD_IVA_L for VDD_IVA_L channel	RW	0
8	SA_VDD_IVA_L	Slave address pointer for VDD_IVA_L channel.  0x0: Select SA_VDD_CORE_L for VDD_IVA_L channel  0x1: Select SA_VDD_IVA_L for VDD_IVA_L channel	RW	0
7:5	RESERVED		R	0x0
4	CMD_VDD_CORE_L	Command values (ON/ON-Low-Power/Retention/OFF voltage values) set selection for VDD_CORE_L channel (This bit has no influence on VDD_CORE_L channel)	RW	0
3	RACEN_VDD_CORE_L	Enable bit for usage of RAC_VDD_CORE_L.  0x0: VDD_CORE_L channel use VOLRA values for register address of VFSM-s commands. VFSM-s commands goes also to voltage configuration register.  0x1: VDD_CORE_L channel use CMDRA values for register address of VFSM-s commands. VFSM-s commands goes to different command configuration register.	RW	0
2	RAC_VDD_CORE_L	Command (ON/ON-Low-Power/Retention/OFF) configuration register address pointer for VDD_CORE_L channel. (This bit has no influence on first VDD_CORE_L channel)	RW	0
1	RAV_VDD_CORE_L	Voltage configuration register address pointer for VDD_CORE_L channel. (This bit has no influence on VDD_CORE_L channel) .	RW	0
0	SA_VDD_CORE_L	Slave address pointer for VDD_CORE_L channel. (This bit has no influence on first VDD_CORE_L channel)	RW	0

**Table 3-834. Register Call Summary for Register PRM\_VC\_CFG\_CHANNEL**

PRCM Module Programming Guide

- [Global Initialization: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[5\]](#)

**Table 3-835. PRM\_VC\_CFG\_I2C\_MODE**

<b>Address Offset</b>	0x0000 00A8		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	I2C configuration register. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DFILTEREN		RESERVED	SRMODEEN	HSMODEEN	HSMCODE										

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	DFILTEREN	This field enables double filter procedure for I2C input lines  0x0: I2C bus digital filter rejects all glitches smaller than 1 sytem clock cycle 0x1: I2C bus digital filter rejects all glitches smaller than 2 sytem clock cycle	RW	0
5	RESERVED		R	0
4	SRMODEEN	Enables the I2C repeated start operation mode. 0x0: Disables the repeated start operation mode 0x1: Enables the repeated start operation mode	RW	1
3	HSMODEEN	Enables I2C bus high-speed mode. 0x0: Disables the I2C high-speed mode 0x1: Enables the I2C high-speed mode	RW	1
2:0	HSMCODE	Master code value for I2C high-speed preamble transmission.	RW	0x0

**Table 3-836. Register Call Summary for Register PRM\_VC\_CFG\_I2C\_MODE**

PRCM Module Programming Guide

- [Global Initialization: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[4\]](#)

**Table 3-837. PRM\_VC\_CFG\_I2C\_CLK**

<b>Address Offset</b>	0x0000 00AC		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	I2C Interface clock configuration parameters. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSSCLL								HSSCLH								SCLL								SCLH							

Bits	Field Name	Description	Type	Reset
31:24	HSSCLL	Number of the system clock cycles, necessary to count the low period of the I2C clock signal, when the I2C interface runs in high-speed mode of operation. The value of the bit field is automatically increased by 7.	RW	0x00

Bits	Field Name	Description	Type	Reset
23:16	HSSCLH	Number of the system clock cycles, necessary to count the high period of the I2C clock signal, when the I2C interface runs in high-speed mode of operation. The value of the bit field is automatically increased by 5.	RW	0x00
15:8	SCLL	Number of the system clock cycles, necessary to count the low period of the I2C clock signal, when the I2C interface runs in fast mode of operation. The value of the bit field is automatically increased by 7.	RW	0x00
7:0	SCLH	Number of the system clock cycles, necessary to count the high period of the I2C clock signal, when the I2C interface runs in fast mode of operation. The value of the bit field is automatically increased by 5.	RW	0x00

**Table 3-838. Register Call Summary for Register PRM\_VC\_CFG\_I2C\_CLK**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-839. PRM\_SRAM\_COUNT**

<b>Address Offset</b>	0x0000 00B0		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Common setup for SRAM LDO transition counters. Applies to all voltage domains. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
STARTUP_COUNT								SLPCNT_VALUE								VSETUPCNT_VALUE								RESERVED	PCHARGEcnt_VALUE							

Bits	Field Name	Description	Type	Reset
31:24	STARTUP_COUNT	Determines the start-up duration of SRAM and ABB LDO. The duration is computed as 16 x NbCycles of system clock cycles. Target is 50us.	RW	0x78
23:16	SLPCNT_VALUE	Delay between retention/off assertion of last SRAM bank and SRAMALLRET signal to LDO is driven high. Counting on system clock. Target is 2us.	RW	0x00
15:8	VSETUPCNT_VALUE	SRAM LDO rampup time from retention to active mode. The duration is computed as 8 x NbCycles of system clock cycles. Target is 30us.	RW	0x00
7:6	RESERVED		R	0x0
5:0	PCHARGEcnt_VALUE	Delay between de-assertion of standby_rta_ret_on and standby_rta_ret_good. Counting on system clock. Target is 600ns.	RW	0x17

**Table 3-840. Register Call Summary for Register PRM\_SRAM\_COUNT**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-841. PRM\_SRAM\_WKUP\_SETUP**

<b>Address Offset</b>	0x0000 00B4		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Setup of memory in WKUP voltage domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DISABLE_RTA_EXPORT															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	DISABLE_RTA_EXPORT	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this.  0x0: HD memory RTA feature is enabled 0x1: HD memory RTA feature is disabled	RW WSpecial	0

**Table 3-842. Register Call Summary for Register PRM\_SRAM\_WKUP\_SETUP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-843. PRM\_LDO\_SRAM\_CORE\_SETUP**

<b>Address Offset</b>	0x0000 00B8		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Setup of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3_EXPORT	ENFUNC2_EXPORT	ENFUNC1_EXPORT	ABBOFF_SLEEP_EXPORT	ABBOFF_ACT_EXPORT	DISABLE_RTA_EXPORT							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overridden 0x1: AIPOFF signal is overridden to 1. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0
5	ENFUNC3_EXPORT	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW WSpecial	0
4	ENFUNC2_EXPORT	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW WSpecial	0
3	ENFUNC1_EXPORT	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW WSpecial	0
2	ABBOFF_SLEEP_EXPORT	Determines whether SRAMNWA is supplied by VDD5 or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: SRAMNWA supplied with VDD5 0x1: SRAMNWA supplied with VDDAR	RW WSpecial	0
1	ABBOFF_ACT_EXPORT	Determines whether SRAMNWA is supplied by VDD5 or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: SRAMNWA supplied with VDD5 0x1: SRAMNWA supplied with VDDAR	RW WSpecial	0
0	DISABLE_RTA_EXPORT	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: HD memory RTA feature is enabled 0x1: HD memory RTA feature is disabled	RW WSpecial	0

**Table 3-844. Register Call Summary for Register PRM\_LDO\_SRAM\_CORE\_SETUP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-845. PRM\_LDO\_SRAM\_CORE\_CTRL**

<b>Address Offset</b>	0x0000 00BC		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SRAM_IN_TRANSITION		SRAMLDO_STATUS		RESERVED										RETMODE_ENABLE	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. Read 0x0: SRAM LDO state machine is stable Read 0x1: SRAM LDO state machine is in transition state	R	0
8	SRAMLDO_STATUS	SRAMLDO status Read 0x0: SRAMLDO is in ACTIVE mode. Read 0x1: SRAMLDO is on RETENTION mode.	R	0
7:1	RESERVED		R	0x00
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0

**Table 3-846. Register Call Summary for Register PRM\_LDO\_SRAM\_CORE\_CTRL**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-847. PRM\_LDO\_SRAM\_MPU\_SETUP**

<b>Address Offset</b>	0x0000 00C0		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Setup of the SRAM LDO for MPU voltage domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3_ECPORT	ENFUNC2_EXPORT	ENFUNC1_EXPORT	ABBOFF_SLEEP_EXPORT	ABBOFF_ACT_EXPORT	DISABLE_RTA_EXPORT							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overridden 0x1: AIPOFF signal is overridden to 1. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0
5	ENFUNC3_EXPORT	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0
4	ENFUNC2_EXPORT	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW WSpecial	0
3	ENFUNC1_EXPORT	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0
2	ABBOFF_SLEEP_EXPORT	Determines whether SRAMNWA is supplied by VDD5 or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: SRAMNWA supplied with VDD5 0x1: SRAMNWA supplied with VDDAR	RW WSpecial	0
1	ABBOFF_ACT_EXPORT	Determines whether SRAMNWA is supplied by VDD5 or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: SRAMNWA supplied with VDD5 0x1: SRAMNWA supplied with VDDAR	RW WSpecial	0
0	DISABLE_RTA_EXPORT	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: HD memory RTA feature is enabled 0x1: HD memory RTA feature is disabled	RW WSpecial	0

**Table 3-848. Register Call Summary for Register PRM\_LDO\_SRAM\_MPU\_SETUP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-849. PRM\_LDO\_SRAM\_MPU\_CTRL**

<b>Address Offset</b>	0x0000 00C4		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Control and status of the SRAM LDO for MPU voltage domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SRAM_IN_TRANSITION		SRAMLDO_STATUS		RESERVED										RETMODE_ENABLE	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. Read 0x0: SRAM LDO state machine is stable Read 0x1: SRAM LDO state machine is in transition state	R	0
8	SRAMLDO_STATUS	SRAMLDO status Read 0x0: SRAMLDO is in ACTIVE mode. Read 0x1: SRAMLDO is on RETENTION mode.	R	0
7:1	RESERVED		R	0x00
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0

**Table 3-850. Register Call Summary for Register PRM\_LDO\_SRAM\_MPU\_CTRL**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-851. PRM\_LDO\_SRAM\_IVA\_SETUP**

<b>Address Offset</b>	0x0000 00C8		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Setup of the SRAM LDO for IVA voltage domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3_EXPORT	ENFUNC2_EXPORT	ENFUNC1_EXPORT	ABBOFF_SLEEP_EXPORT	ABBOFF_ACT_EXPORT	DISABLE_RTA_EXPORT							



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overridden 0x1: AIPOFF signal is overridden to 1. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0
5	ENFUNC3_EXPORT	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0
4	ENFUNC2_EXPORT	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW WSpecial	0
3	ENFUNC1_EXPORT	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0
2	ABBOFF_SLEEP_EXPORT	Determines whether SRAMNWA is supplied by VDD5 or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: SRAMNWA supplied with VDD5 0x1: SRAMNWA supplied with VDDAR	RW WSpecial	0
1	ABBOFF_ACT_EXPORT	Determines whether SRAMNWA is supplied by VDD5 or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: SRAMNWA supplied with VDD5 0x1: SRAMNWA supplied with VDDAR	RW WSpecial	0
0	DISABLE_RTA_EXPORT	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bit field is automatically loaded with an Efuse value from control module. Bit field remains writable after this. 0x0: HD memory RTA feature is enabled 0x1: HD memory RTA feature is disabled	RW WSpecial	0

**Table 3-852. Register Call Summary for Register PRM\_LDO\_SRAM\_IVA\_SETUP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-853. PRM\_LDO\_SRAM\_IVA\_CTRL**

<b>Address Offset</b>	0x0000 00CC		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Control and status of the SRAM LDO for IVA voltage domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SRAM_IN_TRANSITION		SRAMLDO_STATUS	RESERVED						RETMODE_ENABLE						

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. Read 0x0: SRAM LDO state machine is stable Read 0x1: SRAM LDO state machine is in transition state	R	0
8	SRAMLDO_STATUS	SRAMLDO status Read 0x0: SRAMLDO is in ACTIVE mode. Read 0x1: SRAMLDO is on RETENTION mode.	R	0
7:1	RESERVED		R	0x00
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0

**Table 3-854. Register Call Summary for Register PRM\_LDO\_SRAM\_IVA\_CTRL**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-855. PRM\_LDO\_ABB\_MPU\_SETUP**

<b>Address Offset</b>	0x0000 00D0		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Selects the MPU_ABB LDO mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SR2_WTCNT_VALUE						RESERVED		SLEEP_RBB_SEL	ACTIVE_FBB_SEL	ACTIVE_RBB_SEL	SR2EN				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x00
7:4	RESERVED		R	0x0
3	SLEEP_RBB_SEL	Defines ABB LDO mode when voltage is in sleep. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in RBB mode	RW	0
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0
1	ACTIVE_RBB_SEL	Defines ABB LDO mode when voltage is in slow OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in RBB mode	RW	0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0

**Table 3-856. Register Call Summary for Register PRM\_LDO\_ABB\_MPU\_SETUP**

Voltage Management Functional Description

- [ABB LDOs Control: \[0\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[1\]](#)

**Table 3-857. PRM\_LDO\_ABB\_MPU\_CTRL**

<b>Address Offset</b>	0x0000 00D4	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	Control and Status of ABB on MPU voltage domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SR2_IN_TRANSITION	RESERVED	SR2_STATUS	OPP_CHANGE	OPP_SEL											

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. Read 0x0: Read 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0
5	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
4:3	SR2_STATUS	Indicate ABB LDO current operation status Read 0x0: ABB LDO is placed in bypass mode. Read 0x1: ABB LDO is placed in RBB active mode. Read 0x2: ABB LDO is placed in FBB active mode. Read 0x3: Reserved	R	0x0
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL & ACTIVE_FBB_SEL/ACTIVE_RBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW WSpecial	0
1:0	OPP_SEL	Selects the OPP at which the MPU voltage domain is operating 0x0: default : Nominal 0x1: Fast OPP 0x2: Nominal 0x3: Slow OPP	RW	0x0

**Table 3-858. Register Call Summary for Register PRM\_LDO\_ABB\_MPU\_CTRL**

Voltage Management Functional Description

- [ABB LDOs Control: \[0\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[1\]](#)

**Table 3-859. PRM\_LDO\_ABB\_IVA\_SETUP**

<b>Address Offset</b>	0x0000 00D8	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	Selects the IVA_ABB LDO mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SR2_WTCNT_VALUE								RESERVED				SLEEP_RBB_SEL	ACTIVE_FBB_SEL	ACTIVE_RBB_SEL	SR2EN								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x00
7:4	RESERVED		R	0x0
3	SLEEP_RBB_SEL	Defines ABB LDO mode when voltage is in sleep. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in RBB mode	RW	0

Bits	Field Name	Description	Type	Reset
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0
1	ACTIVE_RBB_SEL	Defines ABB LDO mode when voltage is in slow OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in RBB mode	RW	0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0

**Table 3-860. Register Call Summary for Register PRM\_LDO\_ABB\_IVA\_SETUP**

Voltage Management Functional Description

- [ABB LDOs Control: \[0\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[1\]](#)

**Table 3-861. PRM\_LDO\_ABB\_IVA\_CTRL**

<b>Address Offset</b>	0x0000 00DC		
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b>	DEVICE_PRM
<b>Description</b>	Control and Status of ABB on IVA voltage domain. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SR2_IN_TRANSITION		RESERVED	SR2_STATUS		OPP_CHANGE	OPP_SEL									

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. Read 0x0: Read 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0
5	RESERVED		R	0
4:3	SR2_STATUS	Indicate ABB LDO current operation status Read 0x0: ABB LDO is placed in bypass mode. Read 0x1: ABB LDO is placed in RBB active mode. Read 0x2: ABB LDO is placed in FBB active mode. Read 0x3: Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL & ACTIVE_FBB_SEL/ACTIVE_RBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW WSpecial	0
1:0	OPP_SEL	Selects the OPP at which the IVA voltage domain is operating (Fast OPP, Nominal OPP or Slow OPP) 0x0: default : Nominal 0x1: Fast OPP 0x2: Nominal 0x3: Slow OPP	RW	0x0

**Table 3-862. Register Call Summary for Register PRM\_LDO\_ABB\_IVA\_CTRL**

Voltage Management Functional Description

- [ABB LDOs Control: \[0\]](#)

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[1\]](#)

**Table 3-863. PRM\_LDO\_BANDGAP\_SETUP**

<b>Address Offset</b>	0x0000 00E0	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	Control of the bandgap. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STARTUP_COUNT															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STARTUP_COUNT	Determines the start-up duration of BANDGAP. The duration is computed as 32 x NbCycles of system clock cycles. Target is 100us.	RW	0x78

**Table 3-864. Register Call Summary for Register PRM\_LDO\_BANDGAP\_SETUP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-865. PRM\_DEVICE\_OFF\_CTRL**

<b>Address Offset</b>	0x0000 00E4	<b>Instance</b>	DEVICE_PRM
<b>Physical Address</b>	See <a href="#">Table 3-750</a>		
<b>Description</b>	This register is used to control device OFF transition.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EMIF2_OFFWKUP_DISABLE		EMIF1_OFFWKUP_DISABLE		RESERVED							DEVICE_OFF_ENABLE				

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0000 00
9	EMIF2_OFFWKUP_DISABLE	Controls the EMIF2_DEVICE_OFFWKUP_CORESRTACTST notifier sent to EMIF2 upon a device wakeup from off mode. (Warm reset insensitive) 0x0: Notifier is activated. 0x1: Notifier is not activated – stays low.	R	0
8	EMIF1_OFFWKUP_DISABLE	Controls the EMIF1_DEVICE_OFFWKUP_CORESRTACTST notifier sent to EMIF1 upon a device wakeup from off mode. (Warm reset insensitive) 0x0: Notifier is activated. 0x1: Notifier is not activated – stays low.	R	0
7:1	RESERVED		R	0x00
0	DEVICE_OFF_ENABLE	Controls transition to device OFF mode. 0x0: Device is not allowed to perform transition to off mode 0x1: Device is allowed to perform transition to off mode as soon as all power domains in MPU, IVA and CORE voltage are in OFF or OsoftwareRET state (open switch retention)	RW	0

**Table 3-866. Register Call Summary for Register PRM\_DEVICE\_OFF\_CTRL**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-867. PRM\_PHASE1\_CNDP**

<b>Address Offset</b>	0x0000 00E8
<b>Physical Address</b>	See <a href="#">Table 3-750</a>
<b>Description</b>	This register stores the start descriptor address of automatic restore phase1. [warm reset insensitive]
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE1_CNDP																															

Bits	Field Name	Description	Type	Reset
31:0	PHASE1_CNDP	Start descriptor address of automatic restore phase1. Hard-coded to SAR_ROM base address.	R Rreturns	0x4A05 E000

**Table 3-868. Register Call Summary for Register PRM\_PHASE1\_CNDP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-869. PRM\_PHASE2A\_CNDP**

<b>Address Offset</b>	0x0000 00EC	
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b> DEVICE_PRM
<b>Description</b>	This register stores the start descriptor address of automatic restore phase2A. [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE2A_CNDP																															

Bits	Field Name	Description	Type	Reset
31:0	PHASE2A_CNDP	Start descriptor address of automatic restore phase2A. Hard-coded to SAR_ROM base address + 0x30.	R Rreturns	0x4A05 E030

**Table 3-870. Register Call Summary for Register PRM\_PHASE2A\_CNDP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-871. PRM\_PHASE2B\_CNDP**

<b>Address Offset</b>	0x0000 00F0	
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b> DEVICE_PRM
<b>Description</b>	This register stores the start descriptor address of automatic restore phase2B. [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE2B_CNDP																															

Bits	Field Name	Description	Type	Reset
31:0	PHASE2B_CNDP	Start descriptor address of automatic restore phase2B. Hard-coded to SAR_ROM base address + 0x60.	R Rreturns	0x4A05 E060

**Table 3-872. Register Call Summary for Register PRM\_PHASE2B\_CNDP**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

**Table 3-873. PRM\_VC\_ERRST**

<b>Address Offset</b>	0x0000 00F8	
<b>Physical Address</b>	See <a href="#">Table 3-750</a>	<b>Instance</b> DEVICE_PRM
<b>Description</b>	This debug register logs the error status coming from Voltage Controller. Must be cleared by software.	
<b>Type</b>	RW	



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																					
RESERVED				BYPASS_TIMEOUT_ERR			BYPASS_RA_ERR		BYPASS_SA_ERR		RESERVED		VFMSM_TIMEOUT_ERR_MPU		VFMSM_RA_ERR_MPU		VFMSM_SA_ERR_MPU		SMPS_TIMEOUT_ERR_MPU		SMPS_RA_ERR_MPU		SMPS_SA_ERR_MPU		RESERVED		VFMSM_TIMEOUT_ERR_IVA		VFMSM_RA_ERR_IVA		VFMSM_SA_ERR_IVA		SMPS_TIMEOUT_ERR_IVA		SMPS_RA_ERR_IVA		SMPS_SA_ERR_IVA		RESERVED		VFMSM_TIMEOUT_ERR_CORE		VFMSM_RA_ERR_CORE		VFMSM_SA_ERR_CORE		SMPS_TIMEOUT_ERR_CORE		SMPS_RA_ERR_CORE		SMPS_SA_ERR_CORE	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	BYPASS_TIMEOUT_ERR	Bypass command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost.  0x0: No error 0x1: An error has been logged	RW W1toClr	0
25	BYPASS_RA_ERR	Wrong register address error for bypass command  0x0: No error 0x1: An error has been logged	RW W1toClr	0
24	BYPASS_SA_ERR	Wrong slave address error for bypass command  0x0: No error 0x1: An error has been logged	RW W1toClr	0
23:22	RESERVED		R	0x0
21	VFMSM_TIMEOUT_ERR_MPU	MPU voltage FSM command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost.  0x0: No error 0x1: An error has been logged	RW W1toClr	0
20	VFMSM_RA_ERR_MPU	Wrong register address error for MPU voltage FSM  0x0: No error 0x1: An error has been logged	RW W1toClr	0
19	VFMSM_SA_ERR_MPU	Wrong slave address error for MPU voltage FSM  0x0: No error 0x1: An error has been logged	RW W1toClr	0
18	SMPS_TIMEOUT_ERR_MPU	MPU voltage processor command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost.  0x0: No error 0x1: An error has been logged	RW W1toClr	0
17	SMPS_RA_ERR_MPU	Wrong register address error for MPU voltage processor  0x0: No error 0x1: An error has been logged	RW W1toClr	0
16	SMPS_SA_ERR_MPU	Wrong slave address error for MPU voltage processor  0x0: No error 0x1: An error has been logged	RW W1toClr	0
15:14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13	VFSM_TIMEOUT_ERR_IVA	IVA voltage FSM command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW W1toClr	0
12	VFSM_RA_ERR_IVA	Wrong register address error for IVA voltage FSM 0x0: No error 0x1: An error has been logged	RW W1toClr	0
11	VFSM_SA_ERR_IVA	Wrong slave address error for IVA voltage FSM 0x0: No error 0x1: An error has been logged	RW W1toClr	0
10	SMPS_TIMEOUT_ERR_IVA	IVA voltage processor command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW W1toClr	0
9	SMPS_RA_ERR_IVA	Wrong register address error for IVA voltage processor 0x0: No error 0x1: An error has been logged	RW W1toClr	0
8	SMPS_SA_ERR_IVA	Wrong slave address error for IVA voltage processor 0x0: No error 0x1: An error has been logged	RW W1toClr	0
7:6	RESERVED		R	0x0
5	VFSM_TIMEOUT_ERR_CORE	CORE voltage FSM command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW W1toClr	0
4	VFSM_RA_ERR_CORE	Wrong register address error for CORE voltage FSM 0x0: No error 0x1: An error has been logged	RW W1toClr	0
3	VFSM_SA_ERR_CORE	Wrong slave address error for CORE voltage FSM 0x0: No error 0x1: An error has been logged	RW W1toClr	0
2	SMPS_TIMEOUT_ERR_CORE	CORE voltage processor command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW W1toClr	0
1	SMPS_RA_ERR_CORE	Wrong register address error for CORE voltage processor 0x0: No error 0x1: An error has been logged	RW W1toClr	0
0	SMPS_SA_ERR_CORE	Wrong slave address error for CORE voltage processor 0x0: No error 0x1: An error has been logged	RW W1toClr	0

**Table 3-874. Register Call Summary for Register PRM\_VC\_ERRST**

PRCM Register Manual

- [DEVICE\\_PRM Register Summary: \[0\]](#)

### 3.11.20 INSTR\_PRM Registers

#### 3.11.20.1 INSTR\_PRM Register Summary

**Table 3-875. INSTR\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	INSTR_PRM Base Address
PMI_IDENTIFICATION	R	32	0x0000 0000	0x4A30 7F00
PMI_SYS_CONFIG	RW	32	0x0000 0010	0x4A30 7F10
PMI_STATUS	R	32	0x0000 0014	0x4A30 7F14
PMI_CONFIGURATION	RW	32	0x0000 0024	0x4A30 7F24
PMI_CLASS_FILTERING	RW	32	0x0000 0028	0x4A30 7F28
PMI_TRIGGERING	RW	32	0x0000 002C	0x4A30 7F2C
PMI_SAMPLING	RW	32	0x0000 0030	0x4A30 7F30

#### 3.11.20.2 INSTR\_PRM Register Description

**Table 3-876. PMI\_IDENTIFICATION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-875</a>		
<b>Description</b>	PM profiling identification register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
REVISION																																

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	0x-- TI Internal data.

**Table 3-877. Register Call Summary for Register PMI\_IDENTIFICATION**

- PRCM Register Manual
- [INSTR\\_PRM Register Summary: \[0\]](#)

**Table 3-878. PMI\_SYS\_CONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-875</a>		
<b>Description</b>	PM profiling system configuration register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	IDLEMODE	RESERVED	SOFTRESET												

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0000000
5:4	RESERVED	Reserved	R	0x0
3:2	IDLEMODE	Configuration of the local target state management mode	RW	0x2
1	RESERVED	Reserved	R	0
0	SOFTRESET	Software reset	RW	0

**Table 3-879. Register Call Summary for Register PMI\_SYS\_CONFIG**

PRCM Register Manual

- [INSTR\\_PRM Register Summary: \[0\]](#)

**Table 3-880. PMI\_STATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-875</a>		
<b>Description</b>	PM profiling status register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FIFOEMPTY	RESERVED														

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0000000
8	FIFOEMPTY	PM Profiling buffer empty	R	1
7:0	RESERVED	Reserved	R	0x00

**Table 3-881. Register Call Summary for Register PMI\_STATUS**

PRCM Register Manual

- [INSTR\\_PRM Register Summary: \[0\]](#)

**Table 3-882. PMI\_CONFIGURATION**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-875</a>		
<b>Description</b>	PM profiling configuration register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLAIM_3	CLAIM_2	CLAIM_1	RESERVED				RESERVED	RESERVED				RESERVED	RESERVED				EVT_CAPT_EN	RESERVED													

Bits	Field Name	Description	Type	Reset
31:30	CLAIM_3	Ownership	RW	0x0
29	CLAIM_2	Debugger override qualifier	RW	1
28	CLAIM_1	Current owner	R	0

Bits	Field Name	Description	Type	Reset
27:24	RESERVED	Reserved	R	0x0
23	RESERVED	Reserved	R	0
22:16	RESERVED	Reserved	R	0x00
15	RESERVED	Reserved	R	0
14:8	RESERVED	Reserved	R	0x00
7	EVT_CAPT_EN	When HIGH the PM events capture is enabled	RW	0
6:0	RESERVED	Reserved	R	0x00

**Table 3-883. Register Call Summary for Register PMI\_CONFIGURATION**

PRCM Register Manual

- [INSTR\\_PRM Register Summary: \[0\]](#)

**Table 3-884. PMI\_CLASS\_FILTERING**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-875</a>		
<b>Description</b>	PM profiling class filtering register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SNAP_CAPT_EN_03				SNAP_CAPT_EN_02				SNAP_CAPT_EN_01				SNAP_CAPT_EN_00			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	SNAP_CAPT_EN_03	Snapshot capture enable - Class-ID = 0x03	RW	0
2	SNAP_CAPT_EN_02	Snapshot capture enable - Class-ID = 0x02	RW	0
1	SNAP_CAPT_EN_01	Snapshot capture enable - Class-ID = 0x01	RW	0
0	SNAP_CAPT_EN_00	Snapshot capture enable - Class-ID = 0x00	RW	0

**Table 3-885. Register Call Summary for Register PMI\_CLASS\_FILTERING**

PRCM Register Manual

- [INSTR\\_PRM Register Summary: \[0\]](#)

**Table 3-886. PMI\_TRIGGERING**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-875</a>		
<b>Description</b>	PM profiling triggering control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TRIG_STOP_EN		TRIG_START_EN													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1	TRIG_STOP_EN	Enable stop capturing PM events from external trigger detection	RW	0
0	TRIG_START_EN	Enable start capturing PM events from external trigger detection	RW	0

**Table 3-887. Register Call Summary for Register PMI\_TRIGGERING**

PRCM Register Manual

- [INSTR\\_PRM Register Summary: \[0\]](#)

**Table 3-888. PMI\_SAMPLING**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-875</a>		
<b>Description</b>	PM profiling sampling window register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												FCLK_DIV_FACOR				RESERVED								SAMP_WIND_SIZE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x000
19:16	FCLK_DIV_FACOR	FunClk divide factor ranging from 1 to 16	RW	0x0
15:8	RESERVED	Reserved	R	0x00
7:0	SAMP_WIND_SIZE	PM events sampling window size	RW	0x00

**Table 3-889. Register Call Summary for Register PMI\_SAMPLING**

PRCM Register Manual

- [INSTR\\_PRM Register Summary: \[0\]](#)

### 3.11.21 CM1 Instance Summary

**Table 3-890. CM1 Instance Summary**

Module Name	L4 Base Address	Size
INTRCONN_SOCKET_CM1	0x4A00 4000	256Bytes
CKGEN_CM1	0x4A00 4100	512Bytes
MPU_CM1	0x4A00 4300	256Bytes

**Table 3-890. CM1 Instance Summary (continued)**

Module Name	L4 Base Address	Size
DSP_CM1	0x4A00 4400	256Bytes
ABE_CM1	0x4A00 4500	256Bytes
RESTORE_CM1	0x4A00 4E00	256Bytes
INSTR_CM1	0x4A00 4F00	256Bytes

**3.11.22 INTRCONN\_SOCKET\_CM1 Registers**

**3.11.22.1 INTRCONN\_SOCKET\_CM1 Register Summary**

**Table 3-891. INTRCONN\_SOCKET\_CM1 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	INTRCONN_SOCKET_CM1 L4 Base Address
REVISION_CM1	R	32	0x0000 0000	0x4A00 4000
CM_CM1_PROFILING_CLKCTRL	RW	32	0x0000 0040	0x4A00 4040
CM1_DEBUG_CFG	RW	32	0x0000 00F0	0x4A00 40F0

**3.11.22.2 INTRCONN\_SOCKET\_CM1 Register Description**

**Table 3-892. REVISION\_CM1**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	INTRCONN_SOCKET_CM1
<b>Physical Address</b>	See <a href="#">Table 3-891</a>		
<b>Description</b>	This register contains the IP revision code for the CM1 part of the PRCM		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Revision Number	R	0x- - TI Internal data.

**Table 3-893. Register Call Summary for Register REVISION\_CM1**

- PRCM Register Manual
- [INTRCONN\\_SOCKET\\_CM1 Register Summary: \[0\]](#)

**Table 3-894. CM\_CM1\_PROFILING\_CLKCTRL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	INTRCONN_SOCKET_CM1
<b>Physical Address</b>	See <a href="#">Table 3-891</a>		
<b>Description</b>	This register manages the CM1_PROFILING clock. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								MODULEMODE														

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status Read 0x0: Module is fully functional Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle Read 0x3: Module is disabled	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by software. INTRCONN configuration port is not accessible. 0x1: Module is managed automatically by hardware along with CM1 and EMU domain. INTRCONN configuration port is accessible only when EMU domain is on. Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-895. Register Call Summary for Register CM\_CM1\_PROFILING\_CLKCTRL**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_CM1 Register Summary: \[0\]](#)
- [RESTORE\\_CM1 Register Description: \[1\]](#)

**Table 3-896. CM1\_DEBUG\_CFG**

<b>Address Offset</b>	0x0000 00F0
<b>Physical Address</b>	See <a href="#">Table 3-891</a>
<b>Description</b>	This register is used to configure the CM1's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. [warm reset insensitive]
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SEL3				RESERVED				SEL2				RESERVED				SEL1				RESERVED				SEL0			

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30:24	SEL3	Internal signal block select for debug word byte-3	RW	0x03
23	RESERVED		R	0
22:16	SEL2	Internal signal block select for debug word byte-2	RW	0x02
15	RESERVED		R	0
14:8	SEL1	Internal signal block select for debug word byte-1	RW	0x01
7	RESERVED		R	0



Bits	Field Name	Description	Type	Reset
6:0	SELO	Internal signal block select for debug word byte-0	RW	0x00

**Table 3-897. Register Call Summary for Register CM1\_DEBUG\_CFG**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_CM1 Register Summary: \[0\]](#)

### 3.11.23 CKGEN\_CM1 Registers

#### 3.11.23.1 CKGEN\_CM1 Register Summary

**Table 3-898. CKGEN\_CM1 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_CM1 L4 Base Address
CM_CLKSEL_CORE	RW	32	0x0000 0000	0x4A00 4100
CM_CLKSEL_ABE	RW	32	0x0000 0008	0x4A00 4108
CM_DLL_CTRL	RW	32	0x0000 0010	0x4A00 4110
CM_CLKMODE_DPLL_CORE	RW	32	0x0000 0020	0x4A00 4120
CM_IDLEST_DPLL_CORE	R	32	0x0000 0024	0x4A00 4124
CM_AUTOIDLE_DPLL_CORE	RW	32	0x0000 0028	0x4A00 4128
CM_CLKSEL_DPLL_CORE	RW	32	0x0000 002C	0x4A00 412C
CM_DIV_M2_DPLL_CORE	RW	32	0x0000 0030	0x4A00 4130
CM_DIV_M3_DPLL_CORE	RW	32	0x0000 0034	0x4A00 4134
CM_DIV_M4_DPLL_CORE	RW	32	0x0000 0038	0x4A00 4138
CM_DIV_M5_DPLL_CORE	RW	32	0x0000 003C	0x4A00 413C
CM_DIV_M6_DPLL_CORE	RW	32	0x0000 0040	0x4A00 4140
CM_DIV_M7_DPLL_CORE	RW	32	0x0000 0044	0x4A00 4144
CM_SSC_DELTAMSTEP_DPLL_CORE	RW	32	0x0000 0048	0x4A00 4148
CM_SSC_MODFREQDIV_DPLL_CORE	RW	32	0x0000 004C	0x4A00 414C
CM_EMU_OVERRIDE_DPLL_CORE	RW	32	0x0000 0050	0x4A00 4150
CM_CLKMODE_DPLL_MPU	RW	32	0x0000 0060	0x4A00 4160
CM_IDLEST_DPLL_MPU	R	32	0x0000 0064	0x4A00 4164
CM_AUTOIDLE_DPLL_MPU	RW	32	0x0000 0068	0x4A00 4168
CM_CLKSEL_DPLL_MPU	RW	32	0x0000 006C	0x4A00 416C
CM_DIV_M2_DPLL_MPU	RW	32	0x0000 0070	0x4A00 4170
CM_SSC_DELTAMSTEP_DPLL_MPU	RW	32	0x0000 0088	0x4A00 4188
CM_SSC_MODFREQDIV_DPLL_MPU	RW	32	0x0000 008C	0x4A00 418C
CM_BYPCLK_DPLL_MPU	RW	32	0x0000 009C	0x4A00 419C
CM_CLKMODE_DPLL_IVA	RW	32	0x0000 00A0	0x4A00 41A0
CM_IDLEST_DPLL_IVA	R	32	0x0000 00A4	0x4A00 41A4
CM_AUTOIDLE_DPLL_IVA	RW	32	0x0000 00A8	0x4A00 41A8
CM_CLKSEL_DPLL_IVA	RW	32	0x0000 00AC	0x4A00 41AC
CM_DIV_M4_DPLL_IVA	RW	32	0x0000 00B8	0x4A00 41B8
CM_DIV_M5_DPLL_IVA	RW	32	0x0000 00BC	0x4A00 41BC
CM_SSC_DELTAMSTEP_DPLL_IVA	RW	32	0x0000 00C8	0x4A00 41C8
CM_SSC_MODFREQDIV_DPLL_IVA	RW	32	0x0000 00CC	0x4A00 41CC
CM_BYPCLK_DPLL_IVA	RW	32	0x0000 00DC	0x4A00 41DC
CM_CLKMODE_DPLL_ABE	RW	32	0x0000 00E0	0x4A00 41E0
CM_IDLEST_DPLL_ABE	R	32	0x0000 00E4	0x4A00 41E4
CM_AUTOIDLE_DPLL_ABE	RW	32	0x0000 00E8	0x4A00 41E8

**Table 3-898. CKGEN\_CM1 Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_CM1 L4 Base Address
<a href="#">CM_CLKSEL_DPLL_ABE</a>	RW	32	0x0000 00EC	0x4A00 41EC
<a href="#">CM_DIV_M2_DPLL_ABE</a>	RW	32	0x0000 00F0	0x4A00 41F0
<a href="#">CM_DIV_M3_DPLL_ABE</a>	RW	32	0x0000 00F4	0x4A00 41F4
<a href="#">CM_SSC_DELTAMSTEP_DPLL_ABE</a>	RW	32	0x0000 0108	0x4A00 4208
<a href="#">CM_SSC_MODFREQDIV_DPLL_ABE</a>	RW	32	0x0000 010C	0x4A00 420C
RESERVED	RW	32	0x0000 0120	0x4A00 4220
RESERVED	R	32	0x0000 0124	0x4A00 4224
RESERVED	RW	32	0x0000 0128	0x4A00 4228
RESERVED	RW	32	0x0000 012C	0x4A00 422C
RESERVED	RW	32	0x0000 0130	0x4A00 4230
RESERVED	RW	32	0x0000 0138	0x4A00 4238
RESERVED	RW	32	0x0000 013C	0x4A00 423C
RESERVED	RW	32	0x0000 0140	0x4A00 4240
RESERVED	RW	32	0x0000 0148	0x4A00 4248
RESERVED	RW	32	0x0000 014C	0x4A00 424C
<a href="#">CM_SHADOW_FREQ_CONFIG1</a>	RW	32	0x0000 0160	0x4A00 4260
<a href="#">CM_SHADOW_FREQ_CONFIG2</a>	RW	32	0x0000 0164	0x4A00 4264
<a href="#">CM_DYN_DEP_PRESCAL</a>	RW	32	0x0000 0170	0x4A00 4270
<a href="#">CM_RESTORE_ST</a>	RW	32	0x0000 0180	0x4A00 4280

**3.11.23.2 CKGEN\_CM1 Register Description****Table 3-899. CM\_CLKSEL\_CORE**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	CORE module clock selection.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKSEL_L4	RESERVED		CLKSEL_L3	RESERVED		CLKSEL_CORE									

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKSEL_L4	Selects L4 interconnect clock (L4_clk) 0x0: L4_CLK is L3_CLK divided by 1 0x1: L4_CLK is L3_CLK divided by 2	RW	0
7:5	RESERVED		R	0x0
4	CLKSEL_L3	Selects L3 interconnect clock (L3_clk) 0x0: L3_CLK is CORE_CLK divided by 1 0x1: L3_CLK is CORE_CLK divided by 2	RW	0
3:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	CLKSEL_CORE	Selects CORE_CLK configuration 0x0: CORE_CLK is CORE_X2_CLK divided by 1 0x1: CORE_CLK is CORE_X2_CLK divided by 2	RW	0

**Table 3-900. Register Call Summary for Register CM\_CLKSEL\_CORE**

Clock Management Functional Description

- [CM Clock Source: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[3\]](#)
- [CKGEN\\_CM1 Register Description: \[4\] \[5\]](#)
- [RESTORE\\_CM1 Register Description: \[6\] \[7\] \[8\]](#)

**Table 3-901. CM\_CLKSEL\_ABE**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	ABE module clock selection.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SLIMBUS_CLK_GATE		RESERVED	PAD_CLKS_GATE		RESERVED					CLKSEL_OPP					

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10	SLIMBUS_CLK_GATE	Gating control for SLIMBUS_CLK clock tree in ABE. SLIMBUS module always gets the ungated version. 0x0: The clock is gated 0x1: The clock is enabled	RW	0
9	RESERVED		R	0
8	PAD_CLKS_GATE	Gating control for PAD_CLKS clock tree in ABE 0x0: The clock is gated 0x1: The clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	CLKSEL_OPP	Selects the OPP divider ABE domain 0x0: ABE_CLK is divide by 1 of DPLL_ABE_X2_CLK 0x1: ABE_CLK is divide by 2 of DPLL_ABE_X2_CLK 0x2: ABE_CLK is divide by 4 of DPLL_ABE_X2_CLK 0x3: Reserved	RW	0x0

**Table 3-902. Register Call Summary for Register CM\_CLKSEL\_ABE**

Clock Management Functional Description

- [CM Clock Source: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[3\]](#)

**Table 3-903. CM\_DLL\_CTRL**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	Special register for DLL control		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												DLL_OVERRIDE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	DLL_OVERRIDE	Control if DLL lock and code outputs are overridden or not 0x0: Lock and code outputs are not overridden 0x1: Lock output is overridden to 1 and code output is overridden with a value coming from control module.	RW	1

**Table 3-904. Register Call Summary for Register CM\_DLL\_CTRL**

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[0\]](#)
- [CKGEN\\_CM1 Register Description: \[1\]](#)
- [RESTORE\\_CM1 Register Description: \[2\]](#)

**Table 3-905. CM\_CLKMODE\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0020		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register allows controlling the DPLL modes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LP_MODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED	DPLL_EN							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0

Bits	Field Name	Description	Type	Reset
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature Read 0x0: SSC has been turned off on PLL o/ps Read 0x1: SSC has been turned on on PLL o/ps	R	0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. Read 0x0: REGM4XEN mode of the DPLL is disabled	R	0
10	DPLL_LPmode_EN	Set the DPLL in low-power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low-power mode of the DPLL is disabled 0x1: Low-power mode of the DPLL is enabled	RW	0
9	RESERVED		RW	0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0
7:3	RESERVED		RW	0x00
2:0	DPLL_EN	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN bypass mode. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Put the DPLL in MN bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 0x5: Put the DPLL in Idle bypass low-power mode. 0x6: Put the DPLL in Idle bypass fast-relock mode. 0x7: Enables the DPLL in lock mode	RW	0x4

**Table 3-906. Register Call Summary for Register CM\_CLKMODE\_DPLL\_CORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\] \[1\] \[2\] \[5\] \[6\] \[7\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[8\]](#)
- [CKGEN\\_CM1 Register Description: \[9\]](#)
- [RESTORE\\_CM1 Register Description: \[10\] \[11\]](#)

**Table 3-907. CM\_IDLEST\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0024		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_MN_BYPASS	RESERVED							ST_DPLL_CLK							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	ST_MN_BYPASS	DPLL MN_BYPASS status Read 0x1: DPLL is in MN_Bypass Read 0x0: DPLL is not in MN_Bypass	R	0
7:1	RESERVED		R	0x00
0	ST_DPLL_CLK	DPLL lock status Read 0x1: DPLL is LOCKED Read 0x0: DPLL is either in bypass mode or in stop mode.	R	0

**Table 3-908. Register Call Summary for Register CM\_IDLEST\_DPLL\_CORE**

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[0\]](#)

**Table 3-909. CM\_AUTOIDLE\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register provides automatic control over the DPLL activity.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											AUTO_DPLL_MODE				

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000000
2:0	AUTO_DPLL_MODE	<p>DPLL automatic control.</p> <p>0x0: DPLL auto control disabled</p> <p>0x1: The DPLL is automatically put in low-power stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x2: The DPLL is automatically put in fast-relock stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x3: Reserved</p> <p>0x4: Reserved</p> <p>0x5: The DPLL is automatically put in idle bypass low-power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x6: The DPLL is automatically put in idle bypass fast-relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x7: Reserved</p>	RW	0x0

**Table 3-910. Register Call Summary for Register CM\_AUTOIDLE\_DPLL\_CORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)
- [RESTORE\\_CM1 Register Description: \[2\]](#)

**Table 3-911. CM\_CLKSEL\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	RESERVED	DPLL_CLKOUTHIF_CLKSEL	RESERVED	DPLL_MULT								RESERVED	DPLL_DIV										

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL locked mode, 0 - No impact 1 - No impact In DPLL bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0
22:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20	DPLL_CLKOUTHIF_CLKSEL	Selects the source of the DPLL CLKOUTHIF clock. Same as CLKINPHIFSEL pin on the DPLL  0x0: CLKOUTHIF is generated from the DPLL oscillator (DCO)  0x1: CLKOUTHIF is generated from CLKINPHIF	RW	0
19	RESERVED		R	0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN bypass mode. (equal to input M of DPLL; M=2 to 2047 => DPLL multiplies by M).  0x0: Reserved  0x1: Reserved	RW	0x000
7	RESERVED		R	0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).	RW	0x00

**Table 3-912. Register Call Summary for Register CM\_CLKSEL\_DPLL\_CORE**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [DPLL\\_CORE Description: \[1\] \[2\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[3\]](#)
- [RESTORE\\_CM1 Register Description: \[4\]](#)

**Table 3-913. CM\_DIV\_M2\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the M2 divider of the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKOUT		DPLL_CLKOUT_GATE_CTRL		RESERVED		DPLL_CLKOUT_DIVCHACK		DPLL_CLKOUT_DIV							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0000000
9	ST_DPLL_CLKOUT	DPLL CLKOUT status  Read 0x0: The clock output is gated  Read 0x1: The clock output is enabled	R	0
8	DPLL_CLKOUT_GATE_CTRL	Control gating of DPLL CLKOUT  0x0: Automatically gate this clock when there is no dependency for it  0x1: Force this clock to stay enabled even if there is no request	RW	0



Bits	Field Name	Description	Type	Reset
7:6	RESERVED		R	0x0
5	DPLL_CLKOUT_DIVCHACK	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect	R	0
4:0	DPLL_CLKOUT_DIV	DPLL post-divider factor, M2, for internal clock generation (1 to 31); Divide value from 1 to 31. 0x0: Reserved	RW	0x01

**Table 3-914. Register Call Summary for Register CM\_DIV\_M2\_DPLL\_CORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[4\]](#)
- [CKGEN\\_CM1 Register Description: \[5\]](#)
- [RESTORE\\_CM1 Register Description: \[6\] \[7\]](#)

**Table 3-915. CM\_DIV\_M3\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the M3 divider of the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKOUTHIF	DPLL_CLKOUTHIF_GATE_CTRL	RESERVED	DPLL_CLKOUTHIF_DIVCHACK	DPLL_CLKOUTHIF_DIV											

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0000000
9	ST_DPLL_CLKOUTHIF	DPLL CLKOUTHIF status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	DPLL_CLKOUTHIF_GATE_CTRL	Control gating of DPLL CLKOUTHIF 0x0: Gate unconditionally this clock and stop requesting it 0x1: Request this clock and un-gate it when available	RW	0
7:6	RESERVED		R	0x0
5	DPLL_CLKOUTHIF_DIVCHACK	Toggle on this status bit after changing DPLL_CLKOUTHIF_DIV indicates that the change in divider value has taken effect	R	0
4:0	DPLL_CLKOUTHIF_DIV	DPLL post-divider factor, M3, for internal clock generation (1 to 31); Divide value from 1 to 31. 0x0: Reserved	RW	0x01

**Table 3-916. Register Call Summary for Register CM\_DIV\_M3\_DPLL\_CORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[4\]](#)
- [RESTORE\\_CM1 Register Description: \[5\]](#)

**Table 3-917. CM\_DIV\_M4\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT1_PWDN	RESERVED	ST_HSDIVIDER_CLKOUT1	HSDIVIDER_CLKOUT1_GATE_CTRL	RESERVED	HSDIVIDER_CLKOUT1_DIVCHACK	HSDIVIDER_CLKOUT1_DIV									

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x000000
12	HSDIVIDER_CLKOUT1_PWDN	Automatic power down for HSDIVIDER M4 divider and hence CLKOUT1 output when the o/p clock is gated. Read 0x0: Keep M4 divider powered on even when CLKOUT1 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT1	HSDIVIDER CLKOUT1 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT1_GATE_CTRL	Control gating of HSDIVIDER CLKOUT1 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT1_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT1_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT1_DIV	DPLL M4 post-divider factor (1 to 31). 0x0: Reserved	RW	0x04

**Table 3-918. Register Call Summary for Register CM\_DIV\_M4\_DPLL\_CORE**

- Clock Management Functional Description
- [DPLL\\_CORE Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- PRCM Register Manual
- [CKGEN\\_CM1 Register Summary: \[5\]](#)
  - [RESTORE\\_CM1 Register Description: \[6\]](#)

**Table 3-919. CM\_DIV\_M5\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT2_PWDN	RESERVED	ST_HSDIVIDER_CLKOUT2	HSDIVIDER_CLKOUT2_GATE_CTRL	RESERVED	HSDIVIDER_CLKOUT2_DIVCHACK	HSDIVIDER_CLKOUT2_DIV									

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x000000
12	HSDIVIDER_CLKOUT2_PWDN	Automatic power down for HSDIVIDER M5 divider and hence CLKOUT2 output when the o/p clock is gated. Read 0x0: Keep M5 divider powered on even when CLKOUT2 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT2	HSDIVIDER CLKOUT2 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT2_GATE_CTRL	Control gating of HSDIVIDER CLKOUT2 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT2_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT2_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT2_DIV	DPLL M5 post-divider factor (1 to 31). 0x0: Reserved	RW	0x04

**Table 3-920. Register Call Summary for Register CM\_DIV\_M5\_DPLL\_CORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[5\]](#)
- [CKGEN\\_CM1 Register Description: \[6\]](#)
- [RESTORE\\_CM1 Register Description: \[7\] \[8\]](#)

**Table 3-921. CM\_DIV\_M6\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the CLKOUT3 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT3_PWDN	RESERVED	ST_HSDIVIDER_CLKOUT3	HSDIVIDER_CLKOUT3_GATE_CTRL	RESERVED	HSDIVIDER_CLKOUT3_DIVCHACK	HSDIVIDER_CLKOUT3_DIV									

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT3_PWDN	Automatic power down for HSDIVIDER M6 divider and hence CLKOUT3 output when the o/p clock is gated. Read 0x0: Keep M6 divider powered on even when CLKOUT3 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT3	HSDIVIDER CLKOUT3 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT3_GATE_CTRL	Control gating of HSDIVIDER CLKOUT3 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT3_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT3_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT3_DIV	DPLL M6 post-divider factor (1 to 31). 0x0: Reserved	RW	0x08

**Table 3-922. Register Call Summary for Register CM\_DIV\_M6\_DPLL\_CORE**

- Clock Management Functional Description
- [DPLL\\_CORE Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- PRCM Register Manual
- [CKGEN\\_CM1 Register Summary: \[5\]](#)
  - [RESTORE\\_CM1 Register Description: \[6\]](#)

**Table 3-923. CM\_DIV\_M7\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the CLKOUT4 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT4_PWDN	RESERVED	ST_HSDIVIDER_CLKOUT4	HSDIVIDER_CLKOUT4_GATE_CTRL	RESERVED	HSDIVIDER_CLKOUT4_DIVCHACK	HSDIVIDER_CLKOUT4_DIV									

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT4_PWDN	Automatic power down for HSDIVIDER M7 divider and hence CLKOUT4 output when the o/p clock is gated. Read 0x0: Keep M7 divider powered on even when CLKOUT4 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT4	HSDIVIDER CLKOUT4 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT4_GATE_CTRL	Control gating of HSDIVIDER CLKOUT4 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT4_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT4_DIV indicates that the change in divider value has taken effect <sup>(1)</sup>	R	0
4:0	HSDIVIDER_CLKOUT4_DIV	DPLL M7 post-divider factor (1 to 31). 0x0: Reserved	RW	0x04

<sup>(1)</sup> It is toggled only if the programmed divider value differs from the previous one.

**Table 3-924. Register Call Summary for Register CM\_DIV\_M7\_DPLL\_CORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[5\]](#)
- [RESTORE\\_CM1 Register Description: \[6\]](#)

**Table 3-925. CM\_SSC\_DELTAMSTEP\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												DELTAMSTEP																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x00000

**Table 3-926. Register Call Summary for Register CM\_SSC\_DELTAMSTEP\_DPLL\_CORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)
- [RESTORE\\_CM1 Register Description: \[2\]](#)

**Table 3-927. CM\_SSC\_MODFREQDIV\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	Control the Modulation Frequency (Fm) for Spread Spectrum. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED												MODFREQDIV_MANTISSA																				
												MODFREQDIV_EXPONENT				RESERVED																

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x00

**Table 3-928. Register Call Summary for Register CM\_SSC\_MODFREQDIV\_DPLL\_CORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[2\]](#)
- [RESTORE\\_CM1 Register Description: \[3\]](#)

**Table 3-929. CM\_EMU\_OVERRIDE\_DPLL\_CORE**

<b>Address Offset</b>	0x0000 0050
<b>Physical Address</b>	See <a href="#">Table 3-898</a>
<b>Description</b>	This register provides emulation override controls over the CORE DPLL.
<b>Type</b>	RW
<b>Instance</b>	CKGEN_CM1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OVERRIDE_ENABLE	CORE_DPLL_EMU_MULT								RESERVED	CORE_DPLL_EMU_DIV													

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19	OVERRIDE_ENABLE	This bit allows to enable or disable the emulation override controls 0x0: The emulation override controls are disabled 0x1: The emulation override controls are enabled	RW	0
18:8	CORE_DPLL_EMU_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 => DPLL multiplies by M)	RW	0x000
7	RESERVED		R	0
6:0	CORE_DPLL_EMU_DIV	CORE DPLL override divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).	RW	0x00

**Table 3-930. Register Call Summary for Register CM\_EMU\_OVERRIDE\_DPLL\_CORE**

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[0\]](#)

**Table 3-931. CM\_CLKMODE\_DPLL\_MPU**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register allows controlling the DPLL modes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED					DPLL_EN			

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature Read 0x0: SSC has been turned off on PLL o/ps Read 0x1: SSC has been turned on on PLL o/ps	R	0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. Read 0x0: REGM4XEN mode of the DPLL is disabled	R	0
10	DPLL_LPMODE_EN	Set the DPLL in low-power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low-power mode of the DPLL is disabled 0x1: Low-power mode of the DPLL is enabled	RW	0
9	RESERVED		RW	0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0
7:3	RESERVED		RW	0x00



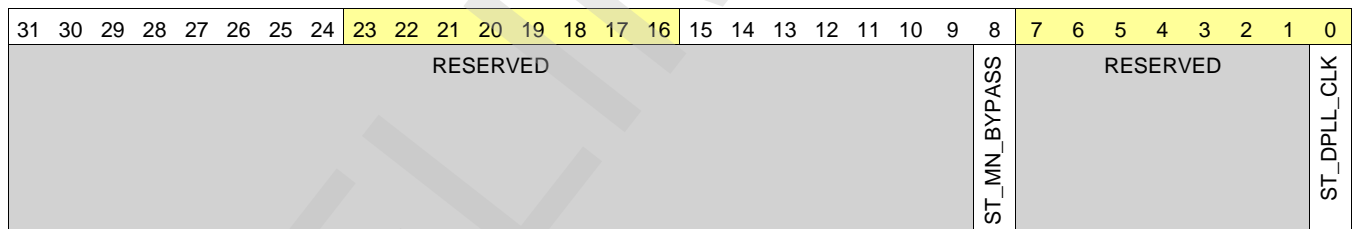
Bits	Field Name	Description	Type	Reset
2:0	DPLL_EN	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN bypass mode.  0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Put the DPLL in MN bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 0x5: Put the DPLL in idle bypass low-power mode. 0x6: Put the DPLL in idle bypass fast-relock mode. 0x7: Enables the DPLL in lock mode	RW	0x4

**Table 3-932. Register Call Summary for Register CM\_CLKMODE\_DPLL\_MPU**

- Clock Management Functional Description
- [DPLL\\_MPU Description: \[0\] \[1\] \[2\] \[5\] \[6\] \[7\]](#)
- PRCM Register Manual
- [CKGEN\\_CM1 Register Summary: \[8\]](#)

**Table 3-933. CM\_IDLEST\_DPLL\_MPU**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
<b>Type</b>	R		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	ST_MN_BYPASS	DPLL MN_BYPASS status Read 0x1: DPLL is in MN_Bypass Read 0x0: DPLL is not in MN_Bypass	R	0
7:1	RESERVED		R	0x00
0	ST_DPLL_CLK	DPLL lock status Read 0x1: DPLL is LOCKED Read 0x0: DPLL is either in bypass mode or in stop mode.	R	0

**Table 3-934. Register Call Summary for Register CM\_IDLEST\_DPLL\_MPU**

- PRCM Register Manual
- [CKGEN\\_CM1 Register Summary: \[0\]](#)

**Table 3-935. CM\_AUTOIDLE\_DPLL\_MPU**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register provides automatic control over the DPLL activity.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AUTO_DPLL_MODE															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in low-power stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in fast-relock stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in idle bypass low-power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in idle bypass fast-relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

**Table 3-936. Register Call Summary for Register CM\_AUTOIDLE\_DPLL\_MPU**

Clock Management Functional Description

- [DPLL\\_MPU Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)

**Table 3-937. CM\_CLKSEL\_DPLL\_MPU**

<b>Address Offset</b>	0x0000 006C		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register provides controls over the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCC_COUNT_MAX								DPLL_BYP_CLKSEL	DCC_EN				DPLL_MULT							RESERVED	DPLL_DIV										

Bits	Field Name	Description	Type	Reset
31:24	DCC_COUNT_MAX	The value "NbCycles" set in this field determines the duration of the clock ramp step during which the output frequency is $F_{dpll}/(2 \cdot M^2)$ . The duration is computed as $32 \times \text{NbCycles}$ of L4 clock cycles (100 MHz). Duration should be $> 1.5 \mu\text{s}$ to allow enough time for DCC to lock. This bit field is relevant only when DCC_EN = 1.	RW	0x05
23	DPLL_BYP_CLKSEL	Only CLKINPULOW bypass clock supported for this PLL	R Returns 1s	1
22:19	DCC_EN	Enable or disable duty cycle correction. Must be enabled only for frequency $> 1 \text{ GHz}$ .  When enabled, the CLKOUTHIF output of the DPLL is used after duty cycle correction instead of CLKOUT. The M3 divider is hard-wired to 1 so the lock frequency $F_{dpll}$ is directly provided to MPU.  0x0: DCC disabled 0x1: DCC enabled	RW	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN bypass mode. (equal to input M of DPLL; $M=2$ to 2047 $\Rightarrow$ DPLL multiplies by M).  0x0: Reserved 0x1: Reserved	RW	0x000
7	RESERVED		R	0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is $N+1$ ).	RW	0x00

**Table 3-938. Register Call Summary for Register CM\_CLKSEL\_DPLL\_MPU**

Clock Management Functional Description

- [DPLL\\_MPU Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[2\]](#)

**Table 3-939. CM\_DIV\_M2\_DPLL\_MPU**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the M2 divider of the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKOUT		DPLL_CLKOUT_GATE_CTRL		RESERVED		DPLL_CLKOUT_DIVCHECK		DPLL_CLKOUT_DIV							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	ST_DPLL_CLKOUT	DPLL CLKOUT status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	DPLL_CLKOUT_GATE_CTRL	Control gating of DPLL CLKOUT 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	DPLL_CLKOUT_DIVCHECK	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect	R	0
4:0	DPLL_CLKOUT_DIV	DPLL post-divider factor, M2, for internal clock generation (1 to 31); Divide value from 1 to 31. 0x0: Reserved	RW	0x01

**Table 3-940. Register Call Summary for Register CM\_DIV\_M2\_DPLL\_MPU**

Clock Management Functional Description

- [DPLL\\_MPU Description: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[4\]](#)

**Table 3-941. CM\_SSC\_DELTAMSTEP\_DPLL\_MPU**

<b>Address Offset</b>	0x0000 0088	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												DELTAMSTEP																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x00000

**Table 3-942. Register Call Summary for Register CM\_SSC\_DELTAMSTEP\_DPLL\_MPU**

Clock Management Functional Description

- [DPLL\\_MPU Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)

**Table 3-943. CM\_SSC\_MODFREQDIV\_DPLL\_MPU**

<b>Address Offset</b>	0x0000 008C	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	Control the Modulation Frequency (Fm) for Spread Spectrum. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODFREQDIV_EXPONENT	RESERVED	MODFREQDIV_MANTISSA													

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x00

**Table 3-944. Register Call Summary for Register CM\_SSC\_MODFREQDIV\_DPLL\_MPU**

Clock Management Functional Description

- [DPLL\\_MPU Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[2\]](#)

**Table 3-945. CM\_BYPCLOCK\_DPLL\_MPU**

<b>Address Offset</b>	0x0000 009C	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	Control MPU PLL BYPASS clock. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKSEL															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	CLKSEL	Select the DPLL MPU bypass clock 0x0: DPLL_MPU bypass clock is CORE_X2_CLK divided by 1 0x1: DPLL_MPU bypass clock is CORE_X2_CLK divided by 2 0x2: DPLL_MPU bypass clock is CORE_X2_CLK divided by 4 0x3: DPLL_MPU bypass clock is CORE_X2_CLK divided by 8	RW	0x0

**Table 3-946. Register Call Summary for Register CM\_BYPCLK\_DPLL\_MPU**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)

**Table 3-947. CM\_CLKMODE\_DPLL\_IVA**

<b>Address Offset</b>	0x0000 00A0	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register allows controlling the DPLL modes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LP_MODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED				DPLL_EN				

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x000000
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature Read 0x0: SSC has been turned off on PLL o/ps Read 0x1: SSC has been turned on on PLL o/ps	R	0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. Read 0x0: REGM4XEN mode of the DPLL is disabled	R	0

Bits	Field Name	Description	Type	Reset
10	DPLL_LPMODE_EN	Set the DPLL in low-power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low-power mode of the DPLL is disabled 0x1: Low-power mode of the DPLL is enabled	RW	0
9	RESERVED		RW	0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0
7:3	RESERVED		RW	0x00
2:0	DPLL_EN	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN bypass mode. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Put the DPLL in MN bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 0x5: Put the DPLL in idle bypass low-power mode. 0x6: Reserved 0x7: Enables the DPLL in lock mode	RW	0x4

**Table 3-948. Register Call Summary for Register CM\_CLKMODE\_DPLL\_IVA**

Clock Management Functional Description

- [DPLL\\_IVA Description: \[0\] \[1\] \[2\] \[5\] \[6\] \[7\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[8\]](#)

**Table 3-949. CM\_IDLEST\_DPLL\_IVA**

<b>Address Offset</b>	0x0000 00A4	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_MN_BYPASS	RESERVED								ST_DPLL_CLK						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0000000
8	ST_MN_BYPASS	DPLL MN_BYPASS status Read 0x1: DPLL is in MN_Bypass Read 0x0: DPLL is not in MN_Bypass	R	0

Bits	Field Name	Description	Type	Reset
7:1	RESERVED		R	0x00
0	ST_DPLL_CLK	DPLL lock status Read 0x1: DPLL is LOCKED Read 0x0: DPLL is either in bypass mode or in stop mode.	R	0

**Table 3-950. Register Call Summary for Register CM\_IDLEST\_DPLL\_IVA**

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[0\]](#)

**Table 3-951. CM\_AUTOIDLE\_DPLL\_IVA**

<b>Address Offset</b>	0x0000 00A8	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides automatic control over the DPLL activity.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AUTO_DPLL_MODE															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000000
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in low-power stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in fast-relock stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in idle bypass low-power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: Reserved 0x7: Reserved	RW	0x0

**Table 3-952. Register Call Summary for Register CM\_AUTOIDLE\_DPLL\_IVA**

Clock Management Functional Description

- [DPLL\\_IVA Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)



**Table 3-953. CM\_CLKSEL\_DPLL\_IVA**

<b>Address Offset</b>	0x0000 00AC		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register provides controls over the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	RESERVED				DPLL_MULT								RESERVED	DPLL_DIV									

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL locked mode, 0 - No impact 1 - No impact In DPLL bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0
22:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN bypass mode. (equal to input M of DPLL; M=2 to 2047 => DPLL multiplies by M). 0x0: Reserved 0x1: Reserved	RW	0x000
7	RESERVED		R	0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).	RW	0x00

**Table 3-954. Register Call Summary for Register CM\_CLKSEL\_DPLL\_IVA**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [DPLL\\_IVA Description: \[1\] \[2\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[3\]](#)

**Table 3-955. CM\_DIV\_M4\_DPLL\_IVA**

<b>Address Offset</b>	0x0000 00B8		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT1_PWDN	RESERVED	ST_HSDIVIDER_CLKOUT1	HSDIVIDER_CLKOUT1_GATE_CTRL	RESERVED	HSDIVIDER_CLKOUT1_DIVCHACK	HSDIVIDER_CLKOUT1_DIV									

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT1_PWDN	Automatic power down for HSDIVIDER M4 divider and hence CLKOUT1 output when the o/p clock is gated.  Read 0x0: Keep M4 divider powered on even when CLKOUT1 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT1	HSDIVIDER CLKOUT1 status  Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT1_GATE_CTRL	Control gating of HSDIVIDER CLKOUT1  0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT1_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT1_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT1_DIV	DPLL M4 post-divider factor (1 to 31).  0x00: Reserved	RW	0x04

**Table 3-956. Register Call Summary for Register CM\_DIV\_M4\_DPLL\_IVA**

Clock Management Functional Description

- [DPLL\\_IVA Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[5\]](#)

**Table 3-957. CM\_DIV\_M5\_DPLL\_IVA**

<b>Address Offset</b>	0x0000 00BC		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT2_PWDN	RESERVED		ST_HSDIVIDER_CLKOUT2	HSDIVIDER_CLKOUT2_GATE_CTRL		RESERVED		HSDIVIDER_CLKOUT2_DIVCHACK	HSDIVIDER_CLKOUT2_DIV						

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT2_PWDN	Automatic power down for HSDIVIDER M5 divider and hence CLKOUT2 output when the o/p clock is gated. Read 0x0: Keep M5 divider powered on even when CLKOUT2 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT2	HSDIVIDER CLKOUT2 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT2_GATE_CTRL	Control gating of HSDIVIDER CLKOUT2 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT2_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT2_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT2_DIV	DPLL M5 post-divider factor (1 to 31). 0x00: Reserved	RW	0x04

**Table 3-958. Register Call Summary for Register CM\_DIV\_M5\_DPLL\_IVA**

Clock Management Functional Description

- [DPLL\\_IVA Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[5\]](#)

**Table 3-959. CM\_SSC\_DELTAMSTEP\_DPLL\_IVA**

<b>Address Offset</b>	0x0000 00C8		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELTAMSTEP																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x00000

**Table 3-960. Register Call Summary for Register CM\_SSC\_DELTAMSTEP\_DPLL\_IVA**

Clock Management Functional Description

- [DPLL\\_IVA Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)

**Table 3-961. CM\_SSC\_MODFREQDIV\_DPLL\_IVA**

<b>Address Offset</b>	0x0000 00CC		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	Control the Modulation Frequency (Fm) for Spread Spectrum. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											MODFREQDIV_EXPONENT		RESERVED	MODFREQDIV_MANTISSA																	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x00

**Table 3-962. Register Call Summary for Register CM\_SSC\_MODFREQDIV\_DPLL\_IVA**

Clock Management Functional Description

- [DPLL\\_IVA Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[2\]](#)

**Table 3-963. CM\_BYPCCLK\_DPLL\_IVA**

<b>Address Offset</b>	0x0000 00DC		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	Control IVA PLL BYPASS clock. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKSEL															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	CLKSEL	Select the DPLL IVA bypass clock 0x0: DPLL_IVA bypass clock is CORE_X2_CLK divided by 1 0x1: DPLL_IVA bypass clock is CORE_X2_CLK divided by 2 0x2: DPLL_IVA bypass clock is CORE_X2_CLK divided by 4 0x3: DPLL_IVA bypass clock is CORE_X2_CLK divided by 8	RW	0x0

**Table 3-964. Register Call Summary for Register CM\_BYPCCLK\_DPLL\_IVA**

- Clock Management Functional Description
- [CM Clock Source: \[0\]](#)
- PRCM Register Manual
- [CKGEN\\_CM1 Register Summary: \[1\]](#)

**Table 3-965. CM\_CLKMODE\_DPLL\_ABE**

<b>Address Offset</b>	0x0000 00E0		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	This register allows controlling the DPLL modes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPmode_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED					DPLL_EN			

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature Read 0x0: SSC has been turned off on PLL o/ps Read 0x1: SSC has been turned on on PLL o/ps	R	0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. Read 0x0: REGM4XEN mode of the DPLL is disabled	RW	0
10	DPLL_LPMODE_EN	Set the DPLL in low-power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low-power mode of the DPLL is disabled 0x1: Low-power mode of the DPLL is enabled	RW	0
9	RESERVED		RW	0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0
7:3	RESERVED		RW	0x00
2:0	DPLL_EN	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN bypass mode. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Put the DPLL in MN bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 0x5: Put the DPLL in idle bypass low-power mode. 0x6: Put the DPLL in idle bypass fast-relock mode. 0x7: Enables the DPLL in lock mode	RW	0x4

**Table 3-966. Register Call Summary for Register CM\_CLKMODE\_DPLL\_ABE**

Clock Management Functional Description

- [DPLL\\_ABE Description: \[0\] \[1\] \[2\] \[5\] \[6\] \[7\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[8\]](#)

**Table 3-967. CM\_IDLEST\_DPLL\_ABE**

<b>Address Offset</b>	0x0000 00E4	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_MN_BYPASS	RESERVED										ST_DPLL_CLK				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	ST_MN_BYPASS	DPLL MN_BYPASS status Read 0x1: DPLL is in MN_Bypass Read 0x0: DPLL is not in MN_Bypass	R	0
7:1	RESERVED		R	0x00
0	ST_DPLL_CLK	DPLL lock status Read 0x1: DPLL is LOCKED Read 0x0: DPLL is either in bypass mode or in stop mode.	R	0

**Table 3-968. Register Call Summary for Register CM\_IDLEST\_DPLL\_ABE**

- PRCM Register Manual
- [CKGEN\\_CM1 Register Summary: \[0\]](#)

**Table 3-969. CM\_AUTOIDLE\_DPLL\_ABE**

<b>Address Offset</b>	0x0000 00E8	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides automatic control over the DPLL activity.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AUTO_DPLL_MODE															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	AUTO_DPLL_MODE	<p>DPLL automatic control;</p> <p>0x0: DPLL auto control disabled</p> <p>0x1: The DPLL is automatically put in low-power stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x2: The DPLL is automatically put in fast-relock stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x3: Reserved</p> <p>0x4: Reserved</p> <p>0x5: The DPLL is automatically put in idle bypass low-power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x6: The DPLL is automatically put in idle bypass fast-relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x7: Reserved</p>	RW	0x0

**Table 3-970. Register Call Summary for Register CM\_AUTOIDLE\_DPLL\_ABE**

Clock Management Functional Description

- [DPLL\\_ABE Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)

**Table 3-971. CM\_CLKSEL\_DPLL\_ABE**

<b>Address Offset</b>	0x0000 00EC	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED								DPLL_BYP_CLKSEL	RESERVED								DPLL_MULT								RESERVED	DPLL_DIV							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23	DPLL_BYP_CLKSEL	Only CLKINPULOW bypass clock supported for this PLL	R Rreturns 1s	1
22:19	RESERVED		R	0x0
18:8	DPLL_MULT	<p>DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN bypass mode. (equal to input M of DPLL; M=2 to 2047 =&gt; DPLL multiplies by M).</p> <p>0x0: Reserved</p> <p>0x1: Reserved</p>	RW	0x000
7	RESERVED		R	0



Bits	Field Name	Description	Type	Reset
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).	RW	0x00

**Table 3-972. Register Call Summary for Register CM\_CLKSEL\_DPLL\_ABE**

Clock Management Functional Description

- [DPLL\\_ABE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[2\]](#)

**Table 3-973. CM\_DIV\_M2\_DPLL\_ABE**

<b>Address Offset</b>	0x0000 00F0	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the M2 divider of the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKOUTX2	DPLL_CLKOUTX2_GATE_CTRL	ST_DPLL_CLKOUT	DPLL_CLKOUT_GATE_CTRL	RESERVED		DPLL_CLKOUT_DIVCHACK	DPLL_CLKOUT_DIV								

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11	ST_DPLL_CLKOUTX2	DPLL CLKOUTX2 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
10	DPLL_CLKOUTX2_GATE_CTRL	Control gating of DPLL CLKOUTX2 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
9	ST_DPLL_CLKOUT	DPLL CLKOUT status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	DPLL_CLKOUT_GATE_CTRL	Control gating of DPLL CLKOUT 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	DPLL_CLKOUT_DIVCHACK	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect	R	0
4:0	DPLL_CLKOUT_DIV	DPLL post-divider factor, M2, for internal clock generation (1 to 31). Divide value from 1 to 31. 0x0: Reserved	RW	0x01

**Table 3-974. Register Call Summary for Register CM\_DIV\_M2\_DPLL\_ABE**

Clock Management Functional Description
• <a href="#">DPLL_ABE Description: [0] [1] [2] [3] [4] [5]</a>
PRCM Register Manual
• <a href="#">CKGEN_CM1 Register Summary: [6]</a>

**Table 3-975. CM\_DIV\_M3\_DPLL\_ABE**

<b>Address Offset</b>	0x0000 00F4	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	This register provides controls over the M3 divider of the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKOUTHIF		DPLL_CLKOUTHIF_GATE_CTRL		RESERVED		DPLL_CLKOUTHIF_DIVCHACK		DPLL_CLKOUTHIF_DIV							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	ST_DPLL_CLKOUTHIF	DPLL CLKOUTHIF status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	DPLL_CLKOUTHIF_GATE_CTRL	Control gating of DPLL CLKOUTHIF 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	DPLL_CLKOUTHIF_DIVCHACK	Toggle on this status bit after changing DPLL_CLKOUTHIF_DIV indicates that the change in divider value has taken effect	R	0
4:0	DPLL_CLKOUTHIF_DIV	DPLL post-divider factor, M3, for internal clock generation (1 to 31). Divide value from 1 to 31. 0x0: Reserved	RW	0x01

**Table 3-976. Register Call Summary for Register CM\_DIV\_M3\_DPLL\_ABE**

Clock Management Functional Description
• <a href="#">DPLL_ABE Description: [0] [1] [2] [3]</a>
PRCM Register Manual
• <a href="#">CKGEN_CM1 Register Summary: [4]</a>

**Table 3-977. CM\_SSC\_DELTAMSTEP\_DPLL\_ABE**

<b>Address Offset</b>	0x0000 0108		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											DELTAMSTEP																				

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x00000

**Table 3-978. Register Call Summary for Register CM\_SSC\_DELTAMSTEP\_DPLL\_ABE**

Clock Management Functional Description

- [DPLL\\_ABE Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)

**Table 3-979. CM\_SSC\_MODFREQDIV\_DPLL\_ABE**

<b>Address Offset</b>	0x0000 010C		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	Control the Modulation Frequency (Fm) for Spread Spectrum. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											MODFREQDIV_EXPONENT																				
RESERVED											MODFREQDIV_MANTISSA																				

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x00

**Table 3-980. Register Call Summary for Register CM\_SSC\_MODFREQDIV\_DPLL\_ABE**

Clock Management Functional Description

- [DPLL\\_ABE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[2\]](#)

**Table 3-981. CM\_SHADOW\_FREQ\_CONFIG1**

<b>Address Offset</b>	0x0000 0160		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	Shadow register to program new DPLL configuration affecting EMIF and GPMC (L3 clock) functional frequency during DVFS. The PRCM h/w automatically applies the new configuration after EMIF/GPMC have been put in idle state.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_CORE_M2_DIV				DPLL_CORE_DPLL_EN				RESERVED				DLL_RESET	DLL_OVERRIDE	RESERVED	FREQ_UPDATE								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x10D
15:11	DPLL_CORE_M2_DIV	Shadow register for <a href="#">CM_DIV_M2_DPLL_CORE</a> .DPLL_CLKOUT_DIV. The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to 1. Divide value from 1 to 31.  0x0: Reserved	RW	0x01
10:8	DPLL_CORE_DPLL_EN	Shadow register for <a href="#">CM_CLKMODE_DPLL_CORE</a> .DPLL_EN. The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to 1.  0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Put the DPLL in MN bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 0x5: Put the DPLL in idle bypass low-power mode. 0x6: Put the DPLL in idle bypass fast-relock mode. 0x7: Enables the DPLL in lock mode	RW	0x5
7:4	RESERVED		R	0x0
3	DLL_RESET	Specify if DLL should be reset or not during the frequency change hardware sequence.  0x0: DLL is not reset during the frequency change hardware sequence 0x1: DLL is reset automatically during the frequency change hardware sequence	RW	1
2	DLL_OVERRIDE	Shadow register for <a href="#">CM_DLL_CTRL</a> .DLL_OVERRIDE.The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to 1.  0x0: Lock and code outputs are not overridden 0x1: Lock output is overridden to 1 and code output is overridden with a value coming from control module.	RW	1
1	RESERVED		RW	0

Bits	Field Name	Description	Type	Reset
0	FREQ_UPDATE	Writing 1 indicates that a new configuration is available. It is automatically cleared by h/w after the configuration has been applied.	RW WSpecial	0

**Table 3-982. Register Call Summary for Register CM\_SHADOW\_FREQ\_CONFIG1**

Clock Management Functional Description

- [DPLLs Cascading: \[0\] \[1\]](#)

Device Low-Power States

- [Device Off Mode Wake-Up Sequences: \[2\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[3\]](#)
- [CKGEN\\_CM1 Register Description: \[4\] \[5\] \[6\]](#)
- [RESTORE\\_CM1 Register Description: \[7\] \[8\] \[9\] \[10\]](#)

**Table 3-983. CM\_SHADOW\_FREQ\_CONFIG2**

<b>Address Offset</b>	0x0000 0164	<b>Instance</b>	CKGEN_CM1
<b>Physical Address</b>	See <a href="#">Table 3-898</a>		
<b>Description</b>	Shadow register to program new DPLL configuration affecting GPMC (L3 clock) functional frequency during DVFS. The PRCM h/w automatically applies the new configuration after EMIF/GPMC have been put in idle state.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_CORE_M5_DIV							CLKSEL_L3	CLKSEL_CORE	GPMC_FREQ_UPDATE						

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:3	DPLL_CORE_M5_DIV	Shadow register for <a href="#">CM_DIV_M5_DPLL_CORE.HSDIVIDER_CLKOUT2_DIV</a> . The main register is automatically loaded with the shadow register value after GPMC IDLE if the <a href="#">CM_SHADOW_FREQ_CONFIG1.FREQ_UPDATE</a> field is set to 1 and <a href="#">GPMC_FREQ_UPDATE</a> is set to 1. Divide value from 1 to 31.  0x0: Reserved	RW	0x01
2	CLKSEL_L3	Shadow register for <a href="#">CM_CLKSEL_CORE.CLKSEL_L3</a> . The main register is automatically loaded with the shadow register value after GPMC IDLE if the <a href="#">CM_SHADOW_FREQ_CONFIG1.FREQ_UPDATE</a> field is set to 1 and <a href="#">GPMC_FREQ_UPDATE</a> is set to 1.  0x0: L3_CLK is CORE_CLK divided by 1 0x1: L3_CLK is CORE_CLK divided by 2	RW	0

Bits	Field Name	Description	Type	Reset
1	CLKSEL_CORE	Shadow register for <a href="#">CM_CLKSEL_CORE</a> .CLKSEL_CORE. The main register is automatically loaded with the shadow register value after GPMC IDLE if the <a href="#">CM_SHADOW_FREQ_CONFIG1</a> .FREQ_UPDATE field is set to 1 and GPMC_FREQ_UPDATE is set to 1. 0x0: CORE_CLK is CORE_X2_CLK divided by 1 0x1: CORE_CLK is CORE_X2_CLK divided by 2	RW	0
0	GPMC_FREQ_UPDATE	Controls whether or not GPMC has to be put automatically into idle during the frequency change operation. 0x0: GPMC is not put automatically into idle during frequency change operation. 0x1: GPMC is put automatically into idle during frequency change operation.	RW	0

**Table 3-984. Register Call Summary for Register CM\_SHADOW\_FREQ\_CONFIG2**

Voltage Management Functional Description

- [GPMC Clocks Frequency Scaling Constraints: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[1\]](#)
- [RESTORE\\_CM1 Register Description: \[2\]](#)

**Table 3-985. CM\_DYN\_DEP\_PRESCAL**

<b>Address Offset</b>	0x0000 0170																																																													
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b> CKGEN_CM1																																																												
<b>Description</b>	Control the time unit of the sliding window for dynamic dependencies (auto-sleep feature).																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> </thead> <tbody> <tr> <td colspan="16">RESERVED</td> <td colspan="12">PRESCAL</td> </tr> </tbody> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																PRESCAL											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
RESERVED																PRESCAL																																														
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																										
31:6	RESERVED		R	0x00000000																																																										
5:0	PRESCAL	Time unit is equal to (PRESCAL + 1) L4 clock cycles.	RW	0x20																																																										

**Table 3-986. Register Call Summary for Register CM\_DYN\_DEP\_PRESCAL**

Device Power Management Introduction

- [Clock Management: \[0\]](#)

PRCM Register Manual

- [EMU\\_CM Register Description: \[1\]](#)
- [CKGEN\\_CM1 Register Summary: \[2\]](#)
- [MPU\\_CM1 Register Description: \[3\]](#)
- [DSP\\_CM1 Register Description: \[4\]](#)
- [RESTORE\\_CM1 Register Description: \[5\]](#)
- [CORE\\_CM2 Register Description: \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [L4PER\\_CM2 Register Description: \[11\]](#)
- [RESTORE\\_CM2 Register Description: \[12\] \[13\] \[14\] \[15\] \[16\]](#)

**Table 3-987. CM\_RESTORE\_ST**

<b>Address Offset</b>	0x0000 0180		
<b>Physical Address</b>	See <a href="#">Table 3-898</a>	<b>Instance</b>	CKGEN_CM1
<b>Description</b>	Automatic restore status. This register is used by the system DMA to write a predefined value at the end of end automatic restore phase. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PHASE2B_COMPLETED		PHASE2A_COMPLETED		PHASE1_COMPLETED											

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	PHASE2B_COMPLETED	Indicates if restore phase 2b is completed. Must be cleared by software before going to device OFF mode.	RW	0
1	PHASE2A_COMPLETED	Indicates if restore phase 2a is completed. Must be cleared by software before going to device OFF mode.	RW	0
0	PHASE1_COMPLETED	Indicates if restore phase 1 is completed. Must be cleared by software before going to device OFF mode.	RW	0

**Table 3-988. Register Call Summary for Register CM\_RESTORE\_ST**

PRCM Register Manual

- [CKGEN\\_CM1 Register Summary: \[0\]](#)

### 3.11.24 MPU\_CM1 Registers

#### 3.11.24.1 MPU\_CM1 Register Summary

**Table 3-989. MPU\_CM1 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	MPU_CM1 L4 Base Address
<a href="#">CM_MPU_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A00 4300
<a href="#">CM_MPU_STATICDEP</a>	RW	32	0x0000 0004	0x4A00 4304
<a href="#">CM_MPU_DYNAMICDEP</a>	RW	32	0x0000 0008	0x4A00 4308
<a href="#">CM_MPU_MPU_CLKCTRL</a>	R	32	0x0000 0020	0x4A00 4320

#### 3.11.24.2 MPU\_CM1 Register Description

**Table 3-990. CM\_MPU\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-989</a>	<b>Instance</b>	MPU_CM1
<b>Description</b>	This register enables the MPU domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_MPU_DPLL_CLK	RESERVED						CLKTRCTRL								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_MPU_DPLL_CLK	This field indicates the state of the MPU_DPLL_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the MPU clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. Read 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-991. Register Call Summary for Register CM\_MPU\_CLKSTCTRL**

## Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\]](#)

## PRCM Register Manual

- [MPU\\_CM1 Register Summary: \[2\]](#)
- [RESTORE\\_CM1 Register Description: \[3\]](#)



**Table 3-992. CM\_MPU\_STATICDEP**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	MPU_CM1
<b>Physical Address</b>	See <a href="#">Table 3-989</a>		
<b>Description</b>	This register controls the static domain dependencies from MPU domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RESERVED																C2C_STATDEP	RESERVED	ALWONCORE_STATDEP	L4WKUP_STATDEP	RESERVED	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	SGX_STATDEP	ISS_STATDEP	DSS_STATDEP	L3INIT_STATDEP	L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	ABE_STATDEP	IVAHD_STATDEP	DSP_STATDEP	MPU_M3_STATDEP

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	C2C_STATDEP	Static dependency towards C2C clock domain Read 0x0: Dependency is disabled	R	0
17	RESERVED		R	0
16	ALWONCORE_STATDEP	Static dependency towards ALWONCORE clock domain Read 0x0: Dependency is disabled	R	0
15	L4WKUP_STATDEP	Static dependency towards L4WKUP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
14	RESERVED		R	0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
11	SDMA_STATDEP	Static dependency towards SDMA clock domain Read 0x0: Dependency is disabled	R	0
10	SGX_STATDEP	Static dependency towards SGX clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
9	ISS_STATDEP	Static dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1

Bits	Field Name	Description	Type	Reset
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
3	ABE_STATDEP	Static dependency towards ABE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	DSP_STATDEP	Static dependency towards DSP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	MPU_M3_STATDEP	Static dependency towards MPU_A3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-993. Register Call Summary for Register CM\_MPU\_STATICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

PRCM Register Manual

- [MPU\\_CM1 Register Summary: \[15\]](#)

**Table 3-994. CM\_MPU\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	MPU_CM1
<b>Physical Address</b>	See <a href="#">Table 3-989</a>		
<b>Description</b>	This register controls the dynamic domain dependencies from MPU domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED										L3_1_DYNDEP	MEMIF_DYNDEP	ABE_DYNDEP	RESERVED										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:6	RESERVED		R	0x00000
5	L3_1_DYNDEP	Dynamic dependency towards L3_1 clock domain Read 0x1: Dependency is enabled	R Returns 1s	1

Bits	Field Name	Description	Type	Reset
4	MEMIF_DYNDEP	Dynamic dependency towards MEMIF clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
3	ABE_DYNDEP	Dynamic dependency towards ABE clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
2:0	RESERVED		R	0x0

**Table 3-995. Register Call Summary for Register CM\_MPU\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [MPU\\_CM1 Register Summary: \[3\]](#)

**Table 3-996. CM\_MPU\_MPU\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	MPU_CM1
<b>Physical Address</b>	See <a href="#">Table 3-989</a>		
<b>Description</b>	This register manages the MPU clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STBYST	IDLEST	RESERVED										MODULEMODE											

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-997. Register Call Summary for Register CM\_MPU\_MPU\_CLKCTRL**

Clock Management Functional Description
• <a href="#">Clock Domain Module Attributes: [0] [1] [2]</a>
PRCM Register Manual
• <a href="#">MPU_CM1 Register Summary: [3]</a>

### 3.11.25 DSP\_CM1 Registers

#### 3.11.25.1 DSP\_CM1 Register Summary

**Table 3-998. DSP\_CM1 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSP_CM1 L4 Base Address
<a href="#">CM_DSP_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A00 4400
<a href="#">CM_DSP_STATICDEP</a>	RW	32	0x0000 0004	0x4A00 4404
<a href="#">CM_DSP_DYNAMICDEP</a>	RW	32	0x0000 0008	0x4A00 4408
<a href="#">CM_DSP_DSP_CLKCTRL</a>	RW	32	0x0000 0020	0x4A00 4420

#### 3.11.25.2 DSP\_CM1 Register Description

**Table 3-999. CM\_DSP\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	DSP_CM1
<b>Physical Address</b>	See <a href="#">Table 3-998</a>		
<b>Description</b>	This register enables the DSP domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_DSP_ROOT_CLK	RESERVED							CLKTRCTRL							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_DSP_ROOT_CLK	This field indicates the state of the DSP_ROOT_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain.  0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.  0x1: SW_SLEEP: Start a software forced sleep transition on the domain.  0x2: SW_WKUP: Start a software forced wake-up transition on the domain.  0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

**Table 3-1000. Register Call Summary for Register CM\_DSP\_CLKSTCTRL**

Reset Management Functional Description

- [DSP Subsystem Power-On Reset Sequence: \[0\]](#)

Clock Management Functional Description

- [Clock Domain Modes: \[1\] \[2\]](#)

PRCM Register Manual

- [DSP\\_CM1 Register Summary: \[3\]](#)

**Table 3-1001. CM\_DSP\_STATICDEP**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	DSP_CM1
<b>Physical Address</b>	See <a href="#">Table 3-998</a>		
<b>Description</b>	This register controls the static domain dependencies from DSP domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
RESERVED																ALWONCORE_STATDEP	L4WKUP_STATDEP	RESERVED	L4PER_STATDEP	L4CFG_STATDEP	RESERVED	ISS_STATDEP	RESERVED	L3INIT_STATDEP	L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	ABE_STATDEP	IVAHD_STATDEP	RESERVED											

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16	ALWONCORE_STATDEP	Static dependency towards ALWONCORE clock domain Read 0x0: Dependency is disabled	R	0
15	L4WKUP_STATDEP	Static dependency towards L4WKUP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
14	RESERVED		R	0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

Bits	Field Name	Description	Type	Reset
11:10	RESERVED		R	0x0
9	ISS_STATDEP	Static dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8	RESERVED		R	0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
3	ABE_STATDEP	Static dependency towards ABE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1:0	RESERVED		R	0x0

**Table 3-1002. Register Call Summary for Register CM\_DSP\_STATICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

PRCM Register Manual

- [DSP\\_CM1 Register Summary: \[9\]](#)

**Table 3-1003. CM\_DSP\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	DSP_CM1
<b>Physical Address</b>	See <a href="#">Table 3-998</a>		
<b>Description</b>	This register controls the dynamic domain dependencies from DSP domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED										L3_1_DYNDEP	RESERVED	ABE_DYNDEP	IVAHD_DYNDEP	RESERVED									

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4

Bits	Field Name	Description	Type	Reset
23:6	RESERVED		R	0x00000
5	L3_1_DYNDEP	Dynamic dependency towards L3_1 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
4	RESERVED		R	0
3	ABE_DYNDEP	Dynamic dependency towards ABE clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
2	IVAHD_DYNDEP	Dynamic dependency towards IVAHD clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
1:0	RESERVED		R	0x0

**Table 3-1004. Register Call Summary for Register CM\_DSP\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [DSP\\_CM1 Register Summary: \[3\]](#)

**Table 3-1005. CM\_DSP\_DSP\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	DSP_CM1
<b>Physical Address</b>	See <a href="#">Table 3-998</a>		
<b>Description</b>	This register manages the DSP clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												STBYST	IDLEST	RESERVED												MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	<p>Control the way mandatory clocks are managed.</p> <p>0x0: Module is disabled by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).</p> <p>0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.</p> <p>Read 0x2: Reserved</p> <p>Read 0x3: Reserved</p>	RW	0x0

**Table 3-1006. Register Call Summary for Register CM\_DSP\_DSP\_CLKCTRL**

Reset Management Functional Description

- [DSP Subsystem Power-On Reset Sequence: \[0\]](#)

Clock Management Functional Description

- [Clock Domain Module Attributes: \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [DSP\\_CM1 Register Summary: \[4\]](#)

### 3.11.26 ABE\_CM1 Registers

#### 3.11.26.1 ABE\_CM1 Register Summary

**Table 3-1007. ABE\_CM1 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ABE_CM1 L4 Base Address
<a href="#">CM1_ABE_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A00 4500
<a href="#">CM1_ABE_L4ABE_CLKCTRL</a>	R	32	0x0000 0020	0x4A00 4520
<a href="#">CM1_ABE_AESS_CLKCTRL</a>	RW	32	0x0000 0028	0x4A00 4528
<a href="#">CM1_ABE_PDM_CLKCTRL</a>	RW	32	0x0000 0030	0x4A00 4530
<a href="#">CM1_ABE_DMIC_CLKCTRL</a>	RW	32	0x0000 0038	0x4A00 4538
<a href="#">CM1_ABE_MCASP_CLKCTRL</a>	RW	32	0x0000 0040	0x4A00 4540
<a href="#">CM1_ABE_MCBSP1_CLKCTRL</a>	RW	32	0x0000 0048	0x4A00 4548
<a href="#">CM1_ABE_MCBSP2_CLKCTRL</a>	RW	32	0x0000 0050	0x4A00 4550
<a href="#">CM1_ABE_MCBSP3_CLKCTRL</a>	RW	32	0x0000 0058	0x4A00 4558
<a href="#">CM1_ABE_SLIMBUS_CLKCTRL</a>	RW	32	0x0000 0060	0x4A00 4560
<a href="#">CM1_ABE_GPTIMER5_CLKCTRL</a>	RW	32	0x0000 0068	0x4A00 4568
<a href="#">CM1_ABE_GPTIMER6_CLKCTRL</a>	RW	32	0x0000 0070	0x4A00 4570
<a href="#">CM1_ABE_GPTIMER7_CLKCTRL</a>	RW	32	0x0000 0078	0x4A00 4578
<a href="#">CM1_ABE_GPTIMER8_CLKCTRL</a>	RW	32	0x0000 0080	0x4A00 4580
<a href="#">CM1_ABE_WDTIMER3_CLKCTRL</a>	RW	32	0x0000 0088	0x4A00 4588

#### 3.11.26.2 ABE\_CM1 Register Description



**Table 3-1008. CM1\_ABE\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>	<b>Instance</b>	ABE_CM1
<b>Description</b>	This register enables the ABE domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_ABE_24M_FCLK	CLKACTIVITY_ABE_ALWON_32K_CLK	CLKACTIVITY_ABE_SYSCLK	CLKACTIVITY_24M_FCLK	CLKACTIVITY_ABE_ICLK2	CLKACTIVITY_DPLL_ABE_X2_CLK	RESERVED						CLKTRCTRL			

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13	CLKACTIVITY_ABE_24M_FCLK	This field indicates the state of the ABE_24M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
12	CLKACTIVITY_ABE_ALWON_32K_CLK	This field indicates the state of the ABE_ALWON_32K_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
11	CLKACTIVITY_ABE_SYSCLK	This field indicates the state of the ABE_SYSCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
10	CLKACTIVITY_24M_FCLK	This field indicates the state of the 24M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
9	CLKACTIVITY_ABE_ICLK2	This field indicates the state of the ABE_ICLK2 interface clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_DPLL_ABE_X2_CLK	This field indicates the state of the DPLL_ABE_X2_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	<p>Controls the clock state transition of the ABE clock domain.</p> <p>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.</p> <p>0x1: SW_SLEEP: Start a software forced sleep transition on the domain.</p> <p>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.</p> <p>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.</p>	RW	0x3

**Table 3-1009. Register Call Summary for Register CM1\_ABE\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[7\]](#)

**Table 3-1010. CM1\_ABE\_L4ABE\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the L4ABE clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED												MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	<p>Module idle status. [warm reset insensitive]</p> <p>Read 0x0: Module is fully functional, including INTRCONN</p> <p>Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion</p> <p>Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock</p> <p>Read 0x3: Module is disabled and cannot be accessed</p>	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	<p>Control the way mandatory clocks are managed.</p> <p>Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.</p>	R Rreturns	0x1

**Table 3-1011. Register Call Summary for Register CM1\_ABE\_L4ABE\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- PRCM Register Manual
- [ABE\\_CM1 Register Summary: \[2\]](#)

**Table 3-1012. CM1\_ABE\_AESS\_CLKCTRL**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the AESS clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKSEL_AESS_FCLK		RESERVED				STBYST	IDLEST	RESERVED										MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL_AESS_FCLK	Selects the ratio of AESS_FCLK to ABE_CLK 0x0: AESS_FCLK is divide by 1 of ABE_CLK 0x1: AESS_FCLK is divide by 2 of ABE_CLK	RW	0
23:19	RESERVED		R	0x00
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1013. Register Call Summary for Register CM1\_ABE\_AESS\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[4\]](#)

**Table 3-1014. CM1\_ABE\_PDM\_CLKCTRL**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the PDM clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																IDLEST	RESERVED																MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1015. Register Call Summary for Register CM1\_ABE\_PDM\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[2\]](#)

**Table 3-1016. CM1\_ABE\_DMIC\_CLKCTRL**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>	<b>Instance</b>	ABE_CM1
<b>Description</b>	This register manages the DMIC clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_INTERNAL_SOURCE		CLKSEL_SOURCE		RESERVED				IDLEST		RESERVED											MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CLKSEL_INTERNAL_SOURCE	Selects the internal clock to be used as the functional clock in case CLKSEL_SOURCE selects the internal clock source as the functional clock source. 0x0: 24MHz clock derived from DPLL_ABE is selected 0x1: ABE_SYSCLK is selected 0x2: 24MHz clock derived from DPLL_PER is selected 0x3: Reserved	RW	0x0
25:24	CLKSEL_SOURCE	Selects the source of the functional clock between, internal source, CLKS pad and Audio SLIMBUS_CLK pad. The switching between the clocks is not guaranteed to be glitchless. 0x0: Functional clock is sourced from an internal clock 0x1: Functional clock is sourced from CLKS pad 0x2: Functional clock is sourced from Audio SIMBUS pad 0x3: Reserved	RW	0x0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disabled by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1017. Register Call Summary for Register CM1\_ABE\_DMIC\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\] \[1\]](#)
- [Clock Domain Module Attributes: \[2\] \[3\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[4\]](#)

**Table 3-1018. CM1\_ABE\_MCASP\_CLKCTRL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the MCASP clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_INTERNAL_SOURCE		CLKSEL_SOURCE		RESERVED				IDLEST	RESERVED											MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CLKSEL_INTERNAL_SOURCE	Selects the internal clock to be used as the functional clock in case CLKSEL_SOURCE selects the internal clock source as the functional clock source.  0x0: 24MHz clock derived from DPLL_ABE is selected 0x1: ABE_SYSCCLK is selected 0x2: 24MHz clock derived from DPLL_PER is selected 0x3: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
25:24	CLKSEL_SOURCE	Selects the source of the functional clock between, internal source, CLKS pad and Audio SLIMBUS_CLK pad. The switching between the clocks is not guaranteed to be glitchless.  0x0: Functional clock is sourced from an internal clock 0x1: Functional clock is sourced from CLKS pad 0x2: Functional clock is sourced from Audio SIMBUS pad 0x3: Reserved	RW	0x0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1019. Register Call Summary for Register CM1\_ABE\_MCASP\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\] \[1\]](#)
- [Clock Domain Module Attributes: \[2\] \[3\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[4\]](#)

**Table 3-1020. CM1\_ABE\_MCBSP1\_CLKCTRL**

Address Offset	0x0000 0048	Instance	ABE_CM1
Physical Address	See <a href="#">Table 3-1007</a>		
Description	This register manages the MCBSP1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_INTERNAL_SOURCE		CLKSEL_SOURCE		RESERVED				IDLEST		RESERVED												MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CLKSEL_INTERNAL_SOURCE	Selects the internal clock to be used as the functional clock in case CLKSEL_SOURCE selects the internal clock source as the functional clock source.  0x0: 24MHz clock derived from DPLL_ABE is selected 0x1: ABE_SYSCLK is selected 0x2: 24MHz clock derived from DPLL_PER is selected 0x3: Reserved	RW	0x0
25:24	CLKSEL_SOURCE	Selects the source of the functional clock between, internal source, CLKS pad and Audio SLIMBUS_CLK pad. The switching between the clocks is not guaranteed to be glitchless.  0x0: Functional clock is sourced from an internal clock 0x1: Functional clock is sourced from CLKS pad 0x2: Functional clock is sourced from Audio SIMBUS pad 0x3: Reserved	RW	0x0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1021. Register Call Summary for Register CM1\_ABE\_MCBSP1\_CLKCTRL**
**Clock Management Functional Description**

- [CM Clock Source: \[0\] \[1\]](#)
- [Clock Domain Module Attributes: \[2\] \[3\]](#)



**Table 3-1021. Register Call Summary for Register CM1\_ABE\_MCBSP1\_CLKCTRL (continued)**

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[4\]](#)

**Table 3-1022. CM1\_ABE\_MCBSP2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the MCBSP2 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_INTERNAL_SOURCE		CLKSEL_SOURCE		RESERVED				IDLEST		RESERVED										MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CLKSEL_INTERNAL_SOURCE	Selects the internal clock to be used as the functional clock in case CLKSEL_SOURCE selects the internal clock source as the functional clock source. 0x0: 24MHz clock derived from DPLL_ABE is selected 0x1: ABE_SYSCLK is selected 0x2: 24MHz clock derived from DPLL_PER is selected 0x3: Reserved	RW	0x0
25:24	CLKSEL_SOURCE	Selects the source of the functional clock between, internal source, CLKS pad and Audio SLIMBUS_CLK pad. The switching between the clocks is not guaranteed to be glitchless. 0x0: Functional clock is sourced from an internal clock 0x1: Functional clock is sourced from CLKS pad 0x2: Functional clock is sourced from Audio SIMBUS pad 0x3: Reserved	RW	0x0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disabled by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1023. Register Call Summary for Register CM1\_ABE\_MCBSP2\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\] \[1\]](#)
- [Clock Domain Module Attributes: \[2\] \[3\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[4\]](#)

**Table 3-1024. CM1\_ABE\_MCBSP3\_CLKCTRL**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the MCBSP3 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_INTERNAL_SOURCE		CLKSEL_SOURCE		RESERVED				IDLEST	RESERVED											MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CLKSEL_INTERNAL_SOURCE	Selects the internal clock to be used as the functional clock in case CLKSEL_SOURCE selects the internal clock source as the functional clock source.  0x0: 24MHz clock derived from DPLL_ABE is selected 0x1: ABE_SYSCCLK is selected 0x2: 24MHz clock derived from DPLL_PER is selected 0x3: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
25:24	CLKSEL_SOURCE	Selects the source of the functional clock between, internal source, CLKS pad and Audio SLIMBUS_CLK pad. The switching between the clocks is not guaranteed to be glitchless.  0x0: Functional clock is sourced from an internal clock 0x1: Functional clock is sourced from CLKS pad 0x2: Functional clock is sourced from Audio SIMBUS pad 0x3: Reserved	RW	0x0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1025. Register Call Summary for Register CM1\_ABE\_MCBSP3\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\] \[1\]](#)
- [Clock Domain Module Attributes: \[2\] \[3\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[4\]](#)

**Table 3-1026. CM1\_ABE\_SLIMBUS\_CLKCTRL**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the SLIMBUS clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED																IDLEST	RESERVED				OPTFCLKEN_SLIMBUS_CLK				OPTFCLKEN_FCLK2	OPTFCLKEN_FCLK1	OPTFCLKEN_FCLK0	RESERVED							MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:12	RESERVED		R	0x0
11	OPTFCLKEN_SLIMBUS_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
10	OPTFCLKEN_FCLK2	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
9	OPTFCLKEN_FCLK1	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
8	OPTFCLKEN_FCLK0	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1027. Register Call Summary for Register CM1\_ABE\_SLIMBUS\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[2\]](#)

**Table 3-1028. CM1\_ABE\_GPTIMER5\_CLKCTRL**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the TIMER5 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED					IDLEST	RESERVED										MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Selects between ABE_SYCLK and ABE_ALWON_32K_CLK as the timer functional clock 0x0: Selects ABE_SYCLK as the functional clock 0x1: Selects ABE_ALWON_32K_CLK as the functional clock	RW	0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1029. Register Call Summary for Register CM1\_ABE\_GPTIMER5\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[3\]](#)

**Table 3-1030. CM1\_ABE\_GPTIMER6\_CLKCTRL**

Address Offset	0x0000 0070		
Physical Address	See <a href="#">Table 3-1007</a>	Instance	ABE_CM1
Description	This register manages the TIMER6 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED					IDLEST	RESERVED										MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Selects between ABE_SYCLK and ABE_ALWON_32K_CLK as the timer functional clock 0x0: Selects ABE_SYCLK as the functional clock 0x1: Selects ABE_ALWON_32K_CLK as the functional clock	RW	0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1031. Register Call Summary for Register CM1\_ABE\_GPTIMER6\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[3\]](#)

**Table 3-1032. CM1\_ABE\_GPTIMER7\_CLKCTRL**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the TIMER7 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKSEL	RESERVED				IDLEST	RESERVED												MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Selects between ABE_SYCLK and ABE_ALWON_32K_CLK as the timer functional clock 0x0: Selects ABE_SYCLK as the functional clock 0x1: Selects ABE_ALWON_32K_CLK as the functional clock	RW	0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1033. Register Call Summary for Register CM1\_ABE\_GPTIMER7\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[3\]](#)

**Table 3-1034. CM1\_ABE\_GPTIMER8\_CLKCTRL**

Address Offset	0x0000 0080		
Physical Address	See <a href="#">Table 3-1007</a>	Instance	ABE_CM1
Description	This register manages the TIMER8 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED							IDLEST	RESERVED											MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Selects between ABE_SYCLK and ABE_ALWON_32K_CLK as the timer functional clock 0x0: Selects ABE_SYCLK as the functional clock 0x1: Selects ABE_ALWON_32K_CLK as the functional clock	RW	0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1035. Register Call Summary for Register CM1\_ABE\_GPTIMER8\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[3\]](#)

**Table 3-1036. CM1\_ABE\_WDTIMER3\_CLKCTRL**

<b>Address Offset</b>	0x0000 0088	<b>Instance</b>	ABE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1007</a>		
<b>Description</b>	This register manages the WDT3 clocks.		
<b>Type</b>	RW		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1037. Register Call Summary for Register CM1\_ABE\_WDTIMER3\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [ABE\\_CM1 Register Summary: \[2\]](#)

### 3.11.27 RESTORE\_CM1 Registers

#### 3.11.27.1 RESTORE\_CM1 Register Summary

**Table 3-1038. RESTORE\_CM1 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	RESTORE_CM1 L4 Base Address
<a href="#">CM_CLKSEL_CORE_RESTORE</a>	RW	32	0x0000 0000	0x4A00 4E00
<a href="#">CM_DIV_M2_DPLL_CORE_RESTORE</a>	RW	32	0x0000 0004	0x4A00 4E04
<a href="#">CM_DIV_M3_DPLL_CORE_RESTORE</a>	RW	32	0x0000 0008	0x4A00 4E08
<a href="#">CM_DIV_M4_DPLL_CORE_RESTORE</a>	RW	32	0x0000 000C	0x4A00 4E0C
<a href="#">CM_DIV_M5_DPLL_CORE_RESTORE</a>	RW	32	0x0000 0010	0x4A00 4E10
<a href="#">CM_DIV_M6_DPLL_CORE_RESTORE</a>	RW	32	0x0000 0014	0x4A00 4E14
<a href="#">CM_DIV_M7_DPLL_CORE_RESTORE</a>	RW	32	0x0000 0018	0x4A00 4E18
<a href="#">CM_CLKSEL_DPLL_CORE_RESTORE</a>	RW	32	0x0000 001C	0x4A00 4E1C
<a href="#">CM_SSC_DELTAMSTEP_DPLL_CORE_RESTORE</a>	RW	32	0x0000 0020	0x4A00 4E20

**Table 3-1038. RESTORE\_CM1 Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	RESTORE_CM1 L4 Base Address
<a href="#">CM_SSC_MODFREQDIV_DPLL_CORE_RESTORE</a>	RW	32	0x0000 0024	0x4A00 4E24
<a href="#">CM_CLKMODE_DPLL_CORE_RESTORE</a>	RW	32	0x0000 0028	0x4A00 4E28
<a href="#">CM_SHADOW_FREQ_CONFIG2_RESTORE</a>	RW	32	0x0000 002C	0x4A00 4E2C
<a href="#">CM_SHADOW_FREQ_CONFIG1_RESTORE</a>	RW	32	0x0000 0030	0x4A00 4E30
<a href="#">CM_AUTOIDLE_DPLL_CORE_RESTORE</a>	RW	32	0x0000 0034	0x4A00 4E34
<a href="#">CM_MPU_CLKSTCTRL_RESTORE</a>	RW	32	0x0000 0038	0x4A00 4E38
<a href="#">CM_CM1_PROFILING_CLKCTRL_RESTORE</a>	RW	32	0x0000 003C	0x4A00 4E3C
<a href="#">CM_DYN_DEP_PRESCAL_RESTORE</a>	RW	32	0x0000 0040	0x4A00 4E40
RESERVED	RW	32	0x0000 0044	0x4A00 4E44
RESERVED	RW	32	0x0000 0048	0x4A00 4E48
RESERVED	RW	32	0x0000 004C	0x4A00 4E4C
RESERVED	RW	32	0x0000 0050	0x4A00 4E50
RESERVED	RW	32	0x0000 0054	0x4A00 4E54
RESERVED	RW	32	0x0000 0058	0x4A00 4E58

**3.11.27.2 RESTORE\_CM1 Register Description****Table 3-1039. CM\_CLKSEL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	RESTORE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>		
<b>Description</b>	Second address map for register <a href="#">CM_CLKSEL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKSEL_L4	RESERVED		CLKSEL_L3	RESERVED		CLKSEL_CORE									

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKSEL_L4	Selects L4 interconnect clock (L4_clk) 0x0: L4_CLK is L3_CLK divided by 1 0x1: L4_CLK is L3_CLK divided by 2	RW	0
7:5	RESERVED		R	0x0
4	CLKSEL_L3	Selects L3 interconnect clock (L3_clk) 0x0: L3_CLK is CORE_CLK divided by 1 0x1: L3_CLK is CORE_CLK divided by 2	RW	0
3:1	RESERVED		R	0x0
0	CLKSEL_CORE	Selects CORE_CLK configuration 0x0: CORE_CLK is CORE_X2_CLK divided by 1 0x1: CORE_CLK is CORE_X2_CLK divided by 2	RW	0

**Table 3-1040. Register Call Summary for Register CM\_CLKSEL\_CORE\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[0\]](#)

**Table 3-1041. CM\_DIV\_M2\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	RESTORE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>		
<b>Description</b>	Second address map for register <a href="#">CM_DIV_M2_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKOUT		DPLL_CLKOUT_GATE_CTRL		RESERVED		DPLL_CLKOUT_DIVCHACK		DPLL_CLKOUT_DIV							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	ST_DPLL_CLKOUT	DPLL CLKOUT status Read 0x0: The clock output is enabled Read 0x1: The clock output is gated	R	0
8	DPLL_CLKOUT_GATE_CTRL	Control gating of DPLL CLKOUT 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	DPLL_CLKOUT_DIVCHACK	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect	R	0
4:0	DPLL_CLKOUT_DIV	DPLL post-divider factor, M2, for internal clock generation (1 to 31); Divide value from 1 to 31. 0x0: Reserved	RW	0x01

**Table 3-1042. Register Call Summary for Register CM\_DIV\_M2\_DPLL\_CORE\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[0\]](#)

**Table 3-1043. CM\_DIV\_M3\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_DIV_M3_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKOUTHIF	DPLL_CLKOUTHIF_GATE_CTRL	RESERVED		DPLL_CLKOUTHIF_DIVCHACK	DPLL_CLKOUTHIF_DIV										

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	ST_DPLL_CLKOUTHIF	DPLL CLKOUTHIF status Read 0x0: The clock output is enabled Read 0x1: The clock output is gated	R	0
8	DPLL_CLKOUTHIF_GATE_CTRL	Control gating of DPLL CLKOUTHIF 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	DPLL_CLKOUTHIF_DIVCHACK	Toggle on this status bit after changing DPLL_CLKOUTHIF_DIV indicates that the change in divider value has taken effect	R	0
4:0	DPLL_CLKOUTHIF_DIV	DPLL post-divider factor, M3, for internal clock generation (1 to 31); Divide value from 1 to 31. 0x0: Reserved	RW	0x01

**Table 3-1044. Register Call Summary for Register CM\_DIV\_M3\_DPLL\_CORE\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[0\]](#)

**Table 3-1045. CM\_DIV\_M4\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 000C		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_DIV_M4_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT1_PWDN	RESERVED	ST_HSDIVIDER_CLKOUT1	HSDIVIDER_CLKOUT1_GATE_CTRL	RESERVED	HSDIVIDER_CLKOUT1_DIVCHACK	HSDIVIDER_CLKOUT1_DIV									

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT1_PWDN	Automatic power down for HSDIVIDER M4 divider and hence CLKOUT1 output when the o/p clock is gated. Read 0x0: Keep M4 divider powered on even when CLKOUT1 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT1	HSDIVIDER CLKOUT1 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT1_GATE_CTRL	Control gating of HSDIVIDER CLKOUT1 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT1_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT1_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT1_DIV	DPLL M4 post-divider factor (1 to 31). 0x0: Reserved	RW	0x01

**Table 3-1046. Register Call Summary for Register CM\_DIV\_M4\_DPLL\_CORE\_RESTORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[1\]](#)

**Table 3-1047. CM\_DIV\_M5\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_DIV_M5_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT2_PWDN	RESERVED	ST_HSDIVIDER_CLKOUT2	HSDIVIDER_CLKOUT2_GATE_CTRL	RESERVED	HSDIVIDER_CLKOUT2_DIVCHACK	HSDIVIDER_CLKOUT2_DIV									

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT2_PWDN	Automatic power down for HSDIVIDER M5 divider and hence CLKOUT2 output when the o/p clock is gated. Read 0x0: Keep M5 divider powered on even when CLKOUT2 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT2	HSDIVIDER CLKOUT2 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT2_GATE_CTRL	Control gating of HSDIVIDER CLKOUT2 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT2_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT2_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT2_DIV	DPLL M5 post-divider factor (1 to 31). 0x0: Reserved	RW	0x01

**Table 3-1048. Register Call Summary for Register CM\_DIV\_M5\_DPLL\_CORE\_RESTORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[1\]](#)

**Table 3-1049. CM\_DIV\_M6\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_DIV_M6_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT3_PWDN	RESERVED		ST_HSDIVIDER_CLKOUT3	HSDIVIDER_CLKOUT3_GATE_CTRL	RESERVED		HSDIVIDER_CLKOUT3_DIVCHACK	HSDIVIDER_CLKOUT3_DIV							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT3_PWDN	Automatic power down for HSDIVIDER M6 divider and hence CLKOUT3 output when the o/p clock is gated. Read 0x0: Keep M6 divider powered on even when CLKOUT3 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT3	HSDIVIDER CLKOUT3 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT3_GATE_CTRL	Control gating of HSDIVIDER CLKOUT3 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT3_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT3_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT3_DIV	DPLL M6 post-divider factor (1 to 31). 0x0: Reserved	RW	0x01

**Table 3-1050. Register Call Summary for Register CM\_DIV\_M6\_DPLL\_CORE\_RESTORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[1\]](#)

**Table 3-1051. CM\_DIV\_M7\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 0018		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_DIV_M7_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT4_PWDN	RESERVED		ST_HSDIVIDER_CLKOUT4	HSDIVIDER_CLKOUT4_GATE_CTRL	RESERVED		HSDIVIDER_CLKOUT4_DIVCHACK	HSDIVIDER_CLKOUT4_DIV							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT4_PWDN	Automatic power down for HSDIVIDER M7 divider and hence CLKOUT4 output when the o/p clock is gated. Read 0x0: Keep M7 divider powered on even when CLKOUT4 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT4	HSDIVIDER CLKOUT4 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT4_GATE_CTRL	Control gating of HSDIVIDER CLKOUT4 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT4_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT4_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT4_DIV	DPLL M7 post-divider factor (1 to 31). 0x0: Reserved	RW	0x01

**Table 3-1052. Register Call Summary for Register CM\_DIV\_M7\_DPLL\_CORE\_RESTORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[1\]](#)



**Table 3-1053. CM\_CLKSEL\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 001C		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_CLKSEL_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	RESERVED	DPLL_CLKOUTHIF_CLKSEL	RESERVED	DPLL_MULT								RESERVED	DPLL_DIV										

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0
22:21	RESERVED		R	0x0
20	DPLL_CLKOUTHIF_CLKSEL	Selects the source of the DPLL CLKOUTHIF clock. Same as CLKINPHIFSEL pin on the DPLL 0x0: CLKOUTHIF is generated from the DPLL oscillator (DCO) 0x1: CLKOUTHIF is generated from CLKINPHIF	RW	0
19	RESERVED		R	0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN bypass mode. (equal to input M of DPLL; M=2 to 2047 => DPLL multiplies by M). 0x0: Reserved 0x1: Reserved	RW	0x000
7	RESERVED		R	0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).	RW	0x00

**Table 3-1054. Register Call Summary for Register CM\_CLKSEL\_DPLL\_CORE\_RESTORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[2\]](#)

**Table 3-1055. CM\_SSC\_DELTAMSTEP\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 0020	
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b> RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_SSC_DELTAMSTEP_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												DELTAMSTEP																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x00000

**Table 3-1056. Register Call Summary for Register CM\_SSC\_DELTAMSTEP\_DPLL\_CORE\_RESTORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[1\]](#)

**Table 3-1057. CM\_SSC\_MODFREQDIV\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 0024	
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b> RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_SSC_MODFREQDIV_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												MODFREQDIV_EXPONENT	RESERVED	MODFREQDIV_MANTISSA																	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x00

**Table 3-1058. Register Call Summary for Register CM\_SSC\_MODFREQDIV\_DPLL\_CORE\_RESTORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\] \[1\]](#)

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[2\]](#)

**Table 3-1059. CM\_CLKMODE\_DPLL\_CORE\_RESTORE**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	RESTORE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>		
<b>Description</b>	Second address map for register <a href="#">CM_CLKMODE_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RESERVED																DPLL_SSC_DOWNSPREAD		DPLL_SSC_ACK		DPLL_SSC_EN		DPLL_REGM4XEN		DPLL_LPMODE_EN		RESERVED		DPLL_DRIFTGUARD_EN		RESERVED			DPLL_EN	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature Read 0x0: SSC has been turned off on PLL o/ps Read 0x1: SSC has been turned on on PLL o/ps	R	0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. Read 0x0: REGM4XEN mode of the DPLL is disabled	R	0
10	DPLL_LPMODE_EN	Set the DPLL in low-power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low-power mode of the DPLL is disabled 0x1: Low-power mode of the DPLL is enabled	RW	0
9	RESERVED		RW	0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0

Bits	Field Name	Description	Type	Reset
7:3	RESERVED		RW	0x00
2:0	DPLL_EN	<p>DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN bypass mode.</p> <p>0x0: Reserved</p> <p>0x1: Reserved</p> <p>0x2: Reserved</p> <p>0x3: Reserved</p> <p>0x4: Put the DPLL in MN bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode.</p> <p>0x5: Put the DPLL in idle bypass low-power mode.</p> <p>0x6: Put the DPLL in idle bypass fast-relock mode.</p> <p>0x7: Enables the DPLL in lock mode</p>	RW	0x4

**Table 3-1060. Register Call Summary for Register CM\_CLKMODE\_DPLL\_CORE\_RESTORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[6\]](#)

**Table 3-1061. CM\_SHADOW\_FREQ\_CONFIG2\_RESTORE**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	RESTORE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>		
<b>Description</b>	Second address map for register <a href="#">CM_SHADOW_FREQ_CONFIG2</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_CORE_M5_DIV							CLKSEL_L3		CLKSEL_CORE		GPMC_FREQ_UPDATE				

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:3	DPLL_CORE_M5_DIV	<p>Shadow register for <a href="#">CM_DIV_M5_DPLL_CORE.HSDIVIDER_CLKOUT2_DIV</a>. The main register is automatically loaded with the shadow register value after GPMC IDLE if the <a href="#">CM_SHADOW_FREQ_CONFIG1.FREQ_UPDATE</a> field is set to 1 and GPMC_FREQ_UPDATE is set to 1. Divide value from 1 to 31.</p> <p>0x0: Reserved</p>	RW	0x04

Bits	Field Name	Description	Type	Reset
2	CLKSEL_L3	Shadow register for <a href="#">CM_CLKSEL_CORE</a> .CLKSEL_L3. The main register is automatically loaded with the shadow register value after GPMC IDLE if the <a href="#">CM_SHADOW_FREQ_CONFIG1</a> .FREQ_UPDATE field is set to 1 and GPMC_FREQ_UPDATE is set to 1.  0x0: L3_CLK is CORE_CLK divided by 1 0x1: L3_CLK is CORE_CLK divided by 2	RW	0
1	CLKSEL_CORE	Shadow register for <a href="#">CM_CLKSEL_CORE</a> .CLKSEL_CORE. The main register is automatically loaded with the shadow register value after GPMC IDLE if the <a href="#">CM_SHADOW_FREQ_CONFIG1</a> .FREQ_UPDATE field is set to 1 and GPMC_FREQ_UPDATE is set to 1.  0x0: CORE_CLK is CORE_X2_CLK divided by 1 0x1: CORE_CLK is CORE_X2_CLK divided by 2	RW	0
0	GPMC_FREQ_UPDATE	Controls whether or not GPMC has to be put automatically into idle during the frequency change operation.  0x0: GPMC is not put automatically into idle during frequency change operation. 0x1: GPMC is put automatically into idle during frequency change operation.	RW	0

**Table 3-1062. Register Call Summary for Register [CM\\_SHADOW\\_FREQ\\_CONFIG2\\_RESTORE](#)**

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[0\]](#)

**Table 3-1063. [CM\\_SHADOW\\_FREQ\\_CONFIG1\\_RESTORE](#)**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	RESTORE_CM1
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>		
<b>Description</b>	Second address map for register <a href="#">CM_SHADOW_FREQ_CONFIG1</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_CORE_M2_DIV				DPLL_CORE_DPPLL_EN				RESERVED				DLL_RESET	DLL_OVERRIDE	RESERVED	FREQ_UPDATE								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x10D
15:11	DPLL_CORE_M2_DIV	Shadow register for <a href="#">CM_DIV_M2_DPLL_CORE</a> .DPLL_CLKOUT_DIV. The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to 1. Divide value from 1 to 31.  0x0: Reserved	RW	0x01

Bits	Field Name	Description	Type	Reset
10:8	DPLL_CORE_DPLL_EN	Shadow register for <a href="#">CM_CLKMODE_DPLL_CORE.DPLL_EN</a> . The main register is automatically loaded with the shadow register value after EMIF IDLE if the <a href="#">FREQ_UPDATE</a> field is set to 1.  0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Put the DPLL in MN bypass mode. The <a href="#">DPLL_MULT</a> register bits are reset to 0 automatically by putting the DPLL in this mode. 0x5: Put the DPLL in idle bypass low-power mode. 0x6: Put the DPLL in idle bypass fast-relock mode. 0x7: Enables the DPLL in lock mode	RW	0x5
7:4	RESERVED		R	0x0
3	DLL_RESET	Specify if DLL should be reset or not during the frequency change hardware sequence.  0x0: DLL is not reset during the frequency change hardware sequence 0x1: DLL is reset automatically during the frequency change hardware sequence	RW	1
2	DLL_OVERRIDE	Shadow register for <a href="#">CM_DLL_CTRL.DLL_OVERRIDE</a> . The main register is automatically loaded with the shadow register value after EMIF IDLE if the <a href="#">FREQ_UPDATE</a> field is set to 1.  0x0: Lock and code outputs are not overridden 0x1: Lock output is overridden to 1 and code output is overridden with a value coming from control module.	RW	1
1	RESERVED		RW	0
0	FREQ_UPDATE	Writing 1 indicates that a new configuration is available. It is automatically cleared by h/w after the configuration has been applied.	RW WSpecial	0

**Table 3-1064. Register Call Summary for Register [CM\\_SHADOW\\_FREQ\\_CONFIG1\\_RESTORE](#)**

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[0\]](#)

**Table 3-1065. [CM\\_AUTOIDLE\\_DPLL\\_CORE\\_RESTORE](#)**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_AUTOIDLE_DPLL_CORE</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_DCOCLKLDO_PWDN				RESERVED		AUTO_DPLL_MODE									

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000000
4	DPLL_DCOCLKLDO_PWDN	<p>Allows powering down the DCOCLKLDO o/p of DPLL if all dividers in HSDIVIDER are powered down. PRCM takes care of reenabling this path for either restarting HSDIVIDER o/p or entering bypass mode.</p> <p>0x0: Keep DCOCLKLDO powered even when all dividers in HSDIVIDER are powered down.</p> <p>0x1: Automatically power down DCOCLKLDO when all o/ps of HSDIVIDER are powered down.</p>	RW	0
3	RESERVED		R	0
2:0	AUTO_DPLL_MODE	<p>DPLL automatic control.</p> <p>0x0: DPLL auto control disabled</p> <p>0x1: The DPLL is automatically put in low-power stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x2: The DPLL is automatically put in fast-relock stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x3: Reserved</p> <p>0x4: Reserved</p> <p>0x5: The DPLL is automatically put in idle bypass low-power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x6: The DPLL is automatically put in idle bypass fast-relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically.</p> <p>0x7: Reserved</p>	RW	0x0

**Table 3-1066. Register Call Summary for Register CM\_AUTOIDLE\_DPLL\_CORE\_RESTORE**

Clock Management Functional Description

- [DPLL\\_CORE Description: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[1\]](#)

**Table 3-1067. CM\_MPU\_CLKSTCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_MPU_CLKSTCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_MPU_DPLL_CLK	RESERVED						CLKTRCTRL								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_MPU_DPLL_CLK	This field indicates the state of the MPU_DPLL_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the MPU clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. Read 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1068. Register Call Summary for Register CM\_MPU\_CLKSTCTRL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[0\]](#)



**Table 3-1069. CM\_CM1\_PROFILING\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 003C		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_CM1_PROFILING_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												IDLEST	RESERVED												MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status Read 0x3: Module is disabled Read 0x2: Module is in Idle Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x0: Module is fully functional	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by software. OCP configuration port is not accessible. 0x1: Module is managed automatically by hardware along with CM1 and EMU domain. OCP configuration port is accessible only when EMU domain is on. Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1070. Register Call Summary for Register CM\_CM1\_PROFILING\_CLKCTRL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[0\]](#)

**Table 3-1071. CM\_DYN\_DEP\_PRESCAL\_RESTORE**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 3-1038</a>	<b>Instance</b>	RESTORE_CM1
<b>Description</b>	Second address map for register <a href="#">CM_DYN_DEP_PRESCAL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															PRESCAL																

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5:0	PRESCAL	Time unit is equal to (PRESCAL + 1) L4 clock cycles.	RW	0x20

**Table 3-1072. Register Call Summary for Register CM\_DYN\_DEP\_PRESCAL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM1 Register Summary: \[0\]](#)

**3.11.28 INSTR\_CM1 Registers****3.11.28.1 INSTR\_CM1 Register Summary****Table 3-1073. INSTR\_CM1 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	INSTR_CM1 Base Address	INSTR_CM2 Base Address
<a href="#">CMI_IDENTICATION</a>	R	32	0x0000 0000	0x4A00 4F00	0x4A00 9F00
<a href="#">CMI_SYS_CONFIG</a>	RW	32	0x0000 0010	0x4A00 4F10	0x4A00 9F10
<a href="#">CMI_STATUS</a>	R	32	0x0000 0014	0x4A00 4F14	0x4A00 9F14
<a href="#">CMI_CONFIGURATION</a>	RW	32	0x0000 0024	0x4A00 4F24	0x4A00 9F24
<a href="#">CMI_CLASS_FILTERING</a>	RW	32	0x0000 0028	0x4A00 4F28	0x4A00 9F28
<a href="#">CMI_TRIGGERING</a>	RW	32	0x0000 002C	0x4A00 4F2C	0x4A00 9F2C
<a href="#">CMI_SAMPLING</a>	RW	32	0x0000 0030	0x4A00 4F30	0x4A00 9F30

**3.11.28.2 INSTR\_PRM Register Description****Table 3-1074. CMI\_IDENTICATION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-1073</a>		
<b>Description</b>	PM profiling identification register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	0x-- TI Internal data.

**Table 3-1075. Register Call Summary for Register CMI\_IDENTICATION**

PRCM Register Manual

- [INSTR\\_CM1 Register Summary: \[0\]](#)

**Table 3-1076. CMI\_SYS\_CONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-1073</a>		
<b>Description</b>	PM profiling system configuration register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		IDLEMODE		RESERVED		SOFTRESET									

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x00000000
5:4	RESERVED	Reserved	R	0x0
3:2	IDLEMODE	Configuration of the local target state management mode	RW	0x2
1	RESERVED	Reserved	R	0
0	SOFTRESET	Software reset	RW	0

**Table 3-1077. Register Call Summary for Register CMI\_SYS\_CONFIG**

PRCM Register Manual

- [INSTR\\_CM1 Register Summary: \[0\]](#)

**Table 3-1078. CMI\_STATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-1073</a>		
<b>Description</b>	PM profiling status register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FIFOEMPTY		RESERVED													

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x00000000
8	FIFOEMPTY	PM Profiling buffer empty	R	1
7:0	RESERVED	Reserved	R	0x00

**Table 3-1079. Register Call Summary for Register CMI\_STATUS**

PRCM Register Manual

- [INSTR\\_CM1 Register Summary: \[0\]](#)

**Table 3-1080. CMI\_CONFIGURATION**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-1073</a>		
<b>Description</b>	PM profiling configuration register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLAIM_3	CLAIM_2	CLAIM_1	RESERVED				RESERVED	RESERVED						RESERVED	RESERVED						EVT_CAPT_EN	RESERVED									

Bits	Field Name	Description	Type	Reset
31:30	CLAIM_3	Ownership	RW	0x0
29	CLAIM_2	Debugger override qualifier	RW	1
28	CLAIM_1	Current owner	R	0
27:24	RESERVED	Reserved	R	0x0
23	RESERVED	Reserved	R	0
22:16	RESERVED	Reserved	R	0x00
15	RESERVED	Reserved	R	0
14:8	RESERVED	Reserved	R	0x00
7	EVT_CAPT_EN	When HIGH the PM events capture is enabled	RW	0
6:0	RESERVED	Reserved	R	0x00

**Table 3-1081. Register Call Summary for Register CMI\_CONFIGURATION**

PRCM Register Manual

- [INSTR\\_CM1 Register Summary: \[0\]](#)

**Table 3-1082. CMI\_CLASS\_FILTERING**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	INSTR_PRM
<b>Physical Address</b>	See <a href="#">Table 3-1073</a>		
<b>Description</b>	PM profiling class filtering register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SNAP_CAPT_EN_03	SNAP_CAPT_EN_02	SNAP_CAPT_EN_01	SNAP_CAPT_EN_00

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	SNAP_CAPT_EN_03	Snapshot capture enable - Class-ID = 0x03	RW	0
2	SNAP_CAPT_EN_02	Snapshot capture enable - Class-ID = 0x02	RW	0
1	SNAP_CAPT_EN_01	Snapshot capture enable - Class-ID = 0x01	RW	0
0	SNAP_CAPT_EN_00	Snapshot capture enable - Class-ID = 0x00	RW	0

**Table 3-1083. Register Call Summary for Register CMI\_CLASS\_FILTERING**

PRCM Register Manual

- [INSTR\\_CM1 Register Summary: \[0\]](#)

**Table 3-1084. CMI\_TRIGGERING**

<b>Address Offset</b>	0x0000 002C		
<b>Physical Address</b>	See <a href="#">Table 3-1073</a>	<b>Instance</b>	INSTR_PRM
<b>Description</b>	PM profiling triggering control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TRIG_STOP_EN		TRIG_START_EN													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1	TRIG_STOP_EN	Enable stop capturing PM events from external trigger detection	RW	0
0	TRIG_START_EN	Enable start capturing PM events from external trigger detection	RW	0

**Table 3-1085. Register Call Summary for Register CMI\_TRIGGERING**

PRCM Register Manual

- [INSTR\\_CM1 Register Summary: \[0\]](#)

**Table 3-1086. CMI\_SAMPLING**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 3-1073</a>	<b>Instance</b>	INSTR_PRM
<b>Description</b>	PM profiling sampling window register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											FCLK_DIV_FACOR		RESERVED					SAMP_WIND_SIZE													

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x000
19:16	FCLK_DIV_FACOR	FunClk divide factor ranging from 1 to 16	RW	0x0
15:8	RESERVED	Reserved	R	0x00
7:0	SAMP_WIND_SIZE	PM events sampling window size	RW	0x00

**Table 3-1087. Register Call Summary for Register CMI\_SAMPLING**

PRCM Register Manual

- [INSTR\\_CM1 Register Summary: \[0\]](#)

### 3.11.29 CM2 Instance Summary

**Table 3-1088. CM2 Instance Summary**

Module Name	L4 Base Address	Size
INTRCONN_SOCKET_CM2	0x4A00 8000	256Bytes
CKGEN_CM2	0x4A00 8100	256Bytes
ALWAYS_ON_CM2	0x4A00 8600	256Bytes
CORE_CM2	0x4A00 8700	2KBytes
IWAHD_CM2	0x4A00 8F00	256Bytes
CAM_CM2	0x4A00 9000	256Bytes
DSS_CM2	0x4A00 9100	256Bytes
SGX_CM2	0x4A00 9200	256Bytes
L3INIT_CM2	0x4A00 9300	256Bytes
L4PER_CM2	0x4A00 9400	512Bytes
RESTORE_CM2	0x4A00 9E00	256Bytes
INSTR_CM2	0x4A00 9F00	256Bytes

### 3.11.30 INTRCONN\_SOCKET\_CM2 Registers

#### 3.11.30.1 INTRCONN\_SOCKET\_CM2 Register Summary

**Table 3-1089. INTRCONN\_SOCKET\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	INTRCONN_SOCKET_CM2 L4 Base Address
<a href="#">REVISION_CM2</a>	R	32	0x0000 0000	0x4A00 8000
<a href="#">CM_CM2_PROFILING_CLKCTRL</a>	RW	32	0x0000 0040	0x4A00 8040
<a href="#">CM2_DEBUG_CFG</a>	RW	32	0x0000 00F0	0x4A00 80F0

#### 3.11.30.2 INTRCONN\_SOCKET\_CM2 Register Description

**Table 3-1090. REVISION\_CM2**

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See <a href="#">Table 3-1089</a>
<b>Description</b>	This register contains the IP revision code for the CM2 part of the PRCM
<b>Type</b>	R
<b>Instance</b>	INTRCONN_SOCKET_CM2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Revision Number	R	0x- - TI Internal data.

**Table 3-1091. Register Call Summary for Register REVISION\_CM2**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_CM2 Register Summary: \[0\]](#)

**Table 3-1092. CM\_CM2\_PROFILING\_CLKCTRL**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 3-1089</a>	<b>Instance</b>	INTRCONN_SOCKET_CM2
<b>Description</b>	This register manages the CM2_PROFILING clocks. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										IDLEST	RESERVED										MODULEMODE										

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status Read 0x0: Module is fully functional Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in Idle Read 0x3: Module is disabled	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by software. INTRCONN configuration port is not accessible. 0x1: Module is managed automatically by hardware along with L3INSTR domain. Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1093. Register Call Summary for Register CM\_CM2\_PROFILING\_CLKCTRL**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_CM2 Register Summary: \[0\]](#)
- [RESTORE\\_CM2 Register Description: \[1\]](#)

**Table 3-1094. CM2\_DEBUG\_CFG**

<b>Address Offset</b>	0x0000 00F0		
<b>Physical Address</b>	See <a href="#">Table 3-1089</a>	<b>Instance</b>	INTRCONN_SOCKET_CM2
<b>Description</b>	This register is used to configure the CM2's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3								SEL2								SEL1								SEL0							

Bits	Field Name	Description	Type	Reset
31:24	SEL3	Internal signal block select for debug word byte-3	RW	0x03
23:16	SEL2	Internal signal block select for debug word byte-2	RW	0x02
15:8	SEL1	Internal signal block select for debug word byte-1	RW	0x01
7:0	SEL0	Internal signal block select for debug word byte-0	RW	0x00

**Table 3-1095. Register Call Summary for Register CM2\_DEBUG\_CFG**

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_CM2 Register Summary: \[0\]](#)

**3.11.31 CKGEN\_CM2 Registers****3.11.31.1 CKGEN\_CM2 Register Summary****Table 3-1096. CKGEN\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_CM2 L4 Base Address
<a href="#">CM_CLKSEL_MPU_M3_ISS_ROOT</a>	RW	32	0x0000 0000	0x4A00 8100
<a href="#">CM_CLKSEL_USB_60MHZ</a>	RW	32	0x0000 0004	0x4A00 8104
<a href="#">CM_SCALE_FCLK</a>	RW	32	0x0000 0008	0x4A00 8108
<a href="#">CM_CORE_DVFS_PERF1</a>	RW	32	0x0000 0010	0x4A00 8110
<a href="#">CM_CORE_DVFS_PERF2</a>	RW	32	0x0000 0014	0x4A00 8114
<a href="#">CM_CORE_DVFS_PERF3</a>	RW	32	0x0000 0018	0x4A00 8118
<a href="#">CM_CORE_DVFS_PERF4</a>	RW	32	0x0000 001C	0x4A00 811C
<a href="#">CM_CORE_DVFS_CURRENT</a>	RW	32	0x0000 0024	0x4A00 8124
<a href="#">CM_IVA_DVFS_PERF_DSP</a>	RW	32	0x0000 0028	0x4A00 8128
<a href="#">CM_IVA_DVFS_PERF_IVAHD</a>	RW	32	0x0000 002C	0x4A00 812C
<a href="#">CM_IVA_DVFS_PERF_ABE</a>	RW	32	0x0000 0030	0x4A00 8130
<a href="#">CM_IVA_DVFS_CURRENT</a>	RW	32	0x0000 0038	0x4A00 8138
<a href="#">CM_CLKMODE_DPLL_PER</a>	RW	32	0x0000 0040	0x4A00 8140
<a href="#">CM_IDLEST_DPLL_PER</a>	R	32	0x0000 0044	0x4A00 8144
<a href="#">CM_AUTOIDLE_DPLL_PER</a>	RW	32	0x0000 0048	0x4A00 8148
<a href="#">CM_CLKSEL_DPLL_PER</a>	RW	32	0x0000 004C	0x4A00 814C
<a href="#">CM_DIV_M2_DPLL_PER</a>	RW	32	0x0000 0050	0x4A00 8150
<a href="#">CM_DIV_M3_DPLL_PER</a>	RW	32	0x0000 0054	0x4A00 8154
<a href="#">CM_DIV_M4_DPLL_PER</a>	RW	32	0x0000 0058	0x4A00 8158
<a href="#">CM_DIV_M5_DPLL_PER</a>	RW	32	0x0000 005C	0x4A00 815C
<a href="#">CM_DIV_M6_DPLL_PER</a>	RW	32	0x0000 0060	0x4A00 8160
<a href="#">CM_DIV_M7_DPLL_PER</a>	RW	32	0x0000 0064	0x4A00 8164
<a href="#">CM_SSC_DELTAMSTEP_DPLL_PER</a>	RW	32	0x0000 0068	0x4A00 8168
<a href="#">CM_SSC_MODFREQDIV_DPLL_PER</a>	RW	32	0x0000 006C	0x4A00 816C
RESERVED	RW	32	0x0000 0070	0x4A00 8170
<a href="#">CM_CLKMODE_DPLL_USB</a>	RW	32	0x0000 0080	0x4A00 8180
<a href="#">CM_IDLEST_DPLL_USB</a>	R	32	0x0000 0084	0x4A00 8184
<a href="#">CM_AUTOIDLE_DPLL_USB</a>	RW	32	0x0000 0088	0x4A00 8188
<a href="#">CM_CLKSEL_DPLL_USB</a>	RW	32	0x0000 008C	0x4A00 818C
<a href="#">CM_DIV_M2_DPLL_USB</a>	RW	32	0x0000 0090	0x4A00 8190
<a href="#">CM_SSC_DELTAMSTEP_DPLL_USB</a>	RW	32	0x0000 00A8	0x4A00 81A8
<a href="#">CM_SSC_MODFREQDIV_DPLL_USB</a>	RW	32	0x0000 00AC	0x4A00 81AC
<a href="#">CM_CLKDCOLDO_DPLL_USB</a>	RW	32	0x0000 00B4	0x4A00 81B4
RESERVED	RW	32	0x0000 00C0	0x4A00 81C0
RESERVED	R	32	0x0000 00C4	0x4A00 81C4
RESERVED	RW	32	0x0000 00C8	0x4A00 81C8
RESERVED	RW	32	0x0000 00CC	0x4A00 81CC
RESERVED	RW	32	0x0000 00D0	0x4A00 81D0
RESERVED	RW	32	0x0000 00E8	0x4A00 81E8



**Table 3-1096. CKGEN\_CM2 Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_CM2 L4 Base Address
RESERVED	RW	32	0x0000 00EC	0x4A00 81EC

**3.11.31.2 CKGEN\_CM2 Register Description**

**Table 3-1097. CM\_CLKSEL\_MPU\_M3\_ISS\_ROOT**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b> CKGEN_CM2
<b>Description</b>	MPU_A3/ISS root clock selection (MPU_A3_ISS_CLK).	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKSEL															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CLKSEL	Select the source for the root clock of MPU_A3/ISS 0x0: Root clock is sourced from CORE_CLK 0x1: Functional clock is sourced from DPLL_PER	RW	0

**Table 3-1098. Register Call Summary for Register CM\_CLKSEL\_MPU\_M3\_ISS\_ROOT**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[1\]](#)

**Table 3-1099. CM\_CLKSEL\_USB\_60MHZ**

<b>Address Offset</b>	0x0000 0004	
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b> CKGEN_CM2
<b>Description</b>	Selects the configuration of the divider generating 60MHz clock for USB from the DPLL_USB o/p.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKSEL															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CLKSEL	Select the configuration of the divider 0x0: Set the divider in bypass mode to support bypass clock from DPLL_USB to pass through without division. 0x1: Set the divider to divide the DPLL o/p (480MHz typical) by 8 to generate 60MHz clock.	RW	1

**Table 3-1100. Register Call Summary for Register CM\_CLKSEL\_USB\_60MHZ**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[1\]](#)

**Table 3-1101. CM\_SCALE\_FCLK**

<b>Address Offset</b>	0x0000 0008	
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b> CKGEN_CM2
<b>Description</b>	This register can be used to scale PER_ABE_NC_FCLK, 96M_FCLK, 48M_FCLK, and 64M_FCLK to half their respective typical frequencies.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SCALE_FCLK															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	SCALE_FCLK	Enable or disable the functional clock scaling. 0x0: The functional clocks run at their default frequencies 0x1: The functional clocks run at half their typical frequencies	RW	0

**Table 3-1102. Register Call Summary for Register CM\_SCALE\_FCLK**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[1\]](#)

**Table 3-1103. CM\_CORE\_DVFS\_PERF1**

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b> CKGEN_CM2
<b>Description</b>	This register allows to system master #1 to specify which level of performance is required from CORE domain (mainly external memory throughput?)	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PERF_REQ															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	PERF_REQ	Current performance request. Unit to be defined by user.	RW	0x00

**Table 3-1104. Register Call Summary for Register CM\_CORE\_DVFS\_PERF1**

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[0\]](#)

**Table 3-1105. CM\_CORE\_DVFS\_PERF2**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register allows to system master #2 to specify which level of performance is require from CORE domain (mainly external memory throughput?)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PERF_REQ															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	PERF_REQ	Current performance request. Unit to be defined by user.	RW	0x00

**Table 3-1106. Register Call Summary for Register CM\_CORE\_DVFS\_PERF2**

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[0\]](#)

**Table 3-1107. CM\_CORE\_DVFS\_PERF3**

<b>Address Offset</b>	0x0000 0018		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register allows to system master #3 to specify which level of performance is require from CORE domain (mainly external memory throughput?)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PERF_REQ															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	PERF_REQ	Current performance request. Unit to be defined by user.	RW	0x00

**Table 3-1108. Register Call Summary for Register CM\_CORE\_DVFS\_PERF3**

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[0\]](#)

**Table 3-1109. CM\_CORE\_DVFS\_PERF4**

<b>Address Offset</b>	0x0000 001C		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register allows to system master #4 to specify which level of performance is require from CORE domain (mainly external memory throughput?)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PERF_REQ															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	PERF_REQ	Current performance request. Unit to be defined by user.	RW	0x00

**Table 3-1110. Register Call Summary for Register CM\_CORE\_DVFS\_PERF4**

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[0\]](#)

**Table 3-1111. CM\_CORE\_DVFS\_CURRENT**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	This register hold the current level of performance achievable by the CORE domain, according to the current OPP setting		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PERF_CURRENT															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	PERF_CURRENT	Current achievable performance level. Unit to be defined by user.	RW	0x00

**Table 3-1112. Register Call Summary for Register CM\_CORE\_DVFS\_CURRENT**

Voltage Management Functional Description

- [DVFS Voting Mechanism \(VDD\\_CORE\\_L and VDD\\_IVA\\_L Voltage Domains\): \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[1\]](#)

**Table 3-1113. CM\_IVA\_DVFS\_PERF\_DSP**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	This register allows to specify which level of performance is required from IVA domain for DSP to operate.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PERF_REQ															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	PERF_REQ	Current performance request. Unit to be defined by user.	RW	0x00

**Table 3-1114. Register Call Summary for Register CM\_IVA\_DVFS\_PERF\_DSP**

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[0\]](#)

**Table 3-1115. CM\_IVA\_DVFS\_PERF\_IVAHD**

<b>Address Offset</b>	0x0000 002C	
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b> CKGEN_CM2
<b>Description</b>	This register allows to specify which level of performance is required from IVA domain for IVAHD to operate.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PERF_REQ															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	PERF_REQ	Current performance request. Unit to be defined by user.	RW	0x00

**Table 3-1116. Register Call Summary for Register CM\_IVA\_DVFS\_PERF\_IVAHD**

- PRCM Register Manual
- [CKGEN\\_CM2 Register Summary: \[0\]](#)

**Table 3-1117. CM\_IVA\_DVFS\_PERF\_ABE**

<b>Address Offset</b>	0x0000 0030	
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b> CKGEN_CM2
<b>Description</b>	This register allows to specify which level of performance is required from IVA domain for ABE to operate.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PERF_REQ															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	PERF_REQ	Current performance request. Unit to be defined by user.	RW	0x00

**Table 3-1118. Register Call Summary for Register CM\_IVA\_DVFS\_PERF\_ABE**

- PRCM Register Manual
- [CKGEN\\_CM2 Register Summary: \[0\]](#)

**Table 3-1119. CM\_IVA\_DVFS\_CURRENT**

<b>Address Offset</b>	0x0000 0038	
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b> CKGEN_CM2
<b>Description</b>	This register hold the current level of performance achievable by the IVA domain, according to the current OPP setting	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PERF_CURRENT															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	PERF_CURRENT	Current achievable performance level. Unit to be defined by user.	RW	0x00

**Table 3-1120. Register Call Summary for Register CM\_IVA\_DVFS\_CURRENT**

Voltage Management Functional Description

- [DVFS Voting Mechanism \(VDD\\_CORE\\_L and VDD\\_IVA\\_L Voltage Domains\): \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[1\]](#)

**Table 3-1121. CM\_CLKMODE\_DPLL\_PER**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	This register allows controlling the DPLL modes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	RESERVED	DPLL_DRIFTGUARD_EN	RESERVED				DPLL_EN				

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x000000
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature Read 0x0: SSC has been turned off on PLL o/ps Read 0x1: SSC has been turned on on PLL o/ps	R	0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. Read 0x0: REGM4XEN mode of the DPLL is disabled	R	0
10	DPLL_LPMODE_EN	Set the DPLL in low-power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low-power mode of the DPLL is disabled 0x1: Low-power mode of the DPLL is enabled	RW	0
9	RESERVED		RW	0

Bits	Field Name	Description	Type	Reset
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0
7:3	RESERVED		RW	0x00
2:0	DPLL_EN	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN bypass mode. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Put the DPLL in MN bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 0x5: Put the DPLL in idle bypass low-power mode. 0x6: Put the DPLL in idle bypass fast-relock mode. 0x7: Enables the DPLL in lock mode	RW	0x4

**Table 3-1122. Register Call Summary for Register CM\_CLKMODE\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\] \[1\] \[2\] \[5\] \[6\] \[7\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[8\]](#)

**Table 3-1123. CM\_IDLEST\_DPLL\_PER**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_MN_BYPASS	RESERVED										ST_DPLL_CLK				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	ST_MN_BYPASS	DPLL MN_BYPASS status Read 0x1: DPLL is in MN_Bypass Read 0x0: DPLL is not in MN_Bypass	R	0
7:1	RESERVED		R	0x00
0	ST_DPLL_CLK	DPLL lock status Read 0x1: DPLL is LOCKED Read 0x0: DPLL is either in bypass mode or in stop mode.	R	0

**Table 3-1124. Register Call Summary for Register CM\_IDLEST\_DPLL\_PER**

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[0\]](#)

**Table 3-1125. CM\_AUTOIDLE\_DPLL\_PER**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	This register provides automatic control over the DPLL activity.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AUTO_DPLL_MODE															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x00000000
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in low-power stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in fast-relock stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in idle bypass low-power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in idle bypass fast-relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

**Table 3-1126. Register Call Summary for Register CM\_AUTOIDLE\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[1\]](#)



**Table 3-1127. CM\_CLKSEL\_DPLL\_PER**

<b>Address Offset</b>	0x0000 004C		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register provides controls over the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	RESERVED				DPLL_MULT								RESERVED	DPLL_DIV									

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0
22:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN bypass mode. (equal to input M of DPLL; M=2 to 2047 => DPLL multiplies by M). 0x0: Reserved 0x1: Reserved	RW	0x000
7	RESERVED		R	0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).	RW	0x00

**Table 3-1128. Register Call Summary for Register CM\_CLKSEL\_DPLL\_PER**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [DPLL\\_PER Description: \[1\] \[2\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[3\]](#)

**Table 3-1129. CM\_DIV\_M2\_DPLL\_PER**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register provides controls over the M2 divider of the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																ST_DPLL_CLKOUTX2		DPLL_CLKOUTX2_GATE_CTRL		ST_DPLL_CLKOUT		DPLL_CLKOUT_GATE_CTRL		RESERVED		DPLL_CLKOUT_DIVCHACK		DPLL_CLKOUT_DIV					

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11	ST_DPLL_CLKOUTX2	DPLL CLKOUTX2 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
10	DPLL_CLKOUTX2_GATE_CTRL	Control gating of DPLL CLKOUTX2 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
9	ST_DPLL_CLKOUT	DPLL CLKOUT status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	DPLL_CLKOUT_GATE_CTRL	Control gating of DPLL CLKOUT 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	DPLL_CLKOUT_DIVCHACK	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect	R	0
4:0	DPLL_CLKOUT_DIV	DPLL post-divider factor, M2, for internal clock generation (1 to 31) 0x0: Reserved	RW	0x01

**Table 3-1130. Register Call Summary for Register CM\_DIV\_M2\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[6\]](#)

**Table 3-1131. CM\_DIV\_M3\_DPLL\_PER**

<b>Address Offset</b>	0x0000 0054		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register provides controls over the M3 divider of the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKOUTHIF		DPLL_CLKOUTHIF_GATE_CTRL		RESERVED		DPLL_CLKOUTHIF_DIVCHACK		DPLL_CLKOUTHIF_DIV							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	ST_DPLL_CLKOUTHIF	DPLL CLKOUTHIF status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	DPLL_CLKOUTHIF_GATE_CTRL	Control gating of DPLL CLKOUTHIF 0x0: Gate unconditionally this clock and stop requesting it 0x1: Request this clock and un-gate it when available	RW	0
7:6	RESERVED		R	0x0
5	DPLL_CLKOUTHIF_DIVCHACK	Toggle on this status bit after changing DPLL_CLKOUTHIF_DIV indicates that the change in divider value has taken effect	R	0
4:0	DPLL_CLKOUTHIF_DIV	DPLL post-divisor factor, M3, for internal clock generation (1 to 31) 0x0: Reserved	RW	0x01

**Table 3-1132. Register Call Summary for Register CM\_DIV\_M3\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[4\]](#)

**Table 3-1133. CM\_DIV\_M4\_DPLL\_PER**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT1_PWDN	RESERVED	ST_HSDIVIDER_CLKOUT1	HSDIVIDER_CLKOUT1_GATE_CTRL	RESERVED	HSDIVIDER_CLKOUT1_DIVCHACK	HSDIVIDER_CLKOUT1_DIV									

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT1_PWDN	Automatic power down for HSDIVIDER M4 divider and hence CLKOUT1 output when the o/p clock is gated.  Read 0x0: Keep M4 divider powered on even when CLKOUT1 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT1	HSDIVIDER CLKOUT1 status  Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT1_GATE_CTRL	Control gating of HSDIVIDER CLKOUT1  0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT1_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT1_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT1_DIV	DPLL M4 post-divider factor (1 to 31).  0x0: Reserved	RW	0x04

**Table 3-1134. Register Call Summary for Register CM\_DIV\_M4\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[5\]](#)

**Table 3-1135. CM\_DIV\_M5\_DPLL\_PER**

<b>Address Offset</b>	0x0000 005C		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT2_PWDN	RESERVED		ST_HSDIVIDER_CLKOUT2	HSDIVIDER_CLKOUT2_GATE_CTRL		RESERVED		HSDIVIDER_CLKOUT2_DIVCHACK	HSDIVIDER_CLKOUT2_DIV						

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT2_PWDN	Automatic power down for HSDIVIDER M5 divider and hence CLKOUT2 output when the o/p clock is gated.  Read 0x0: Keep M5 divider powered on even when CLKOUT2 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT2	HSDIVIDER CLKOUT2 status  Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT2_GATE_CTRL	Control gating of HSDIVIDER CLKOUT2  0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT2_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT2_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT2_DIV	DPLL M5 post-divider factor (1 to 31).  0x0: Reserved	RW	0x04

**Table 3-1136. Register Call Summary for Register CM\_DIV\_M5\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[5\]](#)

**Table 3-1137. CM\_DIV\_M6\_DPLL\_PER**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register provides controls over the CLKOUT3 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT3_PWDN	RESERVED	ST_HSDIVIDER_CLKOUT3	HSDIVIDER_CLKOUT3_GATE_CTRL	RESERVED	HSDIVIDER_CLKOUT3_DIVCHACK	HSDIVIDER_CLKOUT3_DIV									

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT3_PWDN	Automatic power down for HSDIVIDER M6 divider and hence CLKOUT3 output when the o/p clock is gated.  Read 0x0: Keep M6 divider powered on even when CLKOUT3 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT3	HSDIVIDER CLKOUT3 status  Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT3_GATE_CTRL	Control gating of HSDIVIDER CLKOUT3  0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT3_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT3_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT3_DIV	DPLL M6 post-divider factor (1 to 31). 0x0: Reserved	RW	0x04

**Table 3-1138. Register Call Summary for Register CM\_DIV\_M6\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[5\]](#)

**Table 3-1139. CM\_DIV\_M7\_DPLL\_PER**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register provides controls over the CLKOUT4 o/p of the HSDIVIDER.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSDIVIDER_CLKOUT4_PWDN	RESERVED		ST_HSDIVIDER_CLKOUT4	HSDIVIDER_CLKOUT4_GATE_CTRL		RESERVED		HSDIVIDER_CLKOUT4_DIVCHACK	HSDIVIDER_CLKOUT4_DIV						

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HSDIVIDER_CLKOUT4_PWDN	Automatic power down for HSDIVIDER M7 divider and hence CLKOUT4 output when the o/p clock is gated. Read 0x0: Keep M7 divider powered on even when CLKOUT4 is gated.	R	0
11:10	RESERVED		R	0x0
9	ST_HSDIVIDER_CLKOUT4	HSDIVIDER CLKOUT4 status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	HSDIVIDER_CLKOUT4_GATE_CTRL	Control gating of HSDIVIDER CLKOUT4 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:6	RESERVED		R	0x0
5	HSDIVIDER_CLKOUT4_DIVCHACK	Toggle on this status bit after changing HSDIVIDER_CLKOUT4_DIV indicates that the change in divider value has taken effect	R	0
4:0	HSDIVIDER_CLKOUT4_DIV	DPLL M7 post-divider factor (1 to 31). 0x0: Reserved	RW	0x04

**Table 3-1140. Register Call Summary for Register CM\_DIV\_M7\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[5\]](#)

**Table 3-1141. CM\_SSC\_DELTAMSTEP\_DPLL\_PER**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELTAMSTEP																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x00000

**Table 3-1142. Register Call Summary for Register CM\_SSC\_DELTAMSTEP\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[1\]](#)

**Table 3-1143. CM\_SSC\_MODFREQDIV\_DPLL\_PER**

<b>Address Offset</b>	0x0000 006C		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	Control the Modulation Frequency (Fm) for Spread Spectrum. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											MODFREQDIV_EXPONENT		RESERVED	MODFREQDIV_MANTISSA																	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x00

**Table 3-1144. Register Call Summary for Register CM\_SSC\_MODFREQDIV\_DPLL\_PER**

Clock Management Functional Description

- [DPLL\\_PER Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[2\]](#)



**Table 3-1145. CM\_CLKMODE\_DPLL\_USB**

<b>Address Offset</b>	0x0000 0080		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register allows controlling the DPLL modes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	RESERVED								DPLL_EN				

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature Read 0x0: SSC has been turned off on PLL o/ps Read 0x1: SSC has been turned on on PLL o/ps	R	0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0
11:3	RESERVED		R	0x000
2:0	DPLL_EN	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect DPLL low-power stop mode. 0x0: Reserved 0x1: Put the DPLL in low-power stop mode 0x2: Reserved2 0x3: Reserved 0x4: Put the DPLL in MN bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 0x5: Put the DPLL in idle bypass low-power mode. 0x6: Reserved 0x7: Enables the DPLL in lock mode	RW	0x4

**Table 3-1146. Register Call Summary for Register CM\_CLKMODE\_DPLL\_USB**

Clock Management Functional Description

- [DPLL\\_USB Description: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[4\]](#)

**Table 3-1147. CM\_IDLEST\_DPLL\_USB**

<b>Address Offset</b>	0x0000 0084		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_MN_BYPASS	RESERVED						ST_DPLL_CLK								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	ST_MN_BYPASS	DPLL MN_BYPASS status Read 0x1: DPLL is in MN_Bypass Read 0x0: DPLL is not in MN_Bypass	R	0
7:1	RESERVED		R	0x00
0	ST_DPLL_CLK	DPLL lock status Read 0x1: DPLL is LOCKED Read 0x0: DPLL is either in bypass mode or in stop mode.	R	0

**Table 3-1148. Register Call Summary for Register CM\_IDLEST\_DPLL\_USB**

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[0\]](#)

**Table 3-1149. CM\_AUTOIDLE\_DPLL\_USB**

<b>Address Offset</b>	0x0000 0088		
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>	<b>Instance</b>	CKGEN_CM2
<b>Description</b>	This register provides automatic control over the DPLL activity.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AUTO_DPLL_MODE															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in low-power stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in idle bypass low-power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: Reserved 0x7: Reserved	RW	0x0

**Table 3-1150. Register Call Summary for Register CM\_AUTOIDLE\_DPLL\_USB**

Clock Management Functional Description

- [DPLL\\_USB Description: \[0\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[1\]](#)

**Table 3-1151. CM\_CLKSEL\_DPLL\_USB**

<b>Address Offset</b>	0x0000 008C	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	This register provides controls over the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPLL_SD_DIV								DPLL_BYP_CLKSEL	RESERVED	DPLL_MULT								DPLL_DIV													

Bits	Field Name	Description	Type	Reset
31:24	DPLL_SD_DIV	Sigma-Delta divider select (2-255). This factor must be set by s/w to ensure optimum jitter performance. DPLL_SD_DIV = CEILING ((DPLL_MULT/(DPLL_DIV+1)) * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. 0x0: Reserved 0x1: Reserved	RW	0x00
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL locked mode, 0 - No impact 1 - No impact In DPLL bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT	RW	0
22:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19:8	DPLL_MULT	DPLL multiplier factor (2 to 4095). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN bypass mode. (equal to input M of DPLL; M=2 to 4095 => DPLL multiplies by M).  0x0: Reserved 0x1: Reserved	RW	0x000
7:0	DPLL_DIV	DPLL divider factor (0 to 255) (equal to input N of DPLL; actual division factor is N+1).	RW	0x00

**Table 3-1152. Register Call Summary for Register CM\_CLKSEL\_DPLL\_USB**

Clock Management Functional Description

- [DPLL\\_USB Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[2\]](#)

**Table 3-1153. CM\_DIV\_M2\_DPLL\_USB**

<b>Address Offset</b>	0x0000 0090	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	This register provides controls over the M2 divider of the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKOUT		DPLL_CLKOUT_GATE_CTRL		DPLL_CLKOUT_DIVCHACK		DPLL_CLKOUT_DIV									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x00000
9	ST_DPLL_CLKOUT	DPLL CLKOUT status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	DPLL_CLKOUT_GATE_CTRL	Control gating of DPLL CLKOUT 0x0: Automatically gate this clock when there is no dependency for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7	DPLL_CLKOUT_DIVCHACK	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect	R	0
6:0	DPLL_CLKOUT_DIV	DPLL post-divider factor, M2, for internal clock generation (1 to 127) 0x0: Reserved	RW	0x01

**Table 3-1154. Register Call Summary for Register CM\_DIV\_M2\_DPLL\_USB**

- Clock Management Functional Description
- [DPLL\\_USB Description: \[0\] \[1\] \[2\] \[3\]](#)
- PRCM Register Manual
- [CKGEN\\_CM2 Register Summary: \[4\]](#)

**Table 3-1155. CM\_SSC\_DELTAMSTEP\_DPLL\_USB**

<b>Address Offset</b>	0x0000 00A8	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELTAMSTEP																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x00000

**Table 3-1156. Register Call Summary for Register CM\_SSC\_DELTAMSTEP\_DPLL\_USB**

- Clock Management Functional Description
- [DPLL\\_USB Description: \[0\]](#)
- PRCM Register Manual
- [CKGEN\\_CM2 Register Summary: \[1\]](#)

**Table 3-1157. CM\_SSC\_MODFREQDIV\_DPLL\_USB**

<b>Address Offset</b>	0x0000 00AC	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	Control the Modulation Frequency (Fm) for Spread Spectrum. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											MODFREQDIV_EXPONENT		RESERVED	MODFREQDIV_MANTISSA																	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x00

**Table 3-1158. Register Call Summary for Register CM\_SSC\_MODFREQDIV\_DPLL\_USB**

Clock Management Functional Description

- [DPLL\\_USB Description: \[0\] \[1\]](#)

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[2\]](#)

**Table 3-1159. CM\_CLKDCOLDO\_DPLL\_USB**

<b>Address Offset</b>	0x0000 00B4	<b>Instance</b>	CKGEN_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1096</a>		
<b>Description</b>	This register provides controls over the M2 divider of the DPLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ST_DPLL_CLKDCOLDO		DPLL_CLKDCOLDO_GATE_CTRL		RESERVED											

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	ST_DPLL_CLKDCOLDO	DPLL CLKDCOLDO status Read 0x0: The clock output is gated Read 0x1: The clock output is enabled	R	0
8	DPLL_CLKDCOLDO_GATE_CTRL	Control gating of DPLL CLKDCOLDO 0x0: Automatically gate this clock when there is no request for it 0x1: Force this clock to stay enabled even if there is no request	RW	0
7:0	RESERVED		R	0x00

**Table 3-1160. Register Call Summary for Register CM\_CLKDCOLDO\_DPLL\_USB**

PRCM Register Manual

- [CKGEN\\_CM2 Register Summary: \[0\]](#)

### 3.11.32 ALWAYS\_ON\_CM2 Registers

#### 3.11.32.1 ALWAYS\_ON\_CM2 Register Summary

**Table 3-1161. ALWAYS\_ON\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ALWAYS_ON_CM2 L4 Base Address
<a href="#">CM_ALWON_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A00 8600
RESERVED	R	32	0x0000 0020	0x4A00 8620
<a href="#">CM_ALWON_SR_MPU_CLKCTRL</a>	RW	32	0x0000 0028	0x4A00 8628

**Table 3-1161. ALWAYS\_ON\_CM2 Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ALWAYS_ON_CM2 L4 Base Address
CM_ALWON_SR_IVA_CLKCTRL	RW	32	0x0000 0030	0x4A00 8630
CM_ALWON_SR_CORE_CLKCTRL	RW	32	0x0000 0038	0x4A00 8638
CM_ALWON_USBPHY_CLKCTRL	RW	32	0x0000 0040	0x4A00 8640

**3.11.32.2 ALWAYS\_ON\_CM2 Register Description**

**Table 3-1162. CM\_ALWON\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-1161</a>	<b>Instance</b>	ALWAYS_ON_CM2
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED																CLKACTIVITY_SR_CORE_SYSCLK				CLKACTIVITY_SR_IVA_SYSCLK				CLKACTIVITY_SR_MPU_SYSCLK				CLKACTIVITY_L4_AO_ICLK				RESERVED				CLKCTRL			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11	CLKACTIVITY_SR_CORE_SYSCLK	This field indicates the state of the SR_CORE_SYSCLK clock input of the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
10	CLKACTIVITY_SR_IVA_SYSCLK	This field indicates the state of the SR_IVA_SYSCLK clock input of the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
9	CLKACTIVITY_SR_MPU_SYSCLK	This field indicates the state of the SR_MPU_SYSCLK clock input of the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_L4_AO_ICLK	This field indicates the state of the L4_AO_ICLK clock of the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	<p>Controls the clock state transition of the ALWONCORE clock domain.</p> <p>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.</p> <p>Read 0x1: Reserved</p> <p>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.</p> <p>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.</p>	RW	0x3

**Table 3-1163. Register Call Summary for Register CM\_ALWON\_CLKSTCTRL**

PRCM Register Manual

- [ALWAYS\\_ON\\_CM2 Register Summary: \[0\]](#)

**Table 3-1164. CM\_ALWON\_SR\_MPU\_CLKCTRL**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 3-1161</a>	<b>Instance</b>	ALWAYS_ON_CM2
<b>Description</b>	This register manages the SR_MPU clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED																MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	<p>Module idle status. [warm reset insensitive]</p> <p>Read 0x0: Module is fully functional, including INTRCONN</p> <p>Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion</p> <p>Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock</p> <p>Read 0x3: Module is disabled and cannot be accessed</p>	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	<p>Control the way mandatory clocks are managed.</p> <p>0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).</p> <p>Read 0x1: Reserved</p> <p>0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.</p> <p>Read 0x3: Reserved</p>	RW	0x0



**Table 3-1165. Register Call Summary for Register CM\_ALWON\_SR\_MPU\_CLKCTRL**

Clock Management Functional Description  
 • [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual  
 • [ALWAYS\\_ON\\_CM2 Register Summary: \[2\]](#)

**Table 3-1166. CM\_ALWON\_SR\_IVA\_CLKCTRL**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	ALWAYS_ON_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1161</a>		
<b>Description</b>	This register manages the SR_IVA clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED										MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1167. Register Call Summary for Register CM\_ALWON\_SR\_IVA\_CLKCTRL**

Clock Management Functional Description  
 • [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual  
 • [ALWAYS\\_ON\\_CM2 Register Summary: \[2\]](#)

**Table 3-1168. CM\_ALWON\_SR\_CORE\_CLKCTRL**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-1161</a>	<b>Instance</b>	ALWAYS_ON_CM2
<b>Description</b>	This register manages the SR_CORE clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1169. Register Call Summary for Register CM\_ALWON\_SR\_CORE\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [ALWAYS\\_ON\\_CM2 Register Summary: \[2\]](#)

**Table 3-1170. CM\_ALWON\_USBPHY\_CLKCTRL**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	Please refer to <a href="#">Table 3-1161</a>	<b>Instance</b>	ALWAYS_ON_CM2
<b>Description</b>	This register manages the USB PHY 32KHz clock.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OPTFCLKEN_CLK32K	RESERVED														

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	OPTFCLKEN_CLK32K	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:0	RESERVED		R	0x00

**Table 3-1171. Register Call Summary for Register CM\_ALWON\_USBPHY\_CLKCTRL**

PRCM Register Manual

- [ALWAYS\\_ON\\_CM2 Register Summary: \[0\]](#)

### 3.11.33 CORE\_CM2 Registers

#### 3.11.33.1 CORE\_CM2 Register Summary

**Table 3-1172. CORE\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CORE_CM2 L4 Base Address
<a href="#">CM_L3_1_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A00 8700
<a href="#">CM_L3_1_DYNAMICDEP</a>	RW	32	0x0000 0008	0x4A00 8708
<a href="#">CM_L3_1_L3_1_CLKCTRL</a>	R	32	0x0000 0020	0x4A00 8720
<a href="#">CM_L3_2_CLKSTCTRL</a>	RW	32	0x0000 0100	0x4A00 8800
<a href="#">CM_L3_2_DYNAMICDEP</a>	RW	32	0x0000 0108	0x4A00 8808
<a href="#">CM_L3_2_L3_2_CLKCTRL</a>	R	32	0x0000 0120	0x4A00 8820
<a href="#">CM_L3_2_GPMC_CLKCTRL</a>	RW	32	0x0000 0128	0x4A00 8828
<a href="#">CM_L3_2_OCMC_RAM_CLKCTRL</a>	R	32	0x0000 0130	0x4A00 8830
<a href="#">CM_MPU_M3_CLKSTCTRL</a>	RW	32	0x0000 0200	0x4A00 8900
<a href="#">CM_MPU_M3_STATICDEP</a>	RW	32	0x0000 0204	0x4A00 8904
<a href="#">CM_MPU_M3_DYNAMICDEP</a>	RW	32	0x0000 0208	0x4A00 8908
<a href="#">CM_MPU_M3_MPU_M3_CLKCTRL</a>	RW	32	0x0000 0220	0x4A00 8920
<a href="#">CM_SDMA_CLKSTCTRL</a>	RW	32	0x0000 0300	0x4A00 8A00
<a href="#">CM_SDMA_STATICDEP</a>	RW	32	0x0000 0304	0x4A00 8A04
<a href="#">CM_SDMA_DYNAMICDEP</a>	R	32	0x0000 0308	0x4A00 8A08
<a href="#">CM_SDMA_SDMA_CLKCTRL</a>	R	32	0x0000 0320	0x4A00 8A20
<a href="#">CM_MEMIF_CLKSTCTRL</a>	RW	32	0x0000 0400	0x4A00 8B00
<a href="#">CM_MEMIF_DMM_CLKCTRL</a>	R	32	0x0000 0420	0x4A00 8B20
<a href="#">CM_MEMIF_EMIF_FW_CLKCTRL</a>	R	32	0x0000 0428	0x4A00 8B28
<a href="#">CM_MEMIF_EMIF_1_CLKCTRL</a>	RW	32	0x0000 0430	0x4A00 8B30
<a href="#">CM_MEMIF_EMIF_2_CLKCTRL</a>	RW	32	0x0000 0438	0x4A00 8B38
<a href="#">CM_MEMIF_DLL_CLKCTRL</a>	RW	32	0x0000 0440	0x4A00 8B40
RESERVED	RW	32	0x0000 0450	0x4A00 8B50

**Table 3-1172. CORE\_CM2 Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	CORE_CM2 L4 Base Address
RESERVED	RW	32	0x0000 0458	0x4A00 8B58
RESERVED	RW	32	0x0000 0460	0x4A00 8B60
CM_C2C_CLKSTCTRL	RW	32	0x0000 0500	0x4A00 8C00
CM_C2C_STATICDEP	RW	32	0x0000 0504	0x4A00 8C04
CM_C2C_DYNAMICDEP	RW	32	0x0000 0508	0x4A00 8C08
CM_C2C_C2C_CLKCTRL	R	32	0x0000 0520	0x4A00 8C20
RESERVED	R	32	0x0000 0528	0x4A00 8C28
CM_C2C_C2C_FW_CLKCTRL	R	32	0x0000 0530	0x4A00 8C30
CM_L4CFG_CLKSTCTRL	RW	32	0x0000 0600	0x4A00 8D00
CM_L4CFG_DYNAMICDEP	RW	32	0x0000 0608	0x4A00 8D08
CM_L4CFG_L4_CFG_CLKCTRL	R	32	0x0000 0620	0x4A00 8D20
CM_L4CFG_SPINLOCK_CLKCTRL	R	32	0x0000 0628	0x4A00 8D28
CM_L4CFG_MAILBOX_CLKCTRL	R	32	0x0000 0630	0x4A00 8D30
CM_L4CFG_SAR_ROM_CLKCTRL	R	32	0x0000 0638	0x4A00 8D38
CM_L3INSTR_CLKSTCTRL	R	32	0x0000 0700	0x4A00 8E00
CM_L3INSTR_L3_3_CLKCTRL	RW	32	0x0000 0720	0x4A00 8E20
CM_L3INSTR_L3_INSTR_CLKCTRL	RW	32	0x0000 0728	0x4A00 8E28
CM_L3INSTR_INTRCONN_WP1_CLKCTRL	RW	32	0x0000 0740	0x4A00 8E40

**3.11.33.2 CORE\_CM2 Register Description****Table 3-1173. CM\_L3\_1\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_L3_1_ICLK	RESERVED										CLKTRCTRL				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_L3_1_ICLK	This field indicates the state of the L3_1_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the L3_1 clock domain.  0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.  Read 0x1: Reserved Read 0x2: Reserved  0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1174. Register Call Summary for Register CM\_L3\_1\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1175. CM\_L3\_1\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register controls the dynamic domain dependencies from L3_1 domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED								L4CFG_DYNDEP				RESERVED				L3_2_DYNDEP	RESERVED	MEMIF_DYNDEP	ABE_DYNDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:13	RESERVED		R	0x000
12	L4CFG_DYNDEP	Dynamic dependency towards L4CFG clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
11:7	RESERVED		R	0x00
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	RESERVED		R	0
4	MEMIF_DYNDEP	Dynamic dependency towards MEMIF clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
3	ABE_DYNDEP	Dynamic dependency towards ABE clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
2:0	RESERVED		R	0x0

**Table 3-1176. Register Call Summary for Register CM\_L3\_1\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[4\]](#)
- [RESTORE\\_CM2 Register Description: \[5\]](#)

**Table 3-1177. CM\_L3\_1\_L3\_1\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register manages the L3_1 clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST		RESERVED										MODULEMODE			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1178. Register Call Summary for Register CM\_L3\_1\_L3\_1\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1179. CM\_L3\_2\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0100
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>
<b>Instance</b>	CORE_CM2
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_L3_2_ICLK	RESERVED						CLKTRCTRL								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_L3_2_ICLK	This field indicates the state of the L3_2_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the L3_2 clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. Read 0x1: Reserved Read 0x2: Reserved 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1180. Register Call Summary for Register CM\_L3\_2\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1181. CM\_L3\_2\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0108		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register controls the dynamic domain dependencies from L3_2 domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED								L4PER_DYNDEP	RESERVED	SGX_DYNDEP	CAM_DYNDEP	DSS_DYNDEP	L3_INIT_DYNDEP	RESERVED	L3_1_DYNDEP	RESERVED	IVAHD_DYNDEP	RESERVED	MPU_M3_DYNDEP				

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:14	RESERVED		R	0x000
13	L4PER_DYNDEP	Dynamic dependency towards L4PER clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
12:11	RESERVED		R	0x0
10	SGX_DYNDEP	Dynamic dependency towards SGX clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
9	CAM_DYNDEP	Dynamic dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8	DSS_DYNDEP	Dynamic dependency towards DSS clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
7	L3_INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
6	RESERVED		R	0
5	L3_1_DYNDEP	Dynamic dependency towards L3_1 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
4:3	RESERVED		R	0x0
2	IVAHD_DYNDEP	Dynamic dependency towards IVAHD clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
1	RESERVED		R	0
0	MPU_M3_DYNDEP	Dynamic dependency towards CORTEXM3 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1

**Table 3-1182. Register Call Summary for Register CM\_L3\_2\_DYNAMICDEP**

## Clock Management Functional Description

- [Clock Domain Dependency](#): [0] [1] [2] [3] [4] [5] [6] [7] [8]

## PRCM Register Manual

- [CORE\\_CM2 Register Summary](#): [9]
- [RESTORE\\_CM2 Register Description](#): [10]



**Table 3-1183. CM\_L3\_2\_L3\_2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0120		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the L3_2 clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Returns	0x1

**Table 3-1184. Register Call Summary for Register CM\_L3\_2\_L3\_2\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- PRCM Register Manual
- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1185. CM\_L3\_2\_GPMC\_CLKCTRL**

<b>Address Offset</b>	0x0000 0128		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the GPMC clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is temporarily disabled by software. Interconnect access to module are stalled. Can be used to change timing parameter of GPMC module. 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1186. Register Call Summary for Register CM\_L3\_2\_GPMC\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1187. CM\_L3\_2\_OCMC\_RAM\_CLKCTRL**

<b>Address Offset</b>	0x0000 0130		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the OCMC_RAM clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED										MODULEMODE												

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1188. Register Call Summary for Register CM\_L3\_2\_OCMC\_RAM\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1189. CM\_MPU\_M3\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0200	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_MPU_M3_CLK	RESERVED								CLKTRCTRL						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_MPU_M3_CLK	This field indicates the state of the MPU_M3_CLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the MPU_A3 clock domain.  0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.  0x1: SW_SLEEP: Start a software forced sleep transition on the domain.  0x2: SW_WKUP: Start a software forced wake-up transition on the domain.  0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

**Table 3-1190. Register Call Summary for Register CM\_MPU\_M3\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1191. CM\_MPU\_M3\_STATICDEP**

<b>Address Offset</b>	0x0000 0204	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register controls the static domain dependencies from MPU_A3 domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																ALWONCORE_STATDEP	L4WKUP_STATDEP	RESERVED	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	SGX_STATDEP	ISS_STATDEP	DSS_STATDEP	L3INIT_STATDEP	L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	ABE_STATDEP	IVAHD_STATDEP	DSP_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16	ALWONCORE_STATDEP	Static dependency towards ALWONCORE clock domain Read 0x0: Dependency is disabled	R	0
15	L4WKUP_STATDEP	Static dependency towards L4WKUP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
14	RESERVED		RW	0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
11	SDMA_STATDEP	Static dependency towards SDMA clock domain Read 0x0: Dependency is disabled	R	0
10	SGX_STATDEP	Static dependency towards SGX clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
9	ISS_STATDEP	Static dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

Bits	Field Name	Description	Type	Reset
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
3	ABE_STATDEP	Static dependency towards ABE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	DSP_STATDEP	Static dependency towards DSP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
0	RESERVED		R	0

**Table 3-1192. Register Call Summary for Register CM\_MPU\_M3\_STATICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[13\]](#)

**Table 3-1193. CM\_MPU\_M3\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0208
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>
<b>Instance</b>	CORE_CM2
<b>Description</b>	This register controls the dynamic domain dependencies from MPU_A3 domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED										CAM_DYNDEP	RESERVED	L3_2_DYNDEP	RESERVED										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:10	RESERVED		R	0x0000
9	CAM_DYNDEP	Dynamic dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8:7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5:0	RESERVED		R	0x00

**Table 3-1194. Register Call Summary for Register CM\_MPU\_M3\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1195. CM\_MPU\_M3\_MPU\_M3\_CLKCTRL**

<b>Address Offset</b>	0x0000 0220	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register manages the MPU_A3 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED														STBYST		IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1196. Register Call Summary for Register CM\_MPU\_M3\_MPU\_M3\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\] \[2\]](#)
- PRCM Register Manual
- [CORE\\_CM2 Register Summary: \[3\]](#)

**Table 3-1197. CM\_SDMA\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0300	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_DMA_L3_ICLK	RESERVED							CLKTRCTRL							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_DMA_L3_ICLK	This field indicates the state of the DMA_L3_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the SDMA clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. Read 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

**Table 3-1198. Register Call Summary for Register CM\_SDMA\_CLKSTCTRL**

- Clock Management Functional Description
- [Clock Domain Modes: \[0\] \[1\]](#)
- PRCM Register Manual
- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1199. CM\_SDMA\_STATICDEP**

<b>Address Offset</b>	0x0000 0304		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register controls the static domain dependencies from SDMA domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								L4WKUP_STATDEP	RESERVED	L4PER_STATDEP	L4CFG_STATDEP	RESERVED	ISS_STATDEP	DSS_STATDEP	L3INIT_STATDEP	L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	ABE_STATDEP	IVAHD_STATDEP	RESERVED	MPU_M3_STATDEP									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	L4WKUP_STATDEP	Static dependency towards L4WKUP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
14	RESERVED		R	0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
11:10	RESERVED		R	0x0
9	ISS_STATDEP	Static dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
3	ABE_STATDEP	Static dependency towards ABE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0



Bits	Field Name	Description	Type	Reset
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	MPU_M3_STATDEP	Static dependency towards MPU_A3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-1200. Register Call Summary for Register CM\_SDMA\_STATICDEP**

Device Low-Power States

- [Device Off Mode Wake-Up Sequences: \[0\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[1\]](#)
- [RESTORE\\_CM2 Register Description: \[2\]](#)

**Table 3-1201. CM\_SDMA\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0308	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register controls the dynamic domain dependencies from SDMA domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L3_2_DYNDEP		RESERVED													

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x0: Dependency is disabled	R	0
5:0	RESERVED		R	0x00

**Table 3-1202. Register Call Summary for Register CM\_SDMA\_DYNAMICDEP**

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1203. CM\_SDMA\_SDMA\_CLKCTRL**

<b>Address Offset</b>	0x0000 0320		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the SDMA clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STBYST		IDLEST		RESERVED								MODULEMODE											

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Returns	0x1

**Table 3-1204. Register Call Summary for Register CM\_SDMA\_SDMA\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[3\]](#)

**Table 3-1205. CM\_MEMIF\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0400		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_PHY_ROOT_CLK	CLKACTIVITY_DLL_CLK	CLKACTIVITY_L3_EMIF_ICLK	RESERVED							CLKTRCTRL					

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10	CLKACTIVITY_PHY_ROOT_CLK	This field indicates the state of the PHY_ROOT_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
9	CLKACTIVITY_DLL_CLK	This field indicates the state of the DLL_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_L3_EMIF_ICLK	This field indicates the state of the L3_EMIF_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the MEMIF clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. Read 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1206. Register Call Summary for Register CM\_MEMIF\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[4\]](#)
- [RESTORE\\_CM2 Register Description: \[5\]](#)

**Table 3-1207. CM\_MEMIF\_DMM\_CLKCTRL**

<b>Address Offset</b>	0x0000 0420	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register manages the DMM clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST		RESERVED										MODULEMODE			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1208. Register Call Summary for Register CM\_MEMIF\_DMM\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1209. CM\_MEMIF\_EMIF\_FW\_CLKCTRL**

<b>Address Offset</b>	0x0000 0428		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the EMIF_FW clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Returns	0x1

**Table 3-1210. Register Call Summary for Register CM\_MEMIF\_EMIF\_FW\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- PRCM Register Manual
- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1211. CM\_MEMIF\_EMIF\_1\_CLKCTRL**

<b>Address Offset</b>	0x0000 0430		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the EMIF_1 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is temporarily disabled by software. Interconnect access to module are stalled. Can be used to change timing parameter of EMIF1 module. 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1212. Register Call Summary for Register CM\_MEMIF\_EMIF\_1\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1213. CM\_MEMIF\_EMIF\_2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0438	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register manages the EMIF_2 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED																MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is temporarily disabled by software. Interconnect access to module are stalled. Can be used to change timing parameter of EMIF2 module.  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1214. Register Call Summary for Register CM\_MEMIF\_EMIF\_2\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1215. CM\_MEMIF\_DLL\_CLKCTRL**

<b>Address Offset</b>	0x0000 0440	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register manages the DLL clock.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OPTFCLKEN_DLL_CLK	RESERVED														

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	OPTFCLKEN_DLL_CLK	Optional functional clock control.  0x0: Optional functional clock is disabled. DLL_CLK can be gated when MEMIF domain performs sleep transition  0x1: Optional functional clock is enabled. DLL_CLK is guaranteed to not be gated if already running.	RW	0
7:0	RESERVED		R	0x00

**Table 3-1216. Register Call Summary for Register CM\_MEMIF\_DLL\_CLKCTRL**

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1217. CM\_C2C\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0500		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_L3X2_C2C_ICLK			CLKACTIVITY_L4_C2C_ICLK			CLKACTIVITY_L3_C2C_ICLK			RESERVED						CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10	CLKACTIVITY_L3X2_C2C_ICLK	This field indicates the state of the C2C_L3X2_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
9	CLKACTIVITY_L4_C2C_ICLK	This field indicates the state of the L4_C2C_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_L3_C2C_ICLK	This field indicates the state of the L3_C2C_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the C2C clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. Read 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

**Table 3-1218. Register Call Summary for Register CM\_C2C\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[4\]](#)



**Table 3-1219. CM\_C2C\_STATICDEP**

<b>Address Offset</b>	0x0000 0504		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register controls the static domain dependencies from C2C domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L4PER_STATDEP	L4CFG_STATDEP	RESERVED					L3INIT_STATDEP	L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	ABE_STATDEP	IVAHD_STATDEP	RESERVED		

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
11:8	RESERVED		R	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
3	ABE_STATDEP	Static dependency towards ABE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1:0	RESERVED		R	0x0

**Table 3-1220. Register Call Summary for Register CM\_C2C\_STATICDEP**

- Clock Management Functional Description
- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- PRCM Register Manual
- [CORE\\_CM2 Register Summary: \[8\]](#)
  - [RESTORE\\_CM2 Register Description: \[9\]](#)

**Table 3-1221. CM\_C2C\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0508		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register controls the dynamic domain dependencies from C2C domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED										L3_2_DYNDEP	RESERVED	MEMIF_DYNDEP	RESERVED										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:7	RESERVED		R	0x00000
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	RESERVED		R	0
4	MEMIF_DYNDEP	Dynamic dependency towards MEMIF clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
3:0	RESERVED		R	0x0

**Table 3-1222. Register Call Summary for Register CM\_C2C\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1223. CM\_C2C\_C2C\_CLKCTRL**

<b>Address Offset</b>	0x0000 0520		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the C2C clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										STBYST	IDLEST	RESERVED										MODULEMODE									

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	C2C module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	C2C interface idle status. [warm reset insensitive] Read 0x0: C2C interface is in functional state Read 0x1: C2C interface is in a transitory state Read 0x2: C2C interface is in IDLE state	R	0x2
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1224. Register Call Summary for Register CM\_C2C\_C2C\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[3\]](#)

**Table 3-1225. CM\_C2C\_C2C\_FW\_CLKCTRL**

<b>Address Offset</b>	0x0000 0530	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register manages the C2C_FW clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST		RESERVED										MODULEMODE			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1226. Register Call Summary for Register CM\_C2C\_C2C\_FW\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1227. CM\_L4CFG\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0600		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_CFG_L4_ICLK	RESERVED										CLKTRCTRL				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_CFG_L4_ICLK	This field indicates the state of the CFG_L4_ICLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the L4CFG clock domain.  0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.  Read 0x1: Reserved Read 0x2: Reserved  0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1228. Register Call Summary for Register CM\_L4CFG\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1229. CM\_L4CFG\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0608	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register controls the dynamic domain dependencies from L4_CFG domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED				C2C_DYNDEP	RESERVED	ALWONCORE_DYNDEP	L4WKUP_DYNDEP	RESERVED				SDMA_DYNDEP	RESERVED	CAM_DYNDEP	DSS_DYNDEP	L3_INIT_DYNDEP	L3_2_DYNDEP	L3_1_DYNDEP	MEMIF_DYNDEP	RESERVED	DSP_DYNDEP	RESERVED	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:19	RESERVED		R	0x00
18	C2C_DYNDEP	Dynamic dependency towards C2C clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
17	RESERVED		R	1
16	ALWONCORE_DYNDEP	Dynamic dependency towards ALWONCORE clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
15	L4WKUP_DYNDEP	Dynamic dependency towards L4WKUP clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
14:12	RESERVED		R	0x0
11	SDMA_DYNDEP	Dynamic dependency towards SDMA clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
10	RESERVED		R	0
9	CAM_DYNDEP	Dynamic dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8	DSS_DYNDEP	Dynamic dependency towards DSS clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
7	L3_INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1

Bits	Field Name	Description	Type	Reset
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	L3_1_DYNDEP	Dynamic dependency towards L3_1 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
4	MEMIF_DYNDEP	Dynamic dependency towards MEMIF clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
3:2	RESERVED		R	0x0
1	DSP_DYNDEP	Dynamic dependency towards DSP clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
0	RESERVED		R	0

**Table 3-1230. Register Call Summary for Register CM\_L4CFG\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[9\]](#)
- [RESTORE\\_CM2 Register Description: \[10\]](#)

**Table 3-1231. CM\_L4CFG\_L4\_CFG\_CLKCTRL**

<b>Address Offset</b>	0x0000 0620	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register manages the L4_CFG clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED										MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1232. Register Call Summary for Register CM\_L4CFG\_L4\_CFG\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- PRCM Register Manual
- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1233. CM\_L4CFG\_SPINLOCK\_CLKCTRL**

<b>Address Offset</b>	0x0000 0628	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register manages the HW_SEM clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED										MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1234. Register Call Summary for Register CM\_L4CFG\_SPINLOCK\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- PRCM Register Manual
- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1235. CM\_L4CFG\_MAILBOX\_CLKCTRL**

<b>Address Offset</b>	0x0000 0630		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the MAILBOX clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED												MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Returns	0x1

**Table 3-1236. Register Call Summary for Register CM\_L4CFG\_MAILBOX\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1237. CM\_L4CFG\_SAR\_ROM\_CLKCTRL**

<b>Address Offset</b>	0x0000 0638		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the SAR_ROM clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED												MODULEMODE				



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1238. Register Call Summary for Register CM\_L4CFG\_SAR\_ROM\_CLKCTRL**

Clock Management Functional Description

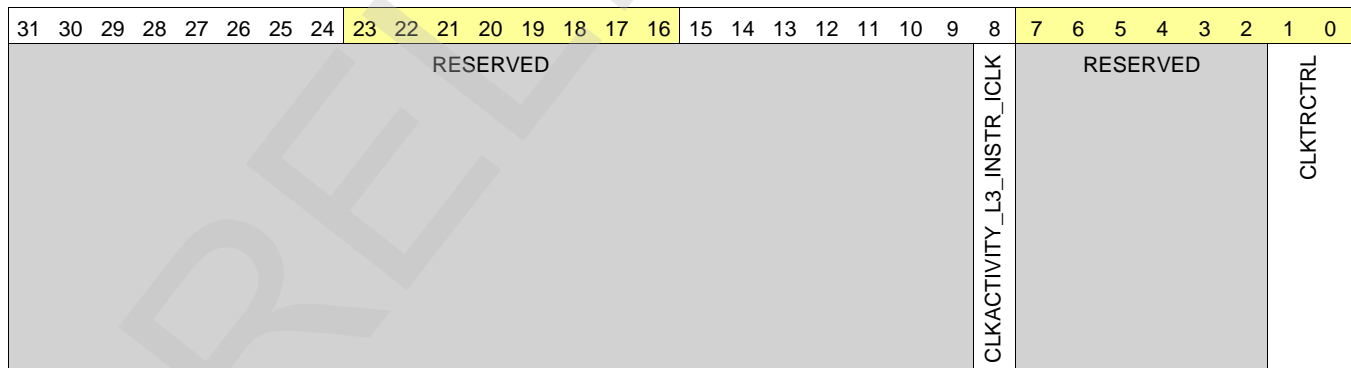
- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1239. CM\_L3INSTR\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0700	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	R		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_L3_INSTR_ICLK	This field indicates the state of the L3_INSTR_GICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the L3INSTR clock domain.  Read 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	R Rreturns 1s	0x3

**Table 3-1240. Register Call Summary for Register CM\_L3INSTR\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\]](#)

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1241. CM\_L3INSTR\_L3\_3\_CLKCTRL**

<b>Address Offset</b>	0x0000 0720		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the L3_3 clocks. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED												MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status.  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved  Read 0x3: Reserved	RW	0x1

**Table 3-1242. Register Call Summary for Register CM\_L3INSTR\_L3\_3\_CLKCTRL**

Clock Management Functional Description  
 • [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual  
 • [CORE\\_CM2 Register Summary: \[2\]](#)  
 • [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1243. CM\_L3INSTR\_L3\_INSTR\_CLKCTRL**

<b>Address Offset</b>	0x0000 0728	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>		
<b>Description</b>	This register manages the L3 INSTRUMENTATION clocks. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED										MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status.  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved  Read 0x3: Reserved	RW	0x1

**Table 3-1244. Register Call Summary for Register CM\_L3INSTR\_L3\_INSTR\_CLKCTRL**

Clock Management Functional Description  
 • [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual  
 • [CORE\\_CM2 Register Summary: \[2\]](#)  
 • [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1245. CM\_L3INSTR\_INTRCONN\_WP1\_CLKCTRL**

<b>Address Offset</b>	0x0000 0740		
<b>Physical Address</b>	See <a href="#">Table 3-1172</a>	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the INTRCONN WP1 clocks. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. Read 0x3: Module is disabled and cannot be accessed Read 0x2: Module is in idle mode (only Interconnect part). It is functional if using separate functional clock Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x0: Module is fully functional, including Interconnect	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any Interconnect access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any Interconnect access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1246. Register Call Summary for Register CM\_L3INSTR\_INTRCONN\_WP1\_CLKCTRL**

PRCM Register Manual

- [CORE\\_CM2 Register Summary: \[0\]](#)
- [RESTORE\\_CM2 Register Description: \[1\]](#)

### 3.11.34 IVAHD\_CM2 Registers

#### 3.11.34.1 IVAHD\_CM2 Register Summary

**Table 3-1247. IVAHD\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	IVAHD_CM2 L4 Base Address
<a href="#">CM_IVAHD_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A00 8F00
<a href="#">CM_IVAHD_STATICDEP</a>	RW	32	0x0000 0004	0x4A00 8F04
<a href="#">CM_IVAHD_DYNAMICDEP</a>	R	32	0x0000 0008	0x4A00 8F08
<a href="#">CM_IVAHD_IVAHD_CLKCTRL</a>	RW	32	0x0000 0020	0x4A00 8F20

**Table 3-1247. IVAHD\_CM2 Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	IVAHD_CM2 L4 Base Address
CM_IVAHD_SL2_CLKCTRL	RW	32	0x0000 0028	0x4A00 8F28

**3.11.34.2 IVAHD\_CM2 Register Description**

**Table 3-1248. CM\_IVAHD\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-1247</a>	<b>Instance</b>	IVAHD_CM2
<b>Description</b>	This register enables the IVAHD domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_IVAHD_CLK	RESERVED						CLKTRCTRL								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_IVAHD_CLK	This field indicates the state of the IVAHD_CLK clock input of the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the IVAHD clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

**Table 3-1249. Register Call Summary for Register CM\_IVAHD\_CLKSTCTRL**

Reset Management Functional Description

- [IVAHD Subsystem Software Warm Reset Sequence: \[0\]](#)

Clock Management Functional Description

- [Clock Domain Modes: \[1\] \[2\]](#)

PRCM Register Manual

- [IVAHD\\_CM2 Register Summary: \[3\]](#)

**Table 3-1250. CM\_IVAHD\_STATICDEP**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 3-1247</a>	<b>Instance</b>	IVAHD_CM2
<b>Description</b>	This register controls the static domain dependencies from IVAHD domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L3_2_STATDEP		L3_1_STATDEP		MEMIF_STATDEP		RESERVED									

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
3:0	RESERVED		R	0x0

**Table 3-1251. Register Call Summary for Register CM\_IVAHD\_STATICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [IVAHD\\_CM2 Register Summary: \[3\]](#)

**Table 3-1252. CM\_IVAHD\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 3-1247</a>	<b>Instance</b>	IVAHD_CM2
<b>Description</b>	This register controls the dynamic domain dependencies from IVAHD domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L3_2_DYNDP		RESERVED													

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x0: Dependency is disabled	R	0
5:0	RESERVED		R	0x00

**Table 3-1253. Register Call Summary for Register CM\_IVAHD\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\]](#)

PRCM Register Manual

- [IVAHD\\_CM2 Register Summary: \[1\]](#)

**Table 3-1254. CM\_IVAHD\_IVAHD\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020	
<b>Physical Address</b>	See <a href="#">Table 3-1247</a>	<b>Instance</b> IVAHD_CM2
<b>Description</b>	This register manages the IVAHD clocks.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STBYST	IDLEST	RESERVED																MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1255. Register Call Summary for Register CM\_IVAHD\_IVAHD\_CLKCTRL**

Reset Management Functional Description

- [IVAHD Subsystem Software Warm Reset Sequence: \[0\] \[1\]](#)

Clock Management Functional Description

- [Clock Domain Module Attributes: \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [IVAHD\\_CM2 Register Summary: \[5\]](#)

**Table 3-1256. CM\_IVAHD\_SL2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	IVAHD_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1247</a>		
<b>Description</b>	This register manages the SL2 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved  Read 0x3: Reserved	RW	0x0

**Table 3-1257. Register Call Summary for Register CM\_IVAHD\_SL2\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [IVAHD\\_CM2 Register Summary: \[2\]](#)



### 3.11.35 CAM\_CM2 Registers

#### 3.11.35.1 CAM\_CM2 Register Summary

Table 3-1258. CAM\_CM2 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CAM_CM2 L4 Base Address
CM_CAM_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9000
CM_CAM_STATICDEP	RW	32	0x0000 0004	0x4A00 9004
CM_CAM_DYNAMICDEP	R	32	0x0000 0008	0x4A00 9008
CM_CAM_ISS_CLKCTRL	RW	32	0x0000 0020	0x4A00 9020
CM_CAM_FDIF_CLKCTRL	RW	32	0x0000 0028	0x4A00 9028

#### 3.11.35.2 CAM\_CM2 Register Description

Table 3-1259. CM\_CAM\_CLKSTCTRL

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-1258</a>	<b>Instance</b>	CAM_CM2
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_FDIF_FCLK			CLKACTIVITY_CAM_PHY_CTRL_CLK			CLKACTIVITY_ISS_CLK			RESERVED					CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10	CLKACTIVITY_FDIF_FCLK	This field indicates the state of the FDIF_FCLK clock input of the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
9	CLKACTIVITY_CAM_PHY_CTRL_CLK	This field indicates the state of the CAM_PHY_CTRL_CLK clock input of the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
8	CLKACTIVITY_ISS_CLK	This field indicates the state of the ISS_CLK clock input of the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the CAM clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

**Table 3-1260. Register Call Summary for Register CM\_CAM\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CAM\\_CM2 Register Summary: \[4\]](#)

**Table 3-1261. CM\_CAM\_STATICDEP**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	CAM_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1258</a>		
<b>Description</b>	This register controls the static domain dependencies from CAM domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																											L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	RESERVED	I/AHD_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
3	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1:0	RESERVED		R	0x0

**Table 3-1262. Register Call Summary for Register CM\_CAM\_STATICDEP**

- Clock Management Functional Description
- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)
- PRCM Register Manual
- [CAM\\_CM2 Register Summary: \[4\]](#)

**Table 3-1263. CM\_CAM\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	CAM_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1258</a>		
<b>Description</b>	This register controls the dynamic domain dependencies from CAM domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L3_2_DYNDEP	RESERVED														

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x0: Dependency is disabled	R	0
5:0	RESERVED		R	0x00

**Table 3-1264. Register Call Summary for Register CM\_CAM\_DYNAMICDEP**

- Clock Management Functional Description
- [Clock Domain Dependency: \[0\]](#)
- PRCM Register Manual
- [CAM\\_CM2 Register Summary: \[1\]](#)

**Table 3-1265. CM\_CAM\_ISS\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	CAM_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1258</a>		
<b>Description</b>	This register manages the ISS clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED												STBYST		IDLEST		RESERVED								OPTFCLKEN_CTRLCLK		RESERVED					MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_CTRLCLK	Optional functional clock control for CAM_PHY_CTRL_GCLK 96Mhz clock. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1266. Register Call Summary for Register CM\_CAM\_ISS\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [CAM\\_CM2 Register Summary: \[3\]](#)

**Table 3-1267. CM\_CAM\_FDIF\_CLKCTRL**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 3-1258</a>	<b>Instance</b>	CAM_CM2
<b>Description</b>	This register manages the FDIF clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_FCLK	RESERVED				STBYST	IDLEST	RESERVED												MODULEMODE								

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	CLKSEL_FCLK	Select the ration of FDIF_FCLK to FUNC_128M_CLK 0x0: FDIF_FCLK is divide by 1 of FUNC_128_CLK 0x1: FDIF_FCLK is divide by 2 of FUNC_128_CLK 0x2: FDIF_FCLK is divide by 4 of FUNC_128_CLK 0x3: Reserved	RW	0x0
23:19	RESERVED		R	0x00
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1268. Register Call Summary for Register CM\_CAM\_FDIF\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [CAM\\_CM2 Register Summary: \[4\]](#)

### 3.11.36 DSS\_CM2 Registers

#### 3.11.36.1 DSS\_CM2 Register Summary

**Table 3-1269. DSS\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSS_CM2 L4 Base Address
<a href="#">CM_DSS_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A00 9100
<a href="#">CM_DSS_STATICDEP</a>	RW	32	0x0000 0004	0x4A00 9104
<a href="#">CM_DSS_DYNAMICDEP</a>	R	32	0x0000 0008	0x4A00 9108
<a href="#">CM_DSS_DSS_CLKCTRL</a>	RW	32	0x0000 0020	0x4A00 9120
RESERVED	RW	32	0x0000 0028	0x4A00 9128

#### 3.11.36.2 DSS\_CM2 Register Description

**Table 3-1270. CM\_DSS\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-1269</a>	<b>Instance</b>	DSS_CM2
<b>Description</b>	This register enables the DSS domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_HDMI_PHY_48M_FCLK CLKACTIVITY_DSS_ALWON_SYS_CLK CLKACTIVITY_DSS_FCLK CLKACTIVITY_DSS_L3_ICLK				RESERVED							CLKTRCTRL				

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11	CLKACTIVITY_HDMI_PHY_48M_FCLK	This field indicates the state of the HDMI_PHY_48MHz_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
10	CLKACTIVITY_DSS_ALWON_SYS_CLK	This field indicates the state of the DSS_ALWON_SYS_CLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
9	CLKACTIVITY_DSS_FCLK	This field indicates the state of the DSS_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_DSS_L3_ICLK	This field indicates the state of the DSS_L3_ICLK (and DSS_L4_ICLK) clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the DSS clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

**Table 3-1271. Register Call Summary for Register CM\_DSS\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [DSS\\_CM2 Register Summary: \[5\]](#)

**Table 3-1272. CM\_DSS\_STATICDEP**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	DSS_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1269</a>		
<b>Description</b>	This register controls the static domain dependencies from DSS domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L3_2_STATDEP		L3_1_STATDEP		MEMIF_STATDEP		RESERVED		IVAHD_STATDEP		RESERVED					

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain Read 0x1: Dependency is enabled	R Returns 1s	1

Bits	Field Name	Description	Type	Reset
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
3	RESERVED		R	0
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1:0	RESERVED		R	0x0

**Table 3-1273. Register Call Summary for Register CM\_DSS\_STATICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [DSS\\_CM2 Register Summary: \[4\]](#)

**Table 3-1274. CM\_DSS\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	DSS_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1269</a>		
<b>Description</b>	This register controls the dynamic domain dependencies from DSS domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L3_1_DYNDEP	RESERVED														

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5	L3_1_DYNDEP	Dynamic dependency towards L3_1 domain Read 0x0: Dependency is disabled	R	0
4:0	RESERVED		R	0x00

**Table 3-1275. Register Call Summary for Register CM\_DSS\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\]](#)

PRCM Register Manual

- [DSS\\_CM2 Register Summary: \[1\]](#)



**Table 3-1276. CM\_DSS\_DSS\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020		
<b>Physical Address</b>	See <a href="#">Table 3-1269</a>	<b>Instance</b>	DSS_CM2
<b>Description</b>	This register manages the DSS clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STBYST		IDLEST		RESERVED				OPTFCLKEN_TV_FCLK		OPTFCLKEN_SYS_CLK		OPTFCLKEN_48MHZ_CLK		OPTFCLKEN_DSSCLK		RESERVED					MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:12	RESERVED		R	0x0
11	OPTFCLKEN_TV_FCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
10	OPTFCLKEN_SYS_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
9	OPTFCLKEN_48MHZ_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
8	OPTFCLKEN_DSSCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	<p>Control the way mandatory clocks are managed.</p> <p>0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).</p> <p>Read 0x1: Reserved</p> <p>0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen.</p> <p>Read 0x3: Reserved</p>	RW	0x0

**Table 3-1277. Register Call Summary for Register CM\_DSS\_DSS\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [DSS\\_CM2 Register Summary: \[4\]](#)

### 3.11.37 SGX\_CM2 Registers

#### 3.11.37.1 SGX\_CM2 Register Summary

**Table 3-1278. SGX\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SGX_CM2 L4 Base Address
<a href="#">CM_SGX_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A00 9200
<a href="#">CM_SGX_STATICDEP</a>	RW	32	0x0000 0004	0x4A00 9204
<a href="#">CM_SGX_DYNAMICDEP</a>	R	32	0x0000 0008	0x4A00 9208
<a href="#">CM_SGX_SGX_CLKCTRL</a>	RW	32	0x0000 0020	0x4A00 9220

#### 3.11.37.2 SGX\_CM2 Register Description

**Table 3-1279. CM\_SGX\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-1278</a>	<b>Instance</b>	SGX_CM2
<b>Description</b>	This register enables the SGX domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_SGX_FCLK		CLKACTIVITY_SGX_L3_ICLK		RESERVED										CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	CLKACTIVITY_SGX_FCLK	This field indicates the state of the SGX_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_SGX_L3_ICLK	This field indicates the state of the SGX_L3_ICLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the SGX clock domain.  0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.  0x1: SW_SLEEP: Start a software forced sleep transition on the domain.  0x2: SW_WKUP: Start a software forced wake-up transition on the domain.  0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

**Table 3-1280. Register Call Summary for Register CM\_SGX\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [SGX\\_CM2 Register Summary: \[3\]](#)

**Table 3-1281. CM\_SGX\_STATICDEP**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	SGX_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1278</a>		
<b>Description</b>	This register controls the static domain dependencies from SGX domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	RESERVED	IVAHD_STATDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

Bits	Field Name	Description	Type	Reset
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
3	RESERVED		R	0
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1:0	RESERVED		R	0x0

**Table 3-1282. Register Call Summary for Register CM\_SGX\_STATICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [SGX\\_CM2 Register Summary: \[4\]](#)

**Table 3-1283. CM\_SGX\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	SGX_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1278</a>		
<b>Description</b>	This register controls the dynamic domain dependencies from SGX domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L3_2_DYNDEP	RESERVED														

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x0: Dependency is disabled	R	0
5:0	RESERVED		R	0x00

**Table 3-1284. Register Call Summary for Register CM\_SGX\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\]](#)

PRCM Register Manual

- [SGX\\_CM2 Register Summary: \[1\]](#)

**Table 3-1285. CM\_SGX\_SGX\_CLKCTRL**

<b>Address Offset</b>	0x0000 0020		
<b>Physical Address</b>	See <a href="#">Table 3-1278</a>	<b>Instance</b>	SGX_CM2
<b>Description</b>	This register manages the SGX clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKSEL_SGX_FCLK	RESERVED				STBYST	IDLEST	RESERVED												MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL_SGX_FCLK	Select the source of SGX_FCLK 0x0: Functional clock is sourced from DPLL_CORE HSDIVIDER 0x1: Functional clock is sourced from DPLL_PER HSDIVIDER	RW	0
23:19	RESERVED		R	0x00
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1286. Register Call Summary for Register CM\_SGX\_SGX\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [SGX\\_CM2 Register Summary: \[4\]](#)

### 3.11.38 L3INIT\_CM2 Registers

#### 3.11.38.1 L3INIT\_CM2 Register Summary

**Table 3-1287. L3INIT\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	L3INIT_CM2 L4 Base Address
CM_L3INIT_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9300
CM_L3INIT_STATICDEP	RW	32	0x0000 0004	0x4A00 9304
CM_L3INIT_DYNAMICDEP	R	32	0x0000 0008	0x4A00 9308
CM_L3INIT_HSMCM1_CLKCTRL	RW	32	0x0000 0028	0x4A00 9328
CM_L3INIT_HSMCM2_CLKCTRL	RW	32	0x0000 0030	0x4A00 9330
CM_L3INIT_HSI_CLKCTRL	RW	32	0x0000 0038	0x4A00 9338
RESERVED	RW	32	0x0000 0040	0x4A00 9340
CM_L3INIT_HSUSBHOST_CLKCTRL	RW	32	0x0000 0058	0x4A00 9358
CM_L3INIT_HSUSBOTG_CLKCTRL	RW	32	0x0000 0060	0x4A00 9360
CM_L3INIT_HSUSBTLL_CLKCTRL	RW	32	0x0000 0068	0x4A00 9368
CM_L3INIT_P1500_CLKCTRL	R	32	0x0000 0078	0x4A00 9378
RESERVED	RW	32	0x0000 0080	0x4A00 9380
RESERVED	RW	32	0x0000 0088	0x4A00 9388
RESERVED	RW	32	0x0000 0090	0x4A00 9390
RESERVED	RW	32	0x0000 0098	0x4A00 9398
RESERVED	RW	32	0x0000 00A8	0x4A00 93A8
RESERVED	RW	32	0x0000 00C0	0x4A00 93C0
RESERVED	RW	32	0x0000 00C8	0x4A00 93C8
CM_L3INIT_FSUSB_CLKCTRL	RW	32	0x0000 00D0	0x4A00 93D0
CM_L3INIT_USBPHY_CLKCTRL	RW	32	0x0000 00E0	0x4A00 93E0

#### 3.11.38.2 L3INIT\_CM2 Register Description

**Table 3-1288. CM\_L3INIT\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	L3INIT_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>		
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKACTIVITY_INIT_32K_FCLK	RESERVED	CLKACTIVITY_INIT_60M_P2_FCLK	CLKACTIVITY_INIT_60M_P1_FCLK	CLKACTIVITY_HSIC_P2_FCLK	CLKACTIVITY_HSIC_P1_FCLK	CLKACTIVITY_UTMI_ROOT_FCLK	CLKACTIVITY_TLL_CH2_FCLK	CLKACTIVITY_TLL_CH1_FCLK	CLKACTIVITY_TLL_CH0_FCLK	CLKACTIVITY_HSIC_P2_480M_FCLK	CLKACTIVITY_HSIC_P1_480M_FCLK	RESERVED	CLKACTIVITY_INIT_HSMCM2_FCLK	CLKACTIVITY_INIT_HSMCM1_FCLK	CLKACTIVITY_INIT_HSI_FCLK	CLKACTIVITY_USB_DPLL_HS_CLK	CLKACTIVITY_USB_DPLL_CLK	CLKACTIVITY_INIT_48MC_FCLK	CLKACTIVITY_INIT_48M_FCLK	RESERVED	CLKACTIVITY_INIT_L4_ICLK	CLKACTIVITY_INIT_L3_ICLK	RESERVED					CLKCTRL			

Bits	Field Name	Description	Type	Reset
31	CLKACTIVITY_INIT_32K_FCLK	This field indicates the state of the INIT_32K_FCLK clock in the domain. [warm reset insensitive]  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing  Read 0x0: Corresponding clock is definitely gated	R	0
30	RESERVED		R	0
29	CLKACTIVITY_INIT_60M_P2_FCLK	This field indicates the state of the INIT_60M_P2_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
28	CLKACTIVITY_INIT_60M_P1_FCLK	This field indicates the state of the INIT_60M_P1_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
27	CLKACTIVITY_HSIK_P2_FCLK	This field indicates the state of the HSIK_P2_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
26	CLKACTIVITY_HSIK_P1_FCLK	This field indicates the state of the HSIK_P1_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
25	CLKACTIVITY_UTMI_ROOT_FCLK	This field indicates the state of the UTMI_ROOT_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
24	CLKACTIVITY_TLL_CH2_FCLK	This field indicates the state of the TLL_CH2_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
23	CLKACTIVITY_TLL_CH1_FCLK	This field indicates the state of the TLL_CH1_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
22	CLKACTIVITY_TLL_CH0_FCLK	This field indicates the state of the TLL_CH0_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
21	CLKACTIVITY_HSIK_P2_480M_FCLK	This field indicates the state of the HSIK_P2_480M_FCLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
20	CLKACTIVITY_HSIC_P1_480M_FCLK	This field indicates the state of the HSIC_P1_480M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
19	RESERVED	Reserved	R	0
18	CLKACTIVITY_INIT_HSMC2_FCLK	This field indicates the state of the INIT_HSMC2_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
17	CLKACTIVITY_INIT_HSMC1_FCLK	This field indicates the state of the INIT_HSMC1_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
16	CLKACTIVITY_INIT_HSI_FCLK	This field indicates the state of the INIT_HSI_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
15	CLKACTIVITY_USB_DPLL_HS_CLK	This field indicates the state of the USB_DPLL_HS_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
14	CLKACTIVITY_USB_DPLL_CLK	This field indicates the state of the USB_DPLL_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
13	CLKACTIVITY_INIT_48MC_FCLK	This field indicates the state of the INIT_48MC_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
12	CLKACTIVITY_INIT_48M_FCLK	This field indicates the state of the INIT_48M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
11:10	RESERVED	Reserved	R	0x0
9	CLKACTIVITY_INIT_L4_ICLK	This field indicates the state of the L4_INIT_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_INIT_L3_ICLK	This field indicates the state of the L3_INIT_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED	Reserved	R	0x00



Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the L3INIT clock domain.  0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.  0x1: SW_SLEEP: Start a software forced sleep transition on the domain.  0x2: SW_WKUP: Start a software forced wake-up transition on the domain.  0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1289. Register Call Summary for Register CM\_L3INIT\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[19\]](#)
- [RESTORE\\_CM2 Register Description: \[20\]](#)

**Table 3-1290. CM\_L3INIT\_STATICDEP**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	L3INIT_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>		
<b>Description</b>	This register controls the static domain dependencies from L3INIT domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																L4WKUP_STATDEP	RESERVED	L4PER_STATDEP	L4CFG_STATDEP	RESERVED								L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	ABE_STATDEP	IVAHD_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	L4WKUP_STATDEP	Static dependency towards L4WKUP clock domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
14	RESERVED		R	0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain  0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
11:7	RESERVED		R	0x00
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain  Read 0x1: Dependency is enabled	R Rreturns 1s	1

Bits	Field Name	Description	Type	Reset
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
3	ABE_STATDEP	Static dependency towards ABE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1:0	RESERVED		R	0x0

**Table 3-1291. Register Call Summary for Register CM\_L3INIT\_STATICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[7\]](#)

**Table 3-1292. CM\_L3INIT\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	L3INIT_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>		
<b>Description</b>	This register controls the dynamic domain dependencies from L3INIT domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L3_2_DYNDEP		L3_1_DYNDEP		RESERVED											

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x0: Dependency is disabled	R	0
5	L3_1_DYNDEP	Dynamic dependency towards L3_1 clock domain Read 0x0: Dependency is disabled	R	0
4:0	RESERVED		R	0x00

**Table 3-1293. Register Call Summary for Register CM\_L3INIT\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[2\]](#)

**Table 3-1294. CM\_L3INIT\_HSMMC1\_CLKCTRL**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>	<b>Instance</b>	L3INIT_CM2
<b>Description</b>	This register manages the MMC1 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKSEL	RESERVED				STBYST	IDLEST	RESERVED												MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Selects the source of the functional clock. 0x0: 64MHz clock derived from DPLL_PER is selected 0x1: 96MHz clock derived from DPLL_PER is selected	RW	0
23:19	RESERVED		R	0x00
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1295. Register Call Summary for Register CM\_L3INIT\_HSMMC1\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[4\]](#)

**Table 3-1296. CM\_L3INIT\_HSMC2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>	<b>Instance</b>	L3INIT_CM2
<b>Description</b>	This register manages the MMC2 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKSEL	RESERVED				STBYST	IDLEST	RESERVED												MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Selects the source of the functional clock. 0x0: 64MHz clock derived from DPLL_PER is selected 0x1: 96MHz clock derived from DPLL_PER is selected	RW	0
23:19	RESERVED		R	0x00
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1297. Register Call Summary for Register CM\_L3INIT\_HSMC2\_CLKCTRL**

## Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\] \[3\]](#)

## PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[4\]](#)

**Table 3-1298. CM\_L3INIT\_HSI\_CLKCTRL**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>	<b>Instance</b>	L3INIT_CM2
<b>Description</b>	This register manages the HSI clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL	RESERVED				STBYST	IDLEST	RESERVED										MODULEMODE										

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:24	CLKSEL	Selects the functional clock source. 0x0: HSI_FCLK is divide by 1 of 192MHz clock 0x1: HSI_FCLK is divide by 2 of 192MHz clock 0x2: HSI_FCLK is divide by 4 of 192MHz clock 0x3: Reserved	RW	0x0
23:19	RESERVED		R	0x00
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain Read 0x: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1299. Register Call Summary for Register CM\_L3INIT\_HSI\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[5\]](#)

**Table 3-1300. CM\_L3INIT\_HSUSBHOST\_CLKCTRL**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>	<b>Instance</b>	L3INIT_CM2
<b>Description</b>	This register manages the USB_HOST_HS clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						CLKSEL_UTMI_P2	CLKSEL_UTMI_P1	RESERVED						STBYST	IDLEST	OPTFCLKEN_FUNC48MCLK	OPTFCLKEN_HSIC480M_P2_CLK	OPTFCLKEN_HSIC480M_P1_CLK	OPTFCLKEN_HSIC60M_P2_CLK	OPTFCLKEN_HSIC60M_P1_CLK	OPTFCLKEN_UTMI_P3_CLK	OPTFCLKEN_UTMI_P2_CLK	OPTFCLKEN_UTMI_P1_CLK	RESERVED			SAR_MODE	RESERVED		MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25	CLKSEL_UTMI_P2	Selects the source of the functional clock for UTMI Port2 on USB Host  0x0: The functional clock is provided by the internal clock source 0x1: The functional clock is provided by an external PHY through an I/O pad.	RW	0
24	CLKSEL_UTMI_P1	Selects the source of the functional clock for UTMI Por1 on USB Host  0x0: The functional clock is provided by the internal clock source 0x1: The functional clock is provided by an external PHY through an I/O pad.	RW	0
23:19	RESERVED		R	0x00
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15	OPTFCLKEN_FUNC48MCLK	USB-HOST optional clock control: FUNC48MCLK  0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
14	OPTFCLKEN_HSIC480M_P2_C LK	USB-HOST optional clock control: HSIC480M_P2_CLK  0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0

Bits	Field Name	Description	Type	Reset
13	OPTFCLKEN_HSIC480M_P1_CLK	USB-HOST optional clock control: HSIC480M_P1_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
12	OPTFCLKEN_HSIC60M_P2_CLK	USB-HOST optional clock control: HSIC60M_P2_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
11	OPTFCLKEN_HSIC60M_P1_CLK	USB-HOST optional clock control: HSIC60M_P1_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
10	OPTFCLKEN_UTMI_P3_CLK	USB-HOST optional clock control: UTMI_P3_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
9	OPTFCLKEN_UTMI_P2_CLK	USB-HOST optional clock control: UTMI_P2_CLK when CLKSEL_UTMI_P2 is 0 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
8	OPTFCLKEN_UTMI_P1_CLK	USB-HOST optional clock control: UTMI_P1_CLK when CLKSEL_UTMI_P1 is 0 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:5	RESERVED		R	0x0
4	SAR_MODE	SAR mode control for the module. Shall not be modify except if module is disabled. 0x0: SAR mode is disabled 0x1: SAR mode is enabled	RW	0
3:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1301. Register Call Summary for Register CM\_L3INIT\_HSUSBHOST\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\] \[1\]](#)
- [Clock Domain Module Attributes: \[2\] \[3\] \[4\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[5\]](#)

**Table 3-1302. CM\_L3INIT\_HSUSBOTG\_CLKCTRL**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>	<b>Instance</b>	L3INIT_CM2
<b>Description</b>	This register manages the USB_OTG_HS clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL_60M	RESERVED				STBYST	IDLEST	RESERVED				OPTFCLKEN_XCLK	RESERVED				MODULEMODE								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL_60M	Selects the source of the 60MHz functional clock. 0x0: The 60MHz clock is sourced from on die UTMI PHY 0x1: The 60MHz clock is sourced from the external ULPI PHY	RW	0
23:19	RESERVED		R	0x00
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_XCLK	USB_OTG optional clock control: XCLK (60MHz clock) 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0



**Table 3-1303. Register Call Summary for Register CM\_L3INIT\_HSUSBOTG\_CLKCTRL**

- Clock Management Functional Description
- [CM Clock Source: \[0\]](#)
  - [Clock Domain Module Attributes: \[1\] \[2\] \[3\]](#)
- PRCM Register Manual
- [L3INIT\\_CM2 Register Summary: \[4\]](#)

**Table 3-1304. CM\_L3INIT\_HSUSBTLL\_CLKCTRL**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	L3INIT_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>		
<b>Description</b>	This register manages the USB_TLL clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST				RESERVED				OPTFCLKEN_USB_CH2_CLK	OPTFCLKEN_USB_CH1_CLK	OPTFCLKEN_USB_CH0_CLK	RESERVED		SAR_MODE	RESERVED	MODULEMODE								

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:11	RESERVED		R	0x00
10	OPTFCLKEN_USB_CH2_CLK	USB-HOST optional clock control: USB_CH2_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
9	OPTFCLKEN_USB_CH1_CLK	USB-HOST optional clock control: USB_CH1_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
8	OPTFCLKEN_USB_CH0_CLK	USB-HOST optional clock control: USB_CH0_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:5	RESERVED		R	0x0
4	SAR_MODE	SAR mode control for the module. Shall not be modify except if module is disabled. 0x0: SAR mode is disabled 0x1: SAR mode is enabled	RW	0
3:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	<p>Control the way mandatory clocks are managed.</p> <p>0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).</p> <p>0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.</p> <p>Read 0x2: Reserved</p> <p>Read 0x3: Reserved</p>	RW	0x0

**Table 3-1305. Register Call Summary for Register CM\_L3INIT\_HSUSBTLL\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1306. CM\_L3INIT\_P1500\_CLKCTRL**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	L3INIT_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>		
<b>Description</b>	This register manages the P1500 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														STBYST	IDLEST	RESERVED										MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	<p>Module standby status. [warm reset insensitive]</p> <p>Read 0x0: Module is functional (not in standby)</p> <p>Read 0x1: Module is in standby</p>	R	1
17:16	IDLEST	<p>Module idle status. [warm reset insensitive]</p> <p>Read 0x0: Module is fully functional, including INTRCONN</p> <p>Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion</p> <p>Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock</p> <p>Read 0x3: Module is disabled and cannot be accessed</p>	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disabled by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1307. Register Call Summary for Register CM\_L3INIT\_P1500\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[2\]](#)

**Table 3-1308. CM\_L3INIT\_FSUSB\_CLKCTRL**

<b>Address Offset</b>	0x0000 00D0	<b>Instance</b>	L3INIT_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>		
<b>Description</b>	This register manages the USB_HOST_FS clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														STBYST	IDLEST	RESERVED										MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disabled by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1309. Register Call Summary for Register CM\_L3INIT\_FSUSB\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\] \[2\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[3\]](#)

**Table 3-1310. CM\_L3INIT\_USBPHY\_CLKCTRL**

<b>Address Offset</b>	0x0000 00E0	<b>Instance</b>	L3INIT_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1287</a>		
<b>Description</b>	This register manages the USBPHY clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST		RESERVED								OPTFCKEN_PHY_48M	RESERVED					MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCKEN_PHY_48M	USBPHY optional clock control: PHY_48M  0x0: Optional functional clock is disabled  0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	<p>Control the way mandatory clocks are managed.</p> <p>0x0: Module is disabled by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).</p> <p>0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.</p> <p>Read 0x2: Reserved</p> <p>Read 0x3: Reserved</p>	RW	0x0

**Table 3-1311. Register Call Summary for Register CM\_L3INIT\_USBPHY\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L3INIT\\_CM2 Register Summary: \[2\]](#)

### 3.11.39 L4PER\_CM2 Registers

#### 3.11.39.1 L4PER\_CM2 Register Summary

**Table 3-1312. L4PER\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_CM2 L4 Base Address
<a href="#">CM_L4PER_CLKSTCTRL</a>	RW	32	0x0000 0000	0x4A00 9400
<a href="#">CM_L4PER_DYNAMICDEP</a>	RW	32	0x0000 0008	0x4A00 9408
RESERVED	RW	32	0x0000 0020	0x4A00 9420
<a href="#">CM_L4PER_GPTIMER10_CLKCTRL</a>	RW	32	0x0000 0028	0x4A00 9428
<a href="#">CM_L4PER_GPTIMER11_CLKCTRL</a>	RW	32	0x0000 0030	0x4A00 9430
<a href="#">CM_L4PER_GPTIMER2_CLKCTRL</a>	RW	32	0x0000 0038	0x4A00 9438
<a href="#">CM_L4PER_GPTIMER3_CLKCTRL</a>	RW	32	0x0000 0040	0x4A00 9440
<a href="#">CM_L4PER_GPTIMER4_CLKCTRL</a>	RW	32	0x0000 0048	0x4A00 9448
<a href="#">CM_L4PER_GPTIMER9_CLKCTRL</a>	RW	32	0x0000 0050	0x4A00 9450
<a href="#">CM_L4PER_ELM_CLKCTRL</a>	R	32	0x0000 0058	0x4A00 9458
<a href="#">CM_L4PER_GPIO2_CLKCTRL</a>	RW	32	0x0000 0060	0x4A00 9460
<a href="#">CM_L4PER_GPIO3_CLKCTRL</a>	RW	32	0x0000 0068	0x4A00 9468
<a href="#">CM_L4PER_GPIO4_CLKCTRL</a>	RW	32	0x0000 0070	0x4A00 9470
<a href="#">CM_L4PER_GPIO5_CLKCTRL</a>	RW	32	0x0000 0078	0x4A00 9478
<a href="#">CM_L4PER_GPIO6_CLKCTRL</a>	RW	32	0x0000 0080	0x4A00 9480
<a href="#">CM_L4PER_HDQ1W_CLKCTRL</a>	RW	32	0x0000 0088	0x4A00 9488
RESERVED	RW	32	0x0000 0090	0x4A00 9490
RESERVED	RW	32	0x0000 0098	0x4A00 9498
<a href="#">CM_L4PER_I2C1_CLKCTRL</a>	RW	32	0x0000 00A0	0x4A00 94A0
<a href="#">CM_L4PER_I2C2_CLKCTRL</a>	RW	32	0x0000 00A8	0x4A00 94A8
<a href="#">CM_L4PER_I2C3_CLKCTRL</a>	RW	32	0x0000 00B0	0x4A00 94B0
<a href="#">CM_L4PER_I2C4_CLKCTRL</a>	RW	32	0x0000 00B8	0x4A00 94B8
<a href="#">CM_L4PER_L4PER_CLKCTRL</a>	R	32	0x0000 00C0	0x4A00 94C0
RESERVED	RW	32	0x0000 00D0	0x4A00 94D0
RESERVED	RW	32	0x0000 00D8	0x4A00 94D8

**Table 3-1312. L4PER\_CM2 Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_CM2 L4 Base Address
CM_L4PER_MCBSP4_CLKCTRL	RW	32	0x0000 00E0	0x4A00 94E0
RESERVED	RW	32	0x0000 00E8	0x4A00 94E8
CM_L4PER_MCSP1_CLKCTRL	RW	32	0x0000 00F0	0x4A00 94F0
CM_L4PER_MCSP2_CLKCTRL	RW	32	0x0000 00F8	0x4A00 94F8
CM_L4PER_MCSP3_CLKCTRL	RW	32	0x0000 0100	0x4A00 9500
CM_L4PER_MCSP4_CLKCTRL	RW	32	0x0000 0108	0x4A00 9508
CM_L4PER_MMCS3_CLKCTRL	RW	32	0x0000 0120	0x4A00 9520
CM_L4PER_MMCS4_CLKCTRL	RW	32	0x0000 0128	0x4A00 9528
RESERVED	RW	32	0x0000 0130	0x4A00 9530
CM_L4PER_SLIMBUS2_CLKCTRL	RW	32	0x0000 0138	0x4A00 9538
CM_L4PER_UART1_CLKCTRL	RW	32	0x0000 0140	0x4A00 9540
CM_L4PER_UART2_CLKCTRL	RW	32	0x0000 0148	0x4A00 9548
CM_L4PER_UART3_CLKCTRL	RW	32	0x0000 0150	0x4A00 9550
CM_L4PER_UART4_CLKCTRL	RW	32	0x0000 0158	0x4A00 9558
CM_L4PER_MMCS5_CLKCTRL	RW	32	0x0000 0160	0x4A00 9560
RESERVED	RW	32	0x0000 0168	0x4A00 9568
RESERVED	RW	32	0x0000 0180	0x4A00 9580
RESERVED	RW	32	0x0000 0184	0x4A00 9584
RESERVED	R	32	0x0000 0188	0x4A00 9588
RESERVED	RW	32	0x0000 01A0	0x4A00 95A0
RESERVED	RW	32	0x0000 01A8	0x4A00 95A8
RESERVED	RW	32	0x0000 01B0	0x4A00 95B0
RESERVED	RW	32	0x0000 01B8	0x4A00 95B8
RESERVED	RW	32	0x0000 01C0	0x4A00 95C0
RESERVED	RW	32	0x0000 01C8	0x4A00 95C8
RESERVED	R	32	0x0000 01D8	0x4A00 95D8

**3.11.39.2 L4PER\_CM2 Register Description****Table 3-1313. CM\_L4PER\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register enables the domain power state transition. It controls the hardware supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKACTIVITY_PER_ABE_24M_FCLK	RESERVED	RESERVED	CLKACTIVITY_PER_MCBSP4_FCLK	RESERVED	CLKACTIVITY_PER_96M_FCLK	CLKACTIVITY_PER_48M_FCLK	CLKACTIVITY_PER_32K_FCLK	CLKACTIVITY_PER_24MC_FCLK	CLKACTIVITY_12M_FCLK_FCLK	CLKACTIVITY_GPT9_FCLK	CLKACTIVITY_GPT4_FCLK	CLKACTIVITY_GPT3_FCLK	CLKACTIVITY_GPT2_FCLK	CLKACTIVITY_GPT11_FCLK	CLKACTIVITY_GPT10_FCLK	CLKACTIVITY_L4_PER_ICLK	RESERVED					CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25	CLKACTIVITY_PER_ABE_24M_FCLK	This field indicates the state of the PER_ABE_24M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
24	RESERVED		R	0
23	RESERVED		R	0
22	CLKACTIVITY_PER_MCBSP4_FCLK	This field indicates the state of the PER_MCBSP4_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
21:20	RESERVED		R	0
19	CLKACTIVITY_PER_96M_FCLK	This field indicates the state of the PER_96M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
18	CLKACTIVITY_PER_48M_FCLK	This field indicates the state of the PER_48M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
17	CLKACTIVITY_PER_32K_FCLK	This field indicates the state of the PER_32K_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
16	CLKACTIVITY_PER_24MC_FCLK	This field indicates the state of the PER_24MC_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
15	CLKACTIVITY_12M_FCLK	This field indicates the state of the FUNC_12M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
14	CLKACTIVITY_GPT9_FCLK	This field indicates the state of the DMT9_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
13	CLKACTIVITY_GPT4_FCLK	This field indicates the state of the DMT4_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
12	CLKACTIVITY_GPT3_FCLK	This field indicates the state of the DMT3_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
11	CLKACTIVITY_GPT2_FCLK	This field indicates the state of the DMT2_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
10	CLKACTIVITY_GPT11_FCLK	This field indicates the state of the DMT11_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
9	CLKACTIVITY_GPT10_FCLK	This field indicates the state of the DMT10_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_L4_PER_ICLK	This field indicates the state of the L4_PER_GICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1314. Register Call Summary for Register CM\_L4PER\_CLKSTCTRL**

Clock Management Functional Description

- [Clock Domain Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[15\]](#)
- [RESTORE\\_CM2 Register Description: \[16\]](#)



**Table 3-1315. CM\_L4PER\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b>	L4PER_CM2
<b>Description</b>	This register controls the dynamic domain dependencies from L4PER domain towards 'target' domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED								DSS_DYNDP	L3_INIT_DYNDP	RESERVED													

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:9	RESERVED		R	0x020
8	DSS_DYNDP	Dynamic dependency towards DSS clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
7	L3_INIT_DYNDP	Dynamic dependency towards L3INIT clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
6:0	RESERVED		R	0x00

**Table 3-1316. Register Call Summary for Register CM\_L4PER\_DYNAMICDEP**

Clock Management Functional Description

- [Clock Domain Dependency: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1317. CM\_L4PER\_GPTIMER10\_CLKCTRL**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b>	L4PER_CM2
<b>Description</b>	This register manages the DMTIMER10 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL	RESERVED				IDLEST	RESERVED								MODULEMODE													

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Select the source of the functional clock 0x0: Selects the SYS_CLK as the source 0x1: Selects the 32KHz as the source	RW	0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1318. Register Call Summary for Register CM\_L4PER\_GPTIMER10\_CLKCTRL**

Clock Management Functional Description

- [PRM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[3\]](#)

**Table 3-1319. CM\_L4PER\_GPTIMER11\_CLKCTRL**

<b>Address Offset</b>	0x0000 0030																																																											
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b> L4PER_CM2																																																										
<b>Description</b>	This register manages the DMTIMER11 clocks.																																																											
<b>Type</b>	RW																																																											
<table border="1" style="width:100%; text-align:center; border-collapse: collapse;"> <thead> <tr> <th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> </thead> <tbody> <tr> <td colspan="7">RESERVED</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">CLKSEL</td> <td colspan="4">RESERVED</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">IDLEST</td> <td colspan="12">RESERVED</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">MODULEMODE</td> </tr> </tbody> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED							CLKSEL	RESERVED				IDLEST	RESERVED												MODULEMODE
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																													
RESERVED							CLKSEL	RESERVED				IDLEST	RESERVED												MODULEMODE																																			
Bits	Field Name	Description	Type	Reset																																																								
31:25	RESERVED		R	0x00																																																								
24	CLKSEL	Select the source of the functional clock 0x0: Selects the SYS_CLK as the source 0x1: Selects the 32KHz as the source	RW	0																																																								
23:18	RESERVED		R	0x00																																																								

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1320. Register Call Summary for Register CM\_L4PER\_GPTIMER11\_CLKCTRL**

Clock Management Functional Description

- [PRM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[3\]](#)

**Table 3-1321. CM\_L4PER\_GPTIMER2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the DMTIMER2 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKSEL	RESERVED								IDLEST	RESERVED								MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Select the source of the functional clock 0x0: Selects the SYS_CLK as the source 0x1: Selects the 32KHz as the source	RW	0
23:18	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1322. Register Call Summary for Register CM\_L4PER\_GPTIMER2\_CLKCTRL**

Clock Management Functional Description

- [PRM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[3\]](#)

**Table 3-1323. CM\_L4PER\_GPTIMER3\_CLKCTRL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the DMTIMER3 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKSEL	RESERVED								IDLEST	RESERVED								MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Select the source of the functional clock 0x0: Selects the SYS_CLK as the source 0x1: Selects the 32KHz as the source	RW	0
23:18	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1324. Register Call Summary for Register CM\_L4PER\_GPTIMER3\_CLKCTRL**

Clock Management Functional Description

- [PRM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[3\]](#)

**Table 3-1325. CM\_L4PER\_GPTIMER4\_CLKCTRL**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the DMTIMER4 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKSEL	RESERVED								IDLEST	RESERVED								MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Select the source of the functional clock 0x0: Selects the SYS_CLK as the source 0x1: Selects the 32KHz as the source	RW	0
23:18	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1326. Register Call Summary for Register CM\_L4PER\_GPTIMER4\_CLKCTRL**

Clock Management Functional Description

- [PRM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[3\]](#)

**Table 3-1327. CM\_L4PER\_GPTIMER9\_CLKCTRL**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the DMTIMER9 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED							IDLEST	RESERVED										MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLKSEL	Select the source of the functional clock 0x0: Selects the SYS_CLK as the source 0x1: Selects the 32KHz as the source	RW	0
23:18	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1328. Register Call Summary for Register CM\_L4PER\_GPTIMER9\_CLKCTRL**

Clock Management Functional Description

- [PRM Clock Source: \[0\]](#)
- [Clock Domain Module Attributes: \[1\] \[2\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[3\]](#)

**Table 3-1329. CM\_L4PER\_ELM\_CLKCTRL**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the ELM clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												IDLEST				RESERVED												MODULEMODE			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1330. Register Call Summary for Register CM\_L4PER\_ELM\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1331. CM\_L4PER\_GPIO2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the GPIO2 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																IDLEST	RESERVED								OPTFCLKEN_DBCLK	RESERVED						MODULEMODE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control.  0x0: Optional functional clock is disabled  0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00



Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1332. Register Call Summary for Register CM\_L4PER\_GPIO2\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1333. CM\_L4PER\_GPIO3\_CLKCTRL**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the GPIO3 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																IDLEST		RESERVED						OPTFCLKEN_DBCLK		RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control.  0x0: Optional functional clock is disabled  0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1334. Register Call Summary for Register CM\_L4PER\_GPIO3\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1335. CM\_L4PER\_GPIO4\_CLKCTRL**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the GPIO4 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								OPTFCLKEN_DBCLK	RESERVED								MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control.  0x0: Optional functional clock is disabled  0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disabled by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1336. Register Call Summary for Register CM\_L4PER\_GPIO4\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1337. CM\_L4PER\_GPIO5\_CLKCTRL**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the GPIO5 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								OPTFCLKEN_DBCLK	RESERVED							MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disabled by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1338. Register Call Summary for Register CM\_L4PER\_GPIO5\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1339. CM\_L4PER\_GPIO6\_CLKCTRL**

<b>Address Offset</b>	0x0000 0080	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the GPIO6 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								OPTFCLKEN_DBCLK	RESERVED								MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1340. Register Call Summary for Register CM\_L4PER\_GPIO6\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)
- [RESTORE\\_CM2 Register Description: \[3\]](#)

**Table 3-1341. CM\_L4PER\_HDQ1W\_CLKCTRL**

<b>Address Offset</b>	0x0000 0088	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the HDQ1W clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST		RESERVED										MODULEMODE			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1342. Register Call Summary for Register CM\_L4PER\_HDQ1W\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1343. CM\_L4PER\_I2C1\_CLKCTRL**

<b>Address Offset</b>	0x0000 00A0	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the I2C1 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED										MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1344. Register Call Summary for Register CM\_L4PER\_I2C1\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- PRCM Register Manual
- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1345. CM\_L4PER\_I2C2\_CLKCTRL**

<b>Address Offset</b>	0x0000 00A8	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the I2C2 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								MODULEMODE														

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1346. Register Call Summary for Register CM\_L4PER\_I2C2\_CLKCTRL**

- Clock Management Functional Description
- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- PRCM Register Manual
- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1347. CM\_L4PER\_I2C3\_CLKCTRL**

<b>Address Offset</b>	0x0000 00B0		
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b>	L4PER_CM2
<b>Description</b>	This register manages the I2C3 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1348. Register Call Summary for Register CM\_L4PER\_I2C3\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1349. CM\_L4PER\_I2C4\_CLKCTRL**

<b>Address Offset</b>	0x0000 00B8		
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b>	L4PER_CM2
<b>Description</b>	This register manages the I2C4 clocks.		
<b>Type</b>	RW		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1350. Register Call Summary for Register CM\_L4PER\_I2C4\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1351. CM\_L4PER\_L4PER\_CLKCTRL**

<b>Address Offset</b>	0x0000 00C0	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the L4PER clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 3-1352. Register Call Summary for Register CM\_L4PER\_L4PER\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1353. CM\_L4PER\_MCBSP4\_CLKCTRL**

<b>Address Offset</b>	0x0000 00E0	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the MCBSP4 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						CLKSEL_INTERNAL_SOURCE	CLKSEL_SOURCE	RESERVED						IDLEST	RESERVED											MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25	CLKSEL_INTERNAL_SOURCE	Selects the internal clock to be used as the functional clock in case CLKSEL_SOURCE selects the internal clock source as the functional clock source.  0x0: 96MHz clock derived from DPLL_PER is selected 0x1: 98MHz clock derived from DPLL_ABE is selected	RW	0

Bits	Field Name	Description	Type	Reset
24	CLKSEL_SOURCE	Selects the source of the functional clock between, internal source and CLK_S pad. The switching between the clocks is not guaranteed to be glitchless.  0x0: Functional clock is sourced from an internal clock 0x1: Functional clock is sourced from CLK_S pad	RW	0
23:18	RESERVED		R	0x00
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x0: Module is fully functional, including INTRCONN  Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion  Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock  Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  Read 0x1: Reserved  0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen.  Read 0x3: Reserved	RW	0x0

**Table 3-1354. Register Call Summary for Register CM\_L4PER\_MCBSP4\_CLKCTRL**

Clock Management Functional Description

- [CM Clock Source: \[0\] \[1\]](#)
- [Clock Domain Module Attributes: \[2\] \[3\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[4\]](#)

**Table 3-1355. CM\_L4PER\_MCSP11\_CLKCTRL**

<b>Address Offset</b>	0x0000 00F0	
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b> L4PER_CM2
<b>Description</b>	This register manages the MCSP11 clocks.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED											MODULEMODE			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1356. Register Call Summary for Register CM\_L4PER\_MCSP11\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1357. CM\_L4PER\_MCSP12\_CLKCTRL**

<b>Address Offset</b>	0x0000 00F8	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the MCSP12 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED																MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1358. Register Call Summary for Register CM\_L4PER\_MCSPi2\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1359. CM\_L4PER\_MCSPi3\_CLKCTRL**

<b>Address Offset</b>	0x0000 0100	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the MCSPi3 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED										MODULEMODE												

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1360. Register Call Summary for Register CM\_L4PER\_MCSPi3\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1361. CM\_L4PER\_MCSPi4\_CLKCTRL**

<b>Address Offset</b>	0x0000 0108	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the MCSPI4 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								MODULEMODE														

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1362. Register Call Summary for Register CM\_L4PER\_MCSPi4\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1363. CM\_L4PER\_MMCSO3\_CLKCTRL**

<b>Address Offset</b>	0x0000 0120		
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b>	L4PER_CM2
<b>Description</b>	This register manages the MMCSO3 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								MODULEMODE														

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1364. Register Call Summary for Register CM\_L4PER\_MMCSO3\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1365. CM\_L4PER\_MMCSO4\_CLKCTRL**

<b>Address Offset</b>	0x0000 0128		
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b>	L4PER_CM2
<b>Description</b>	This register manages the MMCSO4 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1366. Register Call Summary for Register CM\_L4PER\_MMCS4\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1367. CM\_L4PER\_SLIMBUS2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0138	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the SLIMBUS2 clocks.		
<b>Type</b>	RW		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
RESERVED																IDLEST			RESERVED						OPTFCLKEN_SLIMBUS_CLK			OPTFCLKEN_PERABE24M_FCLK			OPTFCLKEN_PER24MC_FCLK			RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:11	RESERVED		R	0x00
10	OPTFCLKEN_SLIMBUS_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
9	OPTFCLKEN_PERABE24M_FCLK	Optional functional clock control for PER_ABE_24M_FCLK clock. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
8	OPTFCLKEN_PER24MC_FCLK	Optional functional clock control for PER_24MC_FCLK clock. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1368. Register Call Summary for Register CM\_L4PER\_SLIMBUS2\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1369. CM\_L4PER\_UART1\_CLKCTRL**

<b>Address Offset</b>	0x0000 0140		
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b>	L4PER_CM2
<b>Description</b>	This register manages the UART1 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								MODULEMODE														

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1370. Register Call Summary for Register CM\_L4PER\_UART1\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1371. CM\_L4PER\_UART2\_CLKCTRL**

<b>Address Offset</b>	0x0000 0148		
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>	<b>Instance</b>	L4PER_CM2
<b>Description</b>	This register manages the UART2 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												IDLEST	RESERVED												MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1372. Register Call Summary for Register CM\_L4PER\_UART2\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1373. CM\_L4PER\_UART3\_CLKCTRL**

<b>Address Offset</b>	0x0000 0150	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the UART3 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												IDLEST	RESERVED												MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1374. Register Call Summary for Register CM\_L4PER\_UART3\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1375. CM\_L4PER\_UART4\_CLKCTRL**

<b>Address Offset</b>	0x0000 0158	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the UART4 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED																MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1376. Register Call Summary for Register CM\_L4PER\_UART4\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

**Table 3-1377. CM\_L4PER\_MMCS5\_CLKCTRL**

<b>Address Offset</b>	0x0000 0160	<b>Instance</b>	L4PER_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1312</a>		
<b>Description</b>	This register manages the MMCS5 clocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED										MODULEMODE				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1378. Register Call Summary for Register CM\_L4PER\_MMCS5\_CLKCTRL**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [L4PER\\_CM2 Register Summary: \[2\]](#)

### 3.11.40 RESTORE\_CM2 Registers

#### 3.11.40.1 RESTORE\_CM2 Register Summary

**Table 3-1379. RESTORE\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	RESTORE_CM2 L4 Base Address
<a href="#">CM_L3_1_CLKSTCTRL_RESTORE</a>	RW	32	0x0000 0000	0x4A00 9E00
<a href="#">CM_L3_2_CLKSTCTRL_RESTORE</a>	RW	32	0x0000 0004	0x4A00 9E04
<a href="#">CM_L4CFG_CLKSTCTRL_RESTORE</a>	RW	32	0x0000 0008	0x4A00 9E08
<a href="#">CM_MEMIF_CLKSTCTRL_RESTORE</a>	RW	32	0x0000 000C	0x4A00 9E0C
<a href="#">CM_L4PER_CLKSTCTRL_RESTORE</a>	RW	32	0x0000 0010	0x4A00 9E10
<a href="#">CM_L3INIT_CLKSTCTRL_RESTORE</a>	RW	32	0x0000 0014	0x4A00 9E14
<a href="#">CM_L3INSTR_L3_3_CLKCTRL_RESTORE</a>	RW	32	0x0000 0018	0x4A00 9E18
<a href="#">CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE</a>	RW	32	0x0000 001C	0x4A00 9E1C
<a href="#">CM_L3INSTR_INTRCONN_WP1_CLKCTRL_RESTORE</a>	RW	32	0x0000 0020	0x4A00 9E20
<a href="#">CM_CM2_PROFILING_CLKCTRL_RESTORE</a>	RW	32	0x0000 0024	0x4A00 9E24
<a href="#">CM_C2C_STATICDEP_RESTORE</a>	RW	32	0x0000 0028	0x4A00 9E28
<a href="#">CM_L3_1_DYNAMICDEP_RESTORE</a>	RW	32	0x0000 002C	0x4A00 9E2C
<a href="#">CM_L3_2_DYNAMICDEP_RESTORE</a>	RW	32	0x0000 0030	0x4A00 9E30
<a href="#">CM_C2C_DYNAMICDEP_RESTORE</a>	RW	32	0x0000 0034	0x4A00 9E34
<a href="#">CM_L4CFG_DYNAMICDEP_RESTORE</a>	RW	32	0x0000 0038	0x4A00 9E38
<a href="#">CM_L4PER_DYNAMICDEP_RESTORE</a>	RW	32	0x0000 003C	0x4A00 9E3C
<a href="#">CM_L4PER_GPIO2_CLKCTRL_RESTORE</a>	RW	32	0x0000 0040	0x4A00 9E40
<a href="#">CM_L4PER_GPIO3_CLKCTRL_RESTORE</a>	RW	32	0x0000 0044	0x4A00 9E44
<a href="#">CM_L4PER_GPIO4_CLKCTRL_RESTORE</a>	RW	32	0x0000 0048	0x4A00 9E48
<a href="#">CM_L4PER_GPIO5_CLKCTRL_RESTORE</a>	RW	32	0x0000 004C	0x4A00 9E4C
<a href="#">CM_L4PER_GPIO6_CLKCTRL_RESTORE</a>	RW	32	0x0000 0050	0x4A00 9E50
<a href="#">CM_L3INIT_HSUSBHOST_CLKCTRL_RESTORE</a>	RW	32	0x0000 0054	0x4A00 9E54
<a href="#">CM_L3INIT_HSUSBTLL_CLKCTRL_RESTORE</a>	RW	32	0x0000 0058	0x4A00 9E58
<a href="#">CM_SDMA_STATICDEP_RESTORE</a>	RW	32	0x0000 005C	0x4A00 9E5C

#### 3.11.40.2 RESTORE\_CM2 Register Description

**Table 3-1380. CM\_L3\_1\_CLKSTCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>	<b>Instance</b>	RESTORE_CM2
<b>Description</b>	Second address map for register <a href="#">CM_L3_1_CLKSTCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_L3_1_ICLK	RESERVED						CLKTRCTRL								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0000000
8	CLKACTIVITY_L3_1_ICLK	This field indicates the state of the L3_1_G1CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the L3_1 clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. Read 0x1: Reserved Read 0x2: Reserved 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1381. Register Call Summary for Register CM\_L3\_1\_CLKSTCTRL\_RESTORE**

- PRCM Register Manual
- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1382. CM\_L3\_2\_CLKSTCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>	<b>Instance</b>	RESTORE_CM2
<b>Description</b>	Second address map for register <a href="#">CM_L3_2_CLKSTCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_L3_2_ICLK	RESERVED						CLKTRCTRL								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_L3_2_ICLK	This field indicates the state of the L3_2_GICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the L3_2 clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. Read 0x1: Reserved Read 0x2: Reserved 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1383. Register Call Summary for Register CM\_L3\_2\_CLKSTCTRL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1384. CM\_L4CFG\_CLKSTCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L4CFG_CLKSTCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_CFG_L4_ICLK	RESERVED						CLKTRCTRL								



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	CLKACTIVITY_CFG_L4_ICLK	This field indicates the state of the L4_CFG_GICLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the L4CFG clock domain.  0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur.  Read 0x1: Reserved  Read 0x2: Reserved  0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1385. Register Call Summary for Register CM\_L4CFG\_CLKSTCTRL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1386. CM\_MEMIF\_CLKSTCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 000C	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_MEMIF_CLKSTCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKACTIVITY_ASYNC_PHY2_CLK	CLKACTIVITY_ASYNC_PHY1_CLK	CLKACTIVITY_ASYNC_DLL_CLK	CLKACTIVITY_PHY_ROOT_CLK	CLKACTIVITY_DLL_CLK	CLKACTIVITY_L3_EMIF_ICLK	RESERVED						CLKTRCTRL			

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x000000
13	CLKACTIVITY_ASYNC_PHY2_CLK	This field indicates the state of the ASYNC_PHY2_CLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
12	CLKACTIVITY_ASYNC_PHY1_CLK	This field indicates the state of the ASYNC_PHY1_CLK clock in the domain. [warm reset insensitive]  Read 0x0: Corresponding clock is definitely gated  Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
11	CLKACTIVITY_ASYNC_DLL_CLK	This field indicates the state of the ASYNC_DLL_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
10	CLKACTIVITY_PHY_ROOT_CLK	This field indicates the state of the PHY_ROOT_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
9	CLKACTIVITY_DLL_CLK	This field indicates the state of the DLL_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_L3_EMIF_ICLK	This field indicates the state of the L3_EMIF_GICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the MEMIF clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. Read 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1387. Register Call Summary for Register CM\_MEMIF\_CLKSTCTRL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1388. CM\_L4PER\_CLKSTCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L4PER_CLKSTCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKACTIVITY_PER_ABE_24M_FCLK	RESERVED	RESERVED	CLKACTIVITY_PER_MCBSP4_FCLK	RESERVED	CLKACTIVITY_PER_96M_FCLK	CLKACTIVITY_PER_48M_FCLK	CLKACTIVITY_PER_32K_FCLK	CLKACTIVITY_PER_24MC_FCLK	CLKACTIVITY_12M_FCLK_FCLK	CLKACTIVITY_GPT9_FCLK	CLKACTIVITY_GPT4_FCLK	CLKACTIVITY_GPT3_FCLK	CLKACTIVITY_GPT2_FCLK	CLKACTIVITY_GPT11_FCLK	CLKACTIVITY_GPT10_FCLK	CLKACTIVITY_L4_PER_ICLK	RESERVED					CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25	CLKACTIVITY_PER_ABE_24M_FCLK	This field indicates the state of the PER_ABE_24M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
24	RESERVED		R	0
23	RESERVED		R	0
22	CLKACTIVITY_PER_MCBSP4_FCLK	This field indicates the state of the PER_MCBSP4_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
21:20	RESERVED		R	0
19	CLKACTIVITY_PER_96M_FCLK	This field indicates the state of the PER_96M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
18	CLKACTIVITY_PER_48M_FCLK	This field indicates the state of the PER_48M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
17	CLKACTIVITY_PER_32K_FCLK	This field indicates the state of the PER_32K_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
16	CLKACTIVITY_PER_24MC_FCLK	This field indicates the state of the PER_24MC_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
15	CLKACTIVITY_12M_FCLK_FCLK	This field indicates the state of the FUNC_12M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
14	CLKACTIVITY_GPT9_FCLK	This field indicates the state of the DMT9_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
13	CLKACTIVITY_GPT4_FCLK	This field indicates the state of the DMT4_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
12	CLKACTIVITY_GPT3_FCLK	This field indicates the state of the DMT3_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
11	CLKACTIVITY_GPT2_FCLK	This field indicates the state of the DMT2_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
10	CLKACTIVITY_GPT11_FCLK	This field indicates the state of the DMT11_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
9	CLKACTIVITY_GPT10_FCLK	This field indicates the state of the DMT10_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_L4_PER_ICLK	This field indicates the state of the L4_PER_GICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED		R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1389. Register Call Summary for Register CM\_L4PER\_CLKSTCTRL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1390. CM\_L3INIT\_CLKSTCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>	<b>Instance</b>	RESTORE_CM2
<b>Description</b>	Second address map for register <a href="#">CM_L3INIT_CLKSTCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	CLKACTIVITY_INIT_60M_P2_FCLK	CLKACTIVITY_INIT_60M_P1_FCLK	CLKACTIVITY_HSI_P2_FCLK	CLKACTIVITY_HSI_P1_FCLK	CLKACTIVITY_UTMI_ROOT_FCLK	CLKACTIVITY_TLL_CH2_FCLK	CLKACTIVITY_TLL_CH1_FCLK	CLKACTIVITY_TLL_CH0_FCLK	CLKACTIVITY_HSI_P2_480M_FCLK	CLKACTIVITY_HSI_P1_480M_FCLK	CLKACTIVITY_INIT_HSMC6_FCLK	CLKACTIVITY_INIT_HSMC2_FCLK	CLKACTIVITY_INIT_HSMC1_FCLK	CLKACTIVITY_INIT_HSI_FCLK	CLKACTIVITY_USB_DPLL_HS_CLK	CLKACTIVITY_USB_DPLL_CLK	CLKACTIVITY_INIT_48MC_FCLK	CLKACTIVITY_INIT_48M_FCLK	RESERVED	RESERVED	CLKACTIVITY_INIT_L4_ICLK	CLKACTIVITY_INIT_L3_ICLK	RESERVED					CLKTRCTRL		

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	CLKACTIVITY_INIT_60M_P2_FCLK	This field indicates the state of the INIT_60M_P2_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
28	CLKACTIVITY_INIT_60M_P1_FCLK	This field indicates the state of the INIT_60M_P1_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
27	CLKACTIVITY_HSI_P2_FCLK	This field indicates the state of the HSI_P2_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
26	CLKACTIVITY_HSI_P1_FCLK	This field indicates the state of the HSI_P1_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
25	CLKACTIVITY_UTMI_ROOT_FCLK	This field indicates the state of the UTMI_ROOT_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
24	CLKACTIVITY_TLL_CH2_FCLK	This field indicates the state of the TLL_CH2_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
23	CLKACTIVITY_TLL_CH1_FCLK	This field indicates the state of the TLL_CH1_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
22	CLKACTIVITY_TLL_CH0_FCLK	This field indicates the state of the TLL_CH0_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
21	CLKACTIVITY_HSIC_P2_480M_FCLK	This field indicates the state of the HSIC_P2_480M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
20	CLKACTIVITY_HSIC_P1_480M_FCLK	This field indicates the state of the HSIC_P1_480M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
19	CLKACTIVITY_INIT_HSMMC6_FCLK	This field indicates the state of the INIT_HSMMC6_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
18	CLKACTIVITY_INIT_HSMMC2_FCLK	This field indicates the state of the INIT_HSMMC2_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
17	CLKACTIVITY_INIT_HSMMC1_FCLK	This field indicates the state of the INIT_HSMMC1_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
16	CLKACTIVITY_INIT_HSI_FCLK	This field indicates the state of the INIT_HSI_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
15	CLKACTIVITY_USB_DPLL_HS_CLK	This field indicates the state of the USB_DPLL_HS_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
14	CLKACTIVITY_USB_DPLL_CLK	This field indicates the state of the USB_DPLL_CLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
13	CLKACTIVITY_INIT_48MC_FCLK	This field indicates the state of the INIT_48MC_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0

Bits	Field Name	Description	Type	Reset
12	CLKACTIVITY_INIT_48M_FCLK	This field indicates the state of the INIT_48M_FCLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
11:10	RESERVED	Reserved	R	0x0
9	CLKACTIVITY_INIT_L4_ICLK	This field indicates the state of the L4_INIT_GICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
8	CLKACTIVITY_INIT_L3_ICLK	This field indicates the state of the L3_INIT_GICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing	R	0
7:2	RESERVED	Reserved	R	0x00
1:0	CLKTRCTRL	Controls the clock state transition of the L3INIT clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

**Table 3-1391. Register Call Summary for Register CM\_L3INIT\_CLKSTCTRL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1392. CM\_L3INSTR\_L3\_3\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L3INSTR_L3_3_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST	RESERVED														MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1393. Register Call Summary for Register CM\_L3INSTR\_L3\_3\_CLKCTRL\_RESTORE**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1394. CM\_L3INSTR\_L3\_INSTR\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L3INSTR_L3_INSTR_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED										MODULEMODE												

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3



Bits	Field Name	Description	Type	Reset
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1395. Register Call Summary for Register CM\_L3INSTR\_L3\_INSTR\_CLKCTRL\_RESTORE**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1396. CM\_L3INSTR\_INTRCONN\_WP1\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L3INSTR_INTRCONN_WP1_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST		RESERVED								MODULEMODE													

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive]  Read 0x3: Module is disabled and cannot be accessed Read 0x2: Module is in idle mode (only Interconnect part). It is functional if using separate functional clock Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x0: Module is fully functional, including Interconnect	R	0x3
15:2	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disabled by software. Any Interconnect access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).  0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any Interconnect access to module is always granted. Module clocks may be gated according to the clock domain state.  Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1397. Register Call Summary for Register  
CM\_L3INSTR\_INTRCONN\_WP1\_CLKCTRL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1398. CM\_CM2\_PROFILING\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_CM2_PROFILING_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST								RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status  Read 0x3: Module is disabled Read 0x2: Module is in Idle Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x0: Module is fully functional	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Control the way mandatory clocks are managed.  0x0: Module is disabled by software. OCP configuration port is not accessible.  0x1: Module is managed automatically by hardware along with L3INSTR domain.  Read 0x2: Reserved Read 0x3: Reserved	RW	0x1

**Table 3-1399. Register Call Summary for Register CM\_CM2\_PROFILING\_CLKCTRL\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1400. CM\_C2C\_STATICDEP\_RESTORE**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_C2C_STATICDEP</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L4PER_STATDEP	L4CFG_STATDEP	RESERVED				L3INIT_STATDEP	L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	ABE_STATDEP	IVAHD_STATDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x000000
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
11:8	RESERVED		R	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
3	ABE_STATDEP	Static dependency towards ABE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1:0	RESERVED		R	0x0

**Table 3-1401. Register Call Summary for Register CM\_C2C\_STATICDEP\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1402. CM\_L3\_1\_DYNAMICDEP\_RESTORE**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L3_1_DYNAMICDEP</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED									L4CFG_DYNDEP	RESERVED				L3_2_DYNDEP	RESERVED	MEMIF_DYNDEP	ABE_DYNDEP	RESERVED					

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:13	RESERVED		R	0x000
12	L4CFG_DYNDEP	Dynamic dependency towards L4CFG clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
11:7	RESERVED		R	0x00
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	RESERVED		R	0
4	MEMIF_DYNDEP	Dynamic dependency towards MEMIF clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
3	ABE_DYNDEP	Dynamic dependency towards ABE clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
2:0	RESERVED		R	0x0

**Table 3-1403. Register Call Summary for Register CM\_L3\_1\_DYNAMICDEP\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1404. CM\_L3\_2\_DYNAMICDEP\_RESTORE**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>	<b>Instance</b>	RESTORE_CM2
<b>Description</b>	Second address map for register <a href="#">CM_L3_2_DYNAMICDEP</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED				C2C_DYNDEP	RESERVED	L4SEC_DYNDEP	L4PER_DYNDEP	RESERVED	GFX_DYNDEP	ISS_DYNDEP	DSS_DYNDEP	L3INIT_DYNDEP	RESERVED	L3_1_DYNDEP	RESERVED	IVAHD_DYNDEP	RESERVED	MPU_M3_DYNDEP					

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:19	RESERVED		R	0x00
18	C2C_DYNDEP	Dynamic dependency towards D2D clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
17:15	RESERVED		R	0x0
14	L4SEC_DYNDEP	Dynamic dependency towards L4SEC clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
13	L4PER_DYNDEP	Dynamic dependency towards L4PER clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
12:11	RESERVED		R	0x0
10	GFX_DYNDEP	Dynamic dependency towards GFX clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
9	ISS_DYNDEP	Dynamic dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8	DSS_DYNDEP	Dynamic dependency towards DSS clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
6	RESERVED		R	0
5	L3_1_DYNDEP	Dynamic dependency towards L3_1 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
4:3	RESERVED		R	0x0
2	IVAHD_DYNDEP	Dynamic dependency towards IVAHD clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
1	RESERVED		R	0
0	MPU_M3_DYNDEP	Dynamic dependency towards CORTEXM3 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1

**Table 3-1405. Register Call Summary for Register CM\_L3\_2\_DYNAMICDEP\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1406. CM\_C2C\_DYNAMICDEP\_RESTORE**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_C2C_DYNAMICDEP</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED										L3_2_DYNDEP	RESERVED	MEMIF_DYNDEP	RESERVED										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:7	RESERVED		R	0x00000
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	RESERVED		R	0
4	MEMIF_DYNDEP	Dynamic dependency towards MEMIF clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
3:0	RESERVED		R	0x0

**Table 3-1407. Register Call Summary for Register CM\_C2C\_DYNAMICDEP\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1408. CM\_L4CFG\_DYNAMICDEP\_RESTORE**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L4CFG_DYNAMICDEP</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED				C2C_DYNDEP	RESERVED	ALWONCORE_DYNDEP	L4WKUP_DYNDEP	RESERVED				SDMA_DYNDEP	RESERVED	ISS_DYNDEP	DSS_DYNDEP	L3INIT_DYNDEP	L3_2_DYNDEP	L3_1_DYNDEP	MEMIF_DYNDEP	RESERVED	DSP_DYNDEP	RESERVED	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:19	RESERVED		R	0x00
18	C2C_DYNDEP	Dynamic dependency towards D2D clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
17	RESERVED		R	1
16	ALWONCORE_DYNDEP	Dynamic dependency towards ALWONCORE clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
15	L4WKUP_DYNDEP	Dynamic dependency towards L4WKUP clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
14:12	RESERVED		R	0x0
11	SDMA_DYNDEP	Dynamic dependency towards SDMA clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
10	RESERVED		R	0
9	ISS_DYNDEP	Dynamic dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8	DSS_DYNDEP	Dynamic dependency towards DSS clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
6	L3_2_DYNDEP	Dynamic dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	L3_1_DYNDEP	Dynamic dependency towards L3_1 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
4	MEMIF_DYNDEP	Dynamic dependency towards MEMIF clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
3:2	RESERVED		R	0x0
1	DSP_DYNDEP	Dynamic dependency towards DSP clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
0	RESERVED		R	0

**Table 3-1409. Register Call Summary for Register CM\_L4CFG\_DYNAMICDEP\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1410. CM\_L4PER\_DYNAMICDEP\_RESTORE**

<b>Address Offset</b>	0x0000 003C		
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>	<b>Instance</b>	RESTORE_CM2
<b>Description</b>	Second address map for register <a href="#">CM_L4PER_DYNAMICDEP</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED								L4SEC_DYNDEP	RESERVED				DSS_DYNDEP	L3INIT_DYNDEP	RESERVED								

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by <a href="#">CM_DYN_DEP_PRESCAL</a> register.	RW	0x4
23:15	RESERVED		R	0x000
14	L4SEC_DYNDEP	Dynamic dependency towards L4SEC clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
13:9	RESERVED		R	0x00
8	DSS_DYNDEP	Dynamic dependency towards DSS clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
6:0	RESERVED		R	0x00

**Table 3-1411. Register Call Summary for Register CM\_L4PER\_DYNAMICDEP\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

**Table 3-1412. CM\_L4PER\_GPIO2\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>	<b>Instance</b>	RESTORE_CM2
<b>Description</b>	Second address map for register <a href="#">CM_L4PER_GPIO2_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												IDLEST	RESERVED				OPTFCLKEN_DBCLK	RESERVED				MODULEMODE									



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1413. Register Call Summary for Register CM\_L4PER\_GPIO2\_CLKCTRL\_RESTORE**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[1\]](#)

**Table 3-1414. CM\_L4PER\_GPIO3\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L4PER_GPIO3_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST		RESERVED						OPTFCLKEN_DBCLK	RESERVED						MODULEMODE								

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1415. Register Call Summary for Register CM\_L4PER\_GPIO3\_CLKCTRL\_RESTORE**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[1\]](#)

**Table 3-1416. CM\_L4PER\_GPIO4\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L4PER_GPIO4_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								OPTFCLKEN_DBCLK	RESERVED								MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1417. Register Call Summary for Register CM\_L4PER\_GPIO4\_CLKCTRL\_RESTORE**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[1\]](#)

**Table 3-1418. CM\_L4PER\_GPIO5\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L4PER_GPIO5_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST		RESERVED								OPTFCLKEN_DBCLK		RESERVED						MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1419. Register Call Summary for Register CM\_L4PER\_GPIO5\_CLKCTRL\_RESTORE**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[1\]](#)

**Table 3-1420. CM\_L4PER\_GPIO6\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L4PER_GPIO6_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IDLEST	RESERVED								OPTFCLKEN_DBCLK	RESERVED								MODULEMODE					

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x00
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:2	RESERVED		R	0x00
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1421. Register Call Summary for Register CM\_L4PER\_GPIO6\_CLKCTRL\_RESTORE**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[1\]](#)

**Table 3-1422. CM\_L3INIT\_HSUSBHOST\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register CM_L3INIT_USB_HOST_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_UTMI_P2	CLKSEL_UTMI_P1	RESERVED				STBYST	IDLEST	OPTFCLKEN_FUNC48MCLK	OPTFCLKEN_HSIC480M_P2_CLK	OPTFCLKEN_HSIC480M_P1_CLK	OPTFCLKEN_HSIC60M_P2_CLK	OPTFCLKEN_HSIC60M_P1_CLK	OPTFCLKEN_UTMI_P3_CLK	OPTFCLKEN_UTMI_P2_CLK	OPTFCLKEN_UTMI_P1_CLK	RESERVED		SAR_MODE	RESERVED		MODULEMODE						

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25	CLKSEL_UTMI_P2	Selects the source of the functional clock for UTMI Port2 on USB Host  0x0: The functional clock is provided by the internal clock source 0x1: The functional clock is provided by an external PHY through an I/O pad.	RW	0
24	CLKSEL_UTMI_P1	Selects the source of the functional clock for UTMI Port1 on USB Host  0x0: The functional clock is provided by the internal clock source 0x1: The functional clock is provided by an external PHY through an I/O pad.	RW	0
23:19	RESERVED		R	0x00
18	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15	OPTFCLKEN_FUNC48MCLK	USB-HOST optional clock control: FUNC48MCLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
14	OPTFCLKEN_HSIC480M_P2_CLK	USB-HOST optional clock control: HSIC480M_P2_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
13	OPTFCLKEN_HSIC480M_P1_CLK	USB-HOST optional clock control: HSIC480M_P1_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
12	OPTFCLKEN_HSIC60M_P2_CLK	USB-HOST optional clock control: HSIC60M_P2_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
11	OPTFCLKEN_HSIC60M_P1_CLK	USB-HOST optional clock control: HSIC60M_P1_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
10	OPTFCLKEN_UTMI_P3_CLK	USB-HOST optional clock control: UTMI_P3_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
9	OPTFCLKEN_UTMI_P2_CLK	USB-HOST optional clock control: UTMI_P2_CLK when CLKSEL_UTMI_P2 is 0 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
8	OPTFCLKEN_UTMI_P1_CLK	USB-HOST optional clock control: UTMI_P1_CLK when CLKSEL_UTMI_P1 is 0 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	SAR_MODE	SAR mode control for the module. Shall not be modify except if module is disabled. 0x0: SAR mode is disabled 0x1: SAR mode is enabled	RW	0
3:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). Read 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. Read 0x3: Reserved	RW	0x0

**Table 3-1423. Register Call Summary for Register CM\_L3INIT\_HSUSBHOST\_CLKCTRL\_RESTORE**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\] \[1\]](#)

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[2\]](#)

**Table 3-1424. CM\_L3INIT\_HSUSBTLL\_CLKCTRL\_RESTORE**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_L3INIT_HSUSBTLL_CLKCTRL</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEST	RESERVED						OPTFCLKEN_USB_CH2_CLK	OPTFCLKEN_USB_CH1_CLK	OPTFCLKEN_USB_CH0_CLK	RESERVED	SAR_MODE	RESERVED	MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status. [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion Read 0x2: Module is in idle mode (only INTRCONN part). It is functional if using separate functional clock Read 0x3: Module is disabled and cannot be accessed	R	0x3
15:11	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
10	OPTFCLKEN_USB_CH2_CLK	USB-HOST optional clock control: USB_CH2_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
9	OPTFCLKEN_USB_CH1_CLK	USB-HOST optional clock control: USB_CH1_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
8	OPTFCLKEN_USB_CH0_CLK	USB-HOST optional clock control: USB_CH0_CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0
7:5	RESERVED		R	0x0
4	SAR_MODE	SAR mode control for the module. Shall not be modify except if module is disabled. 0x0: SAR mode is disabled 0x1: SAR mode is enabled	RW	0
3:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disable by software. Any INTRCONN access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state. Read 0x2: Reserved Read 0x3: Reserved	RW	0x0

**Table 3-1425. Register Call Summary for Register CM\_L3INIT\_HSUSBTLL\_CLKCTRL\_RESTORE**

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\]](#)

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[1\]](#)

**Table 3-1426. CM\_SDMA\_STATICDEP\_RESTORE**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	RESTORE_CM2
<b>Physical Address</b>	See <a href="#">Table 3-1379</a>		
<b>Description</b>	Second address map for register <a href="#">CM_SDMA_STATICDEP</a> . Used only by automatic restore upon wakeup from device OFF mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
RESERVED																L4WKUP_STATDEP	RESERVED	L4PER_STATDEP	L4CFG_STATDEP	RESERVED	ISS_STATDEP	DSS_STATDEP	L3INIT_STATDEP	L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	ABE_STATDEP	IVAHD_STATDEP	RESERVED	MPU_M3_STATDEP																



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	L4WKUP_STATDEP	Static dependency towards L4WKUP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
14	RESERVED		RW	0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
11:10	RESERVED		R	0x0
9	ISS_STATDEP	Static dependency towards ISS clock domain Read 0x0: Dependency is disabled	R	0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
6	L3_2_STATDEP	Static dependency towards L3_2 clock domain Read 0x1: Dependency is enabled	R Rreturns 1s	1
5	L3_1_STATDEP	Static dependency towards L3_1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
4	MEMIF_STATDEP	Static dependency towards MEMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	1
3	ABE_STATDEP	Static dependency towards ABE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
2	IVAHD_STATDEP	Static dependency towards IVAHD clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0
1	RESERVED		R	0
0	MPU_M3_STATDEP	Static dependency towards MPU_A3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0

**Table 3-1427. Register Call Summary for Register CM\_SDMA\_STATICDEP\_RESTORE**

PRCM Register Manual

- [RESTORE\\_CM2 Register Summary: \[0\]](#)

## 3.12 SCRM Register Manual

### 3.12.1 SCRM Instance Summary

**Table 3-1428. SCRM Instance Summary**

Module Name	L4 Base Address	Size
SCRM	0x4A30 A000	256Bytes

### 3.12.2 SCRM Registers

#### 3.12.2.1 SCRM Register Summary

**Table 3-1429. SCRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SCRM L4 Base Address
REVISION_SCRM	R	32	0x0000 0000	0x4A30 A000
CLKSETUPTIME	RW	32	0x0000 0100	0x4A30 A100
PMICSETUPTIME	RW	32	0x0000 0104	0x4A30 A104
ALTCLKSRC	RW	32	0x0000 0110	0x4A30 A110
RESERVED	RW	32	0x0000 0118	0x4A30 A118
C2CLKM	RW	32	0x0000 011C	0x4A30 A11C
EXTCLKREQ	RW	32	0x0000 0200	0x4A30 A200
ACCCLKREQ	RW	32	0x0000 0204	0x4A30 A204
PWRREQ	RW	32	0x0000 0208	0x4A30 A208
AUXCLKREQ0	RW	32	0x0000 0210	0x4A30 A210
AUXCLKREQ1	RW	32	0x0000 0214	0x4A30 A214
AUXCLKREQ2	RW	32	0x0000 0218	0x4A30 A218
AUXCLKREQ3	RW	32	0x0000 021C	0x4A30 A21C
AUXCLKREQ4	RW	32	0x0000 0220	0x4A30 A220
AUXCLKREQ5	RW	32	0x0000 0224	0x4A30 A224
C2CLKREQ	RW	32	0x0000 0234	0x4A30 A234
AUXCLK0	RW	32	0x0000 0310	0x4A30 A310
AUXCLK1	RW	32	0x0000 0314	0x4A30 A314
AUXCLK2	RW	32	0x0000 0318	0x4A30 A318
AUXCLK3	RW	32	0x0000 031C	0x4A30 A31C
AUXCLK4	RW	32	0x0000 0320	0x4A30 A320
AUXCLK5	RW	32	0x0000 0324	0x4A30 A324
RSTTIME_REG	RW	32	0x0000 0400	0x4A30 A400
RESERVED	RW	32	0x0000 0418	0x4A30 A418
C2CRSTCTRL	RW	32	0x0000 041C	0x4A30 A41C
EXTPWRONRSTCTRL	RW	32	0x0000 0420	0x4A30 A420
RESERVED	RW	32	0x0000 0500	0x4A30 A500
RESERVED	RW	32	0x0000 0504	0x4A30 A504
RESERVED	RW	32	0x0000 0508	0x4A30 A508
RESERVED	RW	32	0x0000 050C	0x4A30 A50C
EXTWARMRSTST_REG	RW	32	0x0000 0510	0x4A30 A510
APEWARMRSTST_REG	RW	32	0x0000 0514	0x4A30 A514
RESERVED	RW	32	0x0000 0518	0x4A30 A518
C2CWARMRSTST_REG	RW	32	0x0000 051C	0x4A30 A51C

3.12.2.2 SCRM Register Description

Table 3-1430. REVISION\_SCRM

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register contains the IP revision code for the SCRM.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REV															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reads returns 0.	R	0x000000
7:0	REV	Revision Number	R	0x- - TI Internal data.

Table 3-1431. Register Call Summary for Register REVISION\_SCRM

- SCRM Register Manual
- [SCRM Register Summary: \[0\]](#)

Table 3-1432. CLKSETUPTIME

<b>Address Offset</b>	0x0000 0100	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register holds the clock setup time counters of the system clock source supplier.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DOWNTIME								RESERVED				SETUPTIME											

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reads returns 0.	R	0x000
21:16	DOWNTIME	Holds the number of 32 kHz clock cycles it takes to gate the clock source supplier.	RW	0x00
15:12	RESERVED	Reads returns 0.	R	0x0
11:0	SETUPTIME	Holds the number of 32 kHz clock cycles it takes to stabilize the clock source supplier.	RW	0x000

Table 3-1433. Register Call Summary for Register CLKSETUPTIME

- SCRM Register Manual
- [SCRM Register Summary: \[0\]](#)

**Table 3-1434. PMICSETUPTIME**

<b>Address Offset</b>	0x0000 0104		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register holds the setup time counters for the sleep mode of the PMIC.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WAKEUPTIME								RESERVED								SLEEPTIME							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reads returns 0.	R	0x000
21:16	WAKEUPTIME	Holds the number of 32 kHz clock cycles it takes to exit the PMIC from sleep mode.	RW	0x00
15:6	RESERVED	Reads returns 0.	R	0x000
5:0	SLEEPTIME	Holds the number of 32 kHz clock cycles it takes to enter the PMIC in sleep mode.	RW	0x00

**Table 3-1435. Register Call Summary for Register PMICSETUPTIME**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1436. ALTCLKSRC**

<b>Address Offset</b>	0x0000 0110		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register controls the alternate system clock source supplier.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE_EXT		ENABLE_INT		MODE											

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads returns 0.	R	0x00000000
3	ENABLE_EXT	This bit allows to enable and disable the output alternate system clock version. This bit is intended to be used in order to gate this clock path while the source is stabilizing.  0x0: The alternate system clock version is disabled. 0x1: The alternate system clock version is enabled.	RW	0
2	ENABLE_INT	This bit allows to enable and disable the alternate system clock version used to generate the auxiliary clocks. This bit is intended to be used in order to gate this clock path while the source is stabilizing and also to gate this clock path while switching the auxiliary clock paths on / from this possible source.  0x0: The alternate system clock version is disabled. 0x1: The alternate system clock version is enabled.	RW	0

Bits	Field Name	Description	Type	Reset
1:0	MODE	This bit field defines the functional mode of the alternate system clock supplier.  0x0: The alternate system clock source supplier is powered-down.  0x1: The alternate system clock source supplier is active.  0x2: The alternate system clock source supplier is bypassed.  0x3: Reserved	RW	0x0

Table 3-1437. Register Call Summary for Register ALTCLKSRC

Clock Management Functional Description

- [Clock Domain Module Attributes: \[0\]](#)

SCRM Register Manual

- [SCRM Register Summary: \[1\]](#)

Table 3-1438. C2CCLKM

Address Offset	0x0000 011C	
Physical Address	See <a href="#">Table 3-1429</a>	Instance SCRM
Description	This register controls the clocks of the external C2C interface.	
Type	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													SYSCLK	CLK_32KHZ	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads returns 0.	R	0x0000 0000
1	SYSCLK	This bit allows to enable and disable the system clock version of the external C2C interface.  0x0: The system clock version for the external C2C interface is disabled.  0x1: The system clock version for the external C2C interface is enabled.	RW	0
0	CLK_32KHZ	This bit allows to enable and disable the 32 kHz clock version of the external C2C interface.  0x0: The 32 kHz clock version for the external C2C interface is disabled.  0x1: The 32 kHz clock version for the external C2C interface is enabled.	RW	0

Table 3-1439. Register Call Summary for Register C2CCLKM

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1440. EXTCLKREQ**

<b>Address Offset</b>	0x0000 0200		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register holds qualifiers for the external clock request.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												POLARITY			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0.	R	0x0000 0000
0	POLARITY	This bit defines the active level of the external clock request. 0x0: The external clock request is active low. 0x1: The external clock request is active high.	RW	1

**Table 3-1441. Register Call Summary for Register EXTCLKREQ**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1442. ACCCLKREQ**

<b>Address Offset</b>	0x0000 0204		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register holds qualifiers for the accurate clock request.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												POLARITY			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0.	R	0x0000 0000
0	POLARITY	This bit defines the active level of the accurate clock request. 0x0: The accurate clock request is active low. 0x1: The accurate clock request is active high.	RW	1

**Table 3-1443. Register Call Summary for Register ACCCLKREQ**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1444. PWRREQ**

<b>Address Offset</b>	0x0000 0208		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register holds qualifiers for the external power request.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												POLARITY			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0.	R	0x0000 0000
0	POLARITY	This bit defines the active level of the external power request. 0x0: The external power request is active low. 0x1: The external power request is active high.	RW	1

**Table 3-1445. Register Call Summary for Register PWRREQ**

PRCM Register Manual

- [DEVICE\\_PRM Register Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

SCRM Register Manual

- [SCRM Register Summary: \[9\]](#)

**Table 3-1446. AUXCLKREQ0**

<b>Address Offset</b>	0x0000 0210		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register holds qualifiers for the auxiliary clock request #0.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										MAPPING	ACCURACY	POLARITY			

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000000
4:2	MAPPING	<p>This field allows re-mapping the auxiliary clock request #0 on another auxiliary clock output than auxiliary clock #0.</p> <p>0x0: The auxiliary clock request #0 is mapped on the auxiliary clock #0.</p> <p>0x1: The auxiliary clock request #0 is mapped on the auxiliary clock #1.</p> <p>0x2: The auxiliary clock request #0 is mapped on the auxiliary clock #2.</p> <p>0x3: The auxiliary clock request #0 is mapped on the auxiliary clock #3.</p> <p>0x4: The auxiliary clock request #0 is mapped on the auxiliary clock #4.</p> <p>0x5: The auxiliary clock request #0 is mapped on the auxiliary clock #5.</p> <p>0x6: Reserved</p> <p>0x7: Reserved</p>	RW	0x0
1	ACCURACY	<p>This bit qualifies the auxiliary clock request #0 as an accurate clock request.</p> <p>0x0: An active auxiliary clock request #0 doesn't generate an accurate clock request.</p> <p>0x1: An active auxiliary clock request #0 generates an accurate clock request.</p>	RW	0
0	POLARITY	<p>This bit defines the active level of the auxiliary clock request #0.</p> <p>0x0: The auxiliary clock request #0 is active low.</p> <p>0x1: The auxiliary clock request #0 is active high.</p>	RW	1

**Table 3-1447. Register Call Summary for Register AUXCLKREQ0**

PRCM Subsystem Environment

- [External Clock Signals: \[0\]](#)

SCRM Register Manual

- [SCRM Register Summary: \[1\]](#)

**Table 3-1448. AUXCLKREQ1**

<b>Address Offset</b>	0x0000 0214																																																																			
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>											<b>Instance</b>					SCRM																																																			
<b>Description</b>	This register holds qualifiers for the auxiliary clock request #1.																																																																			
<b>Type</b>	RW																																																																			
<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 2.5%;">31</th><th style="width: 2.5%;">30</th><th style="width: 2.5%;">29</th><th style="width: 2.5%;">28</th><th style="width: 2.5%;">27</th><th style="width: 2.5%;">26</th><th style="width: 2.5%;">25</th><th style="width: 2.5%;">24</th><th style="width: 2.5%;">23</th><th style="width: 2.5%;">22</th><th style="width: 2.5%;">21</th><th style="width: 2.5%;">20</th><th style="width: 2.5%;">19</th><th style="width: 2.5%;">18</th><th style="width: 2.5%;">17</th><th style="width: 2.5%;">16</th><th style="width: 2.5%;">15</th><th style="width: 2.5%;">14</th><th style="width: 2.5%;">13</th><th style="width: 2.5%;">12</th><th style="width: 2.5%;">11</th><th style="width: 2.5%;">10</th><th style="width: 2.5%;">9</th><th style="width: 2.5%;">8</th><th style="width: 2.5%;">7</th><th style="width: 2.5%;">6</th><th style="width: 2.5%;">5</th><th style="width: 2.5%;">4</th><th style="width: 2.5%;">3</th><th style="width: 2.5%;">2</th><th style="width: 2.5%;">1</th><th style="width: 2.5%;">0</th> </tr> </thead> <tbody> <tr> <td colspan="16" style="text-align: center;">RESERVED</td> <td colspan="2" style="text-align: center;">MAPPING</td> <td style="text-align: center;">ACCURACY</td> <td style="text-align: center;">POLARITY</td> </tr> </tbody> </table>																	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																MAPPING		ACCURACY	POLARITY
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																					
RESERVED																MAPPING		ACCURACY	POLARITY																																																	



Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reads returns 0.	R	0x0000000
4:2	MAPPING	This field allows re-mapping the auxiliary clock request #1 on another auxiliary clock output than auxiliary clock #1.  0x0: The auxiliary clock request #1 is mapped on the auxiliary clock #0.  0x1: The auxiliary clock request #1 is mapped on the auxiliary clock #1.  0x2: The auxiliary clock request #1 is mapped on the auxiliary clock #2.  0x3: The auxiliary clock request #1 is mapped on the auxiliary clock #3.  0x4: The auxiliary clock request #1 is mapped on the auxiliary clock #4.  0x5: The auxiliary clock request #1 is mapped on the auxiliary clock #5.  0x6: Reserved 0x7: Reserved	RW	0x1
1	ACCURACY	This bit qualifies the auxiliary clock request #1 as an accurate clock request.  0x0: An active auxiliary clock request #1 doesn't generate an accurate clock request.  0x1: An active auxiliary clock request #1 generates an accurate clock request.	RW	0
0	POLARITY	This bit defines the active level of the auxiliary clock request #1.  0x0: The auxiliary clock request #1 is active low.  0x1: The auxiliary clock request #1 is active high.	RW	1

**Table 3-1449. Register Call Summary for Register AUXCLKREQ1**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1450. AUXCLKREQ2**

<b>Address Offset</b>	0x0000 0218	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register holds qualifiers for the auxiliary clock request #2.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MAPPING		ACCURACY	POLARITY				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reads returns 0.	R	0x0000000
4:2	MAPPING	This field allows re-mapping the auxiliary clock request #2 on another auxiliary clock output than auxiliary clock #2.  0x0: The auxiliary clock request #2 is mapped on the auxiliary clock #0.  0x1: The auxiliary clock request #2 is mapped on the auxiliary clock #1.  0x2: The auxiliary clock request #2 is mapped on the auxiliary clock #2.  0x3: The auxiliary clock request #2 is mapped on the auxiliary clock #3.  0x4: The auxiliary clock request #2 is mapped on the auxiliary clock #4.  0x5: The auxiliary clock request #2 is mapped on the auxiliary clock #5.  0x6: Reserved 0x7: Reserved	RW	0x2
1	ACCURACY	This bit qualifies the auxiliary clock request #2 as an accurate clock request.  0x0: An active auxiliary clock request #2 doesn't generate an accurate clock request.  0x1: An active auxiliary clock request #2 generates an accurate clock request.	RW	0
0	POLARITY	This bit defines the active level of the auxiliary clock request #2.  0x0: The auxiliary clock request #2 is active low.  0x1: The auxiliary clock request #2 is active high.	RW	1

**Table 3-1451. Register Call Summary for Register AUXCLKREQ2**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1452. AUXCLKREQ3**

<b>Address Offset</b>	0x0000 021C																														
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b> SCRM																													
<b>Description</b>	This register holds qualifiers for the auxiliary clock request #3.																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MAPPING		ACCURACY	POLARITY												

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reads returns 0.	R	0x0000000
4:2	MAPPING	This field allows re-mapping the auxiliary clock request #3 on another auxiliary clock output than auxiliary clock #3.  0x0: The auxiliary clock request #3 is mapped on the auxiliary clock #0.  0x1: The auxiliary clock request #3 is mapped on the auxiliary clock #1.  0x2: The auxiliary clock request #3 is mapped on the auxiliary clock #2.  0x3: The auxiliary clock request #3 is mapped on the auxiliary clock #3.  0x4: The auxiliary clock request #3 is mapped on the auxiliary clock #4.  0x5: The auxiliary clock request #3 is mapped on the auxiliary clock #5.  0x6: Reserved 0x7: Reserved	RW	0x3
1	ACCURACY	This bit qualifies the auxiliary clock request #3 as an accurate clock request.  0x0: An active auxiliary clock request #3 doesn't generate an accurate clock request.  0x1: An active auxiliary clock request #3 generates an accurate clock request.	RW	0
0	POLARITY	This bit defines the active level of the auxiliary clock request #3.  0x0: The auxiliary clock request #3 is active low.  0x1: The auxiliary clock request #3 is active high.	RW	1

**Table 3-1453. Register Call Summary for Register AUXCLKREQ3**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1454. AUXCLKREQ4**

<b>Address Offset</b>	0x0000 0220	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register holds qualifiers for the auxiliary clock request #4.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MAPPING		ACCURACY	POLARITY				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reads returns 0.	R	0x0000000
4:2	MAPPING	This field allows re-mapping the auxiliary clock request #4 on another auxiliary clock output than auxiliary clock #4.  0x0: The auxiliary clock request #4 is mapped on the auxiliary clock #0.  0x1: The auxiliary clock request #4 is mapped on the auxiliary clock #1.  0x2: The auxiliary clock request #4 is mapped on the auxiliary clock #2.  0x3: The auxiliary clock request #4 is mapped on the auxiliary clock #3.  0x4: The auxiliary clock request #4 is mapped on the auxiliary clock #4.  0x5: The auxiliary clock request #4 is mapped on the auxiliary clock #5.  0x6: Reserved 0x7: Reserved	RW	0x4
1	ACCURACY	This bit qualifies the auxiliary clock request #4 as an accurate clock request.  0x0: An active auxiliary clock request #4 doesn't generate an accurate clock request.  0x1: An active auxiliary clock request #4 generates an accurate clock request.	RW	0
0	POLARITY	This bit defines the active level of the auxiliary clock request #4.  0x0: The auxiliary clock request #4 is active low.  0x1: The auxiliary clock request #4 is active high.	RW	1

**Table 3-1455. Register Call Summary for Register AUXCLKREQ4**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1456. AUXCLKREQ5**

<b>Address Offset</b>	0x0000 0224																														
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b> SCRM																													
<b>Description</b>	This register holds qualifiers for the auxiliary clock request #5.																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MAPPING		ACCURACY	POLARITY												

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reads returns 0.	R	0x0000000
4:2	MAPPING	This field allows re-mapping the auxiliary clock request #5 on another auxiliary clock output than auxiliary clock #5.  0x0: The auxiliary clock request #5 is mapped on the auxiliary clock #0.  0x1: The auxiliary clock request #5 is mapped on the auxiliary clock #1.  0x2: The auxiliary clock request #5 is mapped on the auxiliary clock #2.  0x3: The auxiliary clock request #5 is mapped on the auxiliary clock #3.  0x4: The auxiliary clock request #5 is mapped on the auxiliary clock #4.  0x5: The auxiliary clock request #5 is mapped on the auxiliary clock #5.  0x6: Reserved 0x7: Reserved	RW	0x5
1	ACCURACY	This bit qualifies the auxiliary clock request #5 as an accurate clock request.  0x0: An active auxiliary clock request #5 doesn't generate an accurate clock request.  0x1: An active auxiliary clock request #5 generates an accurate clock request.	RW	0
0	POLARITY	This bit defines the active level of the auxiliary clock request #5.  0x0: The auxiliary clock request #5 is active low.  0x1: The auxiliary clock request #5 is active high.	RW	1

**Table 3-1457. Register Call Summary for Register AUXCLKREQ5**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1458. C2CLKREQ**

<b>Address Offset</b>	0x0000 0234	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register holds qualifiers for the external C2C interface clock request.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACCURACY		POLARITY													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads returns 0.	R	0x0000 0000
1	ACCURACY	This bit qualifies the external C2C interface clock request as an accurate clock request.  0x0: An active external C2C interface clock request doesn't generate an accurate clock request.  0x1: An active external C2C interface clock request generates an accurate clock request.	RW	0

Bits	Field Name	Description	Type	Reset
0	POLARITY	This bit defines the active level of the external C2C interface clock request.  0x0: The external C2C interface clock request is active low.  0x1: The external C2C interface clock request is active high.	RW	1

**Table 3-1459. Register Call Summary for Register C2CCLKREQ**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1460. AUXCLK0**

<b>Address Offset</b>	0x0000 0310	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register holds qualifiers for the auxiliary clock #0.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKDIV				RESERVED				DISABLECLK	ENABLE	RESERVED				SRCSELECT	POLARITY								

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reads returns 0.	R	0x000
19:16	CLKDIV	This field holds the divider value for the auxiliary clock #0.  0x0: The auxiliary clock #0 is divided by 1. 0x1: The auxiliary clock #0 is divided by 2. 0x2: The auxiliary clock #0 is divided by 3. 0x3: The auxiliary clock #0 is divided by 4. 0x4: The auxiliary clock #0 is divided by 5. 0x5: The auxiliary clock #0 is divided by 6. 0x6: The auxiliary clock #0 is divided by 7. 0x7: The auxiliary clock #0 is divided by 8. 0x8: The auxiliary clock #0 is divided by 9. 0x9: The auxiliary clock #0 is divided by 10. 0xA: The auxiliary clock #0 is divided by 11. 0xB: The auxiliary clock #0 is divided by 12. 0xC: The auxiliary clock #0 is divided by 13. 0xD: The auxiliary clock #0 is divided by 14. 0xE: The auxiliary clock #0 is divided by 15. 0xF: The auxiliary clock #0 is divided by 16.	RW	0x0
15:10	RESERVED	Reads returns 0.	R	0x00
9	DISABLECLK	This bit allows to gate the auxiliary clock #0 without condition. This bit is intended to be used only when the SOC is not clock provider.  0x0: The auxiliary clock #0 is gated upon normal condition: auxiliary clock requests mapped on this path or ENABLE bit set.  0x1: The auxiliary clock #0 is gated without condition.	RW	0

Bits	Field Name	Description	Type	Reset
8	ENABLE	This bit allows to request the auxiliary clock #0 by software. 0x0: The auxiliary clock #0 is not requested by software. 0x1: The auxiliary clock #0 is requested by software.	RW	0
7:3	RESERVED	Reads returns 0.	R	0x00
2:1	SRCSELECT	This field allows selecting the clock source of the auxiliary clock #0. 0x0: The clock source is the system clock. 0x1: The clock source is the version from the CORE DPLL. 0x2: The clock source is the version from the PER DPLL. 0x3: RESERVED.	RW	0x0
0	POLARITY	This bit defines the output level when the auxiliary clock #0 is gated. 0x0: The auxiliary clock #0 is gated low. 0x1: The auxiliary clock #0 is gated high.	RW	0

**Table 3-1461. Register Call Summary for Register AUXCLK0**

PRCM Subsystem Environment

- [External Clock Signals: \[0\]](#)

SCRM Register Manual

- [SCRM Register Summary: \[1\]](#)

**Table 3-1462. AUXCLK1**

<b>Address Offset</b>	0x0000 0314	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register holds qualifiers for the auxiliary clock #1.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKDIV				RESERVED				ENABLE	RESERVED				SRCSELECT	POLARITY									

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reads returns 0.	R	0x000
19:16	CLKDIV	This field holds the divider value for the auxiliary clock #1.  0x0: The auxiliary clock #1 is divided by 1. 0x1: The auxiliary clock #1 is divided by 2. 0x2: The auxiliary clock #1 is divided by 3. 0x3: The auxiliary clock #1 is divided by 4. 0x4: The auxiliary clock #1 is divided by 5. 0x5: The auxiliary clock #1 is divided by 6. 0x6: The auxiliary clock #1 is divided by 7. 0x7: The auxiliary clock #1 is divided by 8. 0x8: The auxiliary clock #1 is divided by 9. 0x9: The auxiliary clock #1 is divided by 10. 0xA: The auxiliary clock #1 is divided by 11. 0xB: The auxiliary clock #1 is divided by 12. 0xC: The auxiliary clock #1 is divided by 13. 0xD: The auxiliary clock #1 is divided by 14. 0xE: The auxiliary clock #1 is divided by 15. 0xF: The auxiliary clock #1 is divided by 16.	RW	0x0
15:9	RESERVED	Reads returns 0.	R	0x00
8	ENABLE	This bit allows to request the auxiliary clock #1 by software.  0x0: The auxiliary clock #1 is not requested by software. 0x1: The auxiliary clock #1 is requested by software.	RW	0
7:3	RESERVED	Reads returns 0.	R	0x00
2:1	SRCSELECT	This field allows selecting the clock source of the auxiliary clock #1.  0x0: The clock source is the system clock. 0x1: The clock source is the version from the CORE DPLL. 0x2: The clock source is the version from the PER DPLL. 0x3: RESERVED.	RW	0x0
0	POLARITY	This bit defines the output level when the auxiliary clock #1 is gated.  0x0: The auxiliary clock #1 is gated low. 0x1: The auxiliary clock #1 is gated high.	RW	0

**Table 3-1463. Register Call Summary for Register AUXCLK1**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)



**Table 3-1464. AUXCLK2**

<b>Address Offset</b>	0x0000 0318
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>
<b>Instance</b>	SCRM
<b>Description</b>	This register holds qualifiers for the auxiliary clock #2.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKDIV				RESERVED				ENABLE	RESERVED				SRCSELECT	POLARITY									

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reads returns 0.	R	0x000
19:16	CLKDIV	This field holds the divider value for the auxiliary clock #2. 0x0: The auxiliary clock #2 is divided by 1. 0x1: The auxiliary clock #2 is divided by 2. 0x2: The auxiliary clock #2 is divided by 3. 0x3: The auxiliary clock #2 is divided by 4. 0x4: The auxiliary clock #2 is divided by 5. 0x5: The auxiliary clock #2 is divided by 6. 0x6: The auxiliary clock #2 is divided by 7. 0x7: The auxiliary clock #2 is divided by 8. 0x8: The auxiliary clock #2 is divided by 9. 0x9: The auxiliary clock #2 is divided by 10. 0xA: The auxiliary clock #2 is divided by 11. 0xB: The auxiliary clock #2 is divided by 12. 0xC: The auxiliary clock #2 is divided by 13. 0xD: The auxiliary clock #2 is divided by 14. 0xE: The auxiliary clock #2 is divided by 15. 0xF: The auxiliary clock #2 is divided by 16.	RW	0x0
15:9	RESERVED	Reads returns 0.	R	0x00
8	ENABLE	This bit allows to request the auxiliary clock #2 by software. 0x0: The auxiliary clock #2 is not requested by software. 0x1: The auxiliary clock #2 is requested by software.	RW	0
7:3	RESERVED	Reads returns 0.	R	0x00
2:1	SRCSELECT	This field allows selecting the clock source of the auxiliary clock #2. 0x0: The clock source is the system clock. 0x1: The clock source is the version from the CORE DPLL. 0x2: The clock source is the version from the PER DPLL. 0x3: RESERVED.	RW	0x0
0	POLARITY	This bit defines the output level when the auxiliary clock #2 is gated. 0x0: The auxiliary clock #2 is gated low. 0x1: The auxiliary clock #2 is gated high.	RW	0

**Table 3-1465. Register Call Summary for Register AUXCLK2**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1466. AUXCLK3**

<b>Address Offset</b>	0x0000 031C	
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b> SCRM
<b>Description</b>	This register holds qualifiers for the auxiliary clock #3.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKDIV				RESERVED				ENABLE	RESERVED				SRCSELECT	POLARITY									

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reads returns 0.	R	0x000
19:16	CLKDIV	This field holds the divider value for the auxiliary clock #3. 0x0: The auxiliary clock #3 is divided by 1. 0x1: The auxiliary clock #3 is divided by 2. 0x2: The auxiliary clock #3 is divided by 3. 0x3: The auxiliary clock #3 is divided by 4. 0x4: The auxiliary clock #3 is divided by 5. 0x5: The auxiliary clock #3 is divided by 6. 0x6: The auxiliary clock #3 is divided by 7. 0x7: The auxiliary clock #3 is divided by 8. 0x8: The auxiliary clock #3 is divided by 9. 0x9: The auxiliary clock #3 is divided by 10. 0xA: The auxiliary clock #3 is divided by 11. 0xB: The auxiliary clock #3 is divided by 12. 0xC: The auxiliary clock #3 is divided by 13. 0xD: The auxiliary clock #3 is divided by 14. 0xE: The auxiliary clock #3 is divided by 15. 0xF: The auxiliary clock #3 is divided by 16.	RW	0x0
15:9	RESERVED	Reads returns 0.	R	0x00
8	ENABLE	This bit allows to request the auxiliary clock #3 by software. 0x0: The auxiliary clock #3 is disabled by software. 0x1: The auxiliary clock #3 is requested by software.	RW	0
7:3	RESERVED	Reads returns 0.	R	0x00
2:1	SRCSELECT	This field allows selecting the clock source of the auxiliary clock #3. 0x0: The clock source is the system clock. 0x1: The clock source is the version from the CORE DPLL. 0x2: The clock source is the version from the PER DPLL. 0x3: RESERVED.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	POLARITY	This bit defines the output level when the auxiliary clock #3 is gated. 0x0: The auxiliary clock #3 is gated low. 0x1: The auxiliary clock #3 is gated high.	RW	0

**Table 3-1467. Register Call Summary for Register AUXCLK3**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1468. AUXCLK4**

<b>Address Offset</b>	0x0000 0320	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register holds qualifiers for the auxiliary clock #4.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKDIV				RESERVED				ENABLE	RESERVED				SRCSELECT	POLARITY									

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reads returns 0.	R	0x000
19:16	CLKDIV	This field holds the divider value for the auxiliary clock #4. 0x0: The auxiliary clock #4 is divided by 1. 0x1: The auxiliary clock #4 is divided by 2. 0x2: The auxiliary clock #4 is divided by 3. 0x3: The auxiliary clock #4 is divided by 4. 0x4: The auxiliary clock #4 is divided by 5. 0x5: The auxiliary clock #4 is divided by 6. 0x6: The auxiliary clock #4 is divided by 7. 0x7: The auxiliary clock #4 is divided by 8. 0x8: The auxiliary clock #4 is divided by 9. 0x9: The auxiliary clock #4 is divided by 10. 0xA: The auxiliary clock #4 is divided by 11. 0xB: The auxiliary clock #4 is divided by 12. 0xC: The auxiliary clock #4 is divided by 13. 0xD: The auxiliary clock #4 is divided by 14. 0xE: The auxiliary clock #4 is divided by 15. 0xF: The auxiliary clock #4 is divided by 16.	RW	0x1
15:9	RESERVED	Reads returns 0.	R	0x00
8	ENABLE	This bit allows to request the auxiliary clock #4 by software. 0x0: The auxiliary clock #4 is not requested by software. 0x1: The auxiliary clock #4 is requested by software.	RW	1
7:3	RESERVED	Reads returns 0.	R	0x00

Bits	Field Name	Description	Type	Reset
2:1	SRCSELECT	This field allows selecting the clock source of the auxiliary clock #4. 0x0: The clock source is the system clock. 0x1: The clock source is the version from the CORE DPLL. 0x2: The clock source is the version from the PER DPLL. 0x3: RESERVED.	RW	0x0
0	POLARITY	This bit defines the output level when the auxiliary clock #4 is gated. 0x0: The auxiliary clock #4 is gated low. 0x1: The auxiliary clock #4 is gated high.	RW	0

**Table 3-1469. Register Call Summary for Register AUXCLK4**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1470. AUXCLK5**

<b>Address Offset</b>	0x0000 0324	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register holds qualifiers for the auxiliary clock #5.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLKDIV				RESERVED				ENABLE	RESERVED				SRCSELECT	POLARITY									

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reads returns 0.	R	0x000
19:16	CLKDIV	This field holds the divider value for the auxiliary clock #5. 0x0: The auxiliary clock #5 is divided by 1. 0x1: The auxiliary clock #5 is divided by 2. 0x2: The auxiliary clock #5 is divided by 3. 0x3: The auxiliary clock #5 is divided by 4. 0x4: The auxiliary clock #5 is divided by 5. 0x5: The auxiliary clock #5 is divided by 6. 0x6: The auxiliary clock #5 is divided by 7. 0x7: The auxiliary clock #5 is divided by 8. 0x8: The auxiliary clock #5 is divided by 9. 0x9: The auxiliary clock #5 is divided by 10. 0xA: The auxiliary clock #5 is divided by 11. 0xB: The auxiliary clock #5 is divided by 12. 0xC: The auxiliary clock #5 is divided by 13. 0xD: The auxiliary clock #5 is divided by 14. 0xE: The auxiliary clock #5 is divided by 15. 0xF: The auxiliary clock #5 is divided by 16.	RW	0x0
15:9	RESERVED	Reads returns 0.	R	0x00

Bits	Field Name	Description	Type	Reset
8	ENABLE	This bit allows to request the auxiliary clock #5 by software. 0x0: The auxiliary clock #5 is not requested by software. 0x1: The auxiliary clock #5 is requested by software.	RW	0
7:3	RESERVED	Reads returns 0.	R	0x00
2:1	SRCSELECT	This field allows selecting the clock source of the auxiliary clock #5. 0x0: The clock source is the system clock. 0x1: The clock source is the version from the CORE DPLL. 0x2: The clock source is the version from the PER DPLL. 0x3: RESERVED.	RW	0x0
0	POLARITY	This bit defines the output level when the auxiliary clock #5 is gated. 0x0: The auxiliary clock #5 is gated low. 0x1: The auxiliary clock #5 is gated high.	RW	0

**Table 3-1471. Register Call Summary for Register AUXCLK5**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1472. RSTTIME\_REG**

<b>Address Offset</b>	0x0000 0400	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>		
<b>Description</b>	This register holds the reset time counter which is used to extend the reset lines beyond the release of the pad reset.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RSTTIME															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads returns 0.	R	0x00000000
3:0	RSTTIME	Holds the number of 32 kHz clock cycles for which the reset duration is extended. Values 0,1 and 2 are not allowed. 0x0: Reserved. 0x1: Reserved. 0x2: Reserved.	RW	0x4

**Table 3-1473. Register Call Summary for Register RSTTIME\_REG**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1474. C2CRSTCTRL**

<b>Address Offset</b>	0x0000 041C		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register controls the release of the external C2C interface reset lines.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WARMRST		COLDRST													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads returns 0.	R	0x0000 0000
1	WARMRST	This bit allows to release the warm reset line of the external C2C interface. [warm reset sensitive]  0x0: Clearing this bit has for effect to release the warm reset line of the external C2C interface. 0x1: Writing this bit has for effect to assert the warm reset line of the external C2C interface.	RW	1
0	COLDRST	This bit allows to release the cold reset line of the external C2C interface.  0x0: Clearing this bit has for effect to release the cold reset line of the external C2C interface. 0x1: Writing this bit has for effect to assert the cold reset line of the external C2C interface.	RW	1

**Table 3-1475. Register Call Summary for Register C2CRSTCTRL**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1476. EXTPWRONRSTCTRL**

<b>Address Offset</b>	0x0000 0420		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register allows the software to perform an external power-on reset.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWRONRST		ENABLE													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads returns 0.	R	0x0000 0000
1	PWRONRST	This bit controls the assertion and the de-assertion of the external power-on reset.  0x0: De-asserts the external power-on reset. 0x1: Asserts the external power-on reset.	RW WSpecial	0

Bits	Field Name	Description	Type	Reset
0	ENABLE	This bit must be set to 1 to allow the software to assert the external power-on reset. 0x0: Prevents the software to assert the external power-on reset. 0x1: Allows the software to assert the external power-on reset.	RW WSpecial	0

**Table 3-1477. Register Call Summary for Register EXTPWRONRSTCTRL**

Reset Management Functional Description

- [SCRM Reset Management Functional Description: \[0\] \[1\]](#)

SCRM Register Manual

- [SCRM Register Summary: \[2\]](#)

**Table 3-1478. EXTWARMRSTST\_REG**

<b>Address Offset</b>	0x0000 0510	
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b> SCRM
<b>Description</b>	This register logs the source of warm reset output. Each bit is set upon release of the warm reset output and must be cleared by software.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
EXTWARMRSTST																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0.	R	0x0000 0000
0	EXTWARMRSTST	This bit logs the external warm reset source. 0x0: No external warm reset occurred. 0x1: An external warm reset occurred.	RW W1toClr	0

**Table 3-1479. Register Call Summary for Register EXTWARMRSTST\_REG**

Reset Management Functional Description

- [SCRM Reset Management Functional Description: \[0\]](#)

SCRM Register Manual

- [SCRM Register Summary: \[1\]](#)

**Table 3-1480. APEWARMRSTST\_REG**

<b>Address Offset</b>	0x0000 0514		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register logs the source of warm reset on the APE. Each bit is set upon release of the APE warm reset and must be cleared by software.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											APEWARMRSTST	RESERVED			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads returns 0.	R	0x0000 0000
1	APEWARMRSTST	This bit logs the APE warm reset source. 0x0: No APE warm reset occurred. 0x1: An APE warm reset occurred.	RW W1toClr	0
0	RESERVED	Reads returns 0.	R	0

**Table 3-1481. Register Call Summary for Register APEWARMRSTST\_REG**

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)

**Table 3-1482. C2CWARMRSTST\_REG**

<b>Address Offset</b>	0x0000 051C		
<b>Physical Address</b>	See <a href="#">Table 3-1429</a>	<b>Instance</b>	SCRM
<b>Description</b>	This register logs the source of warm reset on the external C2C interface. Each bit is set upon release of the external C2C interface warm reset and must be cleared by software.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											C2CWARMRSTST	RESERVED			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads returns 0.	R	0x00000000
3	C2CWARMRSTST	This bit logs the C2C warm reset source. 0x0: No C2C warm reset occurred. 0x1: A C2C warm reset occurred.	RW W1toClr	0
2:0	RESERVED	Reads returns 0.	R	0x0



---

**Table 3-1483. Register Call Summary for Register C2CWARMRSTST\_REG**

---

SCRM Register Manual

- [SCRM Register Summary: \[0\]](#)
- 

PRELIMINARY

### 3.13 SR Register Manual

#### 3.13.1 SR Instance Summary

**Table 3-1484. SR Instance Summary**

Module Name	L4 Base Address	Size
SR_MPU	0x4A0D 9000	256Bytes
SR_IVA	0x4A0D B000	256Bytes
SR_CORE	0x4A0D D000	256Bytes

#### 3.13.2 SR Registers

##### 3.13.2.1 SR Register Summary

**Table 3-1485. SR Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SR_MPU L4 Base Address	SR_IVA L4 Base Address	SR_CORE L4 Base Address
SRCONFIG	RW	32	0x0000 0000	0x4A0D 9000	0x4A0D B000	0x4A0D D000
SRSTATUS	R	32	0x0000 0004	0x4A0D 9004	0x4A0D B004	0x4A0D D004
SENVAL	R	32	0x0000 0008	0x4A0D 9008	0x4A0D B008	0x4A0D D008
SENMIN	R	32	0x0000 000C	0x4A0D 900C	0x4A0D B00C	0x4A0D D00C
SENMAX	R	32	0x0000 0010	0x4A0D 9010	0x4A0D B010	0x4A0D D010
SENAVG	R	32	0x0000 0014	0x4A0D 9014	0x4A0D B014	0x4A0D D014
AVGWEIGHT	RW	32	0x0000 0018	0x4A0D 9018	0x4A0D B018	0x4A0D D018
NVALUERECIPROCAL	RW	32	0x0000 001C	0x4A0D 901C	0x4A0D B01C	0x4A0D D01C
RESERVED	W	32	0x0000 0020	0x4A0D 9020	0x4A0D B020	0x4A0D D020
IRQSTATUS_RAW	RW	32	0x0000 0024	0x4A0D 9024	0x4A0D B024	0x4A0D D024
IRQSTATUS	RW	32	0x0000 0028	0x4A0D 9028	0x4A0D B028	0x4A0D D028
IRQENABLE_SET	RW	32	0x0000 002C	0x4A0D 902C	0x4A0D B02C	0x4A0D D02C
IRQENABLE_CLR	RW	32	0x0000 0030	0x4A0D 9030	0x4A0D B030	0x4A0D D030
SENEROR	R	32	0x0000 0034	0x4A0D 9034	0x4A0D B034	0x4A0D D034
ERRCONFIG	RW	32	0x0000 0038	0x4A0D 9038	0x4A0D B038	0x4A0D D038

##### 3.13.2.2 SR Register Description

**Table 3-1486. SRCONFIG**

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	Configuration bits for the Sensor Core and the Digital Processing.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCUMDATA								SRCLKLENGTH								SRENABLE	SENENABLE	ERRORGENERATORENABLE	MINMAXAVGENABLE	RESERVED				SENNENABLE	SENPENABLE						

Bits	Field Name	Description	Type	Reset
31:22	ACCUMDATA	Number of Values to Accumulate	RW	0x080
21:12	SRCLKLENGTH	Determines frequency of SRCIk	RW	0x200
11	SRENABLE	0: Asynchronously resets MinMaxAvgAccumValid, MinMaxAvgValid, ErrorGeneratorValid, AccumData sensor, SRCIk counter, and MinMaxAvg registers. Also gates the clock for power savings and disables all the digital logic. , 1: Enables the module	RW	0
10	SENENABLE	0: All sensors disabled, 1: Sensors enabled per SenNEnable & SenPEnable	RW	1
9	ERRORGENERATORENABLE	0: Error Generator Module disabled, 1: Error Generator Module enabled	RW	0
8	MINMAXAVGENABLE	0: MinMaxAvg Detector Module disabled, 1: MinMaxAvg Detector Module enabled	RW	0
7:2	RESERVED	Reserved	RW	0x00
1	SENNENABLE	0: Disable SenN sensor, 1: Enable SenN sensor	RW	1
0	SENPENABLE	0: Disable SenP sensor, 1: Enable SenP sensor	RW	0

**Table 3-1487. Register Call Summary for Register SRCONFIG**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

PRCM Module Programming Guide

- [Global Initialization: \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [Changing OPP: \[19\]](#)

SR Register Manual

- [SR Register Summary: \[20\]](#)

**Table 3-1488. SRSTATUS**

<b>Address Offset</b>	0x0000 0004
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	Status bits that indicate that the values in the register are valid or events have occurred.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AVGERRVALID		MINMAXAVGVALID		ERRORGENERATORVALID		MINMAXAVGACCUMVALID									

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	AVGERRVALID	0: AvgError registers are not valid, 1: AvgError registers are valid.	R	0
2	MINMAXAVGVALID	0: SenVal, SenMin, SenMax, SenAvg registers are not valid, 1: SenVal, SenMin, SenMax, SenAvg registers are valid, but not necessarily fully accumulated	R	0
1	ERRORGENERATORVALID	0: SenError register do not have valid data, 1: SenError registers have valid data.	R	0
0	MINMAXAVGACCUMVALID	0: SenVal, SenMin, SenMax, SenAvg registers are not valid, 1: SenVal, SenMin, SenMax, SenAvg registers have valid, final data	R	0

**Table 3-1489. Register Call Summary for Register SRSTATUS**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

SR Register Manual

- [SR Register Summary: \[7\]](#)

**Table 3-1490. SENVAL**

<b>Address Offset</b>	0x0000 0008
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	The current sensor values from the Sensor Core.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENPVAL																SENNVAL															

Bits	Field Name	Description	Type	Reset
31:16	SENPVAL	The latest value of the SenPVal from the sensor core.	R	0x0000
15:0	SENNVAL	The latest value of the SenNVal from the sensor core.	R	0x0000

**Table 3-1491. Register Call Summary for Register SENVAL**

- Voltage Management Functional Description
- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\]](#)
- SR Register Manual
- [SR Register Summary: \[4\]](#)

**Table 3-1492. SENMIN**

**Address Offset** 0x0000 000C  
**Physical Address** See [Table 3-1485](#)  
**Description** The minimum sensor values.  
**Type** R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENPMIN																SENNMIN															

Bits	Field Name	Description	Type	Reset
31:16	SENPMIN	The minimum value of the SenPVal from the sensor core since the last restat operation.	R	0xFFFF
15:0	SENNMIN	The minimum value of the SenNVal from the sensor core since the last restat operation.	R	0xFFFF

**Table 3-1493. Register Call Summary for Register SENMIN**

- Voltage Management Functional Description
- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\]](#)
- SR Register Manual
- [SR Register Summary: \[4\]](#)

**Table 3-1494. SENMAX**

**Address Offset** 0x0000 0010  
**Physical Address** See [Table 3-1485](#)  
**Description** The maximum sensor values.  
**Type** R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENPMAX																SENNMAX															

Bits	Field Name	Description	Type	Reset
31:16	SENPMAX	The maximum value of the SenPVal from the sensor core since the last restat operation.	R	0x0000
15:0	SENNMAX	The maximum value of the SenNVal from the sensor core since the last restat operation.	R	0x0000

**Table 3-1495. Register Call Summary for Register SENMAX**

- Voltage Management Functional Description
- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\]](#)
- SR Register Manual
- [SR Register Summary: \[4\]](#)

**Table 3-1496. SENAVG**

<b>Address Offset</b>	0x0000 0014
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	The average sensor values.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENPAVG																SENNAVG															

Bits	Field Name	Description	Type	Reset
31:16	SENPAVG	The running average of the SenPVal from the sensor core since the last restat operation.	R	0x0000
15:0	SENNAVG	The running average of the SenNVal from the sensor core since the last restat operation.	R	0x0000

**Table 3-1497. Register Call Summary for Register SENAVG**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\]](#)

SR Register Manual

- [SR Register Summary: \[4\]](#)

**Table 3-1498. AVGWEIGHT**

<b>Address Offset</b>	0x0000 0018
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	The weighting factor in the average computation.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
<table border="1"> <tr> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">SENPAVGWEIGHT</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">SENNAVGWEIGHT</td> </tr> </table>																												SENPAVGWEIGHT	SENNAVGWEIGHT		
SENPAVGWEIGHT	SENNAVGWEIGHT																														

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved bits.	R	0x00000000
3:2	SENPAVGWEIGHT	The weighting factor for the SenP averager.	RW	0x0
1:0	SENNAVGWEIGHT	The weighting factor for the SenN averager.	RW	0x0

**Table 3-1499. Register Call Summary for Register AVGWEIGHT**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Module Programming Guide

- [Global Initialization: \[4\] \[5\] \[6\] \[7\]](#)

SR Register Manual

- [SR Register Summary: \[8\]](#)

**Table 3-1500. NVALUERECIPROCAL**

<b>Address Offset</b>	0x0000 001C
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	The reciprocal of the SenN and SenP values used in error generation.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SENPAIN				SENNAIN				SENPRN				SENNRN											

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved bits.	R	0x00
23:20	SENPAIN	The gain value for the SenP reciprocal.	RW	0x0
19:16	SENNAIN	The gain value for the SenN reciprocal.	RW	0x0
15:8	SENPRN	The scale value for the SenP reciprocal.	RW	0x00
7:0	SENNRN	The scale value for the SenN reciprocal.	RW	0x00

**Table 3-1501. Register Call Summary for Register NVALUERECIPROCAL**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

PRCM Module Programming Guide

- [Global Initialization: \[8\] \[9\] \[10\] \[11\]](#)

SR Register Manual

- [SR Register Summary: \[12\]](#)

**Table 3-1502. IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0024
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	MCU raw interrupt status and set.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MCUACCUMINTSTATRAW				MCVALIDINTSTATRAW		MCBOUNDSINTSTATRAW		MCUDISABLEACKINTSTATRAW							

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved bits.	R	0x00000000
3	MCUACCUMINTSTATRAW	0: Accum interrupt status is unchanged, 1: Accum interrupt status is set	RW	0
2	MCVALIDINTSTATRAW	0: Valid interrupt status is unchanged, 1: Valid interrupt status is set	RW	0
1	MCBOUNDSINTSTATRAW	0: Bounds interrupt status is unchanged, 1: Bounds interrupt status is set	RW	0

Bits	Field Name	Description	Type	Reset
0	MCUDISABLEACKINTSTATRAW	0: MCUDisable acknowledge status is unchanged, 1: MCUDisable acknowledge status is set	RW	0

**Table 3-1503. Register Call Summary for Register IRQSTATUS\_RAW**

SR Register Manual

- [SR Register Summary: \[0\]](#)

**Table 3-1504. IRQSTATUS**

<b>Address Offset</b>	0x0000 0028
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	MCU masked interrupt status and clear.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
																MCUACCUMINTSTATENA				MCUVALIDINTSTATENA				MCUBOUNDSINTSTATENA				MCUDISABLEACKINTSTATENA			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved bits	R	0x00000000
3	MCUACCUMINTSTATENA	Read 0: Accum interrupt status is unchanged. Read 1: Accum interrupt status is set. Write 0: Accum interrupt status is unchanged. Write 1: Accum interrupt status is cleared.	RW	0
2	MCUVALIDINTSTATENA	Read 0: Valid interrupt status is unchanged. Read 1: Valid interrupt status is set. Write 0: Valid interrupt status is unchanged. Write 1: Valid interrupt status is cleared.	RW	0
1	MCUBOUNDSINTSTATENA	Read 0: Bounds interrupt status is unchanged. Read 1: Bounds interrupt status is set. Write 0: Bounds interrupt status is unchanged. Write 1: Bounds interrupt status is cleared.	RW	0
0	MCUDISABLEACKINTSTATENA	Read 0: MCUDisable acknowledge status is unchanged. Read 1: MCUDisable acknowledge status is set. Write 0: MCUDisable interrupt status is unchanged. Write 1: MCUDisable interrupt status is cleared.	RW	0

**Table 3-1505. Register Call Summary for Register IRQSTATUS**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Register Manual

- [INTRCONN\\_SOCKET\\_PRM Register Description: \[4\] \[5\] \[6\] \[7\]](#)



**Table 3-1505. Register Call Summary for Register IRQSTATUS (continued)**

- SR Register Manual
- [SR Register Summary: \[8\]](#)

**Table 3-1506. IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 002C
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	MCU interrupt enable flag and set.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MCUACCUMINTENASET		MCUVALIDINTENASET		MCUBOUNDSINTENASET		MCUDISABLEACTINTENASET									

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved bits.	R	0x00000000
3	MCUACCUMINTENASET	Read mode: 0: Accum interrupt generation is disabled/masked, 1: Accum interrupt generation is enabled; Write mode: 0: No change to Accum interrupt enable, 1: Enable Accum interrupt generation.	RW	0
2	MCUVALIDINTENASET	Read mode: 0: Valid interrupt generation is disabled/masked, 1: Valid interrupt generation is enabled; Write mode: 0: No change to Valid interrupt enable, 1: Enable Valid interrupt generation.	RW	0
1	MCUBOUNDSINTENASET	Read mode: 0: Bounds interrupt generation is disabled/masked, 1: Bounds interrupt generation is enabled; Write mode: 0: No change to Bounds interrupt enable, 1: Enable Bounds interrupt generation.	RW	0
0	MCUDISABLEACTINTENASET	Read mode: 0: MCUDisableAck interrupt generation is disabled/masked, 1: MCUDisableAck interrupt generation is enabled; Write mode: 0: No change to MCUDisAck interrupt enable, 1: Enable MCUDisableAck interrupt generation.	RW	0

**Table 3-1507. Register Call Summary for Register IRQENABLE\_SET**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\]](#)

PRCM Module Programming Guide

- [Global Initialization: \[4\] \[5\] \[6\]](#)

SR Register Manual

- [SR Register Summary: \[7\]](#)

**Table 3-1508. IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 0030
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	MCU interrupt enable flag and clear.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MCUACCUMINTENACL			MCUVALIDINTENACL		MCUBOUNDSINTENACL		MCUDISABLEACKINTENACL								

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved bits.	R	0x00000000
3	MCUACCUMINTENACL	Read mode: 0: Accum interrupt generation is disabled/masked, 1: Accum interrupt generation is enabled; Write mode: 0: No change to Accum interrupt enable, 1: Enable Accum interrupt generation.	RW	0
2	MCUVALIDINTENACL	Read mode: 0: Valid interrupt generation is disabled/masked, 1: Valid interrupt generation is enabled; Write mode: 0: No change to Valid interrupt enable, 1: Enable Valid interrupt generation.	RW	0
1	MCUBOUNDSINTENACL	Read mode: 0: Bounds interrupt generation is disabled/masked, 1: Bounds interrupt generation is enabled; Write mode: 0: No change to Bounds interrupt enable, 1: Enable Bounds interrupt generation.	RW	0
0	MCUDISABLEACKINTENACL	Read mode: 0: MCUDisableAck interrupt generation is disabled/masked, 1: MCUDisableAck interrupt generation is enabled; Write mode: 0: No change to MCUDisAck interrupt enable, 1: Enable MCUDisableAck interrupt generation.	RW	0

**Table 3-1509. Register Call Summary for Register IRQENABLE\_CLR**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\]](#)

SR Register Manual

- [SR Register Summary: \[4\]](#)

**Table 3-1510. SENERROR**

<b>Address Offset</b>	0x0000 0034
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	The sensor error from the error generator.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AVGERROR						SENERROR									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved bits.	R	0x0000
15:8	AVGERROR	The average sensor error.	R	0x00
7:0	SENERROR	The percentage of sensor error.	R	0x00

**Table 3-1511. Register Call Summary for Register SENERROR**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)

SR Register Manual

- [SR Register Summary: \[6\]](#)
- [SR Register Description: \[7\]](#)

**Table 3-1512. ERRCONFIG**

<b>Address Offset</b>	0x0000 0038
<b>Physical Address</b>	See <a href="#">Table 3-1485</a>
<b>Description</b>	The sensor error configuration.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WAKEUPENABLE	IDLEMODE	VPBOUNDSINTSTATENA	VPBOUNDSINTENABLE	RESERVED	ERRWEIGHT	ERRMAXLIMIT								ERRMINLIMIT													

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved bits.	R	0x00
26	WAKEUPENABLE	Wakeup from MCU Interrupts enable.	RW	0
25:24	IDLEMODE	0b00: Force-Idle Mode, 0b01: No Idle Mode, 0b10: Smart-Idle Mode #2, 0b11: Smart-Idle-Wkup mode	RW	0x2
23	VPBOUNDSINTSTATENA	0: Bounds interrupt status is unchanged, 1: Bounds interrupt status is cleared.	RW	0
22	VPBOUNDSINTENABLE	0: Bounds interrupt disabled, 1: Bounds interrupt enabled.	RW	0
21:19	RESERVED	Reserved bits.	R	0x0
18:16	ERRWEIGHT	The AvgSenError weight.	RW	0x0
15:8	ERRMAXLIMIT	The upper limit of SenError for interrupt generation.	RW	0x7F
7:0	ERRMINLIMIT	The lower limit of SenError for interrupt generation.	RW	0x80

**Table 3-1513. Register Call Summary for Register ERRCONFIG**

Voltage Management Functional Description

- [Adaptive Voltage Scaling: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

PRCM Module Programming Guide

- [Global Initialization: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

SR Register Manual

- [SR Register Summary: \[15\]](#)

PRELIMINARY

## Dual Cortex-A9 MPU Subsystem

---

---

---

This chapter describes the dual Cortex™-A9 MPU subsystem.

Topic	Page
4.1 Dual Cortex-A9 MPU Subsystem Overview .....	1030
4.2 Dual Cortex-A9 MPU Subsystem Integration .....	1033
4.3 Dual Cortex-A9 MPU Subsystem Functional Description .....	1038
4.4 Dual Cortex-A9 MPU Subsystem Register Manual .....	1047

## 4.1 Dual Cortex-A9 MPU Subsystem Overview

### 4.1.1 Introduction

The dual Cortex™-A9 microprocessor unit (MPU) subsystem of the device is based on the symmetric multiprocessor (SMP) architecture, thus the dual Cortex-A9 MPU subsystem delivers higher performance and optimal power management, debug and emulation capabilities.

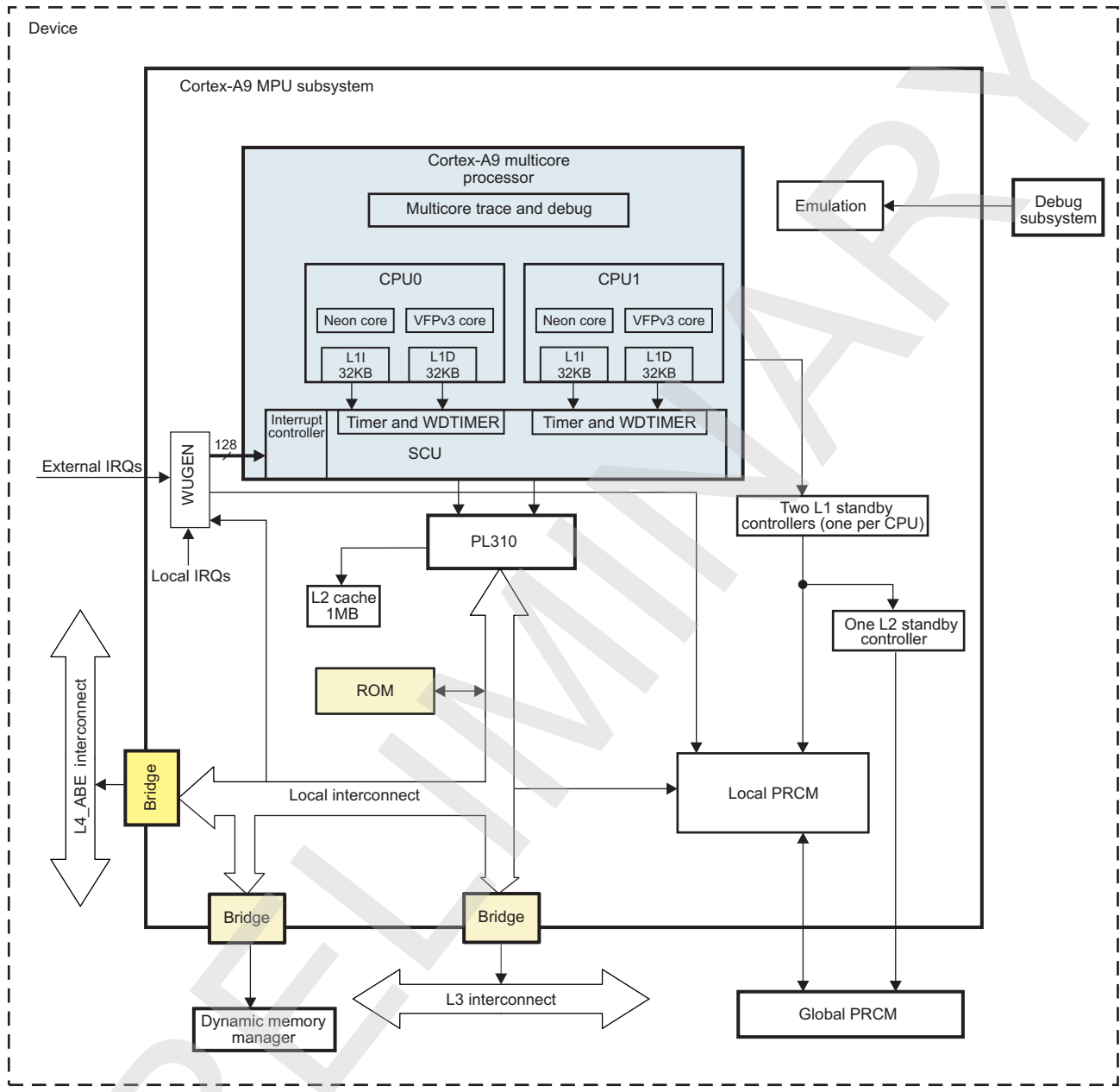
The dual Cortex-A9 MPU subsystem incorporates two Cortex-A9 central processing units (CPUs), level 2 (L2) cache shared between the two CPUs, and uses PL310 as L2 cache controller. Each CPU has 32KB of level 1 (L1) instruction cache, 32KB of L1 data cache, separate dedicated power domain, and includes one Neon™ and Vector Floating Point Unit (VFPv3) coprocessors. The dual Cortex-A9 MPU subsystem also includes standard CoreSight™ components to support SMP debug and emulation, snoop control unit (SCU), interrupt controller (GIC), and clock and reset manager.

The MPU subsystem handles transactions among the ARM® core, the L3 interconnect, the DMM (dynamic memory manager), the L4-ABE, and the interrupt controller (INTC).

From this point below, every reference to the dual Cortex-A9 MPU Subsystem, Cortex-A9 MPU subsystem, and MPU subsystem are equivalent.

[Figure 4-1](#) shows a high-level block diagram of the MPU subsystem.

Figure 4-1. Cortex-A9 MPU Subsystem Overview



mpu\_a9-001

### 4.1.2 Features

The Cortex-A9 MPU subsystem integrates the following:

- ARM subchip
  - Cortex-A9 core revision r1p2
  - SMP architecture
  - Superscalar, dynamic multiple issue technology with an efficient 8-stage pipeline
  - Out-of-order (OoO) instruction dispatch and completion
  - ARM Neon Advanced SIMD (single instruction, multiple data) support for accelerated media and signal processing computation.
  - Include VFPv3 hardware to support single and double-precision add, subtract, divide, multiply and accumulate, and square root operations
  - 32KB L1 instruction and 32KB L1 data cache – 32-byte line size, 4-way set associative
  - Memory management unit (MMU)
    - Two-level translation lookaside buffer (TLB) organization
    - First level is an 32-entry, fully associative micro-TLB implemented on the Instruction and Data side.
    - Second level is a unified, 2-way associative, 128-entry main TLB
    - Supports hardware TLB table-walk
  - SCU ensures memory coherency between the two CPUs
  - Integrated timer and watchdog timer per CPU
  - Interrupt controller with 128 hardware interrupt inputs
- PL310 L2 cache controller (revision r2p0) with 1MB cache size
  - 16-way set associative
  - 32-byte line size
  - Two slave ports and two master ports
  - Includes four 256-bit line-fill-buffers (LFBs) shared by the master ports
  - Each slave port includes two 256-bit line-read-buffers (LRBs)
  - Includes four 256-bit store buffers with merge capability
- Program trace macrocell (PTM)
- Emulation logic (cross-triggers)
- AMBA advanced trace bus (ATB) trace port
- Advanced peripheral bus (APB) port
- ROM memory
- Wake-up generator (WUGEN) module – Responsible for waking up the CPUs
- Standby controllers – The Cortex-A9 MPU implements a two-level standby controller consisting of two L1 standby controllers (one per CPU) and one L2 standby controller. The standby controllers handle the power transitions inside the Cortex-A9 MPU subsystem.
- Interface to audio back-end (ABE) subsystem: Enables direct connection between the Cortex-A9 multicore processor and the ABE module to reduce power consumption during long audio playback
- Power, clock, and reset manager: The Cortex-A9 MPU subsystem implements a local PRCM to handle CPU power domains.

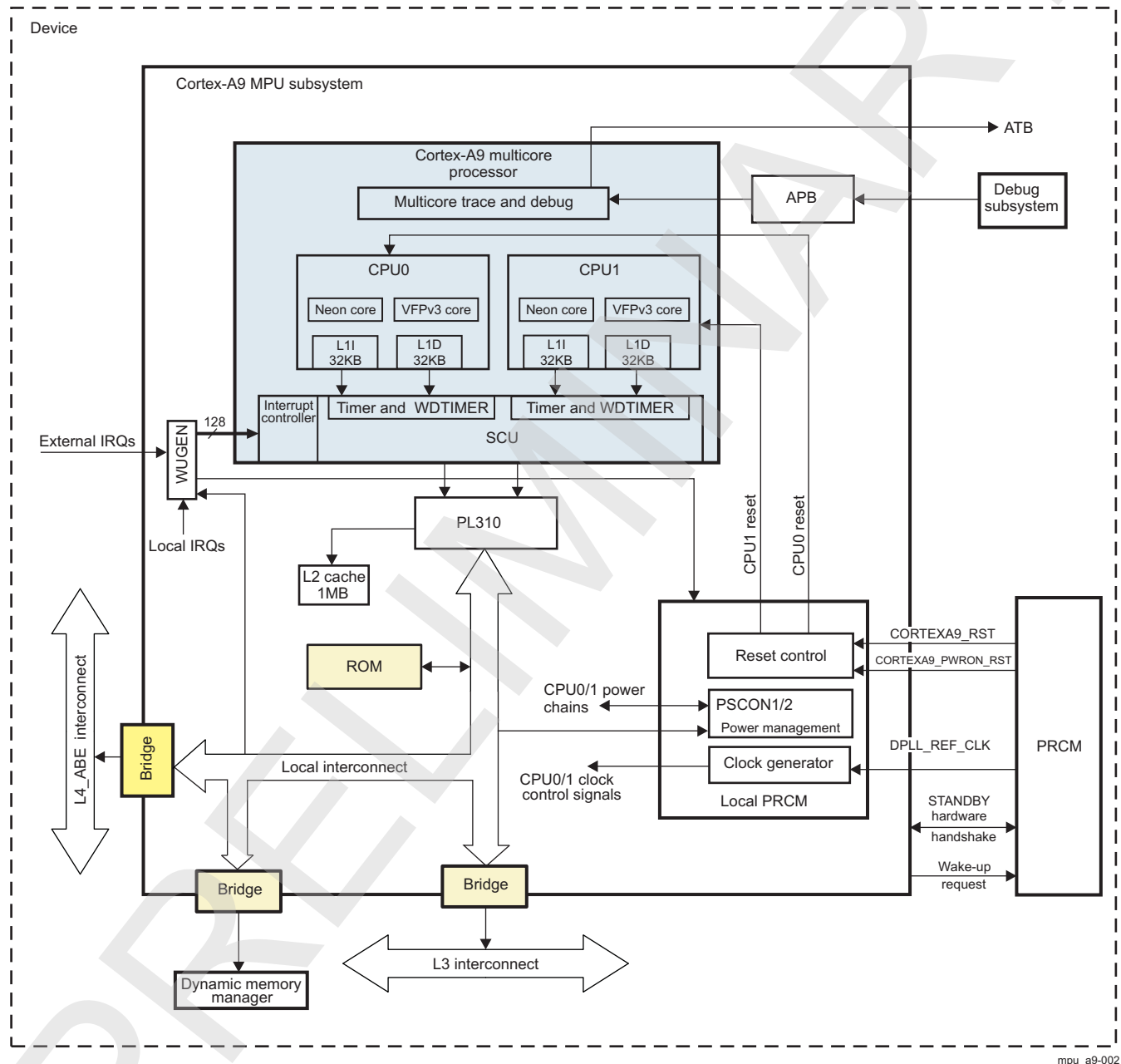


## 4.2 Dual Cortex-A9 MPU Subsystem Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 4-2 shows Cortex-A9 MPU integration.

Figure 4-2. Cortex-A9 MPU Subsystem Integration



mpu\_a9-002

**NOTE:** For more information about the STANDBY hardware handshake and the wake-up request, see Section 3.1.1.1, Clock Management in Chapter 3, Power, Reset, and Clock Management.

Table 4-1 and Table 4-2 summarize the integration of the module in the device.

**Table 4-1. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
CORTEXA9	PD_MPU	L3

**Table 4-2. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
CORTEXA9	MPU_DPLL_CLK	MPU_DPLL_CLK	PRCM	Interface and functional clock
Resets				
CORTEXA9	CORTEXA9_PWRON_RSTN	CORTEXA9_PWRON_RSTN	PRCM	Power-on reset for all the modules inside the MPU system power domain, nonretention
	CORTEXA9_RSTN	CORTEXA9_RSTN	PRCM	Warm reset for all the modules inside the MPU system power domain, nonretention

### 4.2.1 Clock Distribution

The Cortex-A9 MPU clock generator is fed by the MPU DPLL, which can be gated off by the global power, reset, and clock management (PRCM) module when system power domain is in a low-power state. There is a global clock gating for each CPU. Due to the MPU DPLL, the Cortex-A9 MPU subsystem is asynchronous from the rest of the device.

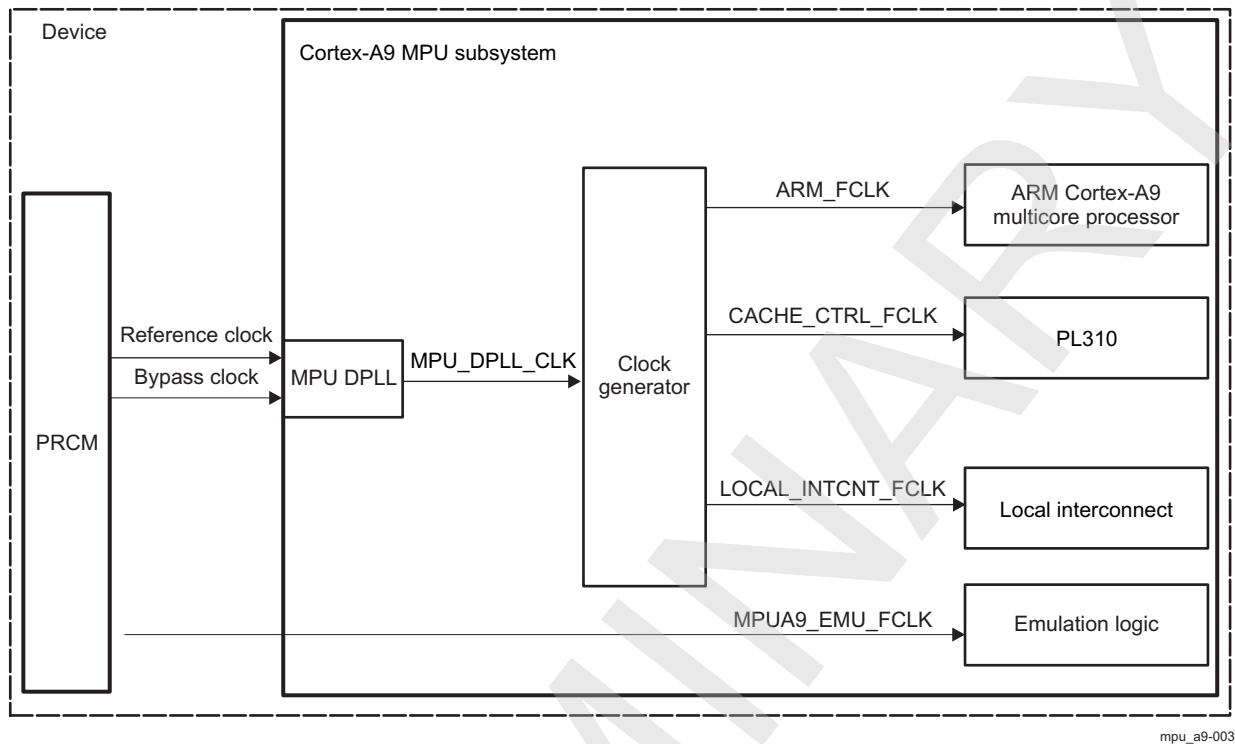
The clock generator generates the following clocks from the MPU DPLL output clock:

- ARM\_FCLK – ARM Cortex-A9 MPCore functional clock
- LOCAL\_INTCNT\_FCLK – Local interconnect functional clock
- CACHE\_CTRL\_FCLK – PL310 cache controller functional clock

The ARM\_FCLK and CACHE\_CTRL\_FCLK are with the same frequency. The LOCAL\_INTCNT\_FCLK frequency is half of the ARM\_FCLK.

Figure 4-3 shows the MPU subsystem clocking scheme.

Figure 4-3. Cortex-A9 MPU Subsystem Clocking Scheme



mpu\_a9-003

**NOTE:** For more information about the MPU DPLL, see [Section 3.6.3.3, Generic DPLL Overview](#) in [Chapter 3, Power, Reset, and Clock Management](#).

The reference clock in [Figure 4-3](#) is relevant to the DPLL\_REF\_CLK clock signal. The ARM Cortex-A9 multicore processor has no wake-up capabilities.

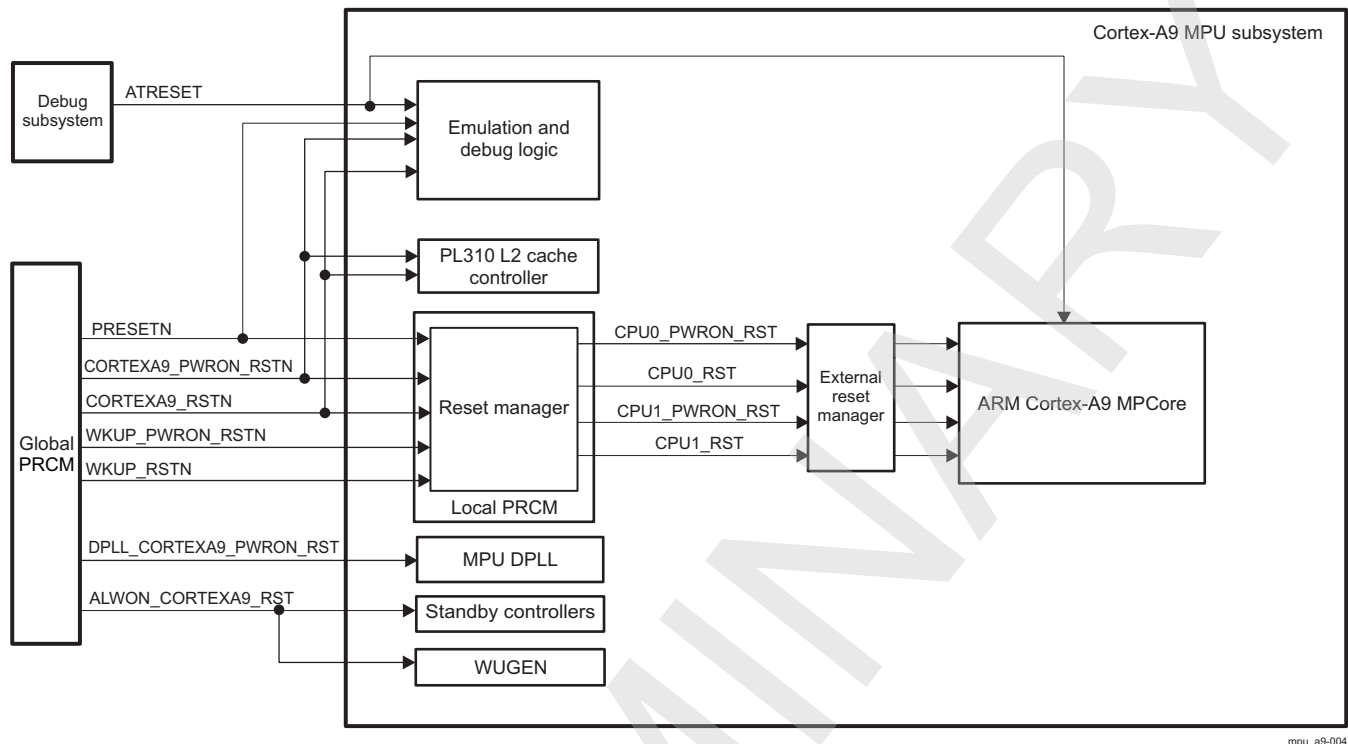
If only one Cortex-A9 CPU is needed, the ARM\_FCLK can be gated independently for each CPU. For more information about how to gate this clock, see [Section 4.3.4.2, Power States of CPU0 and CPU1](#).

The two Cortex-A9 CPUs cannot be clocked at different frequencies.

#### 4.2.2 Reset Distribution

Resets to the MPU subsystem are provided by the global PRCM module and controlled by the local PRCM.

For details about the power domains in the Cortex-A9 MPU subsystem, see [Section 4.3.4.1, Power Domains](#). [Figure 4-4](#) shows the reset scheme of the MPU subsystem.

**Figure 4-4. Cortex-A9 MPU Subsystem Reset Scheme**

mpu\_a9-004

All five external resets which are input to the local PRCM signals are active low. All eight external reset input signals are driven by the global PRCM, except the ATRESETN reset signal which is driven by the device debug subsystem. There are four internal reset signals, generated by the local PRCM.

ATRESETN is asserted at initial device power up and resets the debug and trace modules. ATRESETN affects Cortex-A9 MPU and ARM CPUs debug logic.

The PRESETN reset signal resets the debug logic in the Cortex-A9 MPU (CTI0/1, trace funnel, DAP\_PC) and ARM CPUs.

The WKUP\_PWRON\_RSTN reset signal is global cold reset for the wake-up logic and resets the wake-up domain logic (the PSCON modules) in the local PRCM. A cold reset is typically asserted when power is initially applied to the system. The user can check whether this reset event has occurred through reading the [PRM\\_RSTST\[0\] GLOBAL\\_COLD\\_RESET](#) bit.

The WKUP\_RSTN reset signal is global warm reset and resets the wake-up domain logic (the PSCON modules) in the local PRCM. Warm reset is typically used for resetting a system that has been operating for some time. The user can check whether this reset event has occurred through reading the [PRM\\_RSTST\[1\] GLOBAL\\_WARM\\_RST](#) bit.

The DPLL\_CORTEXA9\_PWRON\_RST reset signal resets the MPU\_DPLL.

The ALWON\_CORTEXA9\_RST reset signal resets the MPU away-on domain: The standby controllers and the WUGEN. The user can check whether the reset has occurred through reading the [WKG\\_CONTROL\\_0\[15\] DOMAIN\\_RST](#) bit for CPU0 and the [WKG\\_CONTROL\\_1\[15\] DOMAIN\\_RST](#) bit for CPU1.

The local PRCM provides two reset signals for each CPU:

- CPU0\_RST and CPU1\_RST reset signals are warm reset events. These reset signals initialize most of the ARM CPUs, except the debug logic (breakpoints and watchpoints are retained during this reset). The user can check whether these reset events have occurred through reading the [RM\\_PDA\\_CPUi\\_RSTCTRL\[0\] RST](#) bit.
- CPU0\_PWRON\_RST and CPU1\_PWRON\_RST reset signals are cold and debug reset events.

For details about clocks, resets, and power domains, and for information about the CORTEXA9\_PWRON\_RSTN and CORTEXA9\_RSTN reset signals, see [Chapter 3, Power, Reset, and Clock Management](#).

PRELIMINARY

## 4.3 Dual Cortex-A9 MPU Subsystem Functional Description

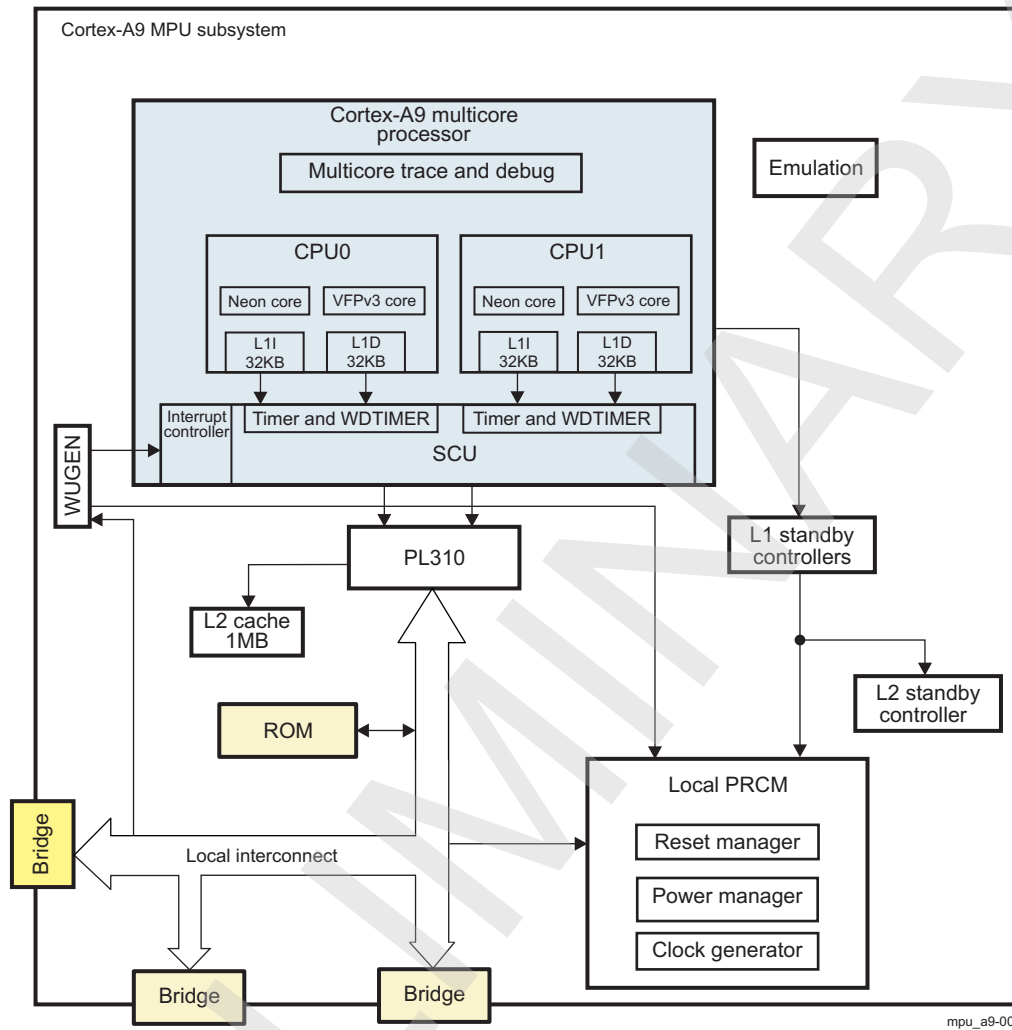
### 4.3.1 Cortex-A9 MPU Subsystem Block Diagram

The Cortex-A9 MPU subsystem integrates the following group of submodules:

- Two ARM Cortex-A9 CPUs. Each CPU contains:
  - ARM version 7 ISA™: Standard ARM instruction set plus Thumb®-2 , Jazelle® RCT and Jazelle DBX Java™ accelerator
  - Neon SIMD coprocessor and VFPv3
- INTC: Handles module interrupts (For details, see [Chapter 17, Interrupt Controller.](#))
- PL310 L2 cache controller (revision r2p0) with 1MB cache and two 64-bit slave and two 64-bit master ports (For details about L2 cache controller, see the ARM PL310 Cache Controller TRM.)
- Local interconnect: Connects the ARM Cortex-A9 multicore processor to the level 3 (L3) interconnect, dynamic memory manager, ABE interconnect, local PRCM, PL310 L2 cache controller, on-chip ROM memory, and the WUGEN.
- Power, clock and reset manager
- On-chip ROM memory – CPU0 (the master CPU) can boot from this memory. The ROM memory size is 48KB, the address range is from 0x4002 8000 to 0x4003 3FFF. For more information, see [Chapter 27, Initialization.](#)
- WUGEN: Responsible for waking up the CPUs, used by the ROM code and OS during SMP boot. For more information, see [Chapter 27, Initialization.](#)
- Standby controllers – handle the power transitions inside the Cortex-A9 MPU subsystem.
- SCU for L1 cache coherency
- One timer and one watchdog unit per core
- Caches memories:
  - 32-KB L1 instruction and 32-KB data caches – 4-way associative on each core
  - 1-MB L2 unified cache – 16-way associative

[Figure 4-5](#) is the Cortex-A9 MPU block diagram.

Figure 4-5. Cortex-A9 MPU Block Diagram



### 4.3.2 ARM Core

The Cortex-A9 MPU subsystem implements ARM version 7 ISA. Table 4-3 lists the main functions of the Cortex-A9 core.

Table 4-3. ARM Core Key Features

Feature	Comment
ARM version 7 ISA	Standard ARM instruction set + Thumb-2, Jazelle RCT/DBX Java accelerator, and media extensions. Backward-compatible with previous ARM ISA versions.
L1 Icache and Dcache	32KB each

**Table 4-3. ARM Core Key Features (continued)**

Feature	Comment
L2 cache (PL310)	Main characteristics are: <ul style="list-style-type: none"> <li>• Physically addressed and physically tagged</li> <li>• 16-way associative</li> <li>• 32-byte line length</li> <li>• Critical word first transactions</li> <li>• Prefetching capability</li> <li>• Pseudo-random victim selection policy</li> <li>• Two 256-bit LFBs in each master port</li> <li>• Two 256-bit LRBs in each slave port</li> <li>• Three 256-bit write buffers (merging capable)</li> <li>• Three 256-bit eviction buffers</li> <li>• Two 64-bit slave ports from SCU</li> <li>• Two 64-bit master ports, one to L3 and one to EMIF</li> </ul>
SCU	The SCU controls by hardware the coherency of the two Cortex-A9 MPCores L1 data caches. The SCU supports up to four outstanding reads for each CPU. The SCU can be given an address range that redirects all memory transactions within this range to the second master ports (to L2 cache controller).
TLB	2 levels: Micro TLB (32 entries on each of instruction and data sides) + Main TLB (instruction/data unified) implemented as a combination of a fully associative array of 4 elements (lockable) + a low-associative, 128 entries, 2-way
Trace interface	One PTM (Program Trace Macrocell) per core
Branch prediction	512 entries per core + global history buffer (GHB) of 4096 2-bit predictors + 8-entry return stack
Enhanced MMU	Mapping sizes are 4KB, 64KB, 1MB, and 16MB.
Neon	Includes advanced SIMD instructions and the ARM VFPv3 instructions

### 4.3.3 Local Interconnect

The local interconnect is used in the device design to connect the two 64-bit buses of the PL310 L2 cache controller to the L3 interconnect (64-bit width), EMIF interconnect port (128-bit width), ABE interconnect port (32-bit width), and local power manager. The local interconnect must do some minimal address decoding to decide where to forward the requests.

Main features:

- Connects to the EMIF through a 128-bit interconnect port
- Connects to the L3 interconnect through a 64-bit port
- Connects to the ABE through a 32-bit port
- Supports Single-Request-Multiple-Data (data handshaking) burst mode to pipeline requests
- Supports multiple outstanding requests

### 4.3.4 Power Management

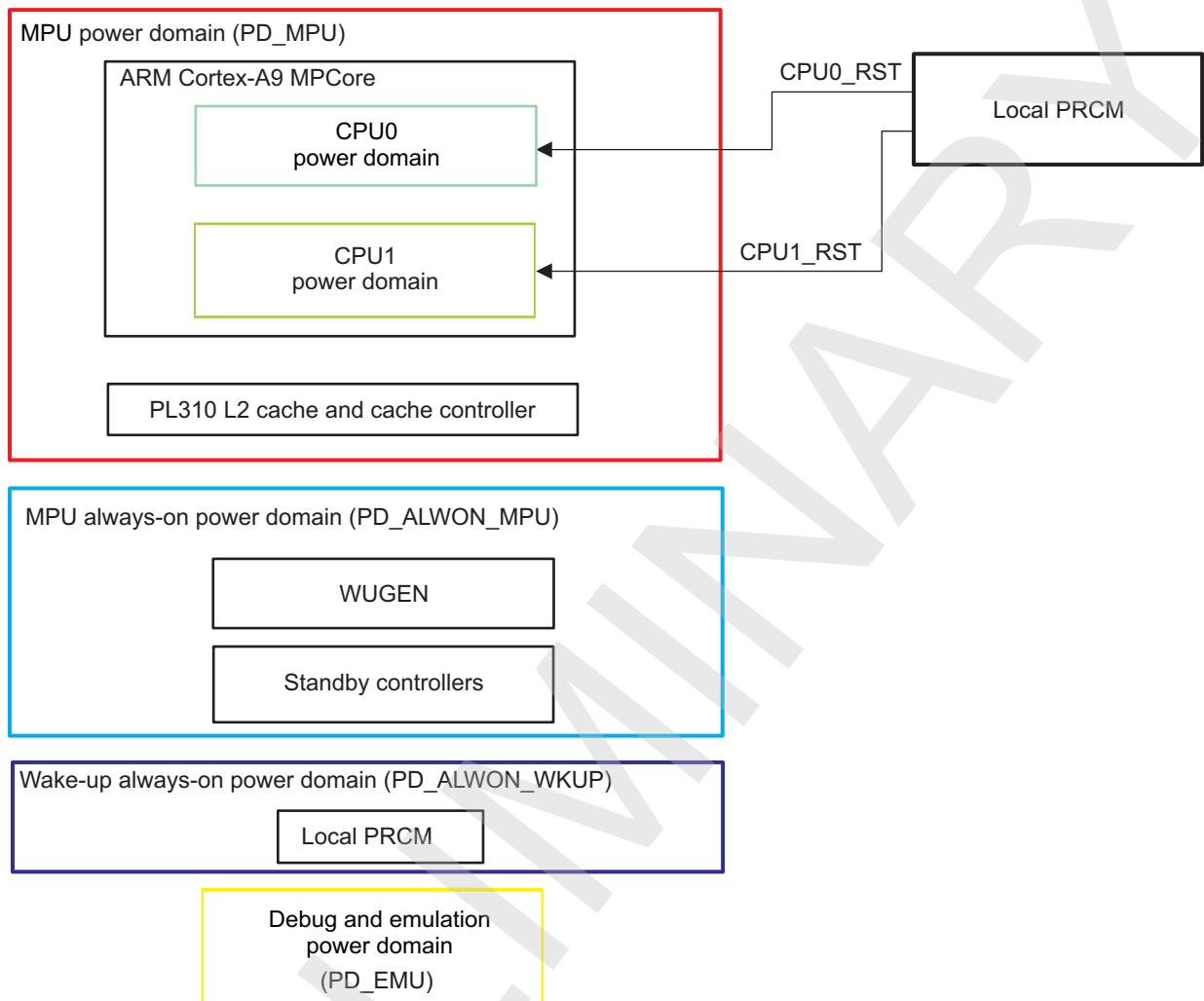
- The Cortex-A9 MPU subsystem implemented a local PRCM to handle local CPU power domain.
- Support cache retention
- The Cortex-A9 MPU subsystem has six power domains.
- The local PRCM handles the power management in conjunction with the global PRCM. The local PRCM has its own registers and is clocked by the system clock, same as the global PRCM.

#### 4.3.4.1 Power Domains

The Cortex-A9 MPU subsystem is divided into six power domains controlled by the local or global PRCM module, as shown in [Figure 4-6](#).



**Figure 4-6. Cortex-A9 MPU Subsystem Power Domain Overview**



mpu\_a9-005

The CPU power domains (CPU0 and CPU1) and L1 cache are controlled by the local PRCM. The system and emulation power domains (PD\_MPU and PD\_EMU) are controlled by the global PRCM. The power-management support for debug and emulation is distributed between local and global PRCM.

The Cortex-A9 local PRCM include two PSCON modules to control the power chains for CPU0 and CPU1. The [PRM\\_PSCON\\_COUNT](#) register is used for that control purpose.

To maximize performance and SMP scalability, both CPUs must run at the same frequency. Do not lower the frequency of one CPU as a power-saving option.

The device-level power domains are directly aligned with voltage domains and thus can be represented as a cross-reference to the different voltage domains.

For information about the physical power domains (PD\_MPU, PD\_EMU, and PD\_ALWON\_MPU) and the related voltage domains, see [Chapter 3, Power, Reset, and Clock Management](#).

#### 4.3.4.2 Power States of CPU0 and CPU1

CPU<sub>x</sub> (where x = 0 or 1) changes power states only when the StandbyWFI signal is asserted. The user can check this in the [WKG\\_CONTROL\\_0/WKG\\_CONTROL\\_1\[8\]](#) STANDBYWFI bit. In conjunction with StandbyWFI, the CPU power state transition is also based on the SCU CPU Power Status register, that the software should program before executing WFI instruction.

Each CPU clock can be gated independently. This clock gating is controlled by several hardware and software conditions.

Clock transition depends on the StandbyWFI assertion + the value of 3 bit fields:

- PWRCTLO from the SCU power status register (internal to ARM)
- CLKTRCTRL from the CM\_PDA\_CPUi\_CLKSTCTRL register (local PRCM)
- POWERSTATE from the PM\_PDA\_CPUi\_PWRSTCTRL register (local PRCM)

The power states of both CPUs does not have to be the same. The power state of CPUx can be:

- Normal (run) mode
- Dormant mode
- Power-off mode

Table 4-4 gives details the power state of the supported CPUx with corresponding local PRCM register values:

**Table 4-4. CPUx Supported Power States**

Hardware Conditions		CPUx Programming Model				Resulting CPUx State			
StandbyWFI StandbyWFE	SCU CPUx Power Status	Local PRCM Power State PM_PDA_CPUi_PWRSTCTRL[1:0] POWERSTATE	Local PRCM Logic Retention State PM_PDA_CPUi_PWRSTCTRL[2] LOGICRETSTATE	Local PRCM L1 Cache Memory Retention State PM_PDA_CPUi_PWRSTCTRL[8] L1_BANK_RETSTATE	Local PRCM Clock Transition Control CM_PDA_CPUi_CLKSTCTRL[1:0] CLKTRCTRL	Logic	L1 Cache	ARM Cortex-A9 Internal Clock Gating	Power State at Local PRCM
CPUs running <sup>(1)</sup>	Any	Any	Any	Any	Any	ON	ON	ON	ON
CPUs in WFE <sup>(2)</sup>	Any	Any	Any	Any	Any	ON	ON	OFF	ON
CPUs in WFI	Normal	Any	Any	Any	NO_SLEEP/SW_WKUP	ON	ON	OFF	ON
		ON	Any	Any	HW_AUTO				
CPUs in WFI	Dormant	RETENTION	Open switch retention (OSW)	Retention	HW_AUTO	OFF	RET	OFF	Open switch retention
CPUs in WFI	Power off	OFF	Any	Any	HW_AUTO	OFF	OFF	OFF	OFF

<sup>(1)</sup> When CPUx is running, no power transition is possible.

<sup>(2)</sup> When CPUx is in WFE mode, no power transition is possible.

The PM\_PDA\_CPUi\_PWRSTCTRL register is static over any power transition. That is, software program it before executing the WFI instruction, then does not change it until CPUx is again in running mode. Another way to say is: once CPUx has reached a low-power state (DORMANT, Power OFF), it cannot move to another low-power state. It must be woken up in order to reach another low-power state. To wake up CPUx the user must:

1. Execute a forced wake-up transition to the CPUx: CM\_PDA\_CPUx\_CLKSTCTRL[1:0] CLKTRCTRL = 0x2
2. The CPUx interrupt handler must set back the automatic hardware transition CM\_PDA\_CPUx\_CLKSTCTRL[1:0] CLKTRCTRL = 0x3

The MPU subsystem power domain (PD\_MPU) must be at the higher or equal power state (state that consume more power) than the higher of the two CPUs. For example, it is illegal for the MPU subsystem power state to be OFF, while the power state of one or both of the CPUs is DORMANT. Software must make sure only legal power states are programmed. When an illegal state is entered, then the behavior of the hardware is unpredictable. Do not program the Cortex-A9 MPU in CSWRET, L2\$ in OFF or either or both CPUs in CSWRET.

Table 4-5 lists the MPU power states handled by the local and global PRCM.

**Table 4-5. MPU Power States Handled by the Local PRCM and Global (Device) PRCM**

CPU State at Local PRCM	PRCM Power State	Logic Retention State	L2 Cache Memory Retention State	PRCM Clock Control	Logic Resulting State	L2 Cache Memory Resulting State	DPLL Clock	MPU Power State at PRCM
At least one is ON	Any	Any	Any	Any	ON	ON	ON	ON
Power states of both CPUs are less than or equal to OSWRET	Any	Any	Any	no sleep/software wake-up	ON	ON	ON	ON
	ON	Any	Any	hardware auto	ON	ON	ON	ON
	INACTIVE	Any	Any	hardware auto	ON	ON	OFF	INACTIVE
Power states of both CPUs are less than or equal to OSWRET	RETENTION	CSW	RETENTION	hardware auto	ON	RETENTION	OFF	CSWRET
	RETENTION	OSW	RETENTION	hardware auto	OFF	RETENTION	OFF	OSWRET
Power states of both CPUs are less than or equal to OSWRET	RETENTION	OSW	OFF	hardware auto	OFF	OFF	OFF	OSWRET
	RETENTION	OSW	OFF	hardware auto	OFF	OFF	OFF	OSWRET
Power states of both CPUs are OFF	OFF	Any	Any	hardware auto	OFF	OFF	OFF	OFF

WFI (Wait For Interrupt) refers to a state where logic is closed switch, memory is ON, and clock is gated off.

DORMANT (for CPU0 or CPU1) refers to a state where logic is in opened or closed switch (depending on programming value), memory is in RETENTION, and clock is gated off.

ON/OFF (for system logic) refers to a state where logic is in opened or closed switch, depending on whether the higher of the two CPU power states is opened or closed switch. For example, if CPU0 is in opened switch dormant mode and CPU1 is in closed switch dormant mode, then system logic must be closed switch.

Once Cortex-A9 MPU has reached a low-power state (INACT, CSWRET, OSWRET, OFF), it cannot move to another low-power state. It must be woken up to reach another low power state.

For the CPU0 and CPU1 power states and power state transitions, refer to the ARM Cortex-A9 MPCore TRM.

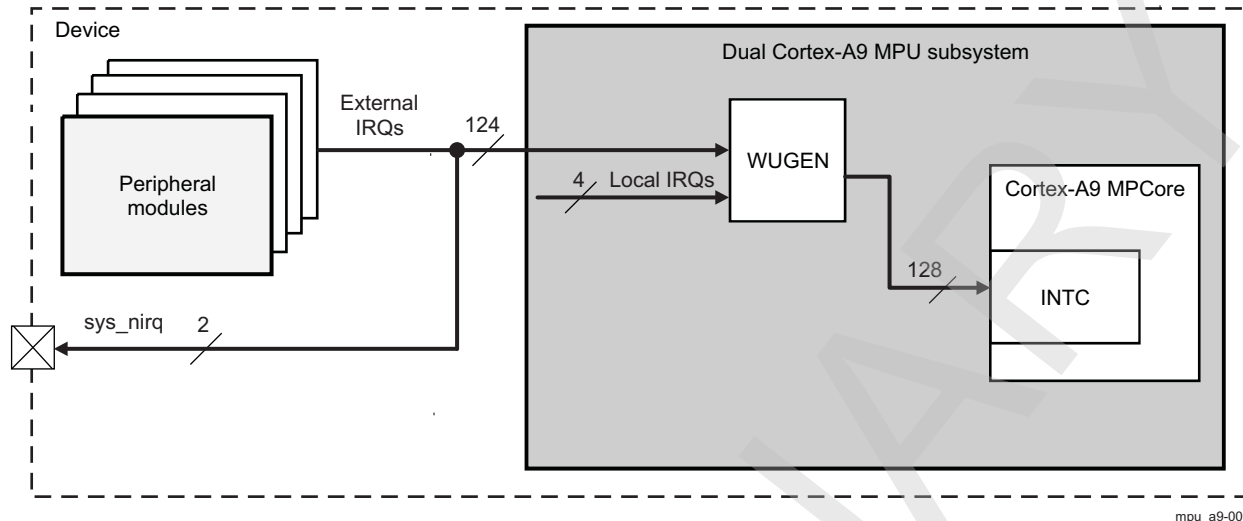
#### 4.3.4.3 WUGEN

The wake-up generator belongs to the MPU always-on power domain (PD\_ALWON\_MPU) and is responsible for generating wake-up event from the incoming interrupts (external and local) according to the enabled bits in the WUGEN registers.

The WUGEN unit has a 128-bit enable field (from [WKG\\_ENB\\_A\\_0](#) to [WKG\\_ENB\\_D\\_0](#) for CPU0 and from [WKG\\_ENB\\_A\\_1](#) to [WKG\\_ENB\\_D\\_1](#) for CPU1) which defines the interrupt that wakes up the corresponding CPU. The enable bits are reset to 0 and can be set to 1 to enable the interrupt to wake up the CPU. All interrupts are enabled after reset, except MA\_IRQ\_8 (which is not supported by the GP device). The Cortex-A9 MPU can access the WUGEN internal configuration registers through the local interconnect.

Software must program interrupt enabling and disabling coherently in the INTC and in the WUGEN enable registers. That is, a given interrupt for a given CPU is either enable at both INTC and WUGEN, or disable at both; no combination is allowed.

Figure 4-7 shows a functional overview of the Cortex-A9 MPU subsystem wake-up generator.

**Figure 4-7. Cortex-A9 MPU Subsystem Wake-up Generator**

#### 4.3.4.4 Power Transition Sequence

There are three types of power transition sequence:

- Power transitions that do not involve the local PRCM or global PRCM  
When the CPU enters the WFI state as long as the CPU power is not shut down or the DPLL is not gated off, then the Cortex-A9 internally gates off the CPU clock and no action is required by local or global PRCM. When an (enabled) interrupt is received, the Cortex-A9 exits the WFI state.
- Power transitions handled by the local PRCM  
When the CPU enters DORMANT or OFF states, the local PRCM must participate to perform the power sequencing. If the system power domain remains ON and the DPLL is not gated off, then the global PRCM does not need to be involved. You can enable and control the CPU domain power mode transitions writing in the [CM\\_PDA\\_CPUi\\_CLKCTRL\[1:0\] CLKTRCTRL](#) bit field.
- Power transitions handled by the local PRCM and global PRCM  
When the CPU enters a power state that causes the system power domain to go into WFI, DORMANT, or OFF state, then the local PRCM and global PRCM must be involved in the power transition.

#### 4.3.4.5 CPU0 and CPU1 Power Domains Description

The PD\_CPU<sub>x</sub> (where x = 1 or 2) contains following reset domains:

- CPU0\_RST
- CPU0\_PWRON\_RST
- CPU1\_RST
- CPU1\_PWRON\_RST

The PD\_CPU<sub>x</sub> contains following clock domains:

- CD\_CPU0
- CD\_CPU1

[Table 4-6](#) lists the logic retention capability for each module of the PD\_CPU<sub>x</sub>.

**Table 4-6. PD\_CPU<sub>x</sub> Modules Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
CPU <sub>x</sub>	No	<a href="#">RM_PDA_CPUi_CONTEXT[0]</a> LOSTCONTEXT_DFF	None

The DFF-based logic is nonretention flip flop. For more information, see the PRCM chapter.

#### 4.3.4.5.1 Power Domain Modes

This section highlights the various power modes supported by the logic and memory areas of the power domain. It also identifies the software control and status bit fields associated. See [Section 3.1.1.2.1, Power Domain](#) in [Chapter 3, Power, Reset, and Clock Management](#), for a functional description of the power-management architecture of a generic power domain.

##### 4.3.4.5.1.1 Logic and Memory Area Power Modes

[Table 4-7](#) lists the power modes supported by the logic area of the power domain.

**Table 4-7. PD\_CPUx Logic Area Power Modes**

OFF	RETENTION-OSWR	RETENTION-CSWR	ON-INACTIVE	ON-ACTIVE
Available	Available	Available	Available	Available

[Table 4-8](#) lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic ON, Logic RETENTION, and Logic OFF columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module - Memory column).

**Table 4-8. PD\_CPUx Memory Area Power Modes**

Memory Bank	Module - Memory	Logic ON	Logic RETENTION	Logic OFF
CPU0_L1		ON	OFF, RETENTION	OFF
	CPU0 - CPU0_L1	always_on	software_control	
CPU1_L1		ON	OFF, RETENTION	OFF
	CPU1 - CPU1_L1	always_on	software_control	

##### 4.3.4.5.1.2 Logic and Memory Area Power Modes Control and Status

[Table 4-9](#) lists the power modes controls for the power domain.

**Table 4-9. PD\_CPUx Power Modes Control Parameters**

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area - State Control (Logic in RETENTION state)	CPUx_L1	PM_PDA_CPUi_PWRSTCTRL[8] L1_BANK_RETSTATE	Read only
Memory Area - State Control (Logic in ON state)	CPUx_L1	PM_PDA_CPUi_PWRSTCTRL[17:16] L1_BANK_ONSTATE	Read only
Logic Area - Retention State Control		PM_PDA_CPUi_PWRSTCTRL[2] LOGICRETSTATE	Read/Write
Power Domain - State Transition Control		PM_PDA_CPUi_PWRSTCTRL[1:0] POWERSTATE	Read/Write

[Table 4-10](#) lists the power modes status for the power domain.

**Table 4-10. PD\_CPUx Power Modes Status Parameters**

Parameter Name	Memory Bank	Status Bit Field
Memory Area - State Status	CPUx_L1	PM_PDA_CPUi_PWRSTST[5:4] L1_BANK_STATEST
Power Domain - State Transition Status		PM_PDA_CPUi_PWRSTST[20] INTRANSITION
Logic Area - State Status		PM_PDA_CPUi_PWRSTST[2] LOGICSTATEST
Power Domain - State Status		PM_PDA_CPUi_PWRSTST[1:0] POWERSTATEST

Table 4-11 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 4-11. Modules Clock Management Modes and Control**

Module	Clock Management Protocol	Status Bit Field	Role
CPUx	Master/Slave	CM_PDA_CPUi_CLKCTRL[0] STBYST	Standby Status

Table 4-12 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

**Table 4-12. Modules Slave Clock Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
CPUx	n/a	Available	n/a	CM_PDA_CPUi_CLKSTCTRL[1:0] MODULEMODE	Read/Write

## 4.4 Dual Cortex-A9 MPU Subsystem Register Manual

### 4.4.1 Cortex-A9 MPU Subsystem Instance Summary

**Table 4-13. Cortex-A9 MPU Instance Summary**

Module Name	Base Address	Size
SCU	0x4824 0000	128 bytes
GIC_Proc_Interface <sup>(1)</sup>	0x4824 0100	256 bytes
Timer	0x4824 0600	256 bytes
GIC_Intr_Distributor <sup>(2)</sup>	0x4824 1000	4KB
PL310	0x4824 2000	4KB
CORTEXA9_SOCKET_PRCM	0x4824 3000	512 bytes
CORTEXA9_PRM	0x4824 3200	512 bytes
CORTEXA9_CPU0	0x4824 3400	1KB
CORTEXA9_CPU1	0x4824 3800	1KB
CORTEXA9_WUGEN	0x4828 1000	4KB

<sup>(1)</sup> Processor interfaces – handles interrupt acknowledgement, interrupt masking, and interrupt completion acknowledgement.

<sup>(2)</sup> Interrupt distributor – handles interrupt detection, interrupt prioritization, and distribution of interrupts to the CPUs.

#### 4.4.2 SCU Registers

For information about the SCU registers and their description, see the ARM® Cortex™-A9 MPCore Technical Reference Manual .

#### 4.4.3 Interrupt Controller Registers

For information about the GIC registers and their description, see the ARM® Cortex™-A9 MPCore Technical Reference Manual .

#### 4.4.4 Timer Registers

For information about the Timer registers and their description, see the ARM® Cortex™-A9 MPCore Technical Reference Manual .

#### 4.4.5 PL310 Registers

For information about the PL310 registers and their description, see the ARM® PL310 Cache Controller Technical Reference Manual .

#### 4.4.6 Local PRCM Revision Register

##### 4.4.6.1 Local PRCM Revision Register Summary

**Table 4-14. Local PRCM Revision Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CORTEXA9_SOCKET_P RCM Base Address
<a href="#">CORTEXA9_PRCM_REVISION</a>	R	32	0x0	0x4824 3000

##### 4.4.6.2 Local PRCM Revision Register Description

**Table 4-15. CORTEXA9\_PRCM\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 4-14</a> .	<b>Instance</b>	CORTEXA9_SOCKET_PRCM
<b>Description</b>	IP Revision register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI Internal data

**Table 4-16. Register Call Summary for Register CORTEXA9\_PRCM\_REVISION**

Cortex-A9 MPU Subsystem Register Manual

- [Local PRCM Revision Register Summary: \[0\]](#)

## 4.4.7 Local PRCM Registers

### 4.4.7.1 Local PRCM Registers Summary

**Table 4-17. Local PRCM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CORTEXA9_PRCM Base Address
<a href="#">PRM_RSTST</a>	RW	32	0x0000 0000	0x4824 3200
<a href="#">PRM_PSCON_COUNT</a>	RW	32	0x0000 0004	0x4824 3204

### 4.4.7.2 Local PRCM Log Register Description

**Table 4-18. PRM\_RSTST**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 4-17</a> .	<b>Instance</b>	CORTEXA9_PRCM
<b>Description</b>	This register logs the global reset sources, thus contains information regarding the cold/warm reset events generated by global PRCM. Each bit is set upon release of the domain reset signal. Must be cleared by software.		
<b>Type</b>	RW (W1toClr - write 0x1 to clear the bit)		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														GLOBAL_WARM_RST	GLOBAL_COLD_RST



Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1	GLOBAL_WARM_RST	Global warm reset event generated by Global PRCM 0x0: No global warm reset. 0x1: Global external warm reset has occurred.	RW W1toClr	0
0	GLOBAL_COLD_RST	Power-on (cold) reset event generated by global PRCM 0x0: No power-on reset. 0x1: Power-on reset has occurred.	RW W1toClr	1

**Table 4-19. Register Call Summary for Register PRM\_RSTST**

Cortex-A9 MPU Subsystem Integration

- [Reset Distribution: \[0\] \[1\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [Local PRCM Registers Summary: \[2\]](#)

**Table 4-20. PRM\_PSCON\_COUNT**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	CORTEXA9_PRM
<b>Physical Address</b>	See <a href="#">Table 4-17</a> .		
<b>Description</b>	Programmable Precharge count for L1Cache. This register is useful to ensure the correct delay between toggles of pscon lines and therefore to avoid Any problem of power switch transitions. The register corresponds to the number of device system clock cycles for the SRAM precharge duration. It means that the local PRCM waits for PCHARGE_TIME cycles of system clock. The pscon modules belongs to the local PRCM modules and are used to control the power of the CPUx.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPAREUSED								SPARE								PCHARGE_TIME															

Bits	Field Name	Description	Type	Reset
31:28	SPAREUSED	4 bits consumed for Last Power State Entered	R	0x0
27:8	SPARE	Spare programmable bits	RW	0x00000
7:0	PCHARGE_TIME	Programmable Precharge count during retention	RW	0x17

**Table 4-21. Register Call Summary for Register PRM\_PSCON\_COUNT**

Cortex-A9 MPU Subsystem Functional Description

- [Power Domains: \[0\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [Local PRCM Registers Summary: \[1\]](#)

#### 4.4.8 Local PRCM CPU0 and CPU1 Registers

##### 4.4.8.1 Local PRCM CPU0 and CPU1 Register Summary

**Table 4-22. Local PRCM CPU0 and CPU1 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CORTEXA9_CPU0 Base Address	CORTEXA9_CPU1 Base Address
<a href="#">PM_PDA_CPUi_PWRSTCTRL</a> <sup>(1)</sup>	RW	32	0x0000 0000	0x4824 3400	0x4824 3800
<a href="#">PM_PDA_CPUi_PWRSTST</a> <sup>(1)</sup>	RW	32	0x0000 0004	0x4824 3404	0x4824 3804
<a href="#">RM_PDA_CPUi_CONTEXT</a> <sup>(1)</sup>	RW	32	0x0000 0008	0x4824 3408	0x4824 3808

**Table 4-22. Local PRCM CPU0 and CPU1 Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	CORTEXA9_CPU0 Base Address	CORTEXA9_CPU1 Base Address
RM_PDA_CPUi_RSTCTRL <sup>(1)</sup>	RW	32	0x0000 000C	0x4824 340C	0x4824 380C
RESERVED	R	32	0x0000 0010	0x4824 3410	0x4824 3810
CM_PDA_CPUi_CLKCTRL <sup>(1)</sup>	R	32	0x0000 0014	0x4824 3414	0x4824 3814
CM_PDA_CPUi_CLKSTCTRL <sup>(1)</sup>	RW	32	0x0000 0018	0x4824 3418	0x4824 3818

(1) i = 0 or 1

**4.4.8.2 Local PRCM CPU0 Register Description****Table 4-23. PM\_PDA\_CPUi\_PWRSTCTRL**

<b>Address Offset</b>	0x0000 0000	<b>Index</b>	i = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 4-22</a> .	<b>Instance</b>	CORTEXA9_CPU0 CORTEXA9_CPU1
<b>Description</b>	This register controls the CPU domain power state to reach upon a domain sleep transition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								RESERVED															
								L1_BANK_ONSTATE								L1_BANK_RETSTATE								LOGICRETSTATE		POWERSTATE					

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:16	L1_BANK_ONSTATE	CPU_L1 memory state when domain is ON. Read 0x3: Memory bank is on when the domain is ON.	R	0x3
15:9	RESERVED	Reserved	R	0x00
8	L1_BANK_RETSTATE	CPU L1 memory state when domain is RETENTION state. Read 0x1: Memory bank is retained when domain is in RETENTION state.	R	1
7:3	RESERVED	Reserved	R	0x0
2	LOGICRETSTATE	Logic state control when power domain is RETENTION 0x0: All CPU logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	1
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x2

**Table 4-24. Register Call Summary for Register PM\_PDA\_CPUi\_PWRSTCTRL**

- Cortex-A9 MPU Subsystem Functional Description
- [Power States: \[0\] \[1\] \[2\] \[3\]](#)
  - [Logic and Memory Area Power Modes Control and Status: \[4\] \[5\] \[6\] \[7\]](#)
- Cortex-A9 MPU Subsystem Register Manual
- [Local PRCM CPU0 and CPU1 Register Summary: \[8\]](#)
  - [Local PRCM CPU0 Register Description: \[9\]](#)

**Table 4-25. PM\_PDA\_CPUi\_PWRSTST**

<b>Address Offset</b>	0x0000 0004	<b>Index</b>	i = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 4-22</a> .	<b>Instance</b>	CORTEXA9_CPU0 CORTEXA9_CPU1
<b>Description</b>	This register provides a status on the CPU domain current power state. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED		INTRANSITION	RESERVED											L1_BANK_STATEST		RESERVED	LOGICSTATEST	POWERSTATEST					
LASTPOWERSTATEENTERED																															

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x000
25:24	LASTPOWERSTATEENTERED	Last low power state entered. The software has to write 0x3 in this field to update this register. 0x0: Power domain was previously in OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously INACTIVE 0x3: Power domain was previously ON	RW W1toSet	0x0
23:21	RESERVED	Reserved	R	0x0
20	INTRANSITION	Domain transition status Read 0x0: No ongoing transition on power domain Read 0x1: Power domain transition is in progress.	R	0
19:6	RESERVED	Reserved	R	0x0000
5:4	L1_BANK_STATEST	CPU_L1 memory state status Read 0x0: Memory is OFF Read 0x1: Memory is RET Read 0x2: Reserved Read 0x3: Memory is ON	R	0x3
3	RESERVED	Reserved	R	0
2	LOGICSTATEST	Logic state status Read 0x0: Logic in domain is OFF Read 0x1: Logic in domain is ON	R	1

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATEST	Current power state status Read 0x0: Power domain is OFF Read 0x1: Power domain is in RETENTION Read 0x2: Power domain is ON-INACTIVE Read 0x3: Power domain is ON-ACTIVE	R	0x3

**Table 4-26. Register Call Summary for Register PM\_PDA\_CPUi\_PWRSTST**

Cortex-A9 MPU Subsystem Functional Description

- [Logic and Memory Area Power Modes Control and Status: \[0\] \[1\] \[2\] \[3\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [Local PRCM CPU0 and CPU1 Register Summary: \[4\]](#)

**Table 4-27. RM\_PDA\_CPUi\_CONTEXT**

<b>Address Offset</b>	0x0000 0008	<b>Index</b>	i = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 4-22</a> .	<b>Instance</b>	CORTEXA9_CPU0 CORTEXA9_CPU1
<b>Description</b>	This register contains dedicated CPU context statuses. [warm reset insensitive]		
<b>Type</b>	RW (W1toClr - write 0x1 to clear the bit)		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTMEM_CPU_L1	RESERVED										LOSTCONTEXT_DFF				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	LOSTMEM_CPU_L1	Specify if memory-based context in CPU_L1 memory bank has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1
7:1	RESERVED	Reserved	R	0x00
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source.  0x0: Context has been maintained 0x1: Context has been lost	RW W1toClr	1

**Table 4-28. Register Call Summary for Register RM\_PDA\_CPUi\_CONTEXT**

Cortex-A9 MPU Subsystem Functional Description

- [CPU0 and CPU1 Power Domains Description: \[0\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [Local PRCM CPU0 and CPU1 Register Summary: \[1\]](#)

**Table 4-29. RM\_PDA\_CPUi\_RSTCTRL**

<b>Address Offset</b>	0x0000 000C	<b>Index</b>	i = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 4-22</a> .	<b>Instance</b>	CORTEXA9_CPU0 CORTEXA9_CPU1
<b>Description</b>	This register controls the assertion/release of the CPU CORE reset. This is basically a software warm reset (that asserts CPUx_RST) per CPU. One CPU can set this bit to reset the other CPU. Actually the CPU can set this bit to reset itself as well (and it will be kept in reset until the other active CPU clears this bit).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RST			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RST	CPU warm local reset control 0x0: Reset is cleared 0x1: Reset is asserted	RW	0

**Table 4-30. Register Call Summary for Register RM\_PDA\_CPUi\_RSTCTRL**

Cortex-A9 MPU Subsystem Integration

- [Reset Distribution: \[0\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [Local PRCM CPU0 and CPU1 Register Summary: \[1\]](#)

**Table 4-31. CM\_PDA\_CPUi\_CLKCTRL**

<b>Address Offset</b>	0x0000 0014	<b>Index</b>	i = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 4-22</a> .	<b>Instance</b>	CORTEXA9_CPU0 CORTEXA9_CPU1
<b>Description</b>	This register manages the CPU clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STBYST			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	STBYST	Module standby status. [warm reset insensitive] Read 0x0: Module is functional (not in standby) Read 0x1: Module is in standby	R	1

**Table 4-32. Register Call Summary for Register CM\_PDA\_CPUi\_CLKCTRL**

Cortex-A9 MPU Subsystem Functional Description

- [Power Transition Sequence: \[0\]](#)
- [Logic and Memory Area Power Modes Control and Status: \[1\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [Local PRCM CPU0 and CPU1 Register Summary: \[2\]](#)

**Table 4-33. CM\_PDA\_CPUi\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0018	<b>Index</b>	i = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 4-22</a> .	<b>Instance</b>	CORTEXA9_CPU0 CORTEXA9_CPU1
<b>Description</b>	This register enables the CPU domain power state transition. It controls the hardware-supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also holds 1 status bit per clock input of the domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
CLKTRCTRL																															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	CLKTRCTRL	Controls the full domain transition of the CPU domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wake-up transition may however occur. Read 0x1: Reserved 0x2: Start a software forced wake-up transition on the domain. The software forced wake-up transition allows the software to wakeup a powered-down CPU by a method other than an externally triggered interrupt. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wake-up transition are based upon hardware conditions. WFI triggers the sleep transition based on settings in <a href="#">CM_PDA_CPUi_CLKSTCTRL</a> and <a href="#">PM_PDA_CPUi_PWRSTCTRL</a> registers.	RW	0x0

**Table 4-34. Register Call Summary for Register CM\_PDA\_CPUi\_CLKSTCTRL**

Cortex-A9 MPU Subsystem Functional Description

- [Power States: \[0\]](#)
- [Logic and Memory Area Power Modes Control and Status: \[1\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [Local PRCM CPU0 and CPU1 Register Summary: \[2\]](#)
- [Local PRCM CPU0 Register Description: \[3\]](#)

## 4.4.9 Wake-Up Generator Registers

### 4.4.9.1 CORTEXA9\_WUGEN Register Summary

**Table 4-35. CORTEXA9\_WUGEN Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Base Address
<a href="#">WKG_CONTROL_0</a>	R	32	0x0000 0000	0x4828 1000
<a href="#">WKG_ENB_A_0</a>	RW	32	0x0000 0010	0x4828 1010
<a href="#">WKG_ENB_B_0</a>	RW	32	0x0000 0014	0x4828 1014
<a href="#">WKG_ENB_C_0</a>	RW	32	0x0000 0018	0x4828 1018
<a href="#">WKG_ENB_D_0</a>	RW	32	0x0000 001C	0x4828 101C
RESERVED	R	32	0x0000 0020	0x4828 1020
RESERVED	R	32	0x0000 0024	0x4828 1024
RESERVED	R	32	0x0000 0028	0x4828 1028

**Table 4-35. CORTEXA9\_WUGEN Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Base Address
RESERVED	R	32	0x0000 002C	0x4828 102C
WKG_CONTROL_1	R	32	0x0000 0400	0x4828 1400
WKG_ENB_A_1	RW	32	0x0000 0410	0x4828 1410
WKG_ENB_B_1	RW	32	0x0000 0414	0x4828 1414
WKG_ENB_C_1	RW	32	0x0000 0418	0x4828 1418
WKG_ENB_D_1	RW	32	0x0000 041C	0x4828 141C
RESERVED	R	32	0x0000 0420	0x4828 1420
RESERVED	R	32	0x0000 0424	0x4828 1424
RESERVED	R	32	0x0000 0428	0x4828 1428
RESERVED	R	32	0x0000 042C	0x4828 142C
RESERVED	R	32	0x0000 0800	0x4828 1800
RESERVED	R	32	0x0000 0804	0x4828 1804
RESERVED	R	32	0x0000 0C00	0x4828 1C00
RESERVED	R	32	0x0000 0C04	0x4828 1C04
RESERVED	R	32	0x0000 0C08	0x4828 1C08
RESERVED	R	32	0x0000 0C0C	0x4828 1C0C

**4.4.9.2 CORTEXA9\_WUGEN Register Description**

**Table 4-36. WKG\_CONTROL\_0**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	CORTEXA9_WUGEN
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .		
<b>Description</b>	Wake-up generator status register for CPU0		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																DOMAIN_RST	CORTEXA9_WARM_RST	CORTEXA9_COLD_RST	WDT_RST	RESERVED	EVENTO	STANDBYWFE	STANDBYWF1	RESERVED									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x00000
15	DOMAIN_RST	MPU always-on power domain (PD_ALWON_MPU) reset status bit. It shows if the the reset occurred previously. 0x0: no reset occur 0x1: reset occur	R	0
14	CORTEXA9_WARM_RST	This bit is set when the CORTEXA9_RSTN signal is asserted. 0x0: CORTEXA9_RSTN reset signal has not been asserted 0x1: CORTEXA9_RSTN reset request has been asserted	R	0

Bits	Field Name	Description	Type	Reset
13	CORTEXA9_COLD_RST	This bit is set when the CORTEXA9_PWRON_RSTN signal is asserted. 0x0: CORTEXA9_PWRON_RSTN reset signal has not been asserted 0x1: CORTEXA9_PWRON_RSTN reset request has been asserted	R	0
12	WDT_RST	This bit is set when the WD timer Reset Request signal from the SCU is asserted. 0x0: WDT reset request has not been asserted 0x1: WDT reset request has been asserted	R	0
11	RESERVED	Reserved	R	0
10	EVENTO	EVENTO status bit. The event output signal is active, when one SEV instruction is executed. This bit is set when a rising edge of EVENTO from CPU is detected. 0x0: Rising edge of EVENTO is not detected 0x1: Rising edge of EVENTO is detected	R	0
9	STANDBYWFE	This bit gives software the visibility to track whether WFE mode have been entered. 0x0: WFE mode has not been entered 0x1: WFE mode has been entered	R	0
8	STANDBYWFI	This bit gives software the visibility to track whether WFI mode have been entered. 0x0: WFI mode has not been entered 0x1: WFI mode has been entered	R	0
7:0	RESERVED	Reserved	R	0

**Table 4-37. Register Call Summary for Register WKG\_CONTROL\_0**

Cortex-A9 MPU Subsystem Integration

- [Reset Distribution: \[0\]](#)

Cortex-A9 MPU Subsystem Functional Description

- [Power States: \[1\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[2\]](#)

**Table 4-38. WKG\_ENB\_A\_0**

<b>Address Offset</b>	0x0000 0010																																
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .																<b>Instance</b>																CORTEXA9_WUGEN
<b>Description</b>	This register enables the interrupts (for CPU0) from MA_IRQ_0 to MA_IRQ_31 write 0x0: disable interrupt write 0x1: enable interrupt																																
<b>Type</b>	RW																																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	WKG_ENB_FOR_INTR31	WKG_ENB_FOR_INTR30	WKG_ENB_FOR_INTR29	WKG_ENB_FOR_INTR28	WKG_ENB_FOR_INTR27	WKG_ENB_FOR_INTR26	WKG_ENB_FOR_INTR25	WKG_ENB_FOR_INTR24	WKG_ENB_FOR_INTR23	WKG_ENB_FOR_INTR22	WKG_ENB_FOR_INTR21	WKG_ENB_FOR_INTR20	WKG_ENB_FOR_INTR19	WKG_ENB_FOR_INTR18	WKG_ENB_FOR_INTR17	WKG_ENB_FOR_INTR16	WKG_ENB_FOR_INTR15	WKG_ENB_FOR_INTR14	WKG_ENB_FOR_INTR13	WKG_ENB_FOR_INTR12	WKG_ENB_FOR_INTR11	WKG_ENB_FOR_INTR10	WKG_ENB_FOR_INTR9	RESERVED	WKG_ENB_FOR_INTR7	RESERVED	RESERVED	WKG_ENB_FOR_INTR4	RESERVED	WKG_ENB_FOR_INTR2	WKG_ENB_FOR_INTR1	WKG_ENB_FOR_INTR0	



Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR31		RW	1
30	WKG_ENB_FOR_INTR30		RW	1
29	WKG_ENB_FOR_INTR29		RW	1
28	WKG_ENB_FOR_INTR28		RW	1
27	WKG_ENB_FOR_INTR27		RW	1
26	WKG_ENB_FOR_INTR26		RW	1
25	WKG_ENB_FOR_INTR25		RW	1
24	WKG_ENB_FOR_INTR24		RW	1
23	WKG_ENB_FOR_INTR23		RW	1
22	WKG_ENB_FOR_INTR22		RW	1
21	WKG_ENB_FOR_INTR21		RW	1
20	WKG_ENB_FOR_INTR20		RW	1
19	WKG_ENB_FOR_INTR19		RW	1
18	WKG_ENB_FOR_INTR18		RW	1
17	WKG_ENB_FOR_INTR17		RW	1
16	WKG_ENB_FOR_INTR16		RW	1
15	WKG_ENB_FOR_INTR15		RW	1
14	WKG_ENB_FOR_INTR14		RW	1
13	WKG_ENB_FOR_INTR13		RW	1
12	WKG_ENB_FOR_INTR12		RW	1
11	WKG_ENB_FOR_INTR11		RW	1
10	WKG_ENB_FOR_INTR10		RW	1
9	WKG_ENB_FOR_INTR9		RW	1
8	RESERVED		R	0 <sup>(1)</sup>
7	WKG_ENB_FOR_INTR7		RW	1
6	RESERVED		R	1
5	RESERVED		R	1
4	WKG_ENB_FOR_INTR4		RW	1
3	RESERVED		R	1
2	WKG_ENB_FOR_INTR2		RW	1
1	WKG_ENB_FOR_INTR1		RW	1
0	WKG_ENB_FOR_INTR0		RW	1

<sup>(1)</sup> The reset value is 0x0 by safety reasons.

**Table 4-39. Register Call Summary for Register WKG\_ENB\_A\_0**

Cortex-A9 MPU Subsystem Functional Description

- [Wake-up Generator \(WUGEN\): \[0\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[1\]](#)

**Table 4-40. WKG\_ENB\_B\_0**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .	<b>Instance</b>	CORTEXA9_WUGEN
<b>Description</b>	This register enables the interrupts (for CPU0) from MA_IRQ_32 to MA_IRQ_63 write 0x0: disable interrupt write 0x1: enable interrupt		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED	WKG_ENB_FOR_INTR62	WKG_ENB_FOR_INTR61	RESERVED	WKG_ENB_FOR_INTR59	WKG_ENB_FOR_INTR58	WKG_ENB_FOR_INTR57	WKG_ENB_FOR_INTR56	RESERVED	RESERVED	WKG_ENB_FOR_INTR53	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	WKG_ENB_FOR_INTR48	WKG_ENB_FOR_INTR47	WKG_ENB_FOR_INTR46	WKG_ENB_FOR_INTR45	WKG_ENB_FOR_INTR44	WKG_ENB_FOR_INTR43	WKG_ENB_FOR_INTR42	WKG_ENB_FOR_INTR41	WKG_ENB_FOR_INTR40	WKG_ENB_FOR_INTR39	WKG_ENB_FOR_INTR38	WKG_ENB_FOR_INTR37	WKG_ENB_FOR_INTR36	RESERVED	WKG_ENB_FOR_INTR34	WKG_ENB_FOR_INTR33	WKG_ENB_FOR_INTR32

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	1
30	WKG_ENB_FOR_INTR62		RW	1
29	WKG_ENB_FOR_INTR61		RW	1
28	RESERVED		R	1
27	WKG_ENB_FOR_INTR59		RW	1
26	WKG_ENB_FOR_INTR58		RW	1
25	WKG_ENB_FOR_INTR57		RW	1
24	WKG_ENB_FOR_INTR56		RW	1
23	RESERVED		R	1
22	RESERVED		R	1
21	WKG_ENB_FOR_INTR53		RW	1
20	RESERVED		R	1
19	RESERVED		R	1
18	RESERVED		R	1
17	RESERVED		R	1
16	WKG_ENB_FOR_INTR48		RW	1
15	WKG_ENB_FOR_INTR47		RW	1
14	WKG_ENB_FOR_INTR46		RW	1
13	WKG_ENB_FOR_INTR45		RW	1
12	WKG_ENB_FOR_INTR44		RW	1
11	WKG_ENB_FOR_INTR43		RW	1
10	WKG_ENB_FOR_INTR42		RW	1
9	WKG_ENB_FOR_INTR41		RW	1
8	WKG_ENB_FOR_INTR40		RW	1
7	WKG_ENB_FOR_INTR39		RW	1
6	WKG_ENB_FOR_INTR38		RW	1
5	WKG_ENB_FOR_INTR37		RW	1
4	WKG_ENB_FOR_INTR36		RW	1
3	RESERVED		R	1
2	WKG_ENB_FOR_INTR34		RW	1
1	WKG_ENB_FOR_INTR33		RW	1

Bits	Field Name	Description	Type	Reset
0	WKG_ENB_FOR_INTR32		RW	1

**Table 4-41. Register Call Summary for Register WKG\_ENB\_B\_0**

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[0\]](#)

**Table 4-42. WKG\_ENB\_C\_0**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	CORTEXA9_WUGEN
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .		
<b>Description</b>	This register enables the interrupts (for CPU0) from MA_IRQ_64 to MA_IRQ_95 write 0x0: disable interrupt write 0x1: enable interrupt		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKG_ENB_FOR_INTR94	WKG_ENB_FOR_INTR93	WKG_ENB_FOR_INTR92	WKG_ENB_FOR_INTR91	WKG_ENB_FOR_INTR90	WKG_ENB_FOR_INTR89	WKG_ENB_FOR_INTR88	WKG_ENB_FOR_INTR87	WKG_ENB_FOR_INTR86	RESERVED	WKG_ENB_FOR_INTR84	WKG_ENB_FOR_INTR83	RESERVED	RESERVED	WKG_ENB_FOR_INTR80	RESERVED	WKG_ENB_FOR_INTR78	WKG_ENB_FOR_INTR77	WKG_ENB_FOR_INTR76	WKG_ENB_FOR_INTR75	WKG_ENB_FOR_INTR74	WKG_ENB_FOR_INTR73	WKG_ENB_FOR_INTR72	WKG_ENB_FOR_INTR71	WKG_ENB_FOR_INTR70	WKG_ENB_FOR_INTR69	WKG_ENB_FOR_INTR68	WKG_ENB_FOR_INTR67	WKG_ENB_FOR_INTR66	WKG_ENB_FOR_INTR65	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	1
30	WKG_ENB_FOR_INTR94		RW	1
29	WKG_ENB_FOR_INTR93		RW	1
28	WKG_ENB_FOR_INTR92		RW	1
27	WKG_ENB_FOR_INTR91		RW	1
26	WKG_ENB_FOR_INTR90		RW	1
25	WKG_ENB_FOR_INTR89		RW	1
24	WKG_ENB_FOR_INTR88		RW	1
23	WKG_ENB_FOR_INTR87		RW	1
22	WKG_ENB_FOR_INTR86		RW	1
21	RESERVED		R	1
20	WKG_ENB_FOR_INTR84		RW	1
19	WKG_ENB_FOR_INTR83		RW	1
18	RESERVED		R	1
17	RESERVED		R	1
16	WKG_ENB_FOR_INTR80		RW	1
15	RESERVED		R	1
14	WKG_ENB_FOR_INTR78		RW	1
13	WKG_ENB_FOR_INTR77		RW	1
12	WKG_ENB_FOR_INTR76		RW	1
11	WKG_ENB_FOR_INTR75		RW	1
10	WKG_ENB_FOR_INTR74		RW	1
9	WKG_ENB_FOR_INTR73		RW	1
8	WKG_ENB_FOR_INTR72		RW	1

Bits	Field Name	Description	Type	Reset
7	WKG_ENB_FOR_INTR71		RW	1
6	WKG_ENB_FOR_INTR70		RW	1
5	WKG_ENB_FOR_INTR69		RW	1
4	WKG_ENB_FOR_INTR68		RW	1
3	WKG_ENB_FOR_INTR67		RW	1
2	WKG_ENB_FOR_INTR66		RW	1
1	WKG_ENB_FOR_INTR65		RW	1
0	RESERVED		R	1

**Table 4-43. Register Call Summary for Register WKG\_ENB\_C\_0**

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[0\]](#)

**Table 4-44. WKG\_ENB\_D\_0**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	CORTEXA9_WUGEN
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .		
<b>Description</b>	This register enables the interrupts (for CPU0) from MA_IRQ_96 to MA_IRQ_127 write 0x0: disable interrupt write 0x1: enable interrupt		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	WKG_ENB_FOR_INTR120	WKG_ENB_FOR_INTR119	RESERVED	RESERVED	RESERVED	RESERVED	WKG_ENB_FOR_INTR114	WKG_ENB_FOR_INTR113	WKG_ENB_FOR_INTR112	WKG_ENB_FOR_INTR111	WKG_ENB_FOR_INTR110	WKG_ENB_FOR_INTR109	RESERVED	WKG_ENB_FOR_INTR107	RESERVED	RESERVED	WKG_ENB_FOR_INTR104	WKG_ENB_FOR_INTR103	WKG_ENB_FOR_INTR102	WKG_ENB_FOR_INTR101	WKG_ENB_FOR_INTR100	WKG_ENB_FOR_INTR99	WKG_ENB_FOR_INTR98	WKG_ENB_FOR_INTR97	WKG_ENB_FOR_INTR96

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	1
30	RESERVED		R	1
29	RESERVED		R	1
28	RESERVED		R	1
27	RESERVED		R	1
26	RESERVED		R	1
25	RESERVED		R	1
24	WKG_ENB_FOR_INTR120		RW	1
23	WKG_ENB_FOR_INTR119		RW	1
22	RESERVED		R	1
21	RESERVED		R	1
20	RESERVED		R	1
19	RESERVED		R	1
18	WKG_ENB_FOR_INTR114		RW	1
17	WKG_ENB_FOR_INTR113		RW	1
16	WKG_ENB_FOR_INTR112		RW	1
15	WKG_ENB_FOR_INTR111		RW	1

Bits	Field Name	Description	Type	Reset
14	WKG_ENB_FOR_INTR110		RW	1
13	WKG_ENB_FOR_INTR109		RW	1
12	RESERVED		R	1
11	WKG_ENB_FOR_INTR107		RW	1
10	RESERVED		R	1
9	RESERVED		R	1
8	WKG_ENB_FOR_INTR104		RW	1
7	WKG_ENB_FOR_INTR103		RW	1
6	WKG_ENB_FOR_INTR102		RW	1
5	WKG_ENB_FOR_INTR101		RW	1
4	WKG_ENB_FOR_INTR100		RW	1
3	WKG_ENB_FOR_INTR99		RW	1
2	WKG_ENB_FOR_INTR98		RW	1
1	WKG_ENB_FOR_INTR97		RW	1
0	WKG_ENB_FOR_INTR96		RW	1

**Table 4-45. Register Call Summary for Register WKG\_ENB\_D\_0**

Cortex-A9 MPU Subsystem Functional Description

- [Wake-up Generator \(WUGEN\): \[0\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[1\]](#)

**Table 4-46. WKG\_CONTROL\_1**

<b>Address Offset</b>	0x0000 0400	<b>Instance</b>	CORTEXA9_WUGEN
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .		
<b>Description</b>	Wake-up generator status register for CPU1		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																DOMAIN_RST	CORTEXA9_WARM_RST	CORTEXA9_COLD_RST	WDT_RST	RESERVED	EVENTO	STANDBYWFE	STANDBYWFI	RESERVED									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x00000
15	DOMAIN_RST	MPU always-on power domain (PD_ALWON_MPU) reset status bit. It shows if the the reset occurred previously. 0x0: no reset occur 0x1: reset occur	R	0
14	CORTEXA9_WARM_RST	This bit is set when the CORTEXA9_RSTN signal is asserted. 0x0: CORTEXA9_RSTN reset signal has not been asserted 0x1: CORTEXA9_RSTN reset request has been asserted	R	0

Bits	Field Name	Description	Type	Reset
13	CORTEXA9_COLD_RST	This bit is set when the CORTEXA9_PWRON_RSTN signal is asserted. 0x0: CORTEXA9_PWRON_RSTN reset signal has not been asserted 0x1: CORTEXA9_PWRON_RSTN reset request has been asserted	R	0
12	WDT_RST	This bit is set when the WD timer Reset Request signal from the SCU is asserted. 0x0: WDT reset request has not been asserted 0x1: WDT reset request has been asserted	R	0
11	RESERVED	Reserved	R	0
10	EVENTO	EVENTO status bit. The event output signal is active, when one SEV instruction is executed. This bit is set when a rising edge of EVENTO from CPU is detected. 0x0: Rising edge of EVENTO is not detected 0x1: Rising edge of EVENTO is detected	R	0
9	STANDBYWFE	This bit gives software the visibility to track whether WFE mode have been entered. 0x0: WFE mode has not been entered 0x1: WFE mode has been entered	R	0
8	STANDBYWFI	This bit gives software the visibility to track whether WFI mode have been entered. 0x0: WFI mode has not been entered 0x1: WFI mode has been entered	R	0
7:0	RESERVED	Reserved	R	0

**Table 4-47. Register Call Summary for Register WKG\_CONTROL\_1**

Cortex-A9 MPU Subsystem Integration

- [Reset Distribution: \[0\]](#)

Cortex-A9 MPU Subsystem Functional Description

- [Power States: \[1\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[2\]](#)

**Table 4-48. WKG\_ENB\_A\_1**

<b>Address Offset</b>	0x0000 0410																																
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .																<b>Instance</b>																CORTEXA9_WUGEN
<b>Description</b>	This register enables the interrupts (for CPU1) from MA_IRQ_0 to MA_IRQ_31 write 0x0: disable interrupt write 0x1: enable interrupt																																
<b>Type</b>	RW																																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	WKG_ENB_FOR_INTR31	WKG_ENB_FOR_INTR30	WKG_ENB_FOR_INTR29	WKG_ENB_FOR_INTR28	WKG_ENB_FOR_INTR27	WKG_ENB_FOR_INTR26	WKG_ENB_FOR_INTR25	WKG_ENB_FOR_INTR24	WKG_ENB_FOR_INTR23	WKG_ENB_FOR_INTR22	WKG_ENB_FOR_INTR21	WKG_ENB_FOR_INTR20	WKG_ENB_FOR_INTR19	WKG_ENB_FOR_INTR18	WKG_ENB_FOR_INTR17	WKG_ENB_FOR_INTR16	WKG_ENB_FOR_INTR15	WKG_ENB_FOR_INTR14	WKG_ENB_FOR_INTR13	WKG_ENB_FOR_INTR12	WKG_ENB_FOR_INTR11	WKG_ENB_FOR_INTR10	WKG_ENB_FOR_INTR9	RESERVED	WKG_ENB_FOR_INTR7	RESERVED	RESERVED	WKG_ENB_FOR_INTR4	RESERVED	WKG_ENB_FOR_INTR2	WKG_ENB_FOR_INTR1	WKG_ENB_FOR_INTR0	

Bits	Field Name	Description	Type	Reset
31	WKG_ENB_FOR_INTR31		RW	1
30	WKG_ENB_FOR_INTR30		RW	1
29	WKG_ENB_FOR_INTR29		RW	1
28	WKG_ENB_FOR_INTR28		RW	1
27	WKG_ENB_FOR_INTR27		RW	1
26	WKG_ENB_FOR_INTR26		RW	1
25	WKG_ENB_FOR_INTR25		RW	1
24	WKG_ENB_FOR_INTR24		RW	1
23	WKG_ENB_FOR_INTR23		RW	1
22	WKG_ENB_FOR_INTR22		RW	1
21	WKG_ENB_FOR_INTR21		RW	1
20	WKG_ENB_FOR_INTR20		RW	1
19	WKG_ENB_FOR_INTR19		RW	1
18	WKG_ENB_FOR_INTR18		RW	1
17	WKG_ENB_FOR_INTR17		RW	1
16	WKG_ENB_FOR_INTR16		RW	1
15	WKG_ENB_FOR_INTR15		RW	1
14	WKG_ENB_FOR_INTR14		RW	1
13	WKG_ENB_FOR_INTR13		RW	1
12	WKG_ENB_FOR_INTR12		RW	1
11	WKG_ENB_FOR_INTR11		RW	1
10	WKG_ENB_FOR_INTR10		RW	1
9	WKG_ENB_FOR_INTR9		RW	1
8	RESERVED		R	0 <sup>(1)</sup>
7	WKG_ENB_FOR_INTR7		RW	1
6	RESERVED		R	1
5	RESERVED		R	1
4	WKG_ENB_FOR_INTR4		RW	1
3	RESERVED		R	1
2	WKG_ENB_FOR_INTR2		RW	1
1	WKG_ENB_FOR_INTR1		RW	1
0	WKG_ENB_FOR_INTR0		RW	1

<sup>(1)</sup> The reset value is 0x0 by safety reasons.

**Table 4-49. Register Call Summary for Register WKG\_ENB\_A\_1**

Cortex-A9 MPU Subsystem Functional Description

- [Wake-up Generator \(WUGEN\): \[0\]](#)

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[1\]](#)

Table 4-50. WKG\_ENB\_B\_1

<b>Address Offset</b>	0x0000 0414		
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .	<b>Instance</b>	CORTEXA9_WUGEN
<b>Description</b>	This register enables the interrupts (for CPU1) from MA_IRQ_32 to MA_IRQ_63 write 0x0: disable interrupt write 0x1: enable interrupt		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED	WKG_ENB_FOR_INTR62	WKG_ENB_FOR_INTR61	RESERVED	WKG_ENB_FOR_INTR59	WKG_ENB_FOR_INTR58	WKG_ENB_FOR_INTR57	WKG_ENB_FOR_INTR56	RESERVED	RESERVED	WKG_ENB_FOR_INTR53	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	WKG_ENB_FOR_INTR48	WKG_ENB_FOR_INTR47	WKG_ENB_FOR_INTR46	WKG_ENB_FOR_INTR45	WKG_ENB_FOR_INTR44	WKG_ENB_FOR_INTR43	WKG_ENB_FOR_INTR42	WKG_ENB_FOR_INTR41	WKG_ENB_FOR_INTR40	WKG_ENB_FOR_INTR39	WKG_ENB_FOR_INTR38	WKG_ENB_FOR_INTR37	WKG_ENB_FOR_INTR36	RESERVED	WKG_ENB_FOR_INTR34	WKG_ENB_FOR_INTR33	WKG_ENB_FOR_INTR32

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	1
30	WKG_ENB_FOR_INTR62		RW	1
29	WKG_ENB_FOR_INTR61		RW	1
28	RESERVED		R	1
27	WKG_ENB_FOR_INTR59		RW	1
26	WKG_ENB_FOR_INTR58		RW	1
25	WKG_ENB_FOR_INTR57		RW	1
24	WKG_ENB_FOR_INTR56		RW	1
23	RESERVED		R	1
22	RESERVED		R	1
21	WKG_ENB_FOR_INTR53		RW	1
20	RESERVED		R	1
19	RESERVED		R	1
18	RESERVED		R	1
17	RESERVED		R	1
16	WKG_ENB_FOR_INTR48		RW	1
15	WKG_ENB_FOR_INTR47		RW	1
14	WKG_ENB_FOR_INTR46		RW	1
13	WKG_ENB_FOR_INTR45		RW	1
12	WKG_ENB_FOR_INTR44		RW	1
11	WKG_ENB_FOR_INTR43		RW	1
10	WKG_ENB_FOR_INTR42		RW	1
9	WKG_ENB_FOR_INTR41		RW	1
8	WKG_ENB_FOR_INTR40		RW	1
7	WKG_ENB_FOR_INTR39		RW	1
6	WKG_ENB_FOR_INTR38		RW	1
5	WKG_ENB_FOR_INTR37		RW	1
4	WKG_ENB_FOR_INTR36		RW	1
3	RESERVED		R	1
2	WKG_ENB_FOR_INTR34		RW	1
1	WKG_ENB_FOR_INTR33		RW	1



Bits	Field Name	Description	Type	Reset
0	WKG_ENB_FOR_INTR32		RW	1

**Table 4-51. Register Call Summary for Register WKG\_ENB\_B\_1**

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[0\]](#)

**Table 4-52. WKG\_ENB\_C\_1**

<b>Address Offset</b>	0x0000 0418	<b>Instance</b>	CORTEXA9_WUGEN
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .		
<b>Description</b>	This register enables the interrupts (for CPU1) from MA_IRQ_64 to MA_IRQ_95 write 0x0: disable interrupt write 0x1: enable interrupt		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKG_ENB_FOR_INTR94	WKG_ENB_FOR_INTR93	WKG_ENB_FOR_INTR92	WKG_ENB_FOR_INTR91	WKG_ENB_FOR_INTR90	WKG_ENB_FOR_INTR89	WKG_ENB_FOR_INTR88	WKG_ENB_FOR_INTR87	WKG_ENB_FOR_INTR86	RESERVED	WKG_ENB_FOR_INTR84	WKG_ENB_FOR_INTR83	RESERVED	RESERVED	WKG_ENB_FOR_INTR80	RESERVED	WKG_ENB_FOR_INTR78	WKG_ENB_FOR_INTR77	WKG_ENB_FOR_INTR76	WKG_ENB_FOR_INTR75	WKG_ENB_FOR_INTR74	WKG_ENB_FOR_INTR73	WKG_ENB_FOR_INTR72	WKG_ENB_FOR_INTR71	WKG_ENB_FOR_INTR70	WKG_ENB_FOR_INTR69	WKG_ENB_FOR_INTR68	WKG_ENB_FOR_INTR67	WKG_ENB_FOR_INTR66	WKG_ENB_FOR_INTR65	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	1
30	WKG_ENB_FOR_INTR94		RW	1
29	WKG_ENB_FOR_INTR93		RW	1
28	WKG_ENB_FOR_INTR92		RW	1
27	WKG_ENB_FOR_INTR91		RW	1
26	WKG_ENB_FOR_INTR90		RW	1
25	WKG_ENB_FOR_INTR89		RW	1
24	WKG_ENB_FOR_INTR88		RW	1
23	WKG_ENB_FOR_INTR87		RW	1
22	WKG_ENB_FOR_INTR86		RW	1
21	RESERVED		R	1
20	WKG_ENB_FOR_INTR84		RW	1
19	WKG_ENB_FOR_INTR83		RW	1
18	RESERVED		R	1
17	RESERVED		R	1
16	WKG_ENB_FOR_INTR80		RW	1
15	RESERVED		R	1
14	WKG_ENB_FOR_INTR78		RW	1
13	WKG_ENB_FOR_INTR77		RW	1
12	WKG_ENB_FOR_INTR76		RW	1
11	WKG_ENB_FOR_INTR75		RW	1
10	WKG_ENB_FOR_INTR74		RW	1
9	WKG_ENB_FOR_INTR73		RW	1
8	WKG_ENB_FOR_INTR72		RW	1

Bits	Field Name	Description	Type	Reset
7	WKG_ENB_FOR_INTR71		RW	1
6	WKG_ENB_FOR_INTR70		RW	1
5	WKG_ENB_FOR_INTR69		RW	1
4	WKG_ENB_FOR_INTR68		RW	1
3	WKG_ENB_FOR_INTR67		RW	1
2	WKG_ENB_FOR_INTR66		RW	1
1	WKG_ENB_FOR_INTR65		RW	1
0	RESERVED		R	1

**Table 4-53. Register Call Summary for Register WKG\_ENB\_C\_1**

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[0\]](#)

**Table 4-54. WKG\_ENB\_D\_1**

<b>Address Offset</b>	0x0000 041C	<b>Instance</b>	CORTEXA9_WUGEN
<b>Physical Address</b>	See <a href="#">Table 4-35</a> .		
<b>Description</b>	This register enables the interrupts (for CPU1) from MA_IRQ_96 to MA_IRQ_127 write 0x0: disable interrupt write 0x1: enable interrupt		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	WKG_ENB_FOR_INTR120	WKG_ENB_FOR_INTR119	RESERVED	RESERVED	RESERVED	RESERVED	WKG_ENB_FOR_INTR114	WKG_ENB_FOR_INTR113	WKG_ENB_FOR_INTR112	WKG_ENB_FOR_INTR111	WKG_ENB_FOR_INTR110	WKG_ENB_FOR_INTR109	RESERVED	WKG_ENB_FOR_INTR107	RESERVED	RESERVED	WKG_ENB_FOR_INTR104	WKG_ENB_FOR_INTR103	WKG_ENB_FOR_INTR102	WKG_ENB_FOR_INTR101	WKG_ENB_FOR_INTR100	WKG_ENB_FOR_INTR99	WKG_ENB_FOR_INTR98	WKG_ENB_FOR_INTR97	WKG_ENB_FOR_INTR96

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	1
30	RESERVED		R	1
29	RESERVED		R	1
28	RESERVED		R	1
27	RESERVED		R	1
26	RESERVED		R	1
25	RESERVED		R	1
24	WKG_ENB_FOR_INTR120		RW	1
23	WKG_ENB_FOR_INTR119		RW	1
22	RESERVED		R	1
21	RESERVED		R	1
20	RESERVED		R	1
19	RESERVED		R	1
18	WKG_ENB_FOR_INTR114		RW	1
17	WKG_ENB_FOR_INTR113		RW	1
16	WKG_ENB_FOR_INTR112		RW	1
15	WKG_ENB_FOR_INTR111		RW	1

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
14	WKG_ENB_FOR_INTR110		RW	1
13	WKG_ENB_FOR_INTR109		RW	1
12	RESERVED		R	1
11	WKG_ENB_FOR_INTR107		RW	1
10	RESERVED		R	1
9	RESERVED		R	1
8	WKG_ENB_FOR_INTR104		RW	1
7	WKG_ENB_FOR_INTR103		RW	1
6	WKG_ENB_FOR_INTR102		RW	1
5	WKG_ENB_FOR_INTR101		RW	1
4	WKG_ENB_FOR_INTR100		RW	1
3	WKG_ENB_FOR_INTR99		RW	1
2	WKG_ENB_FOR_INTR98		RW	1
1	WKG_ENB_FOR_INTR97		RW	1
0	WKG_ENB_FOR_INTR96		RW	1

**Table 4-55. Register Call Summary for Register WKG\_ENB\_D\_1**

Cortex-A9 MPU Subsystem Register Manual

- [CORTEXA9\\_WUGEN Register Summary: \[0\]](#)

PRELIMINARY

## **DSP Subsystem**

---

---

---

This information is not available in the public domain.

**Topic**

**Page**

--

PRELIMINARY

## **IVA-HD Subsystem**

---

---

---

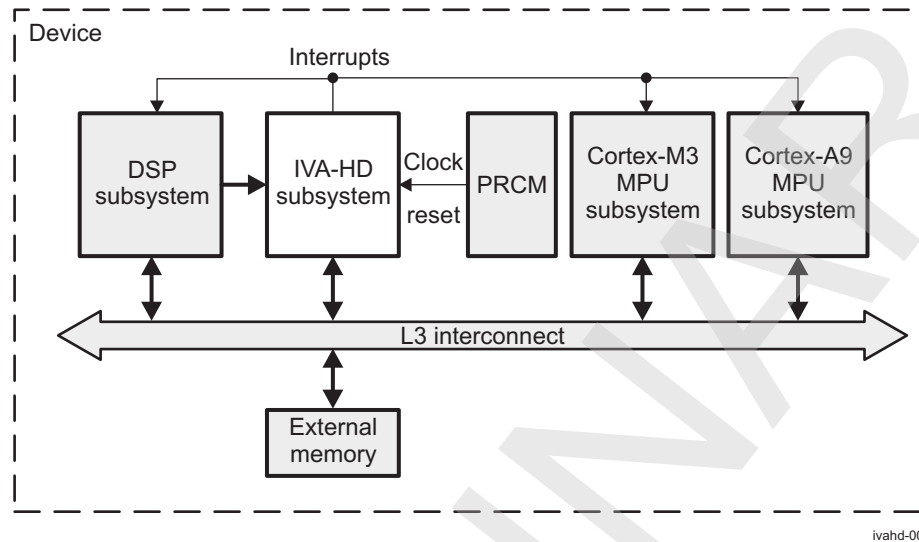
This chapter describes the IVA-HD subsystem in the multimedia device.

Topic	Page
<b>6.1 IVA-HD Overview .....</b>	<b>1072</b>

## 6.1 IVA-HD Overview

The IVA-HD is the image and video imaging hardware accelerator subsystem. Figure 6-1 shows the IVA-HD subsystem in the device.

**Figure 6-1. IVA-HD in the Device**



The IVA-HD supports resolutions up to 1080 p/i with full performance of 30 fps (or 60 fields). The IVA-HD subsystem supports the following codec standards natively; that is, all functions of standards are accelerated (without any intervention of the digital signal processor [DSP]):

- H.264: BP/MP/HP Encode and Decode
- H.264: Fast Profile/RCDO Encode and Decode
- MPEG-4: SP/ASP Encode/Decode (No support for GMC)
- DivX 5.x & higher Encode/Decode (No lower version; for example, 3.11 and 4.x)
- H.263: Profile 0 and 3 for Decode, Profile 0 for Encode
- Sorenson Spark: V0 and V1 Decode (No encode support)
- MPEG-2 SP/MP Encode/Decode
- MPEG-1 Encode/Decode
- VC-1/WMV9/RTV : SP/MP/AP Encode and Decode
- On2® VP6/VP7 Decode
- RealVideo® 8/9/10 Decode
- AVS 1.0 Encode and Decode
- JPEG (also MJPEG) Baseline Encode/Decode
- H.264 Annex G (SVC) Scalable Baseline Profile to 480p-720p30
- H.264 Annex H (MVC) up to 720p30

The IVA-HD subsystem is composed of:

- A primary sequencer, including its memories and an interrupt controller: ICONT1
- A video direct memory access (DMA) processor, which can be used as a secondary sequencer: ICONT2
- A video DMA engine: vDMA
- An entropy coder/decoder: ECD3
- A motion compensation engine: MC3
- A transform and quantization calculation engine: CALC3
- A loop filter acceleration engine: iLF3
- A motion estimation acceleration engine: iME3
- An intraprediction estimation engine: iPE3



- Shared level 2 (L2) interface and memory
- Local interconnect
- A message interface for communication between SyncBoxes
- Mailbox
- A debug module for trace event and software instrumentation: SMSET

IVA-HD 1.0 will use eXpressDSP Digital Media (xDM) standard as the principle software interface. The xDM standard defines application programming interfaces (APIs) through which an application invokes a particular class of codec, such as video decode or audio encode.

xDM developers kit, technical documentation and full compliant codecs can be downloaded from <http://focus.ti.com/docs/toolsw/folders/print/tmdxdaisxdm.html>.

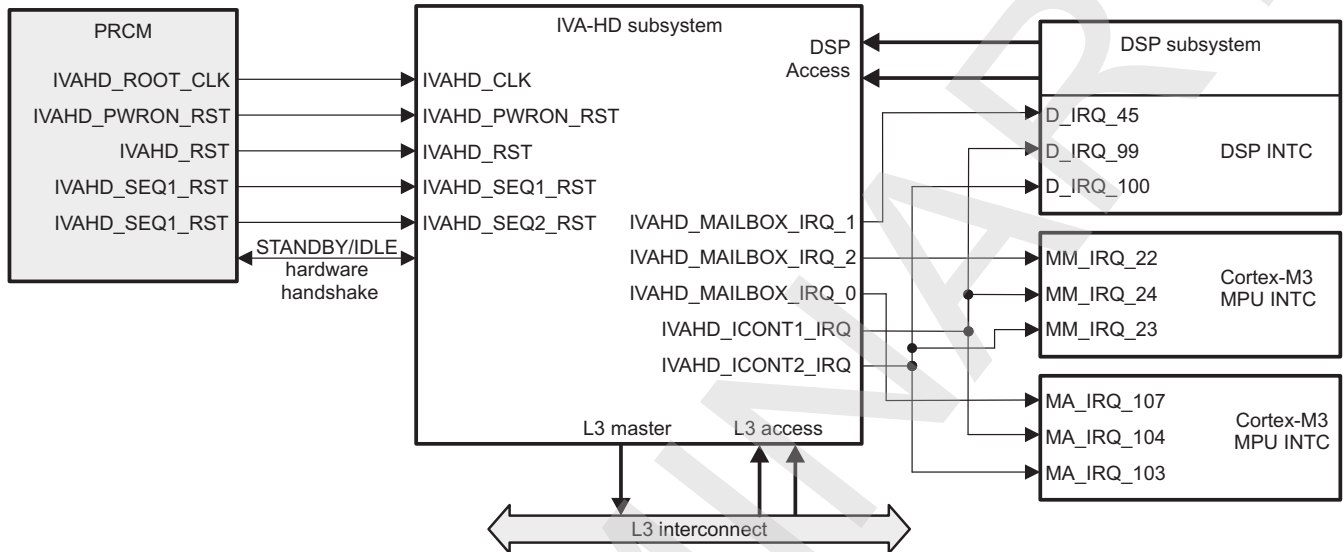
Software released on IVA-HD 1.0 will be xDM-compliant and will be available during 2010.

### 6.1.1 IVA-HD Integration

This section describes IVA-HD in the device, including information about clocks, resets, and hardware requests.

Figure 6-2 shows IVA-HD integration.

**Figure 6-2. IVA-HD Integration**



iva-hd-002

**NOTE:** For more information about the STANDBY/IDLE hardware handshake and the wake-up request, see [Section 3.1.1.1.2, Module-Level Clock Management](#), in [Chapter 3 Power, Reset and Clock Management](#).

Table 6-1 through Table 6-3 summarize the integration of the module in the device.

**Table 6-1. IVA-HD Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
IVA-HD	PD_IVAHD	L3

**Table 6-2. IVA-HD Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
IVA-HD	IVAHD_CLK	IVAHD_ROOT_CLK	CM	IVA-HD clock (functional and interface)
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
IVA-HD	IVAHD_PWRON_RST	IVAHD_PWRON_RST	PRM	IVA-HD power-on reset
IVA-HD	IVAHD_RST	IVAHD_RST	PRM	IVA-HD global reset (all logic is reset)
IVA-HD	IVAHD_SEQ1_RST	IVAHD_SEQ1_RST	PRM	ICONT1 reset
IVA-HD	IVAHD_SEQ2_RST	IVAHD_SEQ2_RST	PRM	ICONT2 reset

**Table 6-3. Hardware Requests**

Module Instance	Source Signal Name	Destination Signal Name	Interrupt Requests	
			Destination	Description
IVA-HD	IVAHD_MAILBOX_IRQ_0	MA_IRQ_107	Cortex-A9 MPU INTC	IVA-HD mailbox user 0 interrupt
IVA-HD	IVAHD_ICONT1_IRQ	MA_IRQ_104	Cortex-A9 MPU INTC	Sync interrupt from ICONT1
IVA-HD	IVAHD_ICONT2_IRQ	MA_IRQ_103	Cortex-A9 MPU INTC	Sync interrupt from ICONT2
IVA-HD	IVAHD_MAILBOX_IRQ_1	D_IRQ_45	DSP INTC	IVA-HD mailbox user 1 interrupt
IVA-HD	IVAHD_ICONT1_IRQ	D_IRQ_99	DSP INTC	Sync interrupt from ICONT1
IVA-HD	IVAHD_ICONT2_IRQ	D_IRQ_100	DSP INTC	Sync interrupt from ICONT2
IVA-HD	IVAHD_MAILBOX_IRQ_2	MM_IRQ_22	Cortex-M3 MPU INTC	IVA-HD mailbox user 2 interrupt
IVA-HD	IVAHD_ICONT1_IRQ	MM_IRQ_24	Cortex-M3 MPU INTC	Sync interrupt from ICONT1
IVA-HD	IVAHD_ICONT2_IRQ	MM_IRQ_23	Cortex-M3 MPU INTC	Sync interrupt from ICONT2

**NOTE:**

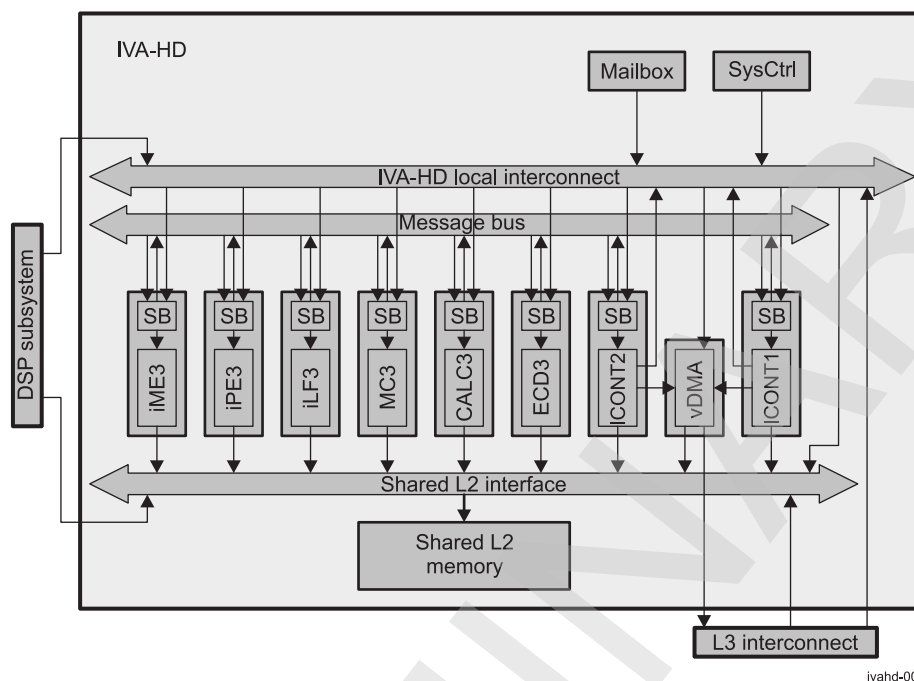
- For more information about the mailbox interrupt, see [Section 19.3.4](#).

## 6.1.2 IVA-HD Functional Description

### 6.1.2.1 IVA-HD Block Diagram

[Figure 6-3](#) shows a block diagram of the IVA-HD subsystem.

Figure 6-3. IVA-HD Block Diagram



NOTE: SB stands for SyncBox.

NOTE: iPE3, MC3, CALC3, and ECD3 include a load and store engine (LSE), which is the same in each of these modules.

#### 6.1.2.1.1 SyncBox

The SyncBox is a configurable module that is in charge of scheduling all embedded hardware module within IVA-HD subsystem. It handles all aspects of synchronization, data sharing, and parameters passing between accelerators. It also offers the possibilities to use asynchronous messages.

#### 6.1.2.1.2 ICONTs

The ICONT module is an ARM968E-S™ based microcontroller with 32KB of instruction tightly coupled memory and 16KB of data tightly coupled memory. It includes an interrupt controller (INTC), a local data mover, its own SyncBox module for synchronizing tasks with other modules and its associated SyncBox handler.

Two identical instances of ICONT, ICONT1 and ICONT2, are present in the IVA-HD subsystem. They can typically be used to perform high-level processing (at frame and slice level), control at macroblock-level bounding box computation and other vDMA processing tasks.

Software can map process equally on either ICONT.

#### 6.1.2.1.3 vDMA

The vDMA is a DMA engine to perform data transfer between external memories and shared L2 memory and also memory copies inside SL2 and inside external memory.

#### 6.1.2.1.4 iME3

The iME3 accelerator is used to perform motion estimation in encode processing. iME3 embeds its own SyncBox module for synchronizing tasks with other modules.

It compares a current macroblock to reference area and provides area in reference region (in terms of offsets) which is least different to the current macroblock. It can also interpolate a block with half or quarter pixel precision, thus producing 4 (half pixel) or 16 (quarter pixel) interpolated blocks from the original block. Additionally, it supports searching of the best matching block within the interpolated planes.

#### **6.1.2.1.5 iPE3**

The iPE3 accelerator is used to perform the intraprediction estimation in encode. The iPE3 embeds its own SyncBox module for synchronizing tasks with other modules and an LSE to transfer data from internal memories to shared L2 memory.

It supports two modes, depending on video standard:

- Spatial intraprediction estimation for H.264 and AVS. It creates intraprediction macroblocks with given intramodes from original macroblock and provides cost estimation between the original macroblock and each pseudo intraprediction macroblock, and then chooses a mode with the smallest cost to recommend it as an optimal intraprediction mode.
- Spatial activity for MPEG-1/2/4 and VC-1. It calculates the spatial activity of the original luminance samples with the specified block size. This mode intends to provide information of the original Luma pixels. The values can be used to decide the coding parameters of the macroblock such as the coding mode and the quantization parameter.

#### **6.1.2.1.6 MC3**

The MC3 accelerator is used to perform the motion compensation. MC3 embeds its own SyncBox module for synchronizing tasks with other HWAs and an LSE to transfer data from internal memories to shared L2 memory.

It creates an interprediction macroblock with given motion vectors and modes from the reference data.

#### **6.1.2.1.7 CALC3**

The CALC3 accelerator is used to perform forward and inverse transform and quantization calculation. CALC3 embeds its own SyncBox module for synchronizing tasks with other HWAs and an LSE to transfer data from internal memories to shared L2 memory.

It can perform transform/inverse transform, Q/iQ and DC/AC prediction.

#### **6.1.2.1.8 iLF3**

The iLF3 accelerator is used to perform deblocking filtering and boundary strength computation. The iLF3 embeds its own SyncBox module for synchronizing tasks with other HWAs.

#### **6.1.2.1.9 ECD3**

The ECD3 accelerator is designed to encode and decode the stream. ECD3 embeds its own SyncBox module for synchronizing tasks with other HWAs and an LSE to transfer data from internal memories to shared L2 memory.

It supports Huffman codes and arithmetic codes.

For encode, the ECD3 encodes the macroblock information and residual data into bitstream. For decode, the ECD3 decodes the bitstream and recovers the macroblock information and residual data.

#### **6.1.2.1.10 SL2 Interface**

The shared L2 interface, SL2IF, is an arbitrator that allows 18 initiators to access to an interleaved set of eight memory banks.

The SL2IF has two sets of interfaces:

- Eighteen 128-bit interfaces for accesses from module's to shared L2 memories.
- Eight 128-bit memory interfaces for direct accesses to memory bank

### 6.1.2.1.11 Message Bus

The message bus, is an arbitrator that allows eight initiators to access eight targets. It is used to dispatch messages generated by the SyncBox of the different IPs.

### 6.1.2.1.12 IVA-HD Local Interconnect

The IVA-HD local interconnect provides connectivity between two external host interconnects (DSP and L3), two local sequencers (ICONT1 and ICONT2) and video hardware accelerators (iME3, iLF3, ECD3, CALC3, MC3, iPE3), video DMA engine (vDMA), and local modules (mailbox and sysctrl).

### 6.1.2.1.13 MailBox

The function of the mailbox is to support a 2-way communication between two hosts through interrupt. It allows the software to establish a communication channel between processors through a set of registers and associated interrupt signals by sending and receiving messages.

The mailbox embedded inside the IVA-HD subsystem implements a 2-way communication between three external users and one internal user. This communication is ensured through three pairs of mailboxes and four-message FIFO depth for each message queue.

---

**NOTE:** The internal user is one of the two ICONTs. ICONT1 and ICONT2 are connected on a shared interrupt line. The choice between ICONT1 and ICONT2 is done by masking mailbox interrupt on ICONT1 or ICONT2. ICONT1 and ICONT2 can access the mailbox through the IVA-HD local interconnect.

---

For a detailed description of the mailbox, including IVA-HD mailbox interrupt mapping, see , *IVA-HD Imaging Controller*.

### 6.1.2.1.14 IVA-HD System Control

The SYSCTRL module of IVA-HD is in charge of:

- Controlling clocks to module's, upon software control and power handshaking state
- Controlling power, reset, and clock management (PRCM) module power handshaking
- Providing status of the above operation
- Supporting synchronization through external event

## 6.1.2.2 IVA-HD Power Management

Table 6-4 describes power-management features available for the IVA-HD module.

---

**NOTE:** For descriptions of IdleMode and StandbyMode features, see [Section 3.1.1.1.2, Module-Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

---

**Table 6-4. Local Power-Management Features**

Feature	Registers	Description
Slave idle modes	IVAHD_SYSCONFIG[3:2] IDLEMODE	No-idle and smart-idle modes are available.
Master standby modes	IVAHD_SYSCONFIG[5:4] STANDBYMODE	No-standby and smart-standby modes are available.

### 6.1.2.3 IVA-HD Memory Mapping

ARM968E-S (of each ICONT) can address a 4-GB memory space. The lower address range (8MB) is allocated to its level 1 (L1) memories (ITCM and DTCM). The higher address range is allocated to L2 memories. Access to L2 memories can be buffered or nonbuffered, depending on the selected address range. For more information, see the ARM968E-STM Technical Reference Manual.

In [Table 6-5](#), the address spaces of buffered regions are shown as EXT\_BUF, and the address spaces of nonbuffered regions are shown as EXT\_NBUF.

**Table 6-5. ICONT Memory Mapping Regions**

Region Name	Address		Size (KB)	Description
	Start	Stop		
ITCM	0x0000 0000	0x0000 7FFF	32	ITCM memory aliased 128 times
	...	...	...	
ITCM (alias 127)	0x003F 8000	0x003F FFFF	32	
DTCM	0x0040 0000	0x0040 3FFF	16	DTCM memory aliased 256 times
	...	...	...	
DTCM (alias 255)	0x007F C000	0x007F FFFF	16	
EXT_BUF	0x0080 0000	0x008F FFFF	1024	External buffered region
	...	...	...	
EXT_BUF (alias 247)	0x0FF0 0000	0x0FFF FFFF	1024	External buffered region
EXT_NBUF (alias)	0x1000 0000	0x100F 0000	1024	External nonbuffered region
	...	...	...	
EXT_NBUF (alias)	0x1FF0 00000	0x1FFF FFFF	1024	External nonbuffered region
	...	...	...	
EXT_BUF (alias)	0xM000 0000	0xM0FF FFFF	1024	External buffered region
	...	...	...	
EXT_NBUF (alias)	0xFFFF 0000	0xFFFF FFFF	1024	External nonbuffered region

Each of these regions (EXT\_BUF and EXT\_NBUF) has the same memory map, aliased throughout the address range of the region. [Table 6-6](#) lists ICONT1 memory mapping, and [Table 6-7](#) lists ICONT2 memory mapping. Offsets shown are from the start of the region.

**Table 6-6. ICONT1 Memory Mapping**

Module	Offset in Region		Size (KB)
	Begin	End	
SL2 - Mem	0x0000 0000	0x0003 FFFF	256
Reserved	0x0004 0000	0x0007 FFFF	256
ICONT1 - DMem	0x0008 0000	0x0008 3FFF	16
Reserved	0x0008 4000	0x0008 7FFF	16
ICONT1 - IMem	0x0008 8000	0x0008 FFFF	32
ICONT2 - DMem	0x0009 0000	0x0009 3FFF	16
Reserved	0x0009 4000	0x0009 7FFF	16
ICONT2 - IMem	0x0009 8000	0x0009 FFFF	32
ECD3 - Mem	0x000A 0000	0x000A 7FFF	32
CALC3 - Mem	0x000A 8000	0x000A BFFF	16
Reserved	0x000A C000	0x000A FFFF	16
MC3 - Mem	0x000B 0000	0x000B 7FFF	32
iPE3 - Mem	0x000B 8000	0x000B 8FFF	4
Reserved	0x000B 9000	0x000C 0FFF	32
SMSET - Reg	0x000C 1000	0x000C 1FFF	4
SMSET - SWI	0x000C 2000	0x000C 27FF	2
Reserved	0x000C 2800	0x000C FFFF	54
vDMA - Reg	0x000D 0000	0x000D 1FFF	8
iLF3 - Reg	0x000D 2000	0x000D 2FFF	4
Reserved	0x000D 3000	0x000D 3FFF	4
iME3 - Reg	0x000D 4000	0x000D 7FFF	16

**Table 6-6. ICONT1 Memory Mapping (continued)**

Module	Offset in Region		Size
	Begin	End	KB
CALC3 - Reg	0x000D 8000	0x000D 87FF	2
iPE3 - Reg	0x000D 8800	0x000D 8FFF	2
MC3 - Reg	0x000D 9000	0x000D 97FF	2
ECD3 - Reg	0x000D 9800	0x000D 9FFF	2
Reserved	0x000D A000	0x000D A3FF	1
SysControl	0x000D A400	0x000D A7FF	1
Mailbox	0x000D A800	0x000D ABFF	1
Reserved	0x000D AC00	0x000D FFFF	21
ICONT1 - SyncBox	0x000E 0000	0x000E 07FF	2
ICONT2 - SyncBox	0x000E 0800	0x000E 0FFF	2
iLF3 - SyncBox	0x000E 1000	0x000E 17FF	2
iME3 - SyncBox	0x000E 1800	0x000E 1FFF	2
CALC3 - SyncBox	0x000E 2000	0x000E 27FF	2
iPE3 - SyncBox	0x000E 2800	0x000E 2FFF	2
MC3 - SyncBox	0x000E 3000	0x000E 37FF	2
ECD3 - SyncBox	0x000E 3800	0x000E 3FFF	2
Reserved	0x000E 4000	0x000E FFFF	48
ICONT1 - Reg <sup>(1)</sup>	0x000F 0000	0x000F 0FFF	4
ICONT2 - Reg	0x000F 1000	0x000F 1FFF	4
ICONT1 - Reg <sup>(2)</sup>	0x000F 2000	0x000F 2FFF	4
Reserved	0x000F 3000	0x000F FFFF	52

(1) ICONT1 private access to its own registers

(2) ICONT1 self-access to its own registers

**Table 6-7. ICONT2 Memory Mapping**

Module	Offset in Region		Size
	Begin	End	KB
SL2 - Mem	0x0000 0000	0x0003 FFFF	256
Reserved	0x0004 0000	0x0007 FFFF	256
ICONT1 - DMem	0x0008 0000	0x0008 3FFF	16
Reserved	0x0008 4000	0x0008 7FFF	16
ICONT1 - IMem	0x0008 8000	0x0008 FFFF	32
ICONT2 - DMem	0x0009 0000	0x0009 3FFF	16
Reserved	0x0009 4000	0x0009 7FFF	16
ICONT2 - IMem	0x0009 8000	0x0009 FFFF	32
ECD3 - Mem	0x000A 0000	0x000A 7FFF	32
CALC3 - Mem	0x000A 8000	0x000A BFFF	16
Reserved	0x000A C000	0x000A FFFF	16
MC3 - Mem	0x000B 0000	0x000B 7FFF	32
iPE3 - Mem	0x000B 8000	0x000B 8FFF	4
Reserved	0x000B 9000	0x000C 0FFF	32
SMSET - Reg	0x000C 1000	0x000C 1FFF	4
SMSET - SWI	0x000C 2000	0x000C 27FF	2
Reserved	0x000C 2800	0x000C FFFF	54
vDMA - Reg	0x000D 0000	0x000D 1FFF	8
iLF3 - Reg	0x000D 2000	0x000D 2FFF	4



**Table 6-7. ICONT2 Memory Mapping (continued)**

Module	Offset in Region		Size
	Start Address	End Address	
Reserved	0x000D 3000	0x000D 3FFF	4
iME3 - Reg	0x000D 4000	0x000D 7FFF	16
CALC3 - Reg	0x000D 8000	0x000D 87FF	2
iPE3 - Reg	0x000D 8800	0x000D 8FFF	2
MC3 - Reg	0x000D 9000	0x000D 97FF	2
ECD3 - Reg	0x000D 9800	0x000D 9FFF	2
Reserved	0x000D A000	0x000D A3FF	1
SysControl	0x000D A400	0x000D A7FF	1
Mailbox	0x000D A800	0x000D ABFF	1
Reserved	0x000D AC00	0x000D FFFF	21
ICONT1 - SyncBox	0x000E 0000	0x000E 07FF	2
ICONT2 - SyncBox	0x000E 0800	0x000E 0FFF	2
iLF3 - SyncBox	0x000E 1000	0x000E 17FF	2
iME3 - SyncBox	0x000E 1800	0x000E 1FFF	2
CALC3 - SyncBox	0x000E 2000	0x000E 27FF	2
iPE3 - SyncBox	0x000E 2800	0x000E 2FFF	2
MC3 - SyncBox	0x000E 3000	0x000E 37FF	2
ECD3 - SyncBox	0x000E 3800	0x000E 3FFF	2
Reserved	0x000E 4000	0x000E FFFF	48
ICONT1 - Reg	0x000F 0000	0x000F 0FFF	4
ICONT2 - Reg	0x000F 1000	0x000F 1FFF	4
Reserved	0x000F 2000	0x000F 2FFF	4
ICONT2 - Reg	0x000F 3000	0x000F 3FFF	4
Reserved	0x000F 4000	0x000F FFFF	48

1. ICONT2 private access to its own registers
2. ICONT2 self-access to its own registers

---

**NOTE:** Definition of ICONT self-access and private access:

- Self-access: Direct access of ICONT to its own memory-mapped registers (MMRs) (reduced latency); that is, without using the IVA-HD local interconnect. Internal ICONT configuration (CFG) registers are not visible. Only IRQ, DM, and SBH registers can be accessed.
- Private access: Access through IVA-HD local interconnect. By default, all ICONT accesses described in [Table 6-6](#) and [Table 6-7](#) are private, except for corresponding self-access. In case of ICONT private access to its own registers, internal ICONT CFG registers are visible (on top of IRQ, DM, and SBH).

---

**NOTE:** Examples of start address:

- Buffered region: 0x0080 0000, 0x0090 0000, 0x00A0 0000, 0x0FF0 0000, 0x2000 0000, etc.
- Nonbuffered region: 0x1000 0000, 0x1010 0000, 0x1020 0000, 0x1FF0 0000, 0x3000 0000, etc.

---

[Table 6-8](#) shows IVA-HD modules and memory addresses for the DSP processor and L3 access.

**Table 6-8. DSP and L3 Interconnect Memory Mapping**

Module	Begin Offset	DSP Private Address		L3 Interconnect Address		Size (KB)
		Begin	End	Begin	End	
ICONT1 - DMem	0x0000 0000	0x01E0 0000	0x01E0 3FFF	0x5A00 0000	0x5A00 3FFF	16
Reserved	0x0000 4000	0x01E0 4000	0x01E0 7FFF	0x5A00 4000	0x5A00 7FFF	16
ICONT1 - IMem	0x0000 8000	0x01E0 8000	0x01E0 FFFF	0x5A00 8000	0x5A00 FFFF	32
ICONT2 - DMem	0x0001 0000	0x01E1 0000	0x01E1 3FFF	0x5A01 0000	0x5A01 3FFF	16
Reserved	0x0001 4000	0x01E1 4000	0x01E1 7FFF	0x5A01 4000	0x5A01 7FFF	16
ICONT2 - IMem	0x0001 8000	0x01E1 8000	0x01E1 FFFF	0x5A01 8000	0x5A01 FFFF	32
ECD3 - Mem	0x0002 0000	0x01E2 0000	0x01E2 7FFF	0x5A02 0000	0x5A02 7FFF	32
CALC3 - Mem	0x0002 8000	0x01E2 8000	0x01E2 BFFF	0x5A02 8000	0x5A02 BFFF	16
Reserved	0x0002 C000	0x01E2 C000	0x01E2 FFFF	0x5A02 C000	0x5A02 FFFF	16
MC3 - Mem	0x0003 0000	0x01E3 0000	0x01E3 7FFF	0x5A03 0000	0x5A03 7FFF	32
iPE3 - Mem	0x0003 8000	0x01E3 8000	0x01E3 8FFF	0x5A03 8000	0x5A03 8FFF	4
Reserved	0x0003 9000	0x01E3 9000	0x01E4 FFFF	0x5A03 9000	0x5A04 FFFF	92
vDMA - Reg	0x0005 0000	0x01E5 0000	0x01E5 1FFF	0x5A05 0000	0x5A05 1FFF	8
iLF3 - Reg	0x0005 2000	0x01E5 2000	0x01E5 2FFF	0x5A05 2000	0x5A05 2FFF	4
Reserved	0x0005 3000	0x01E5 3000	0x01E5 3FFF	0x5A05 3000	0x5A05 3FFF	4
iME3 - Reg	0x0005 4000	0x01E5 4000	0x01E5 7FFF	0x5A05 4000	0x5A05 7FFF	16
CALC3 - Reg	0x0005 8000	0x01E5 8000	0x01E5 87FF	0x5A05 8000	0x5A05 87FF	2
iPE3 - Reg	0x0005 8800	0x01E5 8800	0x01E5 8FFF	0x5A05 8800	0x5A05 8FFF	2
MC3 - Reg	0x0005 9000	0x01E5 9000	0x01E5 97FF	0x5A05 9000	0x5A05 97FF	2
ECD3 - Reg	0x0005 9800	0x01E5 9800	0x01E5 9FFF	0x5A05 9800	0x5A05 9FFF	2
Reserved	0x0005 A000	0x01E5 A000	0x01E5 A3FF	0x5A05 A000	0x5A05 A3FF	1
SysControl	0x0005 A400	0x01E5 A400	0x01E5 A7FF	0x5A05 A400	0x5A05 A7FF	1
Mailbox	0x0005 A800	0x01E5 A800	0x01E5 ABFF	0x5A05 A800	0x5A05 ABFF	1
Reserved	0x0005 AC00	0x01E5 AC00	0x01E5 FFFF	0x5A05 AC00	0x5A05 FFFF	21
ICONT1 - SBox	0x0006 0000	0x01E6 0000	0x01E6 07FF	0x5A06 0000	0x5A06 07FF	2
ICONT2 - SBox	0x0006 0800	0x01E6 0800	0x01E6 0FFF	0x5A06 0800	0x5A06 0FFF	2
iLF3 - SyncBox	0x0006 1000	0x01E6 1000	0x01E6 17FF	0x5A06 1000	0x5A06 17FF	2
iME3 - SyncBox	0x0006 1800	0x01E6 1800	0x01E6 1FFF	0x5A06 1800	0x5A06 1FFF	2
CALC3 - SyncBox	0x0006 2000	0x01E6 2000	0x01E6 27FF	0x5A06 2000	0x5A06 27FF	2
iPE3 - SyncBox	0x0006 2800	0x01E6 2800	0x01E6 2FFF	0x5A06 2800	0x5A06 2FFF	2
MC3 - SyncBox	0x0006 3000	0x01E6 3000	0x01E6 37FF	0x5A06 3000	0x5A06 37FF	2
ECD3 - SyncBox	0x0006 3800	0x01E6 3800	0x01E6 3FFF	0x5A06 3800	0x5A06 3FFF	2
Reserved	0x0006 4000	0x01E6 4000	0x01E6 FFFF	0x5A06 4000	0x5A06 FFFF	48
ICONT1 - Reg	0x0007 0000	0x01E7 0000	0x01E7 0FFF	0x5A07 0000	0x5A07 0FFF	4
ICONT2 - Reg	0x0007 1000	0x01E7 1000	0x01E7 1FFF	0x5A07 1000	0x5A07 1FFF	4
Reserved	0x0007 2000	0x01E7 2000	0x01E7 FFFF	0x5A07 2000	0x5A07 FFFF	56

### 6.1.3 IVA-HD Register Manual

#### CAUTION

This section provides information about ICONT memory mapping (private access). This information represents only the address offsets from the start of relevant buffered (EXT\_BUF) or nonbuffered (EXT\_NBUF) regions. For more information, see [Section 6.1.2.3, IVA-HD Memory Mapping](#).

### 6.1.3.1 IVA-HD Instance Summary

Table 6-9 shows the IVA-HD instance summary.

**Table 6-9. IVA-HD Instance Summary**

Module Name	Base Address (L3 Interconnect)	Base Address (DSP Private Access)	Base Address (ICONT Private Access)	Size
SYSCTRL	0x5A05 A400	0x01E5 A400	0x000D A400	1KB

**NOTE:** Private access is an access through the IVA-HD local interconnect without using the L3 interconnect.

### 6.1.3.2 SYSCTRL Registers

#### 6.1.3.2.1 SYSCTRL Register Summary

Table 6-10 lists the mapping summary for the SYSCTRL registers.

**CAUTION**

SYSCTRL registers are limited to 32-bit access; 16-bit and 8-bit accesses are not allowed and can corrupt register content.

**Table 6-10. SYSCTRL Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (L3 Interconnect)	Physical Address (DSP Private Access)	Physical Address (ICONT Private Access)
IVAHD_REVISION	R	32	0x0000 0000	0x5A05 A400	0x01E5 A400	0x000D A400
IVAHD_HWINFO	R	32	0x0000 0004	0x5A05 A404	0x01E5 A404	0x000D A404
IVAHD_SYSCONFIG	RW	32	0x0000 0010	0x5A05 A410	0x01E5 A410	0x000D A410
RESERVED	RW	32	0x0000 0020	0x5A05 A420	0x01E5 A420	0x000D A420
IVAHD_IRQSTATUS_RAW	RW	32	0x0000 0024	0x5A05 A424	0x01E5 A424	0x000D A424
IVAHD_IRQSTATUS	RW	32	0x0000 0028	0x5A05 A428	0x01E5 A428	0x000D A428
IVAHD_IRQENABLE_SET	RW	32	0x0000 002C	0x5A05 A42C	0x01E5 A42C	0x000D A42C
IVAHD_IRQENABLE_CLR	RW	32	0x0000 0030	0x5A05 A430	0x01E5 A430	0x000D A430
IVAHD_SYNC_IRQSTATUS_RAW	RW	32	0x0000 0034	0x5A05 A434	0x01E5 A434	0x000D A434
IVAHD_SYNC_IRQSTATUS	RW	32	0x0000 0038	0x5A05 A438	0x01E5 A438	0x000D A438
IVAHD_SYNC_IRQENABLE_SET	RW	32	0x0000 003C	0x5A05 A43C	0x01E5 A43C	0x000D A43C
IVAHD_SYNC_IRQENABLE_CLR	RW	32	0x0000 0040	0x5A05 A440	0x01E5 A440	0x000D A440
IVAHD_CLKCTRL	RW	32	0x0000 0050	0x5A05 A450	0x01E5 A450	0x000D A450
IVAHD_CLKST	R	32	0x0000 0054	0x5A05 A454	0x01E5 A454	0x000D A454
IVAHD_STDBYST	R	32	0x0000 0058	0x5A05 A458	0x01E5 A458	0x000D A458

#### 6.1.3.2.2 SYSCTRL Register Description

**Table 6-11. IVAHD\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .	<b>Instance</b>	SYSCTRL
<b>Description</b>	IP revision identifier (X.Y.R). Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI Internal Data

**Table 6-12. Register Call Summary for Register IVAHD\_REVISION**

IVA-HD Overview

- [SYSCTRL Register Summary: \[0\]](#)

**Table 6-13. IVAHD\_HWINFO**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .	<b>Instance</b>	SYSCTRL
<b>Description</b>	Information about the IP module's hardware configuration.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECD3	MC3	IPE3	CALC3	IME3	ILF3	VDMA	ICONT2	ICONT1	SL2BANK	SL2SIZE					

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	ECD3	ECD3 available 0: ECD3 not present 1: ECD3 present	R	1
13	MC3	MC3 available 0: MC3 not present 1: MC3 present	R	1
12	IPE3	iPE3 available 0: iPE3 not present 1: iPE3 present	R	1
11	CALC3	CALC3 available 0: CALC3 not present 1: CALC3 present	R	1
10	IME3	iME3 available 0: iME3 not present 1: iME3 present	R	1
9	ILF3	iLF3 available 0: iLF3 not present 1: iLF3 present	R	1
8	VDMA	vDMA available 0: vDMA not present 1: vDMA present	R	1
7	ICONT2	ICONT2 available 0: ICONT2 not present 1: ICONT2 present	R	1

Bits	Field Name	Description	Type	Reset
6	ICONT1	ICONT1 available 0: ICONT1 not present 1: ICONT1 present	R	1
5:4	SL2BANK	Read 0x0: 1 memory bank Read 0x1: 2 memory bank Read 0x2: 4 memory bank Read 0x3: 8 memory bank	R	0x3
3:0	SL2SIZE	Size of SL2 memory Read 0x1: 16KB Read 0x2: 32KB Read 0x3: 48KB Read 0x4: 64KB Read 0x5: 96KB Read 0x6: 128KB Read 0x7: 160KB Read 0x8: 192KB Read 0x9: 224KB Read 0xA: 256KB Read 0xB: 320KB Read 0xC: 384KB Read 0xD: 448KB Read 0xE: 512KB	R	0xA

**Table 6-14. Register Call Summary for Register IVAHD\_HWINFO**

IVA-HD Overview

- [SYSCTRL Register Summary: \[0\]](#)

**Table 6-15. IVAHD\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	SYSCTRL
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .		
<b>Description</b>	Clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STANDBYMODE		IDLEMODE		RESERVED			

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x00000000
5:4	STANDBYMODE	<p>Configuration of the local initiator state management mode.</p> <p>By definition, initiator may generate read/write transaction as long as it is out of STANDBY state.</p> <p>0x0 and 0x3: Reserved</p> <p>0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only.</p> <p>0x2: Smart-standby mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator. IP module shall not generate (initiator-related) wakeup events.</p>	RW	0x2
3:2	IDLEMODE	<p>Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.</p> <p>0x0 and 0x3: Reserved</p> <p>0x1: No-idle mode: local target never enters idle state. Backup mode, for debug only.</p> <p>0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events.</p>	RW	0x2
1:0	RESERVED	Reserved	R	0x0

**Table 6-16. Register Call Summary for Register IVAHD\_SYSCONFIG**

IVA-HD Overview

- [IVA-HD Power Management: \[0\] \[1\]](#)
- [SYSCCTRL Register Summary: \[2\]](#)

**Table 6-17. IVAHD\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0024		
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .	<b>Instance</b>	SYSCCTRL
<b>Description</b>	<p>Per-event raw interrupt status vector.</p> <p>Raw status is set even if event is not enabled.</p> <p>Write 1 to set the (raw) status, mostly for debug.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																																SYSCCTRL_CLKERR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	SYSCCTRL_CLKERR	<p>Settable raw status for Clock Programming Error event</p> <p>Write 0x0: No action</p> <p>Read 0x0: No event pending</p> <p>Read 0x1: Event pending</p> <p>Write 0x1: Set event (debug)</p>	RW	0

**Table 6-18. Register Call Summary for Register IVAHD\_IRQSTATUS\_RAW**

IVA-HD Overview

- [SYSCTRL Register Summary: \[1\]](#)

**Table 6-19. IVAHD\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	SYSCTRL
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .		
<b>Description</b>	Per-event "enabled" interrupt status vector, line 0. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SYSCTRL_CLKERR															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	SYSCTRL_CLKERR	Clearable, enabled status for Clock Programming Error event Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW	0

**Table 6-20. Register Call Summary for Register IVAHD\_IRQSTATUS**

IVA-HD Overview

- [SYSCTRL Register Summary: \[3\]](#)

**Table 6-21. IVAHD\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	SYSCTRL
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .		
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SYSCTRL_CLKERR															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	SYSCTRL_CLKERR	Clock Programing Error Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0

**Table 6-22. Register Call Summary for Register IVAHD\_IRQENABLE\_SET**

IVA-HD Overview

- [SYSCTRL Register Summary: \[3\]](#)

**Table 6-23. IVAHD\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	SYSCTRL
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .		
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																SYSCTRL_CLKERR																

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	SYSCTRL_CLKERR	Clock Programing Error Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

**Table 6-24. Register Call Summary for Register IVAHD\_IRQENABLE\_CLR**

IVA-HD Overview

- [SYSCTRL Register Summary: \[1\]](#)



**Table 6-25. IVAHD\_SYNC\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .	<b>Instance</b>	SYSCCTRL
<b>Description</b>	Per-event raw interrupt status vector. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SYNC_INPUT7_0															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	SYNC_INPUT7_0	Settable raw status for SYNC INPUT event. For each bit of the bit field: Read 0: No event pending Read 1: Event pending Write 0: No action Write 1: Set event (debug)	RW	0x00

**Table 6-26. Register Call Summary for Register IVAHD\_SYNC\_IRQSTATUS\_RAW**

IVA-HD Overview

- [SYSCCTRL Register Summary: \[1\]](#)

**Table 6-27. IVAHD\_SYNC\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .	<b>Instance</b>	SYSCCTRL
<b>Description</b>	Per-event "enabled" interrupt status vector, line 0. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SYNC_INPUT7_0															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	SYNC_INPUT7_0	Clearable, enabled status for SYNC INPUT event. For each bit of the bit field: Read 0: No (enabled) event pending Read 1: Event pending Write 0: No action Write 1: Clear (raw) event	RW	0x00

**Table 6-28. Register Call Summary for Register IVAHD\_SYNC\_IRQSTATUS**

IVA-HD Overview

- [SYSCCTRL Register Summary: \[1\]](#)

**Table 6-29. IVAHD\_SYNC\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 003C	
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .	<b>Instance</b> SYSCCTRL
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SYNC_INPUT7_0															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	SYNC_INPUT7_0	Enable for interrupt event. For each bit of the bit field: Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 0: No action Write 1: Enable interrupt	RW	0x00

**Table 6-30. Register Call Summary for Register IVAHD\_SYNC\_IRQENABLE\_SET**

IVA-HD Overview

- [SYSCCTRL Register Summary: \[1\]](#)

**Table 6-31. IVAHD\_SYNC\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 0040	
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .	<b>Instance</b> SYSCCTRL
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SYNC_INPUT7_0															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	SYNC_INPUT7_0	Enable for interrupt event. For each bit of the bitfield: Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 0: No action Write 1: Disable interrupt	RW	0x00

**Table 6-32. Register Call Summary for Register IVAHD\_SYNC\_IRQENABLE\_CLR**

IVA-HD Overview

- [SYSCCTRL Register Summary: \[1\]](#)

**Table 6-33. IVAHD\_CLKCTRL**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	SYSCTRL
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .		
<b>Description</b>	IVA-HD clock control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																							SMSSET	MSGIF	ECD3	MC3	IPE3	CALC3	ILF3	IME3	VDMA	ICONT2	ICONT1

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x000000
10	SMSSET	Clock control of SMSSET 0: Exit idle state and start SMSSET clock 1: Request SMSSET to go to idle state and stop SMSSET clock Note: Shutting down SMSSET clock may hang system if SW performs SW instrumentation and/or access to its configuration port.	RW	0
9	MSGIF	Clock control of MSGIF 0: Exit idle state and start MSGIF clock 1: Request MSGIF to go to idle state and stop MSGIF clock	RW	0
8	ECD3	Clock control of ECD3 0: Exit idle state and start ECD3 clock 1: Request ECD3 to go to idle state and stop ECD3 clock	RW	0
7	MC3	Clock control of MC3 0: Exit idle state and start MC3 clock 1: Request MC3 to go to idle state and stop MC3 clock	RW	0
6	IPE3	Clock control of iPE3 0: Exit idle state and start iPE3 clock 1: Request iME3 to go to idle state and stop iPE3 clock	RW	0
5	CALC3	Clock control of CALC3 0: Exit idle state and start CALC3 clock 1: Request CALC3 to go to idle state and stop CALC3 clock	RW	0
4	ILF3	Clock control of iLF3 0: Exit idle state and start iLF3 clock 1: Request iLF3 to go to idle state and stop iLF3 clock	RW	0
3	IME3	Clock control of iME3 0: Exit idle state and start iME3 clock 1: Request iME3 to go to idle state and stop iME3 clock	RW	0
2	VDMA	Clock control of vDMA 0: Exit idle state and start vDMA clock 1: Request vDMA to go to idle state and stop vDMA clock	RW	0
1	ICONT2	Clock control of ICONT2 0: Exit idle state and start ICONT2 clock 1: Request ICONT2 to go to idle state and stop ICONT2 clock	RW	0
0	ICONT1	Clock control of ICONT1 0: Exit idle state and start ICONT1 clock 1: Request ICONT1 to go to idle state and stop ICONT1 clock	RW	0

**Table 6-34. Register Call Summary for Register IVAHD\_CLKCTRL**

IVA-HD Overview

- [SYSCTRL Register Summary: \[4\]](#)

**Table 6-35. IVAHD\_CLKST**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	SYSCTRL
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .		
<b>Description</b>	IVA-HD clock status register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
RESERVED																						SMSSET	MSGIF	ECD3	MC3	IPE3	CALC3	ILF3	IME3	VDMA	ICONT2	ICONT1					

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x000000
10	SMSSET	Clock status of SMSSET 1: SMSSET clock is active 0: SMSSET clock is idled	R	0
9	MSGIF	Clock status of MSGIF 1: MSGIF clock is active 0: MSGIF clock is idled	R	0
8	ECD3	Clock status of ECD3 1: ECD3 clock is active 0: ECD3 clock is idled	R	0
7	MC3	Clock status of MC3 1: MC3 clock is active 0: MC3 clock is idled	R	0
6	IPE3	Clock status of iPE3 1: iPE3 clock is active 0: iPE3 clock is idled	R	0
5	CALC3	Clock status of CALC3 1: CALC3 clock is active 0: CALC3 clock is idled	R	0
4	ILF3	Clock status of iLF3 1: iLF3 clock is active 0: iLF3 clock is idled	R	0
3	IME3	Clock status of iME3 1: iME3 clock is active 0: iME3 clock is idled	R	0
2	VDMA	Clock status of vDMA 1: vDMA clock is active 0: vDMA clock is idled	R	0
1	ICONT2	Clock status of ICONT2 1: ICONT2 clock is active 0: ICONT2 clock is idled	R	0
0	ICONT1	Clock status of ICONT1 1: ICONT1 clock is active 0: ICONT1 clock is idled	R	0

**Table 6-36. Register Call Summary for Register IVAHD\_CLKST**

IVA-HD Overview

- [SYSCTRL Register Summary: \[1\]](#)

**Table 6-37. IVAHD\_STDBYST**

<b>Address Offset</b>	0x0000 0058	
<b>Physical Address</b>	See <a href="#">Table 6-10</a> .	<b>Instance</b> SYSCTRL
<b>Description</b>	IVA-HD STANDBY status	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											vDMA	ICONT2	ICONT1		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	vDMA	vDMA Standby status 0: module is not in Standby 1: module is in Standby	R	1
1	ICONT2	ICONT2 Standby status 0: module is not in Standby 1: module is in Standby	R	1
0	ICONT1	ICONT1 Standby status 0: module is not in Standby 1: module is in Standby	R	1

**Table 6-38. Register Call Summary for Register IVAHD\_STDBYST**

IVA-HD Overview

- [SYSCTRL Register Summary: \[2\]](#)

PRELIMINARY

## Dual Cortex-M3 MPU Subsystem

This chapter describes the dual Cortex-M3 microprocessor unit (MPU) subsystem.

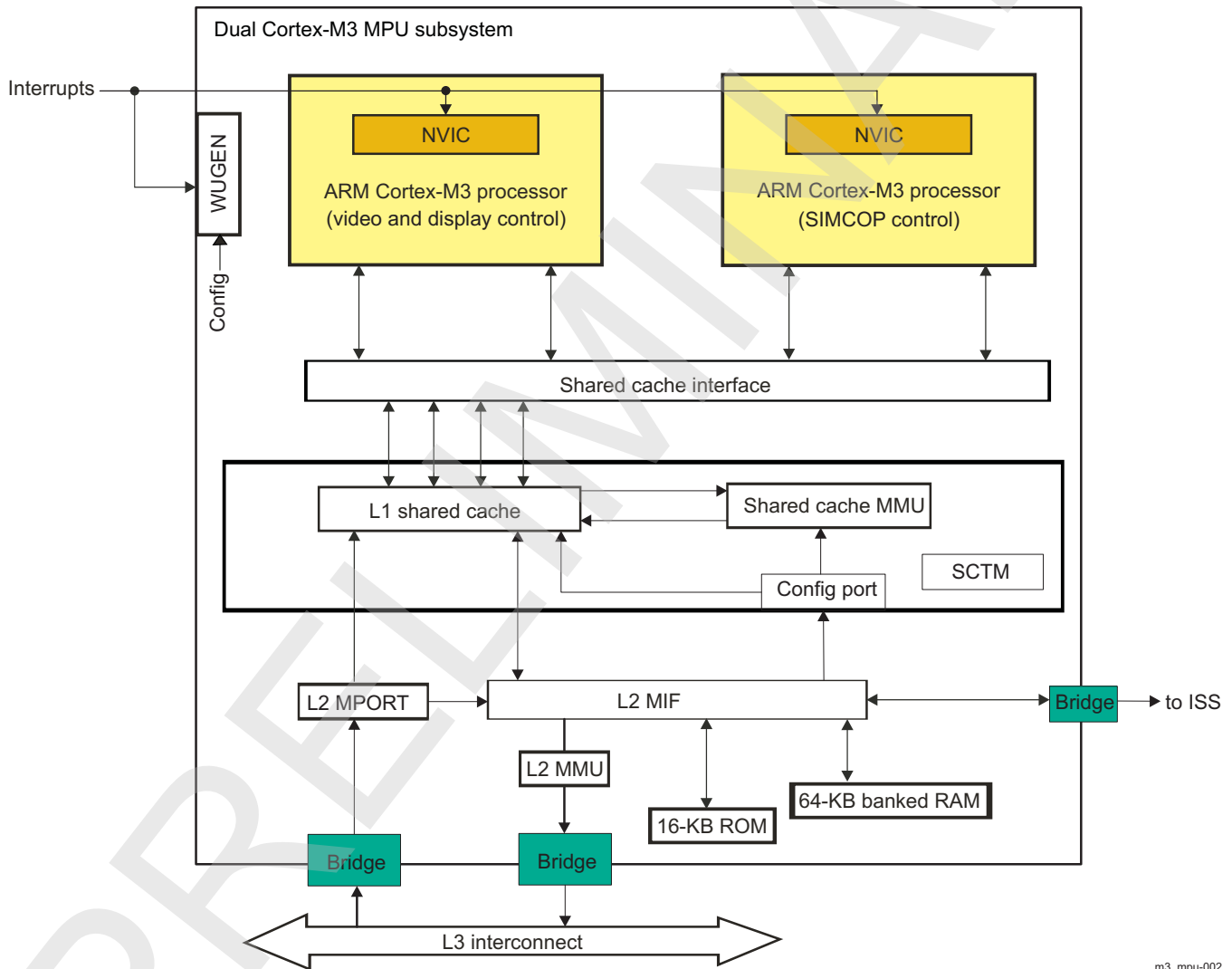
Topic	Page
7.1 Dual Cortex-M3 MPU Subsystem Overview .....	1096
7.2 Dual Cortex-M3 MPU Subsystem Integration .....	1098
7.3 Dual Cortex-M3 MPU Subsystem Functional Description .....	1102
7.4 Dual Cortex-M3 MPU Subsystem Register Manual .....	1112

## 7.1 Dual Cortex-M3 MPU Subsystem Overview

### 7.1.1 Introduction

The dual Cortex™-M3 microprocessor (MPU) subsystem controls the imaging subsystem (ISS) and manages some controls of the video and display subsystem. It contains two ARM® Cortex-M3 processors (CPUs) that share a common level 1 (L1) cache (shared cache). One of the CPUs is dedicated to sequencing still image coprocessor (SIMCOP) accelerators, and the other CPU is dedicated to the ISS and display subsystem control. A single image real-time operating system (RTOS) runs on both cores, thereby minimizing the code size. The integrated interrupt handling of the dual Cortex-M3 MPU allows efficient control of the ISS. Figure 7-1 is a high-level block diagram of the dual Cortex-M3 MPU subsystem.

**Figure 7-1. Dual Cortex-M3 MPU Subsystem Overview**



m3\_mpu-002



### 7.1.2 Features

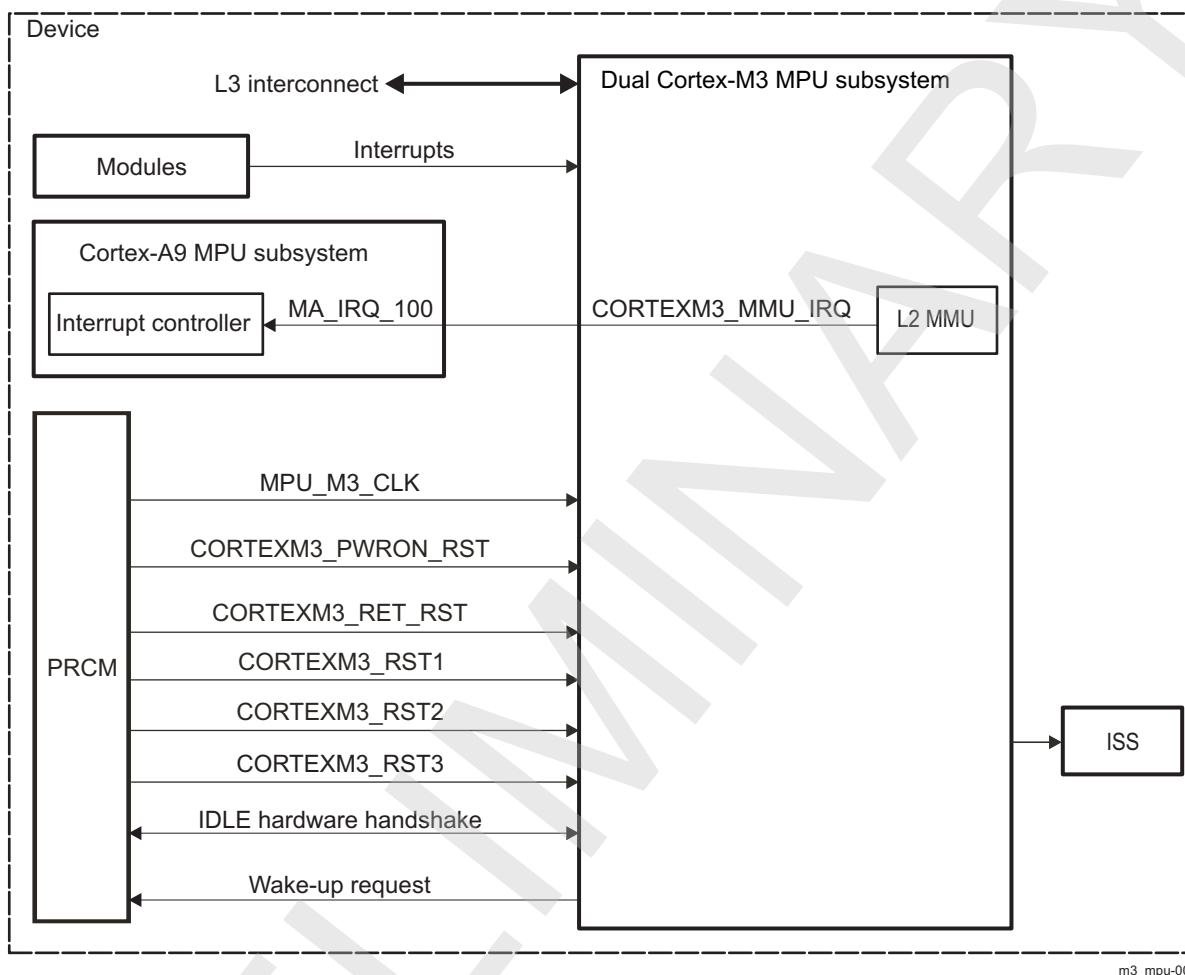
The dual Cortex-M3 MPU subsystem integrates the following:

- Two ARM Cortex-M3 microprocessor:
  - ARMv7-M and Thumb®-2 instruction set architecture
  - Hardware division and single-cycle multiplication computational acceleration
  - Integrated nested vector interrupt controller (NVIC)
  - Integrated bus matrix
  - Registers:
    - Thirteen general-purpose 32-bit registers
    - Link register (LR)
    - Program counter (PC)
    - Program status register, xPSR
    - Two banked SP registers
  - Integrated power management
  - Extensive debug capabilities
- Shared cache interface:
  - Instruction and data interface
  - Supports paralleled accesses
- Level 2 (L2) MIF (master interface) splitter for access to memory or configuration port
- Configuration port – used for shared cache maintenance and shared cache memory management unit (MMU) configuration
- Shared cache:
  - 32KB divided into 16 banks
  - 4-way
  - Cache configuration lock/freeze/preload
  - Internal MMU:
    - 16-entry region-based address translation
    - Read/write control and access type control
    - Execute Never (XN) MMU protection policy
    - Little-endian format
- SCTM – counter timer module
- On-chip ROM and banked RAM memory
- Emulation/debug: Emulation feature embedded in Cortex-M3
- L2 MMU: 32 entries with walking table logic
- Wake-up generator (WUGEN) – generates wake-up request from external interrupts
- Power management:
  - Local power-management control – configurable through the WUGEN registers.
  - Three sleep modes supported, controlled by the local power-management module.
  - Dual Cortex-M3 MPU is clock-gated in all sleep modes.
  - NVIC interrupt interface stays awake.

## 7.2 Dual Cortex-M3 MPU Subsystem Integration

Figure 7-2 shows the signals that interface with other modules.

**Figure 7-2. Dual Cortex-M3 MPU Subsystem Integration Overview**



**NOTE:** Some debug, trace, and emulation features are implemented in the MPU subsystem. This chapter includes only the clock/reset inputs and power-management aspects for these features.

Table 7-1 through Table 7-3 summarize the integration of the module in the device.

**Table 7-1. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
CORTEXM3	PD_CORE	L3

**Table 7-2. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
L2_MMU	CORTEXM3_MMU_IRQ	MA_IRQ_100	Cortex-A9 MPU INTC	L2 MMU fault interrupt

**Table 7-3. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
CORTEXM3	MPU_M3_CLK	MPU_M3_CLK	PRCM module	Interface and functional clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
CORTEXM3	CORTEXM3_PWRON_RST	CORTEXM3_PWRON_RST	PRCM module	Power-on reset, used to reset the whole dual Cortex-M3 subsystem
	CORTEXM3_RET_RST	CORTEXM3_RET_RST	PRCM module	Retention reset to few retention logic inside the shared cache
	CORTEXM3_RST1	CORTEXM3_RST1	PRCM module	Reset signal to the video and display control CPU
	CORTEXM3_RST2	CORTEXM3_RST2	PRCM module	Reset signal to the SIMCOP CPU
	CORTEXM3_RST3	CORTEXM3_RST3	PRCM module	Reset signal to the shared cache and the L2 MMU

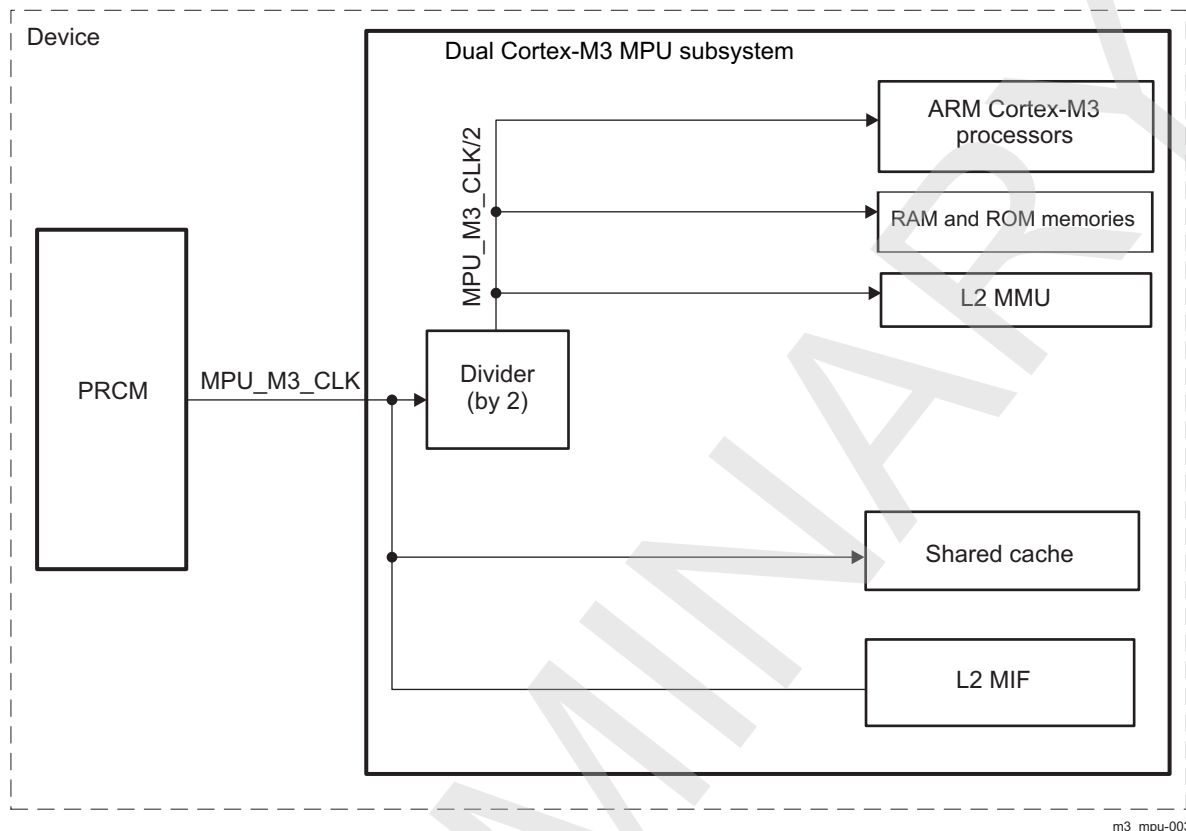
For more information about clocks, resets, and power domains, see [Chapter 3, Power, Reset, and Clock Management](#).

## 7.2.1 Dual Cortex-M3 MPU Subsystem Clock and Reset Distribution

### 7.2.1.1 Clock Distribution

The dual Cortex-M3 subsystem receives only one clock, MPU\_M3\_CLK, which is divided in two for each ARM Cortex-M3 processor, ROM and RAM memory and the L2 MMU. The shared cache and the L2 MIF are directly clocked by the MPU\_M3\_CLK, without any division. For more information, see [Chapter 3, Power, Reset, and Clock Management](#).

[Figure 7-3](#) shows the clocking scheme of the MPU subsystem.

**Figure 7-3. Dual Cortex-M3 MPU Subsystem Clocking Scheme**

### 7.2.1.2 Reset Distribution

Three reset signals controlled by the power, reset, and clock management (PRCM) module let the two ARM Cortex-M3 processors and the rest of the dual Cortex-M3 MPU subsystem be reset independently. These three reset signals are: CORTEXM3\_RST1, CORTEXM3\_RST2, and CORTEXM3\_RST3. The ARM Cortex-M3 processors must come out of reset one at a time:

- At CORTEXM3\_RST1, the video and display control ARM Cortex-M3 processor comes out of reset, but the SIMCOP ARM Cortex-M3 processor is held in reset.
- The video and display control ARM Cortex-M3 controls the reset for the SIMCOP Cortex-M3 (through the PRCM register - RM\_MPU\_M3\_RSTCTRL[1] RST2).

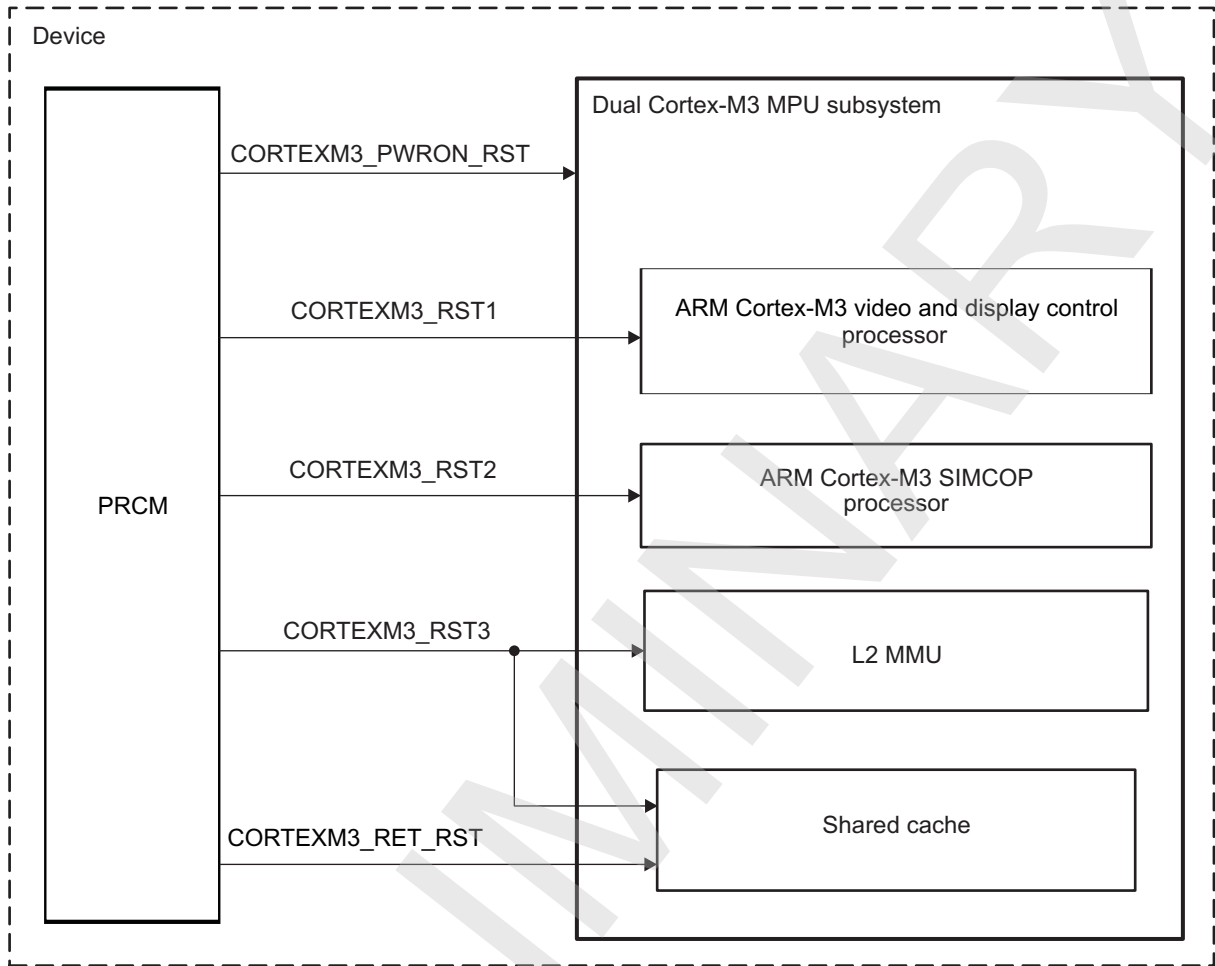
As the video and display control ARM Cortex-M3 controls the reset for the SIMCOP Cortex-M3 (through the PRCM registers), the code running on the video and display control ARM Cortex-M3 decides the mode of operation, whether it is:

- Mode 1: One ARM Cortex-M3 processor is running, and the other processor is held on reset.
- Mode 2: The two ARM Cortex-M3 processors are running.

This decision of which mode to use is driven by the use case. If the software partitioning and performance requirement for a use case requires two Cortex-M3 processor to run simultaneously, then the user must go to mode 2. If SIMCOP Cortex-M3 is not required for a particular use case then the user can remain in mode 1.

Figure 7-4 shows the reset scheme of the MPU subsystem.

Figure 7-4. Dual Cortex-M3 MPU Subsystem Reset Scheme



m3\_mpu-004

## 7.3 Dual Cortex-M3 MPU Subsystem Functional Description

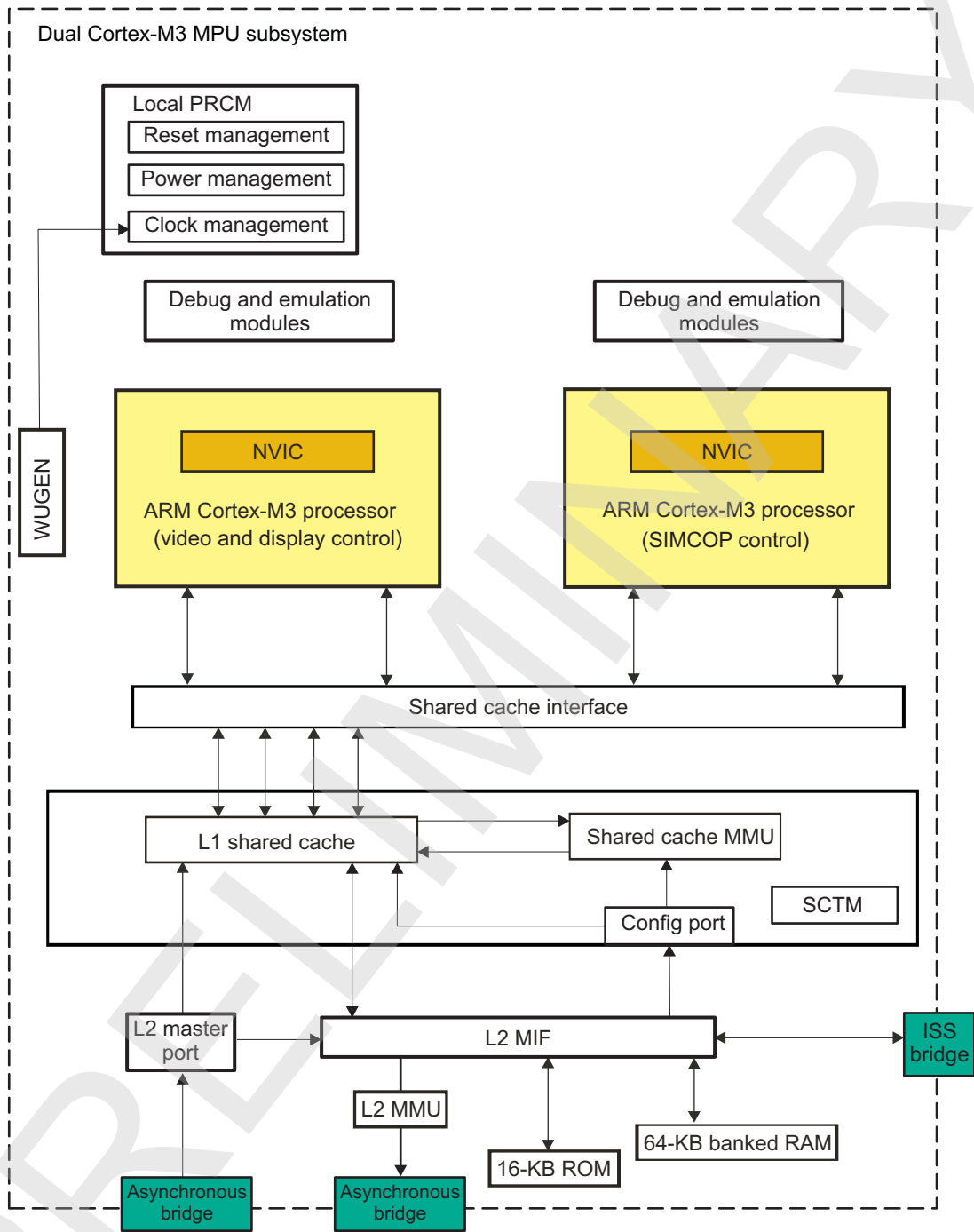
### 7.3.1 Dual Cortex-M3 MPU Subsystem Block Diagram

The dual Cortex-M3 MPU subsystem integrates the following group of submodules:

- Two ARM Cortex-M3 processors: Two cores (r2p0 revision), one dedicated to sequencing SIMCOP accelerators and the other dedicated for video and display subsystem control. For a description of the ARM Cortex-M3 processor, see the *ARM Cortex-M3 Technical Reference Manual*.
- Interrupt controller (NVIC): To facilitate parallel processing, the interrupt mapping is the same for the two cores. Each Cortex-M3 processor receives the same interrupts, except for a few internal interrupts. Every IRQ line is shared between the two ARM processors. By properly configuring the NVIC registers inside each ARM processor, it can be ensured that the shared IRQ is taken by only one of the ARM processors (for more information, see [Chapter 17, Interrupt Controller](#)).
- Shared cache interface: The cache interface converts the data between the different protocols in the subsystem. Four ports are required to support the four buses from the ARM Cortex-M3 processors (two for each processor). The instruction and data connections from each ARM Cortex-M3 are multiplexed, but the ARM Cortex-M3 prevents conflicts on this connection. Default cache policies are provided through the sideband signals and are not used to access the cache. Cacheability is provided through the MMU inside the cache.
- Shared cache: Allows basic maintenance operations, which are performed through a dedicated interface: preload, lock, clean (write out dirty lines, but do not invalidate directly), and invalidate.
- Shared cache MMU: Provides the multiple access cache with region-based address translation, read/write control, access type control, and multilevel cache maintenance. Access to the shared cache MMU is done only under privilege mode. The shared cache MMU can be programmed by the Cortex-A9 MPU subsystem through the Cortex-M3 slave port.
- SCTM: Counter Timer Module, embedded in the shared cache
- Interconnect configuration port: Cache maintenance and MMU configuration are done through an interconnect slave port. Accesses must be performed to a noncacheable area that must be defined within the shared cache MMU. Interconnect accesses are generated from the L2 MIF.
- L2 MMU: Provides address translation for all the accesses done from the Cortex-M3 to the level 3 (L3) interconnect. The L2 MMU can be programmed by the Cortex-A9 MPU subsystem through the Cortex-M3 slave port.
- L3 interconnect port: Allows access to the system memories and peripherals. For the address mapping of the modules in the dual Cortex-M3 MPU subsystem, see [Chapter 2, Memory Mapping](#).
- On-chip ROM and banked RAM memory. The ROM memory is using for boot/initialization purpose. For more information about the initialization of the device, see [Chapter 27, Initialization](#). The address range of the ROM memory is from 0x5500\_0000 to 0x5500\_3FFF. The address range of the on-chip banked RAM memory is from 0x5502\_0000 to 0x5502\_FFFF.

[Figure 7-5](#) is the block diagram of the dual Cortex-M3 MPU subsystem.

Figure 7-5. Dual Cortex-M3 MPU Block Diagram



m3\_mpu-006

### 7.3.2 Power Management

**NOTE:** For information about source clock gating and for a description of the sleep/wake-up transitions, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

### 7.3.2.1 Local Power Management

The user can configure the local power management through the [STANDBY\\_CORE\\_SYSCONFIG](#), [IDLE\\_CORE\\_SYSCONFIG](#), and [WUGEN\\_IRQ\\_EN](#) registers. The user can:

- Configure the different standby modes (the default is smart wake-up standby mode) through the [STANDBY\\_CORE\\_SYSCONFIG\[1:0\]](#) STANDBYMODE bit field
- Configure the different idle modes (the default is smart wake-up idle mode) through the [IDLE\\_CORE\\_SYSCONFIG\[1:0\]](#) IDLEMODE bit field
- Control which interrupts will cause a wakeup by appropriately configuring the [WUGEN\\_MEVT0](#) and [WUGEN\\_MEVT1](#) registers (the default is ALL\_MASKED)

The Cortex-M3 provides three sleep modes:

- On mode: Sleep on exit (wait for the interrupt service routine [ISR] to complete)
- Sleep mode: Wait for interrupt to wake up the processor
- Deep-sleep mode: Long duration sleep; phase-locked loop (PLL) can be stopped

During sleep mode, the system clock can be stopped, but the free-running clock input must still be running to allow the processor to be wakened by an interrupt or an event. The sleep modes are invoked by wait for interrupt (WFI) or wait for event (WFE) instructions. The processor clock is automatically stopped, waiting for an interrupt or an event. Deep-sleep mode also stops the processor clock, but this can also be supported by the PRCM module. A combined signal is generated from the two Cortex-M3 processors in deep-sleep mode to initiate another power state and let the PRCM module handle the next power states. At this time, software must ensure that all shared cache background operations (for example, maintenance) are complete.

[Table 7-4](#) describes local clock gating.

**Table 7-4. Local Clock Gating**

Cortex-M3 CPU Mode	CPU2 On	CPU2 Sleep	CPU2 Deep Sleep
<b>CPU1 On</b>	On	Functional clock 2 stopped locally	Functional clock 2 stopped locally
<b>CPU1 Sleep</b>	Functional clock 1 stopped locally	Functional clock 1 and clock 2 stopped locally	Functional clock 1 and clock 2 stopped locally
<b>CPU1 Deep Sleep</b>	Functional clock 1 stopped locally	Functional clock 1 and clock 2 stopped locally	Standby request to power-management module

### 7.3.2.2 Power Domains

The dual Cortex-M3 MPU subsystem is divided into two power domains (CORE and WKUP), which are controlled by the PRCM module (see [Section 7.3.2.3](#)).

The CORE power domain is the main power domain and includes all the Cortex-M3 components (two ARM Cortex-M3 processors, shared cache, ROM and RAM memories, and emulation\debug modules) except the WUGEN.

The AWLON\_CORE power domain is an always-on power domain. The AWLON\_CORE power domain contains the WUGEN, which generates a wake-up request from external interrupts. By this separate AWLON\_CORE power domain, the wake-up request can be generated even when the CORE power domain is in OFF or RET state.

For information about the CORE and ALWON\_CORE power domains, see [Chapter 3, Power, Reset, and Clock Management](#).

### 7.3.2.3 Voltage Domain

The dual Cortex-M3 MPU subsystem is composed of two voltage domains. Each voltage domain can control individual voltage levels:

- VDD1: Cortex-M3, shared cache/MMU, WUGEN
- VDD\_array: All memories array



The AWLON\_CORE and CORE power domains are in the same voltage domain – the CORE voltage domain (CORE\_VD).

For information about the physical power domains, see [Chapter 3, Power, Reset, and Clock Management](#).

### 7.3.2.4 Power States and Modes

[Table 7-5](#) lists the different power modes and the expected states for each power domain.

**Table 7-5. Dual Cortex-M3 MPU Power Modes**

	Functional Domain	Activity			Power Status	
		Core		Dual Cortex-M3 MPU Subsystem	CORE Power Domain	ALWON_CORE Power Domain
	Modules included	Cortex-M3 processor 1	Cortex-M3 processor 2	WUGEN		
Power modes	Active	Active	Active	Active	ON	ON
	Processor 1 idle	Idle	Active	Active	ON	ON
	Processor 2 idle	Active	Idle	Active	ON	ON
	Core standby	Idle	Idle	Active	ON	ON
	Full idle	Idle	Idle	Idle	ON	ON
	Retention	Idle	Idle	Idle	RET	ON
	Power off	Idle	Idle	Idle	OFF	OFF

[Table 7-6](#) lists the power mode transitions in the dual Cortex-M3 MPU subsystem.

**Table 7-6. Power Mode Transitions**

	Active	Processor 1 Idle	Processor 2 Idle	Standby	Full-Idle	Retention	Power Off
Active		WFE/WFI instruction	WFE/WFI instruction				
Processor 1 idle	Events/interrupts			Deep sleep			
Processor 2 idle	Events/interrupts			Deep sleep			
Standby	Wake-up IRQ				L1/L2/WUGEN functional domain idle		
Full idle	Wake-up IRQ			Wake-up (through interconnect)		PRCM module	PRCM module
Retention					PRCM module or wakeup		PRCM module
Power off					PRCM module or wakeup	PRCM module	

The different power modes and their features are:

- Active mode: All function domains are operable.
- Core 1 and Core 2 idle mode:
  - Only the CPU core is idled (when running WFE/WFI instructions).
  - Only one Cortex-M3 core can be in this mode. Interrupts or events can waken the core.
  - Can go into sleep or deep-sleep mode. Potentially, both cores can be in sleep mode.
  - When both cores are in deep-sleep mode, a standby request is sent to the PRCM module.
  - Software must ensure that all shared cache background operations (for example, maintenance) are complete before the PRCM module asserts an IdleReq. For more information, see [Chapter 5, DSP, Direct Maintenance of Caches](#).
- Core standby mode

- Both cores in the CORE functional domain are in idle mode (an interrupt cannot wake up either of the cores).
- The PRCM module must have acknowledged its acceptance by an MWait signal.
- After this handshake, all power management is under the control of the PRCM module.
- Full-idle mode:
  - The dual Cortex-M3 MPU subsystem functional domain is also idled.
  - After coming to this mode, power states can be moved deeper.
- Retention mode:
  - The voltage of the logic supply is lowered to reduce static power consumption by leakage current. The logic power switch in the CORE power domain is still closed (ON); thus, all logic states are retained.
  - L1 and/or L2 memories can go independently into retention depending on the settings done at the PRCM level.
- Power off mode:
  - The voltage source is shut down. The logic states, including the retention logic, are lost.
  - The WUGEN is not operating and only the PRCM module can trigger the Cortex-M3 wakeup.
  - Reset must be applied to the dual Cortex-M3 MPU subsystem to restart the two ARM Cortex-M3 processors and the memory subsystem.

### 7.3.2.5 Wake-Up Generator

The WUGEN in the dual Cortex-M3 MPU subsystem enables efficient power management. The WUGEN generates a wake-up signal to the PRCM module to enable the dual Cortex-M3 MPU subsystem to recover its functional clocks, which are gated by the PRCM module when at least one request is active. The WUGEN can be configured in standby mode or idle mode through the [STANDBY\\_CORE\\_SYSCONFIG\[1:0\] STANDBYMODE](#) and [IDLE\\_CORE\\_SYSCONFIG\[1:0\] IDLEMODE](#) bit fields.

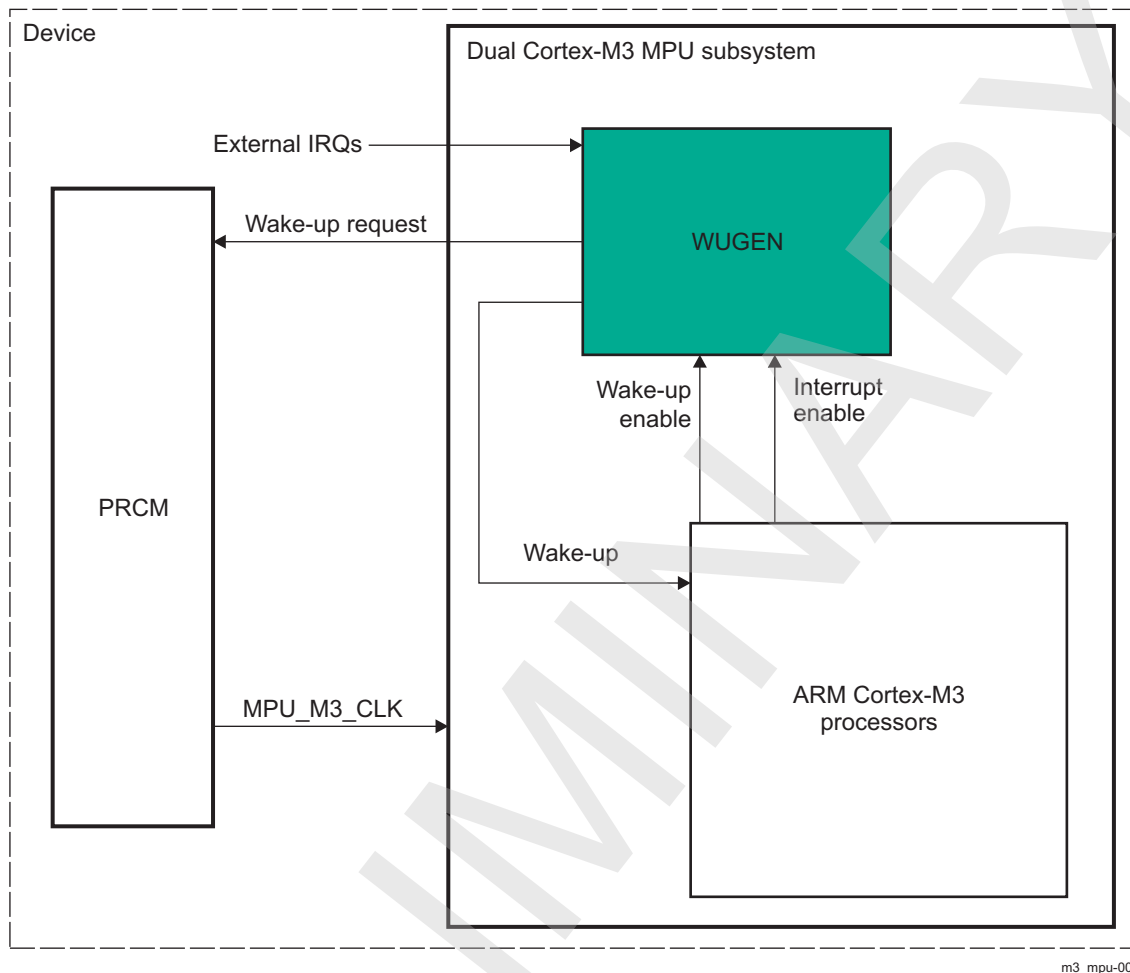
#### 7.3.2.5.1 WUGEN Main Features

The WUGEN allows:

- Gating of the Cortex-M3 MPU clock dynamically, thus reducing power consumption
- Simplifying of dependencies in the PRCM module

[Figure 7-6](#) is an overview of the WUGEN.

Figure 7-6. WUGEN Overview



The wake-up signal to the PRCM module requests the Cortex-M3 functional clock (MPU\_M3\_CLK). The wake-up signal to the ARM Cortex-M3 processors indicates to them that at least one enabled request is active.

Two retention registers [WUGEN\\_MEVT0](#) (for CPU1) and [WUGEN\\_MEVT1](#) (for CPU2), enable the WUGEN requests to the NVICs.

### 7.3.3 Shared Cache

[Table 7-7](#) describes the shared cache configuration in the dual Cortex-M3 MPU subsystem platform.

Table 7-7. Shared Cache Configuration

Parameter	Value
Way	4
Size	32KB
Bank elements	32 bits
Bank number	16
Slave interface data size	32 bits
Master interface data size	64 bits
Line size	256 bits
MMU lookup	Included
Number of slaves	4

**Table 7-7. Shared Cache Configuration (continued)**

Parameter	Value
Number of masters	1
Number of fill/prefetch buffers	Four prefetch buffers
Slave types	Shared cache interface

Shared cache allows basic maintenance operations, which are performed through a dedicated interface:

1. Preload
2. Lock
3. Clean
4. Invalidate

Maintenance of the cache is performed between the start and end addresses. This allows for direct control of memory regions. All maintenance operations occur in the background and can generate an interrupt when they complete. Such operations are protected by software semaphore, because only one operation at a time can be performed. The maintenance operations can also be performed using MMU small entries.

The shared cache is the same as the one in the DSP subsystem. For more information about the shared cache, see [Chapter 5, DSP Subsystem](#).

### 7.3.4 Shared Cache MMU

The MMU for the shared cache provides the multi-access cache with region-based address translation, read/write control, access type control, and multilevel cache maintenance. [Table 7-8](#) describes the MMU configuration in the dual Cortex-M3 MPU subsystem.

**Table 7-8. Shared cache MMU Configuration**

Parameter	Values
Number of large pages	Four entries
Size of large pages	512MB or 32MB (configurable)
Number of medium pages	Two entries
Size of medium pages	256KB or 128MB (configurable)
Number of small pages	10 entries
Size of small pages	16KB or 4KB (configurable)
Number of patch pages	Not included
Size of line pages	256-bit
Number of comparison interfaces	4
Number of comparator sets	1
Write pipeline data comparison	Disabled
Number of shared cache maintenance interfaces	3
Size of entry address	32-bit

The size of the pages is configurable in the shared cache MMU registers. The different MMU page sizes can be used to create smaller policies within a larger region. The shared cache MMU is the same as the one in the DSP subsystem. For more information, see [Chapter 5, DSP Subsystem](#).

### 7.3.5 L2 MMU

An additional MMU provides address translation for the accesses done from the dual Cortex-M3 MPU subsystem to the L3 interconnect. The main characteristics of this MMU are:

- 32 entries
- Compatible with ARMv6 architecture MMU translation tables (protection bits not used)
- Page-based or access-based endianness conversion
- Two-level descriptor hierarchy

- One intermediate page table
- Four page sizes (16MB, 1MB, 64KB, 4KB)
- Page table alignment on 128-byte boundary for ARM11® compatibility

The configuration of the MMU can be done from one of the Cortex-M3 cores or from the L3 interconnect slave port. The accesses done to configure the MMU cannot be part of a burst access.

For more information about the L2 MMU, see [Chapter 20, Memory Management Unit](#).

### 7.3.5.1 L2 MMU Behavior on Page-Fault in Dual Cortex-M3 MPU

**Table 7-9. L2 MMU Behavior on Page-Fault**

Application		Debug	
Table-Walker Enabled	Table-Walker Disabled	Table-Walker Enabled	Table-Walker Disabled
Use Table-Walker to find translation. Update TLB cache if successful, set TRANSLATIONFAULT bit and interrupt if not. The following bits are used for the purpose: MMU_IRQENABLE[1] TRANSLATIONFAULT and MMU_IRQSTATUS[1] TRANSLATIONFAULT.	Set TLBMISS bit and interrupt and stall. The following registers are used for the purpose: MMU_IRQENABLE[0] TLBMISS and MMU_IRQSTATUS[0] TLBMISS.	Use Table-Walker to find translation. Update TLB cache if successful (only if MMU_CNTL[3] EMUTLBUPDATE is set), generate in-band bus error if not.	Set EMUMISS bit and interrupt and stall. The following bits are used for the purpose: MMU_IRQENABLE[2] EMUMISS and MMU_IRQSTATUS[2] EMUMISS.

The MMU fault interrupt line is connected to both Cortex-M3 cores and also propagated outside of Cortex-M3 and connected to the Cortex-A9 MPU (MA\_IRQ\_100). The Cortex-A9 MPU receives the MMU fault and must clean up the fault to resume the execution of the code (or reset Cortex-M3). It is not possible for one of the Cortex-M3 CPUs to clean up the fault caused by the other Cortex-M3 CPU. This is because both the slave port of L2 MMU (which is stalled) and the configuration port of L2 MMU is connected (through a splitter) to the same shared cache master port.

The above default behavior of the L2 MMU can be overridden by setting the MMU\_GP\_REG[0] BUS\_ERR\_BACK\_EN bit to 1. Once this bit is set, all MMU faults (including TLB miss) return a bus error to the Cortex-M3 (interrupt event XLATE\_MMU\_FAULT). This allows the end user to quickly establish the cause of the MMU fault by having appropriate code in the ISR.

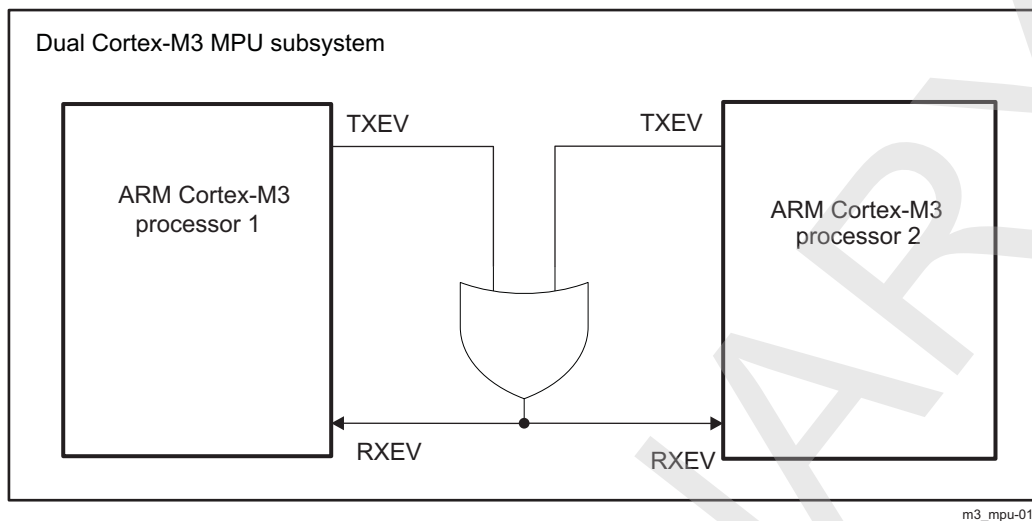
## 7.3.6 Interprocessor Communication (IPC)

### 7.3.6.1 Use of WFE and SEV

The dual Cortex-M3 MPU subsystem provides multiprocessor communication interface for synchronizing tasks. The ARM processors have one output signal, TXEV (transmit event), for sending events and one input signal, RXEV (receive event), for receiving events. [Figure 7-7](#) shows how TXEV and RXEV are connected in the dual Cortex-M3 MPU subsystem.

When a WFE instruction is executed, the processor enters into sleep mode waiting for an event and continues instruction execution when an external event is received. With an SEV (send event) instruction, one processor can wake up the other processor, which is in sleep mode.

The WFE and SEV instructions can help reduce the number of iterations around a lock acquire loop (a spinlock), and thereby reduce power consumption. The basic mechanism involves an observer that is in a spinlock executing a WFE instruction, which suspends execution on that observer until an asynchronous exception or an explicit event (sent by an observer using the SEV instruction) is seen by that observer. The observer that holds the lock uses the SEV instruction to send an event after a lock is released.

**Figure 7-7. Event Communication Connection in Dual Cortex-M3 MPU Subsystem**

### 7.3.6.2 Use of Interrupt for IPC

Each Cortex-M3 can interrupt the other Cortex-M3 by setting up an interrupt register ([CORTEXM3\\_CTRL\\_REG](#)). Because the priority level for that interrupt can be defined, it is possible to choose the task level at which the interrupt will run. For example, if CPU1 was active and CPU2 was idle (WFI state), when CPU1 completes its task it sets the bit for CPU2 in the control register ([CORTEXM3\\_CTRL\\_REG\[16\] INT\\_CORTEX\\_2](#)) and goes into sleep mode. CPU2 wakes up seeing this interrupt, and starts running its task. After the completion of its task, CPU2 sets the interrupt for CPU1 ([CORTEXM3\\_CTRL\\_REG\[0\] INT\\_CORTEX\\_1](#)), and then goes into WFI state. This kind of handshake ensures that if CPU1 and CPU2 are accessing the same resources (memory, registers etc.), only one of the CPUs at a time is active.

### 7.3.6.3 Use of the Bit-Band Feature for Semaphore Operations

The two Cortex-M3 cores share the same memory system, and it is possible to use the bit-band feature to carry semaphore operations. Because the bit-band alias writes are locked read-modify-write transfers, provided that all tasks changed only the lock bit representing themselves, the lock bits of other tasks are not lost, even if two tasks try to write to the same memory location at the same time.

Each Cortex-M3 core supports two bit-band regions.

Bit-band 1 applies to the virtual address space 0x2000 0000 – 0x200F FFFF (1MB). This virtual address space can be mapped to any physical address and bit-banding will apply to that region. It is recommended that the user map the L2 RAM (64KB) to this virtual space and use it only for bit-banding operations. If required, the user can define other available small and medium pages over and above the L2 RAM virtual space and further extend the use of the bit-band feature.

Bit-band 2 applies to the virtual address space 0x4000 0000 – 0x400F FFFF (1MB). The first 16KB of this space (0x4000 0000 – 0x4000 3FFF) are already reserved for small (one page) pages and cannot be remapped by software. The bit-band alias that corresponds to this 16-KB region (0x4200 0000 – 0x4207 FFFF) must also be treated as reserved and no access should be made. The rest of bit-band 2 can be used by appropriately defining the available small and medium pages. In this device, because it is likely that during normal AMMU programming all of L3 is mapped to this region, it is highly recommended that the user use only bit-band 1 for all purposes and bit-band 2 only if it is necessary.

### 7.3.6.4 Private Memory Space

Each ARM Cortex-M3 processor has its own memory space, unaccessible by the other processor. In the private memory space are the NVICs and RW table registers: [CORTEXM3\\_RW\\_PID1](#) and

[CORTEXM3\\_RW\\_PID2](#), [CORTEXM3\\_RW\\_PID1](#) and [CORTEXM3\\_RW\\_PID2](#) are only accessible by the respective Cortex-M3 cores ([CORTEXM3\\_RW\\_PID1](#) is accessible only to the video and display control ARM processor, while [CORTEXM3\\_RW\\_PID2](#) is only accessible by the SIMCOP processor) . These registers are not accessible from the Cortex-A9 MPU. Because they are not shared, they do not require bit-band feature (semaphore) to read and write to them.

## 7.4 Dual Cortex-M3 MPU Subsystem Register Manual

### 7.4.1 Dual Cortex-M3 Subsystem Instance Summary

Table 7-10 summarizes the dual Cortex-M3 MPU instance.

**Table 7-10. Dual Cortex-M3 MPU Subsystem Instance Summary**

Module Name	Base Address	Size
SCACHE_CFG	0x5508 0000	4KB
SCACHE_SCTM	0x5508 0400	4KB
SCACHE_MMU	0x5508 0800	4KB
CORTEXM3_WKUP	0x5508 1000	4KB
CORTEXM3_MMU	0x5508 2000	4KB
CM3_NVIC_CPU1 <sup>(1)</sup>	0xE000 E000	4KB
CM3_NVIC_CPU2 <sup>(2)</sup>	0xE000 E000	4KB
CM3_RW_Table <sup>(3)</sup>	0xE00F E000	4KB

<sup>(1)</sup> Private memory space for CPU1

<sup>(2)</sup> Private memory space for CPU2

<sup>(3)</sup> Private memory space for each CPU

### 7.4.2 Shared Cache Configuration Registers

For information about the shared cache configuration registers and their description, see [Chapter 5, DSP Subsystem](#).

### 7.4.3 Shared Cache SCTM Registers

For information about the shared cache SCTM registers and their description, see [Chapter 5, DSP Subsystem](#).

### 7.4.4 Shared Cache MMU Registers

For information about the shared cache MMU registers and their description, see [Chapter 5, DSP Subsystem](#).

The shared cache MMU in Cortex-M3 has four large pages, two medium pages, and ten small pages. Because of the different page sizes, there are differences with the registers described in [Chapter 5, DSP Subsystem](#), in the enumerated indexes below the register mapping, which are the following: (i = 0 to 3), (j = 0 to 1), (k = 0 to 9). There are also differences in the reset value of the following registers:

- SCACHE\_MMU\_SMALL\_ADDR\_0 with reset value 0x00000000
- SCACHE\_MMU\_SMALL\_ADDR\_1 with reset value 0x40000000
- SCACHE\_MMU\_SMALL\_POLICY\_k with reset value 0x00000000 (k = 0 to 9)
- SCACHE\_MMU\_SMALL\_XLTE\_0 with reset value 0x55020000
- SCACHE\_MMU\_SMALL\_XLTE\_1 with reset value 0x55080000
- SCACHE\_MMU\_SMALL\_XLTE\_k with reset value 0x00000000 (k = 2 to 9)

The page sizes for the 16 entries are:

- Large pages (four): The page size supported are 32 MB and 512 MB.
- Medium pages (two): The page sizes supported are 128 KB and 256 KB.
- Small pages (ten): The page sizes supported are 4 KB and 16 KB.

### 7.4.5 Cortex-M3 L2 MMU Registers

For information about the L2 MMU registers and their description, see [Chapter 20, Memory Management Unit](#).



### 7.4.6 Cortex-M3 NVIC Registers

For information about the Cortex-M3 MPU NVIC registers and their description, see the ARM Cortex-M3 Technical Reference Manual .

### 7.4.7 Cortex-M3 Wake-Up Generator Registers

#### 7.4.7.1 CORTEXM3\_WKUP Register Summary

Table 7-11 summarizes the dual Cortex-M3 MPU register mapping.

**Table 7-11. CORTEXM3\_WKUP Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Cortex-M3 Base Address
<a href="#">CORTEXM3_CTRL_REG</a>	RW	32	0x0000 0000	0x5508 1000
<a href="#">STANDBY_CORE_SYSCONFIG</a>	RW	32	0x0000 0004	0x5508 1004
<a href="#">IDLE_CORE_SYSCONFIG</a>	RW	32	0x0000 0008	0x5508 1008
<a href="#">WUGEN_MEVT0</a>	RW	32	0x0000 000C	0x5508 100C
<a href="#">WUGEN_MEVT1</a>	RW	32	0x0000 0010	0x5508 1010
RESERVED	R	32	0x0000 0014	0x5508 1014

#### 7.4.7.2 Dual Cortex-M3 MPU Wake-Up Generator Register Description

**Table 7-12. CORTEXM3\_CTRL\_REG**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 7-11</a> .	<b>Instance</b>	CORTEXM3_WKUP
<b>Description</b>	The register is used by one CPU to interrupt the other, thus used as a handshake between the two CPUs 0x0: Interrupt is cleared; 0x1: Interrupt is set.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																RESERVED																	
																INT_CORTEX_2																	INT_CORTEX_1

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		RW	0x0000 0000
16	INT_CORTEX_2	Interrupt to ARM Cortex-M3 CPU2	RW	0
15:1	RESERVED		RW	0x0000 0000
0	INT_CORTEX_1	Interrupt to ARM Cortex-M3 CPU1	RW	0

**Table 7-13. Register Call Summary for Register CORTEXM3\_CTRL\_REG**

Dual Cortex-M3 MPU Subsystem Functional Description

- [Use of Interrupt for IPC: \[0\] \[1\] \[2\]](#)

Dual Cortex-M3 MPU Subsystem Register Manual

- [CORTEXM3\\_WKUP Register Summary: \[3\]](#)

**Table 7-14. STANDBY\_CORE\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	CORTEXM3_WKUP
<b>Physical Address</b>	See <a href="#">Table 7-11</a> .		
<b>Description</b>	Standby protocol		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STANDBYMODE															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		RW	0x0000 0000
1:0	STANDBYMODE	0x0: Force-standby mode 0x1: No-standby mode 0x2: Smart-standby mode 0x3: Smart-standby wakeup mode - normal mode to be used	RW	0x3

**Table 7-15. Register Call Summary for Register STANDBY\_CORE\_SYSCONFIG**

Dual Cortex-M3 MPU Subsystem Functional Description

- [Local Power Management: \[0\] \[1\]](#)
- [Wake-Up Generator: \[2\]](#)

Dual Cortex-M3 MPU Subsystem Register Manual

- [CORTEXM3\\_WKUP Register Summary: \[3\]](#)

**Table 7-16. IDLE\_CORE\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	CORTEXM3_WKUP
<b>Physical Address</b>	See <a href="#">Table 7-11</a> .		
<b>Description</b>	Idle protocol		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEMODE															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		RW	0x0000 0000
1:0	IDLEMODE	0x0: Force-idle mode 0x1: No-idle mode 0x2: Smart-idle mode 0x3: Smart-idle wakeup mode - normal mode to be used	RW	0x3

**Table 7-17. Register Call Summary for Register IDLE\_CORE\_SYSCONFIG**

Dual Cortex-M3 MPU Subsystem Functional Description

- [Local Power Management: \[0\] \[1\]](#)
- [Wake-Up Generator: \[2\]](#)

Dual Cortex-M3 MPU Subsystem Register Manual

- [CORTEXM3\\_WKUP Register Summary: \[3\]](#)

**Table 7-18. WUGEN\_MEVT0**

<b>Address Offset</b>	0x0000 000C	<b>Instance</b>	CORTEXM3_WKUP
<b>Physical Address</b>	See <a href="#">Table 7-11</a> .		
<b>Description</b>	This register contains the interrupt mask (LSB) wake-up enable bit per interrupt request 0x0: Interrupt is disabled; 0x1: Interrupt is enabled.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIRQ31	RESERVED	MIRQ29	MIRQ28	MIRQ27	MIRQ26	MIRQ25	MIRQ24	MIRQ23	MIRQ22	MIRQ21	MIRQ20	MIRQ19	MIRQ18	MIRQ17	MIRQ16	MIRQ15	MIRQ14	MIRQ13	MIRQ12	MIRQ11	MIRQ10	MIRQ9	MIRQ8	MIRQ7	RESERVED						

Bits	Field Name	Description	Type	Reset
31	MIRQ31	Interrupt Mask bit 31	RW	0
30	RESERVED	Reserved	R	0
29	MIRQ29	Interrupt Mask bit 29	RW	0
28	MIRQ28	Interrupt Mask bit 28	RW	0
27	MIRQ27	Interrupt Mask bit 27	RW	0
26	MIRQ26	Interrupt Mask bit 26	RW	0
25	MIRQ25	Interrupt Mask bit 25	RW	0
24	MIRQ24	Interrupt Mask bit 24	RW	0
23	MIRQ23	Interrupt Mask bit 23	RW	0
22	MIRQ22	Interrupt Mask bit 22	RW	0
21	MIRQ21	Interrupt Mask bit 21	RW	0
20	MIRQ20	Interrupt Mask bit 20	RW	0
19	MIRQ19	Interrupt Mask bit 19	RW	0
18	MIRQ18	Interrupt Mask bit 18	RW	0
17	MIRQ17	Interrupt Mask bit 17	RW	0
16	MIRQ16	Interrupt Mask bit 16	RW	0
15	MIRQ15	Interrupt Mask bit 15	RW	0
14	MIRQ14	Interrupt Mask bit 14	RW	0
13	MIRQ13	Interrupt Mask bit 13	RW	0
12	MIRQ12	Interrupt Mask bit 12	RW	0
11	MIRQ11	Interrupt Mask bit 11	RW	0
10	MIRQ10	Interrupt Mask bit 10	RW	0
9	MIRQ9	Interrupt Mask bit 9	RW	0
8	MIRQ8	Interrupt Mask bit 8	RW	0
7	MIRQ7	Interrupt Mask bit 7	RW	0
6:0	RESERVED	Reserved	R	0

**Table 7-19. Register Call Summary for Register WUGEN\_MEVT0**

Dual Cortex-M3 MPU Subsystem Functional Description

- [Local Power Management: \[0\]](#)
- [Wake-Up Generator: \[1\]](#)

Dual Cortex-M3 MPU Subsystem Register Manual

- [CORTEXM3\\_WKUP Register Summary: \[2\]](#)

**Table 7-20. WUGEN\_MEVT1**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	CORTEXM3_WKUP
<b>Physical Address</b>	See <a href="#">Table 7-11</a> .		
<b>Description</b>	This register contains the interrupt mask (MSB) wake-up enable bit per interrupt request 0x0: Interrupt is disabled; 0x1: Interrupt is enabled.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIRQ63	MIRQ62	MIRQ61	MIRQ60	MIRQ59	MIRQ58	MIRQ57	MIRQ56	RESERVED	MIRQ54	MIRQ53	MIRQ52	MIRQ51	MIRQ50	RESERVED	MIRQ48	MIRQ47	MIRQ46	RESERVED	RESERVED	RESERVED	MIRQ42	MIRQ41	MIRQ40	MIRQ39	MIRQ38	MIRQ37	MIRQ36	MIRQ35	MIRQ34	MIRQ33	RESERVED

Bits	Field Name	Description	Type	Reset
31	MIRQ63	Interrupt Mask bit 63	RW	0
30	MIRQ62	Interrupt Mask bit 62	RW	0
29	MIRQ61	Interrupt Mask bit 61	RW	0
28	MIRQ60	Interrupt Mask bit 60	RW	0
27	MIRQ59	Interrupt Mask bit 59	RW	0
26	MIRQ58	Interrupt Mask bit 58	RW	0
25	MIRQ57	Interrupt Mask bit 57	RW	0
24	MIRQ56	Interrupt Mask bit 56	RW	0
23	RESERVED	Reserved	R	0
22	MIRQ54	Interrupt Mask bit 54	RW	0
21	MIRQ53	Interrupt Mask bit 53	RW	0
20	MIRQ52	Interrupt Mask bit 52	RW	0
19	MIRQ51	Interrupt Mask bit 51	RW	0
18	MIRQ50	Interrupt Mask bit 50	RW	0
17	RESERVED	Reserved	R	0
16	MIRQ48	Interrupt Mask bit 48	RW	0
15	MIRQ47	Interrupt Mask bit 47	RW	0
14	MIRQ46	Interrupt Mask bit 46	RW	0
13:11	RESERVED	Reserved	R	0
10	MIRQ42	Interrupt Mask bit 42	RW	0
9	MIRQ41	Interrupt Mask bit 41	RW	0
8	MIRQ40	Interrupt Mask bit 40	RW	0
7	MIRQ39	Interrupt Mask bit 39	RW	0
6	MIRQ38	Interrupt Mask bit 38	RW	0
5	MIRQ37	Interrupt Mask bit 37	RW	0
4	MIRQ36	Interrupt Mask bit 36	RW	0
3	MIRQ35	Interrupt Mask bit 35	RW	0
2	MIRQ34	Interrupt Mask bit 34	RW	0
1	MIRQ33	Interrupt Mask bit 33	RW	0

Bits	Field Name	Description	Type	Reset
0	RESERVED	Reserved	R	0

**Table 7-21. Register Call Summary for Register WUGEN\_MEVT1**

Dual Cortex-M3 MPU Subsystem Functional Description

- [Local Power Management: \[0\]](#)
- [Wake-Up Generator: \[1\]](#)

Dual Cortex-M3 MPU Subsystem Register Manual

- [CORTEXM3\\_WKUP Register Summary: \[2\]](#)

## 7.4.8 Cortex-M3 RW Table Registers

### 7.4.8.1 CM3\_RW\_Table Register Summary

**Table 7-22. CM3\_RW\_Table Register Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Cortex-M3 Base Address
<a href="#">CORTEXM3_RW_PID1</a>	RW	32	0x0000 0000	0xE00F E000
<a href="#">CORTEXM3_RW_PID2</a>	RW	32	0x0000 0004	0xE00F E004
RESERVED	R	32	0x0000 0008	0xE00F E008

### 7.4.8.2 Cortex-M3 RW Table Register Description

**Table 7-23. CORTEXM3\_RW\_PID1**

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See <a href="#">Table 7-22</a> .
<b>Description</b>	Peripheral Identification register - allows the user software to differentiate between the two ARM Cortex-M3 processors (two CPUs). The same piece of code running on the two CPUs can result in different execution (in example branch to different location) depending on what address is stored in the register. The address is stored by the BIOS code. The register cannot be accessed when the BIOS code is running (used).
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADD1																															

Bits	Field Name	Description	Type	Reset
31:0	BASEADD1	ROM memory address	RW	0x0000 0000

**Table 7-24. Register Call Summary for Register CORTEXM3\_RW\_PID1**

Dual Cortex-M3 MPU Subsystem Functional Description

- [Private Memory Space: \[0\] \[1\] \[2\]](#)

Dual Cortex-M3 MPU Subsystem Register Manual

- [CM3\\_RW\\_Table Register Summary: \[3\]](#)

**Table 7-25. CORTEXM3\_RW\_PID2**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 7-22</a> .	<b>Instance</b>	CM3_RW_Table
<b>Description</b>	Peripheral Identification register - allows the user software to differentiate between the two ARM Cortex-M3 processors (two CPUs). The same piece of code running on the two CPUs can result in different execution (in example branch to different location) depending on what address is stored in the register. The address is stored by the BIOS code. The register cannot be accessed when the BIOS code is running (used).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADD2																															

Bits	Field Name	Description	Type	Reset
31:0	BASEADD2	ROM memory address	RW	0x0000 0000

**Table 7-26. Register Call Summary for Register CORTEXM3\_RW\_PID2**

Dual Cortex-M3 MPU Subsystem Functional Description

- [Private Memory Space: \[0\] \[1\] \[2\]](#)

Dual Cortex-M3 MPU Subsystem Register Manual

- [CM3\\_RW\\_Table Register Summary: \[3\]](#)

## Imaging Subsystem

This chapter gives a top-level overview of the imaging subsystem (ISS) in the device.

For a full description of SIMCOP module, see [Section 8.4](#), *ISS:Still Image Coprocessor*.

For a full description of ISS interfaces and Interconnects, that is CSI2, CCP2, timing control, burst translation engine, and circular buffer modules, see [Section 8.2](#), *ISS:Interfaces*.

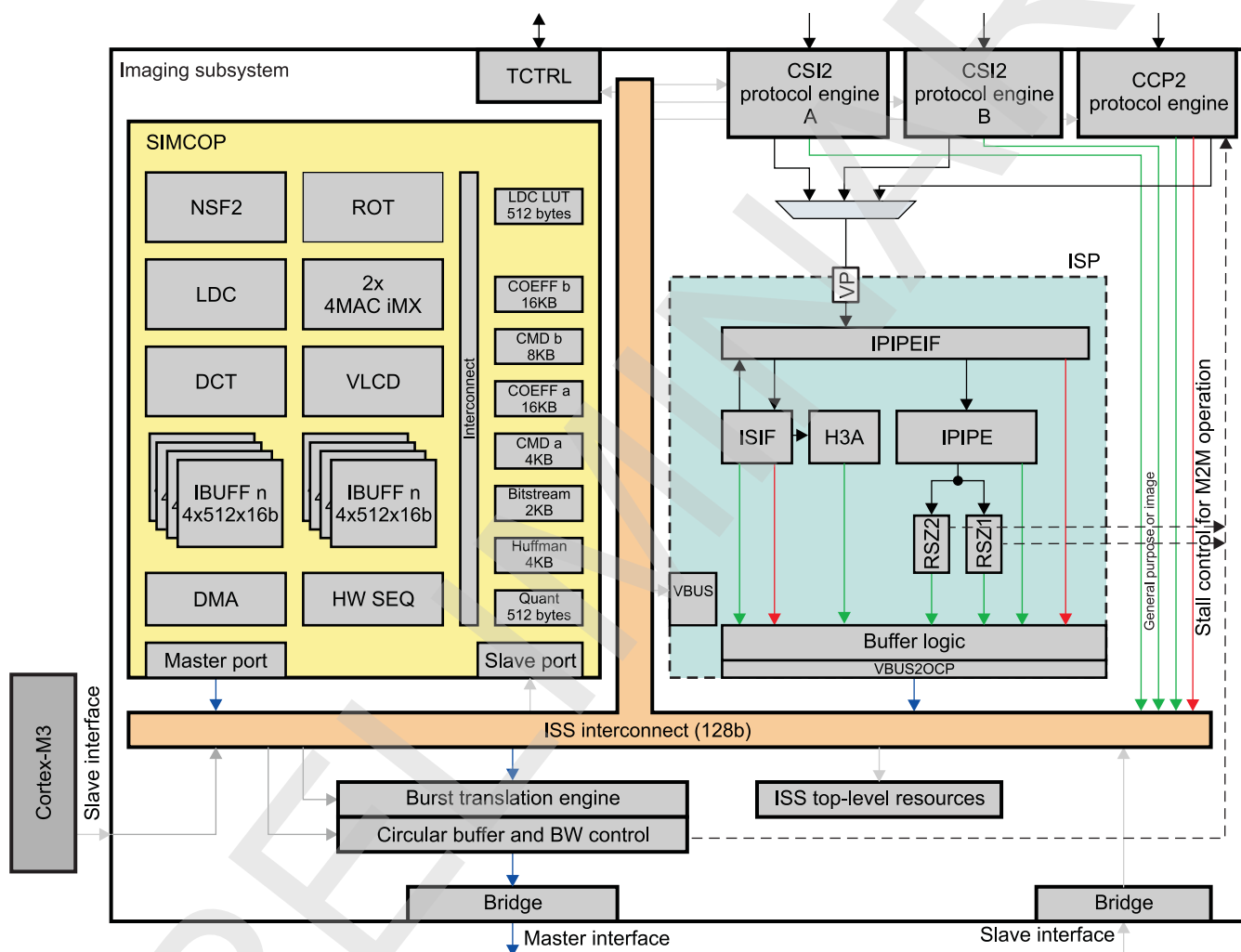
For a full description of ISS ISP, see [Section 8.3](#), *ISS:Image Signal Processor*.

Topic	Page
<b>8.1</b> ISS Overview .....	<b>1120</b>
<b>8.2</b> ISS Interfaces .....	<b>1156</b>
<b>8.3</b> ISS ISP .....	<b>1421</b>
<b>8.4</b> ISS Still Image Coprocessor .....	<b>1875</b>

## 8.1 ISS Overview

The imaging subsystem (ISS) deals with the processing of the pixel data coming from an external image sensor, data from memory (image format encoding and decoding can be done to and from memory), or data from SL2 in IVA-HD for hardware encoding. With its subparts, such as interfaces and interconnects, image signal processor (ISP), and still image coprocessor (SIMCOP), the ISS is a key component in the following multimedia applications: camera viewfinder, video record, and still image capture. Figure 8-1 is an overview of the ISS.

**Figure 8-1. ISS Overview**



camss-351

The direction of the arrows shows the command flow direction: it goes from the master (initiator) to the slave (target). The following color conventions are used for the connections:

- Blue: Bidirectional, 128-bit-wide interface data connection
- Green: Write (ISS -> system memory) data connection. Either 64-bit interface, 128-bit interface, or 32-bit MTC (inside ISP5).
- Red: Read (system memory to ISS) data connection. 128-bit interface port or 32-bit MTC (inside ISP5).
- Gray: 32-bit interface configuration connection
- Solid black: Video port and camera interface related signals
- Dotted black: Data flow stall control signal. Used to slow down ISP for memory-to-memory operation.

the ISS is mainly composed of CCP2 and CSI2-A, CSI2-B camera interfaces, an ISP, and a block-based imaging accelerator (SIMCOP).



The ISS is designed so that it can reach high throughput and low latency with large image sensors. In high-performance mode, the ISS supports a pixel throughput of 200 MPix/s.

Two programmable image processors, iMX4, are included in the SIMCOP subsystem to add further flexibility to implement new algorithms or in case new issues are encountered with the image sensors. The iMX4 processors are also open for third-party algorithms.

The ISS is tightly coupled with a low-interrupt latency microprocessor unit (MPU) subsystem (Cortex™-M3 MPU) that runs a real-time operating system (OS) to reach optimal performance. Mainly, the Cortex-M3 MPU can quickly change the ISS configuration during frame blanking periods and run some sequencing tasks.

The ISS targets the following major use cases:

- Viewfinder with digital zoom, video stabilization, and rotations
- Up to 1080p video record at 30 fps with digital zoom, video stabilization, and rotation
- Up to 16 MPix still image capture with digital zoom and rotation
  - High performance mode: Up to 200 MPix/s throughput
  - High quality and low light modes: Up to 50 MPix/s throughput
- Still image capture during video record

---

**NOTE:** The ISS is not limited to 16 Mpix. Higher resolution can be achieved through multiple passes.

---



---

**NOTE:** For a detailed list of features of a certain submodule, see the related subsection.

---

The ISS offers the following features:

- SIMCOP:
  - Memory-to-memory operation
  - JPEG encode and decode hardware acceleration
  - High-ISO noise filtering (NSF 2.0)
  - Rotation accelerator
  - Warping accelerator
  - Lens distortion correction (YUV space)
  - General-purpose (GP) imaging accelerator (2x iMX4)
  - Direct memory access (DMA) controller
  - Hardware sequencer
- ISP:
  - On-the-fly or memory-to-memory processing
  - Up to 200-MHz pixel throughput
  - Statistic data collection
  - Image pipe interface front-end raw data processing
  - RGB and YUV data processing through ISIF and IPIPE
  - Hardware 3A (H3A) statistics block for real-time auto focus (AF), auto exposure (AE), and auto white balance (AWB)
  - Two image continuous real-time resizers
  - Video port (VP) for interfacing with the receivers and directing data to the ISP
- ISS interfaces:
  - 128-bit-wide data interface to the level 3 (L3) interconnect
  - Burst translation engine (BTE) tightly coupled with the TILER to support efficient rotation
  - Circular buffer for linear space, physically located in memory
  - The ISS relies on the centralized memory management unit (MMU). See [Chapter 20, Memory Management Unit](#).
  - Two CSI2 camera interfaces: CSI2-A (primary) and CSI2-B (secondary)

- CCP2 camera interface (secondary)
- System memory data read-back port (supported by the CCP2 protocol engine module)

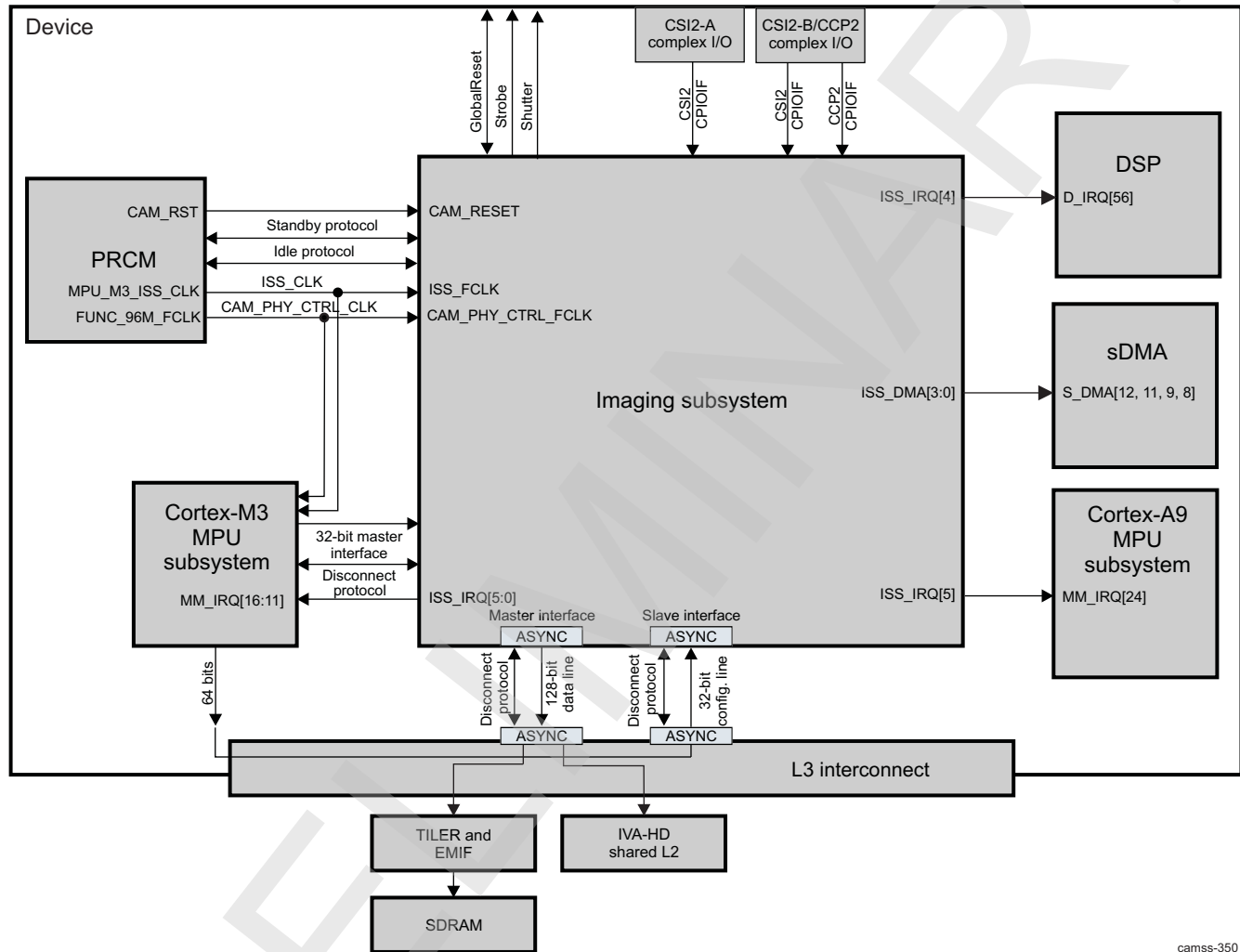
PRELIMINARY

### 8.1.1 ISS Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 8-2 shows the ISS integration.

Figure 8-2. ISS Integration



camss-350

**NOTE:** For more information about the IDLE hardware handshake and wake-up request, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

This section gives an overview of typical uses of the module. For more information about the relationship of the power, reset, and clock management (PRCM) module to the ISS clocks and the reset settings, see the detailed functional description in [Chapter 3, Power, Reset, and Clock Management](#).

The ISS is part of ISS hardware and power management, which comprises the Cortex-M3 MPU, ISS, and a clock generator. These are all independent power domains. The ISS is part of the CAM power domain.

The PRCM module provides a single clock (MPU\_M3\_ISS\_CLK) to ISS power management. This clock generates the clocks for the Cortex-M3 MPU, ISS, and the bridges to L3.

The ISS also supports software reset. A software reset has the same function as a hardware reset except that it does not reset the power-management protocols.

When enabled, MPU\_M3\_ISS\_CLK is gated to provide ISS\_CLK, which, after entering the ISS boundary, is named ISS\_FCLK. MPU\_M3\_ISS\_CLK is provided as long as the Cortex-M3 MPU or ISS modules within the ISS require it.

CAM\_PHY\_CTRL\_CLK is gated from FUNC\_96M\_FCLK, which comes from the PRCM module. When enabled and inside the ISS boundary, it is called CAM\_PHY\_CTRL\_FCLK.

**NOTE:** For a detailed visual representation of the device clocks and how they are handled by the PRCM module before going into the ISS, see the device Clock Tree Tool application and/or [Section 3.6.3.2.4, CM2 Clock Generator](#), in [Chapter 3, Power, Reset, and Clock Management](#).

Table 8-1 through Table 8-3 summarize the integration of the module in the device.

**Table 8-1. ISS Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
ISS	PD_CAM	No	

**Table 8-2. ISS Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
ISS	ISS_FLCK	ISS_CLK	PRCM	ISS global functional clock from the PRCM module
	CAM_PHY_CTRL_FCLK	CAM_PHY_CTRL_CLK	PRCM	Physical layer functional clock from the PRCM module
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
ISS	CAM_RESET	CAM_RST	PRCM	ISS global reset

**Table 8-3. ISS Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
ISS	ISS_IRQ4	D_IRQ_56	DSP INTC	Interrupt generated by ISS to DSP
	ISS_IRQ5	MM_IRQ_24	Cortex-A9 INTC	Interrupt generated by ISS to Cortex-M9
	ISS_IRQ0	MM_IRQ_11	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ1	MM_IRQ_12	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ2	MM_IRQ_13	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ3	MM_IRQ_14	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ4	MM_IRQ_15	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
	ISS_IRQ5	MM_IRQ_16	Cortex-M3 INTC	Interrupt generated by ISS to Cortex-M3
DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
ISS	ISS_DMA0	S_DMA_8	sDMA	Signal connected to the sDMA directly provided by ISP.
	ISS_DMA1	S_DMA_9	sDMA	Signal connected to the sDMA directly provided by ISP
	ISS_DMA2	S_DMA_11	sDMA	Signal connected to the sDMA directly provided by ISP
	ISS_DMA3	S_DMA_12	sDMA	Signal connected to the sDMA directly provided by ISP

### 8.1.1.1 ISS PRCM Interface Integration

#### 8.1.1.1.1 ISS Clock Domains

The ISS has five asynchronous clock domains. Most of the logic uses ISS\_FCLK; the other clock domains are used for interfaces.

Table 8-4 provides a high-level view of clocks. The frequencies provided are maximum values.

**Table 8-4. ISS Local Clock Domains**

Name	Description
PCLK	The CSI2-A, CSI2-B, and CCP2 modules have video port outputs, each having its own pixel clock. These pixel clocks are generated from the functional clock (ISS_FCLK). All three pixel clock sources are multiplexed into one clock provided to the ISP.
CSI2_A_BC	Byte clock provided by the CSI2-A complex I/O. Used by the CSI2-A receiver.
CSI2_B_BC	Byte clock provided by the CSI2-B/CCP2 complex I/O. Used by the CSI2-B receiver.
CCP2_BC	Clock provided by the CCP2 complex I/O. Used by the CCP2 protocol engine.
ISS_FCLK	Functional clock provided by ISS_CLK from the PRCM module. It is used by all ISS submodules and ISS top-level resources.
CAM_PHY_CTRL_FCLK	Functional clock provided by the PRCM module. It is used by the CSI2-A and CSI2-B/CCP2 complex I/Os.

To save power, the ISS can divide the received functional clock (ISS\_CLK) by 2 or 4 from ISS\_CTRL[4:5] ISS\_CLK\_DIV. The configuration clock is always half the functional clock.

The functional clock of some submodules can be cut by software to reduce power consumption by cutting off or turning on the modules from the ISS\_CLKCTRL register. Also, the pixel clocks sent by submodules to ISP can be cut off from ISS\_CLKCTRL VPORT<sub>x</sub>\_CLK.

## 8.1.2 ISS Functional Description

This section provides only a top-level overview of the ISS. The ISS submodules are described in: [Section 8.4, ISS:SIMCOP](#), [Section 8.2, ISS:interfaces](#), and [Section 8.3, ISS:ISP](#).

The following section describes the ISS power-management mechanisms and gives an introduction to and a functional description of the ISP submodules.

### 8.1.2.1 ISS Interrupts

[Table 8-5](#) lists the events generated by the submodules and the top level of the ISS.

Each event that generates an interrupt can be individually enabled by setting the appropriate bit in the [ISS\\_HL\\_IRQENABLE\\_SET\\_i](#) register. The interrupt is disabled by setting the appropriate bit in the [ISS\\_HL\\_IRQENABLE\\_CLR\\_i](#) register.

When an event occurs, the corresponding bit in the [ISS\\_HL\\_IRQSTATUS\\_RAW\\_i](#) register is set regardless of whether or not the event is enabled. Bits in the [ISS\\_HL\\_IRQSTATUS\\_i](#) registers are set only when an enabled event occurs.

Software can clear a pending HS\_VS\_IRQ event by setting the [ISS\\_HL\\_IRQSTATUS\\_i\[17\]](#) HS\_VS\_IRQ bit. Events generated by submodules are automatically cleared at the ISS level when they are cleared at the submodule level.

#### 8.1.2.1.1 ISS Interrupt Merger

The ISS merges the following eight interrupt sources into six physical interrupt lines. All six lines support level and pulse modes.

**Table 8-5. ISS Interrupts**

Interrupt	Source	Description
ISP_IRQ[3:0]	ISP	Interrupt generated by the ISP <sup>(1)</sup>
CSIA_IRQ	CSI2-A	Interrupt generated by the CSI2-A receiver <sup>(1)</sup>
CSIB_IRQ	CSI2-B	Interrupt generated by the CSI2-B receiver <sup>(1)</sup>
CCP2_IRQ[3:0]	CCP2	Interrupt generated by the CCP2 receiver <sup>(1)</sup>
CCP2_IRQ[8]	CCP2	Interrupt generated by the CCP2 receiver <sup>(1)</sup>
CBUFF_IRQ	CBUFF	Interrupt generated by the circular buffer <sup>(1)</sup>
BTE_IRQ	BTE	Interrupt generated by the BTE <sup>(1)</sup>
SIMCOP_IRQ[3:0]	SIMCOP	Interrupt generated by SIMCOP
HS_VS_IRQ	ISS	HS or VS synchronization event. This event is triggered if a rising or falling edge is detected on the HS or VS signal. The rising or falling edge and the HS or VS signal selection are chosen using the ISP_CTRL[0:1] SYNC_DETECT bit field.

<sup>(1)</sup> For more information, see [Section 8.1.2.1.2, ISS Submodule Interrupts](#).

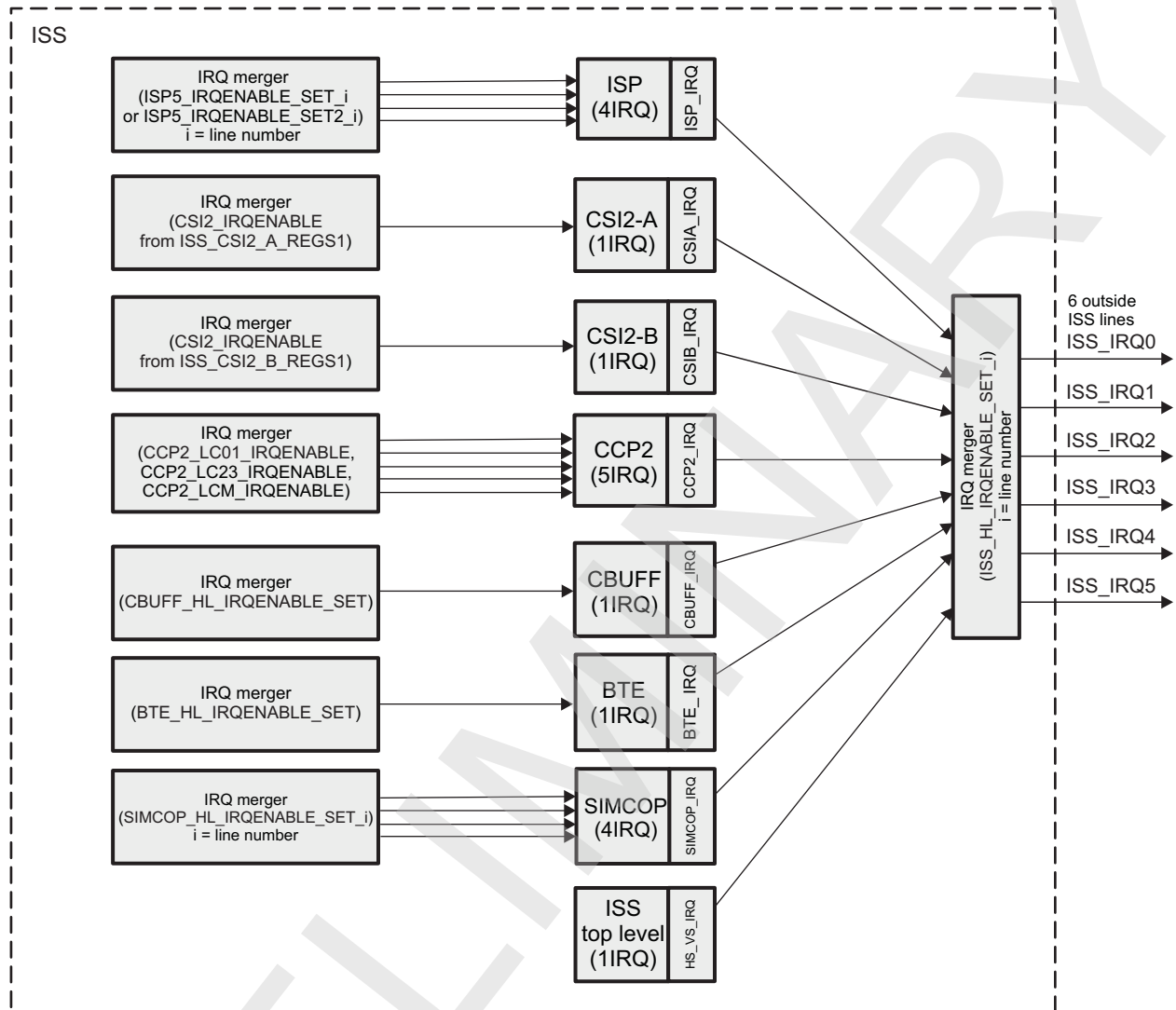
Software can select which interrupt sources are routed to each output line (ISS\_IRQ[0:5]). All six physical interrupt outputs have equivalent functions. Software can use different interrupt lines to group events together by type and therefore reduce interrupt latencies. Typically, one interrupt line is used for low-priority events (errors), and the other interrupt lines are used for high-priority events (for example, SIMCOP sequencing and end-of-frame events to trigger configuration load). The ISS internal interrupt request (IRQ) merger (see [Figure 8-3](#)) relies only on level interrupts provided by submodules. Pulse interrupts provided by submodules are ignored. Typically, all of the interrupts are routed to the Cortex-M3 running the camera driver. In addition, one interrupt is routed to Cortex-A9, which may run the imaging software. Software does not need to clear events from submodules at the ISS level: it clears only the events at the submodule level; the IRQ merger automatically clears the IRQ at this level.

---

**NOTE:** Only the HS\_VS\_IRQ top-level event is cleared at the ISS level.

---

Figure 8-3. ISS Interrupt Merger



camss-702u

**NOTE:** For more information about mapping the six lines to the outside ISS device modules, see [Table 8-3](#).

Only the ISP, SIMCOP, and HL ISS merger IRQ lines are configurable. Each interrupt can be mapped to the required line. This is not the case with CCP2 where, for example, LC01 is hardwired to line0 and line1.

### 8.1.2.1.2 ISS Submodule Interrupts

The ISS shown in [Figure 8-3](#) and listed in [Table 8-3](#) can generate six interrupts.

#### 8.1.2.1.2.1 ISS ISP Interrupts

[Table 8-6](#) summarizes events that cause ISP interrupts.

**Table 8-6. ISS ISP Interrupts**

Event and Register <sup>(1)</sup>	Description
ISP5_IRQENABLE_SET_i[[31] OCP_ERR_IRQ	An interface port error has been received on the ISP master port.
ISP5_IRQENABLE_SET_i[29] IPIPE_INT_DPC_RNEW1	HD interrupt signal to indicate the need to renew the defect pixel correction (DPC) table with new entries. The second 128 entries in the DPC table must be updated when this event triggers. This event is triggered when the 255th entry in the look-up table (LUT) is used. This interrupt is not synchronous to the HD signal.
ISP5_IRQENABLE_SET_i[28] IPIPE_INT_DPC_RNEW0	VD interrupt signal to indicate the need to renew the DPC table with new entries. The first 128 entries in the DPC table must be updated when the event triggers. This event is triggered when the 127th entry in the LUT is used. This interrupt is not synchronous to the HD signal.
ISP5_IRQENABLE_SET_i[27] IPIPE_INT_DPC_INI	Interrupt to signal the need to initialize the DPC table. The DPC table contains two tables of 128 entries. When this signal is used, software must ensure that the 256 table entries are updated with the DPC information.
ISP5_IRQENABLE_SET_i[25] IPIPE_INT_EOF	End of frame interrupt signal
ISP5_IRQENABLE_SET_i[24] H3A_INT_EOF	End of frame interrupt signal
ISP5_IRQENABLE_SET_i[22] RSZ_INT_EOF1	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[23] RSZ_INT_EOF0	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[19] RSZ_FIFO_IN_BLK_ERR	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[18] RSZ_FIFO_IN_OVF	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[17] RSZ_INT_CYC_RZB	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[16] RSZ_INT_CYC_RZA	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[15] RSZ_INT_DMA	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[14] RSZ_INT_LAST_PIX	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[13] RSZ_INT_REG	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[12] H3A_INT	Interrupt generated by the AF and AE/AWB blocks inside the H3A module. It indicates the end of processing a frame and is active high for one configuration bus clock cycle.
ISP5_IRQENABLE_SET_i[11] AF_INT	AF inside generates an interrupt at the end of processing frame; a third interrupt is generated at the same time as the last process to finish.
ISP5_IRQENABLE_SET_i[10] AEW_INT	AEW inside generates an interrupt at the end of processing frame; a third interrupt is generated at the same time as the last process to finish.
ISP5_IRQENABLE_SET_i[9] IPIPEIF_IRQ	IPIPEIF module interrupt is generated at the start position of a frame and is active high for one configuration bus clock cycle.
ISP5_IRQENABLE_SET_i[8] IPIPE_INT_HST	IPIPE module interrupt is generated when histogram is done.
ISP5_IRQENABLE_SET_i[7] IPIPE_INT_BSC	IPIPE module interrupt is generated when boundary signal calculation is done.
ISP5_IRQENABLE_SET_i[6] IPIPE_INT_DMA	IPIPE module interrupt is issued when the SDRAM transfer of boxcar is done. At this time, IPIPE EOF is sent to buffer logic.
ISP5_IRQENABLE_SET_i[5] IPIPE_INT_LAST_PIX	IPIPE module interrupt is issued when the last pixel of a frame comes into IPIPE.
ISP5_IRQENABLE_SET_i[4] IPIPE_INT_REG	IPIPE module interrupt is issued when the register update of the module is allowed.
ISP5_IRQENABLE_SET_i[3] ISIF_INT_3	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[2] ISIF_INT_2	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[1] ISIF_INT_1	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET_i[0] ISIF_INT_0	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET2_i[0] H3A_OVF	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET2_i[1] IPIPEIF_UDF	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET2_i[2] IPIPE_BOXCAR_OVF	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET2_i[3] ISIF_OVF	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .
ISP5_IRQENABLE_SET2_i[4] IPIPE_HST_ERR	See <a href="#">Section 8.1.3</a> and <a href="#">Section 8.3.5</a> .

<sup>(1)</sup> i = 0 to 3



### 8.1.2.1.2.2 ISS CSI2-A and CSI2-B Complex I/O Interrupts

Table 8-7 lists the event generation of the CSI2-A and CSI2-B receivers through the CSI2 interrupt status and interrupt enable registers. The events are checked for status using the CSI2\_IRQSTATUS and CSI2\_IRQENABLE registers.

**Table 8-7. ISS CSI2-A and CSI2-B Interrupts**

Event and Register	Description
CSI2_IRQENABLE[14] OCP_ERR_IRQ	Interface port error
CSI2_IRQENABLE[13] SHORT_PACKET_IRQ	Short packet reception (other than sync events: line start, line end, frame start, and frame end; only data types from 0x8 to 0xF are considered)
CSI2_IRQENABLE[12] ECC_CORRECTION_IRQ	ECC was used to correct a 1-bit error (short packet only).
CSI2_IRQENABLE[11] ECC_NO_CORRECTION_IRQ	ECC was not used to correct the header because the error is larger than 1 bit (short and long packets).
CSI2_IRQENABLE[9] COMPLEXIO_ERR_IRQ	Error signaling from complex I/O: This interrupt is triggered when any error is received from the complex I/O (events are defined in CSI2_COMPLEXIO_IRQSTATUS [see Table 8-8]).
CSI2_IRQENABLE[8] FIFO_OVF_IRQ	FIFO overflow error: This interrupt is triggered when a FIFO overflow is detected. An overflow can occur if there is a mismatch between the data input and output rates. A reset of the module is required to restart correctly.
CSI2_IRQENABLE[7] CONTEXT7	At least one interrupt event enabled from Context 7 occurred (see Table 8-9).
CSI2_IRQENABLE[6] CONTEXT6	At least one interrupt event enabled from Context 6 occurred (see Table 8-9).
CSI2_IRQENABLE[5] CONTEXT5	At least one interrupt event enabled from Context 5 occurred (see Table 8-9).
CSI2_IRQENABLE[4] CONTEXT4	At least one interrupt event enabled from Context 4 occurred (see Table 8-9).
CSI2_IRQENABLE[3] CONTEXT3	At least one interrupt event enabled from Context 3 occurred (see Table 8-9).
CSI2_IRQENABLE[2] CONTEXT2	At least one interrupt event enabled from Context 2 occurred (see Table 8-9).
CSI2_IRQENABLE[1] CONTEXT1	At least one interrupt event enabled from Context 1 occurred (see Table 8-9).
CSI2_IRQENABLE[0] CONTEXT0	At least one interrupt event enabled from Context 0 occurred (see Table 8-9).

Table 8-8 lists CSI2 receiver event generation through the CSI2-A/CSI2-B complex I/O interrupt status and interrupt enable registers. The events are checked and controlled from the CSI2\_COMPLEXIO\_IRQSTATUS and CSI2\_COMPLEXIO\_IRQENABLE registers.

**Table 8-8. ISS CSI2-A and CSI2-B Receivers Complex I/O Interrupts**

Event and Register	Description
CSI2_COMPLEXIO_IRQENABLE[0] ERRSOTHS1	Start of transmission error for lane 1
CSI2_COMPLEXIO_IRQENABLE[1] ERRSOTHS2	Start of transmission error for lane 2
CSI2_COMPLEXIO_IRQENABLE[2] ERRSOTHS3	Start of transmission error for lane 3
CSI2_COMPLEXIO_IRQENABLE[5] ERRSOTSYNCHS1	Start of transmission sync error for lane 1
CSI2_COMPLEXIO_IRQENABLE[6] ERRSOTSYNCHS2	Start of transmission sync error for lane 2
CSI2_COMPLEXIO_IRQENABLE[7] ERRSOTSYNCHS3	Start of transmission sync error for lane 3
CSI2_COMPLEXIO_IRQENABLE[10] ERRESC1	Escape entry error for lane 1
CSI2_COMPLEXIO_IRQENABLE[11] ERRESC2	Escape entry error for lane 2
CSI2_COMPLEXIO_IRQENABLE[12] ERRESC3	Escape entry error for lane 3
CSI2_COMPLEXIO_IRQENABLE[15] ERRCONTROL1	Control error for lane 1
CSI2_COMPLEXIO_IRQENABLE[16] ERRCONTROL2	Control error for lane 2
CSI2_COMPLEXIO_IRQENABLE[17] ERRCONTROL3	Control error for lane 3
CSI2_COMPLEXIO_IRQENABLE[20] STATEULPM1	Lane 1 in ULPM

**Table 8-8. ISS CSI2-A and CSI2-B Receivers Complex I/O Interrupts (continued)**

Event and Register	Description
CSI2_COMPLEXIO_IRQENABLE[21] STATEULPM2	Lane 2 in ULPM
CSI2_COMPLEXIO_IRQENABLE[22] STATEULPM3	Lane 3 in ULPM
CSI2_COMPLEXIO_IRQENABLE[25] STATEALLULPMENTER	All active lanes are entering the ULPM.
CSI2_COMPLEXIO_IRQENABLE[26] STATEALLULPMEXIT	At least one active lane exited the ULPM.

Because the CSI2-A/CSI2-B receivers support eight contexts, the CSI2\_CTX\_IRQSTATUS\_i and CSI2\_CTX\_IRQENABLE\_i registers are present eight times (one time per context).

The events are generated only for the enabled context(s). [Table 8-9](#) describes the generation of the CSI2 receiver event through the CSI2\_CTX\_IRQSTATUS\_i and CSI2\_CTX\_IRQENABLE\_i registers.

**Table 8-9. ISS CSI2-A/CSI2-B Receivers CONTEXT Interrupts**

Event <sup>(1)</sup>	Description
CSI2_CTX_IRQENABLE_i[0] FS_IRQ	Frame start: This interrupt is triggered when a frame-start synchronization code is detected in the CSI2 data stream.
CSI2_CTX_IRQENABLE_i[1] FE_IRQ	Frame end: This interrupt is triggered when a frame-end synchronization code is detected in the CSI2 data stream.
CSI2_CTX_IRQENABLE_i[2] LS_IRQ	Line start: This interrupt is triggered when a line-start synchronization code is detected in the CSI2 data stream.
CSI2_CTX_IRQENABLE_i[3] LE_IRQ	Line end: This interrupt is triggered when a line-end synchronization code is detected in the CSI2 data stream.
CSI2_CTX_IRQENABLE_i[5] CS_IRQ	CS error: This interrupt is triggered when a mismatch between the transmitter and receiver checksums (payload) is detected.
CSI2_CTX_IRQENABLE_i[6] FRAME_NUMBER_IRQ	Frame counter reached: This interrupt is triggered when the frame counter reaches its programmable target value.
CSI2_CTX_IRQENABLE_i[7] LINE_NUMBER_IRQ	Line number reached: The programmable line number is received. The modulo feature can be selected (CSI2_CTX_CTRL1_i.LINE_MODULO). When selected, the interrupt is generated for each line number multiple of the programmed line number (CSI2_CTX_CTRL3_i.LINE_NUMBER); otherwise, the interrupt is generated only for the line number.
CSI2_CTX_IRQENABLE_i[8] ECC_CORRECTION_IRQ	ECC was used to correct a 1-bit error (long packets only).

<sup>(1)</sup> i = 0 to 7

### 8.1.2.1.2.3 ISS CCP2 Interrupts

[Table 8-10](#) summarizes the CCP2 receiver interrupts.

**Table 8-10. ISS CCP2 Receiver Interrupts**

Event and Register	Description
CCP2_LC01_IRQENABLE[31] LC1_OCPERROR_IRQ	CCP2 write master interface port error on logical channel 1
CCP2_LC01_IRQENABLE[27] LC1_FS_IRQ	CCP2 frame-start synchronization code detection on logical channel 1. This interrupt is triggered when a frame-start synchronization code is detected in the CCP2 data stream.
CCP2_LC01_IRQENABLE[26] LC1_LE_IRQ	CCP2 line-end synchronization code detection on logical channel 1. This interrupt is triggered when a line-end synchronization code is detected in the CCP2 data stream.
CCP2_LC01_IRQENABLE[25] LC1_LS_IRQ	CCP2 line-start synchronization code detection on logical channel 1. This interrupt is triggered when a line-start synchronization code is detected in the CCP2 data stream.
CCP2_LC01_IRQENABLE[24] LC1_FE_IRQ	CCP2 frame-end synchronization code detection on logical channel 1. This interrupt is triggered when a frame-end synchronization code is detected in the CCP2 data stream.
CCP2_LC01_IRQENABLE[23] LC1_COUNT_IRQ	CCP2 frame counter reached on logical channel 1. This interrupt is triggered when the frame counter reaches its programmable target value.

**Table 8-10. ISS CCP2 Receiver Interrupts (continued)**

Event and Register	Description
CCP2_LC01_IRQENABLE[21] LC1_FIFO_OVF_IRQ	CCP2 FIFO overflow error for logical channel 1. An overflow can occur if there is a mismatch between the data input and output rates.
CCP2_LC01_IRQENABLE[20] LC1_CRC_IRQ	CCP2 CRC error for logical channel 1. This interrupt is triggered when a mismatch between the transmitter and receiver checksums is detected.
CCP2_LC01_IRQENABLE[19] LC1_FSP_IRQ	CCP2 false-synchronization protection code error for logical channel 1. This interrupt is triggered by the FSP decoder if an illegal combination is detected, but 0xA5 is not present in the bitstream.
CCP2_LC01_IRQENABLE[18] LC1_FW_IRQ	CCP2 frame-width error for logical channel 1. This interrupt is generated if the frame width constraints associated with the current data type are not respected.
CCP2_LC01_IRQENABLE[17] LC1_FSC_IRQ	CCP2 false-synchronization code error for logical channel 1. This interrupt is triggered if the synchronization code order is not respected. This state is shown in the CCP2 receiver finite state-machine (FSM).
CCP2_LC01_IRQENABLE[15] LC0_OCPERROR_IRQ	CCP2 write master interface port error on logical channel 1
CCP2_LC01_IRQENABLE[11] LC0_FS_IRQ	CCP2 frame-start synchronization code detection on logical channel 0. This interrupt is triggered when a frame-start synchronization code is detected in the CCP2 data stream.
CCP2_LC01_IRQENABLE[10] LC0_LE_IRQ	CCP2 line-end synchronization code detection on logical channel 0. This interrupt is triggered when a line-end synchronization code is detected in the CCP2 data stream.
CCP2_LC01_IRQENABLE[9] LC0_LS_IRQ	CCP2 line-start synchronization code detection on logical channel 0. This interrupt is triggered when a line-start synchronization code is detected in the CCP2 data stream.
CCP2_LC01_IRQENABLE[8] LC0_FE_IRQ	CCP2 frame-end synchronization code detection on logical channel 0. This interrupt is triggered when a frame-end synchronization code is detected in the CCP2 data stream.
CCP2_LC01_IRQENABLE[7] LC0_COUNT_IRQ	CCP2 frame counter reached on logical channel 0. This interrupt is triggered when the frame counter reaches its programmable target value.
CCP2_LC01_IRQENABLE[5] LC0_FIFO_OVF_IRQ	CCP2 FIFO overflow error for logical channel 0. An overflow can occur if there is a mismatch between the data input and output rates.
CCP2_LC01_IRQENABLE[4] LC0_CRC_IRQ	CCP2 CRC error for logical channel 0. This interrupt is triggered when a mismatch between the transmitter and receiver checksums is detected.
CCP2_LC01_IRQENABLE[3] LC0_FSP_IRQ	CCP2 false-synchronization code error for logical channel 0. This interrupt is triggered if the synchronization code order is not respected. This state is shown in the CCP2 receiver FSM.
CCP2_LC01_IRQENABLE[2] LC0_FW_IRQ	CCP2 frame-width error for logical channel 0. This interrupt is generated if the frame-width constraints associated with the current data type are not respected.
CCP2_LC01_IRQENABLE[1] LC0_FSC_IRQ	CCP2 false-synchronization code error for logical channel 0. This interrupt is triggered if the synchronization code order is not respected. This state is shown in the CCP2 receiver FSM.
CCP2_LC23_IRQENABLE[31] LC3_OCPERROR_IRQ	CCP2 write master interface port error on logical channel 3
CCP2_LC23_IRQENABLE[27] LC3_FS_IRQ	CCP2 frame-start synchronization code detection on logical channel 3. This interrupt is triggered when a frame-start synchronization code is detected in the CCP2 data stream.
CCP2_LC23_IRQENABLE[26] LC3_LE_IRQ	CCP2 line-end synchronization code detection on logical channel 3. This interrupt is triggered when a line-end synchronization code is detected in the CCP2 data stream.
CCP2_LC23_IRQENABLE[25] LC3_LS_IRQ	CCP2 line-start synchronization code detection on logical channel 3. This interrupt is triggered when a line-start synchronization code is detected in the CCP2 data stream.
CCP2_LC23_IRQENABLE[24] LC3_FE_IRQ	CCP2 frame-end synchronization code detection on logical channel 3. This interrupt is triggered when a frame-end synchronization code is detected in the CCP2 data stream.
CCP2_LC23_IRQENABLE[23] LC3_COUNT_IRQ	CCP2 frame counter reached on logical channel 3. This interrupt is triggered when the frame counter reaches its programmable target value.
CCP2_LC23_IRQENABLE[21] LC3_FIFO_OVF_IRQ	CCP2 FIFO overflow error for logical channel 3. An overflow can occur if there is a mismatch between the data input and output rates.

**Table 8-10. ISS CCP2 Receiver Interrupts (continued)**

Event and Register	Description
CCP2_LC23_IRQENABLE[20] LC3_CRC_IRQ	CCP2 CRC error for logical channel 3. This interrupt is triggered when a mismatch between the transmitter and receiver checksums is detected.
CCP2_LC23_IRQENABLE[19] LC3_FSP_IRQ	CCP2 false-synchronization code error for logical channel 3. This interrupt is triggered if the synchronization code order is not respected. This state is shown in the CCP2 receiver FSM.
CCP2_LC23_IRQENABLE[18] LC3_FW_IRQ	CCP2 frame-width error for logical channel 3. This interrupt is generated if the frame-width constraints associated with the current data type are not respected.
CCP2_LC23_IRQENABLE[17] LC3_FSC_IRQ	CCP2 false-synchronization code error for logical channel 3. This interrupt is triggered if the synchronization code order is not respected. This state is shown in the CCP2 receiver FSM.
CCP2_LC23_IRQENABLE[11] LC2_FS_IRQ	CCP2 frame-start synchronization code detection on logical channel 2. This interrupt is triggered on the detection of a frame-start synchronization code into the CCP2 data stream.
CCP2_LC23_IRQENABLE[10] LC2_LE_IRQ	CCP2 line-end synchronization code detection on logical channel 2. This interrupt is triggered on the detection of a line-end synchronization code into the CCP2 data stream.
CCP2_LC23_IRQENABLE[9] LC2_LS_IRQ	CCP2 line-start synchronization code detection on logical channel 2. This interrupt is triggered when a line-start synchronization code is detected in the CCP2 data stream.
CCP2_LC23_IRQENABLE[8] LC2_FE_IRQ	CCP2 frame-end synchronization code detection on logical channel 2. This interrupt is triggered when a frame-end synchronization code is detected in the CCP2 data stream.
CCP2_LC23_IRQENABLE[7] LC2_COUNT_IRQ	CCP2 frame counter reached on logical channel 2. This interrupt is triggered when the frame counter reaches its programmable target value.
CCP2_LC23_IRQENABLE[5] LC2_FIFO_OVF_IRQ	CCP2 FIFO overflow error for logical channel 2. An overflow can occur if there is a mismatch between the data input and output rates.
CCP2_LC23_IRQENABLE[4] LC2_CRC_IRQ	CCP2 CRC error for logical channel 2. This interrupt is triggered when a mismatch is detected between the transmitter and receiver checksums.
CCP2_LC23_IRQENABLE[3] LC2_FSP_IRQ	CCP2 false-synchronization code error for logical channel 2. This interrupt is triggered if the synchronization code order is not respected. This state is shown in the CCP2 receiver FSM.
CCP2_LC23_IRQENABLE[2] LC2_FW_IRQ	CCP2 frame-width error for logical channel 2. This interrupt is generated if the frame-width constraints associated with the current data type are not respected.
CCP2_LC23_IRQENABLE[1] LC2_FSC_IRQ	CCP2 false-synchronization code error for logical channel 2. This interrupt is triggered if the synchronization code order is not respected. This state is shown in the CCP2 receiver FSM.
CCP2_LCM_IRQENABLE[1] LCM_OCPERROR	CCP2 an interface error occurred on the master read port. This interrupt is triggered when an OCP error is detected on the master read port.
CCP2_LCM_IRQENABLE[0] LCM_EOF	Memory read channel – end of frame: This interrupt is triggered when a frame is read completely from memory.

#### 8.1.2.1.2.4 ISS CBUFF Interrupts

Table 8-11 summarizes the CBUFF interrupts.

**Table 8-11. ISS CBUFF Interrupts**

Event and Register	Description
CBUFF_HL_IRQENABLE_SET[31] IRQ_CTX7_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[30] IRQ_CTX6_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[29] IRQ_CTX5_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[28] IRQ_CTX4_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[27] IRQ_CTX3_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[26] IRQ_CTX2_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[25] IRQ_CTX1_OVR	CBUFF overflow

**Table 8-11. ISS CBUFF Interrupts (continued)**

Event and Register	Description
CBUFF_HL_IRQENABLE_SET[24] IRQ_CTX0_OVR	CBUFF overflow
CBUFF_HL_IRQENABLE_SET[23] IRQ_CTX7_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[22] IRQ_CTX6_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[21] IRQ_CTX5_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[20] IRQ_CTX4_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[19] IRQ_CTX3_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[18] IRQ_CTX2_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[17] IRQ_CTX1_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[16] IRQ_CTX0_INVALID	CBUFF invalid access
CBUFF_HL_IRQENABLE_SET[15] IRQ_CTX7_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[14] IRQ_CTX6_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[13] IRQ_CTX5_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[12] IRQ_CTX4_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[11] IRQ_CTX3_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[10] IRQ_CTX2_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[9] IRQ_CTX1_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[8] IRQ_CTX0_READY	CBUFF WB physical window ready for access by the CPU
CBUFF_HL_IRQENABLE_SET[0] IRQ_OCP_ERR	CBUFF master interface port error

#### 8.1.2.1.2.5 ISS BTE Interrupts

Table 8-12 summarizes the BTE interrupts.

**Table 8-12. ISS BTE Interrupts**

Event and Register	Description
BTE_HL_IRQENABLE_SET[31] IRQ_CTX7_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 7 is received, but not enough frame lines have been prefetched in the buffer. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[30] IRQ_CTX6_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 6 is received, but not enough frame lines have been prefetched in the buffer. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[29] IRQ_CTX5_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 5 is received, but not enough frame lines have been prefetched in the buffer. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[28] IRQ_CTX4_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 4 is received, but not enough frame lines have been prefetched in the buffer. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[27] IRQ_CTX3_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 3 is received, but not enough frame lines have been prefetched in the buffer. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[26] IRQ_CTX2_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 2 is received, but not enough frame lines have been prefetched in the buffer. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .

**Table 8-12. ISS BTE Interrupts (continued)**

Event and Register	Description
BTE_HL_IRQENABLE_SET[25] IRQ_CTX1_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 1 is received, but not enough frame lines have been prefetched in the buffer. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[24] IRQ_CTX0_ERR	Writes enable notification for read request before enough data is prefetched. Reads notify when a read request that requires translation on Context 0 is received, but not enough frame lines have been prefetched in the buffer. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[23] IRQ_CTX7_INVALID	Writes enable invalid access to Context 7. Reads notify when access to an unexpected location in Context 7 is requested, or the start context location access is valid, but the burst length exceeds the Context 7 end. See <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
BTE_HL_IRQENABLE_SET[22] IRQ_CTX6_INVALID	Writes enable invalid access to Context 6. Reads notify when access to an unexpected location in Context 6 is requested, or the start context location access is valid, but the burst length exceeds the Context 6 end. See <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
BTE_HL_IRQENABLE_SET[21] IRQ_CTX5_INVALID	Writes enable invalid access to Context 5. Reads notify when access to an unexpected location in Context 5 is requested, or the start context location access is valid, but the burst length exceeds the Context 5 end. See <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
BTE_HL_IRQENABLE_SET[20] IRQ_CTX4_INVALID	Writes enable invalid access to Context 4. Reads notify when access to an unexpected location in Context 4 is requested, or the start context location access is valid, but the burst length exceeds the Context 4 end. See <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
BTE_HL_IRQENABLE_SET[19] IRQ_CTX3_INVALID	Writes enable invalid access to Context 3. Reads notify when access to an unexpected location in Context 3 is requested, or the start context location access is valid, but the burst length exceeds the Context 3 end. See <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
BTE_HL_IRQENABLE_SET[18] IRQ_CTX2_INVALID	Writes enable invalid access to Context 2. Reads notify when access to an unexpected location in Context 2 is requested, or the start context location access is valid, but the burst length exceeds the Context 2 end. See <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
BTE_HL_IRQENABLE_SET[17] IRQ_CTX1_INVALID	Writes enable invalid access to Context 1. Reads notify when access to an unexpected location in Context 1 is requested, or the start context location access is valid, but the burst length exceeds the Context 1 end. See <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
BTE_HL_IRQENABLE_SET[16] IRQ_CTX0_INVALID	Writes enable invalid access to Context 0. Reads notify when access to an unexpected location in Context 0 is requested, or the start context location access is valid, but the burst length exceeds the Context 0 end. See <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
BTE_HL_IRQENABLE_SET[15] IRQ_CTX7_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 7 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[14] IRQ_CTX6_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 6 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .

**Table 8-12. ISS BTE Interrupts (continued)**

Event and Register	Description
BTE_HL_IRQENABLE_SET[13] IRQ_CTX5_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 5 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[12] IRQ_CTX4_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 4 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[11] IRQ_CTX3_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 3 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[10] IRQ_CTX2_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 2 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[9] IRQ_CTX1_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 1 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[8] IRQ_CTX0_DONE	Writes enable notification for context that has been fully and successfully transferred to TILER. Reads notify when enough frame lines have been prefetched into the buffer and Context 0 is translated to TILER. This interrupt is triggered when flushing completes (if enabled) in one-shot mode. It is triggered once per frame in continuous mode. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .
BTE_HL_IRQENABLE_SET[1] IRQ_INVALID	Writes enable Invalid virtual space access notification. Reads notify when access falls into a translated from the BTE region, but it is 2D access or it does not map to an active context. See <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Virtual Address Space and Context Mapping</a> .
BTE_HL_IRQENABLE_SET[1] IRQ_OCP_ERR	Writes enable notification for error on the master output interface. Reads notify when an error has occurred on the master output interface.

#### 8.1.2.1.2.6 ISS SIMCOP Interrupts

[Table 8-13](#) summarizes the SIMCOP high-level interrupts mapped to the outer boundaries of the SIMCOP. For more information about interrupts generated from inside the SIMCOP modules, see the SIMCOP section.

**Table 8-13. ISS SIMCOP High-Level Interrupts**

Event and Register	Description
SIMCOP_HL_IRQENABLE_SET_i[19] CPU_PROC_START_IRQ	Interrupt used when CPU data processing is used in a macroblock processing pipeline. When the CPU receives this IRQ, data is ready to be processed. When the CPU finishes processing the data, it acknowledges by setting the SIMCOP_HWSEQ_CTRL.CPU_PROC_DONE bit.
SIMCOP_HL_IRQENABLE_SET_i[18] SIMCOP_DMA_IRQ1	Interrupt triggered by SIMCOP DMA
SIMCOP_HL_IRQENABLE_SET_i[16] OCP_ERR_IRQ	SIMCOP master port interface error
SIMCOP_HL_IRQENABLE_SET_i[15] VLCDJ_DECODE_ERR_IRQ	A decode error has been signaled by the VLCDJ module.
SIMCOP_HL_IRQENABLE_SET_i[14] DONE_IRQ	Event triggered when hardware sequencer finishes the sequence: <ul style="list-style-type: none"> <li>The sequence step counter has reached the limit.</li> <li>All accelerator and DMA events for the last sequence step have been received.</li> </ul>
SIMCOP_HL_IRQENABLE_SET_i[13] STEP3_IRQ	Event triggered when a SIMCOP Context 3 is activated by the hardware sequencer.
SIMCOP_HL_IRQENABLE_SET_i[12] STEP2_IRQ	Event triggered when a SIMCOP Context 2 is activated by the hardware sequencer.
SIMCOP_HL_IRQENABLE_SET_i[11] STEP1_IRQ	Event triggered when a SIMCOP Context 1 is activated by the hardware sequencer.
SIMCOP_HL_IRQENABLE_SET_i[10] STEP0_IRQ	Event triggered when a SIMCOP Context 0 is activated by the hardware sequencer.
SIMCOP_HL_IRQENABLE_SET_i[9] LDC_BLOCK_IRQ	A macroblock has been processed.
SIMCOP_HL_IRQENABLE_SET_i[7] ROT_A	Rotational engine interrupt
SIMCOP_HL_IRQENABLE_SET_i[6] IMX_B_IRQ	Event triggered when iMX has executed a SLEEP instruction
SIMCOP_HL_IRQENABLE_SET_i[5] IMX_A_IRQ	Event triggered when iMX has executed a SLEEP instruction
SIMCOP_HL_IRQENABLE_SET_i[4] NSF_IRQ	Event triggered by the NSD2 imaging accelerator when processing of a block is done
SIMCOP_HL_IRQENABLE_SET_i[3] VLCDJ_BLOC_IRQ	A macroblock has been processed (that is, encode and decode).
SIMCOP_HL_IRQENABLE_SET_i[2] DCT_IRQ	DCT operating is complete (configured number of MCUs for YUV4:2:0/4:2:2 mode, or number of blocks for sequential block mode).
SIMCOP_HL_IRQENABLE_SET_i[1] LDC_FRAME_IRQ	A full frame has been processed.
SIMCOP_HL_IRQENABLE_SET_i[0] SIMCOP_DMA_IRQ0	Interrupt triggered by SIMCOP DMA

### 8.1.2.2 ISS Clocks

The clocks of ISS submodules can be cut individually using the [ISS\\_CLKCTRL](#) register. Software can poll the module status reading the appropriate bit in the [ISS\\_CLKSTAT](#) register.

When software wants to enable a submodule:

- Software sets the appropriate bit in the [ISS\\_CLKCTRL](#) register.
- Hardware enables the submodule functional and interface clocks (expected to take a few cycles).
- Hardware sets the appropriate bit in the [ISS\\_CLKSTAT](#) register.

Software must enable the modules in the correct order. The hardware imposes no particular constraint. For example, when data must be provided by the CSI2-A/CSI2-B receiver and processed by the ISP, both modules must be enabled and correctly configured before data arrives. An example of configuration order is: enabling the CSI2-A receiver powers up the complex I/O connected to the external sensor. Additionally, the ISP must be configured and the source interface must be selected. For details and the order of configuration, see the programming module of the particular submodule.

When software wants to shut down a submodule:

- Software ensures that the submodule is idle. Mainly:
  - The submodule must not generate new events.
  - The submodule must not have any pending events.
  - For initiators: The submodule must stop the generation of an interface bridge transaction.



- Software clears the appropriate bit in the [ISS\\_CLKCTRL](#) register.
- For modules having only a master port: Hardware waits until the submodule to be disconnected asserts the MStandBy signal on its master port. It asserts MWait of the submodule.

**NOTE:** The ISS does not assert the MWait signal when a shutdown of the module is not requested by software.

- Hardware cuts the submodule clocks.
- Hardware clears the appropriate bit in the [ISS\\_CLKSTAT](#) register.

[Table 8-14](#) describes the clock gating of the ISS submodule.

**Table 8-14. ISS Submodule Clock Gating**

ISS Resource	Feature On/Off Control
ISS top-level resources	Not applicable. ISS top-level resources cannot be cut. However, top-level resources support the autogating feature.
SIMCOP	<a href="#">ISS_CLKCTRL[0]</a> SIMCOP
ISP	<a href="#">ISS_CLKCTRL[1]</a> ISP
CSI2-A	<a href="#">ISS_CLKCTRL[2]</a> CSI2_A
CSI2-B	<a href="#">ISS_CLKCTRL[3]</a> CSI2_B
CCP2	<a href="#">ISS_CLKCTRL[4]</a> CCP2
ISS interconnect BTE CBUFF TCTRL	These modules cannot be switched off individually. They are required for any processing performed by SIMCOP because they are on the main data path. However, they support autogating to reduce power consumption when activity is low.

When the clock of a submodule is cut and an interface bridge request for this module is received from the ISS configuration interconnect, the ISS clock manager temporarily enables the module clock to handle the access properly.

All ISS submodules are off after reset; software must enable them before they can be used.

### 8.1.2.3 ISS Reset

The ISS can accept a general software reset, propagated through all the hierarchy. This reset can be done to initialize the module and has the same effect as the hardware reset.

1. Set the [ISS\\_HL\\_SYSCONFIG\[0\]](#) SOFTRESET bit to 1.
2. Read the [ISS\\_HL\\_SYSCONFIG\[0\]](#) SOFTRESET bit to check whether it equals 0, which means the reset occurred.

If after five reads, [ISS\\_HL\\_SYSCONFIG\[0\]](#) SOFTRESET still returns 1, it can be assumed that an error occurred during the reset stage.

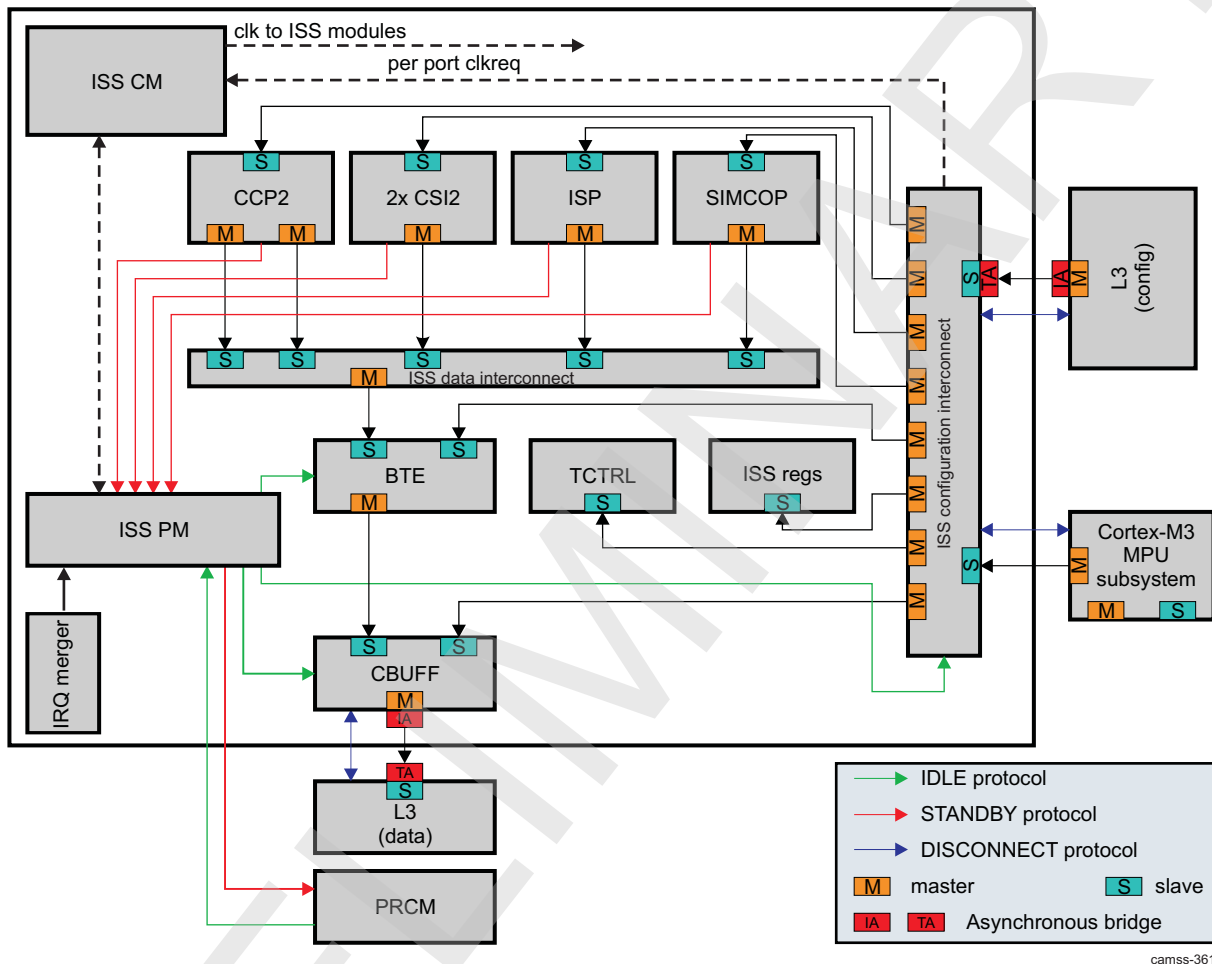
A software reset must not reset the power manager protocols (must not reset the IDLE and STANDBY generic IPs).

### 8.1.2.4 ISS Power Management

#### 8.1.2.4.1 ISS Power-Management Infrastructure Overview

Figure 8-4 is an overview of the ISS power management.

**Figure 8-4. ISS Power Management**



**NOTE:** For power savings, the PRCM module can request idle mode from the ISS. When the ISS is not functional, software must decide when the PRCM module can send the request. For more information, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

#### 8.1.2.4.2 ISS STANDBY Mechanism

The power manager receives STANDBY information from the CCP2, CSI2-A, CSI2-B, ISP, and SIMCOP modules. These modules assert a standby signal when they have no more transactions to perform. The ISS power manager acknowledges by asserting a wait signal.

When those five modules are in standby mode, the ISS power manager initiates a STANDBY sequence for the ISS:

1. The ISS waits while the CCP2, CSI2-A, CSI2-B, ISP, and SIMCOP MStandby is asserted.
2. The ISS power manager acknowledges by asserting the CCP2, CSI2-A, CSI2-B, ISP, and SIMCOP MWait signal.
3. The ISS initiators assert only the MStandBy signal when they receive responses to all sent requests.

Therefore, when all initiators have asserted MStandBy, the ISS interconnect has no more pending traffic (although configurable, only nonposted writes must be used for error reporting).

4. The ISS power manager sends an IDLE request to the BTE. This IDLE request is used to drain BTE data. It does not affect the configuration port.
5. The BTE drains all transactions.
6. The BTE acknowledges the IDLE request.
7. The ISS power manager sends an IDLE request to the CBUFF. This IDLE request is used to drain CBUFF data. It does not affect the configuration port.
8. CBUFF drains all transactions.
9. CBUFF disconnects the interface master port connected to the L3 interconnect.
10. CBUFF acknowledges the IDLE request.
11. The ISS power manager asserts the MStandBy signals connected to the system PRCM module.
12. The PRCM module acknowledges by asserting the MWait signal.

A functional standby transition can be aborted when one of the CCP2, CSI2-A, CSI2-B, ISP, or SIMCOP modules deasserts the MStandBy signal. The corresponding MWait signal is deasserted only when the ISS interconnect, BTE, CBUFF, and ISS interface master port are ready to receive requests.

When one of the CCP2, CSI2-A, CSI2-B, ISP, or SIMCOP modules must perform accesses to the ISS interface master port, it deasserts the MStandBy signal. The ISS power manager executes the following sequence to leave the STANDBY state:

1. The ISS power manager deasserts the MStandby signal.
2. The ISS waits until the PRCM module deasserts the MWait signal.
3. The ISS power manager requests CBUFF to go into functional mode.
4. CBUFF connects the interface port.
5. The ISS power manager requests the BTE to go into functional mode.
6. The ISS power manager waits until CBUFF and BTE acknowledge functional mode.
7. The ISS deasserts the MWait signal of the module requesting access.

Abort of the standby-to-functional mode transition is not supported. The ISS power manager completes the standby-to-functional transition and then allows a new functional-to-standby transition.

Typically, the MStandby signal is used for two purposes. Software chooses one of the following behaviors through PRCM configuration:

- During blanking periods: The ISS asserts the MStandBy signal between frames when it has no more data to send. The PRCM module can use this information to switch off the L3 interconnect and save some dynamic power. However, the PRCM module is not allowed to cut the ISS functional clock in that case, because it is needed to receive the next frame.
- For ISS shutdown: The ISS asserts MStandBy when it has no more transactions to perform. The PRCM module then initiates an IDLE sequence. Once the ISS acknowledges the transition into idle mode, the PRCM module can cut the ISS clock and power.

The internal standby mode can be reached only when CCP2, CSI2-A, CSI2-B, SIMCOP, ISP, ISS data interconnect, BTE, and CBUFF are in IDLE or STANDBY state. Choosing no-idle or no-standby mode for any of these modules prevents the ISS from going into STANDBY state.

Four modes for standby control are supported:

- Smart-standby-wakeup mode: This is the mode normally used. When in this mode, the ISS asserts the MStandBy signal when the MStandBy of all ISS internal initiators is asserted and the ISS data interconnect, BTE, and CBUFF are in IDLE state.
- Smart-standby mode: The ISS has no wake-up event. This mode is equivalent to smart-standby-wakeup mode.
- Force-standby mode: This is a backup mode intended to be used only if smart-standby mode is bugged. When in this mode, the ISS asserts MStandBy unconditionally. Software must ensure that the ISS is in a correct quiet state before programming this mode.
- No-standby mode: This is a backup mode intended to be used only if smart-standby mode is bugged. When in this mode, the ISS never asserts the MStandBy signal.

### 8.1.2.4.3 ISS IDLE Mechanism

The PRCM module can request the ISS to go into IDLE state when the ISS has asserted its MStandBy output.

In a normal case, software must ensure that the ISS is in a quiet state before allowing the PRCM module to send an IDLE request to the ISS:

- The ISS has no more traffic to generate.
- The ISS cannot generate any new interrupts.
- The ISS has no pending interrupts.

When an IDLE request is received from the PRCM module, the ISS power manager verifies that the ISS MWait input has been asserted and that all ISS interrupt outputs are deasserted. It then starts the STANDBY-to-IDLE transition:

1. Send IDLE request to the ISS configuration interconnect.
2. The ISS configuration interconnect requests disconnection of both ISS interface slave ports. Disconnection is done by the master:
  - It stops accepting new requests and drains currently ongoing ones.
  - It waits for completion of all ongoing transactions.
3. The ISS configuration interconnect acknowledges the IDLE transition.
4. The ISS power manager acknowledges the IDLE request from the PRCM module.
5. The PRCM module can cut the ISS clock and power.

The PRCM module first enables the ISS power and clock before requesting the ISS to go into functional state by deasserting the SIdleReq signal. The ISS power manager then executes the wake-up sequence:

1. Request the ISS interconnect to go into functional state.
2. The ISS interconnect connects the ISS slave ports.
3. The ISS interconnect acknowledges transition into functional mode.
4. The ISS power manager acknowledges transition into functional mode (ISS output SIdleAck = 00).

Four modes for IDLE control are supported:

- Smart-idle-wakeup mode [b11]: This is the mode normally used. When in this mode, the ISS acknowledges a request to go idle from the power manager after having performed all hardware operations necessary for the IAF to be in a correct quiet state.
- Smart-idle mode [b10]: This is equivalent to smart-idle-wakeup mode.
- Force-idle mode [b00]: This is a backup mode intended to be used only if smart-idle mode is bugged. When in this mode, the ISS acknowledges a request to go idle from the power manager with no hardware condition. Software must ensure that the ISS is in a correct quiet state before requesting a force-idle transition.
- No-idle mode [b01]: When in this mode, the ISS disregards any request to go idle from the power manager.

### 8.1.3 ISS Register Manual

#### 8.1.3.1 ISS Instance Summary

Table 8-15 is the ISS instance.

**Table 8-15. ISS Instance Summary**

Module Name	Module Base Address	Size
ISS_TOP	0x5200 0000	256 bytes

**NOTE:** This section contains only the ISS TOP registers. For more submodule register details, see the register manual of the particular submodule.

#### 8.1.3.2 ISS Registers

##### 8.1.3.2.1 ISS TOP Register Summary

Table 8-16 summarizes the ISS TOP register mapping.

**Table 8-16. ISS TOP Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_TOP Base Address
ISS_HL_REVISION	R	32	0x0000 0000	0x5200 0000
RESERVED	R	32	0x0000 0004	0x5200 0004
ISS_HL_SYSCONFIG	RW	32	0x0000 0010	0x5200 0010
RESERVED	RW	32	0x0000 001C	0x5200 001C
ISS_HL_IRQSTATUS_R AW <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0020 + (0x10 * i)	0x5200 0020 + (0x10 * i)
ISS_HL_IRQSTATUS_i (1)	RW	32	0x0000 0024 + (0x10 * i)	0x5200 0024 + (0x10 * i)
ISS_HL_IRQENABLE_S ET <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0028 + (0x10 * i)	0x5200 0028 + (0x10 * i)
ISS_HL_IRQENABLE_C LR <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 002C + (0x10 * i)	0x5200 002C + (0x10 * i)
ISS_CTRL	RW	32	0x0000 0080	0x5200 0080
ISS_CLKCTRL	W	32	0x0000 0084	0x5200 0084
ISS_CLKSTAT	R	32	0x0000 0088	0x5200 0088
ISS_PM_STATUS	R	32	0x0000 008C	0x5200 008C

<sup>(1)</sup> i = 0 to 5

##### 8.1.3.2.2 ISS TOP Register Description

Table 8-17 through Table 8-35 describe the ISS TOP registers.

**Table 8-17. ISS\_HL\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 8-16</a> .	<b>Instance</b>	ISS_TOP
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 8-18. Register Call Summary for Register ISS\_HL\_REVISION**

ISS Overview

- [ISS Registers: \[0\]](#)

**Table 8-19. ISS\_HL\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 8-16</a> .	<b>Instance</b>	ISS_TOP
<b>Description</b>	Clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STANDBYMODE	IDLEMODE	RESERVED	SOFTRESET												

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5:4	STANDBYMODE	Master interface power management, standby/Wait control 0x0: Force-standby. MStandby is asserted unconditionally. 0x1: No-standby. MStandby is never asserted. 0x3: Smart-standby 0x2: Smart-standby	RW	0x2
3:2	IDLEMODE	IDLE protocol configuration 0x0: Force-idle 0x1: No-idle 0x3: Smart-idle 0x2: Smart-idle	RW	0x2
1	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
0	SOFTRESET	Software reset. Write 0x0: No action Write 0x1: Initiate software reset Read 0x1: Reset (software or other) ongoing Read 0x0: Reset done, no pending action	RW	0

**Table 8-20. Register Call Summary for Register ISS\_HL\_SYSCONFIG**

ISS Overview

- [ISS Reset: \[0\] \[1\] \[2\]](#)
- [ISS Registers: \[3\]](#)

**Table 8-21. ISS\_HL\_IRQSTATUS\_RAW\_i**

<b>Address Offset</b>	0x0000 0020 + (0x10 * i)	<b>Index</b>	i = 0 to 5
<b>Physical Address</b>	See <a href="#">Table 8-16</a> .	<b>Instance</b>	ISS_TOP
<b>Description</b>	Per-event raw interrupt status vector, line 0. Raw status is set even if event is not enabled by setting the <a href="#">ISS_HL_IRQENABLE_SET_i</a> register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																HS_VS_IRQ	CCP2_IRQ8	SIMCOP_IRQ3	SIMCOP_IRQ2	SIMCOP_IRQ1	SIMCOP_IRQ0	BTE_IRQ	CBUFF_IRQ	CCP2_IRQ3	CCP2_IRQ2	CCP2_IRQ1	CCP2_IRQ0	CSIB_IRQ	CSIA_IRQ	ISP_IRQ3	ISP_IRQ2	ISP_IRQ1	ISP_IRQ0

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17	HS_VS_IRQ	HS or VS synchronization event. This event is triggered if a rising or falling edge is detected on the HS or VS signal (after the video port mux). The rising or falling edge and the HS or VS signal selection is chosen with the <a href="#">ISS_CTRL.SYNC_DETECT</a> bit field. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW	0
16	CCP2_IRQ8	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No event pending	R	0
15	SIMCOP_IRQ3	Event generated by SIMCOP Read 0x1: Event pending Read 0x0: No event pending	R	0
14	SIMCOP_IRQ2	Event generated by SIMCOP Read 0x1: Event pending Read 0x0: No event pending	R	0
13	SIMCOP_IRQ1	Event generated by SIMCOP Read 0x1: Event pending Read 0x0: No event pending	R	0

Bits	Field Name	Description	Type	Reset
12	SIMCOP_IRQ0	Event generated by SIMCOP Read 0x1: Event pending Read 0x0: No event pending	R	0
11	BTE_IRQ	Event generated by the burst translation engine Read 0x1: Event pending Read 0x0: No event pending	R	0
10	CBUFF_IRQ	Event generated by the circular buffer Read 0x1: Event pending Read 0x0: No event pending	R	0
9	CCP2_IRQ3	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No event pending	R	0
8	CCP2_IRQ2	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No event pending	R	0
7	CCP2_IRQ1	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No event pending	R	0
6	CCP2_IRQ0	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No event pending	R	0
5	CSIB_IRQ	Event generated by the CSI2 receiver #b Read 0x1: Event pending Read 0x0: No event pending	R	0
4	CSIA_IRQ	Event generated by the CSI2 receiver #a Read 0x1: Event pending Read 0x0: No event pending	R	0
3	ISP_IRQ3	Combined interrupt event provided by the ISP. Read 0x1: Event pending Read 0x0: No event pending	R	0
2	ISP_IRQ2	Combined interrupt event provided by the ISP. Read 0x1: Event pending Read 0x0: No event pending	R	0
1	ISP_IRQ1	Combined interrupt event provided by the ISP. Read 0x1: Event pending Read 0x0: No event pending	R	0
0	ISP_IRQ0	Combined interrupt event provided by the ISP. Read 0x1: Event pending Read 0x0: No event pending	R	0

**Table 8-22. Register Call Summary for Register ISS\_HL\_IRQSTATUS\_RAW\_i**

ISS Overview

- [ISS Interrupts: \[0\]](#)
- [ISS Registers: \[1\]](#)



**Table 8-23. ISS\_HL\_IRQSTATUS\_i**

<b>Address Offset</b>	0x0000 0024 + (0x10 * i)	<b>Index</b>	i = 0 to 5
<b>Physical Address</b>	See <a href="#">Table 8-16</a> .	<b>Instance</b>	ISS_TOP
<b>Description</b>	Per-event "enabled" interrupt status vector, line 0. Enabled status is not set unless event is enabled by setting the <a href="#">ISS_HL_IRQENABLE_SET_i</a> register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																HS_VS_IRQ	CCP2_IRQ8	SIMCOP_IRQ3	SIMCOP_IRQ2	SIMCOP_IRQ1	SIMCOP_IRQ0	BTE_IRQ	CBUFF_IRQ	CCP2_IRQ3	CCP2_IRQ2	CCP2_IRQ1	CCP2_IRQ0	CSIB_IRQ	CSIA_IRQ	ISP_IRQ3	ISP_IRQ2	ISP_IRQ1	ISP_IRQ0

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17	HS_VS_IRQ	HS or VS synchronization event. This event is triggered if a rising or falling edge is detected on the HS or VS signal (after the video port mux). The rising or falling edge and the HS or VS signal selection is chosen with the <a href="#">ISS_CTRL.SYNC_DETECT</a> bit field.  Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toSet	0
16	CCP2_IRQ8	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
15	SIMCOP_IRQ3	Event generated by SIMCOP Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
14	SIMCOP_IRQ2	Event generated by SIMCOP Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
13	SIMCOP_IRQ1	Event generated by SIMCOP Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
12	SIMCOP_IRQ0	Event generated by SIMCOP Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
11	BTE_IRQ	Event generated by the burst translation engine Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
10	CBUFF_IRQ	Event generated by the circular buffer Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
9	CCP2_IRQ3	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0

Bits	Field Name	Description	Type	Reset
8	CCP2_IRQ2	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
7	CCP2_IRQ1	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
6	CCP2_IRQ0	Event generated by the CCP2 receiver Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
5	CSIB_IRQ	Event generated by the CSI2 receiver #b Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
4	CSIA_IRQ	Event generated by the CSI2 receiver #a Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
3	ISP_IRQ3	Combined interrupt event provided by the ISP. Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
2	ISP_IRQ2	Combined interrupt event provided by the ISP. Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
1	ISP_IRQ1	Combined interrupt event provided by the ISP. Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0
0	ISP_IRQ0	Combined interrupt event provided by the ISP. Read 0x1: Event pending Read 0x0: No (enabled) event pending	R	0

**Table 8-24. Register Call Summary for Register ISS\_HL\_IRQSTATUS\_i**

ISS Overview

- [ISS Interrupts: \[0\] \[1\]](#)
- [ISS Registers: \[2\]](#)

**Table 8-25. ISS\_HL\_IRQENABLE\_SET\_i**

<b>Address Offset</b>	0x0000 0028 + (0x10 * i)	<b>Index</b>	i = 0 to 5
<b>Physical Address</b>	See <a href="#">Table 8-16</a> .	<b>Instance</b>	ISS_TOP
<b>Description</b>	Per-event interrupt enable bit vector Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
RESERVED																HS_VS_IRQ	CCP2_IRQ08	SIMCOP_IRQ03	SIMCOP_IRQ02	SIMCOP_IRQ01	SIMCOP_IRQ00	BTE_IRQ	CBUFF_IRQ	CCP2_IRQ03	CCP2_IRQ02	CCP2_IRQ01	CCP2_IRQ00	CSIB_IRQ	CSIA_IRQ	ISP_IRQ03	ISP_IRQ02	ISP_IRQ01	ISP_IRQ00								

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17	HS_VS_IRQ	<p>HS or VS synchronization event. This event is triggered if a rising or falling edge is detected on the HS or VS signal (after the video port mux). The rising or falling edge and the HS or VS signal selection is chosen with the <a href="#">ISS_CTRL.SYNC_DETECT</a> bit field.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
16	CCP2_IRQ8	<p>Event generated by the CCP2 receiver</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
15	SIMCOP_IRQ3	<p>Event generated by SIMCOP</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
14	SIMCOP_IRQ2	<p>Event generated by SIMCOP</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
13	SIMCOP_IRQ1	<p>Event generated by SIMCOP</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
12	SIMCOP_IRQ0	<p>Event generated by SIMCOP</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
11	BTE_IRQ	<p>Event generated by the burst translation engine</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
10	CBUFF_IRQ	<p>Event generated by the circular buffer</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
9	CCP2_IRQ3	<p>Event generated by the CCP2 receiver</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
8	CCP2_IRQ2	Event generated by the CCP2 receiver Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
7	CCP2_IRQ1	Event generated by the CCP2 receiver Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
6	CCP2_IRQ0	Event generated by the CCP2 receiver Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
5	CSIB_IRQ	Event generated by the CSI2 receiver #b Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
4	CSIA_IRQ	Event generated by the CSI2 receiver #a Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
3	ISP_IRQ3	Combined interrupt event provided by the ISP. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
2	ISP_IRQ2	Combined interrupt event provided by the ISP. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
1	ISP_IRQ1	Combined interrupt event provided by the ISP. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
0	ISP_IRQ0	Combined interrupt event provided by the ISP. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

**Table 8-26. Register Call Summary for Register ISS\_HL\_IRQENABLE\_SET\_i**

ISS Overview

- ISS Interrupts: [0]
- ISS Registers: [1] [2] [3]

**Table 8-27. ISS\_HL\_IRQENABLE\_CLR\_i**

<b>Address Offset</b>	0x0000 002C + (0x10 * i)	<b>Index</b>	i = 0 to 5
<b>Physical Address</b>	See Table 8-16.	<b>Instance</b>	ISS_TOP
<b>Description</b>	Per-event interrupt enable bit vector Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																HS_VS_IRQ	CCP2_IRQ8	SIMCOP_IRQ3	SIMCOP_IRQ2	SIMCOP_IRQ1	SIMCOP_IRQ0	BTE_IRQ	CBUFF_IRQ	CCP2_IRQ3	CCP2_IRQ2	CCP2_IRQ1	CCP2_IRQ0	CSIB_IRQ	CSIA_IRQ	ISP_IRQ3	ISP_IRQ2	ISP_IRQ1	ISP_IRQ0

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17	HS_VS_IRQ	HS or VS synchronization event. This event is triggered if a rising or falling edge is detected on the HS or VS signal (after the video port mux). The rising or falling edge and the HS or VS signal selection is chosen with the ISS_CTRL.SYNC_DETECT bit field. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
16	CCP2_IRQ8	Event generated by the CCP2 receiver Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
15	SIMCOP_IRQ3	Event generated by SIMCOP Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
14	SIMCOP_IRQ2	Event generated by SIMCOP Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
13	SIMCOP_IRQ1	Event generated by SIMCOP Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
12	SIMCOP_IRQ0	Event generated by SIMCOP Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
11	BTE_IRQ	Event generated by the BTE Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
10	CBUFF_IRQ	Event generated by the CBUFF Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
9	CCP2_IRQ3	Event generated by the CCP2 receiver Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
8	CCP2_IRQ2	Event generated by the CCP2 receiver Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
7	CCP2_IRQ1	Event generated by the CCP2 receiver Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
6	CCP2_IRQ0	Event generated by the CCP2 receiver Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
5	CSIB_IRQ	Event generated by the CSI2 receiver b Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
4	CSIA_IRQ	Event generated by the CSI2 receiver a Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
3	ISP_IRQ3	Combined interrupt event provided by the ISP. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
2	ISP_IRQ2	Combined interrupt event provided by the ISP. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
1	ISP_IRQ1	Combined interrupt event provided by the ISP. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
0	ISP_IRQ0	Combined interrupt event provided by the ISP. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

**Table 8-28. Register Call Summary for Register ISS\_HL\_IRQENABLE\_CLR\_i**

ISS Overview

- [ISS Interrupts: \[0\]](#)
- [ISS Registers: \[1\]](#)

**Table 8-29. ISS\_CTRL**

<b>Address Offset</b>	0x0000 0080	<b>Instance</b>	ISS_TOP
<b>Physical Address</b>	See <a href="#">Table 8-16</a> .		
<b>Description</b>	ISS control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSI2_B_TAG_CNT				CSI2_A_TAG_CNT				CCP2W_TAG_CNT				CCP2R_TAG_CNT				RESERVED										ISS_CLK_DIV		INPUT_SEL		SYNC_DETECT	

Bits	Field Name	Description	Type	Reset
31:28	CSI2_B_TAG_CNT	Defines the maximum number of tags that could be used by the CSI2 b write bridge. Note: Tag count must be set to 16 for best performance.	RW	0x0
27:24	CSI2_A_TAG_CNT	Defines the maximum number of tags that could be used by the CSI2 a write bridge. Note: Tag count must be set to 16 for best performance.	RW	0x0
23:20	CCP2W_TAG_CNT	Defines the maximum number of tags that could be used by the CCP2 write bridge Note: Tag count must be set to 16 for best performance.	RW	0x0

Bits	Field Name	Description	Type	Reset
19:16	CCP2R_TAG_CNT	Defines the maximum number of tags that could be used by the CCP2 read bridge Note: Tag count must be set to 16 for best performance.	RW	0x0
15:6	RESERVED		R	0x000
5:4	ISS_CLK_DIV	ISS functional clock division CLK refers to the input clock provided to the ISS. FCLK is the functional clock provided to ISS top level and submodules. CFGCLK is the clock used for the configuration network.  0x0: FCLK=CLK CFGCLK=CLK/2  0x1: FCLK=CLK/2 CFGCLK=CLK/4  0x3: Reserved  0x2: FCLK=CLK/4 CFGCLK=CLK/8	RW	0x0
3:2	INPUT_SEL	Selects ISP input  0x0: CSI2-A  0x1: CSI2-B  0x2: CCP2	RW	0x0
1:0	SYNC_DETECT	Chooses among rising and falling edge for the HS_VS_IRQ synchronization event  0x0: HS falling edge  0x1: HS raising edge  0x3: VS raising edge  0x2: VS falling edge	RW	0x0

**Table 8-30. Register Call Summary for Register ISS\_CTRL**

ISS Overview

- [ISS PRCM Interface Integration: \[0\]](#)
- [ISS Registers: \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 8-31. ISS\_CLKCTRL**

<b>Address Offset</b>	0x0000 0084	<b>Instance</b>	ISS_TOP
<b>Physical Address</b>	See <a href="#">Table 8-16</a> .		
<b>Description</b>	ISS clock control register. Use to enable/disable the interface and functional clock of ISS submodules.		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	VPORT2_CLK	VPORT1_CLK	VPORT0_CLK	RESERVED													CCP2	CSI2_B	CSI2_A	ISP	SIMCOP										

Bits	Field Name	Description	Type	Reset
31	RESERVED		RW	1
30	VPORT2_CLK	Enables the pixel clock from the CCP2 protocol engine  0x0: Disabled  0x1: Enabled	RW	1



Bits	Field Name	Description	Type	Reset
29	VPORT1_CLK	Enables the pixel clock from the CSI2_B protocol engine 0x0: Disabled 0x1: Enabled	RW	1
28	VPORT0_CLK	Enables the pixel clock from the CSI2_A protocol engine 0x0: Disabled 0x1: Enabled	RW	1
27:5	RESERVED		R	0x000000
4	CCP2	CCP2 Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off. Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.	W	0
3	CSI2_B	CSI2_B Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off. Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.	W	0
2	CSI2_A	CSI2_A Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off. Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.	W	0
1	ISP	ISP Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off. Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.	W	0
0	SIMCOP	SIMCOP Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off. Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.	W	0

**Table 8-32. Register Call Summary for Register ISS\_CLKCTRL**

ISS Overview

- [ISS PRCM Interface Integration: \[0\] \[1\]](#)
- [ISS Clocks: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [ISS Registers: \[10\]](#)

**Table 8-33. ISS\_CLKSTAT**

<b>Address Offset</b>	0x0000 0088	<b>Instance</b>	ISS_TOP
<b>Physical Address</b>	See <a href="#">Table 8-16</a> .		
<b>Description</b>	ISS clock status register.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	VPORT2_CLK	VPORT1_CLK	VPORT0_CLK	RESERVED													CCP2	CSI2_B	CSI2_A	ISP	SIMCOP										

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	1
30	VPORT2_CLK	Status of the pixel clock from the CCP2 protocol engine Read 0x1: Enabled Read 0x0: Disabled	R	1
29	VPORT1_CLK	Status of the pixel clock from the CSI2_B protocol engine Read 0x1: Enabled Read 0x0: Disabled	R	1
28	VPORT0_CLK	Status of the pixel clock from the CSI2_A protocol engine Read 0x1: Enabled Read 0x0: Disabled	R	1
27:5	RESERVED		R	0x000000
4	CCP2	CCP2 Read 0x1: The submodule is on. Read 0x0: The submodule is off.	R	0
3	CSI2_B	CSI2_B Read 0x1: The submodule is on. Read 0x0: The submodule is off.	R	0
2	CSI2_A	CSI2_A Read 0x1: The submodule is on. Read 0x0: The submodule is off.	R	0
1	ISP	ISP Read 0x1: The submodule is on. Read 0x0: The submodule is off.	R	0
0	SIMCOP	SIMCOP Read 0x1: The submodule is on. Read 0x0: The submodule is off.	R	0

**Table 8-34. Register Call Summary for Register ISS\_CLKSTAT**

ISS Overview

- [ISS Clocks: \[0\] \[1\] \[2\]](#)
- [ISS Registers: \[3\]](#)

**Table 8-35. ISS\_PM\_STATUS**

<b>Address Offset</b>	0x0000 008C	<b>Instance</b>	ISS_TOP
<b>Physical Address</b>	See <a href="#">Table 8-16</a> .		
<b>Description</b>	ISS power manager status register. SW could know what modules are in functional or STANDBY/IDLE state. This feature could be particularly useful to debug when ISS does not go into STANDBY mode		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			CBUFF_PM		BTE_PM		SIMCOP_PM		ISP_PM		CCP2_PM		CSI2_B_PM		CSI2_A_PM

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:12	CBUFF_PM	Power status of the CBUFF. Read 0x2: Functional Read 0x1: Transition Read 0x0: Idle	R	0x0
11:10	BTE_PM	Power status of the BTE. Read 0x2: Functional Read 0x1: Transition Read 0x0: Idle	R	0x0
9:8	SIMCOP_PM	Power status of the SIMCOP. Read 0x2: Functional Read 0x1: Transition Read 0x0: Standby	R	0x0
7:6	ISP_PM	Power status of the ISP. Read 0x2: Functional Read 0x1: Transition Read 0x0: Standby	R	0x0
5:4	CCP2_PM	Power status of the CCP2. Read 0x2: Functional Read 0x1: Transition Read 0x0: Standby	R	0x0
3:2	CSI2_B_PM	Power status of the CSI2 module b Read 0x2: Functional Read 0x1: Transition Read 0x0: Standby	R	0x0
1:0	CSI2_A_PM	Power status of the CSI2 module a Read 0x2: Functional Read 0x1: Transition Read 0x0: Standby	R	0x0

**Table 8-36. Register Call Summary for Register ISS\_PM\_STATUS**

ISS Overview

- [ISS Registers: \[0\]](#)

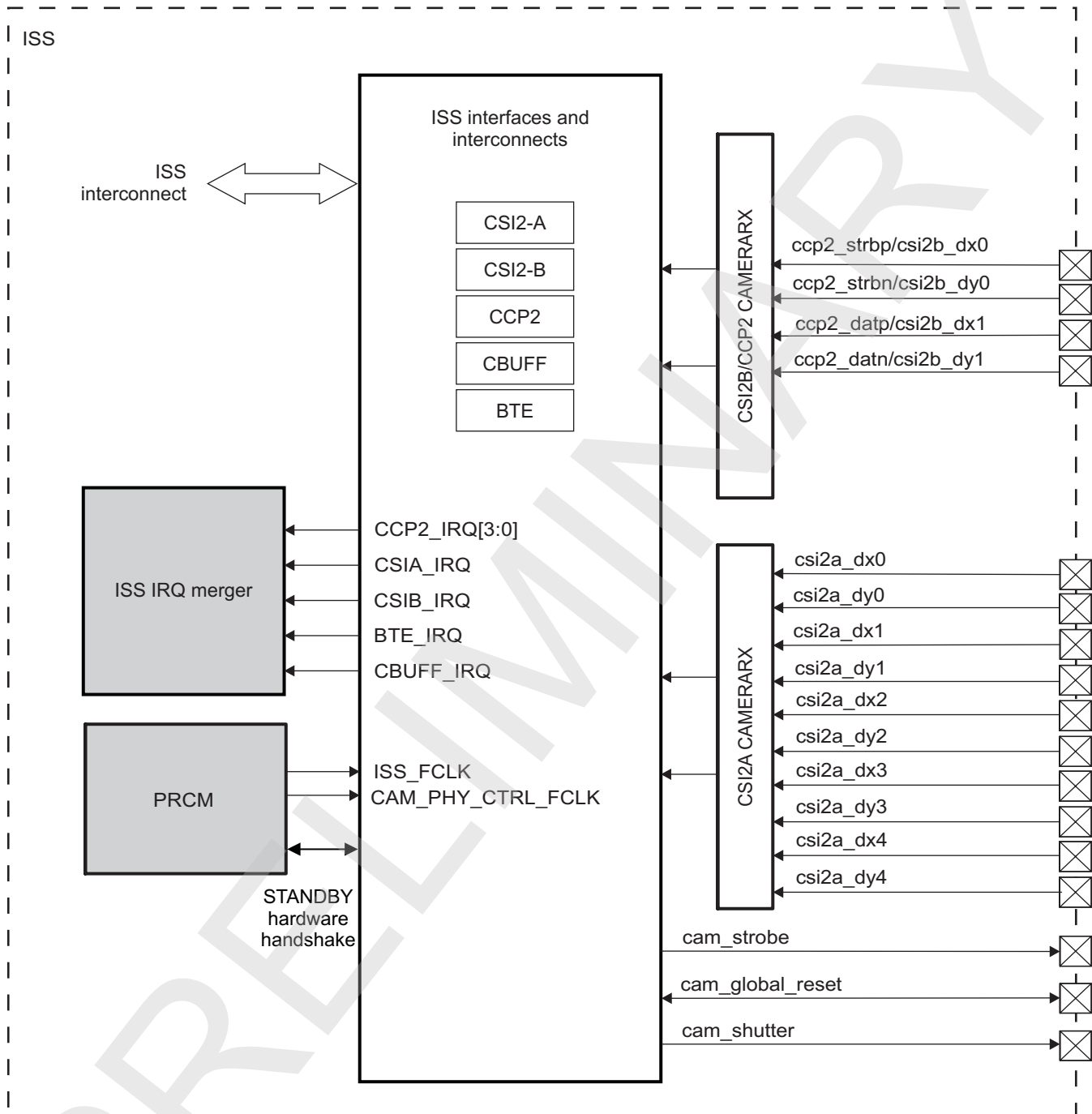
## 8.2 ISS Interfaces

### 8.2.1 ISS Interfaces Overview

Along with the submodules, the ISS has two camera interfaces: primary and secondary. The primary interface (CSI2-A) supports MIPI® CSI2 protocol with four data lanes, and the secondary interface (CSI2-B/CCP2) supports CSI2 protocol with one data lane or CCP2 protocol. All interfaces can use the image signal processor (ISP), but not concurrently. When one interface uses the ISP, the other must send data to memory. However, the ISP can still be used to process this data in memory-to-memory. Time multiplex processing is also possible.

Figure 8-5 shows the ISS interfaces and interconnects.

Figure 8-5. ISS Interfaces and Interconnects Highlight



camss-006

### 8.2.1.1 ISS Interface Features

The camera subsystem supports the following features:

- System interfaces and interconnects:
  - Two 32-bit-wide configuration interfaces:
    - Interface to Cortex™-M3 microprocessor unit (MPU): Synchronous to the functional clock
    - Interface to Cortex™-A9 MPU and system direct memory access (sDMA): Asynchronous from

- the functional clock
- 128-bit-wide data interface to level 3 (L3) interconnect: Asynchronous from the functional clock
  - Shared interface level 2 (L2) in IVA-HD module for hardware encoding
  - Outside connection using the L3 interconnect through the TILER to the synchronous dynamic random access memory (SDRAM) controller (SDRC), which acts as the primary interface between the SDRAM and the ISS functional block
- The ISS has a local interconnect that connects all modules inside the ISS.
- BTE:
  - Tightly coupled with the TILER to support efficient rotation
- CBUFF:
  - Maps a linear space into a circular buffer
  - The buffer is physically located in system memory.
- TCTRL:
  - Control signal generation for flash prestrobe and strobe
  - Camera global reset control
- Camera interfaces:
  - Two CSI2 camera interfaces: CSI2-A (primary) and CSI2-B (secondary)
    - Transfer pixels and data received by the CSI2 digital physical layer receiver to the system memory or to the ISP
    - Use unidirectional data link
    - CSI2-A supports four configurable data links in addition to the clock signaling.
    - CSI2-B supports one configurable data link in addition to the clock signaling.
    - Maximum data rate of 1 Gbps per data lane
    - Data merger for 2-, 3-, or 4-data lane configuration
    - Maximum data rate of 1 Gbps per data lane, possible configurations are:
      - One data lane: 1000 Mbps
      - Two data lanes: 2 × 1000 Mbps
      - Three data lanes: 3 × 1000 Mbps
      - Four data lanes: 4 × 824 Mbps
    - Error detection and correction by the protocol engine
    - Direct memory access (DMA) engine integrated with dedicated first in first out (FIFO)
    - One-dimensional (1D) and two-dimensional (2D) addressing mode
    - Burst support
    - Streaming burst support (64- or 32-bit)
    - Eight contexts to support eight dedicated configurations of virtual channel ID and data types
    - Ping-pong mechanism for double-buffering
    - All primary and secondary MIPI-defined formats are supported.
    - Conversion of the RGB formats
    - On-the-fly differential pulse code modulation (DPCM) decompression
    - On-the-fly image cropping and A-law/DPCM compression
    - Configuration of the complex input/output (I/O) physical layer (PHY) (MIPI D-PHY-compliant receiver PHY solution [D-PHY mode]/CCP2-compatible receiver PHY solution (CCP2 mode)
  - CCP2 camera interface (secondary)
    - Four logical channels
    - Transfer pixels and data received by the complex I/O PHY (CCP2 D-PHY RX to the system memory or the ISP)
    - Use unidirectional data link
    - Maximum data rate of 650 Mbps
    - DMA engine integrated with dedicated FIFO
    - 1D and 2D addressing mode
    - False synchronization code protection

- Ping-pong mechanism for double-buffering
- RGB, RAW, YUV, and JPEG formats supported
- On-the-fly DPCM decompression
- On-the-fly image cropping and A-Law/DPCM compression
- System memory data read back port (supported by the CCP2 protocol engine)
  - RAW 6, 7, 8, 10, 12, 14, 16 formats supported
  - DPCM and A-law decompression
  - Supports image cropping for compressed or uncompressed data

---

**NOTE:** Rotated DPCM data is not supported.

---

## 8.2.2 ISS Interfaces Environment

### 8.2.2.1 ISS Interfaces Signal Descriptions

Table 8-37 summarizes the I/O signals.

**Table 8-37. ISS I/O Description**

Signal Name	I/O <sup>(1)</sup>	Description	Serial Mode CCP2	Serial Mode CSI2
cam_strobe	O	Flash strobe control signal	x	x
cam_shutter	O	Mechanical shutter control signal	x	x
cam_global_reset	I/O	Global reset release shutter signal	x	x
csi2a_dx0	I	Serial CSI2-A mode: Differential clock positive input		x
csi2a_dy0	I	Serial CSI2-A mode: Differential clock negative input		x
csi2a_dx1	I	Serial CSI2-A mode: Differential data lane positive input		x
csi2a_dy1	I	Serial CSI2-A mode: Differential data lane negative input		x
csi2a_dx2	I	Serial CSI2-A mode: Differential data lane positive input		x
csi2a_dy2	I	Serial CSI2-A mode: Differential data lane negative input		x
csi2a_dx3	I	Serial CSI2-A mode: Differential data lane positive input		x
csi2a_dy3	I	Serial CSI2-A mode: Differential data lane negative input		x
csi2a_dx4	I	Serial CSI2-A mode: Differential data lane positive input		x
csi2a_dy4	I	Serial CSI2-A mode: Differential data lane negative input		x
csi2b_dx0	I	Serial CSI2-B mode: Differential clock lane positive input		x
csi2b_dy0	I	Serial CSI2-B mode: Differential clock lane negative input		x
csi2b_dx1	I	Serial CSI2-B mode: Differential data lane positive input		x
csi2b_dy1	I	Serial CSI2-B mode: Differential data lane negative input		x
ccp2_strbp	I	Serial CCP2 mode: Differential clock positive input	x	
ccp2_strbn	I	Serial CCP2 mode: Differential clock negative input	x	

<sup>(1)</sup> I = Input; O = Output



**Table 8-37. ISS I/O Description (continued)**

Signal Name	I/O <sup>(1)</sup>	Description	Serial Mode CCP2	Serial Mode CSI2
ccp2_datp	I	Serial CCP2 mode: Differential data positive input	x	
ccp2_datn	I	Serial CCP2 mode: Differential data negative input	x	

**NOTE:** Lane polarity can be changed in complex I/O. For more information, see [Section 8.2.2.2, ISS Interface Modes](#).

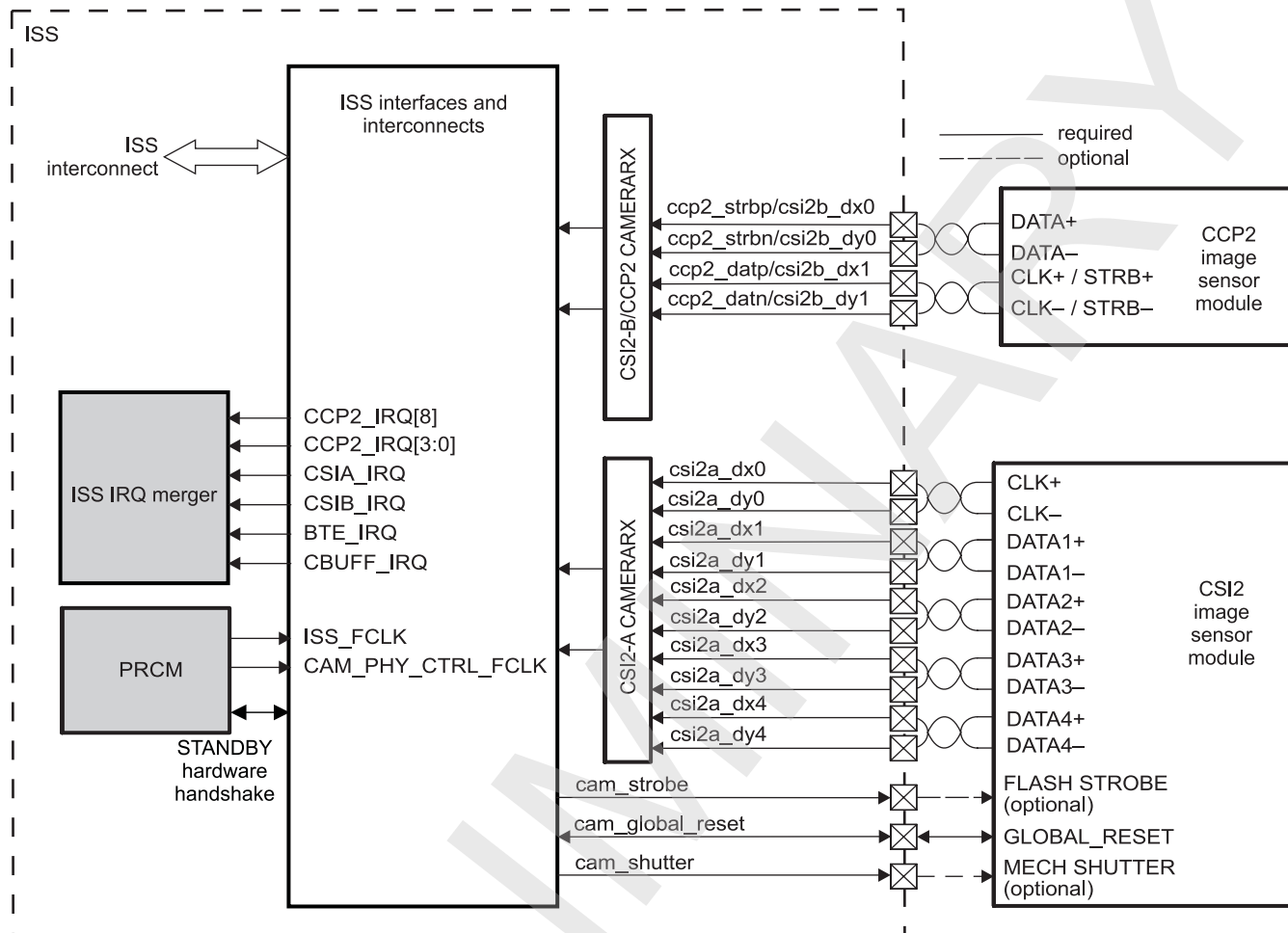
**NOTE:** At least one data lane must be configured for using the CSI2-A and CSI2-B interfaces. The signals are also configurable from the control module. Thus, they are not required to be at a certain location to act as clock or data; this can be configured.

**NOTE:** CCP2 and CSI2-B share pins. Their modes are not supported as a pin configuration mux option. The modes are configured from the CSI2-B/CCP2 CAMERARX configured from the system control module (see [Chapter 18, Control Module](#)).

### 8.2.2.2 ISS Interface Modes

The camera subsystem can manage up to two serial image sensors that can be active at the same time. However, only one data flow can use the ISP. [Figure 8-6](#) shows an example block diagram of the interface configuration. Each serial port clock lane is configurable. The MIPI CSI2 protocol requires only a clock lane setup. The data lane configuration is optional. For more information, see [Section 8.2.5.1.1](#).

Figure 8-6. ISS CSI2-A, CSI2-B/CCP2 Serial Interface Configuration



**NOTE:** Depending on the needs, `cam_strobe`, `cam_global_reset`, and `cam_shutter` can be connected to the image sensor module rather than the CSI2 image sensor module.

**NOTE:** Dedicated power, reset, and clock management (PRCM) module pins for power and ground are provided for external sensors: `vdda_csi2a`, `vdda_csi2b`, `vssa_csi2a`, and `vssa_csi2b`.

A graphical representation of the pins for connecting external sensors to the ISS shows a variety of connectivity configurations that are possible by setting the PHY immediately before the receivers, and a variety of output paths, including some sensor data through the receivers to memory or the ISP. [Table 8-38](#) lists some of the connectivity possibilities; the ISP can take only one input lane at a time. For more information about the operating properties and configuration of the CAMERARX PHY, see [Section 8.2.3, ISS CSI2 PHY](#).

**Table 8-38. ISS Interfaces Connectivity Scheme Example Scenarios**

CSI2-A	On, 4DL CSI2-A <sup>(1)</sup> to ISP	On, 2DL CSI2-A <sup>(2)</sup> to ISP	Off	On, 2DL CSI2-A <sup>(2)</sup> to memory	On, 3DL CSI2-A <sup>(2)</sup> to ISP/memory	Off	Off
CCP2	Off	Off	On, 1DL CCP2 <sup>(1)</sup> to ISP	On, 1DL CCP2 <sup>(1)</sup> to ISP	Off	On, 1DL CCP2 <sup>(1)</sup> to memory	Off
CSI2-B	On, 1DL CSI2-B <sup>(1)</sup> to memory	Off	Off	Off	On, 1DL CSI2-B <sup>(1)</sup> to memory/ISP	Off	On, 1DL CSI2-B <sup>(1)</sup> to memory
<b>Signal</b>	<b>Simultaneous</b>	<b>Simultaneous</b>	<b>Simultaneous</b>	<b>Simultaneous</b>	<b>Simultaneous</b>	<b>Simultaneous</b>	<b>Simultaneous</b>
cam_strobe	x	x	x	x	x	x	x
cam_global_reset	x	x	x	x	x	x	x
cam_shutter	x	x	x	x	x	x	x
csi2a_dx0	x	x		x	x		
csi2a_dy0	x	x		x	x		
csi2a_dx1	x	x		x	x		
csi2a_dy1	x	x		x	x		
csi2a_dx2	x	x		x	x		
csi2a_dy2	x	x		x	x		
csi2a_dx3	x				x		
csi2a_dy3	x				x		
csi2a_dx4	x						
csi2a_dy4	x						
csi2b_dx0/ccp2 _strbp	x		x	x	x	x	x
csi2b_dy0/ccp2 _strbn	x		x	x	x	x	x
csi2b_dx1/ccp2 _datp	x		x	x	x	x	x
csi2b_dy1/ccp2 _datn	x		x	x	x	x	x

<sup>(1)</sup> Full: All data/clock lanes are connected.

<sup>(2)</sup> Limited: Some data/clock lanes are connected.

**NOTE:** The ISP can take only one input at a time. The other inputs must go to memory and then be processed after the ISP processes the input that was received first.

In looking at the sample connections between the ISS through the receivers and external sensors, it is apparent that the lanes of the input can be adjusted for certain needs. For example, it is not necessary that the CSI2-A has four data lanes connected at the same time. The only restriction is that CSI2-B is limited to one data lane. Because CSI2-B and CCP2 share pins, they cannot be used simultaneously. CSI2-A and CSI2-B/CCP2 can function at the same time by sending data to memory. In this manner, external sensors can be connected any way possible for best performance and results.

## 8.2.3 ISS CSI2 PHY

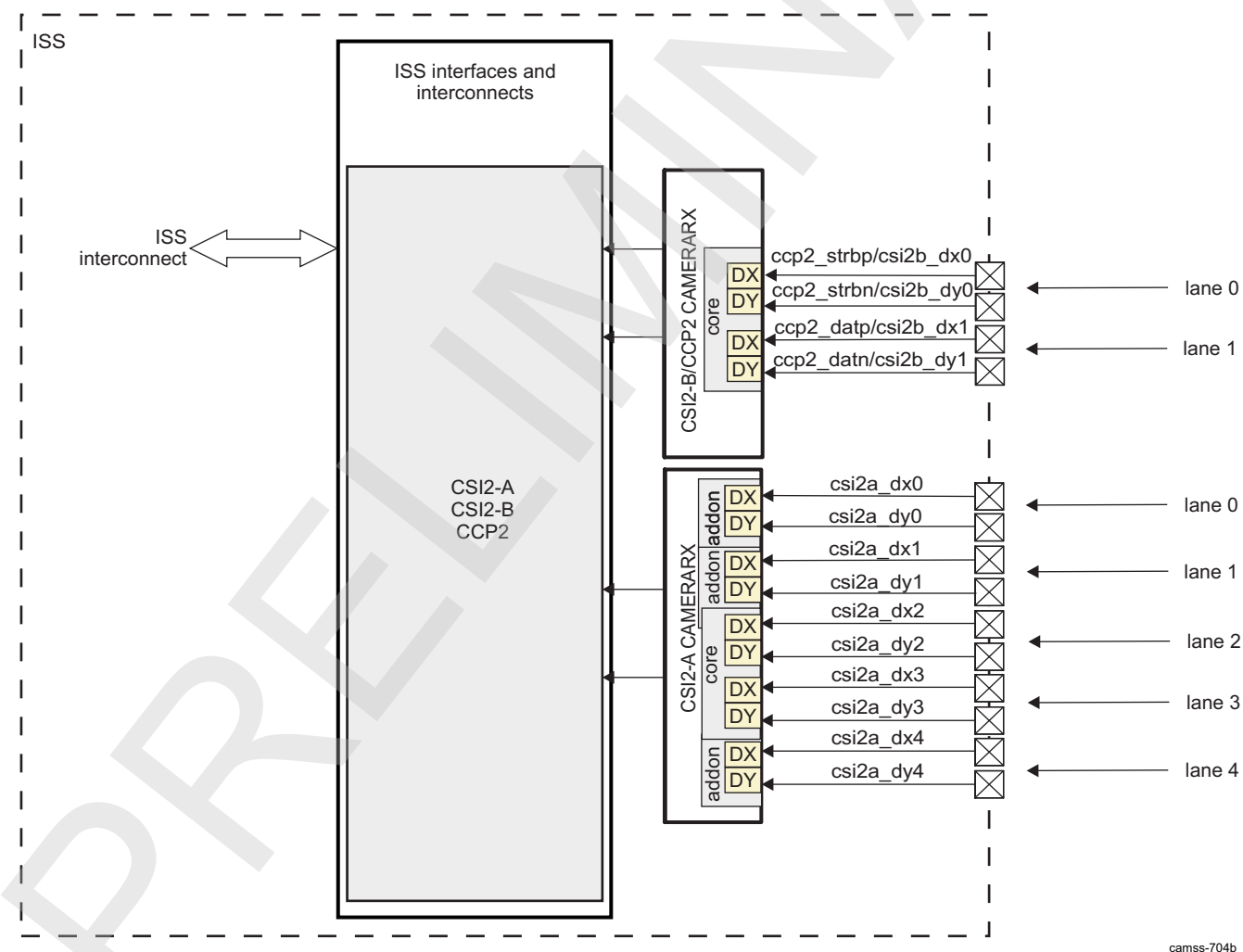
### 8.2.3.1 ISS CSI2 PHY Overview

Two MIPI D-PHY-compliant CAMERARX PHY receivers immediately before the ISS interfaces act as a physical connection and configuration of clock/data lanes with external sensors. CAMERARX PHY supports up to four configurations, depending on the required number of D-PHY data lane external sensors. The selection of CAMERARX in D-PHY mode or CCP2 mode must be done before reset and not on the fly.

The PHY is controlled and must be configured first from the control module for pad configuration. The differential data/clock lanes coming into the CSI2-A and CSI2-B CAMERARX are configured from registers explained in [Section 18.4.8.4](#) (see register CONTROL\_CAMERA\_RX).

The PHY is developed as two modules: CAMERARX\_CORE and CAMERARX\_ADDON. As mentioned previously, two PHYs are integrated in the device. The CSI2-A CAMERARX contains four data lanes, as shown in [Figure 8-7](#); CSI2-B is the same except that it has one data lane plus a clock/strobe lane. The figure also shows case CSI2-A, with four data lanes and one clock, listed in [Table 8-38](#).

**Figure 8-7. ISS Interfaces CSI2-A PHY Diagram Four D-PHY Data Lane Configuration**



**NOTE:** If CCP2 mode is chosen for use in the CSI2-B/CCP2 CAMERARX, then LANE 0 and LANE 1 are used only as datp/datn and strbp/strbn.

**NOTE:** LANE 4 can be used only as a data lane, never as a clock lane. All other configurations are possible.

CSI2-A and CSI2-B/CCP2 CAMERARX represent the overall PHY solution for connecting external sensors to feed the ISS. The MIPI D-PHY function can support up to four data lane modules and one clock lane module. Reverse direction escape mode is not supported. The lane module polarity and positions are configurable; that is, any lane module can be chosen as the clock lane module, and the DX/DY data pad for each lane module can be configured as DP or DN pins defined. The configuration and the selection of D-PHY mode, data/clock, or data/strobes are done through the control module (see [Chapter 18, Control Module](#)). The only exception is the four-data-lane use case, in which one corner lane is allowed to be only a data lane. For the CCP2 function, the CAMERARX PHY solution supports one data/strobe or data/clock lane module. The data lane module and strobe lane module must be physically adjacent. CAMERARX supports serial configuration protocol for the configuration interface.

### 8.2.3.2 ISS CSI2 PHY Functional Description

#### 8.2.3.2.1 ISS CSI2 PHY Functional Configuration

The CSI2 PHY converts the bitstream, divided into 1 to 4 serial data lanes, into a bitstream compatible with the CSI2 receiver and one clock lane.

The [CSI2\\_COMPLEXIO\\_IRQSTATUS](#) register logs complex I/O events of the following types:

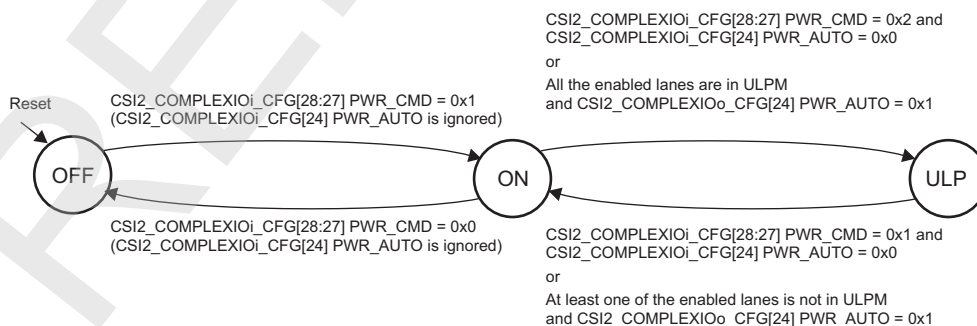
- Line power-state change (all lanes in ultralow-power mode [ULPM], at least one lane exits ULPM, etc.)
- Error on one lane

The CSI2 receiver embeds two registers to configure/read some complex I/O parameters:

- The [REGISTER0](#) register detects clock miss with respect to the MIPI specification and control timing.
- The [REGISTER0](#) register reports completion of reset on the different parts of the module and configures the timing parameters.
- The [CSI2\\_COMPLEXIO\\_CFG](#) registers contain the PWR\_AUTO and PWR\_CMD bit fields, which affect the power management of the two complex I/Os.

The complex I/O has three power modes: on, off, and ultra-low power (ULP). These modes can reflect the ON or ULP states of the five differential lines if the [CSI2\\_COMPLEXIO\\_CFG\[24\]](#) PWR\_AUTO bit is set to 1. If the PWR\_AUTO bit is at reset value (0), the complex I/O power state is controlled by the [CSI2\\_COMPLEXIO\\_CFG\[28:27\]](#) PWR\_CMD bit field, which directly defines the power state. [Figure 8-8](#) shows the complex I/O power finite state-machine (FSM).

**Figure 8-8. ISS CSI2 Complex I/O Power FSM**



camss-253

Another register, [CSI2\\_TIMING](#), is used to control the power state of the complex I/O modules with regard to the differential line state. This register controls the mode of the two complex I/Os (RxMode and NoRxMode) and the delay between the differential lanes in STOP state and the complex I/O on NoRxMode. The [CSI2\\_TIMING\[15\]](#) FORCE\_RX\_MODE\_IO1 bit sets the complex I/O in RxMode or NoRxMode (stopped mode). The FORCE\_RX\_MODE\_IO bit is automatically reset to 0 by hardware when

the counter ends and the FSM returns to NoRxMode. Three bits ([CSI2\\_TIMING\[14\]](#) STOP\_STATE\_X16\_IO1, [CSI2\\_TIMING\[13\]](#) STOP\_STATE\_X4\_IO1, and the [CSI2\\_TIMING\[12:0\]](#) STOP\_STATE\_COUNTER\_IO1 bit field) configure the delay between line stop mode and complex I/O stop mode. The delay represents the number of functional clock (ISS\_FCLK) cycles and can be calculated as follows:

Total delay in ISS\_FCLK cycle = [CSI2\\_TIMING.STOP\\_STATE\\_COUNTER\\_IO](#) x (1+[CSI2\\_TIMING.STOP\\_STATE\\_X16\\_IO](#) x 15) x (1+[CSI2\\_TIMING.STOP\\_STATE\\_X4\\_IO](#) x 3).

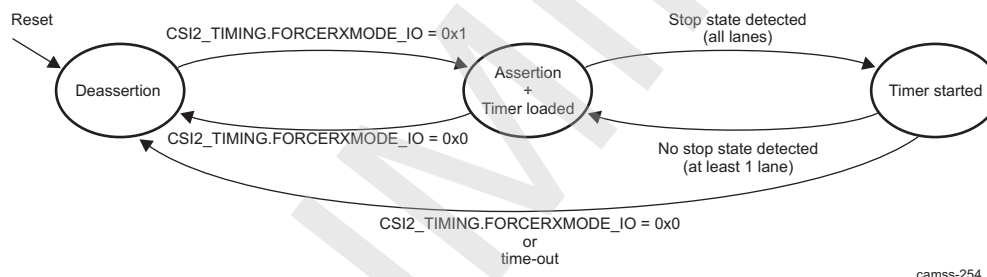
[Table 8-39](#) lists the possible values of the delay, in terms of the ISS\_FCLK cycles, depending on the values of the STOP\_STATE\_X16\_IO and STOP\_STATE\_X4\_IO bits.

**Table 8-39. ISS CSI2 Possible Time-Out Value for RxMode Counter**

STOP_STATE_X16_IO	STOP_STATE_X4_IO	Possible Delay Value (in Functional Clock Cycles)
0x0	0x0	8191 (with step of 1)
0x0	0x1	32764 (with step of 4)
0x1	0x0	131056 (with step of 16)
0x1	0x1	524224 (with step of 64)

The FORCERXMODE signal is used at initialization time (complex I/O). [Figure 8-9](#) describes the ForceRxMode and StopState FSM to assert and deassert the FORCERXMODE signal and to monitor STOPSTATE from the complex I/O.

**Figure 8-9. ISS CSI2 RxMode and StopState FSM**



### 8.2.3.2.2 ISS CSI2 PHY and Link Initialization Sequence

The MIPI D-PHY initialization sequence is not implemented within CAMERARX. The CSI2-A receiver is expected to coordinate the PHY initialization. The controller must ensure that the PHY is held in RESET/WAIT for RX mode until the D-PHY transmitter is powered up and the link comes to the defined state. The controller can use the STOPSTATE and FORCERXMODE signals of CAMERARX for this purpose. STOPSTATE indicates the line states, while FORCERXMODE forces the receiver state-machine into "wait for stop state." One possible initialization sequence is:

To fully initialize the CSIPHY, perform the following steps:

1. Configure all CSI2 receiver registers to be ready to receive signals/data from the CSIPHY:
  - (a) Configure all needed CSI2 registers:
    - (i) Set [CSI2\\_COMPLEXIO\\_CFG\[18:16\]](#) DATA4\_POSITION.
    - (ii) Set [CSI2\\_COMPLEXIO\\_CFG\[14:12\]](#) DATA3\_POSITION.
    - (iii) Set [CSI2\\_COMPLEXIO\\_CFG\[10:8\]](#) DATA2\_POSITION.
    - (iv) Set [CSI2\\_COMPLEXIO\\_CFG\[6:4\]](#) DATA1\_POSITION.
    - (v) Set [CSI2\\_COMPLEXIO\\_CFG\[2:0\]](#) CLOCK\_POSITION.
    - (vi) Set the [CONTROL\\_CAMERA\\_RX\[20:19\]](#) CAMERARX\_CSI22\_CAMMODE or [CONTROL\\_CAMERA\\_RX\[17:16\]](#) CAMERARX\_CSI21\_CAMMODE.

#### CAUTION

This must be done before the CSIPHY is active.

## 2. CSIPHY and link initialization sequence:

## (a) Deassert the CSIPHY reset.

- (i) Set
- `CSI2_COMPLEXIO_CFG[30] RESET_CTRL`
- to 0x1.

The following registers can be set only after deasserting the CSIPHY reset and before asserting the FORCERXMODE signal:

- REGISTER0
- REGISTER1
- REGISTER2

## (b) Assert the FORCERXMODE signal:

- (i) Set
- `CSI2_TIMING[15] FORCE_RX_MODE_IO1`
- to 0x1.

## (c) Connect pulldown on link (DP/DN) by asserting the respective PIPD\* signals (PIPD\* = 0):

For CSI2-A CAMERARX pulldown on signals through padconf registers:

- csi21\_dx4:
  - `CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[8] CSI21_DX4_INPUTENABLE` = 0x1
  - `CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[4] CSI21_DX4_PULLTYPESELECT` = 0x0
  - `CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[3] CSI21_DX4_PULLUDENABLE` = 0x1
- csi21\_dy4:
  - `CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[24] CSI21_DY4_INPUTENABLE` = 0x1
  - `CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[20] CSI21_DY4_PULLTYPESELECT` = 0x0
  - `CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4[19] CSI21_DY4_PULLUDENABLE` = 0x1
- csi21\_dx3:
  - `CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[8] CSI21_DX3_INPUTENABLE` = 0x1
  - `CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[4] CSI21_DX3_PULLTYPESELECT` = 0x0
  - `CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[3] CSI21_DX3_PULLUDENABLE` = 0x1
- csi21\_dy3:
  - `CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[24] CSI21_DY3_INPUTENABLE` = 0x1
  - `CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[20] CSI21_DY3_PULLTYPESELECT` = 0x0
  - `CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3[19] CSI21_DY3_PULLUDENABLE` = 0x1
- csi21\_dx2:
  - `CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[8] CSI21_DX2_INPUTENABLE` = 0x1
  - `CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[4] CSI21_DX2_PULLTYPESELECT` = 0x0
  - `CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[3] CSI21_DX2_PULLUDENABLE` = 0x1
- csi21\_dy2:
  - `CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[24] CSI21_DY2_INPUTENABLE` = 0x1
  - `CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[20] CSI21_DY2_PULLTYPESELECT` = 0x0
  - `CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY3[19]`

- CSI21\_DY2\_PULLUDENABLE = 0x1
- csi21\_dx1:
    - CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1[8] CSI21\_DX1\_INPUTENABLE = 0x1
    - CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1[4] CSI21\_DX1\_PULLTYPESELECT= 0x0
    - CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1[3] CSI21\_DX1\_PULLUDENABLE = 0x1
  - csi21\_dy1:
    - CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1[24] CSI21\_DY1\_INPUTENABLE = 0x1
    - CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1[20] CSI21\_DY1\_PULLTYPESELECT= 0x0
    - CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1[19] CSI21\_DY1\_PULLUDENABLE = 0x1
  - csi21\_dx0:
    - CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0[8] CSI21\_DX0\_INPUTENABLE = 0x1
    - CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0[4] CSI21\_DX0\_PULLTYPESELECT= 0x0
    - CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0[3] CSI21\_DX0\_PULLUDENABLE = 0x1
  - csi21\_dy0:
    - CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0[24] CSI21\_DY0\_INPUTENABLE = 0x1
    - CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0[20] CSI21\_DY0\_PULLTYPESELECT= 0x0
    - CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0[19] CSI21\_DY0\_PULLUDENABLE = 0x1

For CSI2-B/CCP2 CAMERARX pulldown on signals through padconf registers:

- csi22\_dx1:
  - CONTROL\_CORE\_PAD0\_CSI22\_DX1\_PAD1\_CSI22\_DY1[8] CSI22\_DX1\_INPUTENABLE = 0x1
  - CONTROL\_CORE\_PAD0\_CSI22\_DX1\_PAD1\_CSI22\_DY1[4] CSI22\_DX1\_PULLTYPESELECT = 0x0
  - CONTROL\_CORE\_PAD0\_CSI22\_DX1\_PAD1\_CSI22\_DY1[3] CSI22\_DX1\_PULLUDENABLE = 0x1
- csi22\_dy1:
  - CONTROL\_CORE\_PAD0\_CSI22\_DX1\_PAD1\_CSI22\_DY1[24] CSI22\_DY1\_INPUTENABLE = 0x1
  - CONTROL\_CORE\_PAD0\_CSI22\_DX1\_PAD1\_CSI22\_DY1[20] CSI22\_DY1\_PULLTYPESELECT = 0x0
  - CONTROL\_CORE\_PAD0\_CSI22\_DX1\_PAD1\_CSI22\_DY1[19] CSI22\_DY1\_PULLUDENABLE = 0x1
- csi22\_dx0:
  - CONTROL\_CORE\_PAD0\_CSI22\_DX0\_PAD1\_CSI22\_DY0[8] CSI22\_DX0\_INPUTENABLE = 0x1
  - CONTROL\_CORE\_PAD0\_CSI22\_DX0\_PAD1\_CSI22\_DY0[4] CSI22\_DX0\_PULLTYPESELECT = 0x0
  - CONTROL\_CORE\_PAD0\_CSI22\_DX0\_PAD1\_CSI22\_DY0[3] CSI22\_DX0\_PULLUDENABLE = 0x1
- csi22\_dy0:
  - CONTROL\_CORE\_PAD0\_CSI22\_DX0\_PAD1\_CSI22\_DY0[24] CSI22\_DY0\_INPUTENABLE



- = 0x1
- CONTROL\_CORE\_PAD0\_CSI22\_DX0\_PAD1\_CSI22\_DY0[20]  
CSI22\_DY0\_PULLTYPESELECT = 0x0
- CONTROL\_CORE\_PAD0\_CSI22\_DX0\_PAD1\_CSI22\_DY0[19]  
CSI22\_DY0\_PULLUDENABLE = 0x1
- (d) Power up the CSIPHY:
  - (i) Set **CSI2\_COMPLEXIO\_CFG**[28:27] PWR\_CMD to 0x1.
- (e) Check whether the state status reaches the ON state:
  - **CSI2\_COMPLEXIO\_CFG**[26:25] PWR\_STATUS = 0x1
- (f) Wait for STOPSTATE = 1 (for all enabled lane modules):
  - (i) The timer is set through the **CSI2\_TIMING**[14:0] bit field. The reset value can be kept.
  - (ii) Wait until **CSI2\_TIMING**[15] FORCE\_RX\_MODE\_IO1 = 0x0. It is automatically put at 0 when all enabled lanes are in STOPSTATE and the timer is finished.
- (g) Release PIPD\* (= 1).

For CSI2-A CAMERARX pullup on signals through padconf registers:

- csi21\_dx4:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX4\_PAD1\_CSI21\_DY4[4]  
CSI21\_DX4\_PULLTYPESELECT = 0x1
- csi21\_dy4:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX4\_PAD1\_CSI21\_DY4[20]  
CSI21\_DY4\_PULLTYPESELECT = 0x1
- csi21\_dx3:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX3\_PAD1\_CSI21\_DY3[4]  
CSI21\_DX3\_PULLTYPESELECT = 0x1
- csi21\_dy3:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX3\_PAD1\_CSI21\_DY3[20]  
CSI21\_DY3\_PULLTYPESELECT = 0x1
- csi21\_dx2:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX2\_PAD1\_CSI21\_DY2[4]  
CSI21\_DX2\_PULLTYPESELECT = 0x1
- csi21\_dy2:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX2\_PAD1\_CSI21\_DY2[20]  
CSI21\_DY2\_PULLTYPESELECT = 0x1
- csi21\_dx1:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1[4]  
CSI21\_DX1\_PULLTYPESELECT = 0x1
- csi21\_dy1:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1[20]  
CSI21\_DY1\_PULLTYPESELECT = 0x1
- csi21\_dx0:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0[4]  
CSI21\_DX0\_PULLTYPESELECT = 0x1
- csi21\_dy0:
  - CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0[20]  
CSI21\_DY0\_PULLTYPESELECT = 0x1

3. The CSIPHY is initialized and ready/active in CSI2 mode.

### 8.2.3.2.3 ISS CCP2 and Link Initialization Sequence

The CSI2-B interface is expected to coordinate the PHY initialization. The controller must ensure that the PHY is held in RESET state until the CCP transmitter is powered up and the link comes to the defined state. The controller can use the FORCERXMODE signals of CAMERARX for this purpose. FORCERXMODE forces the receiver state-machine into RESET state, while the rest of common logic is powered up and functional. One possible initialization sequence is:

To fully initialize the CSIPHY, perform the following steps:

1. Configure all CSI2-B receiver registers to be ready to receive signals/data from the CSIPHY:
  - (a) Configure all CCP2 registers:
    - (i) Set `CSI2_COMPLEXIO_CFG[10:8] DATA2_POSITION`.
    - (ii) Set `CSI2_COMPLEXIO_CFG[6:4] DATA1_POSITION`.
  - (b) Set `CONTROL_CAMERA_RX[20:19] CAMERARX_CSI22_CAMMODE` or `CONTROL_CAMERA_RX[17:16] CAMERARX_CSI21_CAMMODE`.

#### CAUTION

This must be done before the CSIPHY is active.

2. CSIPHY and link initialization sequence:
  - (a) Deassert the CSIPHY reset:
    - (i) Set `CSI2_COMPLEXIO_CFG[30] RESET_CTRL` to 0x1.
  - (b) Assert the FORCERXMODE signal:
    - (i) Set `CSI2_TIMING[15] FORCE_RX_MODE_IO1` to 0x1.
  - (c) Power up the CSIPHY:
    - (i) Set `CSI2_COMPLEXIO_CFG[28:27] PWR_CMD` to 0x1.
  - (d) Check that the state status reaches the ON state:
    - `CSI2_COMPLEXIO_CFG[26:25] PWR_STATUS` = 0x1
  - (e) Release the FORCERXMODE signal:
    - (i) Set `CSI2_TIMING[15] FORCE_RX_MODE_IO1` to 0x0.
3. The CSIPHY is initialized and ready/active in CSI1/CCP2B mode.

### 8.2.3.2.4 ISS CSI PHY Error Signals

In D-PHY mode, the CSIPHY supports the following error detection and signaling to the associated receiver:

- **ERRSOTHS:** Flags 1-bit errors in the HS start of transmission synchronization pattern. In this error scenario, the CSIPHY continues to receive the data and pass it to the receiver, but confidence in the data may be low, because of the 1-bit error seen in sync. This signal, if asserted, is high for one cycle of RXBYTECLKHS.
- **ERRSOTSYNCHS:** Flags multiple bit errors in the HS start of transmission synchronization pattern. In this case, the CSIPHY cannot achieve proper synchronization and does not pass the received data to the receiver. This signal, if asserted, is high for one cycle of RXBYTECLKHS.
- **ERRCONTROL:** Flags the control sequence error; that is, when the LP sequence observed on line is not recognized as a valid control sequence. This signal, if asserted, is high until the next change in the state of the LP line.
- **ERRESC:** Flags the escape entry error; that is, when the escape entry sequence is unrecognized. This signal, if asserted, is high until the next change in the state of the LP line.
- **ERRSYNCESC:** Flags the low-power data transmission synchronization error. This error is flagged if the number of bits received during a low-power data transmission is not a multiple of 8 bits. This signal, if asserted, is high until the next change in the state of the LP line. In case the number of received bits is 1 less than a multiple of 8, RXVALIDESC is also asserted together with ERRSYNCESC, and an erroneous data byte is output on RXDATAESC. In other cases of this error, RXVALIDESC is not asserted and an erroneous data byte is not sent out.

In CCP2 mode, the CSIPHY supports the following error detection and signaling to the associated receiver:

- CCPERRRESYNC output is an error indicator to the receiver to flag the shifted synchronization error, if the new synchronization sequence is not aligned with a 32-bit word boundary compared to the previously received synchronization sequence. This error flag is asserted for 32 cycles (4 bytes/32 bits).

### 8.2.3.3 ISS CSI2 PHY Register Manual

#### 8.2.3.3.1 ISS CSI2 PHY Instance Summary

Table 8-40 lists the ISS CSI2 PHY instance.

**Table 8-40. ISS CSI2 PHY Instance Summary**

Module Name	L3 Base Address	Size
ISS_CAMERARX_CORE1	0x5200 1170	32 bytes
ISS_CAMERARX_CORE2	0x5200 1570	32 bytes

**NOTE:** ISS\_CAMERARX\_CORE1 is for CSI2-A CAMERARX.  
ISS\_CAMERARX\_CORE2 is for CSI2-B/CCP2 CAMERARX.

#### 8.2.3.3.2 ISS CSI2 PHY Registers

##### 8.2.3.3.2.1 ISS CSI2 PHY Register Summary

Table 8-41 summarizes the ISS CSI2 register mapping.

**Table 8-41. ISS CSI2 PHY Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CAMERARX_CORE1 Base Address	ISS_CAMERARX_CORE2 Base Address
REGISTER0	RW	32	0x0000 0000	0x5200 1170	0x5200 1570
REGISTER1	RW	32	0x0000 0004	0x5200 1174	0x5200 1574
REGISTER2	RW	32	0x0000 0008	0x5200 1178	0x5200 1578

##### 8.2.3.3.2.2 ISS CSI2 PHY Register Description

Table 8-42 through Table 8-46 describe the ISS CSI2 PHY register bits.

**Table 8-42. REGISTER0**

Address Offset	0x0000 0000
Physical Address	See Table 8-41.
Description	First register
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								THS_TERM								THS_SETTLE							
								HSCLOCKCONFIG																							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved fields	NA	0x00
24	HSCLOCKCONFIG	Disable clock missing detector	RW	0
23:16	RESERVED	Read returns zero	R	0x00
15:8	THS_TERM	THS_TERM timing parameter in multiples of DDR clock Effective time for enabling of termination = synchronizer delay + timer delay + LPRX delay + combinational routing delay ~ (1-2)* DDRCLK + THS-TERM + ~ (1-15) ns Programmed value = ceil(12.5 / DDR clock period) - 1	RW	0x04
7:0	THS_SETTLE	THS_SETTLE timing parameter in multiples of DDR clock frequency Effective THS_SETTLE seen on line (starting to look for sync pattern) = synchronizer delay + timer delay + LPRX delay + combinational routing delay - pipeline delay in HS data path. ~ (1-2)* DDRCLK + THS-SETTLE + ~ (1-15) ns - 1*DDRCLK Programmed value = ceil(90 ns / DDR clock period) + 3	RW	0x27

**Table 8-43. Register Call Summary for Register REGISTER0**

ISS Interfaces

- [ISS CS12 PHY Functional Description: \[0\] \[1\] \[2\]](#)
- [ISS CS12 PHY Register Manual: \[3\]](#)

**Table 8-44. REGISTER1**

<b>Address Offset</b>	0x0000 0004
<b>Physical Address</b>	See <a href="#">Table 8-41</a> .
<b>Description</b>	Second register
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESVD_READ_BIT		RESET_DONE_STATUS		RESERVED		CLOCK_MISS_DETECTOR_STATUS		TCLK_TERM								DPHY_HS_SYNC_PATTERN				CTRLCLK_DIV_FACTOR		TCLK_SETTLE									

Bits	Field Name	Description	Type	Reset
31:30	RESVD_READ_BIT	Reserved bit	NA	0x0
29:28	RESET_DONE_STATUS	Reset done read bits. 28: RESETDONERXBYTECLK <b>Note:</b> BYTECLK is provided to the ISS CS12-A, CS12-B, and CCP2 interfaces 29: RESETDONECTRLCLK <b>Note:</b> This is the CAM_PHY_CTRL_FCLK provided to the PHY from the PRCM module.	R	0x0
27:26	RESERVED	Write 0 for future compatibility.	RW	0x0
25	CLOCK_MISS_DETECTOR_STATUS	1: Error in clock missing detector.  0: Clock missing detector successful	R	0

Bits	Field Name	Description	Type	Reset
24:18	TCLK_TERM	TCLK_TERM timing parameter in multiples of CTRLCLK Effective time for enabling of termination = synchronizer delay + timer delay + LPRX delay + combinational routing delay ~ (1-2)* CTRLCLK + TCLK_TERM + ~ (1-15) ns Programmed value = ceil(9.5 / CTRLCLK period) - 1	RW	0x00
17:10	DPHY_HS_SYNC_PATTERN	DPHY mode HS sync pattern in byte order (reverse of received order) See <a href="#">Section 8.2.3.2.4, ISS CSI PHY Error Signals</a> .	RW	0xB8
9:8	CTRLCLK_DIV_FACTOR	Divide factor for CTRLCLK for CLKMISS detector	RW	0x1
7:0	TCLK_SETTLE	TCLK_SETTLE timing parameter in multiples of CTRLCLK Clock Effective TCLK_SETTLE = synchronizer delay + timer delay + LPRX delay + combinational routing delay ~ (1-2)* CTRLCLK + Tclk-settle + ~ (1-15) ns Programmed value = max[3, ceil(155 ns/CTRLCLK period) -1]	RW	0x0E

**Table 8-45. Register Call Summary for Register REGISTER1**

ISS Interfaces

- [ISS CSI2 PHY Functional Description: \[0\]](#)
- [ISS CSI2 PHY Register Manual: \[1\]](#)

**Table 8-46. REGISTER2**

<b>Address Offset</b>	0x0000 0008
<b>Physical Address</b>	See <a href="#">Table 8-41</a> .
<b>Description</b>	Third register
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
TRIGGER_CMD_RXTRIGESC0	TRIGGER_CMD_RXTRIGESC1	TRIGGER_CMD_RXTRIGESC2	TRIGGER_CMD_RXTRIGESC3	CCP2_SYNC_PATTERN																														

Bits	Field Name	Description	Type	Reset
31:30	TRIGGER_CMD_RXTRIGESC0	Mapping of Trigger escape entry command to PPI output RXTRIGGERESC0	RW	0x0
29:28	TRIGGER_CMD_RXTRIGESC1	Mapping of Trigger escape entry command to PPI output RXTRIGGERESC1	RW	0x0
27:26	TRIGGER_CMD_RXTRIGESC2	Mapping of Trigger escape entry command to PPI output RXTRIGGERESC2	RW	0x0
25:24	TRIGGER_CMD_RXTRIGESC3	Mapping of Trigger escape entry command to PPI output RXTRIGGERESC3	RW	0x0
23:0	CCP2_SYNC_PATTERN	CCP2 mode sync pattern in byte order See <a href="#">Section 8.2.3.2.4, ISS CSI PHY Error Signals</a> .	R	0x0000FF

**Table 8-47. Register Call Summary for Register REGISTER2**

ISS Interfaces
<ul style="list-style-type: none"> <li>• <a href="#">ISS CSI2 PHY Functional Description: [0]</a></li> <li>• <a href="#">ISS CSI2 PHY Register Manual: [1]</a></li> </ul>

## 8.2.4 ISS CCP2

### 8.2.4.1 ISS CCP2 Environment

#### 8.2.4.1.1 ISS CCP2 Protocol and Data Formats

This section describes the CCP2 protocol and data formats. The CCP2 receiver is compatible with the CCP2 specification and the MIPI CSI1 specification. [Table 8-48](#) describes the I/O for serial interface CCP2.

**Table 8-48. I/O Description for Serial Interface CCP2**

Signal Name		I/O <sup>(1)</sup>	Description
ccp2_datp	CAM_S_DATA	I	Serial data input. Configurable.
ccp2_datn			
ccp2_strbp	CAM_S_CLK	I	Serial clock or strobe input. Configurable.
ccp2_strbn			

<sup>(1)</sup> I = Input; O = Output

From the device point of view, the CCP2 interface comprises four differential inputs representing two effective signals: the serial data and the clock.

The CCP2 receiver is a serial interface to an image sensor. Signals from the camera are serial input data (CAM\_S\_DATA) and a clock/strobe signal (CAM\_S\_CLK).

#### 8.2.4.1.1.1 ISS CCP2 Synchronization Codes

Each frame is clearly identified by four unique 32-bit synchronization codes: frame start, frame end, line start, and line end, which are embedded in the serial bitstream. The logical channel identification number is also encoded in the synchronization codes.

- Frame-start code (FSC): Identifies the start of a new frame
- Line-start code (LSC): Identifies the start of a new line and is received for every line, except the first line, that starts with an FSC
- Line-end code (LEC): Identifies the end of a line and is received for every line, except the last line, that ends with an FEC
- Frame-end code (FEC): Identifies the end of the last line and the end of the current frame

#### 8.2.4.1.1.2 ISS CCP2 False Synchronization Code Protection

CCP2 supports false synchronization protection (FSP). Additional bytes after the JPEG stream can be written to the SDRAM. They do not prevent JPEG encode and there is no need to preserve them.

#### 8.2.4.1.1.3 ISS CCP2 Image Data Operating Modes and Alignment Constraints

The CCP2 receiver interface has several image data operating modes (see [Table 8-49](#)). The EXP<sub>x</sub> formats (where x = 8, 16, or 32) are used to expand data up to 8, 16, or 32 bits by padding data with zeros. The Storage Increase Versus Packed column indicates memory overhead versus format without data expansion and/or DPCM compression.

**Table 8-49. ISS CCP2 Image Data Operating Modes and Alignment Constraints**

CCP2_LCx_CT RL[7:2] Format	CCP2 Data Format	Bits per Pixel (bpp) (When sending data to memory, N/A when sending to VP)	Data Size Increase in Memory. (When negative, data compression present)	2D Mode Availability <sup>(1)</sup>	Comments
0x0	YUV4:2:2 big endian	16	0%	Yes	
0x1	YUV4:2:2 little endian	16	0%	Yes	
0x2	YUV4:2:0	12	0%	Yes	
0x3	YUV4:2:2 + VP	N/A, data are sent to VP, YUV4:2:2 + VP = RAW8 + VP	N/A	Yes	
0x3	RAW8 + VP	N/A, data are sent to VP, YUV4:2:2 + VP must be used to output RAW8 + VP to memory	N/A	Yes	
0x4	RGB444 + EXP16	16	50%	Yes	
0x5	RGB565	16	0%	Yes	
0x6	RGB888	24	0%	Yes	
0x7	RGB888 + EXP32	32	33%	Yes	
0x8	RAW6 + EXP8	8	33%	Yes	
0x9	RAW6 + DPCM10 + EXP16	16	167%	Yes	DPCM decompression
0xA	RAW6 + DPCM10 + VP	N/A, data are sent to VP	N/A	Yes	DPCM decompression
0xB	RAW10 -> RAW6 DPCM	6	40%	Yes	DPCM compression
0xC	RAW7 + EXP8	8	14%	Yes	
0xD	RAW7 + DPCM10 + EXP16	16	128%	Yes	DPCM decompression
0xE	RAW7 + DPCM10 + VP	N/A, data are sent to VP	N/A	Yes	DPCM decompression
0xF	RAW10 -> RAW6 DPCM + EXP8	8	25%	Yes	DPCM compression
0x10	RAW8, this mode can be used to output RAW6 and RAW7	8	0%	Yes	
0x11	RAW8 + DPCM10 + EXP16	16	100%	Yes	DPCM decompression
0x12	RAW8 + DPCM10 + VP	N/A, data are sent to VP	N/A	Yes	DPCM decompression
0x13	RAW10 -> RAW7 DPCM	7	30%	Yes	DPCM compression
0x14	RAW10	10	0%	Yes	
0x15	RAW10 + EXP16	16	60%	Yes	
0x16	RAW10 + VP	N/A, data are sent to VP	N/A	Yes	
0x17	RAW10 -> RAW7 DPCM + EXP8	8	20%	Yes	
0x18	RAW12	12	0%	Yes	
0x19	RAW12 + EXP16	16	33%	Yes	
0x1A	RAW12 + VP	N/A, data are sent to VP	N/A	Yes	

<sup>(1)</sup> If 2D mode is available, there are no supplementary constraints on data width. 2D mode does not apply when sending to the video port (VP).

**Table 8-49. ISS CCP2 Image Data Operating Modes and Alignment Constraints (continued)**

<b>CCP2_LCx_CTL RL[7:2] Format</b>	<b>CCP2 Data Format</b>	<b>Bits per Pixel (bpp) (When sending data to memory, N/A when sending to VP)</b>	<b>Data Size Increase in Memory. (When negative, data compression present)</b>	<b>2D Mode Availability <sup>(1)</sup></b>	<b>Comments</b>
0x1B	RAW10 -> RAW8 DPCM	8	20%	Yes	DPCM decompression
0x1C	JPEG, 8-bit data	N/A	0%	Yes	
0x1D	JPEG, 8-bit data + FSP	N/A	0%	Yes	
0x1E	RAW10 -> RAW8 DPCM	8	20%	Yes	Data right shift
0x1F	RAW8 DPCM12-> RAW12 + VP	N/A, data are sent to VP	N/A	Yes	
0x20	RAW10 -> RAW8 ALAW	8	20%	Yes	
0x21	RAW8 DPCM10 -> ALAW	8	20%	Yes	

**NOTE:**

- Padding data of a 32-bit pixel data stream is handled the same way regardless of the programmed format. Therefore, there is no increase or decrease in storage because it is not compressed or decompressed.
- EXP8 = Data expansion to 8 bits, padding with zeros
- EXP16 = Data expansion to 16 bits, padding with alpha or zeros  
CCP2\_LCx\_CTRL[15:8] ALPHA can be used to set an alpha value.  
For RGB444 + EXP16:
  - data\_out[31:28] = ALPHA [3:0]
  - data\_out[15:12] = ALPHA [3:0]
- EXP32 = Data expansion to 32 bits, padding with alpha  
CCP2\_LCx\_CTRL[15:8] ALPHA can be used to set an alpha value.  
For RGB888 + EXP32: data\_out[31:24] = ALPHA [7:0]
- FSP = False synchronization code protection decoding. Applies only to JPEG8 data format.
- VP = Output to the video-preprocessing hardware is enabled. Programmers must ensure that only one logical channel is enabled to the video preprocessing hardware. The behavior of the hardware is unpredictable if several logical channels to the video preprocessing hardware are enabled simultaneously.
- DPCM10 = Data decompression to 10 bits. Applies only to RAW6, RAW7, and RAW8 data formats; disabled if CCP2\_CTRL[4] MODE = 0.
- Padding is handled the same way regardless of the programmed format.

**NOTE:** Data written by CSI2 can be read back by the CCP2 read channel. Some constrains apply (see Table 8-52).

**8.2.4.1.1.4 ISS CCP2 Pixel Data Format**

This section summarizes how the CCP2 pixel data formats are transmitted over the serial interface and how the pixels are reconstructed, stored in memory, or passed to the video port.

The CCP2 receiver can cope with all data formats if the data line length sent through the CS1/CCP2 physical protocol is a multiple of 32 bits. This condition is required for the CCP2 receiver to work correctly.



However, some data formats impose stronger line-length constraints to finish pixel reconstruction correctly at the end of the lines. This is imposed by CCP2 protocol-specific requirements, and not the CCP2 receiver. If CCP2 protocol constraints are not respected:

- Only the pixels reconstructed last in every line are erroneous. The missing bits are replaced with zeros to perform pixel reconstruction.
- The FW\_IRQ interrupt is triggered.

#### 8.2.4.1.1.4.1 ISS CCP2 YUV Operating Modes

**NOTE:** Although there are different possibilities of endianness in the bitstream, the device as a whole typically works in little-endian format. Therefore, the use of little-endian format is recommended.

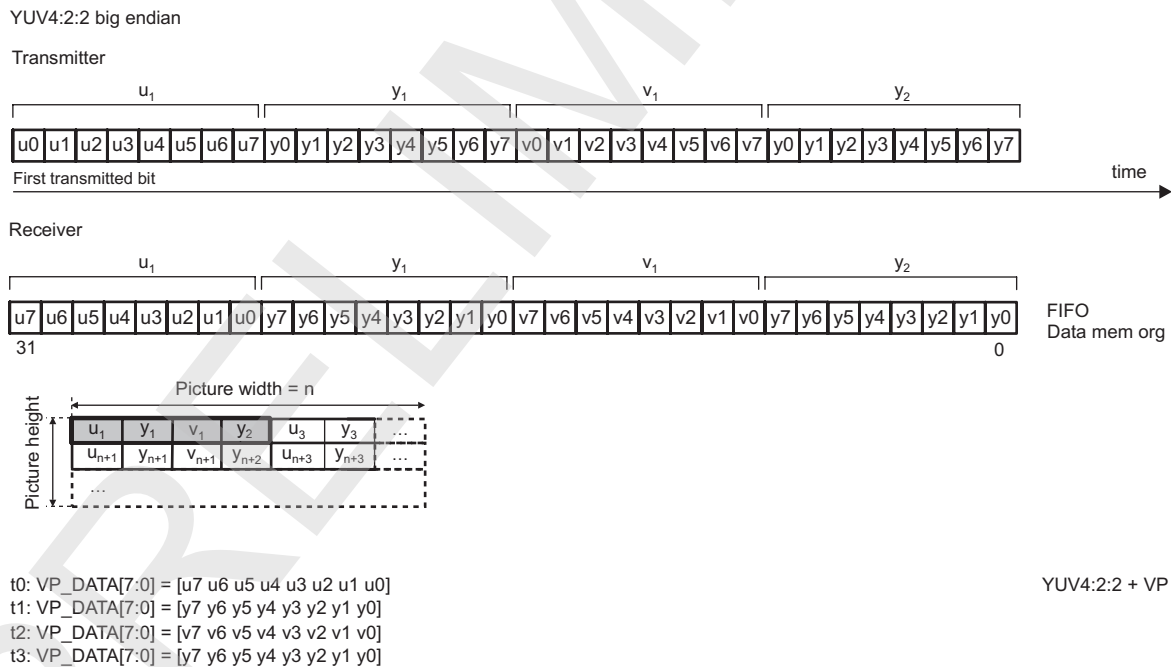
##### 8.2.4.1.1.4.1.1 ISS CCP2 YUV4:2:2

The YUV4:2:2 data format can be stored to memory in little- or big-endian format. The line length sent through the CCP2 receiver protocol must be a multiple of 32 bits.

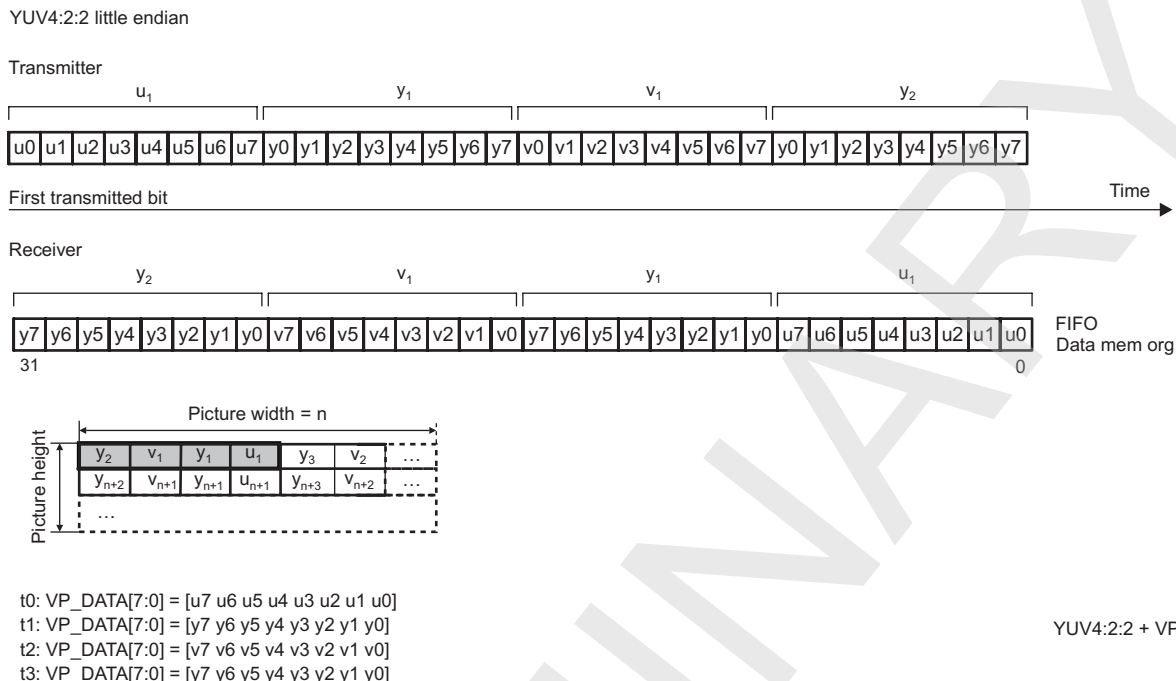
YUV4:2:2 data format can also be sent to the video port.

YUV4:2:2 + VP is used to output RAW8 data to the video port: YUV4:2:2 + VP is equivalent to RAW8 + VP. Figure 8-10 and Figure 8-11 show big-endian and little-endian YUV4:2:2 format, respectively. Set CCP2\_LCx\_CTRL[7:2] FORMAT to 0x0 to select YUV4:2:2 little-endian mode and to 0x1 for YUV4:2:2 big-endian mode.

Figure 8-10. ISS CCP2 YUV4:2:2 Big Endian



**Figure 8-11. ISS CCP2 YUV4:2:2 Little Endian**

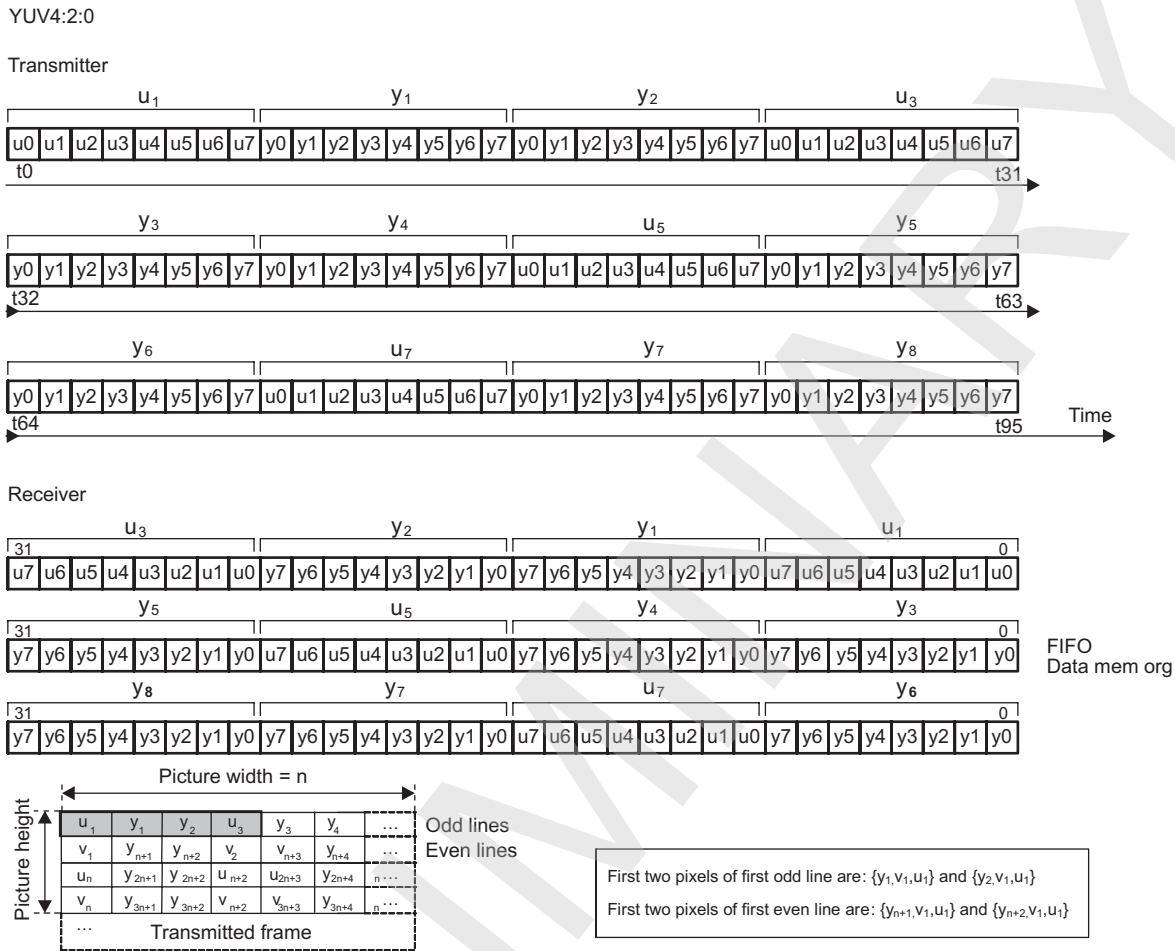


**8.2.4.1.1.4.1.2 ISS CCP2 YUV4:2:0**

The line length sent through the CCP2 receiver protocol is a multiple of 32 bits. Furthermore, the line length all together sent by the CCP2 receiver is a multiple of 3 x 32 bits and the number of lines is even to correctly finish the pixel reconstruction.

The line structure is different for odd and even lines. Odd lines transport the U component, while even lines contain the V component. This is shown in Figure 8-12. Set CCP2\_LCx\_CTRL[7:2] FORMAT to 0x3 to select YUV4:2:0 mode.

Figure 8-12. ISS CCP2 YUV4:2:0



camss-053

### 8.2.4.1.1.4.2 ISS CCP2 RGB Operating Modes

#### 8.2.4.1.1.4.2.1 ISS CCP2 RGB888

RGB888 data format can be output to memory in two formats: without data expansion and with data expansion.

If data expansion is used, the value of the 8 upper bits is programmable and can be set with an alpha value for computer graphics applications. The line length sent through the CCP2 receiver protocol is a multiple of 32 bits. The line length all together sent by the CCP2 receiver is a multiple of 3 x 32 bits to finish pixel reconstruction correctly.

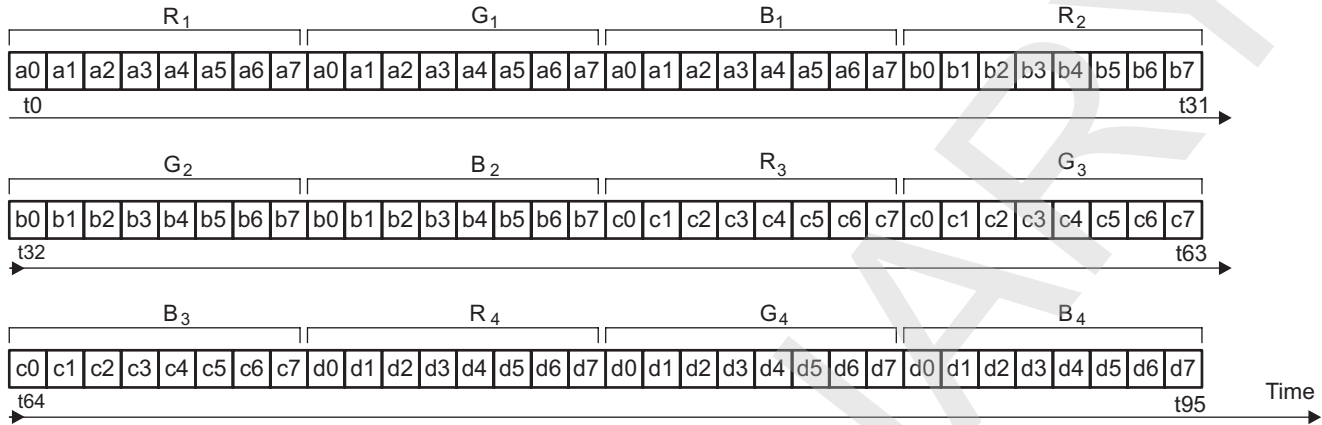
Figure 8-13 is an example of RGB888 format. Set `CCP2_LCx_CTRL[7:2] FORMAT` to 0x6 to select RGB888 mode (24 bits) or to 0x7 to expand RGB888 over 32 bits.

**Figure 8-13. ISS CCP2 RGB888**

RGB888

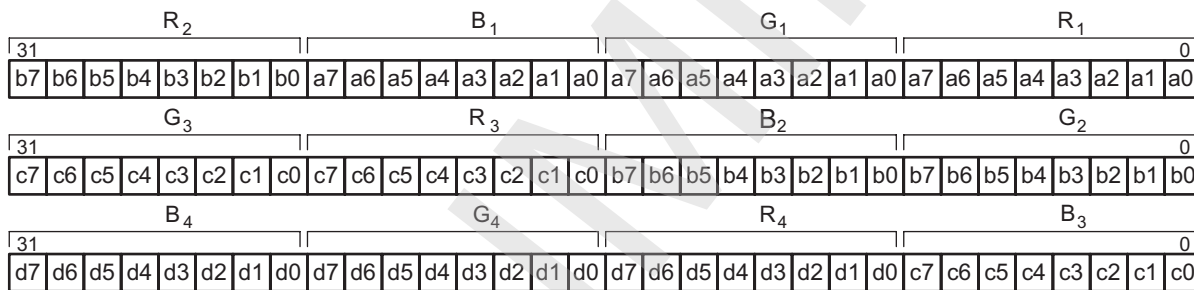
Line width must be a multiple of three 32-bit words.

Transmitter

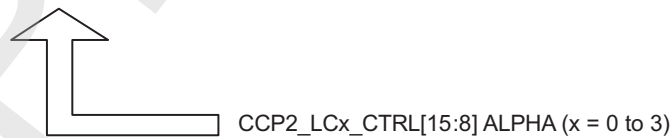
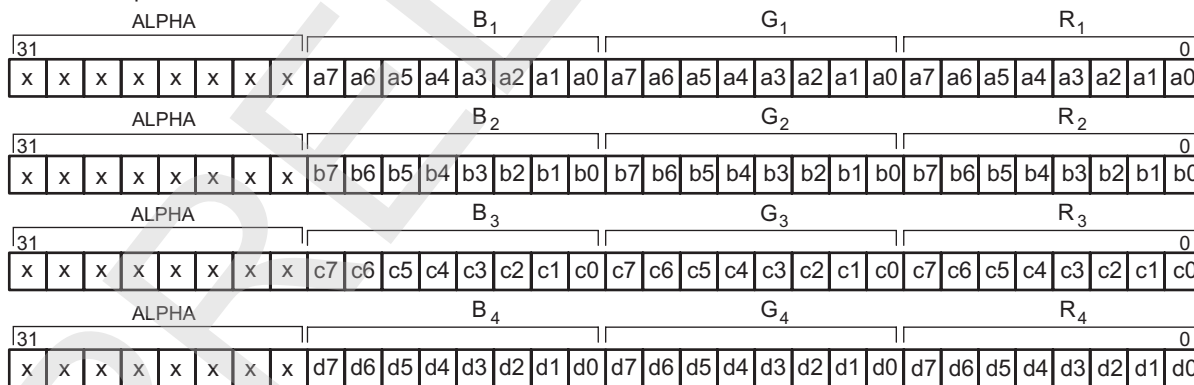


Receiver

Without data expansion



With data expansion

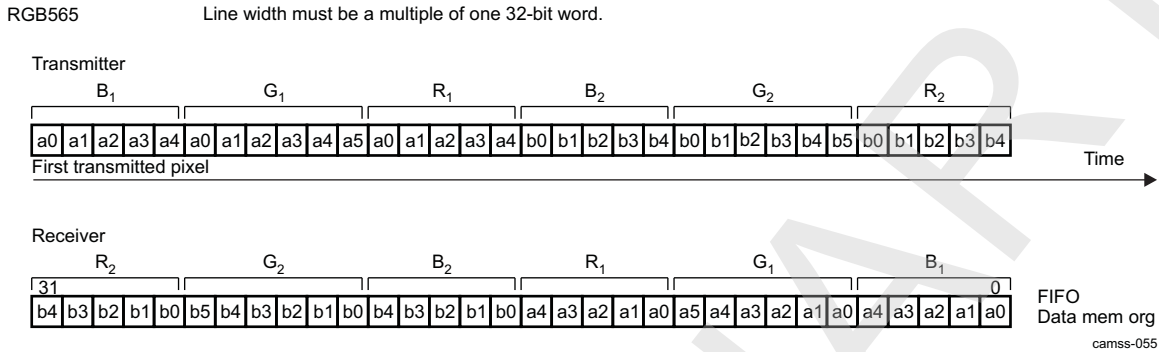


camss-054

### 8.2.4.1.1.4.2.2 ISS CCP2 RGB565

For RGB565, the line length sent through the CCP2 receiver protocol is a multiple of 32 bits (see Figure 8-14). Set `CCP2_LCx_CTRL[7:2] FORMAT` to 0x6 to select RGB565 mode.

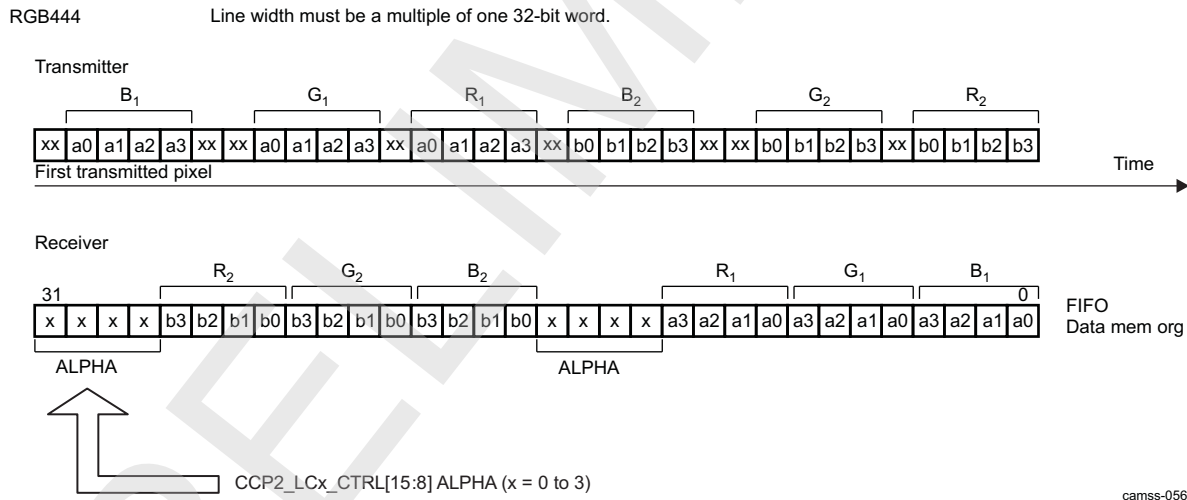
Figure 8-14. ISS CCP2 RGB565



### 8.2.4.1.1.4.2.3 ISS CCP2 RGB444

RGB444 data format is output to memory with data expansion. If data expansion is used, the value of the 4 upper bits is programmable and can be set with an alpha value for computer graphics applications. The line length sent through the CCP2 receiver protocol is a multiple of 32 bits (see Figure 8-15). Set `CCP2_LCx_CTRL[7:2] FORMAT` to 0x4 to select RGB444 mode (12 bits expanded to 16).

Figure 8-15. ISS CCP2 RGB444



### 8.2.4.1.1.4.3 ISS CCP2 RAW Bayer Operating Modes

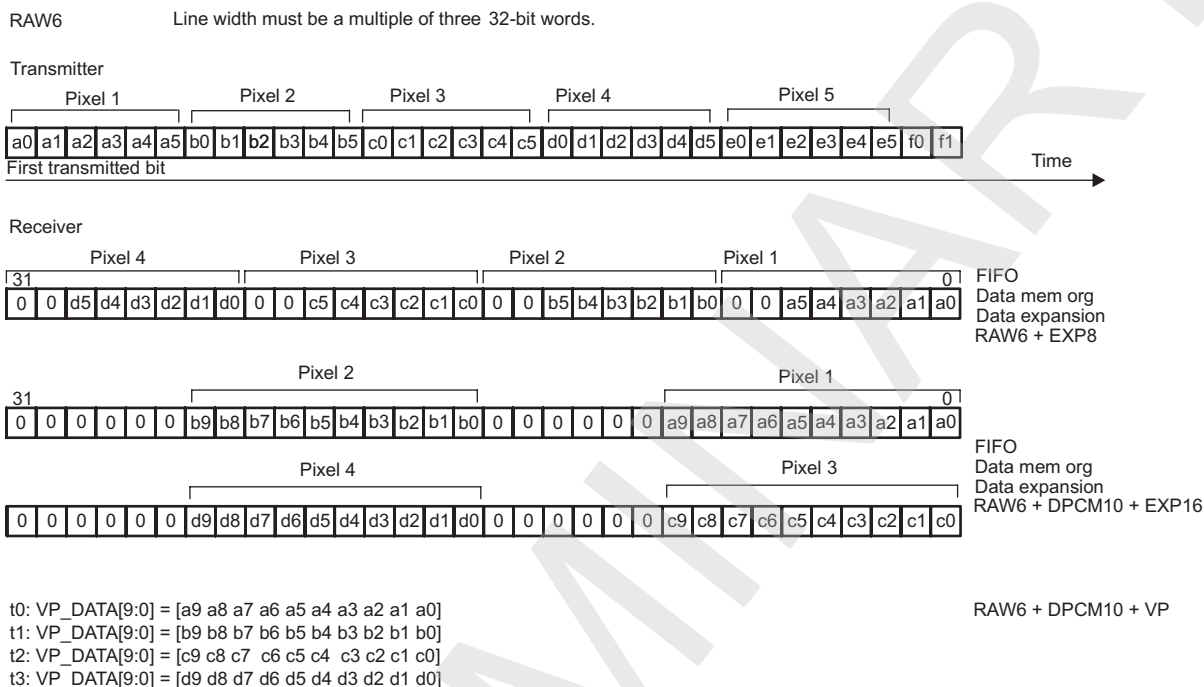
**NOTE:** For more information about packed RAW data, see Section 8.2.4.3.5.1, ISS CCP2 Read Data From Memory.

#### 8.2.4.1.1.4.3.1 ISS CCP2 RAW6

RAW6 data format can be output to memory in two formats: without data expansion and with data expansion. The line length sent through the CSI receiver protocol is a multiple of 32 bits. The line all together sent by the CCP2 receiver is a multiple of 3 x 32 bits to finish pixel reconstruction correctly (the lowest common multiple of 32 and 6 is 96; that is, 3 x 32 bits).

Figure 8-16 shows the RAW6 format. Set `CCP2_LCx_CTRL[7:2] FORMAT` to:

- 0x8 to select RAW6 mode expanded to 8-bit
- 0x9 to select RAW6 mode with DPCM decompression
- 10-bit and expansion to 16-bit
- 0xA to select RAW6 mode with DPCM decompression to 10-bit to video port

**Figure 8-16. ISS CCP2 RAW6**

**NOTE:** Use RAW8 data format to output RAW6 data format to memory.

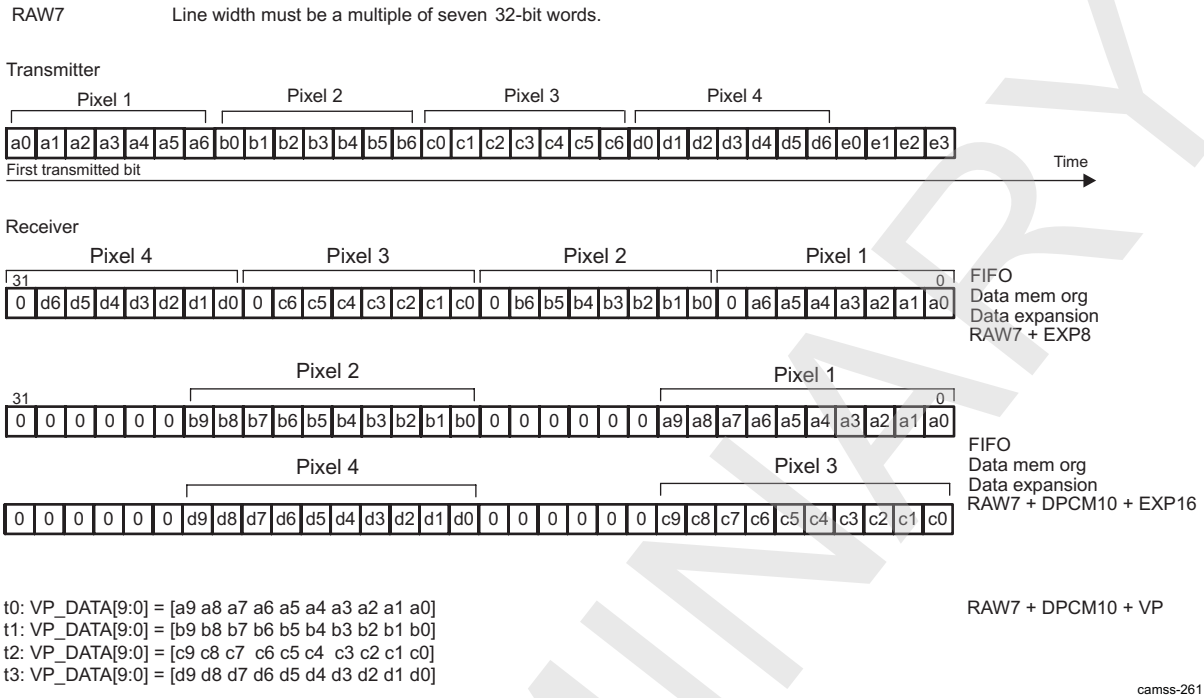
#### 8.2.4.1.1.4.3.2 ISS CCP2 RAW7

RAW7 data format can be output to memory in two formats: without data expansion and with data expansion. The line length sent through the CCP2 receiver protocol is a multiple of 32 bits. The line length all together sent by the CCP2 receiver is a multiple of 7 x 32 bits to finish the pixel reconstruction correctly (the lowest common multiple of 32 and 7 is 224; that is, 7 x 32 bits).

Figure 8-17 shows the RAW7 format. Set `CCP2_LCx_CTRL[7:2]` FORMAT to:

- 0xC to select RAW7 mode expanded to 8-bit
- 0xD to select RAW7 mode with DPCM decompression to 10-bit and expansion to 16-bit
- 0xE to select RAW7 mode with DPCM decompression to 10-bit to video port

**Figure 8-17. ISS CCP2 RAW7**



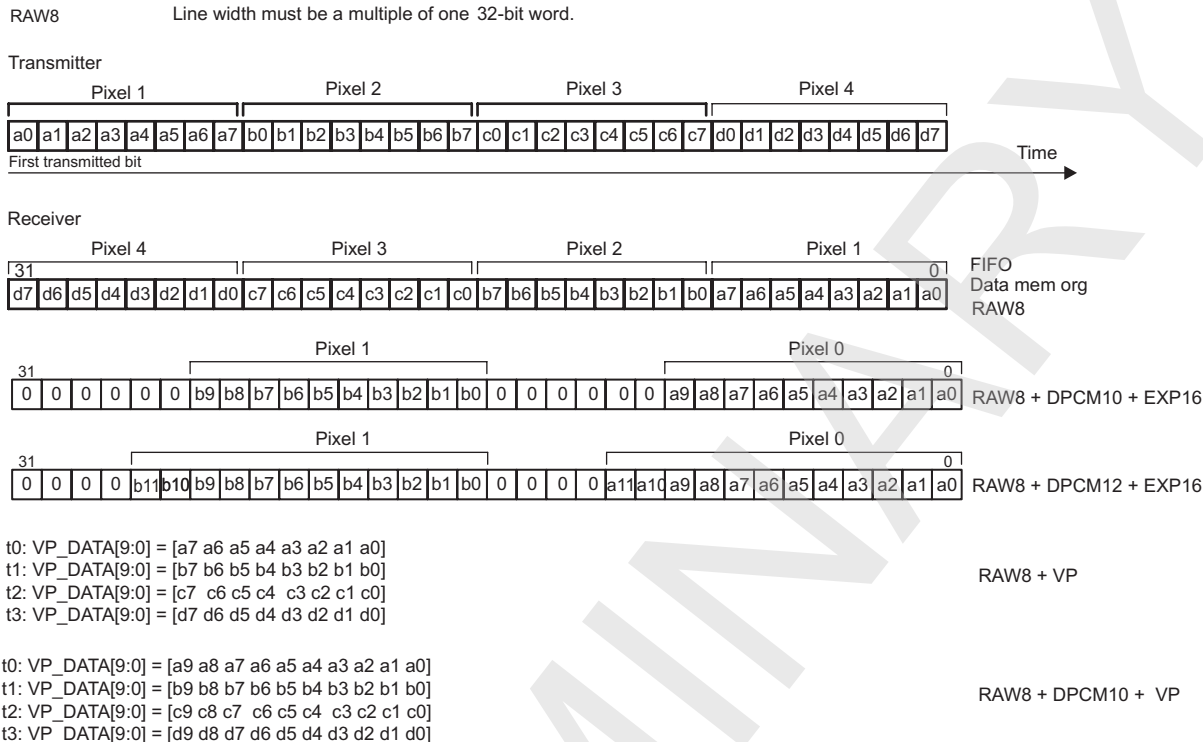
**NOTE:** Use RAW8 data format to output RAW7 data format to memory.

**8.2.4.1.1.4.3.3 ISS CCP2 RAW8**

RAW8 data format can be output to memory. The line length sent through the CCP2 receiver protocol is a multiple of 32 bits.

Figure 8-18 shows RAW8 format. Set CCP2\_LCx\_CTRL[7:2] FORMAT to:

- 0x10 to select RAW8
- 0x11 to select RAW8 mode with DPCM decompression to 10-bit and expansion to 16-bit
- 0x12 to select RAW8 mode with DPCM decompression to 10-bit to video port
- 0x20 to select RAW8 with A-Law decompression to RAW10

**Figure 8-18. ISS CCP2 RAW8**

**NOTE:** Use YUV4:2:2 + VP to output RAW8 data to the video port: YUV4:2:2 + VP is equivalent to RAW8 + VP.

#### 8.2.4.1.1.4.3.4 ISS CCP2 RAW10

RAW10 data format can be output to memory in two formats: without data expansion and with data expansion. If data expansion is used, the 10-bit data are padded with 0s on a 16-bit word. The line length sent through the CCP2 receiver protocol is a multiple of 32 bits. Furthermore, the line length all together sent by the CCP2 receiver is a multiple of 5 x 32 bits to finish pixel reconstruction correctly (the lowest common multiple of 32 and 10 is 320; that is, 10 x 32 bits).

RAW10 data format can be sent to the video port. [Figure 8-19](#) shows the RAW10 format. Set `CCP2_LCx_CTRL[7:2]` FORMAT to:

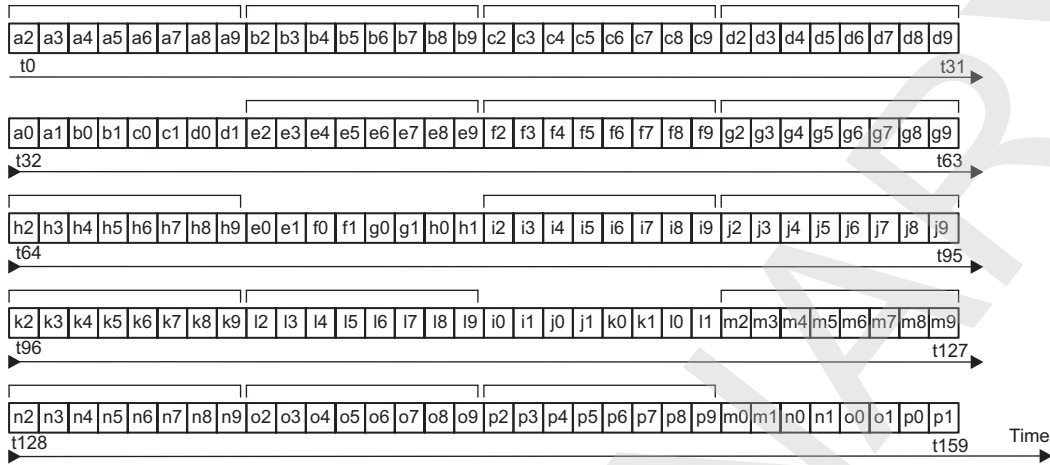
- 0x14 to select RAW10
- 0x13 to select RAW10 mode with DPCM compression to 7-bit
- 0x16 to select RAW10 mode to video port
- 0x17 with DPCM compression to 7-bit with expansion to 8-bit
- 0x1B to select RAW10 compressed to 8-bit
- 0xB with DPCM compression to 6-bit
- 0xF to select RAW10 compressed to 6-bit with expansion to 8-bit
- 0x1E to select RAW10 compressed to 8-bit
- 0x20 to select RAW10 with A-Law compression to RAW8



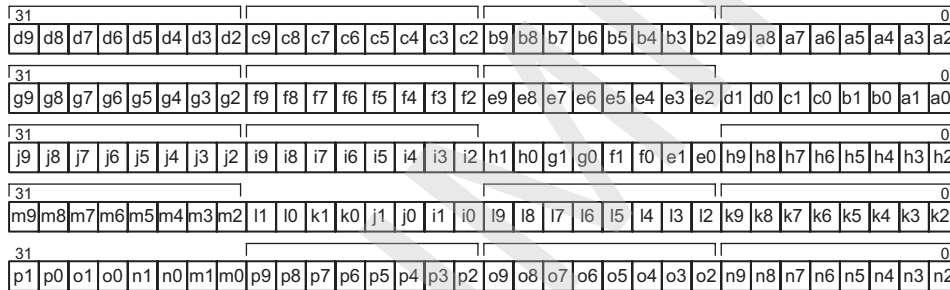
Figure 8-19. ISS CCP2 RAW10

RAW10 Line width must be a multiple of five 32-bit words.

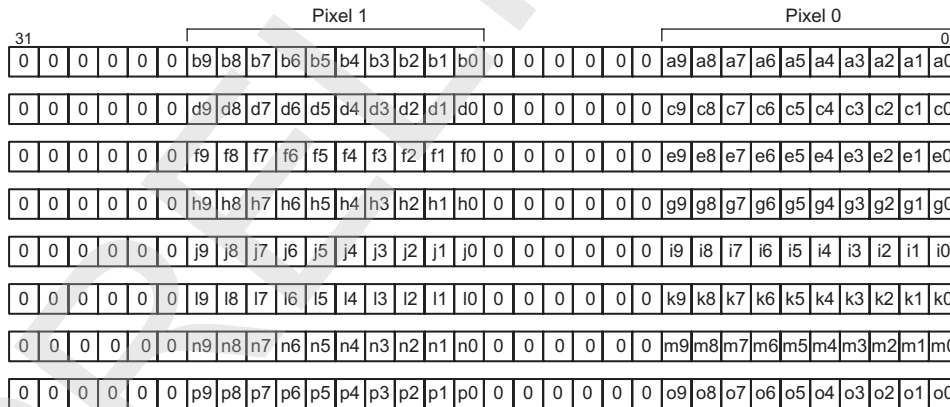
Transmitter



Receiver



FIFO  
Data mem org  
No data expansion  
RAW10



FIFO  
Data mem org  
Data expansion  
RAW10 + EXP16

t0: VP\_DATA[9:0] = [a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]  
t1: VP\_DATA[9:0] = [b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]  
t2: VP\_DATA[9:0] = [c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]  
t3: VP\_DATA[9:0] = [d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

RAW10 + VP

camss-060

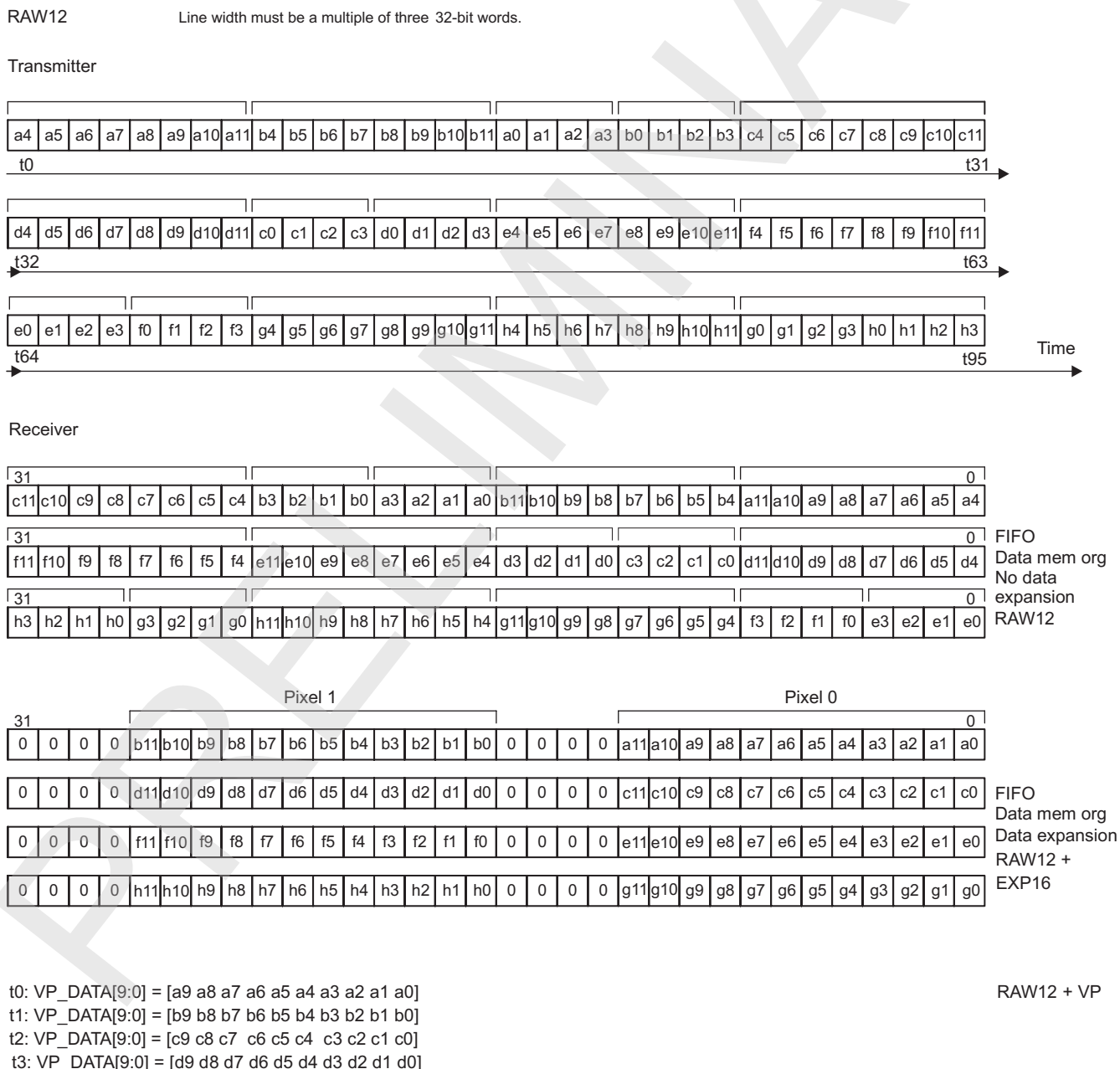
### 8.2.4.1.1.4.3.5 ISS CCP2 RAW12

RAW12 data format can be output to memory in two formats: without data expansion and with data expansion. If data expansion is used, the 12-bit data are padded with 0s on a 16-bit word. The line length sent through the CCP2 PHY is a multiple of 32 bits. The line length is a multiple of 3 x 32 bits to finish pixel reconstruction correctly (the lowest common multiple of 32 and 12 is 96; that is, 3 x 32 bits).

RAW12 data format can be sent to the video port. Figure 8-20 shows RAW12 format. Set CCP2\_LCx\_CTRL[7:2] FORMAT to:

- 0x18 to select RAW12
- 0x19 to select RAW12 mode with expansion to 16-bit
- 0x1A to select RAW12 mode to video port
- 0x17 with DPCM compression to 7-bit with expansion to 8-bit

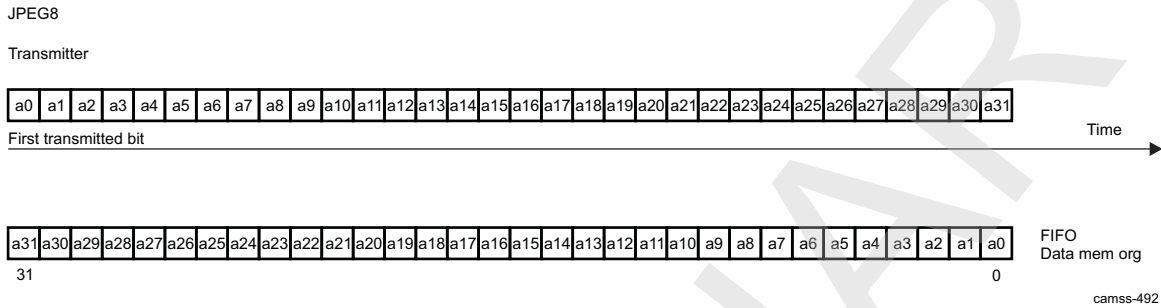
Figure 8-20. ISS CCP2 RAW12



**8.2.4.1.1.4.4 ISS CCP2 JPEG8 Operating Modes**

The line length sent through the CCP2 receiver protocol is a multiple of 32 bits. The FSP code insertion on the transmitter side automatically ensures this line length. It is impossible to know in advance the size of a compressed stream. [Figure 8-21](#) shows the JPEG8 and JPEG8 FSP format. Set `CCP2_LCx_CTRL[7:2] FORMAT` to 0x1C to select JPEG8 format, or to 0x1D to select JPEG8 format with FSP.

**Figure 8-21. ISS CCP2 JPEG8 and JPEG8 FSP**



In JPEG8 mode, the JPEG encoder on the sensor side must avoid generating data equal to the synchronization code and must deliver a synchronization-code-free bitstream for line-start and line-end before and after the transmitted data. For more information, see [Section 8.2.4.1.1.2, ISS CCP2 False Synchronization Protection Code](#).

**8.2.4.1.1.5 ISS CCP2 Data Transfer Through Write Master Port**

The CCP2 receiver module uses its 64-bit master interface to transfer the data stored in the FIFO. The system can set the burst size to be used in the `CCP2_CTRL BURST` register. The recommended burst size is 128 bytes (0x4: 16 x 64-bit bursts are the most efficient for memory output); other sizes should be used only when issues are faced. The module tries to burst data whenever possible to increase system performance and save power. The module uses the appropriate register-selected burst size when the conditions (address is properly aligned, there is enough data in the FIFO to finish the line) are met; otherwise, it uses single requests.

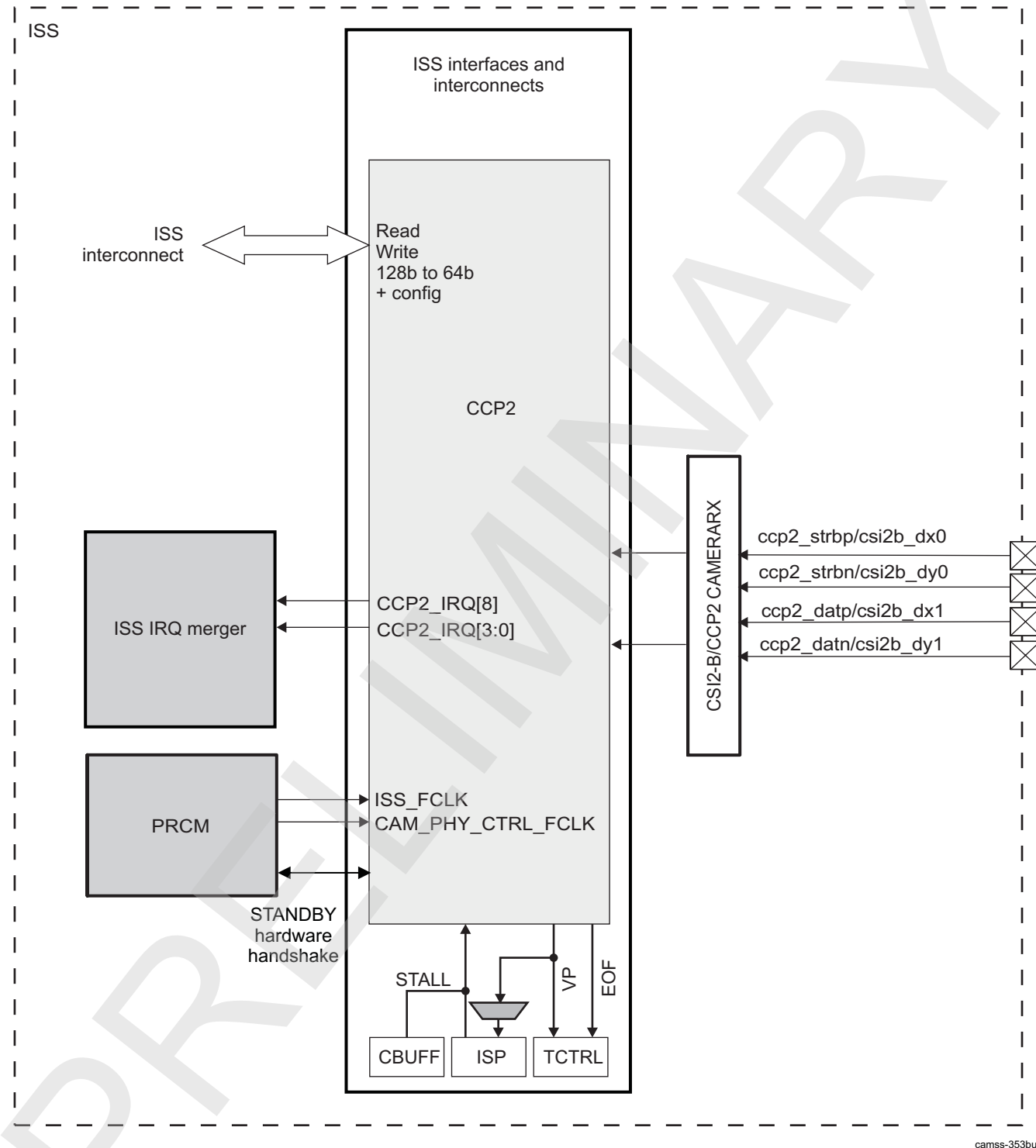
For example, if the line consists of 50 pixels of 64-bit data and `BURST = 0x4` (16 x 64-bit bursts), the module generates four times 16 x 64-bit bursts to transmit the whole line. The FIFO can have information about only two different lines.

The CCP2 engine supports tag count. The count is configurable through a register at the top level of the ISS (see [Section 8.1.3, ISS Register Manual](#)).

**8.2.4.2 ISS CCP2 Integration**

[Figure 8-22](#) is an overview of the integration of the CCP2 interface in the device. The figure shows the CCP2 receiver top-level block diagram. The CCP2 receiver receives serial data from a CCP2-compatible image sensor, converts it to parallel data, extracts the logical channels, detects and extracts the synchronization codes, reformats the data, and outputs it through the VP interface (which is connected to the video preprocessing hardware [ISP]) or the ISS interconnect. The CCP2 can also get its data from memory using a dedicated interface read master port when this feature is enabled at compile time.

Figure 8-22. ISS CCP2 Integration



camss-353bu

**NOTE:** Because CCP2 and CS12-B share pins, they cannot be used simultaneously. For ISP source select, see [Section 8.1.1, ISS Integration](#), for ISP registers input selection and for top-level input select settings.

For power domain, clocks, reset, and hardware requests, see [Section 8.1.2.4, ISS Power Management](#).

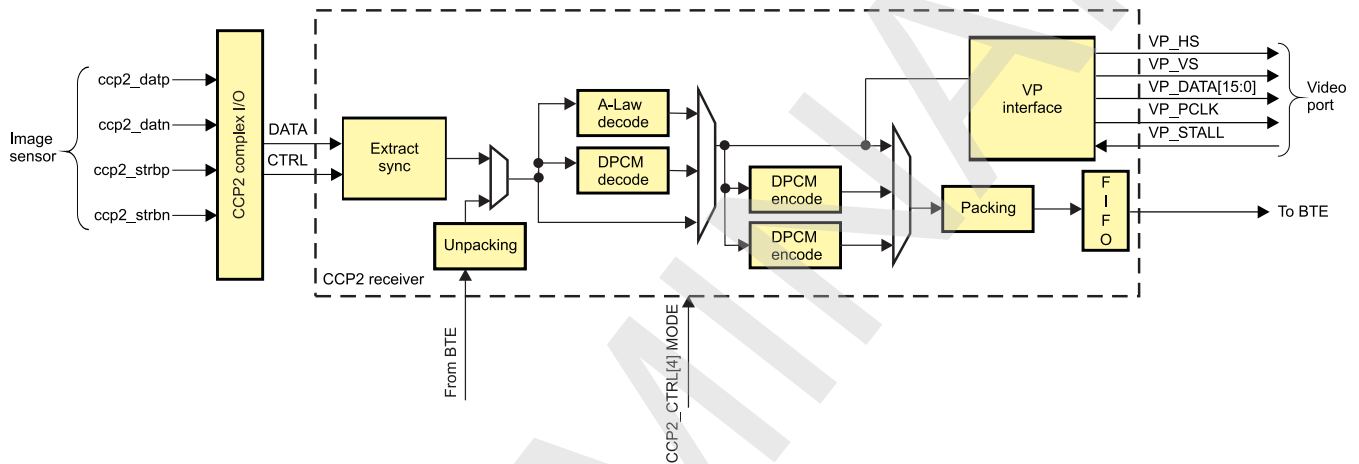
### 8.2.4.3 ISS CCP2 Functional Description

#### 8.2.4.3.1 ISS CCP2 Overview

Figure 8-23 is the top-level block diagram of the CCP2 receiver. The CCP2 receiver receives serial data from a CCP2-compatible image sensor, converts it to parallel data, extracts the logical channels, detects and extracts the synchronization codes, reformats the data, and outputs it through the VP interface (which is connected to the video preprocessing hardware [ISP]) or the ISS interconnect. The CCP2 can also read data from memory getting burst packets from the BTE when this feature has been enabled at compile time. For read data from memory, see Section 8.2.4.3.5, *ISS CCP2 Memory Read Channel*.

The CCP2 receiver video port interface is connected to the video preprocessing hardware.

Figure 8-23. ISS CCP2 Receiver Block Diagram



camss-493

#### 8.2.4.3.2 ISS CCP2 PHY

The CCP2 receiver does not connect directly to a sensor (see Section 8.2.3, *ISS CSI2 PHY*). It uses a complex I/O physical layer. The CCP2 complex I/O is shared with the CSI2-B receiver. The CSI2-B/CCP2 PHY is powered up by configuring the CSI2 receiver configuration register. For more information about the configuration of the PHY, see Section 8.2.3.2.1, *ISS CSI2 PHY Functional Configuration*. The CCP2 serial interface is a unidirectional differential serial interface with two options for the PHY: data/clock or data/strobe signals. After the PHY is configured and initialization is done, the CCP2 mode can be selected and controlled by the CCP2\_CTRL[1] MODE bit.

##### 8.2.4.3.2.1 ISS CCP2 Data/Clock Signaling

Data/clock signaling consists of two parallel signals: data and data clock.

- The data signal carries bit serial data. The CCP2 transmitter writes the data on each falling edge of the clock. The data are usually transmitted byte-wise, least-significant bit (LSB) first.
- The data clock signal carries the clock signal. The CCP2 transmitter writes the data on each falling edge of the clock. The CCP2 receiver reads the data on the rising edge of the clock.

##### 8.2.4.3.2.2 ISS CCP2 Data/Strobe Signaling (CCP2 Only)

Data/strobe signaling consists of two parallel signals: data and data strobe.

- The data signal carries bit serial data.
- The data strobe signal carries the strobe signal. It toggles when the data signal does not change state.

The data signal or the strobe signal changes between 2 data bits. The data and strobe signals must not change simultaneously. The data and strobe signals are used in the receiver to reconstruct the transmission clock. Both fronts of the reconstructed clock are used to sample the data. Set the [CCP2\\_CTRL\[4\] MODE](#) bit and see [Section 8.2.3, ISS CS/2 PHY](#), to set the data/clock or data/strobe function.

### 8.2.4.3.3 ISS CCP2 VP Interface

[Table 8-50](#) summarizes the video interface signals. The video interface connects the CCP2 receiver module to the video preprocessing hardware (ISP). The interface is connected to a 16-bit video port. On the other side of the video port is the ISIF inside the ISP. The ISIF also uses the signals listed in [Table 8-50](#) to synchronize pixel data sent to it by the CCP2 receiver.

**Table 8-50. ISS CCP2 Video interface Signals**

Pin	Type <sup>(1)</sup>	Description
VP_HS	O	Line trigger output signal
VP_VS	O	Frame trigger output signal
VP_DATA[15:0]	O	Parallel output data: bits 0 to 15
VP_PCLK	O	Video port pixel clock. The frequency can be configured.
VP_STALL	I	Stalls data flow when data is read from memory

<sup>(1)</sup> I = Input; O = Output

When data is read from memory and sent to the video port, the data flow can be stalled by asserting the VP\_STALL signal. Doing so does not overflow internal FIFOs: the CCP2 module adapts its read rate automatically.

The response time to the VP\_STALL signal must not exceed two cycles: when VP\_STALL is asserted, the CCP2 module can send 0, 1, or 2 pixels to the video port.

VP\_STALL is asserted and deasserted synchronous to the functional clock.

---

**NOTE:** Stalling the video port for data received from the sensor may lead to internal overflows; VP\_STALL must not be used for this purpose.

---

The pixel clock is generated from the functional clock. Clock pulses are gated based on the selected clock division factor and pixel availability. In other words, software must set the CCP2 receiver to ensure that the pixel clock:

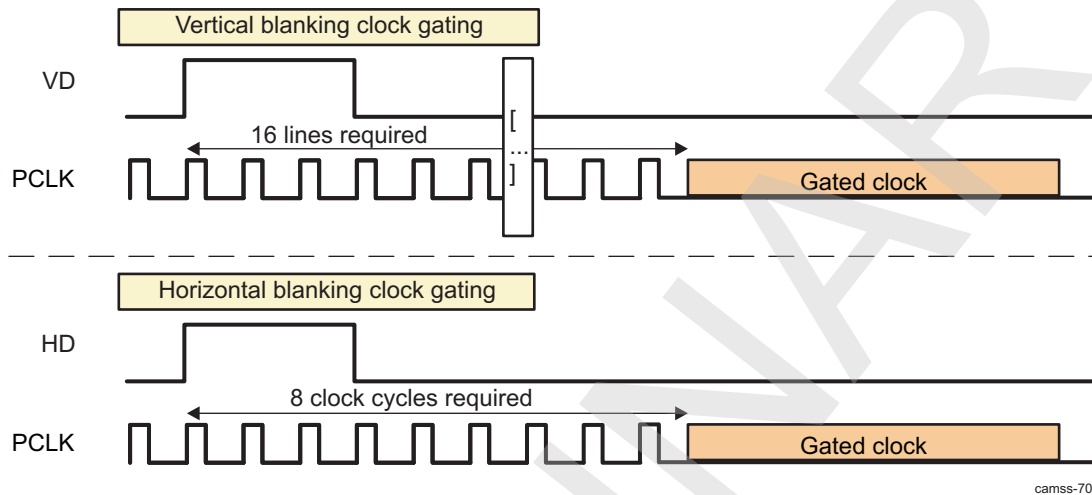
- Never exceeds what the ISP can support: the top value set in the [CCP2\\_CTRL\[31:15\] FRACDIV](#) bit field
- Is sent only when valid pixels or blanking data must be sent

Also, software must set the number of clock pulses during horizontal blanking periods using the [CCP2\\_CTRL1\[1:0\] BLANKING](#) bit field.

- The CCP2 receiver stops generating horizontal blanking clock pulses when the number of pulses defined in the [CCP2\\_CTRL1\[1:0\] BLANKING](#) bit field are generated or when data for the next line arrives from the sensor.
- Free-running horizontal blanking can be selected only when data comes from the sensor.
- When data comes from memory, the CCP2 receiver ensures that the number of horizontal blanking pulses defined in the [CCP2\\_CTRL1\[1:0\] BLANKING](#) bit field is received. The CCP2 receiver stops generating horizontal blanking clock pulses when then number of pulses defined in the [CCP2\\_CTRL1\[1:0\] BLANKING](#) bit field is generated or when data for the next line arrives from the sensor.

**NOTE:** To work properly, the ISP requires a minimum of eight clock cycles in the horizontal blanking period and a minimum of 16 lines in the vertical blanking period. The pixel clock can be gated only when these intervals are respected. This is required to flush the pipeline of the different ISP modules. Figure 8-24 shows VP\_PCLK gating during blanking periods.

**Figure 8-24. ISS CCP2 VP\_PCLK Gating During Blanking Periods**



Vertical blanking generation is controlled through the `CCP2_CTRL[9] VP_CLK_FORCE_ON` bit. The VP\_PCLK clock is enabled during vertical blanking periods when this bit is set. This pushes pixels through the ISP processing pipe. It is needed, for example, when the ISP resizer uses the averager. Otherwise, hardware ensures only that at least four clock pulses are generated before the first pixel of each frame. It may be necessary for the clock to keep running after the frame end to flush internal pipelines. In that case, an interrupt request (IRQ) or status bit is typically present in the attached hardware that indicates when the VP\_PCLK clock is no longer needed (for example, an end of processing interrupt). The module leaves the vertical blanking state when new data is received from the sensor or the memory read channel.

The configured pixel clock is used for active and blanking periods.

Table 8-51 shows how RAW and YUV data is sent over the video port. The data is sent to the ISIF if ISP is used. For the ISIF details about video port data, see Section 8.3 *ISS: ISP*.

**Table 8-51. ISS CCP2 Video Port Data Mapping**

Format	Video Port DATA[15:0]															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW6	0	0	0	0	0	0	0	0	0	0	R5	R4	R3	R2	R1	R0
RAW7	0	0	0	0	0	0	0	0	0	R6	R5	R4	R3	R2	R1	R0
RAW8	0	0	0	0	0	0	0	0	R7	R6	R5	R4	R3	R2	R1	R0
RAW10	0	0	0	0	0	0	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
RAW12	0	0	0	0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
RAW14	0	0	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
RAW16	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
YUV4:2:2	0	0	0	0	0	0	0	0	U7	U6	U5	U4	U3	U2	U1	U0
	0	0	0	0	0	0	0	0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
	0	0	0	0	0	0	0	0	V7	V6	V5	V4	V3	V2	V1	V0
	0	0	0	0	0	0	0	0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

#### 8.2.4.3.4 ISS CCP2 Data Compression

The data compression technique used is DPCM and pulse code modulation (PCM).

The CCP2 receiver performs on-the-fly compression and decompression. The compressed/decompressed data is passed to the video preprocessing hardware or stored in memory.

The data compression method is lossy and does not require any information outside the current encoded/decoded line. This means that all the image lines can be encoded and decoded separately.

Two different predictors are used:

- The simple predictor  
This predictor uses only the previous same color component value as a prediction value. Therefore, only 2-pixel memory is required.
- The advanced predictor  
This predictor uses four previous pixel values, when the prediction value is evaluated. This means that also the other color component values are used, when the prediction value is defined.

The preferable use is that the simple predictor is used with 10 bits to 8 bits or 12 bits to 8 bits conversion (10810 or 12812) and the advanced predictor is used with 10 bits to 7 bits and 10 bits to 6 bits conversions (10710 and 10610). The advanced predictor gives slightly better prediction for pixel value; thus, the image quality can be improved with it. Because the simple predictor is very simple, the processing power and memory requirements are reduced with it, when the image quality is already high enough.

Select the predictor with the [CCP2\\_LCx\\_CTRL\[10\]](#) DPCM\_PRED bit.

#### 8.2.4.3.5 ISS CCP2 Memory Read Channel

The memory channel can perform the following operations:

- Reads data from memory. It is unpacked and DPCM or A-Law decompressed if necessary.
- Sends data to the video preprocessing hardware
- Sends data back to memory. It can be DPCM or A-Law compressed and packed before it is sent to memory.

It cannot receive its input data directly from the sensor, and the logical channels are disabled when the memory channel is enabled.

[Table 8-52](#) summarizes supported modes for memory-to-memory operations.

---

**NOTE:** Video port and memory destinations are mutually exclusive.

---



**Table 8-52. ISS CCP2 Memory-to-Memory Supported Operations**

Memory Input	Memory Output																						
	RAW 6	RAW 6+PA CK	RAW 6+DP CM	RAW 6+PA CK+D PCM	RAW 6+DP CM_A DV	RAW 6+PA CK+D PCM_ADV	RAW 7	RAW 7+PA CK	RAW 7+DP CM	RAW 7+PA CK+D PCM	RAW 7+DP CM_A DV	RAW 7+PA CK+D PCM_ADV	RAW 8	RAW 8+DP CM	RAW 8+DP CM12	RAW 8+AL AW10	RAW 10	RAW 10+P ACK	RAW 12	RAW 12+P ACK	RAW 14	RAW 16	
RAW6																							
RAW6 + PACK																							
RAW6 + DPCM																	X	X					
RAW6 + PACK + DPCM																	X	X					
RAW6 + DPCM_ADV																	X	X					
RAW6 + PACK + DPMC_ADV																	X	X					
RAW7																							
RAW7 + PACK																							
RAW7 + DPCM																	X	X					
RAW7 + PACK + DPCM																	X	X					
RAW7 + DPCM_ADV																	X	X					
RAW7 + PACK + DPMC_ADV																	X	X					
RAW8																							
RAW8 + DPCM																	X	X					
RAW8 + DPCM12																			X	X			

**Table 8-52. ISS CCP2 Memory-to-Memory Supported Operations (continued)**

Memory Input	Memory Output																						
	RAW 6	RAW 6+PA CK	RAW 6+DP CM	RAW 6+PA CK+D PCM	RAW 6+DP CM_A DV	RAW 6+PA CK+D PCM_ADV	RAW 7	RAW 7+PA CK	RAW 7+DP CM	RAW 7+PA CK+D PCM	RAW 7+DP CM_A DV	RAW 7+PA CK+D PCM_ADV	RAW 8	RAW 8+DP CM	RAW 8+DP CM12	RAW 8+AL AW10	RAW 10	RAW 10+P ACK	RAW 12	RAW 12+P ACK	RAW 14	RAW 16	
RAW8 + ALAW10																	X	X					
RAW10			X	X	X	X			X	X	X	X					X						
RAW10 + PACK			X	X	X	X			X	X	X	X					X						
RAW12																							
RAW12 + PACK																							
RAW14																							
RAW16																							

Table 8-53 summarizes supported modes for memory-to-video port operations.

**Table 8-53. ISS CCP2 Memory-to-Video Port Supported Formats**

Memory Input	Video Port Output						
	RAW6	RAW7	RAW8	RAW10	RAW12	RAW14	RAW16
RAW6				X			
RAW6 + PACK				X			
RAW6 +DPCM				X			
RAW6 + PACK + DPCM				X			
RAW6 + DPCM_ADV				X			
RAW6 + DPCM_ADV + PACK				X			
RAW7				X			
RAW7 + PACK				X			
RAW7 + DPCM				X			
RAW7 + PACK + DPCM				X			
RAW7 + DPCM_ADV				X			
RAW7 + DPCM_ADV + PACK				X			

**Table 8-53. ISS CCP2 Memory-to-Video Port Supported Formats (continued)**

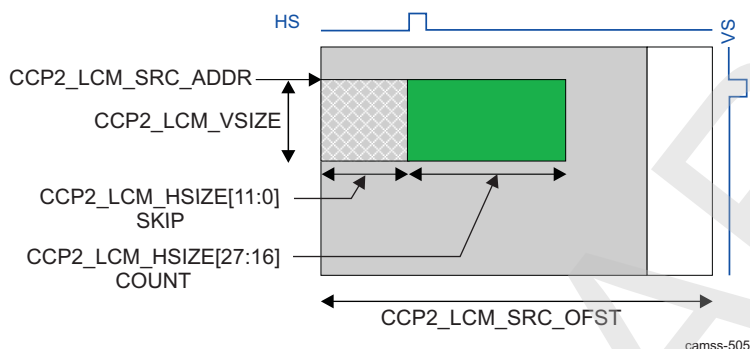
Memory Input	Video Port Output						
	RAW6	RAW7	RAW8	RAW10	RAW12	RAW14	RAW16
RAW8				X			
RAW8 + DPCM				X			
RAW8 + DPCM12					X		
RAW10				X			
RAW10 + PACK				X			
RAW12					X		
RAW12 + PACK					X		
RAW14						X	
RAW16							X

PRELIMINARY

### 8.2.4.3.5.1 ISS CCP2 Read Data From Memory

Figure 8-25 shows the data organization in memory.

**Figure 8-25. ISS CCP2 Data Organization in Memory**



The user chooses the start address and the line length using the [CCP2\\_LCM\\_SRC\\_ADDR](#) and [CCP2\\_LCM\\_SRC\\_OFST](#) registers. The image start address normally must point to the beginning of a line because of packing constraints. However, it does not necessarily point to the first line of the frame in memory. The [CCP2\\_LCM\\_VSIZE\[27:16\]](#) COUNT bit field specifies the total line count to be read from memory.

It is also possible to skip a certain pixel count ([CCP2\\_LCM\\_HSIZE\[11:0\]](#) SKIP) from the start of the line. Thus, they are not sent to the video port or back to memory. The [CCP2\\_LCM\\_HSIZE\[27:16\]](#) COUNT bit field specifies the horizontal size of the image. The pixels after the right boundary of the image are not read from memory.

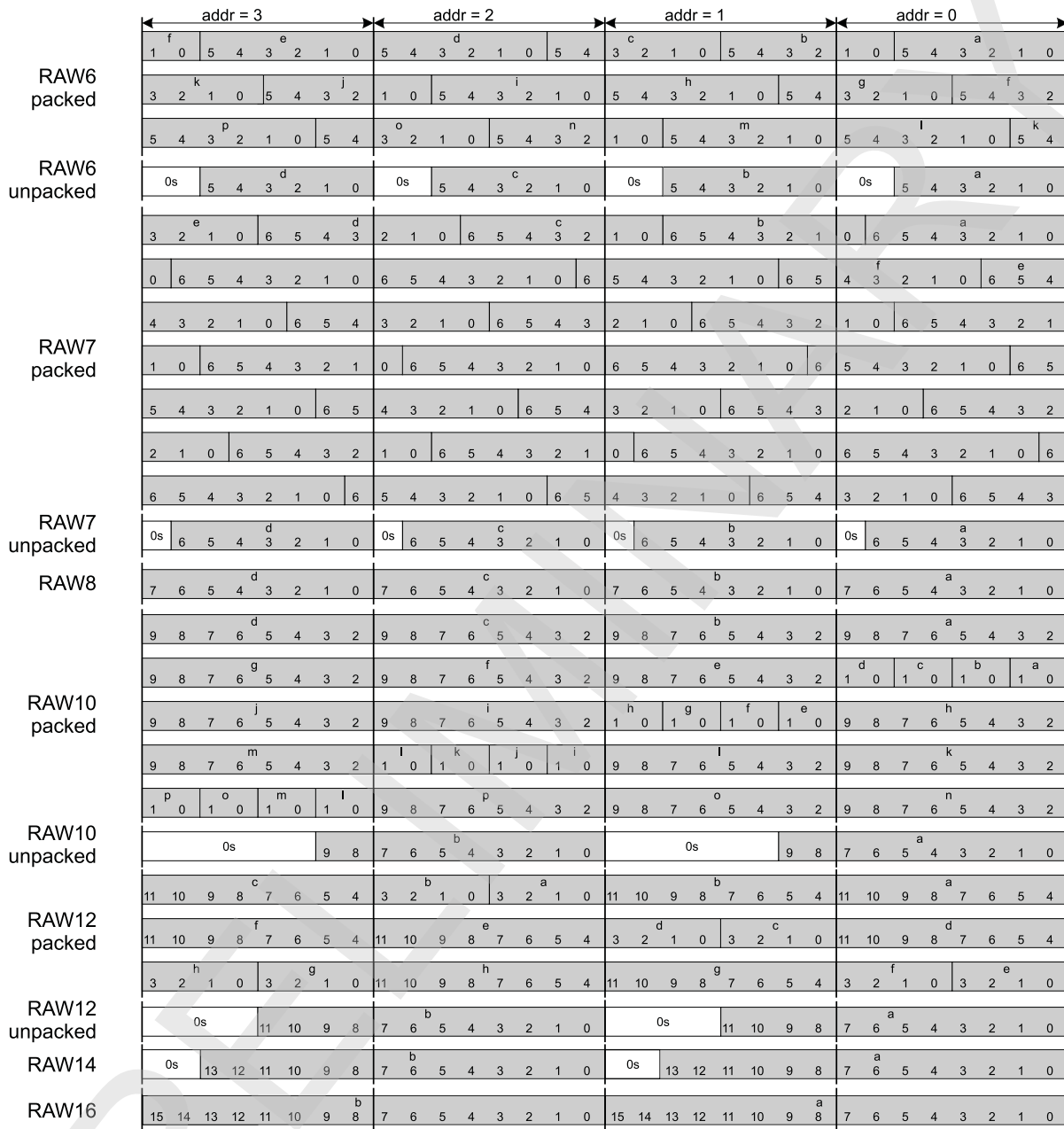
When data are sent to the video port, throughput is imposed by the selected VP\_PCLK. Otherwise, it is imposed by the selected interconnect read port clock. The interconnect read rate can be throttled (limiting the maximum data read speed for memory-to-memory operation) using the [CCP2\\_LCM\\_CTRL\[4:3\]](#) READ\_THROTTLE bit field. Therefore, it is possible to read the unused data at a higher rate than the used video port data rate. This provides better performance than framing the image in the video preprocessing hardware.

The data storage format in memory is defined by the [CCP2\\_LCM\\_CTRL\[18:16\]](#) SRC\_FORMAT and [CCP2\\_LCM\\_CTRL\[23\]](#) SRC\_PACK bit fields.

Not all I/O format combinations are valid. For more information, see [Table 8-52](#) and [Table 8-53](#).

[Figure 8-26](#) shows how data are packed in memory. Pixel order (left to right in the image) is alphabetical (a, b, c). Therefore, data storage is little endian.

Figure 8-26. ISS CCP2 Data Organization Packing in Memory



camss-506u

Table 8-54 summarizes the storage reduction versus unpacked format and image width restrictions when data packing is used. The image width applies to the data width multiple in pixels that must be stored to have storage reduction. Moreover, because each address is 8 bits long, the percentage shows how many bits out of 8 are to be packed in the empty space from another pixel address. A pixel is selected and split. One of the parts is put into another pixel address empty space. When unpacked, each pixel bit is stored continuously again.

Table 8-54. ISS CCP2 Data Packing Benefit and Constraints

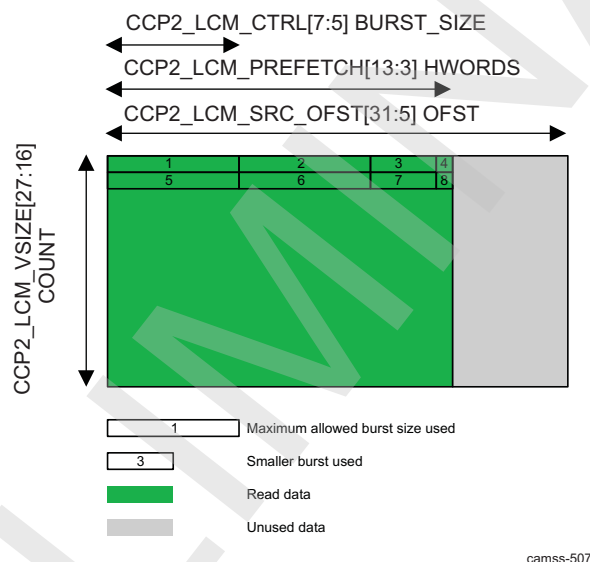
	Bits Per Pixel		Storage Reduction	Width Multiple (Pixels)
	Packed	Unpacked		
RAW6	6	8	25%	16

**Table 8-54. ISS CCP2 Data Packing Benefit and Constraints (continued)**

	Bits Per Pixel		Storage Reduction	Width Multiple (Pixels)
	Packed	Unpacked		
RAW7	7	8	13%	3
RAW8	8	8	0%	4
RAW10	10	16	38%	16
RAW12	12	16	25%	8

#### 8.2.4.3.5.2 ISS CCP2 Memory Read Port Burst Generation

Hardware always uses the largest possible burst size according to the setup. The amount of data read from memory can be higher than what is actually used by the CCP2 receiver. Only full 64-bit burst words are read back from memory. [Figure 8-27](#) shows the data organization and the relationship between the different parameters controlling the burst generation.

**Figure 8-27. ISS CCP2 Data Organization in Memory****NOTE:**

- A minimum burst size of 2 must be selected for correct operation.
- The HWORDS bit field must be even for correct operation.

The  $CCP2\_LCM\_SRC\_ADDR$  register address of the first data to read is aligned to a 32-byte boundary. The read port fetches  $CCP2\_LCM\_PREFETCH[13:3]$  HWORDS of 64-bit words per line using the longest possible burst computed from the  $CCP2\_LCM\_CTRL[7:5]$  BURST\_SIZE bit field and the remaining data to be fetched. Burst size of 128 bytes is preferred. When the CCP2 receiver is configured to fetch more data than required, extra data are dropped internally.

#### 8.2.4.3.5.3 ISS CCP2 Video Port

The video port always receives unpacked data. It can be enabled using the  $CCP2\_LCM\_CTRL[2]$  DST\_PORT bit. Its clock can be selected with the  $CCP2\_CTRL[31:15]$  FRACDIV bit field and gated or not during frame blanking periods using the  $CCP2\_CTRL[9]$  VP\_CLK\_FORCE\_ON bit.

The data format used by the video port is defined by the  $CCP2\_LCM\_CTRL[26:24]$  DST\_FORMAT bit field. For a list of supported modes, see [Table 8-53](#).

**8.2.4.3.5.4 ISS CCP2 Encode, Pack, and Store Data**

This stage is used only when data are sent to memory. Memory destination is selected using the [CCP2\\_LCM\\_CTRL\[2\] DST\\_PORT](#) bit. The output data format is defined by the [CCP2\\_LCM\\_CTRL\[26:24\] DST\\_FORMAT](#) bit field and the [CCP2\\_LCM\\_CTRL\[31\] DST\\_PACK](#) bit. Not all possible combinations are supported; see [Table 8-52](#) for details.

The destination address and offset for the output data of the memory channel are set by the [CCP2\\_LCM\\_DST\\_ADDR](#) and [CCP2\\_LCM\\_DST\\_OFST](#) registers.

Because of alignment constraints on the interconnect port, the output image width restrictions in [Table 8-55](#) apply.

**Table 8-55. ISS CCP2 Output Width Restrictions in Memory-to-Memory Operation**

Format	Bits per Pixel	Width Multiple of <sup>(1)</sup>	Note
RAW6	8	1	Full 32-bit words are written at the end of the line. This last word can eventually include 0s.
RAW6 packed	6	1	
RAW7	8	1	
RAW7 packed	7	1	
RAW8	8	1	
RAW10	16	1	
RAW10 packed	10	16	
RAW12	16	1	Same constraints as RAW8
RAW12 packed	12	8	

<sup>(1)</sup> In continuous mode, lines must be multiples of 128 bits. In 2D mode, lines must start on 128-bit boundaries.

For example, when RAW6 packed data are written to memory, any output width is allowed. However, only full 32-bit words are written to memory. This eventually overwrites some data in memory at the end of a line.

The supported output width is restricted for packed RAW10 and RAW12 data because of the particular bit ordering in those formats (see [Figure 8-26](#)).

When the [DST\\_OFST](#) bit is set to 0, start of lines are aligned on 4-byte boundaries. When [DST\\_OFST](#) is not set to 0, data are aligned on 32-byte boundaries.

---

**NOTE:** The RAW6, RAW7, and DPCM data formats do not apply to the MIPI CSI1-compatible mode.

---

**8.2.4.3.5.5 ISS CCP2 DPCM Decompression History**

The DPCM compression algorithm can encode the difference between consecutive samples in a line instead of the actual samples value to reduce the amount of data to store. The drawback is that lines must always be decoded from the beginning (the first samples of a line are always encoded as PCM).

The CCP2 receiver has a mechanism to preserve the DPCM decode history for each line to avoid decoding the same samples multiple times when vertical frame division mode is used.

The typical use case (also known as vertical frame division mode) is when an image is wider than the ISP can process on the fly, but the image must be processed. Therefore, the image is cut into multiple vertical slices that are processed sequentially by the ISP. The slices are stitched together in the SDRAM through proper address generation in the ISP. Because of various alignment constraints in the ISP, the slices must overlap. Without preserving the DPCM history, all lines from the beginning of the second (and other consecutive) vertical slice would have to be recoded, which would lead to performance degradation.

Writing DPCM history information into the system memory is enabled by setting the `CCP2_LCM_HISTORY[16]` `EN_HIST_WR` bit. The `CCP2_LCM_HISTORY[15:0]` `HIST_EXPORT` bit field defines the position at which history data is written to memory. The position is counted from the beginning line. The first decoded pixel has position 0. The last decoded pixel has position `CCP2_LCM_HSIZE[14:0]` `SKIP` + `CCP2_LCM_HSIZE[30:16]` `COUNT`. The `CCP2_LCM_HISTORY[15:0]` `HIST_EXPORT` bit field is used to choose the resume position and allows support of overlapping vertical slices. History data is written to the SDRAM at `ADDR = CCP2_LCM_DST_ADDR[31:5]` `ADDR` +  $Y \times 8$  bytes (where  $Y$  is the line number). The CCP2 receiver always writes 8 bytes of history data per line to the SDRAM regardless of the chosen DPCM format. The CCP2 receiver uses the interface bursts to send history data to memory.

DPCM history data holds the decoded value of four samples. The CCP2 receiver exports samples of positions:

- `CCP2_LCM_HISTORY[15:0]` `HIST_EXPORT` – 3
- `CCP2_LCM_HISTORY[15:0]` `HIST_EXPORT` – 2
- `CCP2_LCM_HISTORY[15:0]` `HIST_EXPORT` – 1
- `CCP2_LCM_HISTORY[15:0]` `HIST_EXPORT`

Every sample is coded on 16 bits, and several MSBs are unused. The valid range for `CCP2_LCM_HISTORY[15:0]` `HIST_EXPORT` is [ $3..$  `CCP2_LCM_HSIZE[14:0]` `SKIP` + `CCP2_LCM_HSIZE[30:16]` `COUNT` – 1]. History data can be exported even when data is not sent to the video port (`CCP2_LCM_HISTORY[15:0]` `HIST_EXPORT` < `CCP2_LCM_HSIZE[14:0]` `SKIP` is valid).

#### 8.2.4.4 ISS CCP2 Programming Model

This section describes the programming model of the CCP2 receiver.

##### 8.2.4.4.1 ISS CCP2 Hardware Setup/Initialization

This section discusses the configuration of the CCP2 receiver required before image capture can begin. Before using the receiver, a CSIPHY initialization in CCP2 mode must be made for CSI2-B/CCP2 CAMERARX, which is associated with the CCP2 receiver. See [Section 8.2.3.2.3, ISS CCP2 and Link Initialization Sequence](#).

##### 8.2.4.4.1.1 ISS CCP2 Reset Behavior

On hardware or software reset of the ISS, all registers in the CCP2 receiver are reset to their reset values.

##### 8.2.4.4.2 ISS CCP2 Event and Status Checking

When an event occurs, the corresponding bit in the `CCP2_LC01_IRQSTATUS` (or `CCP2_LC23_IRQSTATUS`) register is set. Each event can be individually masked using the `CCP2_LC01_IRQENABLE` register (`CCP2_LC23_IRQENABLE`). Masked events are not transmitted to the interrupt lane, but the `CCP2_LC01_IRQSTATUS` register (or `CCP2_LC23_IRQSTATUS`) is updated.

Events transmitted to the ISS interrupt merger can be mapped to the Cortex-A9 or Cortex-M3 MPU subsystem or IVA3 by unmasking `ISS_HL_IRQENABLE_SET_i[]` `CCP2_IRQj` (where  $i = 0$  to 5,  $j = 0, 1, 2, 3,$  and 8). To clear an event, the following actions are required:

- Clear the event at the CCP2 receiver level by writing 1 to the corresponding bit in the `CCP2_LC01_IRQSTATUS` register (or `CCP2_LC23_IRQSTATUS`).
- Events generated by the submodules and ISS top level clear it. Each event that generates an interrupt can be individually enabled by setting the appropriate bit in the `ISS_HL_IRQENABLE_SET_i` register. It is disabled by setting the appropriate bit in the `ISS_HL_IRQENABLE_CLR_i` register. When an event occurs, the corresponding bit in the `ISS_HL_IRQSTATUS_i` register is set, regardless of whether or not the event is enabled. Bits in the `ISS_HL_IRQSTATUS_i` registers are set only when an enabled event occurs. Software can clear a pending `HS_VS_IRQ` event by setting the appropriate bit in the `CBUFF_HL_IRQSTATUS_i.HS_VS_IRQ` register. Events generated by submodules are automatically cleared at ISS top level when they are cleared at submodule level.



### 8.2.4.4.3 ISS CCP2 Register Accessibility During Frame Processing

There are two types of register accesses in the CCP2 receiver:

- Shadowed registers:
  - These registers/fields can be read and written (if the field is writable) at any time. However, written values take effect only at the start of a frame. Reads return the most-recent write, even though the settings are not used until the next start of frame.
  - The shadowed registers are:
    - [CCP2\\_LCx\\_CTRL](#)
    - [CCP2\\_LCx\\_CODE](#)
    - [CCP2\\_LCx\\_STAT\\_START](#)
    - [CCP2\\_LCx\\_STAT\\_SIZE](#)
    - [CCP2\\_LCx\\_SOF\\_ADDR](#)
    - [CCP2\\_LCx\\_EOF\\_ADDR](#)
    - [CCP2\\_LCx\\_DAT\\_SIZE](#)
    - [CCP2\\_LCx\\_DAT\\_PING\\_ADDR](#)
    - [CCP2\\_LCx\\_DAT\\_PONG\\_ADDR](#)
    - [CCP2\\_LCx\\_DAT\\_OFST](#)
- Busy-locked registers:
  - These registers/fields must not be written if the module is busy.
  - All register fields not listed as shadowed are busy-writable registers.

### 8.2.4.4.4 ISS CCP2 Enable/Disable the Hardware

The CCP2 receiver is globally controlled by the [CCP2\\_CTRL](#) register. The bit fields in this register must not be modified when the CCP2 interface is active (except [CCP2\\_CTRL\[0\] IF\\_EN](#)).

[Table 8-56](#) and [Table 8-57](#) list the procedures to enable and disable, respectively, the interface.

**Table 8-56. ISS CCP2 Interface Enable**

Step	Bit Field	Value
Clear the interface. Data acquisition starts on the following FSC synchronization code.	<a href="#">CCP2_CTRL[0] IF_EN</a>	0x1
Reset the output FIFO of the module; the reset is caused by the 0-to-1 edge transition.	<a href="#">CCP2_CTRL[0] IF_EN</a>	0x1

**Table 8-57. ISS CCP2 Interface Disable**

Step	Bit Field	Value
Disable interface. <ul style="list-style-type: none"> <li>• The interface is disabled immediately if <a href="#">CCP2_CTRL[3] FRAME</a> = 0x0.</li> <li>• If <a href="#">CCP2_CTRL[3] FRAME</a> = 0x1 and <a href="#">CCP2_LCx_CTRL[2] CRC_EN</a> = 0x0, the interface is disabled after the FEC synchronization code is received.</li> <li>• If <a href="#">CCP2_CTRL[3] FRAME</a> = 0x1 and <a href="#">CCP2_LCx_CTRL[2] CRC_EN</a> = 0x1, the interface is disabled only after the 16-bit CRC checksum and 16-bit pad data is received.</li> <li>• Before disabling the interface (<a href="#">IF_EN</a> = 0), it is advised to disable all active channels by writing <a href="#">CCP2_LCx_CTRL[0] CHAN_EN</a> = 0x0. Otherwise, if <a href="#">IF_EN</a> = 0 is set during a vertical blanking period, the reception continues until the FEC synchronization code is received for all active channels.</li> </ul>	<a href="#">CCP2_CTRL[0] IF_EN</a>	0x0

### 8.2.4.4.5 ISS CCP2 Select the Signaling Scheme

Table 8-58 lists the procedure to select the signaling scheme.

**Table 8-58. ISS CCP2 Select the Signaling Scheme**

Step	Bit Field	Value
Selects whether the data/strobe or data/clock signaling scheme is used. For the correct settings as a function of the image sensor class, see Section 8.2.3, <i>ISS CSI2 PHY</i> . Setting <code>CCP2_CTRL[1] PHY_SEL = 0x1</code> has no effect if <code>CCP2_CTRL[4] MODE = 0x0</code> (CSI1 mode). This setting must also be configured in the control module.	See Section 8.2.3, <i>ISS CSI2 PHY</i> . For the register and bit field, see Chapter 18, <i>Control Module</i> .	0x0: Data/clock 0x1: Data/strobe

### 8.2.4.4.6 ISS CCP2 Select the Mode: MIPI CSI1 or CCP2

Table 8-59 lists the procedure to select MIPI CSI1 or CCP2 mode.

**Table 8-59. ISS CCP2 Select MIPI CSI1 or CCP2 Mode**

Step	Bit Field	Value
Selects whether the CCP2 module works in MIPI CSI1 or CCP2-compatible mode	<code>CCP2_CTRL[4] MODE</code>	0x0: Disables the CCP2-specific features (data/strobe, CRC, logical channels, RAW6, RAW7, and DPCM data formats) 0x1: CCP2 mode

### 8.2.4.4.7 ISS CCP2 Burst Settings

Table 8-60 lists the procedure to configure the burst settings.

**Table 8-60. ISS CCP2 Configure Burst Settings**

Step	Bit Field	Value
The module can be forced to perform single 64-bit requests or bursts of 2x, 4x, 8x, and 16 x 64 bits. The module must always use 16 x 64. The FIFO size is 64 x 64 bits.	<code>CCP2_CTRL[6:5] BURST</code> <code>CSI2_CTRL[16] BURST_SIZE_EXPAND</code>	0x0: Single request. 0x1: 2 x 64-bit bursts 0x2: 4 x 64-bit bursts 0x3: 8 x 64-bit bursts 0x4: 16 x 64-bit bursts (with burst expand)
Enable 128 bytes (recommended setting) of the DMA CSI2/CCP DMA engine to burst 128 bytes over the L3 interconnect.	<code>CSI2_CTRL[16] BURST_SIZE_EXPAND</code>	0x0: Use the burst size defined in the <code>BURST_SIZE</code> bit field. 0x1: Allow generation of 16 x 64-bit bursts.

### 8.2.4.4.8 ISS CCP2 Debug Mode

Table 8-61 lists the procedure to enable debug mode.

**Table 8-61. ISS CCP2 Enable Debug Mode**

Step	Bit Field	Value
Enable debug mode.	<code>CCP2_CTRL[13] DBG_EN</code>	0x1

- During debug mode, the input comes from the `CCP2_DBG` register, not from the CCP2 physical interface. The full CCP2 receiver function can be debugged in debug mode. Full 32-bit values must always be written to the `CCP2_DBG` register.
- The following bits have no effect during debug mode:
  - `CCP2_CTRL[0] IF_EN`
  - `CCP2_CTRL[1] PHY_SEL`

- The following examples apply to the **CCP2\_DBG** register:
  - Synchronization codes: **CCP2\_DBG** = 0xFF000000
  - To send the RAW12 pixels 0x673, 0x452, 0x01d, 0xefc, 0xab0, 0x891, 0x326, and 0x547, write **CCP2\_DBG** = 0x01234567 followed by **CCP2\_DBG** = 0x89abcdef and **CCP2\_DBG** = 0x76543210.

**NOTE:** Each write to the **CCP2\_DBG** register sends a full 32-bit word through the CCP2 receiver hardware. When 8- or 16-bit writes are performed to the register, the previous 32-bit value is merged with the newly written one. When the driver writes, for example, 0x01234567 followed by 0x0000 00FF with **BYTEEN** = 0x1 (this write from the MPU subsystem informs that only 8 bits are written), the CCP2 receiver pipeline gets 0x01234567 followed by 0x012345FF.

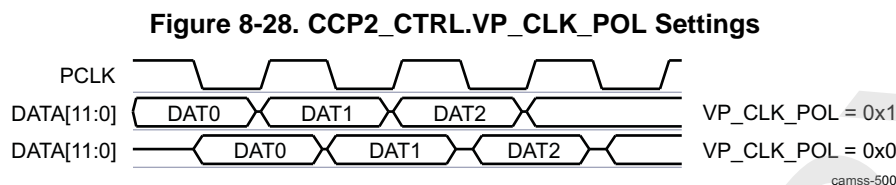
**8.2.4.4.9 ISS CCP2 Video Port**

Table 8-62 lists the procedure to configure the video port.

**Table 8-62. ISS CCP2 Configure Video Port**

Step	Bit Field	Value
Set the video port output frequency. It varies from <b>ISS_FLCK</b> down to <b>ISS_FLCK/65,536</b> MHz.	<b>CCP2_CTRL</b> [31:15] <b>FRACDIV</b>	From 1 to 65,536
Enable video port clock during vertical blanking periods or not.	<b>CCP2_CTRL</b> [9] <b>VP_CLK_FORCE_ON</b>	0x0: The video port clock is gated during vertical blanking periods. 0x1: The video port clock is free-running during vertical blanking periods.
Controls whether the video-port output is the only output interface enabled and applies for all channels. When <b>CCP2_CTRL</b> [11] <b>VP_ONLY_EN</b> = 0x1, the data are output only to the video port; the interface master port is not used. The two parts of the frame (embedded data and pixel data) are output to the video port (instead of pixel data to the video port and embedded data to the interconnect). <ul style="list-style-type: none"> <li>• The video port outputs the embedded data defined by the <b>CCP2_LCx_STAT_START</b> and <b>CCP2_LCx_STAT_SIZE</b> registers without decompression.</li> <li>• The video port outputs the pixel data defined by the <b>CCP2_LCx_DAT_START</b> and <b>CCP2_LCx_DAT_SIZE</b> registers.</li> </ul>	<b>CCP2_CTRL</b> [11] <b>VP_ONLY_EN</b>	
Control the video port pixel clock polarity: (Recommended setting: rising edge)	<b>CCP2_CTRL</b> [12] <b>VP_CLK_POL</b>	0x0: The CCP2 receiver writes the data on the video port on the pixel-clock falling edge. The module connected to the video port samples the data on the pixel clock rising edge. 0x1: The CCP2 receiver writes the data on the video port on the pixel-clock rising edge. The module connected to the video port samples the data on the pixel clock falling edge. <a href="#">Figure 8-28</a> shows the settings for <b>CCP2_CTRL.VP_CLK_POL</b> .

Figure 8-28 shows the settings for `CCP2_CTRL.VP_CLK_POL`.



#### 8.2.4.4.10 ISS CCP2 Logical Channels

The CCP2 receiver supports simultaneous logical channels. Each logical channel is controlled independently with its own set of registers. The four sets of registers are identical, but some reset values are different.

The same description applies to all four logical channels (LCx, where x = 0 to 3).

All the registers in this section can be modified at any time. However, the modifications apply only from the start of the following frame.

#### 8.2.4.4.11 ISS CCP2 Controls

Table 8-63 lists the procedure to enable the logical channels.

**Table 8-63. ISS CCP2 Enable Logical Channels**

Step	Bit Field	Value
Enable logical channels. It has no effect if <code>CCP2_CTRL[4] MODE = 0x0</code> . By default, all logical channels except logical channel 0 are disabled. Only the pixel data of one logical channel can go to the video preprocessing hardware; the SOF and EOF lines are always sent to memory through the interconnect interface.	<code>CCP2_LCx_CTRL[0] CHAN_EN</code>	0x1

#### 8.2.4.4.12 ISS CCP2 Region-of-Interest

Table 8-64 lists the procedure to enable the region-of-interest.

**Table 8-64. ISS CCP2 Enable Region-of-Interest**

Step	Bit Field	Value
Enables the region-of-interest feature (SOF lines, pixel data, and EOF lines): <ul style="list-style-type: none"> <li>If enabled, register settings set the position and size of each region; all data not in a region of interest are ignored.</li> <li>If disabled, all data in the frame are output. <code>CCP2_LCx_CTRL[1] REGION_EN</code> is set to 0x0 for a JPEG bitstream.</li> </ul>	<code>CCP2_LCx_CTRL[1] REGION_EN</code>	0x1

#### 8.2.4.4.13 ISS CCP2 CRC

Table 8-65 lists the procedure to enable the CRC.

**Table 8-65. ISS CCP2 Enable the CRC**

Step	Bit Field	Value
Enable the CRC. If the received checksum and the computed checksum do not match, an interrupt is triggered: the corresponding event is <code>LCx_CRC_IRQ</code> . Setting <code>CCP2_LCx_CTRL[2] CRC_EN = 0x1</code> has no effect if <code>CCP2_CTRL[4] MODE = 0x0</code> .	<code>CCP2_LCx_CTRL[2] CRC_EN</code>	0x1

**8.2.4.4.14 ISS CCP2 Destination Format**

Control the destination format:

- The CCP2 receiver reformats received data to store it in memory or to send it to the video preprocessing.
- The [CCP2\\_LCx\\_CTRL\[7:3\]](#) FORMAT bit field controls the destination-data format:
  - EXP8 = Data expansion to 8 bits, padding with zeros
  - EXP16 = Data expansion to 16 bits, padding with alpha or zeros. The [CCP2\\_CTRL\[15:8\]](#) ALPHA bit field can be used to set an alpha value. For RGB444 + EXP16:
    - data\_out [31:28] = ALPHA [3:0]
    - data\_out [15:12] = ALPHA [3:0]
  - EXP32 = Data expansion to 32 bits, padding with alpha. The [CCP2\\_CTRL\[15:8\]](#) ALPHA bit field can be used to set an alpha value. For RGB888 + EXP32: data\_out [31:24] = ALPHA [7:0]
  - FSP = False synchronization protection code decoding. Applies only to JPEG8 data format.
  - VP = Output to video preprocessing is enabled.

**8.2.4.4.15 ISS CCP2 Frame Acquisition**

[Table 8-66](#) lists the procedure to acquire frames.

**Table 8-66. ISS CCP2 Frame Acquisition**

Step	Bit Field	Value
Program the number of frames that the CCP2 receiver acquires. The value of COUNT is decremented after each frame received.	<a href="#">CCP2_LCx_CTRL[31:24]</a> COUNT	0: The counter is free-running (default value). 1255: Remaining frames to be acquired
Writes to the COUNT bit field are controlled by the COUNT_UNLOCK bit.	<a href="#">CCP2_LCx_CTRL[16]</a> COUNT_UNLOCK	
Configures PING and PONG addresses	<a href="#">CCP2_LCx_DAT_PING_ADDR</a> and <a href="#">CCP2_LCx_DAT_PONG_ADDR</a>	
Indicates whether the PING address or PONG address was used to store the pixel data of the last frame. After reset or after a 0-to-1 edge transition in <a href="#">CCP2_CTRL[0]</a> IF_EN, the pixel data are written in the PING buffer and <a href="#">CCP2_LCx_CTRL[17]</a> PING_PONG = 0x1 (PONG). After the first FEC synchronization code is received, the pixel data are written in the PONG buffer and <a href="#">CCP2_LCx_CTRL[17]</a> PING_PONG = 0x0 (PING). <a href="#">CCP2_LCx_CTRL[17]</a> PING_PONG toggles after every FEC synchronization code.	<a href="#">CCP2_LCx_CTRL[17]</a> PING_PONG	

After the correct number of frames is received, acquisition is automatically disabled ([CCP2\\_LCx\\_CTRL\[0\]](#) CHAN\_EN = 0x0) and the COUNT\_IRQ interrupt is triggered. The programmer can re-enable the acquisition by resetting the [CCP2\\_LCx\\_CTRL\[0\]](#) CHAN\_EN bit to 0x1.

**8.2.4.4.16 ISS CCP2 Synchronization Codes**

The FSC, FEC, LSC, and LEC synchronization codes have default values given by the CCP2 specification. Also, each logical channel is identified by a default identifier.

The [CCP2\\_LCx\\_CODE](#) register enables overwriting of the default values: [CCP2\\_LCx\\_CODE\[11:8\]](#) FSC, [CCP2\\_LCx\\_CODE\[15:12\]](#) FEC, [CCP2\\_LCx\\_CODE\[3:0\]](#) LSC, and [CCP2\\_LCx\\_CODE\[7:4\]](#) LEC overwrite the 4 LSBs of the 32-bit synchronization codes. The default values must not be modified.

**8.2.4.4.17 ISS CCP2 Status Data**

The SOF and EOF status lines can be output to memory.

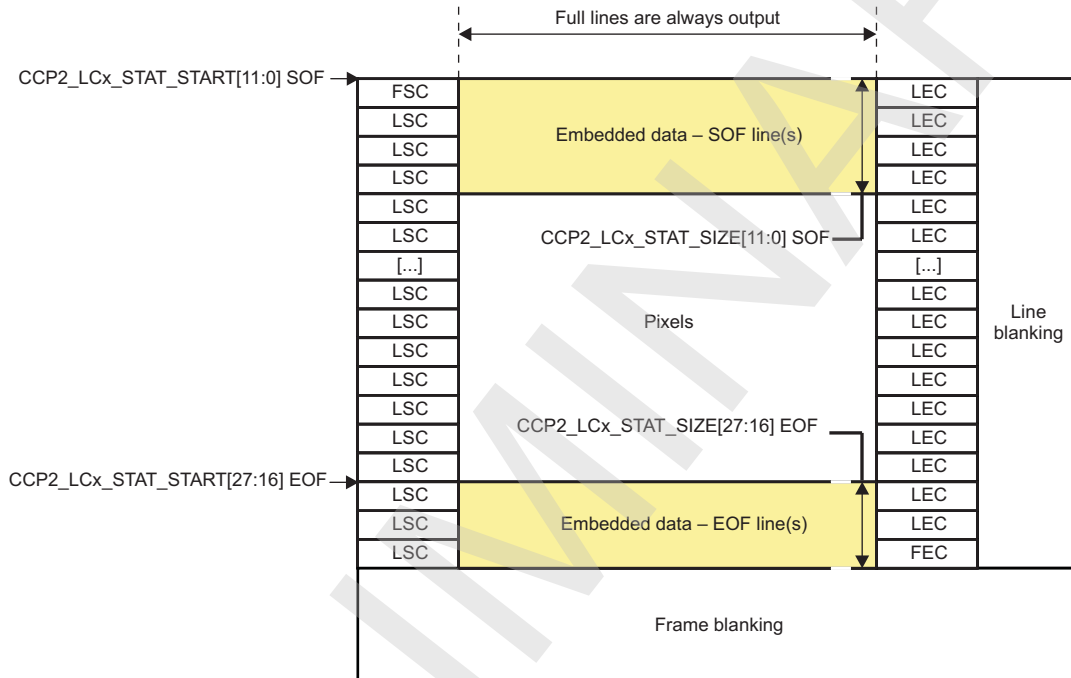
The SOF and EOF status lines always cover full lines. No register settings enable the setting of width.

The [CCP2\\_LCx\\_STAT\\_START](#) register enables the setting of the vertical start position of the SOF and EOF status lines. Because the SOF status line comes first in the CCP2 frame, [CCP2\\_LCx\\_STAT\\_START\[11:0\]](#) SOF = 0x0.

The [CCP2\\_LCx\\_STAT\\_SIZE](#) register enables the setting of the numbers of SOF and EOF status lines. If [CCP2\\_LCx\\_STAT\\_SIZE\[11:0\]](#) SOF = 0x0 and [CCP2\\_LCx\\_STAT\\_SIZE\[27:16\]](#) EOF = 0x0, no status data are output.

[Figure 8-29](#) shows the SOF and EOF region settings. The SOF and EOF status lines and the pixel data must not overlap, but can be consecutive.

**Figure 8-29. ISS CCP2 SOF and EOF Region Settings**



camss-501

The 32-bit destination addresses of the SOF status lines are set by the [CCP2\\_LCx\\_SOF\\_ADDR](#) register.

---

**NOTE:** The destination addresses of the SOF status lines must be aligned on a 32-byte boundary; the 5 LSBs of the address are ignored. The SOF lines are packed together at the destination address.

---

The 32-bit destination addresses of the EOF status lines are set by the [CCP2\\_LCx\\_EOF\\_ADDR](#) register.

---

**NOTE:** The destination addresses of the EOF status lines must be aligned on a 32-byte boundary; the 5 LSBs of the address are ignored. The EOF lines are packed together at the destination address.

---



---

**NOTE:** The CCP2 receiver does not modify the data in the SOF and EOF status lines. The data are received and written with no modifications.

---

#### 8.2.4.4.18 ISS CCP2 Pixel Data Region

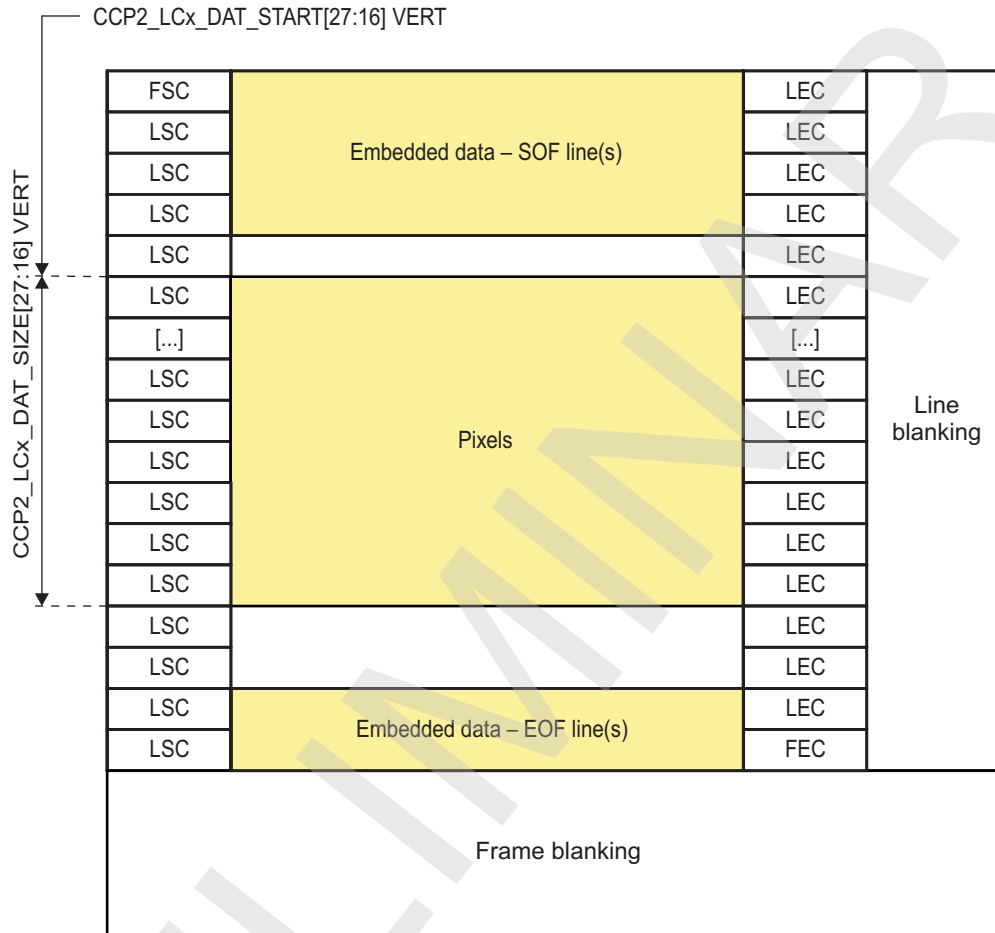
Pixel data can be output to memory or to the video-preprocessing hardware.

The pixel data region covers full lines. The [CCP2\\_LCx\\_DAT\\_SIZE](#) register sets the horizontal size of the pixel region. The vertical size is expressed in lines.

The [CCP2\\_LCx\\_DAT\\_START](#) register enables the setting of the vertical start position of the pixel data. The vertical start position is expressed in lines.

Figure 8-30 shows the pixel region settings.

**Figure 8-30. ISS CCP2 Pixel Data Region Settings**



camss-502

The 32-bit destination addresses of the pixel data are set by the [CCP2\\_LCx\\_DAT\\_PING\\_ADDR](#) and [CCP2\\_LCx\\_DAT\\_PONG\\_ADDR](#) registers.

**NOTE:** The destination address must be aligned on a 32-byte boundary; the 5 LSBs of the address are ignored. The pixel data lines are packed together at the destination address.

It is possible to perform double-buffering (ping-ponging) at the destination by setting different addresses in the [CCP2\\_LCx\\_DAT\\_PING\\_ADDR](#) and [CCP2\\_LCx\\_DAT\\_PONG\\_ADDR](#) registers. It is possible to disable double-buffering by setting up the same address in both registers. The [CCP2\\_LCx\\_CTRL\[17\]](#) PING\_PONG bit must be used by software to determine which address contains the latest frame.

A destination pitch controls the address jump between the address of the first pixel of the previous line and the address of the first pixel of the current line. The destination pitch is set in bytes with the [CCP2\\_LCx\\_DAT\\_OFST](#) register. It applies for [CCP2\\_LCx\\_DAT\\_PING\\_ADDR](#) and [CCP2\\_LCx\\_DAT\\_PONG\\_ADDR](#).

**NOTE:** The destination pitch must be a multiple of 32 bytes; the 5 LSBs of the address are ignored.

The destination data format is set with the [CCP2\\_LCx\\_CTRL\[7:2\]](#) FORMAT bit field.

For the PING frame:

- @Line0 = `CCP2_LCx_DAT_PING_ADDR`
- @Line1 = @Line0 + `CCP2_LCx_DAT_OFST`
- @Line2 = @Line1 + `CCP2_LCx_DAT_OFST`

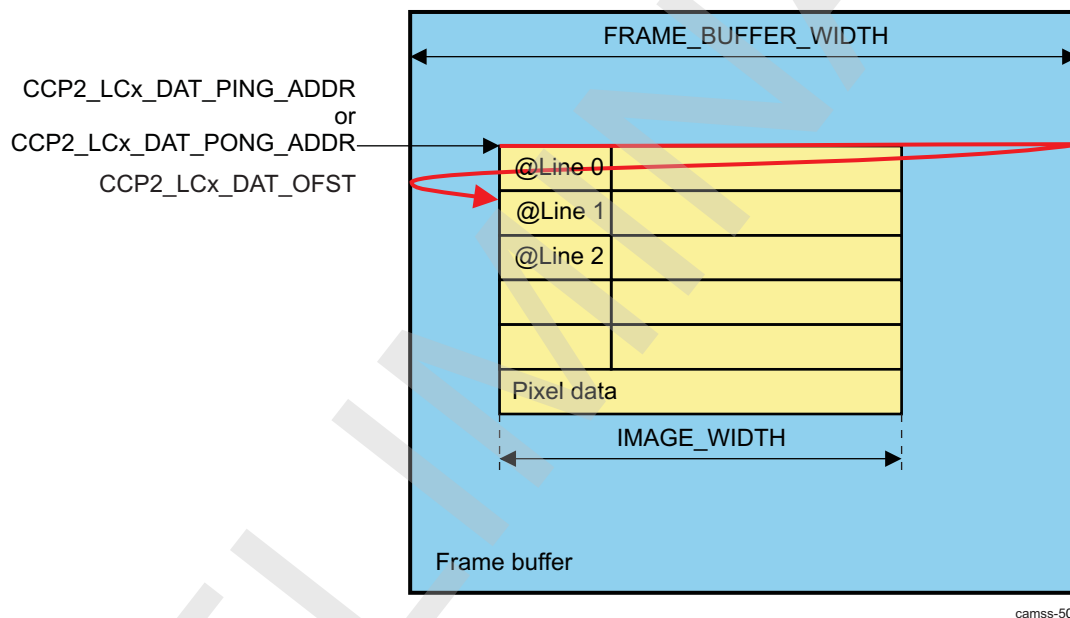
For the PONG frame:

- @Line0 = `CCP2_LCx_DAT_PONG_ADDR`
- @Line1 = @Line0 + `CCP2_LCx_DAT_OFST`
- @Line2 = @Line1 + `CCP2_LCx_DAT_OFST`

When `CCP2_LCx_DAT_OFST = 0x0`, the lines are written contiguously in memory. The destination pitch enables 2D transfers; it is required to write the pixel data directly in the frame buffer, for instance.

In such cases, `CCP2_LCx_DAT_OFST = FRAME_BUFFER_WIDTH`. Figure 8-31 shows the pixel data destination settings.

**Figure 8-31. ISS CCP2 Pixel Data Destination Settings**



camss-503

#### 8.2.4.4.19 ISS CCP2 Memory Read Channel

##### 8.2.4.4.19.1 ISS CCP2 Write Data From Sensor to Memory

Data can be captured from the sensor using any logical channel. To keep the native data format, the channel format must be set to YUV4:2:2 little-endian format.

##### 8.2.4.4.19.2 ISS CCP2 Read Data From Memory

Table 8-67 lists the procedure to configure read data from memory.

**Table 8-67. ISS CCP2 Configure Read Data From Memory**

Step	Bit Field	Value
Disable all logical channels.	<code>CCP2_CTRL[0] IF_EN</code>	0x0
Wait until disabling of the physical interface is effective before enabling the memory read channel.	<code>CCP2_CTRL[3] FRAME</code>	0x1
Configure the burst size to 16 × 64-bit bursts.	<code>CCP2_LCM_CTRL[7:5] BURST_SIZE</code>	0x4



**Table 8-67. ISS CCP2 Configure Read Data From Memory (continued)**

Step	Bit Field	Value
<p>Configure the source data format, location, and framing.</p> <p>In addition to the <a href="#">CCP2_LCM_HSIZE[11:0]</a> SKIP and <a href="#">CCP2_LCM_HSIZE[27:16]</a> COUNT bit fields, firmware must specify the amount of data to be fetched from memory. This value is set in 64-bit word steps and must be a multiple of 32 bytes (four words of 64 bits). The value is computed with the following formula:</p> $\text{HWORDS} = 4 \times \text{ceil}((\text{SKIP} + \text{COUNT}) \times \text{bits\_per\_pixel}) / (8 \times 32) \quad (3)$ <p>The <a href="#">CCP2_LCM_SRC_ADDR</a> and <a href="#">CCP2_LCM_SRC_OFST</a> registers must be aligned on 32-byte boundaries for correct operation. For best performance, both registers must be aligned on 256-byte boundaries. See the example following this table.</p>	<a href="#">CCP2_LCM_CTRL</a> , <a href="#">CCP2_LCM_HSIZE</a> , <a href="#">CCP2_LCM_VSIZE</a> , <a href="#">CCP2_LCM_PREFETCH</a> , <a href="#">CCP2_LCM_SRC_ADDR</a> , <a href="#">CCP2_LCM_SRC_OFST</a>	
Select destination.	<a href="#">CCP2_LCM_CTRL[2]</a> DST_PORT	0x0: Video port 0x1: Memory
If destination = video port, configure clock frequency and its gating during frame blanking periods.	<a href="#">CCP2_CTRL[31:15]</a> FRACDIV, <a href="#">CCP2_CTRL[9]</a> VP_CLK_FORCE_ON	
If needed, configure READ_THROTTLE to reduce the bandwidth in memory-to-memory operation to prevent system overload. It has no effect when data are sent to the video port (controlled by video port clock in this case).	<a href="#">CCP2_LCM_CTRL[4:3]</a> READ_THROTTLE	
If the memory write port is used, the destination format and address must be configured.	<a href="#">CCP2_LCM_DST_ADDR</a> , <a href="#">CCP2_LCM_DST_OFST</a>	
Enable memory read channel. After processing a full frame, this bit is automatically cleared by hardware and an EOF event is triggered.	<a href="#">CCP2_LCM_CTRL[0]</a> CHAN_EN	0x1

**Example:**

- [CCP2\\_LCM\\_CTRL\[7:5\]](#) BURST\_SIZE is set to 16 x 64 bits
  - [CCP2\\_LCM\\_HSIZE\[11:0\]](#) SKIP = 0
  - [CCP2\\_LCM\\_HSIZE\[27:16\]](#) COUNT = 1000
  - [CCP2\\_LCM\\_CTRL\[23\]](#) SRC\_PACK = YES
  - [CCP2\\_LCM\\_CTRL\[18:16\]](#) SRC\_FORMAT = RAW6
  - [CCP2\\_LCM\\_PREFETCH\[13:3\]](#) HWORDS = 96 (>=94)
- Setting the size to 94 produces the following burst sequence: 16, 16, 16, 16, 16, 8, and 4 (7 interconnect requests). However, when it is set to 96, the burst sequence is 6 x 16 (6 interconnect requests).

**8.2.4.5 ISS CCP2 Register Manual**

**8.2.4.5.1 ISS CCP2 Instance Summary**

[Table 8-68](#) summarizes the CCP2 instance.

**Table 8-68. ISS CCP2 Instance Summary**

Module Name	L3 Base Address	Size
<a href="#">ISS_CCP2</a>	0x5200 1C00	512 bytes

### 8.2.4.5.2 ISS CCP2 Registers

#### 8.2.4.5.2.1 ISS CCP2 Register Summary

Table 8-69 lists the CCP2 registers.

**Table 8-69. ISS CCP2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CCP2 Base Address
CCP2_REVISION	R	32	0x0000 0000	0x5200 1C00
CCP2_SYSCONFIG	RW	32	0x0000 0004	0x5200 1C04
CCP2_SYSSTATUS	R	32	0x0000 0008	0x5200 1C08
CCP2_LC01_IRQENABLER	RW	32	0x0000 000C	0x5200 1C0C
CCP2_LC01_IRQSTATUS	RW	32	0x0000 0010	0x5200 1C10
CCP2_LC23_IRQENABLER	RW	32	0x0000 0014	0x5200 1C14
CCP2_LC23_IRQSTATUS	RW	32	0x0000 0018	0x5200 1C18
CCP2_LCM_IRQENABLE	RW	32	0x0000 002C	0x5200 1C2C
CCP2_LCM_IRQSTATUS	RW	32	0x0000 0030	0x5200 1C30
CCP2_CTRL	RW	32	0x0000 0040	0x5200 1C40
CCP2_DBG	W	32	0x0000 0044	0x5200 1C44
CCP2_GNQ	R	32	0x0000 0048	0x5200 1C48
CCP2_CTRL1	RW	32	0x0000 004C	0x5200 1C4C
CCP2_LCx_CTRL <sup>(1)</sup>	RW	32	0x0000 0050 + (x * 0x30)	0x5200 1C50 + (x * 0x30)
CCP2_LCx_CODE <sup>(1)</sup>	RW	32	0x0000 0054 + (x * 0x30)	0x5200 1C54 + (x * 0x30)
CCP2_LCx_STAT_START <sup>(1)</sup>	RW	32	0x0000 0058 + (x * 0x30)	0x5200 1C58 + (x * 0x30)
CCP2_LCx_STAT_SIZE <sup>(1)</sup>	RW	32	0x0000 005C + (x * 0x30)	0x5200 1C5C + (x * 0x30)
CCP2_LCx_SOF_ADDR <sup>(1)</sup>	RW	32	0x0000 0060 + (x * 0x30)	0x5200 1C60 + (x * 0x30)
CCP2_LCx_EOF_ADDR <sup>(1)</sup>	RW	32	0x0000 0064 + (x * 0x30)	0x5200 1C64 + (x * 0x30)
CCP2_LCx_DAT_START <sup>(1)</sup>	RW	32	0x0000 0068 + (x * 0x30)	0x5200 1C68 + (x * 0x30)
CCP2_LCx_DAT_SIZE <sup>(1)</sup>	RW	32	0x0000 006C + (x * 0x30)	0x5200 1C6C + (x * 0x30)
CCP2_LCx_DAT_PING_ADDR <sup>(1)</sup>	RW	32	0x0000 0070 + (x * 0x30)	0x5200 1C70 + (x * 0x30)
CCP2_LCx_DAT_PONG_ADDR <sup>(1)</sup>	RW	32	0x0000 0074 + (x * 0x30)	0x5200 1C74 + (x * 0x30)
CCP2_LCx_DAT_OFST <sup>(1)</sup>	RW	32	0x0000 0078 + (x * 0x30)	0x5200 1C78 + (x * 0x30)
CCP2_LCM_CTRL	RW	32	0x0000 01D0	0x5200 1DD0
CCP2_LCM_VSIZE	RW	32	0x0000 01D4	0x5200 1DD4
CCP2_LCM_HSIZE	RW	32	0x0000 01D8	0x5200 1DD8
CCP2_LCM_PREFETCH	RW	32	0x0000 01DC	0x5200 1DDC
CCP2_LCM_SRC_ADDR	RW	32	0x0000 01E0	0x5200 1DE0
CCP2_LCM_SRC_OFST	RW	32	0x0000 01E4	0x5200 1DE4
CCP2_LCM_DST_ADDR	RW	32	0x0000 01E8	0x5200 1DE8
CCP2_LCM_DST_OFST	RW	32	0x0000 01EC	0x5200 1DEC
CCP2_LCM_HISTORY	RW	32	0x0000 01F0	0x5200 1DF0

<sup>(1)</sup> x = 0 to 3

8.2.4.5.2.2 ISS CCP2 Register Description

Table 8-70 through Table 8-134 describe the CCP2 registers.

Table 8-70. CCP2\_REVISION

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See Table 8-69.
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

Table 8-71. Register Call Summary for Register CCP2\_REVISION

- ISS Interfaces
- ISS CCP2 Register Manual: [0]

Table 8-72. CCP2\_SYSCONFIG

<b>Address Offset</b>	0x0000 0004
<b>Physical Address</b>	See Table 8-69
<b>Description</b>	SYSTEM CONFIGURATION REGISTER This register is the OCP-socket system configuration register.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														MSTANDBY_MODE	RESERVED										SOFT_RESET	AUTO_IDLE					

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:12	MSTANDBY_MODE	Sets the behavior of the master port power management signals.  0x0: Force-standby. MStandby is only asserted when the module is disabled.  0x1: No-standby. MStandby is never asserted.  0x2: Smart-standby: MStandby is asserted based on the activity of the module. The module will try to go to standby during the vertical blanking period.	RW	0x0
11:2	RESERVED		R	0x000

Bits	Field Name	Description	Type	Reset
1	SOFT_RESET	Software reset. Set the bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads return 0.  0x0: Normal mode. 0x1: The module is reset	RW	0
0	AUTO_IDLE	Internal OCP clock gating strategy.  0x0: OCP clock is free-running. 0x1: Automatic OCP clock gating strategy is applied based on the OCP interface activity.	RW	1

**Table 8-73. Register Call Summary for Register CCP2\_SYSCONFIG**

ISS Interfaces

- [ISS CCP2 Register Manual: \[0\]](#)

**Table 8-74. CCP2\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	SYSTEM STATUS REGISTER This register provides status information about the module, excluding the interrupt status register.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESET_DONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads returns 0.	R	0x0000 0000
0	RESET_DONE	Internal reset monitoring  Read 0x1: Reset completed. Read 0x0: Internal module reset is on going.	R	1

**Table 8-75. Register Call Summary for Register CCP2\_SYSSTATUS**

ISS Interfaces

- [ISS CCP2 Register Manual: \[0\]](#)

**Table 8-76. CCP2\_LC01\_IRQENABLE**

<b>Address Offset</b>	0x0000 000C
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .
<b>Instance</b>	ISS_CCP2
<b>Description</b>	<p>INTERRUPT ENABLE REGISTER - LOG CHAN 0 &amp; 1</p> <p>This register regroups all the events related to logical channel 0 and logical channel 1. The events related to logical channel 0 trigger SINTERRUPTN[0]. The events related to logical channel 1 trigger SINTERRUPTN[1]. The channel shall be enabled for events to be generated on that channel.</p>
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
LC1_OCPERROR_IRQ	RESERVED					LC1_FS_IRQ	LC1_LE_IRQ	LC1_LS_IRQ	LC1_FE_IRQ	LC1_COUNT_IRQ	RESERVED	LC1_FIFO_OVF_IRQ	LC1_CRC_IRQ	LC1_FSP_IRQ	LC1_FW_IRQ	LC1_FSC_IRQ	RESERVED	LC0_OCPERROR_IRQ	RESERVED					LC0_FS_IRQ	LC0_LE_IRQ	LC0_LS_IRQ	LC0_FE_IRQ	LC0_COUNT_IRQ	RESERVED	LC0_FIFO_OVF_IRQ	LC0_CRC_IRQ	LC0_FSP_IRQ	LC0_FW_IRQ	LC0_FSC_IRQ	RESERVED

Bits	Field Name	Description	Type	Reset
31	LC1_OCPERROR_IRQ	An OCP error occurred on the master write port. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
30:28	RESERVED		R	0x0
27	LC1_FS_IRQ	Logical channel 1 - Frame start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
26	LC1_LE_IRQ	Logical channel 1 - Line end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
25	LC1_LS_IRQ	Logical channel 1 - Line start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
24	LC1_FE_IRQ	Logical channel 1 - Frame end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
23	LC1_COUNT_IRQ	Logical channel 1 - Frame counter reached. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
22	RESERVED		R	0
21	LC1_FIFO_OVF_IRQ	Logical channel 1 - FIFO overflow error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
20	LC1_CRC_IRQ	Logical channel 1 - CRC error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
19	LC1_FSP_IRQ	Logical channel 1 - FSP error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
18	LC1_FW_IRQ	Logical channel 1 - Frame width error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
17	LC1_FSC_IRQ	Logical channel 1 - False sync code error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
16	RESERVED	Reserved	RW	0
15	LC0_OCPERROR_IRQ	An OCP error occurred on the master write port. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
14:12	RESERVED		R	0x0
11	LC0_FS_IRQ	Logical channel 0 - Frame start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
10	LC0_LE_IRQ	Logical channel 0 - Line end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
9	LC0_LS_IRQ	Logical channel 0 - Line start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
8	LC0_FE_IRQ	Logical channel 0 - Frame end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	LC0_COUNT_IRQ	Logical channel 0 - Frame counter reached. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
6	RESERVED		R	0
5	LC0_FIFO_OVF_IRQ	Logical channel 0 - FIFO overflow error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	LC0_CRC_IRQ	Logical channel 0 - CRC error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
3	LC0_FSP_IRQ	Logical channel 0 - FSP error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
2	LC0_FW_IRQ	Logical channel 0 - Frame width error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
1	LC0_FSC_IRQ	Logical channel 0 - False sync code error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	RESERVED	Reserved	RW	0

**Table 8-77. Register Call Summary for Register CCP2\_LC01\_IRQENABLE**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\]](#)
- [ISS CCP2 Register Manual: \[1\]](#)

**Table 8-78. CCP2\_LC01\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0010
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .
<b>Instance</b>	ISS_CCP2
<b>Description</b>	<p>INTERRUPT STATUS REGISTER - LOG CHAN 0 &amp; 1</p> <p>This register regroups all the events related to logical channel 0 and logical channel 1. The events related to logical channel 0 trigger SINTERRUPTN[0]. The events related to logical channel 1 trigger SINTERRUPTN[1]. The channel shall be enabled for events to be generated on that channel.</p>
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
LC1_OCPERROR_IRQ	RESERVED				LC1_FS_IRQ	LC1_LE_IRQ	LC1_LS_IRQ	LC1_FE_IRQ	LC1_COUNT_IRQ	RESERVED	LC1_FIFO_OVF_IRQ	LC1_CRC_IRQ	LC1_FSP_IRQ	LC1_FW_IRQ	LC1_FSC_IRQ	RESERVED	LC0_OCPERROR_IRQ	RESERVED				LC0_FS_IRQ	LC0_LE_IRQ	LC0_LS_IRQ	LC0_FE_IRQ	LC0_COUNT_IRQ	RESERVED	LC0_FIFO_OVF_IRQ	LC0_CRC_IRQ	LC0_FSP_IRQ	LC0_FW_IRQ	LC0_FSC_IRQ	RESERVED

Bits	Field Name	Description	Type	Reset
31	LC1_OCPERROR_IRQ	An OCP error occurred on the master write port. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
30:28	RESERVED		R	0x0
27	LC1_FS_IRQ	Logical channel 1 - Frame start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
26	LC1_LE_IRQ	Logical channel 1 - Line end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
25	LC1_LS_IRQ	Logical channel 1 - Line start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
24	LC1_FE_IRQ	Logical channel 1 - Frame end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
23	LC1_COUNT_IRQ	Logical channel 1 - Frame counter reached status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
22	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
21	LC1_FIFO_OVF_IRQ	Logical channel 1 - FIFO overflow error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
20	LC1_CRC_IRQ	Logical channel 1 - CRC error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
19	LC1_FSP_IRQ	Logical channel 1 - FSP error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
18	LC1_FW_IRQ	Logical channel 1 - Frame width error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
17	LC1_FSC_IRQ	Logical channel 1 - False sync code error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
16	RESERVED	Reserved	RW	0
15	LC0_OCPERROR_IRQ	An OCP error occurred on the master write port. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
14:12	RESERVED		R	0x0
11	LC0_FS_IRQ	Logical channel 0 - Frame start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
10	LC0_LE_IRQ	Logical channel 0 - Line end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
9	LC0_LS_IRQ	Logical channel 0 - Line start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
8	LC0_FE_IRQ	Logical channel 0 - Frame end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0



Bits	Field Name	Description	Type	Reset
7	LC0_COUNT_IRQ	Logical channel 0 - Frame counter reached status 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
6	RESERVED		R	0
5	LC0_FIFO_OVF_IRQ	Logical channel 0 - FIFO overflow error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
4	LC0_CRC_IRQ	Logical channel 0 - CRC error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
3	LC0_FSP_IRQ	Logical channel 0 - FSP error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
2	LC0_FW_IRQ	Logical channel 0 - Frame width error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
1	LC0_FSC_IRQ	Logical channel 0 - False sync code error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
0	RESERVED	Reserved	RW	0

**Table 8-79. Register Call Summary for Register CCP2\_LC01\_IRQSTATUS**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\]](#)
- [ISS CCP2 Register Manual: \[3\]](#)

**Table 8-80. CCP2\_LC23\_IRQENABLE**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	INTERRUPT ENABLE REGISTER - LOG CHAN 2 & 3 This register regroups all the events related to logical channel 2 and logical channel 3. The events related to logical channel 2 trigger SINTERRUPTN[2]. The events related to logical channel 3 trigger SINTERRUPTN[3]. The channel shall be enabled for events to be generated on that channel.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
LC3_OCPERROR_IRQ	RESERVED				LC3_FS_IRQ	LC3_LE_IRQ	LC3_LS_IRQ	LC3_FE_IRQ	LC3_COUNT_IRQ	RESERVED	LC3_FIFO_OVF_IRQ	LC3_CRC_IRQ	LC3_FSP_IRQ	LC3_FW_IRQ	LC3_FSC_IRQ	RESERVED	LC2_OCPERROR_IRQ	RESERVED				LC2_FS_IRQ	LC2_LE_IRQ	LC2_LS_IRQ	LC2_FE_IRQ	LC2_COUNT_IRQ	RESERVED	LC2_FIFO_OVF_IRQ	LC2_CRC_IRQ	LC2_FSP_IRQ	LC2_FW_IRQ	LC2_FSC_IRQ	RESERVED

Bits	Field Name	Description	Type	Reset
31	LC3_OCPERROR_IRQ	An OCP error occurred on the master write port. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
30:28	RESERVED		R	0x0
27	LC3_FS_IRQ	Logical channel 3 - Frame start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
26	LC3_LE_IRQ	Logical channel 3 - Line end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
25	LC3_LS_IRQ	Logical channel 3 - Line start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
24	LC3_FE_IRQ	Logical channel 3 - Frame end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
23	LC3_COUNT_IRQ	Logical channel 3 - Frame counter reached. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
22	RESERVED		R	0
21	LC3_FIFO_OVF_IRQ	Logical channel 3 - FIFO overflow error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
20	LC3_CRC_IRQ	Logical channel 3 - CRC error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
19	LC3_FSP_IRQ	Logical channel 3 - FSP error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
18	LC3_FW_IRQ	Logical channel 3 - Frame width error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
17	LC3_FSC_IRQ	Logical channel 3 - False sync code error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
16	RESERVED	Reserved	RW	0
15	LC2_OCPERROR_IRQ	An OCP error occurred on the master write port. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
14:12	RESERVED		R	0x0
11	LC2_FS_IRQ	Logical channel 2 - Frame start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
10	LC2_LE_IRQ	Logical channel 2 - Line end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
9	LC2_LS_IRQ	Logical channel 2 - Line start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
8	LC2_FE_IRQ	Logical channel 2 - Frame end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	LC2_COUNT_IRQ	Logical channel 2 - Frame counter reached. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
6	RESERVED		R	0
5	LC2_FIFO_OVF_IRQ	Logical channel 2 - FIFO overflow error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	LC2_CRC_IRQ	Logical channel 2 - CRC error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
3	LC2_FSP_IRQ	Logical channel 2 - FSP error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
2	LC2_FW_IRQ	Logical channel 2 - Frame width error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
1	LC2_FSC_IRQ	Logical channel 2 - False sync code error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	RESERVED	Reserved	RW	0

**Table 8-81. Register Call Summary for Register CCP2\_LC23\_IRQENABLE**

## ISS Interfaces

- [ISS CCP2 Programming Model: \[0\]](#)
- [ISS CCP2 Register Manual: \[1\]](#)

**Table 8-82. CCP2\_LC23\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0018
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .
<b>Instance</b>	ISS_CCP2
<b>Description</b>	<p>INTERRUPT STATUS REGISTER - LOG CHAN 2 &amp; 3</p> <p>This register regroups all the events related to logical channel 2 and logical channel 3. The events related to logical channel 2 trigger SINTERRUPTN[2]. The events related to logical channel 3 trigger SINTERRUPTN[3]. The channel shall be enabled for events to be generated on that channel.</p>
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
LC3_OCPERROR_IRQ	RESERVED				LC3_FS_IRQ	LC3_LE_IRQ	LC3_LS_IRQ	LC3_FE_IRQ	LC3_COUNT_IRQ	RESERVED	LC3_FIFO_OVF_IRQ	LC3_CRC_IRQ	LC3_FSP_IRQ	LC3_FW_IRQ	LC3_FSC_IRQ	RESERVED	LC2_OCPERROR_IRQ	RESERVED				LC2_FS_IRQ	LC2_LE_IRQ	LC2_LS_IRQ	LC2_FE_IRQ	LC2_COUNT_IRQ	RESERVED	LC2_FIFO_OVF_IRQ	LC2_CRC_IRQ	LC2_FSP_IRQ	LC2_FW_IRQ	LC2_FSC_IRQ	RESERVED

Bits	Field Name	Description	Type	Reset
31	LC3_OCPERROR_IRQ	An OCP error occurred on the master write port. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
30:28	RESERVED		R	0x0
27	LC3_FS_IRQ	Logical channel 3 - Frame start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
26	LC3_LE_IRQ	Logical channel 3 - Line end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
25	LC3_LS_IRQ	Logical channel 3 - Line start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
24	LC3_FE_IRQ	Logical channel 3 - Frame end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
23	LC3_COUNT_IRQ	Logical channel 3 - Frame counter reached status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
22	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
21	LC3_FIFO_OVF_IRQ	Logical channel 3 - FIFO overflow error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
20	LC3_CRC_IRQ	Logical channel 3 - CRC error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
19	LC3_FSP_IRQ	Logical channel 3 - FSP error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
18	LC3_FW_IRQ	Logical channel 3 - Frame width error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
17	LC3_FSC_IRQ	Logical channel 3 - False sync code error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
16	RESERVED	Reserved	RW	0
15	LC2_OCPERROR_IRQ	An OCP error occurred on the master write port. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
14:12	RESERVED		R	0x0
11	LC2_FS_IRQ	Logical channel 2 - Frame start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
10	LC2_LE_IRQ	Logical channel 2 - Line end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
9	LC2_LS_IRQ	Logical channel 2 - Line start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
8	LC2_FE_IRQ	Logical channel 2 - Frame end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
7	LC2_COUNT_IRQ	Logical channel 2 - Frame counter reached status 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
6	RESERVED		R	0
5	LC2_FIFO_OVF_IRQ	Logical channel 2 - FIFO overflow error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
4	LC2_CRC_IRQ	Logical channel 2 - CRC error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
3	LC2_FSP_IRQ	Logical channel 2 - FSP error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
2	LC2_FW_IRQ	Logical channel 2 - Frame width error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
1	LC2_FSC_IRQ	Logical channel 2 - False sync code error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
0	RESERVED	Reserved	RW	0

**Table 8-83. Register Call Summary for Register CCP2\_LC23\_IRQSTATUS**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\]](#)
- [ISS CCP2 Register Manual: \[3\]](#)

**Table 8-84. CCP2\_LCM\_IRQENABLE**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	INTERRUPT ENABLE REGISTER - Memory channel This register regroups all the events related to the memory channel 2. The events related to memory channel trigger SINTERRUPTN[8]. The channel shall be enabled for events to be generated on that channel.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LCM_OCPERROR	LCM_EOF		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LCM_OCPERROR	An interconnect error has been returned for a read (interconnect read master) or write (interconnect write master) transaction related to LCM operation 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	LCM_EOF	Memory read channel - End of frame 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

**Table 8-85. Register Call Summary for Register CCP2\_LCM\_IRQENABLE**

ISS Interfaces

- [ISS CCP2 Register Manual: \[0\]](#)

**Table 8-86. CCP2\_LCM\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	INTERRUPT STATUS REGISTER - Memory channel This register regroups all the events related to memory channel. The events related to memory channel trigger SINTERRUPTN[8]. The channel shall be enabled for events to be generated on that channel.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												LCM_OCPERROR	LCM_EOF		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LCM_OCPERROR	An interconnect error has been returned for a read (interconnect read master) or write (interconnect write master) transaction related to LCM operation 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
0	LCM_EOF	Memory read channel - End of frame 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

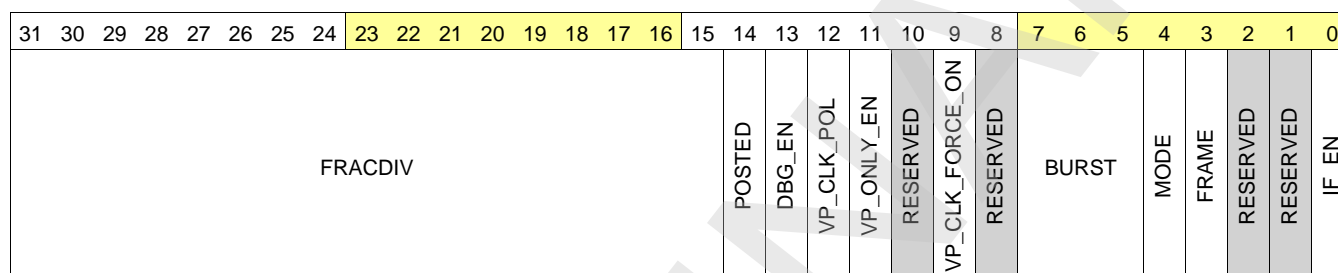
**Table 8-87. Register Call Summary for Register CCP2\_LCM\_IRQSTATUS**

ISS Interfaces

- [ISS CCP2 Register Manual: \[0\]](#)

**Table 8-88. CCP2\_CTRL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	GLOBAL CONTROL REGISTER This register controls the CCP2 receiver. This register shall not be modified dynamically (except IF_EN bit field).		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:15	FRACDIV	Fractional clock divider control for the video port. The mean video port clock is VPBSECLOCK * FRACDIV/65536. Valid range: 1-65536	RW	0x10000
14	POSTED	Selects between posted and non posted writes. 0x0: Non posted 0x1: Posted	RW	0
13	DBG_EN	Enables the debug mode. 0x0: Disable 0x1: Enable	RW	0
12	VP_CLK_POL	VP clock polarity 0x0: The CCP2 receiver writes the data on the VP on the L3 falling edge before the next falling PCLK edge. 0x1: The CCP2 receiver writes the data on the VP on the L3 rising edge before the next rising PCLK edge.	RW	0
11	VP_ONLY_EN	VP only enable. 0x0: The VP is enabled and the OCP master port are enabled. 0x1: The VP is enabled and the OCP master port is disabled. The embedded data and pixel data are output on the VP.	RW	0
10	RESERVED	Read returns reset value	RW	0
9	VP_CLK_FORCE_ON	Controls VP_PCLK gating during frame blanking periods. 0x0: The VP_PCLK is gated during vertical blanking periods. 0x1: The VP_PCLK is free-running during vertical blanking periods.	RW	0
8	RESERVED		R	0



Bits	Field Name	Description	Type	Reset
7:5	BURST	<p>Forces the write burst size used by the module. The write burst size shall never exceed the output FIFO size. The output FIFO size can be read with the <a href="#">CCP2_GNQ.FIFODEPTH</a> bit field.</p> <p>0x0: 1x 64-bit burst = single request.  0x1: 2x 64-bit bursts.  0x3: 8x 64-bit bursts.  0x4: 16x 64-bit bursts.  0x2: 4x 64-bit bursts.</p>	RW	0x0
4	MODE	<p>Selects the receiver operating mode. This bit is only writable when the CCP2MODE input is 1.</p> <p>0x0: MIPI CSI1 compatible mode. When this bit is set all CCP2 settings are ignored. If the settings are not set correctly to MIPI CSI1 values, the behavior of the receiver is unpredictable.  0x1: CCP2 compatible mode</p>	RW	0
3	FRAME	<p>Set the modality in which IF_EN works.</p> <p>0x0: When software writes IF_EN = 0 the interface is disabled immediately.  0x1: When software writes IF_EN = 0 the interface is disabled after the next FEC sync code.</p>	RW	0
2	RESERVED	Read returns reset value	RW	0
1	RESERVED	Read returns reset value	RW	0
0	IF_EN	<p>Enables the physical interface to the module.</p> <p>0x0: The interface is disabled. If FRAME = 0, it is disabled immediately. If FRAME = 1, it is disabled on the next FEC sync code. If FRAME=1, it is advised to disable the logical channels first (CCP2_LCX_CTRL.CHAN_EN=0) before writing IF_EN=0.  0x1: The interface is enabled immediately, the data acquisition starts on the next FSC sync code. Writing 1 to this register when the current value is 0 has the effect to clear the output FIFO. The pixel data of the following frame will be written in the PING buffer, that is, the CCP2_LCX_CTRL.PING_PONG bits are reset to 1 as well.</p>	RW	0

**Table 8-89. Register Call Summary for Register CCP2\_CTRL**

## ISS Interfaces

- [ISS CCP2 Environment: \[0\] \[1\]](#)
- [ISS CCP2 Functional Description: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [ISS CCP2 Programming Model: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\]](#)
- [ISS CCP2 Register Manual: \[39\] \[40\] \[41\]](#)
- [ISS CSI2 Register Manual: \[42\]](#)

**Table 8-90. CCP2\_DBG**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	DEBUG REGISTER This register provides a way to debug the CCP2 receiver with no image sensor connected to the module. The debug mode is enabled by <a href="#">CCP2_CTRL.DBG_EN</a> . Each write to this register provides a full 32bit word to the CCP2 receiver, even when only 8 or 16 bits are written. The newly written value is merged with the previous value (check the example in the programming model section).		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG																															

Bits	Field Name	Description	Type	Reset
31:0	DBG	32-bit input value. Write only register. Reads return 0.	W	0x0000 0000

**Table 8-91. Register Call Summary for Register CCP2\_DBG**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [ISS CCP2 Register Manual: \[8\]](#)

**Table 8-92. CCP2\_GNQ**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	GENERIC PARAMETER REGISTER This register provide a way to read the generic parameters used in the design.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OCPREADPORT	FIFODEPTH	NBCHANNELS					

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5	OCPREADPORT	The OCP master read port, the DPCM encoder and ALAW decompression are only present when this bit is set.	R	1
4:2	FIFODEPTH	Output FIFO size in multiple of 64 bits. Read 0x3: 16 x 64 bits Read 0x4: 32 x 64 bits Read 0x2: 8 x 64 bits Read 0x0: 2 x 64 bits Read 0x1: 4 x 64 bits Read 0x5: 64 x 64 bits	R	0x5

Bits	Field Name	Description	Type	Reset
1:0	NBCHANNELS	Number of logical channels supported by the module. Read 0x3: 8 logical channels Read 0x2: 4 logical channels Read 0x1: 2 logical channels Read 0x0: 1 logical channel	R	0x2

**Table 8-93. Register Call Summary for Register CCP2\_GNQ**

ISS Interfaces

- [ISS CCP2 Register Manual: \[0\] \[1\]](#)

**Table 8-94. CCP2\_CTRL1**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	GLOBAL CONTROL REGISTER (2) This register controls the CCP2 receiver.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
RESERVED																BURST_SIZE_EXPAND	VP_CLK_EN	RESERVED	NON_POSTED_WRITE	RESERVED	VP_ONLY_EN	STREAMING_32_BIT	VP_OUT_CTRL	DBG_EN	BURST_SIZE	ENDIANNESS	FRAME	ECC_EN	RESERVED	IF_EN																

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000
16	BURST_SIZE_EXPAND	Sets the DMA burst size on the L3 interconnect. 0x0: Use the burst size defined in the BURST_SIZE register 0x1: Allow generation of 16x64-bit bursts	RW	0
15	VP_CLK_EN	VP_PCLK enable. 0x0: The VP_PCLK is disabled. 0x1: The VP_PCLK is enabled.	RW	0
14	RESERVED	Read returns reset value	RW	0
13	NON_POSTED_WRITE	Not Posted Writes 0x0: Disable 0x1: Enable	RW	0
12	RESERVED		R	0
11	VP_ONLY_EN	VP only enable. 0x0: The VP is enabled and the ISS interconnect is enabled. 0x1: The VP is enabled and the ISS interconnect is disabled.	RW	0

Bits	Field Name	Description	Type	Reset
10	STREAMING_32_BIT	Indicates if 64-bit or 32-bit streaming burst is used. Valid only if <a href="#">CSI2_CTRL.STREAMING=1</a>  0x0: 64-bit streaming burst is used; byte enable pattern is 0xFF  0x1: 32-bit streaming burst is used; byte enable pattern is 0x0F	RW	0
9:8	VP_OUT_CTRL	VP_PCLK control. Sets the video port output clock as a function of the ISS interconnect interface clock (OCPCLK).  0x0: No division: VP_PCLK = OCPCLK. 0x1: Division by 2: VP_PCLK = OCPCLK / 2. 0x3: Division by 4: VP_PCLK = OCPCLK / 4. 0x2: Division by 3: VP_PCLK = OCPCLK / 3. Example scenarios: - Low VP_PCLK, Memory -> VP: Same as typical memory -> VP, but VP_PCLK = OCPCLK/2 - Typical sensor -> VP: Autoidle enabled, FCLK at optimal rate, sensor provides DPCM compressed RAW12 data at 650 Mbps. Image timings VP_PCLK = (OCPCLK/2) and <a href="#">CCP2_CTRL[31:15] FRACDIV</a> = 0xD000 2600 active pixels/line, 128 blanking pixels, no vertical blanking. (This scenario corresponds to the OTF operation at maximum CCP2 speed. )	RW	0x0
7	DBG_EN	Enables the debug mode.  0x0: Disable 0x1: Enable	RW	0
6:5	BURST_SIZE	Sets the DMA burst size on the L3 interconnect.  0x0: 1x64 OCP writes 0x1: 2x64 OCP writes 0x3: 8x64 OCP writes 0x2: 4x64 OCP writes	RW	0x0
4	ENDIANNESS	Select endianness for YUV4:2:2 8 bit and YUV4:2:0 legacy formats.  0x0: Use native MIPI CSI2 endianness: Little endian for all formats except for YUV4:2:2 8b and YUV4:2:0 Legacy which a big endian. 0x1: Store all pixel formats little endian.	RW	0
3	FRAME	Set the modality in which IF_EN works.  0x0: If IF_EN = 0 the interface is disabled immediately. 0x1: If IF_EN = 1 the interface is disabled after all FEC sync code have been received for the active contexts.	RW	0
2	ECC_EN	Enables the Error Correction Code check for the received header (short and long packets for all virtual channel ids).  0x0: Disabled 0x1: Enabled	RW	0
1	RESERVED	Read returns reset value	RW	0
0	IF_EN	Enables the physical interface to the module.  0x0: The interface is disabled. If FRAME = 0, it is disabled immediately. If FRAME = 1, it is disabled when each context has received the FEC sync code.  0x1: The interface is enabled immediately, the data acquisition starts on the next FSC sync code. Writing 1 to this register when the current value is 0 has the effect to clear the output FIFO. The pixel data of the following frame will be written in the PING buffer, that is, the <a href="#">CSI2_CTX_CTRL.PING_PONG</a> bits are reset to 0 as well.	RW	0

**Table 8-95. Register Call Summary for Register CCP2\_CTRL1**

- ISS Interfaces
- [ISS CCP2 Functional Description: \[0\] \[1\] \[2\] \[3\]](#)
  - [ISS CCP2 Register Manual: \[4\]](#)

**Table 8-96. CCP2\_LCx\_CTRL**

<b>Address Offset</b>	0x0000 0050 + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	CONTROL REGISTER - LOG CHAN 0 This register controls the logical channel 0. This register is shadowed: modifications are taken into account after the next FSC sync code.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT								RESERVED				CRC_EN	DPCM_PRED	PING_PONG	COUNT_UNLOCK	ALPHA								FORMAT				REGION_EN	CHAN_EN		

Bits	Field Name	Description	Type	Reset
31:24	COUNT	Sets the number of frame to acquire. Once the frame acquisition starts, the COUNT value is decremented after every frame. When COUNT reaches 0, the COUNT_IRQ interrupt is triggered and CHAN_EN is set to 0. Writes to this bit field are controlled by the COUNT_UNLOCK bit. COUNT can be overwritten dynamically with a new count value. 0: Infinite number of frames (no count). 1: 1 frame to acquire ... 255: 255 frames to acquire.	RW	0x00
23:20	RESERVED		R	0x0
19	CRC_EN	Enables the cyclic redundancy check. 0x0: Disabled 0x1: Enabled	RW	0
18	DPCM_PRED	Selects the DPCM predictor to be used for the RAW6+DPCM10, RAW7+DPCM10 and RAW8+DPCM12 data formats. The RAW8+DPCM10 data format always use the simple predictor. 0x0: The advanced predictor is used 0x1: The simple predictor is used.	RW	0
17	PING_PONG	Indicates whether the PING or PONG destination address (CCP2_LC0_DAT_PING_ADDR or CCP2_LC0_DAT_PONG_ADDR) was used to write the last frame. This bit field toggles after every FEC sync code. Read 0x1: PONG buffer Read 0x0: PING buffer	R	1
16	COUNT_UNLOCK	Unlock writes to the COUNT bit field. Write 0x0: COUNT bit field is locked. Writes have no effect Write 0x1: COUNT bit field is unlocked. Writes are possible.	W	0
15:8	ALPHA	Alpha value for RGB888 and RBG444.	RW	0x00

Bits	Field Name	Description	Type	Reset
7:2	FORMAT	Data format selection. 0xD: RAW7 + DPCM10 + EXP16 0x15: RAW10 + EXP16 0x1E: RAW10 -> RAW8 RAW10 data from sensor is right shifted to produce RAW8 before it is send to memory 0x8: RAW6 + EXP8 0x5: RGB565 0x1B: RAW10 -> RAW8 DPCM RAW10 data from sensor is DPCM compressed into RAW8 before it is send to memory. 0x2: YUV4:2:0 0x4: RGB444 + EXP16 0x6: RGB888 0x1: YUV4:2:2 LITTLE ENDIAN 0x1D: JPEG8 0x0: YUV4:2:2 BIG ENDIAN 0xB: RAW10 -> RAW6 DPCM RAW10 data from sensor is DPCM compressed into RAW6 before it is send to memory. Used predictor is selected by the DPCM_PRED bit. 0x20: RAW10 -> RAW8 ALAW 0x3: YUV4:2:2 + VP or RAW8 + VP 0x17: RAW10 -> RAW7 DPCM + EXP8 RAW10 data from sensor is DPCM compressed into RAW7 and expanded to 8 bits before it is send to memory. Used predictor is selected by the DPCM_PRED bit. 0x11: RAW8 + DPCM10 + EXP16 0xA: RAW6 + DPCM10 + VP 0x9: RAW6 + DPCM10 + EXP16 0x10: RAW8 This mode can be used to output RAW6 and RAW7 as well. 0x21: RAW8 DPCM10 -> ALAW 0x12: RAW8 + DPCM10 + VP 0x13: RAW10 -> RAW7 DPCM RAW10 data from sensor is DPCM compressed into RAW7 before it is send to memory. Used predictor is selected by the DPCM_PRED bit. 0x18: RAW12 0x14: RAW10 0xE: RAW7 + DPCM10 + VP 0x16: RAW10 + VP 0x1C: JPEG8 + FSP 0x7: RGB888 + EXP32 0x19: RAW12 + EXP16 0x1F: RAW8 DPCM12 -> RAW12 + VP Used predictor is selected by the DPCM_PRED bit. 0x1A: RAW12 + VP 0xF: RAW10 -> RAW6 DPCM + EXP8 RAW10 data from sensor is DPCM compressed into RAW6 and expanded to 8 bits before it is send to memory. Used predictor is selected by the DPCM_PRED bit. 0xC: RAW7 + EXP8	RW	0x00

Bits	Field Name	Description	Type	Reset
1	REGION_EN	Enables the setting of regions of interest in the frame: SOF region, EOF region and DAT region. 0x0: Disabled 0x1: Enabled	RW	0
0	CHAN_EN	Enables the logical channel 0x0: Disabled 0x1: Enabled	RW	0x1 for LC0 0x0 for LC1 0x0 for LC2 0x0 for LC3

**Table 8-97. Register Call Summary for Register CCP2\_LCx\_CTRL**

ISS Interfaces

- [ISS CCP2 Environment: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [ISS CCP2 Functional Description: \[16\]](#)
- [ISS CCP2 Programming Model: \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\]](#)
- [ISS CCP2 Register Manual: \[37\]](#)

**Table 8-98. CCP2\_LCx\_CODE**

<b>Address Offset</b>	0x0000 0054 + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	CODE REGISTER - LOG CHAN 0 This register sets the codes that are used in the 32-bit synchronization codes to recognize the logical channel, frame start, frame end, line start and line end codes. This register applies for logical channel 0 only. The default values are usually not supposed to be modified. Updating this register with new codes under a flowing serial transmission on that channel will cause unexpected result.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CHAN_ID				FEC				FSC				LEC			LSC								

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:16	CHAN_ID	Log chan 0 identifier. The channel identifier is located between bits 4 to 7 in the 32-bit synchronization codes.	RW	0x0
15:12	FEC	Log chan 0 frame end sync code identifier. The sync code identifier is located between bits 0 to 3 in the 32-bit synchronization codes.	RW	0x3
11:8	FSC	Log chan 0 frame start sync code identifier. The sync code identifier is located between bits 0 to 3 in the 32-bit synchronization codes.	RW	0x2
7:4	LEC	Log chan 0 line end sync code identifier. The sync code identifier is located between bits 0 to 3 in the 32-bit synchronization codes.	RW	0x1
3:0	LSC	Log chan 0 line start sync code identifier. The sync code identifier is located between bits 0 to 3 in the 32-bit synchronization codes.	RW	0x0

**Table 8-99. Register Call Summary for Register CCP2\_LCx\_CODE**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [ISS CCP2 Register Manual: \[6\]](#)

**Table 8-100. CCP2\_LCx\_STAT\_START**

<b>Address Offset</b>	0x0000 0058 + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	STATUS LINE START REGISTER - LOG CHAN 0 This register is shadowed: modifications are taken into account after the next FSC sync code.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EOF								RESERVED								SOF							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	EOF	Sets the vertical position of the EOF status lines in regards of the FSC sync code. From 0 to 4095.	RW	0x000
15:12	RESERVED		R	0x0
11:0	SOF	Sets the vertical position of the EOF status lines in regards of the FSC sync code. Should always be 0.	RW	0x000

**Table 8-101. Register Call Summary for Register CCP2\_LCx\_STAT\_START**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS CCP2 Register Manual: \[4\]](#)

**Table 8-102. CCP2\_LCx\_STAT\_SIZE**

<b>Address Offset</b>	0x0000 005C + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	STATUS LINE SIZE REGISTER - LOG CHAN 0 This register is shadowed: modifications are taken into account after the next FSC sync code.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EOF								RESERVED								SOF							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	EOF	Sets the number of EOF status lines From 0 to 4095	RW	0x000
15:12	RESERVED		R	0x0
11:0	SOF	Sets the number of SOF status line(s) From 0 to 4095	RW	0x000

**Table 8-103. Register Call Summary for Register CCP2\_LCx\_STAT\_SIZE**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS CCP2 Register Manual: \[5\]](#)



**Table 8-104. CCP2\_LCx\_SOF\_ADDR**

<b>Address Offset</b>	0x0000 0060 + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	SOF STATUS LINE MEM ADDRESS REGISTER - LOG CHAN 0 This register sets the 32-bit memory address where the SOF data are stored. The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary. This register is shadowed: modifications are taken into account after the next FSC sync code.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most significant bits of the 32-bit address.	RW	0x00000000
4:0	RESERVED		R	0x00

**Table 8-105. Register Call Summary for Register CCP2\_LCx\_SOF\_ADDR**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\]](#)
- [ISS CCP2 Register Manual: \[2\]](#)

**Table 8-106. CCP2\_LCx\_EOF\_ADDR**

<b>Address Offset</b>	0x0000 0064 + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	EOF STATUS LINE MEM ADDRESS REGISTER - LOG CHAN 0 This register sets the 32-bit memory address where the EOF data are stored. The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary. This register is shadowed: modifications are taken into account after the next FSC sync code.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most significant bits of the 32-bit address.	RW	0x00000000
4:0	RESERVED		R	0x00

**Table 8-107. Register Call Summary for Register CCP2\_LCx\_EOF\_ADDR**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\]](#)
- [ISS CCP2 Register Manual: \[2\]](#)

**Table 8-108. CCP2\_LCx\_DAT\_START**

<b>Address Offset</b>	0x0000 0068 + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	DATA START REGISTER - LOG CHAN 0 This register is shadowed: modifications are taken into account after the next FSC sync code.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERT								RESERVED															

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	VERT	Sets the vertical position of the data in regards of the FSC sync code. From 0 to 4095 lines.	RW	0x000
15:0	RESERVED		R	0x0000

**Table 8-109. Register Call Summary for Register CCP2\_LCx\_DAT\_START**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\]](#)
- [ISS CCP2 Register Manual: \[2\]](#)

**Table 8-110. CCP2\_LCx\_DAT\_SIZE**

<b>Address Offset</b>	0x0000 006C + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	DATA SIZE REGISTER - LOG CHAN 0 This register is shadowed: modifications are taken into account after the next FSC sync code.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERT								RESERVED															

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	VERT	Sets the vertical size of the data window. From 0 to 4095 lines. If VERT = "0", no data is output.	RW	0x000
15:0	RESERVED		R	0x0000

**Table 8-111. Register Call Summary for Register CCP2\_LCx\_DAT\_SIZE**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\]](#)
- [ISS CCP2 Register Manual: \[3\]](#)

**Table 8-112. CCP2\_LCx\_DAT\_PING\_ADDR**

<b>Address Offset</b>	0x0000 0070 + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	<p>DATA MEM PING ADDRESS REGISTER - LOG CHAN 0                  This register sets the 32-bit memory address where the pixel data are stored. The destination is double buffered: this register sets the PING address. Double buffering is enabled when the addresses CCP2_LC0_DAT_PING_ADDR and CCP2_LC0_DAT_PONG_ADDR are different.                  The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary.                  This register is shadowed: modifications are taken into account after the next FSC sync code.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most significant bits of the 32-bit address.	RW	0x00000000
4:0	RESERVED		R	0x00

**Table 8-113. Register Call Summary for Register CCP2\_LCx\_DAT\_PING\_ADDR**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [ISS CCP2 Register Manual: \[6\]](#)

**Table 8-114. CCP2\_LCx\_DAT\_PONG\_ADDR**

<b>Address Offset</b>	0x0000 0074 + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	<p>DATA MEM PONG ADDRESS REGISTER - LOG CHAN 0                  This register sets the 32-bit memory address where the pixel data are stored. The destination is double buffered: this register sets the PONG address. Double buffering is enabled when the addresses CCP2_LC0_DAT_PING_ADDR and CCP2_LC0_DAT_PONG_ADDR are different.                  The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary.                  This register is shadowed: modifications are taken into account after the next FSC sync code.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most significant bits of the 32-bit address.	RW	0x00000000
4:0	RESERVED		R	0x00

**Table 8-115. Register Call Summary for Register CCP2\_LCx\_DAT\_PONG\_ADDR**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [ISS CCP2 Register Manual: \[6\]](#)

**Table 8-116. CCP2\_LCx\_DAT\_OFST**

<b>Address Offset</b>	0x0000 0078 + (x * 0x30)	<b>Index</b>	x = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	<p>DATA MEM ADDRESS OFFSET REGISTER - LOG CHAN 0</p> <p>This register sets the offset, which is applied on the destination address after each line is written to memory. This register applies for both CCP2_LC0_DAT_PING_ADDR and CCP2_LC0_DAT_PONG_ADDR.</p> <p>For example, it enables to perform 2D data transfers of the pixel data into a frame buffer. In such case, the pixel data and frame buffer data shall have the same data format.</p> <p>The 5 LSBs are ignored: the offset shall be a multiple of 32 bytes.</p> <p>Only full 64-bits words are written to memory at the end of lines.</p> <p>This register is shadowed: modifications are taken into account after the next FSC sync code.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFST																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	OFST	Line offset programmed in bytes. If OFST = 0, the data is written contiguously in memory. Otherwise, OFST sets the destination offset between the first pixel of the previous line and the first pixel of the current line. NOTE: An OCP access (read/write) is required to properly update the <a href="#">CCP2_LCx_DAT_OFST</a> register	RW	0x0000000
4:0	RESERVED		R	0x00

**Table 8-117. Register Call Summary for Register CCP2\_LCx\_DAT\_OFST**

ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [ISS CCP2 Register Manual: \[8\] \[9\]](#)

**Table 8-118. CCP2\_LCM\_CTRL**

<b>Address Offset</b>	0x0000 01D0	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	Control register for the memory channel. It defines the data format of the source frame stored in memory and how this frame is processed.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST_PACK	DST_DPCM_PRED	DST_COMPR	RESERVED	DST_FORMAT	SRC_PACK	SRC_DPCM_PRED	SRC_DECOMPR	SRC_FORMAT				RESERVED								BURST_SIZE	READ_THROTTLE	DST_PORT	RESERVED	CHAN_EN							

Bits	Field Name	Description	Type	Reset
31	DST_PACK	Data is packed before it is send to memory. Applies to RAW6, RAW7, RAW10, and RAW12 only.  0x0: Disabled 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
30	DST_DPCM_PRED	Selects the DPCM predictor to be used for the RAW6+DPCM10 and RAW7+DPCM10 data formats. The RAW8+DPCM10 data format always use the simple predictor. 0x0: The advanced predictor is used 0x1: The simple predictor is used.	RW	0
29:28	DST_COMPR	Enables data compression of data sent to memory 0x0: No compression 0x1: A-Law compression RAW10 -> RAW8 A-Law when DST_FORMAT=RAW8 other destination formats are invalid. 0x2: DPCM compression RAW10 -> RAW6 DPCM when DST_FORMAT=RAW6 RAW10 -> RAW7 DPCM when DST_FORMAT=RAW7 RAW10 -> RAW8 DPCM when DST_FORMAT=RAW8 other destination formats are invalid.	RW	0x0
27	RESERVED		R	0
26:24	DST_FORMAT	Output format selection. Not every combination between input and output formats are possible. 0x6: RAW16 0x1: RAW7 0x0: RAW6 0x2: RAW8 0x4: RAW12 0x5: RAW14 0x3: RAW10	RW	0x0
23	SRC_PACK	Data stored in memory is packed and must be unpacked. 0x0: Disabled 0x1: Enabled	RW	0
22	SRC_DPCM_PRED	Selects the DPCM predictor to be used for the RAW6+DPCM10, RAW7+DPCM10 and RAW8+DPCM12 data formats. The RAW8+DPCM10 and RAW6 + DPCM12 data format always use the simple predictor. 0x0: The advanced predictor is used 0x1: The simple predictor is used.	RW	0
21:20	SRC_DECOMPR	Enable decompression of incoming data 0x0: No decompression 0x1: A-Law decompression RAW8 A-Law -> RAW10 when SRC_FORMAT=RAW8 other source formats are invalid. 0x3: DPCM decompression RAW6 DPCM -> RAW12 when SRC_FORMAT=RAW6 RAW8 DPCM -> RAW12 when SRC_FORMAT=RAW8 other source formats are invalid. 0x2: DPCM decompression RAW6 DPCM -> RAW10 when SRC_FORMAT=RAW6 RAW7 DPCM -> RAW10 when SRC_FORMAT=RAW7 RAW8 DPCM -> RAW10 when SRC_FORMAT=RAW8 other source formats are invalid.	RW	0x0

Bits	Field Name	Description	Type	Reset
19:16	SRC_FORMAT	Data format of the data stored in memory. As there is no header embedded in the data sent to memory the user is responsible of choosing the adequate format.  0x6: RAW16 0x1: RAW7 0xA: Reserved 0x7: Reserved 0xD: Reserved 0x0: RAW6 0x2: RAW8 0x8: Reserved 0x9: Reserved 0xB: Reserved 0x4: RAW12 0x5: RAW14 0xF: Reserved 0xC: Reserved 0x3: RAW10 0xE: Reserved	RW	0x0
15:8	RESERVED		R	0x00
7:5	BURST_SIZE	Defines the burst size of the master read port  0x0: 1x 64-bit burst = single request. 0x1: 2x 64-bit bursts. 0x3: 8x 64-bit bursts. 0x4: 16x 64-bit bursts. 0x2: 4x 64-bit bursts.	RW	0x0
4:3	READ_THROTTLE	Limit maximum data read speed for memory to memory operation  0x0: Full speed. Throughput is limited by internal processing capabilities. 0x1: 1/2 speed 0x3: 1/8 speed 0x2: 1/4 speed	RW	0x0
2	DST_PORT	Select the destination port  0x0: Data is send to video port, it is always send without compression or packing. The DST_COMPR, DST_DPCM_PRED, DST_PACK, CCP2_LCM_DST_WRITE, and CCP2_LCM_DST_OFST registers have no effect. 0x1: Data is send to memory.	RW	0
1	RESERVED		R	0
0	CHAN_EN	Enables the read from memory channel. Before enabling the memory read channel software shall: - disable the physical interface using the IF_EN bit - wait until disabling of the physical interface is effective (depends on the FRAME bit) Read from memory starts as soon as this bit is set, therefore all CCP2_LCM_x registers must be configured correctly before. This bit is cleared by hardware at the end of the frame.  0x0: Disabled 0x1: Enabled	RW	0

**Table 8-119. Register Call Summary for Register CCP2\_LCM\_CTRL**

ISS Interfaces

- [ISS CCP2 Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [ISS CCP2 Programming Model: \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\]](#)
- [ISS CCP2 Register Manual: \[17\]](#)

**Table 8-120. CCP2\_LCM\_VSIZE**

<b>Address Offset</b>	0x0000 01D4	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	Memory channel vertical framing register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COUNT								RESERVED															

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	COUNT	Defines the line count to be read from memory. From 1 to 8191 lines.	RW	0x0001
15:0	RESERVED		R	0x0000

**Table 8-121. Register Call Summary for Register CCP2\_LCM\_VSIZE**

ISS Interfaces

- [ISS CCP2 Functional Description: \[0\]](#)
- [ISS CCP2 Programming Model: \[1\]](#)
- [ISS CCP2 Register Manual: \[2\]](#)

**Table 8-122. CCP2\_LCM\_HSIZE**

<b>Address Offset</b>	0x0000 01D8	<b>Instance</b>	ISS_CCP2
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .		
<b>Description</b>	Memory read channel horizontal framing register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COUNT								RESERVED								SKIP							

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:16	COUNT	Horizontal count of samples to output after the skipped pixels. Valid values: 1 to 32767.	RW	0x0001
15	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
14:0	SKIP	Horizontal count of samples to skip after the start of the line. When DPCM compressed data is read from memory using this feature is the only valid way to set a horizontal starting position. Valid values: 0 to 32767. 0 disables pixel skipping	RW	0x0000

**Table 8-123. Register Call Summary for Register CCP2\_LCM\_HSIZE**

## ISS Interfaces

- [ISS CCP2 Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [ISS CCP2 Programming Model: \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [ISS CCP2 Register Manual: \[12\] \[13\]](#)

**Table 8-124. CCP2\_LCM\_PREFETCH**

<b>Address Offset</b>	0x0000 01DC	
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b> ISS_CCP2
<b>Description</b>	This register defines the amount of data to be fetched from memory. It must be consistent with the <a href="#">CCP2_LCM_HSIZE</a> register (check programming model).	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HWORDS											RESERVED				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	HWORDS	64 bit words to read from memory for each line of the image. Possible values 1..8191	RW	0x0001
2:0	RESERVED		R	0x0

**Table 8-125. Register Call Summary for Register CCP2\_LCM\_PREFETCH**

## ISS Interfaces

- [ISS CCP2 Functional Description: \[0\]](#)
- [ISS CCP2 Programming Model: \[1\] \[2\]](#)
- [ISS CCP2 Register Manual: \[3\]](#)



**Table 8-126. CCP2\_LCM\_SRC\_ADDR**

<b>Address Offset</b>	0x0000 01E0		
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	Memory channel source address register This register sets the 32-bit memory address where the pixel data are stored. The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most-significant bits of the 32-bit address	RW	0x00000000
4:0	RESERVED		R	0x00

**Table 8-127. Register Call Summary for Register CCP2\_LCM\_SRC\_ADDR**

ISS Interfaces

- [ISS CCP2 Functional Description: \[0\] \[1\]](#)
- [ISS CCP2 Programming Model: \[2\] \[3\]](#)
- [ISS CCP2 Register Manual: \[4\]](#)

**Table 8-128. CCP2\_LCM\_SRC\_OFST**

<b>Address Offset</b>	0x0000 01E4		
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	Memory channel source offset register. This register sets the offset, which is applied on the source address after each line is read from memory. For example, it enables to perform 2D data transfers of the pixel data from a frame buffer. In such case, the pixel data and frame buffer data shall have the same data format. The 5 LSBs are ignored: the offset shall be a multiple of 32 bytes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFST																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	OFST	Line offset programmed in bytes. If OFST = 0, the data is read contiguously from memory. Otherwise, OFST sets the source offset between the first pixel of the previous line and the first pixel of the current line.	RW	0x00000000
4:0	RESERVED		R	0x00

**Table 8-129. Register Call Summary for Register CCP2\_LCM\_SRC\_OFST**

ISS Interfaces

- [ISS CCP2 Functional Description: \[0\]](#)
- [ISS CCP2 Programming Model: \[1\] \[2\]](#)
- [ISS CCP2 Register Manual: \[3\]](#)

**Table 8-130. CCP2\_LCM\_DST\_ADDR**

<b>Address Offset</b>	0x0000 01E8		
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	Memory channel destination address. This register sets the 32-bit memory address where the pixel data are stored. The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most significant bits of the 32-bit address.	RW	0x0000000
4:0	RESERVED		R	0x00

**Table 8-131. Register Call Summary for Register CCP2\_LCM\_DST\_ADDR**

ISS Interfaces

- [ISS CCP2 Functional Description: \[0\] \[1\]](#)
- [ISS CCP2 Programming Model: \[2\]](#)
- [ISS CCP2 Register Manual: \[3\]](#)

**Table 8-132. CCP2\_LCM\_DST\_OFST**

<b>Address Offset</b>	0x0000 01EC		
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .	<b>Instance</b>	ISS_CCP2
<b>Description</b>	Memory channel destination offset register. This register sets the offset, which is applied on the destination address after each line is written to memory. For example, it enables to perform 2D data transfers of the pixel data into a frame buffer. In such case, the pixel data and frame buffer data shall have the same data format. The 5 LSBs are ignored: the offset shall be a multiple of 32 bytes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFST																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	OFST	Line offset programmed in bytes. If OFST = 0, the data is written contiguously to memory if possible. At the end of a line only full 32 bit words will be written, creating eventually gaps at the end of lines. Otherwise, OFST sets the destination offset between the first pixel of the previous line and the first pixel of the current line.	RW	0x0000000
4:0	RESERVED		R	0x00

**Table 8-133. Register Call Summary for Register CCP2\_LCM\_DST\_OFST**

ISS Interfaces

- [ISS CCP2 Functional Description: \[0\]](#)
- [ISS CCP2 Programming Model: \[1\]](#)
- [ISS CCP2 Register Manual: \[2\] \[3\]](#)

**Table 8-134. CCP2\_LCM\_HISTORY**

<b>Address Offset</b>	0x0000 01F0
<b>Physical Address</b>	See <a href="#">Table 8-69</a> .
<b>Description</b>	Controls operation of the DPCM history read/write feature
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EN_HIST_RD	EN_HIST_WR	HIST_EXPORT															

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17	EN_HIST_RD	Enable DPCM history read 0x0: Disable 0x1: Enable	RW	0
16	EN_HIST_WR	Enable DPCM history write 0x0: Disable 0x1: Enable	RW	0
15:0	HIST_EXPORT	Defines the horizontal position at which DPCM history information is written. The first decoded sample of a line has position 0 The last decoded sample has position SKIP+COUNT-1 Valid range [3..SKIP+COUNT-1]	RW	0x0000

**Table 8-135. Register Call Summary for Register CCP2\_LCM\_HISTORY**

ISS Interfaces

- [ISS CCP2 Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [ISS CCP2 Register Manual: \[9\]](#)

## 8.2.5 ISS CSI2

### 8.2.5.1 ISS CSI2 Environment

#### 8.2.5.1.1 ISS CSI2 Protocol and Data Format

The CSI2 supports MIPI CSI2 multiple data type formats. This section describes MIPI CSI2 protocol and data formats. [Table 8-136](#) lists the MIPI CSI2 supported by CSI2 formats in addition to JPEG8. Shading in the primary and secondary MIPI CSI2-defined formats indicates special format extensions of the CSI2 receiver.

[Table 8-136](#) summarizes the pixel formats supported by the CSI2 receiver interface.

**Table 8-136. ISS CSI2 Pixel Format Modes**

CSI2_CTX_CT RL2_i[9:0] Format	CSI2 Data Format	Bits per Pixel (BPP)	Data Size Increases in Memory	2D Mode Availability <sup>(1)</sup>	Comments
0x18	YUV4:2:0 8 bit	12	0%	Yes	
0x19	YUV4:2:0 10 bit	12	0%	Yes	
0x1E	YUV4:2:2 8 bit	16	0%	Yes	
0x1F	YUV4:2:2 10 bit	16	0%	Yes	
0x22	RGB565	16	0%	Yes	
0x24	RGB888	24	0%	Yes	
0x29	RAW7	7	0%	Yes	
0x2A	RAW8	8	8%	Yes	
0x2B	RAW10	10	0%	Yes	
0x2C	RAW12	12	0%	Yes	
0x2D	RAW14	14	0%	Yes	
0xA3	RGB666 + EXP32	32	77%	Yes	
0x68	RAW6 + EXP8	8	33%	Yes	
0x69	RAW7 + EXP8	8	14%	Yes	
0xA0	RGB444 + EXP16	16	33%	Yes	
0xA1	RGB555 + EXP16	16	6%	Yes	
0xAB	RAW10 + EXP16	16	60%	Yes	
0xAC	RAW12 + EXP16	16	33%	Yes	
0xAD	RAW14 + EXP16	16	14%	Yes	
0xE3	RGB666 + EXP32	32	77%	Yes	
0xE4	RGB888 + EXP32	32	33%	Yes	
0x2A8	RAW6 + DPCM10 + EXP16	16	166%	Yes	DPCM decompression
0x229	RAW7 + DPCM10 + EXP16	16	128%	Yes	DPCM decompression
0x2AA	RAW8 + DPCM10 + EXP16	16	100%	Yes	DPCM decompression
0x369	RAW7 + DPCM12 + EXP16	16	128%	Yes	DPCM decompression
0x36A	RAW8 + DPCM12 + EXP16	16	100%	Yes	DPCM decompression
0x3A8	RAW6 + DPCM12 + EXP16	16	166%	Yes	DPCM decompression
0x12	JPEG8	8	0%		

<sup>(1)</sup> If 2D mode is available, there are no supplementary constraints on data width. 2D mode does not apply when sending to the video port (VP).

For more information about how the data formats are transmitted and how the data are stored in memory, see [Section 8.2.5.1.1.4, CSI2 Operating Modes](#).

---

**NOTE:** The VP formats are not included in [Table 8-136](#), because they are not sent to memory; instead they are sent to the ISP for further processing.

---



---

**NOTE:** Data written by CSI2 can be read back by the CCP2 read channel. CCP2 can get data from a sensor or memory. For more information about supported memory operations, see [Table 8-52](#).

---

### 8.2.5.1.1.1 ISS CSI2 Physical Layer

The CSI2-A/CSI2-B receivers are tightly connected to a PHY layer (for more information about the PHY, see [Section 8.2.3, ISS CSI2 PHY](#)). [Table 8-137](#) lists the CSI2-A receiver I/O, and [Table 8-138](#) lists the CSI2-B I/O. The CSI2\_RECEIVER provides access to the complex I/O register through a serial configuration port interface. Accesses performed to the CSI2\_PHY\_CFG\_i register are forwarded to the complex I/O register at address i. The CSI2\_RECEIVER can access up to eight 32-bit registers in each complex I/O (up to two complex I/Os supported).

**Table 8-137. ISS CSI2-A I/O Description**

Signal Name		I/O <sup>(1)</sup>	Description
csi2a_dx0	line 0	I	Serial data/clock input
csi2a_dy0			
csi2a_dx1	line 1	I	Serial data/clock input
csi2a_dy1			
csi2a_dx2	line 2	I	Serial data/clock input
csi2a_dy2			
csi2a_dx3	line 3	I	Serial data/clock input
csi2a_dy3			
csi2a_dx4	line 4	I	Serial data input
csi2a_dy4			

<sup>(1)</sup> I = Input

**Table 8-138. ISS CSI2-B I/O Description**

Signal Name		I/O <sup>(1)</sup>	Description
csi2b_dx0	line 0	I	Serial data/clock input
csi2b_dy0			
csi2b_dx1	line 1	I	Serial data/clock input
csi2b_dy1			

<sup>(1)</sup> I = Input

---

**NOTE:** The serial line can be used as clock lane or data lane (excluding line 4 on the CSI2-A I/O). The MIPI CSI2 protocol requires one clock line (others are data lane or unused lane).

---

Lanes support the two operating modes:

- HS mode: High-speed transmit mode
- Off mode: Lane is off.

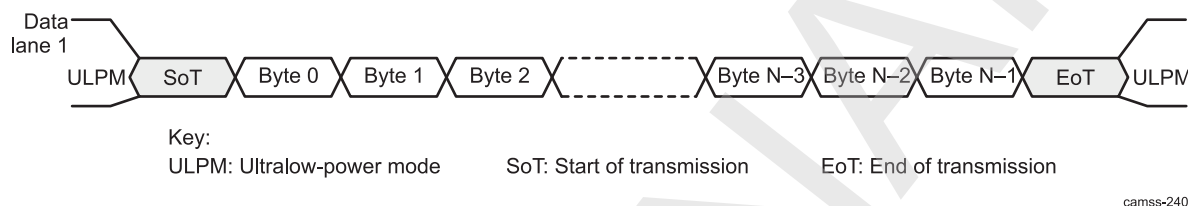
### 8.2.5.1.1.2 ISS CSI2 Lane Merger

The layer consists of lane merger logic to merge the incoming serial stream into a byte stream. The lane merger can merge up to four lanes (CSI2-A) into a single byte stream. The bits are sent with the LSB first. The order of the lanes at the CSI2-A, CSI2-B receiver core depends on the lane configuration. The merger is not used for a single lane.

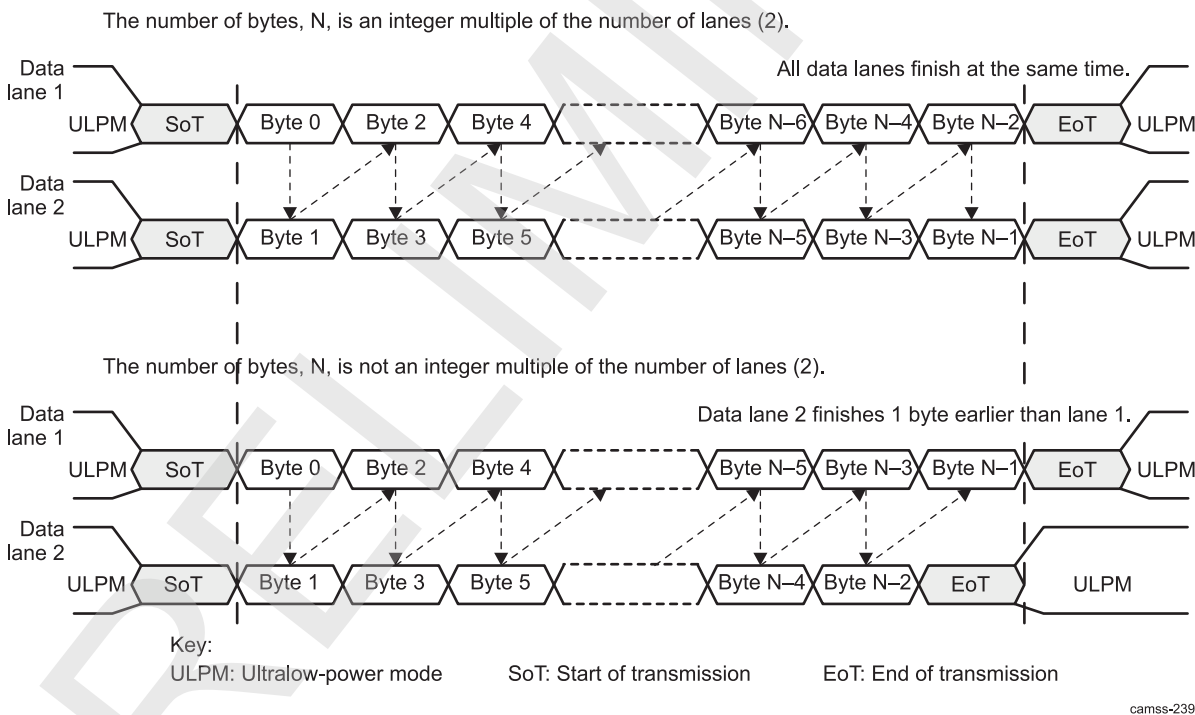
The number of lanes and their configuration can be changed only in ULPM or when all data lanes are in off mode.

Figure 8-32 to Figure 8-35 show the byte position into each serial link for one to four data lane configurations. The byte stream always starts from lane 1. It finishes on one of the lanes, depending on the number of bytes to receive and the number of lanes.

**Figure 8-32. ISS CSI2 One Data-Lane Configuration**

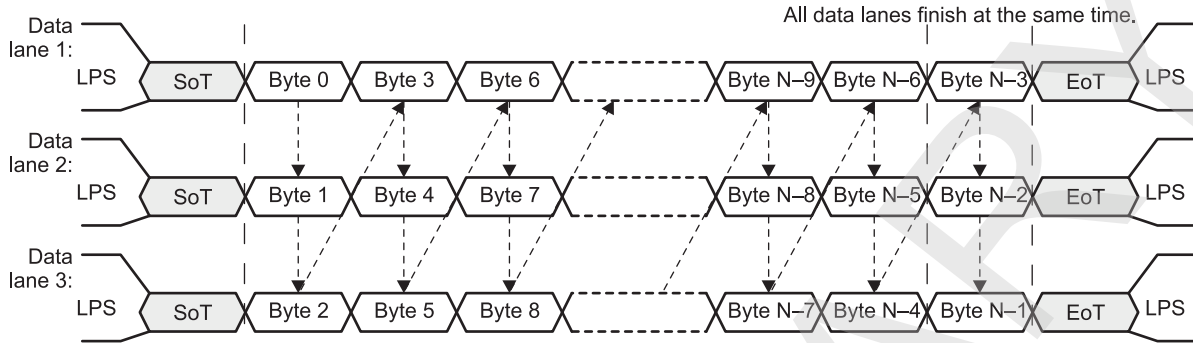


**Figure 8-33. ISS CSI2 Two Data-Lane Merger Configuration**

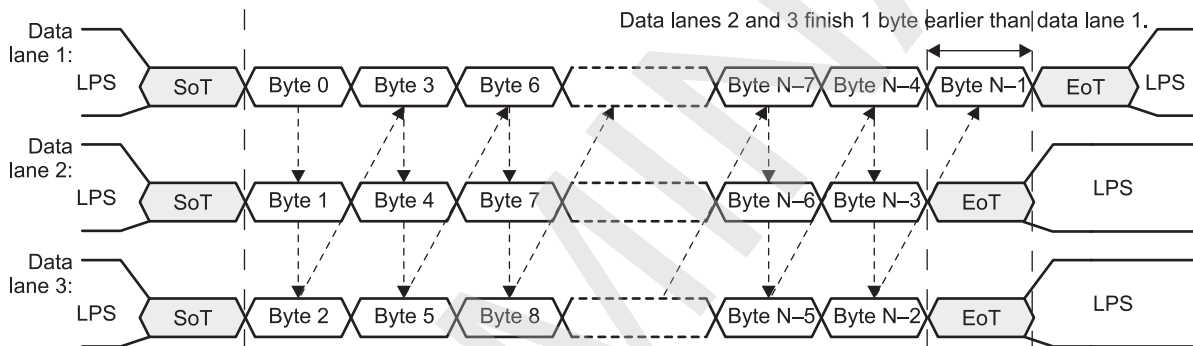


**Figure 8-34. ISS CSI2 Three Data-Lane Merger Configuration**

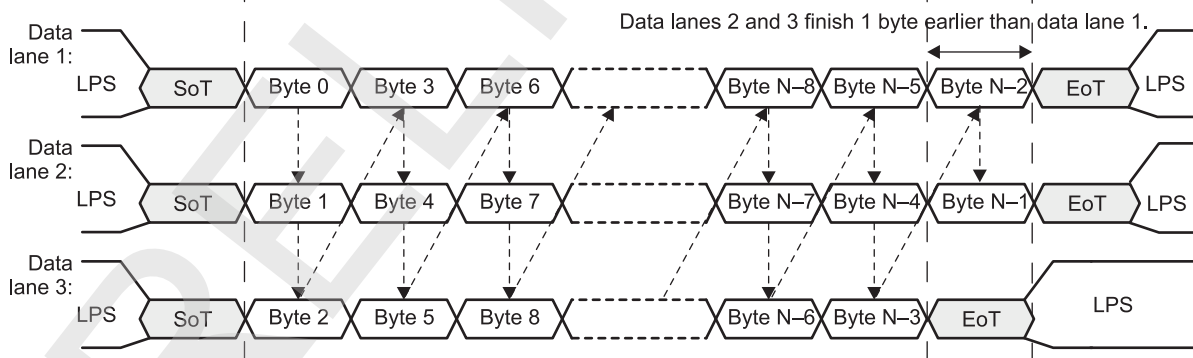
Number of bytes, N, transmitted is an integer multiple of the number of data lanes:



Number of bytes, N, transmitted is not an integer multiple of the number of data lanes (Example 1):



Number of bytes, N, transmitted is not an integer multiple of the number of data lanes (Example 2):

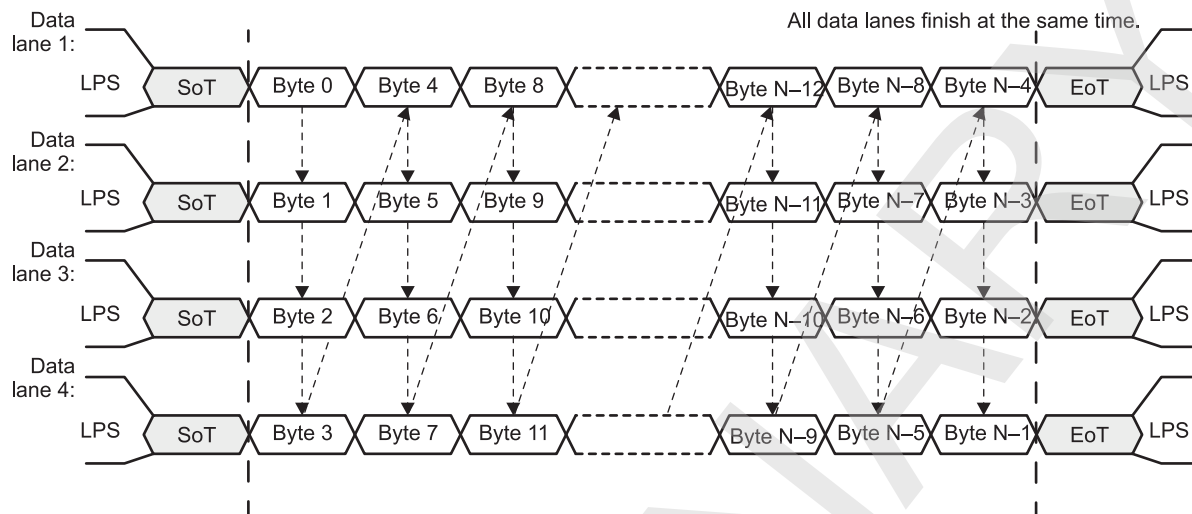


Key:  
 LPM: Low-power mode                      SoT: Start of transmission                      EoT: End of transmission

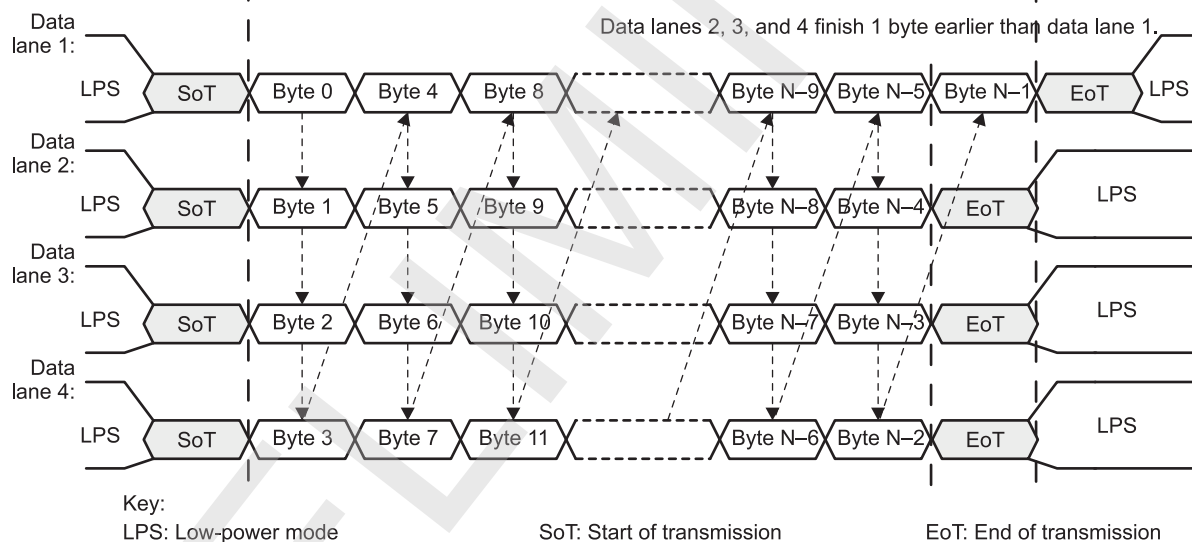
camss-257

**Figure 8-35. ISS CSI2 Four Data-Lane Merger Configuration**

Number of bytes, N, transmitted is not an integer multiple of the number of data lanes:



Number of bytes, N, transmitted is not an integer multiple of the number of data lanes:



camss-258

### 8.2.5.1.1.3 ISS CSI2 Protocol Layer

The low-level protocol (LLP) is a byte-oriented protocol from the lane merger layer. It supports short and long packet formats.

The CSI2 protocol layer defines how image-sensor data is transported onto the physical layer.

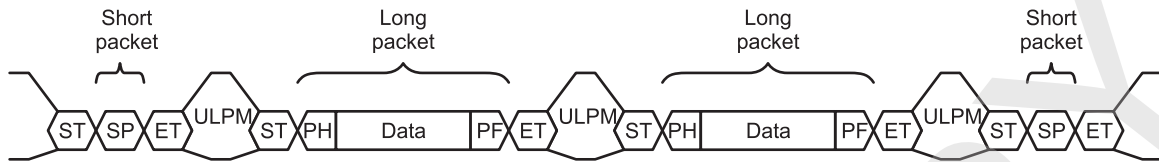
The feature set of the protocol layer implemented by the CSI2 receiver is:

- Transport of arbitrary data (payload-independent)
- 8-bit word size
- Support for up to four interleaved virtual channels on the same link
- Special packets for frame-start, frame-end, line-start, and line-end information
- Descriptor for the type, pixel depth, and format of application-specific payload data
- Error-correction code (ECC) for 1-bit error correction or 2-bit error detection in the header
- 16-bit checksum code for payload error detection

Figure 8-36 shows the CSI2 protocol layer with short and long packets.



**Figure 8-36. ISS CSI2 Protocol Layer With Short and Long Packets**



**Key:**  
 ST: Start of transmission                      ET: End of transmission  
 PH: Packet header                              PF: Packet footer  
 ULPM: Ultralow-power mode                SP: Short packet

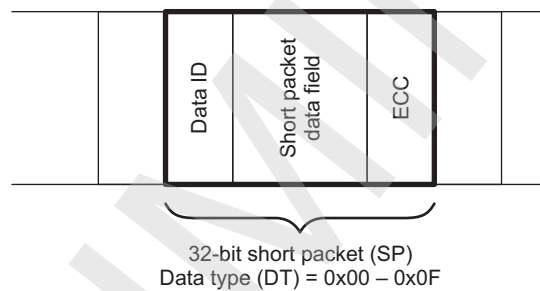
camss-241

Two packets are always separated from each other with a sequence of ET, ULPM, and ST.

### 8.2.5.1.1.3.1 ISS CSI2 Short Packet

A short packet is identified by data types 0x00 to 0x0F. A short packet can be used for frame or line synchronization or for generic data. Figure 8-37 shows the structure of a short packet.

**Figure 8-37. ISS CSI2 Short Packet Structure**



camss-242

For frame-synchronization data types, the short packet data field is the frame number. For line-synchronization data types, the short packet data field is the line number. For generic short packet data types, the content of the short packet data field is user-defined.

The 16-bit frame number, when used, is always nonzero to distinguish it from the use case where the frame number is inoperative and remains set to 0. The behavior of the 16-bit frame number is one of the following:

- The frame number is always 0 and is inoperative.
- The frame number increments by 1 for every FS packet within the same virtual channel and is periodically reset to 1 (1, 2, 1, 2, 1, 2, 1, 2 or 1, 2, 3, 4, 1, 2, 3, 4).

For LSC and LEC synchronization packets, the short packet data field contains a 16-bit line number. This line number is the same for the LS and LE packets corresponding to a given line. Line numbers are logical line numbers and do not necessarily equal physical line numbers. The 16-bit line number, when used, is always nonzero to distinguish it from the use case where the line number is inoperative and remains set to 0.

The behavior of the 16-bit line number is one of the following:

- The line number is always 0 and is inoperative.
- The line number increments by 1 for every LS packet within the same virtual channel and the same data type. The line number is periodically reset to 1 for the first LS packet after an FS packet. The intended use is for progressive scan (noninterlaced) video data streams. The line number must be a nonzero value.
- The line number increments by the same arbitrary step value greater than 1 for every LS packet within the same virtual channel and the same data type. The line number is periodically reset to a nonzero arbitrary start value for the first LS packet after an FS packet. The arbitrary start value can be different

between successive frames. The intended use is for interlaced video data streams.

The ECC byte allows single-bit errors to be corrected and 2-bit errors to be detected in the short packet.

Short packets apply to all contexts using the same virtual channel ID (up to eight contexts support eight dedicated configurations of virtual channel ID and data types). The data type associated with the context is not used to distinguish which context is used when receiving short packets.

#### 8.2.5.1.1.3.2 ISS CSI2 Long Packet

A long packet is identified by data types 0x10 to 0x37. A long packet consists of three elements:

- A 32-bit packet header (PH)
- An application-specific data payload with a variable number of 8-bit data words
- A 16-bit packet footer (PF)

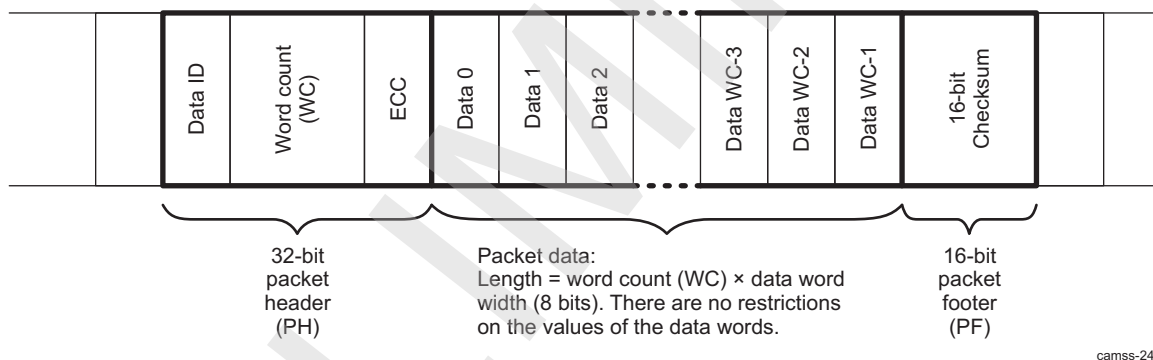
The packet header is composed of three elements:

- An 8-bit data identifier
- A 16-bit word count field
- An 8-bit ECC

The packet footer has one element, a 16-bit checksum.

Figure 8-38 and Table 8-139 show the structure of a long packet.

**Figure 8-38. ISS CSI2 Long Packet Structure**



**Table 8-139. ISS CSI2 Long Packet Structure Description**

Packet Part	Field Name	Size (Bits)	Description
Header	Data ID	8	Contains the virtual channel identifier and the data-type information
	Word count	16	Number of data words in the packet data. A word is 8 bits.
	ECC	8	ECC for data ID and WC field. Allows 1-bit error recovery and 2-bit error detection.
Data	Data	WC-8	Application-specific payload (WC words of 8 bits)
Footer	Checksum	16	16-bit CRC for packet data

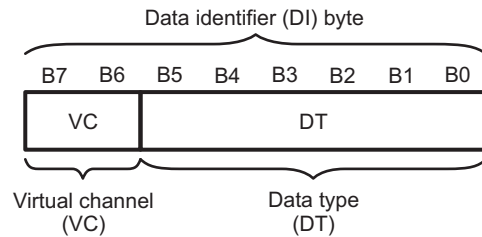
There are no restrictions on the size of the packet data, but each data format can impose additional restrictions on the length of the payload data (for example, a multiple of 4 bytes).

#### 8.2.5.1.1.3.3 ISS CSI2 Data Identifier

The data identifier byte contains the virtual channel (VC) value and the data-type (DT) value, as shown in Figure 8-39. The VC value is in the 2 MSBs of the data identifier byte. The DT value is in the 6 LSBs of the data identifier byte.

Figure 8-39 shows the data identifier structure.

**Figure 8-39. ISS CSI2 Data Identifier Structure**

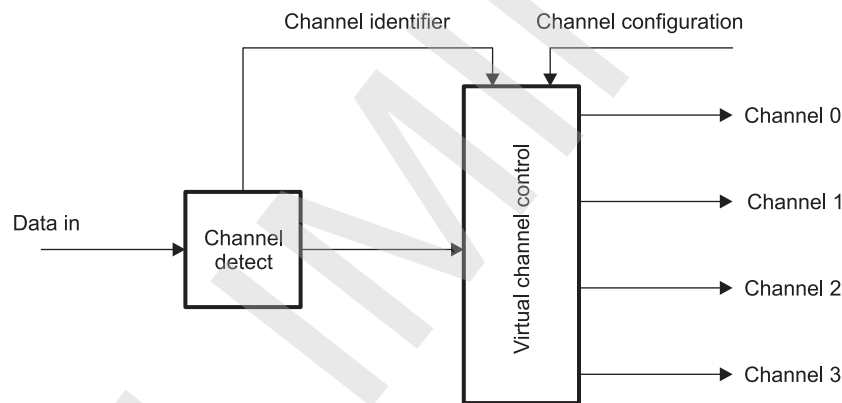


camss-244

### Virtual Channel

The CSI2 protocol layer transports virtual channels. Virtual channels are built of frames. A frame can comprise embedded data and image-sensor data. Two contexts are used to send the two types of data separately. Each frame is identified by unique mandatory synchronization codes: frame start and frame end. Line start and line end synchronization codes are optional for the transmitter. A set of registers is associated with each context defined by the virtual channel ID and the data type. Figure 8-40 shows a virtual channel.

**Figure 8-40. ISS CSI2 Virtual Channel**



camss-245

### Pixel Formats

#### 8.2.5.1.1.3.4 ISS CSI2 Synchronization Codes

Data reception from the image-sensor module uses four synchronization codes embedded in the serial bitstream:

- FSC: Identifies the start of a new frame
- LSC: Identifies the start of a new line; received for every line
- LEC: Identifies the end of a line; received for every line
- FEC: Identifies the end of the last line and the end of the current frame

Table 8-140 summarizes the synchronization code values.

**Table 8-140. ISS CSI2 Synchronization Codes**

Synchronization Code	Value	Comments
FSC	0x0	Mandatory
FEC	0x1	Mandatory
LSC	0x2	Optional
LEC	0x3	Optional
Reserved	0x4 to 0x7	Not used

#### 8.2.5.1.1.3.5 ISS CSI2 Generic Short Packet Codes

When the synchronization code value is from 0x8 to 0xF, the short packet is called a generic short packet. Short packets are not processed by the camera interface hardware. A generic short packet is stored in a register without the ECC and an interrupt can be generated. Therefore, generic short packets must be handled by software.

#### 8.2.5.1.1.3.6 ISS CSI2 Generic Long Packet Codes

The code value 0x10 indicates null packets, which can be received at any time. They are discarded by the protocol engine.

The code value 0x11 indicates blanking packets, which can be received at any time. They are discarded by the protocol engine.

The code value 0x12 indicates embedded 8-bit nonimage data typically used for JPEG.

Code values from 0x13 to 0x17 are reserved.

#### 8.2.5.1.1.3.7 ISS CSI2 Frame Structure

Each frame consists of short packets to indicate SOF and EOF. Optional short packets for start of line and end of line can be sent by the image sensor.

Some information before and after the picture data can be sent as SOF and EOF information by the image sensor to the memory through the L3 port.

For each frame, the pixel data (arbitrary data or user-defined byte data) are valid only after an SOF short packet. If the data are invalid, they are discarded by the protocol engine.

A frame contains embedded data and image-sensor data. [Figure 8-41](#) shows where the embedded data and image-sensor data are in the frame. The frame is scanned in raster order starting from the top-left corner, as shown in [Figure 8-41](#) and [Figure 8-42](#). The following definitions for a frame apply:

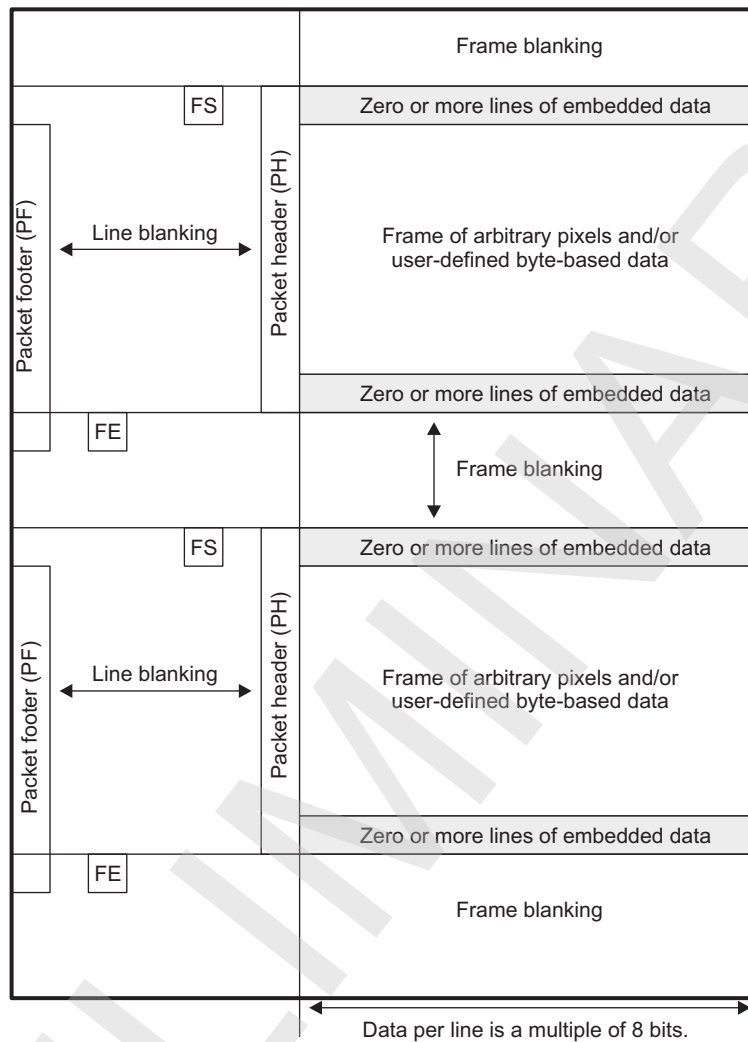
- Zero or more SOF status lines (SOF lines) can be embedded at the beginning of a CSI2 frame.
- The image embedded data is carried using separate data types and virtual channels (see [Section 8.2.4.1.1, ISS CCP2 Protocol and Data Formats](#), and [Section 8.2.5.3.3.4, ISS CSI2 Virtual Channel and Context](#)).
- Zero or more EOF status lines (EOF lines) can be embedded at the end of a CSI2 frame.
- The SOF lines, pixel data, and EOF lines do not overlap.

The CSI2 receiver does not use the information in the status lines. However, it extracts it and stores it in memory for use by software.

Because the data types are different, the data is carried using separate data types called virtual channels. Those must be mapped to the adequate context. The CSI2 receiver uses a different context for embedded data and image-sensor data. See [Section 8.2.5.3.3.4, ISS CSI2 Virtual Channel and Context](#).

Embedded data is supported as a context by the CSI2 receiver; therefore, there is no specific hardware support for embedded data.

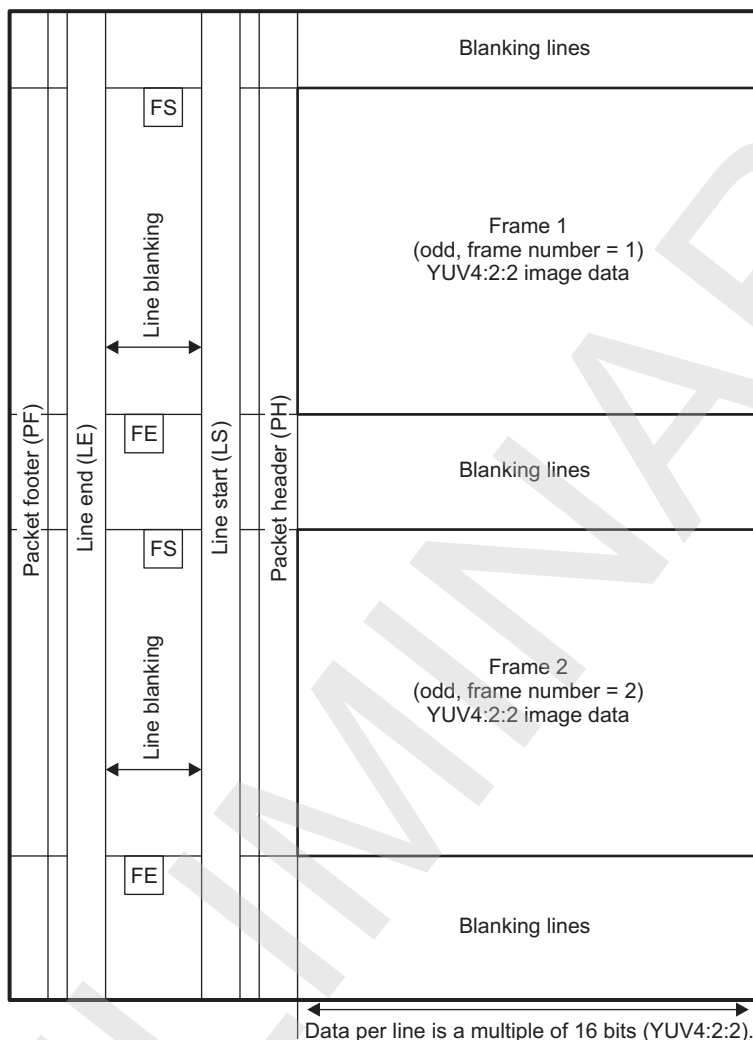
Figure 8-41. ISS CSI2 General Frame Structure (Informative)



**Key:**  
 PH: Packet header  
 FS: Frame start  
 PF: Packet footer  
 FE: Frame end

camss-246

Figure 8-42 shows the frame structure of a YUV4:2:2 interlaced video frame without embedded data.

**Figure 8-42. ISS CSI2 Digital Interlaced Video Frame (Informative)****Key:**

PH: Packet header  
 FS: Frame start  
 LS: Line start

PF: Packet footer  
 FE: Frame end  
 LE: Line end

camss-247

The period between the LEC and the new LSC is the line blanking period. The time between the FEC and the new FSC is the frame blanking period. The receiver works with the line blanking period set to 0. The image data is stored in memory by selecting one of the various operating modes. [Section 8.2.5.1.1.4](#) explains storing image data frames into memory.

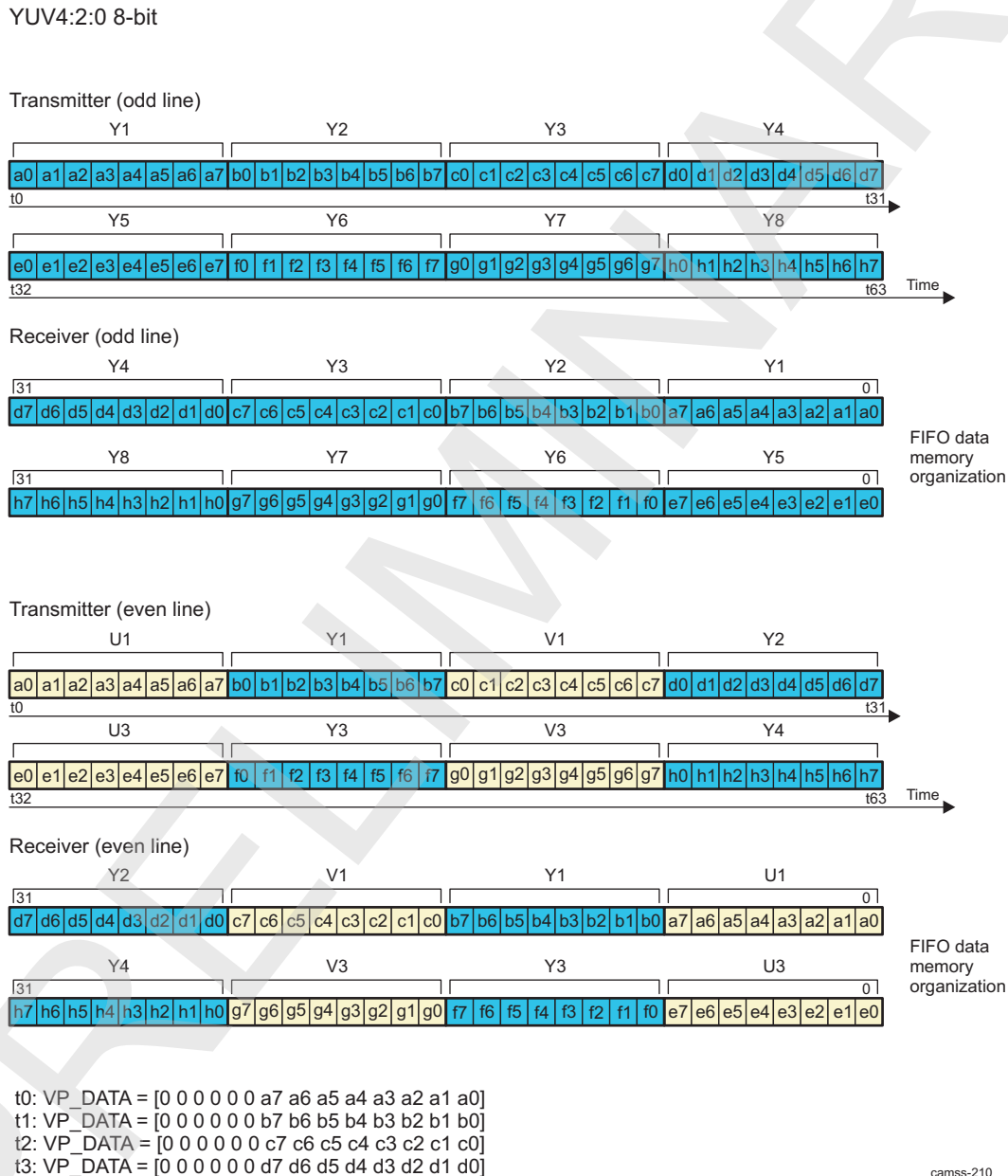
**8.2.5.1.1.4 ISS CSI2 Operating Modes****8.2.5.1.1.4.1 ISS CSI2 YUV Operating Modes****8.2.5.1.1.4.1.1 ISS CSI2 YUV4:2:0 8-Bit**

YUV4:2:0 8-bit data can be stored to memory in little-endian and big-endian format. To set the endianness for YUV4:2:0 use the [CSI2\\_CTRL\[4\]](#) ENDIANNES bit. The line length sent through the CSI2 physical protocol is a multiple of 16 bits for odd lines and 32 bits for even lines.

For correct pixel reconstruction, the line length must be a multiple of 32 bits and the number of lines must

be even. Figure 8-43 shows the storage format for YUV4:2:0 8-bit data. It is shown as little endian. If the data format is big endian, the figure changes accordingly. Set the `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field to 0x18 to select YUV4:2:0 8-bit mode. Even and odd lines do not have the same length. Offset must be set accordingly with the `CSI2_CTX_DAT_OFST_i[16:5]` OFST bit field; for example, if the offset is 0, the data is written in a contiguous way (bit-to-bit of odd and even lines). If the data has an offset, set the destination offset between the first pixel of the previous line and the first pixel of the current line being written to memory.

Figure 8-43. ISS CSI2 YUV4:2:0 8-Bit



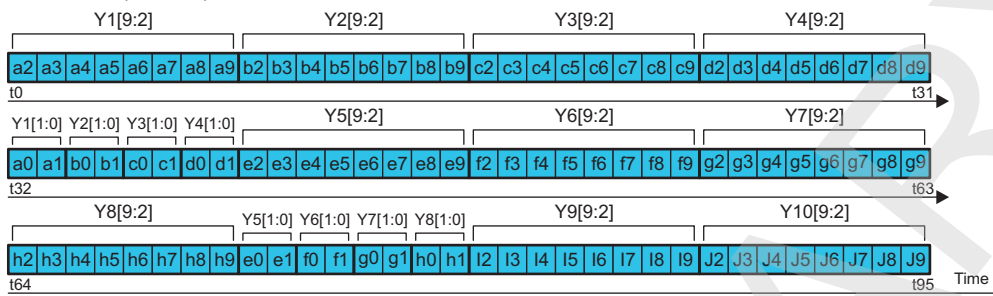
8.2.5.1.1.4.1.2 ISS CSI2 YUV4:2:0 10-Bit

YUV4:2:0 10-bit data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 40 bits for odd lines and 80 bits for even lines. Figure 8-44 shows the storage format for YUV4:2:0 10-bit data. Set the `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field to 0x19 to select YUV4:2:0 10-bit mode. Even and odd lines do not have the same length. Offset must be set accordingly with the `CSI2_CTX_DAT_OFST_i[16:5]` OFST bit field.

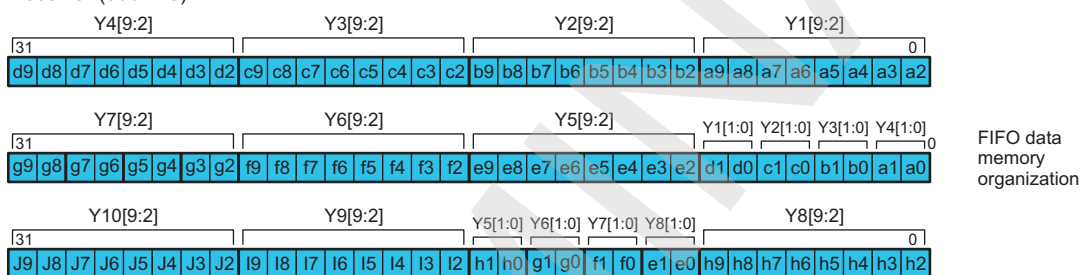
**Figure 8-44. ISS CSI2 YUV4:2:0 10-Bit**

YUV4:2:0 10-bit

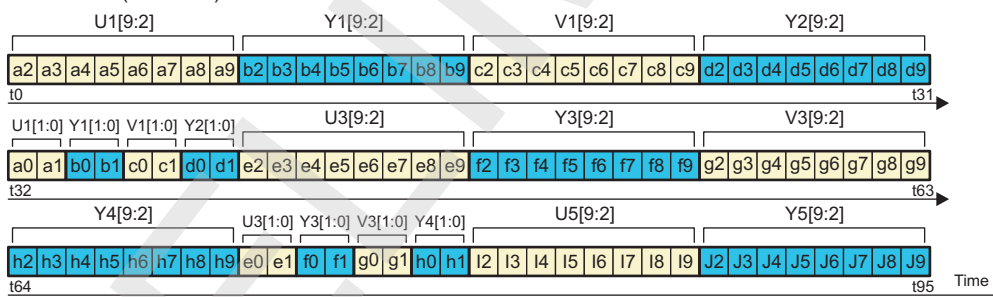
Transmitter (odd line)



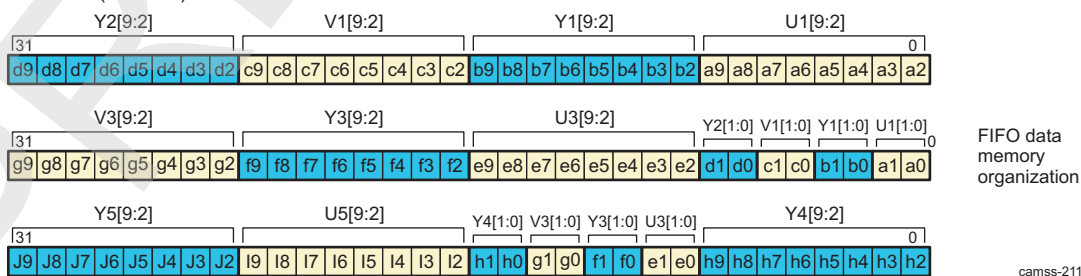
Receiver (odd line)



Transmitter (even line)



Receiver (odd line)



camss-211

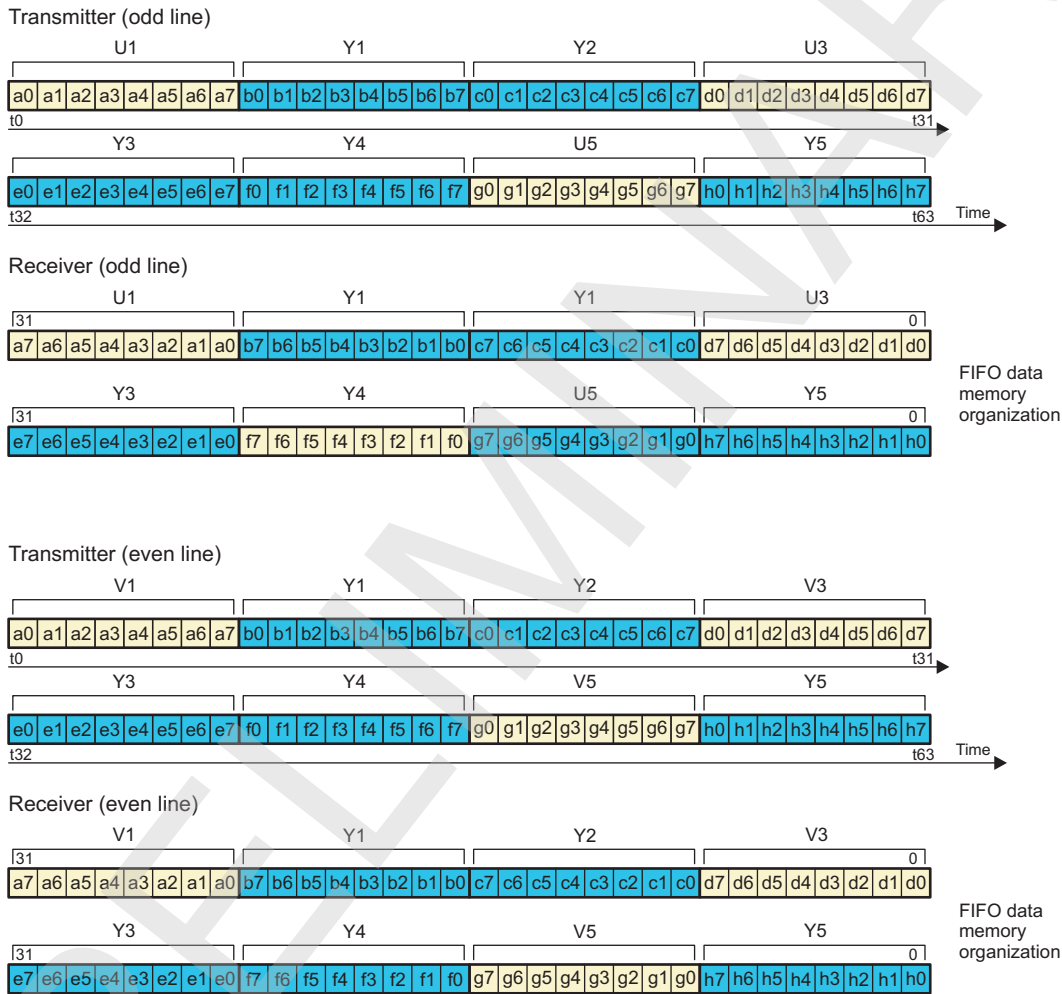


8.2.5.1.1.4.1.3 CS12 YUV4:2:0 8-Bit Legacy

YUV4:2:0 8-bit legacy data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 4 bytes. Figure 8-45 shows the storage format for YUV4:2:0 8-bit legacy data. Set the CSI2\_CTX\_CTRL2\_i[9:0] FORMAT bit field to 0x1A to select YUV4:2:0 8-bit legacy mode.

Figure 8-45. ISS CSI2 YUV4:2:0 8-Bit Legacy

YUV4:2:0 8-bit legacy



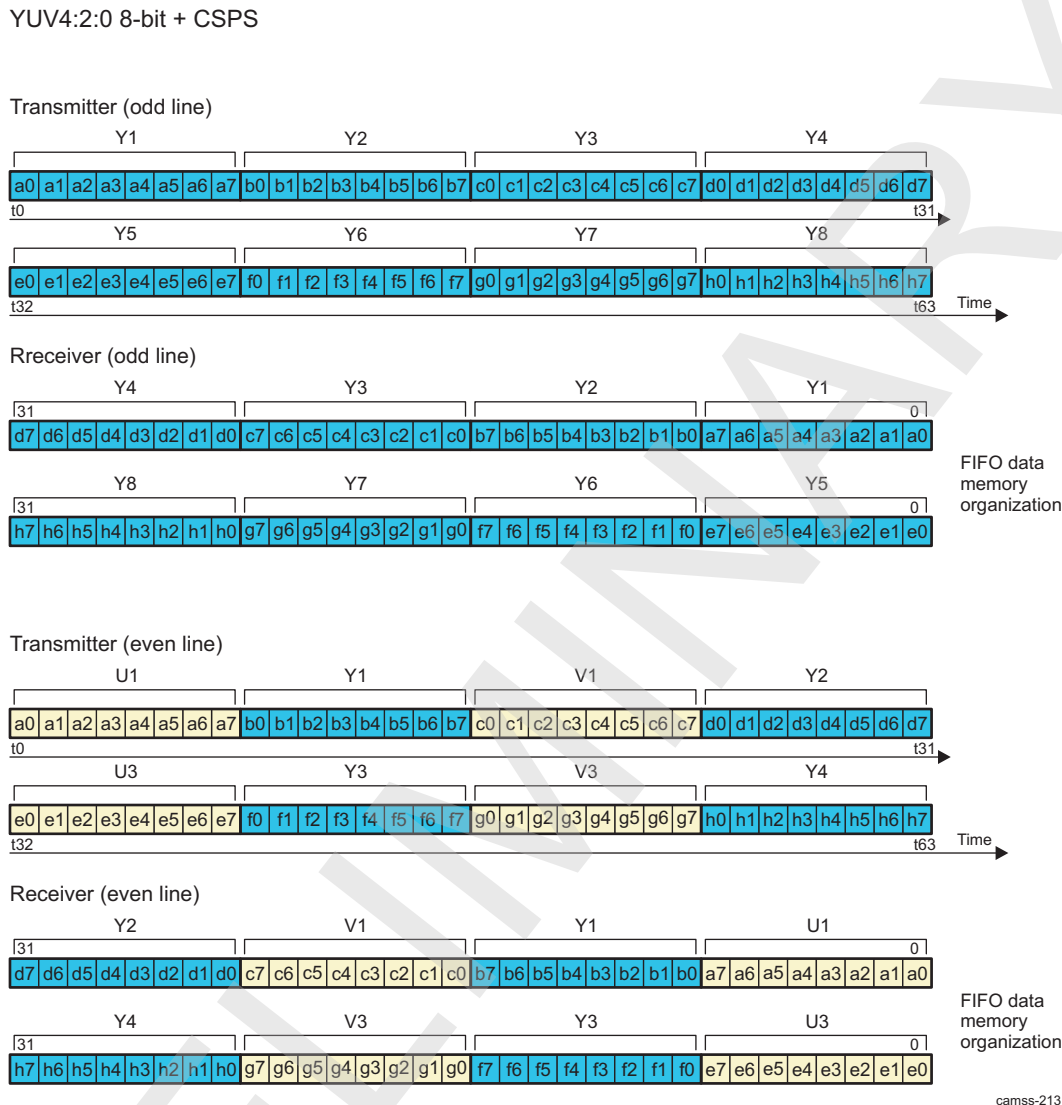
camss-212

8.2.5.1.1.4.1.4 ISS CSI2 YUV4:2:0 8-Bit + CSPS

YUV4:2:0 8-bit CSPS data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 16 bits for odd lines and 32 bits for even lines.

For correct pixel reconstruction, the line length must be a multiple of 3 32 bits and the number of lines must be even. Figure 8-46 shows the storage format for YUV4:2:0 8-bit + CSPS data. Set the CSI2\_CTX\_CTRL2\_i[9:0] FORMAT bit field to 0x1C to select YUV4:2:0 8-bit + CSPS mode.

**Figure 8-46. ISS CSI2 YUV4:2:0 8-Bit + CSPS**



**8.2.5.1.1.4.1.5 ISS CSI2 YUV4:2:0 10-Bit + CSPS**

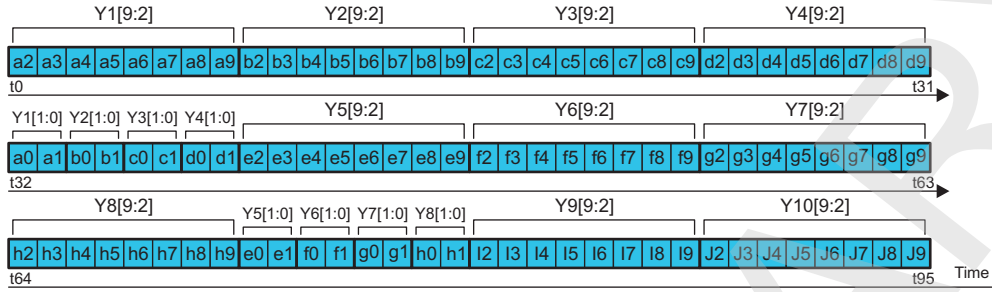
YUV4:2:0 10-bit CSPS data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 40 bits for odd lines and 80 bits for even lines.

For correct pixel reconstruction, the line length must be a multiple of 32 bits and the number of lines must be even. [Figure 8-47](#) shows the storage format for YUV4:2:0 10-bit + CSPS data. Set the `CSI2_CTX_CTRL2_Y[9:0]` FORMAT bit field to 0x1D to select YUV4:2:0 10-bit + CSPS mode.

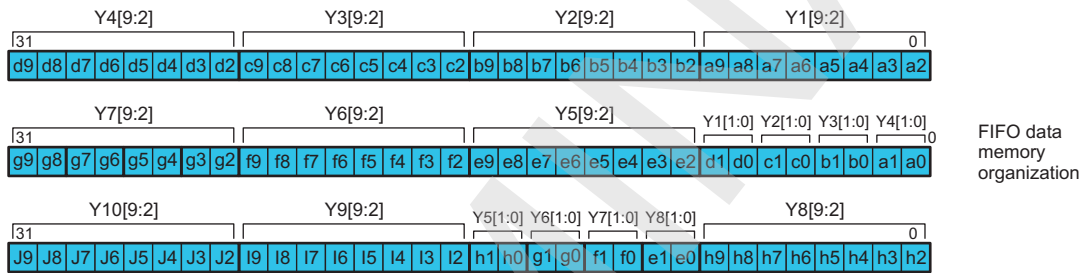
Figure 8-47. ISS CSI2 YUV4:2:0 10-Bit + CSPS

YUV4:2:0 10-bit + CSPS

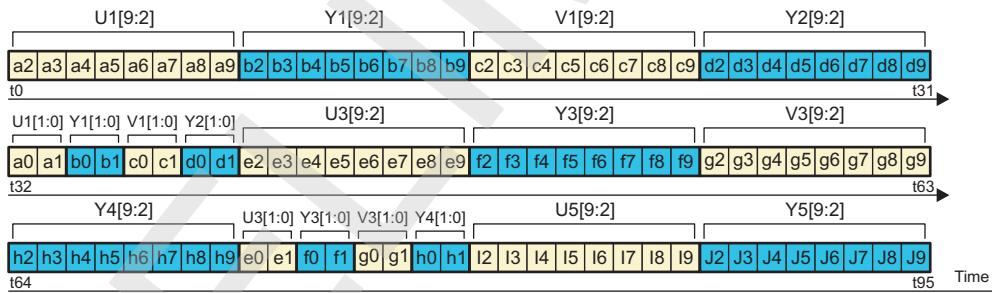
Transmitter (odd line)



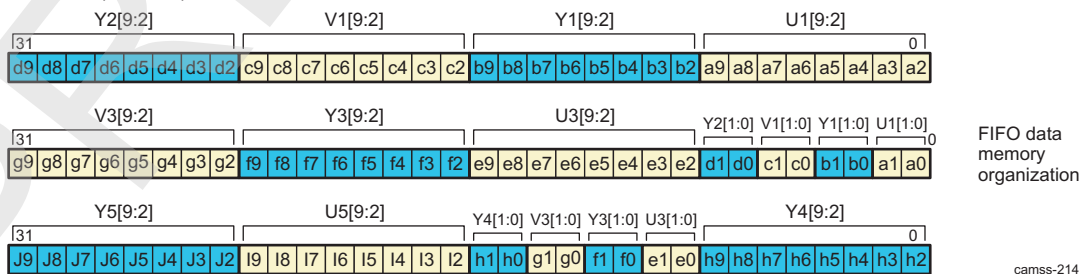
Receiver (odd line)



Transmitter (even line)



Receiver (even line)

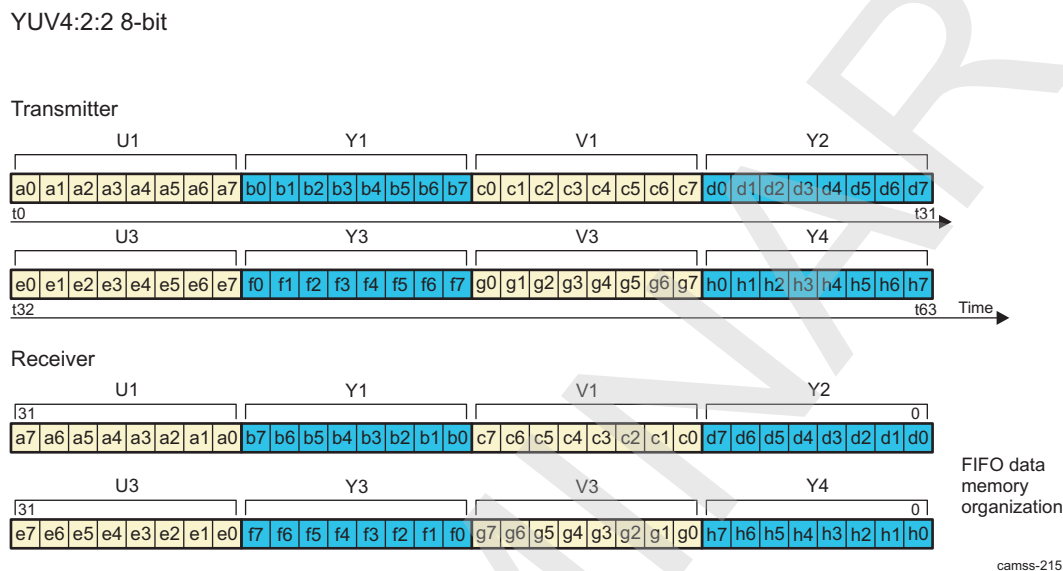


camss-214

### 8.2.5.1.1.4.1.6 ISS CSI2 YUV4:2:2 8-Bit

YUV4:2:2 data can be stored to memory in little-endian and big-endian format. To set the endianness for YUV4:2:2 use the `CSI2_CTRL[4]` ENDIANNESS bit. The line length sent through the CSI2 physical protocol is a multiple of 32 bits. [Figure 8-48](#) shows the storage format for YUV4:2:2 8-bit data. Set the `CSI2_CTX_CTRL2_j[9:0]` FORMAT bit field to 0x1E to select YUV4:2:2 8-bit mode.

**Figure 8-48. ISS CSI2 YUV4:2:2 8-Bit**

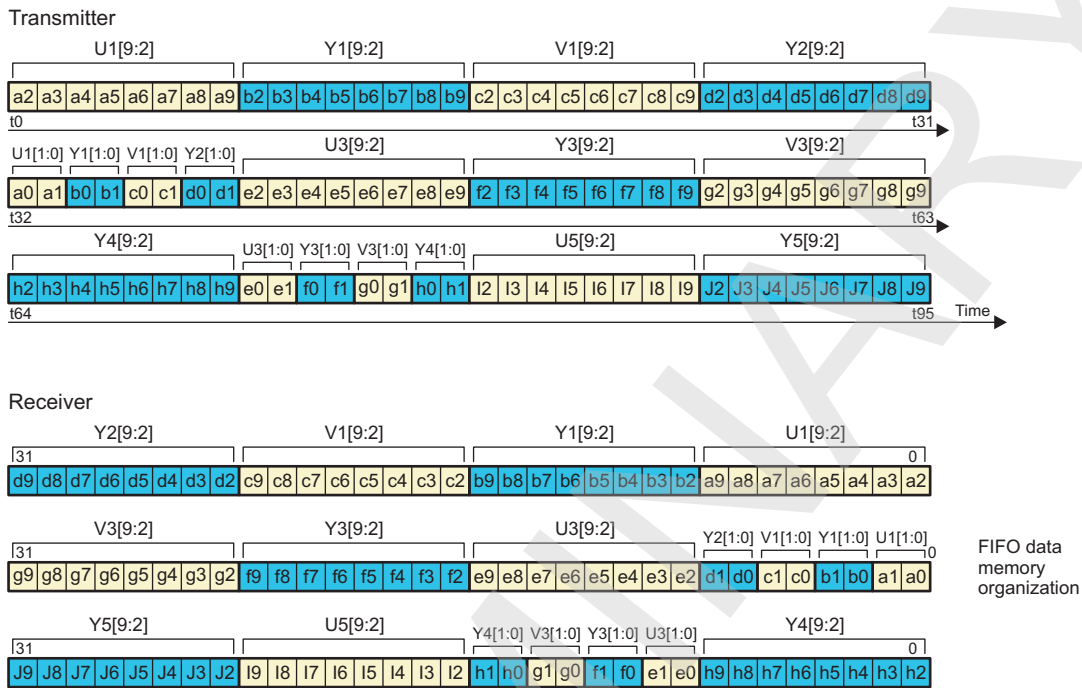


### 8.2.5.1.1.4.1.7 CSI2 YUV4:2:2 10-Bit

YUV4:2:2 data can be stored to memory in little-endian and big-endian format. The line length sent through the CSI2 physical protocol is a multiple of 40 bits. [Figure 8-49](#) shows the storage format for YUV4:2:2 10-bit data. Set the `CSI2_CTX_CTRL2_j[9:0]` FORMAT bit field to 0x1F to select YUV4:2:2 10-bit mode.

Figure 8-49. ISS CSI2 YUV4:2:2 10-Bit

YUV4:2:2 10-bit

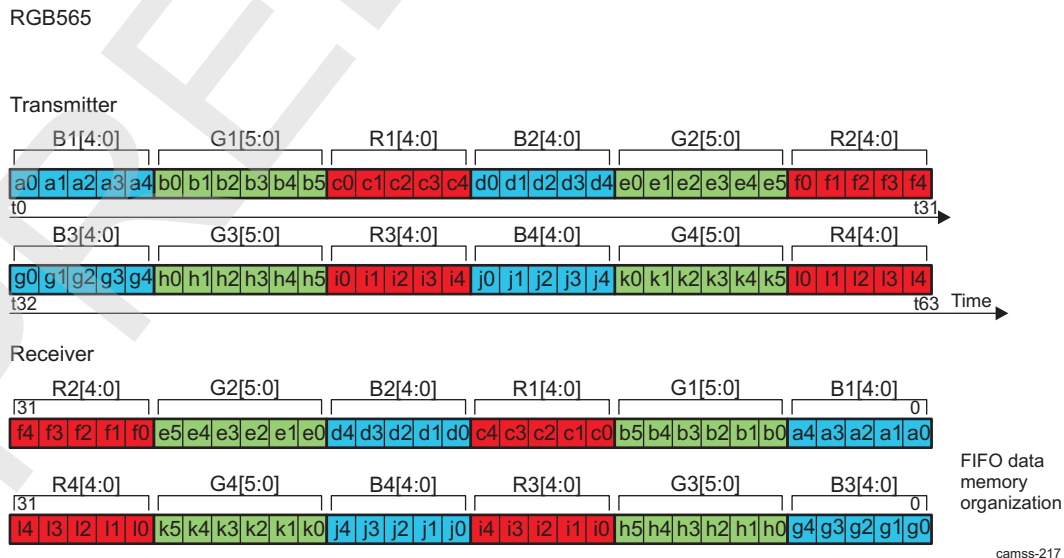


### 8.2.5.1.1.4.2 ISS CSI2 RGB Operating Modes

#### 8.2.5.1.1.4.2.1 ISS CSI2 RGB565

RGB565 data is output to memory without data expansion. The line length sent through the CSI2 physical layer is always a multiple of 16 bits. Figure 8-50 shows the storage format for RGB565 data. Set the CSI2\_CTX\_CTRL2\_i[9:0] FORMAT bit field to 0x22 to select RGB565 mode.

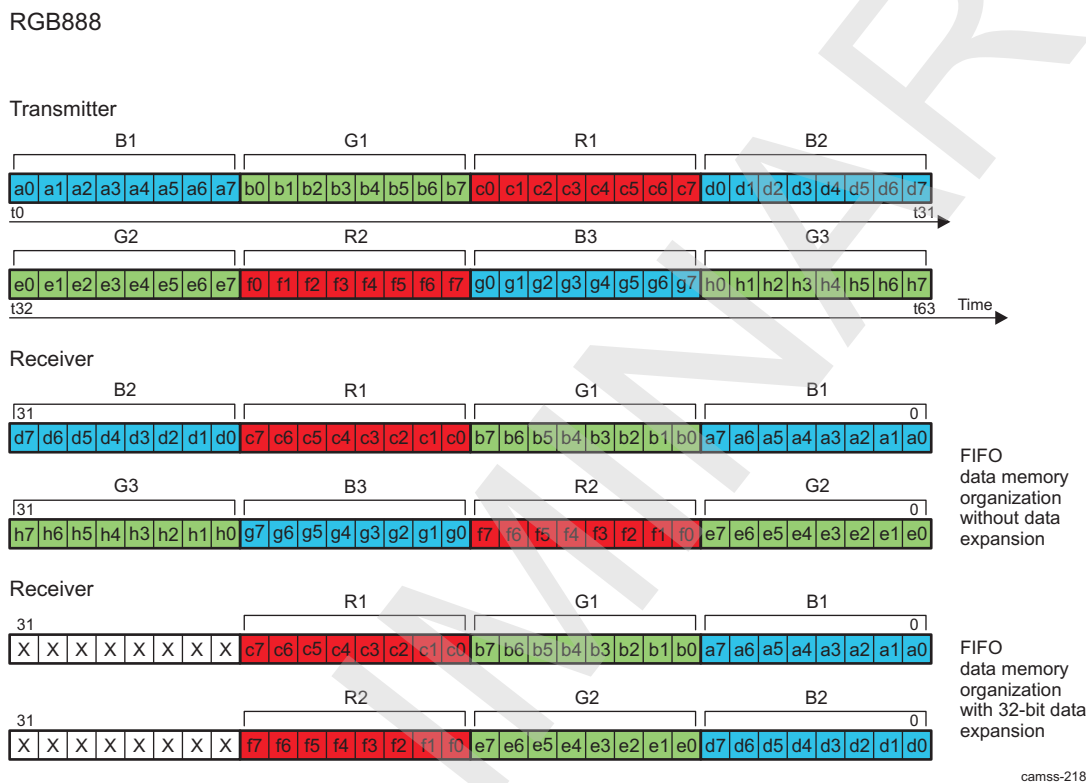
Figure 8-50. ISS CSI2 RGB565



### 8.2.5.1.1.4.2.2 ISS CSI2 RGB888

RGB888 data can be output to memory in two formats: with or without data expansion. If data expansion is used, the value of the 8 upper bits is programmable and can be set with an alpha value for computer graphics applications (the `CSI2_CTX_CTRL3_i[29:16]` ALPHA bit field). Figure 8-51 shows the storage format for RGB888 data. Set the `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field to 0x24 to select RGB888 mode.

Figure 8-51. ISS CSI2 RGB888

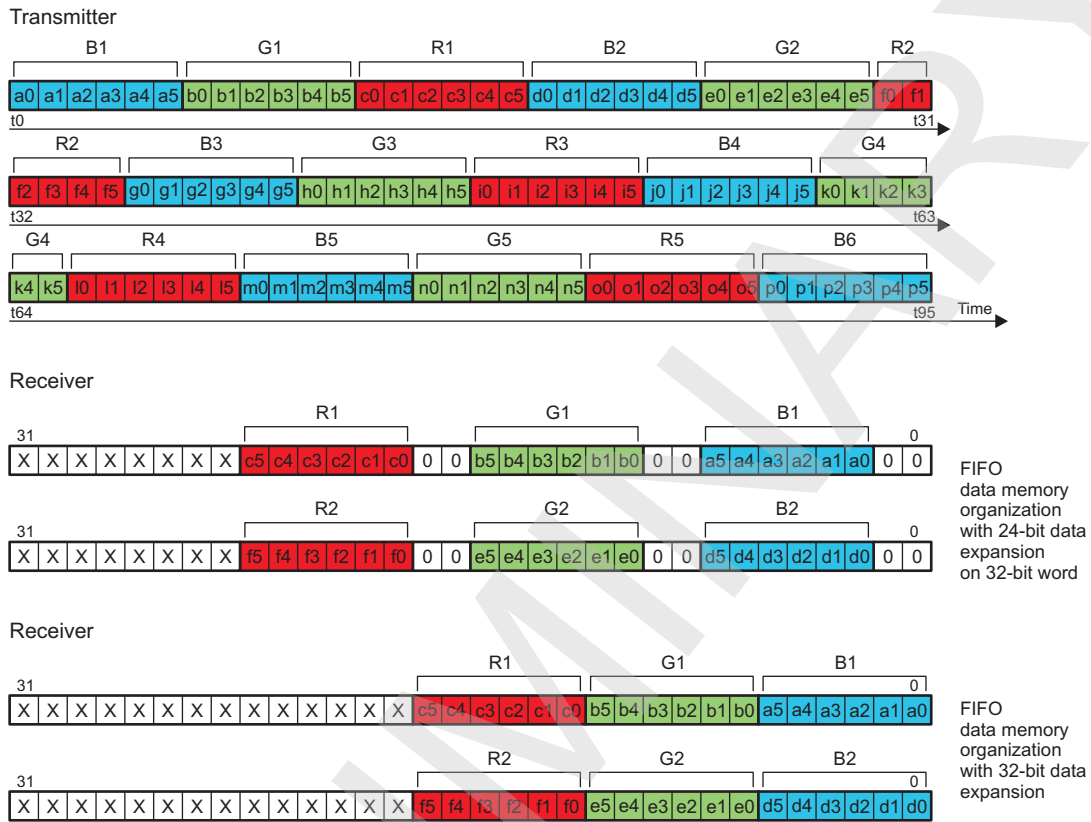


### 8.2.5.1.1.4.2.3 ISS CSI2 RGB666

RGB666 data is always output to memory with data expansion. The value of the 14 upper bits is programmable and can be set with an alpha value for computer graphics applications (the `CSI2_CTX_CTRL3_i[29:16]` ALPHA bit field). The line length sent through the CSI2 physical protocol is a multiple of 8 bits. Furthermore, the line length is a multiple of  $9 \times 8$  bits to finish the pixel reconstruction correctly. Figure 8-52 shows the storage format for RGB666 data. Set the `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field to 0x33 to select RGB666 mode.

Figure 8-52. ISS CSI2 RGB666

RGB666

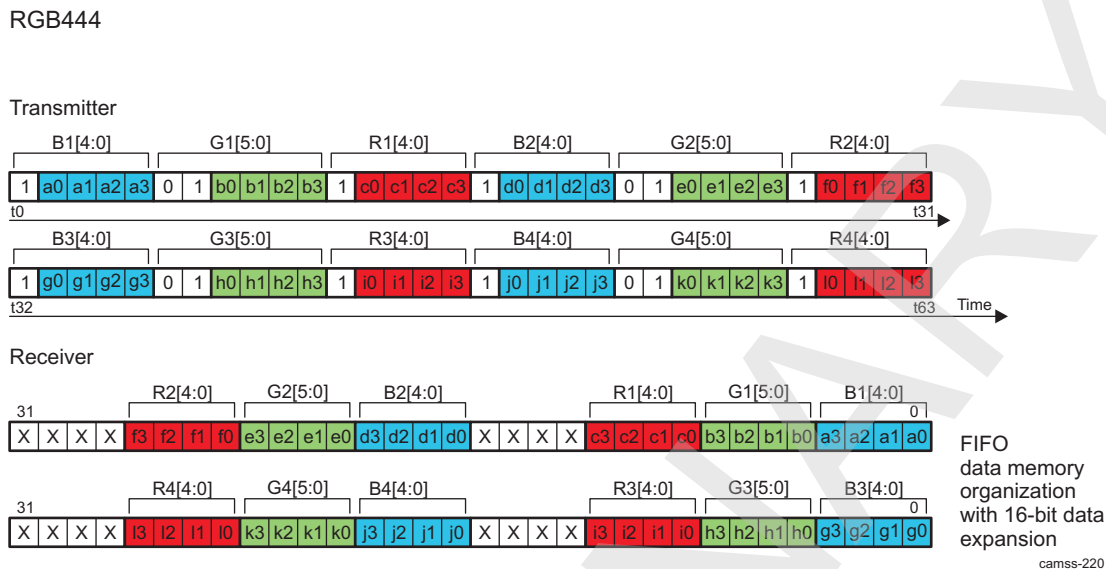


camss-219

8.2.5.1.1.4.2.4 ISS CSI2 RGB444

RGB444 data is output to memory with data expansion. When data expansion is used, the value of the 4 upper bits is programmable and can be set with an alpha value for computer graphics applications (the [CSI2\\_CTX\\_CTRL3\\_j\[29:16\]](#) ALPHA bit field). [Figure 8-53](#) shows the storage format for RGB444 data. Set the [CSI2\\_CTX\\_CTRL2\\_j\[9:0\]](#) FORMAT bit field to 0xA0 to select RGB444 mode.

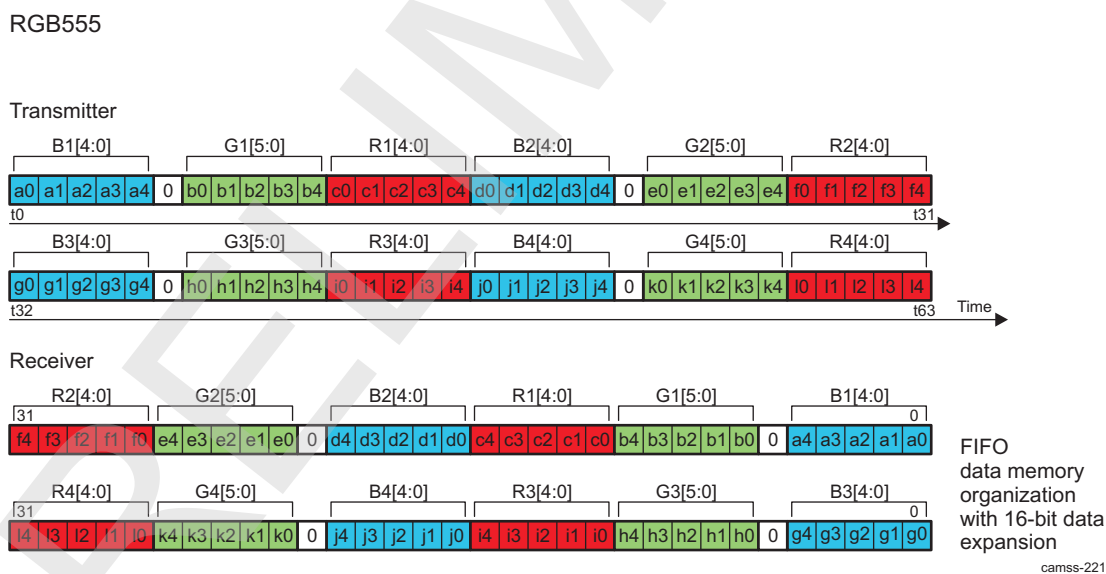
**Figure 8-53. ISS CSI2 RGB444**



**8.2.5.1.1.4.2.5 ISS CSI2 RGB555**

RGB555 data is output to memory with data expansion. Figure 8-54 shows the storage format for RGB555 data. Set the `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field to 0xA1 to select RGB555 mode.

**Figure 8-54. ISS CSI2 RGB555**



**8.2.5.1.1.4.3 ISS CSI2 RAW Bayer RGB Operating Modes**

**8.2.5.1.1.4.3.1 ISS CSI2 RAW6**

RAW6 data can be output to memory with or without data expansion. The line length sent through the CSI2 physical layer is a multiple of 8 bits (6-bit image data + 2-bit expansion). Furthermore, the line length is a multiple of 3 x 8 bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 6 is 24, so 3 x 8 bits). Figure 8-55 shows the storage format for RAW6 data. Set the `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field as follows:

- To 0x28 to select RAW6 mode

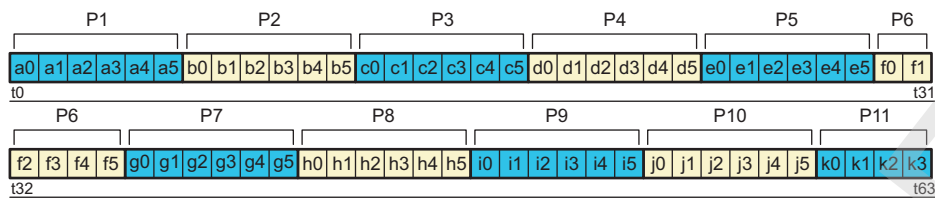


- To 0x68 for RAW6 + 8-bit expansion
- To 0xE8 for RAW6 + DPCM decompression to 10-bit to video port
- To 0x2A8 for RAW6 + DPCM decompression to 10-bit expanded to 16-bit
- To 0x3A8 for RAW6 + DPCM decompression to 12-bit expanded to 16-bit
- To 0x368 for RAW6 + DPCM decompression to 12-bit to video port

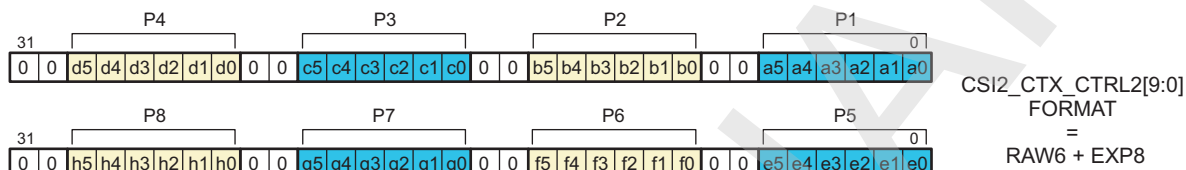
Figure 8-55. ISS CSI2 RAW6

RAW6

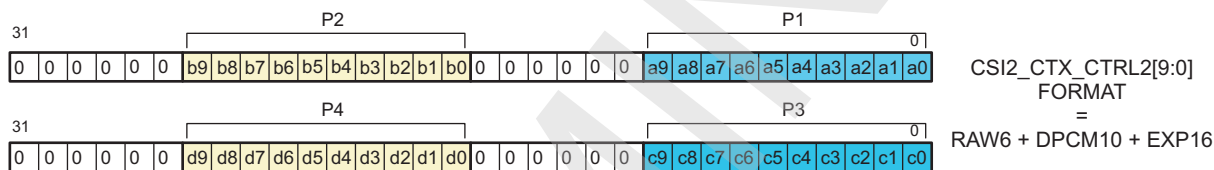
Transmitter



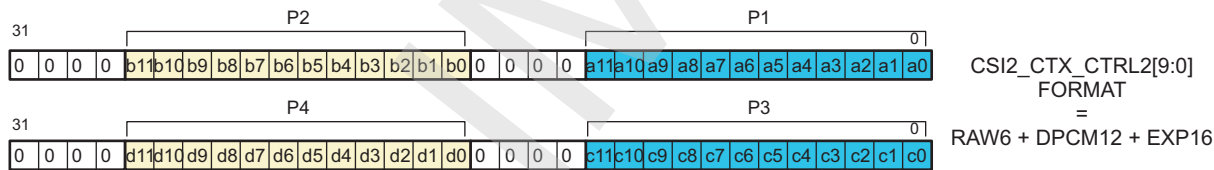
Receiver



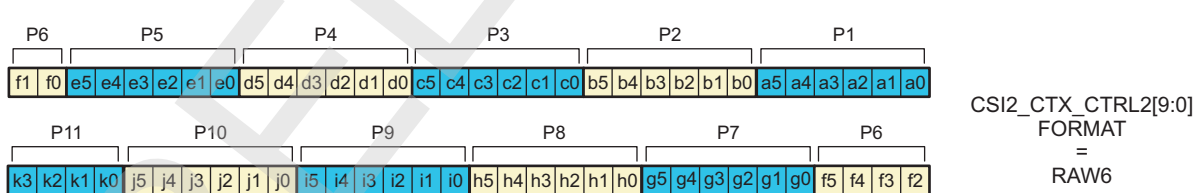
Receiver



Receiver



Receiver



t0: VP\_DATA = [0 0 0 0 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]  
 t1: VP\_DATA = [0 0 0 0 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]  
 t2: VP\_DATA = [0 0 0 0 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]  
 t3: VP\_DATA = [0 0 0 0 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

CSI2\_CTX\_CTRL2[9:0]  
 FORMAT  
 =  
 RAW6 + DPCM10 + VP

t0: VP\_DATA = [0 0 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]  
 t1: VP\_DATA = [0 0 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]  
 t2: VP\_DATA = [0 0 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]  
 t3: VP\_DATA = [0 0 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

CSI2\_CTX\_CTRL2[9:0]  
 FORMAT  
 =  
 RAW6 + DPCM12 + VP

camss-079

### 8.2.5.1.1.4.3.2 ISS CSI2 RAW7

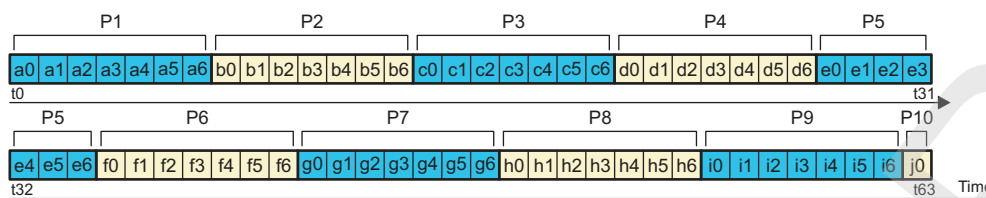
RAW7 data can be output to memory with or without data expansion. The line length sent through the CSI2 physical layer is a multiple of 8 bits. Furthermore, the line length is a multiple of  $7 \times 8$  bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 7 is 56, so  $7 \times 8$  bits). [Figure 8-56](#) shows the storage format for RAW7 data. Set the `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field as follows:

- To 0x29 to select RAW7 mode
- To 0x69 for RAW7 + 8-bit expansion
- To 0x329 for RAW7 + DPCM decompression to 10-bit to video port
- To 0x229 for RAW7 + DPCM decompression to 10-bit expanded to 16-bit
- To 0x369 for RAW7 + DPCM decompression to 12-bit expanded to 16-bit
- To 0x3A9 for RAW7 + DPCM decompression to 12-bit to video port

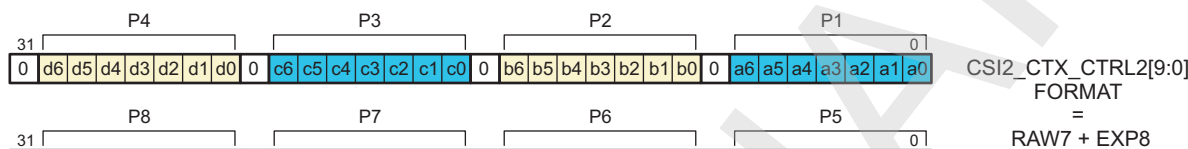
Figure 8-56. ISS CSI2 RAW7

RAW7

Transmitter



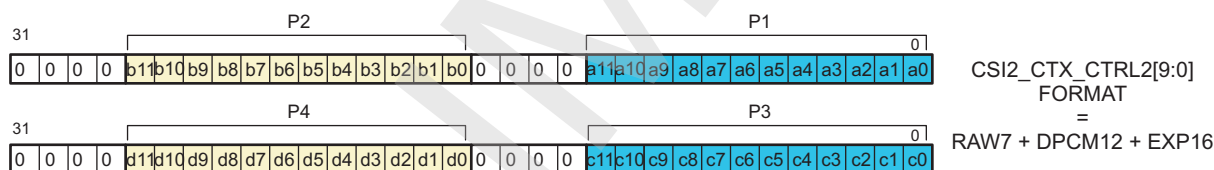
Receiver



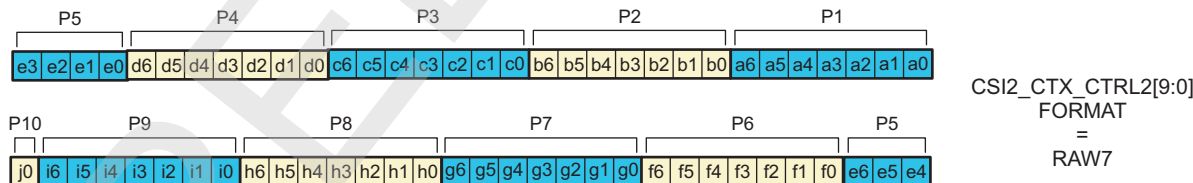
Receiver



Receiver



Receiver



t0: VP\_DATA = [0 0 0 0 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]  
 t1: VP\_DATA = [0 0 0 0 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]  
 t2: VP\_DATA = [0 0 0 0 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]  
 t3: VP\_DATA = [0 0 0 0 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

CSI2\_CTX\_CTRL2[9:0]  
 FORMAT  
 =  
 RAW7 + DPCM10 + VP

t0: VP\_DATA = [0 0 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]  
 t1: VP\_DATA = [0 0 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]  
 t2: VP\_DATA = [0 0 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]  
 t3: VP\_DATA = [0 0 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

CSI2\_CTX\_CTRL2[9:0]  
 FORMAT  
 =  
 RAW7 + DPCM12 + VP

camss-078

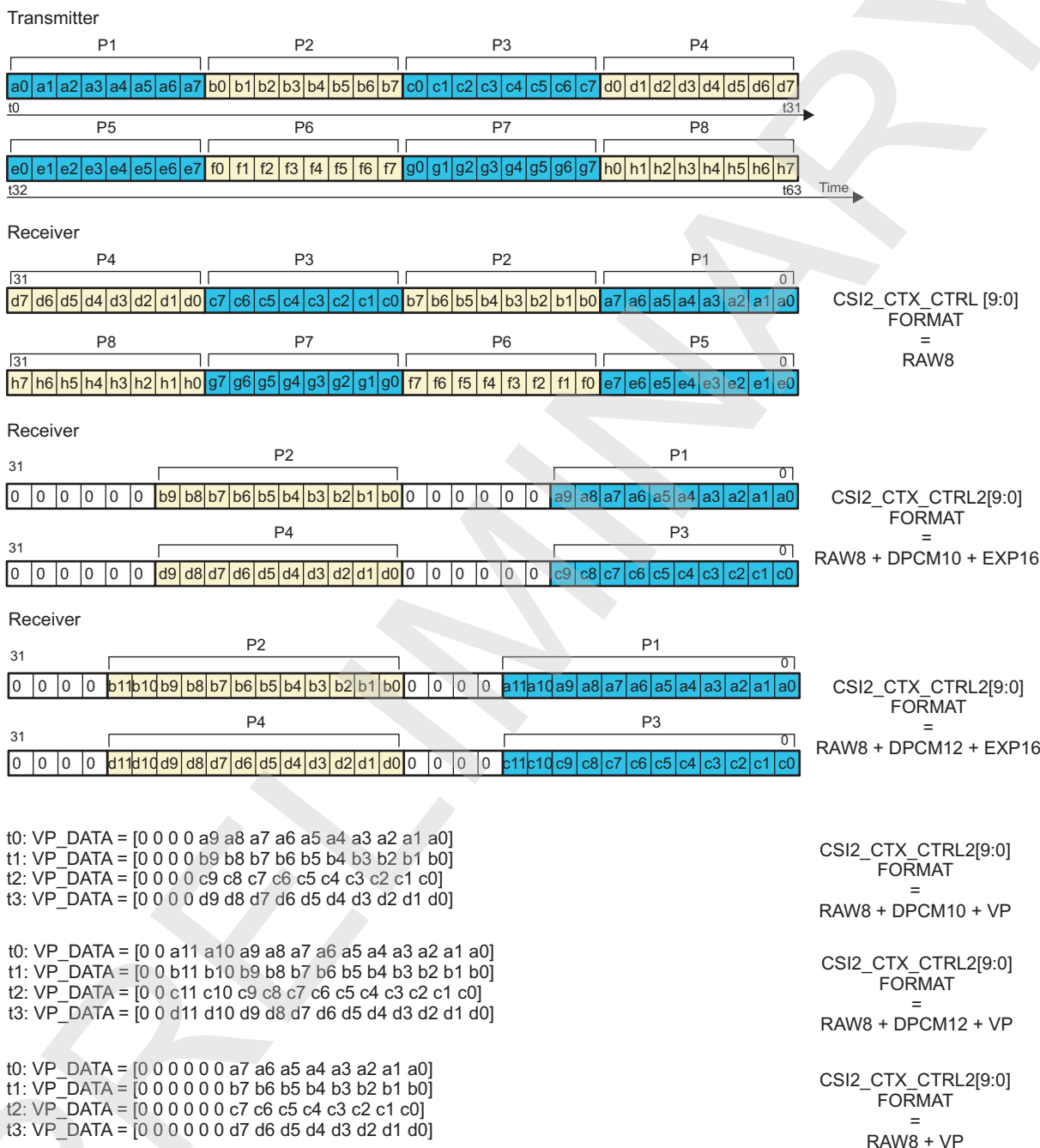
### 8.2.5.1.1.4.3.3 ISS CSI2 RAW8

RAW8 data can be output to memory with or without data expansion. The line length sent through the CSI2 physical layer is always a multiple of 8 bits. [Figure 8-57](#) shows the storage format for RAW8 data. Set the `CSI2_CTX_CTRL2_j[9:0]` FORMAT bit field as follows:

- To 0x2A to select RAW8 mode
- To 0x12A for RAW8 to video port
- To 0x32A for RAW8 + DPCM decompression to 10-bit to video port
- To 0x2AA for RAW8 + DPCM decompression to 10-bit expanded to 16-bit
- To 0x36A for RAW8 + DPCM decompression to 12-bit expanded to 16-bit
- To 0x3AA for RAW8 + DPCM decompression to 12-bit to video port

Figure 8-57. ISS CSI2 RAW8

RAW8



camss-081

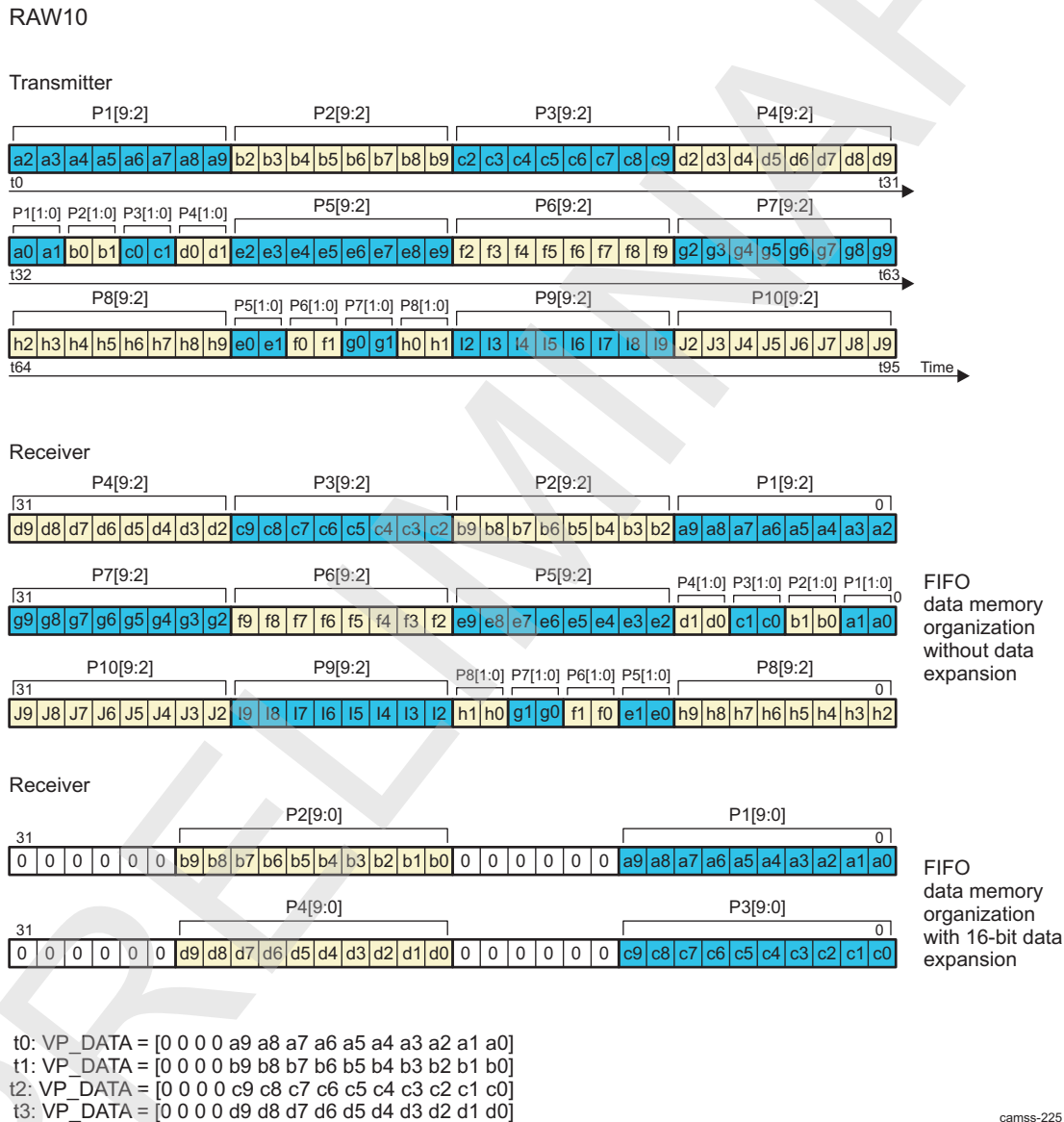
8.2.5.1.1.4.3.4 ISS CSI2 RAW10

RAW10 data can be output memory in two formats: with or without data expansion. It can also be sent to

the video port. If data expansion is used, the 10-bit data are padded with 0s on a 16-bit word. The line length sent through the CSI2 physical layer is a multiple of 8 bits. Furthermore, the line length is a multiple of  $5 \times 8$  bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 10 is 40, so  $5 \times 8$  bits). Figure 8-58 shows the storage format for RAW10 data. Set the `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field as follows:

- To 0x2B to select RAW10 mode
- To 0xAB for RAW10 + 16-bit expansion
- To 0x12F for RAW10 to video port

Figure 8-58. ISS CSI2 RAW10



8.2.5.1.1.4.3.5 ISS CSI2 RAW12

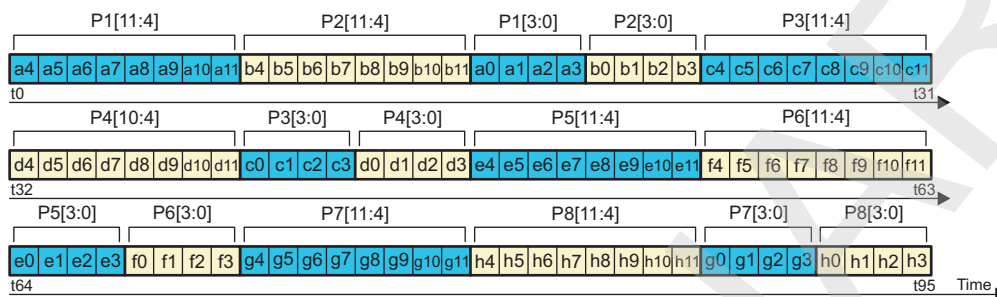
RAW12 data can be output to memory in two formats: with or without data expansion. It can also be sent to the video port. If data expansion is used, the 12-bit data are padded with 0s on a 16-bit word. The line length sent through the CSI2 physical layer is a multiple of 8 bits. Furthermore, the line length is a multiple of  $3 \times 8$  bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 12 is 24, so  $3 \times 8$  bits). Figure 8-59 shows the storage format for RAW12 data. Set the `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field as follows:

- To 0x2C to select RAW12 mode
- To 0xAC for RAW12 + 16-bit expansion
- To 0x12C for RAW12 to video port

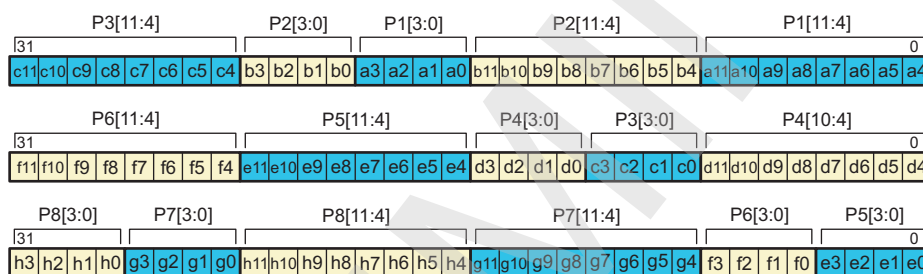
**Figure 8-59. ISS CSI2 RAW12**

RAW12

Transmitter

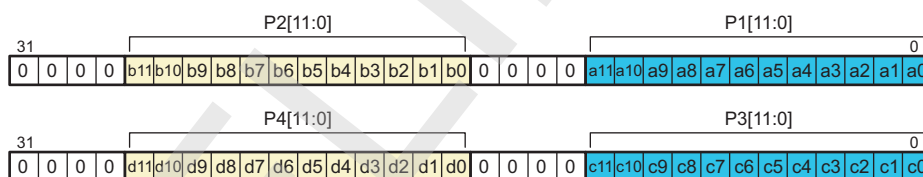


Receiver



FIFO data memory organization without data expansion

Receiver



FIFO data memory organization with 16-bit data expansion

t0: VP\_DATA = [0 0 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]  
 t1: VP\_DATA = [0 0 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]  
 t2: VP\_DATA = [0 0 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]  
 t3: VP\_DATA = [0 0 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

camss-226

**8.2.5.1.1.4.3.6 ISS CSI2 RAW14**

RAW14 data can be output to memory in two formats: with or without data expansion. It can also be sent to the video port. If data expansion is used, the 14-bit data are padded with 0s on a 16-bit word. The line length sent through the CSI2 physical layer is a multiple of 8 bits. Furthermore, the line length is a multiple of 7 x 8 bits to complete the pixel reconstruction correctly (the lowest common multiple of 8 and 14 is 56, so 7 x 8 bits). Figure 8-60 shows the storage format for RAW14 data. Set the CSI2\_CTX\_CTRL2\_i[9:0] FORMAT bit field as follows:

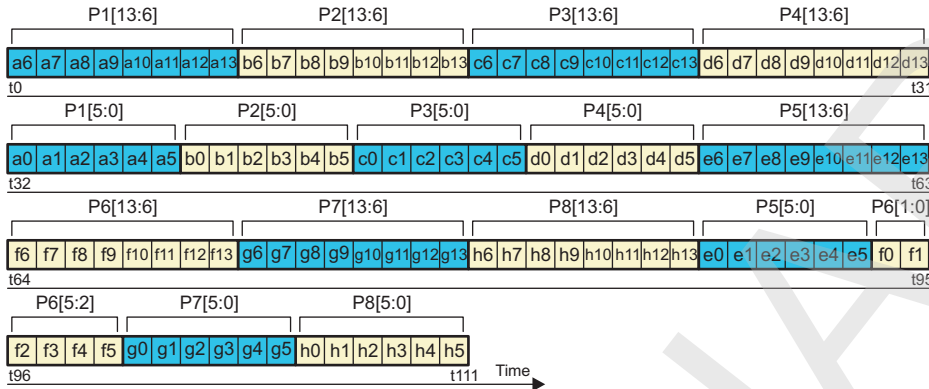
- To 0x2D to select RAW14 mode
- To 0xAD for RAW14 + 16-bit expansion
- To 0x12D for RAW12 to video port



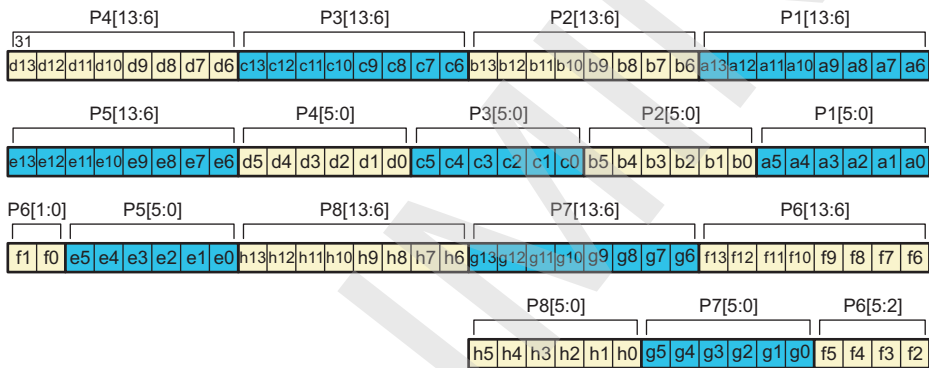
Figure 8-60. ISS CSI2 RAW14

RAW14

Transmitter

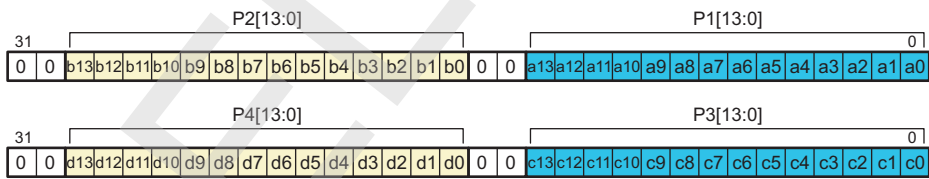


Receiver



FIFO data memory organization without data expansion

Receiver



FIFO data memory organization with 16-bit data expansion

- t0: VP\_DATA = [a13 a12 a11 a10 a9 a8 a7 a6 a5 a4 a3 a2 a1 a0]
- t1: VP\_DATA = [b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0]
- t2: VP\_DATA = [c13 c12 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0]
- t3: VP\_DATA = [d13 d12 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0]

camss-227

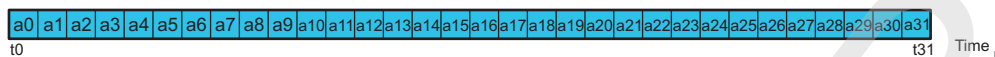
### 8.2.5.1.1.4.4 ISS CSI2 JPEG8 Operating Modes

The size of a compressed stream can be known in advance. Figure 8-61 shows the format for storing JPEG8 data.

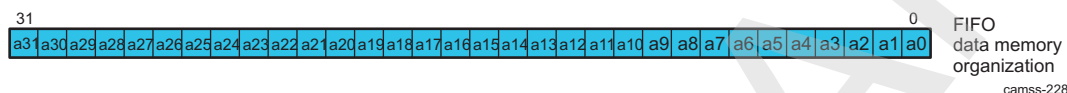
**Figure 8-61. ISS CSI2 JPEG8**

JPEG8 (Embedded 8-bit non image data)

Transmitter



Receiver

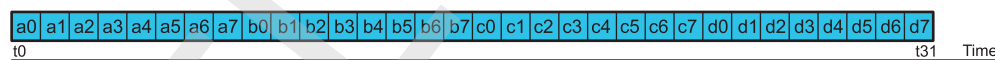
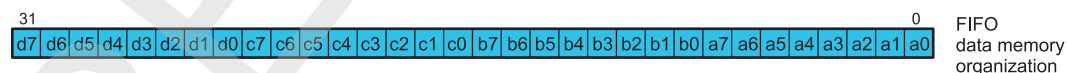
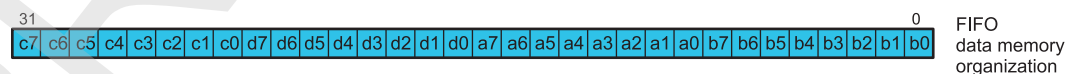
**8.2.5.1.1.4.5 ISS CSI2 Generic Format**

The CSI2 receiver supports a generic format to send data to memory and/or the video port. The generic mode is entered by setting the `CSI2_CTX_CTRL1_i[30]` GENERIC bit. The `CSI2_CTX_CTRL2_i[9:0]` FORMAT bit field defines how the data stream is decoded. However, the MIPI data type code is ignored. Only the virtual channel information is used to map a received data stream to a context. Software must ensure that a MIPI virtual channel used in generic mode is mapped only to a single context. Software can optionally enable byte swapping of the payload data by setting the `CSI2_CTX_CTRL1_i[31]` BYTESWAP bit. This feature must be used only when the amount of payload data per packet is a multiple of 16 bits. The byte swapping is performed before pixel reconstruction.

Figure 8-62 shows the ISS CSI2 generic format.

**Figure 8-62. ISS CSI2 Generic Format**ISS CSI2 Generic: `CSI2_CTX_CTRL1_i[30]` GENERIC = 0x1

Transmitter

Receiver when `CSI2_CTX_CTRL1[31]` BYTESWAP = 0x0Receiver when `CSI2_CTX_CTRL1[31]` BYTESWAP = 0x1

camss\_140-228b

**8.2.5.1.1.4.6 ISS CSI2 MIPI Format Supported Summary**

Table 8-141 summarizes the CSI2 MIPI-supported formats and their output category. By setting the `CSI2_CTX_CTRL2_i` register format, the CSI2 outputs certain types of pixel packet data.

**Table 8-141. ISS CSI2 MIPI Format Supported by the Protocol Engine**

Category	MIPI		CSI2 Protocol Engine Support
	Abbreviation	Register Setting Format Description	Configuration Value for <b>CSI2_CTX_CTRL2_i[9:0] FORMAT</b>
Sync short packet data types <sup>(1)</sup>	Short packet sync code	Mandatory FSC	0x000
	Short packet sync code	Mandatory FEC	0x001
	Short packet sync code	Optional LSC	0x002
	Short packet sync code	Optional LEC	0x003
			0x004
			0x005
			0x006
			0x007
Generic short packet data types <sup>(1)</sup>	Short packet	32-bit without ECC is stored in a register with code value 0x008.	0x008
	Short packet	32-bit without ECC is stored in a register with code value 0x009.	0x009
	Short packet	32-bit without ECC is stored in a register with code value 0x00A.	0x00A
	Short packet	32-bit without ECC is stored in a register with code value 0x00B.	0x00B
	Short packet	32-bit without ECC is stored in a register with code value 0x00C.	0x00C
	Short packet	32-bit without ECC is stored in a register with code value 0x00D.	0x00D
	Short packet	32-bit without ECC is stored in a register with code value 0x00E.	0x00E
	Short packet	32-bit without ECC is stored in a register with code value 0x00F.	0x00F
Generic Long packet data types <sup>(2)</sup>	Null	Discarded	0x010
	Blanking data	Discarded	0x011
	Embedded 8-bit nonimage data (for example, JPEG)	0x12: Embedded 8-bit nonimage data (for example, JPEG)	0x012
		Send to memory when FORMAT = 0	0x013
		Send to memory when FORMAT = 0	0x014
		Send to memory when FORMAT = 0	0x015
		Send to memory when FORMAT = 0	0x016
		Send to memory when FORMAT = 0	0x017
YUV data	YUV4:2:0 8-bit	YUV4:2:0 8-bit	0x018
	YUV4:2:0 10-bit	YUV4:2:0 10-bit	0x019
	YUV4:2:0 8-bit legacy	YUV4:2:0 8-bit legacy	0x01A
	Reserved	Send to memory when FORMAT = 0	0x01B
	YUV4:2:0 8-bit + CSPS	YUV4:2:0 8 bit + CSPS	0x01C
	YUV4:2:0 10-bit + CSPS	YUV4:2:0 10 bit + CSPS	0x01D
	YUV4:2:2 8-bit	YUV4:2:2 8-bit	YUV4:2:2 8-bit
YUV4:2:2 8-bit + VP		YUV4:2:2 8-bit + VP	0x09E

<sup>(1)</sup> To understand ISS synchronization codes and short packets, see [Section 8.2.5.3.3.3, ISS CSI2 Short Packet](#).

<sup>(2)</sup> To understand ISS synchronization codes and long packets, see [Section 8.2.5.1.1.3.2, ISS CSI2 Long Packet](#).

**Table 8-141. ISS CSI2 MIPI Format Supported by the Protocol Engine (continued)**

Category	MIPI		CSI2 Protocol Engine Support	
	Abbreviation	Register Setting Format Description	Configuration Value for CSI2_CTX_CTRL2_i[9:0] FORMAT	
RGB data		YUV4:2:2 8-bit + VP16	0x0DE	
		YUV4:2:2 10-bit	0x01F	
		RGB444	RGB444 + EXP16	0x0A0
		RGB555	RGB555 + EXP16	0x0A1
		RGB565	RGB565	0x022
		RGB666	RGB666 + EXP32	0x0E3
			RGB666 + EXP32_24	0x033
		RGB888	RGB888	0x024
			RGB888 + EXP32	0x0E4
		Reserved	Send to memory when FORMAT = 0	0x025
		Reserved	Send to memory when FORMAT = 0	0x026
	Reserved	Send to memory when FORMAT = 0	0x027	
RAW data	RAW6	RAW6	0x028	
		RAW6 + EXP8	0x068	
		RAW6 + DPCM10 + VP	0x0E8	
		RAW6 + DPCM10 + EXP16	0x2A8	
		RAW6 + DPCM12 + VP	0x368	
		RAW6 + DPCM12 + EXP16	0x3A8	
	RAW7	RAW7	0x029	
		RAW7 + EXP8	0x069	
		RAW7 + DPCM10 + EXP16	0x229	
		RAW7 + DPCM10 + VP	0x329	
		RAW7 + DPCM12 + EXP16	0x369	
		RAW7 + DPCM12 + VP	0x3A9	
	RAW8	RAW8	0x02A	
		RAW8 + VP	0x12A	
		RAW8 + DPCM10 + EXP16	0x2AA	
		RAW8 + DPCM10 + VP	0x32A	
		RAW8 + DPCM12 + EXP16	0x36A	
		RAW8 + DPCM12 + VP	0x3AA	
	RAW10	RAW10	0x02B	
		RAW10 + EXP16	0x0AB	
		RAW10 + VP	0x12F	
	RAW12	RAW12	0x02C	
		RAW12 + EXP16	0x0AC	
		RAW12 + VP	0x12C	
	RAW14	RAW14	0x02D	
		RAW14 + EXP16	0x0AD	
		RAW14 + VP	0x12D	
		Reserved	Send to memory when FORMAT = 0	0x02E
		Reserved	Send to memory when FORMAT = 0	0x02F
	User-defined byte-based data		USER_DEFINED_BYTE_DATA	0x040
		USER_DEFINED_BYTE_DATA + EXP8	0x080	
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C0	

**Table 8-141. ISS CSI2 MIPI Format Supported by the Protocol Engine (continued)**

Category	MIPI		CSI2 Protocol Engine Support
	Abbreviation	Register Setting Format Description	Configuration Value for CSI2_CTX_CTRL2_i[9:0] FORMAT
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x340
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C0
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x140
		USER_DEFINED_BYTE_DATA	0x041
		USER_DEFINED_BYTE_DATA + EXP8	0x081
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C1
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x341
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C1
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x141
		USER_DEFINED_BYTE_DATA	0x042
		USER_DEFINED_BYTE_DATA + EXP8	0x082
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C2
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x342
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C2
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x142
		USER_DEFINED_BYTE_DATA	0x043
		USER_DEFINED_BYTE_DATA + EXP8	0x083
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C3
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x343
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C3
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x143
		USER_DEFINED_BYTE_DATA	0x044
		USER_DEFINED_BYTE_DATA + EXP8	0x084
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C4
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x344
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C4
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x144
		USER_DEFINED_BYTE_DATA	0x045
		USER_DEFINED_BYTE_DATA + EXP8	0x085
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C5
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x345

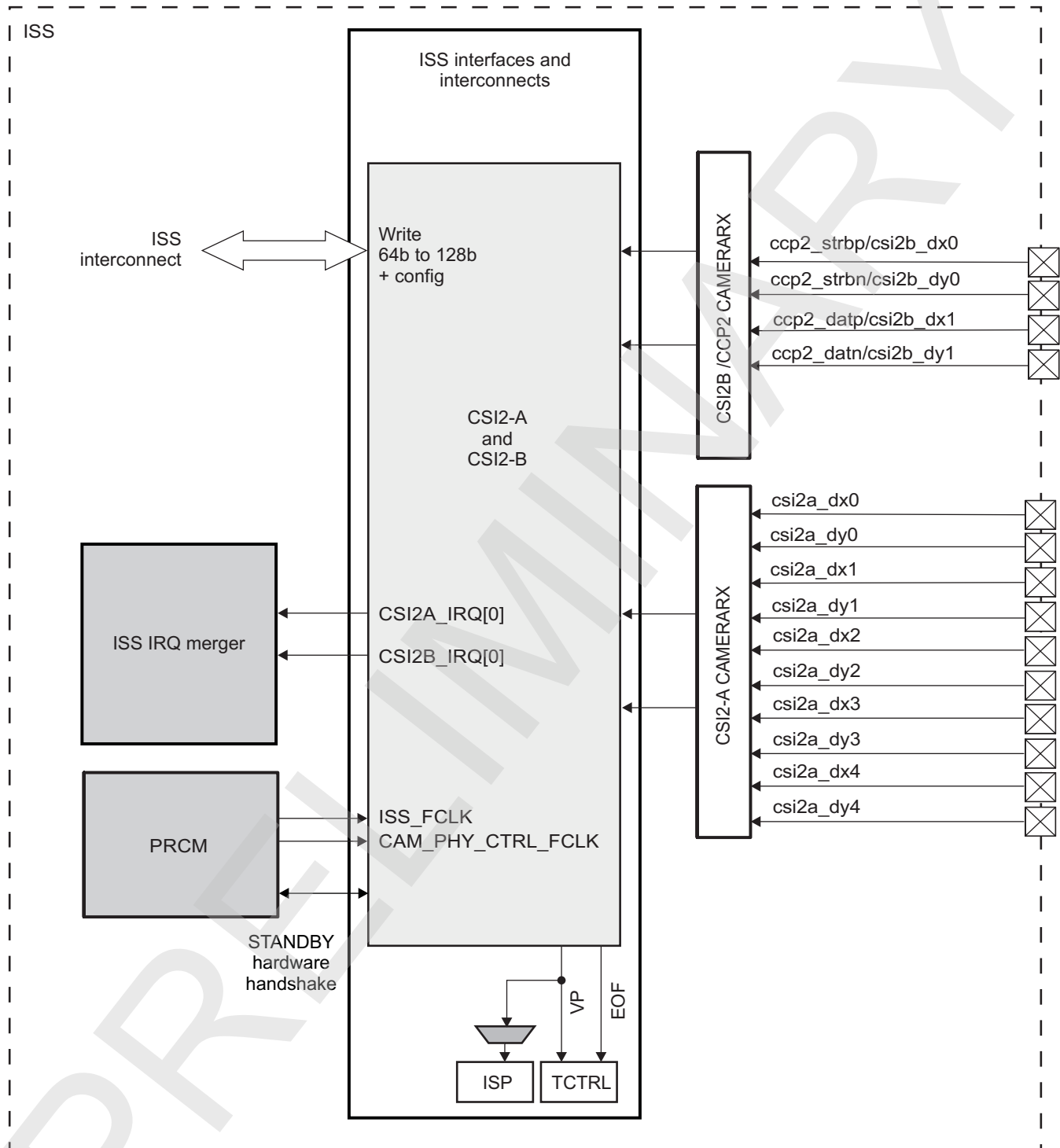
**Table 8-141. ISS CSI2 MIPI Format Supported by the Protocol Engine (continued)**

Category	MIPI		CSI2 Protocol Engine Support
	Abbreviation	Register Setting Format Description	Configuration Value for CSI2_CTX_CTRL2_i[9:0] FORMAT
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C5
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x145
		USER_DEFINED_BYTE_DATA	0x046
		USER_DEFINED_BYTE_DATA + EXP8	0x086
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C6
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x346
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C6
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x146
		USER_DEFINED_BYTE_DATA	0x047
		USER_DEFINED_BYTE_DATA + EXP8	0x087
		USER_DEFINED_BYTE_DATA + DPCM10 + EXP16	0x2C7
		USER_DEFINED_BYTE_DATA + DPCM10 + VP	0x347
		USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	0x1C7
		USER_DEFINED_BYTE_DATA + DPCM12 + VP	0x147
Reserved		Send to memory when FORMAT = 0	0x038
		Send to memory when FORMAT = 0	0x039
		Send to memory when FORMAT = 0	0x03A
		Send to memory when FORMAT = 0	0x03B
		Send to memory when FORMAT = 0	0x03C
		Send to memory when FORMAT = 0	0x03D
		Send to memory when FORMAT = 0	0x03E
		Send to memory when FORMAT = 0	0x03F

### 8.2.5.2 ISS CSI2 Integration

Figure 8-63 is an overview of the integration of the CSI2-A/CSI2-B interface in the device. The figure is the top-level block diagram of the CSI2-A/CSI2-B receiver. The CSI2-A/CSI2-B receiver receives the serial data coming from a CSI2 compatible image sensor, converts it to parallel data, extracts the logical channels, detects and extracts the synchronization codes, reformats the data, and outputs it through the video port or the ISS interconnect interface.

Figure 8-63. ISS CSI2-A/CSI2-B Integration



camss-352b

The CSI2-A/CSI2-B receiver can send data directly to system memory using the master port or send it to the camera ISP using the video port.

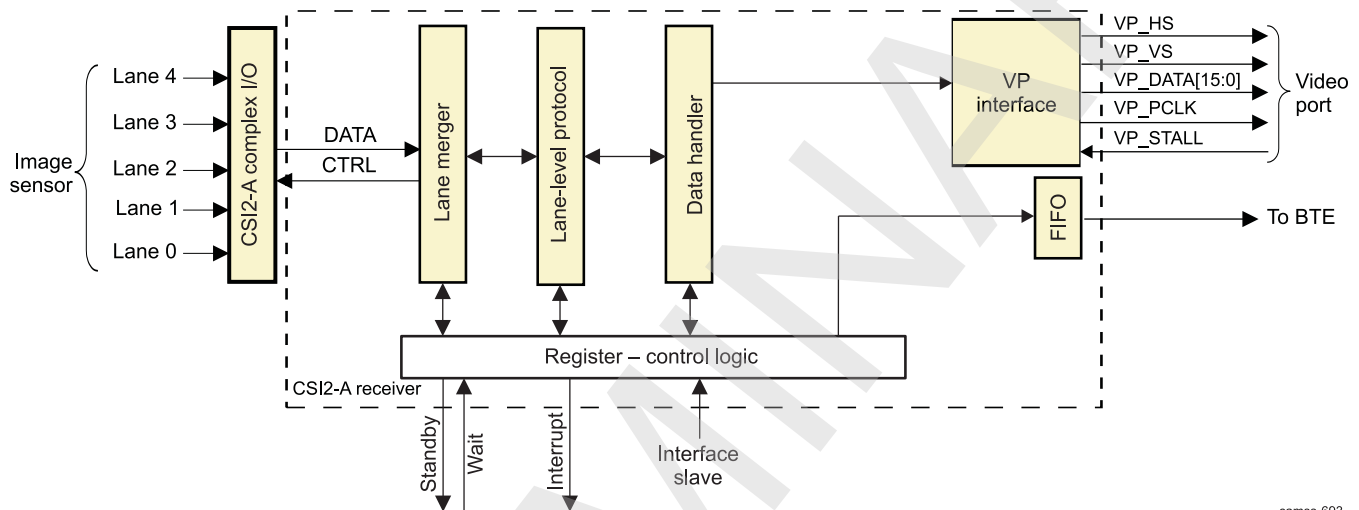
For power domain, clocks, reset, and hardware requests, see [Section 8.1.2.4, ISS Power Management](#).

### 8.2.5.3 ISS CSI2 Functional Description

#### 8.2.5.3.1 ISS CSI2 Overview

Figure 8-64 is the CSI2-A receiver block diagram (it assumes there are four CSI2 image sensor data lines). The only difference for CSI2-B is the number of lanes coming into the complex I/O. The CSI2 receiver receives the byte data coming from a CSI2 D-PHY receiver (up to four data pairs), converts it to byte stream, detects and corrects errors, extracts the virtual channel ID, detects and extracts the synchronization codes, reformats the data, and outputs it through the video port or the ISS interconnect interface. Data communication between CSI2 and ISP is done through the video port.

Figure 8-64. ISS CSI2-A Receiver Block Diagram



camss-693

#### 8.2.5.3.2 ISS CSI2 Features

The CSI2 receiver is a master on the L3 interconnect for storing data in memory and a slave on the level 4 (L4) interconnect for register access.

The main features of the CSI2 receiver are:

- Transfer pixels and data received by the CSI2 PHY to the system memory or video processor
- Unidirectional data link
- Supports up to four data-configurable links in addition to the clock signaling (minimum of one data link and maximum of four depending on the speed)
- Data merger for two, three, or four data lane configurations
- Error detection and correction by the protocol engine
- DMA engine integrated with dedicated FIFO
- 1D and 2D addressing modes
- Up to eight contexts to support eight dedicated configurations of virtual channel ID and data types
- Ping-pong mechanism for double-buffering
- JPEG support for unknown length transfer (no extraction of the thumbnail)
- Supports all primary and secondary MIPI-defined formats (RGB, RAW, and YUV)
- Storage in progressive mode for interlaced stream (using line numbering)
- Conversion to the RGB formats
- Decompressions of the RAW formats
- RAW frame transcoding, including DPCM and A-Law compression
- Fully configurable interface of the complex PHY I/O: position of the clock and data and order of  $\pm$  differential signals for each pair



### 8.2.5.3.3 ISS CSI2 Functional Description

#### 8.2.5.3.3.1 ISS CSI2 Physical Layer Lane Configuration

The CSI2 serial interface is a unidirectional differential serial interface with data/clock for the physical layer.

The maximum CSI2 receiver data transfer capacity is 1000 Mbps per data lane.

Data-clock signaling consists of two to five differential signal pairs: from one to four data lanes and one clock lane:

- The data signal carries the bit-serial data. The CSI2 transmitter in the image sensor sends the data in-quadrature with the dual-data rate (DDR) clock in HS mode; otherwise, the clock is extracted from the received data in LS mode. Data is transmitted byte-wise, LSB first. The CSI2 complex I/O receives the data and sends the byte stream to the CSI2 receiver.
- The clock signal carries the DDR clock signal.

Each physical lane can be a data or clock lane with a restriction to the fourth line, which can only be data (see [Section 8.2.3.1, ISS CSI2 PHY Overview](#)). The clock/data lane must be configured before transmission to indicate the byte order, while merging the received bytes into a byte stream shows the reachable speed per data lane function of data lane numbers.

Lanes are configured through the [CSI2\\_COMPLEXIO\\_CFG](#) registers for CSI2-A and CSI2-B PHY, respectively. The [CSI2\\_COMPLEXIO\\_CFG\[2:0\] CLOCK\\_POSITION](#) bit field and the [CSI2\\_COMPLEXIO\\_CFG\[3\] CLOCK\\_POL](#) bit configure which lane transmits the clock and define its polarity. [DATAI\\_POSITION](#) and [DATAI\\_POL](#) configure the data lanes and their polarity, where *I* is the number of the data lane (*I* = 1 to 4). When the [DATAI\\_POSITION](#) field is set to 0, data lane *I* is not used.

#### CAUTION

The only exception is the four-data-lane use case, in which one corner lane is allowed to be only a data lane. That is, the clock lane cannot be in position 1 or position 5 in the four-data-lane configuration.

#### 8.2.5.3.3.2 ISS CSI2 ECC and Checksum Generation

The CSI2 receiver includes an ECC in the packet header and a checksum in the packet footer for long-packet transmission. These two fields can be used to detect and/or correct errors in the received packet.

##### 8.2.5.3.3.2.1 ISS CSI2 ECC

To detect and correct transmission errors of the header of short and long packets, an 8-bit ECC is included in the header of packets (short and long packet).

The ECC concerns all the fields for a short packet (data ID and short-packet data field) and the packet header for a long packet (data ID and word count). The ECC can only correct one error. Additional errors cannot be repaired, but they are flagged.

The CSI2 receiver ECC is compared against the CSI2 transmitter ECC embedded in the bitstream. If the ECC does not match, an interrupt is triggered to the host central processing unit (CPU).

For long and short packets, the correction is always done if there is only one error per packet header.

An ECC error with or without correction can be reported at two levels, depending on the type of packet. [Table 8-142](#) describes the field in which events are logged. Logging cannot be disabled, but users can set the corresponding bit in the [CSI2\\_IRQENABLE](#) and [CSI2\\_CTX\\_IRQENABLE\\_i](#) registers to prevent event generation at a higher level.

**Table 8-142. ISS CSI2 ECC Event Logging**

	Short Packet	Long Packet
With correction	Global <a href="#">CSI2_IRQSTATUS</a> [12] ECC_CORRECTION_IRQ	Context <a href="#">CSI2_CTX_IRQSTATUS_i</a> [8] ECC_CORRECTION_IRQ
Without correction	Global <a href="#">CSI2_IRQSTATUS</a> [11] ECC_NO_CORRECTION_IRQ	Global <a href="#">CSI2_IRQSTATUS</a> [11] ECC_NO_CORRECTION_IRQ

The ECC check can be disabled (short and long packet) by setting the [CSI2\\_CTRL](#)[2] ECC\_EN bit to 0. Setting the bit to 1 enables the ECC check.

#### 8.2.5.3.3.2 ISS CSI2 Checksum

To detect errors in transmission of the payload of long packets, a 16-bit CRC checksum is computed on the payload of the long packets in the transmitter. This CRC is stored in the packet footer. A CRC is also computed in the CSI2 receiver. If the checksums do not match, an interrupt is triggered to the host CPU.

CRC errors are logged in the CS\_IRQ field of the corresponding context register, [CSI2\\_CTX\\_IRQSTATUS\\_i](#). Logging cannot be disabled, but users can set the corresponding bit in the [CSI2\\_CTX\\_IRQENABLE\\_i](#) register to prevent event generation at a higher level.

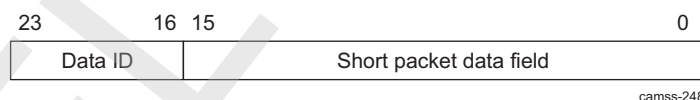
The CRC can be disabled for a specific context by setting the [CSI2\\_CTRL](#)[5] CS\_EN bit to 0. Setting the bit to 1 enables the CRC.

#### 8.2.5.3.3.3 ISS CSI2 Short Packet

There are two types of short packets in the CSI2 receiver:

- Synchronization short packet: Used by the protocol engine to synchronize frame and line (data ID from 0x0 to 0x7)
- Generic short packet: User-dependent; not treated by the protocol engine (data ID from 0x8 to 0xF)

When a generic short packet is received by the CSI2 receiver, the ECC check is performed if it is enabled. Then, the short packet is written in the [CSI2\\_SHORT\\_PACKET](#)[23:0] SHORT\_PACKET bit field. The ECC field is deleted from the short packet. [Figure 8-65](#) shows the SHORT\_PACKET bit field format.

**Figure 8-65. ISS CSI2 SHORT\_PACKET Bit Field Format**

When a short packet is stored, an event is logged in the [CSI2\\_IRQSTATUS](#)[13] SHORT\_PACKET\_IRQ bit. Logging cannot be disabled, but users can set the corresponding bit in the [CSI2\\_IRQENABLE](#) register to prevent event generation at a higher level.

The application reads the [CSI2\\_SHORT\\_PACKET](#) register before the next short packet with a code from 0x8 to 0xF. There is a single register for capturing the generic short packet, because no data type in it is associated with context.

#### 8.2.5.3.3.4 ISS CSI2 Virtual Channel and Context

The CSI2 protocol layer transports virtual channels. The virtual channels separate different data flows interleaved in the same data stream. Each virtual channel is identified by a unique channel identification number in the packet header. This channel identification number is encoded in the 2-bit code.

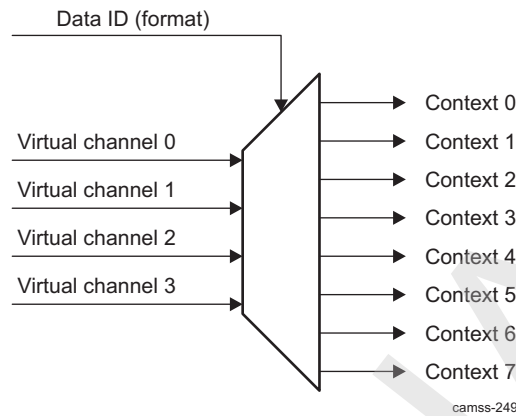
The CSI2 receiver monitors the channel identifier number and demultiplexes the interleaved data streams. The CSI2 receiver supports up to four concurrent virtual channels.

The CSI2 receiver supports eight contexts with their events to control the four possible virtual channels and the different data transmitted through them. A context is linked to a specific data type transported by a given virtual channel. The following bit fields permit configuration of a context:

- [CSI2\\_CTX\\_CTRL2\\_i](#)[12:11] VIRTUAL\_ID: Configures the virtual ID linked to the current context

- [CSI2\\_CTX\\_CTRL2\\_i\[9:0\] FORMAT](#): Configures the data format linked to the current context  
[Figure 8-66](#) shows the relationships between virtual channels and contexts.

**Figure 8-66. ISS CSI2 Virtual Channel to Context**



Each context consists of eight registers: six registers to control the corresponding context and two to log and enable events from the context. All registers in a context can be modified at any time; however, modifications apply only from the start of the following frame.

A context can be enabled independently by setting the [CSI2\\_CTX\\_CTRL1\\_i\[0\] CTX\\_EN](#) bit to 1; setting this bit to 0 disables the corresponding context.

When acquiring frames on a context, users can write the number of frames to capture in the [CSI2\\_CTX\\_CTRL1\\_i\[15:8\] COUNT](#) bit field. Acceptable values are 0 to 255; 0 stands for infinite capture (no count). After each frame is acquired, the count value is decremented by 1. When the count value reaches 0, the [CSI2\\_CTX\\_IRQSTATUS\\_i\[6\] FRAME\\_NUMBER\\_IRQ](#) event is set and the [CTX\\_EN](#) bit is set to 0. To write a value in the [COUNT](#) bit field, the [CSI2\\_CTX\\_CTRL1\\_i\[4\] COUNT\\_UNLOCK](#) bit must be set to 1. If the value of the [COUNT\\_UNLOCK](#) bit is 0, a write in the [COUNT](#) bit field has no effect.

The [CSI2\\_CTX\\_CTRL3\\_i\[15:0\] LINE\\_NUMBER](#) bit field configures the generation of the [CSI2\\_CTX\\_IRQSTATUS\\_i\[7\] LINE\\_NUMBER\\_IRQ](#) event. The [CSI2\\_CTX\\_CTRL1\\_i\[1\] LINE\\_MODULO](#) bit configures how the [LINE\\_NUMBER](#) event is generated:

- 0: The event is generated one time by frame.
- 1: The event is generated modulo [LINE\\_NUMBER](#) (the event can be generated more than once in a frame).

During a frame capture, the [CSI2\\_CTX\\_CTRL2\\_i\[31:16\] FRAME\\_NUMBER](#) bit field shows the number that identifies the frame received.

### 8.2.5.3.3.5 ISS CSI2 DMA Engine

The CSI2 receiver integrates its own DMA engine with dedicated FIFO.

Global DMA configuration is common to the eight channels and is defined in the [CSI2\\_CTRL](#) register. Configuration of the ping-pong address and the offset between lines is specific for a given context; therefore, each context has its own DMA configuration registers.

The DMA engine supports:

- 1D addressing mode (no address line offset, [CSI2\\_CTX\\_DAT\\_OFST\\_i = 0](#))
- 2D addressing mode (address line offset different than 0, [CSI2\\_CTX\\_DAT\\_OFST\\_i = 0](#))

The burst size is defined in the [CSI2\\_CTRL\[6:5\] BURST\\_SIZE](#) bit field and the [CSI2\\_CTRL\[16\] BURST\\_SIZE\\_EXPAND](#) bit. The DMA uses the burst size or smaller sizes down to single open-core protocol (OCP) writes depending on the alignment at the end of lines. The DMA engine can handle burst requests. When the burst requests can be used, as soon as one burst of data is present in the FIFO, the DMA engine initiates a burst write. The burst size is defined in the [CSI2\\_CTRL\[6:5\] BURST\\_SIZE](#) bit field and the [CSI2\\_CTRL\[16\] BURST\\_SIZE\\_EXPAND](#) bit.

**NOTE:** Unless there are specific requirements, CSI2 (also applies to all other ISS initiators) must be configured to use only a burst size of 128 bytes and nonposted writes.

When single requests must be used, as soon as one element (the size depends on the data type and the post-processing: DPCM, EXT, etc.) is present in the FIFO, the DMA engine initiates a single write.

Interleave mode is dedicated by the CSI2 receiver only when the line numbers are received (short packets). The line number is used to calculate the start address of the line.

The DMA starts to write in memory using the `CSI2_CTX_DAT_PING_ADDR_i[31:5]` ADDR bit field for the first frame to be transferred, and then uses the `CSI2_CTX_DAT_PONG_ADDR_i[31:5]` ADDR bit field and the ping address alternately. Thus, the first frame uses the ping address, the second frame uses the pong address, the third frame uses the ping address, and so on.

The `CSI2_CTX_CTRL1_i[3]` PING\_PONG status bit indicates whether the ping address (`CSI2_CTX_DAT_PING_ADDR_i`) or the pong address (`CSI2_CTX_DAT_PONG_ADDR_i`) was used to store the pixel data of the last frame. After reset or after a 0-to-1 edge transition in the `CSI2_CTRL[0]` IF\_EN bit, the pixel data is written in the ping buffer and the `CSI2_CTX_CTRL1_i[3]` PING\_PONG bit = PONG. When the number of FECs received equals the value programmed in the `CSI2_CTX_CTRL1_i[23:16]` FEC\_NUMBER bit field, the pixel data are written in the pong buffer and `CSI2_CTX_CTRL1_i[3]` PING\_PONG = PING. `CSI2_CTX_CTRL1_i[3]` PING\_PONG toggles after the `CSI2_CTX_CTRL1_i[23:16]` FEC\_NUMBER FEC sync code with the virtual channel ID defined is received in the `CSI2_CTX_CTRL2_i[12:11]` VIRTUAL\_ID bit field.

The `CSI2_CTX_CTRL1_i[23:16]` FEC\_NUMBER bit field must be set as follows:

- In progressive mode, set to 1.
- In interlaced mode, set to the number of interlaced frames to recreate a progressive image in the PING\_PONG buffer.

#### 8.2.5.3.3.5.1 ISS CSI2 Progressive Frame to Progressive Storage

After each line, a new start line address is computed, depending on the value of the `CSI2_CTX_DAT_OFST_i[31:5]` OFST bit field:

- If OFST = 0, the new line starts immediately after the last pixel (data are written contiguously in memory).
- Otherwise, the value of OFST sets the offset between the first pixel of the previous line and the first pixel of the current line in memory.

For the ping frame:

$$\text{@Line0} = \text{CSI2\_CTX\_DAT\_PING\_ADDR\_i@Line1} = \text{@Line0} + \text{CSI2\_CTX\_DAT\_OFST\_i@Line2} = \text{@Line1} + \text{CSI2\_CTX\_DAT\_OFST\_i}$$

For the pong frame:

$$\text{@Line0} = \text{CSI2\_CTX\_DAT\_PONG\_ADDR\_i@Line1} = \text{@Line0} + \text{CSI2\_CTX\_DAT\_OFST\_i@Line2} = \text{@Line1} + \text{CSI2\_CTX\_DAT\_OFST\_i}$$

#### 8.2.5.3.3.5.2 ISS CSI2 Interlaced Frame to Progressive Storage

The mode is functional only when the line numbers are transmitted. It is automatically enabled without setting.

For the ping frame:

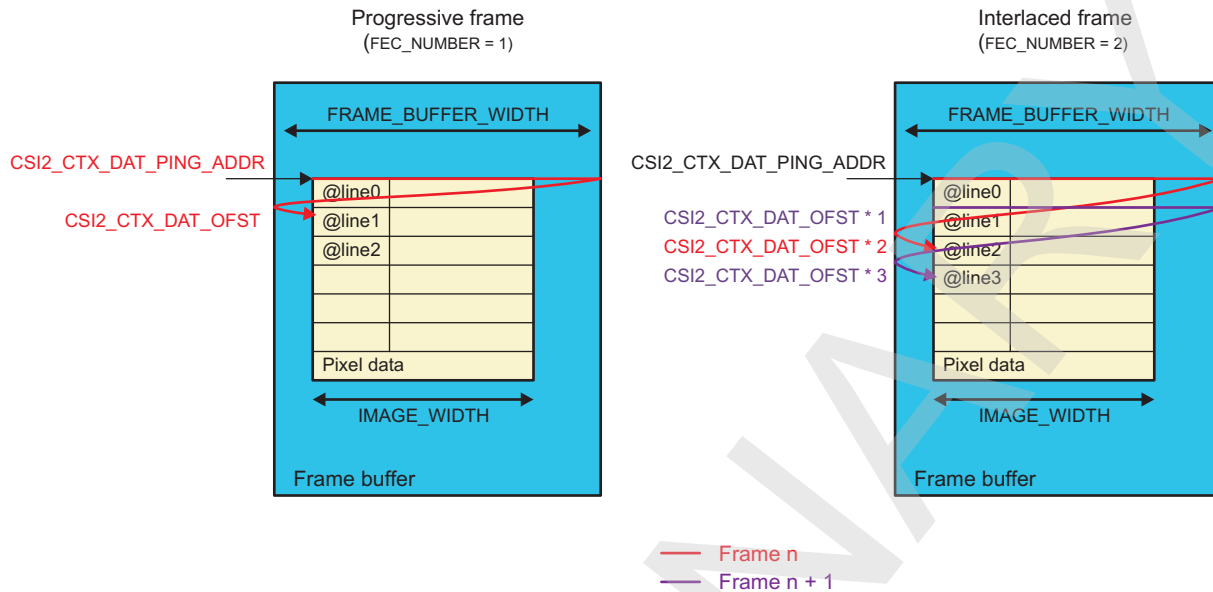
$$\text{@LineX} = \text{CSI2\_CTX\_DAT\_PING\_ADDR\_i} + \text{CSI2\_CTX\_DAT\_OFST\_i} * \text{Line\_Number}$$

For the pong frame:

$$\text{@LineX} = \text{CSI2\_CTX\_DAT\_PONG\_ADDR\_i} + \text{CSI2\_CTX\_DAT\_OFST\_i} * \text{Line\_Number}$$

Figure 8-67 shows how data are stored in memory regarding the DMA configuration.

**Figure 8-67. ISS CSI2 Pixel Data Destination Setting in Progressive and Interlaced Mode**



camss-250

The burst size is defined in the [CSI2\\_CTRL\[6:5\] BURST\\_SIZE](#) bit field for bursts up to 16 × 64 bits or the [CSI2\\_CTRL\[16\] BURST\\_SIZE\\_EXPAND](#) bit for 16 × 128-bit bursts. It can be changed only while the [CSI2\\_CTRL\[0\] IF\\_EN](#) bit is reset to 0. The recommended value is the [CSI2\\_CTRL\[16\] BURST\\_SIZE\\_EXPAND](#) bit set to 1, which defines a burst of 16 × 64 bits (the maximum value); otherwise, by default it is set to 8 × 64 bits. When the BURST\_SIZE\_EXPAND bit is set, the BURST\_SIZE setting has no effect. The DMA uses nonposted writes by default. The [CSI2\\_CTRL\[13\] NON\\_POSTED\\_WRITE](#) bit must be set to 1 to match DMA default configuration. It can be changed only while the [CSI2\\_CTRL\[0\] IF\\_EN](#) bit is reset to 0.

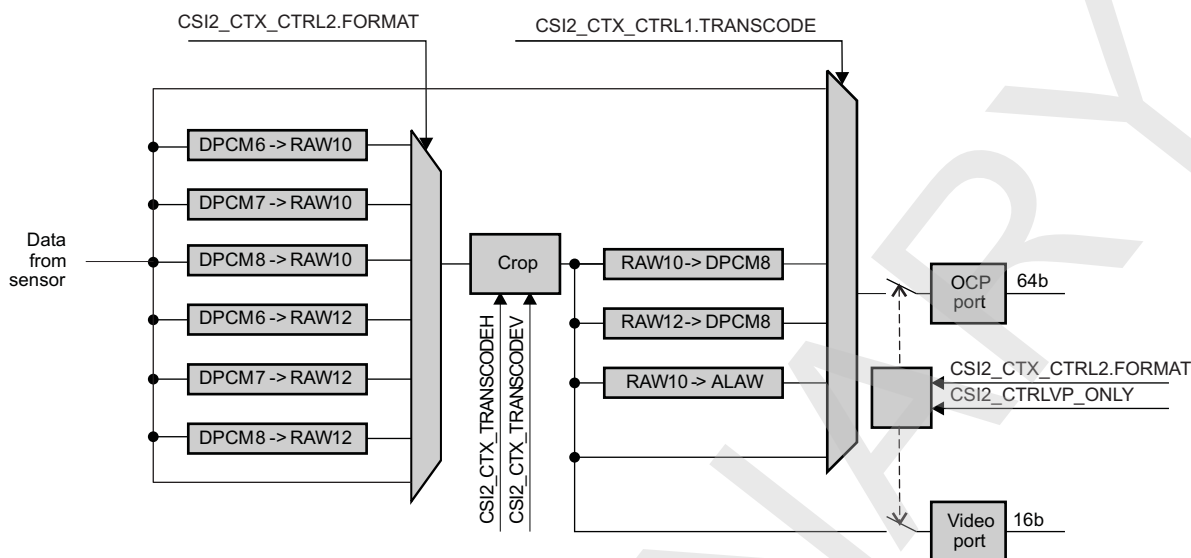
### 8.2.5.3.3.6 ISS CSI2 Transcoding

Image transcoding is used mainly to reduce memory footprint and bandwidth when:

- The sensor does not support DPCM compression. In fact, A-Law and DPCM compressed pixels occupy only 6, 7, or 8 BPP of storage.
- Digital zoom is used
  - Data that is not going to be used by further processing does not need to be stored in system memory.
  - Pixels cannot be accessed from random locations in a DPCM-compressed frame. Transcoding avoids memory-to-memory processing of unused pixels.

[Figure 8-68](#) shows the logical representation of the image transcoding operation.

- Data is extracted from the CSI2 stream by the protocol engine.
- It is DPCM decompressed if necessary. That is the case when the received stream is DPCM-compressed and transcoding has been enabled using the [CSI2\\_CTX\\_CTRL1\\_i\[27:24\] TRANSCODE](#) bit field.
- Data sent to the video port cannot be compressed: it is intended to be processed by the ISS ISP. Data sent to system memory can be optionally compressed.
- Internal data are aligned on MSB when they enter the cropping stage. For example:
  - 4 LSBs are 0s when RAW10 data are handled.
  - 2 LSBs are 0s when RAW12 data are handled.

**Figure 8-68. ISS CSI2 Frame Processing**

camss-255

Table 8-143 shows the input format provided to the cropping engine for a given pixel format provided by the sensor. Formats not listed in the table are not supported for transcoding. The FORMAT and Corresponding Setting Value column corresponds to the value set in the `CSI2_CTX_CTRL2_i[9:0]` FORMAT register.

**Table 8-143. ISS CSI2 Supported Transcoding Input Formats**

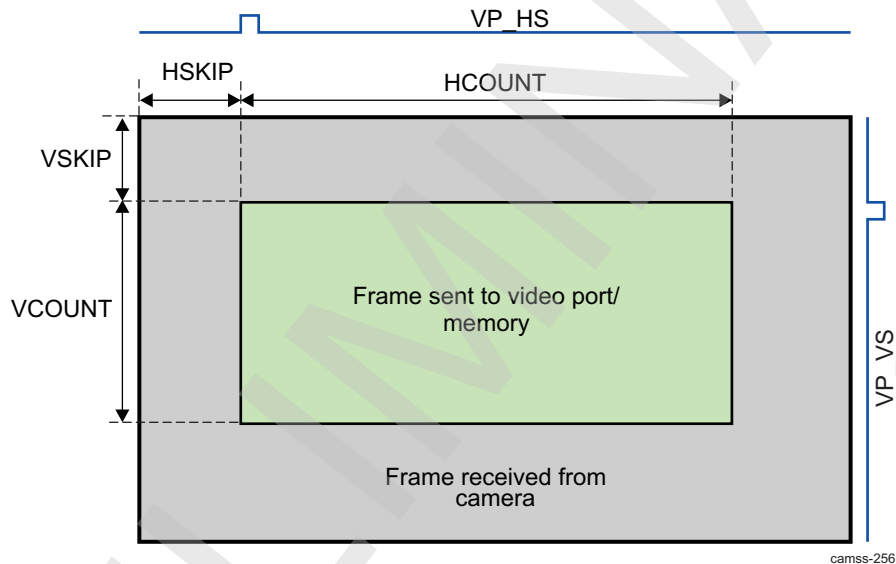
<code>CSI2_CTX_CTRL2_i[9:0]</code> FORMAT and Corresponding Setting Value		Cropping Engine Input Format	DPCM Decomposition Enabled	Video Port Enabled
0x028	RAW6	RAW6		
0x058	RAW6 + EXP8			
0x029	RAW7		RAW7	
0x059	RAW7 + EXP8			
0x02A	RAW8	RAW8		
0x12A	RAW8 + VP			Yes
0x02B	RAW10	RAW10		
0x0AB	RAW10 + EXP16			
0x0E8	RAW6 + DPCM10 + VP		Yes	Yes
0x12F	RAW10 + VP			Yes
0x229	RAW7 + DPCM10 + EXP16		Yes	
0x2A8	RAW6 + DPCM10 + EXP16		Yes	
0x2AA	RAW8 + DPCM10 + EXP16		Yes	
0x329	RAW7 + DPCM10 + VP		Yes	Yes
0x32A	RAW8 + DPCM10 + VP		Yes	Yes
0x2Cn	USER_DEFINED_BYTE_DATA + DPCM10 + EXP16		Yes	
0x34n	USER_DEFINED_BYTE_DATA + DPCM10 + VP	Yes	Yes	
0x02C	RAW12	RAW12		
0x0AC	RAW12 + EXP16			
0x12C	RAW12 + VP			Yes
0x35A	RAW8 DPCM12 + EXP16		Yes	
0x3AA	RAW8 DPCM12 + VP		Yes	Yes
0x1Cn	USER_DEFINED_BYTE_DATA + DPCM12 + EXP16	Yes		

**Table 8-143. ISS CSI2 Supported Transcoding Input Formats (continued)**

CSI2_CTX_CTRL2_i[9:0] FORMAT and Corresponding Setting Value		Cropping Engine Input Format	DPCM Decomposition Enabled	Video Port Enabled
0x14n	USER_DEFINED_BYTE_DATA + DPCM12 + VP		Yes	Yes
0x3A8	RAW6 + DPCM12 + EXP16		Yes	
0x358	RAW6 + DPCM12 + VP		Yes	Yes
0x359	RAW7 + DPCM12 + EXP16		Yes	
0x3A9	RAW7 + DPCM12 + VP		Yes	Yes
0x02D	RAW14	RAW14		
0x0AD	RAW14 + EXP16			
0x12D	RAW14 + VP			Yes

Image cropping parameters are controlled by software. Figure 8-69 shows the cropping operation.

**Figure 8-69. ISS CSI2 Frame Cropping**



**CAUTION**

Hardware does not check for validity of the settings. The following rules must be respected:

- CSI2\_CTX\_TRANSCODEH\_i[12:0] HSKIP + CSI2\_CTX\_TRANSCODEH\_i[28:16] HCOUNT <= image width
- CSI2\_CTX\_TRANSCODEV\_i[12:0] VSKIP + CSI2\_CTX\_TRANSCODEV\_i[28:16] VCOUNT <= image height

Furthermore, the CSI2\_CTX\_TRANSCODEH\_i[28:16] HCOUNT bit field must comply with the following alignment constraints; otherwise, undefined behavior occurs. Table 8-144 shows the transcode alignment constraints

**Table 8-144. ISS CSI2 Transcode Alignment Constraints**

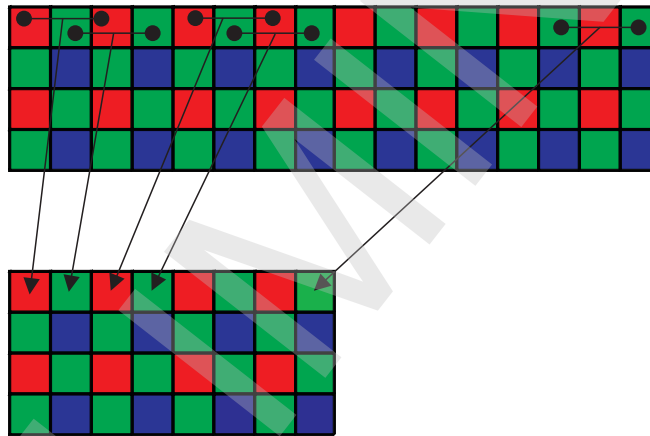
CSI2_CTX_CTRLi[27:24] TRANSCODE Value	Transcode	HCOUNT Must Be Multiple of
0x0	Disabled	1
0x1	DPCM10 RAW8	1

**Table 8-144. ISS CSI2 Transcode Alignment Constraints (continued)**

CSI2_CTX_CTRL1_i[27:24] TRANSCODE Value	Transcode	HCOUNT Must Be Multiple of
0x2	DPCM12 RAW8	1
0x3	ALAW10 RAW8	1
0x4	RAW8	1
0x5	RAW10 + EXP16	1
0x6	RAW10	4
0x7	RAW12 + EXP16	1
0x8	RAW12	2
0x9	RAW10 + EXP16	4

The `CSI2_CTX_CTRL1_i[28]` HSCALE configuration register enables horizontal downscaling of RAW data. It reduces the horizontal size and pixel clock by a factor of 2. The scaler uses a 2-tap horizontal filter operating on samples of the same color plane. The coefficients are:  $[1/2 ; 0 ; 1/2]$

Figure 8-70 shows the scaler operation.

**Figure 8-70. ISS CSI2 Horizontal Scaler**

camss-710u

The scaler can send data to the video port or the interface port. When data goes to the video port, no additional alignment constraints apply. But when data goes to the interface port, HCOUNT/2 must comply with the constraints from Table 8-144 (for example, for RAW10, HCOUNT must be a multiple of 8).

Table 8-145 lists possible combinations of input and output formats supported by the transcoding engine. The Transcode column corresponds to the `CSI2_CTX_CTRL1_i[27:24]` TRANSCODE bit field of a context.

**Table 8-145. ISS CSI2-Supported Transcoding Output Formats**

Cropping Engine Output	Transcode		Supported	Cropping Engine Output	Transcode		Supported
	0	1			0	1	
RAW6	0	Disabled	Yes	RAW10	0	Disabled	Yes
	1	DPCM10 RAW8			1	DPCM10 RAW8	Yes
	2	DPCM12 RAW8			2	DPCM12 RAW8	
	3	ALAW10 RAW8			3	ALAW10 RAW8	Yes
	4	RAW8			4	RAW8	
	5	RAW10 + EXP16			5	RAW10 + EXP16	Yes
	6	RAW10			6	RAW10	Yes
	7	RAW12 + EXP16			7	RAW12 + EXP16	
	8	RAW12			8	RAW12	
	9	RAW14			9	RAW14	



**Table 8-145. ISS CSI2-Supported Transcoding Output Formats (continued)**

Cropping Engine Output	Transcode		Supported	Cropping Engine Output	Transcode		Supported
RAW7	0	Disabled	Yes	RAW12	0	Disabled	Yes
	1	DPCM10 RAW8			1	DPCM10 RAW8	
	2	DPCM12 RAW8			2	DPCM12 RAW8	Yes
	3	ALAW10 RAW8			3	ALAW10 RAW8	
	4	RAW8			4	RAW8	
	5	RAW10 + EXP16			5	RAW10 + EXP16	
	6	RAW10			6	RAW10	
	7	RAW12 + EXP16			7	RAW12 + EXP16	Yes
	8	RAW12			8	RAW12	Yes
	9	RAW14			9	RAW14	
RAW8	0	Disabled	Yes	RAW14	0	Disabled	Yes
	1	DPCM10 RAW8			1	DPCM10 RAW8	
	2	DPCM12 RAW8			2	DPCM12 RAW8	
	3	ALAW10 RAW8			3	ALAW10 RAW8	
	4	RAW8	Yes		4	RAW8	
	5	RAW10 + EXP16			5	RAW10 + EXP16	
	6	RAW10			6	RAW10	
	7	RAW12 + EXP16			7	RAW12 + EXP16	
	8	RAW12			8	RAW12	
	9	RAW14			9	RAW14	Yes

RAW pixels are packed into 64-bit words sent to the OCP master port, as defined in:

- [Section 8.2.5.1.1.4.3.3, ISS CSI2 RAW8](#)
- [Section 8.2.5.1.1.4.3.4, ISS CSI2 RAW10](#)
- [Section 8.2.5.1.1.4.3.5, ISS CSI2 RAW12](#)
- [Section 8.2.5.1.1.4.3.6, ISS CSI2 RAW14](#)

For RAW10 and RAW12, software can choose among packed and nonpacked storage. A-Law and DPCM-compressed pixels are stored as RAW8 data: each RAW8 container holds a compressed data point.

Similarly, RAW data is sent over the video port, as described in:

- [Section 8.2.5.1.1.4.3.3, ISS CSI2 RAW8](#)
- [Section 8.2.5.1.1.4.3.4, ISS CSI2 RAW10](#)
- [Section 8.2.5.1.1.4.3.5, ISS CSI2 RAW12](#)
- [Section 8.2.5.1.1.4.3.6, ISS CSI2 RAW14](#)

Enabling of the OCP/video port is controlled by the [CSI2\\_CTX\\_CTRL2\\_i\[9:0\]](#) FORMAT bit field and the [CSI2\\_CTRL\[11\]](#) VP\_ONLY\_EN and [CSI2\\_CTX\\_CTRL1\\_i\[2\]](#) VPFORCE bits.

To enable transcoding, software configures the context normally and also configures the framing using the [CSI2\\_CTX\\_TRANSCODEV\\_i](#) and [CSI2\\_CTX\\_TRANSCODEH\\_i](#) registers. Software defines the after transcoding with the [CSI2\\_CTX\\_CTRL1\\_i\[27:24\]](#) TRANSCODE bit field.

### 8.2.5.3.3.7 ISS CSI2 EndOfFrame and EndOfLine (EOF and EOL) Pulses

The CSI2 receiver generates two signals to qualify the last pixel of a frame and the last pixel of a line to the TCTRL. It is active during or after the adequate interface bridge transaction and becomes inactive before the first transaction of the next line. Software can enable/disable generation of those signals for each context using the [CSI2\\_CTX\\_CTRL1\\_i\[7\]](#) EOF\_EN and [CSI2\\_CTX\\_CTRL1\\_i\[6\]](#) EOL\_EN bits. When data is sent to both OCP and video ports, the EOL/EOF timing defined for the OCP port is used.

### 8.2.5.3.3.8 ISS CSI2 Data Decompression

The data compression technique used is DPCM and PCM.

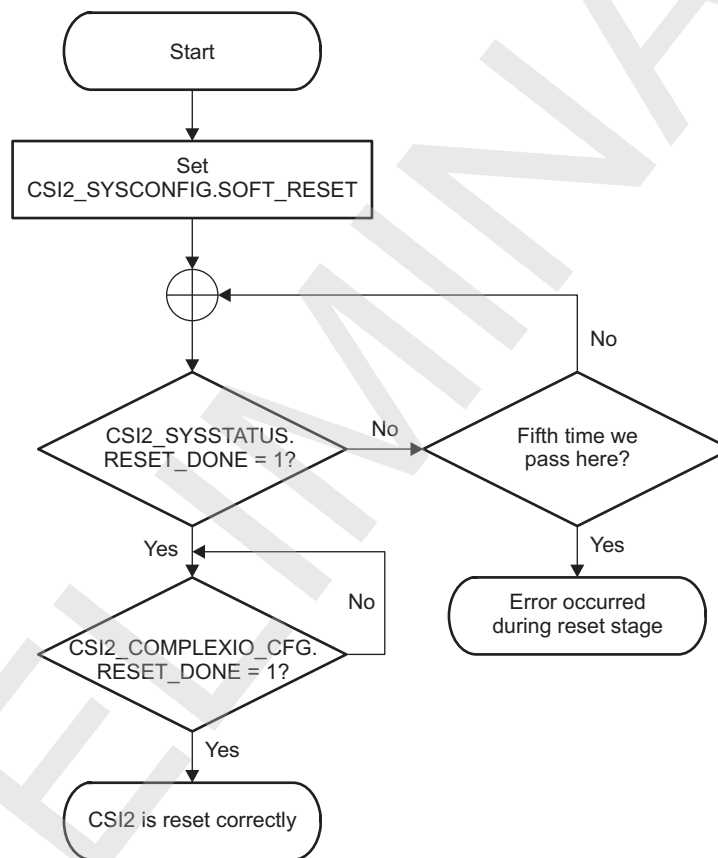
To select the DPCM decompression predictor for the CSI2 Interface, set the `CSI2_CTX_CTRL2_i[10]` DPCM\_PRED bit to 1 for simple predictor or to 0 for advanced predictor.

### 8.2.5.4 ISS CSI2 Programming Model

#### 8.2.5.4.1 ISS CSI2 Reset Management

The CSI2 receiver accepts a general software reset, propagated throughout the hierarchy. This reset can be done to initialize the CSI2 receiver and the complex I/O (A or B) and has the same effect as a hardware reset. [Figure 8-71](#) shows how to reset CSI2 globally.

**Figure 8-71. ISS CSI2 Receiver Global Reset Flow Chart**



camss-252

---

**NOTE:** Before setting the software reset bit to 1 in the `CSI2_SYSCONFIG` register, the user must have access to a CSI2 receiver register.

---



---

**NOTE:** The `CSI2_COMPLEXIO_CFG[29]` RESET\_DONE bit is set to 1 only after the initialization of the CSI2 receiver, CSI2 complex I/O, and external camera completes.

---

#### 8.2.5.4.2 ISS CSI2 Enable Video/Picture Acquisition

Before using the receiver, a CSIPHY initialization in CSI2 mode must be made for CSI2-A CAMERARX, which is associated with the CSI2 receiver. See [Section 8.2.3.2.2, ISS CSI2 PHY and Link Initialization Sequence](#). To start a video/picture acquisition, perform the steps listed in [Table 8-146](#).

**Table 8-146. ISS CSI2 Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Reset the CSI2 receiver.	See <a href="#">Section 8.2.5.4.1, Reset Management</a> .	
Configure the module power management. The module tries to enter smart-standby mode during the vertical blanking period. The <a href="#">CSI2_SYSCONFIG[0]</a> AUTO_IDLE bit keeps its reset value; by default, an automatic port clock gating strategy is applied based on port interface activity.	<a href="#">CSI2_SYSCONFIG[13:12]</a> MSTANDBY_MODE	0x2
Configure the interrupt generation as required. To enable context and/or complex I/O event reporting, enable the corresponding bit field in the <a href="#">CSI2_IRQENABLE</a> register. If the enable bit is at 0, logging is still effective if an event occurs, but is not reported to a higher level.	<a href="#">CSI2_IRQSTATUS</a> and <a href="#">CSI2_IRQENABLE</a>	
Configure the complex I/O interrupt generation as required. If the enable bit is at 0, logging is still effective if an event occurs, but is not reported to a higher level.	<a href="#">CSI2_COMPLEXIO_IRQSTATUS</a> and <a href="#">CSI2_COMPLEXIO_IRQENABLE</a>	
Start complex I/O: Set the <a href="#">CSI2_COMPLEXIO_CFG[28:27]</a> PWR_CMD bit field to 0x1 to pass the complex I/O to the ON state, and then check that the state status reaches the ON state ( <a href="#">CSI2_COMPLEXIO_CFG[26:25]</a> PWR_STATUS = 0x1) (for complex I/O A).	<a href="#">CSI2_COMPLEXIO_CFG[28:27]</a> PWR_CMD	0x1
Configure the complex I/O: <ul style="list-style-type: none"> <li>The complex I/O is fully functional with <a href="#">CSI2_COMPLEX_CFG</a> set at its reset value.</li> <li><a href="#">CSI2_COMPLEX_CFG</a> must be changed according to the data rate being used.</li> </ul>	<a href="#">CSI2_COMPLEXIO_CFG</a>	
Set RXMODE and STOPSTATE FSM to RXMODE state. Users can also configure the delay for the FSM to return from RXMODE to NORXMODE when all lines reach STOPSTATE.	<a href="#">CSI2_TIMING[15]</a> FORCE_RX_MODE_IO1	0x1
Activate ECC correction and error detection on short packets and packet headers. The ECC check corrects the packet if there is one error and generates an error if there is more than one error (unrecoverable error).	<a href="#">CSI2_CTRL[2]</a> ECC_EN	0x1
Start the CSI2 receiver.	<a href="#">CSI2_CTRL[0]</a> IF_EN	0x1
Configure the different contexts to be used.		
Link the context to a virtual channel and a data type.	See <a href="#">Section 8.2.5.4.6, Linking a Context to a Virtual Channel and a Data Type</a> .	
Set the <a href="#">FEC_NUMBER</a> bit field to 0x1 for a progressive video and to 0x2 for an interlaced video. For more information, see <a href="#">Section 8.2.5.3.3.5, DMA Engine</a> .	<a href="#">CSI2_CTX_CTRL1_i[26:23]</a> FEC_NUMBER	0x1 or 0x2
Capture an infinite number of frames (until the interface or the context is disabled).	<a href="#">CSI2_CTX_CTRL1_i[15:8]</a> COUNT and <a href="#">CSI2_CTX_CTRL1_i[4]</a> COUNT_UNLOCK	0x0
Enable the CRC checksum on long packet payload. This allows detection of errors, but cannot correct errors like the ECC for header and short packet. On error detection, an event is triggered (the <a href="#">CSI2_CTX_IRQSTATUS_i[5]</a> CS_IRQ bit).	<a href="#">CSI2_CTX_CTRL1_i[5]</a> CS_EN	
Configure the DMA engine for the current channel: Configure the ping and pong addresses.	<a href="#">CSI2_CTX_DAT_PING_ADDR_i[31:5]</a> ADDR and <a href="#">CSI2_CTX_DAT_PING_ADDR_i[31:5]</a> ADDR	
Set the <a href="#">CSI2_CTX_DAT_OFST_i[15:5]</a> OFST bit field to 0x0 so consecutive lines are stored consecutively in memory (image width and frame-buffer width are equal).	<a href="#">CSI2_CTX_DAT_OFST_i[15:5]</a> OFST	
Keep the ALPHA setting at its reset value (0x0) for RGB padding.	<a href="#">CSI2_CTX_CTRL3_i[29:16]</a> ALPHA	

**Table 8-146. ISS CSI2 Global Initialization (continued)**

Step	Register/Bit Field/Programming Model	Value
Enable the contexts.	CSI2_CTX_CTRL1_i[0] CTX_EN	0x1

#### 8.2.5.4.3 ISS CSI2 Disable Video/Picture Acquisition

There are two ways to end picture acquisition:

- Disable the corresponding context by setting the [CSI2\\_CTX\\_CTRL1\\_i\[0\] CTX\\_EN](#) bit to 0. This stops the acquisition for the current context. Other enabled contexts are still capturing frames and writing them in memory.
- Disable the CSI2 receiver interface by setting the [CSI2\\_CTRL\[0\] IF\\_EN](#) bit to 0. This can have an immediate effect if the [CSI2\\_CTRL\[3\] FRAME](#) bit is set to 0, or it can be effective after all the enabled contexts receive the FEC if the [CSI2\\_CTRL\[3\] FRAME](#) bit is set to 1.

#### 8.2.5.4.4 ISS CSI2 Capture a Finite Number of Frames

The CSI2 receiver can be configured to capture a finite number of frames. To configure the CSI2 receiver in this mode, perform the steps listed in [Table 8-147](#).

**Table 8-147. ISS CSI2 Capture a Finite Number of Frames**

Step	Bit Field	Value
Enable a write to the COUNT bit field.	CSI2_CTX_CTRL1_i[4] COUNT_UNLOCK	0x1
Set the bit field to the number of frames the CSI2 receiver must capture.	CSI2_CTX_CTRL1_i[15:8] COUNT	Valid values are 0 to 255; 0 is infinite capture and 1 to 255 defines the number of frames to capture.
Disable a write to the COUNT bit field.	CSI2_CTX_CTRL1_i[4] COUNT_UNLOCK	0x0

During frame capture, the COUNT bit field is decremented by 1 at each frame capture. Software reads the COUNT bit field to know how many frames must still be captured.

The COUNT bit can be updated during capture if the COUNT\_UNLOCK bit is set to 1.

#### 8.2.5.4.5 ISS CSI2 Configure a Periodic Event During Frame Acquisition

The CSI2 receiver can generate a periodic event. This line number is defined in the [CSI2\\_CTX\\_CTRL3\\_i\[15:0\] LINE\\_NUMBER](#) bit field. The event can be generated once or multiple times per frame, depending on the value of the [CSI2\\_CTX\\_CTRL1\\_i\[1\] LINE\\_MODULO](#) bit:

- If the LINE\_MODULO bit = 0, the event is generated when the line number corresponding to the LINE\_NUMBER bit field is received.
- If the LINE\_MODULO bit = 1, the event is generated when the line number received corresponds to a multiple of the LINE\_NUMBER value (LINE\_NUMBER is used as a modulo).

#### 8.2.5.4.6 ISS CSI2 Linking a Context to a Virtual Channel and a Data Type

The CSI2 receiver supports eight contexts and the CSI2 protocol defines four virtual channels. Therefore, a CSI2 receiver context can be associated with a virtual channel and a data type. Virtual channels are defined by a 2-bit field. Valid data types for the CSI2 receiver with their associated values are described in the [CSI2\\_CTX\\_CTRL2\\_i\[9:0\] FORMAT](#) bit field.

For each context, a [CSI2\\_CTX\\_CTRL2\\_i](#) register defines with which channel and data type the context is associated:

- The VIRTUAL\_ID bit field defines the associated virtual ID transported by the CSI2 protocol from the camera sensor.

- The FORMAT bit field defines the associated data type. The data type is a combination of the data type transported by the CSI2 protocol and the type of storage in memory. A given data type (RGB888) can be stored in memory in different ways (RGB888 or RGB888 + EXP32). Therefore, the FORMAT bit field also defines how DMA stores data in memory.

For example, for the current context to capture a frame from virtual channel 2 and data type RAW12 with data expansion (RAW12 + EXP16), write the value 0x10AC (0x2 11 + 0xAC) in the 16 LSBs of the [CSI2\\_CTX\\_CTRL2\\_i](#) register.

#### 8.2.5.4.7 ISS CSI2 Progressive and Interleaved Frame Configuration

The CSI2 receiver can treat progressive and interlaced frames. There is no progressive or interleaved mode, but the [CSI2\\_CTX\\_CTRL1\\_i\[23:16\]](#) FEC\_NUMBER bit field controls the number of FECs before swapping to the other (ping or pong) buffer. Therefore, two modes are possible:

- FEC\_NUMBER = 1: This is equivalent to progressive mode. After a FEC on the context, the current buffer is switched (ping to pong or pong to ping). The image in the memory buffer consists of one transmitted frame.
- FEC\_NUMBER 1: The current buffer is switched (ping to pong or pong to ping) after the FEC\_NUMBER FEC is received for the context. The image in the memory buffer consists of the FEC\_NUMBER transmitted frame.

For more information about how data is stored in memory through the DMA, see [Section 8.2.5.3.3.5, DMA Engine](#).

---

**NOTE:** If FEC\_NUMBER 1, the camera sensor must send the line number information with the current line. Otherwise, the CSI2 receiver cannot calculate each line address.

---

#### 8.2.5.4.8 ISS CSI2 Progressive and Interleaved Frame Configuration

[Table 8-148](#) lists the procedure to enable debug mode.

**Table 8-148. ISS CSI2 Enable Debug Mode**

Step	Bit	Value
Enable debug mode.	<a href="#">CSI2_CTRL[7]</a> DBG_EN	0x1

- During debug mode the input does not come from the CSI2 receiver interface but from the [CSI2\\_DBG\\_H](#) and [CSI2\\_DBG\\_P](#) registers. The full CSI2 receiver function can be debugged in debug mode. Full 32-bit values must always be written to the [CSI2\\_DBG\\_H](#) register. The [CSI2\\_CTRL\[0\]](#) IF\_EN bit has no affect during debug mode. To reset the FIFO in case of overflow, the [CSI2\\_CTRL\[7\]](#) DBG\_EN bit must be reset to 0, and the interface must be enabled by setting the [CSI2\\_CTRL\[0\]](#) IF\_EN bit to 0x1.
- The [CSI2\\_DBG\\_H](#) register is used to provide short packet and long packet headers.
- The [CSI2\\_DBG\\_P](#) register is used to provide long packet payload.

The following examples apply to the [CSI2\\_DBG\\_H](#) register:

- The sync codes for virtual channel 0 are written as [CSI2\\_DBG\\_H](#) = 0xFF00 0000 or 0xFF00 0001, or 0xFF00 0002 or 0xFF00 0003. To send the RAW12 pixels 0x673, 0x452, 0x01d, 0xefc, 0xab0, 0x891, 0x326, 0x547, write [CSI2\\_DBG\\_H](#) = 0x0123 4567, followed by [CSI2\\_DBG\\_H](#) = 0x89abcdef, and [CSI2\\_DBG\\_H](#) = 0x7654 3210.

### 8.2.5.5 ISS CSI2 Register Manual

#### 8.2.5.5.1 ISS CSI2 Instance Summary

[Table 8-149](#) summarizes the CSI2 instance.

**Table 8-149. ISS CSI2 Instance Summary**

Module Name	L3 Base Address	Size
ISS_CSI2_A_REGS1	0x5200 1000	368 bytes
ISS_CSI2_A_REGS2	0x5200 11C0	64 bytes
ISS_CSI2_B_REGS1	0x5200 1400	368 bytes
ISS_CSI2_B_REGS2	0x5200 15C0	64 bytes

### 8.2.5.5.2 ISS CSI2 REGS1 Registers

#### 8.2.5.5.2.1 ISS CSI2 REGS1 Register Summary

Table 8-150 summarizes the CSI2 REGS1 registers.

**Table 8-150. ISS CSI2 REGS1 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CSI2_A_REGS 1 Base Address	ISS_CSI2_B_REGS 1 Base Address
CSI2_REVISION	R	32	0x0000 0000	0x5200 1000	0x5200 1400
CSI2_SYSCONFIG	RW	32	0x0000 0010	0x5200 1010	0x5200 1410
CSI2_SYSSTATUS	R	32	0x0000 0014	0x5200 1014	0x5200 1414
CSI2_IRQSTATUS	RW	32	0x0000 0018	0x5200 1018	0x5200 1418
CSI2_IRQENABLE	RW	32	0x0000 001C	0x5200 101C	0x5200 141C
CSI2_CTRL	RW	32	0x0000 0040	0x5200 1040	0x5200 1440
CSI2_DBG_H	W	32	0x0000 0044	0x5200 1044	0x5200 1444
RESERVED	R	32	0x0000 0048	0x5200 1048	0x5200 1448
RESERVED	RW	32	0x0000 004C	0x5200 104C	0x5200 144C
CSI2_COMPLEXIO_CFG	RW	32	0x0000 0050	0x5200 1050	0x5200 1450
CSI2_COMPLEXIO_IRQSTATUS	RW	32	0x0000 0054	0x5200 1054	0x5200 1454
RESERVED	RW	32	0x0000 0058	0x5200 1058	0x5200 1458
CSI2_SHORT_PACKET	R	32	0x0000 005C	0x5200 105C	0x5200 145C
CSI2_COMPLEXIO_IRQENABLE	RW	32	0x0000 0060	0x5200 1060	0x5200 1460
RESERVED	RW	32	0x0000 0064	0x5200 1064	0x5200 1464
CSI2_DBG_P	W	32	0x0000 0068	0x5200 1068	0x5200 1468
CSI2_TIMING	RW	32	0x0000 006C	0x5200 106C	0x5200 146C
CSI2_CTX_CTRL1 <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0070 + (0x20 * i)	0x5200 1070 + (0x20 * i)	0x5200 1470 + (0x20 * i)
CSI2_CTX_CTRL2 <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0074 + (0x20 * i)	0x5200 1074 + (0x20 * i)	0x5200 1474 + (0x20 * i)
CSI2_CTX_DAT_OFS <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0078 + (0x20 * i)	0x5200 1078 + (0x20 * i)	0x5200 1478 + (0x20 * i)
CSI2_CTX_DAT_PING_ADDR <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 007C + (0x20 * i)	0x5200 107C + (0x20 * i)	0x5200 147C + (0x20 * i)
CSI2_CTX_DAT_PONG_ADDR <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0080 + (0x20 * i)	0x5200 1080 + (0x20 * i)	0x5200 1480 + (0x20 * i)
CSI2_CTX_IRQENABLE <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0084 + (0x20 * i)	0x5200 1084 + (0x20 * i)	0x5200 1484 + (0x20 * i)
CSI2_CTX_IRQSTATUS <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0088 + (0x20 * i)	0x5200 1088 + (0x20 * i)	0x5200 1488 + (0x20 * i)
CSI2_CTX_CTRL3 <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 008C + (0x20 * i)	0x5200 108C + (0x20 * i)	0x5200 148C + (0x20 * i)

<sup>(1)</sup> i = 0 to 7

8.2.5.5.2.2 ISS CSI2 REGS1 Register Description

Table 8-151 through Table 8-191 describe the CSI2 REGS1 registers.

Table 8-151. CSI2\_REVISION

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See Table 8-150.
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

Table 8-152. Register Call Summary for Register CSI2\_REVISION

ISS Interfaces

- [ISS CSI2 Register Manual: \[0\]](#)

Table 8-153. CSI2\_SYSCONFIG

<b>Address Offset</b>	0x0000 0010
<b>Physical Address</b>	See Table 8-150.
<b>Description</b>	SYSTEM CONFIGURATION REGISTER This register is the OCP-socket system configuration register.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														MSTANDBY_MODE	RESERVED										SOFT_RESET	AUTO_IDLE					

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:12	MSTANDBY_MODE	Sets the behavior of the master port power management signals.  0x0: Force-standby. MStandby is only asserted when the module is disabled.  0x1: No-standby. MStandby is never asserted.  0x2: Smart-standby: MStandby is asserted based on the activity of the module. The module will try to go to standby during the vertical blanking period.	RW	0x0
11:2	RESERVED		R	0x000

Bits	Field Name	Description	Type	Reset
1	SOFT_RESET	Software reset. Set the bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads return 0.  0x0: Normal mode. 0x1: The module is reset Note: Before setting the software reset bit to 1 in <a href="#">CSI2_SYSCONFIG</a> register, the user must have access to a CSI2 receiver register.	RW	0
0	AUTO_IDLE	Internal OCP gating strategy  0x0: OCP clock is free-running. 0x1: Automatic OCP clock gating strategy is applied based on the OCP interface activity.	RW	1

**Table 8-154. Register Call Summary for Register CSI2\_SYSCONFIG**

ISS Interfaces

- [ISS CSI2 Programming Model: \[0\] \[1\] \[2\]](#)
- [ISS CSI2 Register Manual: \[3\] \[4\]](#)

**Table 8-155. CSI2\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	SYSTEM STATUS REGISTER This register provides status information about the module, excluding the interrupt status register.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																RESERVED																	RESET_DONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0.	R	0x0000 0000
0	RESET_DONE	Internal reset monitoring Read 0x1: Reset completed. Read 0x0: Internal module reset is on going.	R	1

**Table 8-156. Register Call Summary for Register CSI2\_SYSSTATUS**

ISS Interfaces

- [ISS CSI2 Register Manual: \[0\]](#)



**Table 8-157. CSI2\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0018
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	INTERRUPT STATUS REGISTER - All contexts This register associates one bit for each context in order to determine which context has generated the interrupt. The context shall be enabled for events to be generated on that context. If the context is disabled, the interrupt is not generated.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OCP_ERR_IRQ	SHORT_PACKET_IRQ	ECC_CORRECTION_IRQ	ECC_NO_CORRECTION_IRQ	RESERVED	COMPLEXIO_ERR_IRQ	FIFO_OVF_IRQ	CONTEXT7	CONTEXT6	CONTEXT5	CONTEXT4	CONTEXT3	CONTEXT2	CONTEXT1	CONTEXT0	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14	OCP_ERR_IRQ	OCP Error Interrupt 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
13	SHORT_PACKET_IRQ	Short packet reception status (other than synch events: Line Start, Line End, Frame Start, and Frame End: data type between 0x8 and x0F only shall be considered). 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
12	ECC_CORRECTION_IRQ	ECC has been used to do the correction of the only 1-bit error status (short packet only). 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
11	ECC_NO_CORRECTION_IRQ	ECC error status (short and long packets). No correction of the header because of more than 1-bit error. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
10	RESERVED	Reserved	R	0
9	COMPLEXIO_ERR_IRQ	Error signaling from complex I/O: status of the PHY errors received from the complex I/O (events are defined in <a href="#">CSI2_COMPLEXIO_IRQSTATUS</a> for the complex I/O). Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0

Bits	Field Name	Description	Type	Reset
8	FIFO_OVF_IRQ	FIFO overflow error status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
7	CONTEXT7	Context 7 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
6	CONTEXT6	Context 6 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
5	CONTEXT5	Context 5 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
4	CONTEXT4	Context 4 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
3	CONTEXT3	Context 3 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
2	CONTEXT2	Context 2 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
1	CONTEXT1	Context 1 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0
0	CONTEXT0	Context 0 Read 0x1: READS: Event is true (pending). Read 0x0: READS: Event is false.	R	0

**Table 8-158. Register Call Summary for Register CSI2\_IRQSTATUS**

## ISS Interfaces

- [ISS CSI2 Functional Description: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS CSI2 Programming Model: \[4\]](#)
- [ISS CSI2 Register Manual: \[5\]](#)

**Table 8-159. CSI2\_IRQENABLE**

<b>Address Offset</b>	0x0000 001C
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	INTERRUPT ENABLE REGISTER - All contexts This register associates one bit for each context in order to enable/disable each context individually.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OCP_ERR_IRQ	SHORT_PACKET_IRQ	ECC_CORRECTION_IRQ	ECC_NO_CORRECTION_IRQ	RESERVED	COMPLEXIO_ERR_IRQ	FIFO_OVF_IRQ	CONTEXT7	CONTEXT6	CONTEXT5	CONTEXT4	CONTEXT3	CONTEXT2	CONTEXT1	CONTEXT0	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14	OCP_ERR_IRQ	OCP Error Interrupt 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
13	SHORT_PACKET_IRQ	Short packet reception (other than synch events: Line Start, Line End, Frame Start, and Frame End: data type between 0x8 and x0F only shall be considered). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
12	ECC_CORRECTION_IRQ	ECC has been used to correct the only 1-bit error (short packet only). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
11	ECC_NO_CORRECTION_IRQ	ECC error (short and long packets). No correction of the header because of more than 1-bit error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
10	RESERVED	Reserved	RW	0
9	COMPLEXIO_ERR_IRQ	Error signaling from complex I/O: the interrupt is triggered when any error is received from the complex I/O (events are defined in <a href="#">CSI2_COMPLEXIO_IRQSTATUS</a> for the complex I/O). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
8	FIFO_OVF_IRQ	FIFO overflow enable 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	CONTEXT7	Context 7 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
6	CONTEXT6	Context 6 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
5	CONTEXT5	Context 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	CONTEXT4	Context 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
3	CONTEXT3	Context 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
2	CONTEXT2	Context 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
1	CONTEXT1	Context 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	CONTEXT0	Context 0 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

**Table 8-160. Register Call Summary for Register CSI2\_IRQENABLE**

ISS Interfaces

- [ISS CSI2 Functional Description: \[0\] \[1\]](#)
- [ISS CSI2 Programming Model: \[2\] \[3\]](#)
- [ISS CSI2 Register Manual: \[4\]](#)

**Table 8-161. CSI2\_CTRL**

<b>Address Offset</b>	0x0000 0040
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	GLOBAL CONTROL REGISTER This register controls the CSI2 RECEIVER module. This register shall not be modified dynamically (except IF_EN bit field).
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
RESERVED																BURST_SIZE_EXPAND	VP_CLK_EN	RESERVED	NON_POSTED_WRITE	RESERVED	VP_ONLY_EN	STREAMING_32_BIT	VP_OUT_CTRL	DBG_EN	BURST_SIZE	ENDIANNESS	FRAME	ECC_EN	RESERVED	IF_EN																

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000
16	BURST_SIZE_EXPAND	Sets the DMA burst size on the L3 interconnect. 0x0: Use the burst size defined in the BURST_SIZE register 0x1: Allow generation of 16x64-bit bursts	RW	0
15	VP_CLK_EN	VP clock enable. 0x0: The VP clock is disabled. 0x1: The VP clock is enabled.	RW	0
14	RESERVED	Read returns reset value	RW	0
13	NON_POSTED_WRITE	Not posted writes 0x0: Disable 0x1: Enable	RW	0

Bits	Field Name	Description	Type	Reset
12	RESERVED		R	0
11	VP_ONLY_EN	VP only enable. 0x0: The VP is enabled and the OCP master port is enabled. 0x1: The VP is enabled and the OCP master port is disabled.	RW	0
10	STREAMING_32_BIT	Indicates if 64-bit or 32-bit streaming burst is used. Valid only if <a href="#">CSI2_CTRL.STREAMING=1</a> 0x0: 64-bit streaming burst is used; byte enable pattern is 0xFF 0x1: 32-bit streaming burst is used; byte enable pattern is 0x0F	RW	0
9:8	VP_OUT_CTRL	VP_PCLK control. Sets the VP_PCLK as a function of the ISS interconnect interface clock (OCPCLK). 0x0: No division: VP_PCLK = OCPCLK. 0x1: Division by 2: VP_PCLK = OCPCLK / 2. 0x3: Division by 4: VP_PCLK = OCPCLK / 4. 0x2: Division by 3: VP_PCLK = OCPCLK / 3. Example scenarios: - Low VP_PCLK, Memory -> VP: Same as typical memory -> VP, but VP_PCLK = OCPCLK/2 - Typical sensor -> VP: Autoidle enabled, FCLK at optimal rate, sensor provides DPCM compressed RAW12 data at 650 Mbps. Image timings VP_PCLK = (OCPCLK/2) and <a href="#">CCP2_CTRL[31:15] FRACDIV = 0xD000</a> 2600 active pixels/line, 128 blanking pixels, no vertical blanking. (This scenario corresponds to the OTF operation at maximum CCP2 speed.)	RW	0x0
7	DBG_EN	Enables the debug mode. 0x0: Disable 0x1: Enable	RW	0
6:5	BURST_SIZE	Sets the DMA burst size on the L3 interconnect. 0x0: 1x64 OCP writes 0x1: 2x64 OCP writes 0x3: 8x64 OCP writes 0x2: 4x64 OCP writes	RW	0x0
4	ENDIANNESS	Select endianness for YUV4:2:2 8 bit and YUV4:2:0 legacy formats. 0x0: Use native MIPI CSI2 endianness: Little endian for all formats except for YUV4:2:2 8b and YUV4:2:0 Legacy which a big endian. 0x1: Store all pixel formats little endian.	RW	0
3	FRAME	Set the modality in which IF_EN works. 0x0: If IF_EN = 0 the interface is disabled immediately. 0x1: If IF_EN = 1 the interface is disabled after all FEC sync code have been received for the active contexts.	RW	0
2	ECC_EN	Enables the Error Correction Code check for the received header (short and long packets for all virtual channel ids). 0x0: Disabled 0x1: Enabled	RW	0
1	RESERVED	Read returns reset value	RW	0

Bits	Field Name	Description	Type	Reset
0	IF_EN	<p>Enables the physical interface to the module.</p> <p>0x0: The interface is disabled. If FRAME = 0, it is disabled immediately. If FRAME = 1, it is disabled when each context has received the FEC sync code.</p> <p>0x1: The interface is enabled immediately, the data acquisition starts on the next FSC sync code. Writing 1 to this register when the current value is 0 has the effect to clear the output FIFO. The pixel data of the following frame will be written in the PING buffer, that is, the CSI2_CTX_CTRL.PING_PONG bits are reset to 0 as well.</p>	RW	0

**Table 8-162. Register Call Summary for Register CSI2\_CTRL**

## ISS Interfaces

- [ISS CCP2 Programming Model: \[0\] \[1\]](#)
- [ISS CCP2 Register Manual: \[2\]](#)
- [ISS CSI2 Environment: \[3\] \[4\]](#)
- [ISS CSI2 Functional Description: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)
- [ISS CSI2 Programming Model: \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\]](#)
- [ISS CSI2 Register Manual: \[29\] \[30\] \[31\] \[32\] \[33\]](#)

**Table 8-163. CSI2\_DBG\_H**

<b>Address Offset</b>	0x0000 0044
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	<p>DEBUG REGISTER (Header)</p> <p>This register provides a way to debug the CSI2 RECEIVER module with no image sensor connected to the module. The debug mode is enabled by <a href="#">CSI2_CTRL.DBG_EN</a>. Only full 32-bit values shall be written. The register is used to write short packets and header of long packets.</p>
<b>Type</b>	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBG																															

Bits	Field Name	Description	Type	Reset
31:0	DBG	32-bit input value.	W	0x0000 0000

**Table 8-164. Register Call Summary for Register CSI2\_DBG\_H**

## ISS Interfaces

- [ISS CSI2 Programming Model: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [ISS CSI2 Register Manual: \[8\]](#)

**Table 8-165. CSI2\_COMPLEXIO\_CFG**

<b>Address Offset</b>	0x0000 0050
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	COMPLEXIO CONFIGURATION REGISTER for the complex I/O This register contains the lane configuration for the order and position of the lanes (clock and data) and the polarity order for the control of the PHY differential signals in addition to the control bit for the power FSM.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESET_CTRL	RESET_DONE	PWR_CMD	PWR_STATUS	PWR_AUTO	RESERVED						DATA4_POL	DATA4_POSITION	DATA3_POL	DATA3_POSITION	DATA2_POL	DATA2_POSITION	DATA1_POL	DATA1_POSITION	CLOCK_POL	CLOCK_POSITION										

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30	RESET_CTRL	Controls the reset of the complex I/O 0x0: Complex I/O reset active. 0x1: Complex I/O reset deasserted.	RW	0
29	RESET_DONE	Internal reset monitoring of the power domain using the byte clock provided by the associated CSIPHY (see <a href="#">Section 8.1.1, ISS Clock Domains</a> ). Read 0x1: Reset completed. Read 0x0: Internal module reset is on going.	R	0
28:27	PWR_CMD	Command for power control of the complex I/O 0x0: Command to change to OFF state 0x1: Command to change to ON state 0x2: Command to change to Ultralow-Power state	RW	0x0
26:25	PWR_STATUS	Status of the power control of the complex I/O Read 0x2: Complex I/O in Ultralow-Power state Read 0x1: Complex I/O in ON state Read 0x0: Complex I/O in OFF state	R	0x0
24	PWR_AUTO	Automatic switch between ULP and ON states based on ULPM signals from complex I/O 0x0: Disable 0x1: Enable	RW	0
23:20	RESERVED		R	0x0
19	DATA4_POL	+/- differential pin order of data lane 4. 0x0: +/- pin order 0x1: -/+ pin order	RW	0
18:16	DATA4_POSITION	Position and order of the data lane 4. The values 6 and 7 are reserved. 0x1: Data lane 4 is at the position 1. 0x0: Not used/connected 0x2: Data lane 4 is at the position 2. 0x4: Data lane 4 is at the position 4. 0x5: Data lane 4 is at the position 5. 0x3: Data lane 4 is at the position 3.	RW	0x0

Bits	Field Name	Description	Type	Reset
15	DATA3_POL	+/- differential pin order of data lane 3. 0x0: +/- pin order 0x1: -/+ pin order	RW	0
14:12	DATA3_POSITION	Position and order of the data lane 3. The values 6 and 7 are reserved. 0x1: Data lane 3 is at the position 1. 0x0: Not used/connected 0x2: Data lane 3 is at the position 2. 0x4: Data lane 3 is at the position 4. 0x5: Data lane 3 is at the position 5. 0x3: Data lane 3 is at the position 3.	RW	0x0
11	DATA2_POL	+/- differential pin order of DATA lane 2. 0x0: +/- pin order (csi2_dx=+ and csi2_dy=-) 0x1: -/+ pin order (csi2_dx=- and csi2_dy=+)	RW	0
10:8	DATA2_POSITION	Position and order of the data lane 2. The values 6 and 7 are reserved. 0x1: Data lane 2 is at the position 1. 0x0: Not used/connected 0x2: Data lane 2 is at the position 2. 0x4: Data lane 2 is at the position 4. 0x5: Data lane 2 is at the position 5. 0x3: Data lane 2 is at the position 3.	RW	0x0
7	DATA1_POL	+/- differential pin order of data lane 1. 0x0: +/- pin order (csi2_dx=+ and csi2_dy=-) 0x1: -/+ pin order (csi2_dx=- and csi2_dy=+)	RW	0
6:4	DATA1_POSITION	Position and order of the DATA lane 1. 0, 6 and 7 are reserved. The data lane 1 is always present. 0x0: Not used/connected 0x4: Data lane 1 is at the position 4. 0x1: Data lane 1 is at the position 1. 0x5: Data lane 1 is at the position 5. 0x3: Data lane 1 is at the position 3. 0x2: Data lane 1 is at the position 2.	RW	0x0
3	CLOCK_POL	+/- differential pin order of clock lane. 0x0: +/- pin order (csi2_dx=+ and csi2_dy=-) 0x1: -/+ pin order (csi2_dx=- and csi2_dy=+)	RW	0
2:0	CLOCK_POSITION	Position and order of the clock lane. 0, 6 and 7 are reserved. The clock lane is always present. 0x0: Not used/connected 0x4: Reserved 0x1: Clock lane is at the position 1. 0x5: Clock lane is at the position 5. 0x3: Clock lane is at the position 3. 0x2: Clock lane is at the position 2.	RW	0x0

**Table 8-166. Register Call Summary for Register CSI2\_COMPLEXIO\_CFG**

## ISS Interfaces

- [ISS CSI2 PHY Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [ISS CSI2 Functional Description: \[16\] \[17\] \[18\]](#)
- [ISS CSI2 Programming Model: \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [ISS CSI2 Register Manual: \[24\]](#)



**Table 8-167. CSI2\_COMPLEXIO\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0054
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	INTERRUPT STATUS REGISTER - All errors from complex I/O #1
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					STATEALLULPMEXIT	STATEALLULPMENTER	STATEULPM5	STATEULPM4	STATEULPM3	STATEULPM2	STATEULPM1	ERRCONTROL5	ERRCONTROL4	ERRCONTROL3	ERRCONTROL2	ERRCONTROL1	ERRESC5	ERRESC4	ERRESC3	ERRESC2	ERRESC1	ERRSOTSYNCHS5	ERRSOTSYNCHS4	ERRSOTSYNCHS3	ERRSOTSYNCHS2	ERRSOTSYNCHS1	ERRSOTHS5	ERRSOTHS4	ERRSOTHS3	ERRSOTHS2	ERRSOTHS1

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	STATEALLULPMEXIT	At least one of the active lanes has exit the ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
25	STATEALLULPMENTER	All active lanes are entering in ULPM. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
24	STATEULPM5	Lane 5 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
23	STATEULPM4	Lane 4 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
22	STATEULPM3	Lane 3 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
21	STATEULPM2	Lane 2 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
20	STATEULPM1	Lane 1 in ULPM 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
19	ERRCONTROL5	Control error for lane 5 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
18	ERRCONTROL4	Control error for lane 4 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
17	ERRCONTROL3	Control error for lane 3 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
16	ERRCONTROL2	Control error for lane 2 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
15	ERRCONTROL1	Control error for lane 1 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
14	ERRESC5	Escape entry error for lane 5 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
13	ERRESC4	Escape entry error for lane 4 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
12	ERRESC3	Escape entry error for lane 3 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
11	ERRESC2	Escape entry error for lane 2 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
10	ERRESC1	Escape entry error for lane 1 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
9	ERRSOTSYNCHS5	Start of transmission sync error for lane 5 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
8	ERRSOTSYNCHS4	Start of transmission sync error for lane 4 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
7	ERRSOTSYNCHS3	Start of transmission sync error for lane 3 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
6	ERRSOTSYNCHS2	Start of transmission sync error for lane 2 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
5	ERRSOTSYNCHS1	Start of transmission sync error for lane 1 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
4	ERRSOTHS5	Start of transmission error for lane 5 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
3	ERRSOTHS4	Start of transmission error for lane 4 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
2	ERRSOTHS3	Start of transmission error for lane 3 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
1	ERRSOTHS2	Start of transmission error for lane 2 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
0	ERRSOTHS1	Start of transmission error for lane 1 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

**Table 8-168. Register Call Summary for Register CSI2\_COMPLEXIO\_IRQSTATUS**

## ISS Interfaces

- [ISS CSI2 PHY Functional Description: \[0\]](#)
- [ISS CSI2 Programming Model: \[1\]](#)
- [ISS CSI2 Register Manual: \[2\] \[3\] \[4\]](#)

**Table 8-169. CSI2\_SHORT\_PACKET**

<b>Address Offset</b>	0x0000 005C
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	SHORT PACKET INFORMATION - This register sets the 24-bit DATA_ID + Short Packet Data Field when the data type is between 0x8 and 0xF
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SHORT_PACKET																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads returns 0.	R	0x00
23:0	SHORT_PACKET	Short Packet information: DATA ID + DATA FIELD	R	0x000000

**Table 8-170. Register Call Summary for Register CSI2\_SHORT\_PACKET**

ISS Interfaces

- [ISS CSI2 Functional Description: \[0\] \[1\]](#)
- [ISS CSI2 Register Manual: \[2\]](#)

**Table 8-171. CSI2\_COMPLEXIO\_IRQENABLE**

<b>Address Offset</b>	0x0000 0060
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	INTERRUPT ENABLE REGISTER - All errors from complex I/O
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RESERVED								STATEALLULPMEXIT	STATEALLULPMENTER	STATEULPM5	STATEULPM4	STATEULPM3	STATEULPM2	STATEULPM1	ERRCONTROL5	ERRCONTROL4	ERRCONTROL3	ERRCONTROL2	ERRCONTROL1	ERRESC5	ERRESC4	ERRESC3	ERRESC2	ERRESC1	ERRSOTSYNCHS5	ERRSOTSYNCHS4	ERRSOTSYNCHS3	ERRSOTSYNCHS2	ERRSOTSYNCHS1	ERRSOTHS5	ERRSOTHS4	ERRSOTHS3	ERRSOTHS2	ERRSOTHS1

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	STATEALLULPMEXIT	At least one of the active lanes has exit the ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
25	STATEALLULPMENTER	All active lanes are entering in ULPM. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
24	STATEULPM5	Lane 5 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
23	STATEULPM4	Lane 4 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
22	STATEULPM3	Lane 3 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
21	STATEULPM2	Lane 2 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
20	STATEULPM1	Lane 1 in ULPM 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
19	ERRCONTROL5	Control error for lane 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
18	ERRCONTROL4	Control error for lane 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
17	ERRCONTROL3	Control error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
16	ERRCONTROL2	Control error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
15	ERRCONTROL1	Control error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
14	ERRESC5	Escape entry error for lane 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
13	ERRESC4	Escape entry error for lane 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
12	ERRESC3	Escape entry error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
11	ERRESC2	Escape entry error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
10	ERRESC1	Escape entry error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
9	ERRSOTSYNCHS5	Start of transmission sync error for lane 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
8	ERRSOTSYNCHS4	Start of transmission sync error for lane 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	ERRSOTSYNCHS3	Start of transmission sync error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
6	ERRSOTSYNCHS2	Start of transmission sync error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
5	ERRSOTSYNCHS1	Start of transmission sync error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	ERRSOTHS5	Start of transmission error for lane 5 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
3	ERRSOTHS4	Start of transmission error for lane 4 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
2	ERRSOTHS3	Start of transmission error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
1	ERRSOTHS2	Start of transmission error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	ERRSOTHS1	Start of transmission error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

**Table 8-172. Register Call Summary for Register CSI2\_COMPLEXIO\_IRQENABLE**

ISS Interfaces

- [ISS CSI2 Programming Model: \[0\]](#)
- [ISS CSI2 Register Manual: \[1\]](#)

**Table 8-173. CSI2\_DBG\_P**

<b>Address Offset</b>	0x0000 0068																																																																
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .																																																																
<b>Description</b>	DEBUG REGISTER (Payload) This register provides a way to debug the CSI2 RECEIVER module with no image sensor connected to the module. The debug mode is enabled by <a href="#">CSI2_CTRL.DBG_EN</a> . Only full 32-bit values shall be written. The register is used to write payload of long packets.																																																																
<b>Type</b>	W																																																																
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">DBG</td> </tr> </table>		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DBG																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																		
DBG																																																																	
<b>Bits</b>	31:0																																																																
<b>Field Name</b>	DBG																																																																
<b>Description</b>	32-bit input value.																																																																
<b>Type</b>	W																																																																
<b>Reset</b>	0x0000 0000																																																																

**Table 8-174. Register Call Summary for Register CSI2\_DBG\_P**

ISS Interfaces

- [ISS CSI2 Programming Model: \[0\] \[1\]](#)
- [ISS CSI2 Register Manual: \[2\]](#)

**Table 8-175. CSI2\_TIMING**

<b>Address Offset</b>	0x0000 006C
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .
<b>Description</b>	TIMING REGISTER This register controls the CSI2 RECEIVER module. This register shall not be modified while <a href="#">CSI2_CTRL.IF_EN</a> is set to 1. It is used to indicate the number of L3 cycles for the Stop State monitoring.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED													FORCE_RX_MODE_IO1	STOP_STATE_X16_IO1	STOP_STATE_X4_IO1	STOP_STATE_COUNTER_IO1												

Bits	Field Name	Description	Type	Reset
31	RESERVED	Read returns reset value	RW	0
30	RESERVED	Read returns reset value	RW	1
29	RESERVED	Read returns reset value	RW	1
28:16	RESERVED	Read returns reset value	RW	0x1FFF
15	FORCE_RX_MODE_IO1	Control of ForceRxMode signal  0x0: Deassertion of ForceRxMode. The hardware reset the bit at the end of the Force RX Mode assertion. The software can reset the bit in order to stop the assertion of the ForceRXMode signal prior to the completion of the period. 0x1: Assertion of ForceRxMode	RW	0
14	STOP_STATE_X16_IO1	Multiplication factor for the number of L3 cycles defined in STOP_STATE_COUNTER bit field  0x0: The number of L3 cycles defined in STOP_STATE_COUNTER is multiplied by 1x 0x1: The number of L3 cycles defined in STOP_STATE_COUNTER is multiplied by 16x	RW	1
13	STOP_STATE_X4_IO1	Multiplication factor for the number of L3 cycles defined in STOP_STATE_COUNTER bit field  0x0: The number of L3 cycles defined in STOP_STATE_COUNTER is multiplied by 1x 0x1: The number of L3 cycles defined in STOP_STATE_COUNTER is multiplied by 4x	RW	1
12:0	STOP_STATE_COUNTER_IO1	Stop State counter for monitoring. It indicates the number of L3 to monitor for Stop State before deasserting ForceRxMode (complex I/O 1). The value is from 0 to 8191.	RW	0x1FFF

**Table 8-176. Register Call Summary for Register CSI2\_TIMING**

ISS Interfaces

- [ISS CSI2 PHY Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [ISS CSI2 Programming Model: \[13\]](#)
- [ISS CSI2 Register Manual: \[14\]](#)

**Table 8-177. CSI2\_CTX\_CTRL1\_i**

<b>Address Offset</b>	0x0000 0070 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .		
<b>Description</b>	CONTROL REGISTER - Context This register controls the Context. This register is shadowed: modifications are taken into account after the next FSC sync code.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYTESWAP	GENERIC	RESERVED	HSCALE	TRANSCODE				FEC_NUMBER								COUNT								EOF_EN	EOL_EN	CS_EN	COUNT_UNLOCK	PING_PONG	VP_FORCE	LINE_MODULO	CTX_EN

Bits	Field Name	Description	Type	Reset
31	BYTESWAP	Allows swapping bytes two by two in the payload data. It does not affect: - short packets - long packet header or footers - CRC calculation The purpose is to by swap data send to the OCP port and/or video port 0x0: Disabled 0x1: Enabled	RW	0
30	GENERIC	Enables the generic mode. 0x0: Disabled. Data is received according to CSI2_CTX_CTRL1.FORMAT and the long packet code transmitted in the MIPI stream is used. 0x1: Enabled. Data is received according to CSI2_CTX_CTRL1.FORMAT and the long packet code transmitted in the MIPI stream is ignored.	RW	0
29	RESERVED		R	0x0
28	HSCALE	Enable horizontal downscaling by a factor of two. Applies to RAW data when transcoding is enabled. 0x0: Disable 0x1: Enable	RW	0x0



Bits	Field Name	Description	Type	Reset
27:24	TRANSCODE	<p>Enables image transcoding. When this features is enabled:</p> <ul style="list-style-type: none"> <li>- the data format from the camera is defined by the FORMAT register</li> <li>- the format after transcode is defined by the TRANSCODE register. The memory storage / video port formats is defined by the TRANSCODE register</li> </ul> <p>0x6: Outputs uncompressed RAW10 data. Data in memory is organized as regular packed RAW10 data</p> <p>0x1: Outputs DPCM compressed RAW10 data. After compression, pixels are coded on 8 bits. Data in memory is organized as regular RAW8 data</p> <p>0x7: Outputs uncompressed RAW12 data. Data in memory is organized as regular RAW12+EXP16 data</p> <p>0x0: Feature disabled.</p> <p>0x2: Outputs DPCM compressed RAW12 data. After compression, pixels are coded on 8 bits. Data in memory is organized as regular RAW8 data</p> <p>0x8: Outputs uncompressed RAW12 data. Data in memory is organized as regular packed RAW12 data</p> <p>0x9: Outputs uncompressed RAW14 data.</p> <p>0x4: Outputs uncompressed RAW8 data. Data in memory is organized as regular RAW8 data</p> <p>0x5: Outputs uncompressed RAW10 data. Data in memory is organized as regular RAW10+EXP16 data</p> <p>0x3: Outputs A-Law compressed RAW10 data. After compression, pixels are coded on 8 bits. Data in memory is organized as regular RAW8 data.</p>	RW	0x0
23:16	FEC_NUMBER	<p>Number of FEC to receive between using swap of CSI2_CTX_DAT_PING_ADDR and CSI2_CTX_DAT_PONG_ADDR for the calculation of the address in memory (must be used only in interlace mode, otherwise set to 1).</p>	RW	0x01
15:8	COUNT	<p>Sets the number of frame to acquire. Once the frame acquisition starts, the COUNT value is decremented after every frame. When COUNT reaches 0, the FRAME_NUMBER_IRQ interrupt is triggered and CTX_EN is set to 0.</p> <p>Writes to this bit field are controlled by the COUNT_UNLOCK bit. During the same OCP write access, the bit field COUNT_UNLOCK shall be written in addition to COUNT bit field in order to change the COUNT value. COUNT can be overwritten dynamically with a new count value.</p> <p>0: Infinite number of frames (no count). 1: 1 frame to acquire ... 255: 255 frames to acquire</p>	RW	0x00
7	EOF_EN	<p>Indicates if the end of frame signal shall be asserted at the end of the line.</p> <p>Read 0x1: The end of frame signal is asserted at the end of each frame.</p> <p>Read 0x0: The end of frame signal is not asserted at the end of each frame.</p>	RW	0

Bits	Field Name	Description	Type	Reset
6	EOL_EN	Indicates if the end of line signal shall be asserted at the end of the line.  Read 0x1: The end of line signal is asserted at the end of each frame.  Read 0x0: The end of line signal is not asserted at the end of each frame.	RW	0
5	CS_EN	Enables the checksum check for the received payload (long packet only).  0x0: Disabled 0x1: Enabled	RW	0
4	COUNT_UNLOCK	Unlock writes to the COUNT bit field.  Write 0x0: COUNT bit field is locked. Writes have no effect  Write 0x1: COUNT bit field is unlocked. Writes are possible.	W	0
3	PING_PONG	Indicates whether the PING or PONG destination address (CSI2_CTX_DAT_PING_ADDR or CSI2_CTX_DAT_PONG_ADDR) was used to write the last frame.  This bit field toggles after every FEC_NUMBER FEC sync code received for the current context.  Read 0x1: PONG buffer Read 0x0: PING buffer	R	1
2	VP_FORCE	Forces sending of the data to both VPORT and OCP. Only applies to formats that existing in two versions: - One sending data to OCP port only - One sending data to VPORT only (tagged with the +VP extension)  The format version sending data only to OCP should be chosen.  0x0: Disabled 0x1: Enabled	RW	0
1	LINE_MODULO	Line modulo configuration  0x0: CSI2_CTX_CTRL3.LINE_NUMBER is used once per frame for the generation of the LINE_NUMBER_IRQ.  0x1: CSI2_CTX_CTRL3.LINE_NUMBER is used as a modulo number for the generation of the LINE_NUMBER_IRQ (multiple times the interrupt can be generated for each frame)	RW	0
0	CTX_EN	Enables the context  0x0: Disabled 0x1: Enabled	RW	0

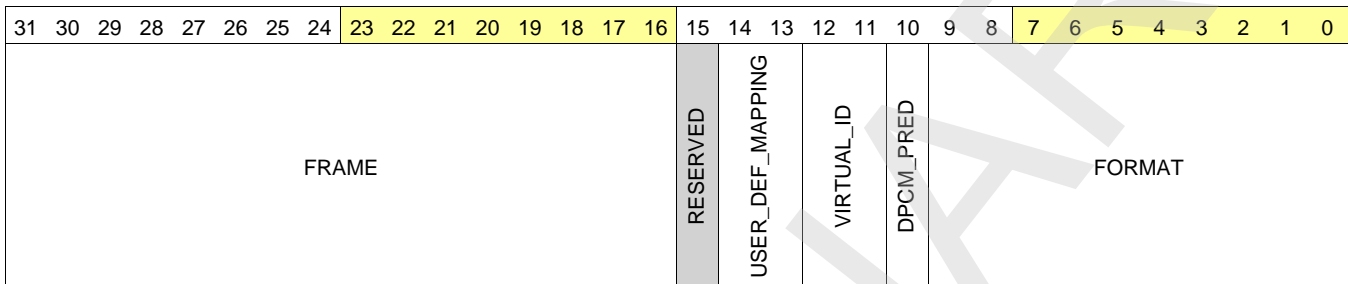
**Table 8-178. Register Call Summary for Register CSI2\_CTX\_CTRL1\_i**

## ISS Interfaces

- [ISS CS12 Environment: \[0\] \[1\]](#)
- [ISS CS12 Functional Description: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)
- [ISS CS12 Programming Model: \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\]](#)
- [ISS CS12 Register Manual: \[31\]](#)

**Table 8-179. CSI2\_CTX\_CTRL2\_i**

<b>Address Offset</b>	0x0000 0074 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .		
<b>Description</b>	CONTROL REGISTER - Context This register controls the Context. This register is shadowed: modifications are taken into account after the next FSC sync code (except for VIRTUAL_ID and FORMAT fields). The change of VIRTUAL_ID and FORMAT has to occur only when the context is disabled (CSI2_CTX_CTRL1.CTX_EN).		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:16	FRAME	Frame number received	R	0x0000
15	RESERVED		R	0
14:13	USER_DEF_MAPPING	Selects the pixel format of USER_DEFINED in FORMAT 0x0: RAW6 0x1: RAW7 0x2: RAW8 (not valid if FORMAT is USER_DEFINED_8_BIT_DATA_TYPE_x_EXP8 with x from 1 to 8)	RW	0x0
12:11	VIRTUAL_ID	Virtual channel ID 0x0: Virtual Channel ID 0 0x1: Virtual Channel ID 1 0x3: Virtual Channel ID 3 0x2: Virtual Channel ID 2	RW	0x0
10	DPCM_PRED	Selects the DPCM predictor. 0x0: The advanced predictor is used. Not supported for 10- 8- 10 algorithm. Performance limited to 1 pixel/cycle. 0x1: The simple predictor is used.	RW	0
9:0	FORMAT	Data format selection. 0x3A9: RAW7 DPCM12 + VP 0x80: USER_DEFINED_8_BIT_DATA_TYPE_1 + EXP8 0x84: USER_DEFINED_8_BIT_DATA_TYPE_5 + EXP8 0x346: USER_DEFINED_8_BIT_DATA_TYPE_7 + DPCM10 + VP 0x1D: YUV4:2:0 10bit + CSPS 0x2C6: USER_DEFINED_8_BIT_DATA_TYPE_7 + DPCM10 + EXP16 0x41: USER_DEFINED_8_BIT_DATA_TYPE_2 0x1C5: USER_DEFINED_8_BIT_DATA_TYPE_6_DPCM12_EXP 16 0xAC: RAW12 + EXP16 0x145: USER_DEFINED_8_BIT_DATA_TYPE_6_DPCM12_VP	RW	0x000

Bits	Field Name	Description	Type	Reset
		0x12: Embedded 8-bit nonimage data (e.g. JPEG)		
		0x46: USER_DEFINED_8_BIT_DATA_TYPE_7		
		0x83: USER_DEFINED_8_BIT_DATA_TYPE_4 + EXP8		
		0x2B: RAW10		
		0x9E: YUV4:2:2 8bit + VP		
		0x1C0: USER_DEFINED_8_BIT_DATA_TYPE_1_DPCM12_EXP 16		
		0x140: USER_DEFINED_8_BIT_DATA_TYPE_1_DPCM12_VP		
		0x329: RAW7 + DPCM10 + VP		
		0x22: RGB565		
		0x68: RAW6 + EXP8		
		0x43: USER_DEFINED_8_BIT_DATA_TYPE_4		
		0xE3: RGB666 + EXP32		
		0x1C7: USER_DEFINED_8_BIT_DATA_TYPE_8_DPCM12_EXP 16		
		0x143: USER_DEFINED_8_BIT_DATA_TYPE_4_DPCM12_VP		
		0xA0: RGB444 + EXP16		
		0x87: USER_DEFINED_8_BIT_DATA_TYPE_8 + EXP8		
		0x85: USER_DEFINED_8_BIT_DATA_TYPE_6 + EXP8		
		0x0: OTHERS (except NULL and BLANKING packets)		
		0x341: USER_DEFINED_8_BIT_DATA_TYPE_2 + DPCM10 + VP		
		0xAD: RAW14 + EXP16		
		0x2C3: USER_DEFINED_8_BIT_DATA_TYPE_4 + DPCM10 + EXP16		
		0xE4: RGB888 + EXP32		
		0x343: USER_DEFINED_8_BIT_DATA_TYPE_4 + DPCM10 + VP		
		0x368: RAW6 DPCM12 + VP		
		0x12C: RAW12 + VP		
		0x42: USER_DEFINED_8_BIT_DATA_TYPE_3		
		0xA1: RGB555 + EXP16		
		0x12D: RAW14 + VP		
		0x144: USER_DEFINED_8_BIT_DATA_TYPE_5_DPCM12_VP		
		0x1C4: USER_DEFINED_8_BIT_DATA_TYPE_5_DPCM12_EXP 16		
		0x2A: RAW8		
		0x3AA: RAW8 DPCM12 + VP		
		0x2C1: USER_DEFINED_8_BIT_DATA_TYPE_2 + DPCM10 + EXP16		
		0x1C3: USER_DEFINED_8_BIT_DATA_TYPE_4_DPCM12_EXP 16		
		0x342: USER_DEFINED_8_BIT_DATA_TYPE_3 + DPCM10 + VP		
		0x142: USER_DEFINED_8_BIT_DATA_TYPE_3_DPCM12_VP		
		0x1A: YUV4:2:0 8bit legacy		

Bits	Field Name	Description	Type	Reset
		0x36A: RAW8 DPCM12 + EXP16		
		0x32A: RAW8 + DPCM10 + VP		
		0x229: RAW7 + DPCM10 + EXP16		
		0x12A: RAW8 + VP		
		0x28: RAW6		
		0x1C1: USER_DEFINED_8_BIT_DATA_TYPE_2_DPCM12_EXP 16		
		0x2A8: RAW6 + DPCM10 + EXP16		
		0x45: USER_DEFINED_8_BIT_DATA_TYPE_6		
		0xDE: Same as YUV4:2:2 8bit + VP but data is send as 16-bit wide words to video port. Could be used together with the GENERIC and BYTESWAP features.		
		0x340: USER_DEFINED_8_BIT_DATA_TYPE_1 + DPCM10 + VP		
		0x40: USER_DEFINED_8_BIT_DATA_TYPE_1		
		0x19: YUV4:2:0 10 bit		
		0x344: USER_DEFINED_8_BIT_DATA_TYPE_5 + DPCM10 + VP		
		0x2C2: USER_DEFINED_8_BIT_DATA_TYPE_3 + DPCM10 + EXP16		
		0x141: USER_DEFINED_8_BIT_DATA_TYPE_2_DPCM12_VP		
		0x369: RAW7 DPCM12 + EXP16		
		0x69: RAW7 + EXP8		
		0x2C4: USER_DEFINED_8_BIT_DATA_TYPE_5 + DPCM10 + EXP16		
		0x1C: YUV4:2:0 8 bit + CSPS		
		0x347: USER_DEFINED_8_BIT_DATA_TYPE_8 + DPCM10 + VP		
		0x146: USER_DEFINED_8_BIT_DATA_TYPE_7_DPCM12_VP		
		0x82: USER_DEFINED_8_BIT_DATA_TYPE_3 + EXP8		
		0x47: USER_DEFINED_8_BIT_DATA_TYPE_8		
		0x2AA: RAW8 + DPCM10 + EXP16		
		0x1E: YUV4:2:2 8 bit		
		0x33: RGB666 + EXP32_24		
		0x2C: RAW12		
		0x2C7: USER_DEFINED_8_BIT_DATA_TYPE_8 + DPCM10 + EXP16		
		0x147: USER_DEFINED_8_BIT_DATA_TYPE_8_DPCM12_VP		
		0x81: USER_DEFINED_8_BIT_DATA_TYPE_2 + EXP8		
		0x24: RGB888		
		0x2D: RAW14		
		0x345: USER_DEFINED_8_BIT_DATA_TYPE_6 + DPCM10 + VP		
		0xAB: RAW10 + EXP16		
		0x3A8: RAW6 DPCM12 + EXP16		
		0x44: USER_DEFINED_8_BIT_DATA_TYPE_5		
		0x2C0: USER_DEFINED_8_BIT_DATA_TYPE_1 + DPCM10 + EXP16		
		0x18: YUV4:2:0 8 bit		

Bits	Field Name	Description	Type	Reset
		0x12F: RAW10 + VP		
		0xE8: RAW6 + DPCM10 + VP		
		0x1F: YUV4:2:2 10 bit		
		0x2C5: USER_DEFINED_8_BIT_DATA_TYPE_6 + DPCM10 + EXP16		
		0x1C2: USER_DEFINED_8_BIT_DATA_TYPE_3_DPCM12_EXP16		
		0x1C6: USER_DEFINED_8_BIT_DATA_TYPE_7_DPCM12_EXP16		
		0x29: RAW7		
		0x86: USER_DEFINED_8_BIT_DATA_TYPE_7 + EXP8		

**Table 8-180. Register Call Summary for Register CSI2\_CTX\_CTRL2\_i**

## ISS Interfaces

- ISS CSI2 Environment: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21]
- ISS CSI2 Functional Description: [22] [23] [24] [25] [26] [27] [28] [29]
- ISS CSI2 Programming Model: [30] [31] [32]
- ISS CSI2 Register Manual: [33]

**Table 8-181. CSI2\_CTX\_DAT\_OFST\_i**

Address Offset	0x0000 0078 + (0x20 * i)	Index	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .		
<b>Description</b>	DATA MEM ADDRESS OFFSET REGISTER - Context This register sets the offset, which is applied on the destination address after each line is written to memory. This register applies for both CSI2_CTX_DAT_PING_ADDR and CSI2_CTX_DAT_PONG_ADDR. For example, it enables to perform 2D data transfers of the pixel data into a frame buffer. In such case, the pixel data and frame buffer data shall have the same data format. The 5 LSBs are ignored: the offset shall be a multiple of 32 bytes. This register is shadowed: modifications are taken into account after the next FSC sync code. Only full 32-bit values shall be written.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OFST								RESERVED															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:5	OFST	Line offset programmed in bytes (signed value 2s complement). If OFST = 0, the data is written contiguously in memory. Otherwise, OFST sets the destination offset between the first pixel of the previous line and the first pixel of the current line.	RW	0x000
4:0	RESERVED		R	0x00

**Table 8-182. Register Call Summary for Register CSI2\_CTX\_DAT\_OFST\_i**

## ISS Interfaces

- ISS CSI2 Environment: [0] [1]
- ISS CSI2 Functional Description: [2] [3] [4]
- ISS CSI2 Programming Model: [5]
- ISS CSI2 Register Manual: [6]

**Table 8-183. CSI2\_CTX\_DAT\_PING\_ADDR\_i**

<b>Address Offset</b>	0x0000 007C + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .		
<b>Description</b>	<p>DATA MEM PING ADDRESS REGISTER - Context</p> <p>This register sets the 32-bit memory address where the pixel data are stored. The destination is double buffered: this register sets the PING address. Double buffering is enabled when the addresses CSI2_CTX_DAT_PING_ADDR and CSI2_CTX_DAT_PONG_ADDR are different.</p> <p>The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary.</p> <p>This register is shadowed: modifications are taken into account after the next FSC sync code. Only full 32-bit values shall be written.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most-significant bits of the 32-bit address.	RW	0x00000000
4:0	RESERVED		R	0x00

**Table 8-184. Register Call Summary for Register CSI2\_CTX\_DAT\_PING\_ADDR\_i**

ISS Interfaces

- [ISS CSI2 Functional Description: \[0\] \[1\]](#)
- [ISS CSI2 Programming Model: \[2\] \[3\]](#)
- [ISS CSI2 Register Manual: \[4\]](#)

**Table 8-185. CSI2\_CTX\_DAT\_PONG\_ADDR\_i**

<b>Address Offset</b>	0x0000 0080 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .		
<b>Description</b>	<p>DATA MEM PONG ADDRESS REGISTER - Context</p> <p>This register sets the 32-bit memory address where the pixel data are stored. The destination is double-buffered: this register sets the PONG address. Double-buffering is enabled when the addresses CSI2_CTX_DAT_PING_ADDR and CSI2_CTX_DAT_PONG_ADDR are different.</p> <p>The 5 LSBs are ignored: the address shall be aligned on a 32-byte boundary.</p> <p>This register is shadowed: modifications are taken into account after the next FSC sync code. Only full 32-bit values shall be written.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	27 most-significant bits of the 32-bit address.	RW	0x00000000
4:0	RESERVED		R	0x00

**Table 8-186. Register Call Summary for Register CSI2\_CTX\_DAT\_PONG\_ADDR\_i**

ISS Interfaces

- [ISS CSI2 Functional Description: \[0\] \[1\]](#)
- [ISS CSI2 Register Manual: \[2\]](#)

**Table 8-187. CSI2\_CTX\_IRQENABLE\_i**

<b>Address Offset</b>	0x0000 0084 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .		
<b>Description</b>	INTERRUPT ENABLE REGISTER - Context This register regroups all the events related to context.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_CORRECTION_IRQ	LINE_NUMBER_IRQ	FRAME_NUMBER_IRQ	CS_IRQ	RESERVED	LE_IRQ	LS_IRQ	FE_IRQ	FS_IRQ							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	ECC_CORRECTION_IRQ	Context - ECC has been used to correct the only 1-bit error (long packet only). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	LINE_NUMBER_IRQ	Context - Line number is reached. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
6	FRAME_NUMBER_IRQ	Context - Frame counter reached. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
5	CS_IRQ	Context - Check-Sum of the payload mismatch detection 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	RESERVED		R	0
3	LE_IRQ	Context - Line end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
2	LS_IRQ	Context - Line start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
1	FE_IRQ	Context - Frame end sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	FS_IRQ	Context - Frame start sync code detection. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

**Table 8-188. Register Call Summary for Register CSI2\_CTX\_IRQENABLE\_i**

ISS Interfaces

- [ISS CSI2 Functional Description: \[0\] \[1\]](#)
- [ISS CSI2 Register Manual: \[2\]](#)



**Table 8-189. CSI2\_CTX\_IRQSTATUS\_i**

<b>Address Offset</b>	0x0000 0088 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .		
<b>Description</b>	INTERRUPT STATUS REGISTER - Context This register regroups all the events related to Context.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_CORRECTION_IRQ	LINE_NUMBER_IRQ	FRAME_NUMBER_IRQ	CS_IRQ	RESERVED	LE_IRQ	LS_IRQ	FE_IRQ	FS_IRQ							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	ECC_CORRECTION_IRQ	Context - ECC has been used to do the correction of the only 1-bit error status (long packet only). 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
7	LINE_NUMBER_IRQ	Context - Line number reached status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
6	FRAME_NUMBER_IRQ	Context - Frame counter reached status 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
5	CS_IRQ	Context - Check-Sum mismatch status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
4	RESERVED		R	0
3	LE_IRQ	Context - Line end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
2	LS_IRQ	Context - Line start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
1	FE_IRQ	Context - Frame end sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
0	FS_IRQ	Context - Frame start sync code detection status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

**Table 8-190. Register Call Summary for Register CSI2\_CTX\_IRQSTATUS\_i**

ISS Interfaces

- [ISS CSI2 Functional Description: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS CSI2 Programming Model: \[4\]](#)
- [ISS CSI2 Register Manual: \[5\]](#)

**Table 8-191. CSI2\_CTX\_CTRL3\_i**

<b>Address Offset</b>	0x0000 008C + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-150</a> .		
<b>Description</b>	CONTROL REGISTER - Context This register controls the Context. This register is shadowed: modifications are taken into account after the next FSC sync code.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ALPHA								LINE_NUMBER															

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	ALPHA	Alpha value for RGB888, RGB666 and RBG444.	RW	0x0000
15:0	LINE_NUMBER	Line number for the interrupt generation	RW	0x0000

**Table 8-192. Register Call Summary for Register CSI2\_CTX\_CTRL3\_i**

ISS Interfaces

- [ISS CSI2 Environment: \[0\] \[1\] \[2\]](#)
- [ISS CSI2 Functional Description: \[3\]](#)
- [ISS CSI2 Programming Model: \[4\] \[5\]](#)
- [ISS CSI2 Register Manual: \[6\]](#)

### 8.2.5.5.3 ISS CSI2 REGS2 Registers

#### 8.2.5.5.3.1 ISS CSI2 REGS2 Register Summary

[Table 8-193](#) summarizes the CSI2 REGS2 registers.

**Table 8-193. ISS CSI2 REGS2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CSI2_A_REGS 2 Base Address	ISS_CSI2_B_REGS 2 Base Address
CSI2_CTX_TRANS CODEH <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0000 + (0x8 * i)	0x5200 11C0 + (0x8 * i)	0x5200 15C0 + (0x8 * i)
CSI2_CTX_TRANS CODEV <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0004 + (0x8 * i)	0x5200 11C4 + (0x8 * i)	0x5200 15C4 + (0x8 * i)

<sup>(1)</sup> i = 0 to 7

**8.2.5.5.3.2 ISS CSI2 REGS2 Register Description**

Table 8-194 and Table 8-196 describe the CSI2 REGS2 registers.

**Table 8-194. CSI2\_CTX\_TRANSCODEH<sub>i</sub>**

<b>Address Offset</b>	0x0000 0000 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See Table 8-193.		
<b>Description</b>	Transcode configuration register: defines horizontal frame cropping		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HCOUNT								RESERVED								HSKIP							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	HCOUNT	Pixels to output per line when the values is between 1 and 8191. Pixels HSKIP-WIDTH pixels are output when HCOUNT=0. WIDTH corresponds to the image width provided by the sensor.	RW	0x0000
15:13	RESERVED		R	0x0
12:0	HSKIP	Pixel to skip horizontally. Valid values: 0-8191	RW	0x0000

**Table 8-195. Register Call Summary for Register CSI2\_CTX\_TRANSCODEH<sub>i</sub>**

ISS Interfaces

- ISS CSI2 Functional Description: [0] [1] [2] [3]
- ISS CSI2 Register Manual: [4]

**Table 8-196. CSI2\_CTX\_TRANSCODEV<sub>i</sub>**

<b>Address Offset</b>	0x0000 0004 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See Table 8-193.		
<b>Description</b>	Transcode configuration register: defines vertical frame cropping		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VCOUNT								RESERVED								VSKIP							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	VCOUNT	Lines to output per frame when the values is between 1 and 8191. Pixels VSKIP-HEIGHT pixels are output when VCOUNT=0. HEIGHT corresponds to the image height provided by the sensor.	RW	0x0000
15:13	RESERVED		R	0x0
12:0	VSKIP	Pixel to skip vertically Valid values: 0-8191	RW	0x0000

**Table 8-197. Register Call Summary for Register CSI2\_CTX\_TRANSCODEV\_i**

ISS Interfaces

- [ISS CSI2 Functional Description: \[0\] \[1\] \[2\]](#)
- [ISS CSI2 Register Manual: \[3\]](#)

## 8.2.6 ISS TCTRL

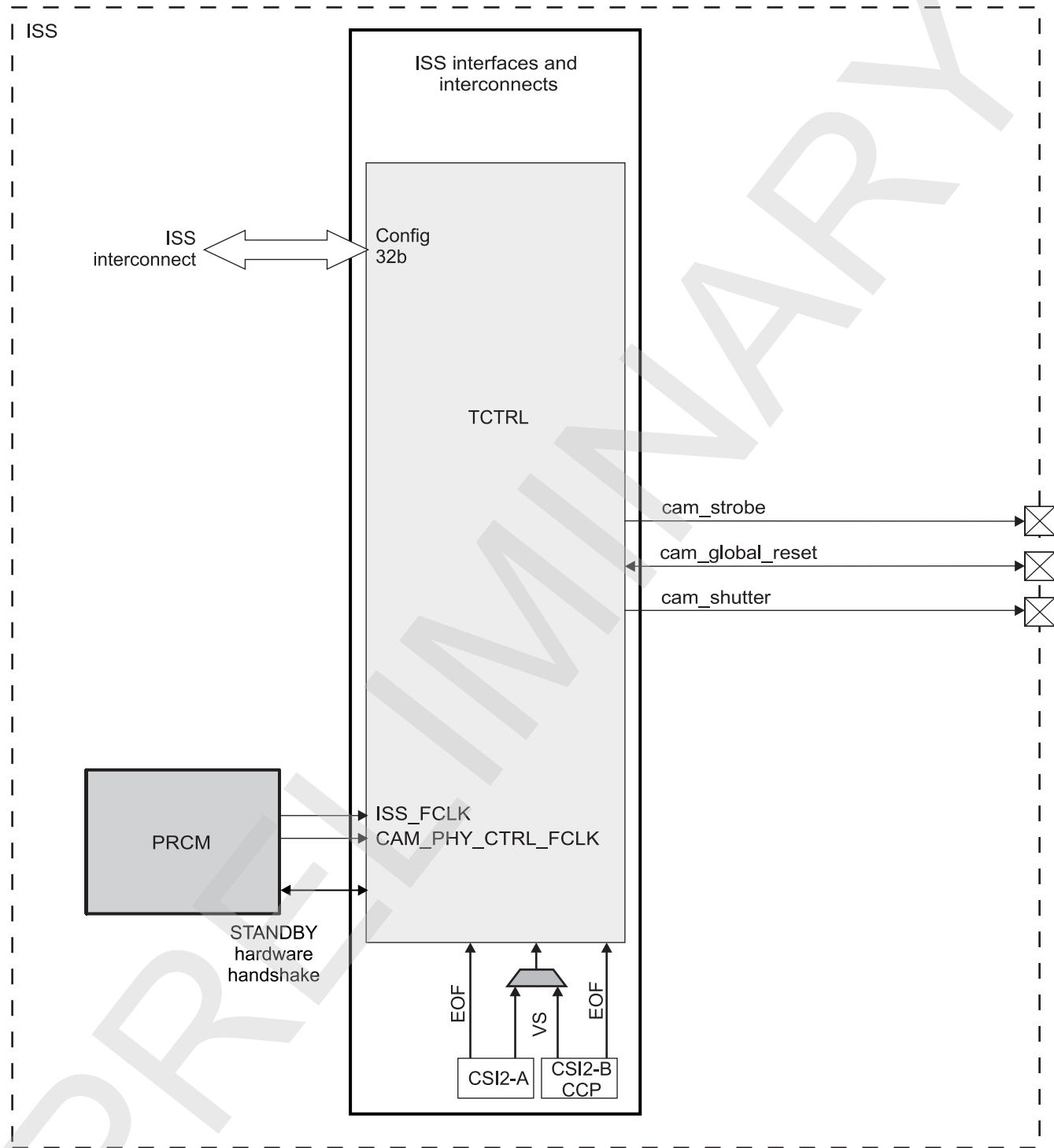
### 8.2.6.1 ISS TCTRL Environment

There are no particular environment attributes. See [Section 8.2.2, ISS Interfaces Environment](#).

### 8.2.6.2 ISS TCTRL Integration

[Figure 8-72](#) shows the integration of the TCTRL.

Figure 8-72. ISS TCTRL Integration



camss-355

For information about the power domain, clocks, reset, and hardware requests, see [Section 8.1.2.4, ISS Power Management](#).

### 8.2.6.3 ISS TCTRL Functional Description

#### 8.2.6.3.1 ISS TCTRL Features

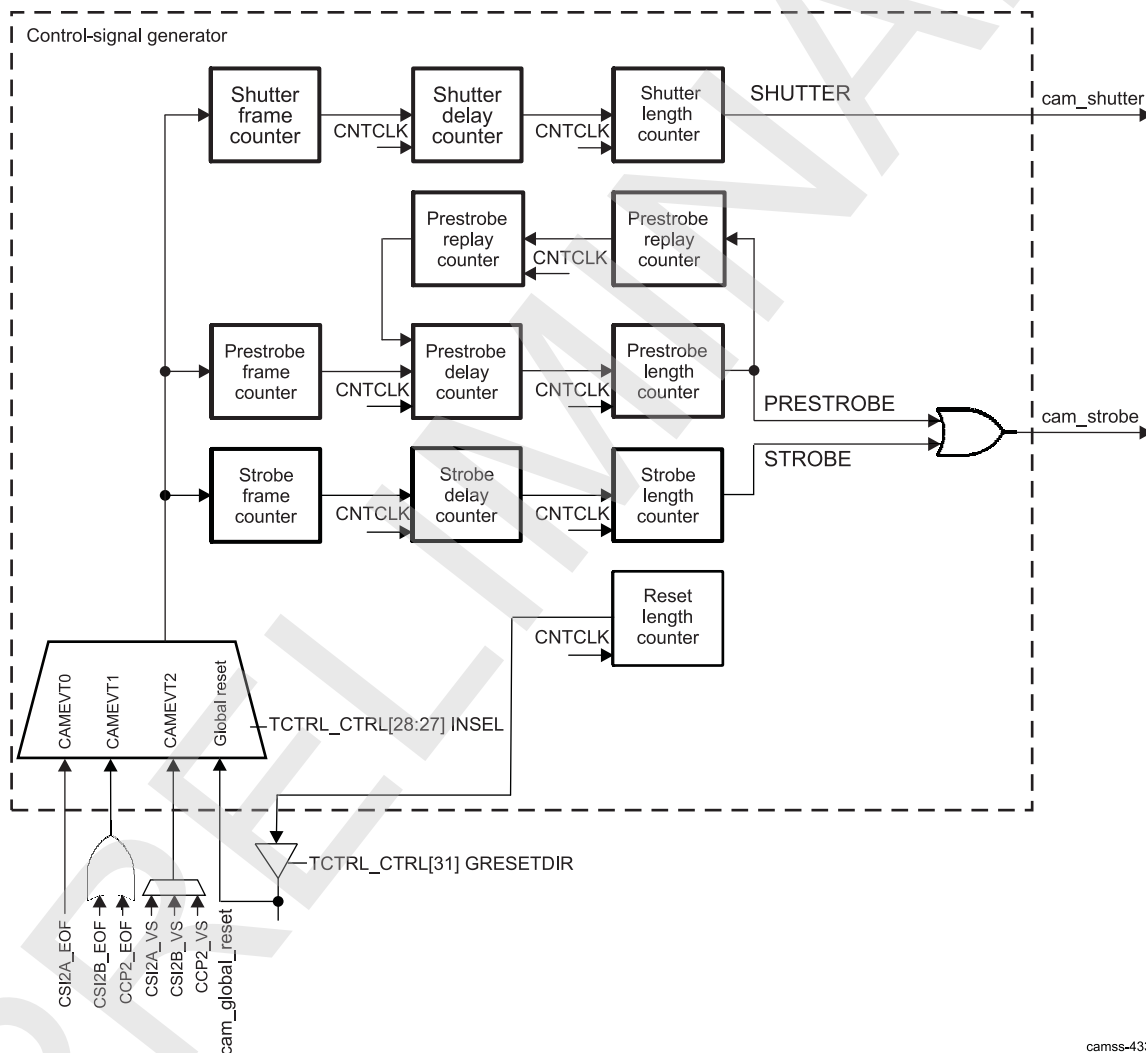
The TCTRL generates the control signals (cam\_strobe and cam\_shutter) for the flash prestrobe, flash strobe, and mechanical shutters.

The TCTRL includes a timing generator and a control-signal generator.

#### 8.2.6.3.2 ISS TCTRL Control-Signal Generator

The control-signal generator generates the prestrobe, strobe, and shutter signals: cam\_strobe and cam\_shutter. Figure 8-73 shows the principle of control-signal generation.

Figure 8-73. TCTRL Control-Signal Generation



The control-signal generator gathers precise timings for the cam\_strobe and cam\_shutter signals, to assert and deassert the signals at known times. The timing control-signal generator can be synchronized on the vertical synchronization signal coming from the CSI2-A, CSI2-B, CCP2 or on an externally generated cam\_global\_reset signal.

A multiplexer controls which of the CSI2-A, CSI2-B, or CCP2 interfaces drives control-signal generation. This multiplexer can also select the externally generated cam\_global\_reset signal as the trigger event. The TCTRL\_CTRL[31] GRESETDIR bit defines the direction of cam\_global\_reset.

- The externally generated `cam_global_reset` is used as a trigger when `TCTRL_CTRL[31] GRESETDIR = 0` and `TCTRL_CTRL[28:27] INSEL = 3`.
- The internally generated `cam_global_reset` is used as a trigger when `TCTRL_CTRL[31] GRESETDIR = 1` and `TCTRL_CTRL[28:27] INSEL = 3`.

The `cam_global_reset` signal can also be generated internally by the control-signal generator under software control. In this case, the prestrobe and shutter signals are synchronized on internally generated `cam_global_reset`. The multiplexer controls whether control-signal generation must be triggered by the internal or external `cam_global_reset`.

The prestrobe-, strobe-, and shutter-control signals can be individually enabled at any time. These signals must not be disabled by software.

The clock divider generates the `CNTCLK` clock based on the functional clock. The clock divider is programmable. [Table 8-198](#) lists the possible frequencies as a function of the divisor values.

**Table 8-198. ISS TCTRL Control-Signal Generator: CNTCLK Frequencies**

Divisor Value <code>TCTRL_CTRL[18:10] DIVC</code>	CNTCLK Clock	CNTCLK Precision (ns)
0 (default)	Clock gated. No clock.	N/A
1	200 MHz	5
2	100 MHz	10
3	66,667 MHz	15
4	50 MHz	20
...	...	...
510	0.392 MHz	2550
511	0.391 MHz	2555

There are three counters per control signal, for a total of nine counters. Each counter is programmable.

- The frame counter is decreased each time a full new frame is received.
  - A new frame is detected by the TCTRL module when `CAMEVTx` is received.
  - The frame counter determines how many whole frames must be ignored before the delay counter is triggered. The frame counters can be set to 0 to bypass the frame counters.
- The delay counter determines the control-signal activation delay. The counter is decreased at every `CNTCLK` clock cycle. When the counter reaches 0, the control signal is asserted. If the delay counter is set to 0, the control signal is asserted immediately.
- The activation-length counter determines the length of control-signal assertion. The counter is decreased at every `CNTCLK` clock cycle. When the counter reaches 0, the signal is deasserted and the control-signal enable bit is disabled. If the activation length is set to 0, the control signal is not asserted and the control-signal enable bit is disabled.

The polarity of the following signals can be individually selected:

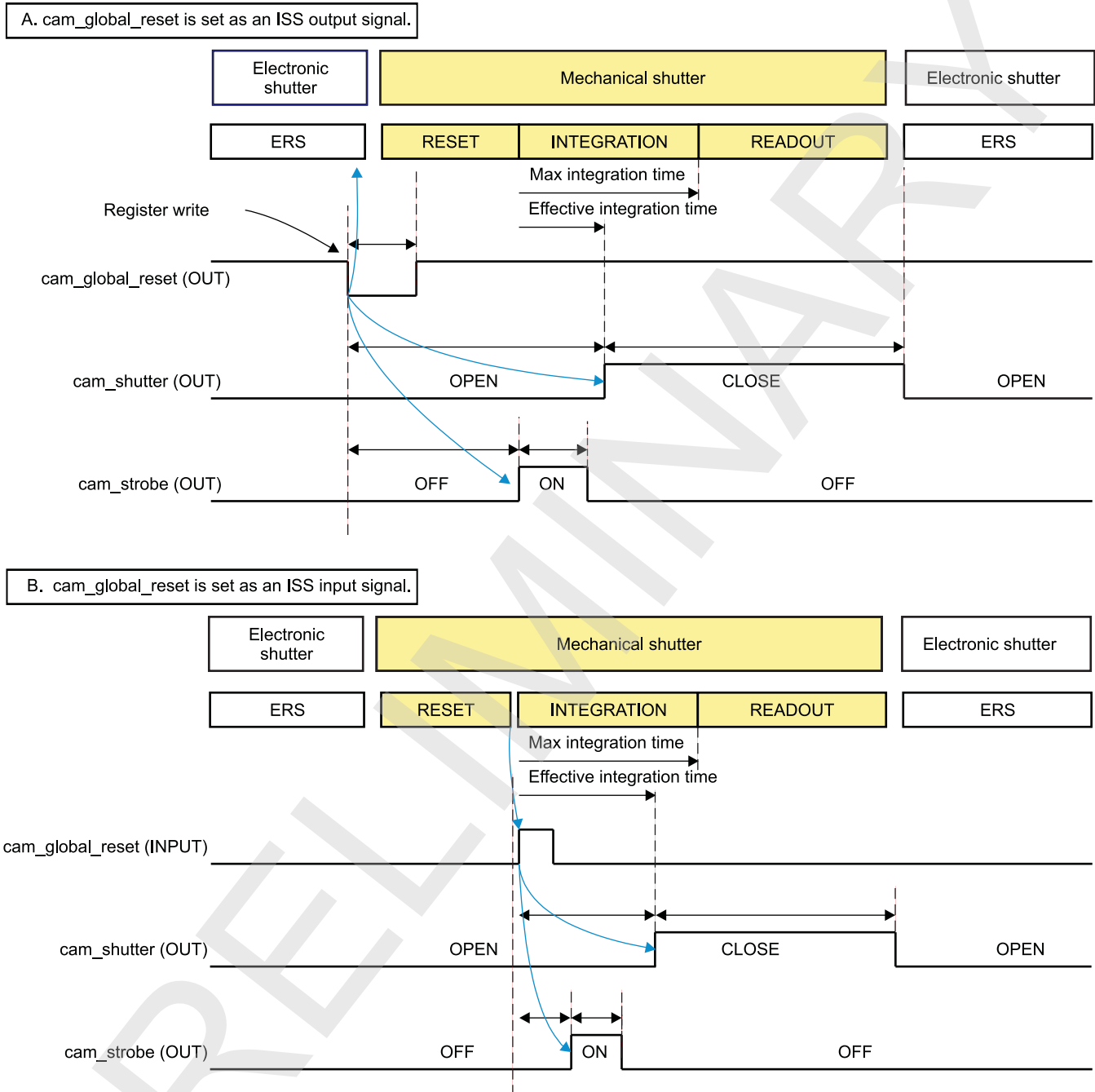
- `TCTRL_CTRL[26] STRBPSTRBPOL` for the prestrobe and strobe signals
- `TCTRL_CTRL[24] SHUTPOL` for the shutter signal
- `TCTRL_CTRL[30] GRESETPOL` for `cam_global_reset`

Software can trigger the generation of `cam_global_reset` to the camera module. The length of signal-activation is programmable. The counter is decreased at every `CNTCLK` clock cycle. When the counter reaches 0, the signal is deasserted and the global reset enable bit is disabled (the `TCTRL_CTRL[29] GRESETEN` bit). If the activation length is set to 0, the control signal is not asserted and the control-signal enable bit is disabled. The polarity of `cam_global_reset` can be selected (the `TCTRL_CTRL[30] GRESETPOL` bit).

[Figure 8-74](#) shows the use of `cam_global_reset` set as an input or output signal. `cam_global_reset` is asynchronous, edge-sensitive, and asserted for at least one interconnect clock cycle.



**Figure 8-74. ISS TCTRL Use of cam\_global\_reset With Global Reset Release Camera Modules**



camss-434

There are two types of shutter mechanisms: mechanical and electronic. A mechanical shutter is used only for high-resolution sensors. The three control signals (cam\_global\_reset, cam\_shutter, and cam\_strobe) are useful with a mechanical shutter. High frame rates can be achieved only with an electronic shutter. When an electronic shutter is used, none of the three control signals is used.

- Mechanical shutter mechanism:
  - Reset: All pixels of the sensor are reset to their black value. When the sensor has a global reset feature, the mechanical shutter can be open during reset.
  - Integration: The light received by the sensor is transformed into electrical charges that are stored inside pixels. At the end of the integration time, the shutter must be closed. Exposure time is

- defined by the time between reset release and shutter close.
- Readout: The charges accumulated in pixels are converted to digital values that are sent to the camera receiver.
- Electronic rolling shutter (ERS) mechanism:
  - Each line of the sensor is reset separately and read after a fixed amount of time. Exposure time is defined by the time between reset and read.

#### 8.2.6.4 ISS TCTRL Programming Model

The following settings must be done before enabling the TCTRL.

##### 8.2.6.4.1 ISS TCTRL Camera-Control Signal Generator

Two configurations apply:

- First configuration: The control signals are based on the vertical synchronization information coming from the camera module or from externally generated `cam_global_reset`.
- Second configuration: The control signals are based on internally generated `cam_global_reset`.

##### 8.2.6.4.1.1 ISS TCTRL Vertical Sync-Based Control-Signal Generation or Externally Generated `cam_global_reset`

To enable control-signal generation in the first configuration, follow the procedure listed in [Table 8-199](#).

**Table 8-199. ISS TCTRL Enabling the Control-Signal Generation in First Configuration**

Step	Bit Field	Value
Select the input that triggers the control signals. The trigger signal can come from the CSI2-A, or CSI2-B/CCP2 interface, or the externally generated <code>cam_global_reset</code> signal.	<a href="#">TCTRL_CTRL[28:27]</a> INSEL	0x0: CSI2A 0x1: CSI2B or CCP2 0x2: VS signal 0x3: Global reset
Set the global reset as input signal. Writes to the <a href="#">TCTRL_CTRL[29]</a> GRESETEN bit do not trigger the <code>CAM_STROBE</code> and <code>CAM_SHUTTER</code> signals and do not generate the <code>CAM_GLOBAL_RESET</code> signal.	<a href="#">TCTRL_CTRL[31]</a> GRESETDIR	0x0
The following bits are cleared automatically to 0 after the signal assertion:	<ul style="list-style-type: none"> <li>• <a href="#">TCTRL_CTRL[21]</a> SHUTEN</li> <li>• <a href="#">TCTRL_CTRL[22]</a> PSTRBEN</li> <li>• <a href="#">TCTRL_CTRL[23]</a> STRBEN</li> </ul>	
Set the polarity of the <code>SHUTTER</code> , <code>STROBE/PRESTROBE</code> , and <code>cam_global_reset</code> signals.	<ul style="list-style-type: none"> <li>• <a href="#">TCTRL_CTRL[24]</a> SHUTPOL</li> <li>• <a href="#">TCTRL_CTRL[26]</a> STRBPSTRBPOL</li> <li>• <a href="#">TCTRL_CTRL[30]</a> GRESETPOL</li> </ul>	
Set the clock divisor value, which generates the CNTCLK clock. The clock is set by <code>CNTCLK = ISS_FCLK/TCTRL_CTRL[18:10] DIVC</code> .	<a href="#">TCTRL_CTRL[18:10]</a> DIVC	0x0: Disable CNTCLK 0 to 511: Divider
Set the frame counters.	<ul style="list-style-type: none"> <li>• <a href="#">TCTRL_FRAME[5:0]</a> SHUT</li> <li>• <a href="#">TCTRL_FRAME[11:6]</a> PSTRB</li> <li>• <a href="#">TCTRL_FRAME[17:12]</a> STRB</li> </ul>	0: TCTRL does not delay any frame in input. 1 to 63
Set the delay counters.	<ul style="list-style-type: none"> <li>• <a href="#">TCTRL_SHUT_DELAY</a></li> <li>• <a href="#">TCTRL_PSTRB_DELAY</a></li> <li>• <a href="#">TCTRL_STRB_DELAY</a></li> </ul>	The possible values are 0 to $2^{25} - 1$ cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration is $(2^{25} - 1) \times 511/200$ MHz = 85.73157376 s ( <a href="#">TCTRL_CTRL[18:10]</a> DIVC = 511).

**Table 8-199. ISS TCTRL Enabling the Control-Signal Generation in First Configuration (continued)**

Step	Bit Field	Value
Set the signal durations.	<ul style="list-style-type: none"> <li>TCTRL_SHUT_LENGTH</li> <li>TCTRL_PSTRB_LENGTH</li> <li>TCTRL_STRB_LENGTH</li> </ul>	The possible values are 0 to $2^{24} - 1$ cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration for a 200 MHz input clock is $(2^{24} - 1) \times 511/200 \text{ MHz} = 42.865784325 \text{ s}$ (TCTRL_CTRL.DIVC = 511).
Enable the SHUTTER signal.	TCTRL_CTRL[21] SHUTEN	0x1
Enable the PRESTROBE signal.	TCTRL_CTRL[22] PSTRBEN	0x1
Enable the STROBE signal.	TCTRL_CTRL[23] STRBEN	0x1

#### 8.2.6.4.1.2 ISS TCTRL Internally Generated cam\_global\_reset-Based Control-Signal Generation

To enable control-signal generation in the second configuration, follow the procedure listed in [Table 8-200](#).

**Table 8-200. ISS TCTRL Enabling the Control-Signal Generation in Second Configuration**

Step	Bit Field	Value
Select the input to global reset, to loop back the internally generated GLOBAL_RESET. Vertical synchronization events do not trigger the CAM_STROBE and CAM_SHUTTER signals.	TCTRL_CTRL[28:27] INSEL	0x3: Global reset
Set the global reset as output signal.	TCTRL_CTRL[31] GRESETDIR	0x1
The following bits are cleared automatically to 0 after the signal assertion:	<ul style="list-style-type: none"> <li>TCTRL_CTRL[21] SHUTEN</li> <li>TCTRL_CTRL[22] PSTRBEN</li> <li>TCTRL_CTRL[23] STRBEN</li> <li>TCTRL_CTRL[29] GRESETEN</li> </ul>	0x0
Set the polarity of the SHUTTER, STROBE/PRESTROBE, and cam_global_reset signals.	<ul style="list-style-type: none"> <li>TCTRL_CTRL[24] SHUTPOL</li> <li>TCTRL_CTRL[26] STRBPSTRBPOL</li> <li>TCTRL_CTRL[30] GRESETPOL</li> </ul>	
Set the clock divisor value, which generates the CNTCLK clock. The clock is set by $\text{CNTCLK} = \text{ISS\_FCLK}/\text{TCTRL\_CTRL}[18:10] \text{ DIVC}$ .	TCTRL_CTRL[18:10] DIVC	0x0: Disable CNTCLK 0 to 511: Divider
Set the frame counters.	<ul style="list-style-type: none"> <li>TCTRL_FRAME[5:0] SHUT</li> <li>TCTRL_FRAME[11:6] PSTRB</li> <li>TCTRL_FRAME[17:12] STRB</li> </ul>	0: TCTRL does not delay any frame in input. 1 to 63
Set the delay counters.	<ul style="list-style-type: none"> <li>TCTRL_SHUT_DELAY</li> <li>TCTRL_PSTRB_DELAY</li> <li>TCTRL_STRB_DELAY</li> </ul>	The possible values are 0 to $2^{25} - 1$ cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration is $(2^{25} - 1) \times 511/200 \text{ MHz} = 85.73157376 \text{ s}$ (TCTRL_CTRL[18:10] DIVC = 511).

**Table 8-200. ISS TCTRL Enabling the Control-Signal Generation in Second Configuration (continued)**

Step	Bit Field	Value
Set the signal durations.	<ul style="list-style-type: none"> <li>TCTRL_SHUT_LENGTH</li> <li>TCTRL_PSTRB_LENGTH</li> <li>TCTRL_STRB_LENGTH</li> </ul>	The possible values are 0 to $2^{24} - 1$ cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration for a 200 MHz input clock is $(2^{24} - 1) \times 511/200 \text{ MHz} = 42.865784325 \text{ s}$ (TCTRL_CTRL.DIVC = 511).
Set the cam_global_reset assertion time.	TCTRL_GRESET_LENGTH	The possible values are 0 to $2^{24} - 1$ cycles. The cycles are at the CNTCLK clock frequency. The maximum signal duration for a 200 MHz input clock is $(2^{24} - 1) \times 511/200 \text{ MHz} = 42.865784325 \text{ s}$ (TCTRL_CTRL.DIVC = 511).
Enable the SHUTTER signal.	TCTRL_CTRL[21] SHUTEN	0x1
Enable the PRESTROBE signal.	TCTRL_CTRL[22] PSTRBEN	0x1
Enable the STROBE signal.	TCTRL_CTRL[23] STRBEN	0x1
Enable the cam_global_reset control-signal generation.	TCTRL_CTRL[29] GRESETEN	0x1

**NOTE:** Setting the following bits to 1 simultaneously leads to unpredictable behavior:

- TCTRL\_CTRL[21] SHUTEN
- TCTRL\_CTRL[22] PSTRBEN
- TCTRL\_CTRL[23] STRBEN
- TCTRL\_CTRL [29] GRESETEN

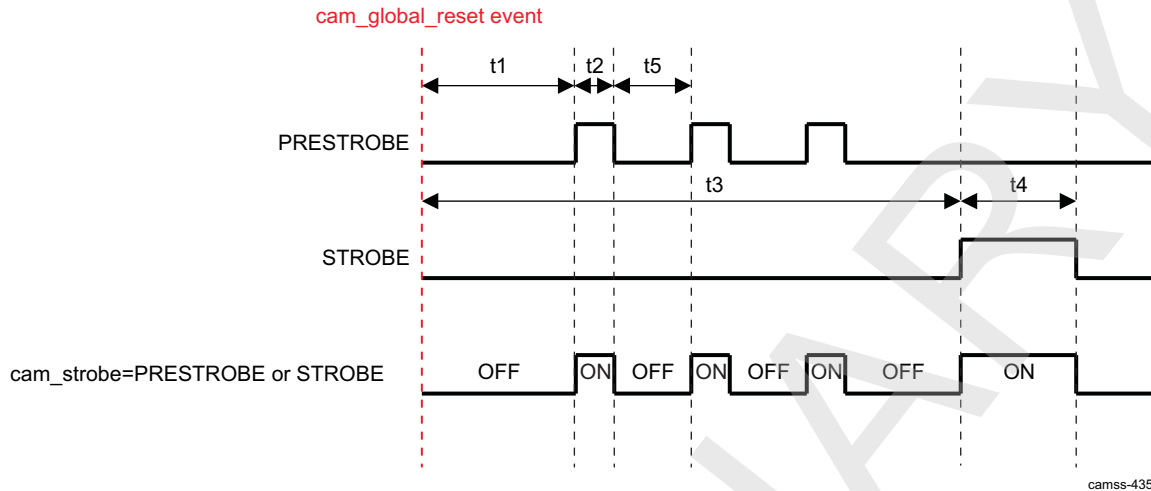
The following bits must be set before TCTRL\_CTRL[29] GRESETEN is enabled:

- TCTRL\_CTRL[21] SHUTEN
- TCTRL\_CTRL[22] PSTRBEN
- TCTRL\_CTRL[23] STRBEN

#### 8.2.6.4.1.3 ISS TCTRL STROBE and PRESTROBE Signal Generation for Red-Eye Removal

The STROBE and PRESTROBE signal generation enables a strobe flash for red-eye removal. The process is shown in [Figure 8-75](#). The dotted line corresponds to known timings from which the delay counters start decreasing: cam\_global\_reset event.

Figure 8-75. cam\_strobe Signal-Generation for Red-Eye Removal



- t1: Set by the [TCTRL\\_PSTRB\\_DELAY](#) register
- t2: Set by the [TCTRL\\_PSTRB\\_LENGTH](#) register
- t5: Set by the [TCTRL\\_PSTRB\\_REPLAY\[24:0\]](#) DELAY bit field. The number of times the pulse is repeated is controlled by the [TCTRL\\_PSTRB\\_REPLAY\[31:25\]](#) COUNTER bit field. In the previous example, [TCTRL\\_PSTRB\\_REPLAY\[31:25\]](#) COUNTER = 2.
  - The possible delay values are 0 to  $2^{25} - 1$  cycle. The cycles are at the CNTCLK clock frequency. The maximum signal duration is  $(2^{25} - 1) \times 511/200$  MHz = 85.73157376 s ([TCTRL\\_CTRL\[18:11\]](#) DIVC = 511).
  - The possible count values are 0 to 127 additional pulses.
- t3: Set by the [TCTRL\\_STRB\\_DELAY](#) register
- t4: Set by the [TCTRL\\_STRB\\_LENGTH](#) register

## 8.2.6.5 ISS TCTRL Register Manual

### 8.2.6.5.1 ISS TCTRL Instance Summary

[Table 8-201](#) summarizes the TCTRL instance.

Table 8-201. ISS TCTRL Instance Summary

Module Name	L3 Base Address	Size
<a href="#">ISS_TCTRL</a>	0x5200 0400	256 bytes

### 8.2.6.5.2 ISS TCTRL Registers

#### 8.2.6.5.2.1 ISS TCTRL Register Summary

[Table 8-202](#) summarizes the TCTRL registers.

Table 8-202. ISS TCTRL Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_TCTRL Base Address
<a href="#">TCTRL_REVISION</a>	R	32	0x0000 0000	0x5200 0400
<a href="#">TCTRL_SYSCONFIG</a>	RW	32	0x0000 0004	0x5200 0404
<a href="#">TCTRL_SYSSTATUS</a>	R	32	0x0000 0008	0x5200 0408
<a href="#">TCTRL_STRB_LENGTH</a>	RW	32	0x0000 0010	0x5200 0410

**Table 8-202. ISS TCTRL Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_TCTRL Base Address
TCTRL_PSTRB_LEN GH	RW	32	0x0000 0014	0x5200 0414
TCTRL_SHUT_LEN GH	RW	32	0x0000 0018	0x5200 0418
TCTRL_GRESET_LEN GTH	RW	32	0x0000 001C	0x5200 041C
TCTRL_STRB_DELAY	RW	32	0x0000 0020	0x5200 0420
TCTRL_PSTRB_DELAY	RW	32	0x0000 0024	0x5200 0424
TCTRL_SHUT_DELAY	RW	32	0x0000 0028	0x5200 0428
TCTRL_CTRL	RW	32	0x0000 0030	0x5200 0430
TCTRL_PSTRB_REPLA Y	RW	32	0x0000 0034	0x5200 0434
TCTRL_FRAME	RW	32	0x0000 0038	0x5200 0438

**8.2.6.5.2.2 ISS TCTRL Register Description**

Table 8-203 through Table 8-227 describe the TCTRL registers.

**Table 8-203. TCTRL\_REVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	ISS_TCTRL
<b>Physical Address</b>	See Table 8-202.		
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 8-204. Register Call Summary for Register TCTRL\_REVISION**

ISS Interfaces

- [ISS TCTRL Register Manual: \[0\]](#)

**Table 8-205. TCTRL\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	ISS_TCTRL
<b>Physical Address</b>	See Table 8-202.		
<b>Description</b>	OCP-SOCKET SYSTEM CONFIGURATION REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SOFT_RESET	AUTO_IDLE		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	SOFT_RESET	Software reset. Set the bit to 1 to trigger the module reset. The bit is automatically reset by the hardware. During reads return 0.  0x0: Normal mode. 0x1: The module is reset.	RW	0
0	AUTO_IDLE	Internal OCP and functional clock gating strategy  0x0: OCP and functional clocks are free-running 0x1: Automatic clock gating strategy is applied, based on the OCP interface activity for interface clock and on the functional activity for functional clocks.	RW	1

**Table 8-206. Register Call Summary for Register TCTRL\_SYSCONFIG**

ISS Interfaces

- [ISS TCTRL Register Manual: \[0\]](#)

**Table 8-207. TCTRL\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	ISS_TCTRL
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .		
<b>Description</b>	OCP-SOCKET SYSTEM STATUS REGISTER		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESET_DONE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	RESET_DONE	Internal reset monitoring  Read 0x1: Reset completed. Read 0x0: Internal module reset is ongoing.	R	0

**Table 8-208. Register Call Summary for Register TCTRL\_SYSSTATUS**

ISS Interfaces

- [ISS TCTRL Register Manual: \[0\]](#)

**Table 8-209. TCTRL\_STRB\_LENGTH**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	ISS_TCTRL
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .		
<b>Description</b>	TIMING CONTROL - STROBE LENGTH REGISTER This register is used by the TIMING CTRL module to generate the STROBE signal.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LENGTH																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:0	LENGTH	Sets the length of the CAM_STROBE signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the <a href="#">TCTRL_CTRL.DIVC</a> bit field. After signal assertion, the <a href="#">TCTRL_CTRL.STRBEN</a> bit is automatically cleared. The possible values are 0 to $2^{24}-1$ cycles.	RW	0x000000

**Table 8-210. Register Call Summary for Register TCTRL\_STRB\_LENGTH**

## ISS Interfaces

- [ISS TCTRL Programming Model: \[0\] \[1\] \[2\]](#)
- [ISS TCTRL Register Manual: \[3\] \[4\]](#)

**Table 8-211. TCTRL\_PSTRB\_LENGTH**

<b>Address Offset</b>	0x0000 0014	
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .	<b>Instance</b> ISS_TCTRL
<b>Description</b>	TIMING CONTROL - PRESTROBE LENGTH REGISTER This register is used by the TIMING CTRL module to generate the PRESTROBE signal.	
<b>Type</b>	RW	
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED		
LENGTH		

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:0	LENGTH	Sets the length of the CAM_PRESTROBE signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the <a href="#">TCTRL_CTRL.DIVC</a> bit field. After signal assertion, the <a href="#">TCTRL_CTRL.PSTRBEN</a> bit is automatically cleared. The possible values are 0 to $2^{24}-1$ cycles.	RW	0x000000

**Table 8-212. Register Call Summary for Register TCTRL\_PSTRB\_LENGTH**

## ISS Interfaces

- [ISS TCTRL Programming Model: \[0\] \[1\] \[2\]](#)
- [ISS TCTRL Register Manual: \[3\] \[4\] \[5\] \[6\] \[7\]](#)

**Table 8-213. TCTRL\_SHUT\_LENGTH**

<b>Address Offset</b>	0x0000 0018	
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .	<b>Instance</b> ISS_TCTRL
<b>Description</b>	TIMING CONTROL - SHUTTER LENGTH REGISTER This register is used by the TIMING CTRL module to generate the SHUTTER signal.	
<b>Type</b>	RW	
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED		
LENGTH		



Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:0	LENGTH	Sets the length of the CAM_SHUTTER signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the <a href="#">TCTRL_CTRL.DIVC</a> bit field. After signal assertion, the <a href="#">TCTRL_CTRL.SHUTEN</a> bit is automatically cleared. The possible values are 0 to $2^{24}-1$ cycles.	RW	0x000000

**Table 8-214. Register Call Summary for Register TCTRL\_SHUT\_LENGTH**

ISS Interfaces

- [ISS TCTRL Programming Model: \[0\] \[1\]](#)
- [ISS TCTRL Register Manual: \[2\] \[3\]](#)

**Table 8-215. TCTRL\_GRESET\_LENGTH**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	ISS_TCTRL
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .		
<b>Description</b>	TIMING CONTROL - GLOBAL SHUTTER LENGTH REGISTER This register is used by the TIMING CTRL module to generate the CAM.GRESET signal.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LENGTH																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:0	LENGTH	Sets the length of the CAM_GLOBAL_RESET signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the <a href="#">TCTRL_CTRL.DIVC</a> bit field. After signal assertion, the <a href="#">TCTRL_CTRL.GRESETEN</a> bit is automatically cleared. The possible values are 0 to $2^{24}-1$ cycles. The polarity of the CAM_GLOBAL_RESET signal is set by the <a href="#">TCTRL_CTRL.GRESETPOL</a> bit.	RW	0x000000

**Table 8-216. Register Call Summary for Register TCTRL\_GRESET\_LENGTH**

ISS Interfaces

- [ISS TCTRL Programming Model: \[0\]](#)
- [ISS TCTRL Register Manual: \[1\] \[2\]](#)

**Table 8-217. TCTRL\_STRB\_DELAY**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	ISS_TCTRL
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .		
<b>Description</b>	TIMING CONTROL - STROBE DELAY REGISTER This register is used by the TIMING CTRL module to generate the STROBE signal.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELAY																							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:0	DELAY	Sets the delay for the CAM_STROBE signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the <a href="#">TCTRL_CTRL.DIVC</a> bit field. The possible values are 0 to $2^{25}-1$ cycles.	RW	0x0000000

**Table 8-218. Register Call Summary for Register TCTRL\_STRB\_DELAY**

ISS Interfaces

- [ISS TCTRL Programming Model: \[0\] \[1\] \[2\]](#)
- [ISS TCTRL Register Manual: \[3\] \[4\]](#)

**Table 8-219. TCTRL\_PSTRB\_DELAY**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	ISS_TCTRL
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .		
<b>Description</b>	TIMING CONTROL - PRE STROBE DELAY REGISTER This register is used by the TIMING CTRL module to generate the PRESTROBE signal.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELAY																							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:0	DELAY	Sets the delay for the CAM_PSTROBE signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the <a href="#">TCTRL_CTRL.DIVC</a> bit field. The possible values are 0 to $2^{25}-1$ cycles.	RW	0x0000000

**Table 8-220. Register Call Summary for Register TCTRL\_PSTRB\_DELAY**

ISS Interfaces

- [ISS TCTRL Programming Model: \[0\] \[1\] \[2\]](#)
- [ISS TCTRL Register Manual: \[3\] \[4\] \[5\]](#)

**Table 8-221. TCTRL\_SHUT\_DELAY**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	ISS_TCTRL
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .		
<b>Description</b>	TIMING CONTROL - SHUTTER DELAY REGISTER This register is used by the TIMING CTRL module to generate the SHUTTER signal.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELAY																							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:0	DELAY	Sets the delay for the CAM_SHUTTER signal assertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the <a href="#">TCTRL_CTRL.DIVC</a> bit field. The possible values are 0 to $2^{25}-1$ cycles.	RW	0x0000000

**Table 8-222. Register Call Summary for Register TCTRL\_SHUT\_DELAY**

ISS Interfaces

- [ISS TCTRL Programming Model: \[0\] \[1\]](#)
- [ISS TCTRL Register Manual: \[2\] \[3\]](#)

**Table 8-223. TCTRL\_CTRL**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	ISS_TCTRL
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .		
<b>Description</b>	TIMING CONTROL - CONTROL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GRESETDIR	GRESETPOL	GRESETEN	INSEL	STRBPSTRBPOL	RESERVED	SHUTPOL	STRBEN	PSTRBEN	SHUTEN	RESERVED	DIVC						RESERVED														

Bits	Field Name	Description	Type	Reset
31	GRESETDIR	Sets the direction of the GLOBAL_RESET signal. 0x0: INPUT – GLOBAL_RESET is an input to the TIMING CONTROL module. GLOBAL_RESET is externally generated. 0x1: OUTPUT – GLOBAL_RESET is an output of the TIMING CONTROL module. GLOBAL_RESET is internally generated. If GRESETEN is set to 1, the internally generated GLOBAL_RESET will trigger the generation of the PRESTROBE, STROBE and SHUTTER signals. The frame counters are ignored.	RW	0
30	GRESETPOL	Sets the polarity of the global reset signal: CAM_GLOBAL_RESET. It applies whatever the direction of the GLOBAL_RESET signal: input or output. 0x0: active high 0x1: active low	RW	0
29	GRESETEN	Triggers the generation of the CAM_GLOBAL_RESET signal. The signal is asserted immediately. If enabled, the CAM_GLOBAL_RESET signal will be asserted for <a href="#">TCTRL_GRESET_LENGTH</a> cycles. After the signal assertion, the enable bit is automatically cleared to 0. The polarity of the GLOBAL_RESET signal is set with <a href="#">TCTRL_CTRL.GRESETPOL</a> . Enabling this bit triggers the generation of the CAM_SHUTTER and CAM_STROBE signals (if previously enabled). The frame counters shall be set to 0 when this bit is set to 1 and GRESETDIR is set a OUTPUT.	RW	0
28:27	INSEL	Sets the mode that will trigger the SHUTTER, PRESTROBE and STROBE signals.  0x0: Synchronization event from camera 0 0x1: Synchronization event from camera 1  0x3: GRESET – The CAM_GLOBAL_RESET input signal will trigger the SHUTTER, PRESTROBE and STROBE signals. In this mode, there are no frame counters. The delay counters start decrementing as soon as the GLOBAL_RESET signal is asserted. The polarity of the GLOBAL_RESET signal is set with <a href="#">TCTRL_CTRL.GRESETPOL</a> .  0x2: Synchronization event from camera 2	RW	0x0

Bits	Field Name	Description	Type	Reset
26	STRBPSTRBPOL	Sets the polarity of the strobe and prestrobe signals. 0x0: Active high 0x1: Active low	RW	0
25	RESERVED		R	0
24	SHUTPOL	Sets the polarity of the mechanical shutter signal: CAM_SHUTTER 0x0: Active high 0x1: Active low	RW	0
23	STRBEN	Flash strobe signal enable. If enabled, the STROBE signal will be asserted after <a href="#">TCTRL_FRAME</a> .STRB frames have been received and a delay of <a href="#">TCTRL_STRB_DELAY</a> cycles have passed. The STROBE signal is asserted for <a href="#">TCTRL_STRB_LENGTH</a> cycles. After the signal assertion, the enable bit is automatically cleared to 0. This signal shall not be disabled by software.	RW	0
22	PSTRBEN	Flash prestrobe signal enable. If enabled, the PRESTROBE signal will be asserted after <a href="#">TCTRL_FRAME</a> .PSTRB frames have been received and a delay of <a href="#">TCTRL_PSTRB_DELAY</a> cycles have passed. The PRESTROBE signal is asserted for <a href="#">TCTRL_PSTRB_LENGTH</a> cycles. After the signal assertion, the enable bit is automatically cleared to 0. This signal shall not be disabled by software.	RW	0
21	SHUTEN	Mechanical shutter signal enable. If enabled, the SHUTTER signal will be asserted after <a href="#">TCTRL_FRAME</a> .SHUT frames have been received and a delay of <a href="#">TCTRL_SHUT_DELAY</a> cycles have passed. The SHUTTER signal is asserted for <a href="#">TCTRL_SHUT_LENGTH</a> cycles. After the signal assertion, the enable bit is automatically cleared to 0. This signal shall not be disabled by software.	RW	0
20:19	RESERVED		R	0x0
18:10	DIVC	Sets the clock divisor value for the CNTCLK clock generation based on the CLK input clock. CNTCLK is an internal clock used by the TIMING CTRL module counters. Usually, CNTCLK = CLK / DIVC, except for some particular values shown hereafter. 0x0: No clock. CNTCLK is gated.	RW	0x000
9:0	RESERVED		R	0x000

**Table 8-224. Register Call Summary for Register TCTRL\_CTRL**

ISS Interfaces

- [ISS TCTRL Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [ISS TCTRL Programming Model: \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\] \[44\] \[45\] \[46\] \[47\] \[48\] \[49\] \[50\] \[51\] \[52\] \[53\]](#)
- [ISS TCTRL Register Manual: \[54\] \[55\] \[56\] \[57\] \[58\] \[59\] \[60\] \[61\] \[62\] \[63\] \[64\] \[65\] \[66\] \[67\] \[68\] \[69\]](#)

**Table 8-225. TCTRL\_PSTRB\_REPLAY**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .	<b>Instance</b>	ISS_TCTRL
<b>Description</b>	TIMING CONTROL - PRESTROBE REPLAY REGISTER This register is used by the TIMING CTRL module to generate the prestrobe signal.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER								DELAY																							

Bits	Field Name	Description	Type	Reset
31:25	COUNTER	Sets the number of PRESTROBE pulses after the original pulse. If this bit is set to 0, the PRESTROBE signal behavior is only controlled by <a href="#">TCTRL_FRAME.STRB</a> , <a href="#">TCTRL_PSTRB_DELAY</a> , and <a href="#">TCTRL_PSTRB_LENGTH</a> . If <a href="#">TCTRL_PSTRB_LENGTH</a> =0, there is no replay. This bit is useful when one wants to enable red-eye removal.	RW	0x00
24:0	DELAY	Sets the delay for the PRESTROBE signal reassertion in cycles of the CNTCLK clock. The CNTCLK frequency is generated with the <a href="#">TCTRL_CTRL.DIVC</a> bit field. The possible values are 0 to $2^{25}-1$ cycles. If <a href="#">TCTRL_PSTRB_LENGTH</a> =0, there is no replay. This bit field shall not be set to 0 if the COUNTER is set to a value different of 0. This bit is useful when one wants to enable red-eye removal.	RW	0x0000000

**Table 8-226. Register Call Summary for Register TCTRL\_PSTRB\_REPLAY**

- ISS Interfaces
- [ISS TCTRL Programming Model: \[0\] \[1\] \[2\]](#)
  - [ISS TCTRL Register Manual: \[3\]](#)

**Table 8-227. TCTRL\_FRAME**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 8-202</a> .	<b>Instance</b>	ISS_TCTRL
<b>Description</b>	TIMING CONTROL - FRAME REGISTER This register is used by the TIMING CTRL module to generate the SHUTTER, PRESTROBE, and STROBE signals.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STRB				PSTRB				SHUT															

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:12	STRB	Frame counter for the STROBE signal generation. From 0 to 63 frames. This bit field is ignored if <a href="#">TCTRL.INSEL</a> =GRESET.	RW	0x00
11:6	PSTRB	Frame counter for the PRESTROBE signal generation. From 0 to 63 frames. This bit field is ignored if <a href="#">TCTRL.INSEL</a> =GRESET.	RW	0x00
5:0	SHUT	Frame counter for the SHUTTER signal generation. From 0 to 63 frames. This bit field is ignored if <a href="#">TCTRL.INSEL</a> =GRESET.	RW	0x00

---

**Table 8-228. Register Call Summary for Register TCTRL\_FRAME**

---

## ISS Interfaces

- [ISS TCTRL Programming Model: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
  - [ISS TCTRL Register Manual: \[6\] \[7\] \[8\] \[9\] \[10\]](#)
-

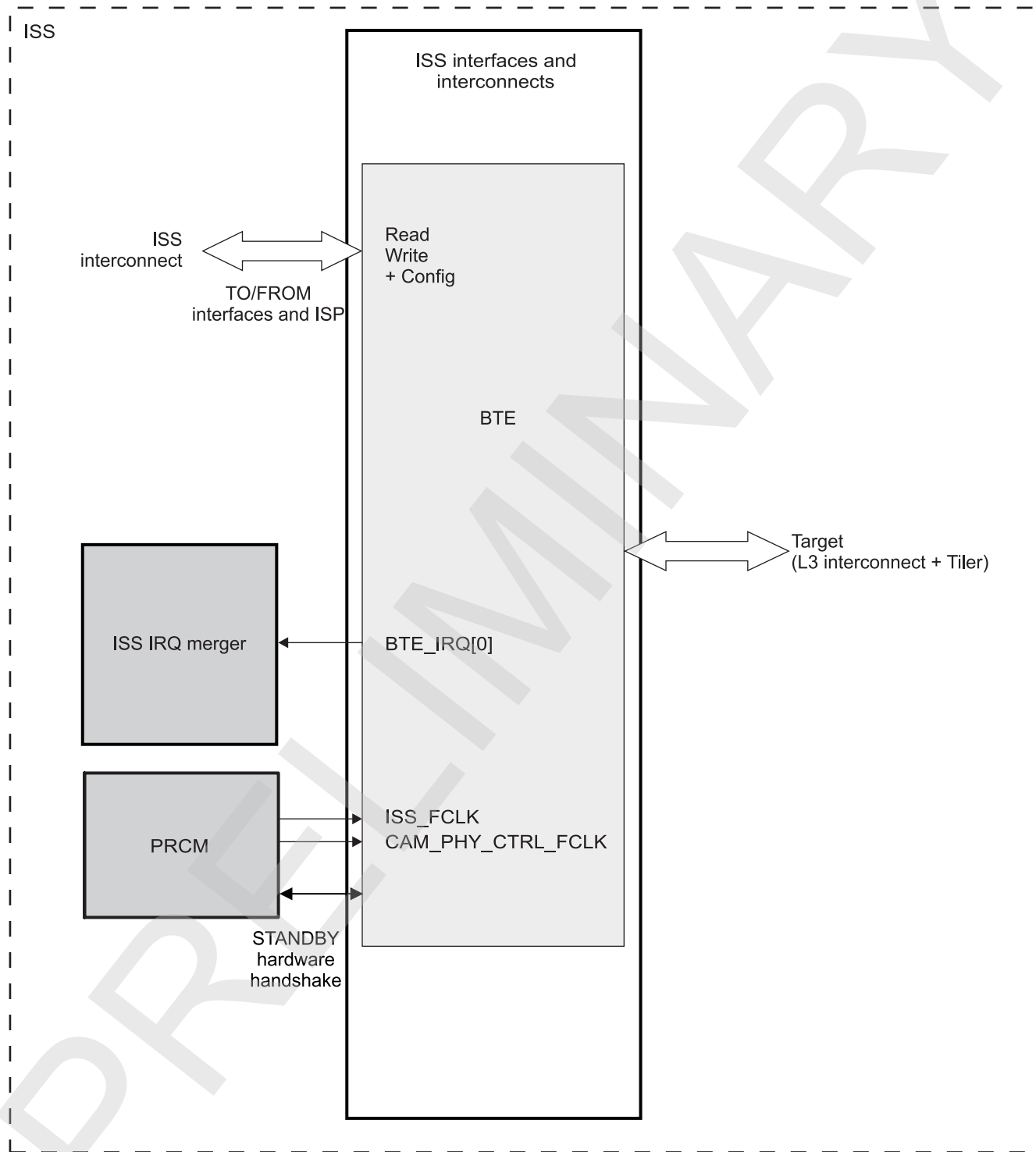
## 8.2.7 ISS BTE

### 8.2.7.1 ISS BTE Environment

There are no particular environment attributes (see [Section 8.2.2, ISS Interfaces Environment](#)).

### 8.2.7.2 ISS BTE Integration

Figure 8-76. ISS BTE Integration



camss-318

For power domain, clocks, reset, and hardware requests, see [Section 8.1.2.4, ISS Power Management](#).



### 8.2.7.2.1 ISS BTE PRCM Interface

#### 8.2.7.2.1.1 ISS BTE Power Management

##### 8.2.7.2.1.1.1 ISS BTE PRCM Handshake

The BTE supports the IDLE protocol to flush outstanding transactions. When an IDLE request is received, the BTE:

- Completes ongoing requests on OCPI (request, data and response phases) and stalls the port (SCmdAccept = 0)
- Flushes all contexts (same behavior than when the [BTE\\_CONTEXT\\_CTRL\\_i\[2\]](#) FLUSH bit is set by software) when the [BTE\\_CONTEXT\\_CTRL\\_i\[11\]](#) AUTOFLUSH bit is set. Otherwise, no context flushing is triggered by an IDLE request.
- Completes ongoing requests on OCPO (request, data and response phases)
- Acknowledges the IDLE request.

### 8.2.7.3 ISS BTE Functional Description

#### 8.2.7.3.1 ISS BTE Features

The BTE increases access efficiency of raster initiators to tiled SDRAM. In fact, the TILER expects 2D-bursts corresponding to a row or column of subtiles for maximal efficiency. For more information about SDRAM and TILER, see [Section 15.2, Dynamic Memory Manager](#).

The BTE is connected between one or multiple raster initiators. It can translate reads and writes. For reads, BTE prefetches sufficient data from tiled memory to translate raster requests. For writes, BTE buffers raster requests until it has sufficient data to generate requests to tiled memory. The features of the BTE are:

- Interfaces:
  - 32-bit-wide configuration interface (OCPC)
  - 128-bit-wide slave data port (OCPI)
  - 128-bit-wide master data port (OCPO)
- Incrementing to 2D burst translation for read and writes:
  - Four contexts. A context is a virtual frame buffer attached to a data flow requiring translation.
  - One-shot and continuous mode
- Local memories for temporal storage:
  - Cannot use external memories for temporal data storage
- Transparent for accesses that do not require translation. Requests are forwarded from OCPI to OCPO without modification.
- Local buffer

#### 8.2.7.3.2 ISS BTE Functional Description Details

The main tasks and capabilities of the BTE are:

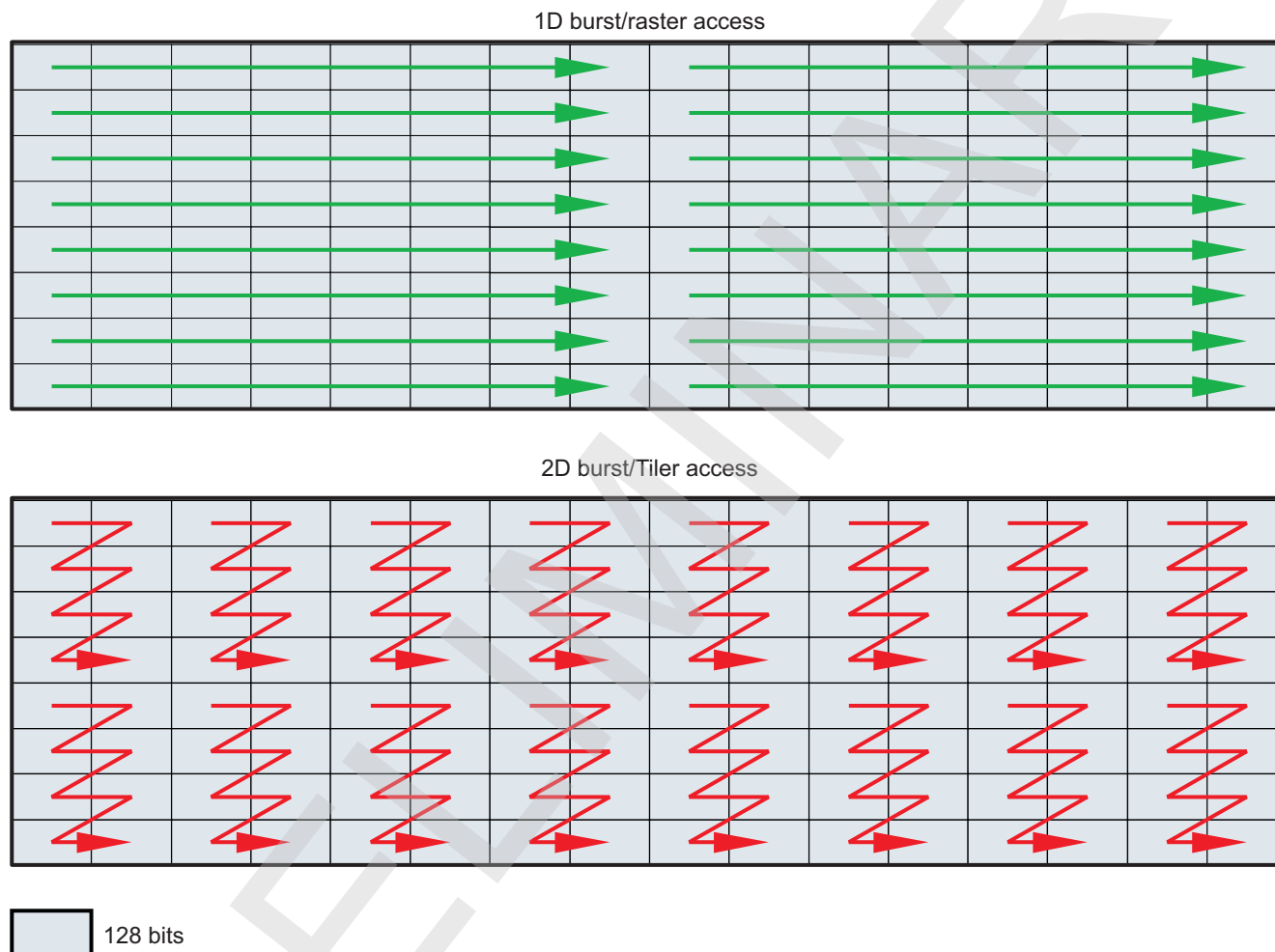
- Forward OCP transactions that do not need translation.
- For OCP writes requiring translation:
  - Store data received from OCPI to local buffers.
  - Read data from local buffers and send it to OCPO.
- For OCP read requiring translation:
  - Store data received from OCPO to local buffers.
  - Read data from local buffers and send it to OCPI.
- BTE DMA capability

### 8.2.7.3.2.1 ISS BTE Burst Translation Principle

The BTE receives raster accesses from the OCPI port and generates TILER accesses on OCPO.

The BTE keeps track of the amount of data written into the local memories. For translated writes, 2D bursts are sent to OCPO when there is sufficient data in the local memory to generate TILER accesses. For translated reads, 2D burst are generated to fill the local memory with data that is returned when raster accesses are received on OCPI. [Figure 8-77](#) shows the BTE burst translation principle.

**Figure 8-77. ISS BTE Burst Translation Principle**



camss-320

In the following sections, three address spaces are described:

- **Virtual:** Corresponds to the translated address region on OCPI. Accesses performed to this address space are translated by the BTE. It can be seen as a 64KB x 8 k lines frame buffer. Locations in this space are expressed as byte addresses for OCP compliance. However, the access granularity is 128-bit (4 LSBs expected to be zeros). For more information about this space, see [Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping](#).
- **Physical:** Corresponds to the addresses used to access the physical buffer of the BTE. The physical space is addressed in 128-bit-wide words. However, this document refers to byte addresses to preserve homogeneity.
- **Tiler:** Corresponds to addresses used by translated accesses. Locations in this space are expressed as byte addresses for OCP compliance. However, the access granularity is 128-bit (4 LSBs expected to be zeros). For more information about this space, see [Section 8.2.7.3.2.5, ISS BTE TILER Space Accesses: 2D Burst Generation](#).

### 8.2.7.3.2.2 ISS BTE Virtual Address Space and Context Mapping

The location of the virtual space in the address map of OCPI is set by the `BTE_CTRL[11:8]` BASE bit field. It always occupies 64KB × 8k lines = 512MB. Software must map a virtual space into an unused region (a region, for example, that the ISS top level cannot access).

The virtual space is decomposed into contexts. A context corresponds to a 2D region in the virtual space that requires burst translation. It can also be seen as a virtual frame buffer.

Accesses to different contexts can be interleaved at OCP transaction level. OCP transactions spanning multiple contexts are not allowed.

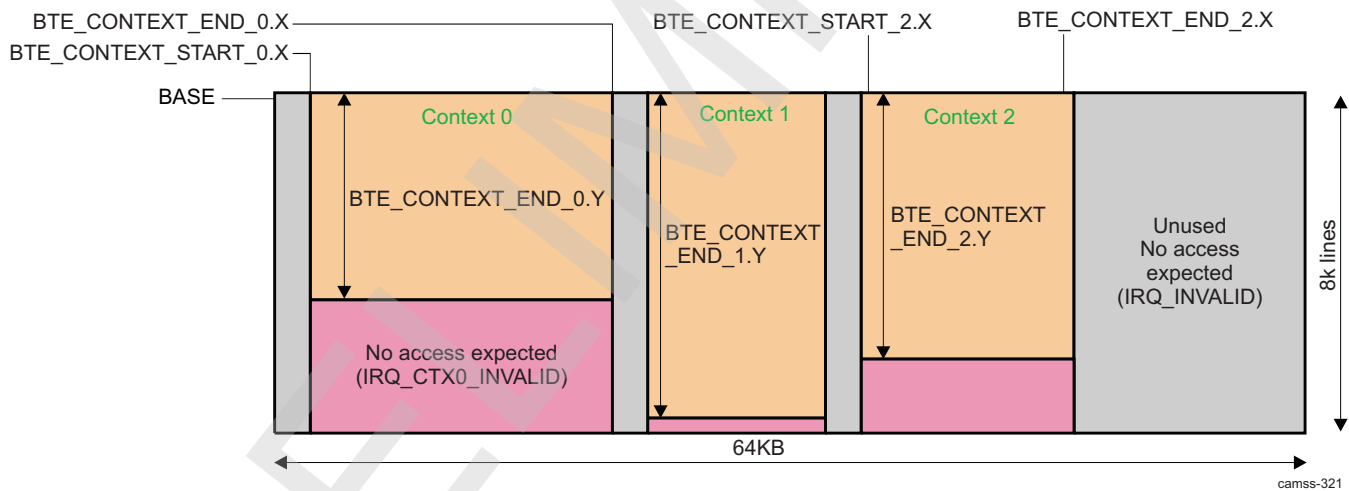
When the BTE receives an access from the OCPI port, it performs the following checks:

- If the access address is not between BASE 512MB and (BASE + 1) 512MB 1, it is handled as a transparent: request and data phases are simply forwarded to the OCPO without modification.
- If the access falls into the translated region:
  - If the access is a 2D burst, an `IRQ_INVALID` event is generated.
  - If the access does not map to an active context, an `IRQ_INVALID` event is generated. The BTE ignores bits [28:16] for this test.
  - Otherwise, context mapping is performed (see [Figure 8-78](#)).

Invalid requests are not forwarded to the OCPO.

Software must ensure that contexts do not overlap. The BTE hardware does not check for this condition. Wrong setup is likely to lead to corrupted data.

**Figure 8-78. ISS BTE Context Mapping**



The BTE internally keeps track of every context where the next access is expected. It internally maintains a 2D pointer, referred to as  $(SX_i, SY_i)$  in the remainder of this document. The expected byte address for an access into context  $x$  is:

$$\text{ADDR} = \text{BTE\_CTRL}[11:8] \text{ BASE } 512\text{MB} + \text{SX}_i \text{ 16 bytes} + \text{SY}_i \text{ 64KB}$$

If an access to an unexpected location in a given context is received from OCPI, the BTE generates an `IRQ_CTXx_INVALID` event. The BTE provides a valid response on OCPI but does not store any data into the internal buffer. Subsequent accesses are handled normally: in other words, the BTE does not enter any specific error mode. When this happens, typically an initiator configuration is not aligned with the BTE context configuration.

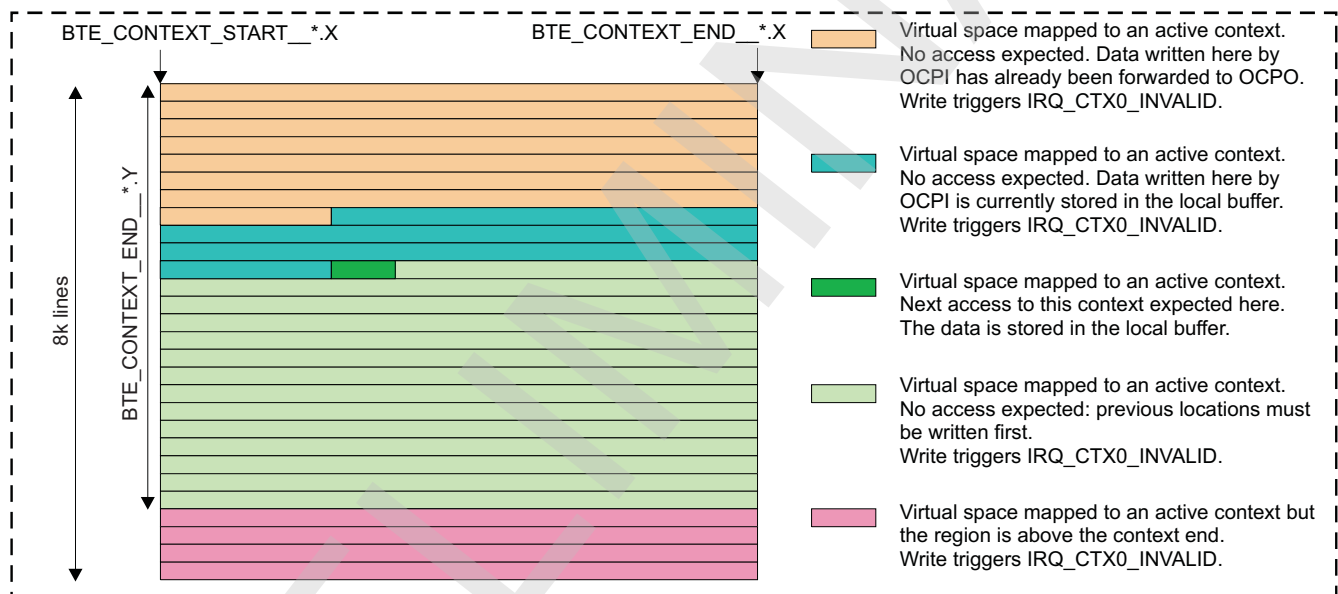
The (SX<sub>i</sub>, SY<sub>i</sub>) pointer of a context is updated on every access to this context. It is reset to SX<sub>i</sub> = BTE\_CONTEXT\_START<sub>i</sub>[15:7] X and SY<sub>x</sub> = 0 when a context is enabled. SX<sub>i</sub> is incremented by 1 for every received 16-byte word, except when it equals BTE\_CONTEXT\_END<sub>i</sub>[15:4] X. In that case it is reset to BTE\_CONTEXT\_START<sub>i</sub>[15:7] X and SY<sub>i</sub> is incremented. SY<sub>i</sub> is reset when it reaches the bottom-right corner of the context.

Therefore, the BTE can only translate raster accesses:

- Image data must be provided line by line, starting from the top-left corner of the context.
- A line stride of 64KB is expected.
- The maximum supported image height is 8k lines.

Figure 8-79 is an example of an active context. There is only one location in an active context where an access from OCPI is expected (green). Accesses to any other location trigger an IRQ\_CTXx\_INVALID event. Reads to contexts in write mode (BTE\_CONTEXT\_CTRL<sub>i</sub>[7:6] MODE = 0) or writes to contexts in read mode (BTE\_CONTEXT\_CTRL<sub>i</sub>[7:6] MODE = 1) trigger an IRQ\_CTXx\_INVALID event. When an access to an expected location is received but the burst length exceeds the context end, an IRQ\_CTXx\_INVALID event is triggered.

**Figure 8-79. ISS BTE Expected Access Locations in the Virtual Space**



camss-322

Some alignment constraints are summarized as follows and are shown in Figure 8-80. They apply only to translated accesses.

- The context start addresses are aligned on 16-byte boundaries.
- The context width is aligned on 16-byte boundaries.
- The OCPI initiator must not use BYTEEN to qualify subwords in the middle of lines. The BTE interprets each BYTEEN as 0xFFFF. Access of partial words triggers an IRQ\_CTXx\_INVALID error.
- The line length sent by the initiator can be 1-byte aligned. The initiator can use BYTEEN to qualify valid data for the last access of the line. The BTE treats those accesses as full 16-byte writes and writes 0 data into its local buffers to complete the access. Those 0s are forwarded to OCPO when burst translation is performed. Software must ensure this data is discarded when the buffer is read back.
- The BTE does not impose any specific constraint on vertical alignment on the TILER tile and subtile grids. However, it is strongly recommended to configure the burst generation in such a way the requests performed to the TILER do not cross tile or subtile boundaries. For more information, see Section 8.2.7.3.2.5, *ISS BTE TILER Space Accesses: 2D Burst Generation*.
- The BTE accepts only one outstanding transaction per OCP tag on the OCPI port. When an initiator tries to generate an OCP request on a tag ID that is already used (no response has been returned on

OCPI), the OCPI port is stalled.

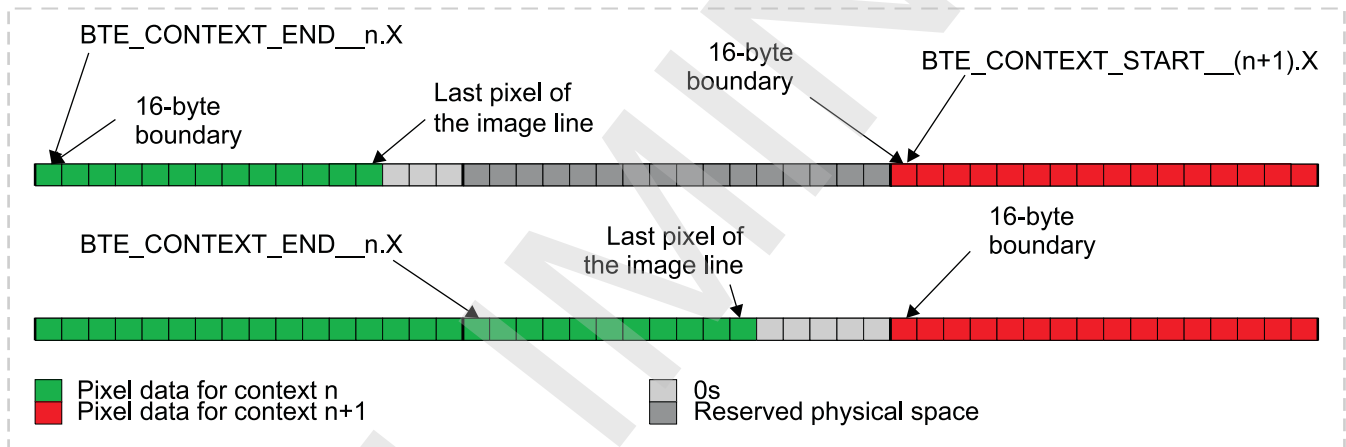
- The amount of memory allocated for each context must be a multiple of 512 bytes. Software must leave  $\text{mod}(\text{BTE\_CONTEXT\_END}_i \text{ BTE\_CONTEXT\_START}_i + 1, 8)$  unused locations between the end of the context and the start of the next one. No accesses to this space from OCPI are allowed, but the BTE can use it to store data.
- While programming a BTE context, the BTE context End X value of all contexts must be less than  $\text{MEMORY}/64$  in 128-bit address (equals  $\text{MEMORY}/4$  in byte address)

**NOTE:** The generic BTE MEMORY size can be found by reading the BTE\_HL\_HWINFO[18:0] MEMORY bit field.

- While reading from a BTE-translated location, the CCP2 read burst size must be greater than or equal to ( $\geq$ ) 16 bytes (128 bits) for correct operation. Bus width translation is performed in the ISS. Therefore, 64-bit accesses lead to 128-bit accesses with  $\text{BYTEEN} = 0x00FF$ . On the BTE side,  $\text{BYTEEN}$  is always processed as  $0xFFFF$ ; thus, access to a BTE-translated 128-bit location is not correct if software uses 64-bit access.

Figure 8-80 shows the BTE context alignment constraints.

**Figure 8-80. ISS BTE Context Alignment Constraints**



camss-323

### 8.2.7.3.2.3 ISS BTE TILER Context Configuration Example

Global configuration and context configuration must be done before traffic from the master can be enabled.

This section provides a configuration example used to write YUV4:2:0 data into a 90-degree rotated buffer ( $S = 1, Y = 1, X = 0$ ). The YUV frame is made up of two objects: Y data (8 bits) and UV data (16 bits).

The BTE provides data orientation and format information to the TILER using a 33-bit address. It can be controlled by software using the  $\text{BTE\_CONTEXT\_BASE}_i$  and  $\text{BTE\_CONTEXT\_CTRL}_i[12]$  ADDR32 bits. ADDR32 must be set to 1. Bits [31:27] of the address control the accessed view format. Table 8-229 is an example of a BTE TILER context configuration.

Figure 8-81 shows BTE TILER mode addressing in 90- or 270-degree orientation.

**Table 8-229. ISS BTE TILER Context Configuration Example**

32	31	30	29	28	27	26 ... 4	3 ... 0
T	Orientation			Mode		Virtual Address	
1	S	/Y	/X	M1	M0	A26 ... A4	0

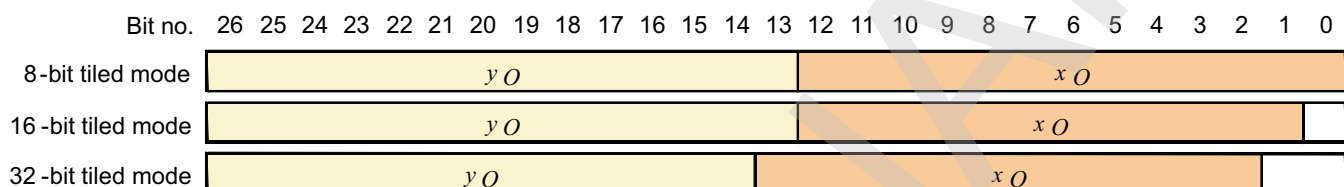
For the example described here, the following settings are used:

90-degree view, 8-bit data      ADDR[32:27] = 0b1 110 00  
 90-degree view, 16-bit data    ADDR[32:27] = 0b1 110 01

Software must also configure the [BTE\\_CONTEXT\\_CTRL\\_i\[9:8\]](#) GRID bit field to match the view and format set by the base address:

90-degree view, 8-bit data      GRID = 1    Stride = 8k  
 90-degree view, 16-bit data    GRID = 1    Stride = 8k

**Figure 8-81. ISS BTE TILER Mode Addressing in 90- or 270-Degree Orientation**



camss-701

Bits [26:0] of the [BTE\\_CONTEXT\\_BASE\\_i](#) register are used to address a pixel in the virtual space of TILER. It must point to the top-left corner of the 2D object.

Software must also configure the other context register before it can start it by setting the [BTE\\_CONTEXT\\_CTRL\\_i\[0\]](#) START bit.

#### 8.2.7.3.2.4 ISS BTE Local Memory Management

##### 8.2.7.3.2.4.1 ISS BTE Introduction

The amount of data buffered is defined by the [BTE\\_CONTEXT\\_CTRL\\_i\[29:16\]](#) TRIGGER bit field. In write mode, translated 2D writes to OCPO are issued when the buffer fill level is greater than or equal to the value of the [BTE\\_CONTEXT\\_CTRL\\_i\[29:16\]](#) TRIGGER bit field. In read mode, translated 2D reads are sent to OCPO when the buffer level is less than the value of the [BTE\\_CONTEXT\\_CTRL\\_i\[29:16\]](#) TRIGGER bit field.

##### 8.2.7.3.2.4.2 ISS BTE Buffer Flushing

###### 8.2.7.3.2.4.2.1 ISS BTE One-Shot Mode

One-shot mode is enabled by setting the [BTE\\_CONTEXT\\_CTRL\\_i\[10\]](#) ONESHOT bit.

During normal operation, a data transfer from local memory to the TILER is automatically triggered when sufficient data is available in the buffer.

Three lines of data remain in the buffer once the OCPI initiator stops sending data into the context.

The BTE supports two ways to flush data remaining in the buffer:

- The last data (bottom-right corner) in a context is written. The last data is defined by the [BTE\\_CONTEXT\\_END\\_i\[15:4\]](#) X and [BTE\\_CONTEXT\\_END\\_i\[28:16\]](#) Y bit fields.
- Software writes the [BTE\\_CONTEXT\\_CTRL\\_i\[2\]](#) FLUSH bit. This is typically done when the context has been stopped before the full frame has been written.

An autoflush mode controls automatic context flushing when an IDLE request is received. It can be activated by setting the [BTE\\_CONTEXT\\_CTRL\\_i\[11\]](#) AUTOFLUSH bit to 1.

During a context flush, all remaining data in the buffer are written to the TILER.

Buffer flushing has lower priority than burst translation active contexts or forwarding transparent accesses. This prevents performance degradation.

If software wants to abort context operation at a random location, it must disable the context.

**8.2.7.3.2.4.2 ISS BTE Continuous Mode**

When continuous mode is selected (the `BTE_CONTEXT_CTRL_i[10]` ONESHOT bit set to 0), reception of frame *n* pushes the data remaining from frame 1 out of the memory to OCPO. This mode is particularly useful when vertical blanking periods are too short to perform a buffer flush at the end of the frame. Also, it avoids creating traffic peaks due to buffer flushing.

The frame height must be a multiple of eight lines and the `BTE_CONTEXT_CTRL_i[14:13]` INITSY bit field must be 0x0 in continuous mode.

**8.2.7.3.2.4.3 ISS BTE Buffer Prefetch**

The buffer must be prefilled before read requests requiring translation can be accepted. Buffer prefetch starts with the top-left corner of the frame when a context is enabled by setting the `BTE_CONTEXT_CTRL_i[0]` START bit. When enough lines have been prefetched, an `IRQ_CTXx_DONE` event is triggered to inform software that the context is ready to perform request translation. Typically, software enables the data flow requiring translation in response to this event. Context ready for transaction and last transaction can be controlled by `BTE_CONTEXT_START` and `BTE_CONTEXT_END`, respectively.

When a read request requiring translation is received while prefetch is ongoing, an `IRQ_CTXx_ERR` event is triggered. It informs software that the read traffic was enabled too early. The BTE returns 0s to OCPI for the failing request (it does not hold the response until real data is available).

After prefetch completion, when a request requiring translation is received but the required data is missing (the BTE is waiting for a response from OCPO), the BTE delays the response on OCPI until the missing data is received on OCPI.

This behavior avoids stalling the OCPI port for too long (that is, a prefetch of up to 3.25 lines of data). However, it avoids getting errors because of slow OCPI responses.

**8.2.7.3.2.4.4 ISS BTE Bandwidth Limiter**

Translated and transparent traffic has higher priority than prefetch and flushing traffic. However, overall system bandwidth is limited. Requesting too much bandwidth for prefetch and flushing traffic may increase latencies for higher priority traffic. That could affect higher priority traffic.

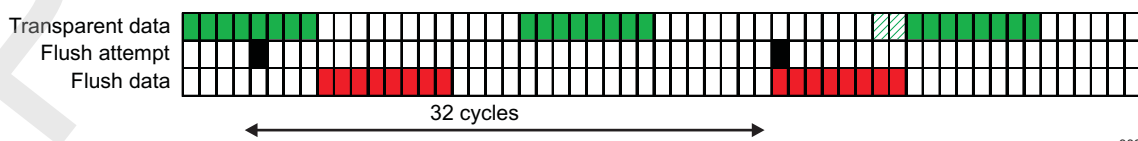
Software can limit the speed of prefetching and buffer flushing by using the `BTE_CTRL[31:22]` `BW_LIMITER` bit field. Typically, this register is used to avoid the buffer prefetch and flush traffic using all the available system bandwidth. This register does not slow down the translated or transparent traffic.

The example in [Figure 8-82](#) assumes:

- 200-MHz functional clock
- 800 Mbps of transparent traffic = one 8 × 128-bit burst every 32 cycles

Without the bandwidth limiter, prefetch and flush traffic may use up to 3.2 0.8 = 2.4Gbps. Using `BTE_CTRL[31:22]` `BW_LIMITER` = 24 ensures that at a maximum one flush/prefetch request is issued every 32 cycles. A prefetch/flush request may be delayed by higher priority traffic of OCP port stalls. To avoid excessive traffic slowdown, the BTE tries to catch up by requesting the next flush/prefetch transaction earlier.

**Figure 8-82. BTE Bandwidth Limiter Example**



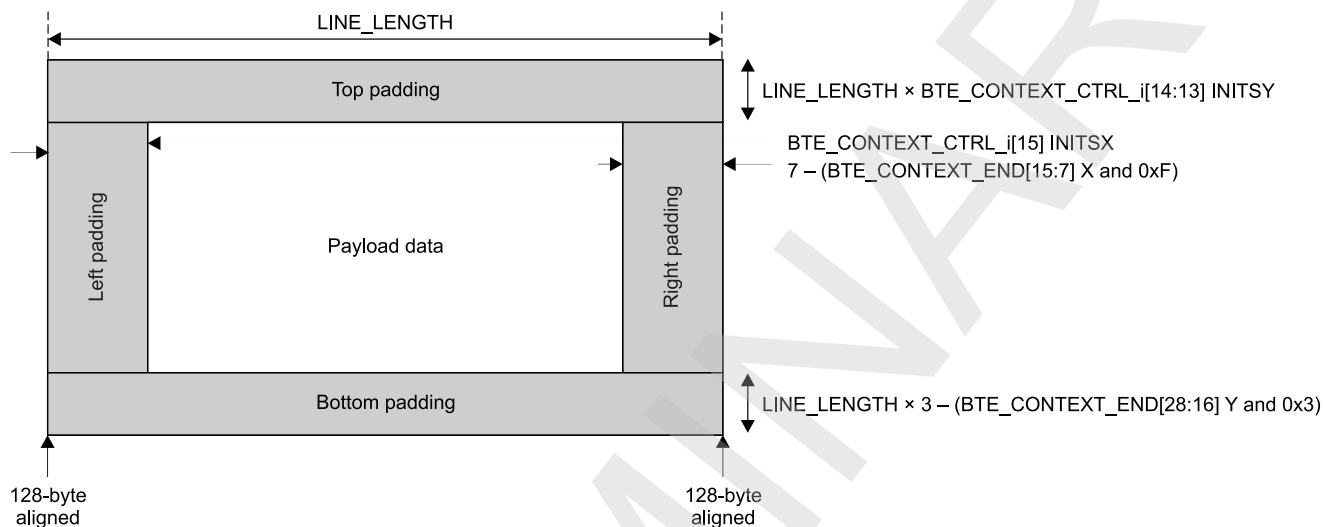
camss-332

### 8.2.7.3.2.5 ISS BTE TILER Space Accesses: 2D Burst Generation

#### 8.2.7.3.2.5.1 ISS BTE Buffer Fill Level

The BTE maintains an internal counter to keep track of the buffer fill level. This internal counter is used to detect when a translated OCP request is sent to the OCPO port. The counter also accounts for padding data. Figure 8-83 shows the BTE buffer fill-level padding.

**Figure 8-83. ISS BTE Buffer Fill-Level Padding**



camss-333

$$\text{LINE\_LENGTH} = \text{BTE\_CONTEXT\_END\_i}[15:4] \times \&0xFF8 - (\text{BTE\_CONTEXT\_START\_i}[15:7] \times \&0xFF8) + 8 \text{ words of 16 bytes.}$$

Figure 8-83 is a visual representation of the following explanation about how read or write events are triggered determined by the internal buffer size.

When context is started, the buffer fill level is initialized to  $\text{LINE\_LENGTH} \times \text{BTE\_CONTEXT\_CTRL\_i}[14:13] \text{ INITSY} + \text{BTE\_CONTEXT\_CTRL\_i}[15] \text{ INITSX}$  (see the previous equation for the  $\text{LINE\_LENGTH}$  calculation). If the BTE reads or writes, the level of the internal buffer is incremented or decremented, respectively, by the burst size. Here, only full 16-byte accesses are performed. Other OCP BYTEEN patterns are forced to 0xFF.

The  $\text{BTE\_CONTEXT\_CTRL\_i}[29:16] \text{ TRIGGER}$  bit field triggers a buffer level read or write even. In write mode, if flushing, the data level is greater than 0; if not flushing, the BTE translates to OCPO when the level is greater than or equal to the value set by the register. If the level is smaller than the  $\text{BTE\_CONTEXT\_CTRL\_i}[29:16] \text{ TRIGGER}$  software setup, then the BTE reads data.

#### 8.2.7.3.2.5.2 ISS BTE OCP Request Generation

Except on borders:

- TILER bursts are 32 bytes  $\times$  4 lines of data blocks.
- Bursts are aligned on subtile boundaries.

The BTE maintains an internal 2D pointer ( $\text{DX\_i}$ ,  $\text{DY\_i}$ ) that corresponds to the top-left corner of the next access to be issued to the TILER. It is initialized to (0, 0) when a context is enabled or wraps around. It is updated each time an access to the TILER is performed.

Table 8-230 lists all supported TILER formats and views.



**Table 8-230. ISS BTE Supported TILER Formats and Views**

Modes	View			Description	OFST	Subtile Grid		1KB Tile Grid		X		Y		GRID	Subtile Aligned		Tile Aligned	
	S	/Y	/X			X	Y	X	Y	LSB	MSB	LSB	MSB		X	Y	X	Y
8-bit	0	0	0	0-degree view	16384	4	4	32	32	0	13	14	26	0	[1:0]	[15:14]	[4:0]	[18:14]
	0	0	1	0-degree view with vertical mirror														
	0	1	0	0-degree view with horizontal mirror														
	0	1	1	180-degree view														
	1	0	0	90-degree view with vertical mirror	8192	4	4	32	32	0	12	13	26	1	[1:0]	[14:13]	[4:0]	[17:13]
	1	0	1	270-degree view														
	1	1	0	90-degree view														
	1	1	1	90-degree view with horizontal mirror														
16-bit	0	0	0	0-degree view	32768	8	2	64	16	0	14	15	26	2	[2:0]	[15]	[5:0]	[18:15]
	0	0	1	0-degree view with vertical mirror														
	0	1	0	0-degree view with horizontal mirror														
	0	1	1	180-degree view														
	1	0	0	90-degree view with vertical mirror	8192	4	4	32	32	0	12	13	26	1	[1:0]	[14:13]	[4:0]	[17:13]
	1	0	1	270-degree view														
	1	1	0	90-degree view														
	1	1	1	90-degree view with horizontal mirror														
32-bit	0	0	0	0-degree view	32768	8	2	64	16	0	14	15	26	2	[2:0]	[15]	[5:0]	[18:15]
	0	0	1	0-degree view with vertical mirror														
	0	1	0	0-degree view with horizontal mirror														
	0	1	1	180-degree view														
	1	0	0	90-degree view with vertical mirror	16384	8	2	64	16	0	13	14	26	3	[2:0]	[14]	[5:0]	[17:14]
	1	0	1	270-degree view														
	1	1	0	90-degree view														
	1	1	1	90-degree view with horizontal mirror														

The `BTE_CONTEXT_BASE_i` and `BTE_CONTEXT_CTRL_i[9:8]` GRID registers must be configured by software to choose the format and view. The format must match the data stored into the virtual space. The choice of the view depends on the desired behavior.

The GRID bit field controls the used OCP stride and OCP address generation. The OCP address is generated using the following formula:

$$\text{OCP\_ADDR} = \text{BTE\_CONTEXT\_BASE}_n + \text{DX}_x + \text{DY}_x \ll \text{Y\_LSB}$$

`Y_LSB` corresponds to the Y LSB column of [Table 8-230](#).

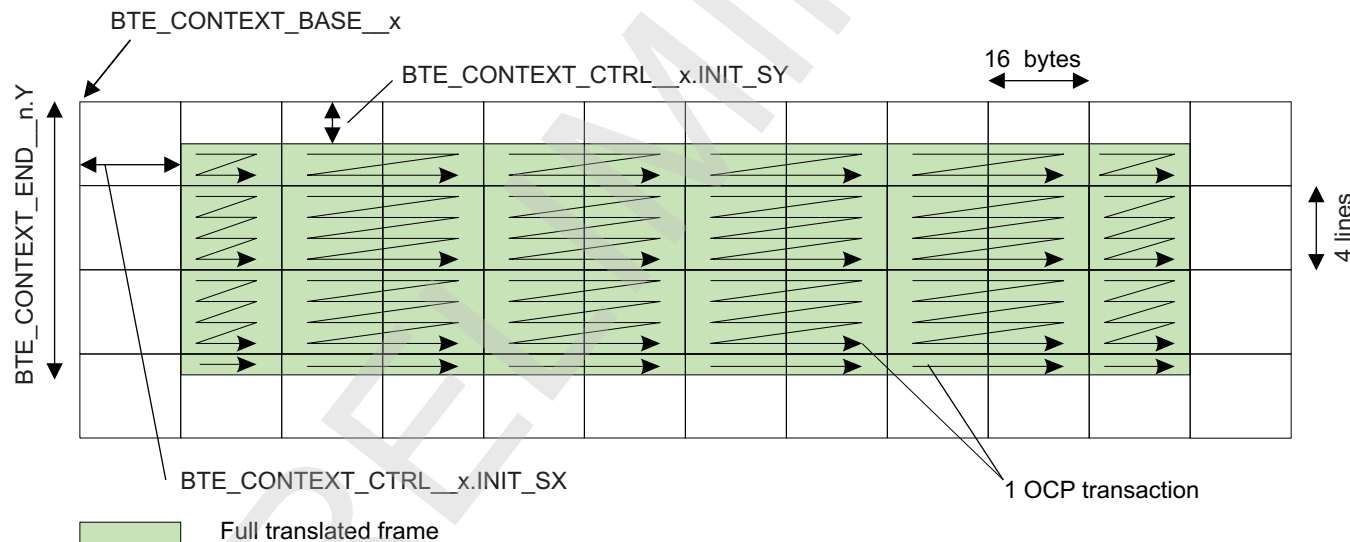
The OCP stride corresponds to the `OFST` column of [Table 8-230](#). The tile and subtile grid information is currently not used by hardware.

The BTE does not automatically realign 2D bursts on tile or subtile boundaries. However, software can force the  $(\text{SX}_i, \text{SY}_i)$  (see [Section 8.2.7.3.2.4, ISS BTE Local Memory Management](#), for the definition of  $\text{SY}_i$ ) reset value to be used. It is set through the `BTE_CONTEXT_CTRL_i[15]` `INITSX` bit and the `BTE_CONTEXT_CTRL_i[14:13]` `INITSY` bit field.

The length and height of the 2D burst is adapted by the BTE to avoid sending dummy data to the TILER.

[Figure 8-84](#) is an example of BTE 2D burst generation. The vertical start and end of the full 2D frame are not vertically aligned on the grid.

**Figure 8-84. ISS BTE 2D Burst Generation**



camss-336

### 8.2.7.3.2.6 ISS BTE Posted and Nonposted Write Support

The BTE can handle posted and nonposted writes received on OCPI. Normally, only nonposted writes should be used.

For transparent accesses, posted and nonposted writes are forwarded to OCPO. The response provided by OCPO is returned to OCPI.

For translated accesses, the response to posted and nonposted writes is provided by the BTE. It does not wait for the response of the translated request that is sent to the TILER. In other words, the BTE has no true nonposted write support for translated accesses. The BTE does not ensure that the data has effectively been written to the destination memory when it returns the response to a nonposted translated write.

True nonposted write support is ensured for transparent accesses. Only nonposted writes must be used. Select nonposted write mode through the [BTE\\_CTRL\[5\] POSTED](#) bit.

#### **8.2.7.3.2.7 ISS BTE Error Reporting**

Unexpected accesses are flagged using interrupts. Also, when an SResp = ERR is received on OCPO, an interrupt is triggered. If the response corresponds to a transparent access, it is forwarded to OCPI.

The BTE is not an OCP checker: It expects only valid and supported transactions from the external world.

Also, it is not intended to detect all types of software errors; few cases are detected. Those cases are described in the functional description sections.

#### **8.2.7.3.2.8 ISS BTE Interrupts**

All events generated by the module are merged into a single event at ISS level. This event can be enabled from ISS level by the [ISS\\_HL\\_IRQENABLE\\_SET\\_i\[11\] BTE\\_IRQ](#) bit. [Table 8-231](#) lists the BTE interrupt events.

**Table 8-231. ISS BTE Interrupt Events**

Event	Description
IRQ_OCP_ERR	OCP error received from OCPO master port
IRQ_INVALID	An access to a location that is not mapped to any context has been performed. For more information, see <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
IRQ_CTXx_DONE	Context has been fully transferred to the TILER. This interrupt is triggered when flushing completes in one-shot mode. It is triggered once per frame in continuous mode.
IRQ_CTXx_INVALID	Unexpected address sequence or access direction (read of a context in write mode or write of a context in read mode). For more information, see <a href="#">Section 8.2.7.3.2.2, ISS BTE Virtual Address Space and Context Mapping</a> .
IRQ_CTXx_ERR	Can occur only when a context is configured in read mode. This request triggers when a read request is received but insufficient data is buffered to perform the translation. For more information, see <a href="#">Section 8.2.7.3.2.4.3, ISS BTE Buffer Prefetch</a> .

### 8.2.7.3.2.9 ISS BTE Debug Support

The BTE has no specific debug support.

## 8.2.7.4 ISS BTE Programming Model

### 8.2.7.4.1 ISS BTE Reset

The BTE can accept a general software reset, propagated through all the hierarchy. This reset can be done to initialize the module and has the same effect as the hardware reset.

1. Set the [BTE\\_HL\\_SYSCONFIG\[0\] SOFTRESET](#) bit to 1.
2. Read the [BTE\\_HL\\_SYSCONFIG\[0\] SOFTRESET](#) bit to check whether it is set to 1, which means the reset occurred.

If after five reads, the [BTE\\_HL\\_SYSCONFIG\[0\] SOFTRESET](#) bit still returns 0, assume that an error occurred during the reset stage.

Programmers must not set the [BTE\\_HL\\_SYSCONFIG\[0\] SOFTRESET](#) bit to 1 if the BTE is integrated in a subsystem; it is safer to use the software reset at subsystem level.

---

**NOTE:** A software reset does not reset the IDLE protocol signals.

---

### 8.2.7.4.2 ISS BTE Interrupts

All events are mapped to a single interrupt output, BTE\_IRQ. [Table 8-232](#) lists the procedure to configure or manage the BTE interrupts.

**Table 8-232. ISS BTE Configure/Manage Interrupts**

Step	Bit Field	Value
Each event that generates an interrupt can be individually enabled by setting the appropriate bit.	<a href="#">BTE_HL_IRQENABLE_SET</a>	
Each event that generates an interrupt can be individually disabled by setting the adequate bit.	<a href="#">BTE_HL_IRQENABLE_CLR</a>	
When an event occurs, the corresponding bit in the <a href="#">BTE_HL_IRQSTATUS_RAW</a> register is set, regardless of whether or not the event is enabled. Bits in the <a href="#">BTE_HL_IRQSTATUS</a> registers are set only when an enabled event occurs.	<a href="#">BTE_HL_IRQSTATUS_RAW</a> and <a href="#">BTE_HL_IRQSTATUS</a>	
Software can clear a pending event by setting the adequate bit in the <a href="#">BTE_HL_IRQSTATUS</a> register.	<a href="#">BTE_HL_IRQSTATUS</a>	

### 8.2.7.4.3 ISS BTE Context Configuration

Global configuration and context configuration must be done before traffic from the master can be enabled.

This section provides a configuration example used to write YUV4:2:0 data into a 90-degree rotated buffer (S = 1, Y = 1, X = 0). The YUV frame consists of two objects: Y data (8 bits) and UV data (16 bits).

The BTE provides data orientation and format information to the TILER using a 33-bit address. It can be controlled by software using the [BTE\\_CONTEXT\\_BASE\\_i](#) and [BTE\\_CONTEXT\\_CTRL\\_i\[12\]](#) ADDR32 registers.

ADDR32 must be set to 1. Bits [31:27] of the address control the accessed view and data format. [Table 8-233](#) gives the format of the TILER address. [Figure 8-85](#) shows BTE tiled mode addressing in 90- or 270-degree orientation.

**Table 8-. TILER Address Format**

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
T		Orientation		Mode		Virtual Address																																
1		S > ×		M1 M0		A26 ... A4																															0	

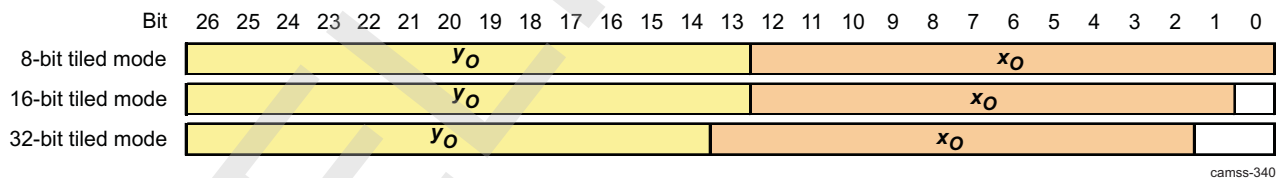
For the example described here, the following settings are used.

- 90-degree view, 8-bit data      ADDR[32:27] = 0b1 110 00
- 90-degree view, 16-bit data    ADDR[32:27] = 0b1 110 01

Software must also configure the [BTE\\_CONTEXT\\_CTRL\\_i\[9:8\]](#) GRID bit field to match the view and format set by the base address:

- 90-degree view, 8-bit data      GRID = 1    Stride = 8k
- 90-degree view, 16-bit data    GRID = 1    Stride = 8k

**Figure 8-85. ISS BTE Tiled Mode Addressing in 90- or 270-Degree Orientation (S = 1)**



Bits [26:0] of the [BTE\\_CONTEXT\\_BASE\\_i](#) register are used to address a pixel in the virtual space of the TILER. It must point to the top-left corner of the 2D object.

Software must also configure the other context registers before it can start it by setting the [BTE\\_CONTEXT\\_CTRL\\_i\[0\]](#) START bit.

### 8.2.7.4.4 ISS BTE Change Context Configuration

All contexts operate independently. Software can change the configuration of an inactive context while other contexts are active and perform request translation.

When software must change the configuration of an active context, it must follow the sequence in [Table 8-234](#).

**Table 8-234. ISS BTE Change Context Configuration**

Step	Bit Field	Value
1		Ensure that the initiator does not send any more data to this context until it is re-enabled.

**Table 8-234. ISS BTE Change Context Configuration (continued)**

Step	Bit Field	Value
Disable the context. <ul style="list-style-type: none"> <li>The STOP condition is considered on a valid OCP boundary.</li> <li>It preserves the internal states so that buffer flushing can be done.</li> <li>The BTE no longer translates requests received for this context. Any transactions received for an inactive context trigger error interrupts.</li> <li>The BTE completes all outstanding transactions on OCPO.</li> </ul>	<a href="#">BTE_CONTEXT_CTRL_i[1]</a> STOP	0x1
Flush all remaining data for the context, if needed. If software simply wants to abort the transfer to recover from some error condition, flushing is not needed. The context is automatically reset when it is enabled again.	<a href="#">BTE_CONTEXT_CTRL_i[2]</a> FLUSH	0x1
Wait until the context completes pending OCP transaction and buffer flush (if enabled). It sets the <a href="#">IRQ_CTXx_DONE</a> when it becomes idle.	<a href="#">BTE_HL_IRQSTATUS</a> . <a href="#">IRQ_CTXx_DONE</a>	Read 0x0
Change the context configuration.		
Enable the context by setting the <a href="#">BTE_CONTEXT_CTRL_i[0]</a> START bit. Setting the START bit resets the internal state-machine of the context.	<a href="#">BTE_CONTEXT_CTRL_i[0]</a> START	0x1

Alternatively, software can change the context mode to one-shot and wait until the [CTXx\\_DONE\\_IRQ](#) is triggered.

**NOTE:** Once a context is disabled it cannot be resumed simply by writing the START bit. In fact, doing so resets the internal FSM. If data is lost in the buffer, it will be lost.

## 8.2.7.5 ISS BTE Register Manual

### 8.2.7.5.1 ISS BTE Instance Summary

[Table 8-235](#) lists the BTE instance.

**Table 8-235. ISS BTE Instance Summary**

Module Name	L3 Base Address	Size
<a href="#">ISS_BTE</a>	0x5200 2000	512 bytes

### 8.2.7.5.2 ISS BTE Registers

#### 8.2.7.5.2.1 ISS BTE Register Summary

[Table 8-236](#) summarizes the BTE registers.

**Table 8-236. ISS BTE Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_BTE Base Address
<a href="#">BTE_HL_REVISION</a>	R	32	0x0000 0000	0x5200 2000
<a href="#">BTE_HL_HWINFO</a>	R	32	0x0000 0004	0x5200 2004
<a href="#">BTE_HL_SYSCONFIG</a>	RW	32	0x0000 0010	0x5200 2010
RESERVED	RW	32	0x0000 001C	0x5200 201C

**Table 8-236. ISS BTE Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_BTE Base Address
<a href="#">BTE_HL_IRQSTATUS_RAW</a>	RW	32	0x0000 0020	0x5200 2020
<a href="#">BTE_HL_IRQSTATUS</a>	RW	32	0x0000 0024	0x5200 2024
<a href="#">BTE_HL_IRQENABLE_SET</a>	RW	32	0x0000 0028	0x5200 2028
<a href="#">BTE_HL_IRQENABLE_CLR</a>	RW	32	0x0000 002C	0x5200 202C
<a href="#">BTE_CTRL</a>	RW	32	0x0000 0030	0x5200 2030
<a href="#">BTE_CTRL1</a>	RW	32	0x0000 0034	0x5200 2034
<a href="#">BTE_CONTEXT_CTRL_i<sup>(1)</sup></a>	RW	32	0x0000 0040 + (0x20 * i)	0x5200 2040 + (0x20 * i)
<a href="#">BTE_CONTEXT_BASE_i<sup>(1)</sup></a>	RW	32	0x0000 0044 + (0x20 * i)	0x5200 2044 + (0x20 * i)
<a href="#">BTE_CONTEXT_STAR_T_j<sup>(1)</sup></a>	RW	32	0x0000 0048 + (0x20 * i)	0x5200 2048 + (0x20 * i)
<a href="#">BTE_CONTEXT_END_i<sup>(2)</sup></a>	RW	32	0x0000 004C + (0x20 * i)	0x5200 204C + (0x20 * i)

<sup>(1)</sup> i = 0 to 7

<sup>(2)</sup> i = 0 to 7

**8.2.7.5.2.2 ISS BTE Register Description**

Table 8-237 through Table 8-261 describe the BTE registers.

**Table 8-237. BTE\_HL\_REVISION**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .	<b>Instance</b> ISS_BTE
<b>Description</b>	IP revision identifier (X.Y.R) Used by software to track features, bugs, and compatibility	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 8-238. Register Call Summary for Register BTE\_HL\_REVISION**

ISS Interfaces

- [ISS BTE Register Manual: \[0\]](#)

**Table 8-239. BTE\_HL\_HWINFO**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	ISS_BTE
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .		
<b>Description</b>	Information about the hardware configuration of the IP module; that is, typically, the HDL generics (if any) of the module.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESPFIFO	CONTEXTS	RESERVED																					

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:21	RESPFIFO	Response FIFO size Read 0x3: 64 x 128 bits Read 0x4: 128 x 128 bits Read 0x2: 32 x 128 bits Read 0x0: Reserved Read 0x6: Reserved Read 0x1: 16 x 128 bits Read 0x7: Reserved Read 0x5: Reserved	R	0x2
20:19	CONTEXTS	Number of contexts Read 0x3: Reserved Read 0x2: 8 contexts Read 0x1: 4 contexts Read 0x0: 2 contexts	R	0x1
18:0	RESERVED	Reserved	R	0x056A0

**Table 8-240. Register Call Summary for Register BTE\_HL\_HWINFO**

ISS Interfaces

- [ISS BTE Register Manual: \[0\]](#)

**Table 8-241. BTE\_HL\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	ISS_BTE
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .		
<b>Description</b>	Clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEMODE	RESERVED	SOFTRESET													



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.  0x0: An IDLE request is acknowledged unconditionally 0x1: An IDLE request is never acknowledged 0x3: Reserved. Do not use 0x2: Smart-idle mode. Acknowledgment to an IDLE request is given based on the internal activity of the module.	RW	0x2
1	RESERVED		R	0
0	SOFTRESET	Software reset.  Write 0x0: No action Write 0x1: Initiate software reset Read 0x1: Reset (software or other) ongoing Read 0x0: Reset done, no pending action	RW	0

**Table 8-242. Register Call Summary for Register BTE\_HL\_SYSCONFIG**

ISS Interfaces

- [ISS BTE Programming Model: \[0\] \[1\]](#)
- [ISS BTE Register Manual: \[2\]](#)

**Table 8-243. BTE\_HL\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	ISS_BTE
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .		
<b>Description</b>	Per-event raw interrupt status vector. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_ERR	IRQ_CTX6_ERR	IRQ_CTX5_ERR	IRQ_CTX4_ERR	IRQ_CTX3_ERR	IRQ_CTX2_ERR	IRQ_CTX1_ERR	IRQ_CTX0_ERR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_DONE	IRQ_CTX6_DONE	IRQ_CTX5_DONE	IRQ_CTX4_DONE	IRQ_CTX3_DONE	IRQ_CTX2_DONE	IRQ_CTX1_DONE	IRQ_CTX0_DONE	RESERVED					IRQ_INVALID	IRQ_OCP_ERR	

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_ERR	Read request received before sufficient data has been prefetched.  Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
30	IRQ_CTX6_ERR	Read request received before sufficient data has been prefetched.  Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
29	IRQ_CTX5_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
28	IRQ_CTX4_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
27	IRQ_CTX3_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
26	IRQ_CTX2_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
25	IRQ_CTX1_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
24	IRQ_CTX0_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
15	IRQ_CTX7_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
14	IRQ_CTX6_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
13	IRQ_CTX5_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
12	IRQ_CTX4_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
11	IRQ_CTX3_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
10	IRQ_CTX2_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
9	IRQ_CTX1_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
8	IRQ_CTX0_DONE	Write mode: Context has been fully transferred to the TILER Read mode: Context prefetch has completed. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
7:2	RESERVED		R	0x00
1	IRQ_INVALID	Invalid access to the virtual space Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
0	IRQ_OCP_ERR	OCP error received from OCP master port. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

**Table 8-244. Register Call Summary for Register BTE\_HL\_IRQSTATUS\_RAW**

## ISS Interfaces

- [ISS BTE Programming Model: \[0\] \[1\]](#)
- [ISS BTE Register Manual: \[2\]](#)

**Table 8-245. BTE\_HL\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0024
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .
<b>Instance</b>	ISS_BTE
<b>Description</b>	Per-event "enabled" interrupt status vector. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_ERR	IRQ_CTX6_ERR	IRQ_CTX5_ERR	IRQ_CTX4_ERR	IRQ_CTX3_ERR	IRQ_CTX2_ERR	IRQ_CTX1_ERR	IRQ_CTX0_ERR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_DONE	IRQ_CTX6_DONE	IRQ_CTX5_DONE	IRQ_CTX4_DONE	IRQ_CTX3_DONE	IRQ_CTX2_DONE	IRQ_CTX1_DONE	IRQ_CTX0_DONE	RESERVED						IRQ_INVALID	IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
30	IRQ_CTX6_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
29	IRQ_CTX5_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
28	IRQ_CTX4_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
27	IRQ_CTX3_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
26	IRQ_CTX2_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
25	IRQ_CTX1_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
24	IRQ_CTX0_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
15	IRQ_CTX7_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
14	IRQ_CTX6_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
13	IRQ_CTX5_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
12	IRQ_CTX4_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
11	IRQ_CTX3_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
10	IRQ_CTX2_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
9	IRQ_CTX1_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_DONE	Write mode: Context has been fully transferred to the TILER Read mode: Context prefetch has completed. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
7:2	RESERVED		R	0x00
1	IRQ_INVALID	Invalid access to the virtual space Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
0	IRQ_OCP_ERR	OCF error received from OCF master port. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

**Table 8-246. Register Call Summary for Register BTE\_HL\_IRQSTATUS**

ISS Interfaces

- [ISS BTE Programming Model: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS BTE Register Manual: \[5\]](#)

**Table 8-247. BTE\_HL\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	ISS_BTE
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .		
<b>Description</b>	Per-event interrupt enable bit vector Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_ERR	IRQ_CTX6_ERR	IRQ_CTX5_ERR	IRQ_CTX4_ERR	IRQ_CTX3_ERR	IRQ_CTX2_ERR	IRQ_CTX1_ERR	IRQ_CTX0_ERR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_DONE	IRQ_CTX6_DONE	IRQ_CTX5_DONE	IRQ_CTX4_DONE	IRQ_CTX3_DONE	IRQ_CTX2_DONE	IRQ_CTX1_DONE	IRQ_CTX0_DONE	RESERVED					IRQ_INVALID	IRQ_OCP_ERR	

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0



Bits	Field Name	Description	Type	Reset
30	IRQ_CTX6_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
29	IRQ_CTX5_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
28	IRQ_CTX4_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
27	IRQ_CTX3_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
26	IRQ_CTX2_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
25	IRQ_CTX1_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
24	IRQ_CTX0_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
15	IRQ_CTX7_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
14	IRQ_CTX6_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
13	IRQ_CTX5_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
12	IRQ_CTX4_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
11	IRQ_CTX3_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
10	IRQ_CTX2_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
9	IRQ_CTX1_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
8	IRQ_CTX0_DONE	Write mode: Context has been fully transferred to the TILER Read mode: Context prefetch has completed. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
7:2	RESERVED		R	0x00
1	IRQ_INVALID	Invalid access to the virtual space Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
0	IRQ_OCP_ERR	OCP error received from OCP master port. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

**Table 8-248. Register Call Summary for Register BTE\_HL\_IRQENABLE\_SET**

ISS Interfaces

- [ISS BTE Programming Model: \[0\]](#)
- [ISS BTE Register Manual: \[1\]](#)

**Table 8-249. BTE\_HL\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 002C
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .
<b>Instance</b>	ISS_BTE
<b>Description</b>	Per-event interrupt enable bit vector Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_ERR	IRQ_CTX6_ERR	IRQ_CTX5_ERR	IRQ_CTX4_ERR	IRQ_CTX3_ERR	IRQ_CTX2_ERR	IRQ_CTX1_ERR	IRQ_CTX0_ERR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_DONE	IRQ_CTX6_DONE	IRQ_CTX5_DONE	IRQ_CTX4_DONE	IRQ_CTX3_DONE	IRQ_CTX2_DONE	IRQ_CTX1_DONE	IRQ_CTX0_DONE	RESERVED					IRQ_INVALID	IRQ_OCP_ERR	

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
30	IRQ_CTX6_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
29	IRQ_CTX5_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
28	IRQ_CTX4_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
27	IRQ_CTX3_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
26	IRQ_CTX2_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
25	IRQ_CTX1_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
24	IRQ_CTX0_ERR	Read request received before sufficient data has been prefetched. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
15	IRQ_CTX7_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
14	IRQ_CTX6_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
13	IRQ_CTX5_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
12	IRQ_CTX4_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
11	IRQ_CTX3_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
10	IRQ_CTX2_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
9	IRQ_CTX1_DONE	Context has been fully transferred to the TILER Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_DONE	Write mode: Context has been fully transferred to the TILER Read mode: Context prefetch has completed. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
7:2	RESERVED		R	0x00
1	IRQ_INVALID	Invalid access to the virtual space Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
0	IRQ_OCP_ERR	OCP error received from OCP master port. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

**Table 8-250. Register Call Summary for Register BTE\_HL\_IRQENABLE\_CLR**

ISS Interfaces

- [ISS BTE Programming Model: \[0\]](#)
- [ISS BTE Register Manual: \[1\]](#)

**Table 8-251. BTE\_CTRL**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	ISS_BTE
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .		
<b>Description</b>	BTE control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BW_LIMITER								RESERVED								BASE				RESERVED	POSTED	RESERVED	TAG_CNT								

Bits	Field Name	Description	Type	Reset
31:22	BW_LIMITER	Minimum number of OCP cycles between two consecutive buffer flushing or prefetch requests. Used to limit the bandwidth used to fill/empty buffers. 0: Maximum speed. Up to 1 request every 8 cycles (3.2GB @ 200 MHz) 1: Up to 1 request every 9 cycles. 1023: Minimum speed. Up to 1 request every 1031 cycles (24MB @ 200 MHz)	RW	0x000
21:12	RESERVED		R	0x000
11:8	BASE	Base address of the virtual space translated by the BTE. Start address = BASE*512MB End address = (BASE+1)*512MB 1 For example: BASE=3 => 0x 0 6000 0000 - 0x 0 7FFF FFFF	RW	0x0
7:6	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
5	POSTED	Select among posted and nonposted writes for translated requests. 0x0: Use non posted writes 0x1: Use posted writes	RW	0
4	RESERVED		R	0
3:0	TAG_CNT	BTE could use up to TAG_CNT+1 tags on OCPO. There could only be one outstanding request per tag. TAG_CNT does not control the number of requests it could handle on OCPI. This register is internally shadowed. Modifications are taken into account when there are no outstanding transactions on OCPO. TAG ID 0 to TAG_CNT are used on OCPO.	RW	0xF

**Table 8-252. Register Call Summary for Register BTE\_CTRL**

ISS Interfaces

- [ISS BTE Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS BTE Register Manual: \[5\]](#)

**Table 8-253. BTE\_CTRL1**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	ISS_BTE
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .		
<b>Description</b>	BTE control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESP_FIFO_THR															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6:0	RESP_FIFO_THR	The BTE stops accepting new requests from OCPI (on a clean burst boundary) when the response FIFO contains more than RESP_FIFO_THR words. The reset value is FIFO_SIZE - 16 - 1. FIFO_SIZE = $8 * 2^{\text{RESP_FIFO}}$	RW	0x0F

**Table 8-254. Register Call Summary for Register BTE\_CTRL1**

ISS Interfaces

- [ISS BTE Register Manual: \[0\]](#)

**Table 8-255. BTE\_CONTEXT\_CTRL\_i**

<b>Address Offset</b>	0x0000 0040 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .	<b>Instance</b>	ISS_BTE
<b>Description</b>	Context control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	TRIGGER																INITSX	INITSY	ADDR32	AUTOFLUSH	ONESHOT	GRID	MODE	RESERVED	FLUSH	STOP	START				



Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	TRIGGER	Threshold used to trigger translated requests to OCPO. Unit: words or 16 bytes Valid range: 3 lines + 2 ... 4 lines WRITE: a 2D write is issued to OCPO when the internal buffer level (including masked accesses) is superior or equal to TRIGGER READ: a 2D read is issued to OCPO when the internal buffer level (including masked accesses) is inferior to TRIGGER.	RW	0x0000
15	INITSX	Reset value to be used for SX__x. Check the section describing the local buffer management for details.	RW	0
14:13	INITSY	Reset value to be used for SY__x. Check the section describing the local buffer management for details.	RW	0x0
12	ADDR32	Controls the value of the OCP address bit 32 to be used for translated accesses	RW	1
11	AUTOFLUSH	Controls automatic context flushing when an IDLE request is received 0x0: Disabled 0x1: Enabled	RW	0
10	ONESHOT	Selects one-shot or continuous mode 0x0: The context is automatically re-enabled when its end is reached. 0x1: The context is disabled when the end of a frame has been reached.	RW	0
9:8	GRID	Grid used to access the TILER 0x0: Stride = 16k Subtile = 4x4 bytes Tile = 32x32 bytes 0x1: Stride = 8k Subtile = 4x4 bytes Tile = 32x32 bytes 0x3: Stride = 16k Subtile = 8x2 bytes Tile = 64x16 bytes 0x2: Stride = 32k Subtile = 8x2 bytes Tile = 64x16 bytes	RW	0x0
7:6	MODE	Select the translation mode for the context 0x0: Write translation 0x1: Read translation 0x3: reserved 0x2: Direct access to local buffer	RW	0x0
5:3	RESERVED		R	0x0
2	FLUSH	Flushes all remaining data of the context to the TILER. Write 0x0: No effect Write 0x1: Flush	W	0
1	STOP	Stops the context on a clean OCP transaction boundary. Write 0x0: No effect Write 0x1: Stop the context	W	0

Bits	Field Name	Description	Type	Reset
0	START	Resets the contexts internal state and enables the context on a clean OCP transaction boundary. Write 0x0: No effect Write 0x1: Reset + Enable	W	0

**Table 8-256. Register Call Summary for Register BTE\_CONTEXT\_CTRL\_i**

## ISS Interfaces

- [ISS BTE Integration: \[0\] \[1\]](#)
- [ISS BTE Functional Description: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [ISS BTE Programming Model: \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\]](#)
- [ISS BTE Register Manual: \[31\]](#)

**Table 8-257. BTE\_CONTEXT\_BASE\_i**

<b>Address Offset</b>	0x0000 0044 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .	<b>Instance</b>	ISS_BTE
<b>Description</b>	Address of the frame buffer in the TILER address space.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	Address	RW	0x00000000
4:0	RESERVED		R	0x00

**Table 8-258. Register Call Summary for Register BTE\_CONTEXT\_BASE\_i**

## ISS Interfaces

- [ISS BTE Functional Description: \[0\] \[1\] \[2\]](#)
- [ISS BTE Programming Model: \[3\] \[4\]](#)
- [ISS BTE Register Manual: \[5\]](#)

**Table 8-259. BTE\_CONTEXT\_START\_i**

<b>Address Offset</b>	0x0000 0048 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .	<b>Instance</b>	ISS_BTE
<b>Description</b>	Top-left corner of the context.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED																X												RESERVED											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:7	X	Address, in 128-byte words	RW	0x000
6:0	RESERVED		R	0x00

**Table 8-260. Register Call Summary for Register BTE\_CONTEXT\_START\_i**

## ISS Interfaces

- [ISS BTE Functional Description: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS BTE Register Manual: \[4\]](#)

**Table 8-261. BTE\_CONTEXT\_END\_i**

<b>Address Offset</b>	0x0000 004C + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-236</a> .	<b>Instance</b>	ISS_BTE
<b>Description</b>	Bottom-right corner of the context.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				Y												X								RESERVED							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	Y	Last line number for the context (0 corresponds to a context of 1 line)	RW	0x0000
15:4	X	Address, in 128-bit words, of the last column of the context	RW	0x000
3:0	RESERVED		R	0x0

**Table 8-262. Register Call Summary for Register BTE\_CONTEXT\_END\_i**

ISS Interfaces

- [ISS BTE Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS BTE Register Manual: \[5\]](#)

## 8.2.8 ISS CBUFF

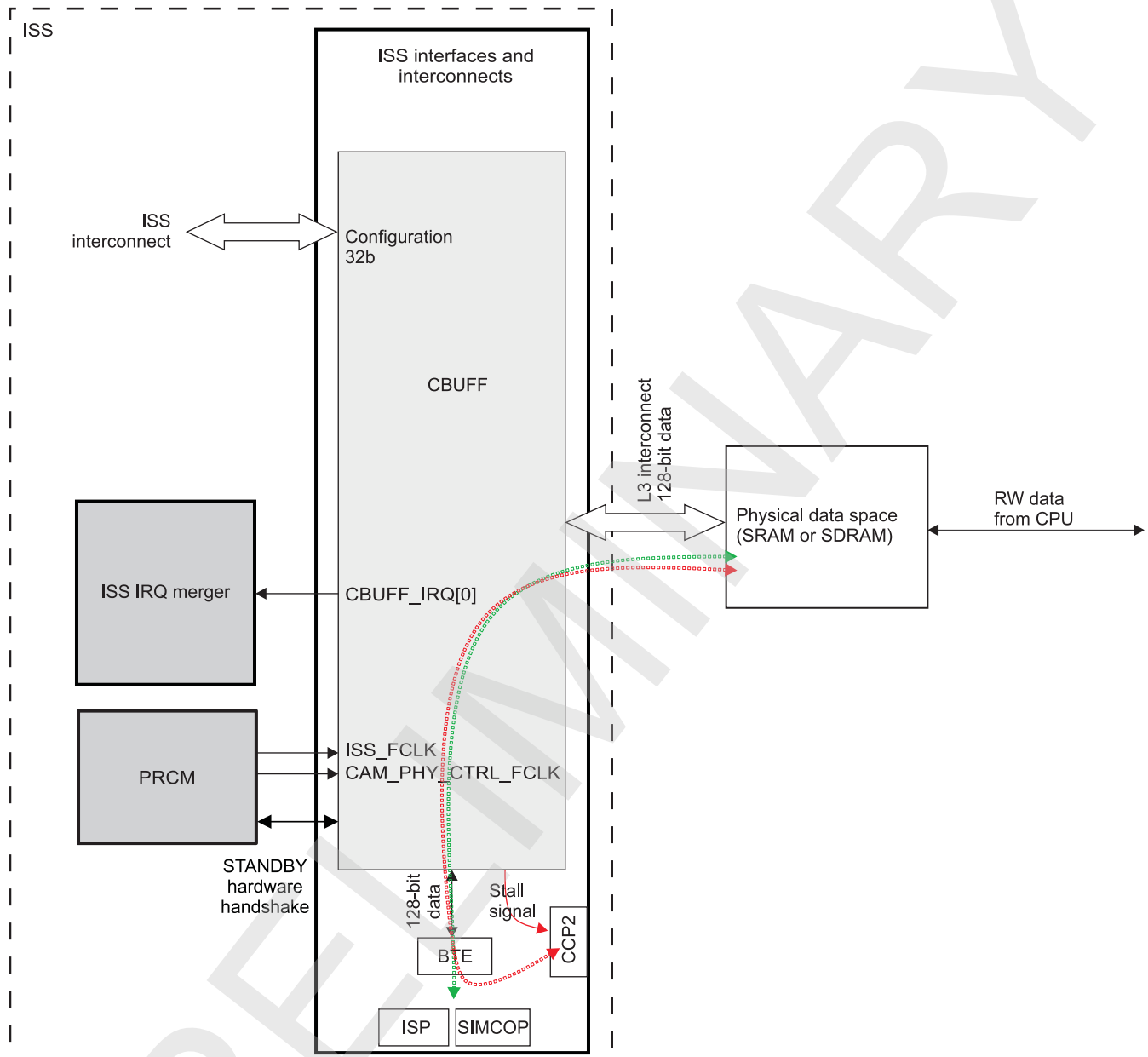
### 8.2.8.1 ISS CBUFF Environment

There are no particular environment attributes. See [Section 8.2.2, ISS Interfaces Environment](#).

### 8.2.8.2 ISS CBUFF Integration

[Figure 8-86](#) shows the integration of the CBUFF in the ISS. Because the CBUFF maps a virtual memory space from the physical memory, it therefore communicates with the ISS ISP and ISS interface modules for data to and from memory. This figure shows the normal data flow for further processing by the ISP and/or still-image coprocessor (SIMCOP) (in green) and the stall functionality of CBUFF (in red), which stall the data for a certain amount of time. An example of stalling is when the ISP processes the data faster than the input from CCP2 to memory. The CBUFF then must stall the data flow until a sufficient amount of data can be read from memory by the ISP.

Figure 8-86. ISS CBUFF Integration



RED - Stalled data flow  
 GREEN - Normal data flow

camss-679

For power domain, clocks, reset, and hardware requests, see [Section 8.1.2.4, ISS Power Management](#).

### 8.2.8.2.1 ISS CBUFF PRCM Interface

#### 8.2.8.2.1.1 ISS CBUFF Reset and Idle Mechanism

A reset signal is provided by the PRCM module to the top-level ISS power and clock-management module.

For standby, when none of the ISS modules require CBUFF execution actions, ISS PM executes the standby sequence, which reaches the CBUFF. When an IDLE request arrives, the CBUFF data is drained. After all transactions are drained, the CBUFF acknowledges the IDLE request to the ISS PM/CM. The IDLE request/acknowledge steps are:

When an IDLE request is received in smart-idle mode:

1. CBUFF stops accepting any new OCP requests on a clean OCP transaction boundary.
2. Waits until the interrupt output becomes inactive (no more enabled event pending)
3. Waits until all outstanding OCP transactions are complete on OCPO and OCPI. CBUFF keeps track of issued requests, data phases, and responses for that purpose.
4. CBUFF disconnects the OCPO port.
5. CBUFF acknowledges the IDLE request.

When a wake-up request is received in smart-idle mode:

1. CBUFF connects the OCPO port.
2. Starts accepting new requests from OCPI
3. Acknowledges the functional state

Idle mode is controlled through the [CBUFF\\_HL\\_SYSCONFIG\[3:2\]](#) IDLEMODE bit field. For software reset, it is recommended to use the global ISS reset; if a reset is required, it can be set from the [CBUFF\\_HL\\_SYSCONFIG\[0\]](#) SOFTRESET bit.

### 8.2.8.2.2 ISS CBUFF Interrupts

All events generated by the module are merged into a single event at ISS level. This event can be mapped to the MPU subsystem by enabling the [ISS\\_HL\\_IRQENABLE\\_SET\\_i\[10\]](#) CBUFF\_IRQ bit. [Table 8-263](#) lists the procedure to manage CBUFF interrupts.

**Table 8-263. ISS CBUFF Interrupt Management**

Description	Bit Field	Value
Each event that generates an interrupt can be individually enabled by setting the appropriate bit.	<a href="#">CBUFF_HL_IRQENABLE_SET</a>	
Each event that generates an interrupt can be individually disabled by setting the appropriate bit.	<a href="#">CBUFF_HL_IRQENABLE_CLR</a>	
When an event occurs, the corresponding bit in the <a href="#">CBUFF_HL_IRQSTATUS_RAW</a> register is set regardless of whether or not the event is enabled. Bits in the <a href="#">CBUFF_HL_IRQSTATUS</a> registers are only set only when an enabled event occurs	<a href="#">CBUFF_HL_IRQSTATUS_RAW</a> and <a href="#">CBUFF_HL_IRQSTATUS</a>	
Software can clear a pending event by setting the appropriate bit in the <a href="#">CBUFF_HL_IRQSTATUS</a> register.	<a href="#">CBUFF_HL_IRQSTATUS</a>	

The CBUFF can generate three events per context and one global event. All events are merged into one physical interrupt line. [Table 8-264](#) describes the CBUFF-generated events.

**Table 8-264. ISS CBUFF-Generated Events**

Event	Description
IRQ_CTXx_READY	Read mode: CPU can write data to the physical window pointed by <a href="#">CBUFF_CTX_STATUS_i[3:0]</a> WB.
	Write mode: CPU can read data from the physical window pointed by <a href="#">CBUFF_CTX_STATUS_i[3:0]</a> WB.
	Read/write mode: The OCPI initiator has completed writing a physical window.
IRQ_CTXx_INVALID	Invalid access
	OCPI writes the virtual space of context i in read mode.
	OCPI reads the virtual space of context i in write mode.
	OCPI writes the virtual space of context i outside the <a href="#">CBUFF_CTX_STATUS_i[11:8]</a> WA window in write or read/write mode.
	OCPI reads the virtual space of context i outside the <a href="#">CBUFF_CTX_STATUS_i[11:8]</a> WA window in read mode.

**Table 8-264. ISS CBUFF-Generated Events (continued)**

Event	Description
	OCPI reads the virtual space of context <i>i</i> outside the <code>CBUFF_CTX_STATUS_i[3:0]</code> WB window in read/write mode.
	CPU writes the DONE bit when physical windows are not ready for the CPU.
	This event indicates a wrong configuration of the CBUFF, the OCPI initiator or bogus software. When it happens, context <i>i</i> goes into an error state. In this state all accesses to the virtual space of context <i>i</i> are cancelled: they are not forwarded to the physical space. The purpose is to prevent corruption of the physical memory. Of course, the CBUFF still returns OCP responses to OCPI to ensure the integrity of the OCP.
	The error state can be left by disabling the context <i>i</i> and re-enabling it. Before doing so, software must ensure that there are no more outstanding requests to the virtual space of context <i>i</i> .
IRQ_CTXx_OVR	Physical space overflow or underflow event  This event indicates a bandwidth mismatch between data producer and data consumer. When it happens, context <i>i</i> does not go into error state. However, the data in the physical space is likely to be corrupted.
IRQ_OCP_ERR	OCP error received in the OCPO master port. The OCP response is forwarded to OCPI normally.

### 8.2.8.3 ISS CBUFF Functional Description

The CBUFF maps a virtual space to a physical space by address translation. It does not change the data or store it locally.

#### 8.2.8.3.1 ISS CBUFF Features

The ISS CBUFF features are:

- Fully transparent for accesses out of the configured virtual space
- Three functional modes:
  - Read mode: Read requests received from OCPI and forwarded after translation to OCPO. Writes are handled by an external process and acknowledged by the CPU.
  - Write mode: Write request received from OCPI and forwarded after translation to OCPO. Reads are handled by an external process and acknowledged by the CPU.
  - Read/write mode: Read and write requests received from OCPI and forwarded after translation to OCPO
- Four independent contexts
- Virtual address space (linear) mapped into a physical space (circular)
- Maximum physical buffer size of 16 × 16MB:
  - Physical space consists of 2, 4, 8, or 16 windows.
  - Maximum allowed window size is 16MB.
- Support of 2D addressing modes
- Strong error detection mechanisms to prevent data corruption caused by bogus configuration
- Addresses are 128-bit aligned, but window fill level managing is byte accurate
- Bandwidth control feedback loop to stall in initiator connected to OCPI

#### 8.2.8.3.2 ISS CBUFF Functional Description

The CBUFF maps a virtual address space to a physical space also called circular buffer.

The CBUFF can handle up to eight contexts. For most data formats, primarily four contexts are used. In cases where YUV4:2:0 data is exchanged, two contexts are consumed by the SIMCOP (JPEG encode) coming from the resizer module, which is the ISP output module. Moreover, a context is a virtual full-frame buffer that maps to a configurable number of physical windows.

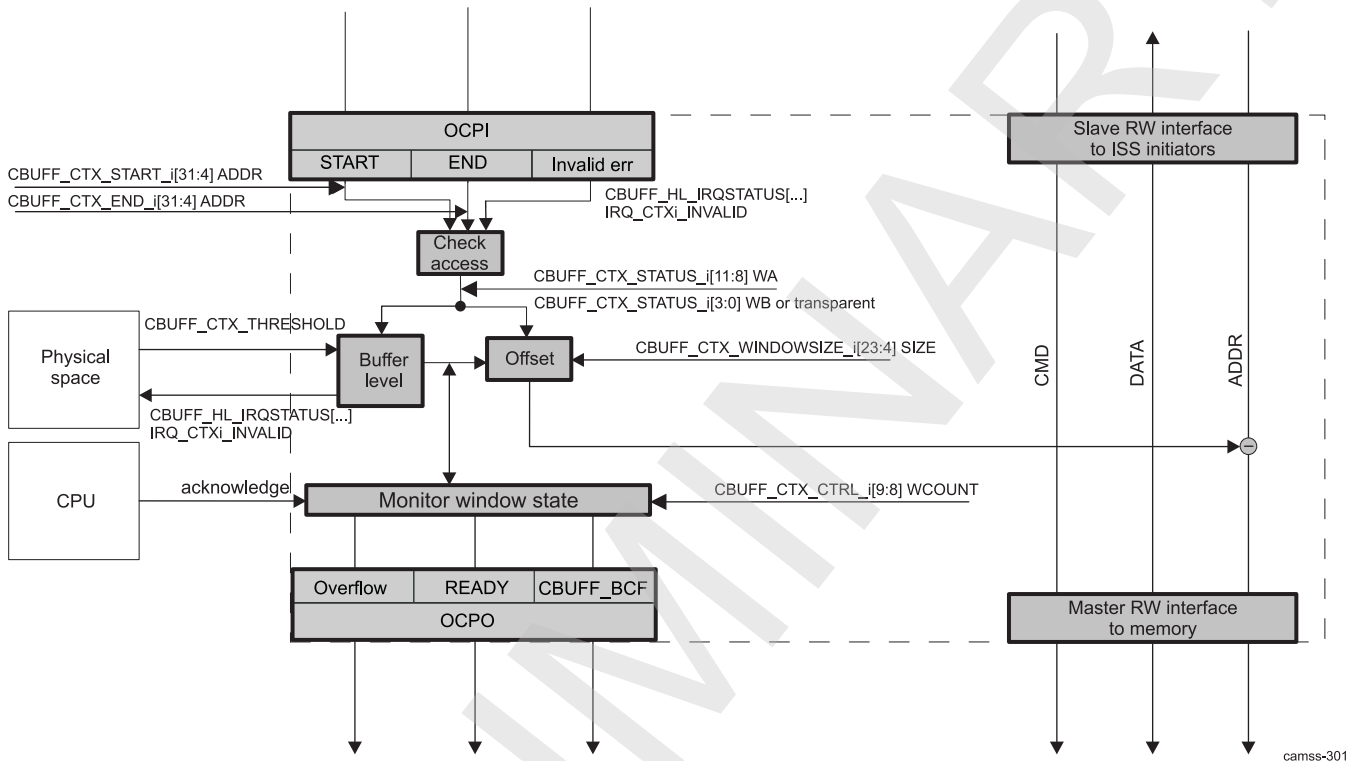
This section gives an overview of typical uses of the CBUFF.

### 8.2.8.3.2.1 ISS CBUFF Top-Level Diagram

Figure 8-87 shows the functional principle diagram. It does not include an exhaustive list of the interface signals or internal status registers.

A more detailed functional description is provided in the following sections.

**Figure 8-87. ISS CBUFF Top-Level Diagram**



camss-301

### 8.2.8.3.2.2 ISS CBUFF Functional Modes

The CBUFF supports three functional modes (see Table 8-265).

**Table 8-265. ISS CBUFF Functional Modes**

Mode	Data Written by	Data Read by
Write mode	ISS initiator	CPU-controlled process
Read mode	CPU-controlled process	ISS initiator
Read/write mode	ISS initiator	ISS initiator

#### 8.2.8.3.2.2.1 ISS CBUFF Write Mode

In write mode, the physical space is written by the CBUFF and it is read by the CPU. An `IRQ_CTXx_READY` event is set each time a physical window is available to be read by the CPU. This happens when the `CBUFF_CT_X_STATUS_i[11:8] WA` pointer is moved by the CBUFF.

The CBUFF sets an `IRQ_CTXx_READY` event to inform the CPU that it can access the `CBUFF_CT_X_STATUS_i[3:0] WB` physical window. The CBUFF cannot monitor CPU accesses to the physical window. The CPU must indicate when it has completed the processing of the `CBUFF_CT_X_STATUS_i[3:0] WB` window by setting the `CBUFF_CT_X_CTRL_i[2:1] DONE` bit field. This increments the window index `CBUFF_CT_X_STATUS_i[3:0] WB` by one modulo the window count (defined by the `CBUFF_CT_X_CTRL_i[9:8] WCOUNT` bit field).



The CBUFF ensures that one IRQ\_CTXx\_READY event is sent to the CPU per physical window to be read. In other words, when a new IRQ\_CTXx\_READY event occurs before the previous one(s) is acknowledged by the CPU, it is not lost. The CBUFF memorizes the event and triggers the interrupt line again when the CPU clears it.

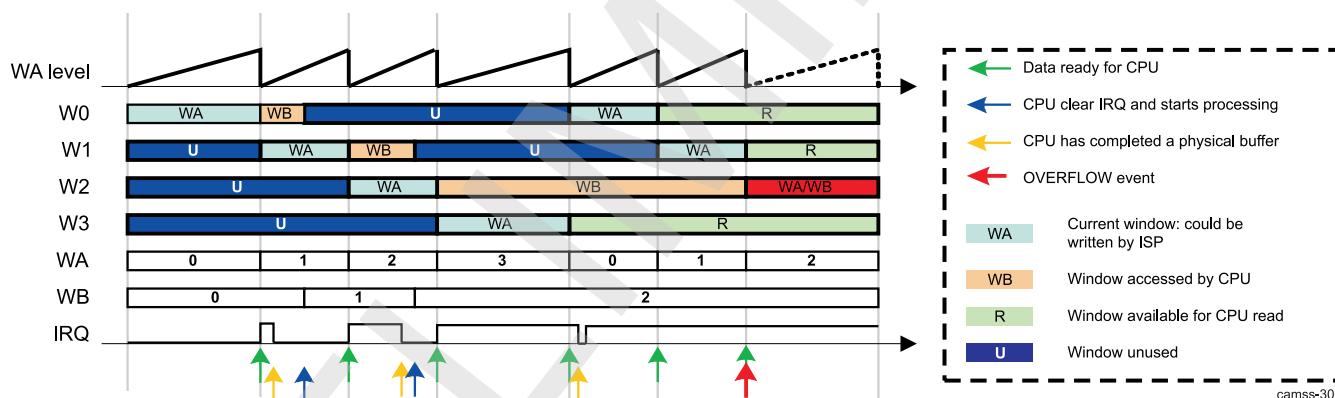
When the CPU reads the physical space too slowly, the CBUFF\_CTX\_STATUS\_i[11:8] WA window pointer falls into the CBUFF\_CTX\_STATUS\_i[3:0] WB window. This generates an IRQ\_CTXx\_OVR event when the OCPI initiator performs an access to that window. The CPU receives the IRQ\_CTXx\_READY event only if it is allowed to access a physical window. Therefore, it cannot generate an IRQ\_CTXx\_OVR event in a normal case. The OCPI initiator accesses are tracked based on activity on the OCP port.

When an IRQ\_CTXx\_OVR event occurs, the buffer content is likely to be corrupted. However, the CBUFF context continues processing data normally; it does not go into error state. Software must reset data generation and the CBUFF in a clean manner.

Figure 8-88 shows an example scenario with CBUFF\_CTX\_CTRL\_i[9:8] WCOUNT = 1 (four windows). In normal operation, the CPU processes data at least at the same speed as it is written to the physical space. When this is not true, the number of windows to be read by the CPU increases. When no more physical windows are available for OCPI writes, an overflow occurs.

In the previous example, the CPU takes more time than expected to read the third buffer. Therefore, physical windows are not freed up. The OCPI initiator continues to write data into the physical window. That leads to an overflow when the OCPI initiator writes into the physical window that is read by the CPU (CBUFF\_CTX\_STATUS\_i[11:8] WA = CBUFF\_CTX\_STATUS\_i[3:0] WB and writes into CBUFF\_CTX\_STATUS\_i[11:8] WA detected).

Figure 8-88. ISS CBUFF Write Mode CPU Interaction Example



- (1) When there is no physical window available to be read by the CPU and the CPU writes the DONE bit, an IRQ\_CTXx\_INVALID event occurs.
- (2) The bandwidth control feedback (BCF) feature can be used to prevent overflow. It must be supported by the module writing data into the virtual space (typically, an ISS).

### 8.2.8.3.2.2 ISS CBUFF Read Mode

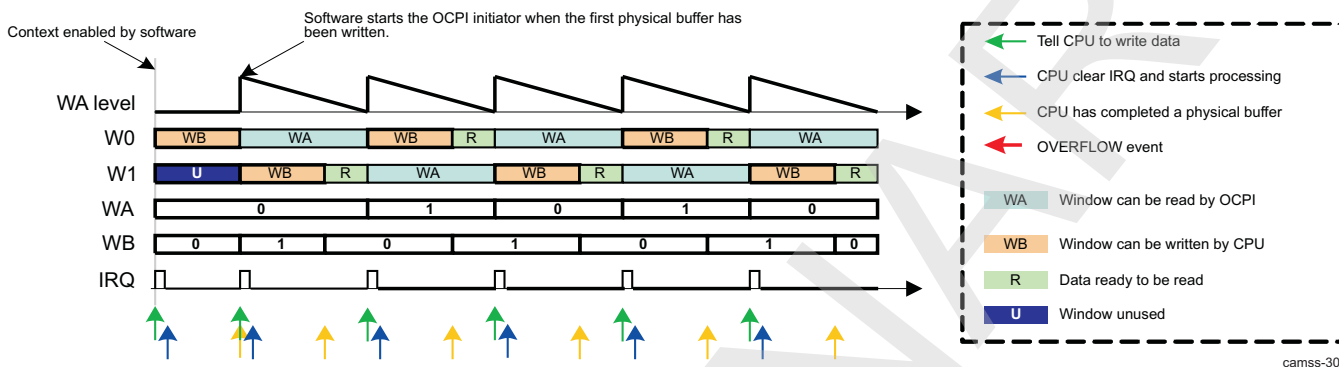
In read mode, the physical space is read by the CBUFF and it is written by the CPU. An IRQ\_CTXx\_READY event is set each time a physical window is available to be written by the CPU. This is true until all free buffers are used.

The CBUFF sets an IRQ\_CTXx\_READY event to inform the CPU that it can write the CBUFF\_CTX\_STATUS\_i[3:0] WB window. The CBUFF cannot monitor CPU accesses to the physical space. It must indicate when it completes writing the CBUFF\_CTX\_STATUS\_i[3:0] WB window by setting the CBUFF\_CTX\_CTRL\_i[2:1] DONE bit field. This increments the CPU window index CBUFF\_CTX\_STATUS\_i[3:0] WB by one modulo the window count (defined by the CBUFF\_CTX\_CTRL\_i[9:8] WCOUNT bit field).

The CBUFF ensures that one IRQ\_CTXx\_READY event is sent to the CPU per physical window to be written. In other words, when a new IRQ\_CTXx\_READY event occurs before the previous one(s) is acknowledged by the CPU, it is not lost. The CBUFF memorizes the event and triggers the interrupt line again when the CPU clears it.

The following is an example of a normal operation in which the CPU writes data faster than it is read by the OCPI initiator. **Figure 8-89** uses two physical windows (CBUFF\_CTX\_CTRL\_i[9:8] WCOUNT = 0).

**Figure 8-89. ISS CBUFF CPU Writes Data Faster Than it Is Read by the OCPI Initiator**



Software must enable the ISS initiator only when at least one window is written by the CPU. Otherwise, an OVERFLOW event occurs when BCF is not used.

The CPU receives an interrupt each time a physical window is available to receive data. It clears the interrupt and then starts filling the physical window. When the buffer is completely written, it sets the CBUFF\_CTX\_CTRL\_i[2:1] DONE bit field. This happens before the OCPI initiator has read all data from the previous physical window: the CPU must wait until the next IRQ\_CTXx\_READY event is received before it can write again to the physical space.

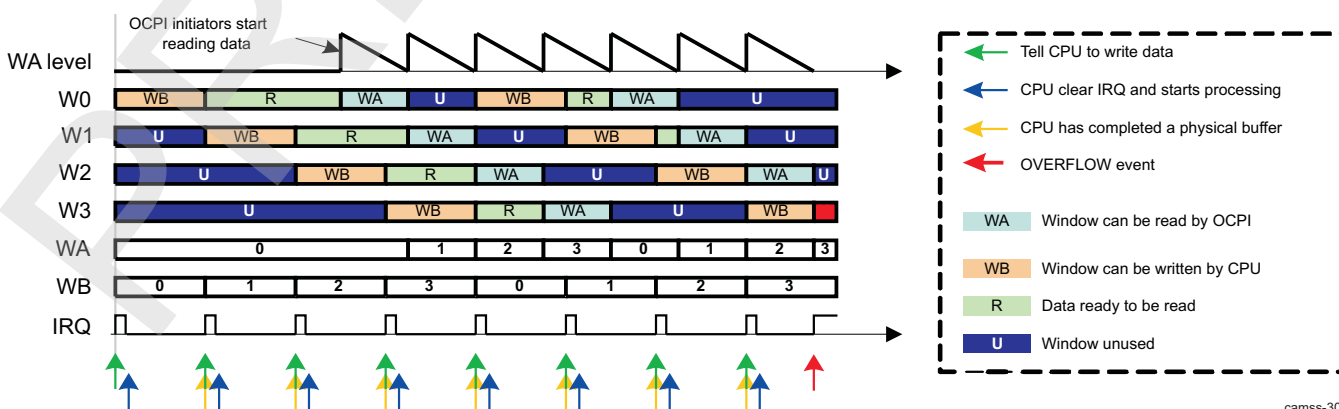
When the CPU writes buffers too slowly, the CBUFF\_CTX\_STATUS\_i[11:8] WA window falls into the CBUFF\_CTX\_STATUS\_i[3:0] WB window. This generates an IRQ\_CTXx\_OVR event only when the OCPI initiator performs reads to that window. The CPU receives the IRQ\_CTXx\_READY event only if it is allowed to access a physical window. Therefore, it does not generate an IRQ\_CTXx\_OVR event in a normal case. ISS accesses are tracked based on activity on the OCPI port.

When no buffer is available to be written by the CPU and the CPU writes the DONE bit, an IRQ\_CTXx\_INVALID event occurs.

When an IRQ\_CTXx\_OVR event occurs, the OCPI initiator reading the virtual space is likely to receive dummy data. However, the CBUFF context continues to process data normally; it does not go into error state. Software must reset the data consumer (OCPI initiator) and CBUFF context in a clean manner.

**Figure 8-90** is an example of CBUFF read mode CPU interaction.

**Figure 8-90. CBUFF Read Mode CPU Interaction Example**



The BCF feature can be used to prevent overflow. It must be supported by the module reading data from the virtual space (typically, an ISS).

### 8.2.8.3.2.2.3 ISS CBUFF Read/Write Mode

Reads and writes are performed by OCPI initiators such as a camera interface, ISP, or SIMCOP. Address translation is performed for read and write data flows. The WA pointer is used for the write data flow, and the WB pointer is used for the read data flow. Therefore, address translation for the write data flow is the same as for write mode.

In this mode, the OCPI read data flow is stalled when there is not enough data to read in the physical space. A typical application is to store data from the camera in a CBUFF and to read it back by the ISP. When the camera is slower than the ISP, the ISP stalls.

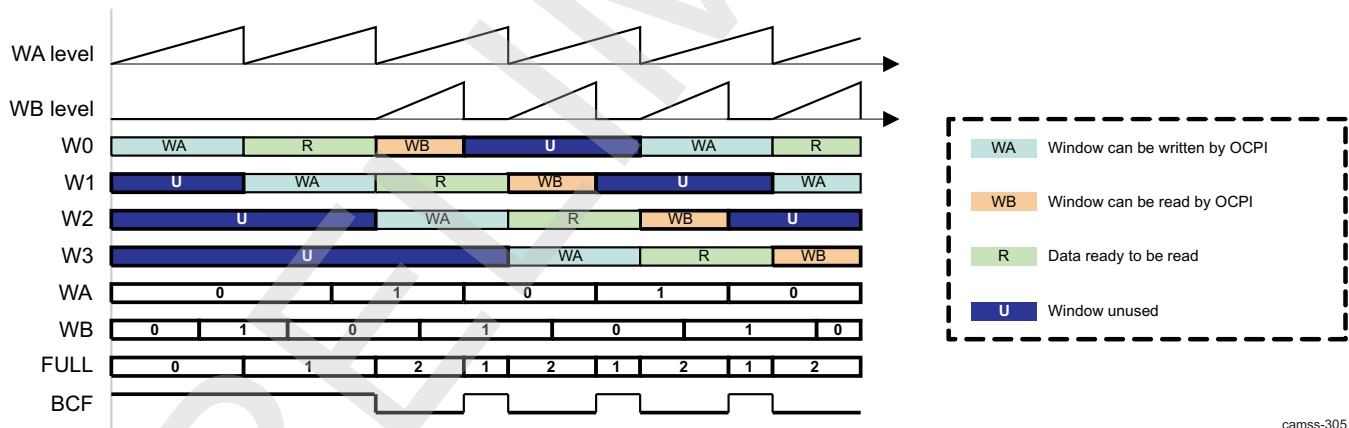
This mode does not rely on CPU synchronization. Synchronization is always performed using the BCF hardware mechanism. IRQ\_CTXx\_READY events are triggered in this mode when the OCPI write initiator has filled a physical window. Software can use this event for debug or performance bench marking purposes. Writes to the CBUFF\_CTX\_CTRL\_i[2:1] DONE bit field are ignored in this mode.

The IRQ\_CTXx\_OVR event is triggered when an underflow occurs. In a normal case this should not occur, because the BCF signal is used to stall the read data flow when insufficient data is available in the physical space.

Figure 8-91 assumes:

- Read/write mode
- Four physical windows
- CBUFF\_CTX\_THRESHOLD\_S\_i = CBUFF\_CTX\_THRESHOLD\_F\_i
- The OCPI read initiator is faster than the OCPI write initiator. The BCF feature is used to stall the read initiator. CBUFF\_CTX\_CTRL\_i[7:4] BCF = 2

Figure 8-91. ISS CBUFF Read/Write Mode Example



camss-305

### 8.2.8.3.2.3 ISS CBUFF Events and Status Checking

#### 8.2.8.3.2.3.1 ISS CBUFF Operations

A CBUFFx\_READY\_IRQ event is generated each time the CPU can read data from the CBUFF. The CPU can clear the event when it starts processing the data to avoid masking other events. The CPU can keep track of the location on the data internally or use the CBUFF registers to compute it.

The formula used is: ADDR = CBUFF\_CTX\_STATUS\_i[3:0] WB CBUFF\_CTX\_WINDOWSIZE\_i + CBUFF\_CTX\_PHY\_i

When the CPU is done with processing, it must free the buffer by setting the CBUFF\_CTX\_CTRL\_i[10] DONE bit. Otherwise, an overflow event may occur.

The CBUFF does not keep track of EOF events. They must be managed by the CPU using the EOF event of the module that writes into the CBUFF. At the EOF, data may remain in the current write windows. For example, when the window size is set to 8 lines and the image size is 20 lines, only two window-ready events are generated for a linear addressing scheme. The remaining four lines can be read after the EOF event.

No automatic reset of the CBUFF FSM occurs at the end of the image frame. Software must reset the context by clearing the `CBUFF_CTX_CTRL_i[0]` ENABLEABLE bit when the frame is completely processed. A new frame can start only when the `CBUFF_CTX_CTRL_i[0]` ENABLEABLE bit is set.

#### 8.2.8.3.2.3.2 ISS CBUFF Status Checking

The CBUFF provides read-only access to the `CBUFF_CTX_STATUS_i[11:8]` WA and `CBUFF_CTX_STATUS_i[3:0]` WB pointers through the `CBUFF_CTX_STATUS_i` register. For example, the `CBUFF_CTX_STATUS_i[3:0]` WB index can be used by the CPU to compute the address of a physical window. Those indexes can also be used to evaluate latency margins.

#### 8.2.8.3.2.3.3 ISS CBUFF Register Accessibility During Frame Processing

All registers are busy-writeable registers. These registers/fields can be read or written even if the module is busy. Changes to the underlying settings occur instantly. However, module behavior is unpredictable when registers are changed during processing.

For correct operation, software must:

1. Disable all accesses to the virtual space managed by the context.
2. Disable the context by clearing the `CBUFF_CTX_CTRL_i[0]` ENABLEABLE bit.
3. Change the configuration.
4. Re-enable CBUFFx by setting the `CBUFF_CTX_CTRL_i[0]` ENABLEABLE bit.

#### 8.2.8.3.2.4 ISS CBUFF Memory-to-Memory Operation BCF

The BCF mechanism matches the bandwidth between two processes.

The BCF feature can be used in all three CBUFF modes:

- Read mode: A CPU-controlled process writes data into physical space. The BCF signal is deasserted when the physical space contains enough data to start the read initiator connected to OCPI.
- Write mode: The BCF signal controls the write initiator connected to OCPI. When the CPU-controlled process does not read the data fast enough from the physical space, the BCF signal is asserted to stall filling of the buffer. It is deasserted when enough space is available in the physical space.
- Read/write mode: The BCF signal controls the read initiator connected to OCPI. Another initiator connected to OCPI fills the physical space. The BCF signal is deasserted when the physical space contains data that can be read by the read initiator. It is deasserted when insufficient data is available in the physical space.

The `CBUFF_BCF` output is controlled based on two factors:

- The window count available for the OCPI initiator to read/write
- The amount of data in the last available window (pretrigger)

The `CBUFF_BCF` signal is enabled by the `CBUFF_CTX_CTRL_i[7:4]` BCF bit field. It defines:

- Write mode: The amount of required free windows to allow writing from the ISS. In other words, when less than the required amount of BCF windows is available for ISS writes, the stall mechanism is triggered. The number of free windows is initialized to the total window count of the context. It is decreased by 1 each time the OCPI initiator finishes writing a window. It is increased by 1 each time the CPU finishes reading a window.
- Read and read/write modes: The minimum amount of required full windows to allow reading from the OCPI. When fewer than `CBUFF_CTX_CTRL_i[7:4]` BCF windows are available for ISS read, the stall mechanism is triggered. The number of full windows is initialized to 0. It is decreased by 1 each time the OCPI initiator completes reading a window. It is increased by 1 each time the CPU/OCPI initiator finishes writing a window.

Figure 8-92 is an example of BCF use. It assumes:

- Write mode: OCPI writes data into physical space and the CPU reads it.
- Two physical windows
- $CBUFF\_CTX\_THRESHOLD\_S\_i = CBUFF\_CTX\_THRESHOLD\_F\_i$
- The OCPI write initiator is faster than the CPU.
- $CBUFF\_CTX\_CTRL\_i[7:4] BCF = 1$

Figure 8-92. ISS CBUFF Write Mode CPU Interaction Example BCF Used

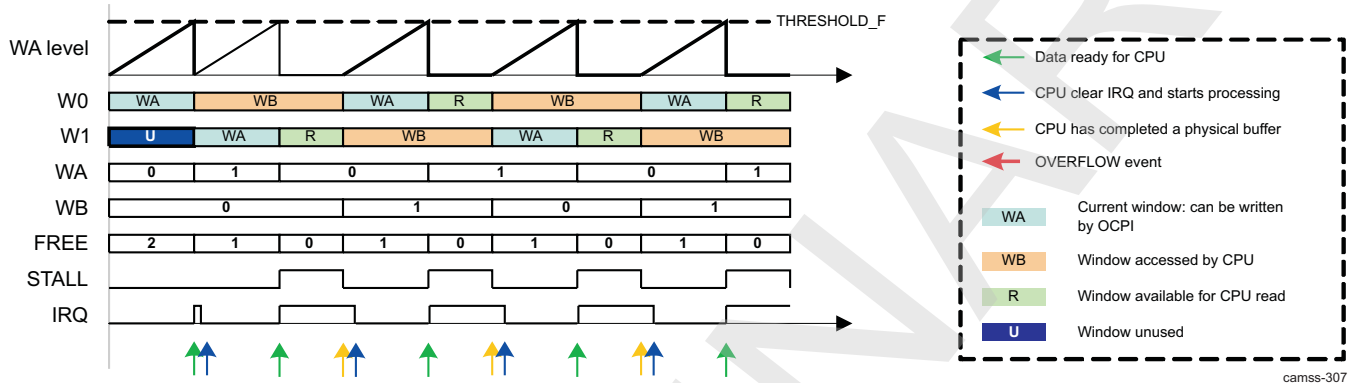


Figure 8-93 is another example. It assumes:

- Read mode: The CPU writes data into physical space and the OCPI reads it.
- Four physical windows
- $CBUFF\_CTX\_THRESHOLD\_S\_i = CBUFF\_CTX\_THRESHOLD\_F\_i$
- The OCPI read initiator is faster than the CPU. It starts after the CPU.
- $CBUFF\_CTX\_CTRL\_i[7:4] BCF = 2$

Figure 8-93. ISS CBUFF Read Mode CPU Interaction Example (1)

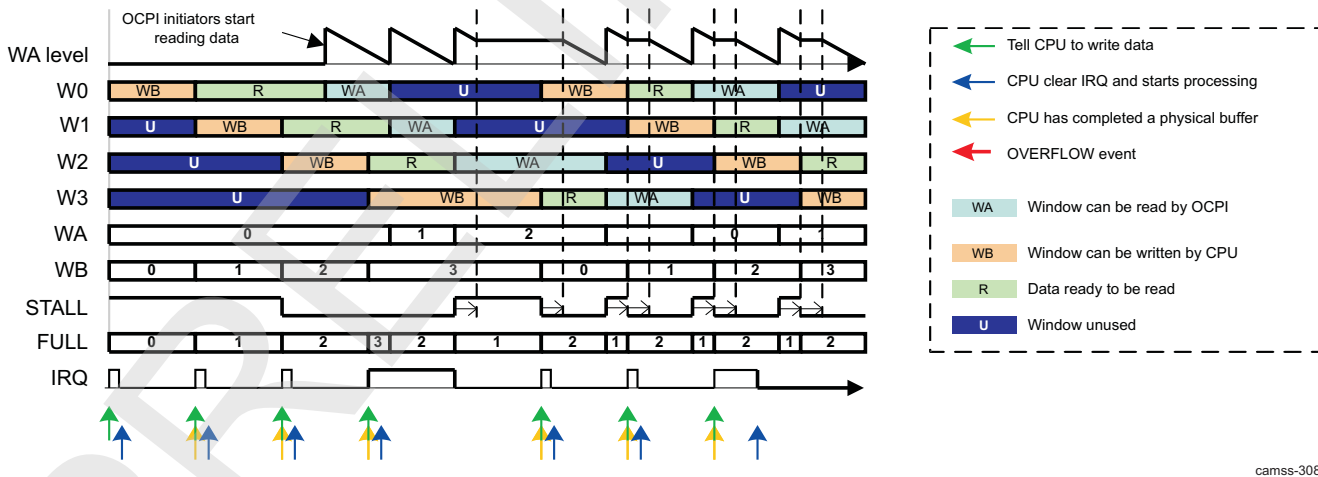
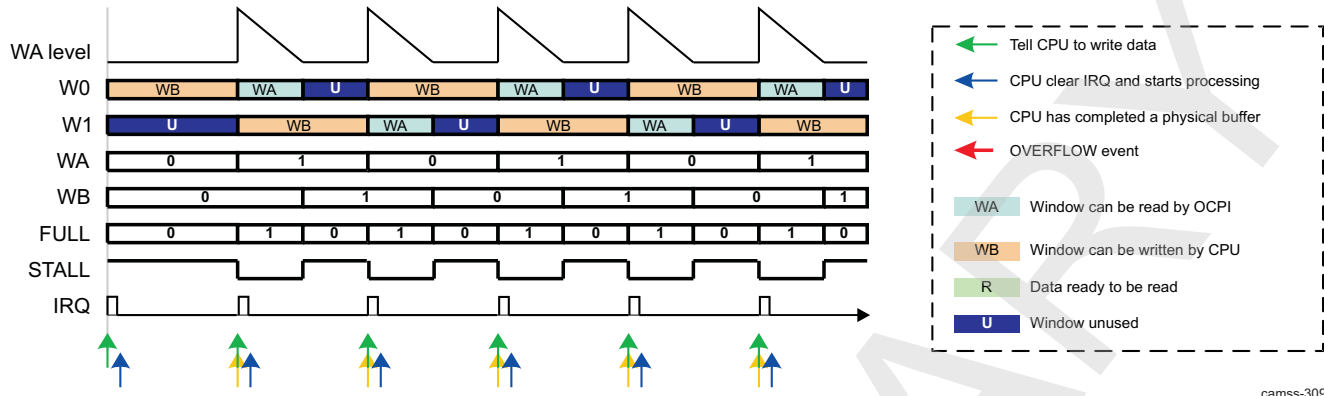


Figure 8-94 is another example. It assumes:

- Read mode: The CPU writes data into physical space and the OCPI reads it.
- Two physical windows
- $CBUFF\_CTX\_THRESHOLD\_S\_i = CBUFF\_CTX\_THRESHOLD\_F\_i$
- The OCPI read initiator is faster than the CPU. It starts after the CPU.
- $CBUFF\_CTX\_CTRL\_i[7:4] BCF = 1$

Figure 8-94. ISS CBUFF Read Mode CPU Interaction Example (2)



camss-309

Figure 8-95 is another example. It assumes:

- Read mode: The CPU writes data into physical space and the OCPI reads it.
- Four physical windows
- $\text{CBUFF\_CTX\_THRESHOLD\_S}_i = \text{CBUFF\_CTX\_THRESHOLD\_F}_i$
- The OCPI read initiator is faster than the CPU. It starts after the CPU.
- $\text{CBUFF\_CTX\_CTRL}_i[7:4] \text{ BCF} = 1$

CBUFF supports pretriggering the BCF signal for finer latency compensation control. This mechanism is active only for the last window available for OCPI access. It typically improves the physical space use, which is particularly useful when on-chip SRAM is used as a ping-pong buffer.

Pretriggering is controlled through the  $\text{CBUFF\_CTX\_THRESHOLD\_S}_i$  register. It defines:

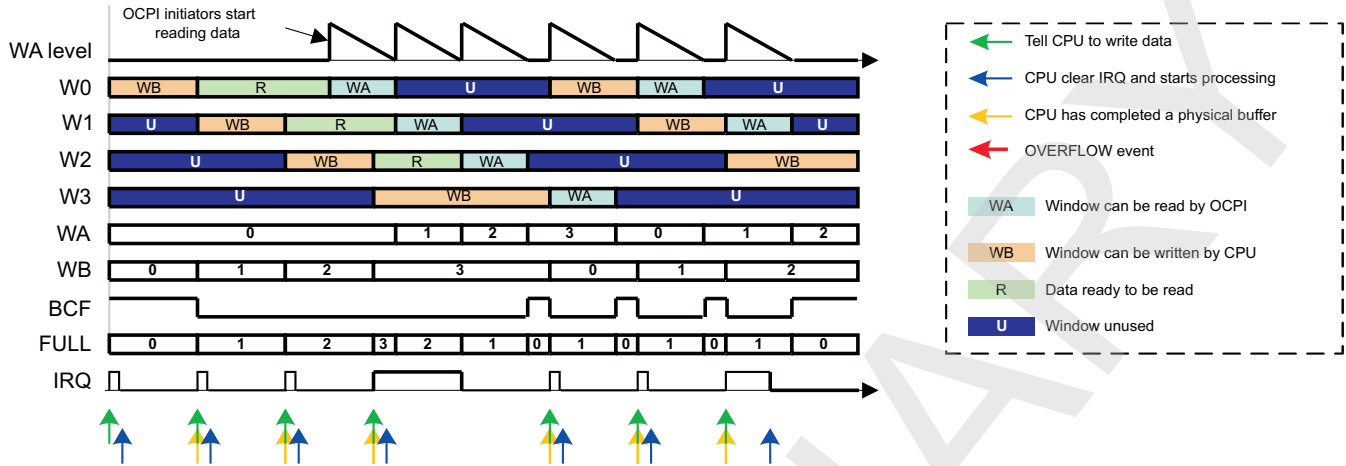
- Write mode: When the fill level of the last window available for ISS writes is greater than or equal to  $\text{CBUFF\_CTX\_THRESHOLD\_S}_i$ , the  $\text{CBUFF\_BCF}$  signal is asserted.
- Read mode: When the amount of data in the last window available for ISS reads is less than  $\text{CBUFF\_CTX\_THRESHOLD\_F}_i / \text{CBUFF\_CTX\_THRESHOLD\_S}_i$ , the  $\text{CBUFF\_BCF}$  signal is asserted.

BCF pretriggering is disabled by setting  $\text{CBUFF\_CTX\_THRESHOLD\_S}_i = \text{CBUFF\_CTX\_THRESHOLD\_F}_i$ .

Figure 8-95 is an example of BCF pretriggering, assuming the following:

- Write mode: The CPU reads data from the physical space and the OCPI writes it.
- Two physical windows
- $\text{CBUFF\_CTX\_THRESHOLD\_S}_i = 1/3 \text{ CBUFF\_CTX\_THRESHOLD\_F}_i$
- $\text{CBUFF\_CTX\_CTRL}_i[7:4] \text{ BCF} = 1$

Figure 8-95. ISS CBUFF BCF Pretrigger Example: Write Mode



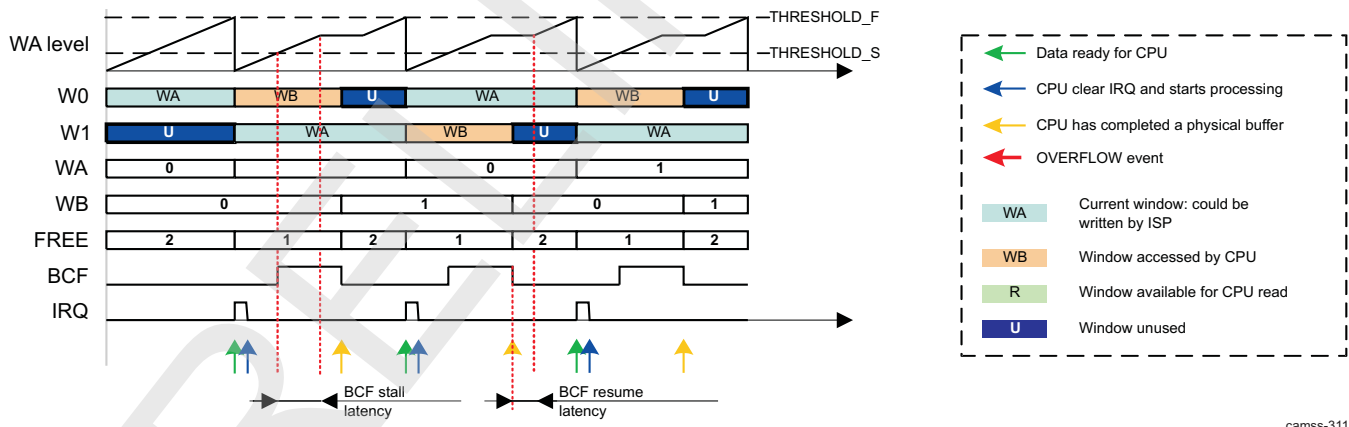
camss-310

The CBUFF\_BCF output signal is a logical OR of all internal BCF signals of CBUFF contexts. For example, when contexts 0 and 1 have enabled the BCF feature, both contexts can request a data flow stall.

Figure 8-96 is an example of pretriggering in read mode:

- Read mode: CPU writes data into the physical space and the OCPI reads it.
- Four physical windows
- $CBUFF\_CTX\_THRESHOLD\_S\_i = 2/3 CBUFF\_CTX\_THRESHOLD\_F\_i$
- The OCPI read initiator is faster than the CPU. It starts after the CPU.
- $CBUFF\_CTX\_CTRL\_i[7:4] BCF = 2$

Figure 8-96. ISS CBUFF BCF Pretrigger Example: Read Mode



camss-311

### 8.2.8.3.2.5 ISS CBUFF TILER Support

The CBUFF can translate 2D (BLOCK) bursts intended for the TILER. However, software must ensure that a given burst fits in a window.

The expected value for ADDR[32] is defined by the CBUFF\_CTX\_CTRL\_i[11] TILERMODE bit.

- When ADDR[32] = CBUFF\_CTX\_CTRL\_i[11] TILERMODE, the access is processed normally. ADDR[32] is not used for further processing.
- Otherwise, the access is treated as transparent.

### 8.2.8.3.2.6 ISS CBUFF Window Management and Address Remapping Details

This section explains the internal address remapping and windows management algorithm. Internally, the module maintains some variables in addition to the configuration registers. The module manages multiple contexts in parallel. [Table 8-266](#) lists the CBUFF internal variables.

**Table 8-266. ISS CBUFF Internal Variables**

Variable	Description
WAX	Current window index for context x. Possible values are 0 to allowed window count. The current value can be read using the <a href="#">CBUFF_CTX_STATUS_i[11:8]</a> WA bit field.
WBx	Window in the physical space that can be accessed by the CPU in read/write modes. Window that is read from OCPI in read/write mode. Possible values are 0 to allowed window count. The current value can be read using the <a href="#">CBUFF_CTX_STATUS_i[3:0]</a> WB bit field.
VPAX	Start address, in the virtual space, of the <a href="#">CBUFF_CTX_STATUS_i[11:8]</a> WA window
	Used as a base pointer for the read (read mode) or write data flow (write, read/write modes)
	This is an internal quantity that cannot be accessed by software.
VPBx	Start address, in the virtual space, of the <a href="#">CBUFF_CTX_STATUS_i[3:0]</a> WB window
	Used as a base pointer for the read data flow.
	This is an internal quantity that cannot be accessed by software.
OFFSETAx	This is an internal quantity that cannot be accessed by software.
OFFSETBx	Address offset used when WAX or WBx is accessed
LEVELAx	This is an internal quantity that cannot be accessed by software.
LEVELBx	Address offset used when WAX or WBx is accessed

#### 8.2.8.3.2.6.1 ISS CBUFF Startup

The status of a CBUFF context is reset when it is disabled. This does not affect the configuration registers or the CBUFF\_IRQSTATUS register. [Table 8-267](#) lists the internal state after reset.

**Table 8-267. ISS CBUFF Internal State After Reset**

Variable	Description
WAX	0
WBx	0
VPAX VPBx	<a href="#">CBUFF_CTX_START_i</a>
OFFSETAx OFFSETBx	<a href="#">CBUFF_CTX_START_i</a> – <a href="#">CBUFF_CTX_PHY_i</a>
LEVELAx LEVELBx	0

#### 8.2.8.3.2.6.2 ISS CBUFF Access Identification

For each access to the virtual space (OCPI slave port), the CBUFF first checks the address to classify the transaction into one of the categories listed in [Table 8-268](#).

**Table 8-268. ISS CBUFF Address identification**

Address ID	Variable	Condition
0+2*x	WA_CBUFFx	<a href="#">CBUFF_CTX_CTRL_i[0]</a> ENABLE = 1 and ADDR >= VPAX and ADDR < VPAX + <a href="#">CBUFF_CTX_WINDOWSIZE_i</a> and ADDR <= <a href="#">CBUFF_CTX_END_i</a> and Access type = write when read/write mode is selected
1+2*x	WB_CBUFFx	<a href="#">CBUFF_CTX_CTRL_i[0]</a> ENABLE = 1 and <a href="#">CBUFF_CTX_CTRL_i[2:1]</a> MODE = 2 and



**Table 8-268. ISS CBUFF Address identification (continued)**

Address ID	Variable	Condition
		Access type = read and ADDR >= VPBx and ADDR < VPBx + CBUFF_CTX_WINDOWSIZE <sub>i</sub> and ADDR <= CBUFF_CTX_END <sub>i</sub>
16+x	ERR_CBUFFx	CBUFF_CTX_CTRL <sub>i</sub> [0] ENABLE = 1 and ADDR >= CBUFF_CTX_START <sub>i</sub> and ADDR <= CBUFF_CTX_END <sub>i</sub>
24	TRANSPARENT	Always true

Lower IDs correspond to higher priorities if multiple conditions are true. For example, when the current virtual window of CBUFF 0 is accessed, at least the tests for categories WA\_CBUFFx and ERR\_CBUFFx are true. The final category is WA\_CBUFFx, because it has a higher priority.

**NOTE:** Tests must be performed in parallel to match the desired performance.

Further processing depends on the category:

- **TRANSPARENT:** Accesses flow through the module without changing its internal state or any translation.
- **ERR\_CBUFFx:** The module goes into error state for the concerned context and sets the [CBUFF\\_HL\\_IRQSTATUS.IRQ\\_CTXx\\_INVALID](#) bit. When the module is in error state for CBUFFx, all accesses to that buffer are cancelled. In other words, any access that has an address between [CBUFF\\_CTX\\_START<sub>i</sub>](#) and [CBUFF\\_CTX\\_END<sub>i</sub>](#) is not transmitted to the OCPO master port. There are two ways to leave the error state:
  - Hardware reset
  - Disable and re-enable the context in error state.
 Accesses outside of the virtual space from the context in error state are not affected.
- **WA\_CBUFFx and WB\_CBUFFx:** The internal state is updated and address translation is performed when the performed access type (read or write) is compatible with the current mode (read, write, or read/write mode). Otherwise, a [CBUFF\\_HL\\_IRQSTATUS.IRQ\\_CTXx\\_INVALID](#) event is set and CBUFFx goes into the error state.

### 8.2.8.3.2.6.3 ISS CBUFF Address Translation

An offset is selected depending on the access category (see [Section 8.2.8.3.2.6.1, ISS CBUFF Startup](#)) and the internal state of the accessed buffer. [Table 8-269](#) lists possible cases.

**Table 8-269. ISS CBUFF Address Translation**

Condition	Address Translation
CBUFF_CTX_CTRL <sub>i</sub> [0] ENABLE = 1 and ADDR >= CBUFF_CTX_START <sub>i</sub> and ADDR <= CBUFF_CTX_END <sub>i</sub> and CBUFFx in error state	Access cancelled
Category = WA_CBUFFx	ADDR <sub>OUT</sub> = ADDR <sub>IN</sub> -OFFSET <sub>Ax</sub>
Category = WB_CBUFFx	Read/write mode only ADDR <sub>OUT</sub> = ADDR <sub>IN</sub> -OFFSET <sub>Bx</sub>
Category = ERR_CBUFFx	Access cancelled
Category = TRANSPARENT	ADDR <sub>OUT</sub> = ADDR <sub>IN</sub>

### 8.2.8.3.2.6.4 ISS CBUFF Window Fill Level

Each time an access is performed into an active window (WA\_CBUFF<sub>x</sub> when context *x* is enabled) the window level is updated. The corresponding LEVEL<sub>y</sub> is incremented according to the BYTEEN input of the OCPI slave port. All possible BYTEEN patterns, including nonaligned ones, are supported. Table 8-270 shows some examples. The basic idea is to count the number of 1s in the BYTEEN input for each 128-bit word and to sum the values.

**Table 8-270. ISS CBUFF Window-Level Increment**

BYTEEN	LEVEL <sub>y</sub> Increment	Comment
0x0000	0	No access
0x0001	1	8-bit access
0x0002	1	8-bit access
0x0003	2	16-bit access
...	...	...
0x0007	3	24-bit access
...	...	...
0x000F	4	32-bit access
...	...	...
0x00F0	4	32-bit access
...	...	...
0xFFFF	16	128-bit access

The window level is compared to CBUFF<sub>x</sub>\_THRESHOLD. Table 8-271 lists the situations that may occur:

**Table 8-271. ISS CBUFF Window-Level Comparison<sup>(1)</sup>**

Condition	Description
LEVEL <sub>A</sub> <sub>x</sub> >= CBUFF_CTX_THRESHOLD_F <sub>i</sub>	The CBUFF_CTX_STATUS <sub>i</sub> [11:8] WA bit field of context <i>x</i> is full (write mode) or empty (read mode). Internal window indexes, levels, and offsets are updated.
LEVEL <sub>B</sub> <sub>x</sub> >= CBUFF_CTX_THRESHOLD_F <sub>i</sub>	Read/write mode only Window used to handle the read flow of context <i>x</i> is empty. Internal window indexes, levels, and offsets are updated.

<sup>(1)</sup> All situations described in Table 8-271 are mutually exclusive because only one level is updated each cycle.

### 8.2.8.3.2.6.5 ISS CBUFF Window Pointer and Offset Update

The following description refers to the update of WA in write mode:

When the current window of a context is full:

- A new window is opened. The update is done in a circular manner: the first physical window in is reused after the last one.
  - WA (WA + 1) modulo the number defined by CBUFF\_CTX\_CTRL<sub>i</sub>[9:8] WCOUNT
  - LEVEL<sub>A</sub> <- 0
  - VPA <- VPA + CBUFF\_CTX\_WINDOWSIZE<sub>i</sub>
- When the window is moved from the last buffer to the first:
  - OFFSET<sub>A</sub> <- OFFSET<sub>A</sub> + CBUFF\_CTX\_WINDOWSIZE<sub>i</sub> << (CBUFF\_CTX\_CTRL<sub>i</sub>[9:8] WCOUNT + 1)
- Otherwise, OFFSET<sub>A</sub> does not change.

The algorithm used to update WA in read mode is the same, except that *full* refers to *all data of the window has been read or window empty*.

For read/write mode, the same algorithm is used to translate reads to the WB<sub>x</sub> window. Make the following changes:

- Replace WA with WB.

- Replace LEVELA with LEVELB.
- Replace VPA with VPB.
- Replace OFFSETA with OFFSETB.

**8.2.8.3.2.7 ISS CBUFF Error State**

Contexts may go into error state when they receive unexpected accesses. In that case, the CBUFF does not send dummy transactions to the OCPO port. It does not forward the failing transactions to OCPO and returns valid responses (SResp = DVA) on OCPI to preserve the integrity of the OCP.

Responses may be received on OCPO while the CBUFF responds to failing transactions on OCPI. The OCPO response phase is stalled (MRespAccept = 0) during that time to prevent corruption. OCPO cannot be stalled longer than one full OCP transaction. Responses received on OCPO are handled before another internally generated response can be sent back to the OCPI. In that case, command and data phases are eventually stalled. Not stalling OCPO for too long is required to avoid affecting system performance.

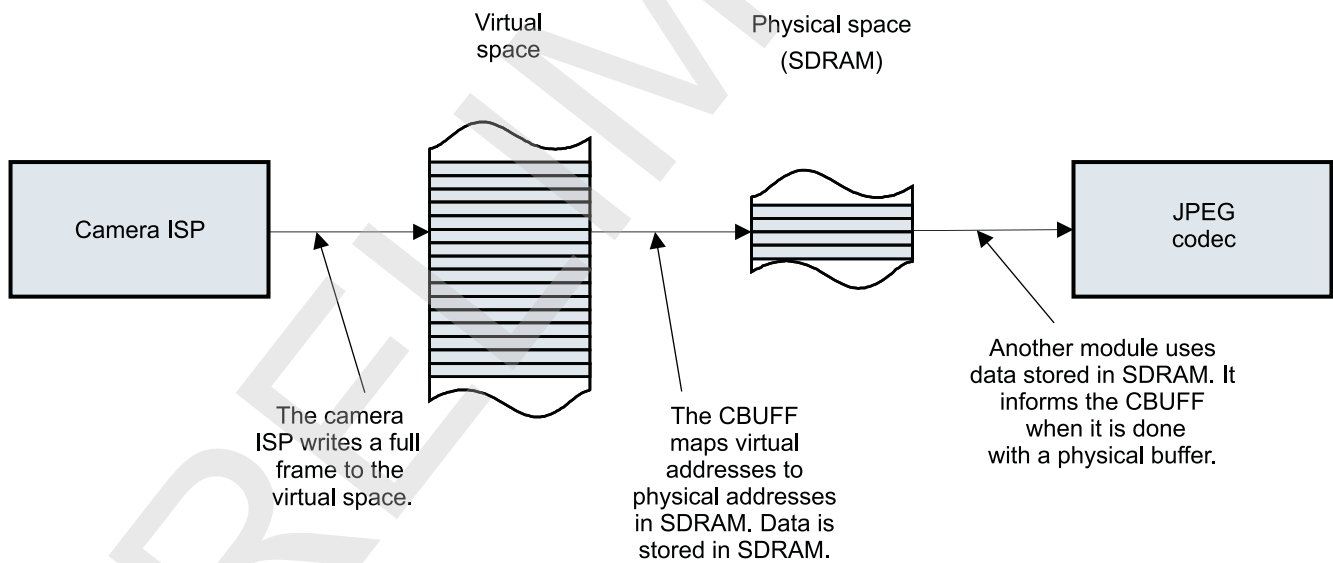
During normal operation (that is, no context is in error state), the OCPO response phase is never stalled.

**8.2.8.3.2.8 ISS CBUFF Typical Configurations**

**8.2.8.3.2.8.1 ISS CBUFF Single-Slice Buffer**

An OCP master (typically, the ISS) writes data with an incremental addressing scheme to the virtual space. The physical space is smaller than the virtual space. Therefore, physical space locations are read and written multiple times. Figure 8-97 shows the CBUFF single-slice buffer in write mode.

**Figure 8-97. ISS CBUFF Single-Slice Buffer (Write Mode)**

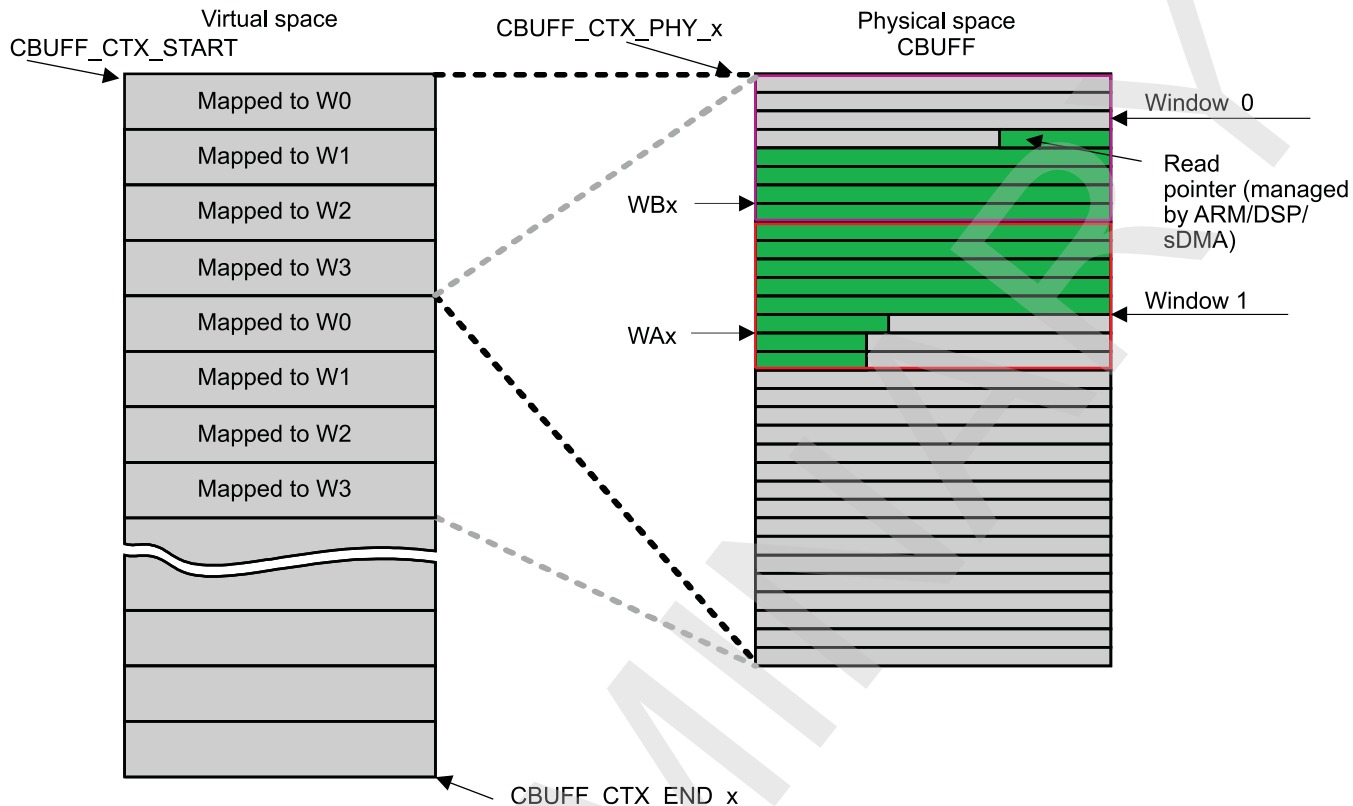


camsa-313

Physically, this CBUFF is in on-chip SRAM or SDRAM. The virtual space is defined by a start and end addresses. The physical space is defined by a start address, a window size, and a window count. It is contiguous in memory. When the CPU accesses physical memory for processing, it must know if the SDRAM is available for it to access and if the CBUFF is not using it. The CPU and CBUFF cannot track each other. For example, an interrupt must be triggered from the SIMCOP to the CBUFF to notify it when the processor is done working with SDRAM. For more information about the software configuration for these interrupts, see Section 8.2.8.3.2.2.1, ISS CBUFF Write Mode.

Figure 8-98 shows the buffer organization for a 4-window buffer.

**Figure 8-98. ISS CBUFF Single-Slice Buffer Example (Write Mode)**



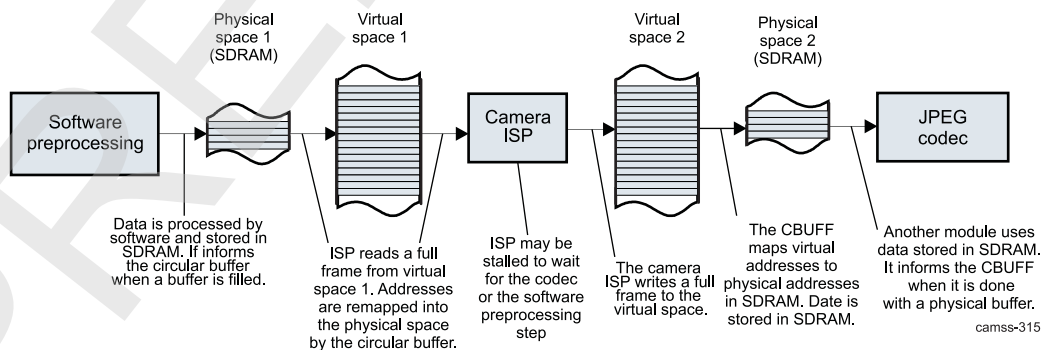
camss-314

The CBUFF can manage multiple contexts in single-slice mode.

**8.2.8.3.2.8.2 ISS CBUFF Extended-Slice Buffer**

In extended-slice mode, at least two contexts managed by the CBUFF are used together. The two contexts provide address translation, one for the read data flow and the other for a write data flow. Figure 8-99 is an example of the CBUFF extended-slice buffer.

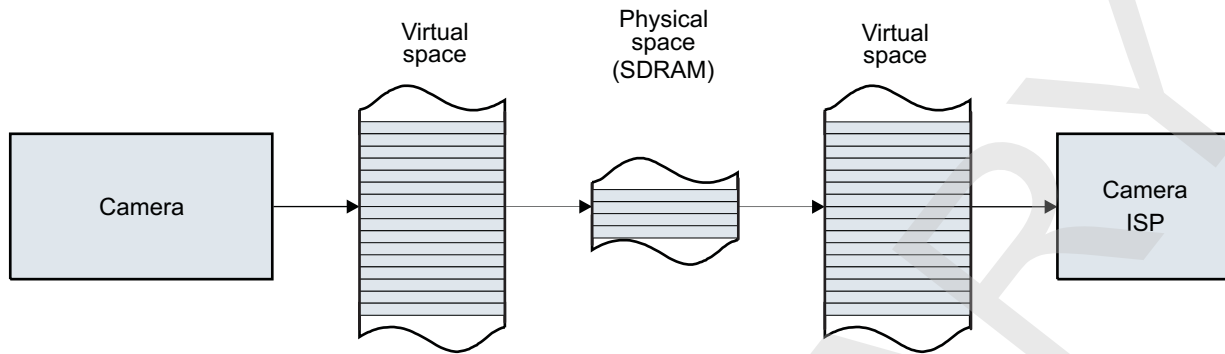
**Figure 8-99. ISS CBUFF Extended-Slice Buffer Example**



**8.2.8.3.2.9 ISS CBUFF FIFO Mode**

The CBUFF can behave like a FIFO to buffer data between two initiators connected to the OCPI. A typical use case is a camera interface writing data to the FIFO and an ISP reading data from the FIFO. Figure 8-100 shows the CBUFF FIFO mode.

Figure 8-100. ISS CBUFF FIFO Mode



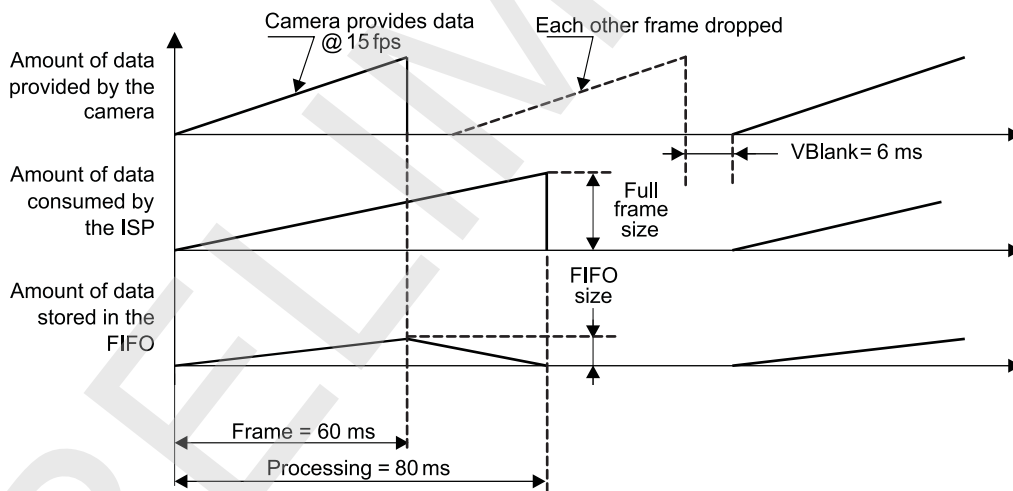
camss-316

The CBUFF is used as a FIFO when the camera provides data at a higher rate than the ISP can process. To avoid storing one or multiple full-frame buffers, a FIFO buffer is used to store the data that cannot be processed immediately.

For this example, assume that the camera provides 16 MPix at 15 frames per second. A new frame is provided every 66 ms. The ISP needs 80 ms to process this frame. Therefore, the processing cannot be done on the fly, and each other frame must be dropped.

Figure 8-101 shows the accumulated amount of data provided by the camera since the start of the frame; the amount of data consumed by the ISP since it has started processing the frame; and the difference between the two, which must be stored in the FIFO.

Figure 8-101. ISS CBUFF FIFO Use Example



camss-317

The FIFO is emulated as 16 windows (`CBUFF_CTX_CTRL_i[9:8] WCOUNT = 0x3`). The ISP is started by software when a few kilobytes of data are written to the physical space. Alternatively, the BCF mechanism can be used, but the minimum FIFO size is more complex to calculate.

The FIFO level reaches its maximum when the camera performs the last write of the frame. The camera provides the data at  $1 \text{ byte/pixel} \times 16 \text{ MPix}/60 \text{ ms} = 266 \text{ Mbps}$ . The ISP reads data at  $1 \text{ byte/pixel} \times 16 \text{ MPix}/80 \text{ ms} = 200 \text{ Mbps}$ . Therefore, the minimum FIFO size is  $60 \text{ ms} \times (266 \text{ Mbps} - 200 \text{ Mbps}) = 3.96 \text{ Mbps}$ , which is four times less than a full-frame buffer.

Each window holds 256KB of data. The camera starts writing to a new physical window every  $256\text{KB}/266 \text{ Mbps} = 962 \mu\text{s}$ . The ISP starts reading from a new physical window every 1280  $\mu\text{s}$ .

The FIFO mode can also be used when the ISP is slower than the camera. In that case, the ISP is stalled when insufficient data is available in the physical space.

## 8.2.8.4 ISS CBUFF Programming Model

### 8.2.8.4.1 ISS CBUFF Reset Behavior

Upon hardware or software reset of the CBUFF, all registers in the CBUFF are reset to their reset values.

This software reset has the same effect as a hardware reset. ISS high-level software reset ensures that traffic is stopped on clear boundary because the CBUFF is sending data to and from other modules. ISS top-level reset is preferred. For more information about ISS software reset, see [Section 8.1.2.3, ISS Reset](#). Submodule reset is not preferred but is available through the following registers:

1. Set the [CBUFF\\_HL\\_SYSCONFIG\[0\] SOFTRESET](#) bit to 1.
2. Read the [CBUFF\\_HL\\_SYSCONFIG\[0\] SOFTRESET](#) bit to check whether it is set to 1, which means the reset occurred.

The reset is performed without waiting until all OCP traffic stops. To avoid OCP corruption, software must ensure there is no more ongoing traffic before performing a reset.

### 8.2.8.4.2 ISS CBUFF Register Setup

All registers of the context to be used must be initialized for correct operation. [Table 8-272](#) lists the procedure for the CBUFF register setup.

**Table 8-272. ISS CBUFF Setup Register**

Step	Bit Field	Value
Set operation mode.	<a href="#">CBUFF_CTX_CTRL_i[2:1] MODE</a>	0x0: Write mode 0x1: Read mode 0x2: Read/write mode 0x3: Reserved
Define the virtual address range managed by the CBUFF. It usually corresponds to the address region where one image frame is written by the OCPI initiator.	<a href="#">CBUFF_CTX_START_i</a> and <a href="#">CBUFF_CTX_END_i</a>	
Define the start address of the physical buffer.	<a href="#">CBUFF_CTX_PHY_i</a>	
Set the window count and size. The window size usually depends on the use of the buffer. Eight or 16 video lines correspond to a current size for JPEG video compression. A higher window count provides better latency-related overflow protection.	<a href="#">CBUFF_CTX_CTRL_i[9:8] WCOUNT</a> and <a href="#">CBUFF_CTX_WINDOWSIZE_i</a>	
When the 2D addressing capability is not used, set to the window size in <a href="#">CBUFF_CTX_THRESHOLD_F_i</a> . Otherwise, it is set to a smaller value depending on the buffer organization. For example, when each window corresponds to lines by 4096 pixels, but the ISP sends lines of only 2560 pixels, <a href="#">CBUFF_CTX_WINDOWSIZE_i</a> = 8 4096 and <a href="#">CBUFF_CTX_THRESHOLD_F_i</a> = 8 2560.	<a href="#">CBUFF_CTX_THRESHOLD_F_i</a>	
BCF signal-generation configuration is optional.	<a href="#">CBUFF_CTX_THRESHOLD_S_i</a> and <a href="#">CBUFF_CTX_CTRL_i[7:4] BCF</a>	
Enable the module. It can be disabled by clearing the ENABLE bit. This must be done only when there are no more outstanding requests to the virtual space managed by CBUFFx. All internal FSMs and counters of the CBUFF are reset when it is disabled. Pending interrupts are not affected.	<a href="#">CBUFF_CTX_CTRL_i[0] ENABLE</a>	

## 8.2.8.5 ISS CBUFF Register Manual

### 8.2.8.5.1 ISS CBUFF Instance Summary

[Table 8-273](#) lists the CBUFF instance.

**Table 8-273. ISS CBUFF Instance Summary**

Module Name	L3 Base Address	Size
ISS_CBUFF	0x5200 1800	512 bytes

### 8.2.8.5.2 ISS CBUFF Registers

#### 8.2.8.5.2.1 ISS CBUFF Register Summary

Table 8-274 summarizes the CBUFF registers.

**Table 8-274. ISS CBUFF Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CBUFF Base Address
CBUFF_HL_REVISION	R	32	0x0000 0000	0x5200 1800
CBUFF_HL_HWINFO	R	32	0x0000 0004	0x5200 1804
CBUFF_HL_SYSCONFIG	RW	32	0x0000 0010	0x5200 1810
RESERVED	RW	32	0x0000 001C	0x5200 181C
CBUFF_HL_IRQSTATUS_RAW	RW	32	0x0000 0020	0x5200 1820
CBUFF_HL_IRQSTATUS	RW	32	0x0000 0024	0x5200 1824
CBUFF_HL_IRQENABLE_SET	RW	32	0x0000 0028	0x5200 1828
CBUFF_HL_IRQENABLE_CLR	RW	32	0x0000 002C	0x5200 182C
CBUFF_FRAG_ADDR <sub>j</sub> <sup>(1)</sup>	RW	32	0x0000 0080 + (0x4 * j)	0x5200 1880 + (0x4 * j)
CBUFF_CTX_CTRL <sub>i</sub> <sup>(2)</sup>	RW	32	0x0000 0100 + (0x20 * i)	0x5200 1900 + (0x20 * i)
CBUFF_CTX_START <sub>i</sub> <sup>(2)</sup>	RW	32	0x0000 0104 + (0x20 * i)	0x5200 1904 + (0x20 * i)
CBUFF_CTX_END <sub>i</sub> <sup>(2)</sup>	RW	32	0x0000 0108 + (0x20 * i)	0x5200 1908 + (0x20 * i)
CBUFF_CTX_WINDOW_SIZE <sub>i</sub> <sup>(2)</sup>	RW	32	0x0000 010C + (0x20 * i)	0x5200 190C + (0x20 * i)
CBUFF_CTX_THRESH_OLD_F <sub>i</sub> <sup>(2)</sup>	RW	32	0x0000 0110 + (0x20 * i)	0x5200 1910 + (0x20 * i)
CBUFF_CTX_THRESH_OLD_S <sub>i</sub> <sup>(2)</sup>	RW	32	0x0000 0114 + (0x20 * i)	0x5200 1914 + (0x20 * i)
CBUFF_CTX_STATUS <sub>i</sub> <sup>(2)</sup>	R	32	0x0000 0118 + (0x20 * i)	0x5200 1918 + (0x20 * i)
CBUFF_CTX_PHY <sub>i</sub> <sup>(2)</sup>	RW	32	0x0000 011C + (0x20 * i)	0x5200 191C + (0x20 * i)

<sup>(1)</sup> j = 0 to 15

<sup>(2)</sup> i = 0 to 7

#### 8.2.8.5.2.2 ISS CBUFF Register Description

Table 8-275 through Table 8-305 describe the CBUFF registers.

**Table 8-275. CBUFF\_HL\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	IP revision identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 8-276. Register Call Summary for Register CBUFF\_HL\_REVISION**

ISS Interfaces

- [ISS CBUFF Register Manual: \[0\]](#)

**Table 8-277. CBUFF\_HL\_HWINFO**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	Information about the IP module's hardware configuration.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CONTEXTS	ENABLE_FRAGMENTATION

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:1	CONTEXTS	Number of contexts Read 0x3: Reserved Read 0x2: 8 contexts Read 0x1: 4 contexts Read 0x0: 2 contexts	R	0x1
0	ENABLE_FRAGMENTATION	Provides information to software if fragmentation support is available Read 0x1: Yes Read 0x0: No	R	0

**Table 8-278. Register Call Summary for Register CBUFF\_HL\_HWINFO**

ISS Interfaces

- [ISS CBUFF Register Manual: \[0\]](#)



**Table 8-279. CBUFF\_HL\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	Clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE	RESERVED	SOFTRESET	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:2	IDLEMODE	<p>Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.</p> <p>0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. Backup mode, for debug only.</p> <p>0x1: No-idle mode: local target never enters IDLE state. Backup mode, for debug only.</p> <p>0x3: Reserved</p> <p>0x2: Smart-idle mode: local target's IDLE state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wake-up events.</p>	RW	0x2
1	RESERVED		R	0
0	SOFTRESET	<p>Software reset</p> <p>Write 0x0: No action</p> <p>Write 0x1: Initiate software reset</p> <p>Read 0x1: Reset (software or other) ongoing</p> <p>Read 0x0: Reset done, no pending action</p>	RW	0

**Table 8-280. Register Call Summary for Register CBUFF\_HL\_SYSCONFIG**

ISS Interfaces

- [ISS CBUFF Integration](#): [0] [1]
- [ISS CBUFF Programming Model](#): [2] [3]
- [ISS CBUFF Register Manual](#): [4]

**Table 8-281. CBUF\_HL\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	ISS_CBUF
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .		
<b>Description</b>	Per-event raw interrupt status vector. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
IRQ_CTX7_OVR	IRQ_CTX6_OVR	IRQ_CTX5_OVR	IRQ_CTX4_OVR	IRQ_CTX3_OVR	IRQ_CTX2_OVR	IRQ_CTX1_OVR	IRQ_CTX0_OVR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_READY	IRQ_CTX6_READY	IRQ_CTX5_READY	IRQ_CTX4_READY	IRQ_CTX3_READY	IRQ_CTX2_READY	IRQ_CTX1_READY	IRQ_CTX0_READY	RESERVED										IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
30	IRQ_CTX6_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
29	IRQ_CTX5_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
28	IRQ_CTX4_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
27	IRQ_CTX3_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
26	IRQ_CTX2_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
25	IRQ_CTX1_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
24	IRQ_CTX0_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
15	IRQ_CTX7_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
14	IRQ_CTX6_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
13	IRQ_CTX5_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
12	IRQ_CTX4_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
11	IRQ_CTX3_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
10	IRQ_CTX2_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
9	IRQ_CTX1_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
8	IRQ_CTX0_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
7:1	RESERVED		R	0x00
0	IRQ_OCP_ERR	OCP error received in the master port. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

**Table 8-282. Register Call Summary for Register CBUFF\_HL\_IRQSTATUS\_RAW**

ISS Interfaces

- [ISS CBUFF Integration: \[0\] \[1\]](#)
- [ISS CBUFF Register Manual: \[2\]](#)

**Table 8-283. CBUFF\_HL\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	ISS_CBUFF
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .		
<b>Description</b>	Per-event "enabled" interrupt status vector. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IRQ_CTX7_OVR	IRQ_CTX6_OVR	IRQ_CTX5_OVR	IRQ_CTX4_OVR	IRQ_CTX3_OVR	IRQ_CTX2_OVR	IRQ_CTX1_OVR	IRQ_CTX0_OVR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_READY	IRQ_CTX6_READY	IRQ_CTX5_READY	IRQ_CTX4_READY	IRQ_CTX3_READY	IRQ_CTX2_READY	IRQ_CTX1_READY	IRQ_CTX0_READY	RESERVED								IRQ_OCP_ERR

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
30	IRQ_CTX6_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
29	IRQ_CTX5_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
28	IRQ_CTX4_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
27	IRQ_CTX3_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
26	IRQ_CTX2_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
25	IRQ_CTX1_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
24	IRQ_CTX0_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
15	IRQ_CTX7_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
14	IRQ_CTX6_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
13	IRQ_CTX5_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
12	IRQ_CTX4_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
11	IRQ_CTX3_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
10	IRQ_CTX2_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
9	IRQ_CTX1_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
8	IRQ_CTX0_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
7:1	RESERVED		R	0x00
0	IRQ_OCP_ERR	OCP error received in the master port. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

**Table 8-284. Register Call Summary for Register CBUFF\_HL\_IRQSTATUS**

ISS Interfaces

- [ISS CBUFF Integration: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS CBUFF Functional Description: \[4\] \[5\]](#)
- [ISS CBUFF Register Manual: \[6\]](#)

**Table 8-285. CBUFF\_HL\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	ISS_CBUFF
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .		
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_OVR	IRQ_CTX6_OVR	IRQ_CTX5_OVR	IRQ_CTX4_OVR	IRQ_CTX3_OVR	IRQ_CTX2_OVR	IRQ_CTX1_OVR	IRQ_CTX0_OVR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_READY	IRQ_CTX6_READY	IRQ_CTX5_READY	IRQ_CTX4_READY	IRQ_CTX3_READY	IRQ_CTX2_READY	IRQ_CTX1_READY	IRQ_CTX0_READY	RESERVED					IRQ_OCP_ERR		

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
30	IRQ_CTX6_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
29	IRQ_CTX5_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
28	IRQ_CTX4_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
27	IRQ_CTX3_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
26	IRQ_CTX2_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
25	IRQ_CTX1_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
24	IRQ_CTX0_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
15	IRQ_CTX7_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
14	IRQ_CTX6_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
13	IRQ_CTX5_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
12	IRQ_CTX4_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
11	IRQ_CTX3_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
10	IRQ_CTX2_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
9	IRQ_CTX1_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
8	IRQ_CTX0_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
7:1	RESERVED		R	0x00
0	IRQ_OCP_ERR	OCP error received in the master port. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

**Table 8-286. Register Call Summary for Register CBUFF\_HL\_IRQENABLE\_SET**

ISS Interfaces

- [ISS CBUFF Integration: \[0\]](#)
- [ISS CBUFF Register Manual: \[1\]](#)

**Table 8-287. CBUFF\_HL\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	ISS_CBUFF
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .		
<b>Description</b>	Per-event interrupt enable bit vector, line 0. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ_CTX7_OVR	IRQ_CTX6_OVR	IRQ_CTX5_OVR	IRQ_CTX4_OVR	IRQ_CTX3_OVR	IRQ_CTX2_OVR	IRQ_CTX1_OVR	IRQ_CTX0_OVR	IRQ_CTX7_INVALID	IRQ_CTX6_INVALID	IRQ_CTX5_INVALID	IRQ_CTX4_INVALID	IRQ_CTX3_INVALID	IRQ_CTX2_INVALID	IRQ_CTX1_INVALID	IRQ_CTX0_INVALID	IRQ_CTX7_READY	IRQ_CTX6_READY	IRQ_CTX5_READY	IRQ_CTX4_READY	IRQ_CTX3_READY	IRQ_CTX2_READY	IRQ_CTX1_READY	IRQ_CTX0_READY	RESERVED					IRQ_OCP_ERR		

Bits	Field Name	Description	Type	Reset
31	IRQ_CTX7_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
30	IRQ_CTX6_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
29	IRQ_CTX5_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
28	IRQ_CTX4_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
27	IRQ_CTX3_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
26	IRQ_CTX2_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
25	IRQ_CTX1_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
24	IRQ_CTX0_OVR	Buffer overflow event. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
23	IRQ_CTX7_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
22	IRQ_CTX6_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
21	IRQ_CTX5_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
20	IRQ_CTX4_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
19	IRQ_CTX3_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
18	IRQ_CTX2_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

## ISS Interfaces

www.ti.com

Bits	Field Name	Description	Type	Reset
17	IRQ_CTX1_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
16	IRQ_CTX0_INVALID	Invalid access. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
15	IRQ_CTX7_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
14	IRQ_CTX6_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
13	IRQ_CTX5_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
12	IRQ_CTX4_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
11	IRQ_CTX3_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
10	IRQ_CTX2_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
9	IRQ_CTX1_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
8	IRQ_CTX0_READY	The WB physical window is ready to be accessed by the CPU. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
7:1	RESERVED		R	0x00
0	IRQ_OCP_ERR	OCP error received in the master port. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

**Table 8-288. Register Call Summary for Register CBUFF\_HL\_IRQENABLE\_CLR**

ISS Interfaces

- [ISS CBUFF Integration: \[0\]](#)
- [ISS CBUFF Register Manual: \[1\]](#)

**Table 8-289. CBUFF\_FRAG\_ADDR\_j**

<b>Address Offset</b>	0x0000 0080 + (0x4 * j)	<b>Index</b>	j = 0 to 15																																																											
	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF																																																											
<b>Description</b>	Start address of the physical buffer of the CBUFF context 0. This register only exists when fragmentation support is enabled.																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">ADDR</td> <td colspan="11">RESERVED</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ADDR																RESERVED										
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
ADDR																RESERVED																																														
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																										
31:4	ADDR	Address, in 128-bit words	RW	0x0000000																																																										
3:0	RESERVED		R	0x0																																																										

**Table 8-290. Register Call Summary for Register CBUFF\_FRAG\_ADDR\_j**

ISS Interfaces

- [ISS CBUFF Register Manual: \[0\]](#)

**Table 8-291. CBUFF\_CTX\_CTRL\_i**

<b>Address Offset</b>	0x0000 0100 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	Context control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TILERMODE	DONE	WCOUNT	BCF				RESERVED	MODE		ENABLE					

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11	TILERMODE	Sets the expected value for ADDR[32]. If ADDR[32]=TILERMODE, ADDR[31:4] is processed and eventually translated. Otherwise, the access is handled as transparent, regardless of the other address bits.	RW	0
10	DONE	Write this bit to 1 to indicate the CPU has finished processing its physical buffer. This bit is automatically cleared by hardware, reads always return 0. This bit has no effect when MODE=2 (read/write) Write 0x0: No effect. Write 0x1: The CPU has completely processed the WB physical buffer.	W	0
9:8	WCOUNT	Window count 0x0: 2 windows 0x1: 4 windows 0x3: 16 windows 0x2: 8 windows	RW	0x0
7:4	BCF	This register controls the bandwidth control feedback loop output. 0: Control loop disabled. 1-15: The control feedback loop enabled. Behavior depends on functional mode, see <a href="#">Section 8.2.8.3.2.4, ISS CBUFF Memory-to-Memory Operation BCF</a> .	RW	0x0
3	RESERVED		R	0
2:1	MODE	Selects the functional mode of this context 0x0: Write mode. ISS writes and CPU reads the physical space. CPU accesses are out of the scope of the CBUFF module; therefore, only writes are permitted between CBUFF_CTX_START_x and CBUFF_CTX_END_x. 0x1: Read mode. Hardware reads and CPU writes the physical space. CPU accesses are out of the scope of the CBUFF module; therefore, only reads are permitted between CBUFF_CTX_START_x and CBUFF_CTX_END_x. 0x2: Read/Write mode. Read and writes are monitored by the CBUFF. WB is used to track current read positions WA is used to track current write position.	RW	0x0



Bits	Field Name	Description	Type	Reset
0	ENABLE	<p>Enable/disable</p> <p>0x0: Disables the context. This resets the internal state of the context. All accesses received on OCPI are transmitted to OCPO without modification. Disabling the context takes effect immediately. Software must ensure that no more accesses to the context are outstanding before disabling it. Otherwise memory corruption may occur.</p> <p>0x1: Enable the context. All accesses between CBUFF_CTX_START_x and CBUFF_CTX_END_x are processed by the CBUFF.</p>	RW	0

**Table 8-292. Register Call Summary for Register CBUFF\_CTX\_CTRL\_i**

ISS Interfaces

- [ISS CBUFF Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [ISS CBUFF Programming Model: \[32\] \[33\] \[34\] \[35\]](#)
- [ISS CBUFF Register Manual: \[36\]](#)

**Table 8-293. CBUFF\_CTX\_START\_i**

<b>Address Offset</b>	0x0000 0104 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	Start address of the virtual space managed by the context		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Address, in 128-bit words	RW	0x00000000
3:0	RESERVED		R	0x0

**Table 8-294. Register Call Summary for Register CBUFF\_CTX\_START\_i**

ISS Interfaces

- [ISS CBUFF Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS CBUFF Programming Model: \[5\]](#)
- [ISS CBUFF Register Manual: \[6\]](#)

**Table 8-295. CBUFF\_CTX\_END\_i**

<b>Address Offset</b>	0x0000 0108 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	End address of the virtual space managed by the context		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Address, in 128-bit words	RW	0x00000000
3:0	RESERVED		R	0x0

**Table 8-296. Register Call Summary for Register CBUFF\_CTX\_END\_i**

## ISS Interfaces

- [ISS CBUFF Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS CBUFF Programming Model: \[5\]](#)
- [ISS CBUFF Register Manual: \[6\]](#)

**Table 8-297. CBUFF\_CTX\_WINDOWSIZE\_i**

<b>Address Offset</b>	0x0000 010C + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	Defines the size of a window		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SIZE																RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:4	SIZE	Size, in 128-bit words	RW	0x000000
3:0	RESERVED		R	0x0

**Table 8-298. Register Call Summary for Register CBUFF\_CTX\_WINDOWSIZE\_i**

## ISS Interfaces

- [ISS CBUFF Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS CBUFF Programming Model: \[5\] \[6\]](#)
- [ISS CBUFF Register Manual: \[7\]](#)

**Table 8-299. CBUFF\_CTX\_THRESHOLD\_F\_i**

<b>Address Offset</b>	0x0000 0110 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	Threshold value used to check if a write window is full		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								THRESHOLD																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:0	THRESHOLD	Threshold value, in bytes	RW	0x000000

**Table 8-300. Register Call Summary for Register CBUFF\_CTX\_THRESHOLD\_F\_i**

## ISS Interfaces

- [ISS CBUFF Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [ISS CBUFF Programming Model: \[11\] \[12\] \[13\]](#)
- [ISS CBUFF Register Manual: \[14\]](#)

**Table 8-301. CBUFF\_CTX\_THRESHOLD\_S\_i**

<b>Address Offset</b>	0x0000 0114 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	Threshold value used to control the BCF synchronization mechanism		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								THRESHOLD																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:0	THRESHOLD	Threshold value, in bytes	RW	0x000000

**Table 8-302. Register Call Summary for Register CBUFF\_CTX\_THRESHOLD\_S\_i**

ISS Interfaces

- [ISS CBUFF Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [ISS CBUFF Programming Model: \[11\]](#)
- [ISS CBUFF Register Manual: \[12\]](#)

**Table 8-303. CBUFF\_CTX\_STATUS\_i**

<b>Address Offset</b>	0x0000 0118 + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	Status register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WA				RESERVED				WB											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:8	WA	Valid values depend on the CBUFF_CTX_CTRL_x.WCOUNT register.	R	0x0
7:4	RESERVED		R	0x0
3:0	WB	Valid values depend on the CBUFF_CTX_CTRL_x.WCOUNT register.	R	0x0

**Table 8-304. Register Call Summary for Register CBUFF\_CTX\_STATUS\_i**

ISS Interfaces

- [ISS CBUFF Integration: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS CBUFF Functional Description: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\]](#)
- [ISS CBUFF Register Manual: \[29\]](#)

**Table 8-305. CBUFF\_CTX\_PHY\_i**

<b>Address Offset</b>	0x0000 011C + (0x20 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 8-274</a> .	<b>Instance</b>	ISS_CBUFF
<b>Description</b>	Start address of the first physical buffer managed by the context when fragmentation support is disabled.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Address, in 128 bit words.	RW	0x0000000
3:0	RESERVED		R	0x0

**Table 8-306. Register Call Summary for Register CBUFF\_CTX\_PHY\_i**

ISS Interfaces

- [ISS CBUFF Functional Description: \[0\] \[1\]](#)
- [ISS CBUFF Programming Model: \[2\]](#)
- [ISS CBUFF Register Manual: \[3\]](#)

## 8.3 ISS ISP

This section describes the image signal processor.

### 8.3.1 ISS ISP Overview

The image signal processor (ISP) is part of the image subsystem (ISS) of the device and is a key component for imaging and video applications. It describes all ISP modules in the multimedia device, that is, the video port (VP), pattern generator (PG), image pipe interface (IPIPEIF), image pipe module (IPIPE), resizer (RSZ), hardware 3A (H3A), image sensor interface (ISIF), and buffer logic (BL). For better understanding, see the first top-level ISS diagram and feature list in [Section 8.1, ISS Overview](#).

#### 8.3.1.1 ISS ISP Features

The video-processing hardware removes the need for expensive camera modules to perform processing functions. The ISS can support the following features:

- On-the-fly or memory-to-memory processing
- Up to 200-MHz pixel throughput.
- Statistic data collection:
  - On-the-fly or memory-to-memory operation
  - Data collection for auto exposure
  - Data collection for auto white balance
  - Data collection for auto focus
  - Boundary signal calculation for video stabilization
- IPIPE front end: RAW data processing:
  - On-the-fly or memory-to-memory processing
  - 16-bit-wide RAW BAYER data path between image and sensor linearization module
  - 12-bit-wide RAW BAYER data path between sensor linearization module and gamma correction module. Gamma correction module outputs 10-bit data.
  - Programmable Bayer RGB positions
  - Sensor data linearization for dynamic range extension
  - Programmable 2D lens shading compensation (LSC) correction
  - Per-pixel gain and offset control
  - Black level compensation
  - Boxcar filter
  - Data collection for histogram generation
  - Defect pixel correction (LUT\_DPC) with look-up table (LUT)
  - Defect pixel correction (OTF\_DPC) with on-the-fly detection and correction
  - 2D noise filtering
  - Green imbalance correction (GIC)
  - Digital gains and offset
  - 8- to 10-bit A-law decompression and 10- to 8-bit A-law compression.
- IPIPE back end: RGB and YUV data processing:
  - RGB-to-RGB color correction
  - Gamma correction (GC).
  - RGB -> YUV4:2:2: Color conversion, cosited chroma filtering and downsampling
  - 2D edge enhancement (EE)
  - 3D LUT for color correction
  - False chroma suppression (FCS)
- Two resizers:
  - Performance: input and output rates up to 200 MPix/s
  - YUV4:2:2 to RGB56, ARGB888, YUV4:2:2 and YUV4:2:0 data output
  - YUV4:2:0 to YUV4:2:0 data output

- RAW to RAW data output
- Range from x1/4096 to x20. Supports memory to memory rescaling.

The ISP comprises the following modules:

- Pattern generator (PG)
- IPIPE interface (IPIPEIF)
- Image sensor interface (ISIF) accelerator
- Auto exposure, auto white balance, and auto focus engine (H3A)
- Image pipe accelerator (IPIPE)
- Two resizer (RSZ) accelerators
- Buffer logic (BL): Receives module requests, performs arbitration, and creates read/write bursts to the memory subsystem

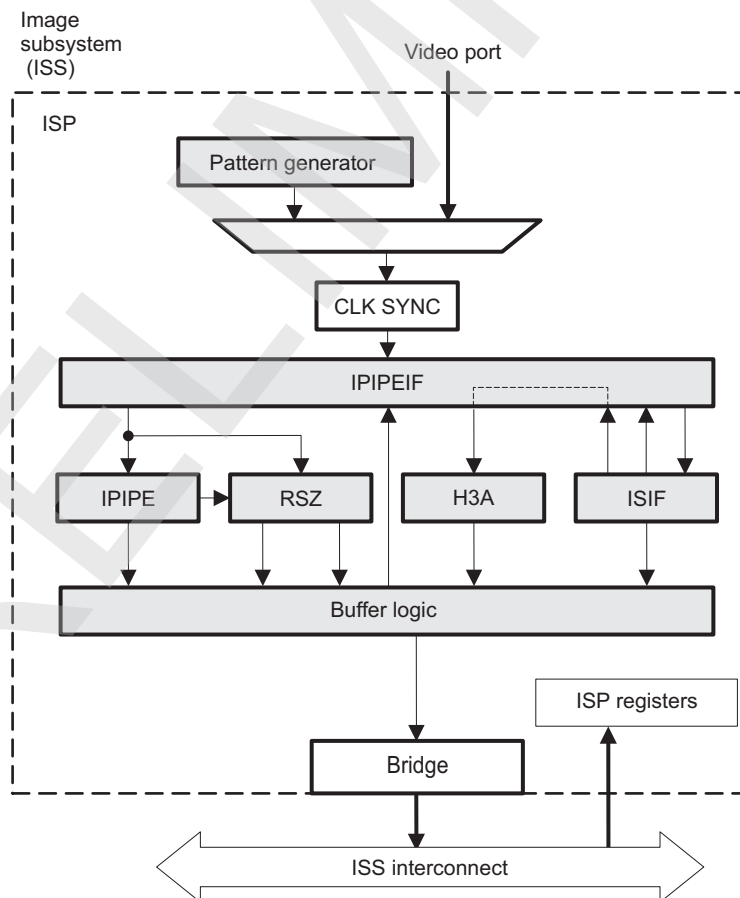
### 8.3.1.2 ISS ISP Block Diagram

Figure 8-102 is the ISP top-level block diagram. The ISS supports three simultaneous pixel flows (CCP2, CSI2 A, and CSI2 B), but only one of them can use the video-processing hardware at a time; the others can go directly to memory.

The ISP master port is connected to the level 3 (L3) interconnect, and the slave port is connected to the level 4 (L4) interconnect.

Figure 8-102 is the ISP block diagram.

**Figure 8-102. ISS ISP Block Diagram**



camss-003

### 8.3.2 ISS ISP Integration

This section describes the integration of the ISP modules in the ISS and includes information about clocks, resets, and hardware requests.

The ISP is part of the ISS of the device and is a key component for imaging and video applications such as camera viewfinder, video record, and still image capture.

ISS is coupled with a low interrupt latency ISS microprocessor unit (MPU) subsystem running a real-time operating system to reach optimal performance. Primarily, the ISS MPU subsystem can quickly change the ISS configuration during frame blanking periods and run some sequencing tasks.

#### 8.3.2.1 ISS ISP PRCM Interface

The ISP and its internal modules, as integrated in the ISS, use the same power and clock management details. See [Section 8.1.2, ISS Functional Description](#), for the ISP top-level power, reset, and clock management (PRCM) Interface.

##### 8.3.2.1.1 ISS ISP Clocks

The PRCM module, through the local power and clock management inside the ISS, provides a unique PCLK that can be enabled from the [ISP5\\_CTRL](#) register.

The modules inside the ISP require three clocks:

- PCLK: This clock is asynchronous to the other clocks. It is provided by the module sending the data to the IPIPEIF module on the VP.
- ISS\_FCLK: This clock is synchronous with the configuration clock domain. This is the clock used for the MTC interface.
- GCK\_MMR: This is the clock for the configuration bus. It is created from the ISS\_FCLK and runs at half the speed of the ISS\_FCLK.

##### 8.3.2.1.2 ISS ISP Reset

The ISP supports global software reset along with internal hardware reset, if needed.

Software reset is done through the [ISP5\\_SYSCONFIG\[1\]](#) SOFTRESET bit. Before issuing a software reset, the ISP must be in standby mode. The following must be done:

- Ensure that the interfaces must be stopped from sending data and/or the ISP modules are disabled. Before reset, the last interrupt triggered by the ISP when the frame processing completes is RSZ\_INT\_DMA. RSZ\_INT\_DMA must be used to enable clean termination of the processing. Software must wait a few hundred cycles to trigger a soft reset after RSZ\_INT\_DMA is asserted; this is to ensure that the BL is completely drained.
- Ensure that [ISP5\\_SYSCONFIG\[5:4\]](#) STANDBYMODE = 2 (smart standby). Write the [ISP5\\_CTRL\[24\]](#) MSTANDBY bit to 1 and poll for [ISP5\\_CTRL\[20\]](#) MSTANDBY\_WAIT = 1. Then, the soft reset can be applied ( [ISP5\\_SYSCONFIG\[1\]](#) SOFTRESET = 1).

In case an ISP overflow or underflow event happens (for example, RSZ\_FIFO\_OVF, ISIF\_OVF, etc.), it is not sufficient to reset the ISP. In that case a reset must take place at the ISS level.

##### 8.3.2.2 ISS ISP Interrupt Tree

[Table 8-307](#) summarizes the submodule interrupts that can be mapped to the four ISP interrupt output lines. After this, these output lines are mapped to the ISS top interrupt request (IRQ) merger (for more information, see [Section 8.1.2](#)). Software enables the explained before merger from the top-level ISS resources registers in [Section 8.1.2](#).

**Table 8-307. ISS ISP Interrupt Tree Table**

Register	Modules	Destination	Comments
<a href="#">ISP5_IRQENABLE_SET_i[9]</a> IPIPEIF_IRQ	IPIPEIF	ISP to ISS merger four IRQ lines	See <a href="#">Section 8.3.2.4.1</a> .
<a href="#">ISP5_IRQENABLE_SET2_i[1]</a> IPIPEIF_UDF			

**Table 8-307. ISS ISP Interrupt Tree Table (continued)**

Register	Modules	Destination	Comments			
ISP5_IRQENABLE_SET_i[29] IPIPE_INT_DPC_RNEW1	IPIPE		See <a href="#">Section 8.3.2.5.1</a> .			
ISP5_IRQENABLE_SET_i[28] IPIPE_INT_DPC_RNEW0						
ISP5_IRQENABLE_SET_i[27] IPIPE_INT_DPC_INI						
ISP5_IRQENABLE_SET_i[8] IPIPE_INT_HST						
ISP5_IRQENABLE_SET_i[7] IPIPE_INT_BSC						
ISP5_IRQENABLE_SET_i[6] IPIPE_INT_DMA						
ISP5_IRQENABLE_SET_i[5] IPIPE_INT_LAST_PIX						
ISP5_IRQENABLE_SET_i[4] IPIPE_INT_REG						
ISP5_IRQENABLE_SET_i[25] IPIPE_INT_EOF						
ISP5_IRQENABLE_SET2_i[4] IPIPE_HST_ERR						
ISP5_IRQENABLE_SET2_i[2] IPIPE_BOXCAR_OVF						
ISP5_IRQENABLE_SET_i[12] H3A_INT				H3A		See <a href="#">Section 8.3.2.7</a> .
ISP5_IRQENABLE_SET_i[24] H3A_INT_EOF						
ISP5_IRQENABLE_SET2_i[0] H3A_OVF						
ISP5_IRQENABLE_SET_i[23] RSZ_INT_EOF1	RSZ		See <a href="#">Section 8.3.2.6.2</a> .			
ISP5_IRQENABLE_SET_i[22] RSZ_INT_EOF0						
ISP5_IRQENABLE_SET_i[19] RSZ_FIFO_IN_BLK_ERR						
ISP5_IRQENABLE_SET_i[18] RSZ_FIFO_OVF						
ISP5_IRQENABLE_SET_i[17] RSZ_INT_CYC_RZB						
ISP5_IRQENABLE_SET_i[16] RSZ_INT_CYC_RZA						
ISP5_IRQENABLE_SET_i[15] RSZ_INT_DMA						
ISP5_IRQENABLE_SET_i[14] RSZ_INT_LAST_PIX						
ISP5_IRQENABLE_SET_i[13] RSZ_INT_REG	ISIF		See <a href="#">Section 8.3.2.8.1</a> .			
ISP5_IRQENABLE_SET_i[3] ISIF_INT_3						
ISP5_IRQENABLE_SET_i[2] ISIF_INT_2						
ISP5_IRQENABLE_SET_i[1] ISIF_INT_1						
ISP5_IRQENABLE_SET_i[0] ISIF_INT_0						
ISP5_IRQENABLE_SET2_i[3] ISIF_OVF						

### 8.3.2.3 ISS ISP PG Integration

See [Section 8.3.2](#).

### 8.3.2.4 ISS ISP IPIPEIF Integration

#### 8.3.2.4.1 ISS ISP IPIPEIF Interrupts

The IPIPEIF module generates two interrupts, one generic and one:

- IPIPEIF\_IRQ: This event is triggered to the BL module when a new frame starts (VS signal). The interrupt is active low and is asserted for one GCK\_MMR clock cycle.
- IPIPEIF\_UDF: Interrupt generated when an underflow happens in the IPIPEIF module. Underflow events are nonrecoverable at the ISP level and a soft reset is required at the ISS level.

The interrupts are enabled from the [ISP5\\_IRQENABLE\\_SET\\_i\[9\]](#) IPIPEIF\_IRQ and [ISP5\\_IRQENABLE\\_SET2\\_i\[1\]](#) IPIPEIF\_UDF bits (where  $i = 0$  to  $3$  for the line that will be mapped to the four lines of the ISP). Then, each line from the ISP is sent to the ISS top level, where it is muxed with other ISS modules for a total output of six interrupt lines. See [Section 8.1.2](#), *ISP Top -Level Integration*.



### 8.3.2.5 ISS ISP IPIPE Integration

#### 8.3.2.5.1 ISS ISP IPIPE Interrupts

IPIPE can generate several interrupts:

- IPIPE\_INT\_DPC\_RNEW: This event is triggered when there is permission to initialize LUT-DPC table lines 0 and 1.
- IPIPE\_INT\_DPC\_INI: This event is triggered when the DPC table is initialized.
- IPIPE\_INT\_HST: This event is triggered when the histogram is done.
- IPIPE\_INT\_BSC: This event is triggered when boundary signal calculation is done.
- IPIPE\_INT\_DMA: This event is triggered when the boxcar SDRAM transfer is done. On this timing, IPIPE\_INT\_EOF is sent to the BL. This event is active high for one GCK\_MMR clock cycle.
- IPIPE\_INT\_LAST\_PIX: This event is triggered when the last pixel of a frame comes into IPIPE. This event is active high for one GCK\_MMR clock cycle.
- IPIPE\_INT\_EOF: This event is triggered for end of frame.
- IPIPE\_BOXCAR\_OVF: This event is generated when an overflow happens in the IPIPE-BOXCAR output buffer. The interrupt avoids polling the IPIPE\_SRC\_STA[0] VAL0 bit for errors. Overflow events are nonrecoverable at the ISP level and a soft reset is required at the ISS level.
- IPIPE\_HST\_ERR: This event is triggered when the MPU or system direct memory access (sDMA) is still reading the memory that is being used by the module. This is an indication that the read operation was not fast enough.

The interrupts are enabled from the [ISP5\\_IRQENABLE\\_SET\*i\*](#) and [ISP5\\_IRQENABLE\\_SET2\*i\*](#) registers (where  $i = 0$  to 3 for the line that will be mapped to the four lines of the ISP). Then, each line from the ISP is sent to the ISS top level, where it is muxed with other ISS modules for a total output of six interrupt lines. See [Section 8.1.2, ISP Top-Level Integration](#).

#### 8.3.2.5.2 ISS ISP DMA Requests

The ISP generally outputs four direct memory access (DMA) requests, which can be used to read or write memories inside the IPIPE module. These memories are:

- BSC memory: This memory must be read during a vertical blanking period. It is used by the video stabilization application.
- HIST memory: This memory must be read during a vertical blanking period. It is used by the 3A application. The HIST data is double-buffered from frame-to-frame. Software must select the memory that is to be used by setting the IPIPE\_HST\_TBL[0] SEL bit.
- GAMMA memory: This memory must be set during a vertical blanking period. The imaging application typically uses multiple gamma tables.
- DPC memory: This memory must be set during a frame acquisition. The memory is not big enough to store all faulty pixels for a given frame.

To generate the DMA requests, the following events must be used:

- The IPIPE\_INT\_BSC event is used to generate the DMA request for the BSC memory. It maps on DMA line 0, ISS\_DMA0.
- The IPIPE\_INT\_HST event is used to generate the DMA request for the HIST memory. It maps on DMA line 1, ISS\_DMA1. The HIST data is double-buffered from frame-to-frame. Software must select the memory that is to be used by setting the IPIPE\_HST\_TBL[0] SEL bit. When the DMA request is set, it is required to read 4KB from the ping buffer address (0x2000) or the pong buffer address (0x3000). Software must ensure that when one buffer is selected no other accesses (for example, MPU) take place in the other buffer.
- The IPIPE\_INT\_LAST\_PIX event is used to generate the DMA request for the GAMMA memory. This same event can also be used to initialize the DPC LUT (not the preferred method; it is better to use IPIPE\_INT\_DPC\_INI). Basically, when the last pixel is output from the IPIPE module, it is safe to modify the IPIPE memories. It maps on DMA line 3, ISS\_DMA3.
- The IPIPE\_INT\_DPC\_INI event signals that DPC table memory initialization can take place. IPIPE\_INT\_DPC\_INI is used to generate two back-to-back DMA requests, the first one mapping on IPIPE\_INT\_DPC\_RNEW0 and the second one mapping on IPIPE\_INT\_DPC\_RNEW1. After

initialization (steady state), the IPIPE\_INT\_DPC\_RNEW0 and IPIPE\_INT\_DPC\_RNEW1 events are used to generate the DMA request for the DPC LUT renewal. It maps on DMA line 2, ISS\_DMA2. To select which event is used to initialize the DPC (IPIPE\_INT\_LAST\_PIX or IPIPE\_INT\_DPC\_INI), set the `ISP5_CTRL[25] DPC_EVT_INI` bit.

---

**NOTE:** The size of the DPC table in SDRAM must be exactly a multiple of the number of DMA requests. It ensures that during burst capture mode the DMA always load the correct data from frame-to-frame. The DMA automatically warps back to the start of the table after all expected DMA requests are received. There is a total of  $\text{ceil}(\text{nb\_faulty\_pixel}/128) + 2$  DMA requests per frame when the IPIPE\_INT\_DPC\_INI EVENT is used. Hence, the size of the DPC table in the SDRAM is  $(\text{ceil}(\text{nb\_faulty\_pixel}/128) + 2) \times 128 \times 32$  bits.

---



---

**NOTE:** If DMA channels are not used but the MPU is used, an error check must be performed for IPIPE HST . See `ISP5_CTRL[26] HST_RD_CHK` for details.

---

The DMA request assertion and deassertion to the sDMA is automatic and no software intervention is needed. The ISP contains two registers (`ISP5_DMAENABLE_SET` and `ISP5_DMAENABLE_CLR`) that must be used to enable or disable generation of the DMA requests.

The DMA request deassertion is based on late deassertion; that is, the DMA request is disabled only when all the data corresponding to the transfer size have been read or written. To deassert the DMA request and generate the `hw_eoi` signal, hardware counts the number of 32-bit accesses that are done in the memory range of the corresponding DMA request through the ISP slave port. When the number of accesses corresponding to the DMA request is done, the DMA request is cleared. Multiple DMA requests can be active simultaneously.

Software must not attempt to read or write in the memory range of the DMA requests that are enabled because CPU accesses, instead of sDMA accesses, will be counted. Software can freely access ISP memories for which the DMA request is disabled, and can access registers while the sDMA performs the transfers.

### 8.3.2.6 ISS ISP RSZ Integration

#### 8.3.2.6.1 ISS ISP RSZ PRCM Interface

##### 8.3.2.6.1.1 ISS ISP RSZ Reset

The resizer module has no standalone software reset. RSZ must be reset at the ISP level. See [Section 8.3.2.1.2, ISS ISP Reset](#).

##### 8.3.2.6.2 ISS ISP RSZ Interrupts

RSZ can generate several interrupts:

- `RSZ_INT_EOF0`: This event is triggered for end of frame.
- `RSZ_INT_EOF1`: This event is triggered for end of frame.
- `RSZ_FIFO_IN_BLK_ERR`: This event is triggered when the minimum vertical blanking period has not been respected, thus causing errors in the input data buffering submodule. This event is triggered when the `RSZ_INT_REG` event of frame N is triggered before `RSZ_INT_DMA` of frame N + 1. This event typically happens at the transition between two frames because there is not enough vertical blanking between frames. Hardware cannot recover from this error. It requires a reset. It is a requirement that despite the error, the RSZ module finishes correctly: ongoing requests are completed and further requests are blocked.
- `RSZ_FIFO_OVF`: This event is triggered when overflow happens in the input data buffering submodule. It typically occurs while processing a frame, because the VP pixel clock is too high. Because hardware cannot recover from this error, a reset is required. It is a requirement that despite the error, the RSZ module finishes correctly: ongoing requests are completed and further requests are blocked.

This event can be also triggered at the master write interface (MTC) where an MTC stall signal is generated.

- **RSZ\_INT\_CYC\_RZA/RSZ\_INT\_CYC\_RZB:** This event is triggered as circular interrupt every time that [RSZ\\_IRQ\\_RZA](#) / [RSZ\\_IRQ\\_RZB](#) output lines are written out to the RZA\_SDR\_Y/RZB\_SDR\_Y buffer. The range can go from 1 to 8192 lines. Usually, this value must be such that the circular buffer vertical size (set by the RZBA\_SDR\_Y\_PTR\_E / RZBB\_SDR\_Y\_PTR\_E register) is a multiple of [RSZ\\_IRQ\\_RZA/RSZ\\_IRQ\\_RZA](#).
- **RSZ\_INT\_DMA:** This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and the RSZ core has returned to idle. This event is active high for one GCK\_MMR clock cycle.
- **RSZ\_INT\_LAST\_PIX:** This event is triggered when the last pixel of the valid area is received. This event is active high for one GCK\_MMR clock cycle.
- **RSZ\_INT\_REG:** This event is triggered when the new value of the shadowed registers, if updated, take effect on the next RSZ\_INT\_REG event. Then again, shadowed registers can be updated for the next frame after the RSZ\_INT\_REG event is triggered. This event is active high for one GCK\_MMR clock cycle.

The interrupts are enabled from the [ISP5\\_IRQENABLE\\_SET\\_i](#) register (where i = 0 to 3 for the line that will be mapped to the four lines of the ISP). Then, each line from the ISP is sent to the ISS top level, where it is muxed with other ISS modules for a total output of six interrupt lines. See [Section 8.1.2, ISP Top-Level Integration](#).

### 8.3.2.7 ISS ISP H3A Integration

#### 8.3.2.7.1 ISS ISP H3A Interrupts

H3A can generate the following interrupts:

- **H3A\_INT:** This event is triggered at the end of the last window or last pixel, whichever completes last. This always triggers at the same time as H3A\_INT\_EOF.
- **H3A\_INT\_EOF:** This event is triggered and generated at the end of the last window. This event is active high for one GCK\_MMR clock cycle.
- **H3A\_OVF:** This interrupt is generated when an overflow happens in the H3A output buffer. The interrupt avoids polling the [H3A\\_PCR\[21\]](#) OVF register for errors. Overflow events are nonrecoverable at the ISP level and a soft reset is required at the ISS level. The event is active high for one GCK\_MMR clock cycle.

The interrupts are enabled from [ISP5\\_IRQENABLE\\_SET\\_i](#) and [ISP5\\_IRQENABLE\\_SET2\\_i](#) (where i = 0 to 3 for the line that will be mapped to the four lines of the ISP). Then, each line from the ISP is sent to the ISS top level, where it is muxed with other ISS modules for a total output of six interrupt lines. See [Section 8.1.2, ISP Top-Level Integration](#).

### 8.3.2.8 ISS ISP ISIF Integration

#### 8.3.2.8.1 ISS ISP ISIF Interrupts

ISIF can generate several interrupts:

- **ISIF\_INT\_0:** This event is triggered when the VD0 interrupt on line 0 is configured. The VD0 interrupt can be configured based on the VD position. It is asserted after receiving the number of horizontal lines (horizontal pulse signals) set in VDINT0. See [Section 8.3.3.7.19.1](#) for more details.
- **ISIF\_INT\_1:** This event is triggered when the VD1 interrupt on line 1 is configured. The VD1 interrupt can be configured based on the VD position. It is asserted after receiving the number of horizontal lines (horizontal pulse signals) set in VDINT1. See [Section 8.3.3.7.19.1](#) for more details.
- **ISIF\_INT\_2:** This event is triggered when the VD2 interrupt on line 2 is configured. The VD2 interrupt can be configured based on the VD position. It is asserted after receiving the number of horizontal lines (horizontal pulse signals) set in VDINT2. See [Section 8.3.3.7.19.1](#) for more details.
- **ISIF\_INT\_3:** This event is triggered LSC interrupt is an interrupt issued by the 2D-LSC block. See [Section 8.3.3.7.10.1.5](#) for more details.
- **ISIF\_OVF:** Interrupt generated when an underflow happens in the ISIF module. The interrupt avoids polling the [ISIF\\_MODESET\[11\]](#) OVF register for errors.

The interrupts are enabled from the [ISP5\\_IRQENABLE\\_SET\\_i](#) and [ISP5\\_IRQENABLE\\_SET2\\_i](#) registers (where  $i = 0$  to 3 for the line that will be mapped to the four lines of the ISP). Then, each line from ISP is sent to the ISS top level where it is muxed with other ISS modules for a total output of six interrupt lines. See [Section 8.1.2](#), *ISP Top-Level integration*.

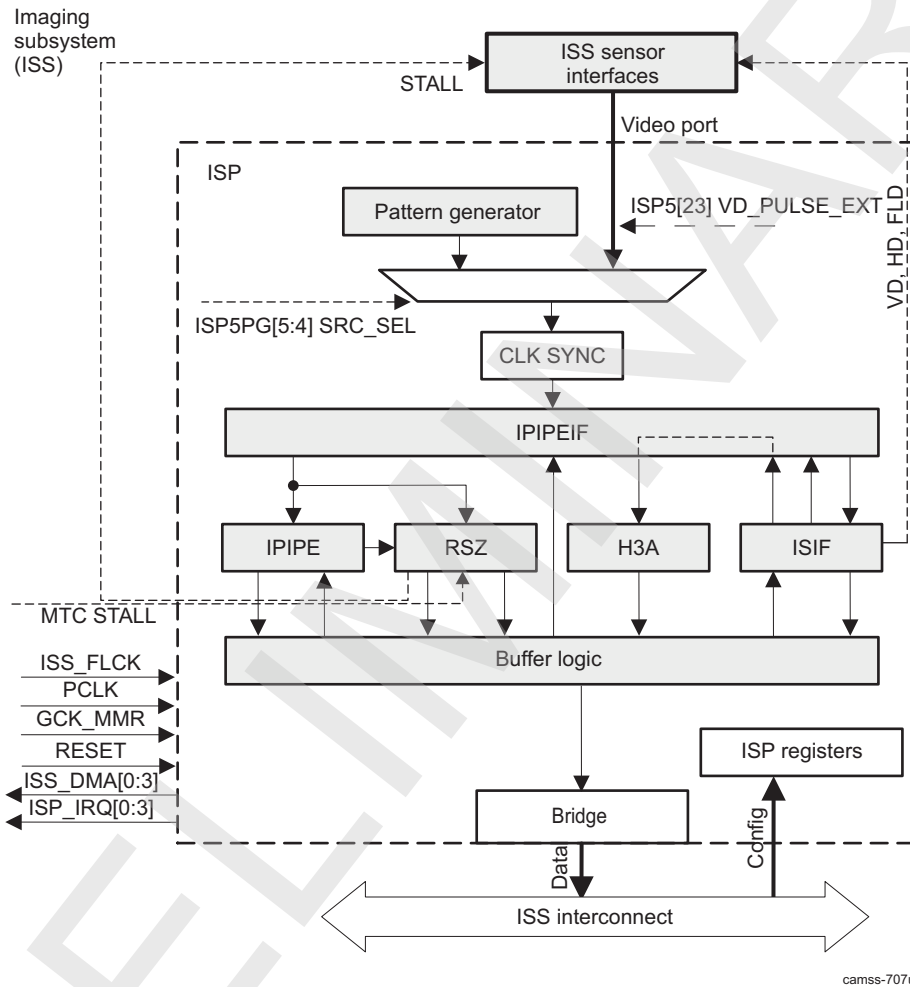
### 8.3.2.9 ISS ISP BL Integration

See [Section 8.3.2](#).

### 8.3.3 ISS ISP Functional Description

The functionality of the ISP is part of the overall performance of the ISS. For the top-level ISS diagram with ISP inside and for a features list, see [Section 8.1, ISS Overview](#). [Figure 8-103](#) is an overview of the ISP module. It outputs DMA and interrupt requests, has clocks coming in and a stall signal to the CCP2 interface receiver from the resizer. It also shows the top-level configuration for input to IPIPEIF. For the scaled-in functional details of each submodule inside the ISP, see its functional description section.

**Figure 8-103. ISS ISP High-Level Diagram**



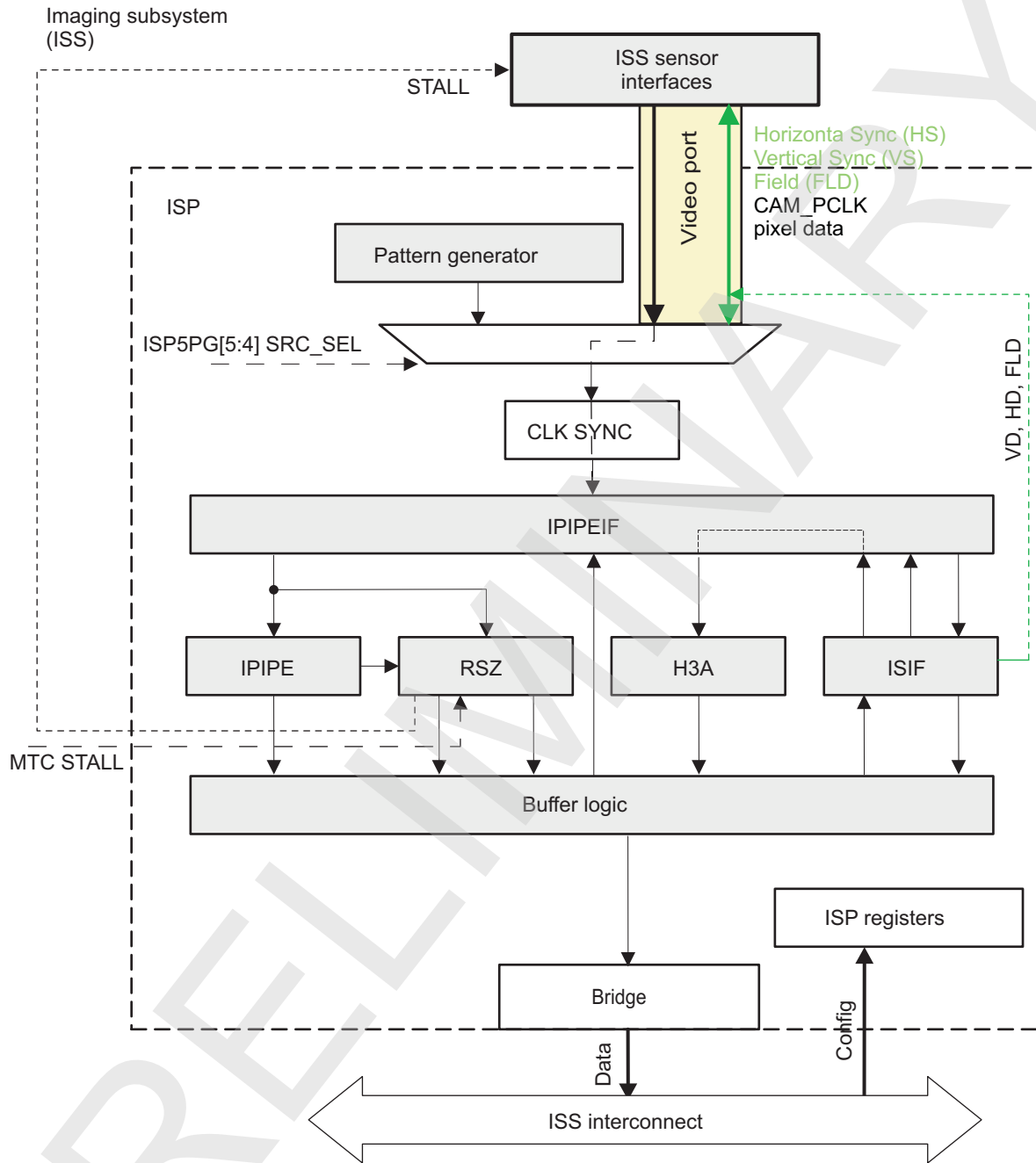
#### 8.3.3.1 ISS ISP VP Functional Description

##### 8.3.3.1.1 ISS ISP VP Overview

The VP supports a parallel interface that is used for interfacing with image sensors. The ISP VP can transport 8- to 16-bit RAW data and 8-/16-bit YCbCr data.

[Figure 8-104](#) lists the VP module connections to other submodules of the ISP.

**Figure 8-104. ISS ISP VP High-Level Diagram**



camss-699

**8.3.3.1.2 ISS ISP VP Data Formats**

The VP can be used to connect external camera receivers to the ISS. Data paths inside the ISP hardware depend on the image format sourced by the sensor (RAW RGB, YUV4:2:2, JPEG, etc.). [Table 8-308](#) shows how the CCP2/CSI2 modules are connected to the VP in function of the image format.

Table 8-308. ISS ISP VP Format Mapping

Source		Format	Connected to															ISIF GW DI	Data Provided to ISIF Linearization Engine																		
CCP 2	CSI2		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1		0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
X	X	RAW16	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0		
X	X	RAW14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0		
			0	0	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	1	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0		
X	X	RAW12	0	0	0	0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0		
			0	0	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	1	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0		
			R11	R10	R9	R8	R7	R5	R5	R4	R3	R2	R1	R0	0	0	0	0	2	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0		
X	X	RAW10	0	0	0	0	0	0	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0		
			0	0	0	0	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	1	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0		
			0	0	R9	R8	R7	R5	R5	R4	R3	R2	R1	R0	0	0	0	0	2	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0		
			R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	3	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0		
X	X	RAW8								R8	R7	R6	R5	R4	R3	R2	R1	R0	0	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0			
										R7	R6	R5	R4	R3	R2	R1	R0			1	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0		
									R7	R6	R5	R4	R3	R2	R1	R0					2	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0	
						R7	R6	R5	R4	R3	R2	R1	R0									3	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0
			R7	R5	R5	R4	R3	R2	R1	R0												4	R7	R6	R5	R4	R3	R2	R1	R0	0	0	0	0	0	0	0
		YUV 16-bit	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	C7	C6	C5	C4	C3	C2	C1	C0	0																		
		YUV 8-bit									YC7	YC6	YC5	YC4	YC3	YC2	YC1	YC0	0																		

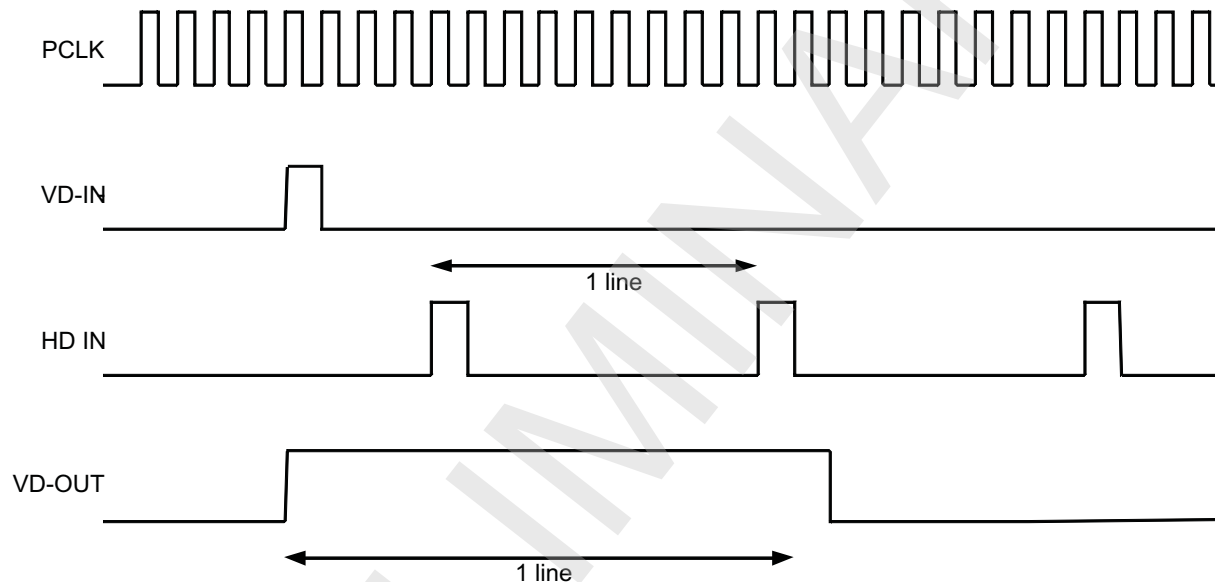
8.3.3.1.3 ISS ISP VP Top-Level Communication With CCP2 RX and CSI2 RX

At the ISS level, the VP is connected to the VP of the CCP2 RX and CSI2 RX modules. VP implementation differences force the introduction of a bridge between the CCP2 RX/CIS2 RX modules and the VP. The role of the bridge is to perform VD pulse extension. The CCP2 RX module assumes that the VD signal is active for at least one pixel clock cycle, and the CSI2 RX module assumes that the VD signal is asserted for four pixel clock cycles. However, the ISP assumes that the VD pulse is active on at least one line.

Figure 8-105 shows how the VD pulse extension works. Assume that VD-IN is the VD signal at the input of the pulse extension bridge, and VD-OUT is the VD signal at the output of the pulse extension bridge.

The VD-OUT signal is asserted at the same time as VD-IN. The VD-OUT signal is kept high until one full line is received. A line is delimited by two rising edges of the HD signal. The VD-OUT pulse is deasserted on the next cycle after the falling edge of the HD signal.

**Figure 8-105. ISS ISP VP VD Pulse**



camss-691

The `ISP5_CTRL[23] VD_PULSE_EXT` bit controls whether the VD extension bridge is enabled or disabled. By default, the bridge is enabled. When the bridge is disabled, the VD pulse must be unmodified:  $VD\_OUT = VD\_IN$ . At the ISS level, it is expected that `ISP5_CTRL [23] VD_PULSE_EXT = 1` when the VP gets data from the CSI2 RX modules and `ISP5_CTRL[23] VD_PULSE_EXT = 0` when the VP gets data from the parallel interface or CCP2 RX module.

**CAUTION**

A minimum of four lines per frame are required on the VP when the VD pulse extension bridge is enabled; therefore, the VD extension bridge is not functional if a 1-/2-/3-line frame is sent to the VP.



#### **8.3.3.1.4 ISS ISP VP Pixel Clock Inversion**

The ISP always uses the rising edge of the pixel clock to sample the pixel data. The ISP provides the capability to invert the pixel clock so it can shift the resampling of a pixel clock period by half. This is controlled by the [ISP5\\_CTRL\[22\] PCLK\\_INV](#) bit. By default, the inversion is disabled. The 5 bits in [Table 8-309](#) are resynchronized from the GCK\_MMR clock domain to the PCLK clock domain. There must be at least three clock cycles between the time these bits are modified and the HD/VD pulse for start of frame comes.

PRELIMINARY

**Table 8-309. ISS ISP VP GCK\_MMR to PCLK Clock Resynchronization**

Module	Register	Bit Field
ISP	<a href="#">ISP5_CTRL</a>	VD_PULSE_EXT
ISIF	<a href="#">ISIF_MODESET</a>	HDVDD
ISIF	<a href="#">ISIF_MODESET</a>	FIDD
ISP	<a href="#">ISP5_CTRL</a>	ISIF_CLK_ENABLE

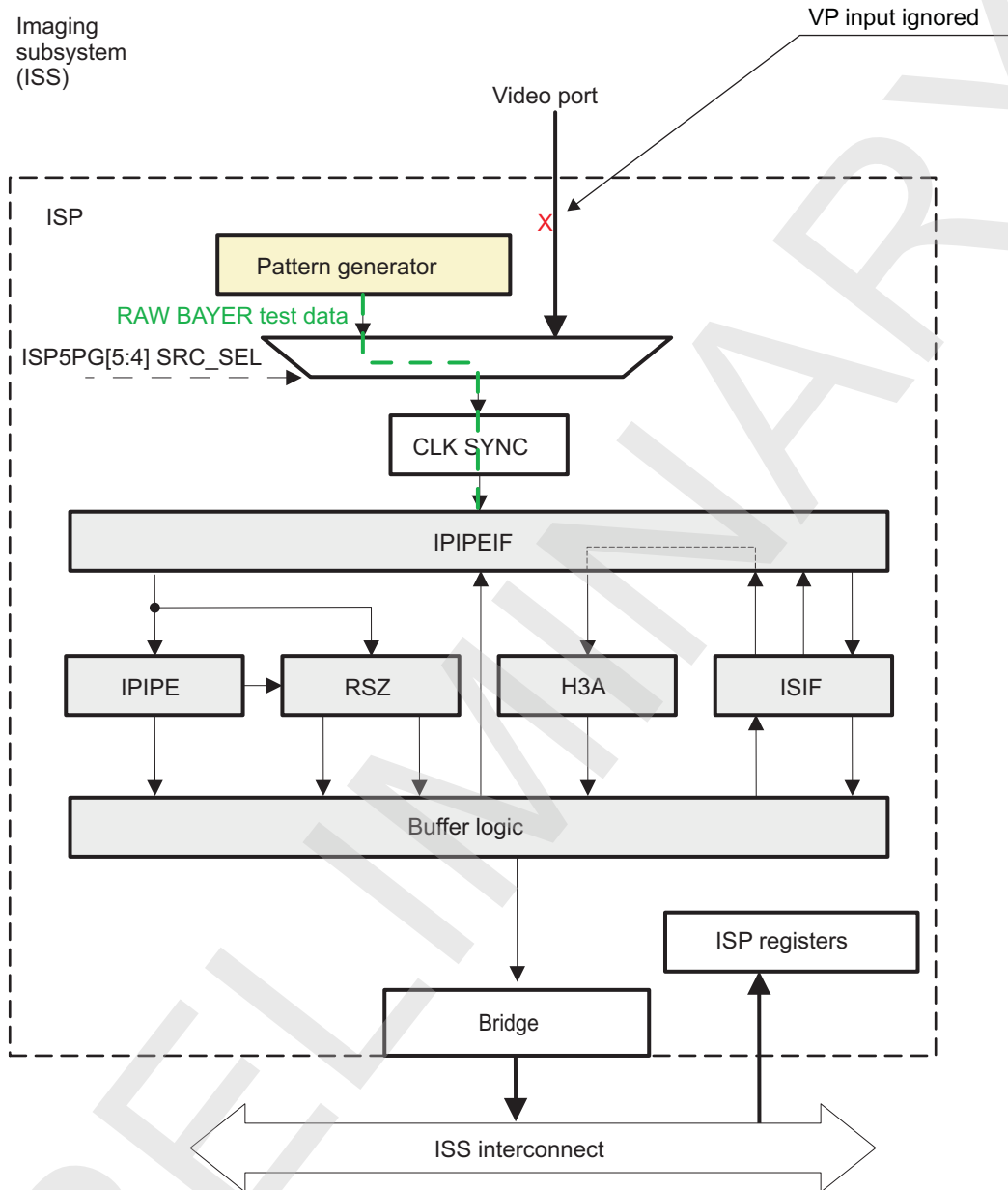
### 8.3.3.2 ISS ISP PG Functional Description

#### 8.3.3.2.1 ISS ISP PG Overview

The pattern generator (PG) implements an internal data generation mechanism to test the external pins and can generate RAW data (to the IPIPEIF, ISIF, and other ISP modules) without the need for an external image sensor.

[Figure 8-106](#) show the PG module connections to other submodules of the ISP.

Figure 8-106. ISS ISP PG High-Level Diagram



camss-686

### 8.3.3.2.2 ISS ISP PG Functional Description

The data pattern is as follows:

- Data remain unchanged from frame to frame.
- Data are always generated as a BAYER pattern with RED (only on even/even pixel/line; other locations are RED = 0), GREEN (only on odd/even pixel/line and even/odd pixel/line; other locations are GREEN = 0) and BLUE (only on odd/odd pixel/line; other locations are BLUE = 0) colors.
- The frame is divided into four equal horizontal sections:
  - 1st horizontal section is further divided into eight equal vertical sections
    - 1st vertical section is RED only with maximum value (depending on number of bits programmed).
    - 2nd vertical section is RED and GREEN with maximum value.

- 3rd vertical section is RED, GREEN, and BLUE with maximum value.
- 4th vertical section RED and BLUE with maximum value.
- 5th vertical section is all zeros.
- 6th vertical section is GREEN and BLUE with maximum value.
- 7th vertical section is GREEN with maximum value.
- 8th vertical section is BLUE with maximum value.
- 2nd horizontal section is all RED with the pixel value linearly incrementing (reset to zero for each line) from zero onwards.
- 3rd horizontal section is all GREEN with the pixel value linearly incrementing (reset to zero for each line) from zero onwards.
- 4th horizontal section is all BLUE with the pixel value linearly incrementing (reset to zero for each line) from zero onwards.

By default, the ISP source is the VP. To select PG as the source for the ISP, set the `ISP5_PG[5:4] SRC_SEL` bit field to 0x3.

The `ISP5_PG[2]` WEN bit controls write-enable signal generation for the PG module. There are two modes: the write enable signal is enabled all the time or it is enabled and disabled every eight clock cycles. The `ISP5_PG[1]` HDPOL and `ISP5_PG[0]` VDPOL bits control the polarity of the horizontal and vertical synchronization signals generated by the PG module. These signals can be active high or active low.

The `ISP5_PG_PULSE_CTRL` register controls the HS and VS signal widths for the PG module.

- The `ISP5_PG_PULSE_CTRL[12:0]` HDW bit field sets the duration of the HD signal in terms of pixel clock cycles. The actual duration of the signal is `ISP5_PG_PULSE_CTRL[12:0] HDW + 1` cycle. The minimum signal duration is one cycle, and the maximum signal duration is 4096 cycles.
- The `ISP5_PG_PULSE_CTRL[27:16]` VDW bit field sets the duration of the VD signal in terms of lines. The actual duration of the signal is `ISP5_PG_PULSE_CTRL[27:16] VDW + 1` line. The minimum signal duration is one line, and the maximum signal duration is 4096 lines.

The `ISP5_PG_FRAME_SIZE` register controls the PG module horizontal and vertical frame sizes.

- The `ISP5_PG_FRAME_SIZE[31:16]` PPLN bit field sets the number of pixels per line a frame generated by the PG module. The actual number of pixels in a line is set by `ISP5_PG_FRAME_SIZE[31:16] FRAME_PPLN + 1`. The minimum number of pixels is 1.
- `ISP5_PG_FRAME_SIZE[15:0]` HLPFR sets the number of lines per frame generated by the PG module. The actual number lines is set by `ISP5_PG_FRAME_SIZE[15:0] HLPFR + 1`. The minimum number of lines is one.

The `ISP5_PG[3]` EN bit enables or disables the PG module.

### 8.3.3.3 ISS ISP IPIPEIF Functional Description

#### 8.3.3.3.1 ISS ISP IPIPEIF Overview

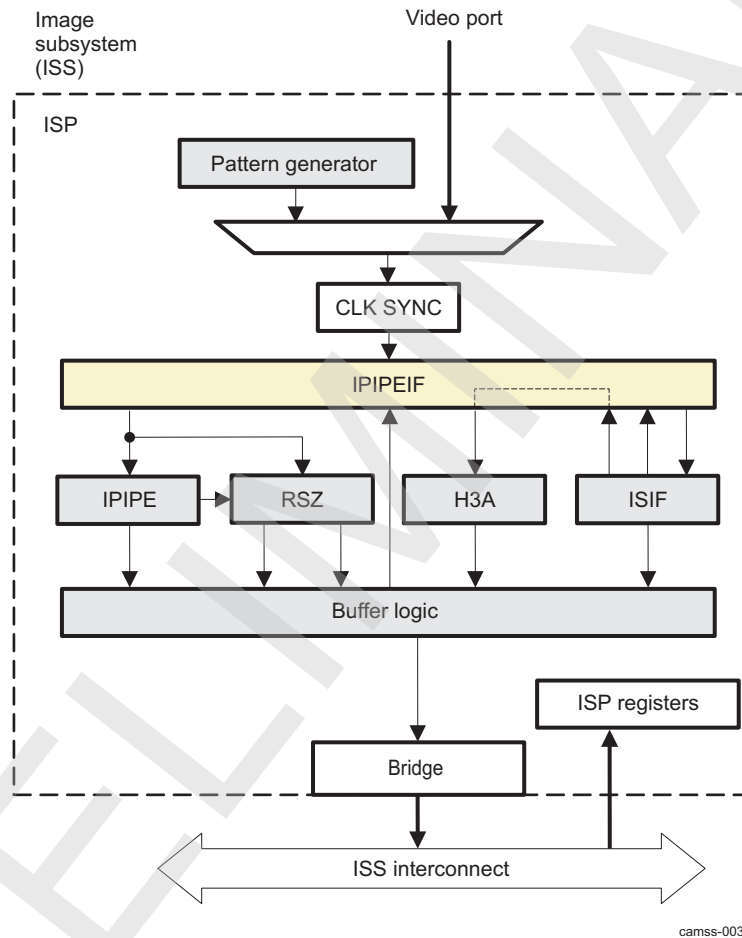
The IPIPEIF module provides data and synchronization signals (HD, VD) for the ISIF, IPIPE, RSZ, and H3A modules. The data source of this module is the VP, ISIF, or SDRAM using BL, and the selected data is output to ISIF, IPIPE, H3A, and RSZ. This module supports the following features:

- Supports up to 16 bpp data on the VP
- Supports up to 200-MHz pixel clock on the VP, up to 8K × 8K imaging resolution
- Supports RAW and YUV data formats on the VP and BL ports
- Supports dark-frame subtract of 8-bit RAW image stored in SDRAM from image from VP
- Supports dark-frame subtract of 8-bit RAW image stored in SDRAM from image from ISIF
- Supports dark-frame subtract of 8-bit RAW image from VP from image read from SDRAM through the BL
- Supports simple defect correction to prevent the subtraction of defect pixels
- Supports 8-10, 8-12 DPCM of 10-8, 12-8 DPCM compressed data in SDRAM
- Supports simple and advanced DPCM predictor

- Supports inverse A-Law decompression of RAW data 10-8 A-Law compressed from SDRAM
- Supports 8-bit, 12-bit unpacking of 8-bit,12-bit packed SDRAM data
- Supports gain multiply for output data to IPIPE module
- Supports horizontal Bayer rescaler in the data paths to the IPIPE and H3A modules: supports (1, 2, 1) averager filter and supports horizontal pixel decimation
- Supports data rate control when reading data from SDRAM: fraction clock divider.
- Supports (1, 2, 1) averager filter and supports horizontal pixel decimation in the data path to the IPIPE for YUV data.

Figure 8-107 show the IPIPEIF module connections to other submodules of the ISP.

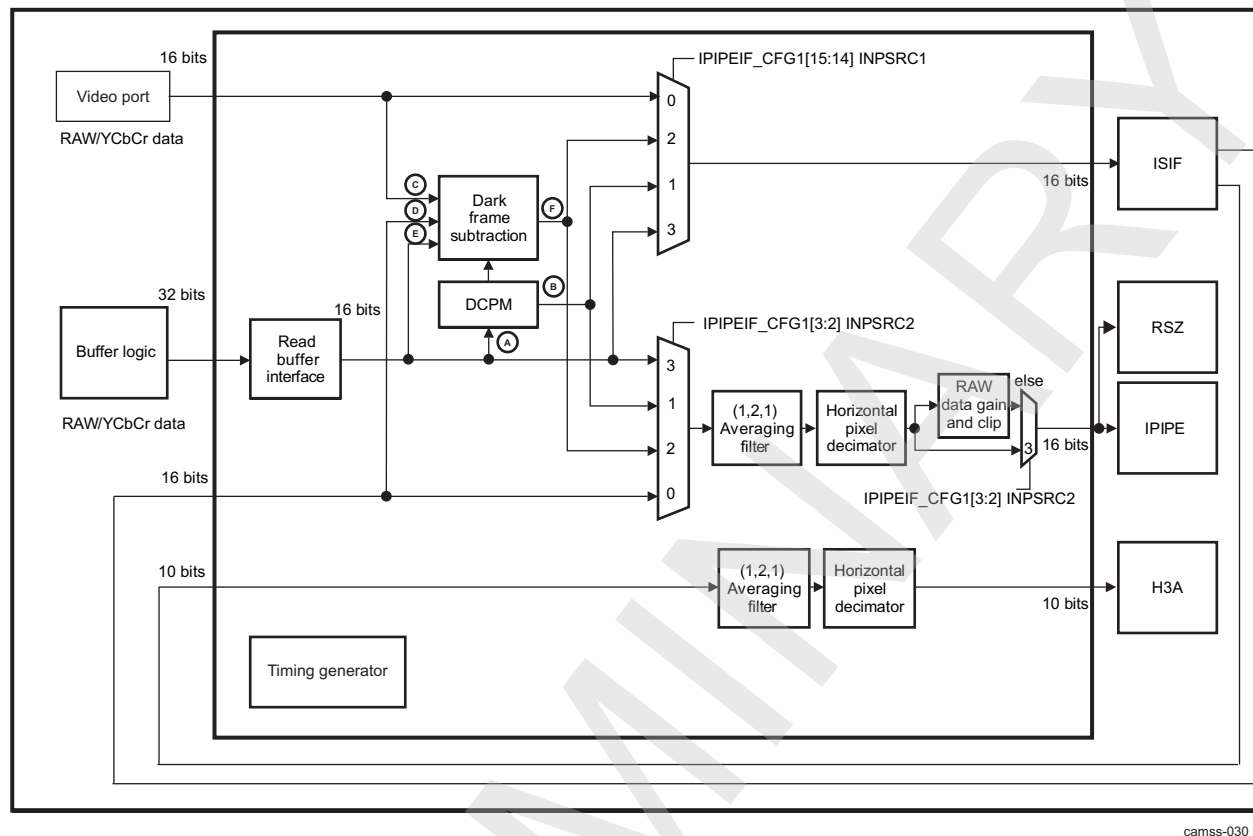
**Figure 8-107. ISS ISP IPIPEIF High-Level Diagram**



### 8.3.3.3.2 ISS ISP IPIPEIF Top-Level Block Diagram

The following sections describe the function of each subblock in the IPIPEIF, as shown in Figure 8-108.

Figure 8-108. ISS ISP IPIPEIF Top-Level Block Diagram



camss-030

**NOTE:** When the IPIPEIF receives data from the VP, the timing generator must be configured for HD, VD, and WEN. For more information, see [Section 8.3.3.3.5, ISS ISP IPIPEIF Timing Generation](#).

### 8.3.3.3.3 ISS ISP IPIPEIF Input Interface

The IPIPEIF module comprises two major interface blocks: VP and BL. The data types can be RAW or YUV.

#### 8.3.3.3.3.1 ISS ISP IPIPEIF Input From VP

The VP typically receives data from the image sensor. At the ISS level, it is connected to the serial interface receivers.

#### 8.3.3.3.3.2 ISS ISP IPIPEIF Input From BL

The BL is the interface with the memory (SDRAM). In that case, the SDRAM address and line offset registers must be programmed in units of 32 bytes.

- SDRAM start address (byte) =  $(\text{IPIPEIF\_ADDRU}[10:0] \text{ ADDRU}) \ll 16 + (\text{IPIPEIF\_ADDRL}[15:0] \text{ ADDR L})$
- SDRAM address offset (byte) =  $\text{IPIPEIF\_ADOFS}[11:0] \text{ ADOFS}$

Two types of data can be stored in memory: pixel data and dark frame data.

For pixel data, the HD and VD signals are reconstructed with:

- [IPIPEIF\\_HNUM](#)
- [IPIPEIF\\_VNUM](#)

- IPIPEIF\_LPFR
- IPIPEIF\_PPLN

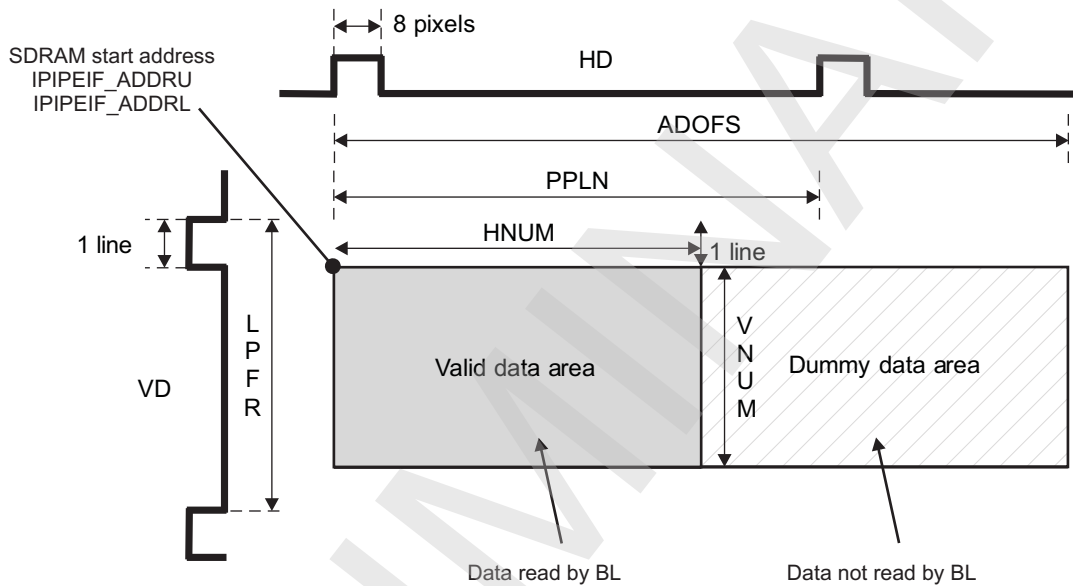
The IPIPEIF\_HNUM and IPIPEIF\_VNUM registers define the number of pixels per line and lines per frame to read from the SDRAM, and the IPIPEIF\_LPFR and IPIPEIF\_PPLN registers define the interval of VD and HD, respectively.

Vertical blanking for the frame is defined with the following equation:  $IPIPEIF\_LPFR \cdot IPIPEIF\_VNUM \cdot 1$ .

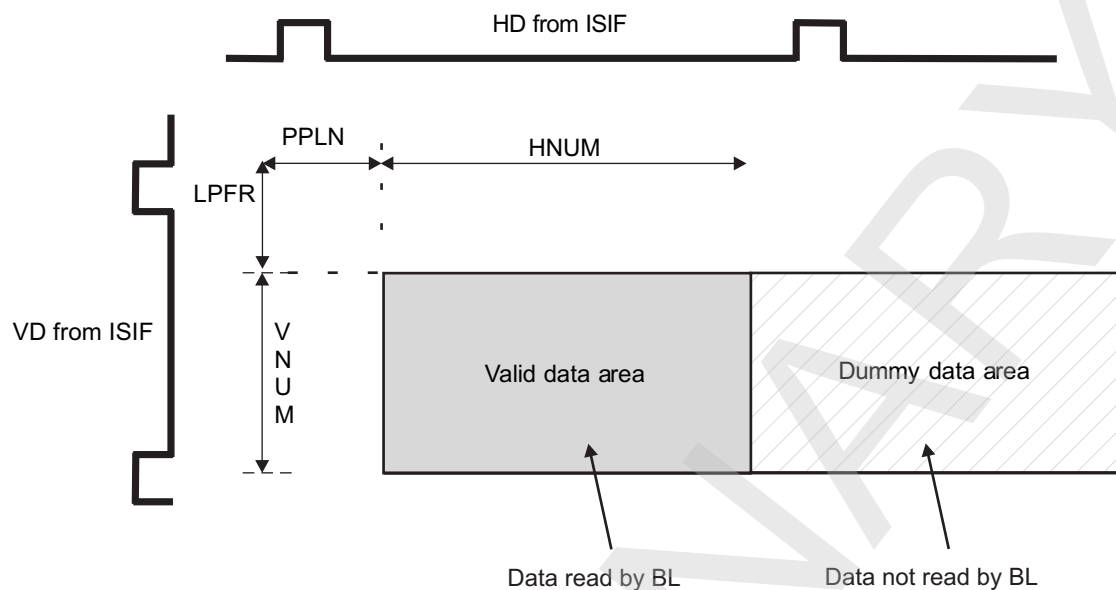
Horizontal blanking for the frame is defined with the following equation:  $IPIPEIF\_PPLN \cdot IPIPEIF\_HNUM$ .

Figure 8-109 shows the global frame definition for all SDRAM input modes, except for dark frame subtract.

**Figure 8-109. ISS ISP IPIPEIF Global Frame Definition in SDRAM Input Modes (Except Dark Frame)**



For dark frame data, the HD and VD signals come from the VP through the ISIF. The IPIPEIF\_PPLN and IPIPEIF\_LPFR registers should be used to indicate the horizontal and vertical start position of the subtraction from the ISIF data, as shown in Figure 8-110. The value of the IPIPEIF\_LPFR [12:0] LPFR bit field must be greater than 0 because the first line from the VP or ISIF cannot be subtracted from. The IPIPEIF\_HNUM and IPIPEIF\_VNUM registers must be used to set the number of valid pixels horizontally and the number of valid lines vertically.

**Figure 8-110. ISS ISP IPIPEIF Global Frame Definition in Dark Frame Subtract Mode**


#### 8.3.3.3.2.1 ISS ISP IPIPEIF Double-Buffer Input Function When Reading From BL

The IPIPEIF module supports a double-buffer input function. This feature is most useful when SDRAM space is limited, because it enables to read continuously from two buffers and to push data to the rest of the ISP (ISIF, H3A, etc.) for further processing.

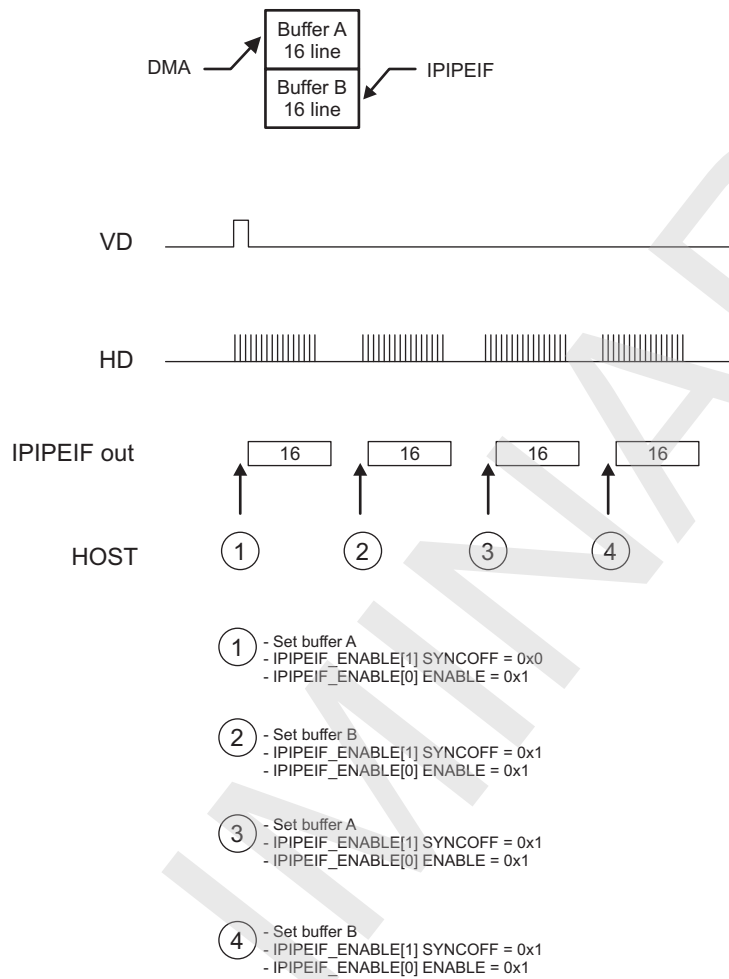
Consider the following configuration where data are read from two buffers, A and B. The intent is not only to read continuously from these buffers but also to ensure that the ISP modules consider the data as being from the same frame; that is, VD is generated the first time buffer A is read, but it must not toggle until all the frames are read.

The IPIPEIF module can mask the VD sync signal by writing the `IPIPEIF_ENABLE[1] SYNCOFF` bit but such that the IPIPEIF module drives the data to the ISP modules as if it is a continuous frame data.

In the following example, there are 16 lines per trigger and input circular addressing.  $VNUM = 16$  (see [Figure 8-109](#)), and the VD signal is generated only for the first frame (see [Figure 8-111](#)).



Figure 8-111. ISS ISP IPIPEIF Double-Buffer Functionality



camss-615

### 8.3.3.3.4 ISS ISP IPIPEIF Data Path Selection

The data path configuration through the IPIPEIF module is set with the [IPIPEIF\\_CFG1\[15:14\] INPSRC1](#) and [IPIPEIF\\_CFG1\[3:2\] INPSRC2](#) bit fields. [Table 8-310](#) lists the possible combinations for these two bit fields.

Table 8-310. ISS ISP IPIPEIF IPIPEIF\_CFG1[15:14] INPSRC1 and IPIPEIF\_CFG1[3:2] INPSRC2 Possible Combinations

IPIPEIF_CFG1[15:14] INPSRC1	IPIPEIF_CFG1[3:2] INPSRC2	Description	Common Use
0	0	This data path is described in <a href="#">Section 8.3.3.3.4.1, ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 0</a> .	Video record, view finder, on-the-fly still image capture applications
0	1	This data path is described in <a href="#">Section 8.3.3.3.4.2, ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 1</a> .	Memory to IPIPE to memory
0	2	This data path is described in <a href="#">Section 8.3.3.3.4.3, ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 2</a> .	Image capture with on-the-fly dark frame subtraction

**Table 8-310. ISS ISP IPIPEIF IPIPEIF\_CFG1[15:14] INPSRC1 and IPIPEIF\_CFG1[3:2] INPSRC2 Possible Combinations (continued)**

IPIPEIF_CFG1[15:14] INPSRC1	IPIPEIF_CFG1[3:2] INPSRC2	Description	Common Use
0	3	This data path is described in <a href="#">Section 8.3.3.3.4.4</a> , <i>ISS ISP IPIPEIF INPSRC1 = 0 and INPSRC2 = 3</i> .	On-the-fly data acquisition done in VP, forwarded to the ISIF, and then to the H3A through IPIPEIF while data from memory is processed and forwarded to the IPIPE module and then stored in memory YUV4:2:2 or RAW data processing with the IPIPE and RESIZER modules from memory to memory fetched by IPIPEIF
1	0	This data path is described in <a href="#">Section 8.3.3.3.4.5</a> , <i>ISS ISP IPIPEIF INPSRC1 = 1 and INPSRC2 = 0</i> .	Memory-to-ISIF- to memory operation
1	1	This data path is possible but there is no use case associated.	N/A
1	2	This data path is not supported.	N/A
1	3	This data path is not supported.	N/A
2	0	This data path is described in <a href="#">Section 8.3.3.3.4.6</a> , <i>ISS ISP IPIPEIF INPSRC1 = 2 and INPSRC2 = 0</i> .	Dark frame subtraction is performed and data sent to the ISIF module for further processing, back to IPIPEIF, and then to IPIPE and RSZ.
2	1	This data path is not supported.	N/A
2	2	This data path is not supported.	N/A
2	3	This data path is not supported.	N/A
3	0	This data path is described in <a href="#">Section 8.3.3.3.4.7</a> , <i>ISS ISP IPIPEIF INPSRC1 = 3 and INPSRC2 = 0</i> .	Memory-to-ISIF-to memory operation
3	1	This data path is not supported.	N/A
3	2	This data path is not supported.	N/A
3	3	This data path is possible but there is no use case associated.	N/A

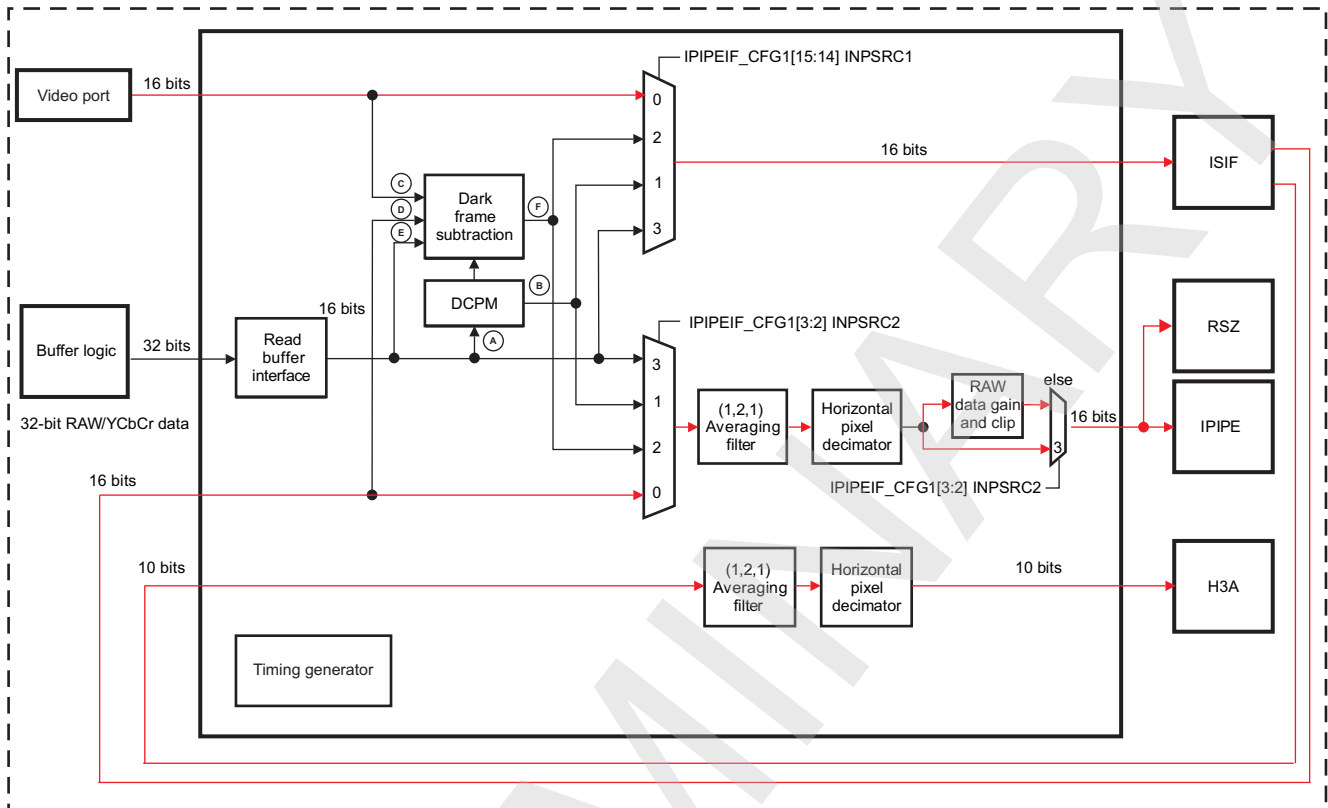
#### 8.3.3.3.4.1 ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 0

Set the IPIPEIF\_CFG1[15:14] INPSRC1 bit field to 0 and the IPIPEIF\_CFG1[3:2] INPSRC2 bit field to 0.

This configuration can be used for the video record, viewfinder, and on-the-fly still image capture applications. The full ISP processing capability is used in a single pass.

[Figure 8-112](#) shows the data path.

Figure 8-112. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 0 Data Path



camss-603

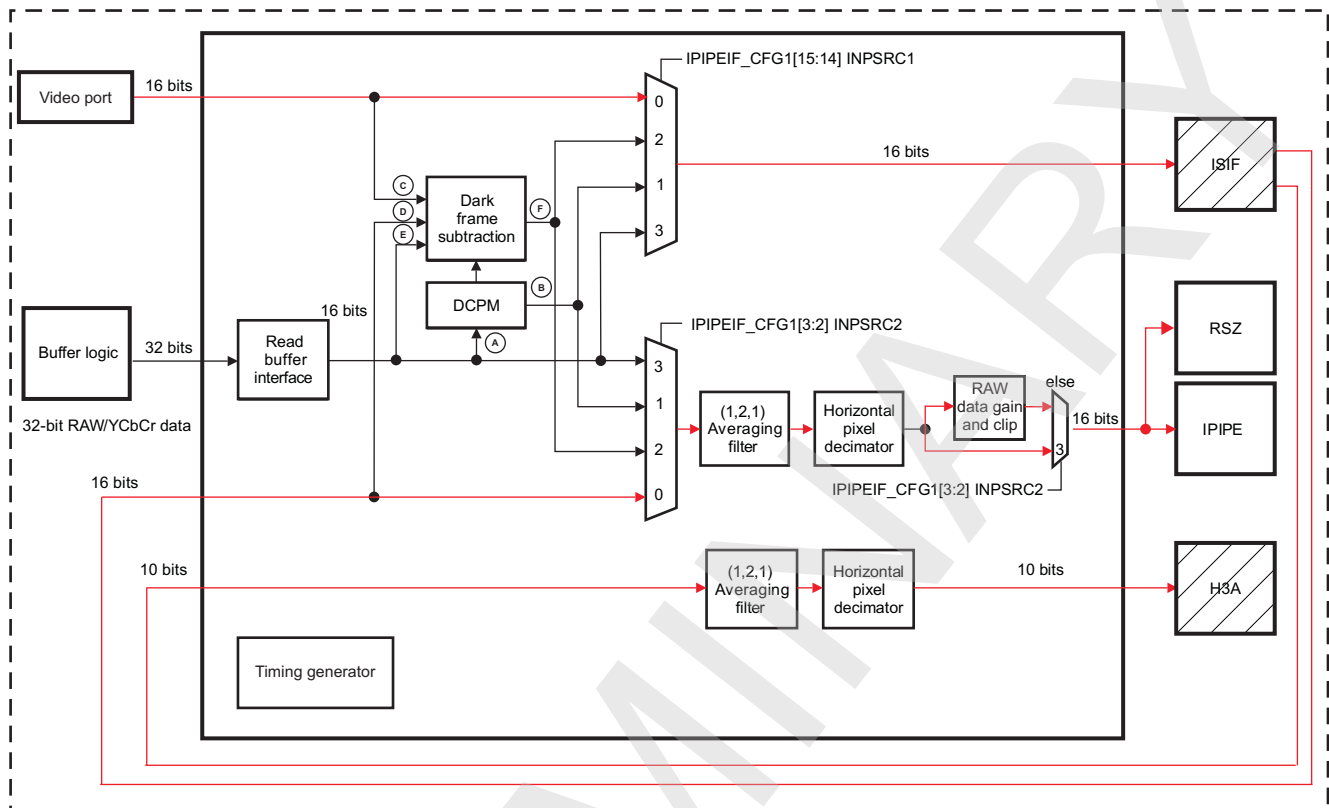
8.3.3.3.4.2 ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 1

Set the `IPIPEIF_CFG1[15:14] INPSRC1` bit field to 0 and the `IPIPEIF_CFG1[3:2] INPSRC2` bit field to 1.

This configuration can be used to process data with the IPIPE module from memory to memory. The data stored in SDRAM can be decompressed (A-law or DPCM) before being forwarded to the IPIPE module.

**NOTE:** In this configuration, the ISIF and H3A modules are assumed to be disabled.

Figure 8-113 shows the data path.

**Figure 8-113. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 1 Data Path**


camss-604

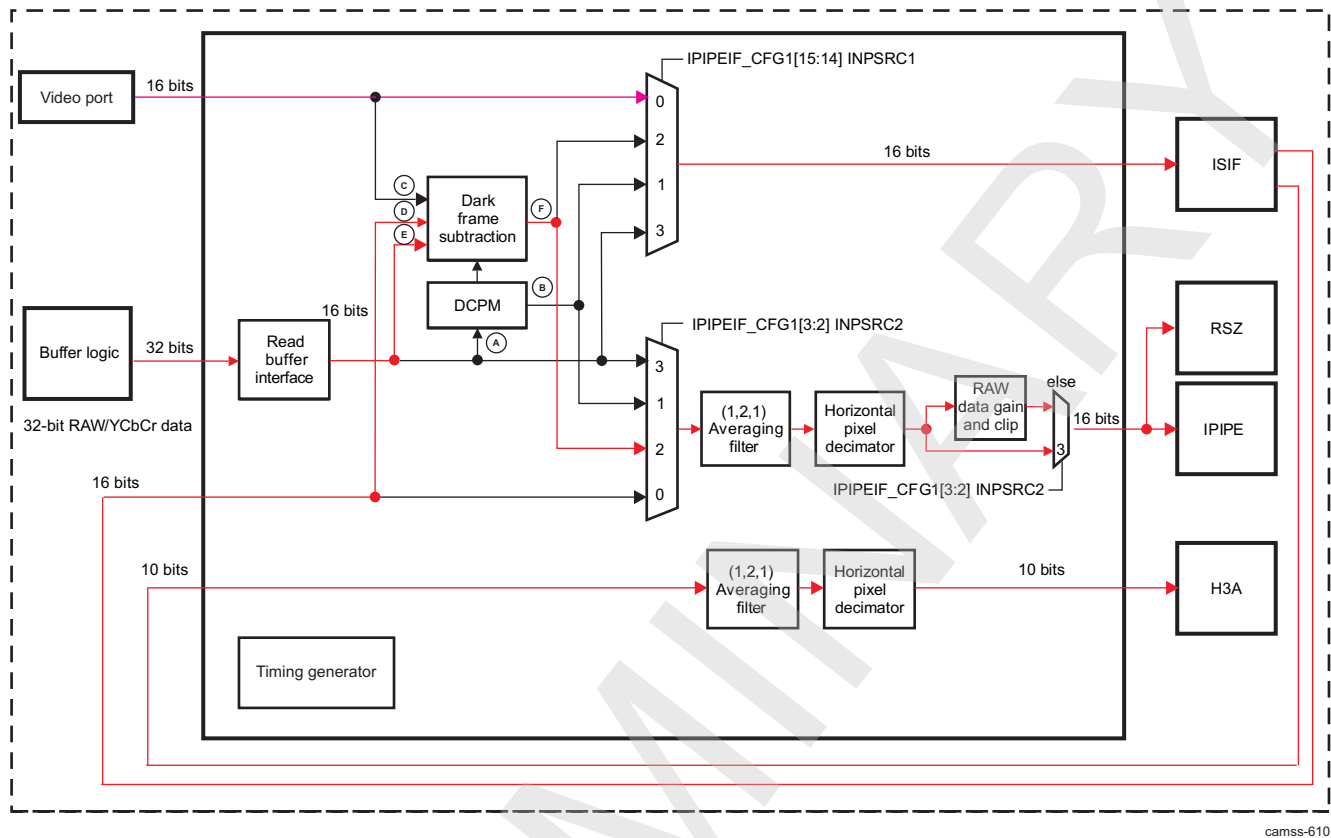
#### 8.3.3.3.4.3 ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 2

Set `IPIPEIF_CFG1[15:14] INPSRC1` to 0 and `IPIPEIF_CFG1[3:2] INPSRC2` to 2.

This configuration can be used for image capture with on-the-fly dark-frame subtraction. In the first case, the dark frame can come from BL and data from the ISIF. In the second case, the dark frame can come from the VP and data from BL.

Figure 8-114 shows the data path.

Figure 8-114. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 2 Data Paths



camss-610

#### 8.3.3.3.4.4 ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 3

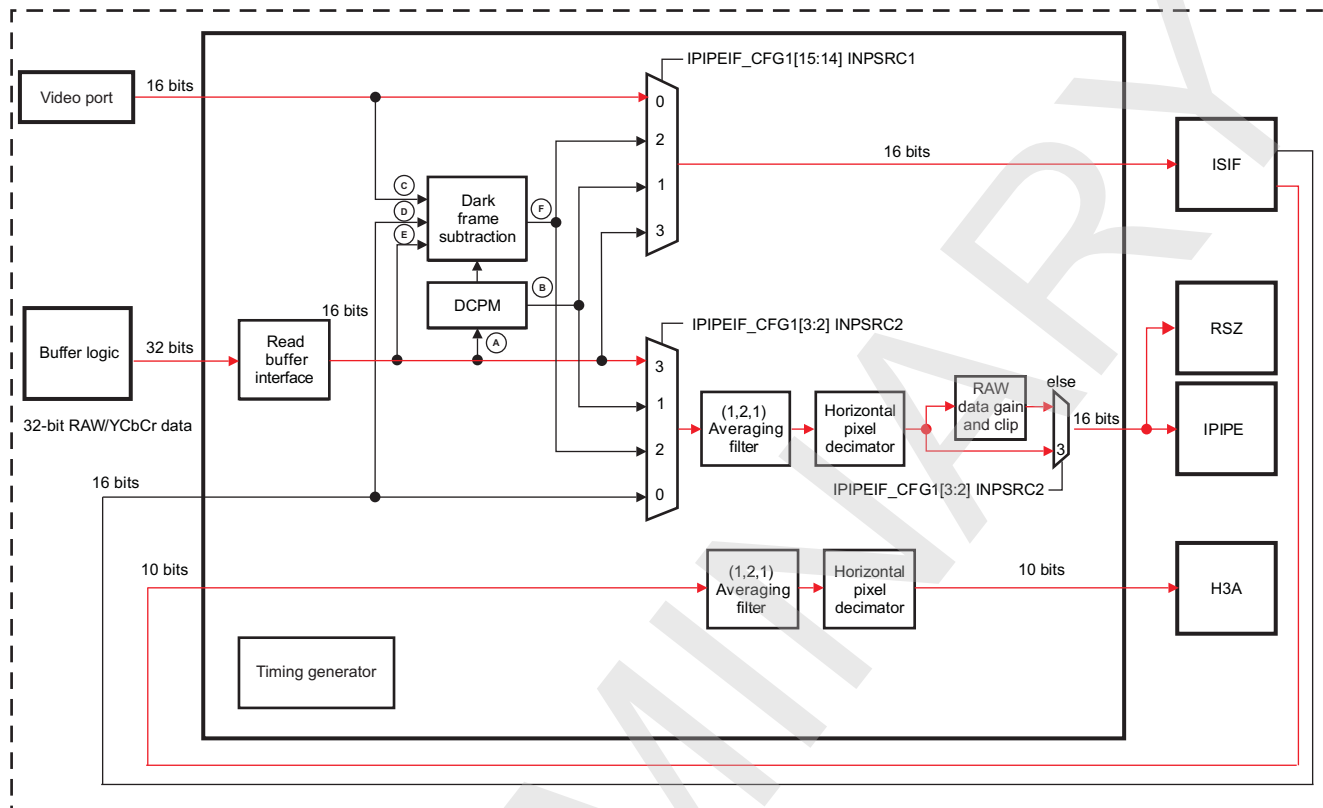
Set the `IPIPEIF_CFG1[15:14] INPSRC1` bit field to 0 and the `IPIPEIF_CFG1[3:2] INPSRC2` bit field to 3.

In the first case, on-the-fly data acquisition is done with the VP, forwarded to the ISIF, and then sent to the H3A through the IPIPEIF while data from memory is processed and forwarded to the IPIPE module and then stored in memory.

In the second case, the configuration can be used to process YUV4:2:2 or RAW data with the IPIPE and RESIZER modules from memory to memory. The YUV4:2:2 or RAW data stored in the SDRAM is fetched and forwarded to the IPIPE and RSZ modules. ISIF and H3A are assumed to be disabled in this configuration.

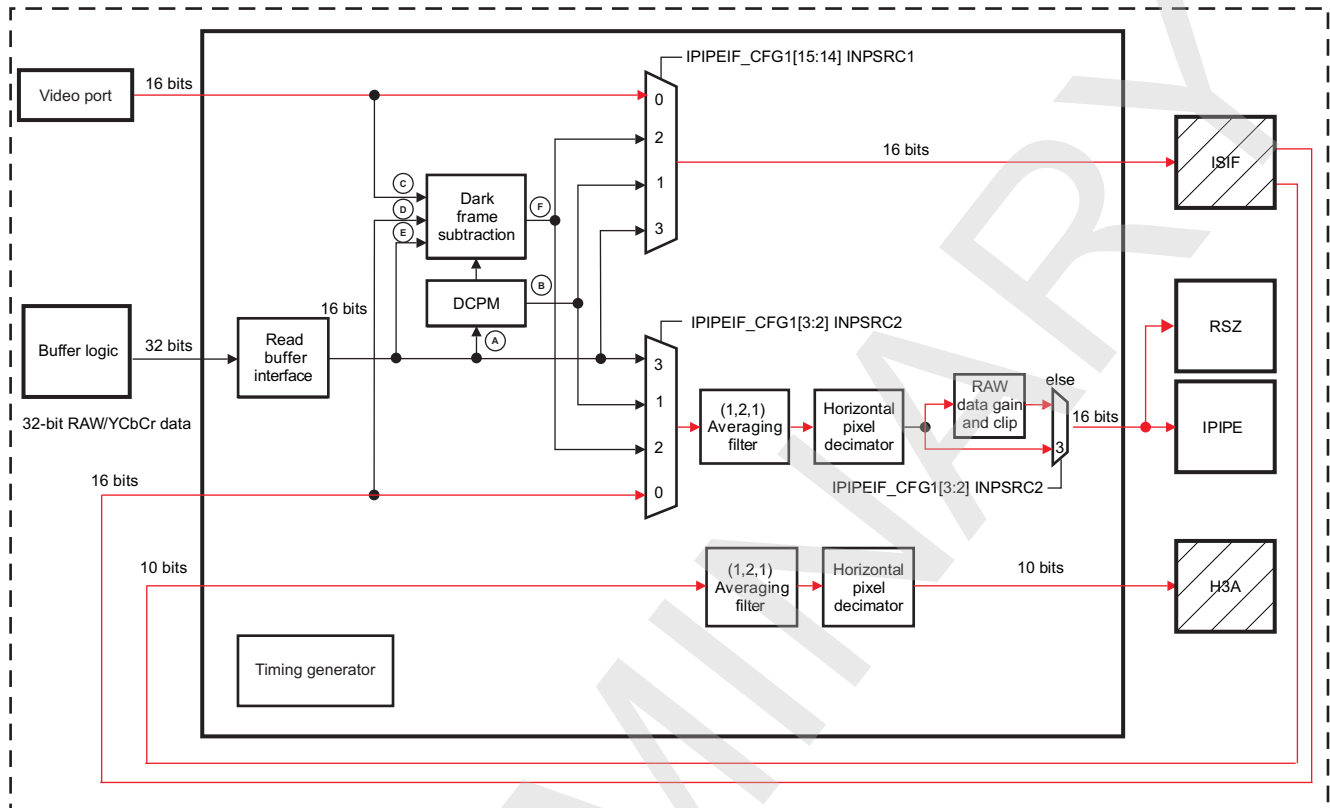
Figure 8-115 and Figure 8-116 show the two possible data paths.

Figure 8-115. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 3 Data Paths: First Case



camss-606

Figure 8-116. ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 3 Data Paths: Second Case



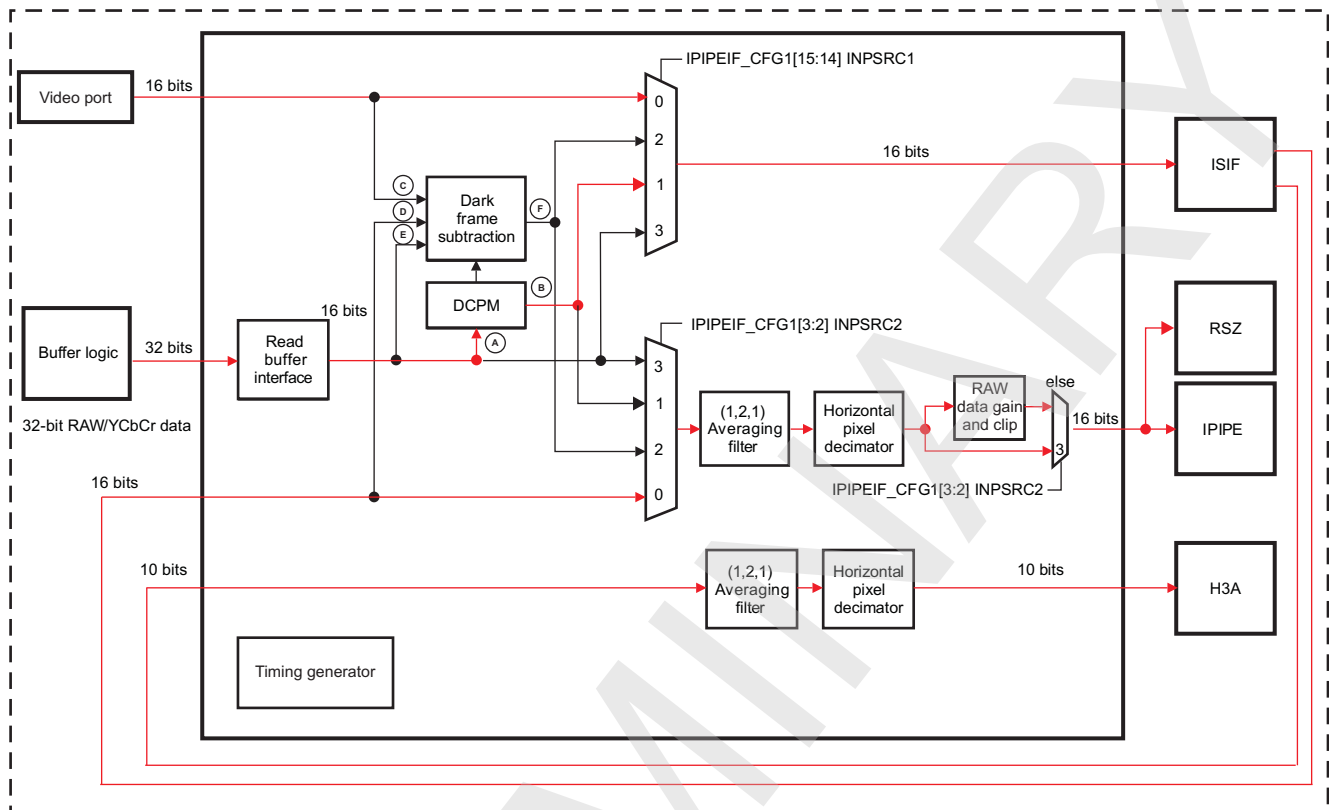
camss-605

8.3.3.3.4.5 ISS ISP IPIPEIF INPSRC1= 1 and INPSRC2 = 0

Set the `IPIPEIF_CFG1[15:14] INPSRC1` bit field to 1 and the `IPIPEIF_CFG1[3:2] INPSRC2` bit field to 0.

This configuration is a memory-to-memory operation. RAW data is read by the BL interface, decompressed, and pushed to the ISIF. The ISIF processes the data and sends it back to the IPIPEIF module before the data is pushed to the IPIPE and H3A modules.

Figure 8-117 shows the data path.

**Figure 8-117. ISS ISP IPIPEIF INPSRC1= 1 and INPSRC2 = 0 Data Path**


camss-607

#### 8.3.3.3.4.6 ISS ISP IPIPEIF INPSRC1= 2 and INPSRC2 = 0

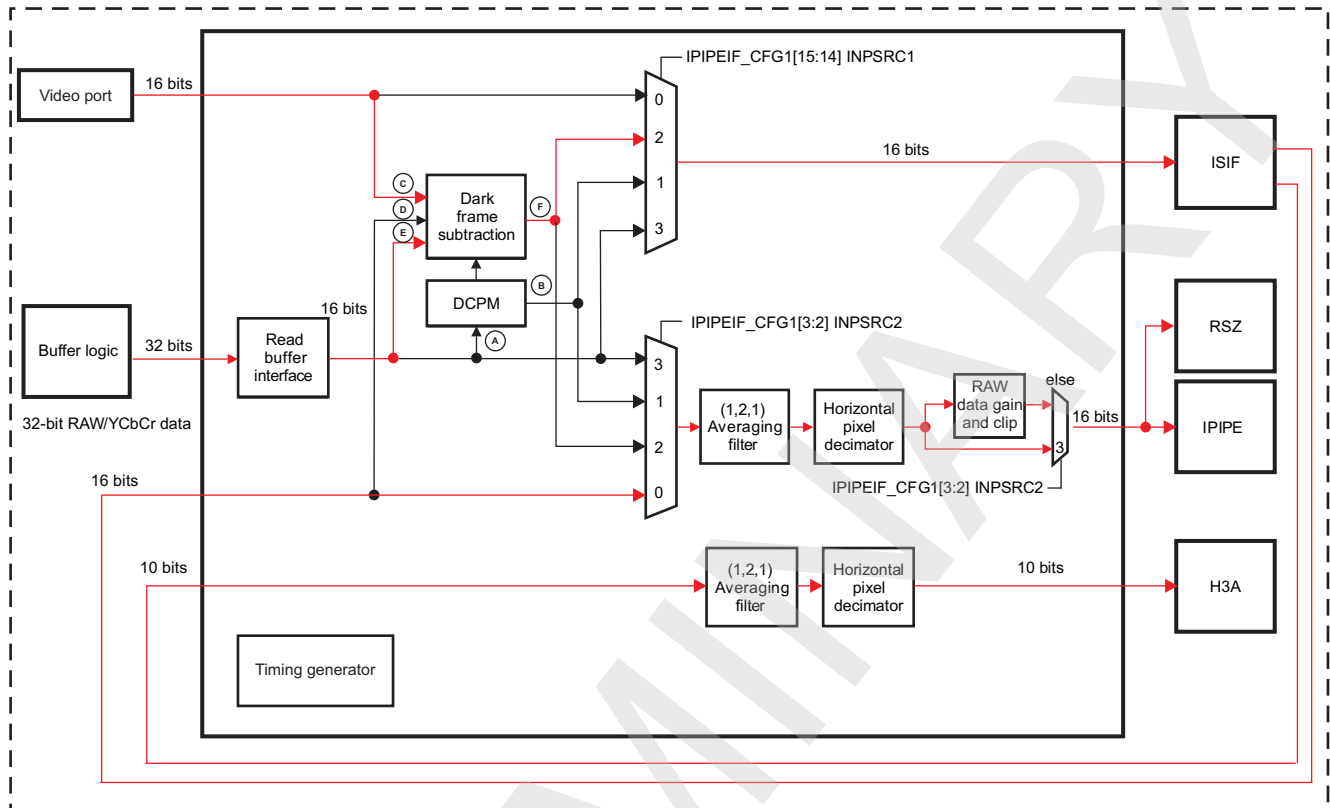
Set the `IPIPEIF_CFG1[15:14] INPSRC1` bit field to 2 and the `IPIPEIF_CFG1[3:2] INPSRC2` bit field to 0.

In this configuration, dark frame subtraction is performed and data is sent to the ISIF module. The ISIF processes the data and sends it back to the IPIPEIF module before the data is pushed to the IPIPE or RSZ modules. There are two possible dark frame subtractions: the first is with data coming from the VP and the dark frame coming from BL; the second is with data coming from BL and the dark frame coming from the VP.

Figure 8-118 shows the data path.



Figure 8-118. ISS ISP IPIPEIF INPSRC1= 2 and INPSRC2 = 0 Data Path



camss-608

8.3.3.3.4.7 ISS ISP IPIPEIF INPSRC1= 3 and INPSRC2 = 0

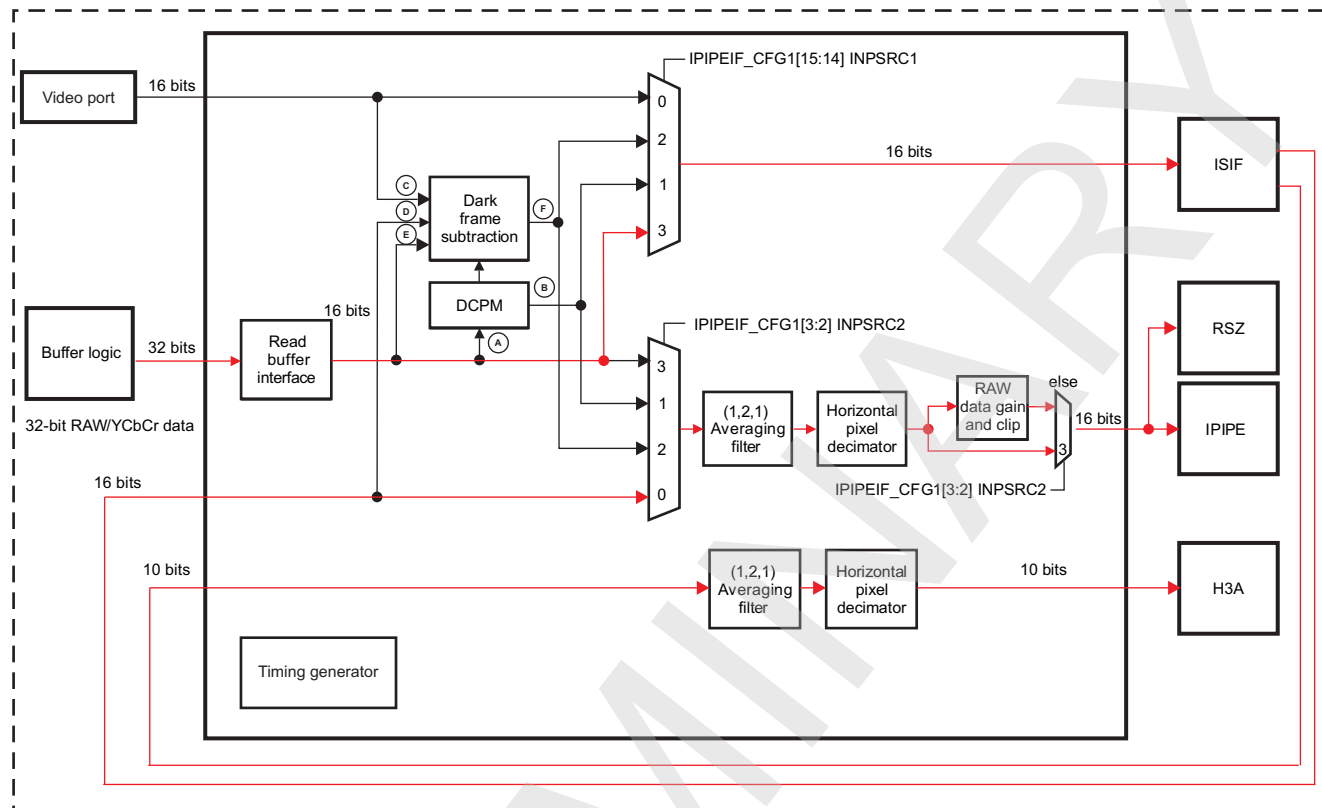
Set the `IPIPEIF_CFG1[15:14] INPSRC1` bit field to 3 and the `IPIPEIF_CFG1[3:2] INPSRC2` bit field to 0.

This configuration is a memory-to-memory operation. Data is loaded from the SDRAM. Input data is expected as 16 bpp. The ISIF processes the data and sends it back to the IPIPEIF module before the data is pushed to the IPIPE or RSZ module.

In this configuration data is assumed to be YUV only, and H3A and RAW data gain are assumed to be disabled.

Figure 8-119 shows the data path.

Figure 8-119. ISS ISP IPIPEIF INPSRC1= 3 and INPSRC2 = 0 Data Path

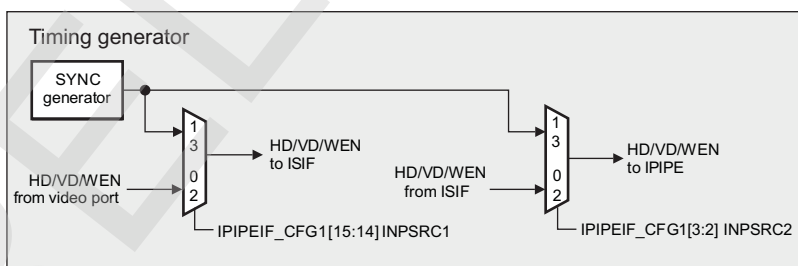


camss-609

### 8.3.3.3.5 ISS ISP IPIPEIF Timing Generation

Figure 8-120 shows the ISS ISP IPIPEIF timing generator submodule.

Figure 8-120. ISS ISP IPIPEIF Timing Generator Submodule Description



camss-602

When the IPIPEIF module input source is from the VP (`IPIPEIF_CFG1[15:14] INPSRC1 = 0 or 2`) or the ISIF (`IPIPEIF_CFG1[3:2] INPSRC2 = 0 or 2`), the `IPIPEIF_CFG1[10] CLKSEL` bit must be set to 0 so that data is latched using the PCLK, HD, and VD signals from the VP.

When the IPIPEIF module input source is not from the VP (`IPIPEIF_CFG1[15:14] INPSRC1 = 1 or 3`), the `IPIPEIF_CFG1[10] CLKSEL` bit must be set to 1 so that the IPIPEIF module generates its proper PCLK, HD, and VD signals (through the use of the SYNC generator). The `IPIPEIF_CLKDIV` register is then used to select a divide ratio of the SDRAM (DMA) clock for the pixel clock frequency, which is used to clock the data into the PCLK. See [Section 8.3.3.3.5.1, ISS ISP IPIPEIF Fractional Clock Divider](#).

When the `IPIPEIF_CFG1[15:14] INPSRC1` or `IPIPEIF_CFG1[3:2] INPSRC2` bit field is not set to 0, the IPIPEIF SDRAM data reading and timing generation can be enabled (`IPIPEIF_ENABLE[0] ENABLE`) in one-shot mode or continuous mode (`IPIPEIF_CFG1[0] ONESHOT`).

**8.3.3.3.5.1 ISS ISP IPIPEIF Fractional Clock Divider**

When the input data of the IPIPEIF module does not come from the VP but from memory, it is useful to have control of the rate at which the data is fetched from memory to avoid overflow conditions or to avoid peak bandwidth requirements. The IPIPEIF\_CFG1[10] CLKSEL bit is equal to 1 for fractional divider use.

The ISP clock ISS\_FCLK is divided to generate the pixel clock, which goes to the ISIF and IPIPE modules when data is read from memory (IPIPEIF\_CFG1[15:14] INPSRC1 = IPIPEIF\_CFG1[3:2] INPSRC2 = 1 or 3). The IPIPEIF\_CLKDIV register selects the divider ratio: M and N values in the IPIPEIF\_CLKDIV[15:0] CLKDIV bit field.

Given an input clock of clock rate ISS\_FCLK, the fractional clock divider generates an output clock with average clock rate f\_out.

Where f\_out = ISS\_FCLK x M/N, and M = 1 through 256, and N = 1 through 256.

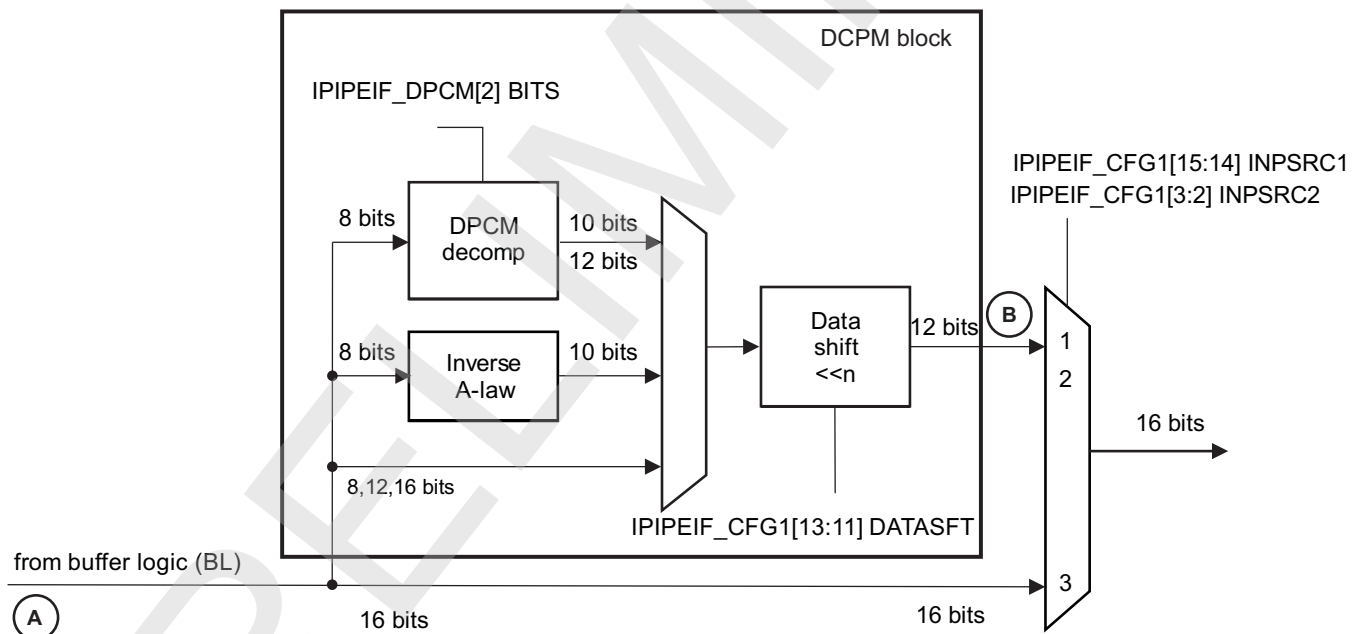
The fractional clock divider logic is synchronous and uses only the positive clock edge of the input clock.

**8.3.3.3.6 ISS ISP IPIPEIF Decompression (DCPM) Subblock: Unpack and Decompression Function**

The IPIPEIF module can read RAW data from memory. The RAW data can be previously packed/compressed into memory. Unpack, A-Law decompression, and DPCM decompression are available in the IPIPEIF module.

Figure 8-121 shows the DCPM subblock.

**Figure 8-121. ISS ISP IPIPEIF DCPM Subblock**



camss-032

Table 8-311 provides the possible configuration of the DCPM block.

**Table 8-311. ISS ISP IPIPEIF DCPM Block Possible Configuration**

Number of Bits per Pixel at DCPM Block Input	Number of Bits per Pixel at DCPM Block Output	Description	Registers
16	16	It can correspond to YUV4:2:2 or RAW16 data.	IPIPEIF_CFG1[9:8] UNPACK = 0x0 In this configuration, IPIPEIF_CFG1[3:2] INPSRC2 = 0x3. In this configuration data bypasses the DCPM block.

**Table 8-311. ISS ISP IPIPEIF DCPM Block Possible Configuration (continued)**

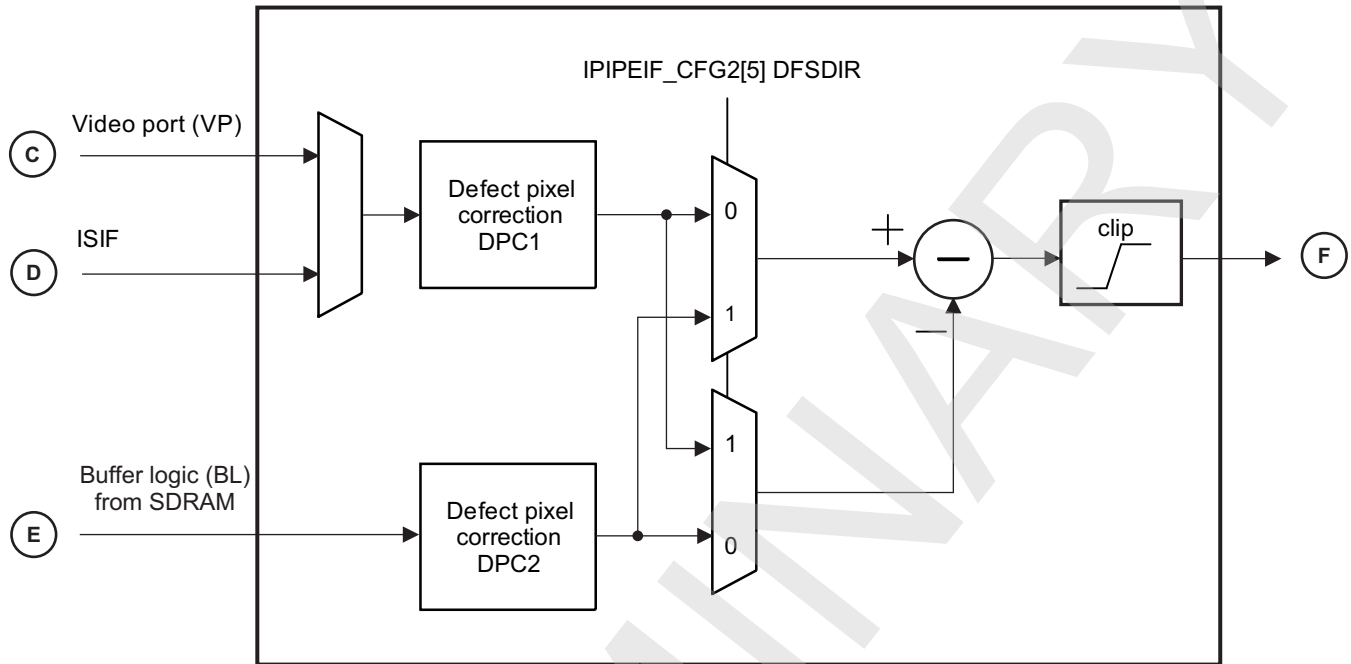
Number of Bits per Pixel at DCPM Block Input	Number of Bits per Pixel at DCPM Block Output	Description	Registers
8	8	PACK8: YUV	<a href="#">IPIPEIF_CFG1[9:8]</a> UNPACK = 0x1 <a href="#">IPIPEIF_DPCM[0]</a> ENA = 0x0 <a href="#">IPIPEIF_CFG1[13:11]</a> DATASFT = 0x01. In this configuration an 8-bit packed RAW data is used in DFS mode or YUV4:2:0 pass through. (See section 1.3.3.13)
8	10	DPCM10: The 8-bit input data are DPCM-decompressed in 10 bits.	<a href="#">IPIPEIF_CFG1[9:8]</a> UNPACK = 0x1 <a href="#">IPIPEIF_DPCM[0]</a> ENA = 0x1 <a href="#">IPIPEIF_DPCM[2]</a> BITS= 0x0 Set the <a href="#">IPIPEIF_DPCM[1]</a> PRED <sup>(1)</sup> <sup>(2)</sup> <a href="#">IPIPEIF_CFG1[13:11]</a> DATASFT = 0x2
8	12	DPCM12: The 8-bit input data are DPCM-decompressed in 12 bits.	<a href="#">IPIPEIF_CFG1[9:8]</a> UNPACK = 0x1 <a href="#">IPIPEIF_DPCM[0]</a> ENA = 0x1 <a href="#">IPIPEIF_DPCM[2]</a> = 0x1 Set the <a href="#">IPIPEIF_DPCM[1]</a> PRED <sup>(1)</sup> <sup>(2)</sup> <a href="#">IPIPEIF_CFG1[13:11]</a> DATASFT = 0x0
8	10	ALAW10: The 8-bit input data were previously A-law compressed. The 8 bits are A-law decompressed and padded with 0.	<a href="#">IPIPEIF_CFG1[9:8]</a> UNPACK = 0x2 <a href="#">IPIPEIF_CFG1[13:11]</a> DATASFT = 0x2
12	12	PACK12: The input is coded on 12 bits and is packed in the SDRAM. This mode is compatible with the ISIF module packing capability. If inverse A-law and DPCM decomposition are not enabled, the data read from the SDRAM can be shifted by the <a href="#">IPIPEIF_CFG1[13:11]</a> DATASFT bit field to select which 12-bits to use. MSB input data bit must be shifted such that it corresponds to bit 11 after the shift.	<a href="#">IPIPEIF_CFG1[9:8]</a> UNPACK = 0x3 Set the <a href="#">IPIPEIF_CFG1[13:11]</a> DATASFT

<sup>(1)</sup> The simple predictor uses only the value of the previous same color component as a prediction value. Therefore, only 2-pixel memory is required. It is typically used for 10–8–10 or 12–8–12 bit conversions.

<sup>(2)</sup> The advanced predictor uses four previous pixel values, when the prediction value is evaluated. This means that the values of the other color components are also used, when the prediction value is defined. Therefore, the advanced predictor is slightly better than the simple predictor but consumes more power and memory. It can, however, improve image quality. It is typically used for 10–7–10 and 10–6–10 bit conversions.

8.3.3.3.7 ISS ISP IPIPEIF Dark-Frame Subtraction Functionality

Figure 8-122. ISS ISP IPIPEIF Dark-Frame Subtraction Subblock



camss-033

The dark-frame subtract function is used to remove noise from the sensor. Typically, the ISIF module previously writes a dark frame (frame captured when the shutter is closed) to SDRAM using 8-bits of linear data packed into 2 pixels per 16 bits.

In this mode, RAW data from the ISIF and SDRAM is used. Data can also be read from SDRAM with the [IPIPEIF\\_CFG1\[9:8\] UNPACK](#) bit field set to 1. Each pixel read from SDRAM is subtracted from each pixel sent from the VP or ISIF.

The mux at the input of the dark-frame subtraction subblock is implicitly controlled by the selection of the [IPIPEIF\\_CFG1\[15:14\] INPSRC1](#) and [IPIPEIF\\_CFG1\[3:2\] INPSRC2](#) bit fields:

- When [IPIPEIF\\_CFG1\[15:14\] INPSRC1](#) = 0x2, the selected input ports are VP and BL. Therefore, the dark frame operation can be:
  - Dark frame = VP – BL
  - Dark frame = BL – VP
- When [IPIPEIF\\_CFG1\[3:2\] INPSRC2](#) = 0x2, the selected input ports are ISIF and BL. Therefore, the dark frame operation can be:
  - Dark frame = ISIF – BL
  - Dark frame = BL – ISIF

The output of the dark frame subtract operation is 12-bits wide (U12Q0). There must be adequate SDRAM bandwidth if this feature is enabled. If the data fetched from memory arrives late, there is an underflow bit ([IPIPEIF\\_DTUF](#)) that must be triggered to know it .

8.3.3.3.7.1 ISS ISP IPIPEIF Defect Pixel Correction

**NOTE:** For the defect pixel correction (DPC) memory access locations, see [Section 8.3.3.9](#).

A simple DPC can be applied to the ISIF or VP input data path and SDRAM input data path, respectively. This DPC algorithm is intended to correct hot pixels during RAW dark frame acquisition or dark frame readout from SDRAM before dark frame subtraction.

The following code describes DPC algorithm:

```

If (image(n) < TH) || ((image(2) > TH)&& (image(n+2) > TH))
    image(n) = image(n)
Else if image(2) > TH
    image(n) = image(n+2)
Else if image(n+2) > TH
    image(n) = image(2)
Else
    image(n) = (image(2) + image(n+2)) / 2
// Where TH is equal to IPIPEIF_DPC1[11:0] TH for DPC1
// Where TH is equal to IPIPEIF_DPC2[11:0] TH for DPC2

```

The `IPIPEIF_DPC1[12]` ENA bit enables DPC for the VP/ISIF input path, and the `IPIPEIF_DPC2[12]` ENA bit enables DPC for the SDRAM input path. The algorithm requires a threshold value that is set by the `IPIPEIF_DPC1[11:0] TH` or `IPIPEIF_DPC2[11:0] TH` bit field that is a 12-bit unsigned value.

### 8.3.3.3.7.2 ISS ISP IPIPEIF DFS Subtraction Direction

The `IPIPEIF_CFG2[5] DFSDIR` bit selects how the DFS subtraction is performed.

- Set the `IPIPEIF_CFG2[5] DFSDIR` bit to 0 when the RAW data is coming from the VP/ISIF and the dark frame is stored in SDRAM.
- Set the `IPIPEIF_CFG2[5] DFSDIR` bit to 1 when the RAW data is coming from SDRAM and the dark frame is coming from the VP/ISIF.

After subtraction, a clip ensures that the value is not negative.

Table 8-312 lists the different modes supported in DFS.

**Table 8-312. ISS ISP IPIPEIF DFS Modes Supported**

Description	DFDIR Value
Dark frame subtract of 8-bit RAW image stored in SDRAM from image from VP	<code>IPIPEIF_CFG2[5] DFSDIR = 0x0</code>
Dark frame subtract of 8-bit RAW image stored in SDRAM from image from ISIF	<code>IPIPEIF_CFG2[5] DFSDIR = 0x0</code>
Dark frame subtract of 8-bit RAW image from VP from image read from SDRAM through the BL	<code>IPIPEIF_CFG2[5] DFSDIR = 0x1</code>

**NOTE:** DFS input depends on the `INPSRC1` and `INPSRC2` settings.

### 8.3.3.3.8 ISS ISP IPIPEIF (1, 2, 1) Averaging Filter for IPIPE Data Path

The averaging filter acts as an anti-aliasing low-pass filter for the horizontal pixel decimator. Usually, when horizontal pixel decimation is enabled (`IPIPEIF_CFG1[1] DECIM`), the averaging filter must also be enabled to avoid aliasing artifacts. The averaging filter can be enabled by setting the `IPIPEIF_CFG1[7] AVGFILT` bit. It operates on every other pixel (same color) in RAW Bayer input or every Y component in YCbCr data in the following equation:

$$\text{output} = (\text{input}[i - 1] + 2 * \text{input}[i] + \text{input}[i + 1]) \gg 2$$

The averaging filter operates on every other pixel (same color) in RAW Bayer input or YUV data.

If the data is YUV4:2:2, the option to average and decimate is given under the conditions listed in Table 8-313.

**Table 8-313. ISS ISP IPIPEIF Averaging Filter Conditions for YUV4:2:2 Data**

<code>IPIPEIF_CFG1[3:2] INPSRC2</code>	<code>IPIPEIF_CFG2[3] YUV16</code>	Comments
0	1	YUV4:2:2 data is coming from the ISIF module. Averager and decimation is possible on the data path to the IPIPE module.
1	1	YUV4:2:2 data is read from SDRAM. Averager and decimation is possible on the data path to the IPIPE module.

**Table 8-313. ISS ISP IPIPEIF Averaging Filter Conditions for YUV4:2:2 Data (continued)**

PIPEIF_CFG1[3:2] INPSRC2	PIPEIF_CFG2[3] YUV16	Comments
3		YUV4:2:2 data is read from SDRAM. Averager and decimation is possible on the data path to the IPIPE module.
Other value	Other value	For YUV4:2:2 data, averager and decimation is not possible on the data path to the IPIPE module.

The averager implements a (1, 2, 1) FIR filter on Luma and Chroma. The following registers have a part in the behavior of the YUV data averaging and decimation:

- PIPEIF\_INIRSZ[12:0] INIRSZ
- PIPEIF\_CFG1[7] AVGFILT
- PIPEIF\_CFG1[1] DECIM

**8.3.3.3.9 ISS ISP IPIPEIF Horizontal Pixel Decimator (Downsizer) for IPIPE Data Path**

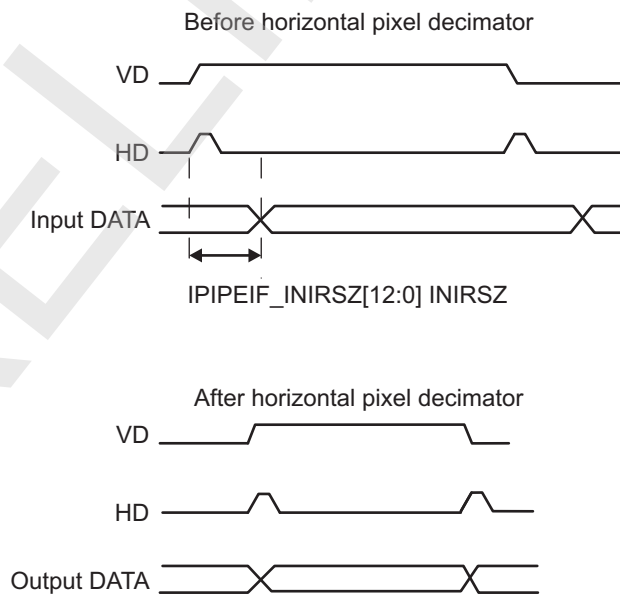
The IPIPE input is limited to 5376 pixels per horizontal line due to line memory width restrictions in the IPIPE.

To process image sensor resolutions with more than 5376 pixels per line with no resolution loss, vertical frame division mode (FDM) must be used; that is, the image must be divided into vertical chunks of less than 5376 pixels, and each chunk must be processed by the ISP sequentially. FDM is memory-to-memory processing and is not supported on the fly.

Alternatively, if a loss in resolution is acceptable, the line width decimator (the PIPEIF\_CFG1[1] DECIM bit) can be enabled to downsample the input lines to a width equal to or less than the 5376 pixel maximum. The resize ratio (16/RSZ) can be configured by programming the PIPEIF\_RSZ[6:0] RSZ bit field to be within the range from 16 to 112 to give a resampling range from 1x to 1/7x.

When ALNSYNC is enabled (PIPEIF\_INIRSZ[13] ALNSYNC = 0x1), the PIPEIF\_INIRSZ[12:0] INIRSZ pixels are skipped (from the HD position) before the horizontal pixel decimator, as shown in Figure 8-123.

**Figure 8-123. ISS ISP IPIPEIF Resizer Offset Definition**



camss-614

### 8.3.3.3.10 ISS ISP IPIPEIF RAW Data Gain for IPIPE Data Path

A gain factor ranging from 0.00195(1/512) to 1.99805(1023/512) is multiplied to the RAW output of the IPIPEIF. The gain is not applied if the input data is YCbCr. The gain constant is set in the IPIPEIF\_GAIN[9:0] GAIN bit field using U10Q9 format.

The output value is clipped after gain control through the value of the IPIPEIF\_OCLIP[11:0] OCLIP bit field.

### 8.3.3.3.11 ISS ISP IPIPEIF (1, 2, 1) Averaging Filter for H3A Data Path

The averaging filter acts as an anti-aliasing low-pass filter for the horizontal pixel decimator. Usually, when horizontal pixel decimation is enabled (the IPIPEIF\_RSZ3A[9] DECIM bit), the averaging filter must also be enabled to avoid aliasing artifacts. The averaging filter can also be used to reduce noise before H3A statistics generation. It operates on every other pixel (the same color) in a RAW Bayer input or every Y component in YCbCr data. The averaging filter can be enabled by setting the IPIPEIF\_RSZ3A[8] AVGFILT bit, and it operates with the following equation:

$$\text{output} = (\text{input}[i - 1] + 2 * \text{input}[i] + \text{input}[i + 1]) \gg 2$$

### 8.3.3.3.12 ISS ISP IPIPEIF Horizontal Pixel Decimator (Downsizer) for H3A Data Path

The H3A input is limited to 3008 pixels per horizontal line due to line memory width restrictions in the H3A.

To process image-sensor resolutions with more than 3008 pixels per line with no resolution loss, vertical frame division mode (FDM must be used); that is, the image must be divided into vertical chunks of less than 3008 pixels, and each chunk must be processed by the ISP sequentially. FDM is memory-to-memory processing and is not supported on the fly.

Alternatively, if loss of resolution is acceptable, the line width decimator (the IPIPEIF\_RSZ3A[9] DECIM bit) can be enabled to downsample the input lines to a width equal to or less than the 3008 pixel maximum. The resize ratio (16/RSZ) can be configured by programming the IPIPEIF\_RSZ3A[6:0] RSZ bit field to be within the range from 16 to 112 to give a resampling range from 1x to 1/7x.

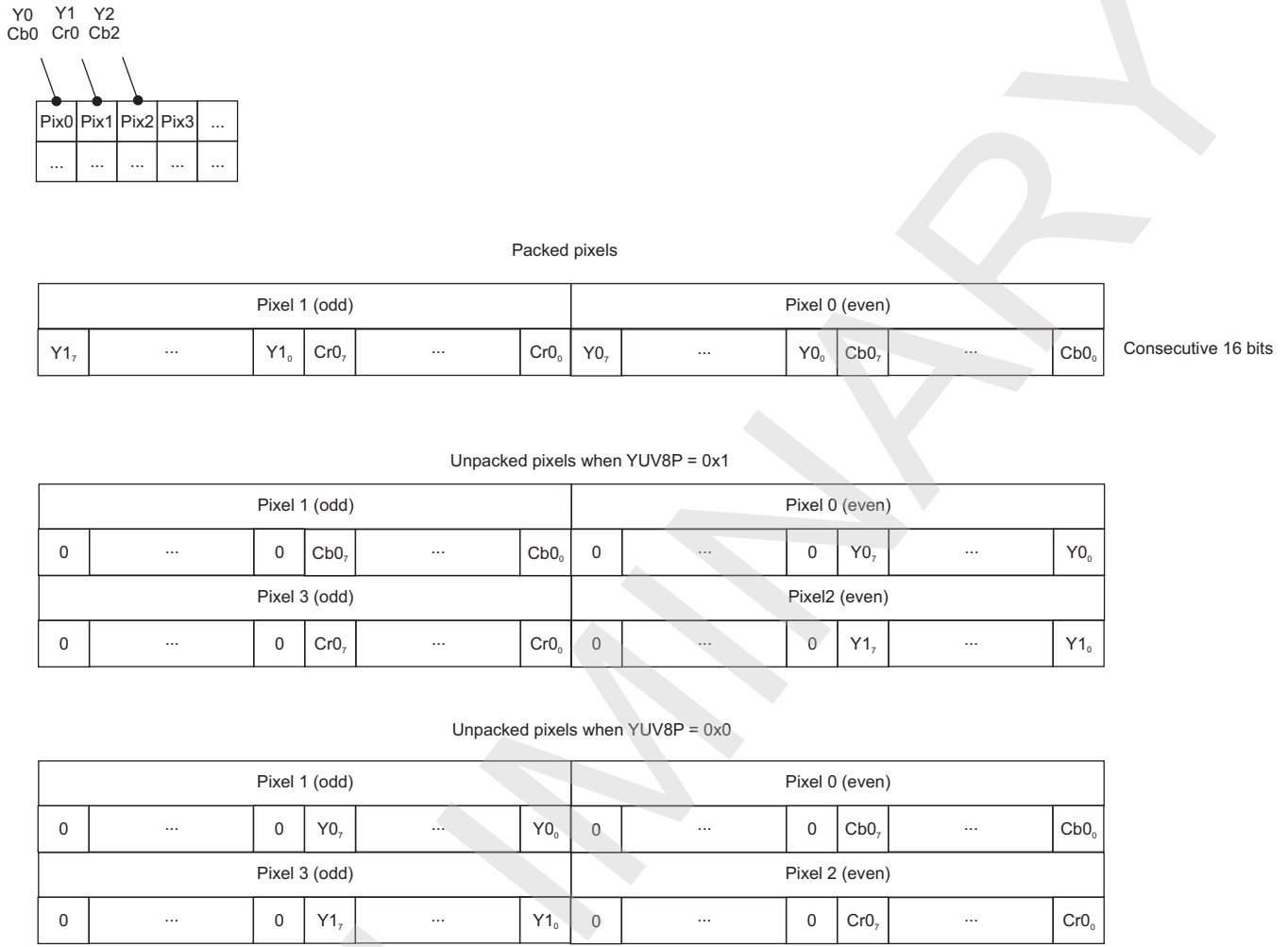
When ALNSYNC is enabled (IPIPEIF\_INIRSZ3A[13] ALNSYNC = 0x1), the IPIPEIF\_INIRSZ3A[12:0] INIRSZ pixels are skipped (from the HD position) before the horizontal pixel decimator (see Figure 8-123).

### 8.3.3.3.13 ISS ISP IPIPEIF YUV4:2:2 8-bits packed Data Input Coming from ISIF Module

This section applies when data coming from the ISIF is 8 bits (IPIPEIF\_CFG2[6] YUV8 = 0x1). When IPIPEIF\_CFG1[3:2] INPSRC2 = 0 and IPIPEIF\_CFG2[3] YUV16 = 1, the 8-bit YUV data are transformed into 16-bit YUV data. The way the data are unpacked from 8 bits to 16 bits is controlled by the IPIPEIF\_CFG2[7] YUV8P bit. See Figure 8-124.



Figure 8-124. ISS ISP IPIPEIF YUV8P Settings



camss-616

8.3.3.3.14 ISS ISP IPIPEIF YUV4:2:0 Data Input for Memory-to-Memory Resize Operations

The ISP RSZ module can resize YUV4:2:0 data. The YUV4:2:0 data can come from the memory (through BL) or from the VP through the CCP2 RX module, which reads data from SDRAM and pushes it to the IPIPEIF module VP. The possible data paths are:

- SDRAM -> IPIPEIF (BL) -> RESIZER -> SDRAM
- SDRAM -> CCP2 RX -> IPIPEIF (VP) -> RESIZER -> SDRAM
- SDRAM -> IPIPEIF (BL) -> IPIPE -> RESIZER -> SDRAM
- SDRAM -> CCP2 RX -> IPIPEIF (BL) -> IPIPE -> RESIZER -> SDRAM

When the data comes from BL, the IPIPEIF module must be set up to process the luminance data first, and then the chrominance data.

- For 420Y (first pass):
  - IPIPEIF\_CFG1[15:14] INPSRC1 = 0x1 (SDRAM data input)
  - IPIPEIF\_CFG1[3:2] INPSRC2 = 0x1 (SDRAM data input)
  - IPIPEIF\_CFG1[13:11] DATASFT = 0x0 (no data shift)
  - IPIPEIF\_CFG1[9:8] UNPACK = 0x1 (data packed on 8 bits)
  - IPIPEIF\_CFG2[3] YUV16 = 0x1 (data\_input [7:0] = 0 and data\_input [15:8] = valid)
- For 420C (second pass):

- IPIPEIF\_CFG1[15:14] INPSRC1 = 0x1 (SDRAM data input)
- IPIPEIF\_CFG1[3:2] INPSRC2 = 0x1 (SDRAM data input)
- IPIPEIF\_CFG1[13:11] DATASFT = 0x0 (no data shift)
- IPIPEIF\_CFG1[9:8] UNPACK = 0x1 (data packed on 8 bits)
- IPIPEIF\_CFG2[3] YUV16 = 0x0 (data\_input[7:0] = valid and data\_input [15:8] = 0)

### 8.3.3.3.15 ISS ISP IPIPEIF Module Events and Status Checking

The IPIPEIF module generates an IPIPEIF event through the IPIPEIF\_IRQ interrupt at the end of each frame. This interrupt is set through the [ISP5\\_IRQENABLE\\_SET\\_i\[9\]](#) IPIPEIF\_IRQ bit. The input interrupt source generation is selected through the [IPIPEIF\\_CFG2\[0\]](#) INTSW bit in a certain configuration. The following pseudo code describes INTSW.

```

if (IPIPEIF_CFG2[0] INTSW==0) // Interrupt source from VP
  if (IPIPEIF_CFG1[15:14] INPSRC1==1,2 or 3)
    if (CFG1.ONESHOT==1) // In one shot mode
      Interrupt happens at the end of frame
    else // In continuous mode
      Interrupt is the start position of VD which is generated by IPIPEIF timing generator
  else // IPIPEIF_CFG1[15:14] INPSRC1==0, data is from VP
    Interrupt is the start position of VD from VP
else // Interrupt source from ISIF: IPIPEIF_CFG2[0] INTSW==1
  if (IPIPEIF_CFG1[3:2] INPSRC2==1,2 or 3)
    if (CFG1.ONESHOT==1) // In one shot mode
      Interrupt happens at the end of frame
    else
      Interrupt is the start position of VD which is generated by IPIPEIF timing generator
  else // IPIPEIF_CFG1[3:2] INPSRC2==0, data is from ISIF
    Interrupt is the start position of VD from ISIF

```

Moreover to this interrupt, the host must check the IPIPEIF\_DTUF status flag of the [ISP5\\_IRQSTATUS\\_RAW2\\_i\[1\]](#) IPIPEIF\_UDF bit (if this is enabled and mapped to the ISP IRQ lines) to see if an underflow occurred. For more information, see [Section 8.3.4.3.4](#).

## 8.3.3.4 ISS ISP IPIPE Functional Description

### 8.3.3.4.1 ISS ISP IPIPE Overview

- The input interface extracts valid region from the Bayer RAW data:
  - Up to 12-bit input pixel resolution
  - Requires at least 8 pixels for horizontal blanking and four lines for vertical blanking. In one-shot mode; 16 blanking lines after processing area are required.
  - Maximum horizontal and vertical offset of IPIPE processing area from synchronous signal is 65534.
  - Support RGB Bayer pattern for input
- DPC module fixes defect pixels using two methods: LUT-based and on-the-fly adaptive.
- 2D noise-filter module reduces noise in RAW data.
- Green-imbalance-correction (GIC) module reduces Gb/Gr difference to remove line crawl noise.
- White balance module applies offset and gain adjustments to each color.
- Color filter array (CFA) interpolation module implements CFA interpolation. The output from the CFA interpolation module is RGB-4:4:4 formatted data. CFA also reduces aliasing caused by undersampling by digital anti-aliasing (DAA)
- RGB2RGB blending module applies a 3 × 3 matrix transform to the RGB data generated by the CFA interpolation module.
- Gamma correction module independently applies gamma correction to each RGB component. Gamma is implemented using a piece-wise linear interpolation approach with a 512-entry LUT for each color.
- 2nd RGB2RGB blending module applies a 3 × 3 matrix transform to the RGB data after gamma correction.
- 3D-LUT converts RGB data to RGB data using 9 × 9 × 9 table and tetrahedral interpolation.
- RGB2YCbCr conversion module applies 3 × 3 matrix transformation to the RGB data to convert it to YCbCr data. This module also implements offset. Global brightness and contrast enhancement module

fixes brightness and contrast tone.

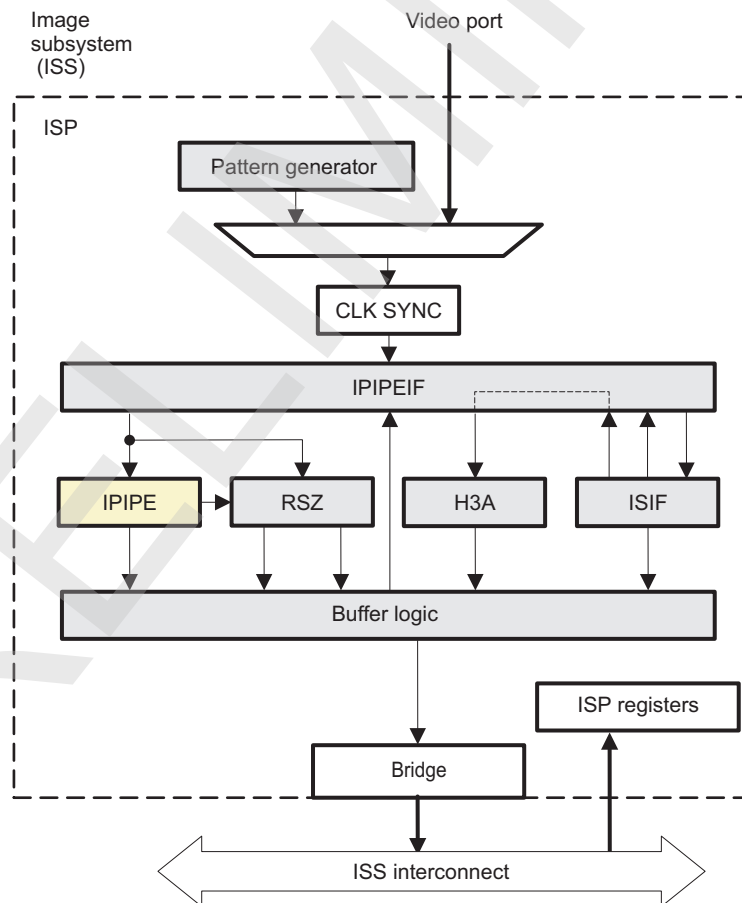
- 4:2:2 conversion module applies the chroma low pass filter and downsampling to Cb and Cr to convert 4:4:4 data to 4:2:2 data.
- 2D edge enhancer module improves image clarity with luminance nonlinear filter.
- Chroma artifact reduction module reduces color artifacts using gain control and 2D median filter.
- Output interface module transfers data from IPIPE to SDRAM, in the form of one YCbCr (4:2:2 or 4:2:0), RGB (32/16 bits), or Bayer data.
- Histogram function can record histograms of up to four distinct areas into up to 256 bins.
- Boxcar function makes 1/8 or 1/16 size (1/64 or 1/256 in area) images.
- Boundary signal calculator (BSC) makes vectors of row and column summations.

IPIPE has four different processing paths:

- Case 1: IPIPE reads CCD RAW data and applies all IPIPE functions and stores the YCbCr (or RGB) data to SDRAM.
- Case 2: IPIPE reads CCD RAW data and stores the Bayer data after white balance to SDRAM.
- Case 3: IPIPE reads YCbCr-422 data and applies edge enhance, chroma suppression, and resize to output YCbCr data to SDRAM.
- Case 4: IPIPE reads YCbCr-420 data and applies resize to output YCbCr data to SDRAM.

Figure 8-125 shows the connections from the IPIPE module to other submodules of the ISP.

**Figure 8-125. ISS ISP IPIPE High-Level Diagram**



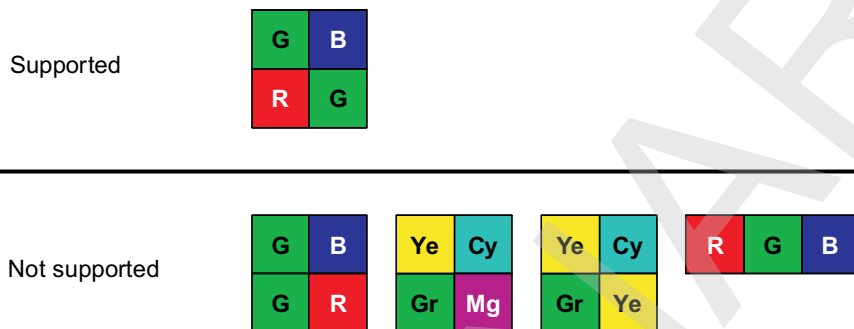
camss-003d

### 8.3.3.4.2 ISS ISP IPIPE Top-Level Block Diagram

The IPIPE is a programmable hardware image-processing module that generates image data in YCbCr-4:2:2 or YCbCr-4:2:0 format from RAW CCD/CMOS data. The IPIPE module supports output of Bayer data.

The IPIPE module supports RAW data in Bayer format, as shown in Figure 8-126. Other RGB formats or complimentary color formats are not supported.

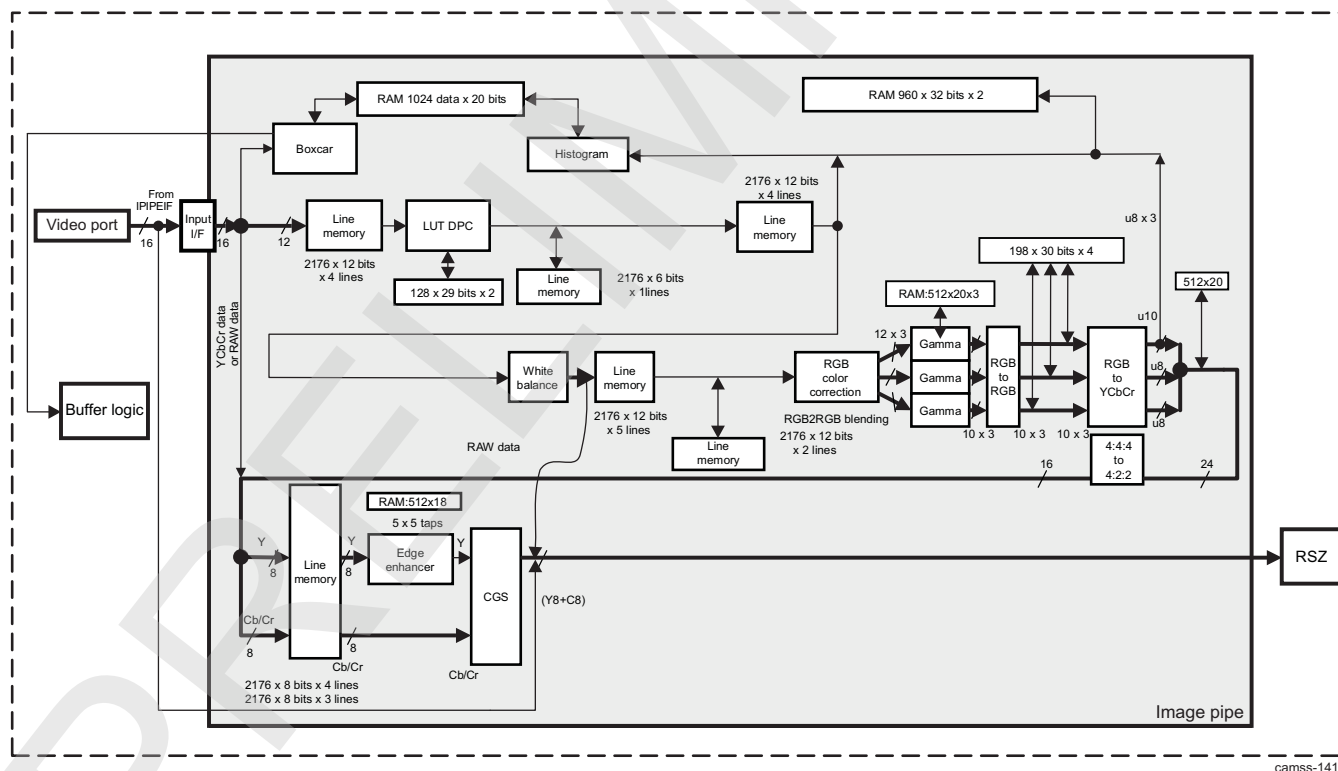
Figure 8-126. ISS ISP IPIPE Supported CFA Format



camss-142

As shown in Figure 8-127, many internal modules are used to process Bayer data into YCbCr data

Figure 8-127. ISS ISP IPIPE Module Block Diagram



camss-141b

### 8.3.3.4.3 ISS ISP IPIPE Input Interface

The IPIPE module receives 12-bit RAW image data or 16-bit YCbCr data through the IPIPEIF module. The IPIPE module can work with up to 5376 pixels in each horizontal line, except in RAW pass-through mode. If the image width is larger than 5376, the image must be scaled down at the IPIPEIF module level. Otherwise, the input image must be split into several blocks.

If the input data is YCbCr, all RGB processing modules are skipped, and only edge enhancer and chroma suppression are applied to the input data.

If the input data is YCbCr-4:2:0, only Y or C can be processed at a time, and only resizer process can be applied. Because the resizer is outside the ISIF module, the data is passed to it directly by skipping the RGB and YCbCr processing modules.

In RAW pass-through mode, images up to 8190 pixels per line can be processed. In RAW pass through mode, the input data is written out directly to SDRAM.

The IPIPE module is enabled through the [IPIPE\\_SRC\\_EN\[0\]](#) EN bit.

The IPIPEIF module must be selected as the IPIPE module source with the [IPIPE\\_SRC\\_MODE\[1\]](#) WRT bit set to 1 from the input port of the IPIPEIF. This is required to enable and transfer data properly from the interface to the IPIPE.

The IPIPE module has two processing modes, which can be selected through the [IPIPE\\_SRC\\_MODE\[0\]](#) OST bit:

- One-shot mode: [IPIPE\\_SRC\\_MODE\[0\]](#) OST = 0x1
- Free-run mode: [IPIPE\\_SRC\\_MODE\[0\]](#) OST = 0x0

The input and output formats are selected in the [IPIPE\\_SRC\\_FMT\[1:0\]](#) FMT bit field (see [Table 8-314](#)).

**Table 8-314. ISS ISP IPIPE Input and Output Selections**

<a href="#">IPIPE_SRC_FMT[1:0]</a> FMT	IPIPE Module Input	IPIPE Module Output
0x0	RAW Bayer	YCbCr or RGB
0x1	RAW Bayer	RAW Bayer
0x2	RAW Bayer	Disabled
0x3	YCbCr 16 bits	YCbCr

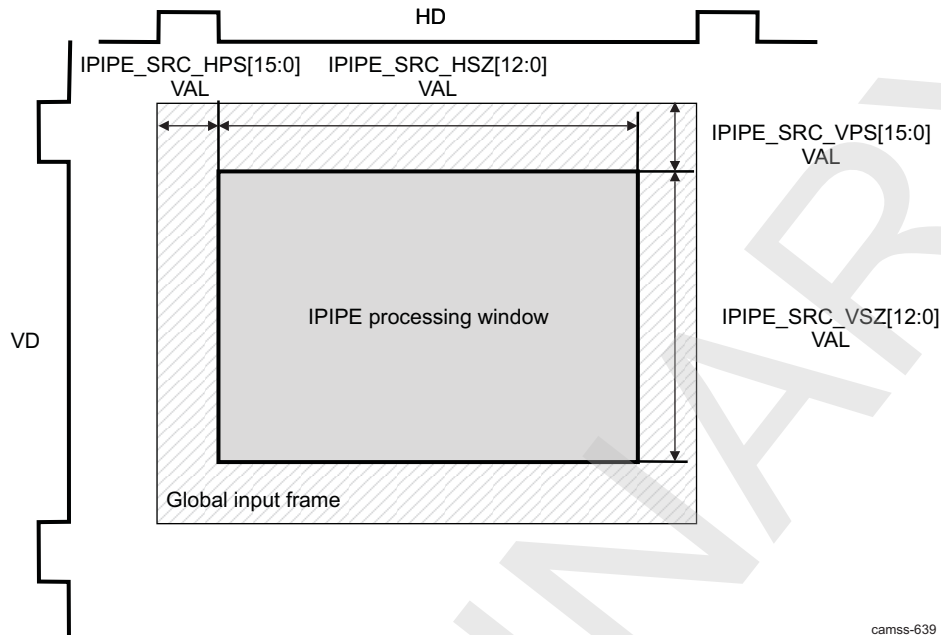
The input to the IPIPE module is in the formats (YCbCr-8bit is not allowed) shown in [Figure 8-128](#).

**Figure 8-128. ISS ISP IPIPE Module Input Format**

IPIPE input	RAW	LOW	LOW	LOW	LOW	RAW11	RAW10	RAW9	RAW8	RAW7	RAW6	RAW5	RAW4	RAW3	RAW2	RAW1	RAW0
	YCbCr 16b	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Cb/Cr7	Cb/Cr6	Cb/Cr5	Cb/Cr4	Cb/Cr3	Cb/Cr2	Cb/Cr1	Cb/Cr0
	Y 8bit	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	LOW	LOW	LOW	LOW	LOW	LOW	LOW	LOW
	CbCr 8bit	LOW	LOW	LOW	LOW	LOW	LOW	LOW	LOW	Cb/Cr7	Cb/Cr6	Cb/Cr5	Cb/Cr4	Cb/Cr3	Cb/Cr2	Cb/Cr1	Cb/Cr0

camss-143

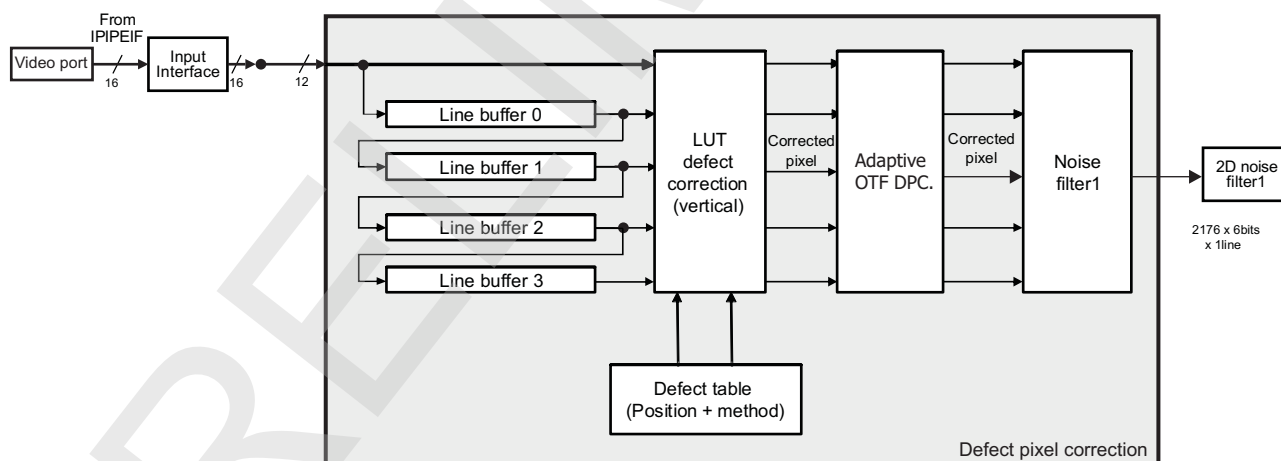
The window to process can be defined by its vertical and horizontal start position ([IPIPE\\_SRC\\_VPS](#) and [IPIPE\\_SRC\\_HPS](#)) and vertical and horizontal size ([IPIPE\\_SRC\\_VSZ](#) and [IPIPE\\_SRC\\_HSZ](#)). [Figure 8-129](#) shows the processing window settings.

**Figure 8-129. ISS ISP IPIPE Module Processing Window Settings**

camss-639

**8.3.3.4.4 ISS ISP IPIPE Defect Pixel Correction**

The DPC module corrects defect pixels using two methods: look-up-table-based method (LUT DPC) and on-the-fly adaptive method (OTF DPC). [Figure 8-130](#) shows defect pixel correction.

**Figure 8-130. ISS ISP IPIPE Defect Pixel Correction**

camss-146b

**8.3.3.4.4.1 ISS ISP IPIPE LUT Defect Pixel Correction (LUT DPC)**

LUT DPC is the first stage of the IPIPE image-processing pipeline. The LUT DPC module corrects defects in input data. It supports up to a 256 defect point table. However, the table may be renewed as required during image processing. Therefore, the maximum amount of defect information is limited only by system level performance.

The module uses two sets of  $128 \times 29$  memories to hold defect information. The table contains the information of horizontal position (13 bits), vertical position (13 bits), and correction method (3 bits), as shown in [Table 8-315](#). The LUT DPC is enabled through the `IPIPE_DPC_LUT_EN[0]` EN bit.

**Table 8-315. ISS ISP IPIPE Defect Information Packing**

Correction Method	Vertical Position	Horizontal Position
28..26	25..13	12..0

The information must be listed in the order of "from left to right" and "from the top to the bottom". The first position in the defect information table and the number of defects that are used may be specified. The address of the table must be programmed in the `IPIPE_DPC_LUT_ADR[9:0]` ADR bit field. Thus, the address of the first valid data is stated.

The LUT type can be:

- With a finite number of entries:
  - `IPIPE_DPC_LUT_SEL[1]` TBL = 0x0
  - The size of the LUT is set in the `IPIPE_DPC_LUT_SIZ[9:0]` SIZ bit field.
- With an infinite number of entries:
  - `IPIPE_DPC_LUT_SEL[1]` TBL = 0x1

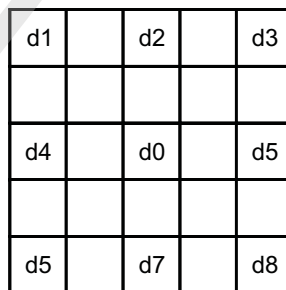
The correction methods, set in [Table 8-315](#), are described in [Table 8-316](#).

**Table 8-316. ISS ISP IPIPE Correction Method Description**

Correction Method	d0 =	Comment
0	Black or white dot	Replace with a black (or white) dot to force OTF-DPC to work on the pixel. White or black dot replacement can be selected through the <code>IPIPE_DPC_LUT_SEL[0]</code> DOT field.
1	d4	Copy from left
2	d5	Copy from right
3	$(d4 + d5)/2$	Horizontal interpolation
4	$(d2 + d7)/2$	Vertical interpolation
5	d2	Copy from top
6	d7	Copy from bottom
7	$(d2 + d4 + d5 + d7)/2$	2D interpolation

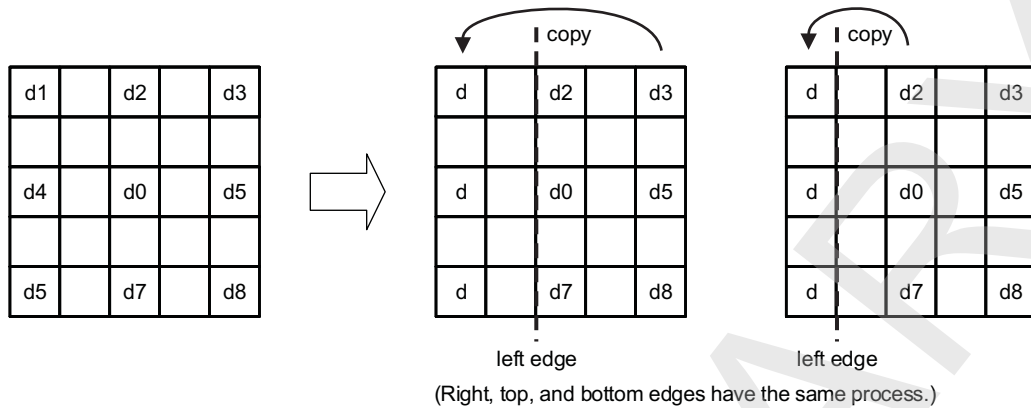
The pixels in the defect correction algorithm are numbered as shown in [Figure 8-131](#).

**Figure 8-131. ISS ISP IPIPE Pixel Numbering in Defect Correction Algorithm**



camss-145

The pixels at the edges are mirrored in the way described in [Figure 8-132](#). The figure shows the typical correction by overwriting far edge pixels and mirroring them with other edge pixels. The example shows how by using a noise filter the correct-by-definition pixels are copied over bad pixels.

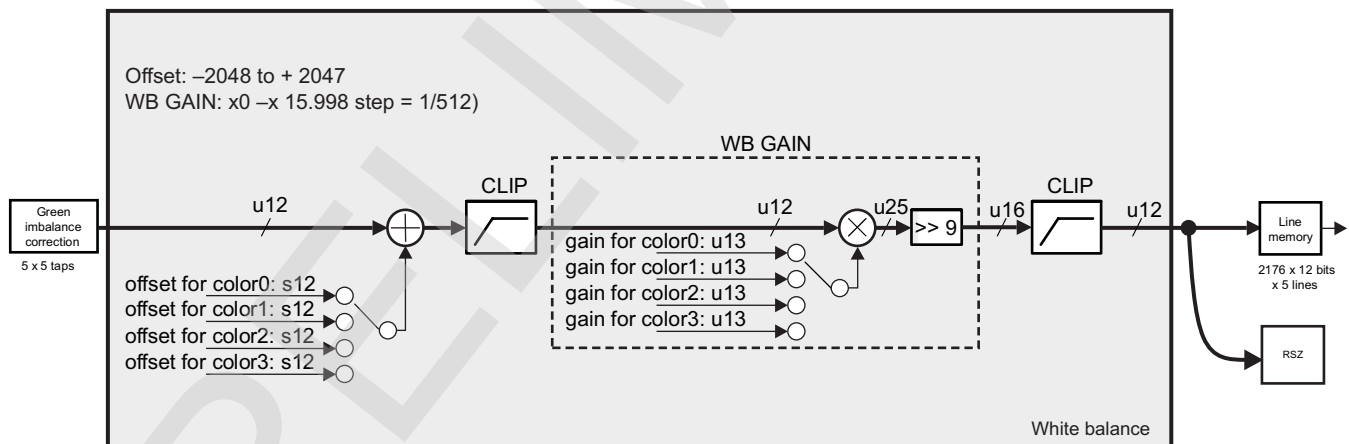
**Figure 8-132. ISS ISP IPIPE Mirroring in Defect Correction and Noise Filter**


camss-147

### 8.3.3.4.5 ISS ISP IPIPE White Balance

The white balance module executes white balance to each color component. White balance gain adjusts the ratio of each color existing in a CFA pattern. An offset can be applied before white balance correction (IPIPE\_WB2\_OFT\_R, GR, Gb, or B registers).

Figure 8-133 is a block diagram of the white balance module. In the white balance gain adjuster, the RAW data is multiplied by a selected gain (IPIPE\_WB2\_WGN\_R, Gr, Gb, or B registers) corresponding to the color. The white balance gain can be selected from four 13-bit values. Firmware can assign any combination of 4 pixels in horizontal and vertical directions. The precision of each gain is shown in the figure.

**Figure 8-133. ISS ISP IPIPE White Balance**


camss-155

### 8.3.3.4.6 ISS ISP IPIPE RGB2RGB Blending Module

The RGB2RGB blending module transforms the RGB data generated by the CFA interpolation module using a  $3 \times 3$  square matrix transformation in combination with an added offset. The RGB to RGB blending is calculated using the formula shown in Figure 8-134. Each gain range is from -8 to +7.996 with step  $1/256 = 0.004$ , and is set in the IPIPE\_RGB1\_MUL\_RR to IPIPE\_RGB1\_MUL\_BB registers. The offset range for each component is from -4096 to 4095, and is set in the IPIPE\_RGB1\_OFT\_OR to IPIPE\_RGB1\_OFT\_OB registers.



**Figure 8-134. ISS ISP IPIPE RGB2RGB Conversion Formula**

$$\begin{pmatrix} R\_out \\ G\_out \\ B\_out \end{pmatrix} = \begin{pmatrix} gain\_RR & gain\_GR & gain\_BR \\ gain\_RG & gain\_GG & gain\_BG \\ gain\_RB & gain\_GB & gain\_BB \end{pmatrix} \begin{pmatrix} R\_in \\ G\_in \\ B\_in \end{pmatrix} + \begin{pmatrix} offset\_r \\ offset\_G \\ offset\_B \end{pmatrix}$$

camss-650

**8.3.3.4.7 ISS ISP IPIPE Gamma Correction Module**

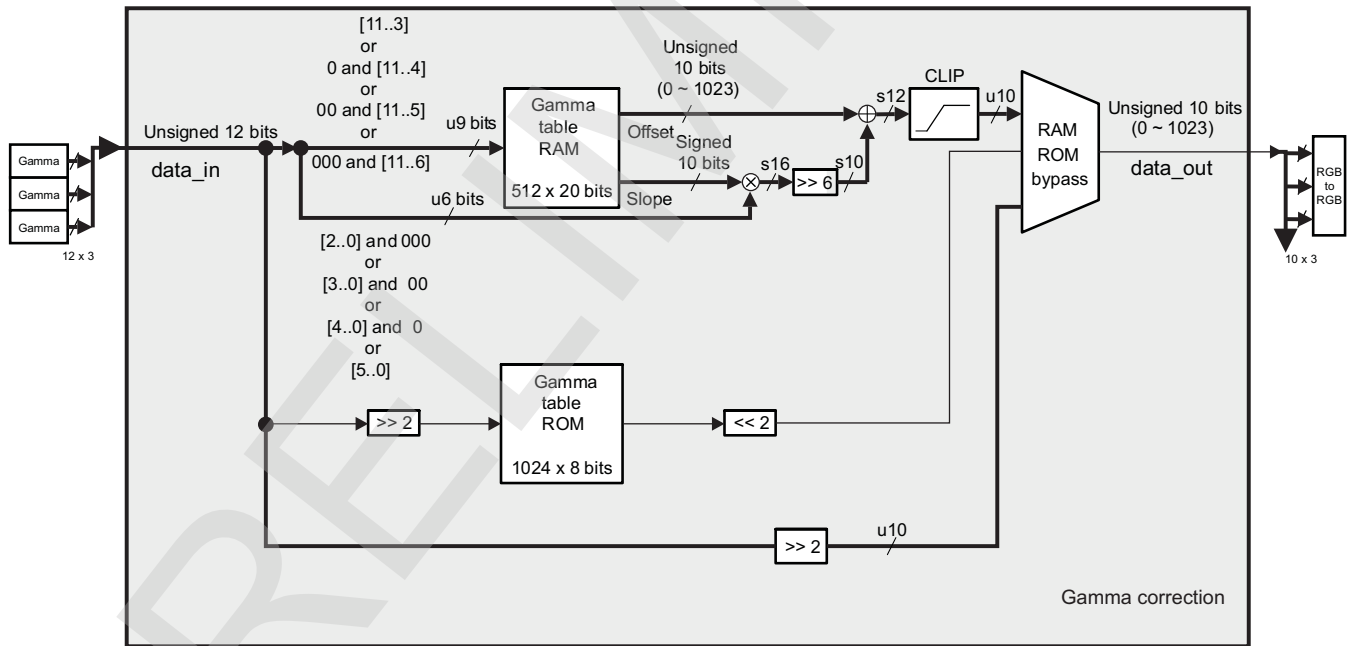
**NOTE:** For the memory access locations of the gamma correction module, see [Section 8.3.3.9](#).

The gamma correction module performs a gamma correction independently for each color in the RGB color space by using a piece-wise linear interpolation. ROM tables and RAM tables are selectable through the `IPIPE_GMM_CFG[4]` TBL bit. Each ROM table and RAM table has 512-entries, and each entry accommodates a 10-bit offset and 10-bit slope. The range of slope value is from -512 to +511. The ROM table has 1024 entries and output 8-bit value.

Figure 8-135 is a block diagram of the gamma correction module. It is composed of two tables and one selector. When the BYPASS bit is asserted, the input data is divided by 16 (the `IPIPE_GMM_CFG[0]` BYPR, `IPIPE_GMM_CFG[1]` BYPG, and `IPIPE_GMM_CFG[2]` BYPB bits).

Figure 8-136 shows an example of the gamma curve. Figure 8-137 shows offset and slope packing.

**Figure 8-135. ISS ISP IPIPE Gamma Correction Module Block Diagram**



camss-157

Figure 8-136. ISS ISP IPIPE Gamma Curve Example

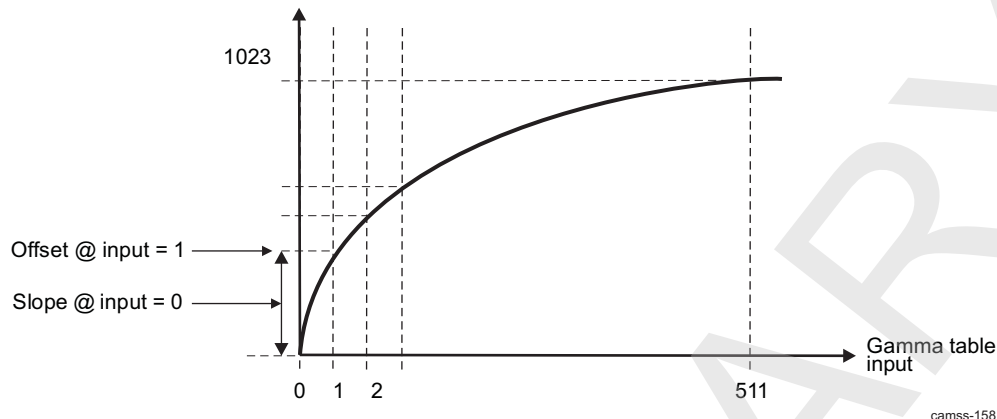
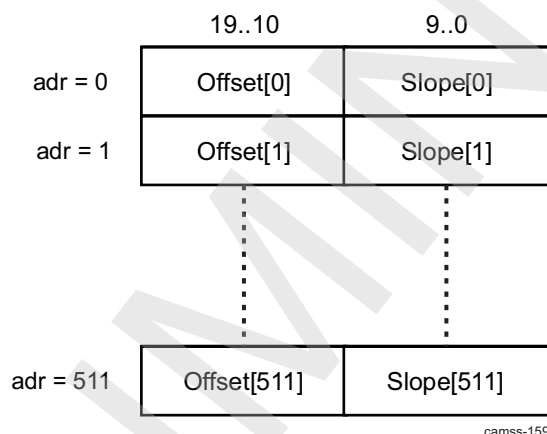


Figure 8-137. ISS ISP IPIPE Gamma Table Offset/Slope Packing



#### 8.3.3.4.8 ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix

The second RGB2RGB blending module transforms the RGB data after gamma correction using the  $3 \times 3$  square matrix transformation in combination with an added offset. The RGB-to-RGB blending is calculated using the formula shown in Figure 8-138. Each gain range is from  $-4$  to  $+3.996$  with step  $1/256 = 0.004$  (s3.8), and is set in the `IPIPE_RGB2_MUL_RR` to `IPIPE_RGB2_MUL_BB` registers. The offset is  $-1024$  to  $1023$  (s11), and is set in the `IPIPE_RGB2_OFT_OR` to `IPIPE_RGB2_OFT_OB` registers.

Figure 8-138. ISS ISP IPIPE RGB2RGB 2nd Conversion Formula

$$\begin{pmatrix} R\_out \\ G\_out \\ B\_out \end{pmatrix} = \begin{pmatrix} gain\_RR & gain\_GR & gain\_BR \\ gain\_RG & gain\_GG & gain\_BG \\ gain\_RB & gain\_GB & gain\_BB \end{pmatrix} \begin{pmatrix} R\_in \\ G\_in \\ B\_in \end{pmatrix} + \begin{pmatrix} offset\_r \\ offset\_G \\ offset\_B \end{pmatrix}$$

camss-650

#### 8.3.3.4.9 ISS ISP IPIPE RGB2YCbCr Conversion Matrix

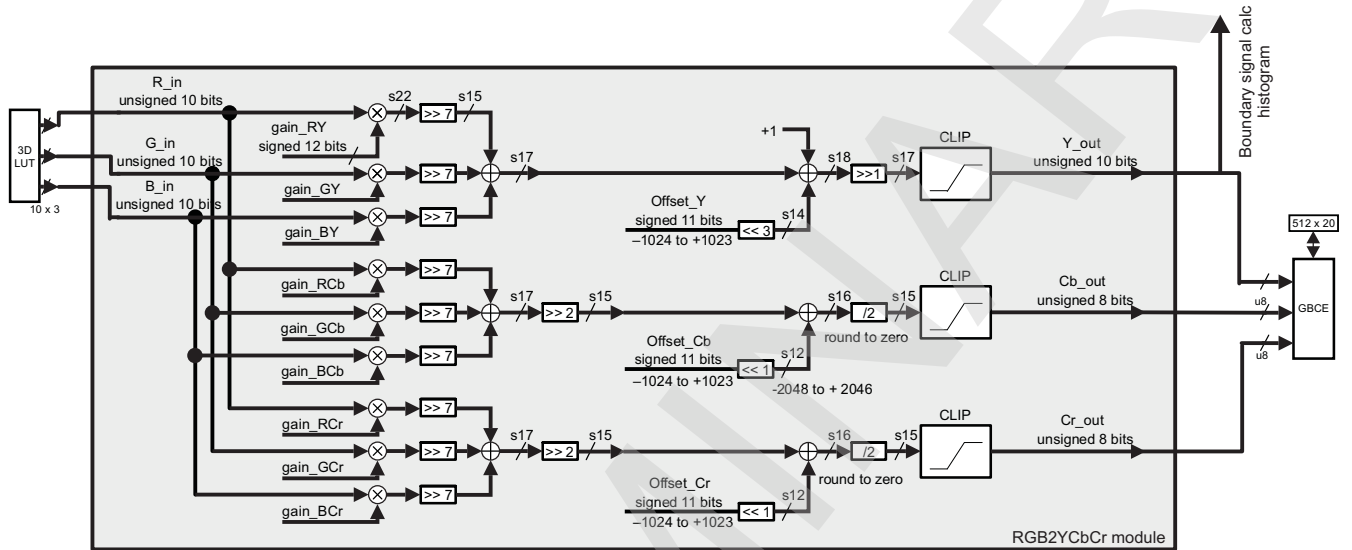
This module transforms the RGB data to YCbCr data format using a  $3 \times 3$  matrix transformation in combination with an added offset. While transferring, the brightness control and contrast control can be adjusted using the `IPIPE_YUV_ADJ[8:15]` BRT and `IPIPE_YUV_ADJ[8:15]` CRT bit fields, respectively. Then, the transform is calculated using the formula shown in Figure 8-139. Each gain range is from  $-8$  to  $+7.996$  with step  $1/256 = 0.004$ , configured in the `IPIPE_YUV_MUL_RY` to `IPIPE_YUV_MUL_BCr` registers. The offset is  $-1024$  to  $1023$  for Y, Cb, and Cr, configured in the `IPIPE_YUV_OFT_Y` to `IPIPE_YUV_OFT_Cr` registers. Figure 8-140 is the block diagram of the RGB to RGB blending module. The output is calculated by the equation.

Figure 8-139. ISS ISP IPIPE RGB2RGB 2nd Conversion Formula

$$\begin{pmatrix} Y\_out \\ Cb\_out \\ Cr\_out \end{pmatrix} = \begin{pmatrix} gain\_RY & gain\_GY & gain\_BY \\ gain\_RCb & gain\_GCb & gain\_BCb \\ gain\_RCr & gain\_GCr & gain\_BCr \end{pmatrix} \begin{pmatrix} R\_in \\ G\_in \\ B\_in \end{pmatrix} + \begin{pmatrix} offset\_Y \\ offset\_Cb \\ offset\_Cr \end{pmatrix}$$

camss-654

Figure 8-140. ISS ISP IPIPE RGB2YCbCr Module Block Diagram

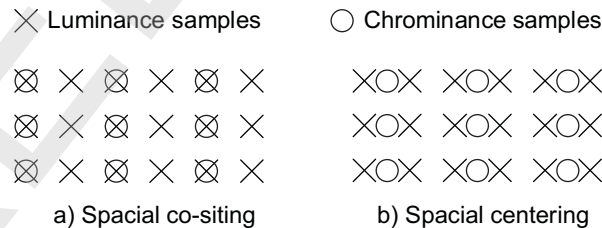


camss-161

8.3.3.4.10 ISS ISP IPIPE 4:2:2 Conversion Module

The 4:2:2 conversion module converts the image data to YCbCr-4:2:2 format by taking the average of every two Cb and Cr components. Y and Cb/Cr sampling point of spatial cosited or spatial centering are selectable using the IPIPE\_YUV\_PHS[0] POS register. Horizontal 3 taps and 4- or 2-tap filters are used for spatial cosited and spatial centering, respectively. The module is enabled from the IPIPE\_YUV\_PHS[1] PLF bit.

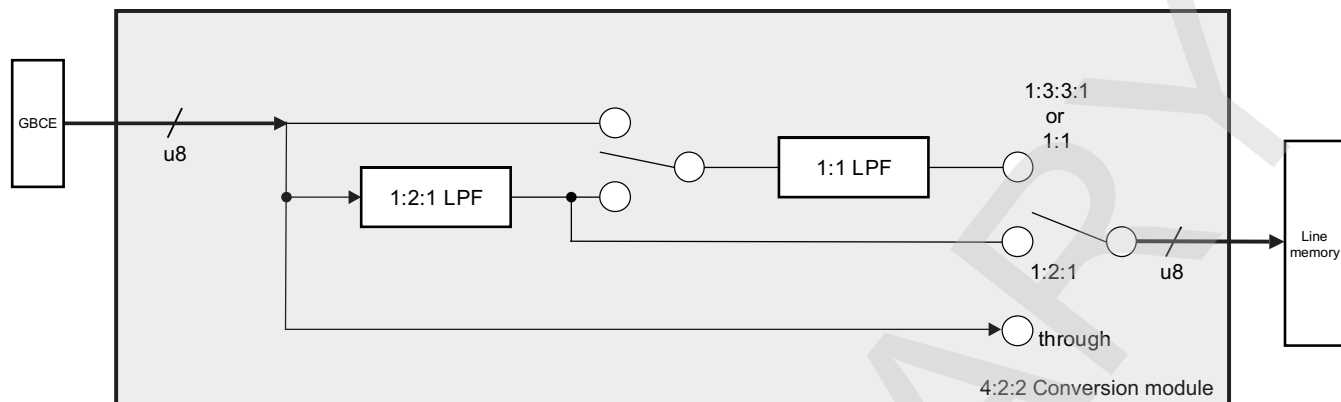
Figure 8-141. ISS ISP IPIPE Chroma Subsampling Position



camss164

Figure 8-142 is a block diagram of the 4:2:2 conversion module.

Figure 8-142. ISS ISP IPIPE 4:2:2 Conversion Module Block Diagram



camss-165

### 8.3.3.4.11 ISS ISP IPIPE 2D Edge Enhancer

**NOTE:** For the location of 2D edge-enhancer memory accesses, see [Section 8.3.3.9](#).

The edge-enhancer module operates on the luminance (Y data) component of images to improve the image quality and can be enabled from the `IPIPE_YEE_EN[0]` EN bit. Edges in input images are detected by a 2D high-pass filter, and its sharpness is increased by the value from a non-linear table. [Figure 8-143](#) shows a block diagram of the luminance non-linear edge-enhancer. Entry for the non-linear table is 10-bit and the output is in signed 9-bit.

In edge-enhancer mode when `IPIPE_YEE_TYP[0]` SEL is set to 0, linear filter with programmable coefficient is applied to the Y input. Here, M is a  $5 \times 5$  matrix with programmable coefficients (`IPIPE_YEE_MUL_xx`, where  $x = 00, 01, 02, 10, 11, 12, 20, 21, 22$ ). A down shift of high pass filter is applied to the edge enhancer from the `IPIPE_YEE_SHF[3:0]` SHF bit field (`shgHPF` in the formula).

Then, the HPF value is shrunk by a threshold value (u6) specified by the `IPIPE_YEE_THR` register (`thresholdHPF` in the formula), and clipped to signed 10 bits to get the index for the LUT.

Figure 8-143. ISS ISP IPIPE 2D Edge-Enhancer Indexing

$$index = clip( shrink( HPF, threshold_{HPF} ), -512, 511 )$$

$$shrink(x, threshold) = \begin{cases} x + threshold & x < -threshold \\ 0 & -threshold \leq x \leq threshold \\ x - threshold & threshold < x \end{cases}$$

$$clip(x, limit_{LOW}, limit_{HIGH}) = \begin{cases} -limit_{LOW} & x < -limit_{LOW} \\ x & -limit_{LOW} \leq x \leq limit_{HIGH} \\ limit_{HIGH} & limit_{HIGH} < x \end{cases}$$

camss-661

Moreover, the edge-enhancement intensity is looked up from the LUT through the formula shown in [Figure 8-144](#) and in [Table 8-317](#).

Figure 8-144. ISS ISP IPIPE 2D Edge Intensity LUT Formula

$$E_{int} = LUT[index]$$

camss-662

**Table 8-317. ISS ISP IPIPE Edge-Enhancer LUT Mapping**

Address (32-bit Word Address)	Bit Position	LUT Index
0x00000h	8..0	1
	17..9	2
0x00001h	8..0	2
	17..9	3
0x00002h	8..0	4
	17..9	5
0x00003h	8..0	6
	17..9	7
⋮	⋮	⋮
0x000FFh	8..0	510
	17..9	511
0x00100h	8..0	-512
	17..0	-511
0x00101h	8..0	-510
	17..9	-509
⋮	⋮	⋮
0x001FD	8..0	-6
	17..9	-5
0x001FE	8..0	-4
	17..9	-3
0x001FFh	8..0	-2
	17..9	-1

Figure 8-145 shows the LUT packing, and Figure 8-146 shows the 2D edge-enhancer block diagram.

**Figure 8-145. ISS ISP IPIPE 2D Edge-Enhancer LUT Packing**

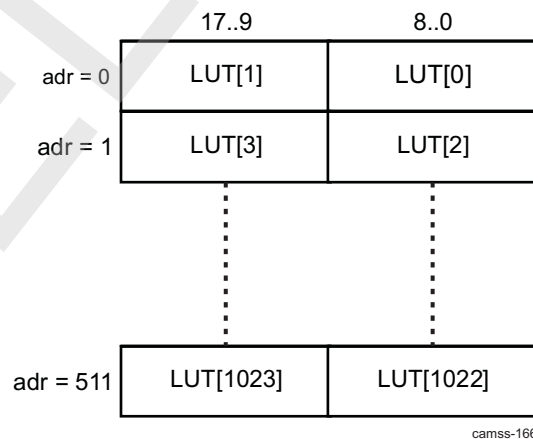
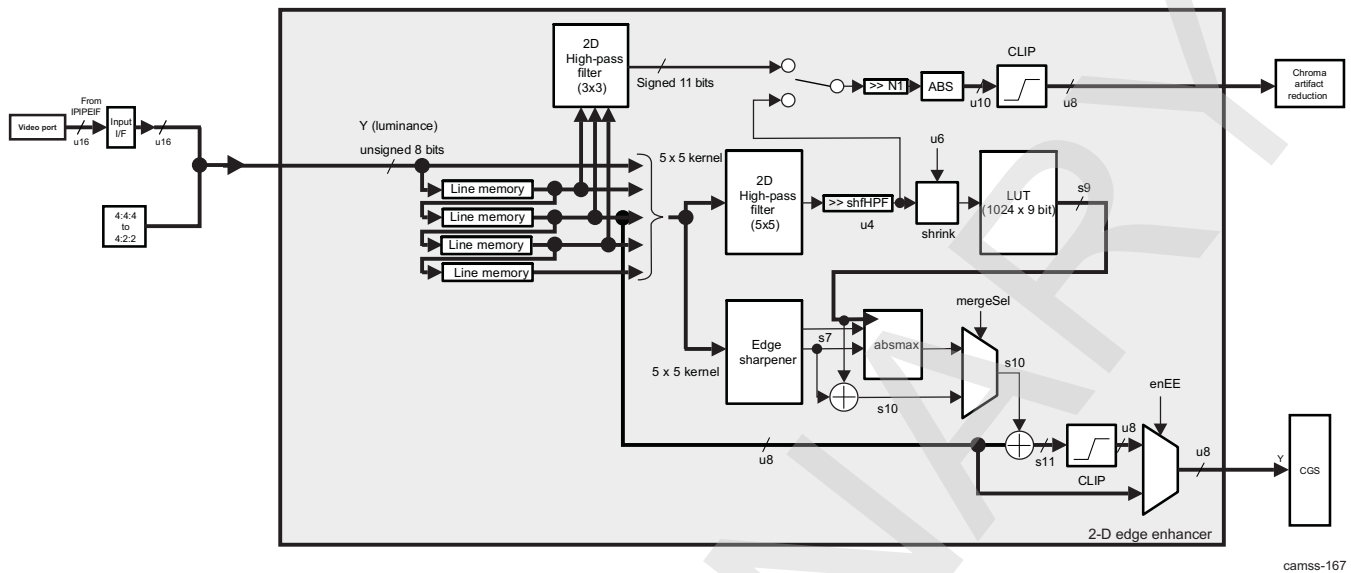


Figure 8-146. ISS ISP IPIPE 2D Edge-Enhancer Block Diagram



In edge sharpener mode, enabled when `IPIPE_YEE_TYP[0] SEL = 1`, edge clarity is enhanced without producing a Halo artifact. In this module, edge intensity is derived by the 2D linear filter with fixed coefficients shown in Figure 8-147.

Figure 8-147. ISS ISP IPIPE Edge Sharpener Details

$$S_{i,j} = \begin{pmatrix} 0 & -1 & -2 & -1 & 0 \\ -1 & 0 & 2 & 0 & -1 \\ -2 & 2 & 8 & 2 & -2 \\ -1 & 0 & 2 & 0 & -1 \\ 0 & -1 & -2 & -1 & 0 \end{pmatrix}$$

$$\text{sharpness}(h,v) = \text{clip} \left( \text{shrink} \left( g \sum_{j=-2}^2 \sum_{i=-2}^2 S_{i,j} Y(h+i,v+j), -\text{threshold}_{\text{LOW}}, \text{threshold}_{\text{LOW}} \right) \gg 6, \text{threshold}_{\text{HIGH}} \right)$$

camss-663

The gain ( $g$ ) and threshold values for the shrink/clip function ( $\text{threshold}_{\text{LOW}}$ ,  $\text{threshold}_{\text{HIGH}}$ ) are determined by the `IPIPE_YEE_E_GAN`, `IPIPE_YEE_THR_1` and `IPIPE_YEE_THR_2` registers. The bit width of  $g$  and  $\text{threshold}_{\text{HIGH}}$  is in U6, and  $\text{threshold}_{\text{LOW}}$  is in U12Q6.

This edge intensity is then clipped by a threshold value in the formula shown in Figure 8-148.

Figure 8-148. ISS ISP IPIPE 2D Edge-Intensity Clipping Formula

$$S_{int} = \begin{cases} \text{clip}(\text{sharpness}, \text{grad}) & \text{Haloreduction on} \\ \text{sharpness} & \text{Haloreduction off} \end{cases}$$

camss-664

The threshold value ( $\text{grad}$ ) is a function of the activity around the target pixel, which is derived from gradient values. Gain and offset are specified by `IPIPE_YEE_G_GAN` and `IPIPE_YEE_G_OFFT`

Capping with gradient value prevents overly enhancing edges, and suppresses halo artifacts around edges.

The output from edge enhancer and edge sharpener are merged with the function shown in Figure 8-149.

**Figure 8-149. ISS ISP IPIPE 2D Edge Enhancer and Sharpener Merger Formula**

$$E_{merge} = \begin{cases} E_{int} + S_{int} & mergedsel = 1 \\ \text{abs max}(E_{int}, S_{int}) & mergedsel = 0 \end{cases}$$

$$\text{abs max}(x, y) = \begin{cases} x & \text{abs}(y) \leq \text{abs}(x) \\ y & \text{otherwise} \end{cases}$$

camss-665

The  $E_{merge}$  value is added to the Y input value to make the final output.

For chroma suppression, another 2D high pass filter (HPF) is implemented. One of the four coefficient sets shown in [Figure 8-150](#) is selectable.

**Figure 8-150. ISS ISP IPIPE 2D Edge Chroma-Suppression Coefficient Sets**

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \text{ or } \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

camss-667

At the end of the edge enhancer process, brightness and contrast adjustment are applied to the Y signal. The formula shown in [Figure 8-151](#) describes the process.

**Figure 8-151. ISS ISP IPIPE 2D Edge-Brightness and Contrast Adjustments Formula**

$$Y_{ctr\_brt} = \text{clip8}(\text{clip8}((Y_{EE} \times \text{CTR}) \gg 4) + \text{BRT})$$

camss-668

### 8.3.3.4.12 ISS ISP IPIPE Histogram

---

**NOTE:** The boxcar function can be used simultaneously with the histogram function if needed.

---

**NOTE:** For the locations of histogram memory access, see [Section 8.3.3.9](#).

---

The histogram module counts the number of pixels that have a value in a region and can be enabled from the `IPIPE_HST_EN[0]` EN bit. Moreover, if enabled, the `IPIPE_HST_MODE` register can be set to work constantly or one time. If the `IPIPE_HST_MODE[0]` OST bit is set to 1, the histogram is disabled by clearing the `IPIPE_HST_EN[0]` EN bit to 0 after one run (one-shot mode).

After enabling the module, the following features are available:

- The data to be summed is taken from the DPC memory or RGB2YCbCr module. The choice is made in the `IPIPE_HST_SEL[2]` SEL bit.
- When data are collected from the DPC memory, the sampled colors are R/G/B/Y. Y is derived in the following method:

$$Y = (\text{HST\_MUL\_R} * R + \text{HST\_MUL\_GR} * Gr + \text{HST\_MUL\_GB} * Gb + \text{HST\_MUL\_B} * B) \tag{4}$$

For the G histogram, Gb, Gr, or the average is used, through the `IPIPE_HST_SEL[1:0]` TYP bit field.

- Two sets of 512 x 20-bit memory are used.
- The number of bins can be set from 32 to 256 in the `IPIPE_HST_PARA[13:12]` BIN bit field.
- The number of regions (areas) from 1 to 4; each region can be enabled through `IPIPE_HST_PARA[x]` RGNx (where x = 0 to 3). The positions of the regions are defined in `IPIPE_HST_x_VPS` and `IPIPE_HST_x_HPS`, and the vertical and horizontal size are defined by `IPIPE_HST_x_VSZ` and `IPIPE_HST_x_HSZ`, respectively (where x = 0 to 3).
- The number of regions x the number of bins <= 256.
- Each region can be turned on/off counting.
- The regions have priority orders.

- Each region has its own start coordinate X/Y (12 bits) and horizontal/vertical sizes (12 bits)
- When regions are overlapped, the value in the overlapped region is accumulated only in the region with the highest priority.
- The number of colors to be counted is from 1 to 4. Each color in all regions can be turned off counting (the `IPIPE_HST_PARA[7:4]` bit field).
- The value of each pixel is down-shifted (0 ~ 11) before counting using the `IPIPE_HST_PARA[11:8]` SHF bit field.
- When the value of a bin reaches ( $2^{20} - 1$ ), the value is saturated until the memory is cleared.
- Number of bins: 32, 64, 128, or 256

The histogram memory can be cleared at the VD signal. When the memory is cleared, the first line of each frame cannot be sampled by the histogram if the width of the frame is larger than 512. If the width of the frame is smaller than 512, the first ceil ( $512/\text{width}$ ) lines cannot be collected, where  $\text{ceil}(x)$  is the smallest integer value above  $x$ . If the clearing function is not enabled, the histogram bins are accumulated over the previous values.

The histogram has two banks of memories, which can be switched alternatively. The two memory banks are slipped into four histogram memory tables. Only two tables can be used at a time: output memory tables 0 and 1, or tables 2 and 3. To initialize tables, the `IPIPE_HST_TBL[1]` CLR bit is set to 1, and to select which set of tables is to be used the `IPIPE_HST_TBL[0]` SEL bit can be switched between 0 and 1.

A gain for each color can be applied using the `IPIPE_HST_MUL_x` registers, where  $x = R, GR, GB, \text{ or } B$ .

#### 8.3.3.4.13 ISS ISP IPIPE Boxcar

The boxcar module generates a boxcar by taking mosaic image data and averaging the red, green, and blue pixels in an ( $8 \times 8$ ) or ( $16 \times 16$ ) block to produce one red, green, and blue output, as shown in [Figure 8-152](#) and in [Figure 8-153](#). Here, similar to the histogram module, the boxcar is enabled from the `IPIPE_BOX_EN[0]` EN bit, and if the mode is set to run once (one shot) (`IPIPE_BOX_MODE[0]` OST = 1), the enable bit is cleared after the run. The size of the blocks is determined from the `IPIPE_BOX_SHF[0]` SEL bit, where if set to 0 =  $8 \times 8$ , and 1 =  $16 \times 16$ .

The result of this operation is a full-color image with ( $1/64$ ) or ( $1/256$ ) area of the original image. The maximum input horizontal width is 8190 pixels when a  $16 \times 16$  block is used; the width is 4096 when an  $8 \times 8$  block is used. Also, the image size (width and height) must be multiple of 16 for a  $16 \times 16$  block, and multiple of 8 for an  $8 \times 8$  block. Boxcar operation works on 12-bit Bayer data and outputs 16-bit data. The output data is 48-bit RGB data for each  $8 \times 8$  or  $16 \times 16$  block. The 48-bit data is aligned in 64-bit format in SDRAM as shown in [Section 8.3.3.4.13](#). The first address of SDRAM access is specified by the `IPIPE_BOX_SDR_SAD_H` and `IPIPE_BOX_SDR_SAD_L` registers. The output data are written to SDRAM continuously line by line; there is no address offset between lines. After the image transfer of each frame completes, the `ipipe_eof` signal is sent to buffer logic. This signal is issued at the same timing as `ipipe_int_dma`.

**Figure 8-152. ISS ISP IPIPE Boxcar Operation ( $8 \times 8$  Block)**

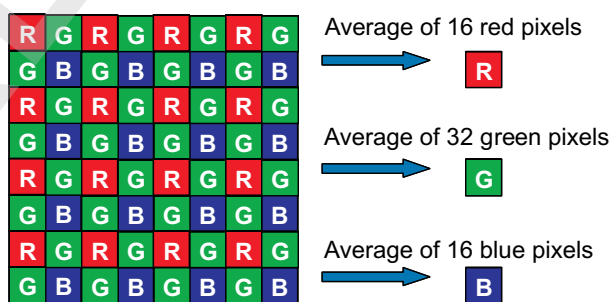




Figure 8-153. ISS ISP IPIPE Boxcar Operation (16 × 16 Block)

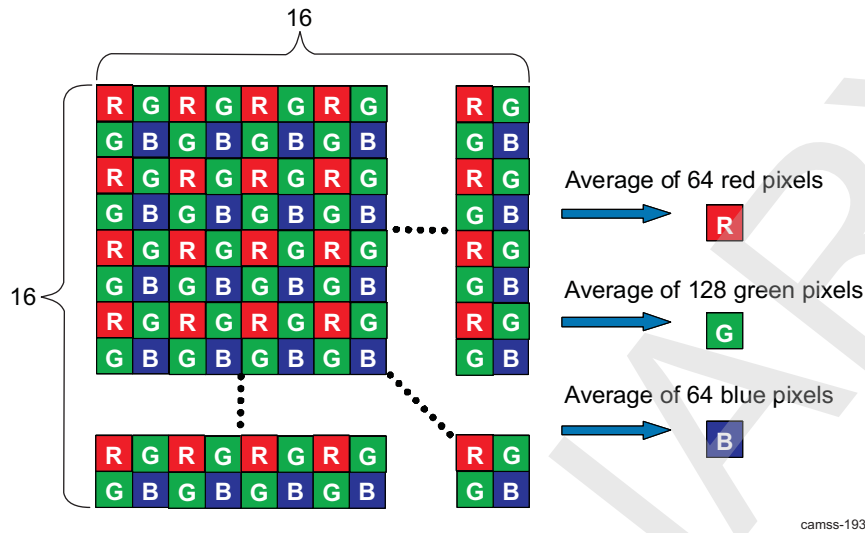
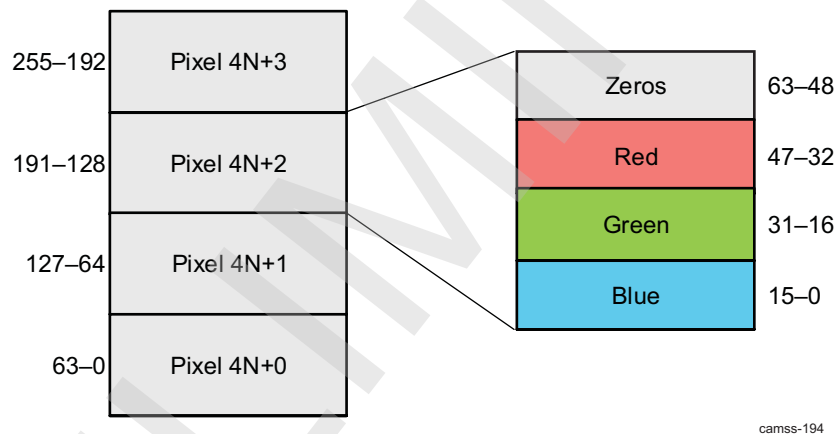


Figure 8-154. ISS ISP IPIPE Boxcar Data Packing in SDRAM



The right shift value is specified by the `IPIPE_BOX_SHF` register, which has a range of 0 to 4. (The shift down is performed to fit the 20-bit accumulated value into 16-bit output.) For green signal processing, a divide-by-two operation rounds off the least-significant bit (LSB).

### 8.3.3.5 ISS ISP RSZ Functional Description

#### 8.3.3.5.1 ISS ISP RSZ Overview

The RSZ module rescales images into various sizes ranging from x1/4096 scale-down to x16 scale-up. It also works in conjunction with the rotational engine (ROT) in SIMCOP for rotating images. The RSZ data slave interfaces support a parallel video port (VP). The RSZ module can produce two output images simultaneously, because there are two independent resizer engines. The input data used by the two resizer engines is the same.

The RSZ module has the following capabilities:

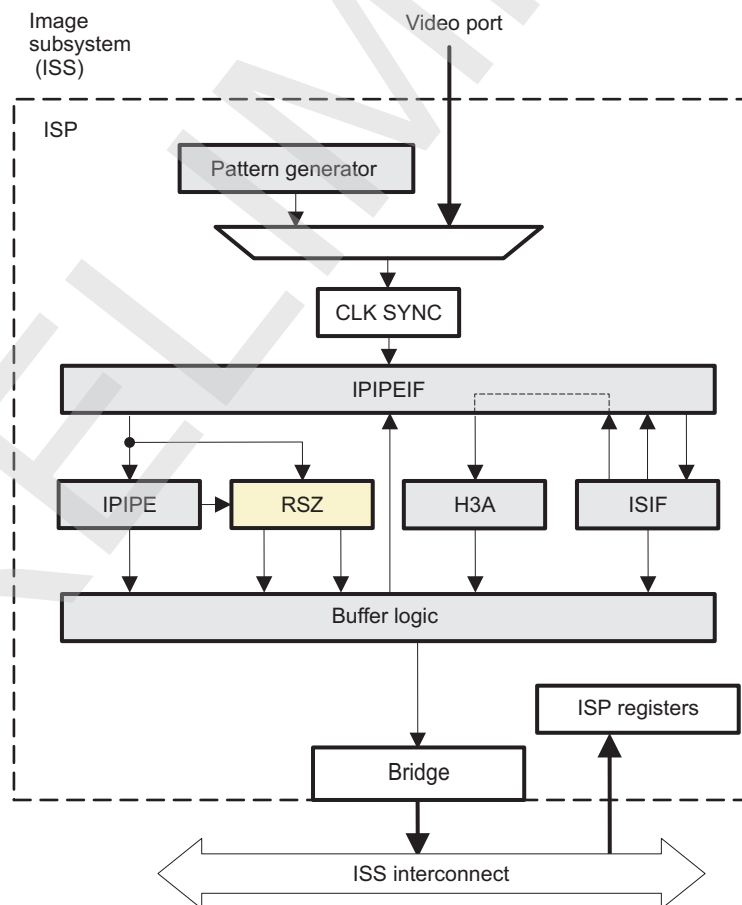
- Input/output data formats:
  - Common input data for the two resizers
  - Independent output data formats for the two resizers
  - Supports YUV4:2:2 input and YUV4:2:2/YUV4:2:0 or RGB5:6:5/ARGB32 output.
  - Supports YUV4:2:0 input data and YUV4:2:0
    - YUV4:2:0 input format not supported natively

- Two passes required: luma followed by chroma or vice versa
- Only supported from memory to memory
- Supports RAW Bayer input and RAW Bayer output
  - RAW format invariant. Takes whichever RAW format at the input and writes it out unmodified: 16 bits are read, 16 bits are written out.
  - No resizing can take place on RAW data.
- Resizer capabilities:
  - Input image cropping
    - Common for the two resizer engines: Same input data before and after cropping
    - Supported on YUV4:2:2 and RAW data formats
    - Supported on pass-through mode data path
    - Supported on bypass mode data path
  - Dual resizer engines: RSZ-A and RSZ-B
    - Up to x20 upsampling and x1/4096 downsampling on both engines
    - Up to 200-MHz pixel throughput on both resizer engines
    - Programmable data rate control to smooth the peak memory bandwidth
  - RSZ-A
    - Horizontal resolution of up to 8K pixels on RSZ-A.
  - RSZ-B
    - Horizontal resolution of up to 8K pixels on resizer RSZ-B
  - Independent Y and Cb/Cr phases on horizontal and vertical axis
    - Enables to take care of different YUV4:2:0 phases used in different video formats
    - Enables frame division mode: Images can be stitched together with the right phase.
  - Rescaling: two modes supported
    - Normal mode for upscale and downscale: Higher flexibility but lower downscale quality
    - Downscale mode for downscale only: Lower flexibility but higher downscale quality
  - Filtering: two modes supported
    - Independent settings for the horizontal and vertical directions
    - 3-tap low pass filter with 2-tap linear interpolation
    - 4-tap cubic interpolation
  - Flip support of the output image
    - Horizontal flip
    - Vertical flip
  - Pixel duplication on the top/bottom, left/right sides
    - Avoids losing pixels at the image boundaries because of the filtering
  - Support pass-through and bypass modes: Resizer engines bypassed
    - Pass-through mode
      - RAW and YUV4:2:2 data support
      - Lower power consumption mode to transfer data to memory
      - Can transfer images larger than 8K pixels to memory
    - Bypass mode
      - RAW and YUV4:2:2 data support
      - Input buffer used. Can benefit from additional buffering in case the BL module memory is not big enough and back pressure occurs.
- Slave data interface: VP interface
  - Two VP interfaces: The programming model selects which VP is used to input data to the RSZ module. Both VPs cannot be active simultaneously.
    - VP 1: Typically connected to the IPIPE module.
    - VP 2: Typically connected to the IPIPEIF module.
  - Up to 200-MHz pixel clock

- Master data interface:
  - Two interfaces to the BL module
    - Up to 200 MHz
    - 32-bit wide, 32-byte long requests
    - Accesses are aligned on 32-byte boundaries
    - Used to transfer data to memory
    - Each interface is dedicated to a single output image.
  - Addressing modes
    - Linear
    - Circular
- Configuration interface:
  - Up to 100 MHz
  - 32-bit wide
  - Used to configure the resizer registers
- Power management:
  - Independent clock domains for the two resizers
    - Each resizer engine can be gated off separately
- Error management:
  - FIFO overflow detection on the input buffers

Figure 8-155 show the RSZ module connections to other submodules of the ISP.

**Figure 8-155. ISS ISP RSZ High-Level Diagram**



camss-003c

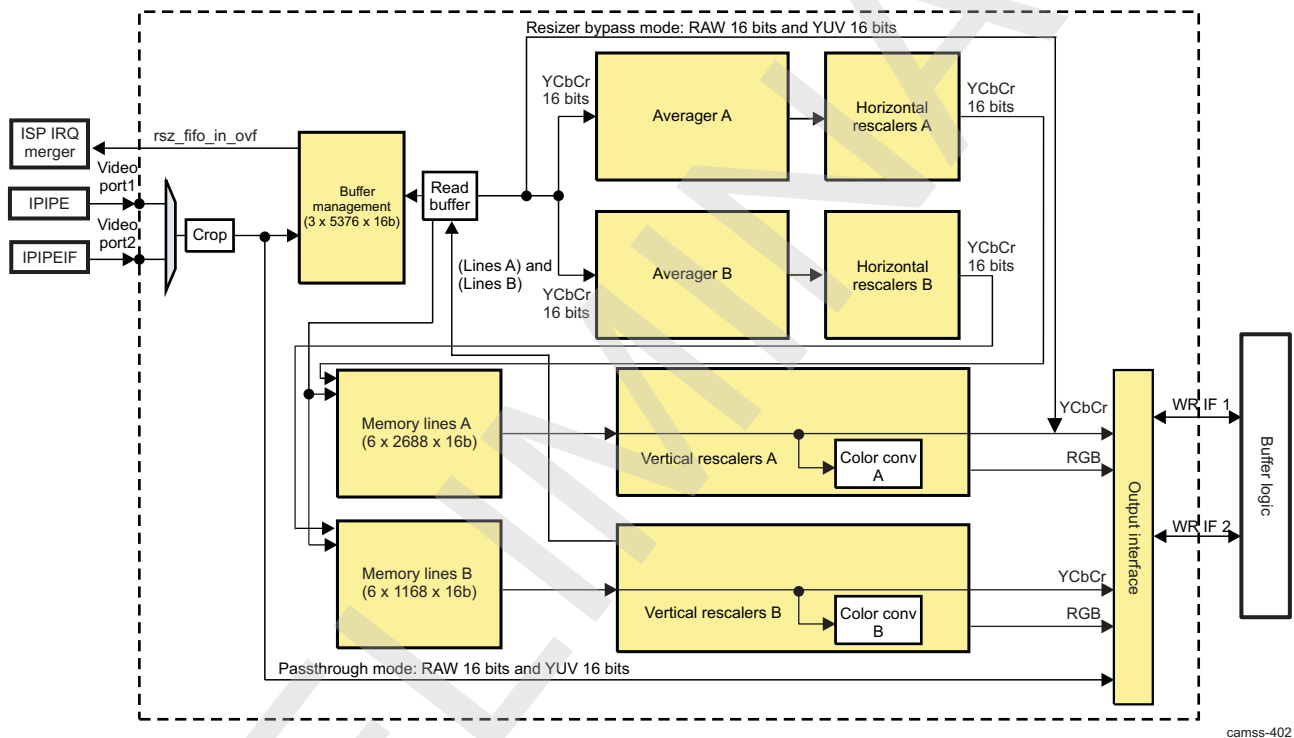
### 8.3.3.5.2 ISS ISP RSZ Top-Level Block Diagram

Figure 8-156 is the top-level block diagram of the RSZ module. The RSZ module comprises the following submodules: cropping, input data buffering, data requestor, averager, data saturation, and resizer Interpolation (comprised of horizontal rescaler, vertical rescaler, color conversion, and output interface) (see the following sections for more information).

The RSZ module comprises two independent resizer engines with the same capabilities (except for the memory line size). The input data can come from VP 1 or VP 2. Software must determine and control which interface is selected.

The RSZ module includes one VBUSP slave port, which is used to control the RSZ registers. It also includes two MTC master ports, which are used to pass the pixels to the BL. The BL in turn creates the burst requests to the memory subsystem (see Section 8.3.3.5.3, *ISS ISP RSZ Interfaces*).

Figure 8-156. ISS ISP RSZ Top-Level Block Diagram



camss-402

### 8.3.3.5.3 ISS ISP RSZ Interfaces

The RSZ module has the following data interfaces:

- One 32-bit read/write point-to-point pending VBUSP interface
- Two slave VP interfaces for transport YUV and RAW data
- TWO MTC interfaces to BL for RZA A and RZA B, with only write capabilities

#### 8.3.3.5.3.1 ISS ISP RSZ VBUSP interface

The VBUSP interface is a 32-bit read/write capable interface. The VBUSP interface must be programmed in a way that back-to-back requests are possible for read and write. The `RSZ_GCK_MMR[0]` MMR bit enables the memory register access from the VBUSP interface to enable transfer and signal such as MMR request, direction, enable write/read data can be enabled.

8.3.3.5.3.2 ISS ISP RSZ Video Port Interfaces

The VP interfaces are slave interfaces; one is connected to IPIPE, and the other to IPIPEIF. These interfaces are for data transfer. Table 8-318 lists the format supported across IPIPE/IPIPEIF and RSZ. Signals coming from IPIPE and IPIPEIF can be write-enable signals. The RSZ\_SRC\_MODE[1] WRT bit is set whether or not the write enable signals are considered. This is a line-valid qualifier. This signal is sampled on the rising edge of HD, and the value is used for the full line.

Table 8-318. ISS ISP RSZ VP Supported Formats

	VP Signals: From IPIPE and IPIPEIF Modules (dat[15:0] Register)															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW16	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
YUV4:2: 2 16 bits	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Cb7 Cr7	Cb6 Cr6	Cb5 Cr5	Cb4 Cr4	Cb3 Cr3	Cb2 Cr2	Cb1 Cr1	Cb0 Cr0
YUV4:2: 0 Y data	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Low	Low	Low	Low	Low	Low	Low	Low
YUV4:2: 0 Cb/Cr data	Low	Low	Low	Low	Low	Low	Low	Low	Cb7 Cr7	Cb6 Cr6	Cb5 Cr5	Cb4 Cr4	Cb3 Cr3	Cb2 Cr2	Cb1 Cr1	Cb0 Cr0

**NOTE:** The formats are set from the IPIPE and IPIPEIF registers. For more information, see Section 8.3.3.4 and Section 8.3.3.3.

8.3.3.5.3.3 ISS ISP RSZ MTC interfaces

The RSZ module includes two write-only MTC interfaces. Their implementation enables passing a maximum of eight 32-byte requests in ten clock cycles. The RSZ must be programmed to obtain smooth and average bandwidth to buffer logic module by setting a minimum interval between two successive requests (set the RSZ\_DMA\_RZA[15:0] RZA and RSZ\_DMA\_RZB[15:0] RZB bit fields for the A and B resizers as appropriate). This setting is not expected to be dynamic. It can be fixed setting from request to request and frame to frame. When the bandwidth is set appropriately, between the first valid translated pixel and the EOF signal sent to buffer logic, the RSZ\_DMA\_STA[0] STATUS bit can be seen, and it is high if the transfer over the MTC interfaces is active. Figure 8-157 shows how RSZ\_DMA\_RZx for resizers A and B affects the MTC data request generator.

Figure 8-157. ISS ISP RSZ MTC DMA Bandwidth Control

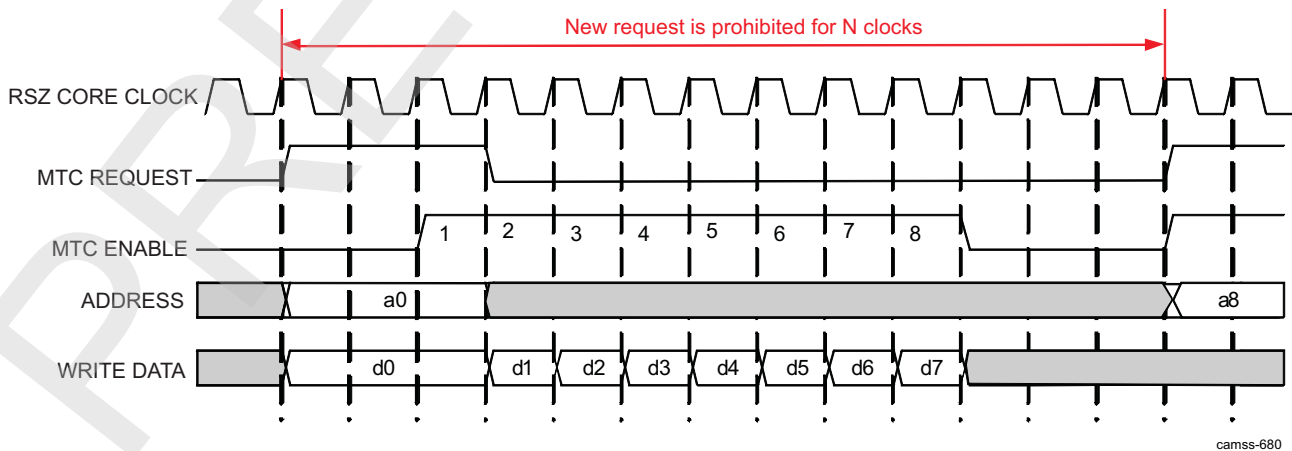
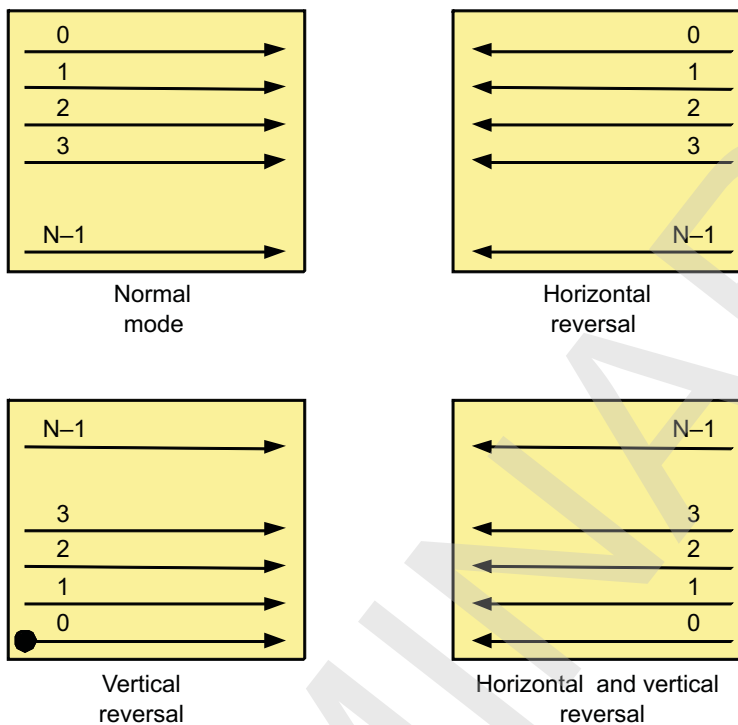


Figure 8-158 shows the pixel order in memory written by the MTC. The arrows do not represent the order in which data is written. Data are always written from left to right, whether horizontal reversal is enabled or not.

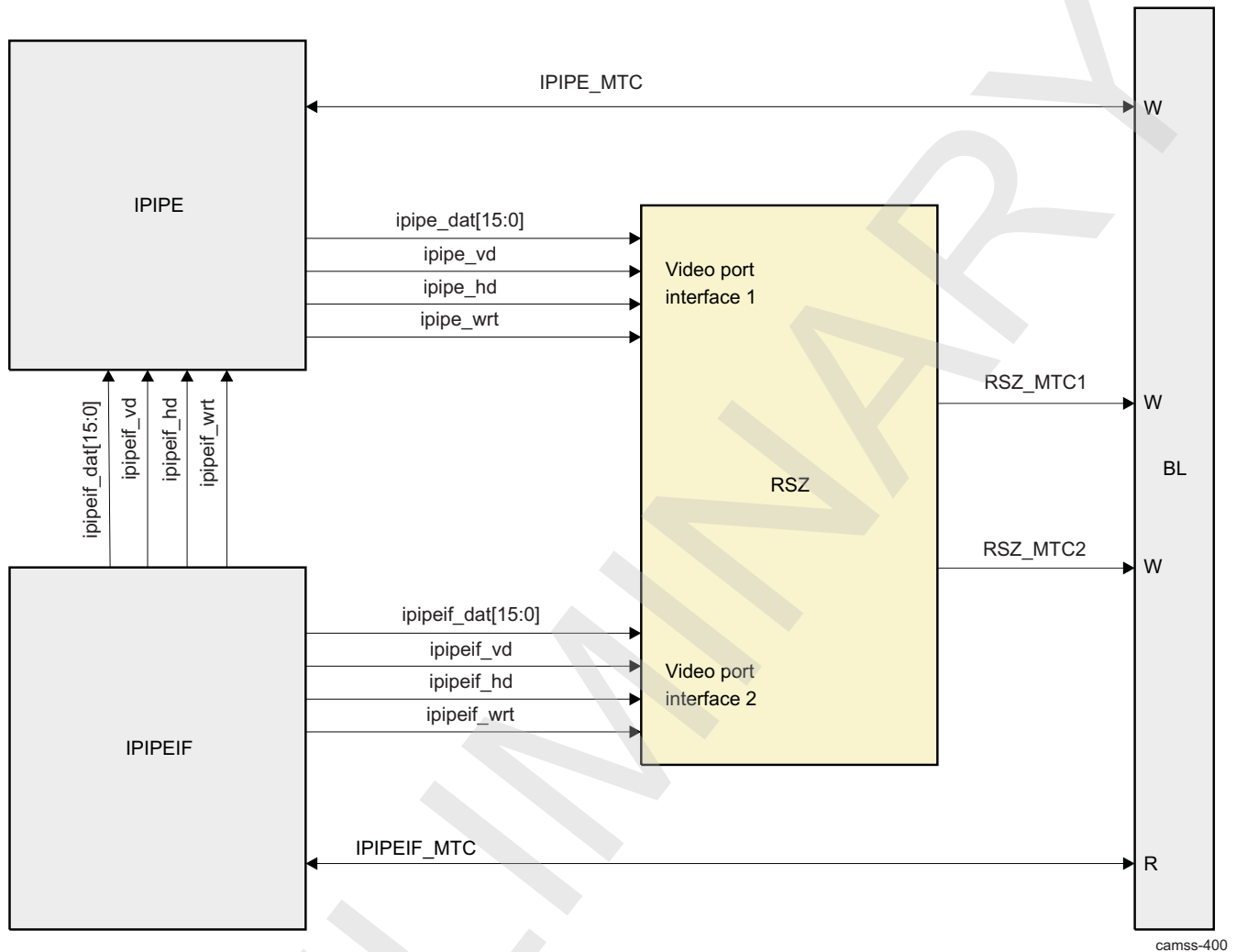
**Figure 8-158. ISS ISP RSZ MTC Image Data Storage Pixel Order**


camss-684

#### 8.3.3.5.4 ISS ISP RSZ Integration

Figure 8-159 shows how the VP and interfaces of the RSZ module are connected to surrounding modules at the ISP level. The RSZ module gets data from the IPIPEIF module or IPIPE module.

Figure 8-159. ISS ISP RSZ Typical Module Integration: High-Level Summary



The following constraints apply to the RSZ module:

- The data coming from the IPIPEIF module can be RAW or YUV4:2:2 data. Because the RSZ module can rescale only YUV4:2:2 data, the RSZ module must be configured in pass-through mode when RAW data is received on VP 2. It is possible to bypass the RSZ engine if YUV4:2:2 data is sent but rescaling is not needed (bypass mode).
- The data coming from the IPIPE module can be RAW or YUV4:2:2 data. Eventually, YUV4:2:0 data can be sent through this path, but the data must be sent in two passes. Because the RSZ module can rescale only YUV4:2:2 data, the RSZ module must be configured in pass-through mode when RAW data is received on VP 1. It is possible to bypass the RSZ engine if YUV4:2:2 data is sent but rescaling is not needed (bypass mode).

VP 2 provides a way to bypass the IPIPE module when YUV4:2:2 data is received from the images sensor or when YUV4:2:2 is read back from memory with the CCP2 RX or ISIF module.

Table 8-319 summarizes the different RSZ configuration possibilities as a function of the input data format.

Table 8-319. ISS ISP RSZ Data Flow vs. Input Data Format Constraints

VP 1 Data Format	VP 2 Data Format	RSZ-A Configuration	RSZ-B Configuration	Comments
RAW	N/A	Disabled	Disabled	RSZ module in pass-through mode

**Table 8-319. ISS ISP RSZ Data Flow vs. Input Data Format Constraints (continued)**

VP 1 Data Format	VP 2 Data Format	RSZ-A Configuration	RSZ-B Configuration	Comments
N/A	RAW	Disabled	Disabled	RSZ module in pass-through mode
YUV4:2:2	N/A	Disabled	Disabled	RSZ module in pass-through mode or bypass mode
YUV4:2:2	N/A	Enabled	Disabled	One output image
YUV4:2:2	N/A	Disabled	Enabled	One output image
YUV4:2:2	N/A	Enabled	Enabled	Two output images
N/A	YUV4:2:2	Disabled	Disabled	RSZ module in pass-through mode or bypass mode
N/A	YUV4:2:2	Enabled	Disabled	One output image
N/A	YUV4:2:2	Disabled	Enabled	One output image
N/A	YUV4:2:2	Enabled	Enabled	Two output images

### 8.3.3.5.5 ISS ISP RSZ Functional Description

To start up, the RSZ configuration can be set from the [RSZ\\_SYSCONFIG](#) register, which provides enabling the RSZ-A and RSZ-B clocks. The RSZ module does not have standalone reset and status check. Software reset must be done at the ISP level. Moreover, when enabled, the RSZ module can control the input data buffer, and when the `rsz_stall_input` signals are set from the [RSZ\\_IN\\_FIFO\\_CTRL](#) register, the RSZ module generates a stall signal that can be used by the master module sending data to the RSZ module when the data threshold is too high.

The [RSZ\\_SRC\\_EN\[0\]](#) EN bit starts the resizer processing. If the processing mode is set to one shot (one run and then turn off) from the [RSZ\\_SRC\\_MODE\[0\]](#) OST bit, the EN bit is cleared to 0.

The resizer can be configured to be bypassed in certain cases (see [Figure 8-159](#) for the module constraints) from the [RSZ\\_SRC\\_FMT0\[1\]](#) BYPASS bit. The data can be sent directly from here to the output interface (bypass mode) or imported to the module buffer, but not manipulated and sent to the output interface (pass-through mode). The master device sending data to the resizer can be switched between IPIPEIF and IPIPE using the [RSZ\\_SRC\\_FMT0\[0\]](#) SEL bit. The resizer understanding of the data input is set from the [RSZ\\_SRC\\_FMT1](#) register (for more information, see [Table 8-321](#)). The [RSZ\\_SEQ.VRVX](#) and [RSZ\\_SEQ.HRVX](#) registers can be set to flip the image horizontally or vertically, respectively (see [Figure 8-158](#)).

Depending on the mode to which the resizer is set, the core clock can be enabled from the [RSZ\\_GCK\\_SDR](#) register. [Table 8-320](#) summarizes the behavior of the module for the different settings.

**Table 8-320. ISS ISP RSZ Module Modes: Register Settings**

Configuration Number	RSZ_SRC_EN	RZA_EN	RZA_CLK_EN	RZB_EN	RZB_CLK_EN	RSZ_GCK_SDR_CORE	RSZ_SRC_FMT0.BYPASS	Comments
0	0	X	X	X	X	X	X	Data cannot go through the RSZ module. Interrupts are not issued.
1	1	0	X	0	X	1	0	Resizer A is disabled. Resizer B is disabled. It is best to have <code>RZA_EN = RZB_EN = 0</code> to save power, but <code>RZA_EN = RZB_EN = 1</code> is also supported.
2	1	1	1	0	X	1	0	This configuration is supported but does not make sense because data cannot go through the module. Resizer A is enabled. Resizer B is disabled.



**Table 8-320. ISS ISP RSZ Module Modes: Register Settings (continued)**

								It is best to have <b>RZB_EN</b> = 0 to save power, but <b>RZB_EN</b> =1 is supported as well.
3	1	0	X	1	1	1	0	Resizer A is disabled. Resizer B is enabled.
								It is best to have <b>RZA_EN</b> = 0 to save power, but <b>RZA_EN</b> = 1 is also supported.
4	1	1	1	1	1	1	0	Resizer A is enabled. Resizer B is enabled
5	1	X	X	X	X	0	0	Bypass mode is enabled. Resizer core functional clock is disabled.
6	1	X	X	X	X	0	1	Pass-through mode enabled. Resizer core functional clock is disabled.
7	1	X	X	X	X	1	1	Pass-through mode is enabled. Resizer core functional clock is enabled.
								Not a preferred configuration. Configuration (6) saves power.

**Table 8-321. ISS ISP RSZ Module Input Control: Register Settings**

<b>RSZ_SRC_FMT1.IN420</b>	<b>RSZ_SRC_FMT1.COL</b>	<b>Comments</b>
0	X	YUV4:2:2 input. Chrominance is cosited.
1	0	YUV4:2:0 input. Valid data is Y, C is dummy. On the VP, YUV4:2:2 data is always assumed.
1	1	YUV4:2:0 input. Valid data is C, Y is dummy. On the VP, YUV4:2:2 data is always assumed.

The **RSZ\_YUV\_PHS[0]** POS bit sets the chrominance output. The RSZ module does not change the relative position of the chroma samples versus the luma samples between the input and output, and the chroma position at the output of the IPIPE module and at the output of the RSZ module must be identical. In other words, **RSZ\_YUV\_PHS.POS** = **IPIPE\_YUV\_PHS.POS**.

Settings are common for both resizer engines inside the RSZ module. Each engine (RZA or RZB) can be enabled from the **RZx\_EN** register: select the mode from **RZx\_MODE**, and select the input and output in the YUV color scheme from the **RZx\_420** register (valid only if YUV 4:2:2 is the input set from **RSZ\_SRC\_FMT1.IN420**). [Table 8-322](#) summarizes the combination of settings available in the **RZx\_420** register.

**Table 8-322. ISS ISP RSZ-A/RSZ-B Output Format Selection**

<b>RZx_420.YEN</b>	<b>RZx_420.CEN</b>	<b>Comments</b>
0	0	Input is YUV4:2:2. Output is YUV4:2:2 if <b>RZX_RGB_EN</b> = 0 and RGB if <b>RZB_RGB_EN</b> = 1.
0	1	Input is YUV4:2:2. Output is the chroma of YUV4:2:0. <b>RZX_RGB_EN</b> is ignored. Must be used to rescale YUV4:2:0 data: 1st/2nd pass
1	0	Input is YUV4:2:2. Output is the luma of YUV4:2:0. <b>RZX_RGB_EN</b> is ignored. Must be used to rescale YUV4:2:0 data: 2nd/1st pass
1	1	Input is YUV4:2:2. Output is YUV4:2:0. <b>RZX_RGB_EN</b> is ignored.

### 8.3.3.5.5.1 ISS ISP RSZ Operating Modes

The RSZ module offers two basic rescaling modes. These modes are not built-in but are particular configurations, which means that other hybrid modes can be programmed. The normal mode provides more flexibility (the rescale ratio granularity is smaller) than downscale mode, but downscale mode produces better image quality (averager performs anti-aliasing):

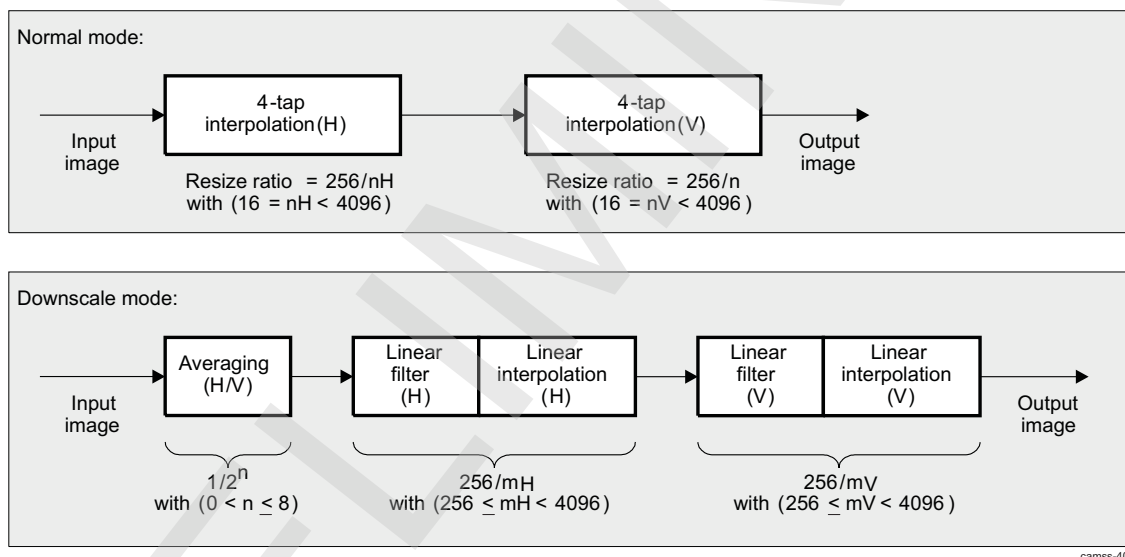
- Normal mode: The scaling process is carried out using interpolation with a 4-tap filter. The interpolation method is a 4-tap cubic convolution or a 3-tap linear filter + 2-tap linear interpolation. The user controls the type of interpolation that is used. The possible rescale ratios range from 1/16x to 20x.
- Downscale mode: The scaling process is the same as for normal mode, but an averaging function is placed before. It enables reaching much higher reduction factors while avoiding anti-aliasing artifacts. The interpolation method is a 4-tap cubic convolution or a 3-tap linear filter + 2-tap linear interpolation. The possible rescale ratios range from 1/4096x to 1x.

**NOTE:** The selection of the mode is independent for each resizer engine. One resizer engine can be configured in normal mode, while another is configured in downscale mode.

The RSZ module can produce two output images simultaneously, because there are two independent resizer engines. The input data used by the two resizer engines is the same. The RSZ output image sizes are limited to `RSZ_GNC[0:12]` `RSZA_MEM_LINE_SIZE` pixels/line for RSZ-A (5376 pixels/line) and `RSZB_MEM_LINE_SIZE[16:28]` for RSZ-B (2336 pixels/line).

Figure 8-160 shows the RSZ operating modes. Note that nothing prevents to use linear interpolation in normal mode or bicubic interpolation in downscale mode, similarly, one may want to mix the interpolation modes for horizontal and vertical filtering. This is fully programmable.

**Figure 8-160. ISS ISP RSZ Operating Modes**



camss-401

### 8.3.3.5.5.1 ISS ISP RSZ Operating Modes and Maximum Input Clock

The maximum output pixel clock on both resizers is 200 MHz (100 percent optimal power performance [OPP]), that is, a pixel throughput of 200 MPix/s. Moreover, hardware takes care of the following constraints:

- When both resizer engines are configured to perform downscaling, there is no particular constraint on the VP pixel clock. The VP pixel clock can be as high as 200 MHz.
- When one resizer engine is configured to perform upscaling and the second resizer engine is configured to perform downscaling, the VP pixel clock must be limited. The VP must be lower than:

$$\text{clk\_pix} \leq (200 \text{ MHz} / (\text{Vertical Upscale Ratio} * \text{Horizontal Upscale Ratio})) \quad (5)$$

For example, if a 4x upscale ratio happens horizontally and vertically, then the input pixel clock must be lower than  $200 / (4 * 4) = 12.5$  MHz.

It is the reason why it is not possible to perform digital zoom upscaling on the fly. It is necessary to acquire the pixels to memory first and to read them back at a pace that does not exceed the previously discussed constraints. At the ISS level, data can be read back from memory from the CCP2 RX or ISIF module.

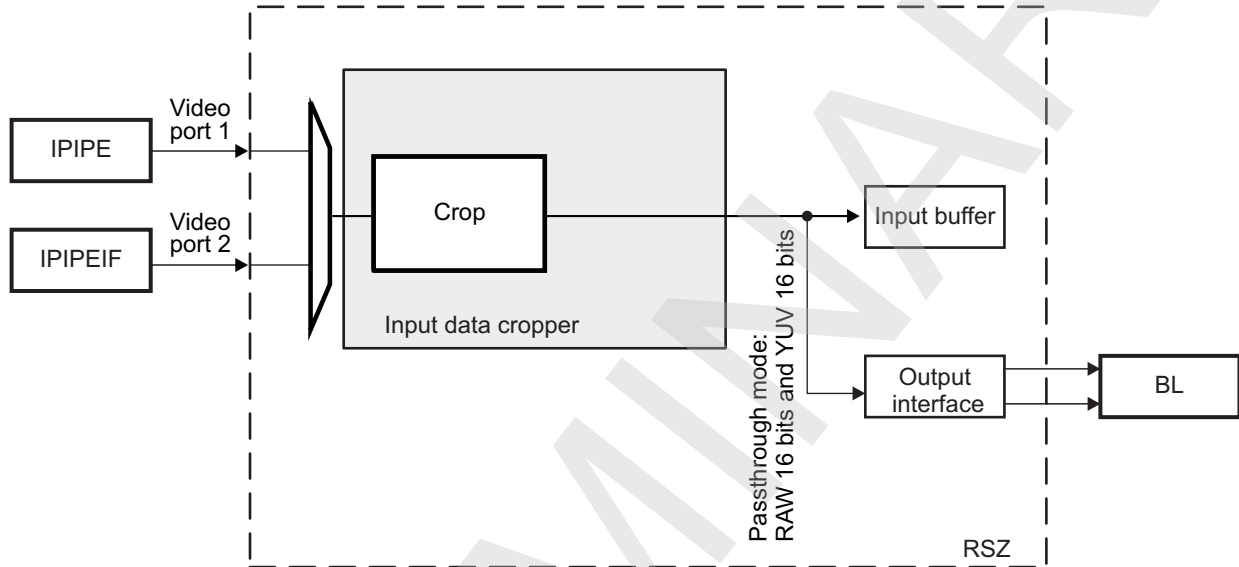
- When the two resizer engines are configured to perform upscaling, the VP pixel clock must be limited.

In that case, the VP frequency is limited by the resizer engine having the larger rescale ratios.

### 8.3.3.5.5.2 ISS ISP RSZ Input Data Cropper

The data coming from the VPs into the RSZ module can be cropped: this applies to RAW and YUV4:2:2 data. It is mandatory to crop the data as early as possible in the RSZ processing pipeline to reduce power consumption. It is mandatory to crop the data before storing it in the input data buffer. Figure 8-161 is the block diagram of the RSZ input data cropper.

Figure 8-161. ISS ISP RSZ Input Data Cropper Block Diagram



camss-708

The input data are in YUV4:2:2 interleaved format or RAW format.

For YUV4:2:2 format, the data come as Cb0, Y0, Cr0, Y1, Cb2, Y2, Cr2, Y3, etc. Y denotes the luma component value and Cb/Cr denotes the chroma component values. There are as many Y components as Cb/Cr components per line.

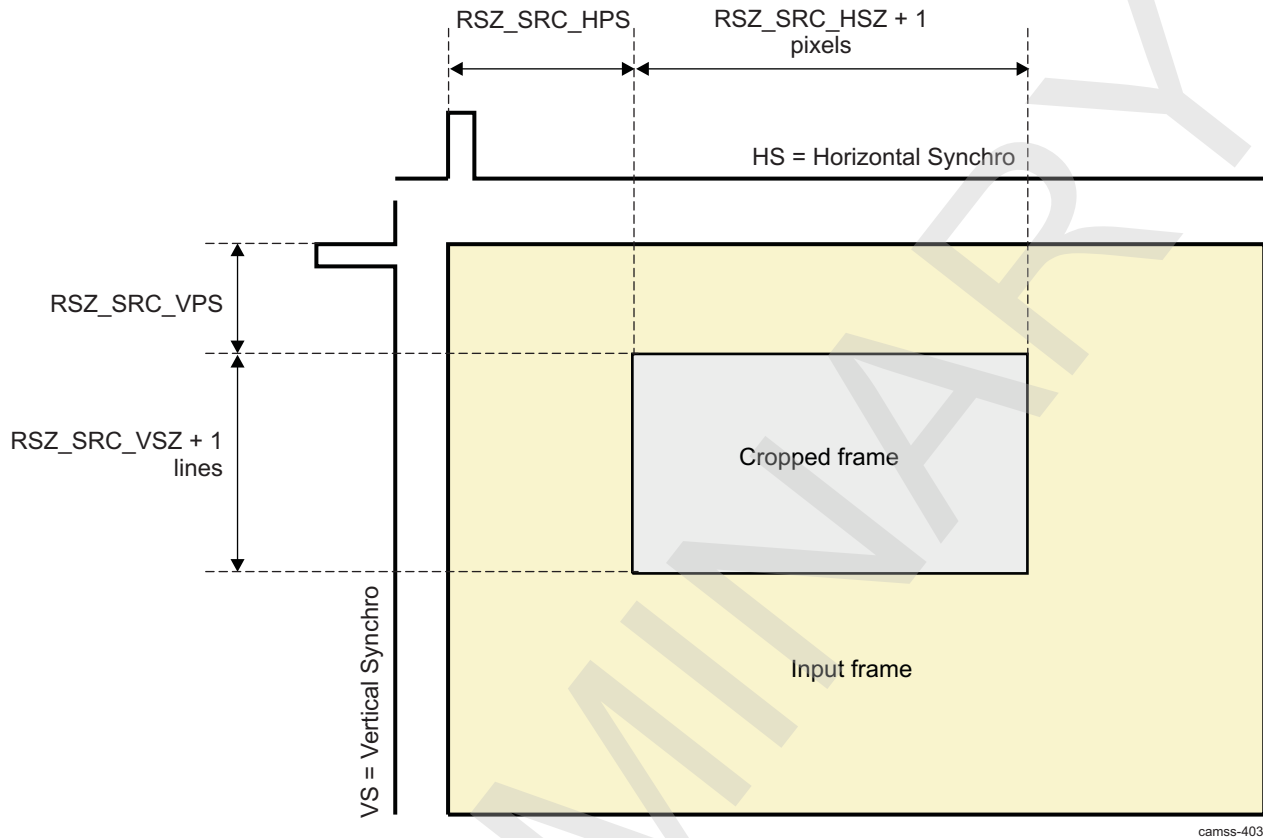
Figure 8-162 shows input data cropping. Only the cropped data is stored in the input data buffer. The names used in the figure correspond to the register names. If no cropping is desired, the [RSZ\\_SRC\\_HPS](#) and [RSZ\\_SRC\\_VPS](#) registers, which set the horizontal and vertical positions, must be set to 0; the start size from the [RSZ\\_SRC\\_HSZ](#) register must be set to the input image width minus 1, and the [RSZ\\_SRC\\_VSZ](#) register must be set to the input image height minus 1. These are typical settings for both resize engines inside the RESIZER. After setting, more flexibility is present through [RZx\\_i\\_VPS](#) and [RZx\\_i\\_HPS](#) for vertical and horizontal positioning, respectively, of the input/output (where x = A or B, i = I or O).

Depending on the input data format, different constraints apply to the registers that set the cropping parameters:

- For YUV4:2:2 format, the vertical start positions of the cropped frame can be even or odd. However, the horizontal start position must be even: the reason is to always start with the same pattern: Cb<sub>2n</sub>, Y<sub>2n</sub>, Cr<sub>2n</sub>, Y<sub>2n+1</sub>... For the same reason, the horizontal size of the cropped frame ([RSZ\\_SRC\\_HSZ](#) + 1) must be an even number. Finally, the vertical size of the cropped frame can be odd or even.
- For RAW format, the vertical start position of the cropped frame can be even or odd. The vertical size can be even or odd. The horizontal resolution must be even.

These features and constraints are common for both resizer engines inside the resizer module.

Figure 8-162 shows the input data cropping.

**Figure 8-162. ISS ISP RSZ Input Data Cropping**

### 8.3.3.5.5.3 ISS ISP RSZ Averager

#### 8.3.3.5.5.3.1 ISS ISP RSZ Use Cases

The two RSZ engines can have independent averager settings:

- Both resizers can use it.
- One resizer can bypass it and the other can use it.
- Both resizers can bypass it.

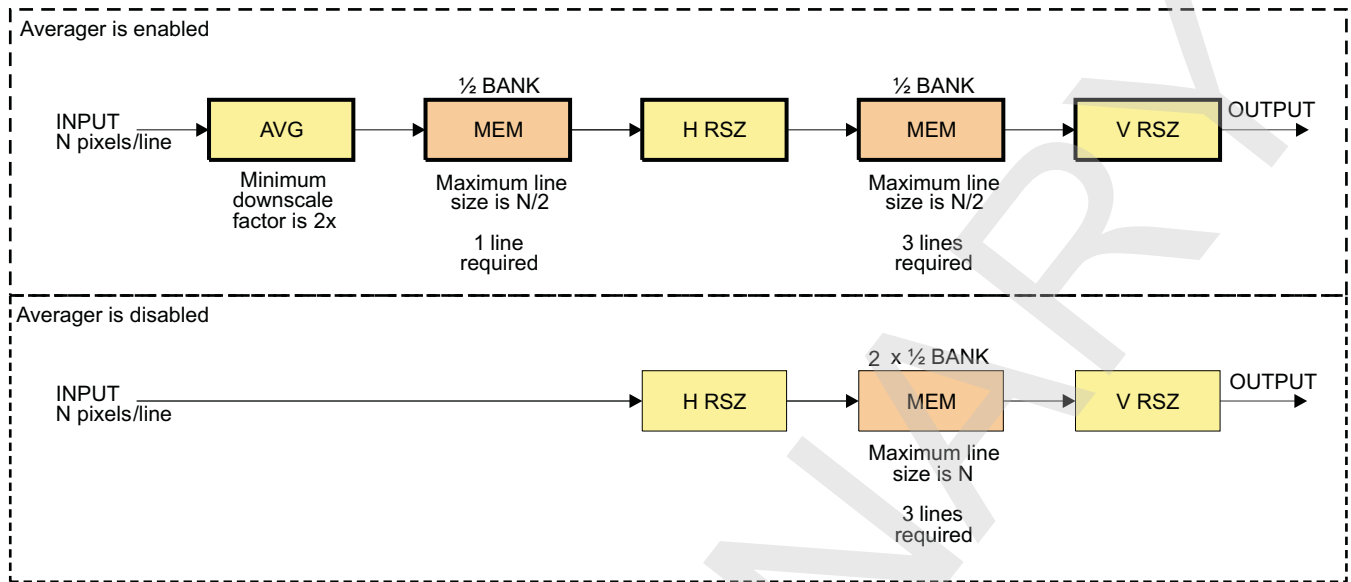
#### 8.3.3.5.5.3.2 ISS ISP RSZ Memory Utilization

Vertical averaging requires memory to perform pixel data accumulation. It shares the vertical memory lines that are used for vertical filtering: this is the reason why the vertical memory lines are organized as two banks of half lines.

- The averagers output lines that are at most half the size of the input image in one memory bank.
- The horizontal resizers write their output data in the second memory banks.

Figure 8-163 shows the use of memory when the averager is enabled or disabled.

Figure 8-163. ISS ISP RSZ Averager Memory Utilization

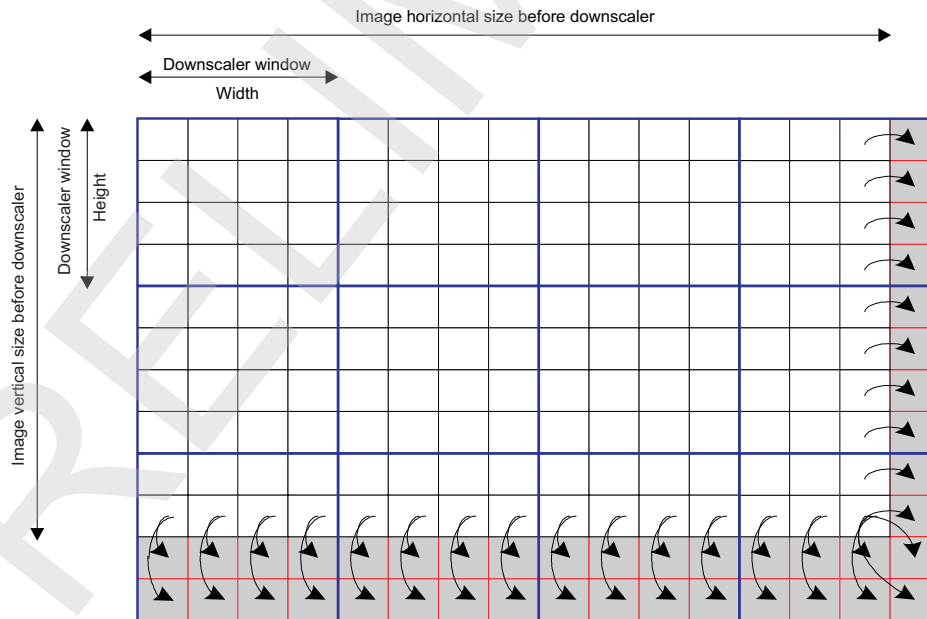


camss-407

8.3.3.5.5.3.3 ISS ISP RSZ Border Conditions

Figure 8-164 shows the averager behavior for border conditions. If the input image is not big enough, border duplication must occur.

Figure 8-164. ISS ISP RSZ Averager Border Conditions



camss-408

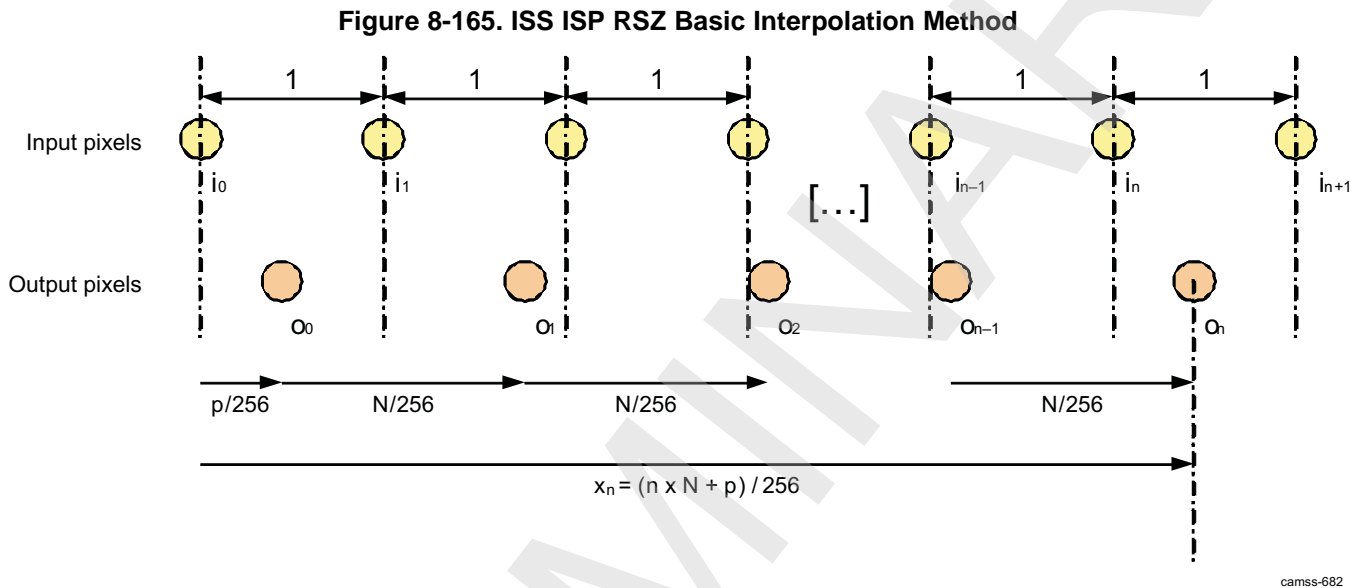
Downscaling is enabled from the RZi\_DWN\_EN register. Moreover, the vertical averaging size is set by the RZA\_DWN\_AV[3:5] V bit field. The actual downscale ratio is given by  $1/2^{(RZA\_DWN\_AV[3:5] V + 1)}$ . The range is from 1/2 to 1/256 in power of two. The horizontal averaging size is set by the RZA\_DWN\_AV[0:2] H bit field. The actual downscale ratio is given by  $1/2^{(RZA\_DWN\_AV[0:2] H + 1)}$ . The equations are the same for RSZ-B. The range goes from 1/2 to 1/256 in power of 2.

#### 8.3.3.5.5.4 ISS ISP RSZ Interpolation

Figure 8-165 shows the basic interpolation method used in the RESIZER module. The following assumptions are made:

- The distance between each input pixel is 1.
- The magnification ration is given by  $256/N$  and  $p/256$  is the initial phase of the output data.

The output pixels are also evenly spaced. The distance between each output pixel is given by  $N/256$ . In the example in Figure 8-165,  $N$  is greater than 256. The position of the  $n^{\text{th}}$  output pixel is given by  $(n \times N + p) / 256$ .



camss-682

Assuming the intensity of input pixels is  $i_0, i_1, i_2$ , etc. and the resized (output) pixels are  $o_0, o_1, o_2$ , etc., the  $n^{\text{th}}$  output pixel ( $o_n$ ) is determined using the nearest 4 input pixels as follows:

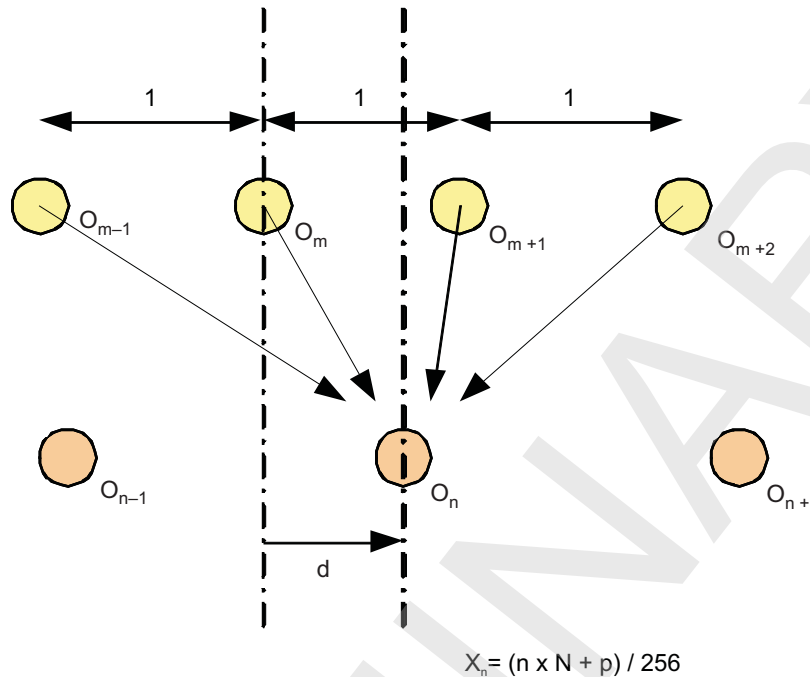
$$o_n = h(1+d) \times i_1 + h(d) \times i_m + h(d-1) \times i_{m+1} + h(d-2) \times i_{m+2}$$

In the previous equation  $h(x)$  is the interpolation main function. The RESIZER module supports linear and bicubic convolution interpolation functions.

Figure 8-166 shows the interpolation principle at the  $n^{\text{th}}$  output pixel ( $o_n$ ) at position  $x_n$ . Furthermore, the  $m$  and  $d$  parameters are as follows:

$$m = \text{floor}((n \times N + p) / 256) \text{ and } d = ((n \times N + p) / 256) - m$$

Figure 8-166. ISS ISP RSZ Interpolation Filtering



camss-683

For each resizer (RSZ-A or RSZ-B), and for chrominance and luminance, the interpolation method for vertical interpolation can be a 2-tap linear interpolation or a 4-tap cubic convolution (default) method. The choice is made in RZi\_V\_TYP[0] Y for luminance and RZi\_V\_TYP[1] C for chrominance, where  $i = A$  or  $B$  and is the resizer number. It is similar for horizontal interpolation from RZi\_H\_TYP[0] Y for luminance and RZi\_H\_TYP[1] C for chrominance, where  $i = A$  or  $B$  and is the resizer number.

#### 8.3.3.5.5.4.1 ISS ISP RSZ Liner Interpolation Input Data

Before data interpolation, a low pass filtering (LPF) operation is required on the input data. The following equation gives the LPF function. The equation is evaluated at pixel position  $d_i$  but neighbor pixels  $d_{i-1}$  and  $d_{i+1}$  are required. The gain value  $g$  is set up by the register RZi\_V\_LPF and RZi\_H\_LPF. Different gains are possible horizontally and vertically as well as for Luma and Chroma.

$$LPF_g(d_{i-1}, d_i, d_{i+1}) = d_i + g \times (d_{i-1} - 2d_{i+1} + d_{i+2})/128$$

#### 8.3.3.5.5.4.1.1 ISS ISP RSZ Cubic Convolution Mode

The input data is not modified in bicubic mode. Basically, the input is equal to the output.

#### 8.3.3.5.5.4.1.2 ISS ISP RSZ Phase Settings

The initial value for the phase value for vertical resizing is set by the RZi\_V\_PHS\_Y for luminance and RZi\_V\_PHS\_C for chrominance. These values are in the U14Q8 fractional format (values in the range [0 – 63.996]). When YUV4:2:2 data are output, the phase value for Luma and Chroma must be aligned; that is,  $RZi_V_PHS_Y = RZi_V_PHS_C$ .

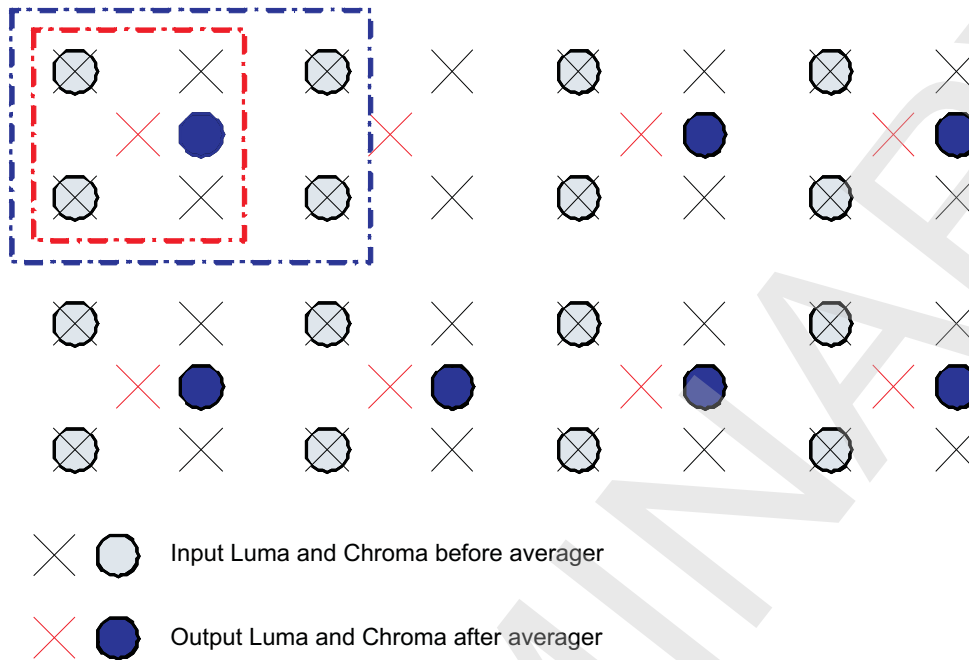
The following constraint equation applies:  $|RZi_V_PHS_Y - RZx_V_PHS_C| \leq RZi_V_DIF$ . This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two Luma pixels. The absolute value is used; therefore, the initial Luma phase can be greater than the initial Chroma phase or vice versa. As a reminder, the distance between two output pixels for Luma is given by RZi\_V\_DIF.

The initial value for the phase value for horizontal resizing is set by the RZi\_H\_PHS bit field. The RZi\_H\_PHS\_ADJ register enables adjusting the horizontal phase for the Luma component when averaging is enabled (the averager disrupts the relative sampling point between luminance and chrominance when YUV4:2:2 cosited data is input). The relative phase between Luma and Chroma is different before and after the horizontal averager. The vertical phase is not affected by the averager. [Figure 8-167](#) shows the effect of the averager on the phases. RZi\_H\_PHS\_ADJ is expected to be equal to zero if the averager is disabled.

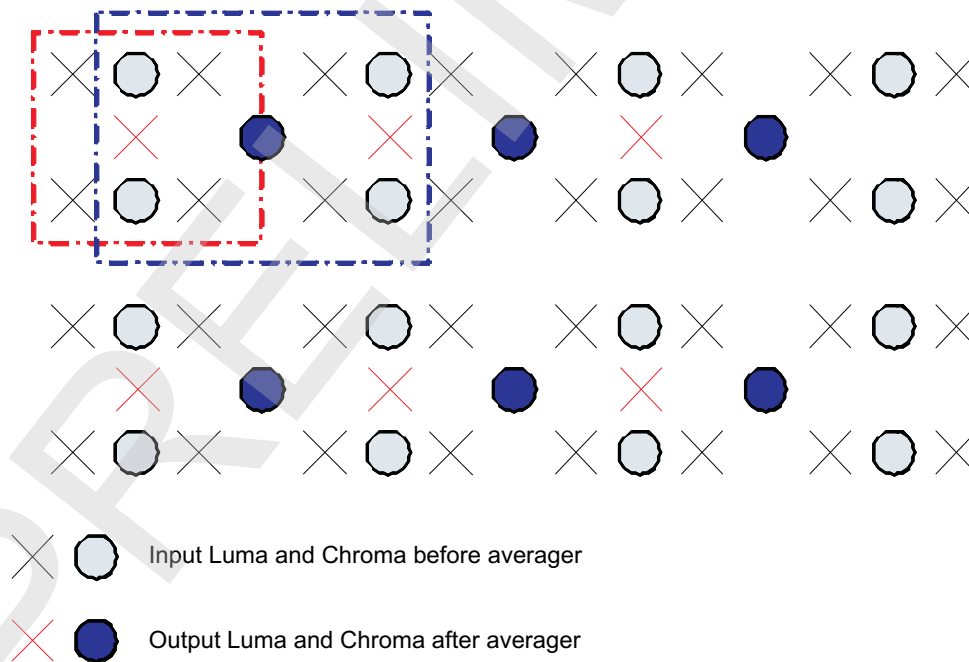


**Figure 8-167. ISS ISP RSZ-A/RSZ-B Phase Averager Effect**

Input Chroma is co-sited: relative input I/O phases btw Y and UV are different, correction is needed.



Input Chroma is centered: relative input I/O phases btw Y and UV are identical, no correction is needed.



camss-685

### 8.3.3.5.5.5 ISS ISP RSZ Data Saturator

After vertical rescaling and before color conversion, the output data is saturated (clipped) to programmable values that are given by the following registers:

- [RSZ\\_YUV\\_Y\\_MIN](#)
- [RSZ\\_YUV\\_Y\\_MAX](#)
- [RSZ\\_YUV\\_C\\_MIN](#)
- [RSZ\\_YUV\\_C\\_MAX](#)

The maximum Y value is set up with the [RSZ\\_YUV\\_Y\\_MAX](#) register. If the Input Y value is greater than the MAX value, it is clipped to MAX.

The minimum Y value is set up with the [RSZ\\_YUV\\_Y\\_MIN](#) register. If the Input Y value is smaller than the MIN value, it is clipped to MIN.

The maximum Cb/Cr value is set up with the [RSZ\\_YUV\\_C\\_MAX](#) register. If the Input Cb/Cr value is greater than the MAX value, it is clipped to MAX.

The minimum Cb/Cr value is set up with the [RSZ\\_YUV\\_C\\_MIN](#) register. If the Input Cb/Cr value is smaller than the MIN value, it is clipped to MIN.

### 8.3.3.5.6 ISS ISP RSZ Color Convertor

As mentioned previously, the resizer can support RAW, YUV4:2:0, and YUV4:2:2 formats. The resizer engines can also support RGB output: RGB5:6:5 and ARGB32.

The RGB5:6:5 data is 16 bits wide and consists of 5 bits for red, 6 bits for green, and 5 bits for blue.

The following table shows the way RGB5:6:5 is stored to memory. This data format is compatible with the display controller. Only the little-endian memory representation is supported.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R1					G1						B1					R0				G0			B0								

The ARGB32 data is 32 bits wide and consists of 8 bits for alpha, 8 bits for red, 8 bits for green, and 8 bits for blue. The alpha value is global and is set for the entire frame; registers control the alpha value: the RZx\_RGB\_BLD register controls the alpha values of resizer A and resizer B.

The following table shows the way ARGB32 is stored to memory. This data format is compatible with the display controller. This representation is endianness invariant: it is the same for little endian and big endian.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A								R								G								B							

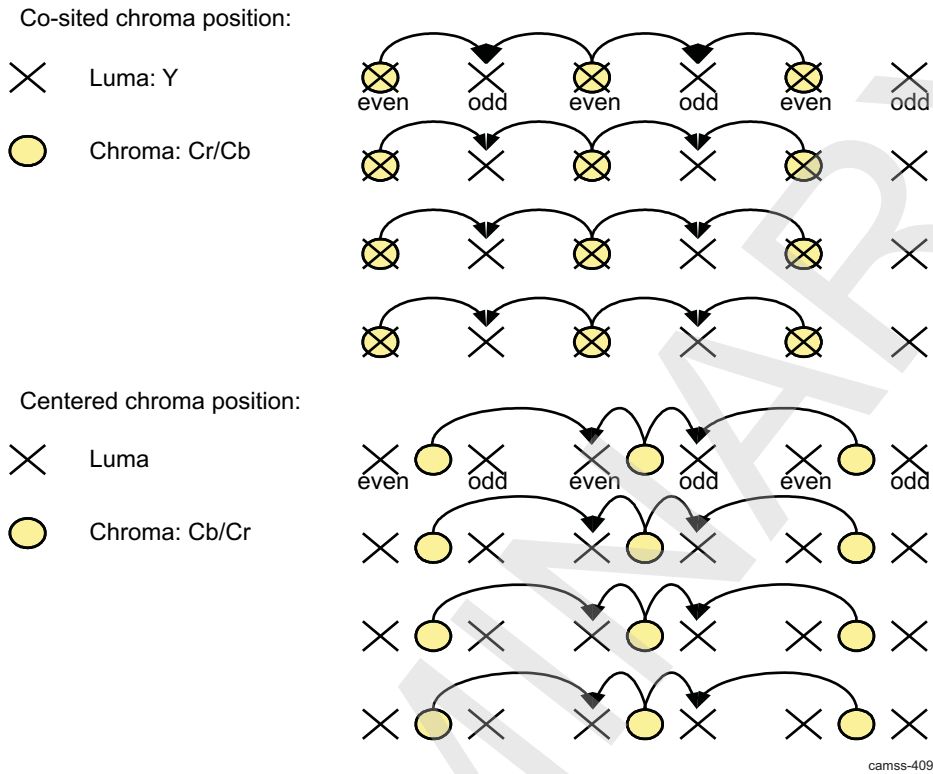
The RGB output is enabled by setting the RZx\_RGB\_EN[0] RGB\_EN bit to 1.

The RGB format is set by the RZx\_RGB\_TYP[0] TYP bit (0 for ARGB32 format and 1 for RGB5:6:5).

To handle the horizontal border conditions, the leftmost Chroma sample or the right-most Chroma sample is duplicated on the left or the right.

Software must make it possible to remove 2 pixels on the left and/or right to take care of the issues that Chroma duplication introduces on the borders. The RZx\_RGB\_TYP[1] MSK0 and RZx\_RGB\_TYP[2] MSK1 bits control this feature.

**Figure 8-168. ISS ISP RSZ Chroma Position and Upsampling**



**8.3.3.5.5.7 ISS ISP RSZ Output Interface**

The output interface receives the data generated by the two resizer engines and generates the addresses and the port requests to the BL module.

- The port 1 interface is dedicated to the RSZ-A module. If the RSZ module is set up in pass-through mode, then the data is output on the port 1 interface. This interface can transfer RAW, YUV4:2:2, YUV4:2:0 and RGB data
- The port 2 interface is dedicated to the RSZ-B module. This interface can transfer RAW, YUV4:2:2, YUV4:2:0 and RGB data.

The YUV4:2:0 data format is handled differently from the other formats because the output data are written at two different memory locations: luminance in one buffer and chrominance in a second buffer. For all other formats the data are written in the same buffer.

Each data format must be stored in memory in a dedicated manner, which is summarized in [Table 8-323](#).

**Table 8-323. ISS ISP RSZ Output Interface: Data Formats**

Output Format	Bytes per Pixel	Output Buffers per Image	Interface Supporting the Data Format
RAW	2	1	MTC port 1
YUV4:2:2	2 average	1	MTC port 1 port 2
YUV4:2:0	1.5 average	2	MTC port 1 port 2
RGB16	2	1	MTC port 1 port 2
ARGB32	4	1	MTC port 1 port 2

**8.3.3.5.5.7.1 ISS ISP RSZ Circular Buffer**

Figure 8-169 shows the parameters that are required to set up the circular buffers. As mentioned previously, there can be up to four circular buffers in case the two resizer engines are outputting YUV4:2:0 data.

The circular buffer management requires the following parameters (REZ-A or RSZ-B A or B, chrominance or luminance Y or C, low or high part of the address, L or H. Sets the base address of the circular buffer):

- Baseline address (BAD, in registers, where x is the resizer A or B, i is the
- Start address (SAD, in RZx\_SDR\_i\_SAD\_j registers)
- Start pointer (PTR\_S, in RZx\_SDR\_i\_PTR\_S registers)
- End pointer (PTR\_E, in RZx\_SDR\_i\_PTR\_E registers)
- Line offset (OFT, in RZx\_SDR\_i\_OFT registers)

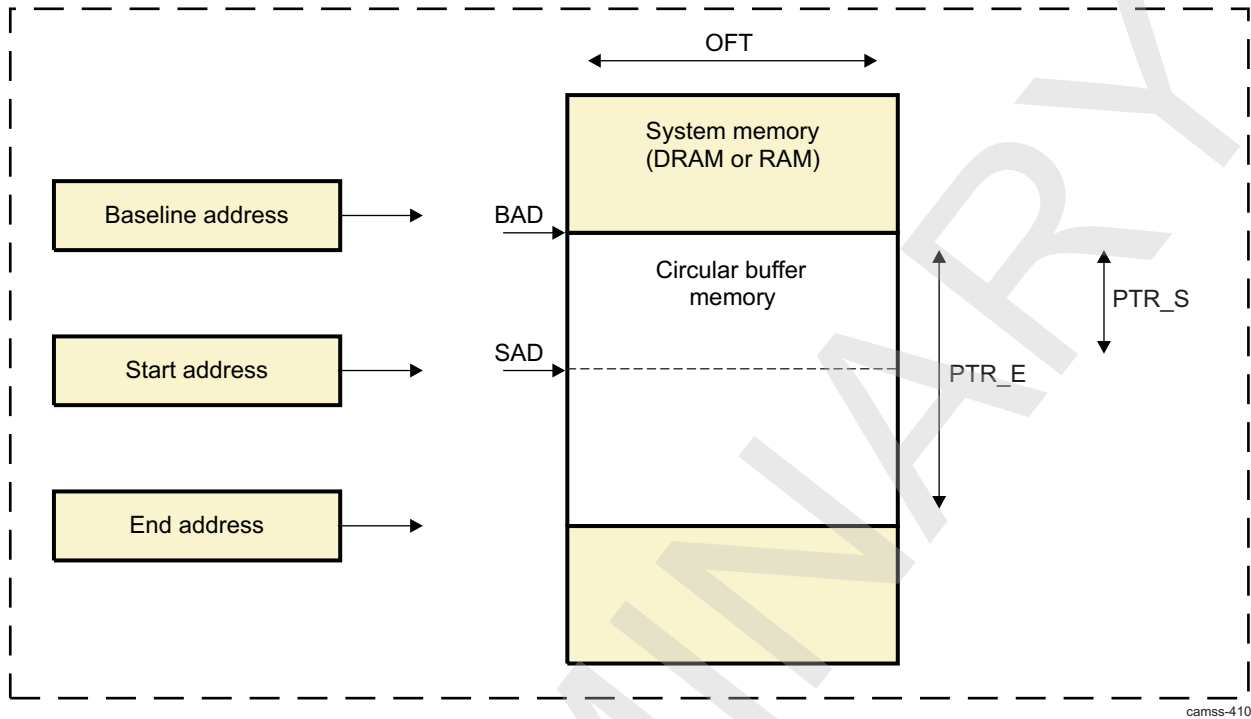
Circular Buffer Parameter	Register (for RSZ-A and RSZ-B)	Description
Baseline address	RZA_SDR_Y_BAD_H RZA_SDR_Y_BAD_L RZA_SDR_C_BAD_H RZA_SDR_C_BAD_L RZB_SDR_Y_BAD_H RZB_SDR_Y_BAD_L RZB_SDR_C_BAD_H RZB_SDR_C_BAD_L	Sets the base address of the circular buffer
Start address	RZA_SDR_Y_SAD_H RZA_SDR_Y_SAD_L RZA_SDR_C_SAD_H RZA_SDR_C_SAD_L RZB_SDR_Y_SAD_H RZB_SDR_Y_SAD_L RZB_SDR_C_SAD_H RZB_SDR_C_SAD_L	Sets the start address of the circular buffer. The first data output is written to this address. If the first line of a frame must be written at the beginning of the circular buffer memory, then SAD = BAD and PTR_S = 0.
Start pointer	RZA_SDR_Y_PTR_S RZA_SDR_C_PTR_S RZB_SDR_Y_PTR_S RZB_SDR_C_PTR_S	Sets the initial value of the circular buffer internal counter. It must be set up as PTR_S = (SAD - BAD)/OFT. PTR_S is expressed in the number of lines.
End pointer	RZA_SDR_Y_PTR_E RZA_SDR_C_PTR_E RZB_SDR_Y_PTR_E RZB_SDR_C_PTR_E	Sets the size of the circular buffer. PTR_E is expressed in the number of lines. The circular buffer can contain up to PTR_E lines.
Line offset	RZA_SDR_Y_OFT RZA_SDR_C_OFT RZB_SDR_Y_OFT RZB_SDR_C_OFT	This is the offset expressed in bytes between two lines in the circular buffer. Here: Line 0 = SAD, Line 1 = SAD + 1 x OFT, Line 2 = SAD + 2 x OFT, etc. OFT does not necessarily correspond to the size of a line in a frame; it can be bigger.

More generally, the following equations hold:

- $SAD = BAD + (PTR\_S \times OFT)$  and  $PTR\_S < PTR\_E$

Interrupts can be triggered every time a certain number of lines are written to the circular buffer. There are independent settings for each resizer and for each possible output of each resizer.

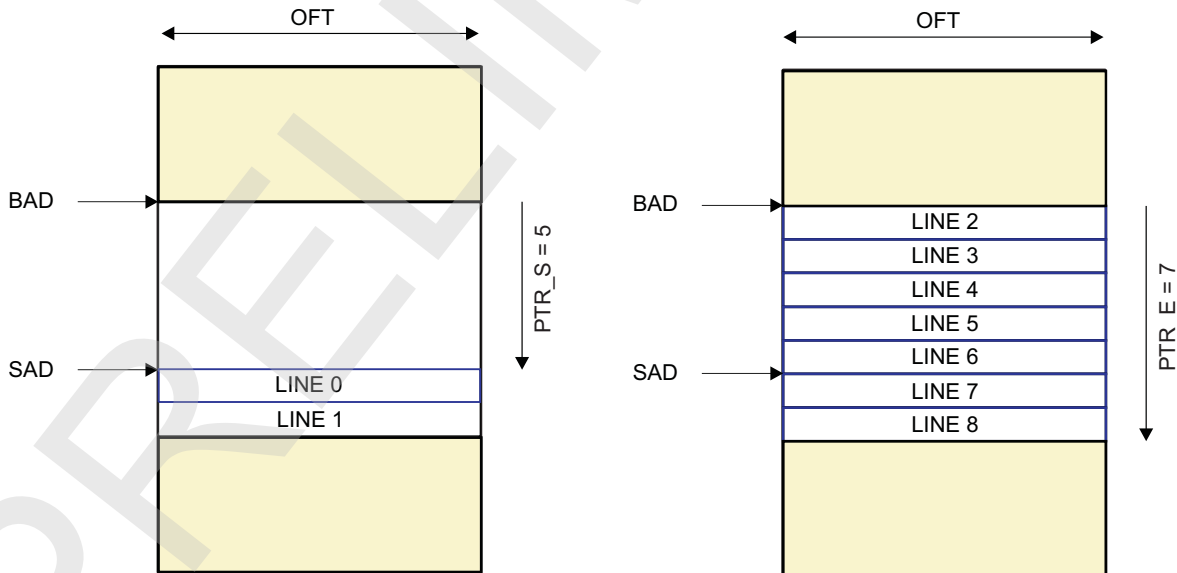
Figure 8-169. ISS ISP RSZ and Circular Buffer Settings



camss-410

Figure 8-170 shows how the data are stored in the circular buffer over time when vertical flip is disabled. In this example, PTR\_S = 5 and PTR\_E = 7. There can be up to PTR\_E = 7 lines in the circular buffer.

Figure 8-170. ISS ISP RSZ and Circular Buffer Settings – Example 1



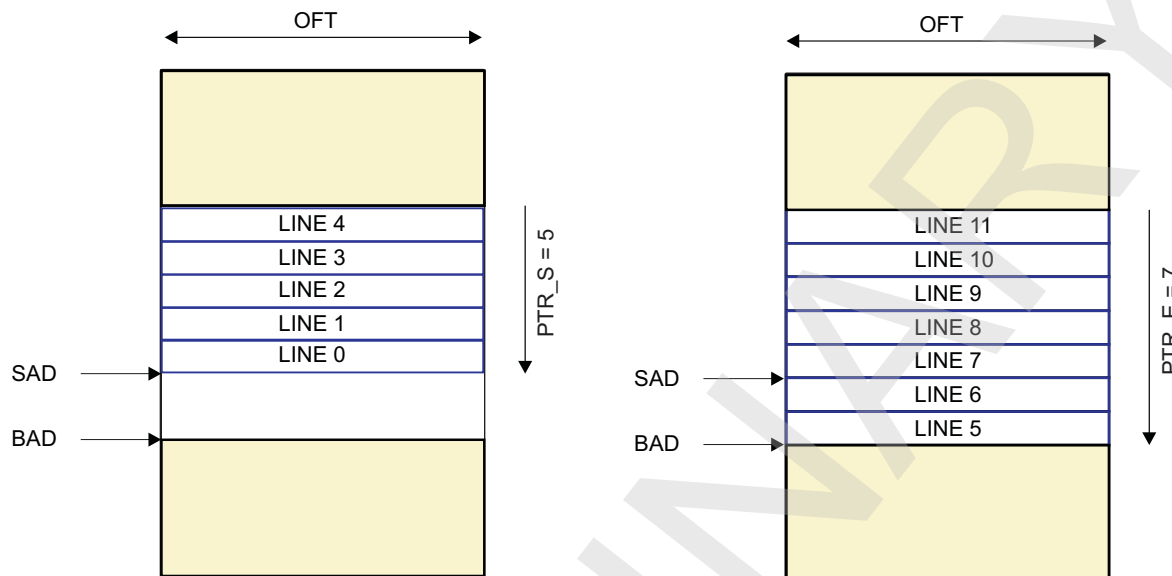
Vertical flip enabled:

1. Start from SAD.
2. Output PTR\_E – PTR\_S lines.
3. Wrap to BAD.
4. Output PTR\_E lines and continue wrapping to BAD.

camss-412

Figure 8-171 shows how the data are stored in the circular buffer over time when vertical flip is enabled. In this example, PTR\_S = 5 and PTR\_E = 7. There can be up to PTR\_E = 7 lines in the circular buffer.

**Figure 8-171. ISS ISP RSZ and Circular Buffer Settings – Example 2**



Vertical flip enabled:

1. Start from SAD.
2. Output PTR\_E – PTR\_S lines.
3. Wrap to BAD.
4. Output PTR\_E lines and continue wrapping to BAD.

camss-413

### 8.3.3.6 ISS ISP H3A Functional Description

#### 8.3.3.6.1 ISS ISP H3A Overview

The H3A module supports the control loops for autofocus, auto white balance, and auto exposure by collecting metrics about the imaging/video data. The metrics are used to adjust parameters for processing the imaging/video data. There are two main blocks in the H3A module:

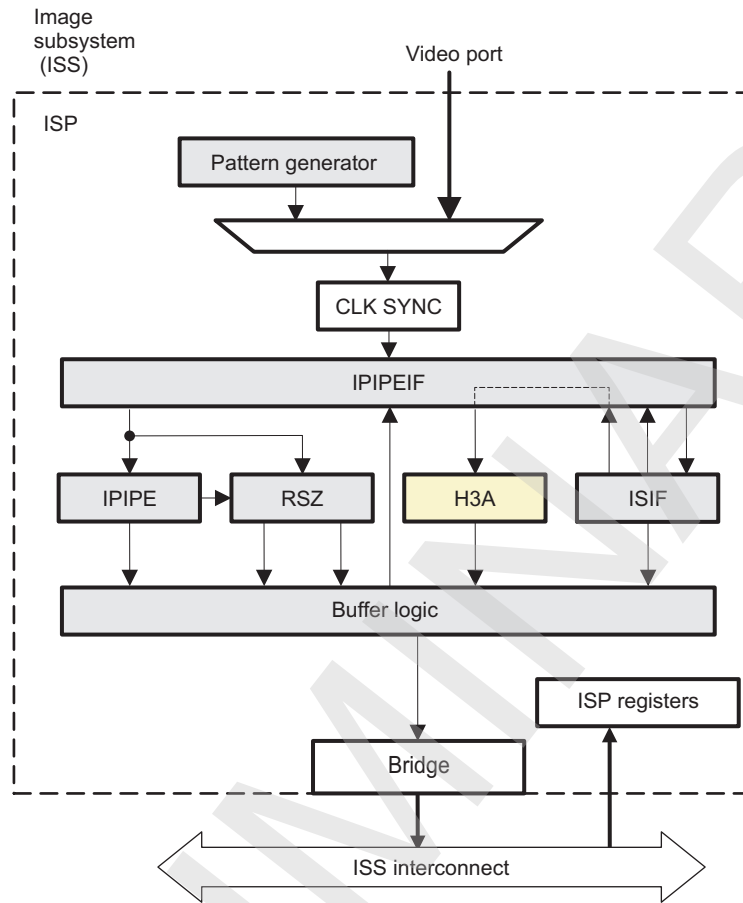
- Autofocus (AF) engine:
  - The AF submodule extracts and filters the red, green, and blue data from input image data and provides the accumulation or peaks of the data in a specified region. The specified region is a 2D block of data referred to as a paxel. The AF engine supports the following features:
    - Peak mode in a paxel
      - Accumulation of the maximum focus value (FV) of each line in a paxel
    - Accumulation mode in a paxel
    - Accumulation of horizontal and vertical focus value in a paxel
    - Up to 12 paxels in the horizontal direction and up to 12 paxels in the vertical direction with vertical focus
    - Up to 36 paxels in the horizontal direction and up to 128 paxels in the vertical direction with horizontal focus only
    - Programmable width and height for the paxel/window
    - Programmable red, green, and blue position within a 2 × 2 matrix
    - Separate horizontal start for paxel and filtering
    - Programmable vertical and horizontal line increments within a paxel

- Horizontal FV uses parallel infinite impulse response (IIR) filters configured in a dual-biquad configuration with individual coefficients (two filters with 11 coefficients each). The filters are intended to compute the sharpness/peaks in the frame to focus on.
- Vertical FV uses a 5-tap FIR filter with 8-bit coefficients. With horizontal steps each paxel has up to 32 columns to be maintained for vertical FV calculation.
- Auto exposure and auto white balance (AE/AWB) engine:
 

The AE/AWB engine accumulates values and checks for saturated values in a subsampling of the video data. In the case of the AE/AWB, the 2D block of data is referred to as a window. Thus, other than having different names, paxels and windows are essentially the same. However, the numbers, dimensions, and starting positions of AF paxels and AE/AWB windows are programmable separately. AE/AWB supports the following features:

  - Accumulate clipped pixels along with all nonsaturated pixels in each window per color
  - Accumulate the sum of squared pixels in each window per color
  - Minimum and maximum pixel values in each window per color
  - Supports for up to 36 horizontal windows with sum + { sum\_sq or min+max} output
  - Support for up to 56 horizontal windows with sum output
  - Support for up to 128 vertical windows
  - Programmable width and height for the windows. All windows in the frame are the same size.
  - Separate vertical start coordinate and height for a black row of paxels that is different than the remaining color paxels
  - Programmable horizontal sampling points in a window
  - Programmable vertical sampling points in a window
- Maximum pixel throughput of 200 MPix/s
- Double-buffer for paxel/window accumulation
- H3A data path is 10 bits.
- Maximum input size is 3008 pixels.

Figure 8-172 shows the H3A module connections to other submodules of the ISP.

**Figure 8-172. ISS ISP H3A High-Level Diagram**

camss-003b

**8.3.3.6.2 ISS ISP H3A Top-Level Block Diagram**

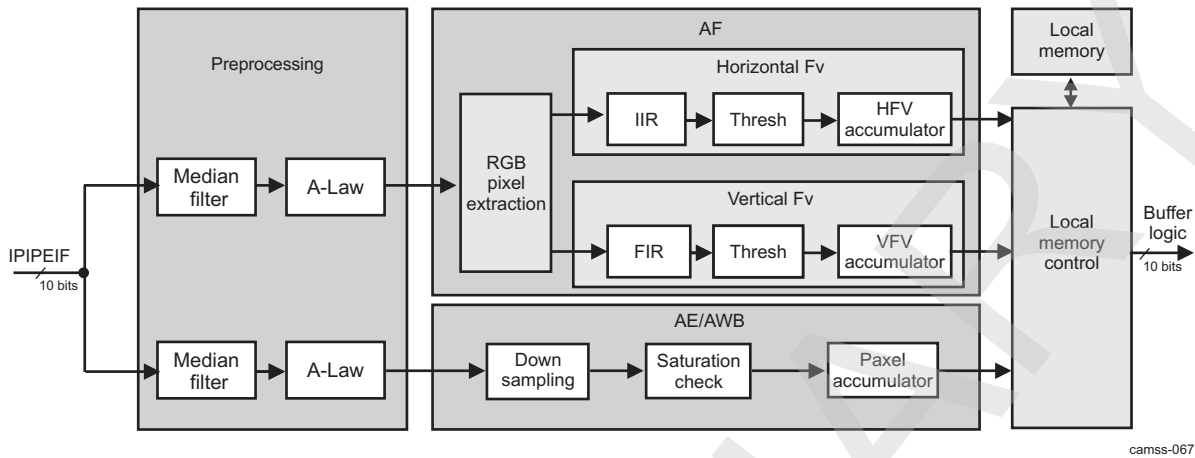
The block diagram in [Figure 8-173](#) shows the process of the AF and AE/AWB data paths through the H3A module.

The data flow before H3A is:

1. Data comes from the VP (VP) or BL.
2. The data is processed by the ISIF.
3. The data is processed by the IPIPEIF.
4. The data is 10 bits from the IPIPEIF at H3A input.



Figure 8-173. ISS ISP H3A Top-Level Block Diagram

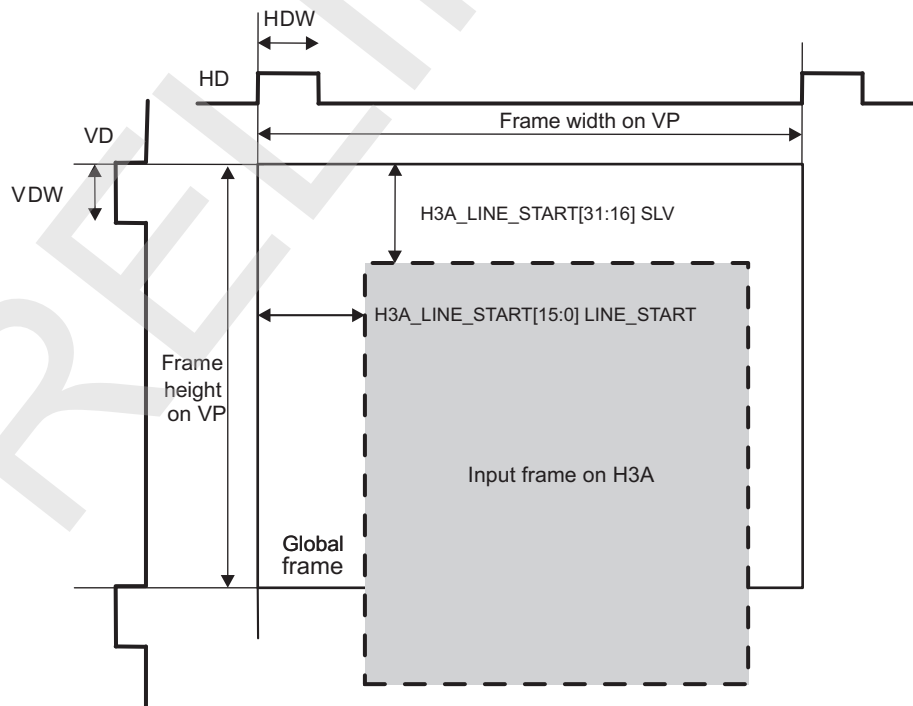


camss-067

8.3.3.6.3 ISS ISP H3A Line Framing Logic

In certain cases the number of clock cycles between HD pulses is greater than the line buffer included in the H3A. To solve this problem a framing module was added before the line buffer. The framing module uses the [H3A\\_LINE\\_START](#) register to find the position of the first pixel to place into the line buffer. All other registers reference this point as the 0 pixel for their start positions. The line size is 3008 pixels. After 3008 clock cycles the framing logic disables the line buffer and waits until the next HD. If the next HD comes before 3008 clock cycles, then the active region ends immediately and the counter waits for the [H3A\\_LINE\\_START](#) register count to be reached again. For the vertical position the [H3A\\_LINE\\_START\[31:16\]](#) SLV bit field can be used to determine where the start point of the frame is relative to the rising edge of VD. This logic allows an active frame to cross VD boundaries and remain in the same frame.

Figure 8-174. ISS ISP H3A Frame Format Settings



camss-073

---

**NOTE:** (Frame width on VP) - (H3A\_LINE\_START[15:0] LINE\_START) must be less than or equal to 3008, because the H3A memory lines are limited to 3008 pixels.

---

#### 8.3.3.6.4 ISS ISP H3A Optional Preprocessing

The input to the H3A module is 10-bit RAW data from the IPIPEIF. A 10-bit to 8-bit A-Law compression step can be enabled and disabled separately for the AF engine (the H3A\_PCR[1] AF\_ALAW\_EN bit) and the AE/AWB engine (the H3A\_PCR[17] AEW\_ALAW\_EN bit). A-Law compression offers added protection against overflowing the accumulators.

If the A-Law table is enabled, the output is 10 bits, with the upper two bits filled with 0.

For the AF process, a horizontal median filter can be enabled and disabled (the H3A\_PCR[2] AF\_MED\_EN bit) before A-Law compression. This filter is useful for reducing temperature-induced noise. The horizontal median filter calculates the absolute difference between the current pixel (i) and pixel (i - 2), and between the current pixel (i) and pixel (i + 2). If the absolute difference exceeds a threshold, and the sign of the differences is the same, the average of pixel (i - 2) and pixel (i + 2) replaces pixel (i). The threshold of the horizontal median filter can be set in the H3A\_PCR[10:3] MED\_TH bit field.

#### 8.3.3.6.5 ISS ISP H3A Autofocus Engine

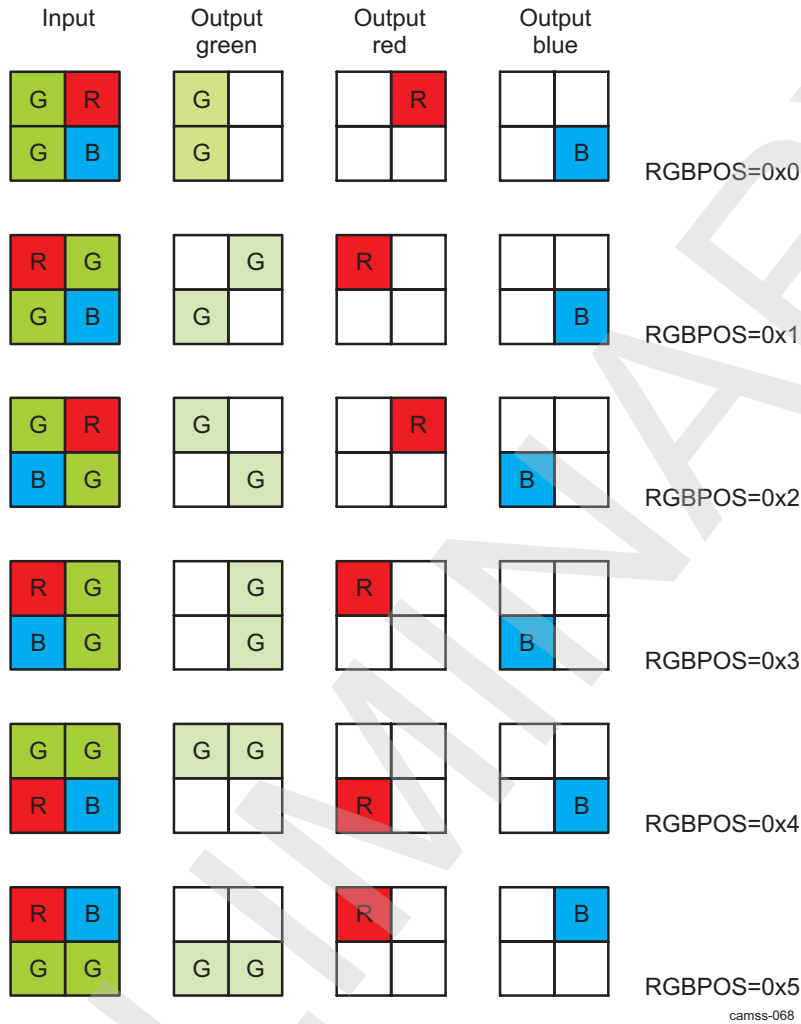
The AF engine works by extracting each green (Gr or Gb) pixel from the video stream and subtracts a fixed offset of 128 or 512 (depending on whether A-Law is enabled or disabled) from the pixel value. The offset value is then passed through an IIR filter and the absolute value of the filter output is the focus value (FV). Both FV and FV<sup>2</sup> are produced. The FV and FV<sup>2</sup> values can be accumulated or the maximum for each line/column can be accumulated. The following sections describe this process in more detail.

##### 8.3.3.6.5.1 ISS ISP H3A Poxel Extraction

From the poxel starting coordinate (the H3A\_AFPAXSTART[27:16] PAXSH and H3A\_AFPAXSTART[11:0] PAXSV bit fields) specifies the starting point of the poxel grid, with respect to first pixel of the input image frame.

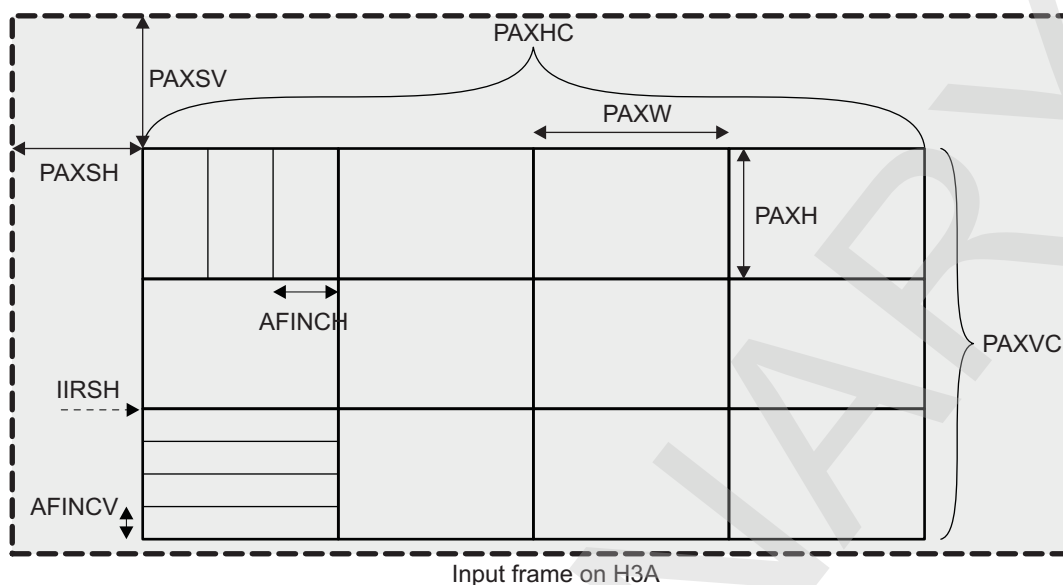
The poxel starting coordinate also indicates which color pixels are extracted if VF is enabled (that is, if H3A\_PCR[20] AF\_VF\_EN = 1). Normally, either Gr or Gb is used for AF, but it is not important to the hardware whether it is red, green, or blue. If VF is not enabled, then the red, green, and blue pixel extraction is controlled by the H3A\_PCR [13:11] RGBPOS bit field to extract the correct colors from the input stream. Figure 8-176 shows the available options for this bit field. The red and blue pixel positions are interchangeable. For each 2 × 2 grid, the green pixels are summed to create a single value. Because of this, the amplitude of the green output contains 2 pixels, while the red and blue outputs each contain 1 pixel.

Figure 8-175. ISS ISP H3A Red, Green, and Blue Pixel Extraction Examples



Each paxel is  $H3A\_AFPAX1[23:16]$  PAXW  $\times$   $H3A\_AFPAX1[7:0]$  PAXH (width  $\times$  height) pixels. Inside each paxel, horizontal FV can skip lines, operating on one every  $H3A\_AFPAX2[16:13]$  AFINCV lines. Vertical FV can skip columns, operating on one every  $H3A\_AFPAX2[20:17]$  AFINCH columns. Up to 32 columns are supported for each paxel. If floor (PAXW/AFINCH)  $\geq$  32, only the first 32 designated columns are operated on. Because PAXW, PAXH, AFINCV, and AFINCH are all even numbers, AF always operates on the same green color, Gr or Gb. IIR filters for the horizontal FVs start operation at column  $H3A\_AFIIRSH[11:0]$  IIRSH.

Figure 8-176. ISS ISP H3A Horizontal/Vertical FV Poxel Configuration



camss-071

**NOTE:**  $(H3A\_AFPAXSTART[27:16] PAXSH) + (H3A\_AFPAX2[5:0] PAXHC) \times (H3A\_AFPAX1[23:16] PAXW) \leq [(Frame\ width\ on\ VP) - (H3A\_LINE\_START[15:0] LINE\_START)] \leq 3008$

Table 8-324 lists the register fields that configure the size and number of pixels.

Table 8-324. ISS ISP H3A Poxel Register Field Descriptions

Register Field	Bit Width	Description
<a href="#">H3A_AFPAX1[23:16]</a> PAXW	8	Poxel width (in pixels)
<a href="#">H3A_AFPAX1[7:0]</a> PAXH	8	Poxel height (in lines)
<a href="#">H3A_AFPAX2[5:0]</a> PAXHC	6	Poxel count for horizontal direction
<a href="#">H3A_AFPAX2[12:6]</a> PAXVC	7	Poxel count for vertical direction
<a href="#">H3A_AFPAX2[16:13]</a> AFINCV	4	Line increments in a poxel
<a href="#">H3A_AFPAX2[20:17]</a> AFINCH	4	Column increments in a poxel
<a href="#">H3A_AFPAXSTART[27:16]</a> PAXSH	12	Poxel start position H
<a href="#">H3A_AFPAXSTART[11:0]</a> PAXSV	12	Poxel start position V
<a href="#">H3A_AFIIRSH[11:0]</a> IIRSH	12	IIR filter start position

The H3A AF engine also has an option for an advanced or normal stats collection mode. When 0xCA00 is written to the [H3A\\_ADVANCED\[31:15\]](#) ID bit field, then [H3A\\_ADVANCED\[0\]](#) AF\_MODE can be used to toggle between normal and advanced AF stats collection mode. When the advanced AF stats collection mode is enabled, the ZEROS section of the AF poxel packet is filled with the sum of the maximum FVs, regardless of the color, from HFV\_1 and HFV\_2.

### 8.3.3.6.5.2 ISS ISP H3A Horizontal FV Calculator

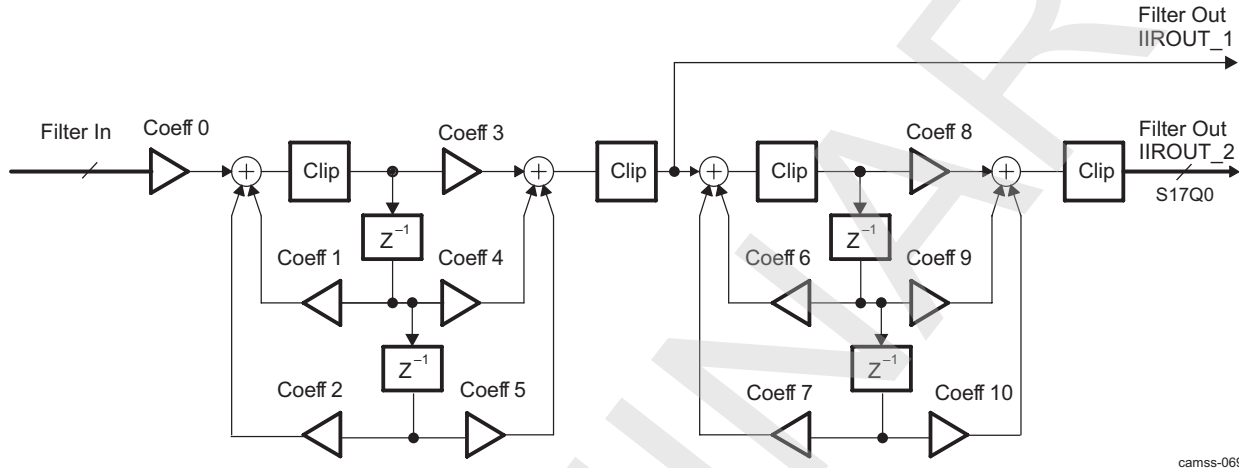
The FV calculator takes the unsigned red/green/blue extracted data and subtracts 128 or 512 (depending on whether A-Law is enabled) to place the data in the range  $[-128:127]$  or  $[-512:511]$ .

After removing the offset, the data is sent through two parallel IIR filters configured in a dual-biquad configuration. Each filter uses a unique set of 11 programmable coefficients. Each coefficient is 12-bits-wide with 6 bits of decimal, S12Q6 ([H3A\\_AFCOEF010](#) to [H3A\\_AFCOEF0010](#) for SET0, and [H3A\\_AFCOEF110](#) to [H3A\\_AFCOEF1010](#) for SET1). The filter-shift registers are cleared on each

horizontal line at the position set by the register IIR horizontal start register (the [H3A\\_AFIIRSH](#) [11:0] IIRSH bit field). The absolute values of the output (16-bits-wide with 4 bits of decimal, U16Q4) of both filters are then sent to the AF accumulator module. Signed clipping is performed during the FV calculation. If the input value is  $m$  bits (signed) and the required output value is  $n$  bits, clipping transforms the input to between  $-2^1$  and  $2^1$ . Values lower than  $-2^1$  are set to  $-2^1$ , and values higher than  $2^1$  are set to  $2^1$ .

Figure 8-177 shows the IIR filter model.

Figure 8-177. ISS ISP H3A IIR Filter Model



### 8.3.3.6.5.3 ISS ISP H3A HFV Accumulator

The horizontal focus value (HFV) accumulator takes the output of the horizontal IIR filter and accumulates values for each paxel. The size and number of paxels is configurable by registers.

Table 8-324 lists the register fields that configure the size and number of paxels:

- In peak mode ([H3A\\_PCR](#)[14] FVMODE = 0x1), the maximum value is accumulated.
- In sum mode ([H3A\\_PCR](#)[14] FVMODE = 0x0), all HFV<sub>n</sub> are accumulated in a paxel.

The following equations detail the calculation for:

- Sum of pixel values used in HFV: The pixel values that are used for filtering and accumulation of HFV are also accumulated in this sum of pixel values.
- HFV<sub>n</sub> (HFV<sub>n\_peak</sub> for peak mode or HFV<sub>n\_sum</sub> for sum mode)
- HFV\_count<sub>n</sub>
- HFV\_sq\_n (HFV\_sq\_n\_peak for peak mode or HFV\_sq\_n\_sum for sum mode)

$n = 1$  or  $2$  for IIR1 and IIR2, respectively.

For each paxel, these six values are available for each R, G, and B component.

```

for (k=0; k< PAXH) // Loop on paxel rows
{
    rowpeak_n = 0;
    for (l=0; l<PAXW; l++) // Loop on values within a row
    {
        aIIRout_n = ABS(IIRout_n);
        if (aIIRout_n >= threshold_n)
        {
            hfval = aIIRout_n - threshold_n;
            HFV_count_n++;
        }
        else hfval = 0;
        if (hfval > rowpeak_n)
        {
            rowpeak_n = HFV_n;
        }
        HFV_n_sum += hfval;
        HFV_sq_n_sum += (hfval* hfval + RNDADD) >> RNDSHIFT;
    }
}
    
```

```

} // Finished looping on values in a row
HFV_n_peak += rowpeak_n;
HFV_sq_n_peak += (rowpeak_n * rowpeak_n + RNDADD) >> RNDSHIFT;
}

```

- threshold\_n is [H3A\\_HVF\\_THR\[15:0\]](#) HTHR1 and [H3A\\_HVF\\_THR\[31:16\]](#) HTHR2, respectively.
- IIRout\_n is the IIRout\_1 and IIRout\_2 outputs, respectively.
- HFV\_count\_n and HFV\_sq\_n are not sent to the DMA interface if VF is disabled.
- RNDADD and RNDSHIFT depend on whether input pixels are 8-bit or 10-bit, and achieves rounding. This is automatically performed by the module.
- If VF is enabled, only the green color channel values are output to the DMA interface.
- In sum mode:
  - HFV\_n = HFV\_n\_sum
  - HFV\_sq\_n = HFV\_sq\_n\_sum
- In peak mode:
  - HFV\_n = HFV\_n\_peak
  - HFV\_sq\_n = HFV\_sq\_n\_peak

#### 8.3.3.6.5.4 ISS ISP H3A VFV Calculator

The VFV calculator takes the unsigned extracted data through two FIR filters, each with a set of five coefficients (VCOEF1\_x, where x = 0..4, in the [H3A\\_VFV\\_CFG1](#) and [H3A\\_VFV\\_CFG2](#) registers for FIR 1, and VCOEF2\_x, where x = 0..4, in [H3A\\_VFV\\_CFG3](#) and [H3A\\_VFV\\_CFG4](#) registers). Each coefficient is 8 bits wide with 4 bits of decimal (S8Q4). The filter outcome is downshifted by 4 bits and taken absolute value to produce a 16-bit unsigned value. This is then sent to threshold [H3A\\_VFV\\_CFG2\[31:16\]](#) VTHR1 for FIR 1, and [H3A\\_VFV\\_CFG4\[31:16\]](#) VTHR2 for FIR 2, and square logic to produce VFV\_n and VFV\_sq\_n.

#### 8.3.3.6.5.5 ISS ISP H3A VFV Accumulator

The VFV accumulator takes the output of the vertical FIR filters and accumulates values for each pixel. The size and number of pixels is configurable by registers.

[Table 8-324](#) lists the register fields that configure the size and number of pixels.

The following equations detail the calculation for:

- VFV\_n
- VFV\_count\_n
- VFV\_sq\_n

n = 1 or 2 for FIR1 and FIR2, respectively.

For each pixel, these six values are available for each R, G, and B component.

```

FIR_coef_n = [VCOEFn_0, VCOEFn_1, VCOEFn_2, VCOEFn_3, VCOEFn_4]; /* coefficient values in S8.4
format */
aFIRout_n = (ABS(inner_product(extracted_G, FIR_coef_n)) + 8) >> 4;
if (aFIRout_n >= threshold_n)
{
  VFV_n = aFIRout_n - threshold_n
  VFV_count_n++;
}
else VFV_n = 0;
VFV_sq_n = (VFV_n * VFV_n + RNDADD) >> RNDSHIFT;

```

- threshold\_n is [H3A\\_VFV\\_CFG2\[31:16\]](#) VTHR1 and [H3A\\_VFV\\_CFG4\[31:16\]](#) VTHR2, respectively.
- FIRout\_n is the FIRout\_1 and FIRout\_2 outputs, respectively.
- RNDADD and RNDSHIFT depend on whether the input pixels are 8-bit or 10-bit, and achieves rounding. This is automatically performed by the module.

### 8.3.3.6.6 ISS ISP H3A AE/AWB Engine

The AE/AWB engine starts by dividing the frames into windows, and then subsamples each window into 2 x 2 blocks. For each subsampled 2 x 2 block, each pixel is accumulated. Also, each pixel is compared to a limit set in a register. If any pixels in a 2 x 2 block are greater than or equal to the limit, the block is not counted in the unsaturated block counter. Pixels greater than the limit are replaced by the limit, and the value of the pixel is accumulated.

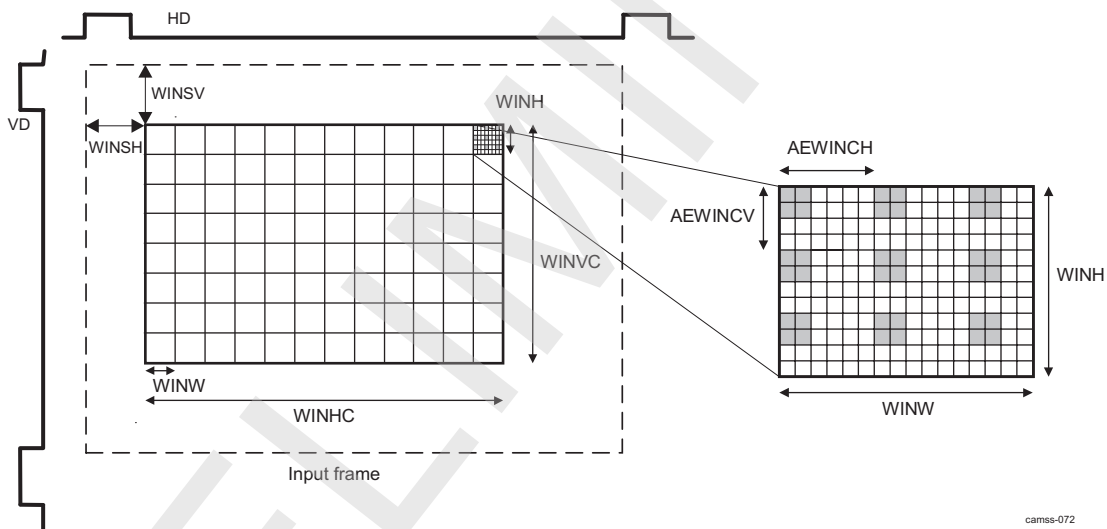
The AE/AWB module has three output format modes, which are set through the `H3A_AEWCFG[9:8]` AEFMT bit field:

- Sum of square mode: `H3A_AEWCFG[9:8]` AEFMT = 0x0
- Min/max mode: `H3A_AEWCFG[9:8]` AEFMT = 0x1
- Sum-only mode: `H3A_AEWCFG[9:8]` AEFMT = 0x2

#### 8.3.3.6.6.1 ISS ISP H3A Sub\_sampler

The subsampler partitions the frame into windows using the size, count, and starting location parameters shown on the left in [Figure 8-174](#). Each window is further sampled down to a set of 2 x 2 blocks. The horizontal and vertical distances between the start of blocks within a window is programmable using the parameters shown on the right in [Figure 8-174](#).

**Figure 8-178. ISS ISP H3A AE/AWB Window Configurations**



[Table 8-325](#) lists the register fields that configure the window and block sizes, counts, and starting positions.

**Table 8-325. ISS ISP H3A AE/AWB Window Register Field Descriptions**

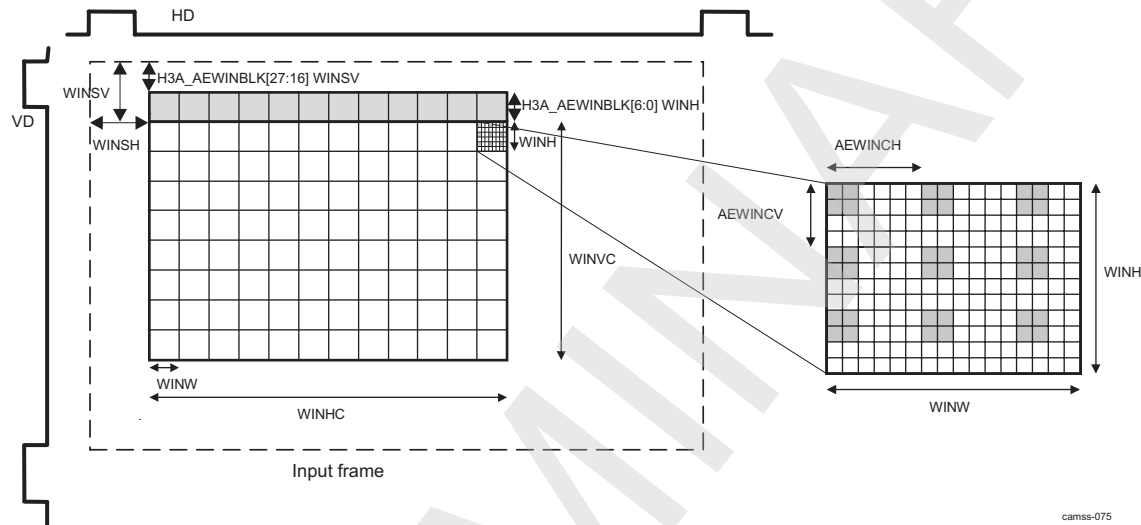
Register Field	Bit Width	Description
<code>H3A_AEWWIN1[20:13]</code> WINW	7	Window width (in pixels)
<code>H3A_AEWWIN1[31:24]</code> WINH	7	Window height (in lines)
<code>H3A_AEWWIN1[5:0]</code> WINHC	6	Window count for horizontal direction
<code>H3A_AEWWIN1[12:6]</code> WINVC	7	Window count for vertical direction
<code>H3A_AEWINSTART[11:0]</code> WINSH	12	Window start position H
<code>H3A_AEWINSTART[27:16]</code> WINSV	12	Window start position V
<code>H3A_AEWSUBWIN[3:0]</code> AEWINCH	4	Horizontal distance between subsamples
<code>H3A_AEWSUBWIN[11:8]</code> AEWINCV	4	Vertical distance between subsamples

### 8.3.3.6.6.2 ISS ISP H3A Additional Black Row of AE/AWB Windows

In addition to the 128 rows of windows, the AE/AWB module provides support for an additional row of windows for black data. This data may be useful in determining the DC offset noise of the rest of the data. The black row of windows can be before or after the regular rows of windows. The vertical start line for the black row of windows is specified in the `H3A_AEWINBLK[27:16]` `WINSV` bit field, and the height is specified in the `H3A_AEWINBLK[6:0]` `WINH` bit field. The horizontal starting pixel and horizontal width of the black row of windows are the same as for the regular rows of windows.

Figure 8-179 shows a black row of windows before rows of windows.

**Figure 8-179. ISS ISP H3A Black Row of Windows Before Regular Rows of Windows**



**Table 8-326. ISS ISP H3A AE/AWB Window with Additional Black Row Register Field Descriptions**

Register Field	Bit Width	Description
<code>H3A_AEWWIN1[20:13]</code> WINW	7	Window width (in pixels)
<code>H3A_AEWWIN1[31:24]</code> WINH	7	Window height (in lines)
<code>H3A_AEWWIN1[5:0]</code> WINHC	6	Window count for horizontal direction
<code>H3A_AEWWIN1[12:6]</code> WINVC	7	Window count for vertical direction
<code>H3A_AEWINSTART[11:0]</code> WINSH	12	Window start position H
<code>H3A_AEWINSTART[27:16]</code> WINSV	12	Window start position V
<code>H3A_AEWSUBWIN[3:0]</code> AEWINCH	4	Horizontal distance between subsamples
<code>H3A_AEWSUBWIN[11:8]</code> AEWINCV	4	Vertical distance between subsamples
<code>H3A_AEWINBLK[27:16]</code> WINSV	12	Window start position H for single black line
<code>H3A_AEWINBLK[6:0]</code> WINH	7	Window height (in lines) for single black line

### 8.3.3.6.6.3 ISS ISP H3A Saturation Check

The saturation check module compares the data from the subsampler to the value programmed in the `H3A_PCR [31:22]` `AVE2LMT` bit field. This value is the maximum clipping value. If all 4 pixels in the  $2 \times 2$  block are less than the `AVE2LMT` value, the value of the unsaturated block counter is incremented. There is one unsaturated block counter per window. The unsaturated block counters are later written to memory.

### 8.3.3.6.6.4 ISS ISP H3A AE/AWB Accumulators

The output from the saturation check module and the subsampler module are separately accumulated for each pixel in every  $2 \times 2$  pixel block for each window. Therefore, there are eight accumulators per window (one accumulator for each pixel in a  $2 \times 2$  pixel block, times two sets of accumulators: clipped/saturated data and presaturated data). Each of the 4 pixels in the  $2 \times 2$  pixel grid is associated with a color (R, Gr, B, Gb); however, the output of these accumulators is referenced by position in the grid, not color.



The accumulators are 16 bits wide, and the accumulated data is 10 bits wide. Therefore, when a window contains more than 64 pixels of the same color, an overflow risk exists. This risk can be reduced by enabling the A-Law conversion in the preprocessing stage. See [Section 8.3.3.6.4](#) for details.

The AE/AWB module has a shift value for the accumulation of pixel values that is set in the [H3A\\_AEWCFG\[3:0\]](#) SUMSHFT bit field.

**8.3.3.6.7 ISS ISP H3A DMA Interface**

The DMA interface module takes the data from the AF engine and AE/AWB engine and builds packets to be sent out to the memory through the BL module.

The data interface has separate start pointers for the AF and AE/AWB engines.

- The starting address for the AF engine is the [H3A\\_AFBUFST\[31:5\]](#) AFBUFST bit field.
- The starting address for the AE/AWB engine is the [H3A\\_AEWBUFST\[31:5\]](#) AEWBUFST bit field.

The DMA interface module continuously loops through this data as it builds the packets. To optimize the transfer sizes, the DMA interface sends out an AF or AE transfer for each row of pixels or windows. This requires that each horizontal row of pixels or windows starts and ends on a 32-byte boundary. If a horizontal row of pixels or windows ends on a non-32 byte boundary, the hardware packs zeroes. The counts for the AEW that occur every eight windows is sent in the row with the 8<sup>th</sup> consecutive window.

[Table 8-327](#) lists the packet formats for AF with vertical AF disabled.

**Table 8-327. ISS ISP H3A AF Packet Format With Vertical AF Disabled**

Buffer Start Address (Byte Address) <a href="#">H3A_AFBUFST</a>	31	16 15	0
Sum of pixel values used in HFV			(Paxel 0)
HFV_1 (peak or sum)			(Paxel 0)
HFV_2 (peak or sum)			(Paxel 0)
ZEROES			(Paxel 0)
Sum of pixel values used in HFV			(Paxel 0)
HFV_1 (peak or sum)			(Paxel 0)
HFV_2 (peak or sum)			(Paxel 0)
ZEROES			(Paxel 0)
Sum of pixel values used in HFV			(Paxel 0)
HFV_1 (peak or sum)			(Paxel 0)
HFV_2 (peak or sum)			(Paxel 0)
ZEROES			(Paxel 0)
Sum of pixel values used in HFV			(Paxel 1)
HFV_1 (peak or sum)			(Paxel 1)
HFV_2 (peak or sum)			(Paxel 1)
ZEROES			(Paxel 1)
...			

[Table 8-328](#) shows the packet formats for AF with vertical AF enabled.

**Table 8-328. ISS ISP H3A AF Packet Format With Vertical AF Enabled**

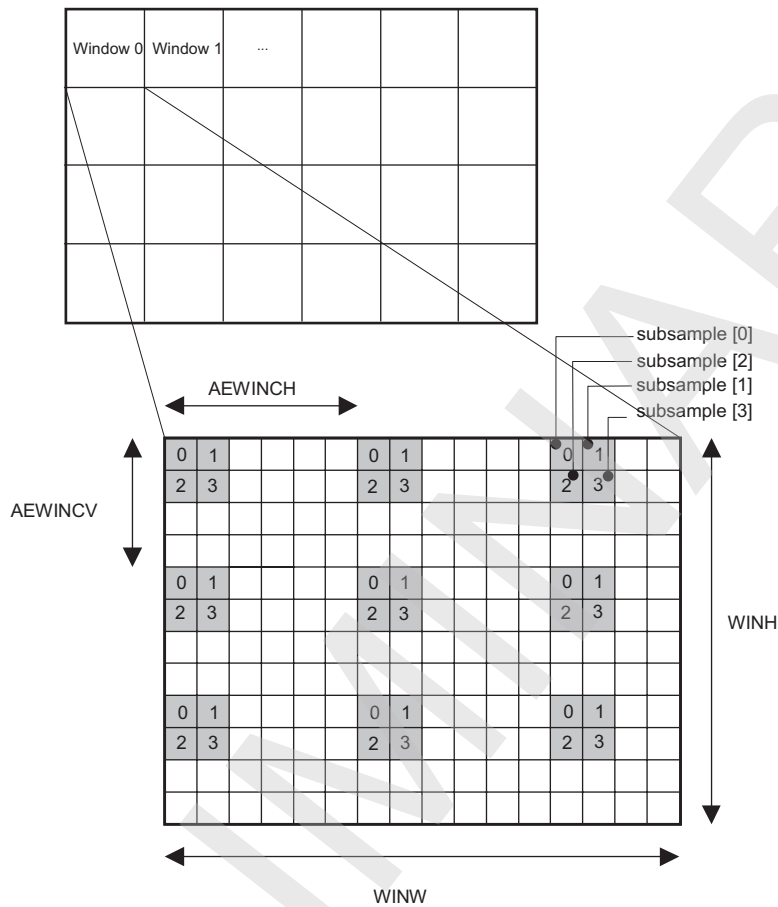
Buffer Start Address (Byte Address) <a href="#">H3A_AFBUFST</a>	31	16 15	0
Sum of pixel values used in HFV			(Paxel 0)
HFV_1 (peak or sum)			(Paxel 0)
HFV_sq_1 (peak or sum)			(Paxel 0)
HFV_count_1			(Paxel 0)

**Table 8-328. ISS ISP H3A AF Packet Format With Vertical AF Enabled  
(continued)**

Buffer Start Address (Byte Address) H3A_AFBUFST	31	16 15	0
HFV_2 (peak or sum)			(Paxel 0)
HFV_sq_2 (peak or sum)			(Paxel 0)
HFV_count_2			(Paxel 0)
ZEROES			(Paxel 0)
VFV_1			(Paxel 0)
VFV_sq_1			(Paxel 0)
VFV_count_1			(Paxel 0)
ZEROES			(Paxel 0)
VFV_2			(Paxel 0)
VFV_sq_2			(Paxel 0)
VFV_count_2			(Paxel 0)
ZEROES			(Paxel 0)
Sum of pixel values used in HFV			(Paxel 1)
HFV_1 (peak or sum)			(Paxel 1)
HFV_sq_1 (peak or sum)			(Paxel 1)
HFV_count_1			(Paxel 1)
...			

Figure 8-180 shows the windows and subsample definition used in tables.

Figure 8-180. ISS ISP H3A AE/AWB Window and Subsample Definition



camss-624

Table 8-329 lists the packet formats for AE/AWB for sum of square mode ( $H3A\_AEWCFCG[9:8]$  AEFMT = 0x0) .

Table 8-329. ISS ISP H3A AE/AWB Packet Format for Sum of Square Mode

	31	16	15	0	
Buffer address (byte address) $H3A\_AEWBUF$ ST	Subsample Accum[1]		Subsample Accum[0]		Window 0 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
	Sum of squares[0]				
	Sum of squares[1]				
	Sum of squares[2]				
	Sum of squares[3]				
$H3A\_AEWBUF$ ST + 32 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 1 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		

**Table 8-329. ISS ISP H3A AE/AWB Packet Format for Sum of Square Mode (continued)**

	31	16 15	0	
	Sum of squares[0]			
	Sum of squares[1]			
	Sum of squares[2]			
	Sum of squares[3]			
H3A_AEWBUF ST + 64 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 2 data	
	Subsample Accum[3]	Subsample Accum[2]		
	Saturator Accum[1]	Saturator Accum[0]		
	Saturator Accum [3]	Saturator Accum[2]		
	Sum of squares[0]			
	Sum of squares[1]			
	Sum of squares[2]			
	Sum of squares[3]			
H3A_AEWBUF ST + 96 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 3 data	
	Subsample Accum[3]	Subsample Accum[2]		
	Saturator Accum[1]	Saturator Accum[0]		
	Saturator Accum [3]	Saturator Accum[2]		
	Sum of squares[0]			
	Sum of squares[1]			
	Sum of squares[2]			
	Sum of squares[3]			
H3A_AEWBUF ST + 128 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 4 data	
	Subsample Accum[3]	Subsample Accum[2]		
	Saturator Accum[1]	Saturator Accum[0]		
	Saturator Accum [3]	Saturator Accum[2]		
	Sum of squares[0]			
	Sum of squares[1]			
	Sum of squares[2]			
	Sum of squares[3]			
H3A_AEWBUF ST + 160 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 5 data	
	Subsample Accum[3]	Subsample Accum[2]		
	Saturator Accum[1]	Saturator Accum[0]		
	Saturator Accum [3]	Saturator Accum[2]		
	Sum of squares[0]			
	Sum of squares[1]			
	Sum of squares[2]			
	Sum of squares[3]			
H3A_AEWBUF ST + 192 bytes	Subsample Accum[1]	Subsample Accum[0]	Window 6 data	
	Subsample Accum[3]	Subsample Accum[2]		
	Saturator Accum[1]	Saturator Accum[0]		
	Saturator Accum [3]	Saturator Accum[2]		
	Sum of squares[0]			
	Sum of squares[1]			
	Sum of squares[2]			
	Sum of squares[3]			

**Table 8-329. ISS ISP H3A AE/AWB Packet Format for Sum of Square Mode (continued)**

	31	16	15	0	
H3A_AEWBUF ST + 224 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 7 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
	Sum of squares[0]				
	Sum of squares[1]				
	Sum of squares[2]				
	Sum of squares[3]				
H3A_AEWBUF ST + 256 bytes	Unsaturated count, win 1		Unsaturated count, win 0		Unsaturated block count for the above 8 windows
	Unsaturated count, win 3		Unsaturated count, win 2		
	Unsaturated count, win 5		Unsaturated count, win 4		
	Unsaturated count, win 7		Unsaturated count, win 6		
	Data for next 8 windows, and so on. If the total number of windows is not a multiple of 8, the unsaturated counters are written immediately following the last window data. For example, if the total number of windows (including the black row) are 43, the first 40 windows are written out as per the 272-byte boundary above. Then the remaining three windows are written at +0, +32, and +64 bytes. The counts are written out at +96 instead of +256-byte boundary.				

Table 8-330 lists the packet formats for AE/AWB in minimum-maximum mode (H3A\_AEWCFG[9:8] AEFMT = 0x1).

**Table 8-330. ISS ISP H3A AE/AWB Packet Format for Minimum-Maximum Mode**

	31	16	15	0	
Buffer address (byte address) H3A_AEWBUF ST	Subsample Accum[1]		Subsample Accum[0]		Window 0 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
	Minimum[1]		Minimum[0]		
	Minimum[3]		Minimum[2]		
	Maximum[1]		Maximum[0]		
	Maximum[3]		Maximum[2]		
H3A_AEWBUF ST + 32 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 1 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
	Minimum[1]		Minimum[0]		
	Minimum[3]		Minimum[2]		
	Maximum[1]		Maximum[0]		
	Maximum[3]		Maximum[2]		
H3A_AEWBUF ST + 64 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 2 data
	Subsample Accum[3]		Subsample Accum[2]		

**Table 8-330. ISS ISP H3A AE/AWB Packet Format for Minimum-Maximum Mode  
(continued)**

		31	16 15	0
H3A_AEWBUF ST + 96 bytes		Saturator Accum[1]		Saturator Accum[0]
		Saturator Accum [3]		Saturator Accum[2]
		Minimum[1]		Minimum[0]
		Minimum[3]		Minimum[2]
		Maximum[1]		Maximum[0]
		Maximum[3]		Maximum[2]
		Subsample Accum[1]		Subsample Accum[0]
H3A_AEWBUF ST + 128 bytes		Subsample Accum[3]		Subsample Accum[2]
		Saturator Accum[1]		Saturator Accum[0]
		Saturator Accum [3]		Saturator Accum[2]
		Minimum[1]		Minimum[0]
		Minimum[3]		Minimum[2]
		Maximum[1]		Maximum[0]
		Maximum[3]		Maximum[2]
H3A_AEWBUF ST + 160 bytes		Subsample Accum[1]		Subsample Accum[0]
		Subsample Accum[3]		Subsample Accum[2]
		Saturator Accum[1]		Saturator Accum[0]
		Saturator Accum [3]		Saturator Accum[2]
		Minimum[1]		Minimum[0]
		Minimum[3]		Minimum[2]
		Maximum[1]		Maximum[0]
H3A_AEWBUF ST + 192 bytes		Maximum[3]		Maximum[2]
		Subsample Accum[1]		Subsample Accum[0]
		Subsample Accum[3]		Subsample Accum[2]
		Saturator Accum[1]		Saturator Accum[0]
		Saturator Accum [3]		Saturator Accum[2]
		Minimum[1]		Minimum[0]
		Minimum[3]		Minimum[2]
H3A_AEWBUF ST + 224 bytes		Maximum[1]		Maximum[0]
		Maximum[3]		Maximum[2]
		Subsample Accum[1]		Subsample Accum[0]
		Subsample Accum[3]		Subsample Accum[2]
		Saturator Accum[1]		Saturator Accum[0]
		Saturator Accum [3]		Saturator Accum[2]
		Minimum[1]		Minimum[0]
	Minimum[3]		Minimum[2]	

**Table 8-330. ISS ISP H3A AE/AWB Packet Format for Minimum-Maximum Mode (continued)**

31		16	15	0	
H3A_AEWBUF ST + 256 bytes	Maximum[1]		Maximum[0]		Unsaturated block count for the above 8 windows
	Maximum[3]		Maximum[2]		
	Unsaturated count, win 1		Unsaturated count, win 0		
	Unsaturated count, win 3		Unsaturated count, win 2		
	Unsaturated count, win 5		Unsaturated count, win 4		
	Unsaturated count, win 7		Unsaturated count, win 6		
	Data for next 8 windows, and so on. If the total number of windows is not a multiple of 8, the unsaturated counters are written immediately following the last window data. For example, if the total number of windows (including the black row) are 43, the first 40 windows are written out as per the 272-byte boundary above. Then the remaining three windows are written at +0, +32, and +64 bytes. The counts are written out at +96 instead of +256-byte boundary.				

Table 8-331 lists the packet formats for AE/AWB in sum-only mode (H3A\_AEWCFG[9:8] AEFMT = 0x2).

**Table 8-331. ISS ISP H3A AE/AWB Packet Format for Sum-Only Mode**

31		16	15	0	
Buffer address (byte address) H3A_AEWBUF ST	Subsample Accum[1]		Subsample Accum[0]		Window 0 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
H3A_AEWBUF ST + 32 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 1 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
H3A_AEWBUF ST + 64 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 2 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
H3A_AEWBUF ST + 96 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 3 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
H3A_AEWBUF ST + 128 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 4 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
H3A_AEWBUF ST + 160 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 5 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		

**Table 8-331. ISS ISP H3A AE/AWB Packet Format for Sum-Only Mode  
(continued)**

	31	16	15	0	
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
H3A_AEWBUF ST + 192 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 6 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
H3A_AEWBUF ST + 224 bytes	Subsample Accum[1]		Subsample Accum[0]		Window 7 data
	Subsample Accum[3]		Subsample Accum[2]		
	Saturator Accum[1]		Saturator Accum[0]		
	Saturator Accum [3]		Saturator Accum[2]		
H3A_AEWBUF ST + 256 bytes	Unsaturated count, win 1		Unsaturated count, win 0		Unsaturated block count for the above 8 windows
	Unsaturated count, win 3		Unsaturated count, win 2		
	Unsaturated count, win 5		Unsaturated count, win 4		
	Unsaturated count, win 7		Unsaturated count, win 6		
	Data for next 8 windows, and so on. If the total number of windows is not a multiple of 8, the unsaturated counters are written immediately following the last window data. For example, if the total number of windows (including black row) are 43, the first 40 windows are written out as per the 272-byte boundary above. Then the remaining three windows are written at +0, +32, and +64 bytes. The counts are written out at +96 instead of +256-byte boundary.				

### 8.3.3.6.8 ISS ISP H3A Events and Status Checking

The AF and AEW engines generate an interrupt event at the end of processing each frame. However, these two interrupts are internally tied together so that only one H3A interrupt signal is generated. If the AF engine and AEW engine do not process the same frame concurrently, this should not be an issue. However, if they do run concurrently, one of two outcomes may occur:

- The H3A interrupt may seem to trigger only once for each frame. This can happen when the processing for the AF and AEW engines finishes at or near the same time. The interrupt service routine does not have enough time to clear the interrupt flag for the first interrupt before the second interrupt occurs.
- The H3A interrupt may trigger twice for each frame. This can happen when the AF engine or the AEW engine finishes processing the frame much earlier than the other engine. In this case, the interrupt service routine does have enough time to clear the interrupt flag for the first interrupt by the time the second interrupt occurs.

The outcome depends on the difference in location of the last paxel/window in the frame (determines when processing is finished), the frequency of the relative clocks in the system, the occurrence and triggering of other interrupts in the system, and the latencies of the context switching and interrupt service routine execution.

The [H3A\\_PCR\[15\] BUSYAF](#) and/or [H3A\\_PCR\[18\] BUSYAEAWB](#) status bits are set when the start of frame occurs (if the [H3A\\_PCR\[0\] AF\\_EN](#) and/or [H3A\\_PCR\[16\] AEW\\_EN](#) bits are 1 at that time). They are automatically reset to 0 at the end of processing a frame. The [H3A\\_PCR\[15\] BUSYAF](#) and/or [H3A\\_PCR\[18\] BUSYAEAWB](#) status bits may be polled to determine the end of frame status.



### 8.3.3.7 ISS ISP ISIF Functional Description

#### 8.3.3.7.1 ISS ISP ISIF Overview

The image sensor interface (ISIF) module receives RAW or YUV4:2:2 data from the IPIPEIF module. The module outputs data back to the IPIPEIF module and can also output data to memory through the BL (Buffer Logic) module. The ISIF module can process the incoming data and supports the following functions:

- Maximum supported image size is 32,768 X 32,768
- Supports up to 16-bit analog front end
- Sensor Data linearization
- Supports Bayer and Foveon® input data format (RGB and CMYG color support)
- Supports VGA read out mode
- Supports various image data format
- Color space conversion
- Digital clamp with Horizontal/Vertical offset drift compensation
- Vertical line defect correction
- Programmable 2D-matrix LSC
- Gain and offset control
- Programmable H/V culling pattern
- Max pixel rate clock of 200 MPix/s on the VP interface.
- 10-to-8 bits A-Law compression table inside
- 12-bit pack supported when written to memory

Figure 8-181 show the ISIF module connections to other submodules of the ISP.

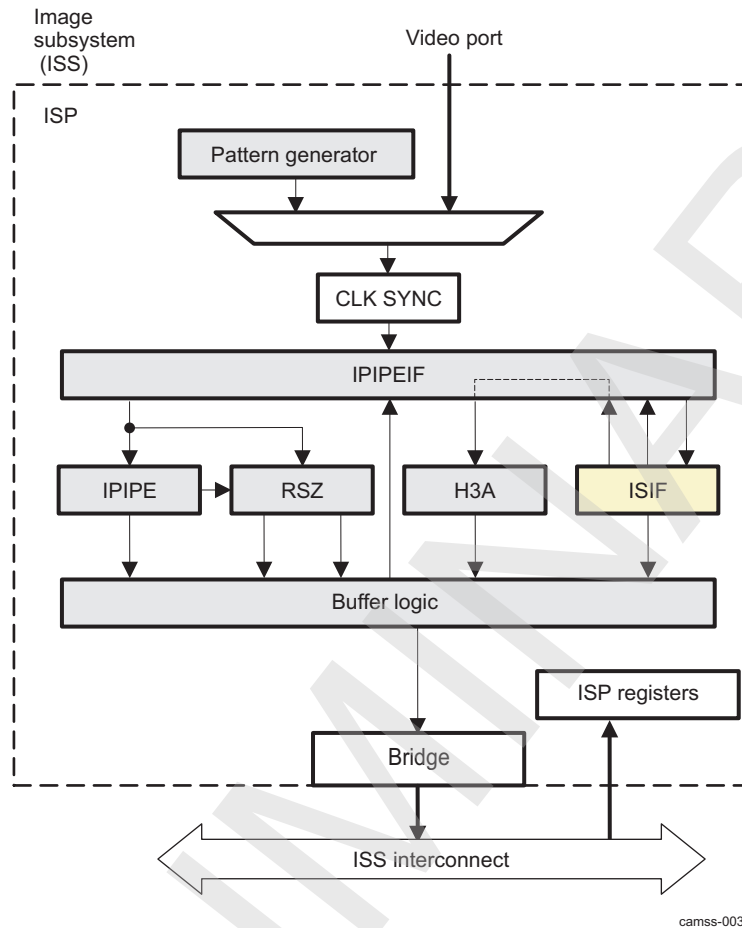
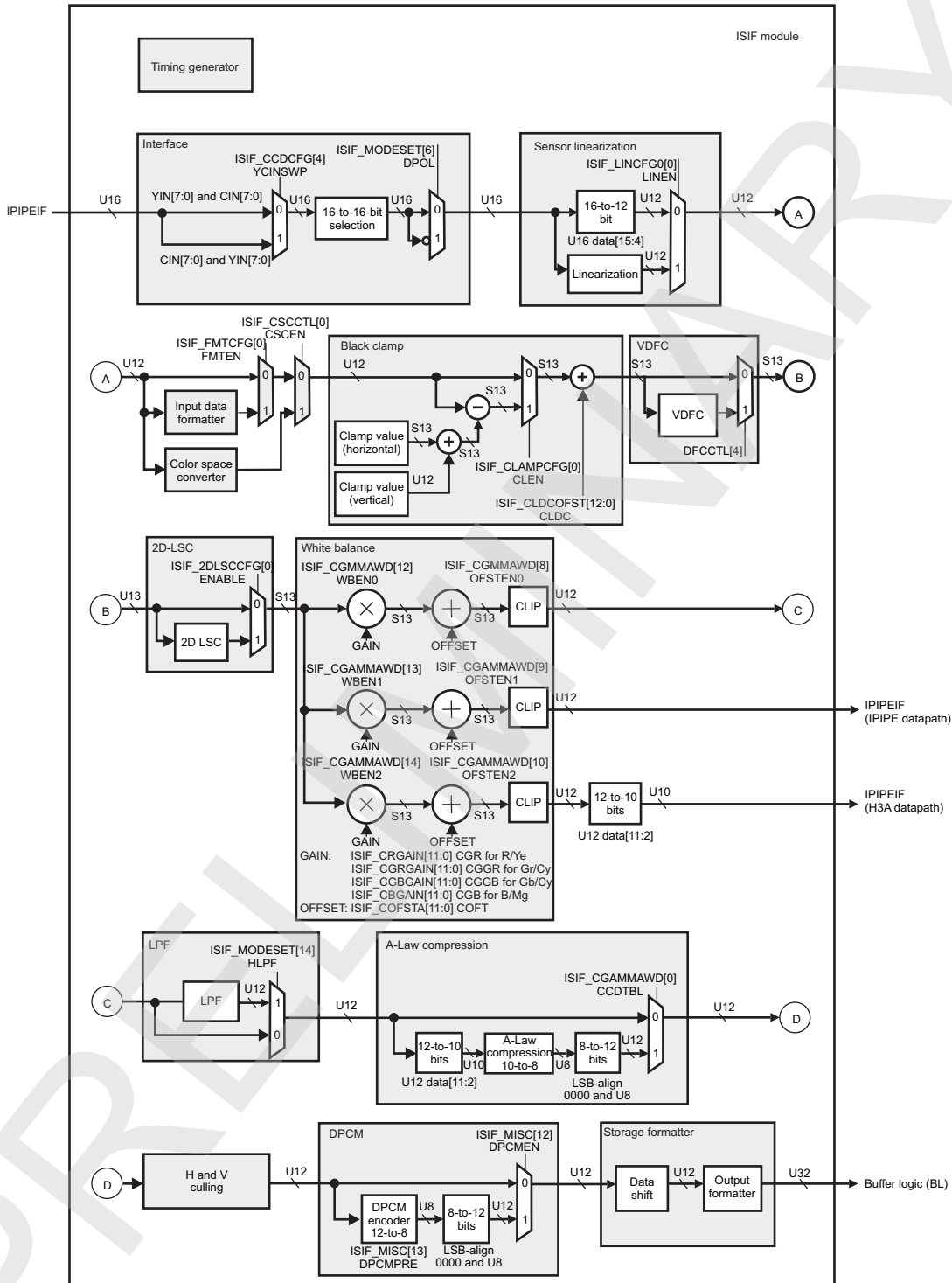
**Figure 8-181. ISS ISP ISIF High-Level Diagram****8.3.3.7.2 ISS ISP ISIF Top-Level Block Diagram**

Figure 8-182 shows the different blocks of the ISIF module.

Figure 8-182. ISS ISP ISIF Top-Level Block Diagram



camss-074

The following sections describe the blocks in the ISIF module.

### 8.3.3.7.3 ISS ISP ISIF Input Interface

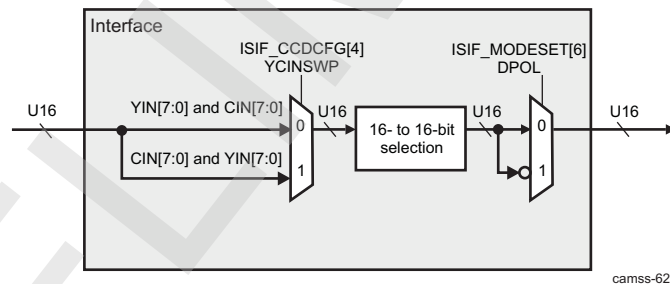
The input interface is a video interface. It comprises the horizontal(HD)I and vertical(VD) synchronization signal, pixel clock (PCLK), and data(DATA). [Table 8-332](#) gives more information about these different signals. The ISIF uses the HD and VD signals provided by the sensor through the VP and IPIPEIF. The pixel clock clocks data into the ISIF at a maximum rate of 200MHz.

**Table 8-332. ISS ISP ISIF Input Interface Signals**

Name	I/O	Function
VD	I	Vertical sync signal
HD	I	Horizontal sync signal
PCLK	I	Pixel clock. This signal is the pixel clock used to load image data into the ISIF. The clock controller can configure to trigger on the rising or falling edge of the PCLK signal.
DATA	I	Data. The data interface is a 16-bit interface. When the ISIF is configured to write data to SDRAM, the write enable signal allows an external device to control which data is to be written to SDRAM. The data input can be configured from the <a href="#">ISIF_MODESET.INPMOD</a> register where it can be set to RAW, YCbCr (16 bits and 8 bits). The polarity of the data can be changed from the <a href="#">ISIF_MODESET.DPOL</a> as shown in <a href="#">Figure 8-183</a> .

### 8.3.3.7.4 ISS ISP ISIF Interface

**Figure 8-183. ISS ISP ISIF Interface Block Diagram**



The ISIF supports 8- to 16-bit-wide RAW data signals and 8-/16-bit YCbCr signals, as described in [Table 8-333](#). The interface can be set in the three different modes from the [ISIF\\_MODESET\[13:12\]](#) INPMOD bit field. The [ISIF\\_CCDCFG\[11\]](#) Y8POS bit selects the y signal positioning whenever YUV4:2:2 is input. Moreover, if CCIR656 input is used the width of selected bit can be set through the [ISIF\\_CCDCFG\[5\]](#) BT656 bit.

**Table 8-333. ISS ISP ISIF Data Input Formats**

ISIF Input Port Name	RAW Data	16-bit YCbCr	8-bit YCbCr
Y17	C_DATA15	Y7	
Y16	C_DATA14	Y6	
Y15	C_DATA13	Y5	
Y14	C_DATA12	Y4	
Y13	C_DATA11	Y3	
Y12	C_DATA10	Y2	
Y11	C_DATA9	Y1	
Y10	C_DATA8	Y0	

**Table 8-333. ISS ISP ISIF Data Input Formats (continued)**

ISIF Input Port Name	RAW Data	16-bit YCbCr	8-bit YCbCr
CI7	C_DATA7	Cb7,Cr7	Y7,Cb7,Cr7
CI6	C_DATA6	Cb6,Cr6	Y6,Cb6,Cr6
CI5	C_DATA5	Cb5,Cr5	Y5,Cb5,Cr5
CI4	C_DATA4	Cb4,Cr4	Y4,Cb4,Cr4
CI3	C_DATA3	Cb3,Cr3	Y3,Cb3,Cr3
CI2	C_DATA2	Cb2,Cr2	Y2,Cb2,Cr2
CI1	C_DATA1	Cb1,Cr1	Y1,Cb1,Cr1
CI0	C_DATA0	Cb0,Cr0	Y0,Cb0,Cr0

Y and C input signals can be swapped through [ISIF\\_CCDCFG\[4\]](#) YCINSWP bit.

In case of RAW data at ISIF input, a 16- to-16-bit selection can be done: when the number of RAW data lines is less than 16, data can be connected to the upper or lower lines of C\_DATA[15:0]. Lines not connected must be tied low. As shown in [Table 8-334](#), the [ISIF\\_CGAMMAWD\[4:1\]](#) GWDI bit field must be configured correctly so that the MSB of the input is connected to the MSB of the 16-bit data bus in ISIF.

**Table 8-334. ISS ISP ISIF Raw Data Connection: Selects MSB Position of Input Data**

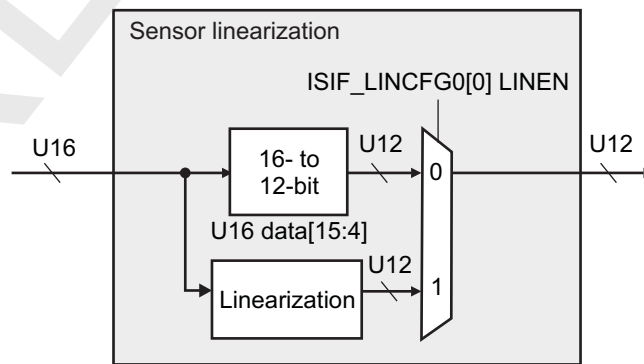
ISIF_CGAMMAWD[4:1] GWDI	16-to-16-bit Selection
0	C_DATA [15:0] = C_DATA[15:0]
1	C_DATA [15:0] = C_DATA[14:0] & 0
2	C_DATA [15:0] = C_DATA[13:0] & 00
3	C_DATA [15:0] = C_DATA[12:0] & 000
4	C_DATA [15:0] = C_DATA[11:0] & 0000
5	C_DATA [15:0] = C_DATA[10:0] & 00000
6	C_DATA [15:0] = C_DATA[9:0] & 000000
7	C_DATA [15:0] = C_DATA[8:0] & 0000000
8	C_DATA [15:0] = C_DATA[7:0] & 00000000

The polarity of the input image data can be switched through the [ISIF\\_MODESET\[6\]](#) DPOL bit.

**8.3.3.7.5 ISS ISP ISIF Sensor Linearization**

**NOTE:** For the memory access locations of the sensor linearization table, see [Section 8.3.3.9](#).

**Figure 8-184. ISS ISP ISIF Sensor Linearization Block Diagram**



camss-627

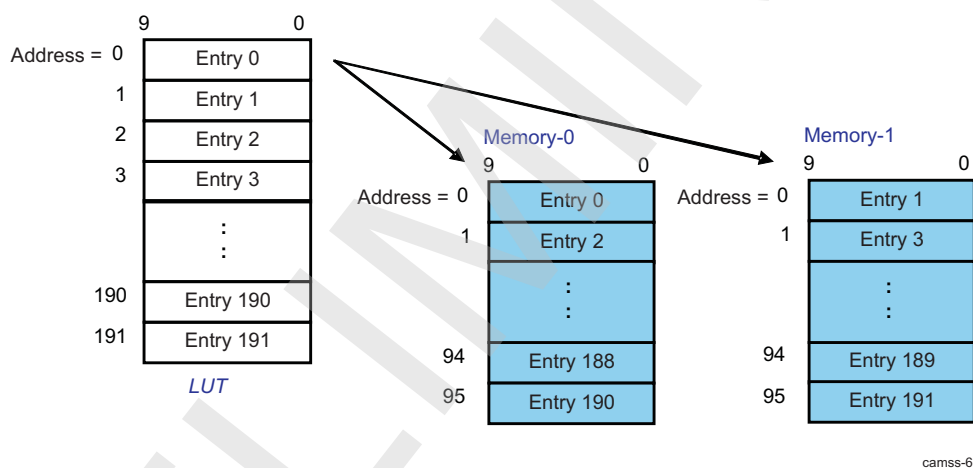
The sensor linearization module can correct for the non-linear response of image sensors. A LUT is programmed with an offset value to add to the original pixel value based on the original pixel value.

The LUT is a sampling of the linearization correction curve based on calibration of the image sensor. Intermediate values between sampling points are computed using linear interpolation. The entire correction curve is divided into seven regions, as shown in [Table 8-335](#). The regions for the darkest part and the brightest part of the response curve have dense sampling. The linearization mode can be a uniform or a non-uniform sampling and can be set through the `ISIF_LINCFG0[1] LINMD` bit.

**Table 8-335. ISS ISP ISIF Linearization LUT**

Region	Number of Sample Points	LUT Address
<code>table_in[15:11] == 00000</code>	32	<code>table_in[10:6]</code>
<code>table_in[15:11] == 00001</code>	4	<code>table_in[10:9] + 32</code>
<code>table_in[15:12] == 0001</code>	4	<code>table_in[11:10] + 36</code>
<code>table_in[15:13] == 001</code>	4	<code>table_in[12:11] + 40</code>
<code>table_in[15:14] == 01</code>	4	<code>table_in[13:12] + 44</code>
<code>table_in[15:14] == 10</code>	16	<code>table_in[13:10] + 48</code>
<code>table_in[15:14] == 01</code>	128	<code>table_in[13:7] + 64</code>

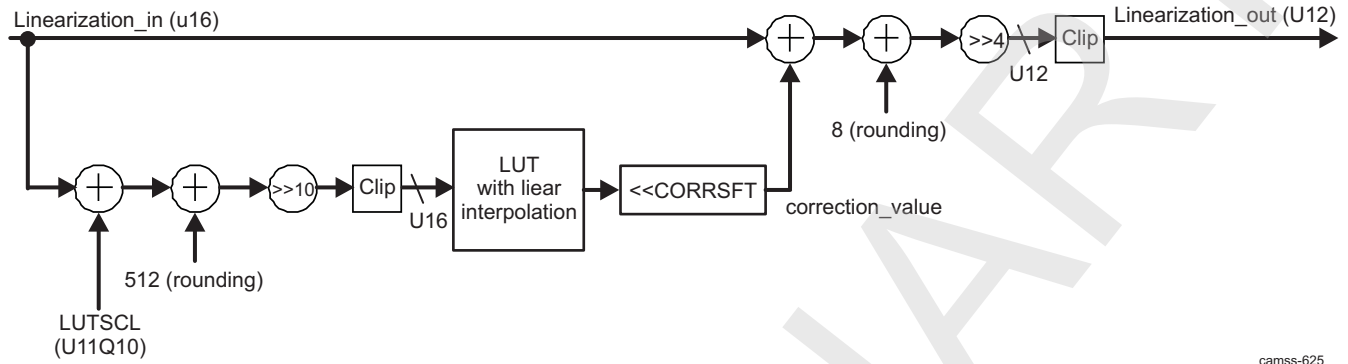
The LUT has 192 entries and is split into two 96 ×10-bit memories, as shown in [Figure 8-185](#). The table is mapped in the memory map. The LUT entries are interleaved between memory 0 and memory 1.

**Figure 8-185. ISS ISP ISIF Linearization LUT Memories****Table 8-336. ISS ISP ISIF LUT Memory Region**

Memory Region	Address Range	Description
Memory 0	0xC000 – 0xC17F	ISIF linearity compensation LUT 0
Memory 1	0xC400 – C57F	ISIF linearity compensation LUT 1

A scale factor is applied to the input before lookup through the `ISIF_LINCFG1[10:0]` LUTSCL bit field. The LUT entries are signed 10-bit data (u16). After linear interpolation, the correction value is left-shifted by a programmable amount (the `ISIF_LINCFG0[6:4]` CORRSFT bit field), and then added to the input. This is then converted to unsigned 12-bit by right shift, followed by clipping.

Figure 8-186. ISS ISP ISIF Linearization Block Diagram

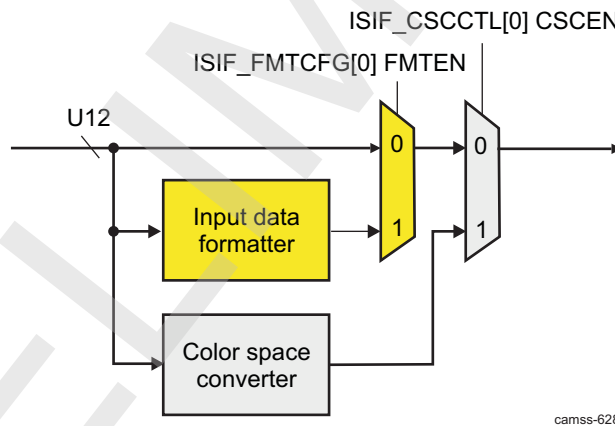


To enable the linearization module, set the `ISIF_LINCFG0[0]` LINEN bit to 1.

If the linearization module is disabled (`ISIF_LINCFG0[0]` LINEN = 0x0), a 16- to-12-bit transformation is done, and the upper 12-bits of U16 input are sent to the next block.

### 8.3.3.7.6 ISS ISP ISIF Input Data Formatter

Figure 8-187. ISS ISP ISIF Input Data Formatter Block Diagram



There are two functional blocks: input data formatter and color space converter, which use two 5376 x 12-bit memories (corresponds to one line of maximum 5376 pixels with each pixel equal to 12 bits). Only one of the function blocks can be enabled.

The input data formatter block allows the ISIF to handle a wide variety of current and future readout schemes other than Bayer format. Two line memories and a programmable address generator are used to translate those patterns into a standard Bayer pattern (or any other pattern). This allows the back-end processing (noise filters, interpolation, histogram, 3A statistics) to remain unchanged.

The input data formatter block also supports divided input lines. In case an input line is divided into multiple lines and fed to the ISIF, the formatter gathers the divided lines and organizes a single line. Up to four divided lines can be supported.

The input data formatter is enabled through the `ISIF_FMTCFG[0]` FMTEN bit.

The input data formatter can split an input line into 1, 2, 3, or 4 output lines, or can combine the divided 1, 2, 3, or 4 input lines into a single line.

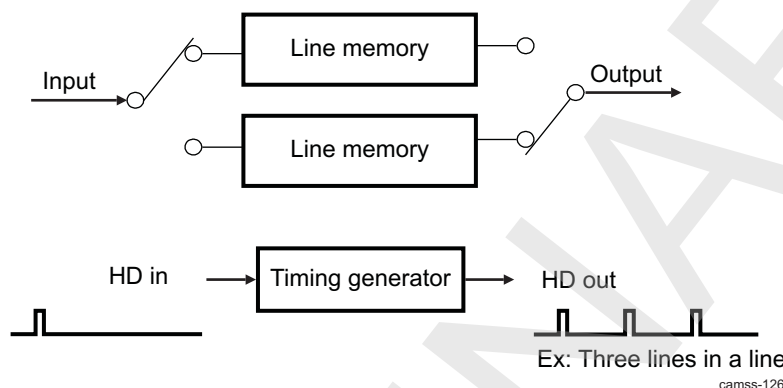
- Set the `ISIF_FMTCFG[1]` FMTCBL bit to 0 for split mode.

- Set the [ISIF\\_FMTCFG\[1\]](#) FMTCBL bit to 1 for combine mode.
- Select the number of lines in the [ISIF\\_FMTCFG\[5:4\]](#) LNUM bit field.

The Input data formatter can work in normal or line alternative mode. The choice is done through the [ISIF\\_FMTCFG\[2\]](#) LNALT bit.

[Figure 8-188](#) shows an example of generating three output lines from an input line with a new, internally generated HD signal.

**Figure 8-188. ISS ISP ISIF Splits an Input Line Into Three Output Lines**



This HD signal then gates the downstream processing rather than the original sensor HD signal. Descriptions of how to configure the formatter are provided in the following sections.

Because the size of the line memories is 5376 × 12-bits, the following restrictions apply for the data formatter:

- Split mode:
  - The maximum number of pixels that can be supported in an output line if the input line is transformed into 1 output line is 5376
  - The maximum number of pixels that can be supported in an output line if the input line is transformed into 2 output lines is 2688
  - The maximum number of pixels that can be supported in an output line if the input line is transformed into 3 output lines is 1792
  - The maximum number of pixels that can be supported in an output line if the input line is transformed into 4 output lines is 1344
- Combine mode
  - The maximum number of pixels that can be supported in an input line if 1 input line is transformed into an output line is 5376.
  - The maximum number of pixels that can be supported in an input line if 2 input lines are transformed into an output line is 2368.
  - The maximum number of pixels that can be supported in an input line if 3 input lines are transformed into an output line is 1792.
  - The maximum number of pixels that can be supported in an input line if 4 input lines are transformed into an output line is 1344.

#### 8.3.3.7.6.1 ISS ISP ISIF Formatter Area Settings

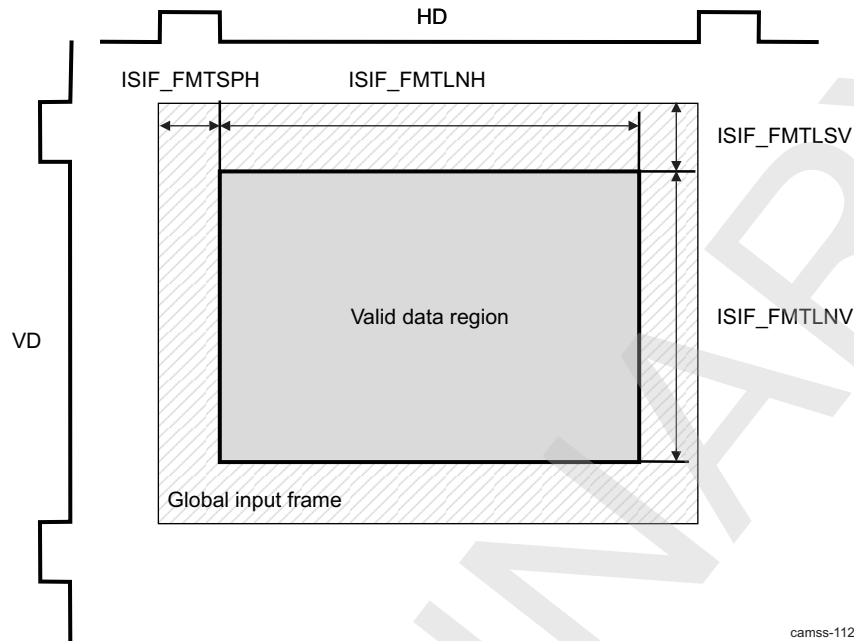
As shown in [Figure 8-189](#), the following registers are used to set the formatter area:

- [ISIF\\_FMTSPH](#)
- [ISIF\\_FMTLNH](#)
- [ISIF\\_FMTLSV](#)
- [ISIF\\_FMTLNV](#)

[Table 8-337](#) describes these registers. The input line is input to the formatter, and the output line is output from the formatter.



Figure 8-189. ISS ISP ISIF Input Data Formatter Area Settings



camss-112

Table 8-337. ISS ISP ISIF Input Data Formatter Area Setting Registers

Register	Description
ISIF_FMTSPH	The first valid pixel of an input line
ISIF_FMTLNH	Valid length of a input line = FMTLNH + 1
ISIF_FMTLSV	The first valid input line
ISIF_FMTLNV	The number of the valid input lines = FMTLNV + 1

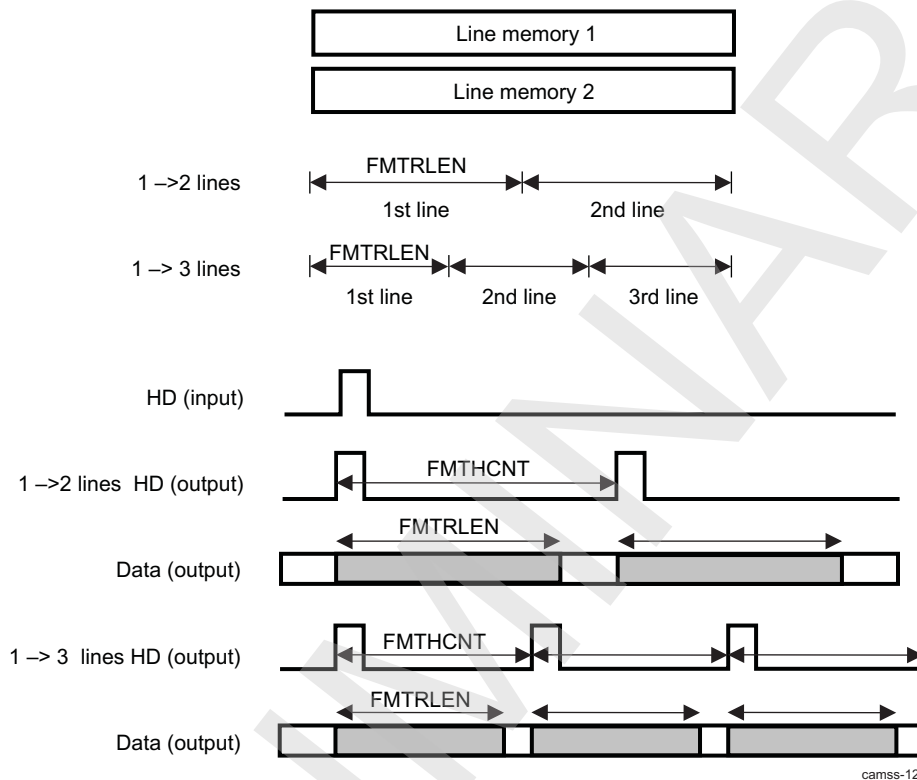
Table 8-338. ISS ISP ISIF Output Data Formatter Area Setting Registers

Register	Description
ISIF_FMTRLEN	The length of an output line
ISIF_FMTHCNT	HD interval for output lines
ISIF_SPH	The first pixel in an output line to be stored to SDRAM
ISIF_LNH	Number of pixels in an output line to be stored to SDRAM = LNH + 1
ISIF_LNV	The number of the output lines to be stored to SDRAM = LNV + 1

The number of pixels in an output line must be set to the [ISIF\\_FMRLEN](#) register, and the HD output interval must be set to the [ISIF\\_FMTHCNT](#) register. It is not necessary to set the [ISIF\\_FMTHCNT](#) register if multiple input lines are combined into a single line.

[Figure 8-190](#) shows an example of splitting an input line into two or three output lines.

**Figure 8-190. ISS ISP ISIF Data Formatter Output Control Example**



### 8.3.3.7.6.2 ISS ISP ISIF Formatter Programming

The data formatter derives its flexibility by supporting up to 16 different addresses and a program that can contain up to 32 entries.

#### Address pointer

There are 16 address pointer registers ([ISIF\\_FMTAPTR0](#) to [ISIF\\_FMTAPTR15](#)), which contain:

- The [ISIF\\_FMTAPTRx\[14:13\]](#) LINE bit field: 2-bit line number to specify the output line to which it belongs: 0, 1, 2, or 3. It is valid only for the line splitting.
- The [ISIF\\_FMTAPTRx\[12:0\]](#) INIT bit field: 13-bit initial address for pointer x (where x = 0 to 15)

Each of the address values is auto-incremented or auto-decremented by a programmable value (the [ISIF\\_FMTCFG\[11:8\]](#) FMTAINC bit field).

#### Program

There are 32 program entry registers, which contain:

- In the [ISIF\\_FMTPGMVFO](#) and [ISIF\\_FMTPGMVF1](#) registers: The PGMxxEN fields (where xx = 00 to 31) set the program entry valid flag.
- In the [ISIF\\_FMTPGMAPS0](#) to [ISIF\\_FMTPGMAPS7](#) registers: The PGMxxAPTR fields specify the program xx address pointer (where xx = 00 to 31).
- In the [ISIF\\_FMTPGMAPU0](#) and [ISIF\\_FMTPGMAPU1](#) registers: The PGMxxUPDT fields (where xx = 00 to 31) set the program xx address update (increment or decrement).

Because each of the program entries has a valid flag, the formatter can also support images larger than the memory limit (length: 4480) by not choosing some of the entries.

The total program memory of 32 entries is divided into two or four SETs:

- Program memory of 16 entries each for odd and even lines to split the line:
  - SET0 for even input line: Program entry 015
  - SET1 for odd input line: Program entry 1631
- Program memory of eight entries for up to four input lines to combine the lines:
  - SET0 for first input line: Program entry 07
  - SET1 for second input line: Program entry 815
  - SET2 for third input line: Program entry 1623
  - SET3 for fourth input line: Program entry 2431

The number of program entries per SET must be specified by the **ISIF\_FMTPLEN** register as follows:

- Number of program entries for SET0: **ISIF\_FMTPLEN**[3:0] FMTPLEN0
- Number of program entries for SET1: **ISIF\_FMTPLEN**[7:4] FMTPLEN1
- Number of program entries for SET2: **ISIF\_FMTPLEN**[10:8] FMTPLEN2
- Number of program entries for SET3: **ISIF\_FMTPLEN**[14:12] FMTPLEN3

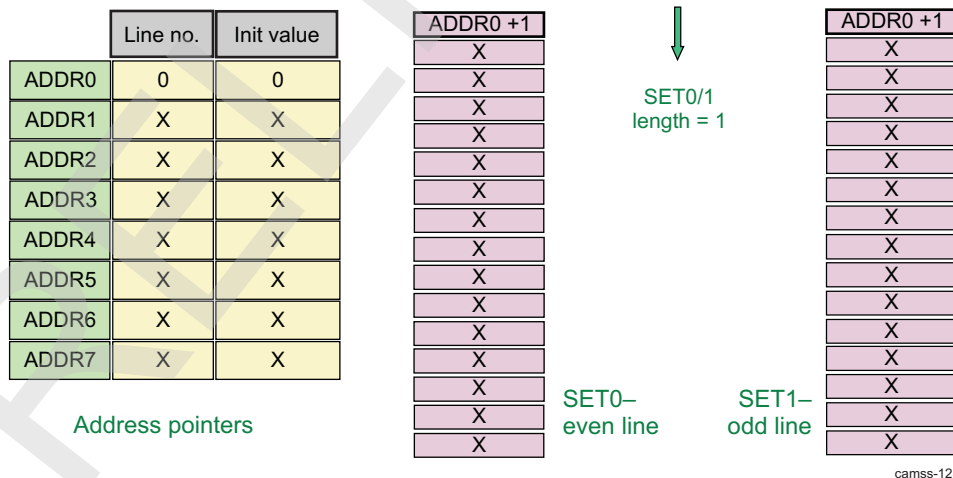
The program entry must be set from the lower registers within a SET. For instance, start from program entry 16 for odd input lines.

The following examples show the programmability of the data formatter:

- Register settings:
  - **ISIF\_FMTCFG**[0] FMTEN = 0x1
  - **ISIF\_FMTCFG**[1] FMTCBL = 0x0
  - **ISIF\_FMTCFG**[11:8] FMTAINC = 0x0 (add or subtract 1)
- 1 input line -> 1 output line

**NOTE:** ADDR0 is set to an example value. Addresses with no value in the following example are marked with X.

**Figure 8-191. ISS ISP ISIF Conventional Read-Out Pattern**

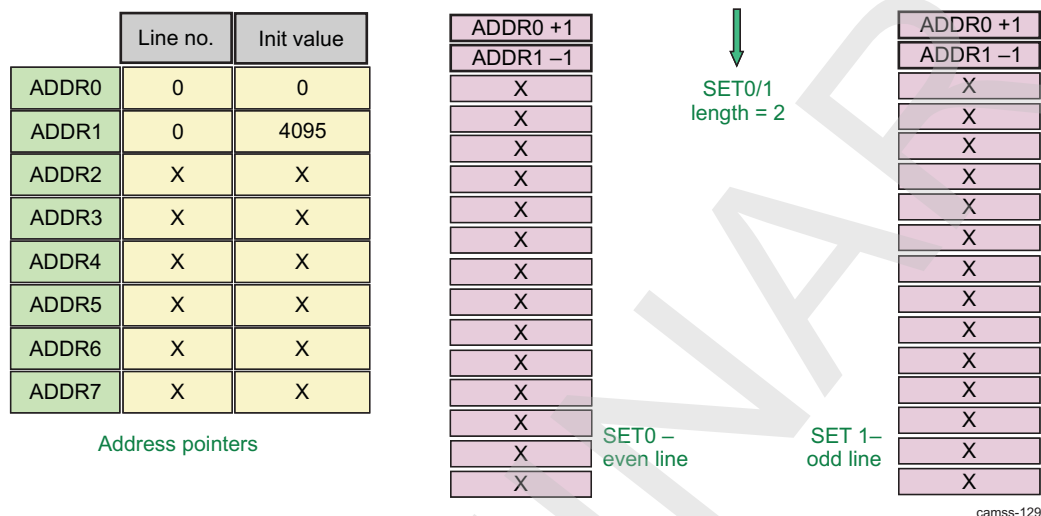


The following examples show the programmability of the data formatter with a 2-tap analog front end (AFE):

- Registers settings:
  - **ISIF\_FMTCFG**[0] FMTEN = 0x1
  - **ISIF\_FMTCFG**[1] FMTCBL = 0x0
  - **ISIF\_FMTCFG**[11:8] FMTAINC = 0x0 (add or subtract 1)
  - **ISIF\_FMTCFG**[5:4] LNUM = 0x0

- One input line (4096) -> one output line with left and right read-out
- Input -> First pixel, last pixel, first pixel + 1, last pixel - 1, and so on
- Input -> 0, 4095, 1, 4094, 2, 4093, 3, 4092, ..., 2047 and 2048
- Output -> 0, 1, 2, 3, ..., 4094 and 4095

**Figure 8-192. ISS ISP ISIF Conventional Read-Out Pattern With 2-tap AFE**



### 8.3.3.7.6.3 ISS ISP ISIF Combine the Divided Input Lines

The formatter can gather the divided input lines and organize a single line. Figure 8-193 shows an example generating a single output line from three input lines, and masking two out of three HD input pulses.

**Figure 8-193. ISS ISP ISIF Combine Three Input Lines Into Single Line**

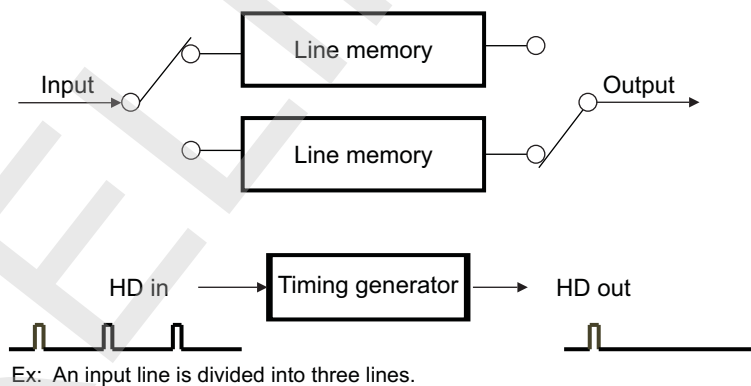


Figure 8-194 is an example that shows the register setting.

- Multiple input lines combined into one output line (use three input lines to one output line in this example)
- Input -> 4, 5, 10, 11, ..., 2, 3, 8, 9, ..., 0, 1, 6, 7, ... (three lines read from sensor)
- Output -> 0, 1, 2, 3, ... (1 line output)
- Register settings:
  - `ISIF_FMTCFG[0] FMTEN = 0x1`
  - `ISIF_FMTCFG[1] FMTCBL = 0x1`
  - `ISIF_FMTCFG[11:8] FMTAINC = 0x5` (add or subtract 6)
  - `ISIF_FMTCFG[5:4] LNUM = 0x2`

- SETs recycled based on LNUM

Figure 8-194. ISS ISP ISIF Example of Combining Three Input Lines Into a Single Line

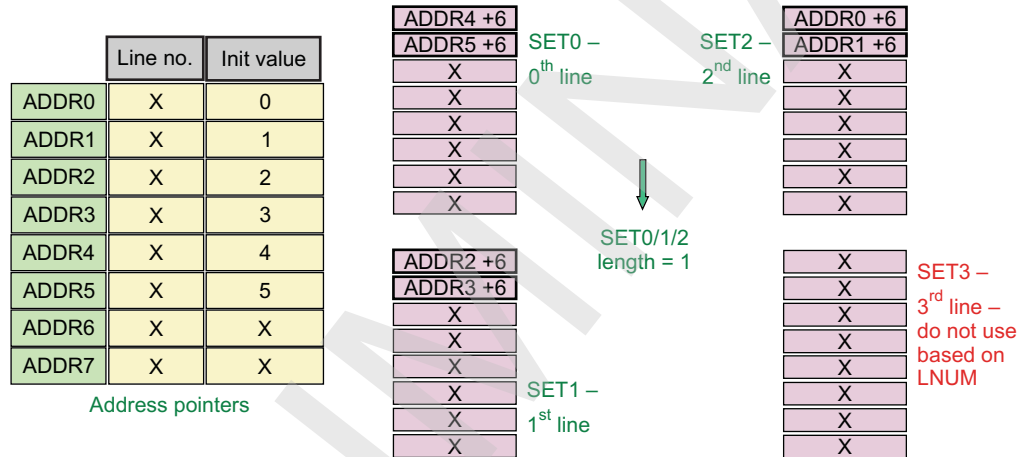
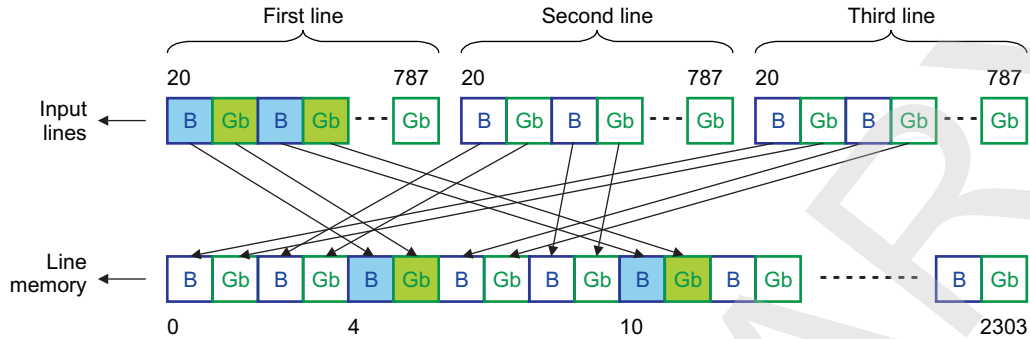


Table 8-339. ISS ISP ISIF Example of Combining Three Input Lines Into a Single Line: Register Setting Example

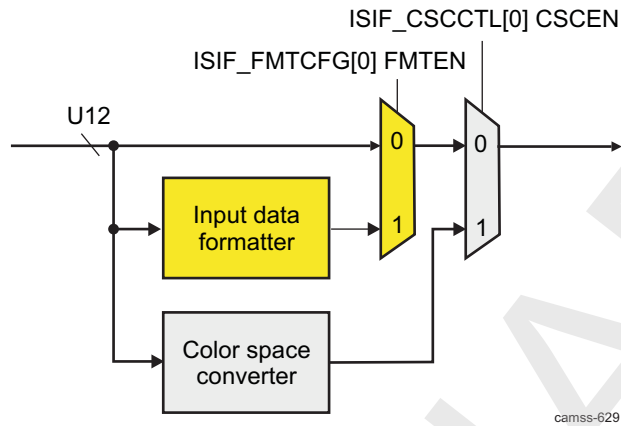
Step	Configuration Required	Size
Formatter enable	ISIF_FMTCFG[0] FMTEN	1
Combine input lines	ISIF_FMTCFG[1] FMTCBL	1
Address increment = FMTAINC + 1 = 6	ISIF_FMTCFG[11:8] FMTAINC	5
The first valid pixel of a divided line	ISIF_FMTSPH[12:0] FMTSPH	20
Valid length of a divided line = FMTLNH + 1 = 768	ISIF_FMTLNH[12:0] FMTLNH	767
The first valid divided line	ISIF_FMTLSV[12:0] FMTSLV	16
The number of the valid divided lines = FMTLNV + 1 = 4590	ISIF_FMTLNV[14:0] FMTLNV	4589
The length of an organized line = (FMTLNH + 1) x (LNUM + 1) = 2304	ISIF_FMTRLEN[12:0] FMTRLEN	2304
Split/combine line number = LNUM + 1 = 3	ISIF_FMTCFG[5:4] LNUM	2
Number of PGM entries for SET0 = FMTPLEN0 + 1 = 2	ISIF_FMTPLEN[3:0] FMTPLEN0	1
Number of PGM entries for SET1 = FMTPLEN1 + 1 = 2	ISIF_FMTPLEN[7:4] FMTPLEN1	1
Number of PGM entries for SET2 = FMTPLEN2 + 1 = 2	ISIF_FMTPLEN[10:8] FMTPLEN2	1
Address Pointer 0, INIT = 0	ISIF_FMTAPTR0[12:0] INIT	0
Address Pointer 1, INIT = 1	ISIF_FMTAPTR1[12:0] INIT	1

**Table 8-339. ISS ISP ISIF Example of Combining Three Input Lines Into a Single Line: Register Setting Example (continued)**

Step	Configuration Required	Size
Address Pointer 2, INIT = 2	ISIF_FMTAPTR2[12:0] INIT	2
Address Pointer 3, INIT = 3	ISIF_FMTAPTR3[12:0] INIT	3
Address Pointer 4, INIT = 4	ISIF_FMTAPTR4[12:0] INIT	4
Address Pointer 5, INIT = 5	ISIF_FMTAPTR5[12:0] INIT	5
Program 0 Valid flag	ISIF_FMTPGMVFO[0] PGM00EN	1
Program 1 Valid flag	ISIF_FMTPGMVFO[1] PGM01EN	1
Program 8 Valid flag	ISIF_FMTPGMVFO[8] PGM08EN	1
Program 9 Valid flag	ISIF_FMTPGMVFO[9] PGM09EN	1
Program 16 Valid flag	ISIF_FMTPGMV1[0] PGM16EN	1
Program 17 Valid flag	ISIF_FMTPGMV1[1] PGM17EN	1
Increment Address pointer = 0x0 Program 0 Address pointer = ADDR4 + 6	ISIF_FMTPGMAPU0[0] PGM0UPDT ISIF_FMTPGMAPS0[3:0] PGM0APTR	4
Increment Address pointer = 0x0 Program 1 Address pointer = ADDR5 + 6	ISIF_FMTPGMAPU0[1] PGM1UPDT ISIF_FMTPGMAPS0[7:4] PGM1APTR	5
Increment Address pointer = 0x0 Program 8 Address pointer = ADDR2 + 6	ISIF_FMTPGMAPU0[8] PGM8UPDT ISIF_FMTPGMAPS2[3:0] PGM8APTR	2
Increment Address pointer = 0x0 Program 9 Address pointer = ADDR3 + 6	ISIF_FMTPGMAPU0[9] PGM9UPDT ISIF_FMTPGMAPS2[7:4] PGM9APTR	3
Increment Address pointer = 0x0 Program 16 Address pointer = ADDR0 + 6	ISIF_FMTPGMAPU1[1] PGM17UPDT ISIF_FMTPGMAPS4[3:0] PGM16APTR	0
Increment Address pointer = 0x0 Program 17 Address pointer = ADDR1 + 6	ISIF_FMTPGMAPU1[0] PGM16UPDT ISIF_FMTPGMAPS4[7:4] PGM17APTR	1

8.3.3.7.7 ISS ISP ISIF Color Space Converter

Figure 8-195. ISS ISP ISIF Color Space Converter Block Diagram

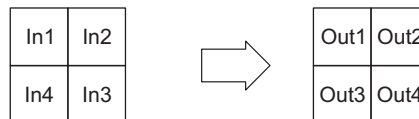


The color space converter (CSC) includes four 8-bit × 12-bit multipliers and one adder for the color space conversion. These multiplier/adder units are used for the operation described in Figure 8-196. Data are taken from two input lines during the operation.

Coefficients are signed 8-bit (decimal is 5 bits). Coefficients are set through the following registers:

- Coefficient M00: ISIF\_CSCM0[7:0] CSCM00
- Coefficient M01: ISIF\_CSCM0[15:8] CSCM01
- Coefficient M02: ISIF\_CSCM1[7:0] CSCM02
- Coefficient M03: ISIF\_CSCM1[15:8] CSCM03
- Coefficient M10: ISIF\_CSCM2[7:0] CSCM10
- Coefficient M11: ISIF\_CSCM2[15:8] CSCM11
- Coefficient M12: ISIF\_CSCM3[7:0] CSCM12
- Coefficient M13: ISIF\_CSCM3[15:8] CSCM13
- Coefficient M20: ISIF\_CSCM4[7:0] CSCM20
- Coefficient M21: ISIF\_CSCM4[15:8] CSCM21
- Coefficient M22: ISIF\_CSCM5[7:0] CSCM22
- Coefficient M23: ISIF\_CSCM5[15:8] CSCM23
- Coefficient M30: ISIF\_CSCM6[7:0] CSCM30
- Coefficient M31: ISIF\_CSCM6[15:8] CSCM31
- Coefficient M32: ISIF\_CSCM7[7:0] CSCM32
- Coefficient M33: ISIF\_CSCM7[15:8] CSCM33

Figure 8-196. ISS ISP ISIF Color Space Converter Operation



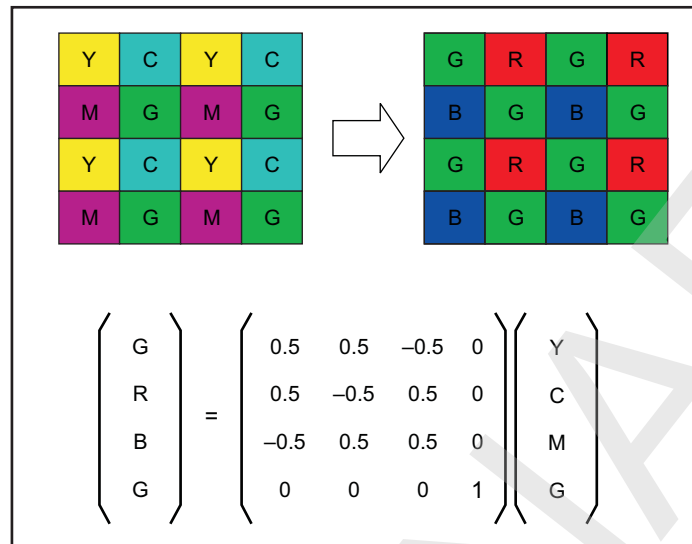
$$\begin{pmatrix} Out1 \\ Out2 \\ Out3 \\ Out4 \end{pmatrix} = \begin{pmatrix} M00 & M01 & M02 & M03 \\ M10 & M11 & M12 & M13 \\ M20 & M21 & M22 & M23 \\ M30 & M31 & M32 & M33 \end{pmatrix} \begin{pmatrix} In1 \\ In2 \\ In3 \\ In4 \end{pmatrix}$$

M00–M33: Signed 8-bit data with 5-bit decimal the value range  $-4 \leq M_{xx} < 4$

camss-618

The CSC can convert CMYG filtered CCD data to Bayer matrix (RGBG) data, as shown in Figure 8-197.

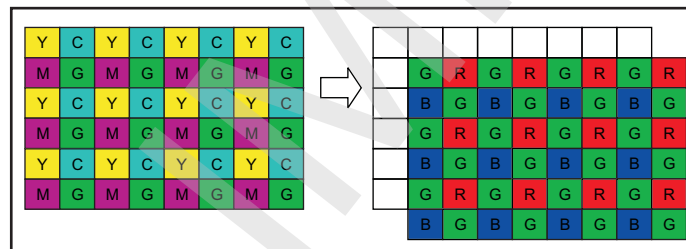
**Figure 8-197. ISS ISP ISIF Color Space Converter Operation: CMYG to RGBG**



camss-113

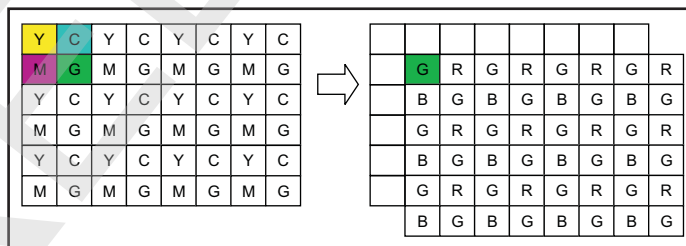
Figure 8-198 through Figure 8-200 show which input pixels are used for the operation. There is one-line latency between the input and the output.

**Figure 8-198. ISS ISP ISIF Color Space Conversion Example**



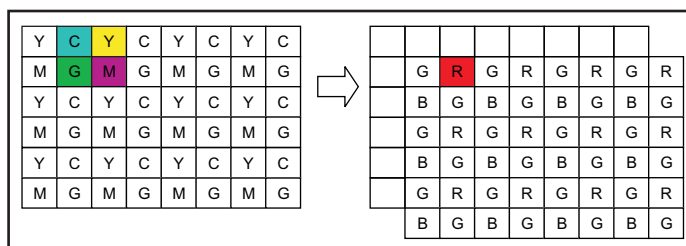
camss-114

**Figure 8-199. ISS ISP ISIF 1st Pixel/1st Line Generation**



camss-115

**Figure 8-200. ISS ISP ISIF 2nd Pixel/1st Line Generation**

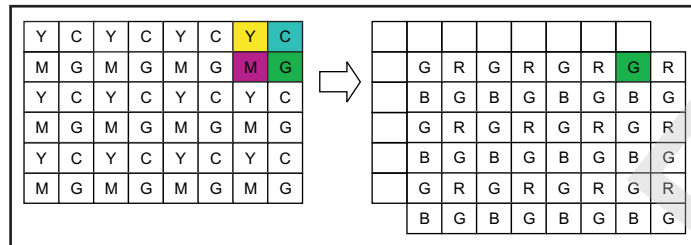


camss-116



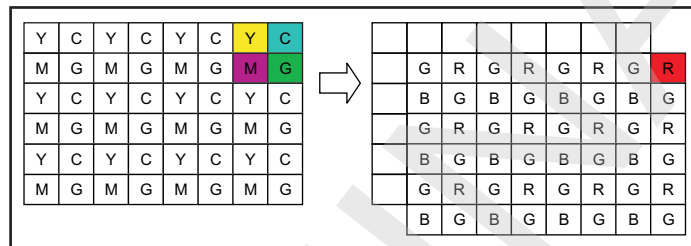
As shown in [Figure 8-201](#) through [Figure 8-204](#), the operation for the last pixel and the second last pixel uses the same input data.

**Figure 8-201. ISS ISP ISIF 2nd Last Pixel/1st Line Generation**



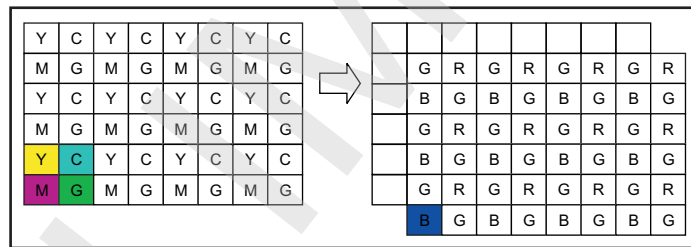
camss-117

**Figure 8-202. ISS ISP ISIF Last Pixel/1st Line Generation**



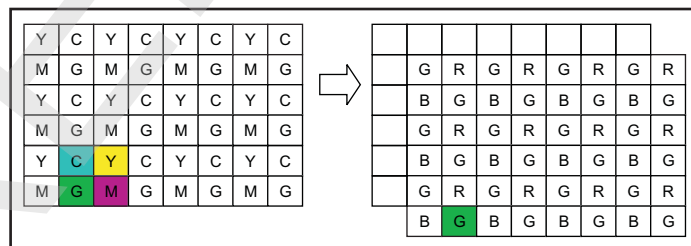
camss-118

**Figure 8-203. ISS ISP ISIF 1st Pixel/Last Line Generation**



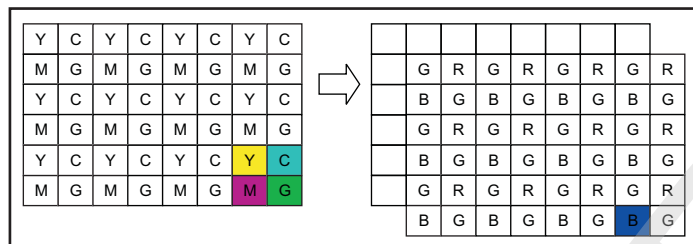
camss-119

**Figure 8-204. ISS ISP ISIF 2nd Pixel/Last Line Generation**

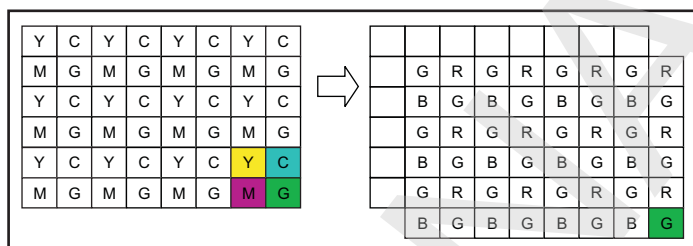


camss-120

Also, the operation for the last line and the second last line uses the same input data (see [Figure 8-205](#) and [Figure 8-206](#)).

**Figure 8-205. ISS ISP ISIF 2nd Last Pixel/Last Line Generation**

camss-121

**Figure 8-206. ISS ISP ISIF Last Pixel/Last Line Generation**

camss-122

In addition to the registers specific to the color space converter, some of the registers are shared with the input data formatter to configure the valid area:

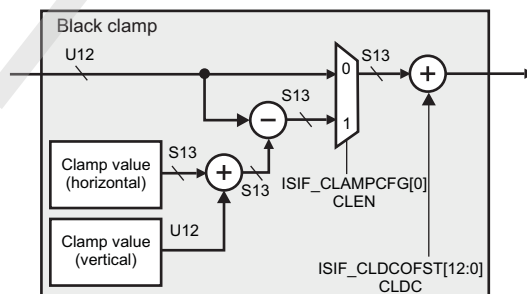
- [ISIF\\_FMTSPH](#)
- [ISIF\\_FMTLNH](#)
- [ISIF\\_FMTLSV](#)
- [ISIF\\_FMTLNV](#)

There must be at least 1 invalid pixel at the end of the line and one invalid line at the end of the frame.

To enable the color space conversion, set the [ISIF\\_CSCCTL\[0\]](#) CSCEN bit to 1.

### 8.3.3.7.8 ISS ISP ISIF Black Clamp

**NOTE:** For the memory access locations of the ISIF clamp, see [Section 8.3.3.9](#).

**Figure 8-207. ISS ISP ISIF Black Clamp Block Diagram**

camss-630

The clamp value is calculated based on the pixel value of the OB region of the sensor. The clamp value is calculated separately for horizontal and vertical directions to compensate the offset drift in both horizontal and vertical directions. The sum of the horizontal and vertical clamp values is subtracted from the image data, and then the additional DC offset is added (the [ISIF\\_CLDCOFST\[12:0\]](#) CLDC bit field, a S13Q0 value). This value is added whether the black clamp module is enabled or not.

The horizontal clamp is disabled through the [ISIF\\_CLAMPCFG\[2:1\]](#) CLHMD bit field.

To enable the black clamp module, set the **ISIF\_CLAMPCFG[0]** CLEN bit to 1. The **ISIF\_PPLN[15:0]** PPLN bit field sets the pixel per line, and the number of pixel clock periods in one line HD period equals PPLN + 1 pixel clock. The **ISIF\_PPLN[15:0]** bit field is not used when the input is already HD/VD.

### 8.3.3.7.8.1 ISS ISP ISIF Clamp Value for Horizontal Direction

The clamp value for horizontal direction is calculated using the pixel values at the upper OB region.

The maximum pixel value to be used for the clamp value calculation can be limited to 1023 if the pixel value limitation is enabled (**ISIF\_CLHWIN0[6]** CLHLMT = 1).

Clamp value calculation for horizontal direction can be disabled in case there is no upper OB region. The operating modes are:

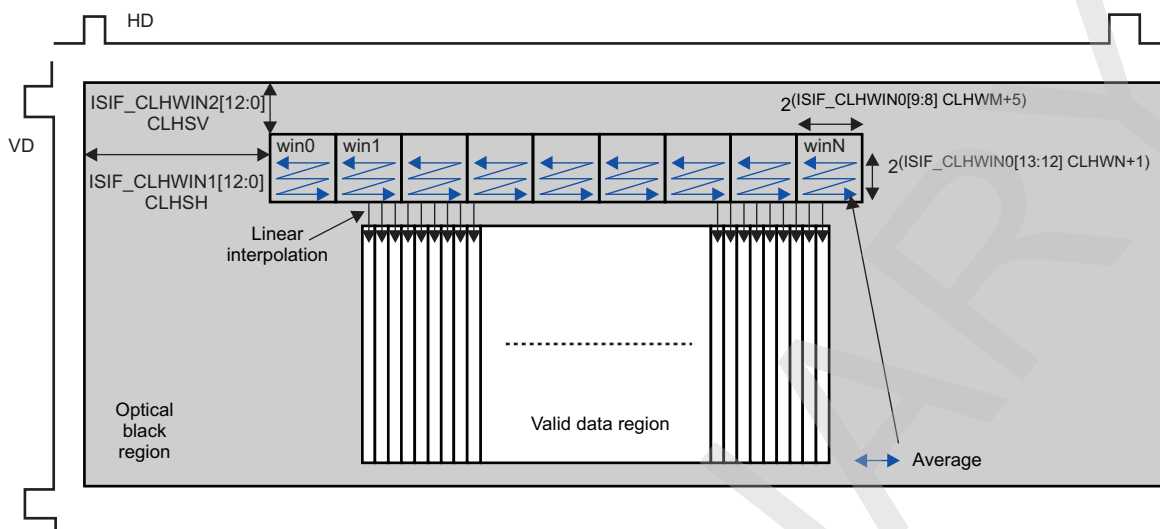
- Horizontal clamp value calculation is enabled. The calculated horizontal clamp value is subtracted from the image data along with the vertical clamp value (**ISIF\_CLAMPCFG[2:1]** CLHMD = 0x1).
- Horizontal clamp value is not updated. The horizontal clamp value used for the previous image is subtracted from the image data along with the vertical clamp value (**ISIF\_CLAMPCFG[2:1]** CLHMD = 0x2).
- Horizontal clamp value is not updated. Only the vertical clamp value is subtracted from the image data (**ISIF\_CLAMPCFG[2:1]** CLHMD = 0x0).

The number of windows in a row is set with the **ISIF\_CLHWIN0[4:0]** CLHWC bit field.

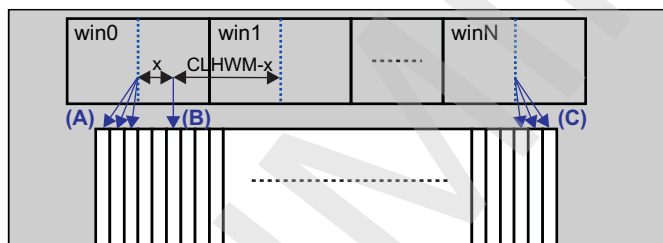
Up to 32 windows in a row can be set for clamp value calculation. All the windows have the same size in a format  $[2^{(\text{ISIF\_CLHWIN0}[9:8] \text{ CLHWM}+5)}]$  pixels by  $[2^{(\text{ISIF\_CLHWIN0}[13:12] \text{ CLHWN}+1)}]$  lines.

The **ISIF\_CLHWIN2[12:0]** CLHSV and **ISIF\_CLHWIN1[12:0]** CLHSH bit fields enable setting the position of the first optical black clamp window in the frame. The pixel and line offset are in a range [0:8191]. The **ISIF\_HDW** register sets the width of the HD.

Figure 8-208. ISS ISP ISIF Clamp Value for Horizontal Direction



Windows settings details



camss-132

The clamp value for horizontal direction calculation steps are:

1. Calculate the average of the pixel value in each window (ave\_win<sub>0</sub> to ave\_win<sub>N</sub>).

Calculation Steps	win0	win1	win2	...	wini	wini + 1	...	winN
Average of the pixel value	ave_win <sub>0</sub>	ave_win <sub>1</sub>	ave_win <sub>2</sub>		ave_win <sub>i</sub>	ave_win <sub>i+1</sub>		ave_win <sub>N</sub>

2. Set the average of the left-most window or the right-most window as the base value B<sub>V</sub>:

- B<sub>V</sub> = ave\_win<sub>0</sub> (if ISIF\_CLHWIN0[5] CLHWBS = 0x0, case 1)
- B<sub>V</sub> = ave\_win<sub>N</sub> (if ISIF\_CLHWIN0[5] CLHWBS = 0x1, case 2)

3. Subtract the base value from the average of each window. Use this value as a clamp value for each window.

Calculation Steps	win0	...	wini	wini+1	...	winN
Clamp value for each window	clamp_win <sub>0</sub> = ave_win <sub>0</sub> - B <sub>V</sub>		clamp_win <sub>i</sub> = ave_win <sub>i</sub> - B <sub>V</sub>	clamp_win <sub>i+1</sub> = ave_win <sub>i+1</sub> - B <sub>V</sub>		clamp_win <sub>N</sub> = ave_win <sub>N</sub> - B <sub>V</sub>

4. Acquire the horizontal distance (X and CLHWM - X) from the valid pixel to be processed to the center of the closest two windows.
5. Calculate the clamp value of the valid pixel by linear interpolation, using the clamp value of the closest two windows (i and i + 1).
  - Case 1: interpolated\_clamp\_win<sub>X</sub> = (clamp\_win<sub>i+1</sub> - clamp\_win<sub>i</sub>) \* X / CLHWM
  - Case 2: interpolated\_clamp\_win<sub>X</sub> = (clamp\_win<sub>i</sub> - clamp\_win<sub>N</sub>) \* (CLHWM - X) / CLHWM + (clamp\_win<sub>i+1</sub> - clamp\_win<sub>N</sub>) \* X / CLHWM
6. If the valid pixel is on the left of the center of the left-most window, the clamp value of the left-most

window is applied. If the valid pixel is on the right of the center of the right-most window, the clamp value of the right-most window is applied.

The clamp values calculated (A), (B) and (C) are:

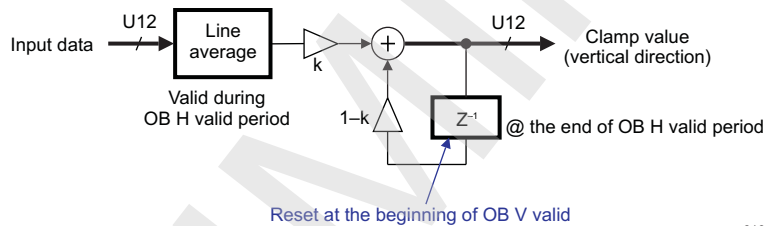
- Case1: Base is win0 (Left-most window: `ISIF_CLHWIN0[5]` CLHWBS = 0x0)
  - (A): zero
  - (B):  $(\text{clamp\_win}_{i+1} - \text{clamp\_win}_i) * X / \text{CLHWM}$
  - (C):  $(\text{clamp\_win}_N - \text{clamp\_win}_0)$
- Case2: Base is winN (Right-most window: `ISIF_CLHWIN0[5]` CLHWBS = 0x1)
  - (A):  $(\text{clamp\_win}_0 - \text{clamp\_win}_N)$
  - (B):  $(\text{clamp\_win}_i - \text{clamp\_win}_N) * (\text{CLHWM} - X) / \text{CLHWM} + (\text{clamp\_win}_{i+1} - \text{clamp\_win}_N) * X / \text{CLHWM}$
  - (C): zero

Each interpolated value `interpolated_clamp_winx` is then subtracted to the associated column.

### 8.3.3.7.8.2 ISS ISP ISIF Clamp Value for Vertical Direction

The clamp value for vertical direction is calculated using the pixel values at the left or right OB region. Line average is calculated for the OB H valid period ( $2^{(\text{ISIF\_CLVWIN0}[2:0] \text{ CLVOBH} + 1)}$ ). The averages for the previous lines are also added back to reduce the difference between the lines, as shown in [Figure 8-209](#).

**Figure 8-209. ISS ISP ISIF Clamp Value for Vertical Direction Calculation**



- Clamp Value ( $V_n$ ) = Line Average ( $V_n$ ) \* k + Clamp Value ( $V_1$ ) \* (1-k)
- $k = \text{ISIF\_CLVWIN0}[15:8] \text{ CLVCOEF}$

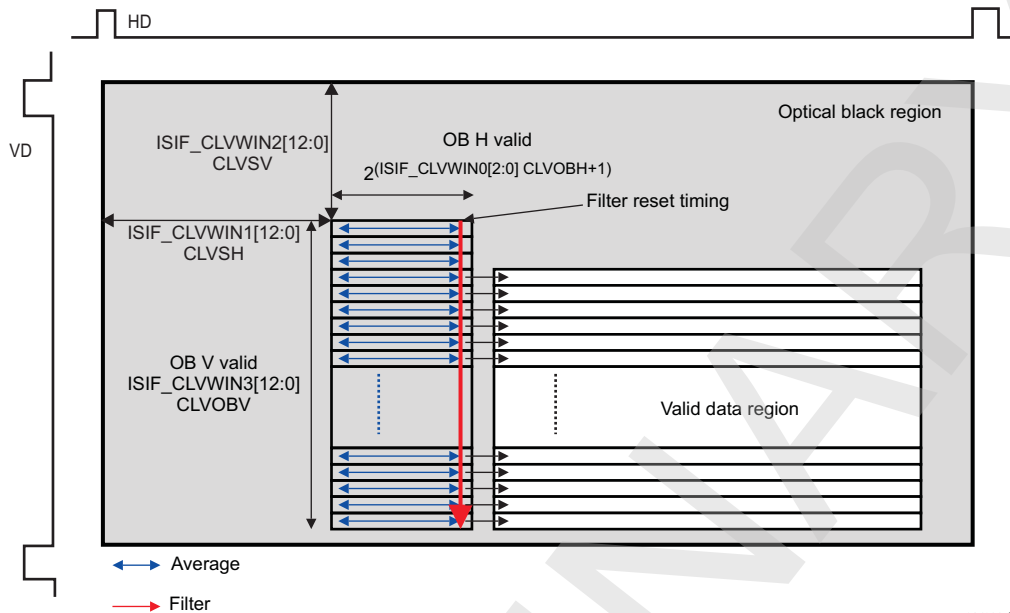
The position of the first vertical black clamp window is set with the `ISIF_CLVWIN2[12:0]` CLVSV and `ISIF_CLVWIN1[12:0]` CLVSH bit fields. The number of vertical windows is set with the `ISIF_CLVWIN3[12:0]` CLVOBV bit field. `ISIF_VDW` sets the width of the VD.

The accumulator, which holds the vertical clamp value for the previous line, is reset at the beginning of the OB V valid. The reset value can be selected through the `ISIF_CLVWIN0[5:4]` CLVRVSL bit field:

- `ISIF_CLVWIN0[5:4]` CLVRVSL = 0x0: The base value is calculated for horizontal direction (left-most window win0 or right-most winn set with `ISIF_CLHWIN0[5]` CLHWBS).
- `ISIF_CLVWIN0[5:4]` CLVRVSL = 0x1: The base value is set through the configuration register (`ISIF_CLVRV[11:0]` CLVRV).
- `ISIF_CLVWIN0[5:4]` CLVRVSL = 0x2: No update (same as the previous image).

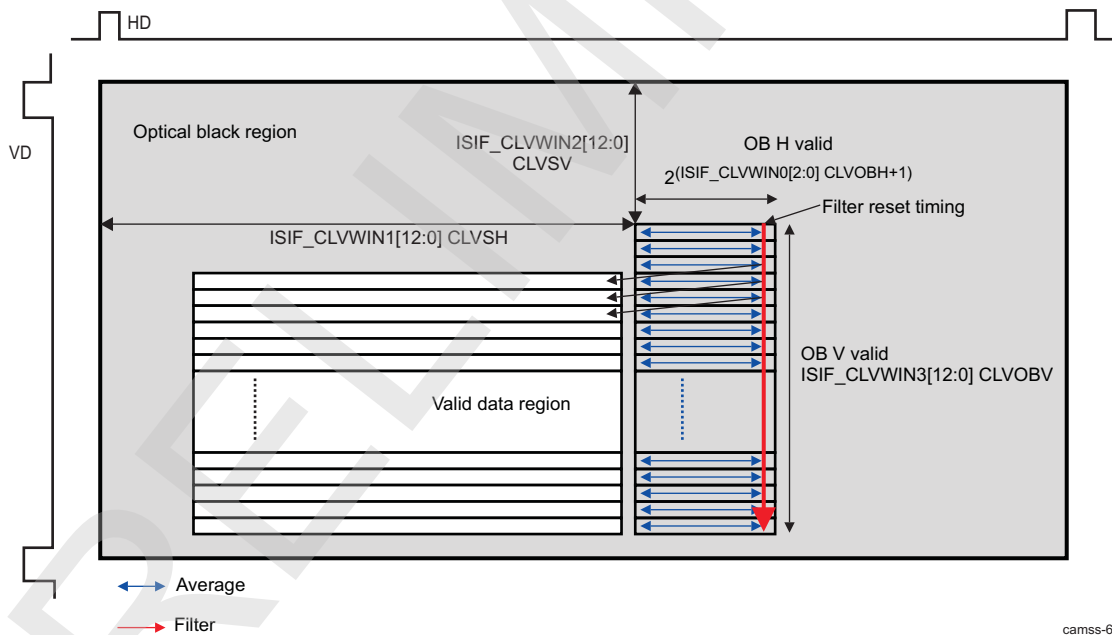
The following figures show the OB valid settings and associated vertical clamp value calculation when OB region is at the left (see [Figure 8-210](#)) and when OB region is at the right (see [Figure 8-211](#)). Each line average value is subtracted from the associated line valid region data.

**Figure 8-210. ISS ISP ISIF Clamp Value for Vertical Direction With OB Region at the Left**



camss-133

**Figure 8-211. ISS ISP ISIF Clamp Value for Vertical Direction With OB Region at the Right**

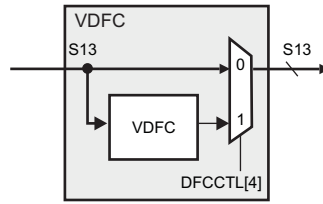


camss-620

**8.3.3.7.9 ISS ISP ISIF Vertical Line Defect Correction (VDFC)**

Figure 8-212 shows the block diagram of the vertical line defect (VDFC) correction.

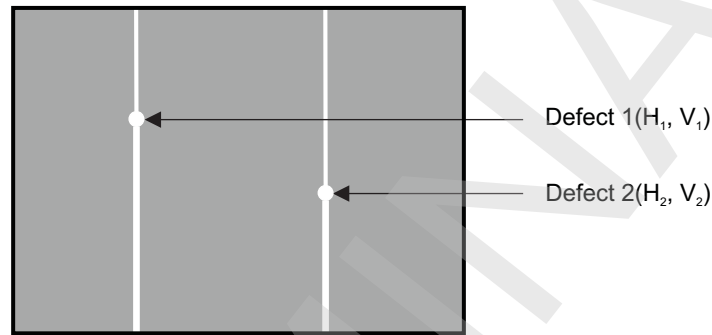
Figure 8-212. ISS ISP ISIF Vertical Line Defect Correction Block Diagram



camss-631

The VDFC block can correct up to eight vertical line defects (see Figure 8-213).

Figure 8-213. ISS ISP ISIF Vertical Line Defects



camss-134

The correction method is common to all the defects and can be selected through the `ISIF_DFCCTL[6:5]` `VDFCSL` bit field.

There are two different methods to correct vertical line defects (`ISIF_DFCCTL[6:5]` `VDFCSL`):

- Method 1: Data is replaced by an average (`ISIF_DFCCTL[6:5]` `VDFCSL` = 0x2):
  - The defect is replaced by the average of pixel (i - 2) and pixel (i + 2)
- Method 2: Data is subtracted by a defect level (`ISIF_DFCCTL[6:5]` `VDFCSL` = 0x0 or 0x1):
  - A saturation level is defined in the `ISIF_VDFSATLV[11:0]` `VDFSLV` bit field.
  - The coordinates of the defect:
    - Are defined in the `ISIF_DFCMEM0[12:0]` `DFCMEM0` and `ISIF_DFCMEM1[12:0]` `DFCMEM1` bit fields
    - Are 13 bits wide for horizontal and vertical direction, so an image size up to 8192 x 8192 is supported
  - If the data is not saturated (data < `VDFSLV`):
    - The defect is corrected by subtracting the defect level. A different defect level is defined for:
      - The point of the defect (V = `Vdefect`): `SUB1` defect level is defined in the `ISIF_DFCMEM2[7:0]` `DFCMEM2` bit field.
      - The pixels lower than the defect (V < `Vdefect`): `SUB2` defect level is defined in the `ISIF_DFCMEM3[7:0]` `DFCMEM3` bit field.
      - The pixels upper than the defect (V > `Vdefect`): Defect level is defined in the `ISIF_DFCMEM4[7:0]` `DFCMEM4` bit field.
    - Each defect level (value to be subtracted from the data) described previously can be up-shifted through the `ISIF_DFCCTL[10:8]` `VDFLSFT` bit field.
    - Vertical line defect correction for upper pixels can be disabled through the `ISIF_DFCCTL[7]` `VDFCUDA` bit.
  - If the data is saturated (`VDFSLV`), there are two possibilities:
    - `ISIF_DFCCTL[6:5]` `VDFCSL` = 0x0: Data is simply fed through (not subtracted)
    - `ISIF_DFCCTL[6:5]` `VDFCSL` = 0x1: Horizontal interpolation  $((i - 2) + (i + 2))/2$  (data is replaced by interpolation or data is subtracted with interpolation)

The [ISIF\\_LPFR](#) register sets the number of half lines per frame or field: VD period = (L PFR+ 1)/2 lines. LPFR is not used when HD and VD are inputs.

The following paragraphs concern only method 2 correction.

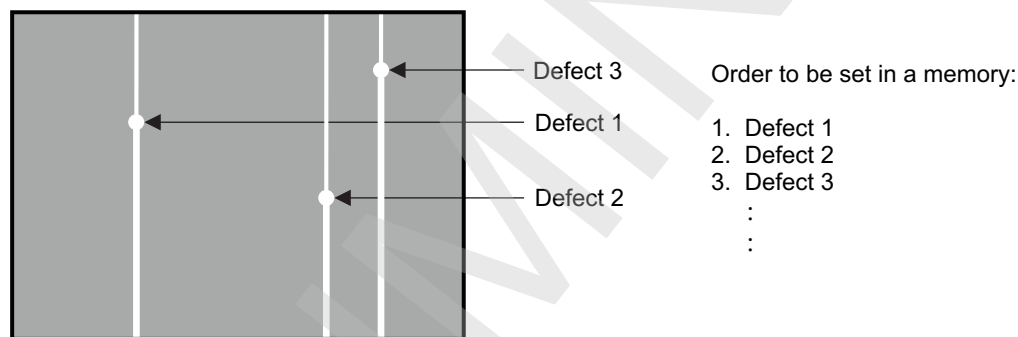
The coordinates of the defects and the defect levels to be subtracted from the data must be set to the processing shown in [Table 8-340](#).

**Table 8-340. ISS ISP ISIF Vertical Line Defect Table in Memory**

Bit	Defect Information
12:0	Vertical position of the defects
25:13	Horizontal position of the defects
33:26	Defect level of the vertical line defect position ( $V = V_{\text{defect}}$ )
41:34	Defect level of the pixels upper than the vertical line defect ( $V < V_{\text{defect}}$ )
49:42	Defect level of the pixels lower than the vertical line defect ( $V > V_{\text{defect}}$ )

The defect must be set from left to right, as shown in [Figure 8-214](#).

**Figure 8-214. ISS ISP ISIF Vertical Line Defects**



camss-135

Vertical line defect correction is enable by setting the [ISIF\\_DFCCTL\[4\]](#) VDFCEN bit to 1, but the procedure explained in [Section 8.3.3.7.9.1](#), *ISS ISP ISIF Vertical Line Defect Table Update Procedure*, must be respected.

#### 8.3.3.7.9.1 ISS ISP ISIF Vertical Line Defect Table Update Procedure

The following procedure must be respected to write the vertical line defect table in memory.

1. [ISIF\\_DFCMEMCTL\[4\]](#) DFCMCLR = 0x1
2. Ensure that [ISIF\\_DFCCTL\[4\]](#) VDFCEN is disabled (0x0).
3. Write the V coordinate of the first defect to the [ISIF\\_DFCMEM0\[12:0\]](#) DFCMEM0 bit field.
4. Write the H coordinate of the first defect to the [ISIF\\_DFCMEM1\[12:0\]](#) DFCMEM1 bit field.
5. Set the defect level to:
  - [ISIF\\_DFCMEM2\[7:0\]](#) DFCMEM2
  - [ISIF\\_DFCMEM3\[7:0\]](#) DFCMEM3
  - [ISIF\\_DFCMEM4\[7:0\]](#) DFCMEM4
6. Set the [ISIF\\_DFCMEMCTL\[0\]](#) DFCMWR bit to 1 with the [ISIF\\_DFCMEMCTL\[2\]](#) DFCMARST bit set to 1.
7. Wait until the [ISIF\\_DFCMEMCTL\[0\]](#) DFCMWR bit is cleared.
8. Write the next data to:
  - [ISIF\\_DFCMEM0\[12:0\]](#) DFCMEM0
  - [ISIF\\_DFCMEM1\[12:0\]](#) DFCMEM1
  - [ISIF\\_DFCMEM2\[7:0\]](#) DFCMEM2
  - [ISIF\\_DFCMEM3\[7:0\]](#) DFCMEM3

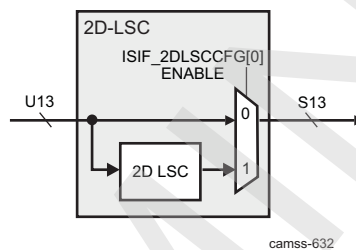


- [ISIF\\_DFCMEM4\[7:0\]](#) DFCMEM4
9. Set the [ISIF\\_DFCMEMCTL\[0\]](#) DFCMWR bit to 1 with the [ISIF\\_DFCMEMCTL\[2\]](#) DFCMARST bit cleared.
  10. Repeat 4~5 times until all entries (up to 8) are written to the vertical line defect table.
  11. In case the defect entry is less than 8, an extra write cycle is required to fill the next table location with a certain value.
  12. Clear the [ISIF\\_DFCMEM0\[12:0\]](#) DFCMEM0 bit field to all 0, set the [ISIF\\_DFCMEM1\[12:0\]](#) DFCMEM1 to all 1, and set the [ISIF\\_DFCMEMCTL\[0\]](#) DFCMWR bit to 1 with the [ISIF\\_DFCMEMCTL\[2\]](#) DFCMARST bit cleared.
  13. Enable VDFC by setting the [ISIF\\_DFCCTL\[1\]](#) VDFCEN bit.

### 8.3.3.7.10 ISS ISP ISIF Lens Shading Correction (2D-LSC)

**NOTE:** For the memory access locations of the 2D-LSC table, see [Section 8.3.3.9](#).

**Figure 8-215. ISS ISP ISIF 2D-LSC Block Diagram**



LSC is useful for correcting optical artifacts that cause image brightness to decrease starting from the center of the image and going out to the edges.

The LSC module implements a per pixel offset and gain adjustment in the RAW Bayer domain (2 × 2 color pattern). The offset is applied before gain multiplication.

The offset and gains are stored in a LUT, which is stored in SDRAM and is loaded in real time. The submodule prefetches the data from SDRAM such that no underflow occurs. Underflow occurs when the offset and gain data required for the current pixel are not available.

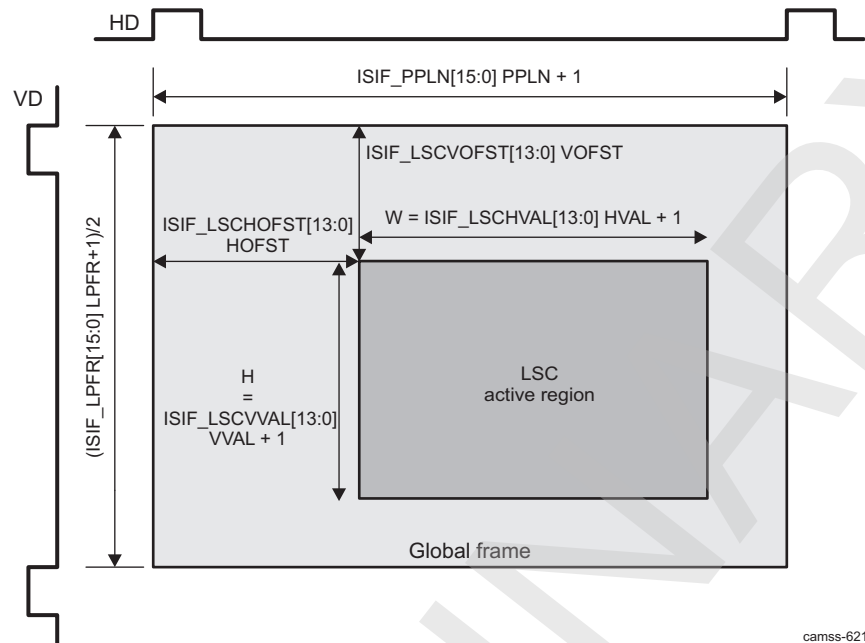
The data stored in the LUT is downsampled; that is, there is no gain or offset per pixel. The downsampling factor is programmable. High downsampling ratios lead to a smaller LUT, lower accuracy, and lower memory bandwidth. A low downsampling ratio leads to a bigger LUT, higher accuracy, and higher memory bandwidth.

When the offset and gain values are loaded, they are upsampled to the incoming image resolution. The missing table values are computed through bilinear interpolation.

To enable the 2D-LSC module, set the [ISIF\\_2DLSCCFG\[0\]](#) ENABLE bit to 1.

#### 8.3.3.7.10.1 ISS ISP ISIF 2D-LSC Active Region Settings

The gain and offset maps are internally up-sampled to full resolution before being applied to the image. In order to account for all the possible cropping schemes and zoom ratios, the 2D-LSC can be configured such that a single gain map can be stored in memory that maps to sensor lens. The 2D-LSC active region is defined by [Figure 8-216](#).

**Figure 8-216. ISS ISP ISIF 2D-LSC Active Region for ISIF Input Frame**

camss-621

#### 8.3.3.7.10.1.1 ISS ISP ISIF 2D-LSC Gain and Offset Tables

The gain and offset map are MxN downsampled:

- M is the horizontal sampling factor.
- N is the vertical sampling factor.
- M and N are {8, 16, 32, 64, 128} independently.
- $N \leq M$ . M is set in the [ISIF\\_2DLSCCFG\[14:12\]](#) GAIN\_MODE\_M bit field.
- N is set in the [ISIF\\_2DLSCCFG\[10:8\]](#) GAIN\_MODE\_N bit field.

The starting point of the preconfigured lens shading map can be modified in software to align with the ISIF input image frame. The location of the gain and offset mask data in memory is specified by :

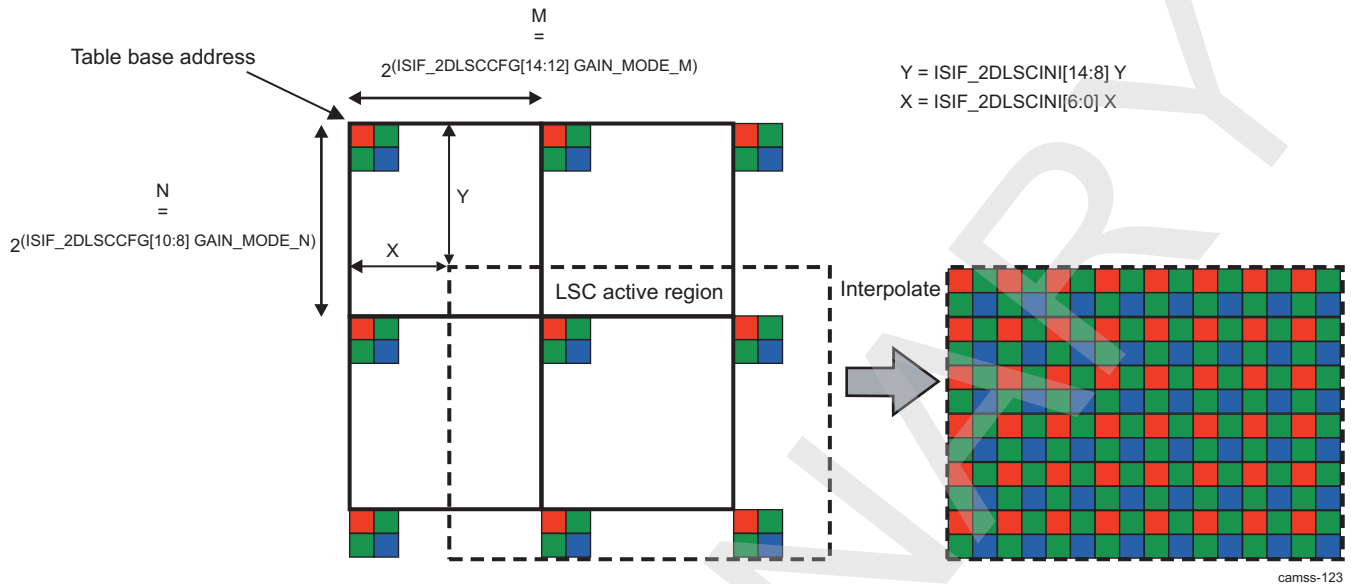
- For gain map:
  - Table base address: [ISIF\\_2DLSCGRBU\[15:0\]](#) BASE31\_16 and [ISIF\\_2DLSCGRBL\[15:0\]](#) BASE15\_0
- For offset map:
  - Table base address: [ISIF\\_2DLSCORBU\[15:0\]](#) BASE and [ISIF\\_2DLSCORBL\[15:0\]](#) BASE

The input address must be aligned to a 4-byte boundary.

In the full resolution case, the address is set to the beginning of the map. When the LSC active region is defined over a cropped region of the full image, the SDRAM input address can be set to the upper-left corner of the grid closest to the beginning of the active region, and the [ISIF\\_2DLSCINI\[6:0\]](#) X and [ISIF\\_2DLSCINI\[14:8\]](#) Y bit fields mark the offset into the upsampled gain and offset map where the active region begins. [Figure 8-217](#) shows the LSC active region with respect to the gain and offset map grid. Because (X, Y) deals with the pixel phase inside a gain and offset map grid, X and Y must each be less than M and N, respectively.

Before applying the gain or offset table, the table is internally upsampled and interpolated back to full resolution via bilinear interpolation.

**Figure 8-217. ISS ISP ISIF Gain and Offset Mask Upsampling via Bilinear Interpolation**



The gain table format is set in the [ISIF\\_2DLSCCFG\[3:1\] GAIN\\_FORMAT](#) bit field. 8-bit entries are supported in the gain map (in U8Q8, U8Q7, U8Q6, and U8Q5 format with optional base of 1.0 to shift the range up).

8-bit entries are supported in the offset map (in S8Q0 format). An optional shifting up for offsets is possible. The shift up value is selected through the [ISIF\\_2DLSCOFST\[6:4\] OFSTSFT](#) bit field. A scaling factor for offsets is defined in the [ISIF\\_2DLSCOFST\[15:8\] OFSTSF](#) bit field.

The offset control in the 2D-LSC module can be enabled or disabled through the [ISIF\\_2DLSCOFST\[0\] OFSTEN](#) bit.

For an LSC active region size of  $W \times H$  (the output and input are the same size), and the gain and offset maps are  $M \times N$  downsampled, a table with the following values is needed:

- $(\text{ceil}[(X + W) / M] + 1) \times (\text{ceil}[(Y + H) / N] + 1) \times 4$  bytes data in external memory organized as:
    - $(\text{ceil}[(\text{Init}X + W) / M] + 1)$  lines of data
    - Each line having at least  $(\text{ceil}[(\text{Init}Y + H) / N] + 1) \times 4$  data points
- Extra data at end of each line can be skipped by the line offset register parameter:
- For gain map:
    - Line offset: [ISIF\\_2DLSCGROF\[15:0\] OFFSET](#)
  - For offset map:
    - Line offset: [ISIF\\_2DLSCOROF\[15:0\] OFFSET](#)
  - Each line offset must start at a 32-bit aligned boundary.

### 8.3.3.7.10.1.2 ISS ISP ISIF 2D-LSC Gain and Offset Table Upsampling

Upsampling of the pixel-by-pixel gains is performed by locating the four same-color anchors for each destination gain value and applying bilinear interpolation.

The gain and offset mask function is neutral to color pattern. The starting color of the gain and offset mask must be consistent with the starting color of the image, and can be any color. To align starting colors, the X and Y values must be even. The 2D-LSC engine upsamples each phase of the mask data as a separate plane and applies the upsampled mask to the image with the same color phasing. In other words, the red gains are interpolated with red gains, and applied to the red input pixels. The same is done for each of the other three colors in the color pattern. The 2D-LSC module is designed to work with Bayer CFA data, having the R/Gr/Gb/B color pattern. For the purpose of functional description, assume red is the starting color, but any other starting color or other  $2 \times 2$  pattern can be used by placing color gains in the appropriate order.

### 8.3.3.7.10.1.3 ISS ISP ISIF Application of Gain and Offset to Image Pixels

The gain value interpolated for each pixel is multiplied with a corresponding input pixel. An offset is applied before the gain. The product is rounded to the nearest integer and then clipped or saturated to the valid range of 13 bits.

The following equation describes the operation of the LSC in terms of offset and gain:

$$\text{out}[x,y] = (\text{in}[x,y] + a * (\text{ofst}[x,y] \ll T)) * \text{gain}[x,y] \quad (6)$$

- $\text{in}[x, y]$  are the input pixels, 13 bits signed.
- $\text{ofst}[x, y]$  are the upsampled offset points.
- $T$  is the upshift value applied to the result of the offset interpolation points (0 – 5) set through the [ISIF\\_2DLSCOFST\[6:4\]](#) OFSTSFT bit field.
- $a$  is the offset gain value in U8Q7 format set through the [ISIF\\_2DLSCOFST\[15:8\]](#) OFSTSF bit field.
- $\text{gain}[x, y]$  are the upsampled gain points.
- $\text{out}[x, y]$  are the resulting output pixels, 13 bits signed.

### 8.3.3.7.10.1.4 ISS ISP ISIF Enabling/ Disabling the 2D-LSC Module

LSC operates on a single frame or continuously, depending on the firmware programming.

Upon power-on reset (POR), the 2D-LSC module is disabled and input pixels are copied to the output, bypassing any shading operation.

When enabling or disabling the 2D-LSC, caution must be taken on the timing of register modifications. To avoid causing a prefetch error or other unexpected behavior, the following safeguards must be implemented:

1. While configuring the 2D-LSC registers, the input clock into the ISIF should be toggling.
2. All of the 2D-LSC registers must be configured appropriately before enabling the [ISIF\\_2DLSCCFG\[0\]](#) ENABLE bit.
3. After setting the ENABLE bit to 1, the hardware immediately begins fetching the first two rows of gain and offset data entries from external memory. When this is complete, the [ISIF\\_2DLSCIRQST\[2\]](#) PREFETCH\_COMPETED status flag is set.

---

**NOTE:** If the ENABLE bit is disabled before the [ISIF\\_2DLSCIRQST\[3\]](#) SOF status flag is set, the [ISIF\\_2DLSCIRQST\[1\]](#) PREFETCH\_ERROR flag is set and the state of the 2D-LSC submodule may lead to unexpected errors. Therefore, the ENABLE bit must not be disabled until after the [ISIF\\_2DLSCIRQST\[3\]](#) SOF status flag is set.

---

4. Appropriate gains and offsets are applied to the image pixels. Pixels outside the LSC active region are passed through unaltered. When the 2D-LSC operation on the active region completes, the [ISIF\\_2DLSCIRQST\[0\]](#) DONE status flag is set.
5. At this point:
  - If the ENABLE bit is still set to 1, the hardware immediately begins to prefetch the gain and offset data entries for the next frame and waits for the active region of the next frame to arrive.
  - If the ENABLE bit is set to 0, it stops LSC operation once the active region is passed, and goes into idle until the ENABLE bit is written to 1 again.

---

**NOTE:** To provide a mechanism for firmware to recover from the LSC module waiting indefinitely for the input image, if LSC\_ENABLE is written to 0 after it has started gain/offset map prefetching, but before the LSC gets to the next active region, the LSC operation is aborted and turned idle, and any prefetched gain/offset entries are discarded. This can happen before or after the next start-of-frame.

---

**NOTE:** Therefore, because of the constraints set in point 3, the ENABLE register bit must be disabled only after the `ISIF_2DLSCIRQST[3]` SOF status flag is set and before the `ISIF_2DLSCIRQST[0]` DONE status flag is set for that same frame.

It is suggested that when the 2D-LSC or the whole ISIF needs to be disabled for switching modes, the `ISIF_2DLSCIRQST[3]` SOF interrupt be enabled so that software knows when it is safe to disable the 2D-LSC. Then the ISIF can be disabled after the `ISIF_2DLSCIRQST[0]` DONE status signal is set for that frame.

**NOTE:** The LSC\_ENABLE bit, once written to 1, must not be cleared until at least one `vpi_clk` clock cycle after start-of-frame, to ensure correct processing.

### 8.3.3.7.10.1.5 ISS ISP ISIF 2D-LSC Events and Status Checking

The 2D-LSC submodule can generate events on a single interrupt line. These events are further remapped at the ISP level in `ISP5_IRQENABLE_SET__0.ISIF_INT_3` - `ISP5_IRQENABLE_SET__3.ISIF_INT3`.

Four 2D-LSC events can be generated:

- **DONE:** LSC done. This event triggers when the LSC submodules transition from ACTIVE state to IDLE state.
- **PREFETCH\_ERROR:** Gain table prefetch error. This event triggers when the tables stored in SDRAM are read too slowly. After this event is asserted, the LSC disables the LSC computation until the beginning of the next frame.
- **PREFETCH\_COMPLETE:** Gain table prefetch complete. This event triggers when data prefetching from SDRAM completes. Data prefetching must complete by the time the first pixel of a frame comes. The event triggers when the buffer contains three full rows of data.
- **SOF:** This event signals the start of the LSC valid region. The LSC configuration registers for the next frame can be updated after the LSC SOF triggers.

The `ISIF_2DLSCIRQEN` register can be configured to select which events are masked and which are propagated to the LSC interrupt signal. The `ISIF_2DLSCIRQST` register can be read and cleared to identify which events have occurred.

In addition, the 2D-LSC module provides the following status bit:

- **BUSY:** This indicates that LSC has entered the active region vertically. This bit remains on during horizontal blanking, and turns off only after the entire active region of the current frame is processed.

### 8.3.3.7.10.1.6 ISS ISP ISIF Supported On-the-Fly 2D-LSC Configurations

The 2D-LSC prefetch memory is equal to  $2 \times 1536 \times 32$  bits. This memory is sized to fetch three lines of 8-bit gain and 8-bit offset  $\times$  four color components per paxel. Given an image sensor of horizontal resolution H, there are  $\text{floor}[(H / \text{ISIF\_2DLSCCFG}[14:12] \text{ GAIN\_MODE\_M}) + 1]$  paxels per line, where M is the horizontal LSC paxel size.

Table 8-341 shows the LSC horizontal paxel size, which can be supported for different image sensor resolutions. When  $M = 8$ , some resolutions cannot be supported on the fly (orange-shaded cells in the table); the way to process such large images is to use vertical frame division.

**Table 8-341. ISS ISP ISIF Supported On-the-Fly LSC Configurations**

MPix	Aspect Ratio		Line Size	Horizontal LSC Paxel Size: $M = 2^{\text{ISIF\_2DLSCCFG}[14:12] \text{ GAIN\_MODE\_M}}$				
				8	16	32	64	128
Maximum	–	–	5376	2019	1011	507	255	129
16	16	9	5333	2003	1003	503	253	128
16	4	3	4619	1735	869	436	220	111
16	3	2	4899	1840	922	462	233	118
12	16	9	4619	1735	869	436	220	111

**Table 8-341. ISS ISP ISIF Supported On-the-Fly LSC Configurations (continued)**

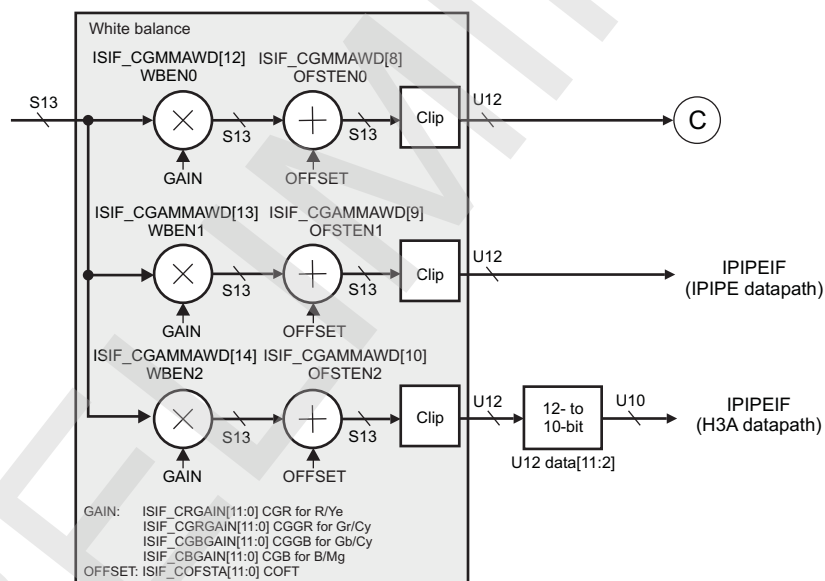
MPix	Aspect Ratio		Line Size	Horizontal LSC Poxel Size: $M = 2^{\lfloor \text{ISIF\_2DLSCCFG}[14:12] \text{ GAIN\_MODE\_M} \rfloor}$				
				8	16	32	64	128
12	4	3	4000	1503	753	378	191	97
12	3	2	4243	1594	798	401	202	102
10	16	9	4216	1584	794	398	201	102
10	4	3	3651	1372	688	345	174	89
10	3	2	3873	1455	729	366	185	94
8	16	9	3771	1417	710	357	180	91
8	4	3	3266	1228	615	309	156	80
8	3	2	3464	1302	653	328	165	84

**8.3.3.7.10.1.7 ISS ISP ISIF Bandwidth Requirements on BL Read Port**

See [Section 8.3.3.7.18.2](#) for details.

**8.3.3.7.11 ISS ISP ISIF White Balance**

[Figure 8-218](#) shows the white balance block diagram.

**Figure 8-218. ISS ISP ISIF White Balance Block Diagram**

Color pattern settings are set through the [ISIF\\_CCOLP](#) register. Moreover, through this register the pixel position from 0 to 3 can be set to the needed Bayer universal camera filter color pattern (RGB/CYGM).

The CFA pattern can be in two modes, stripe or mosaic, and is set through the [ISIF\\_CGAMMAWD\[5\]](#) CFAP bit.

There are color-dependent gain controls for the three outputs: BL output, IPIPEIF (IPIPE path) output, and IPIPEIF (H3A path) output. Gain applied to each data is selected according to the pixel position and the color pattern settings. Gain factors are common for the three data paths. Gain is in U11Q9 format, which ranges from 0 to  $3 + 511/512$ . The gain factor is set through the following registers:

- R/Ye gain: [ISIF\\_CRGAIN\[11:0\]](#) CGR
- Gr/Cy gain: [ISIF\\_CGRGAIN\[11:0\]](#) CGGR
- Gb/Cy gain: [ISIF\\_CGBGAIN\[11:0\]](#) CGGB
- B/Mg gain: [ISIF\\_CBGAIN\[11:0\]](#) CGB

Gain control can be enabled or disabled individually for each path.

- Enable or disable gain for the BL path: `ISIF_CGAMMAWD[12]` `WBEN0`
- Enable or disable gain for the IPIPEIF (IPIPE) path: `ISIF_CGAMMAWD[13]` `WBEN1`
- Enable or disable gain for the IPIPEIF (H3A) path: `ISIF_CGAMMAWD[14]` `WBEN2`

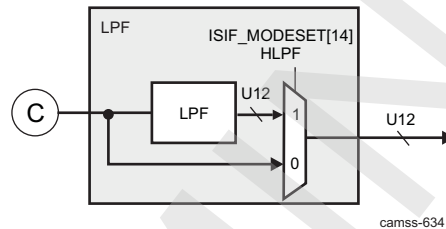
After the gain control, a single offset value can be added to each path individually. This offset is common for the three paths and is set through the `ISIF_COFSTA[11:0]` `COFT` bit field. The offset value is `U12`, which ranges from 0 to 4095. Data (`S13`) are then truncated to `U12`.

- Enable or disable offset for the BL path: `ISIF_CGAMMAWD[8]` `OFSTEN0`
- Enable or disable offset for the IPIPEIF (IPIPE) path: `ISIF_CGAMMAWD[9]` `OFSTEN1`
- Enable or disable offset for the IPIPEIF (H3A) path: `ISIF_CGAMMAWD[10]` `OFSTEN2`

### 8.3.3.7.12 ISS ISP ISIF Low-Pass Filter (LPF)

Figure 8-219 shows the low-pass filter block diagram.

Figure 8-219. ISS ISP ISIF Low-Pass Filter Block Diagram



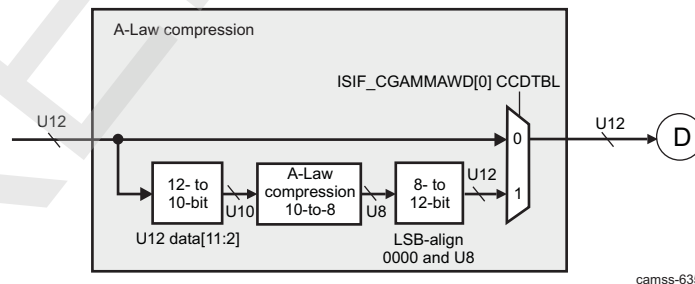
An optional horizontal low-pass anti-aliasing filter (LPF) can be applied (the `ISIF_MODESET[14]` `HLPF` bit) after reframing. The low-pass filter consists of a simple 3-tap (1/4, 1/2, and 1/4) filter. Two pixels on the left and two pixels on the right of each line are cropped if the filter is enabled. Use of the LPF is intended for bandwidth reduction if culling is enabled.

**NOTE:** For YUV data, the LPF must be disabled (`ISIF_MODESET[14]` `HLPF` = 0x0).

### 8.3.3.7.13 ISS ISP ISIF A-Law Compression

Figure 8-220 shows the A-Law compression block diagram.

Figure 8-220. ISS ISP ISIF A-Law Compression Block Diagram



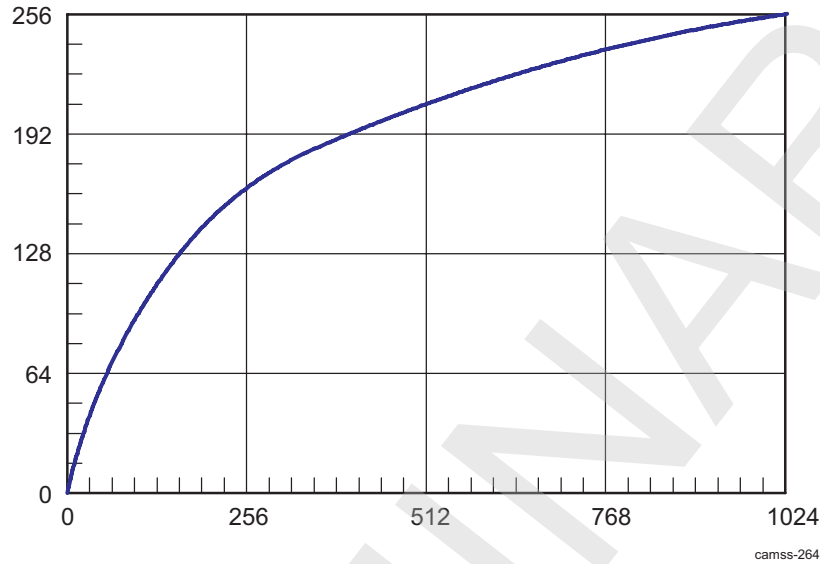
An optional 10-to-8-bit A-Law compression using a fixed A-Law table can be applied (`ISIF_CGAMMAWD[0]` `CCDTBL`) as the final processing stage. Using this compression causes the data width to be reduced to 8 bits and allows packing to 8 bits/pixel when saving to memory. Because data resolution can be greater than 10 bits at this stage, the 10 bits for input to the A-Law operation must be selected (`ISIF_CGAMMAWD[4:1]` `GWDI`).

The IPIPEIF module has an inverse A-Law table (A-Law decompression) option so that this nonlinear operation can be reversed if this saved data is to be read back in for further processing.

**NOTE:** Do not use A-Law compression (ISIF\_CGAMMAWD[0] CCDTBL = 0) with YUV data.

Figure 8-221 shows the A-Law table diagram, and Figure 8-222 shows the A-Law table values.

**Figure 8-221. ISS ISP ISIF A-Law Table Diagram**





**Figure 8-222. ISS ISP ISIF A-Law Table Values**

Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
0	0	64	64	128	112	192	140	256	161	320	176	384	189	448	200
1	1	65	65	129	113	193	141	257	161	321	176	385	189	449	200
2	2	66	66	130	113	194	141	258	161	322	177	386	189	450	200
3	3	67	67	131	114	195	142	259	161	323	177	387	189	451	200
4	4	68	68	132	114	196	142	260	162	324	177	388	190	452	200
5	5	69	69	133	115	197	142	261	162	325	177	389	190	453	200
6	6	70	70	134	115	198	143	262	162	326	177	390	190	454	201
7	7	71	71	135	116	199	143	263	162	327	178	391	190	455	201
8	8	72	72	136	116	200	143	264	163	328	178	392	190	456	201
9	9	73	73	137	117	201	144	265	163	329	178	393	190	457	201
10	10	74	74	138	117	202	144	266	163	330	178	394	191	458	201
11	11	75	75	139	118	203	144	267	163	331	178	395	191	459	201
12	12	76	76	140	118	204	145	268	164	332	179	396	191	460	201
13	13	77	77	141	119	205	145	269	164	333	179	397	191	461	202
14	14	78	78	142	119	206	145	270	164	334	179	398	191	462	202
15	15	79	78	143	120	207	146	271	164	335	179	399	191	463	202
16	16	80	79	144	120	208	146	272	165	336	179	400	192	464	202
17	17	81	80	145	121	209	146	273	165	337	180	401	192	465	202
18	18	82	81	146	121	210	147	274	165	338	180	402	192	466	202
19	19	83	82	147	122	211	147	275	166	339	180	403	192	467	202
20	20	84	83	148	122	212	147	276	166	340	180	404	192	468	203
21	21	85	84	149	123	213	148	277	166	341	181	405	193	469	203
22	22	86	84	150	123	214	148	278	166	342	181	406	193	470	203
23	23	87	85	151	124	215	148	279	167	343	181	407	193	471	203
24	24	88	86	152	124	216	149	280	167	344	181	408	193	472	203
25	25	89	87	153	125	217	149	281	167	345	181	409	193	473	203
26	26	90	88	154	125	218	149	282	167	346	182	410	193	474	204
27	27	91	88	155	125	219	150	283	168	347	182	411	194	475	204
28	28	92	89	156	126	220	150	284	168	348	182	412	194	476	204
29	29	93	90	157	126	221	150	285	168	349	182	413	194	477	204
30	30	94	91	158	127	222	151	286	168	350	182	414	194	478	204
31	31	95	91	159	127	223	151	287	168	351	183	415	194	479	204
32	32	96	92	160	128	224	151	288	169	352	183	416	194	480	204
33	33	97	93	161	128	225	152	289	169	353	183	417	195	481	205
34	34	98	93	162	129	226	152	290	169	354	183	418	195	482	205
35	35	99	94	163	129	227	152	291	169	355	183	419	195	483	205
36	36	100	95	164	129	228	152	292	170	356	184	420	195	484	205
37	37	101	96	165	130	229	153	293	170	357	184	421	195	485	205
38	38	102	96	166	130	230	153	294	170	358	184	422	195	486	205
39	39	103	97	167	131	231	153	295	170	359	184	423	196	487	205
40	40	104	98	168	131	232	154	296	171	360	184	424	196	488	206
41	41	105	98	169	132	233	154	297	171	361	185	425	196	489	206
42	42	106	99	170	132	234	154	298	171	362	185	426	196	490	206
43	43	107	100	171	132	235	155	299	171	363	185	427	196	491	206
44	44	108	100	172	133	236	155	300	172	364	185	428	196	492	206
45	45	109	101	173	133	237	155	301	172	365	185	429	197	493	206
46	46	110	102	174	134	238	155	302	172	366	185	430	197	494	206
47	47	111	102	175	134	239	156	303	172	367	186	431	197	495	207
48	48	112	103	176	134	240	156	304	173	368	186	432	197	496	207
49	49	113	103	177	135	241	156	305	173	369	186	433	197	497	207
50	50	114	104	178	135	242	157	306	173	370	186	434	197	498	207
51	51	115	105	179	136	243	157	307	173	371	186	435	198	499	207
52	52	116	105	180	136	244	157	308	173	372	187	436	198	500	207
53	53	117	106	181	136	245	157	309	174	373	187	437	198	501	207
54	54	118	106	182	137	246	158	310	174	374	187	438	198	502	208
55	55	119	107	183	137	247	158	311	174	375	187	439	198	503	208
56	56	120	108	184	137	248	158	312	174	376	187	440	198	504	208
57	57	121	108	185	138	249	159	313	175	377	188	441	198	505	208
58	58	122	109	186	138	250	159	314	175	378	188	442	199	506	208
59	59	123	109	187	139	251	159	315	175	379	188	443	199	507	208
60	60	124	110	188	139	252	159	316	175	380	188	444	199	508	208
61	61	125	110	189	139	253	160	317	175	381	188	445	199	509	208
62	62	126	111	189	140	254	160	318	176	382	188	446	199	510	209
63	63	127	112	191	140	255	160	319	176	383	189	447	199	511	209

camss-262

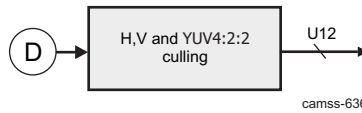
Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law	Input	A-Law
512	209	576	217	640	224	704	231	768	237	832	243	896	248	960	253
513	209	577	217	641	225	705	231	769	237	833	243	897	248	961	253
514	209	578	217	642	225	706	231	770	237	834	243	898	248	962	253
515	209	579	217	643	225	707	231	771	237	835	243	899	248	963	253
516	209	580	218	644	225	708	232	772	238	836	243	900	248	964	253
517	210	581	218	645	225	709	232	773	238	837	243	901	248	965	253
518	210	582	218	646	225	710	232	774	238	838	243	902	248	966	253
519	210	583	218	647	225	711	232	775	238	839	243	903	249	967	253
520	210	584	218	648	225	712	232	776	238	840	243	904	249	968	253
521	210	585	218	649	225	713	232	777	238	841	244	905	249	969	253
522	210	586	218	650	226	714	232	778	238	842	244	906	249	970	254
523	210	587	218	651	226	715	232	779	238	843	244	907	249	971	254
524	211	588	219	652	226	716	232	780	238	844	244	908	249	972	254
525	211	589	219	653	226	717	232	781	238	845	244	909	249	973	254
526	211	590	219	654	226	718	233	782	238	846	244	910	249	974	254
527	211	591	219	655	226	719	233	783	239	847	244	911	249	975	254
528	211	592	219	656	226	720	233	784	239	848	244	912	249	976	254
529	211	593	219	657	226	721	233	785	239	849	244	913	249	977	254
530	211	594	219	658	226	722	233	786	239	850	244	914	249	978	254
531	211	595	219	659	227	723	233	787	239	851	244	915	249	979	254
532	212	596	220	660	227	724	233	788	239	852	244	916	250	980	254
533	212	597	220	661	227	725	233	789	239	853	245	917	250	981	254
534	212	598	220	662	227	726	233	790	239	854	245	918	250	982	254
535	212	599	220	663	227	727	233	791	239	855	245	919	250	983	254
536	212	600	220	664	227	728	233	792	239	856	245	920	250	984	255
537	212	601	220	665	227	729	234	793	239	857	245	921	250	985	255
538	212	602	220	666	227	730	234	794	240	858	245	922	250	986	255
539	212	603	220	667	227	731	234	795	240	859	245	923	250	987	255
540	213	604	220	668	227	732	234	796	240	860	245	924	250	988	255
541	213	605	221	669	228	733	234	797	240	861	245	925	250	989	255
542	213	606	221	670	228	734	234	798	240	862	245	926	250	990	255
543	213	607	221	671	228	735	234	799	240	863	245	927	250	991	255
544	213	608	221	672	228	736	234	800	240	864	245	928	250	992	255
545	213	609	221	673	228	737	234	801	240	865	246	929	250	993	255
546	213	610	221	674	228	738	234	802	240	866	246	930	251	994	255
547	214	611	221	675	228	739	235	803	240	867	246	931	251	995	255
548	214	612	221	676	228	740	235	804	240	868	246	932	251	996	255
549	214	613	221	677	228	741	235	805	240	869	246	933	251	997	255
550	214	614	222	678	229	742	235	806	241	870	246	934	251	998	255
551	214	615	222	679	229	743	235	807	241	871	246	935	251	999	255
552	214	616	222	680	229	744	235	808	241	872	246	936	251	1000	255
553	214	617	222	681	229	745	235	809	241	873	246	937	251	1001	255
554	214	618	222	682	229	746	235	810	241	874	246	938	251	1002	255
555	215	619	222	683	229	747	235	811	241	875	246	939	251	1003	255
556	215	620	222	684	229	748	235	812	241	876	246	940	251	1004	255
557	215	621	222	685	229	749	235	813	241	877	246	941	251	1005	255
558	215	622	222	686	229	750	236	814	241	878	247	942	251	1006	255
559	215	623	223	687	229	751	236	815	241	879	247	943	252	1007	255
560	215	624	223	688	230	752	236	816	241	880	247	944	252	1008	255
561	215	625	223	689	230	753	236	817	242	881	247	945	252	1009	255
562	215	626	223	690	230	754	236	818	242	882	247	946	252	1010	255
563	216	627	223	691	230	755	236	819	242	883	247	947	252	1011	255
564	216	628	223	692	230	756	236	820	242	884	247	948	252	1012	255
565	216	629	223	693	230	757	236	821	242	885	247	949	252	1013	255
566	216	630	223	694	230	758	236	822	242	886	247	950	252	1014	255
567	216	631	223	695	230	759	236	823	242	887	247	951	252	1015	255
568	216	632	224	696	230	760	236	824	242	888	247	952	252	1016	255
569	216	633	224	697	230	761	237	825	242	889	247	953	252	1017	255
570	216	634	224	698	231	762	237	826	242	890	247	954	252	1018	255
571	217	635	224	699	231	763	237	827	242	891	248	955	252	1019	255
572	217	636	224	700	231	764	237	828	242	892	248	956	252	1020	255
573	217	637	224	701	231	765	237	829	243	893	248	957	253	1021	255
574	217	638	224	702	231	766	237	830	243	894	248	958	253	1022	255
575	217	639	224	703	231	767	237	831	243	895	248	959	253	1023	255

camss-263

### 8.3.3.7.14 ISS ISP ISIF Culling

Figure 8-223 shows the culling block diagram.

**Figure 8-223. ISS ISP ISIF Culling Block Diagram**



The culling block performs a programmable decimation function for horizontal, vertical, and YUV4:2:2 data directions. The horizontal and vertical decimation of image data can be controlled by two registers.

The horizontal culling operation allows selected pixel data to be culled (deleted) from a line. The **ISIF\_CULH** register specifies the horizontal culling pattern for even and odd lines:

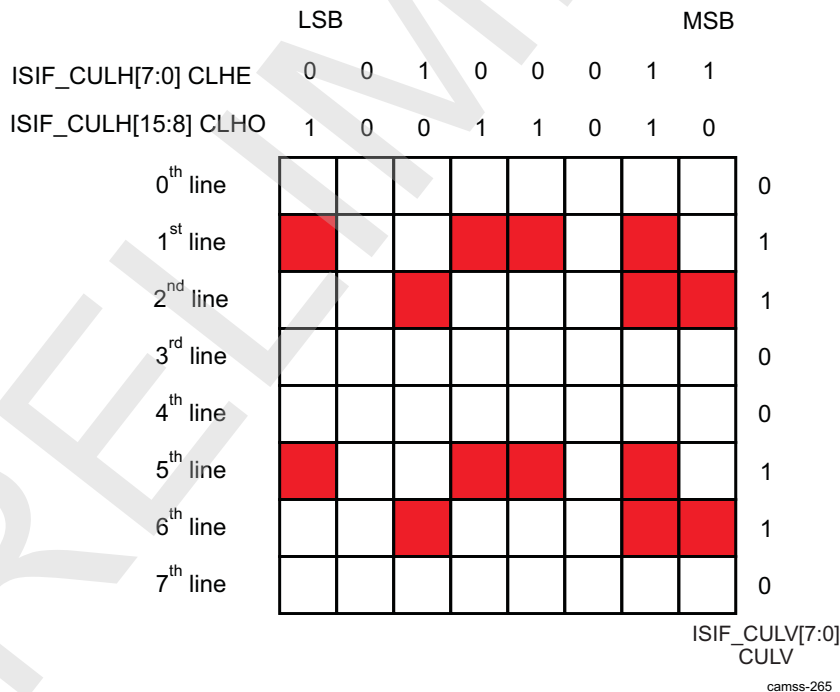
- Even lines: **ISIF\_CULH**[15:8] CLHE
- Odd lines: **ISIF\_CULH**[16:0] CLHO

The vertical culling operation allows selected lines to be culled from a frame. The **ISIF\_CULV** register specifies the pattern for the vertical direction. The LSBs of CULV represent the top line of the CCD; the MSB is the seventh line.

**Figure 8-224** is an example of how register values apply the decimation pattern to the data. The red pixels are saved to memory and the white pixels are discarded. In this example, **CULH** = 0x59C4 and **CULV** = 0x0066.

**NOTE:** Culling can be used with YUV data, but care must be taken to preserve the YUV4:2:2 output format.

**Figure 8-224. ISS ISP ISIF Example for Decimation Pattern**



**8.3.3.7.15 ISS ISP ISIF 12-to-8 bit DCPM Compression Block**

In ISIF, there is a DCPM compression block, which is between the culling module and the storage formatter module. This block can compress 12-bit image data to 8 bits for bandwidth reduction in transmission between the ISIF and SDRAM. An 8- to-12-bit DCPM decoder at the IPIPEIF decompresses data for IPIPE processing.

Two different predictors are used for the compression system. The first predictor is simple (simple predictor), and the second predictor is slightly more complex (advanced predictor). Because the advanced predictor gives a slightly better prediction for the pixel value, the image quality can be improved using it. Because the simple predictor is very simple, the processing power and memory requirements are reduced using it, when the image quality is already sufficiently high.

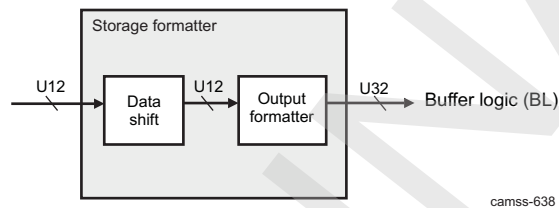
- Advanced predictor: This predictor uses only the previous same color component value as a prediction value. Therefore, only 2-pixel memory is required.
- This predictor uses four previous pixel values when the prediction value is evaluated. This means that the other color component values are also used when the prediction value has been defined.

The function is controlled from the [ISIF\\_MISC\[12\] DPCMEN](#) and [ISIF\\_MISC\[13\] DPCMPRE](#) bits.

### 8.3.3.7.16 ISP ISIF Storage Formatter

[Figure 8-225](#) shows the storage formatter block diagram.

**Figure 8-225. ISS ISP ISIF Storage Formatter Block Diagram**



Data are stored to the lower bits of a 16-bit SDRAM word, or can be 8- or 12-bit packed. The [ISIF\\_HSIZE\[11:0\] HSIZE](#) bit field can specify the memory address offsets between lines of memory (offset in 32-byte units). If set, the [ISIF\\_HSIZE\[12\] ADCR](#) bit can decrement the memory address line and the line can be horizontally flipped in memory.

In case of RAW data, a data shift module is used: data to be stored can be right-shifted according to the value set at the [ISIF\\_MODESET\[10:8\] CCDW](#) bit field, as described in [Table 8-342](#).

**Table 8-342. ISS ISP ISIF RAW Data Shifting**

ISIF_MODESET[10:8] CCDW	Output Format	
	MSB	LSB
000	0000 & U12 data[11:0]	
001	00000 & U12 data[11:1]	
010	000000 & U12 data[11:2]	
011	0000000 & U12 data[11:3]	
100	00000000 & U12 data[11:4]	

Table 8-343 shows the format where data are stored to the lower bits of a 16-bit word and the format where data are packed to 8 bits. The unused bits are filled with zeros.

**Table 8-343. ISS ISP ISIF SDRAM Data Format**

	Upper Word		Lower Word	
	MSB(31)	LSB(16)	MSB(15)	LSB(0)
12 bit	0	Pixel 1	0	Pixel 0
11 bit	0	Pixel 1	0	Pixel 0
10 bit	0	Pixel 1	0	Pixel 0
9 bit	0	Pixel 1	0	Pixel 0
8 bit	0	Pixel 1	0	Pixel 0
8-bit packed	Pixel 3	Pixel 2	Pixel 1	Pixel 0

Table 8-344 shows the format where data are packed to 12 bits.

**Table 8-344. ISS ISP ISIF SDRAM Data Format for 12-bit Packed**

	Upper Word		Lower Word	
	MSB(31)	LSB(16)	MSB(15)	LSB(0)
12 bit	Pixel2[7:0]	Pixel1	Pixel 0	
	Pixel5[3:0]	Pixel4	Pixel3	Pixel2[11:8]
	Pixel7	Pixel6	Pixel5[11:4]	

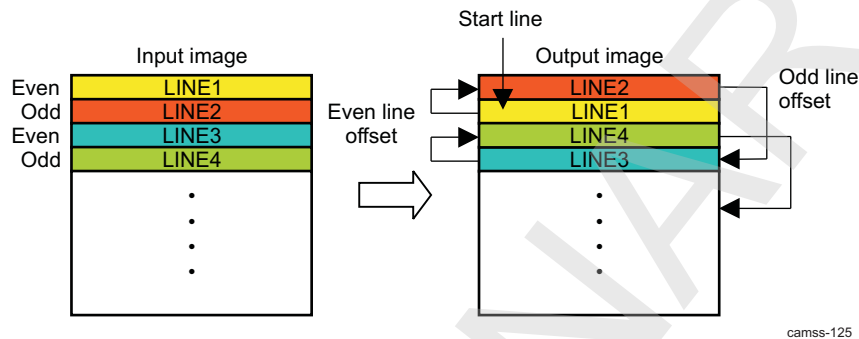
In case of YUV, YUV data is stored in memory in packed YUV4:2:2 mode, using 2 pixels per 32 bits, as shown in Table 8-345.

The output formatter can configure to any image format by using the SDRAM line offset register and offset control registers. Figure 8-226 shows how to construct a frame format in SDRAM. The ISIF\_CADU[10:0] CADU bit field specifies the memory destination (upper 11 bits) to SDRAM (the address is the value of the set bit multiplied by 32 bytes). On the other hand, the ISIF\_CADL[15:0] CADL bit field sets the memory destination to SDRAM (lower 16 bits) (the address is the value of the set bit multiplied by 32 bytes).

- ISIF\_SPH
- ISIF\_LNH
- ISIF\_SLV0
- ISIF\_SLV1

- ISIF\_LNV

**Figure 8-226. ISS ISP ISIF Frame Image Format Conversion**



### 8.3.3.7.17 ISS ISP ISIF YCbCr signal Processing

The ISIF accepts 4:2:2 sampled YCbCr input data. The luminance and color difference signals are 8 bits each, scaled 0 to 255. The color difference signals are multiplexed into one 8-bit bus beginning with a Cb sample. The Y and CbCr buses can be input parallel (16-bit mode) or be time multiplexed and input as a single bus (8-bit mode).

The 16- or 8-bit YCbCr data is stored in SDRAM as 4:2:2 format. [Table 8-345](#) lists the data format in SDRAM. Y data typically has a range of 16 to 235; however, it is possible to subtract a DC value from the Y signal.

**Table 8-345. ISS ISP ISIF Memory Output Format for YUV Data**

Memory Address	Upper Word		Lower Word	
	MSB(31)	LSB(16)	MSB(15)	LSB(0)
N	Y1	Cr0	Y0	Cb0
N + 1	Y3	Cr1	Y2	Cb1
N + 2	Y5	Cr2	Y4	Cb2

### 8.3.3.7.18 ISS ISP ISIF Expected Bandwidth on BL Ports

The ISIF module has a write port and a read port connected to the BL. This section summarizes the expected bandwidth on these ports.

#### 8.3.3.7.18.1 ISS ISP ISIF Write Port

The write port is used to write pixels to memory after the data have passed through the storage formatter. Data storage to SDRAM is controlled by the `ISIF_SYNCEN[1]` DWEN bit. The module allows writing the data as 16, 12, and 8 bits per pixel. The bit width is controlled by the `ISIF_CCDCFG[1:0]` SDRPACK bit field.

The write port generates a burst of 32 bytes on the MTC interface. The delay between consecutive bursts is proportional to the input pixel clock. Hence, the write port does not request peak bandwidth traffic.

[Table 8-346](#) lists the estimated delay between 32-byte MTC requests for different pixel clock frequencies and assumes the L3 clock is 200 MHz.

**Table 8-346. ISP ISIF ISIF Module: Write Port Bandwidth**

Pixel Clock	Max Bandwidth 2 bytes/s pixel MB/s	Expected Delay btw MTC Requests
200	400	16 cycles = 80 ns
100	200	32 cycles = 160 ns
10	20	320 cycles = 1600 ns

#### 8.3.3.7.18.2 ISS ISP ISIF Read Port

The read port is used to read gain and offset data from SDRAM required for the LSC computation. When LSC is enabled, 8-bit gain values are read and 8-bit offset values can optionally also be read. The LSC gain computation can be enabled or disabled by setting the `ISIF_2DLSCCFG[0]` ENABLE bit. The LSC offset computation can be enabled by setting the `ISIF_2DLSCOFST[0]` OFSTEN bit.

The LSC submodule fetches four 8-bit gain values per paxel and optionally four 8-bit offset values per paxel. This is a maximum 8 bytes per paxel.

The bandwidth that is generated by the LSC module is also proportional to the paxel size. The paxel size is set up by the `ISIF_2DLSCCFG[14:12]` GAIN\_MODE\_M and `ISIF_2DLSCCFG[10:8]` GAIN\_MODE\_N bit fields. The possible values are 8, 16, 32, 64 and 128. Smaller values lead to higher memory bandwidth requirements. Hence, the worst case is achieved by setting an 8 × 8 paxel size.

When the LSC submodule is enabled it automatically prefetches two lines of gain values and two lines of offset values (if this is enabled). When the first VD comes, it again requests one line of gain values and one line of offset values (if this is enabled). Then, it again fetches one line of gain values and one line of offset values (if this is enabled) after `ISIF_2DLSCCFG[10:8] GAIN_MODE_N` lines. It continues to do so until the last row of paxels. For the last row of paxels, it fetch two lines of gain values and two lines of offset values (if this is enabled), which are used for the following frame.

By default, the LSC submodule creates peak bandwidth requirements. To avoid this, the MTC bandwidth limiter must be used to space the request over time.

The MTC bandwidth limiter must be used to smooth the bandwidth requirements of the LSC module. The MTC bandwidth limiter can be set with the `ISP5_BL_MTC_1.ISIF_R` register.

The principle is that instead of reading the gain and offset data as fast as possible, use the time that it takes for `ISIF_2DLSCCFG[10:8] GAIN_MODE_N` lines to pass through the ISP to read the data.

[Table 8-347](#) gives the estimated delay between 32-byte MTC requests for different pixel clock frequencies and assumes the L3 clock at 200 MHz.

**Table 8-347. ISS ISP ISIF Read Port Bandwidth**

Pixel Clock	Max Bandwidth MB/s	Expected Delay btw MTC Requests
200	25.07	255 cycles = 1275 ns
100	12.5	510 cycles = 2550 ns
10	1.25	5103 cycles = 2515 ns

When the bandwidth limiter is used, ensure that there is enough time for the data prefetching.

- The LSC submodule must be enabled at least 2 x `ISIF_2DLSCCFG[10:8] GAIN_MODE_N` lines before the first VD.
- There must be at least `ISIF_2DLSCCFG[10:8] GAIN_MODE_N` lines of blanking. If there is not enough blanking, multiply the bandwidth requirement by 2 (that is, ensure two lines of gain and offset data can be fetched within the time of `ISIF_2DLSCCFG[10:8] GAIN_MODE_N` lines.

### 8.3.3.7.19 ISS ISP ISIF Events and Status Checking

The ISIF module can generate four different interrupts: VDINT0, VDINT1, VDINT2, and 2DLSCINT. The `ISIF_SYNCEN[0] SYEN` bit must be enabled to receive any of the ISIF interrupts.

#### 8.3.3.7.19.1 ISS ISP ISIF VDINT0, VDINT1, and VDINT2 Interrupts

As shown in [Figure 8-227](#), the VDINT0, VDINT1, and VDINT2 interrupts occur relative to the VD pulse. The trigger timing is selected by using `ISIF_MODESET[2] VDPOL` bit setting. VDINT0, VDINT1, and VDINT2 occur after receiving the number of horizontal lines (HD pulse signals) set in the `ISIF_VDINT0[14:0] CDV0`, `ISIF_VDINT1[14:0] CDV1`, and `ISIF_VDINT2[14:0] CDV2` register fields, respectively.

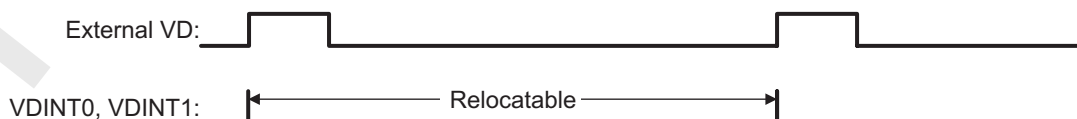
---

**NOTE:** In the case of BT.656 input mode, there is a VD at the beginning of each field. Therefore, there are two interrupts for each frame (that is, one for each field).

---

If the `ISIF_MODESET[2] VDPOL` bit is set to 0, the VDINT0, VDINT1, and VDINT2 HD counters begin counting HD pulses from the rising edge of the external VD.

**Figure 8-227. ISS ISP ISIF VDINT0/VDINT1/VDINT2 Interrupt Behavior When VDPOL = 0**

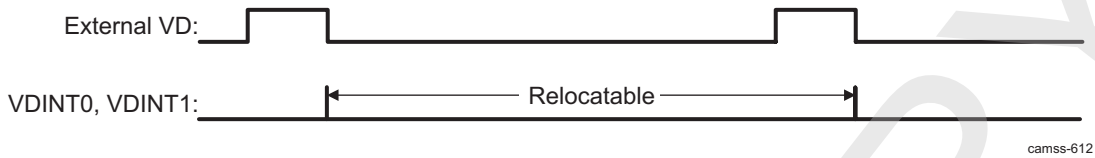


camss-611



If the [ISIF\\_MODESET](#)[2] VDPOL bit is set to 1, the VDINT0, VDINT1, and VDINT2 HD counters begin counting HD pulses from the falling edge of the external VD.

**Figure 8-228. ISS ISP ISIF VDINT0/VDINT1/VDINT2 Interrupt Behavior When VDPOL = 1**



### 8.3.3.7.19.2 ISS ISP ISIF 2DLSCINT Interrupt

See [Section 8.3.3.7.10.1.5](#) for details.

### 8.3.3.7.19.3 ISS ISP ISIF Status Checking

The [ISIF\\_MODESET](#)[15] MDFS bit is set when the field status is on an even field, and it is cleared when the field status is on an odd field.

The 2D-LSC has a register that monitors the status of the LSC. See [Section 8.3.3.7.10.1.5](#) for details.

## 8.3.3.8 ISS ISP BL Functional Description

### 8.3.3.8.1 ISS ISP BL Overview

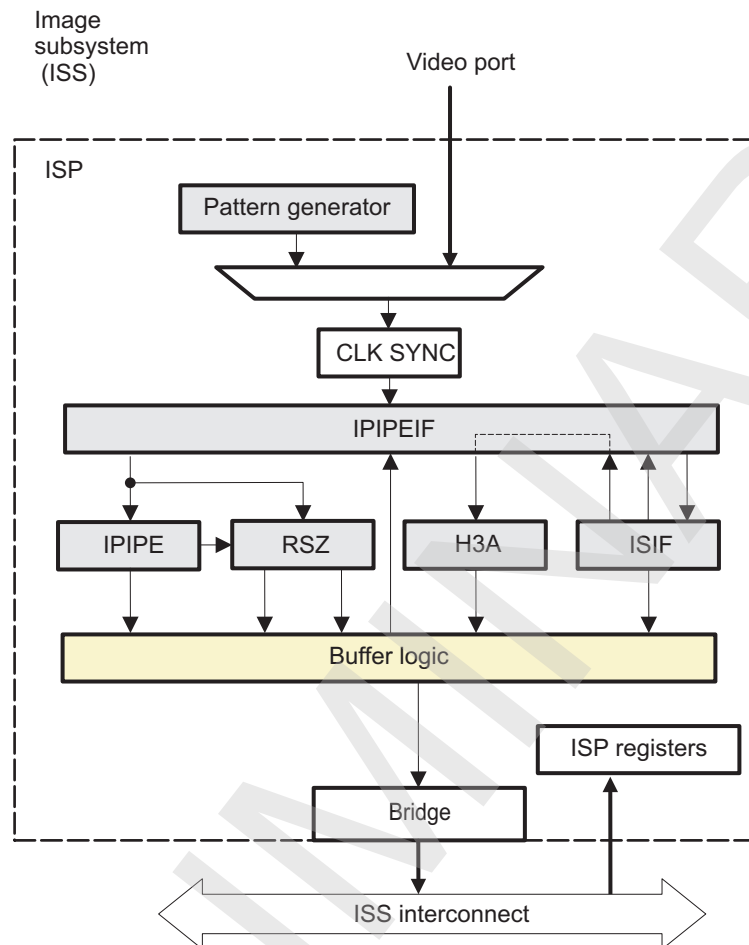
The BL module arbitrates and merges ISP master module memory requests. The BL module also generates interrupts upon frame completion for the following modules. The interrupt generation is delayed until the actual transfer completes (ack is returned). The following interrupts are delayed by the BL module:

- RSZ module RSZ-A EOF
- RSZ module resizer RSZ-B EOF
- H3A module EOF
- IPIPE module EOF

The BL uses two different types of interfaces:

- MTC protocol is used between the ISP modules and the BL.
- VBUSM is use on the BL master port interface.

[Figure 8-229](#) show the BL module connections to other submodules of the ISP.

**Figure 8-229. ISS ISP BL High-Level Diagram**

camss-003f

**8.3.3.8.2 ISS ISP BL Functional Description**

The BL merges the ISP master-module memory requests to memory (read/write). The BL interfaces with all the ISP modules through a 32-bit-wide bus following.

The ISP modules make memory requests of 32 bytes. Additional signals, SOF for read and EOF for write are included to deal with boundary conditions in frame transitions.

The BL arbitration is divided into two parts: a bug hog and a fixed priority arbitration. Bus hog refers to the property of the buffer logic that gives higher priority to the module that last sent or received data. RESIZER module MTC write port 0 and RESIZER module MTC write port 1 are excluded from the bus hog.

The buffer logic is to be programmed to maximize the memory bandwidth: it makes maximum burst requests of 128 bytes ( $8 \times 128$  bits) for reads and writes. The BL can generate burst sizes of  $2 \times 128$ ,  $4 \times 128$ ,  $6 \times 128$ , and  $8 \times 128$  bits.

---

**NOTE:** The ISP interface only supports burst sizes of 1, 2, 4, and  $8 \times 128$  bits. If the BL generates a  $6 \times 128$ -bit request, it is broken into a  $4 \times 128$ -bit request, followed by another  $2 \times 128$ -bit request.

---

To use the memory bandwidth efficiently, the BL interfaces with the memory through a high-bandwidth bus (128 bits wide).

The BL handles memory requests for the following modules:

- IPIPEIF module read port
- ISIF-LSC module read port
- ISIF module write port
- IPIPE-BOXCAR module write port
- H3A module write port
- RESIZER module write port 0
- RESIZER module write port 1

From a use case point of view, the following sharing and priority arrangement is used. All reads have higher priority than writes. For reads: IPIPEIF > ISIF-LSC; for writes: ISIF > IPIPE-BOXCAR > RESIZER 0 > RESIZER 1 > H3A

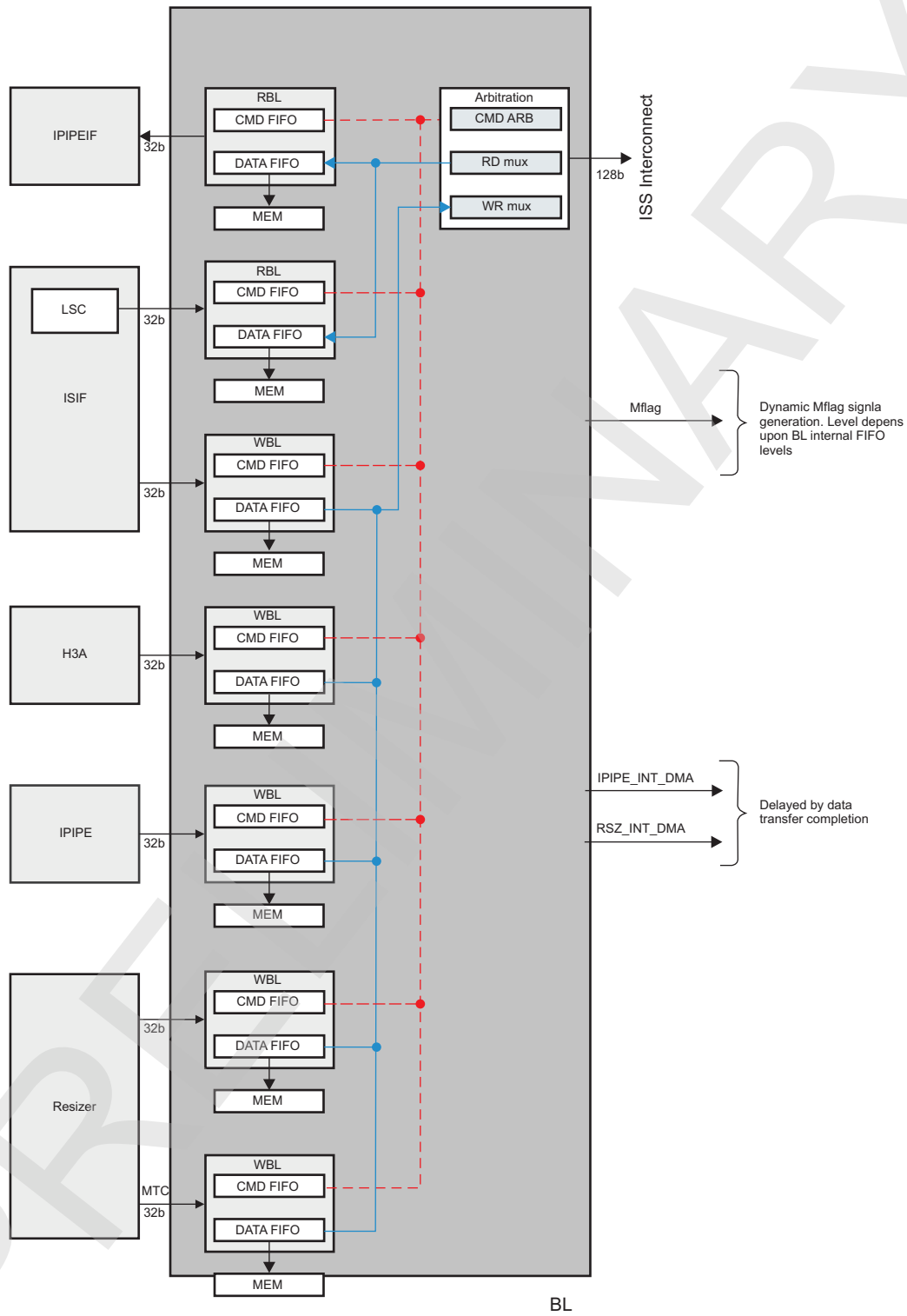
---

**NOTE:** BL can generate a static or a dynamic MFlag signal. The MFlag signal is used by the ISS arbitration to consider the urgency of the requests coming from the ISP. The dynamic MFlag feature is enabled from the [ISP5\\_CTRL\[21\]](#) MFLAG bit.

---

[Figure 8-230](#) shows the BL top-level block diagram. The figure highlights the two clock domains that are used.

Figure 8-230. ISS ISP BL Block Diagram



camss-080u

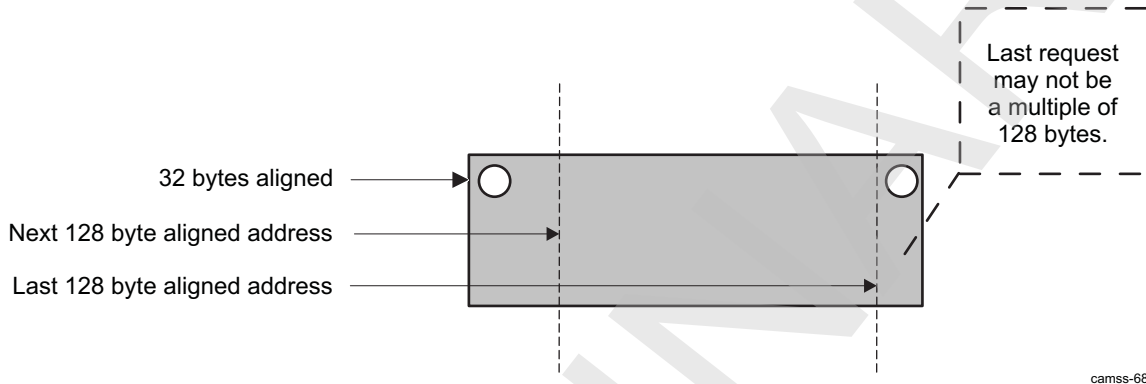
**NOTE:** The BL module has no registers. The configurations come from the top level of the ISP. See [Table 8-389](#) for register details.

### 8.3.3.8.3 ISS ISP BL Address Alignment

The BL module ensures maximum memory efficiency by re-aligning data to a 128-byte address boundary. In all cases, the BL accesses are 32-byte-aligned: address [4:0] is always 0.

This is required when the input address is a multiple of 32 bytes, not 128 bytes. The BL issues a non-aligned burst until it reaches a 128-byte boundary, and then keeps making a 128-byte request until the end of the line. Eventually, although the last burst in a line may not be a multiple of 128 bytes, it will always be a multiple of 32 bytes, as shown in Figure 8-231.

Figure 8-231. ISS ISP BL Address Alignment



camss-687

### 8.3.3.8.4 ISS ISP BL Out-of-Order Responses

BL supports out-of-order responses. The out-of-order response is handled by having up to 16 outstanding CIDs (reads + writes). The maximum number of outstanding CIDs is set up by the `ISP5_CTRL[7:4]` `VBUSM_CIDS` bit field. There is one outstanding request per CID.

If any of the CIDs are not outstanding, a command is accepted from the highest prioritized buffer with a request. The CID availability is cleared when status complete is received. The outstanding commands can be all reads or all writes or any combination. The CID that is allocated is the lowest number of the 16 that are available when the command is accepted. As soon as a CID is released it is used if a command is available.

The `VBUSM2OCP` module transforms the `VBUSM` CIDs into `OCP MFlag` signals. Note that there can be only one outstanding per tag. It is not authorized to make a new request on a tag which is already waiting for a response.

### 8.3.3.8.5 ISS ISP BL Stalling

The BL can stall the requests from the initiator modules for reads and writes

#### 8.3.3.8.5.1 ISS ISP BL Stalling Write Requests

One reason for the BL to stall write requests is that usually there is back pressure on the system memory and requests cannot be issued as fast as required. In that case the BL needs to stall requests from the initiator modules. The BL module cannot know whether the request stalling will lead to an initiator overflow or not. The `RESIZER` module can support back pressure from the BL module when it is configured in bypass mode. The input buffer memory can be used to store data from the `IPIPE` and `IPIPEIF` modules. If the `RESIZER` input buffer overflows, an interrupt is issued.

#### 8.3.3.8.5.2 ISS ISP BL Stalling Read Requests

The BL may also stall read requests from the `ISIF-LSC` and `IPIPEIF` modules. When data is coming from an image sensor there is no way to stop the sensor. If the read request is stalled too long, the `LSC` module may eventually underflow. When the `IPIPEIF` module reads all its data from memory, the `IPIPEIF` stalls transfers to successive modules by masking the clock.

### 8.3.3.8.6 ISS ISP BL Dynamic and Static MFlag Generation

**NOTE:** The following applies when the `ISP5_CTRL[21]` MFLAG = 1. The MFlag value is static.

The BL outputs `cpriority[2:0]` and `cepriority[2:0]` signals on its VBUS master port interface. On the VBUSM side, these signals are aligned with the request (`creq`). Both values are actually repeating register input signals. The lower value (0) corresponds to the higher priority.

The MFlag signal does not need to be aligned with the request. The signal can change value anytime.

In the ISP VBUSM2 interface bridge, the `cepriority` signal is ignored and only the `cpriority` signal is used. Because the priority value is set up by a register setting, the value is not going to be dynamically modified obviously. [Table 8-348](#) shows how the `cpriority` values are mapped to the interface MFlag signal, which is present on the ISP master interface and set in the `ISP5_CTRL[3:1]` VBUSM\_CPRIORITY bit field.

**Table 8-348. ISS ISP BL `cpriority` to MFlag With `ISP5_CTRL[21]` MFLAG = 1**

MFlag[1:0]	Description
00	Normal priority <code>cpriority[2:0]</code> = 4, 5, 6, 7
01	Medium priority <code>cpriority[2:0]</code> = 2, 3
10	Reserved
11	High priority <code>cpriority[2:0]</code> = 0, 1

**NOTE:** The following applies when the `ISP5_CTRL[21]` MFLAG = 0. The MFlag value is dynamic.

This feature is enabled at reset. The general idea for the dynamic MFlag generation is to monitor the FIFO levels. Thresholds are used to increase or decrease the MFlag values. There are different implementations for read and write requestors.

An individual MFlag value is generated for each FIFO and then ORed altogether and exported at the BL boundary. The MFlag signal generation does not affect the BL arbitration scheme.

- Writes: To prevent overflows in the BL, the dynamic MFlag signal generation gives higher priority to FIFOs that are almost full:
  - Low FIFO level = 50%
  - High FIFO level = 75%
  - FIFOs with more than 75% fill level have high priority: MFlag = 11
  - FIFOs between 50 and 75% fill level have medium priority: MFlag = 01
  - FIFOs below 50% fill level have normal priority: MFlag = 00

[Table 8-349](#) gives the low- and high-level priority thresholds for write initiators.

**Table 8-349. ISS ISP BL MFlag Write Low- and High-Level Priority Thresholds**

	ISIF	H3A	IPIPE	RSZ
Access type	Write	Write	Write	Write
Buffer size	64 x 128	64 x 128	48 x 128	64 x 136
50% low level	32 x 128	32 x 128	24 x 128	32 x 136
75% high level	48 x 128	48 x 128	36 x 128	48 x 128

- Reads: The dynamic MFlag signal generation depends on the "reserved data units" in the initiator data FIFO. The reserved data units correspond to read commands waiting to be sent on the interface bus plus the read commands that have been sent on the interface bus and for which data responses have not yet arrived.
- The total FIFO size in bytes can be expressed as reserved data units (bytes) + data bytes stored (bytes) + empty space (bytes). By definition, empty space is lower than a burst size (128 bytes).
  - Low FIFO level = 25%
  - High FIFO level = 50%
  - Data bytes stored + empty space (bytes)  $\geq$  50% of FIFO size: MFlag = 00

- Data bytes stored + empty space (bytes) < 50% of FIFO size: MFlag = 01
- Data bytes stored + empty space (bytes) < 25% of FIFO size: MFlag = 11

Table 8-350 gives the low- and high-level priority thresholds for read initiators

**Table 8-350. ISS ISP BL MFlag Read Low- and High-Level Priority Thresholds**

	IPIPEIF	ISIF-LSC
Access type	Read	Read
Buffer size	64 x 128	32 x 128
25% buffer size	16 x 128	8 x 128
50% buffer size	32 x 128	16 x 128

**8.3.3.8.7 ISS ISP BL VBUSM2OCP Last Beat Command Delay**

The VBUSM2OCP module bridge implements the following function to work around a limitation of the BL module, which does not send back-to-back requests to the ISS, thereby leading to possible situations where the ISP loses arbitration at the ISS level.

To fully benefit from the dynamic MFlag generation (see Section 8.3.3.8.6), the following function is present in the VBUSM2OCP module bridge:

- The delay occurs only if BL MFlag >= ISP5\_BL\_VBUSM[5] MFLAG\_THRES.
- The MFlag value used is whatever is available when the last beat comes on the interface bus.
- The last beat of the interface request (read or write) is held until one cycle before a new command (read or write). This is achieved by masking the last beat of the interface command at the ISP interface.

The last beat is unmasked on the first event of (i) one cycle before a new interface command, or (ii) the delay counter that uses the ISP5\_BL\_VBUSM[4:0] LASTCMD\_DLY value counter expires (has decremented to zero). The ISP5\_BL\_VBUSM[4:0] LASTCMD\_DLY bit field must be set before the request on the BL starts. If the value of the ISP5\_BL\_VBUSM[4:0] LASTCMD\_DLY bit field is changed during the pending requests, the delay counter is not updated.

**8.3.3.8.8 ISS ISP BL Peak Memory Bandwidth Reduction**

To limit the peak memory bandwidth generated by the IPIPEIF (read port), ISIF (read port), and H3A (write port) modules, a bandwidth limiter is placed between the modules and the BL. The resizer module has this function built in and therefore does not need a bandwidth limiter.

The bandwidth limiter enables control of the minimum interval between two consecutive memory requests.

This function is controlled by the ISP5\_BL\_MTC\_1 and ISP5\_BL\_MTC\_2 registers. When the registers are set to 0, the function is not modified (that is, the bandwidth limiter is disabled). For the resizer module, it is controlled by the following registers: RSZ\_DMA\_RZA and RSZ\_DMA\_RZB.

**8.3.3.9 ISS ISP Memory Mapping**

A total of 64KB is reserved for the ISP registers and memories. Table 8-351 describes the memory map.

**Table 8-351. ISS ISP Memory Mapping**

Memory Mapping	Start	End	Size	Comments
ISS ISP5 SYS1	0x5201 0000	0x5201 009F	160	ISP5 configuration registers (set 1)
ISS ISP5 SYS2	0x5201 00A0	0x5201 03FF	864	ISP5 configuration registers (set 2)
ISS RESIZER registers	0x5201 0400	0x5201 07FF	1024	RESIZER configuration registers
ISS IPIPE registers	0x5201 0800	0x5201 0FFF	2048	IPIPE configuration registers
ISS ISIF registers	0x5201 1000	0x5201 11FF	512	ISIF configuration registers

**Table 8-351. ISS ISP Memory Mapping (continued)**

Memory Mapping	Start	End	Size	Comments
ISS IPIPEIF registers	0x5201 1200	0x5201 13FF	512	IPIPEIF configuration registers
ISS H3A registers	0x5201 1400	0x5201 15FF	512	H3A configuration registers
Reserved	0x5201 1600	0x5201 17FF	512	Reserved
Reserved	0x5201 1800	0x5201 1BFF	1024	Reserved
Reserved	0x5201 1C00	0x5201 1DFF	512	Reserved
Reserved	0x5201 1E00	0x5201 1FFF	512	Reserved
HST memory 0	0x5201 2000	0x5201 27FF	2048	IPIPE histogram
HST memory 1	0x5201 2800	0x5201 2FFF	2048	IPIPE histogram
HST memory 2	0x5201 3000	0x5201 37FF	2048	IPIPE histogram
HST memory 3	0x5201 3800	0x5201 3FFF	2048	IPIPE histogram
Reserved	0x5201 4000	0x5201 6EFF	16384	Reserved
DPC table 0	0x5201 8000	0x5201 81FF	1024	IPIPE Defect(Fault) Pixel Correction address table
DPC table 1	0x5201 8400	0x5201 85FF	1024	IPIPE Defect(Fault) Pixel Correction address table
YEE table	0x5201 8800	0x5201 8FFF	2048	IPIPE Y-data Edge Enhance table
Reserved	0x5201 9000	0x5201 A6FF	6144	Reserved
GAMR table	0x5201 A800	0x5201 AFFF	2048	IPIPE Gamma correction table (R)
GAMG table	0x5201 B000	0x5201 B7FF	2048	IPIPE Gamma correction table (G)
GAMB table	0x5201 B800	0x5201 BFFF	2048	IPIPE Gamma correction table (B)
LIN table0	0x5201 C000	0x5201 C17F	1024	ISIF Linearization table
LIN table1	0x5201 C400	0x5201 C57F	1024	ISIF Linearization table
DCCLAMP	0x5201 C800	0x5201 C9FF	2048	ISIF Digital Clamp
LSC table0	0x5201 D000	0x5201 E7FF	6144	ISIF Lens Shading gain table
LSC table1	0x5201 E800	0x5201 FFFF	6144	ISIF Lens Shading gain table



### 8.3.4 ISS ISP Programming Model

**NOTE:** The preferred way to perform memory-to-memory processing with ISP is to use the CCP2 receiver module at the ISS level (see [Section 8.2 ISS Interfaces](#)). It is possible to use the IPIPEIF read port for memory-to-memory processing, but it is not the preferred way because it does not provide enough granularity on the fractional clock divider for up to 20x digital zoom.

#### 8.3.4.1 ISS ISP PG Programming Model

The procedure in [Table 8-352](#) initializes the pattern generator.

**Table 8-352. ISS ISP PG Settings**

Step	Register/Bit Field/Programming Model	Value
Select the PG as the ISP source.	ISP5_PG[5:4] SRC_SEL	0x3
Select the mode for write-enable signal generation.	ISP5_PG[2] WEN	
Select the polarity of the horizontal sync signal.	ISP5_PG[1] HDPOL	
Select the polarity of the vertical sync signal.	ISP5_PG[0] VDPOL	
Sets the duration of the HD signal	ISP5_PG_PULSE_CTRL[12:0] HDW	
Sets the duration of the VD signal	ISP5_PG_PULSE_CTRL[27:16] VDW	
Sets the number of pixels per line	ISP5_PG_FRAME_SIZE[31:16] PPLN	
Sets the number of lines per frame	ISP5_PG_FRAME_SIZE[15:0] HLPFR	
Enables the pattern generator	ISP5_PG[3] EN	0x1

#### 8.3.4.2 ISS ISP ISIF Programming Model

This section discusses issues related to the software control of the ISIF. It lists the registers that are required to be programmed in different modes, describes how to enable and disable the ISIF and how to check the status of the ISIF, discusses the different register access types, and enumerates several programming constraints.

##### 8.3.4.2.1 ISS ISP ISIF Hardware Setup/Initialization

This section discusses the configuration of the ISIF required before image processing can begin.

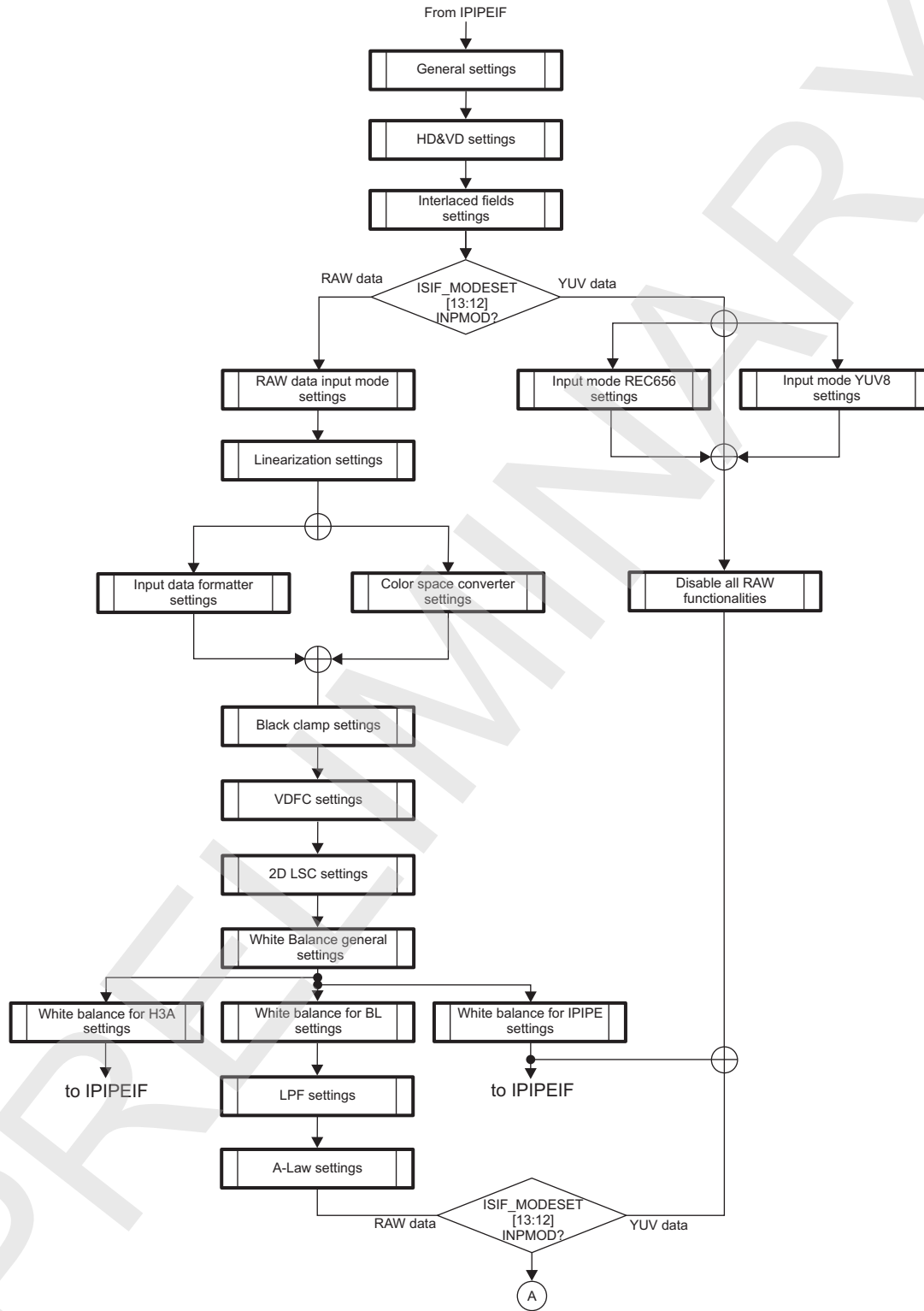
###### 8.3.4.2.1.1 ISS ISP ISIF Reset Behavior

Upon hardware reset of the ISP, all of the registers in the ISIF are reset to their reset values except the defect table registers. Because the defect table registers are stored in internal RAM, they do not have reset values. If the reset is a chip-level POR (reset after power is applied), the defect table register values are unknown. If the reset is an ISP module reset (when power remains active), the contents of this memory remain the same as before the reset.

###### 8.3.4.2.1.2 ISS ISP ISIF Register Setup

Before enabling the ISIF, the hardware must be properly configured through register writes. [Figure 8-232](#) and [Figure 8-233](#) show the sequence to be used for RAW and YUV data before enabling the ISIF. The register settings for each process in the following sequence are described in [Table 8-353](#).

Figure 8-232. ISS ISP ISIF Initialization Flow Chart – Part One



camss-600

Figure 8-233. ISS ISP ISIF Initialization Flow Chart – Part Two

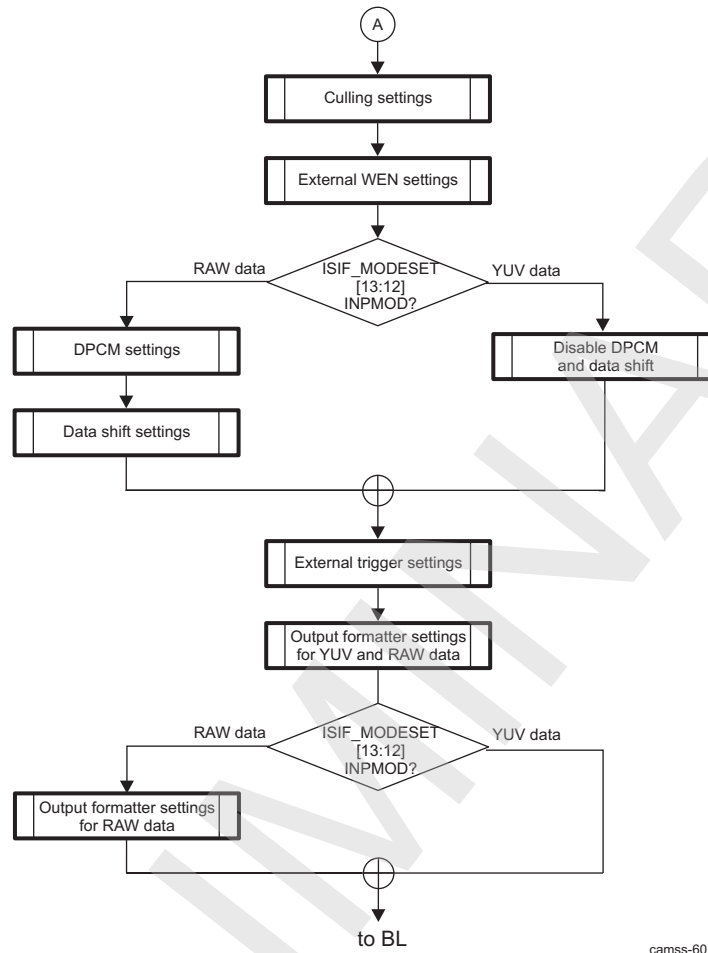


Table 8-353. ISS ISP ISIF Required Configuration Parameters

Step	Configuration Required	Value
<b>General settings</b>		
Set the Field Indicator signal direction.	ISIF_MODESET[1] FIDD	
Set the VD signal polarity.	ISIF_MODESET[2] VDPOL	
Set the HD signal polarity.	ISIF_MODESET[3] HDPOL	
Set the Field Indicator signal polarity.	ISIF_MODESET[4] FIPOL	
<b>HD and VD settings</b>		
Set the HD and VD signal directions.	ISIF_MODESET[0] HDVDD	
<b>If: HD and VD are set as output (HDVDD = 0x1):</b>		
Set the HD width.	ISIF_HDW[11:0] HDW	
Set the VD width.	ISIF_VDW[11:0] VDW	
Set the HD period.	ISIF_PPLN[15:0] PPLN	
Set the VD period.	ISIF_LPFR[15:0] LPFR	
<b>End</b>		
<b>Interlaced fields settings</b>		
Select the type of image sensor (progressive or interlaced)	ISIF_MODESET[7] CCDMD	
<b>Input mode settings</b>		
Set the data input mode.	ISIF_MODESET[13:12] INPMOD	

**Table 8-353. ISS ISP ISIF Required Configuration Parameters (continued)**

Step	Configuration Required	Value
Input mode settings for REC656 data		
Select CCIR Rec.656 interface or not.	ISIF_REC656IF[1] R656ON	
<b>If:</b> input is REC656 (R656ON = 0x1)		
Set error correction of FVH code.	ISIF_REC656IF[0] ECCFVH	
Selects bit width of CCIR656.	ISIF_CCDCFG[5] BT656	
<b>e End</b>		
Input mode settings for YCC 8 bit data (if INPMOD = 0x2)		
Selects Y position signal	ISIF_CCDCFG[11] Y8POS	
RAW data processing: input settings		
Enable or disable MSB inverse of CIN port (YUV format).	ISIF_CCDCFG[13] MSBINVI	
Select Y and C swapping.	ISIF_CCDCFG[4] YCINSWP	
Selects MSB position of input data (16 bits-to-16 bits)	ISIF_CGAMMAWD[4:1] GWDI	
Set the image sensor data polarity.	ISIF_MODESET[6] DPOL	
Select the CFA pattern mode.	ISIF_CGAMMAWD[5] CFAP	
Specifies the color pattern	ISIF_CCOLP[7:6] CP0_F0 ISIF_CCOLP[5:4] CP1_F0 ISIF_CCOLP[3:2] CP2_F0 ISIF_CCOLP[1:0] CP3_F0	
Linearization settings		
Enable linearization or not.	ISIF_LINCFG0[0] LINEN	
<b>If:</b> Linearization enabled (LINEN = 0x1)		
Select linearization mode.	ISIF_LINCFG0[1] LINMD	
Select the shift value.	ISIF_LINCFG0[6:4] CORRSFT	
Set the scale factor for LUT.	ISIF_LINCFG1[10:0] LUTSCL	
Set up linearization LUT.		
<b>End</b>		
Input data formatter settings		
Enable data formatter or not.	ISIF_FMTCFG[0] FMTEN	
<b>If:</b> Data formatter is enabled (FMTEN = 0x1)		
Select the combine input lines.	ISIF_FMTCFG[1] FMTCBL	
Select the mode normal or alternative.	ISIF_FMTCFG[2] LNALT	
Select the split/combine number of lines.	ISIF_FMTCFG[5:4] LNUM	
Set the address increment.	ISIF_FMTCFG[11:8] FMTAINC	
Set the number of program entries per SET.	ISIF_FMTPLEN[3:0] FMTPLEN0 ISIF_FMTPLEN[7:4] FMTPLEN1 ISIF_FMTPLEN[10:8] FMTPLEN2 ISIF_FMTPLEN[14:12] FMTPLEN3	
Set the first pixel in a line.	ISIF_FMTSPH[12:0] FMTSPH	
Set the number of pixels in a line.	ISIF_FMTLNH[12:0] FMTLNH	
Set the start line vertical.	ISIF_FMTLSV[12:0] FMTSLV	
Set the number of lines in a vertical.	ISIF_FMTLVN[14:0] FMTLVN	
Set the number of pixels in an output line.	ISIF_FMTRLN[12:0] FMTRLN	
Set the HD interval for output lines.	ISIF_FMTHCNT[12:0] FMTHCNT	
Set up to 16 address pointers.	ISIF_FMTAPTRx[14:13] LINE (x = 0 to 15) ISIF_FMTAPTRx[12:0] INIT (x = 0 to 15)	

**Table 8-353. ISS ISP ISIF Required Configuration Parameters (continued)**

Step	Configuration Required	Value
Set the 32 possible program entry valid flag.	ISIF_FMTPGMVF0	
	ISIF_FMTPGMVF1	
Set the 32 possible address pointers.	ISIF_FMTPGMAPS0	
	ISIF_FMTPGMAPS1	
	ISIF_FMTPGMAPS2	
	ISIF_FMTPGMAPS3	
	ISIF_FMTPGMAPS4	
	ISIF_FMTPGMAPS5	
	ISIF_FMTPGMAPS6	
	ISIF_FMTPGMAPS7	
Set the 32 possible address update (increment or decrement).	ISIF_FMTPGMAPU0	
	ISIF_FMTPGMAPU1	E
<b>End</b>		
Color space converter settings		
Enable or disable color space converter.	ISIF_CSCCTL[0] CSCEN	
<b>If:</b> Color space converter is enabled (CSCEN = 0x1)		
Set the color space converter coefficients.	ISIF_CSCM0	
	ISIF_CSCM1	
	ISIF_CSCM2	
	ISIF_CSCM3	
	ISIF_CSCM4	
	ISIF_CSCM5	
	ISIF_CSCM6	
	ISIF_CSCM7	
<b>End</b>		
Black Clamp settings		
Enable or disable black clamp.	ISIF_CLAMPCFG[0] CLEN	
<b>If:</b> Black clamp is enabled (CLEN = 0x1)		
Set the DC offset for black clamp.	ISIF_CLDCOFST[12:0] CLDC	
<i>[Horizontal Black Clamp]</i>		
Set the horizontal clamp mode.	ISIF_CLAMPCFG[2:1] CLHMD	
Set the vertical dimension of a window.	ISIF_CLHWIN0[13:12] CLHWN	
Set the horizontal dimension of a window.	ISIF_CLHWIN0[9:8] CLHWM	
Enable or disable limitation for horizontal.	ISIF_CLHWIN0[6] CLHLMT	
Select base window.	ISIF_CLHWIN0[5] CLHWBS	
Set the window count per color.	ISIF_CLHWIN0[4:0] CLHWC	
Set window start position (H).	ISIF_CLHWIN1[12:0] CLHSH	
Set the window start position (V).	ISIF_CLHWIN2[12:0] CLHSV	
<i>[Vertical Black Clamp]</i>		
Set the black clamp start position.	ISIF_CLSV[12:0] CLSV	
Set the vertical black clamp reset value.	ISIF_CLVRV[11:0] CLVRV	
Set the line average coefficient.	ISIF_CLVWIN0[15:8] CLVCOEF	
Select the reset value for the clamp value of the previous line.	ISIF_CLVWIN0[5:4] CLVRVSL	
Select the optical black H valid.	ISIF_CLVWIN0[2:0] CLVOBH	
Set the window start position (H).	ISIF_CLVWIN1[12:0] CLVSH	
Set the window start position (V).	ISIF_CLVWIN2[12:0] CLVSV	
Select the optical black V valid.	ISIF_CLVWIN3[12:0] CLVOBV	

**Table 8-353. ISS ISP ISIF Required Configuration Parameters (continued)**

Step	Configuration Required	Value
<b>End</b>		
Set the DC offset for black clamp (RAW data only).	ISIF_CLDCOFST[12:0] CLDC	DC offset available for YUV
Vertical line defect correction (VDFC) settings		
Disable vertical line defect correction.	ISIF_DFCCTL[4] VDFCEN	0x0
<b>If:</b> Vertical line defect correction will be enabled (VDFCEN = 0x1)		
Select the mode.	ISIF_DFCCTL[6:5] VDFCSL	
Select upper pixels correction enable or disable.	ISIF_DFCCTL[7] VDFCUDA	
Set the shift value.	ISIF_DFCCTL[10:8] VDFLSFT	
Set the saturation level.	ISIF_VDFSATLV[11:0] VDFS LV	
Clear memories.	ISIF_DFCMEMCTL[4] DFCMCLR	0x1
Vertical line defect table update procedure.	See Section 8.3.3.7.9.1 for details. Use the following registers: ISIF_DFCMEMCTL[2] DFCMARST ISIF_DFCMEMCTL[0] DFCMWR ISIF_DFCMEM0 ISIF_DFCMEM1 ISIF_DFCMEM2 ISIF_DFCMEM3 ISIF_DFCMEM4	
<b>Rnd</b>		
Enable vertical line defect correction.	ISIF_DFCCTL[4] VDFCEN	0x1
2D Lens Shading Compensation (LSC) settings		
Disable lens shading compensation.	ISIF_2DLSCCFG[0] ENABLE	0x0
<b>If:</b> 2D-LSC will be enabled (ENABLE = 0x1)		
Set the H direction data offset.	ISIF_LSCHOFST[13:0] HOFST	
Set the V direction data offset.	ISIF_LSCVOFST[13:0] VOFST	
Set the number of valid pixels in H direction.	ISIF_LSCHVAL[13:0] HVAL	
Set the number of valid lines in V direction.	ISIF_LSCVVAL[13:0] VVAL	
Define the horizontal dimension of a paxel.	ISIF_2DLSCCFG[14:12] GAIN_MODE_M	
Define the vertical dimension of a paxel.	ISIF_2DLSCCFG[10:8] GAIN_MODE_N	
Set gain format table.	ISIF_2DLSCCFG[3:1] GAIN_FORMAT	
Enable or disable offset control.	ISIF_2DLSCOFST[0] OFSTEN	
Select shift up value for offsets.	ISIF_2DLSCOFST[6:4] OFSTSFT	
Set scaling factor for offset.	ISIF_2DLSCOFST[15:8] OFSTSF	
Set the initial Y position.	ISIF_2DLSCINI[14:8] Y	
Set the initial X position.	ISIF_2DLSCINI[6:0] X	
Set the gain table base address.	ISIF_2DLSCGRBU[15:0] BASE31_16 ISIF_2DLSCGRBL[15:0] BASE15_0	
Set the gain table offset (length of one row).	ISIF_2DLSCGROF[15:0] OFFSET	
Set the offset table base address.	ISIF_2DLSCORBU[15:0] BASE ISIF_2DLSCORBL[15:0] BASE	
Set the offset table offset (length of one row).	ISIF_2DLSCOROF	

**Table 8-353. ISS ISP ISIF Required Configuration Parameters (continued)**

Step	Configuration Required	Value
Enable useful interrupts	ISIF_2DLSCIRQEN[3] SOF ISIF_2DLSCIRQEN[2] PREFETCH_COMPLETED ISIF_2DLSCIRQEN[1] PREFETCH_ERROR ISIF_2DLSCIRQEN[0] DONE	
Set up LSC gain table and offset table in SDRAM. See Figure 8-219 for details.		
Wait seven clock periods before enabling LSC.		
Enable lens shading compensation.	ISIF_2DLSCCFG[0] ENABLE	0x1
<b>End</b>		
White balance color settings		
Set R/Ye gain.	ISIF_CRGAIN[11:0] CGR	
Set Gr/Cy gain.	ISIF_CGRGAIN[11:0] CGGR	
Set Gr/Cy gain.	ISIF_CGBGAIN[11:0] CGGB	
Set B/Mg gain.	ISIF_CBGAIN[11:0] CGB	
Set offset.	ISIF_COFSTA[11:0] COFT	
<i>[For BL output]</i>		
Enable or disable white balance for BL path.	ISIF_CGAMMAWD[12] WBNEN0	
Enable or disable offset control for BL path.	ISIF_CGAMMAWD[8] OFSTEN0	
<i>[For IPIPE (through IPIPEIF) output]</i>		
Enable or disable white balance for IPIPE path.	ISIF_CGAMMAWD[13] WBEN1	
Enable or disable offset control for IPIPE path.	ISIF_CGAMMAWD[9] OFSTEN1	
<i>[For H3A (through IPIPEIF) output]</i>		
Enable or disable white balance for H3A path.	ISIF_CGAMMAWD[14] WBEN2	
Enable or disable offset control for H3A path.	ISIF_CGAMMAWD[10] OFSTEN2	
LPF settings (for BL output only) (RAW data only)		
Enable or disable low pass filter.	ISIF_MODESET[14] HLPF	
A-Law compression settings (for BL output only) (RAW data only)		
Enable or disable A-Law compression.	ISIF_CGAMMAWD[0] CCDTBL	
Culling settings (for BL output only) (RAW or YUV data)		
Set the culling pattern in odd lines.	ISIF_CULH[15:8] CLHO	
Set the culling pattern in even lines.	ISIF_CULH[7:0] CLHE	
Set the culling pattern in vertical lines.	ISIF_CULV[7:0] CULV	
External WEN settings (for BL output only)		
Select external WEN use or not.	ISIF_MODESET[5] SWEN	
<b>If:</b> External WEN is used (SWEN = 0x1):		
Specifies the CCD valid area	ISIF_CCDCFG[8] WENLOG	
<b>End</b>		
DPCM settings (for BL output only) RAW data only		
Select the predictor for DPCM encoder.	ISIF_MISC[13] DPCMPRE	
Enable or disable DPCM encoding.	ISIF_MISC[12] DPCMEN	
Data shift settings (for BL output only) (RAW data only)		
Select the data shift value when image is written to memory	ISIF_MODESET[10:8] CCDW	

**Table 8-353. ISS ISP ISIF Required Configuration Parameters (continued)**

Step	Configuration Required	Value
External trigger settings (for BL output only)		
<b>If:</b> External trigger is selected (EXTRG = 0x1):		
Select the trigger source signal.	ISIF_CCDCFG[9] TRGSEL	
<b>End</b>		
Output formatter (for BL output only) RAW and YUV		
Set the memory address decrement.	ISIF_HSIZE[12] ADCR	
Set the memory address offset between lines.	ISIF_HSIZE[11:0] HSIZE	
Set the first pixel in a line to be stored in memory.	ISIF_SPH[14:0] SPH	
Set the number of pixels in a line to be stored in memory.	ISIF_LNH[14:0] LNH	
Set the start line vertical for field 0.	ISIF_SLV0[14:0] SLV0	
Set the start line vertical for field 1.	ISIF_SLV1[14:0] SLV1	
Set the number of lines to be stored in memory.	ISIF_LNV[14:0] LNV	
Enable or disable the storage of image in memory.	ISIF_SYNCEN[1] DWEN	
Set the memory destination address.	ISIF_CADU[10:0] CADU ISIF_CADL[10:0] CADL	
Enable or disable MSB inverse of COUT port.	ISIF_CCDCFG[14] MSBINVO	
Enable or disable byte swap when SDRAM capturing.	ISIF_CCDCFG[12] BSWD	
Select Y and C swapping.	ISIF_CCDCFG[2] YCOUNTSWP	
Select SDRAM pack mode.	ISIF_CCDCFG[1:0] SDRPACK	
Enable or disable VD/HD output.	ISIF_SYNCEN[0] SYEN	

#### 8.3.4.2.2 ISS ISP ISIF Enable/Disable Hardware

The ISIF is enabled by setting the ISIF\_SYNCEN[0] SYEN bit. This should be done after all the required registers discussed in the previous section are programmed.

With respect to the write enable bit and output address, the following procedure must be followed:

1. Set the data output address (ISIF\_CADU and ISIF\_CADL).
2. Enable HD/VD and WEN at the same time (ISIF\_MODESET[1] DWEN and ISIF\_SYNCEN[0] SYEN).

If the ISIF\_SYNCEN[0] SYEN bit is written before the output address and the SDRAM write enable bit (not recommended but may be required for a particular mode), data begins to be written to the old address value and not the one recently programmed. The desired response can be achieved if the following procedure is followed:

1. Enable HD/VD (ISIF\_SYNCEN[0] SYEN).
2. Set the output address (ISIF\_CADU and ISIF\_CADL).
3. Wait for the next VD.
4. Enable WEN (ISIF\_MODESET[1] DWEN).

The ISIF always operates in continuous mode. In other words, after enabling the ISIF, it continues to process sequential frames until the ISIF\_SYNCEN[0] SYEN bit is cleared by software. When this happens, the frame being processed is disabled immediately, and does not continue to process the current frame.

When the HD/VD signals are set to outputs ( ISIF\_MODESET[0] HDVDD = 0x1), fetching and processing of the frame begin immediately upon setting the ISIF\_SYNCEN[0] SYEN bit.



When the HD/VD signals are set to inputs ( [ISIF\\_MODESET\[0\]](#) HDVDD = 0x0), processing of the frame is dependent on the input timing of the external sensor/decoder. To ensure that data from the external device is not missed, the ISIF must be enabled before data transmission from the external device. In this way, the ISIF waits for data from the external device.

#### 8.3.4.2.3 ISS ISP ISIF Register Accessibility During Frame Processing

There are two types of register access in the ISIF:

- Shadowed registers (event latched registers)  
Shadowed registers are those that can be read and written at any time, but the written values take effect (are latched) only at certain times based on some event. Reads still return the most recent write even though the settings are not used until the specific event occurs.
- Busy-writeable registers  
These registers/fields can be read or written even if the module is busy. Changes to the underlying settings takes place instantly.

The following registers/fields can all be set as shadow registers, or optionally set as busy-writable registers. When the [ISIF\\_CCDCFG\[15\]](#) VDLC bit is set to 0, these registers are shadowed; when the [ISIF\\_CCDCFG\[15\]](#) VDLC bit is set to 1, these registers are busy-writeable. All other ISIF registers not included in this list are always busy-writeable.

<a href="#">ISIF_SYNCEN[1]</a> DWEN	<a href="#">ISIF_SLV1</a>	<a href="#">ISIF_CGRGAIN</a>	<a href="#">ISIF_FMTLSV</a>
<a href="#">ISIF_MODESET[14]</a> HLPF	<a href="#">ISIF_CULH</a>	<a href="#">ISIF_CGBGAIN</a>	<a href="#">ISIF_FMTLNV</a>
<a href="#">ISIF_HDW</a>	<a href="#">ISIF_CULV</a>	<a href="#">ISIF_CBGAIN</a>	<a href="#">ISIF_LSCHOFS</a>
<a href="#">ISIF_VDW</a>	<a href="#">ISIF_HSIZE</a>	<a href="#">ISIF_COFSTA</a>	<a href="#">ISIF_LSCVOFS</a>
<a href="#">ISIF_PPLN</a>	<a href="#">ISIF_SDOFS</a>	<a href="#">ISIF_CLAMPCFG[0]</a> CLEN	<a href="#">ISIF_DFCCTL</a>
<a href="#">ISIF_LPFR</a>	<a href="#">ISIF_CADU</a>	<a href="#">ISIF_MISC</a>	<a href="#">ISIF_VDFSATLV</a>
<a href="#">ISIF_SPH</a>	<a href="#">ISIF_CADL</a>	<a href="#">ISIF_CGAMMAWD</a>	
<a href="#">ISIF_LNH</a>	<a href="#">ISIF_CCOLP</a>	<a href="#">ISIF_FMTSPH</a>	
<a href="#">ISIF_SLV0</a>	<a href="#">ISIF_CRGAIN</a>	<a href="#">ISIF_FMTLNH</a>	

#### 8.3.4.2.4 ISS ISP ISIF Interframe Operations

Between frames, it may be necessary to enable or disable functions or to modify the memory pointers. Because the [ISIF\\_SYNCEN\[1\]](#) DWEN bit and the memory pointer registers are shadowed, these modifications can occur any time before the end of the frame and the data will be latched in for the next frame. Likewise, the 2D-LSC registers can be changed after receiving the LSC SOF interrupt but before it starts to prefetch the gain values for the next frame (the end of the LSC active region is reached). The host controller can perform these changes upon receiving an interrupt.

#### 8.3.4.2.5 ISS ISP ISIF Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the ISIF. It can be used as a quick checklist. More detailed register setting constraints can be found in the individual register descriptions.

- PCLK cannot be higher than 200 MHz.
- If SDRAM output port is enabled:
  - The memory output line offset and address must be on 32-byte boundaries.
  - [ISIF\\_LNH\[14:0\]](#) LNH -1 must be a multiple of 32 bytes.
  - [ISIF\\_SPH](#), [ISIF\\_LNH](#), [ISIF\\_SLV0](#), [ISIF\\_SLV1](#), and [ISIF\\_LNV](#) must be cleared to 0 within the same VD period that the [ISIF\\_SYNCEN\[1\]](#) DWEN bit is cleared to 0.
  - [ISIF\\_SPH](#), [ISIF\\_LNH](#), [ISIF\\_SLV0](#), [ISIF\\_SLV1](#), and [ISIF\\_LNV](#) must be set from 0 to the appropriate values within the same VD period that the [ISIF\\_SYNCEN\[1\]](#) DWEN bit is set to 1.
- In RAW input mode:

- [ISIF\\_CCDCFG\[4\]](#) YCINSWP must be set to 0.
- If DPCM compression is enabled:
  - Horizontal culling must not be used. Use the input formatter instead.
- For 2D-LSC:
  - $N \leq M$  (where  $M$  = horizontal downsampling factor,  $N$  = vertical downsampling factor)
  - The [ISIF\\_2DLSCINI](#) register values must be even numbers.
  - Maximum widths with respect to selected  $M$  value (see [Table 8-354](#))

**Table 8-354. ISS ISP ISIF Maximum Line Width Versus M Value**

M	Maximum Line Width
8	2040
16	4080
32	8160
64	16,320
128	16,384

### 8.3.4.3 ISS ISP IPIPEIF Programming Model

This section discusses issues related to the software control of the IPIPEIF module. It lists the registers that are required to be programmed in different modes, describes how to enable and disable the IPIPEIF module and how to check the status of the IPIPEIF module, discusses the different register access types, and enumerates several programming constraints.

#### 8.3.4.3.1 ISS ISP IPIPEIF Hardware Setup/Initialization

This section discusses the configuration of the IPIPEIF module required before image processing can begin.

##### 8.3.4.3.1.1 ISS ISP IPIPEIF Reset Behavior

Upon hardware reset of the ISP ([ISP5\\_SYSCONFIG\[1\]](#) SOFTRESET = 0x1), all of the registers in the IPIPEIF are reset to their reset values.

##### 8.3.4.3.1.2 ISS ISP IPIPEIF Register Setup

Before enabling the IPIPEIF, the hardware must be properly configured through register writes. [Table 8-355](#) identifies the register parameters that must be programmed before enabling the IPIPEIF module (depending on the functions needed).

**Table 8-355. ISS ISP IPIPEIF Required Configuration Parameters**

Step	Configuration Required	Value
General settings		
Select the input sources for IPIPEIF.	<a href="#">IPIPEIF_CFG1[15:14]</a> INPSRC1 <a href="#">IPIPEIF_CFG1[3:2]</a> INPSRC2	
Select VD sync polarity.	<a href="#">IPIPEIF_CFG2[2]</a> VDPOL	
Select HD sync polarity.	<a href="#">IPIPEIF_CFG2[1]</a> HDPOL	
Select the interrupt source.	<a href="#">IPIPEIF_CFG2[0]</a> INTSW	
Select the input clock source.	<a href="#">IPIPEIF_CFG1[10]</a> CLKSEL	
Set the clock divider value.	<a href="#">IPIPEIF_CFG1[15:0]</a> CLKDIV	
Set the data type: YUV or RAW.	<a href="#">IPIPEIF_CFG2[3]</a> YUV16	
Settings to perform if data is from BL		
Set HD setting.	<a href="#">IPIPEIF_PPLN[12:0]</a> PPLN	
Set VD setting.	<a href="#">IPIPEIF_LPFR[12:0]</a> LPFR	
Set the number of valid pixels in a line.	<a href="#">IPIPEIF_HNUM[12:0]</a> HNUM	

**Table 8-355. ISS ISP IPIPEIF Required Configuration Parameters (continued)**

Step	Configuration Required	Value
Set the number of valid lines.	IPIPEIF_VNUM[12:0] VNUM	
Set memory address information.	IPIPEIF_ADDRU[10:0] ADDRU IPIPEIF_ADDRL[15:0] ADDRDL IPIPEIF_ADOFS[11:0] ADOFS	
Set one-shot mode, if needed.	IPIPEIF_CFG1[0] ONESHOT	
Use SYNCOFF function only for double-buffering. See Section 8.3.3.3.2.1 for details.	IPIPEIF_ENABLE[1] SYNCOFF	
Settings to perform for RAW data from ISIF or BL		
Select the unpack function.	IPIPEIF_CFG1[9:8] UNPACK	
Settings to perform for YUV for data from ISIF		
Set the data type to YUV.	IPIPEIF_CFG2[3] YUV16	
If YUV16 = 0x1, enables or not the conversion from 8 bits to 16 bits	IPIPEIF_CFG2[6] YUV8	
If YUV16 = 0x1, set the way the data is unpacked.	IPIPEIF_CFG2[7] YUV8P	
DCPM function		
Enable or disable DPCM decompression.	IPIPEIF_DPCM[0] ENA	
Select DPCM prediction mode.	IPIPEIF_DPCM[1] PRED	
Select DPCM bit mode.	IPIPEIF_DPCM[2] BITS	
Select inverse A-Law function.	IPIPEIF_CFG1[9:8] UNPACK	
Select SDRAM read data shift. - For RAW data - DPCM enabled and A- Law disabled - or DPCM disabled	IPIPEIF_CFG1[13:11] DATASFT	
Dark frame subtraction (DFS) function		
RAW data from VP (video port), BL(buffer logic, SDRAM) or ISIF		
Set defect pixel correction (DPC1) for VP or ISIF inputs.	IPIPEIF_DPC1[12] ENA	
Set the associated threshold for DPC1.	IPIPEIF_DPC1[11:0] TH	
Set defect pixel correction (DPC2) for BL inputs.	IPIPEIF_DPC2[12] ENA	
Set the associated threshold for DPC1.	IPIPEIF_DPC2[11:0] TH	
Set the direction of subtraction.	IPIPEIF_CFG2[5] DFSDIR	
Set the averaging filter, horizontal pixel decimator, and gain function for IPIPE data path.		
Select averaging filter function.	IPIPEIF_CFG1[7] AVGFILT	
Select horizontal pixel decimation function.	IPIPEIF_CFG1[1] DECIM	
Set horizontal resizing value.	IPIPEIF_RSZ[6:0] RSZ	
Set the resizer initial position.	IPIPEIF_INIRSZ[13] ALNSYNC IPIPEIF_INIRSZ[12:0] INIRSZ	
Set the data gain (only for RAW data).	IPIPEIF_GAIN[9:0] GAIN	
Set the output clipping value.	IPIPEIF_OCLIP[11:0] OCLIP	
Set the averaging filter and horizontal pixel decimator function for H3A data path.		
Select averaging filter function.	IPIPEIF_RSZ3A[8] AVGFILT	
Select horizontal pixel decimation function.	IPIPEIF_RSZ3A[9] DECIM	
Set horizontal resizing value.	IPIPEIF_RSZ3A[6:0] RSZ	

**Table 8-355. ISS ISP IPIPEIF Required Configuration Parameters (continued)**

Step	Configuration Required	Value
Set the resizer initial position.	<a href="#">IPIPEIF_RSZ3A[13]</a> ALNSYNC <a href="#">IPIPEIF_RSZ3A[12:0]</a> INIRSZ	

For information about YUV data coming and unpacking from the ISIF module, see [Section 8.3.3.3.13](#).

#### 8.3.4.3.2 ISS ISP IPIPEIF Enable/Disable Hardware

When [IPIPEIF\\_CFG1\[15:14\]](#) INPSRC1 or [IPIPEIF\\_CFG1\[3:2\]](#) INPSRC2 = 0, the IPIPEIF does not need to be enabled: it receives data from the VP and pushes it to the ISIF, H3A, IPIPE, and RESIZER modules.

If [IPIPEIF\\_CFG1\[15:14\]](#) INPSRC1 or [IPIPEIF\\_CFG1\[3:2\]](#) INPSRC2 ≠ 0, the IPIPEIF module begins to fetch data from the BL by setting the [IPIPEIF\\_ENABLE\[0\]](#) ENABLE bit. Writing the enable bit must be the last step of the configuration.

When the input source is the BL, the IPIPEIF can optionally operate in one-shot mode or continuous mode by setting the [IPIPEIF\\_CFG1\[0\]](#) ONESHOT parameter. If one-shot mode is enabled, then after enabling the IPIPEIF, the [IPIPEIF\\_ENABLE\[0\]](#) ENABLE bit is automatically turned off (set to 0) and only a single frame is processed from memory. In this mode, fetching and processing of the frame begins immediately upon setting the [IPIPEIF\\_ENABLE\[0\]](#) ENABLE bit.

When the input source is the ISIF, processing of the frame is dependent on the timing of the ISIF. To ensure that data from the ISIF is not missed, the IPIPEIF must be enabled before the ISIF. In this way, the IPIPEIF waits for data from the ISIF.

When the IPIPEIF is in continuous mode, it can be disabled by clearing the [IPIPEIF\\_ENABLE\[0\]](#) ENABLE bit after processing of the last frame. The disable takes place immediately because it is a busy-writeable register.

An EOF interrupt can indicate to other modules that the frame treatment is finished. See [Section 8.3.3.3.15](#) for details.

#### 8.3.4.3.3 ISS ISP IPIPEIF Register Accessibility During Frame Processing

There are two types of register access in the IPIPEIF:

- Shadow registers

These registers/fields can be read and written (if the field is writeable) at any time. However, the written values take effect only at the start of a frame (VD rising edge). Reads still return the most recent write even though the settings are not used until the next start of frame. The following are the shadowed registers in the IPIPEIF.

<a href="#">IPIPEIF_PPLN</a>	<a href="#">IPIPEIF_ADDRU</a>	<a href="#">IPIPEIF_CFG1[1]</a> DECIM
<a href="#">IPIPEIF_HNUM</a>	<a href="#">IPIPEIF_ADOFS</a>	<a href="#">IPIPEIF_CFG1[7]</a> AVGFILT
<a href="#">IPIPEIF_VNUM</a>	<a href="#">IPIPEIF_ENABLE[1]</a> SYNCOFF	<a href="#">IPIPEIF_RSZ</a>
<a href="#">IPIPEIF_GAIN</a>	<a href="#">IPIPEIF_RSZ3A[9]</a> DECIM	<a href="#">IPIPEIF_RSZ3A[8]</a> AVGFILT
<a href="#">IPIPEIF_RSZ3A[6:0]</a> RSZ		

- Busy-writeable registers

These registers/fields can be read or written even if the module is busy. Changes to the underlying settings takes place instantaneously. Registers that are not shadowed are busy-writeable.

Only for busy-writeable registers, the ideal procedure for changing the IPIPEIF registers is IF (PCR.BUSY == 0) OR IF (EOF interrupt occurs):

- Disable IPIPEIF ([IPIPEIF\\_ENABLE\[0\]](#) ENABLE = 0x0).
- Change registers.
- Enable IPIPEIF ([IPIPEIF\\_ENABLE\[0\]](#) ENABLE = 0x1).

**8.3.4.3.4 ISS ISP IPIPEIF Interframe Operations**

Between frames, it may be necessary to enable or disable functions or to modify the memory pointers. Because several of the registers are shadowed, these modifications can take place any time before the end of the frame, and the data is latched in for the next frame. The host controller can perform these changes upon receiving an interrupt.

When reading input data from the BL, the host must check IPIPEIF\_DTUF or the [ISP5\\_IRQSTATUS\\_RAW2\\_i\[1\]](#) IPIPEIF\_UDF register (if this event is enabled and mapped to the ISP IRQ lines) during vertical blanking to see if an underflow occurred.

If an underflow occurs it means the system is congested because too much bandwidth is being generated. It is most likely that the scenario being passed is too memory bandwidth-intensive. If the bit is set, software must clear the bit. If the process is taking place from memory-to-memory, the software may try to re-initiate the data flow. Software must decide the sequence to execute after an underflow.

**8.3.4.3.5 ISS ISP IPIPEIF Summary of Constraints**

The following is a list of register configuration constraints to adhere to when programming the IPIPEIF module. It can be used as a quick checklist. More detailed register setting constraints can be found in the individual register descriptions.

- If BL is the source for the IPIPEIF:
  - The memory output line offset and address must be on 32-byte boundaries.
  - In DFS block, [IPIPEIF\\_LPFR](#) must be > 0, because the first line cannot be fetched.
  - [IPIPEIF\\_PPLN](#) > [IPIPEIF\\_HNUM](#)
  - [IPIPEIF\\_LPFR](#) > [IPIPEIF\\_VNUM](#) + 1
- There are restrictions on [IPIPEIF\\_CFG1\[15:14\]](#) INPSRC1 and [IPIPEIF\\_CFG1\[3:2\]](#) INPSRC2 combinations. See [Table 8-310](#) for details.

**8.3.4.4 ISS ISP IPIPE Programming Model**

This section covers the low-level hardware programming sequences for the configuration and use of the ISS PIPE module.

**8.3.4.4.1 Global Initialization**

**8.3.4.4.1.1 Surrounding Modules Global Initialization**

This initialization of surrounding modules is based on the integration of the ISS ISP.

**8.3.4.4.1.2 ISS IPIPE Global Initialization**

**8.3.4.4.1.2.1 Main Sequence – ISS ISP IPIPE Global Initialization**

This procedure initializes the ISS ISP PIPE modules after a POR or software reset.

**Table 8-356. ISS Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Set the IPIPE processing modes.	<a href="#">IPIPE_SRC_MODE</a> [0] OST	0x-
Select the input and output formats.	<a href="#">IPIPE_SRC_FMT</a> [1:0] FMT	0x-
Select the IPIPEIF module as a source.	<a href="#">IPIPE_SRC_MODE</a> [1] WRT	0x1
Set the vertical start position of the window to process.	<a href="#">IPIPE_SRC_VPS</a> [15:0] VAL	0x-
Set the horizontal start position of the window to process.	<a href="#">IPIPE_SRC_HPS</a> [15:0] VAL	0x-
Set the vertical size of the processing area.	<a href="#">IPIPE_SRC_VSZ</a> [12:0] VAL	0X-

**Table 8-356. ISS Global Initialization (continued)**

Step	Register/Bit Field/Programming Model	Value
Set the horizontal size of the processing area.	IPIPE_SRC_HSZ[12:1] VAL	0x-
Enable clk_arm_g0.	IPIPE_GCK_MMR[0] REG	0x-
Enable the clk_pix_g3.	IPIPE_GCK_PIX[3] G3	0x-
Enable the clk_pix_g2.	IPIPE_GCK_PIX[2] G2	0x-
Enable the clk_pix_g1.	IPIPE_GCK_PIX[1] G1	0x-
Enable the clk_pix_g0.	IPIPE_GCK_PIX[0] G0	0x-
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

#### 8.3.4.4.1.2.2 Subsequence – ISS ISP IPIPE Defect Pixel Correction Initialization

##### 8.3.4.4.1.2.2.1 Subsequence – ISS ISP IPIPE LUT Defect Pixel Correction (LUT DPC)

**Table 8-357. ISS ISP IPIPE LUT Defect Pixel Correction (LUT DPC)**

Step	Register/Bit Field/Programming Model	Value
Set the address of the first valid data in the LUT.	IPIPE_DPC_LUT_ADR[9:0] ADR	0x-
Set the LUT type.	IPIPE_DPC_LUT_SEL[1] TBL	0x-
Set the size of the LUT.	IPIPE_DPC_LUT_SIZ[9:0] SIZ	0x-
Set the dot replacement correction method .	IPIPE_DPC_LUT_SEL[0] DOT	0x-
Enable LUT DPC.	IPIPE_DPC_LUT_EN[0] EN	0x1

#### 8.3.4.4.1.2.3 Subsequence – ISS ISP IPIPE White Balance Initialization

**Table 8-358. ISS ISP IPIPE White Balance Initialization**

Step	Register/Bit Field/Programming Model	Value
Set the offset before white balance.	IPIPE_WB2_OFT_R[11:0] VAL IPIPE_WB2_OFT_GR[11:0] VAL IPIPE_WB2_OFT_GB[11:0] VAL IPIPE_WB2_OFT_B[11:0] VAL	0x-
Set the white balance gain.	IPIPE_WB2_WGN_R[12:0] VAL IPIPE_WB2_WGN_GR[12:0] VAL IPIPE_WB2_WGN_GB[12:0] VAL IPIPE_WB2_WGN_B[12:0] VAL	0x-

#### 8.3.4.4.1.2.4 Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization

**Table 8-359. ISS ISP IPIPE RGB2RGB Blending Module Initialization**

Step	Register/Bit Field/Programming Model	Value
Set gain range.	IPIPE_RGB1_MUL_RR[11:0] VAL IPIPE_RGB1_MUL_GR[11:0] VAL IPIPE_RGB1_MUL_BR[11:0] VAL IPIPE_RGB1_MUL_RG[11:0] VAL IPIPE_RGB1_MUL_GG[11:0] VAL IPIPE_RGB1_MUL_BG[11:0] VAL IPIPE_RGB1_MUL_RB[11:0] VAL IPIPE_RGB1_MUL_GB[11:0] VAL IPIPE_RGB1_MUL_BB[11:0] VAL	0x-
Se the offset range for each component.	IPIPE_RGB1_OFT_OR[12:0] VAL IPIPE_RGB1_OFT_OG[12:0] VAL IPIPE_RGB1_OFT_OB[12:0] VAL	0x-

**8.3.4.4.1.2.5 Subsequence – ISS ISP IPIPE Gamma Correction Module Initialization**
**Table 8-360. ISS ISP IPIPE Gamma Correction Module Initialization**

Step	Register/Bit Field/Programming Model	Value
Select the gamma table.	IPIPE_GMM_CFG[4] TBL	0x-
Set the size of the gamma table.	IPIPE_GMM_CFG[6:5]SIZ	0x-
(optional) Insert bypass bit for each color.	IPIPE_GMM_CFG[0] BYPR IPIPE_GMM_CFG[1] BYPG IPIPE_GMM_CFG[2] BYPB	0x1

**8.3.4.4.1.2.6 Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization**
**Table 8-361. ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization**

Step	Register/Bit Field/Programming Model	Value
Set the gain range.	IPIPE_RGB2_MUL_RR[10:0] VAL IPIPE_RGB2_MUL_GR[10:0] VAL IPIPE_RGB2_MUL_BR[10:0] VAL IPIPE_RGB2_MUL_RG[10:0] VAL IPIPE_RGB2_MUL_GG[10:0] VAL IPIPE_RGB2_MUL_BG[10:0] VAL IPIPE_RGB2_MUL_RB[10:0] VAL IPIPE_RGB2_MUL_GB[10:0] VAL IPIPE_RGB2_MUL_BB[10:0] VAL	0x-
Set the offset.	IPIPE_RGB2_OFT_OR[10:0] VAL IPIPE_RGB2_OFT_OG[10:0] VAL IPIPE_RGB2_OFT_OB[10:0] VAL	0x-

**8.3.4.4.1.2.7 Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization**
**Table 8-362. ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization**

Step	Register/Bit Field/Programming Model	Value
Set the brightness control.	IPIPE_YUV_ADJ[15:8] BRT	0x-
Set the contrast control.	IPIPE_YUV_ADJ[7:0] CRT	0x-
Configure the gain range.	IPIPE_YUV_MUL_RY[11:0]VAL IPIPE_YUV_MUL_GY[11:0]VAL IPIPE_YUV_MUL_BY[11:0]VAL IPIPE_YUV_MUL_RCB[11:0]VAL IPIPE_YUV_MUL_GCB[11:0]VAL IPIPE_YUV_MUL_BCB[11:0]VAL IPIPE_YUV_MUL_RCR[11:0]VAL IPIPE_YUV_MUL_GCR[11:0]VAL IPIPE_YUV_MUL_BCR[11:0]VAL	0x-
Set the output offset value for Y.	IPIPE_YUV_OFT_Y[10:0] VAL	0x-
Set the output offset value for Cr.	IPIPE_YUV_OFT_CR[10:0] VAL	0x-
Set the output offset value for Cb.	IPIPE_YUV_OFT_CB[10:0] VAL	0x-

**8.3.4.4.1.2.8 Subsequence – ISS ISP IPIPE 4:2:2 Conversion Module Initialization**
**Table 8-363. ISS ISP IPIPE 4:2:2 Conversion Module Initialization**

Step	Register/Bit Field/Programming Model	Value
If: RAW BAYER data set as an input?	IPIPE_SRC_FMT.[1:0] FMT	= 0x0
Select the Y and Cb/Cr sampling point.	IPIPE_YUV_PHS[0] POS	0x-
Enable 4:2:2 conversion module.	IPIPE_YUV_PHS[1] PLF	0x-
<b>ELSE</b>		

### 8.3.4.4.1.2.9 Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization

**Table 8-364. ISS ISP IPIPE 2D Edge Enhancer Initialization**

Step	Register/Bit Field/Programming Model	Value
Set the merging method.	IPIPE_YEE_TYP[0] SEL	0x-
Set the downshift length of HPF in the edge enhancer.	IPIPE_YEE_SHF[3:0] SHF	0x-
Set the edge enhancer lower threshold before referring to the LUT.	IPIPE_YEE_THR[5:0] VAL	0x-
Set the multiplier coefficient in the HPF.	IPIPE_YEE_MUL_00[9:0] VAL IPIPE_YEE_MUL_01[9:0] VAL IPIPE_YEE_MUL_02[9:0] VAL IPIPE_YEE_MUL_10[9:0] VAL IPIPE_YEE_MUL_11[9:0] VAL IPIPE_YEE_MUL_12[9:0] VAL IPIPE_YEE_MUL_20[9:0] VAL IPIPE_YEE_MUL_21[9:0] VAL IPIPE_YEE_MUL_22[9:0] VAL	0x-
Set the edge sharpener HPF value lower limit.	IPIPE_YEE_E_THR_1[11:0] VAL	0x-
Set the edge sharpener HPF value upper limit.	IPIPE_YEE_E_THR_2[5:0] VAL	0x-
Set the edge sharpener gain value on gradient.	IPIPE_YEE_G_GAN[7:0] VAL	0x-
Set the edge sharpener gain value.	IPIPE_YEE_E_GAN[7:0] VAL	0x-
Set the edge sharpener offset value on gradient.	IPIPE_YEE_G_OFT[5:0] VAL	0x-
Enable the 2D edge enhancer.	IPIPE_YEE_EN[0] EN	0x1

### 8.3.4.4.1.2.10 Subsequence – ISS ISP IPIPE Histogram Initialization

**Table 8-365. ISS ISP IPIPE Histogram Initialization**

Step	Register/Bit Field/Programming Model	Value
Select input source.	IPIPE_HST_SEL[2] SEL	0x-
Select the Bayer mode.	IPIPE_HST_SEL[1:0] TYP	0x-
Set the number of bins.	IPIPE_HST_PARA[13:12] BIN	0x-
Enable region(area) 0.	IPIPE_HST_PARA[0] RGN0	0x-
Enable region(area) 1.	IPIPE_HST_PARA[1] RGN1	0x-
Enable region(area) 2.	IPIPE_HST_PARA[2] RGN2	0x-
Enable region(area) 3.	IPIPE_HST_PARA[3] RGN3	0x-
Set the vertical position of region 0.	IPIPE_HST_0_VPS[12:1] VAL	0x-
Set the vertical position of region 1.	IPIPE_HST_1_VPS[12:1] VAL	0x-
Set the vertical position of region 2.	IPIPE_HST_2_VPS[12:1] VAL	0x-
Set the vertical position of region 3.	IPIPE_HST_3_VPS[12:1] VAL	0x-
Set the horizontal position of region 0.	IPIPE_HST_0_HPS[12:1] VAL	0x-
Set the horizontal position of region 1.	IPIPE_HST_0_HPS[12:1] VAL	0x-
Set the horizontal position of region 2.	IPIPE_HST_0_HPS[12:1] VAL	0x-
Set the horizontal position of region 3.	IPIPE_HST_0_HPS[12:1] VAL	0x-
Set the vertical size of region 0.	IPIPE_HST_0_VSZ[12:1] VAL	0x-
Set the vertical size of region 1.	IPIPE_HST_1_VSZ[12:1] VAL	0x-
Set the vertical size of region 2.	IPIPE_HST_2_VSZ[12:1] VAL	0x-
Set the vertical size of region 3.	IPIPE_HST_3_VSZ[12:1] VAL	0x-
Set the horizontal size of region 0.	IPIPE_HST_0_HSZ[12:1] VAL	0x-
Set the horizontal size of region 1.	IPIPE_HST_1_HSZ[12:1] VAL	0x-
Set the horizontal size of region 2.	IPIPE_HST_2_HSZ[12:1] VAL	0x-
Set the horizontal size of region 3.	IPIPE_HST_3_HSZ[12:1] VAL	0x-



**Table 8-365. ISS ISP IPIPE Histogram Initialization (continued)**

Step	Register/Bit Field/Programming Model	Value
Enable selection of the color pattern 0 (R).	IPIPE_HST_PARA[4] COL0	0x-
Enable selection of the color pattern 1 (G).	IPIPE_HST_PARA[5] COL1	0x-
Enable selection of the color pattern 2 (B).	IPIPE_HST_PARA[6] COL2	0x-
Enable selection of the color pattern 3 (Y).	IPIPE_HST_PARA[7] COL3	0x-
Set the shift length of the input data.	IPIPE_HST_PARA[11:8] SHF	0x-
Enable/disable histogram memory clear.	IPIPE_HST_TBL[1] CLR	0x-
Select table for store the histogram data.	IPIPE_HST_TBL[1] CLR	0x-
<b>IF:</b> Is input from noise filter?	IPIPE_HST_SEL[2] SEI	0x0
Gain for color for R.	IPIPE_HST_MUL_R[7:0]GAIN	0x-
Gain for color for GR.	IPIPE_HST_MUL_GR[7:0]GAIN	0x-
Gain for color for GB.	IPIPE_HST_MUL_GB[7:0]GAIN	0x-
Gain for color for B.	IPIPE_HST_MUL_B[7:0]GAIN	0x-
<b>ENDIF</b>		
Select processing mode.	IPIPE_HST_MODE[0]OST	0x-
Enable histogram.	IPIPE_HST_EN[0] EN	0x1

#### 8.3.4.4.1.2.11 Subsequence – ISS ISP IPIPE Boxcar Initialization

**Table 8-366. ISS ISP IPIPE Boxcar Initialization**

Step	Register/Bit Field/Programming Model	Value
Select the processing mode.	IPIPE_BOX_MODE[0] OST	0x-
Set the size of blocks.	IPIPE_BOX_TYP[0] SEL	0x-
Set the higher 16 bits of the first address of output in memory.	IPIPE_BOX_SDR_SAD_H[15:0] VAL	0x-
Set the lower 16 bits of the first address of output in memory.	IPIPE_BOX_SDR_SAD_L[15:5] VAL	0x-
Set the downshift value.	IPIPE_BOX_SHF[2:0] VAL	0x-
Enable boxcar.	IPIPE_BOX_EN[0] EN	0x1

#### 8.3.4.4.2 ISS ISP IPIPE Operational Modes Configuration

##### 8.3.4.4.2.1 ISS ISP IPIPE Processing Path: Case 1 Configuration

**Table 8-367. ISS ISP IPIPE Processing Path: Case 1 Configuration**

Step	Register/Bit Field/Programming Model	Value
Set the IPIPEIF module as a source.	IPIPE_SRC_MODE[1] WRT	0x1
Select one-shot processing mode.	IPIPE_SRC_MODE[0] OST	0x1
Select processing path: Case 1	IPIPE_SRC_FMT[1:0] FMT	0x0
Configure IPIPE DPC.	See <a href="#">Section 8.3.4.4.1.2.2, Subsequence – ISS ISP IPIPE Defect Pixel Correction Initialization.</a>	
Configure IPIPE white balance.	See <a href="#">Section 8.3.4.4.1.2.3, Subsequence – ISS ISP IPIPE White Balance Initialization.</a>	
Configure IPIPE RGB2RGB blending module.	See <a href="#">Section 8.3.4.4.1.2.4, Subsequence – ISS ISP IPIPE RGB2RGB Blending Module Initialization.</a>	
Configure IPIPE gamma correction module.	See <a href="#">Section 8.3.4.4.1.2.5, Subsequence – ISS ISP IPIPE Gamma Correction Module Initialization.</a>	
Configure IPIPE 2nd RGB2RGB conversion matrix.	See <a href="#">Section 8.3.4.4.1.2.6, Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization.</a>	

**Table 8-367. ISS ISP IPIPE Processing Path: Case 1 Configuration (continued)**

Step	Register/Bit Field/Programming Model	Value
Configure IPIPE RGB2YCbCr conversion matrix.	See <a href="#">Section 8.3.4.4.1.2.7, Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization.</a>	
Configure IPIEE histogram.	See <a href="#">Section 8.3.4.4.1.2.10, Subsequence – ISS ISP IPIPE Histogram Initialization.</a>	
Configure IPIPE 2D edge enhancer.	See <a href="#">Section 8.3.4.4.1.2.9, Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization.</a>	
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

**8.3.4.4.2 ISS ISP IPIPE Processing Path: Case 2 Configuration****Table 8-368. ISS ISP IPIPE Processing Path: Case 2 Configuration**

Step	Register/Bit Field/Programming Model	Value
Set the IPIPEIF module as a source.	IPIPE_SRC_MODE[1] WRT	0x1
Select one-shot processing mode.	IPIPE_SRC_MODE[0] OST	0x1
Select processing path: Case 2.	IPIPE_SRC_FMT[1:0] FMT	0x1
Configure IPIPE DPC.	See <a href="#">Section 8.3.4.4.1.2.2, Subsequence – ISS ISP IPIPE Defect Pixel Correction Initialization.</a>	
Configure IPIPE white balance.	See <a href="#">Section 8.3.4.4.1.2.3, Subsequence – ISS ISP IPIPE White Balance Initialization.</a>	
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

**8.3.4.4.3 ISS ISP IPIPE Processing Path: Case 3 Configuration****Table 8-369. ISS ISP IPIPE Processing Path: Case 3 Configuration**

Step	Register/Bit Field/Programming Model	Value
Set the IPIPEIF module as a source.	IPIPE_SRC_MODE[1] WRT	0x1
Select one-shot processing mode.	IPIPE_SRC_MODE[0] OST	0x1
Select processing path: Case 3.	IPIPE_SRC_FMT[1:0] FMT	0x2
Configure IPIPE boxcar.	See <a href="#">Section 8.3.4.4.1.2.11, Subsequence – ISS ISP IPIPE Boxcar Initialization.</a>	
Configure IPIEE histogram	See <a href="#">Section 8.3.4.4.1.2.10, Subsequence – ISS ISP IPIPE Histogram Initialization.</a>	
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

**8.3.4.4.4 ISS ISP IPIPE Processing Path: Case 4 Configuration****Table 8-370. ISS ISP IPIPE Processing Path: Case 4 Configuration**

Step	Register/Bit Field/Programming Model	Value
Set the IPIPEIF module as a source.	IPIPE_SRC_MODE[1] WRT	0x1
Select one-shot processing mode.	IPIPE_SRC_MODE[0] OST	0x1
Select processing path: Case 4.	IPIPE_SRC_FMT[1:0] FMT	0x3
Configure IPIPE 2D edge enhancer.	See <a href="#">Section 8.3.4.4.1.2.9, Subsequence - ISS ISP IPIPE 2D Edge Enhancer Initialization.</a>	
Enable IPIPE module.	IPIPE_SRC_EN[0] EN	0x1

### 8.3.4.5 ISS ISP RSZ Programming Model

This section discusses programming configuration steps related to software control of the RSZ. It lists the registers that must be programmed in different modes, describes how to enable and disable the RSZ and how to check the status of the image resizing procedure, and discusses the different register access types and several programming constraints.

#### 8.3.4.5.1 ISS ISP RSZ Hardware Setup/Initialization

This section discusses the configuration of the RSZ required before image processing can begin.

##### 8.3.4.5.1.1 ISS ISP RSZ Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the RSZ module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the RSZ. For more information, see [Section 8.3.3.5.4, ISS ISP RSZ Integration..](#)

**Table 8-371. ISS ISP RSZ Surrounding Modules Global Initialization**

Surrounding Modules	Comments
PRCM	Module Interface and functional clocks must be enabled. RSZ is part of ISP, which is part of ISS. To enable the clocks for ISS, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> . Also see <a href="#">Section 8.3.2.1.1, ISS ISP Clocks</a> .
(optional) MPU INTC	The MPU INTC must be configured to enable interrupts from the RSZ. For information about how to configure the local ISS interrupt channels, see <a href="#">Section 8.3.2.2, ISS ISP Interrupt Tree</a> , and <a href="#">Section 8.1.2.1.1, ISS Interrupt Merger</a> . Then, to configure the outside ISS boundary channels, see <a href="#">Chapter 17, Interrupt Controllers</a> .
(optional) IPIPE	Configure the IPIPE module to process and pass data to the RSZ module. See <a href="#">Section 8.3.4.4, ISS ISP IPIPE Programming Model</a> .
(optional) IPIPEIF	Configure the IPIPEIF module to process and pass data to the RSZ module. See <a href="#">Section 8.3.4.3, ISS ISP IPIPEIF Programming Model</a> .
(optional) BL	Configure the BL for reading back data from memory. See <a href="#">Section 8.3.4.7, ISS ISP BL Programming Model</a> .
(optional) CSI2	If the data comes from the sensor, configure the CSI2-A interface. See <a href="#">Section 8.2.5.4, ISS CSI2 Programming Model</a> .
(optional) CCP2	If the data comes from the sensor, configure the CCP2 interface. See <a href="#">Section 8.2.4.4, ISS CCP2 Programming Model</a> .

**NOTE:** The MPU INTC configurations are necessary if the interrupt-based communication modes are used. The IPIPE or IPIPEIF configuration is also necessary for putting data into the RSZ. Moreover, if data comes from memory, the BL must be configured. If data comes from the image sensor CSI2 or CCP2, the interface must be configured.

##### 8.3.4.5.1.2 ISS ISP RSZ Initial Register Setup

Before enabling the RSZ, the hardware must be properly configured through register writes. [Table 8-372](#) identifies the sequence to be used before enabling the hardware.

**Table 8-372. ISS ISP RSZ Initial Register Setup**

Step	Register/Bit Field/Programming Model	Value
Set the MMR clock to enable RSZ register read/write access.	<a href="#">RSZ_GCK_MMR[0]</a> MMR	0x1

**Table 8-372. ISS ISP RSZ Initial Register Setup (continued)**

Step	Register/Bit Field/Programming Model	Value
Determine whether or not a bypass mode will be used and set the RSZ core functional clock accordingly.	RSZ_GCK_SDR[0] CORE	0x-
<b>IF: Is the core clock enabled?</b>	RSZ_GCK_SDR[0] CORE	= 0x1
Enable RSZ engine clocks.	RSZ_SYSCONFIG[8] RSZA_CLK_EN RSZ_SYSCONFIG[9] RSZB_CLK_EN	0x1
<b>ELSE: The input data buffer is not used?</b>	RSZ_GCK_SDR[0] CORE	= 0x0
Set the bypass mode accordingly.	RSZ_SRC_FMT0[1] BYPASS	0x-
<b>ENDIF</b>		
Enable the RSZ core clock.	RSZ_GCK_SDR[0] CORE	0x1
Determine the upscale ratio and functional clock, and adjust the fractional clock divider as appropriate.	RSZ_FRACDIV[15:0] RSZ_FRACDIV	0x-
Set the low threshold of the RSZ input data buffer as appropriate.	RSZ_IN_FIFO_CTRL[28:16] THRLD_LOW	0x-
Set the high threshold of the RSZ input data buffer as appropriate.	RSZ_IN_FIFO_CTRL[12:0] THRLD_HIGH	0x-
Determine whether IPIPE or IPEPIF will be the source of input and set this bit accordingly.	RSZ_SRC_FMT0[0] SEL	0x-
Determine whether the RSZ will be processing as long as data is present in the input buffer, or whether it will wait for a WEN signal from the input source (IPIPE or IPIPEIF) to process lines only arrived during WEN high state.	RSZ_SRC_MODE[1] WRT	0x-
Select the processing mode (one shot or free running). This bit controls the RSZ module. There are also additional mode settings for the two RSZ engines within. After RSZ reset, the mode is automatically set to free running.	RSZ_SRC_MODE[0] OST	0x-
Configure the number of interrupt intervals for writing lines into CBUFF for the two RSZ engines.	RSZ_IRQ_RZA[12:0] RZA RSZ_IRQ_RZB[12:0] RZB	0x-
Enable the RSZ module.	RSZ_SRC_EN[0] EN	0x1
Enable the RSZ engine A.	RZA_EN[0] EN	0x1
Enable the RSZ engine B.	RZB_EN[0] EN	0x1

**NOTE:** RSZ engine A or RSZ engine B must be enabled after setting all needed configuration parameters.

### 8.3.4.5.1.3 ISS ISP RSZ Reset Behavior

Because the RSZ module has no software reset, software can issue one at the ISP level. Moreover, upon hardware reset, all the registers in the RSZ are reset to their reset values. [Table 8-373](#) identifies the proper software sequence before and after RSZ reset issues at the ISP level.

**Table 8-373. ISS ISP RSZ Reset Behavior**

Step	Register/Bit Field/Programming Model	Value
<b>ELSE IF: Is FIFO overflow or blanking error present?</b>	ISP5_IRQSTATUS_[18] RSZ_FIFO_IN_OVF or ISP5_IRQSTATUS_[19] RSZ_FIFO_IN_BLK_ERR	= 0x1

**Table 8-373. ISS ISP RSZ Reset Behavior (continued)**

Step	Register/Bit Field/Programming Model	Value
Clear the FIFO blanking event.	ISP5_IRQENABLE_CLR_i[19] RSZ_FIFO_IN_BLK_ERR	0x1
Clear the overflow event.	SP5_IRQENABLE_CLR_i[18] RSZ_FIFO_IN_OVF	0x1
<b>ELSE: No FIFO overflow or blanking error is present?</b>		
Wait until there is no DMA process and the module ready for reset.	ISP5_SYSCONFIG[1] SOFTRESET and ISP5_IRQSTATUS_i[15] RSZ_INT_DMA	= 0x0
<b>ENDIF</b>		
Disable the source data.	RSZ_SRC_EN[0] EN	0x0
Reset the RSZ at the ISP level.	ISP5_SYSCONFIG[1] SOFTRESET	0x1
Check whether the RSZ has been reset properly.	ISP5_SYSCONFIG[1] SOFTRESET	0x-

**8.3.4.5.2 ISS ISP RSZ Global Image Processing Settings**

This section discusses the configuration of the RSZ global settings before/during image frame processing. [Table 8-374](#) identifies the global RSZ functional sequence, which includes global control, frame settings, bandwidth control and reversal output image frames.

**Table 8-374. ISS ISP RSZ Global Image Processing Settings**

Step	Register/Bit Field/Programming Model	Value
Initialize the RSZ module.	See <a href="#">Section 8.3.4.5.1</a> .	
Determine input data type and configure the source parameters.	See <a href="#">Section 8.3.4.5.2.1</a> .	
Determine if vertical flip of the output image will be performed by the two RSZ engines and set accordingly.	RSZ_SEQ[1] VRVA RSZ_SEQ[3] VRVB	0x-
Determine if horizontal flip of the output image will be performed by the two RSZ engines and set accordingly.	RSZ_SEQ[0] HRVA RSZ_SEQ[2] HRVB	0x-
Set the baseline address of the RSZ output to CBUFF. Best performance can be achieved by assigning address on a 128-byte boundary.	RZA_SDR_Y_BAD_H[15:0] Y_BAD_H RZA_SDR_Y_BAD_L[15:0] Y_BAD_L RZB_SDR_Y_BAD_H[15:0] Y_BAD_H RZB_SDR_Y_BAD_L[15:0] Y_BAD_L RZA_SDR_C_BAD_H[15:0] C_BAD_H RZA_SDR_C_BAD_L[15:0] C_BAD_L RZB_SDR_C_BAD_H[15:0] C_BAD_H RZB_SDR_C_BAD_L[15:0] C_BAD_L	0x-
Set the start address of the RSZ output to CBUFF. The first data output will be written to this address. If the first line of a frame is written at the beginning of the CBUFF memory, then SAD = BAD and PTR_S = 0. It is strongly advised to set up this address on a 128-byte boundary, which will lead to the best performance.	RZA_SDR_Y_SAD_H[15:0] Y_SAD_H RZA_SDR_Y_SAD_L[15:0] Y_SAD_L RZB_SDR_Y_SAD_H[15:0] Y_SAD_H RZB_SDR_Y_SAD_L[15:0] Y_SAD_L RZA_SDR_C_SAD_H[15:0] C_SAD_H RZA_SDR_C_SAD_L[15:0] C_SAD_L RZB_SDR_C_SAD_H[15:0] C_SAD_H RZB_SDR_C_SAD_L[15:0] C_SAD_L	0x-
Set the start pointer of the CBUFF internal counter. It must be set up as PTR_S = (SAD – BAD) / OFT. PTR_S is expressed in number of lines.	RZA_SDR_Y_PTR_S[12:0] Y_PTR_S RZB_SDR_Y_PTR_S[12:0] Y_PTR_S RZA_SDR_C_PTR_S[12:0] C_PTR_S RZB_SDR_C_PTR_S[12:0] C_PTR_S	0x-
Set the end pointer of the CBUFF internal counter. PTR_E is expressed in number of lines. The CBUFF can contain up to PTR_E lines.	RZA_SDR_Y_PTR_E[12:0] Y_PTR_E RZB_SDR_Y_PTR_E[12:0] Y_PTR_E RZA_SDR_C_PTR_E[12:0] C_PTR_E RZB_SDR_C_PTR_E[12:0] C_PTR_E	0x-

**Table 8-374. ISS ISP RSZ Global Image Processing Settings (continued)**

Step	Register/Bit Field/Programming Model	Value
Set the line of offset expressed in bytes between two lines in the CBUFF (Line 0 = SAD, Line 1 = SAD + 1 x OFT, Line 2 = SAD + 2 x OFT). PTR_E is expressed in number of lines. The CBUFF can contain up to PTR_E lines. Note: OFT does not necessarily correspond to the size of a line in a frame; it can be much bigger. The line offset must be a multiple of 128 bytes.	RZA_SDR_Y_OFT[16:0] Y_OFT RZB_SDR_Y_OFT[16:0] Y_OFT RZA_SDR_C_OFT[16:0] C_OFT RZB_SDR_C_OFT[16:0] C_OFT	0x-
Determine output data type, whether it will be flipped, and configure the output memory addresses.	See <a href="#">Section 8.3.4.5.2.2</a> .	

**NOTE:** When data output is 4:2:2 or 4:2:0-Y-only, the following values are not needed:

- RZx\_SDR\_C\_BAD\_x
- RZx\_SDR\_C\_SAD\_x
- RZx\_SDR\_C\_PTR\_S
- RZx\_SDR\_C\_PTR\_E
- RZx\_SDR\_C\_OFT

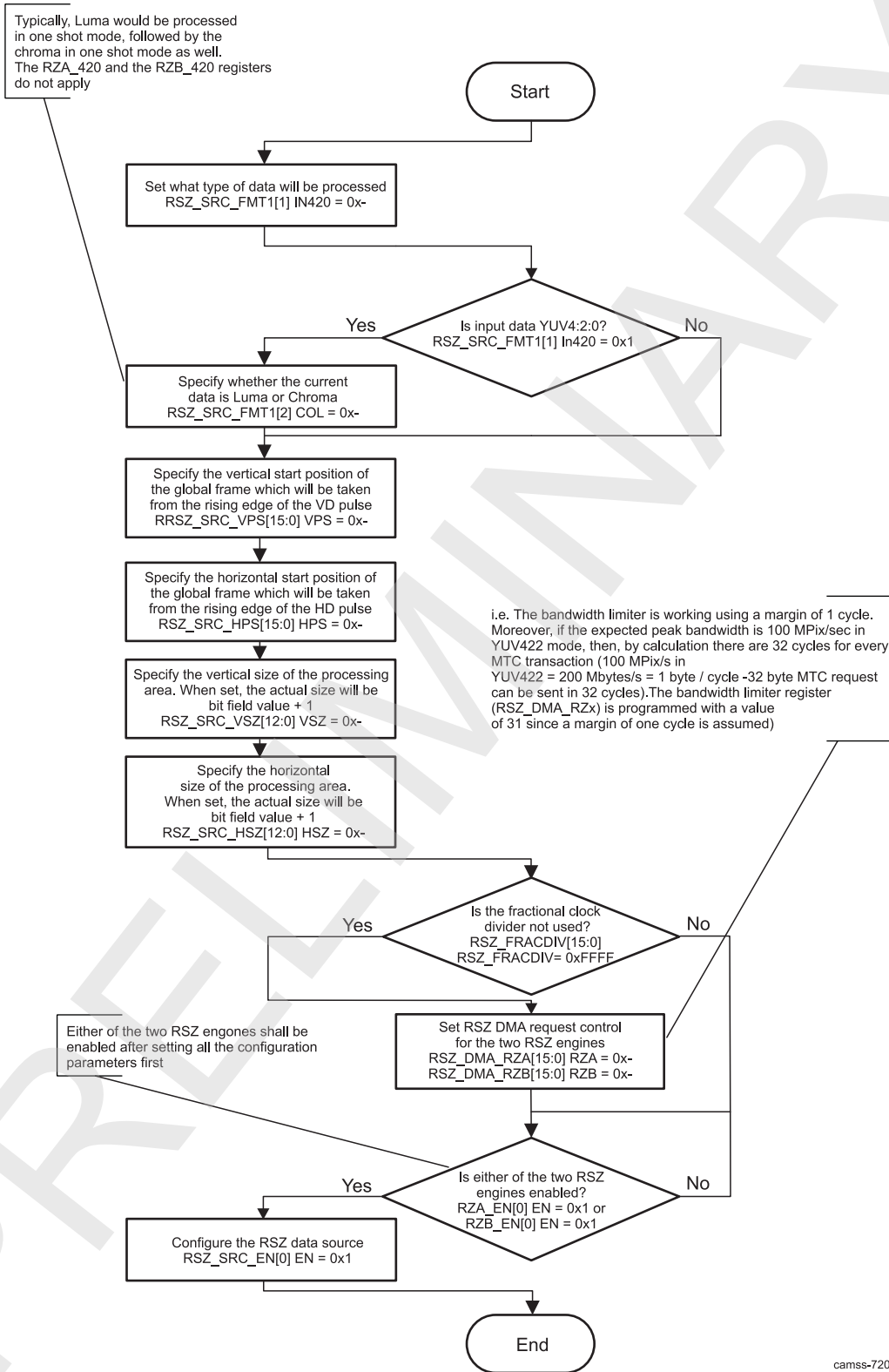
When output is 4:2:0-C-only, the following values are not needed:

- RZx\_SDR\_Y\_BAD\_x
- RZx\_SDR\_Y\_SAD\_x
- RZx\_SDR\_Y\_PTR\_S
- RZx\_SDR\_Y\_PTR\_E
- RZx\_SDR\_Y\_OFT

#### 8.3.4.5.2.1 ISS ISP RSZ Global Image Processing Settings – Subsequence 1

This procedure listed in [Figure 8-234](#) determines which of the RSZ engines are enabled and the type of input data, and configures the source data parameters

Figure 8-234. ISS ISP RSZ Global Image Processing Settings – Subsequence 1



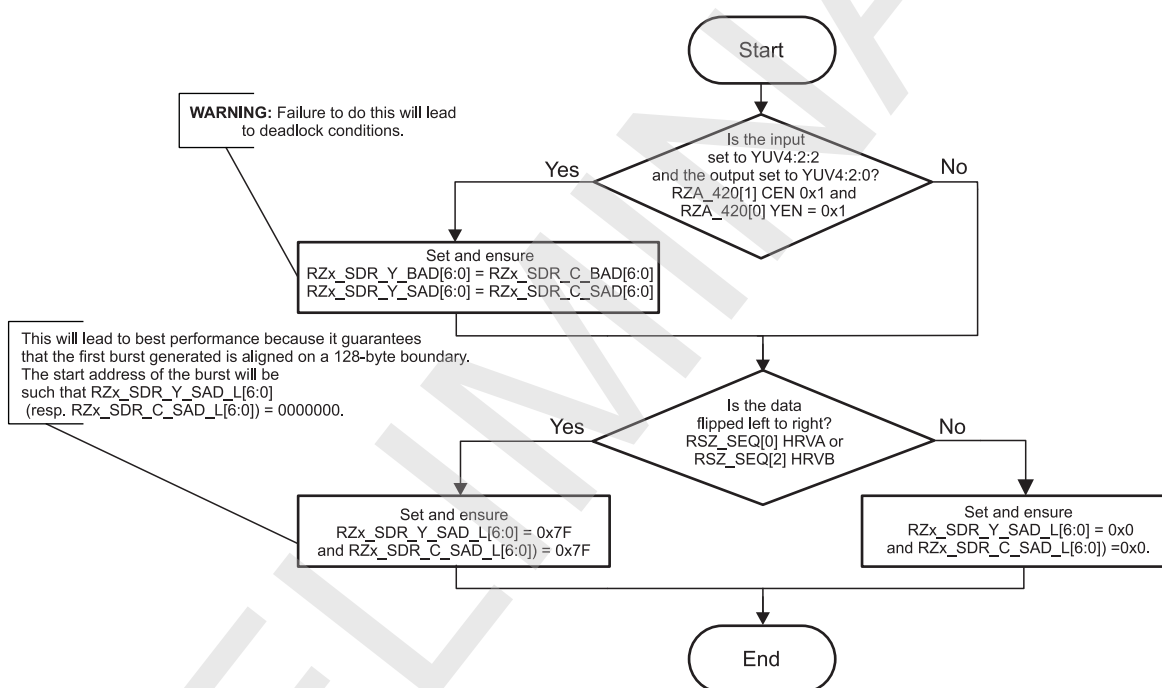
**Table 8-375. Register Call Summary for ISS ISP RSZ Global Image Processing Settings – Subsequence 1**

Register Name	Register Name	Register Name
<a href="#">RZA_420[1]</a> CEN	<a href="#">RZA_420[0]</a> YEN	<a href="#">RSZ_SRC_FMT1[1]</a> IN420
<a href="#">RZB_420[0]</a> CEN	<a href="#">RZB_420[1]</a> YEN	<a href="#">RSZ_SRC_FMT1[2]</a> COL
<a href="#">RSZ_SRC_EN[0]</a> EN	<a href="#">RSZ_SRC_VPS[15:0]</a> VPS	<a href="#">RSZ_SRC_HPS[15:0]</a> HPS
<a href="#">RSZ_SRC_VSZ[12:0]</a> VSZ	<a href="#">RSZ_SRC_HSZ[12:0]</a> HSZ	<a href="#">RSZ_FRACDIV[15:0]</a> <a href="#">RSZ_FRACDIV</a>
<a href="#">RSZ_DMA_RZA[15:0]</a> RZA	<a href="#">RSZ_DMA_RZB[15:0]</a> RZB	

### 8.3.4.5.2.2 ISS ISP RSZ Global Image Processing Settings – Subsequence 2

The procedure listed in [Figure 8-235](#) determines the type of output data and whether it will be flipped. Then it configures the RSZ engines accordingly.

**Figure 8-235. ISS ISP RSZ Global Image Processing Settings – Subsequence 2**



camss-721

**Table 8-376. Register Call Summary for ISS ISP RSZ Global Image Processing Settings – Subsequence 2**

Register Name	Register Name	Register Name
<a href="#">RZA_420[1]</a> CEN	<a href="#">RZA_420[0]</a> YEN	<a href="#">RZB_420[1]</a> CEN
<a href="#">RZB_420[0]</a> YEN	<a href="#">RZA_SDR_Y_SAD_L</a>	<a href="#">RZA_SDR_C_SAD_L</a>
<a href="#">RZB_SDR_Y_SAD_L</a>	<a href="#">RZB_SDR_C_SAD_L</a>	<a href="#">RSZ_SEQ[0]</a> HRVA
<a href="#">RSZ_SEQ[2]</a> HRVB		

### 8.3.4.5.3 ISS ISP RSZ Engines Interframe Image Processing Settings

This section discusses the configuration of the RSZ interframe image processing. [Table 8-377](#) identifies the setup sequence for the two different engines within the RSZ module.



**Table 8-377. ISS ISP RSZ Engines Interframe Image Processing Settings**

Step	Register/Bit Field/Programming Model	Value
<b>IF: Is the resizer module set to free-running mode?</b>	RSZ_SRC_MODE[0] OST	= 0x0
Set either of the engines to free-running mode to keep popping data out of the internal buffer. Otherwise, it will lead to an overflow event after the first frame.	RZA_MODE[0] MODE RZB_MODE[0] MODE	0x0
<b>ELSE: Is the resizer module set to one-shot mode?</b>		
Set both RSZ engines to one-shot mode.	RZA_MODE[0] MODE RZB_MODE[0] MODE	0x1
<b>ENDIF</b>		
Set the output format. See <a href="#">Table 8-322</a> .	RZA_420[0] YEN RZA_420[1] CEN RZB_420[0] YEN RZB_420[1] CEN	0x-
On the side of the <a href="#">RSZ_SRC_VPS</a> , <a href="#">RSZ_SRC_HPS</a> , <a href="#">RSZ_SRC_VSZ</a> , and <a href="#">RSZ_SRC_HSZ</a> registers, set the engine complementary function to crop within the global frame and assign proper vertical start position of the input frame.	RZA_I_VPS[12:0] VPS RZB_I_VPS[12:0] VPS	0x-
Assign proper horizontal start position of the input frame.	RZA_I_HPS[12:0] HPS RZB_I_HPS[12:0] HPS	0x-
Assign proper vertical size of the output frame.	RZA_O_VSZ[12:0] VPS RZB_O_VSZ[12:0] VPS	0x-
Assign proper horizontal size of the output frame.	RZA_O_HSZ[12:1] HPS RZB_O_HSZ[12:1] HPS	0x-
Set the phase position for the Chroma and Luma element and configure the averager.	See <a href="#">Section 8.3.4.5.3.1</a>	
Set the vertical resize value for the two engines. (vertical resize ration = 256 / RZx_V_DIF).	RZA_V_DIF[13:0] V RZB_V_DIF[13:0] V	0x-
Set the horizontal resize value for the two engines. (horizontal resize ration = 256 / RZx_H_DIF).	RZA_H_DIF[13:0] H RZB_H_DIF[13:0] H	0x-
Select vertical method of resizing filtering (linear or cubic interpolation) for the Luma and Chroma elements.	RZA_V_TYP[0] Y RZB_V_TYP[0] Y RZA_V_TYP[1] C RZB_V_TYP[1] C	0x-
Select horizontal method of resizing filtering (linear or cubic interpolation) for the Luma and Chroma elements.	RZA_H_TYP[0] Y RZB_H_TYP[0] Y RZA_H_TYP[1] C RZB_H_TYP[1] C	0x-
<b>IF: Is the interpolation method linear?</b>	RZx_V_TYP[0] Y <b>AND</b> RZx_V_TYP[1] C RZx_H_TYP[0] Y <b>AND</b> RZx_H_TYP[1] C	= 0x1
Set the needed vertical LPF intensity for the Luma and Chroma elements.	RZA_V_LPF[5:0] Y RZA_V_LPF[11:6] C RZB_V_LPF[5:0] Y RZB_V_LPF[11:6] C	0x-
Set the needed horizontal LPF intensity for the Luma and Chroma elements.	RZA_H_LPF[5:0] Y RZA_H_LPF[11:6] C RZB_H_LPF[5:0] Y RZB_H_LPF[11:6] C	0x-
<b>ENDIF</b>		
Set the Chroma saturation MAX and MIN values.	RSZ_YUV_C_MAX[7:0] MAX RSZ_YUV_C_MIN[7:0] MIN	0x-
Set the Luma saturation MAX and MIN values.	RSZ_YUV_Y_MAX[7:0] MAX RSZ_YUV_Y_MIN[7:0] MIN	0x-

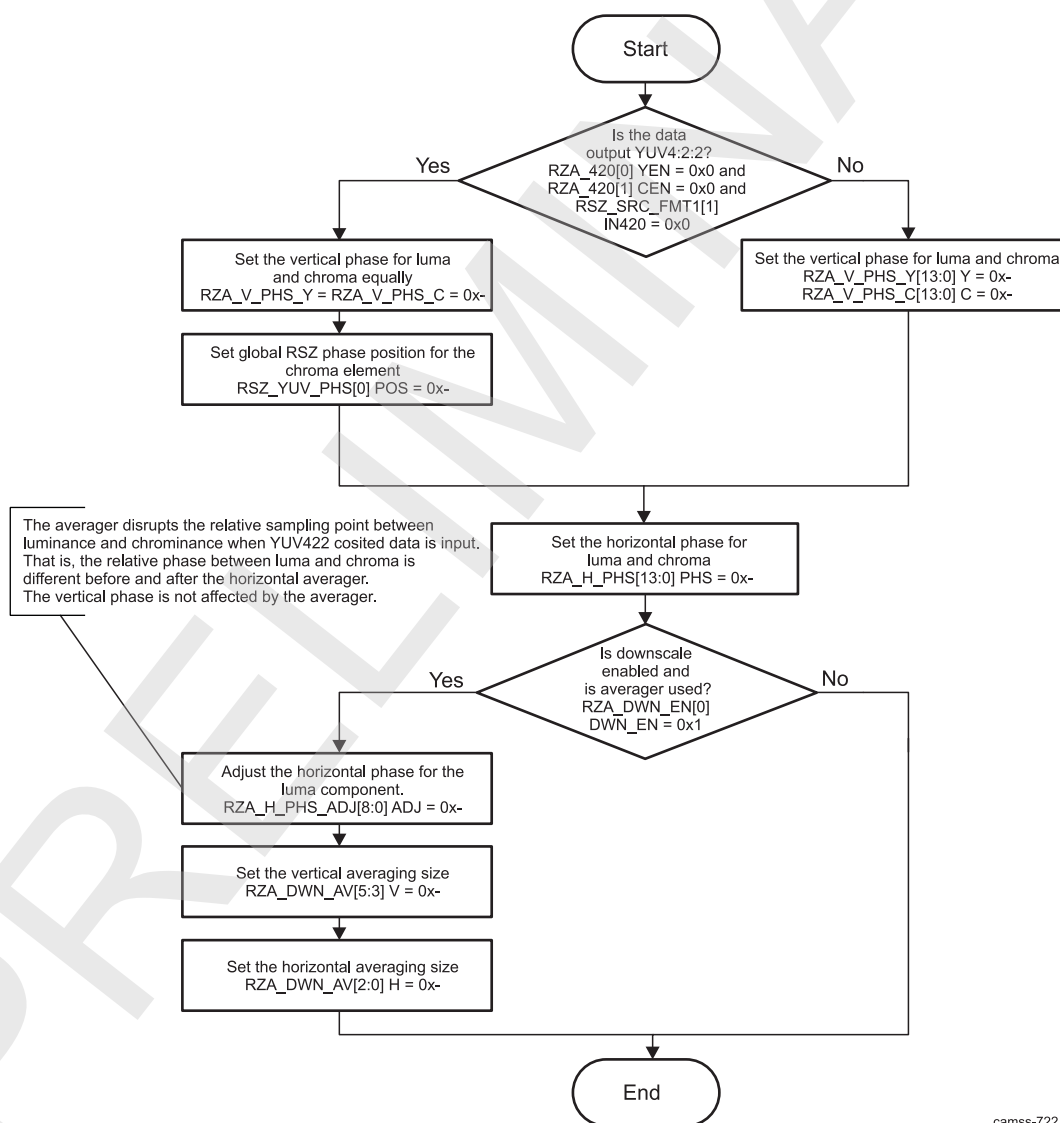
**Table 8-377. ISS ISP RSZ Engines Interframe Image Processing Settings (continued)**

Step	Register/Bit Field/Programming Model	Value
(optional) Set the mode to RGB conversion, configure the alpha value and set additional pixel masking.	See <a href="#">Section 8.3.4.5.3.2</a> .	

### 8.3.4.5.3.1 ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1

The procedure shown in [Figure 8-236](#) sets the phase position for the Chroma and Luma elements and configures the averager.

**NOTE:** This procedure configures RSZ engine A. The procedure for RSZ engine B is identical.

**Figure 8-236. ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1**

**NOTE:** When masking is used, boundaries affect the leftmost/rightmost 2 pixels in up-conversion.

**Table 8-378. Register Call Summary for ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1**

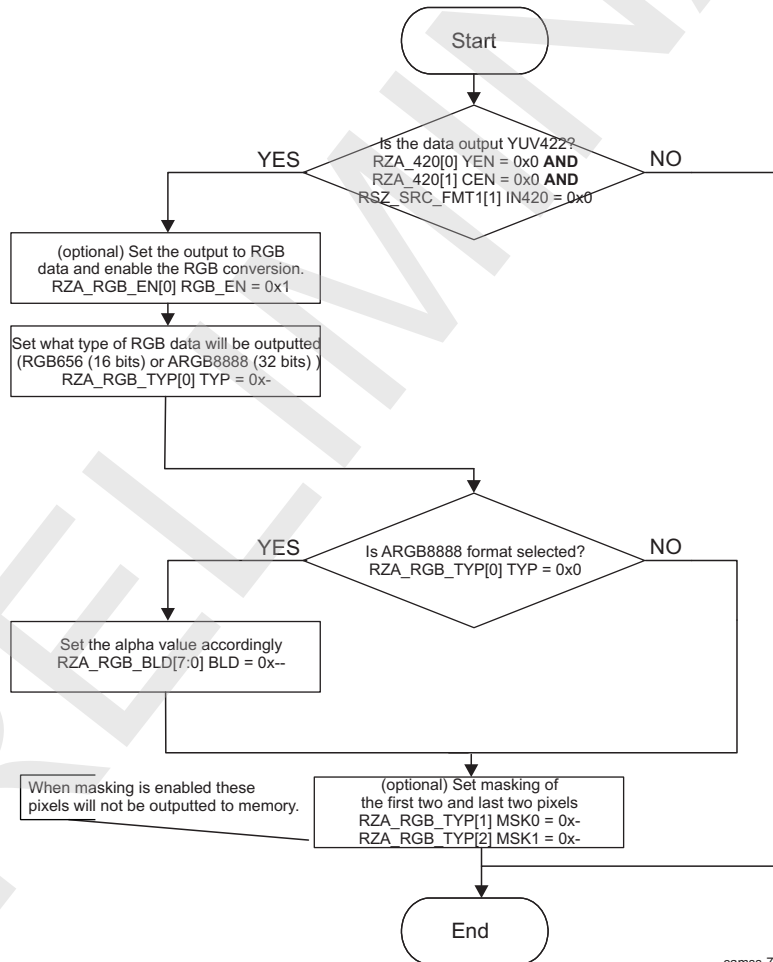
Register Name	Register Name	Register Name
RZA_420[0] YEN	RZA_420[1] CEN	RSZ_SRC_FMT1[1] IN420
RZA_V_PHS_Y	RZA_V_PHS_C	RSZ_YUV_PHS[0] POS
RZA_V_PHS_Y[13:0] Y	RZA_V_PHS_C[13:0] C	RZA_H_PHS [13:0] PHS
RZA_DWN_EN[0] DWN_EN	RZA_H_PHS_ADJ[8:0] ADJ	RZA_DWN_AV[5:3] V
RZA_DWN_AV[2:0] H		

**8.3.4.5.3.2 ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2**

This procedure shown in Figure 8-237 sets the mode to RGB conversion, configures the alpha value, and sets additional pixel masking.

**NOTE:** This procedure configures RSZ engine A. The procedure for RSZ engine B is identical.

**Figure 8-237. ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2**



camss-723

**Table 8-379. Register Call Summary for ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2**

Register Name	Register Name	Register Name
RZA_420[0] YEN	RZA_420[1] CEN	RSZ_SRC_FMT1[1] IN420
RZA_RGB_EN[0] RGB_EN	RZA_RGB_TYP[0] TYP	RZA_RGB_BLD[7:0] BLD

**Table 8-379. Register Call Summary for ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2 (continued)**

Register Name	Register Name	Register Name
<a href="#">RZA_RGB_TYP[1]</a> MSK0	<a href="#">RZA_RGB_TYP[2]</a> MSK1	

#### 8.3.4.5.4 ISS ISP RSZ Programming Constraints

The RSZ module contains shadowed and nonshadowed registers. Shadowed registers can be updated at anytime during the resizing operation, but the new setting does not take effect until the next `rsz_int_reg` event. Shadowed registers can be updated for the next frame after the `rsz_int_reg` event triggers.

Nonshadowed registers must be programmed before enabling the RSZ module or between frames (that is, after the `rsz_int_dma` has come [and the entire frame has come on the VPORT IF] and before the VD of the next frame.) [Table 8-380](#) list the nonshadowed registers.

**Table 8-380. ISS ISP RSZ Nonshadowed Registers**

Nonshadowed Registers
<a href="#">RSZ_SYSCONFIG</a>
<a href="#">RSZ_IN_FIFO_CTRL</a>
<a href="#">RSZ_GCK_MMR</a>
<a href="#">RSZ_GCK_SDR</a>
<a href="#">RSZ_SRC_MODE</a>
<a href="#">RSZ_SRC_FMT0</a>
<a href="#">RSZ_SRC_VPS</a>
<a href="#">RSZ_SRC_HPS</a>
<a href="#">RSZ_SRC_EN</a>

#### 8.3.4.6 ISS ISP H3A Programming Model

This section discusses issues related to the software control of the H3A module. It lists which registers are required to be programmed in different modes, how to enable and disable the H3A, how to check the status of the H3A, discusses the different register access types, and enumerates several programming constraints.

##### 8.3.4.6.1 ISS ISP H3A Hardware Setup/Initialization

This section discusses the configuration of the H3A required before image processing can begin.

###### 8.3.4.6.1.1 ISS ISP H3A Reset Behavior

Upon hardware reset of the ISP, all of the registers in the H3A are reset to their reset values.

###### 8.3.4.6.1.2 ISS ISP H3A Register Setup

For register configuration purposes, the AF engine and the AEW engine of the H3A can be configured independently. Because there are separate enable bits for each engine, this section is divided into the AF engine and the AEW engine.

###### 8.3.4.6.1.2.1 ISS ISP H3A AF Engine

Before enabling the AF engine, the hardware must be properly configured through register writes.

[Table 8-381](#) lists the register parameters that must be programmed before enabling the AF engine of the H3A.

**Table 8-381. ISS ISP H3A AF Engine Required Configuration Parameters**

Step	Configuration Required	Value
<b>AF optional preprocessing settings</b>		
Note: A suggestion is to use the averaging filter in the IPIPEIF to reduce noise before generating AF statistics.		
Enable or disable the median filter.	H3A_PCR[2] AF_MED_EN	
Set the median filter threshold, if the filter is enabled.	H3A_PCR[10:3] MED_TH	
Enable or disable A-Law compression.	H3A_PCR[1] AF_ALAW_EN	
<b>Set AF general settings.</b>		
Set the focus value accumulation mode.	H3A_PCR[14] FVMODE	
Set the input start information.	H3A_LINE_START[31:16] SLV H3A_LINE_START[15:0] LINE_START	
Set the output SDRAM destination start address.	H3A_AFBUFST[31:5] AFBUFST	
<b>RGB pixel extraction and paxel settings</b>		
Set the RGB positions in the pixel.	H3A_PCR[13:11] RGBPOS	
Set the paxel width.	H3A_AFPAX1[23:16] PAXW	
Set the paxel height.	H3A_AFPAX1[7:0] PAXH	
Set the paxel horizontal start position.	H3A_AFPAXSTART[27:16] PAXSH	
Set the paxel vertical start position.	H3A_AFPAXSTART[11:0] PAXSV	
Set the column increment.	H3A_AFPAX2[20:17] AFINCH	
Set the line increment.	H3A_AFPAX2[16:13] AFINCV	
Set the vertical paxel count.	H3A_AFPAX2[12:6] PAXVC	
Set the horizontal paxel count.	H3A_AFPAX2[5:0] PAXHC	
<b>Horizontal focus value calculator settings</b>		
Set the horizontal threshold for the two IIR.	H3A_HVF_THR[31:16] HTHR2 H3A_HVF_THR[15:0] HTHR1	
Set the IIR horizontal start position.	H3A_AFIIRSH[11:0] IIRSH	
Set the coefficients for the SET 0 IIR.	H3A_AFcoef010 H3A_AFcoef032 H3A_AFcoef054 H3A_AFcoef076 H3A_AFcoef098 H3A_AFcoef0010	
Set the coefficients for the SET 1 IIR.	H3A_AFcoef110 H3A_AFcoef132 H3A_AFcoef154 H3A_AFcoef176 H3A_AFcoef198 H3A_AFcoef1010	
<b>Vertical focus value calculator settings</b>		
Enable or disable vertical AF focus value calculation.	H3A_PCR[20] AF_VF_EN	
Set the vertical FIR 1 coefficients and threshold.	H3A_VFV_CFG1 H3A_VFV_CFG2	
Set the vertical FIR 2 coefficients and threshold.	H3A_VFV_CFG3 H3A_VFV_CFG4	

The following references offer guidelines on how to program the filter coefficients and make use of the H3A output.

- V. Peddigari, M. Gamadia, and N. Kehtarnavaz, *Real-time implementation issues in passive automatic focusing for digital still cameras*, Journal of Imaging Science and Technology, vol. 49, no. 2, pp. 114-123, Mar/Apr 2005.
- M. Gamadia and N. Kehtarnavaz, *A real-time continuous automatic focus algorithm for digital cameras*, in Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation 2006, pp. 163 - 167, Mar. 2006.
- M. Gamadia, N. Kehtarnavaz, and K. Roberts-Hoffman, *Low-light auto-focus enhancement for digital and cell-phone camera image pipelines*, IEEE Transactions on Consumer Electronics, vol. 53, no. 2, pp. 249-257, May 2007.

#### 8.3.4.6.1.2.2 ISS ISP H3A AEW Engine

Before enabling the AEW engine, the hardware must be properly configured through register writes. [Table 8-382](#) lists the register parameters that must be programmed before enabling the AEW engine of the H3A.

**Table 8-382. ISS ISP H3A AEW Engine Required Configuration Parameters**

Step	Configuration Required	Value
AEW optional preprocessing settings		
Enable or disable median filter.	<a href="#">H3A_PCR[19]</a> AEW_MED_EN	
Set the median filter threshold.	<a href="#">H3A_PCR[10:3]</a> MED_TH	
Enable or disable A-Law compression.	<a href="#">H3A_PCR[17]</a> AEW_ALAW_EN	
Set AEW general settings.		
Set the saturation limit.	<a href="#">H3A_PCR[31:22]</a> AVE2LMT	
Set the AE/AWB output format.	<a href="#">H3A_AEWCFG[9:8]</a> AEFMT	
Set the AE/AWB shift value for sum of pixels.	<a href="#">H3A_AEWCFG[3:0]</a> SUMFST	
Set the input start information.	<a href="#">H3A_LINE_START[31:16]</a> SLV <a href="#">H3A_LINE_START[15:0]</a> LINE_START	
Set the output SDRAM destination start address.	<a href="#">H3A_AEWBUFST[31:5]</a> AEWBUFST	
Set the AE/AWB window configuration settings.		
Set the window width (in pixels).	<a href="#">H3A_AEWWIN1[20:13]</a> WINW	
Set the window height (in lines).	<a href="#">H3A_AEWWIN1[31:24]</a> WINH	
Set the window count for horizontal direction.	<a href="#">H3A_AEWWIN1[5:0]</a> WINHC	
Set the window count for vertical direction.	<a href="#">H3A_AEWWIN1[12:6]</a> WINVC	
Set the window start position H.	<a href="#">H3A_AEWINSTART[11:0]</a> WINSH	
Set the window start position V.	<a href="#">H3A_AEWINSTART[27:16]</a> WINSV	
Set the horizontal distance between subsamples.	<a href="#">H3A_AEWSUBWIN[3:0]</a> AEWINCH	
Set the vertical distance between subsamples.	<a href="#">H3A_AEWSUBWIN[11:8]</a> AEWINCV	
Set the vertical start position for single black line of windows.	<a href="#">H3A_AEWINBLK[27:16]</a> WINSV	
Set the height for the single black line of windows.	<a href="#">H3A_AEWINBLK[6:0]</a> WINH	

#### 8.3.4.6.2 ISS ISP H3A Enable/Disable Hardware

Setting the [H3A\\_PCR\[0\]](#) AF\_EN bit enables the AF engine, and the [H3A\\_PCR\[16\]](#) AEW\_EN bit enables the AEW engine. This must be done after all of the required registers discussed in the previous section are programmed.

The H3A operates in continuous mode. Processing of the frame is dependent upon the timing of the IPIPEIF. To ensure that data from the IPIPEIF is not missed, the H3A must be enabled before the IPIPEIF. In this way, the H3A waits for data from the IPIPEIF. The AF engine or the AEW engine can be disabled by clearing the [H3A\\_PCR\[0\] AF\\_EN](#) or [H3A\\_PCR\[16\] AEW\\_EN](#) bit, respectively, during the processing of the last frame. The disable is latched in at the end of the frame in which it was written.

### 8.3.4.6.3 ISS ISP H3A Register Accessibility During Frame Processing

There are two types of register access in the H3A module:

- Shadow registers

These registers/fields can be read and written (if the field is writable) at any time. However, the written values take effect only at the start of a frame. Reads still return the most recent write even though the settings are not used until the next start of frame.

The only shadowed registers in the H3A module are:

- [H3A\\_AFPAX1](#)
- [H3A\\_AFPAX2](#)
- [H3A\\_AFPAXSTART](#)
- [H3A\\_AFIIRSH](#)
- [H3A\\_AEWWIN1](#)
- [H3A\\_AEWINSTART](#)
- [H3A\\_AEWINBLK](#)
- [H3A\\_AEWSUBWIN](#)
- [H3A\\_AEWCFG](#)
- [H3A\\_AEWBUFST](#)

- Busy-lock registers

All other registers, except those described previously, belong to this category.

Busy-lock registers cannot be written when the module is busy ([H3A\\_PCR\[15\] BUSYAF == 1](#) OR [H3A\\_PCR\[18\] BUSYAEAWB == 1](#)). Writes are allowed to occur, but no change occurs in the registers (blocked writes from the hardware perspective, but allowed write from the software perspective). Once the busy bit in the [H3A\\_PCR](#) register ([H3A\\_PCR\[15\] BUSYAF](#) or [H3A\\_PCR\[18\] BUSYAEAWB](#) bit) is reset to 0, the busy-lock registers can be written.

The ideal procedure for changing the H3A registers if ([H3A\\_PCR\[15\] BUSYAF == 0](#) or [H3A\\_PCR\[18\] BUSYAEAWB == 0](#)) or if (EOF interrupt occurs) is:

1. Disable AF or AE/AWB.
2. Change registers.
3. Enable AF or AE/AWB.

### 8.3.4.6.4 ISS ISP H3A Interframe Operations

Between frames, it may be necessary to modify the memory pointers before processing the next frame. Because the [H3A\\_PCR](#) register and memory pointer registers are shadowed, these modifications can take place any time before the end of the frame, and the data will be latched in for the next frame. The host controller can perform these changes upon receiving an interrupt.

### 8.3.4.6.5 ISS ISP H3A Summary of Constraints

The following is a list of register configuration constraints to adhere to when programming the H3A. It can be used as a quick checklist. More detailed register setting constraints can be found in the individual register descriptions.

- The output addresses ([H3A\\_AFBUFST\[31:5\] AFBUFST](#) and ) must be on 64-byte boundaries
- Each horizontal row of paxels ( ) or windows ( ) starts on a 32-byte boundary.
- If the pixel clock frequency < ISP\_FCLK/2 and vertical focus is enabled, the constraints listed in [Table 8-383](#) apply.

**Table 8-383. ISS ISP H3A Constraints When PCLK < ISP\_FCLK/2 and Vertical Focus Enabled**

Field	Constraint
H3A_AEWWIN1[20:13] WINW	>7
H3A_AFPAX1[23:16] PAXW	>7
H3A_AFIIRSH[11:0] IIRSH	None
H3A_AFPAXSTART[27:16] PAXSH	$\geq$ H3A_AFIIRSH[11:0] IIRSH + 2
H3A_AFPAX2[20:17] AFINCH	None
H3A_AFPAX2[5:0] PAXHC	$\leq$ 11

- the If pixel clock frequency < ISP\_FCLK/2 and vertical focus is disabled, the constraints listed in [Table 8-384](#) apply.

**Table 8-384. ISS ISP H3A Constraints When PCLK < ISP\_FCLK/2 and Vertical Focus Disabled**

Field	Constraint
H3A_AEWWIN1[20:13] WINW	>7
H3A_AFPAX1[23:16] PAXW	>7
H3A_AFIIRSH[11:0] IIRSH	Must be even
H3A_AFPAXSTART[27:16] PAXSH	$\geq$ H3A_AFIIRSH[11:0] IIRSH + 2 and must be even
H3A_AFPAX2[20:17] AFINCH	$(1 + \text{H3A\_AFPAX1}[23:16] \text{ PAXW} / \text{H3A\_AFPAX2}[20:17] \text{ AFINCH}) * \text{H3A\_AFPAX2}[5:0] \text{ PAXHC}$ must be between 4 and 384
H3A_AFPAX2[5:0] PAXHC	$\leq$ 35

- If the pixel clock frequency  $\geq$  ISP\_FCLK/2 and vertical focus is enabled, the constraints listed in [Table 8-385](#) apply.

**Table 8-385. ISS ISP H3A Constraints When PCLK  $\geq$  ISP\_FCLK/2 and Vertical Focus Enabled**

Field	Constraint
H3A_AEWWIN1[20:13] WINW	>15
H3A_AFPAX1[23:16] PAXW	>15
H3A_AFIIRSH[11:0] IIRSH	None
H3A_AFPAXSTART[27:16] PAXSH	$\geq$ H3A_AFIIRSH[11:0] IIRSH + 2
H3A_AFPAX2[20:17] AFINCH	None
H3A_AFPAX2[5:0] PAXHC	$\leq$ 11

- If the pixel clock frequency  $\geq$  ISP\_FCLK/2 and vertical focus is disabled, the constraints in [Table 8-386](#) apply.

**Table 8-386. ISS ISP H3A Constraints When PCLK  $\geq$  ISP\_FCLK/2 and Vertical Focus Disabled**

Field	Constraint
H3A_AEWWIN1[20:13] WINW	>15
H3A_AFPAX1[23:16] PAXW	>15
H3A_AFIIRSH[11:0] IIRSH	Must be even
H3A_AFPAXSTART[27:16] PAXSH	$\geq$ H3A_AFIIRSH[11:0] IIRSH + 2 and must be even
H3A_AFPAX2[20:17] AFINCH	$(1 + \text{H3A\_AFPAX1}[23:16] \text{ PAXW} / \text{H3A\_AFPAX2}[20:17] \text{ AFINCH}) * \text{H3A\_AFPAX2}[5:0] \text{ PAXHC}$ must be between 4 and 384 and $\text{H3A\_AFPAX2}[20:17] \text{ AFINCH} \text{ modulo } \text{H3A\_AFPAX1}[23:16] \text{ PAXW} \neq 1$
H3A_AFPAX2[5:0] PAXHC	$\leq$ 35

**AF engine**

- The paxel horizontal start value must be greater than or equal to the IIR horizontal start position.
- The paxel start/end and IIR filter start positions must not be set within the first 2 and the last 2 pixels (to check).
- The width (H3A\_AFPAX1[23:16] PAXW) and height (H3A\_AFPAX1[7:0] PAXH) of the paxels must be



an even number.

- The minimum width of the paxel ([H3A\\_AFPAX1\[23:16\] PAXW](#)) must be 8 pixels.
- The number of columns to increment in a paxel ([H3A\\_AFPAX2\[20:17\] AFINCH](#)) must be even and is restricted to 2 to 32.
- The number of lines to increment in a paxel ([H3A\\_AFPAX2\[16:13\] AFINCV](#)) must be even and is restricted to 0 to 30.
- The maximum number of vertical paxels in a frame ([H3A\\_AFPAX2\[12:6\] PAXVC](#)) must not exceed 128.
- The number of paxels in the horizontal direction ([H3A\\_AFPAX2\[5:0\] PAXHC](#)) has a valid range from 1 to 35.
- If vertical mode is enabled:
  - Paxel horizontal start position ([H3A\\_AFPAXSTART\[27:16\] PAXSH](#)) must be even.
  - The lower bit of the [H3A\\_AFPAXSTART\[27:16\] PAXSH](#) bit field and the lower bit of the [H3A\\_AFIIRSH\[11:0\] IIRSH](#) bit field must be equal.
  - The [H3A\\_AFPAXSTART\[11:0\] PAXSV](#) bit field must be  $\geq 8$ .
- If vertical mode is not enabled, the [H3A\\_AFIIRSH\[11:0\] IIRSH](#) bit field must be even.
- Paxels cannot overlap the last pixel in a line.
- Paxels must be adjacent to one another.

**AEW engine**

- The width ([H3A\\_AEWWIN1\[20:13\] WINW](#)) and height ([H3A\\_AEWWIN1\[31:24\] WINH](#)) of the windows must be an even number.
- The minimum width of the window ([H3A\\_AEWWIN1\[20:13\] WINW](#)) must be 8 pixels.
- The window height ([H3A\\_AEWWIN1\[31:24\] WINH](#)) has a valid range from 2 to 512.
- The maximum number of vertical windows in a frame ([H3A\\_AEWWIN1\[12:6\] WINVC](#)) must not exceed 128.
- The number of horizontal windows ([H3A\\_AEWWIN1\[5:0\] WINHC](#)) has a valid range from 1 to 35.
- Vertical and horizontal window start position ([H3A\\_AEWWINSTART](#)) has a valid range from 0 to 4095.
- Vertical window start position for single black lines ([H3A\\_AEWINBLK\[27:16\] WINSV](#)) has a valid range from 0 to 4095.
- Horizontal window start position for single black lines ([H3A\\_AEWINBLK\[6:0\] WINH](#)) must be even and has a valid range from 2 to 256
- Subsampling windows can start only on even numbers.
- Vertical and horizontal sampling point increment ([H3A\\_AEWSUBWIN](#)) has a valid range from 2 to 32.

**8.3.4.7 ISS ISP BL Programming Model**

The procedure listed in [Table 8-387](#) initializes the buffer logic.

**Table 8-387. ISS ISP BL Settings**

Step	Register/Bit Field/Programming Model	Value
Sets the memory access priority registers	<a href="#">ISP5_MPSR</a>	
Sets the minimum interval between two memory requests for ISIF read port	<a href="#">ISP5_BL_MTC_1[31:16] ISIF_R</a>	
Sets the minimum interval between two memory requests for IPIPEIF read port	<a href="#">ISP5_BL_MTC_1[15:0] IPIPEIF_R</a>	
Sets the minimum interval between two memory requests for H3A write port	<a href="#">ISP5_BL_MTC_1[31:16] H3A_W</a>	
Set the maximum number of CIDs/tags that the BL can use.	<a href="#">ISP5_CTRL[7:4] VBUSM_CIDS</a>	
Set the BL VBUSM priority setting.	<a href="#">ISP5_CTRL[3:1] VBUSM_CPRIORITY</a>	
Set write posted/nonposted.	<a href="#">ISP5_CTRL[0] OCP_WRNP</a>	
Enable the BL clock.	<a href="#">ISP5_CTRL[15] BL_CLK_ENABLE</a>	0x1

### 8.3.5 ISS ISP Register Manual

#### 8.3.5.1 ISS ISP Instance Summary

**Table 8-388. ISS ISP Instance Summary**

Module Name	Module Base Address	Size
ISS_ISP5_SYS1	0x5201 0000	160 bytes
ISS_ISP5_SYS2	0x5201 00A0	864 bytes
ISS_RESIZER	0x5201 0400	1KB
ISS_IPIPE	0x5201 0800	2KB
ISS_ISIF	0x5201 1000	512 bytes
ISS_IPIPEIF	0x5201 1200	128 bytes
ISS_H3A	0x5201 1400	512 bytes

#### 8.3.5.2 ISS ISP5 SYS1 Registers

##### 8.3.5.2.1 ISS ISP5 SYS1 Register Summary

**Table 8-389. ISS ISP5 SYS1 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISP5_SYS1 Base Address
ISP5_REVISION	R	32	0x0000 0000	0x5201 0000
ISP5_HWINFO1	R	32	0x0000 0004	0x5201 0004
ISP5_HWINFO2	R	32	0x0000 0008	0x5201 0008
ISP5_SYSCONFIG	RW	32	0x0000 0010	0x5201 0010
RESERVED	RW	32	0x0000 0020	0x5201 0020
ISP5_IRQSTATUS_RA W <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0024 + (0x10 * i)	0x5201 0024 + (0x10 * i)
ISP5_IRQSTATUS <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0028 + (0x10 * i)	0x5201 0028 + (0x10 * i)
ISP5_IRQENABLE_SET <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 002C + (0x10 * i)	0x5201 002C + (0x10 * i)
ISP5_IRQENABLE_CLR <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0030 + (0x10 * i)	0x5201 0030 + (0x10 * i)
ISP5_DMAENABLE_SE T	RW	32	0x0000 0064	0x5201 0064
ISP5_DMAENABLE_CL R	RW	32	0x0000 0068	0x5201 0068
ISP5_CTRL	RW	32	0x0000 006C	0x5201 006C
ISP5_PG	RW	32	0x0000 0070	0x5201 0070
ISP5_PG_PULSE_CTR L	RW	32	0x0000 0074	0x5201 0074
ISP5_PG_FRAME_SIZE	RW	32	0x0000 0078	0x5201 0078
ISP5_MPSR	RW	32	0x0000 007C	0x5201 007C
ISP5_BL_MTC_1	RW	32	0x0000 0080	0x5201 0080
ISP5_BL_MTC_2	RW	32	0x0000 0084	0x5201 0084
ISP5_BL_VBUSM	RW	32	0x0000 0088	0x5201 0088

<sup>(1)</sup> i = 0 to 3

##### 8.3.5.2.2 ISS ISP5 SYS1 Register Description

**Table 8-390. ISP5\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 8-389</a>	<b>Instance</b>	ISS_ISP5_SYS1
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 8-391. Register Call Summary for Register ISP5\_REVISION**

ISS ISP

- [ISS ISP5 SYS1 Register Summary: \[0\]](#)

**Table 8-392. ISP5\_HWINFO1**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 8-389</a>	<b>Instance</b>	ISS_ISP5_SYS1
<b>Description</b>	GENERIC PARAMETER REGISTER Information about the hardware configuration of the IP module.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		ISIF_RFM_LINE_SIZE										RESERVED		IPIPE_LINE_SIZE																	

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	ISIF_RFM_LINE_SIZE	Memory line size for the data reformatter in the ISIF module.	R	0x1500
15:13	RESERVED		R	0x0
12:0	IPIPE_LINE_SIZE	Memory line size for the IPIPE module	R	0x1500

**Table 8-393. Register Call Summary for Register ISP5\_HWINFO1**

ISS ISP

- [ISS ISP5 SYS1 Register Summary: \[0\]](#)

**Table 8-394. ISP5\_HWINFO2**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 8-389</a>	<b>Instance</b>	ISS_ISP5_SYS1
<b>Description</b>	GENERIC PARAMETER REGISTER Information about the hardware configuration of the IP module.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																H3A_LINE_SIZE															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	H3A_LINE_SIZE	Memory line size for the H3A module	R	0x0BC0

**Table 8-395. Register Call Summary for Register ISP5\_HWINFO2**

ISS ISP

- [ISS ISP5\\_SYS1 Register Summary: \[0\]](#)

**Table 8-396. ISP5\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 8-389</a>	<b>Instance</b>	ISS_ISP5_SYS1
<b>Description</b>	Clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STANDBYMODE	RESERVED	SOFTRESET	AUTO_IDLE												

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5:4	STANDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. 0x0: Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only. 0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only. 0x3: Reserved 0x2: Smart-standby mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator. IP module must not generate (initiator-related) wake-up events. Generation of the MStandby signal must be initiated by the firmware by writing <code>ISP5_CTRL.MSTANDBY = 1</code> .	RW	0x2
3:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	SOFTRESET	<p>Software reset.</p> <p>The soft reset will cause the MStandby to be asserted as the reset value of the <code>ISP5_CTRL.MSTANDBY</code> bit is 1. After a soft reset, the software must ensure not to perform any access for 16 clock cycles (OCP-slave port frequency) after writing this bit. The OCP slave port is running at half the frequency of the functional clock. Before issuing a soft reset, the software must ensure that no more traffic is being generated by the ISP. Basically, it means that the camera module must be stopped from sending data and/or that the ISP modules are disabled. The last interrupt triggered by the ISP design upon completion of the frame processing is <code>rsz_int_dma</code>. This <code>rsz_int_dma</code> event must be used to enable clean termination of the processing. The software must wait a few hundred cycles to trigger the soft reset after upon assertion of the <code>rsz_int_dma</code>, this is to ensure that the BL is completely drained.</p> <p>Software must set the ISP in standby mode before issuing the soft reset:</p> <p>Set <code>ISP5_SYSCONFIG.STANDBYMODE = 2</code> (smart standby).</p> <p>Set <code>ISP5_CTRL.MSTANDBY</code> to 1.</p> <p>Poll for <code>ISP5_CTRL.MSTANDBY_WAIT = 1</code>.</p> <p>Then, the soft reset can be applied (<code>ISP5_SYSCONFIG.SOFTRESET = 1</code>).</p> <p>Write 0x0: No action</p> <p>Write 0x1: Initiate software reset</p> <p>Read 0x1: Reset (software or other) ongoing</p> <p>Read 0x0: Reset done, no pending action</p>	RW	0
0	AUTO_IDLE	Auto clock gating. Always enabled.	R	1

**Table 8-397. Register Call Summary for Register ISP5\_SYSCONFIG**

ISS ISP

- [ISS ISP Reset: \[0\] \[1\] \[2\]](#)
- [ISS ISP IPIPEIF Reset Behavior: \[3\]](#)
- [ISS ISP RSZ Reset Behavior: \[4\] \[5\] \[6\]](#)
- [ISS ISP5 SYS1 Register Summary: \[7\]](#)
- [ISS ISP5 SYS1 Register Description: \[8\] \[9\]](#)

**Table 8-398. ISP5\_IRQSTATUS\_RAW\_i**

<b>Address Offset</b>	0x0000 0024 + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-389</a>	<b>Instance</b>	ISS_ISP5_SYS1
<b>Description</b>	<p>Per-event raw interrupt status vector.</p> <p>Raw status is set even if event is not enabled.</p> <p>Write 1 to set the (raw) status, mostly for debug.</p> <p>The ISP outputs four interrupt lines, <code>ISP_IRQ0</code> to <code>ISP_IRQ3</code>. Any internal ISP event can be merged on the four lines. A same event must be enabled on only one interrupt line.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPERR_IRQ	RESERVED	IPIPE_INT_DPC_RNEW1	IPIPE_INT_DPC_RNEW0	IPIPE_INT_DPC_INI	RESERVED	IPIPE_INT_EOF	H3A_INT_EOF	RSZ_INT_EOF1	RSZ_INT_EOF0	RESERVED	RSZ_FIFO_IN_BLK_ERR	RSZ_FIFO_OVF	RSZ_INT_CYC_RZB	RSZ_INT_CYC_RZA	RSZ_INT_DMA	RSZ_INT_LAST_PIX	RSZ_INT_REG	H3A_INT	AF_INT	AEW_INT	IPIPEIF_IRQ	IPIPE_INT_HST	IPIPE_INT_BSC	IPIPE_INT_DMA	IPIPE_INT_LAST_PIX	IPIPE_INT_REG	ISIF_INT_3	ISIF_INT_2	ISIF_INT_1	ISIF_INT_0	

Bits	Field Name	Description	Type	Reset
31	OCP_ERR_IRQ	An OCP error has been received on the ISP master port. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
30	RESERVED		R	0
29	IPIPE_INT_DPC_RNEW1	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
28	IPIPE_INT_DPC_RNEW0	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
27	IPIPE_INT_DPC_INI	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
26	RESERVED		R	0
25	IPIPE_INT_EOF	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
24	H3A_INT_EOF	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
23	RSZ_INT_EOF1	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer B engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
22	RSZ_INT_EOF0	RESIZER module event: This event signals that the BL has received the EOF signal from the resizer A engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory. Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
21:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	RSZ_FIFO_IN_BLK_ERR	<p>This event signals that the minimum vertical blanking period has not been respected causing errors in the input data buffering submodule.</p> <p>This event will be triggered when the <code>rsz_int_reg</code> event of frame N is triggered before the <code>rsz_int_dma</code> of frame N + 1.</p> <p>This event would typically happen at the transition between two frames because there is not enough vertical blanking between frames: the firmware must take care to ensure enough vertical blanking.</p> <p>The hardware cannot recover from this error. It will be required to perform a reset of the IP.</p> <p>Write 0x0: No action</p> <p>Write 0x1: Set event (debug)</p> <p>Read 0x1: Event pending</p> <p>Read 0x0: No event pending</p>	RW W1toSet	0
18	RSZ_FIFO_OVF	<p>This event signals that overflow happened in the input data buffering submodule.</p> <p>This event would typically happen while processing a frame because the video port pixel clock is too high: the firmware must take care to use a lower pixel clock at the input of the resizer module. Depending on the mode being used, the overflow can happen at different places:</p> <ol style="list-style-type: none"> <li>1. Bypass mode: overflow happened in the input circular buffer.</li> <li>2. Pass through mode: overflow happened on the module output interface (MTC)</li> <li>3. Normal resize mode: overflow happened in the input circular buffer.</li> </ol> <p>The hardware cannot recover from this error. It will be required to perform a reset of the IP.</p> <p>Write 0x0: No action</p> <p>Write 0x1: Set event (debug)</p> <p>Read 0x1: Event pending</p> <p>Read 0x0: No event pending</p>	RW W1toSet	0
17	RSZ_INT_CYC_RZB	<p>RESIZER module event:</p> <p>This event is the circular interrupt for RESIZER #B. An event can be triggered every time that <code>RSZ_IRQ_RZB</code> output lines have been written out to the <code>RZB_SDR_Y</code> buffer. The range can go from 1 to 8192 lines. Usually, this value should be such that the circular buffer vertical size (set by the <code>RZB_SDR_Y_PTR_E</code> register) is a multiple of <code>RSZ_IRQ_RZB</code>.</p> <p>Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first.</p> <p>Write 0x0: No action</p> <p>Write 0x1: Set event (debug)</p> <p>Read 0x1: Event pending</p> <p>Read 0x0: No event pending</p>	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
16	RSZ_INT_CYC_RZA	<p>RESIZER module event: This event is the circular interrupt for RESIZER #A. An event can be triggered every time that <a href="#">RSZ_IRQ_RZA</a> output lines have been written out to the RZA_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, the circular buffer vertical size (set by the <a href="#">RZA_SDR_Y_PTR_E</a> register) should be a multiple of <a href="#">RSZ_IRQ_RZA</a>.</p> <p>Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first.</p> <p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
15	RSZ_INT_DMA	<p>This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and that the resizer core has returned to idle. <code>rsz_int_dma</code> is a true indication that all processing is finished for the particular frame on both resizer engines.</p> <p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
14	RSZ_INT_LAST_PIX	<p>This event is triggered when the last pixel of the valid area is received.</p> <p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
13	RSZ_INT_REG	<p>This event is triggered when the first pixel of the valid area is received. Shadowed registers can be updated at any time but the new value will take effect on the next <code>rsz_int_reg</code> event.</p> <p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
12	H3A_INT	<p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
11	AF_INT	<p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
10	AEW_INT	<p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0



Bits	Field Name	Description	Type	Reset
9	IPIPEIF_IRQ	IPIPEIF module interrupt Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
8	IPIPE_INT_HST	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
7	IPIPE_INT_BSC	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
6	IPIPE_INT_DMA	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
5	IPIPE_INT_LAST_PIX	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
4	IPIPE_INT_REG	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
3	ISIF_INT_3	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
2	ISIF_INT_2	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
1	ISIF_INT_1	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0
0	ISIF_INT_0	Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending	RW W1toSet	0

**Table 8-399. Register Call Summary for Register ISP5\_IRQSTATUS\_RAW\_i**

ISS ISP

- [ISS ISP5 SYS1 Register Summary: \[0\]](#)

**Table 8-400. ISP5\_IRQSTATUS\_i**

<b>Address Offset</b>	0x0000 0028 + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-389</a>	<b>Instance</b>	ISS_ISP5_SYS1
<b>Description</b>	<p>Per-event "enabled" interrupt status vector.            Enabled status is not set unless event is enabled.            Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).            The ISP outputs four interrupt lines, ISP_IRQ0 to ISP_IRQ3. Any internal ISP event can be merged on the four lines. A same event must be enabled on only one interrupt line.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPERR_IRQ	RESERVED	IPIPE_INT_DPC_RNEW1	IPIPE_INT_DPC_RNEW0	IPIPE_INT_DPC_INI	RESERVED	IPIPE_INT_EOF	H3A_INT_EOF	RSZ_INT_EOF1	RSZ_INT_EOF0	RESERVED	RSZ_FIFO_IN_BLK_ERR	RSZ_FIFO_OVF	RSZ_INT_CYC_RZB	RSZ_INT_CYC_RZA	RSZ_INT_DMA	RSZ_INT_LAST_PIX	RSZ_INT_REG	H3A_INT	AF_INT	AEW_INT	IPIPEIF_IRQ	IPIPE_INT_HST	IPIPE_INT_BSC	IPIPE_INT_DMA	IPIPE_INT_LAST_PIX	IPIPE_INT_REG	ISIF_INT_3	ISIF_INT_2	ISIF_INT_1	ISIF_INT_0	

Bits	Field Name	Description	Type	Reset
31	OCPERR_IRQ	An OCP error has been received on the ISP master port. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
30	RESERVED		R	0
29	IPIPE_INT_DPC_RNEW1	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
28	IPIPE_INT_DPC_RNEW0	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
27	IPIPE_INT_DPC_INI	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
26	RESERVED		R	0
25	IPIPE_INT_EOF	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
24	H3A_INT_EOF	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
23	RSZ_INT_EOF1	<p>RESIZER module event: This event signals that the BL has received the EOF signal from the resizer B engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory.</p> <p>Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending</p>	RW W1toClr	0
22	RSZ_INT_EOF0	<p>RESIZER module event: This event signals that the BL has received the EOF signal from the resizer A engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory.</p> <p>Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending</p>	RW W1toClr	0
21:20	RESERVED		R	0x0
19	RSZ_FIFO_IN_BLK_ERR	<p>This event signals that the minimum vertical blanking period has not been respected causing errors in the input data buffering submodule. This event will be triggered when the rsz_int_reg event of frame N is triggered before the rsz_int_dma of frame N + 1. This event would typically happen at the transition between two frames because there is not enough vertical blanking between frames: the firmware must take care to ensure enough vertical blanking. The hardware cannot recover from this error. It will be required to perform a reset of the IP.</p> <p>Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending</p>	RW W1toClr	0
18	RSZ_FIFO_OVF	<p>This event signals that overflow happened in the input data buffering submodule or in the RSZ output interface. This event would typically happen while processing a frame because the video port pixel clock is too high: the firmware must take care to use a lower pixel clock at the input of the resizer module. The hardware cannot recover from this error. It will be required to perform a reset of the IP.</p> <p>Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending</p>	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
17	RSZ_INT_CYC_RZB	<p>RESIZER module event: This event is the circular interrupt for RESIZER #B. An event can be triggered every time that <a href="#">RSZ_IRQ_RZB</a> output lines have been written out to the RZB_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, this value should be such that the circular buffer vertical size (set by the <a href="#">RZB_SDR_Y_PTR_E</a> register) is a multiple of <a href="#">RSZ_IRQ_RZB</a>. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first.</p> <p>Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending</p>	RW W1toClr	0
16	RSZ_INT_CYC_RZA	<p>RESIZER module event: This event is the circular interrupt for RESIZER #A. An event can be triggered every time that <a href="#">RSZ_IRQ_RZA</a> output lines have been written out to the RZA_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, the circular buffer vertical size (set by the <a href="#">RZA_SDR_Y_PTR_E</a> register) should be a multiple of <a href="#">RSZ_IRQ_RZA</a>. Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first.</p> <p>Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending</p>	RW W1toClr	0
15	RSZ_INT_DMA	<p>This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and that the resizer core has returned to idle. <a href="#">rsz_int_dma</a> is a true indication that all processing is finished for the particular frame on both resizer engines.</p> <p>Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending</p>	RW W1toClr	0
14	RSZ_INT_LAST_PIX	<p>This event is triggered when the last pixel of the valid area is received.</p> <p>Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending</p>	RW W1toClr	0
13	RSZ_INT_REG	<p>This event is triggered when the first pixel of the valid area is received. Shadowed registers can be updated at any time but the new value will take effect on the next <a href="#">rsz_int_reg</a> event.</p> <p>Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending</p>	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
12	H3A_INT	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
11	AF_INT	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
10	AEW_INT	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
9	IPIPEIF_IRQ	IPIPEIF module interrupt Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
8	IPIPE_INT_HST	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
7	IPIPE_INT_BSC	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
6	IPIPE_INT_DMA	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
5	IPIPE_INT_LAST_PIX	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
4	IPIPE_INT_REG	Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
3	ISIF_INT_3	LSC interrupt issued by 2D-LSC block. Four types of 2D-LSC can be generated and mapped to the INT_3 line. For more information, see <a href="#">Section 8.3.3.7.10.1.5, ISS ISP ISIF 2D-LSC Events and Status Checking</a> . Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
2	ISIF_INT_2	<p>VD interrupt 2 event. Read this bit to check the interrupt mapped to the INT_2 line. This interrupt is also set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see <a href="#">Section 8.3.3.7.19.1</a>, <i>ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts</i>.</p> <p>Write 0x0: No action</p> <p>Write 0x1: Clear (raw) event</p> <p>Read 0x1: Event pending</p> <p>Read 0x0: No (enabled) event pending</p>	RW W1toClr	0
1	ISIF_INT_1	<p>VD interrupt 1 event. Read this bit to check the interrupt mapped to the INT_1 line. This interrupt is also set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see <a href="#">Section 8.3.3.7.19.1</a>, <i>ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts</i>.</p> <p>Write 0x0: No action</p> <p>Write 0x1: Clear (raw) event</p> <p>Read 0x1: Event pending</p> <p>Read 0x0: No (enabled) event pending</p>	RW W1toClr	0
0	ISIF_INT_0	<p>VD interrupt 0 event. Read this bit to check the interrupt mapped to the INT_0 line. This interrupt is also set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see <a href="#">Section 8.3.3.7.19.1</a>, <i>ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts</i>.</p> <p>Write 0x0: No action</p> <p>Write 0x1: Clear (raw) event</p> <p>Read 0x1: Event pending</p> <p>Read 0x0: No (enabled) event pending</p>	RW W1toClr	0

**Table 8-401. Register Call Summary for Register ISP5\_IRQSTATUS\_i**

## ISS ISP

- [ISS ISP RSZ Reset Behavior: \[0\] \[1\]](#)
- [ISS ISP5 SYS1 Register Summary: \[2\]](#)

**NOTE:** ISP submodule interrupts are mapped to ISP top-level lines. Moreover, ISP top-level lines are mapped to the ISS top interrupt merger level. For information about how IRQ lines are handled at the ISP level, see [Table 8-307](#). For information about how IRQs are handled at the ISS top-level before the signals leave ISS boundaries, see [Section 8.1.2.1.1](#), *ISS Interrupt Merger*.

**Table 8-402. ISP5\_IRQENABLE\_SET\_i**

<b>Address Offset</b>	0x0000 002C + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-389</a>	<b>Instance</b>	ISS_ISP5_SYS1
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register. The ISP outputs four interrupt lines, ISP_IRQ0 to ISP_IRQ3. Any internal ISP event can be merged on the four lines. A same event must be enabled on only one interrupt line.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPERR_IRQ	RESERVED	IPIPE_INT_DPC_RNEW1	IPIPE_INT_DPC_RNEW0	IPIPE_INT_DPC_INI	RESERVED	IPIPE_INT_EOF	H3A_INT_EOF	RSZ_INT_EOF1	RSZ_INT_EOF0	RESERVED	RSZ_FIFO_IN_BLK_ERR	RSZ_FIFO_OVF	RSZ_INT_CYC_RZB	RSZ_INT_CYC_RZA	RSZ_INT_DMA	RSZ_INT_LAST_PIX	RSZ_INT_REG	H3A_INT	AF_INT	AEW_INT	IPIPEIF_IRQ	IPIPE_INT_HST	IPIPE_INT_BSC	IPIPE_INT_DMA	IPIPE_INT_LAST_PIX	IPIPE_INT_REG	ISIF_INT_3	ISIF_INT_2	ISIF_INT_1	ISIF_INT_0	

Bits	Field Name	Description	Type	Reset
31	OCPERR_IRQ	An OCP error has been received on the ISP master port. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
30	RESERVED		R	0
29	IPIPE_INT_DPC_RNEW1	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
28	IPIPE_INT_DPC_RNEW0	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
27	IPIPE_INT_DPC_INI	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
26	RESERVED		R	0
25	IPIPE_INT_EOF	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
24	H3A_INT_EOF	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
23	RSZ_INT_EOF1	<p>RESIZER module event: This event signals that the BL has received the EOF signal from the resizer B engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
22	RSZ_INT_EOF0	<p>RESIZER module event: This event signals that the BL has received the EOF signal from the resizer A engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
21:20	RESERVED		R	0x0
19	RSZ_FIFO_IN_BLK_ERR	<p>This event signals that the minimum vertical blanking period has not been respected causing errors in the input data buffering submodule. This event will be triggered when the rsz_int_reg event of frame N is triggered before the rsz_int_dma of frame N + 1. This event would typically happen at the transition between two frames because there is not enough vertical blanking between frames: the firmware must take care to ensure enough vertical blanking. The hardware cannot recover from this error. It will be required to perform a reset of the IP.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
18	RSZ_FIFO_OVF	<p>This event signals that overflow happened in the input data buffering submodule or in the RSZ output interface. This event would typically happen while processing a frame because the video port pixel clock is too high: the firmware must take care to use a lower pixel clock at the input of the resizer module. The hardware cannot recover from this error. It will be required to perform a reset of the IP.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0



Bits	Field Name	Description	Type	Reset
17	RSZ_INT_CYC_RZB	<p>RESIZER module event: This event is the circular interrupt for RESIZER #B. An event can be triggered every time that <a href="#">RSZ_IRQ_RZB</a> output lines have been written out to the RZB_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, this value should be such that the circular buffer vertical size (set by the <a href="#">RZB_SDR_Y_PTR_E</a> register) is a multiple of <a href="#">RSZ_IRQ_RZB</a>.</p> <p>Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
16	RSZ_INT_CYC_RZA	<p>RESIZER module event: This event is the circular interrupt for RESIZER #A. An event can be triggered every time that <a href="#">RSZ_IRQ_RZA</a> output lines have been written out to the RZA_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, the circular buffer vertical size (set by the <a href="#">RZA_SDR_Y_PTR_E</a> register) should be a multiple of <a href="#">RSZ_IRQ_RZA</a>.</p> <p>Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
15	RSZ_INT_DMA	<p>This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and that the resizer core has returned to idle. <code>rsz_int_dma</code> is a true indication that all processing is finished for the particular frame on both resizer engines.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
14	RSZ_INT_LAST_PIX	<p>This event is triggered when the last pixel of the valid area is received.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
13	RSZ_INT_REG	<p>This event is triggered when the first pixel of the valid area is received. Shadowed registers can be updated at any time but the new value will take effect on the next <code>rsz_int_reg</code> event.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
12	H3A_INT	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
11	AF_INT	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
10	AEW_INT	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
9	IPIPEIF_IRQ	IPIPEIF module interrupt Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
8	IPIPE_INT_HST	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
7	IPIPE_INT_BSC	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
6	IPIPE_INT_DMA	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
5	IPIPE_INT_LAST_PIX	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
4	IPIPE_INT_REG	Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0
3	ISIF_INT_3	LSC interrupt issued by the 2D-LSC block. Four types of 2D-LSC can be generated and mapped to the INT_3 line. For more information, see <a href="#">Section 8.3.3.7.10.1.5, ISS ISP ISIF 2D-LSC Events and Status Checking</a> . Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
2	ISIF_INT_2	<p>VD interrupt 2 event. Set this bit to enable the interrupt and map it to the INT_2 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see <a href="#">Section 8.3.3.7.19.1, ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts</a>.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
1	ISIF_INT_1	<p>VD interrupt 1 event. Set this bit to enable the interrupt and map it to the INT_1 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see <a href="#">Section 8.3.3.7.19.1, ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts</a>.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0
0	ISIF_INT_0	<p>VD interrupt 0 event. Set this bit to enable the interrupt mapped to INT_0 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see <a href="#">Section 8.3.3.7.19.1, ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts</a>.</p> <p>Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toSet	0

**Table 8-403. Register Call Summary for Register ISP5\_IRQENABLE\_SET\_i**

ISS ISP

- ISS ISP Interrupt Tree: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24]
- ISS ISP IPIPEIF Interrupts: [25]
- ISS ISP IPIPE Interrupts: [26]
- ISS ISP RSZ Interrupts: [27]
- ISS ISP H3A Interrupts: [28]
- ISS ISP ISIF Interrupts: [29]
- ISS ISP IPIPEIF Module Events and Status Checking: [30]
- ISS ISP5 SYS1 Register Summary: [31]

**NOTE:** Setting the ISP submodule interrupts and mapping them to the ISP lines requires a configuration to receive the IRQ events at a higher ISS interrupt merger level. For information about how IRQ lines are handled at the ISP level, see [Table 8-307](#). For information about how IRQs are handled at the ISS top-level before the signals leave ISS boundaries, see [Section 8.1.2.1.1, ISS Interrupt Merger](#).

**Table 8-404. ISP5\_IRQENABLE\_CLR\_i**

<b>Address Offset</b>	0x0000 0030 + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-389</a>	<b>Instance</b>	ISS_ISP5_SYS1
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register. The ISP outputs four interrupt lines, ISP_IRQ0 to ISP_IRQ3. Any internal ISP event can be merged on the four lines. A same event must be enabled on only one interrupt line.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPE_ERR_IRQ	RESERVED	IPIPE_INT_DPC_RNEW1	IPIPE_INT_DPC_RNEW0	IPIPE_INT_DPC_INI	RESERVED	IPIPE_INT_EOF	H3A_INT_EOF	RSZ_INT_EOF1	RSZ_INT_EOF0	RESERVED	RSZ_FIFO_IN_BLK_ERR	RSZ_FIFO_OVF	RSZ_INT_CYC_RZB	RSZ_INT_CYC_RZA	RSZ_INT_DMA	RSZ_INT_LAST_PIX	RSZ_INT_REG	H3A_INT	AF_INT	AEW_INT	IPIPEIF_IRQ	IPIPE_INT_HST	IPIPE_INT_BSC	IPIPE_INT_DMA	IPIPE_INT_LAST_PIX	IPIPE_INT_REG	ISIF_INT_3	ISIF_INT_2	ISIF_INT_1	ISIF_INT_0	

Bits	Field Name	Description	Type	Reset
31	OCPE_ERR_IRQ	An OCP error has been received on the ISP master port. Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
30	RESERVED		R	0
29	IPIPE_INT_DPC_RNEW1	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
28	IPIPE_INT_DPC_RNEW0	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
27	IPIPE_INT_DPC_INI	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
26	RESERVED		R	0
25	IPIPE_INT_EOF	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW	0
24	H3A_INT_EOF	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW	0

Bits	Field Name	Description	Type	Reset
23	RSZ_INT_EOF1	<p>RESIZER module event: This event signals that the BL has received the EOF signal from the resizer B engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW	0
22	RSZ_INT_EOF0	<p>RESIZER module event: This event signals that the BL has received the EOF signal from the resizer A engine which happens one the last transfer in the frame has happened. Note that because the BL has FIFOs it may take some additional time before true data transfer completion to memory.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW	0
21:20	RESERVED		R	0x0
19	RSZ_FIFO_IN_BLK_ERR	<p>This event signals that the minimum vertical blanking period has not been respected causing errors in the input data buffering submodule. This event will be triggered when the rsz_int_reg event of frame N is triggered before the rsz_int_dma of frame N + 1. This event would typically happen at the transition between two frames because there is not enough vertical blanking between frames: the firmware must take care to ensure enough vertical blanking. The hardware cannot recover from this error. It will be required to perform a reset of the IP.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0
18	RSZ_FIFO_OVF	<p>RESIZER module event: This event signals that overflow happened in the input data buffering submodule or in the RSZ output interface.. This event would typically happen because the video port pixel clock is too high.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
17	RSZ_INT_CYC_RZB	<p>RESIZER module event: This event is the circular interrupt for RESIZER #B. An event can be triggered every time that <a href="#">RSZ_IRQ_RZB</a> output lines have been written out to the RZB_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, this value should be such that the circular buffer vertical size (set by the <a href="#">RZB_SDR_Y_PTR_E</a> register) is a multiple of <a href="#">RSZ_IRQ_RZB</a>.</p> <p>Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0
16	RSZ_INT_CYC_RZA	<p>RESIZER module event: This event is the circular interrupt for RESIZER #A. An event can be triggered every time that <a href="#">RSZ_IRQ_RZA</a> output lines have been written out to the RZA_SDR_Y buffer. The range can go from 1 to 8192 lines. Usually, the circular buffer vertical size (set by the <a href="#">RZA_SDR_Y_PTR_E</a> register) should be a multiple of <a href="#">RSZ_IRQ_RZA</a>.</p> <p>Note that at the time the interrupt is triggered, the actual data write has not taken place. It may take a few hundred of cycles to complete the data write into system memory. This is not an issue since the start of the buffer is read first.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0
15	RSZ_INT_DMA	<p>RESIZER module event: This event is triggered when the last EOF (of the two MTC interfaces) is sent out to the BL and that the resizer core has returned to idle. <code>rsz_int_dma</code> is a true indication that all processing is finished for the particular frame on both resizer engines.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0
14	RSZ_INT_LAST_PIX	<p>RESIZER module event: This event is triggered when the last pixel of the valid area is received.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0
13	RSZ_INT_REG	<p>RESIZER module event: This event is triggered when the first pixel of the valid area is received. Shadowed registers can be updated at any time but the new value will take effect on the next <code>rsz_int_reg</code> event.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
12	H3A_INT	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
11	AF_INT	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
10	AEW_INT	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
9	IPIPEIF_IRQ	IPIPEIF module interrupt Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
8	IPIPE_INT_HST	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
7	IPIPE_INT_BSC	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
6	IPIPE_INT_DMA	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
5	IPIPE_INT_LAST_PIX	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
4	IPIPE_INT_REG	Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0
3	ISIF_INT_3	Set this bit to disable the LSC interrupt issued by the 2D-LSC block. Four types of 2D-LSC can be generated and mapped to the INT_3 line. For more information, see <a href="#">Section 8.3.3.7.10.1.5, ISS ISP ISIF 2D-LSC Events and Status Checking</a> . Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
2	ISIF_INT_2	<p>VD interrupt 2 event. Set this bit to disable the interrupt mapped to the INT_2 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see <a href="#">Section 8.3.3.7.19.1, ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts</a>.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0
1	ISIF_INT_1	<p>VD interrupt 1 event. Set this bit to disable the interrupt mapped to the INT_1 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see <a href="#">Section 8.3.3.7.19.1, ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts</a>.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0
0	ISIF_INT_0	<p>VD interrupt 0 event. Set this bit to disable the interrupt mapped to the INT_0 line. This interrupt is set based on the VD pulse position after receiving a configured number of horizontal pulse signals. For more information, see <a href="#">Section 8.3.3.7.19.1, ISS ISP ISIF VDINT0, VDINT1 and VDINT2 Interrupts</a>.</p> <p>Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled (masked)</p>	RW W1toClr	0

**Table 8-405. Register Call Summary for Register ISP5\_IRQENABLE\_CLR\_i**

ISS ISP

- [ISS ISP RSZ Reset Behavior: \[0\]](#)
- [ISS ISP5 SYS1 Register Summary: \[1\]](#)

**NOTE:** Setting or disabling the ISP submodule interrupts mapped to the ISP lines requires a configuration to receive or disable the IRQ events at a higher ISS interrupt merger level. For information about how IRQ lines are handled at the ISP level, see [Table 8-307](#). For information about how IRQs are handled at the ISS top-level before the signals leave ISS boundaries, see [Section 8.1.2.1.1, ISS Interrupt Merger](#).

**Table 8-406. ISP5\_DMAENABLE\_SET**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	ISS_ISP5_SYS1
<b>Physical Address</b>	See <a href="#">Table 8-389</a>		
<b>Description</b>	Per-line DMA enable bit vector Write 1 to set (enable DMA request generation). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IPIPE_INT_DPC_RNEW1		IPIPE_INT_LAST_PIX		IPIPE_INT_DPC_RNEW0		IPIPE_INT_HST		IPIPE_INT_BSC							

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000000
4	IPIPE_INT_DPC_RNEW1	Enable for ISP DMA request generation on line #2 This DMA request must be set to transfer the DPC data from memory to the IPIPE internal RAM. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0
3	IPIPE_INT_LAST_PIX	Enable for ISP DMA request generation on line #3 This DMA request must be set to transfer the GAMMA data from memory to the IPIPE internal RAM or to initialize the DPC table. One must set the <a href="#">ISP5_CTRL.DMA3_CFG</a> register before enabling this DMA request. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0
2	IPIPE_INT_DPC_RNEW0	Enable for ISP DMA request generation on line #2 This DMA request must be set to transfer the DPC data from memory to the IPIPE internal RAM. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0
1	IPIPE_INT_HST	Enable for ISP DMA request generation on line #1 This DMA request must be set to transfer the HIST data from the IPIPE internal RAM to memory. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0
0	IPIPE_INT_BSC	Enable for ISP DMA request generation on line #0 This DMA request must be set to transfer the BSC data from the IPIPE internal RAM to memory. Write 0x0: No action Write 0x1: Enable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toSet	0

**Table 8-407. Register Call Summary for Register ISP5\_DMAENABLE\_SET**

ISS ISP

- [ISS ISP DMA Requests: \[0\]](#)
- [ISS ISP5 SYS1 Register Summary: \[1\]](#)

**Table 8-408. ISP5\_DMAENABLE\_CLR**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	ISS_ISP5_SYS1
<b>Physical Address</b>	See <a href="#">Table 8-389</a>		
<b>Description</b>	Per-line DMA clear bit vector Write 1 to clear (disable DMA request generation). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IPIPE_INT_DPC_RNEW1	IPIPE_INT_LAST_PIX	IPIPE_INT_DPC_RNEW0	IPIPE_INT_HST	IPIPE_INT_BSC											

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x00000000
4	IPIPE_INT_DPC_RNEW1	Clear for ISP DMA request generation on line ISS_DMA2. This DMA request must be set to transfer the DPC data from memory to the IPIPE internal RAM.  Write 0x0: No action Write 0x1: Disable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toClr	0
3	IPIPE_INT_LAST_PIX	Clear for ISP DMA request generation on ISS_DMA3. This DMA request must be set to transfer the GAMMA data from memory to the IPIPE internal RAM.  Write 0x0: No action Write 0x1: Disable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toClr	0
2	IPIPE_INT_DPC_RNEW0	Clear for ISP DMA request generation on ISS_DMA2. This DMA request must be set to transfer the DPC data from memory to the IPIPE internal RAM.  Write 0x0: No action Write 0x1: Disable DMA Read 0x1: DMA enabled Read 0x0: DMA disabled	RW W1toClr	0



Bits	Field Name	Description	Type	Reset
31:30	DMA3_CFG	<p>This bit selects the DMA transfer configuration which is used with the ISS_DMA3 DMA request signal. This DMA request is generated from IPIPE_INT_LAST_PIXEL event. One can choose to use this DMA request to transfer the DPC initialization data, the gamma table, or both.</p> <p>0x0: No DMA request associated with ISS_DMA3.</p> <p>0x1: DPC DMA request associated with ISS_DMA3. Expected DMA transfer size is 2KB in the range 0x8000-0x87FF. DPC_EVT_INI must be set to 0.</p> <p>0x3: DPC + GAMMA DMA request associated with ISS_DMA3. Expected DMA transfer size is 8KB in the range 0x8000-0x87FF and 0xA800-0xBFFF. DPC_EVT_INI must be set to 0.</p> <p>0x2: GAMMA DMA request associated with ISS_DMA3. Expected DMA transfer size is 6KB in the range 0xA800-0xBFFF.</p>	RW	0x0
29:28	RESERVED		R	0x0
27	RESERVED		R	0x0
26	HST_RD_CHK	<p>When the HISTOGRAM computation is enabled and the HST DMA request is not used to read out the data, this register ensures that the data is read fast enough, else an interrupt <a href="#">ISP5_IRQSTATUS2_i[5]</a> IPIPE_HST_ERR is triggered.</p> <p>The hardware sets automatically this bit to 1 when software can start reading the memory.</p>	RW	0
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p><b>CAUTION</b></p> <p>It is the software responsibility to set this bit to 0 after reading the data. Once the MPU has read the histogram data, it must clear this register, else the <a href="#">ISP5_IRQSTATUS2_i[5]</a> IPIPE_HST_ERR will occur.</p> </div>				
<p>Write 0x0: Clears the signal to avoid error generation. The software must write this bit to 0 after the last data read.</p> <p>Write 0x1: Reserved</p> <p>Read 0x1: The MPU can read the data from the memory. Needs to complete fast enough to avoid the interrupt generation.</p> <p>Read 0x0: No interrupt generation can happen</p>				
25	DPC_EVT_INI	<p>Select the IPIPE module event to be used to generate the DMA requests for the DPC submodule.</p> <p>0x0: IPIPE_INT_LAST_PIX event is selected.</p> <p>0x1: IPIPE_INT_DPC_INI event is selected.</p>	RW	0

Bits	Field Name	Description	Type	Reset
24	MSTANDBY	<p>MStandby signal assertion and de-assertion control for power management transitions.</p> <p>After software reset, this bit is asserted.</p> <p>Write "1" to transition from normal mode to idle mode. The firmware needs to ensure that no more ISP processing is ongoing before setting up this bit.</p> <p>Write "0" to transition from idle mode to normal mode. The software should poll <code>ISP5_CTRL.MSTANDBY_WAIT = 0</code> after writing <code>ISP5_CTRL.MSTANDBY = 0</code> in a transition from idle to normal mode.</p> <p>0x0: De-assert MStandby signal. May not be immediate due to power management handshaking btw the MStandby and Wait signals.</p> <p>0x1: Assert MStandby signal</p>	RW	1
23	VD_PULSE_EXT	<p>VD pulse extension enable</p> <p>This bit enables or disables the VD extension bridge. By default, the bridge is enabled. At ISS level, it is expected that <code>ISP5_CTRL.VD_PULSE_EXT = 1</code> when the VPORT gets data from the CSI2 RX module and <code>ISP5_CTRL.VD_PULSE_EXT = 0</code> when the VPORT gets data from the parallel interface or the CCP2 RX module. There must be at least three clock cycles between the time this bit is modified and the HD/VD pulse for start of frame comes.</p> <p>0x0: Disabled</p> <p>0x1: Enabled</p>	RW	1
22	PCLK_INV	<p>Pixel clock inversion</p> <p>This bit enables or disables pixel clock inversion. The ISP always samples the data on the rising edge of the pixel clock. Enabling the inversion shifts the resampling period by 1/2 a pixel clock period.</p> <p>0x0: Normal</p> <p>0x1: Inversed</p>	RW	0
21	MFLAG	<p>MFlag signal generation control</p> <p>This bit controls how the OCP MFlag signal is generated on the ISS NOC.</p> <p>0x0: The MFlag value is dynamic.</p> <p>0x1: The MFlag value is static. The value is set with the <code>ISP5_CTRL[3:1] VBUSM_CPRIORITY</code>.</p>	RW	0
20	MSTANDBY_WAIT	<p>MStandby / Wait power management status bit.</p> <p>The power management framework of the ISP is based on the handshaking of the MStandby and Wait signals. The software is not supposed to write inside the ISP slave port and initiate traffic when <code>ISP5_CTRL.MSTANDBY</code> bit is written. The software can poll this bit to know when Wait signal is deasserted.</p> <p>Read 0x1: MStandby signal is asserted.</p> <p>Read 0x0: MStandby signal is deasserted.</p>	R	-
19:16	RESERVED		R	0x0
15	BL_CLK_ENABLE	<p>BL clock enable</p> <p>0x0: Disable</p> <p>0x1: Enable</p>	RW	0
14	ISIF_CLK_ENABLE	<p>ISIF clock enable</p> <p>The ISP will return <code>OCP_ERROR</code> if one tries to program the module MMR or memory when the clock is disabled. There must be at least three clock cycles between the time this bit is modified and the HD/VD pulse for start of frame comes.</p> <p>0x0: Disable</p> <p>0x1: Enable</p>	RW	0

Bits	Field Name	Description	Type	Reset
13	H3A_CLK_ENABLE	H3A clock enable The ISP will return OCP_ERROR if one tries to program the module MMR or memory when the clock is disabled. 0x0: Disable 0x1: Enable	RW	0
12	RSZ_CLK_ENABLE	RESIZER clock enable The ISP will return OCP_ERROR if one tries to program the module MMR or memory when the clock is disabled. 0x0: Disable 0x1: Enable	RW	0
11	IPIPE_CLK_ENABLE	IPIPE clock enable The ISP will return OCP_ERROR if one tries to program the module MMR or memory when the clock is disabled. 0x0: Disable 0x1: Enable	RW	0
10	IPIPEIF_CLK_ENABLE	IPIPEIF clock enable The ISP will return OCP_ERROR if one tries to program the module MMR or memory when the clock is disabled. 0x0: Disable 0x1: Enable	RW	0
9	SYNC_ENABLE	PCLK Sync module enable 0x0: Disable. must be used only when the video port is not receiving data; for example, data is read from the IPIPEIF module memory read port. 0x1: Enable. must only be used when the video port is not receiving data.	RW	0
8	PSYNC_CLK_SEL	PCLK Sync clock select. This bit selects the clock which is used to resynchronize the input pixel clock. 0x0: MMR_CLK. Can be used if the input pixel clock is always lower than 100 MHz. 0x1: DMA_CLK. must be used if the pixel clock is higher than 100 MHz.	RW	0
7:4	VBUSM_CIDS	BL MAX VBUSM CIDS The BL module supports up to 16 CIDs/tags. This bit field sets up the maximum number of CIDs/tags that the BL can use. The actual number of CIDs/tags is setup by VBUSM_CIDS + 1. Tag number 0 to VBUSM_CIDS are used.	RW	0xF
3:1	VBUSM_CPRIORITY	BL VBUSM priority setting 0x6: Normal Priority VBUSM cpriority[2:0] = 6 0x1: High Priority VBUSM cpriority[2:0] = 1 0x7: Normal Priority VBUSM cpriority[2:0] = 7 0x0: High Priority VBUSM cpriority[2:0] = 0 0x2: Medium Priority VBUSM cpriority[2:0] = 2 0x4: Normal Priority VBUSM cpriority[2:0] = 4 0x5: Normal Priority VBUSM cpriority[2:0] = 5 0x3: Medium Priority VBUSM cpriority[2:0] = 3	RW	0x4

Bits	Field Name	Description	Type	Reset
0	OCP_WRNP	ISP OCP master port non-posted write control. 0x0: All writes are non posted. 0x1: All writes are posted.	RW	0

**Table 8-411. Register Call Summary for Register ISP5\_CTRL**

ISS ISP

- [ISS ISP Clocks: \[0\]](#)
- [ISS ISP Reset: \[1\] \[2\]](#)
- [ISS ISP DMA Requests: \[3\] \[4\]](#)
- [ISS ISP VP Top-Level Communication With CCP2 RX and CSI2 RX: \[6\] \[7\] \[8\]](#)
- [ISS ISP VP Pixel Clock Inversion: \[9\] \[10\] \[11\]](#)
- [ISS ISP BL Functional Description: \[12\]](#)
- [ISS ISP BL Out-of-Order Responses: \[13\]](#)
- [ISS ISP BL Dynamic and Static MFlag Generation: \[14\] \[15\] \[16\]](#)
- [ISS ISP BL Programming Model: \[17\] \[18\] \[19\] \[20\]](#)
- [ISS ISP5 SYS1 Register Summary: \[21\]](#)
- [ISS ISP5 SYS1 Register Description: \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\]](#)
- [ISS ISP5 SYS2 Register Description: \[34\]](#)

**Table 8-412. ISP5\_PG**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	ISS_ISP5_SYS1
<b>Physical Address</b>	See <a href="#">Table 8-389</a>		
<b>Description</b>	PATTERN GENERATOR REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SRC_SEL		EN	WEN	HDPOL	VDPOL										

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5:4	SRC_SEL	Input mux selection 0x0: ISP video port is selected. 0x1: Reserved 0x3: Pattern generator is selected. 0x2: Reserved	RW	0x0
3	EN	0x0: Pattern generator off 0x1: Enable pattern generator	RW	0
2	WEN	0x0: WEN is always activated 0x1: WEN is on 8 cycles and off 8 cycles	RW	0
1	HDPOL	0x0: Active high 0x1: Active low	RW	0
0	VDPOL	0x0: Active high 0x1: Active low	RW	0

**Table 8-413. Register Call Summary for Register ISP5\_PG**

ISS ISP

- [ISS ISP PG Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP PG Programming Model: \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [ISS ISP5 SYS1 Register Summary: \[10\]](#)

**Table 8-414. ISP5\_PG\_PULSE\_CTRL**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	ISS_ISP5_SYS1
<b>Physical Address</b>	See <a href="#">Table 8-389</a>		
<b>Description</b>	PATTERN GENERATOR REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VDW								RESERVED		HDW													

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	VDW	Pattern generator VD width Width = VDW+1	RW	0x000
15:13	RESERVED		R	0x0
12:0	HDW	Pattern generator HD width Width = HDW+1	RW	0x0000

**Table 8-415. Register Call Summary for Register ISP5\_PG\_PULSE\_CTRL**

ISS ISP

- [ISS ISP PG Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP PG Programming Model: \[5\] \[6\]](#)
- [ISS ISP5 SYS1 Register Summary: \[7\]](#)

**Table 8-416. ISP5\_PG\_FRAME\_SIZE**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	ISS_ISP5_SYS1
<b>Physical Address</b>	See <a href="#">Table 8-389</a>		
<b>Description</b>	PATTERN GENERATOR REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPLN								HLPFR																							

Bits	Field Name	Description	Type	Reset
31:16	PPLN	Pattern Generator: pixels per line, PPLN+1	RW	0x0000
15:0	HLPFR	Pattern Generator: half lines per frame, HLPFR+1	RW	0x0000

**Table 8-417. Register Call Summary for Register ISP5\_PG\_FRAME\_SIZE**

ISS ISP

- [ISS ISP PG Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP PG Programming Model: \[5\] \[6\]](#)
- [ISS ISP5 SYS1 Register Summary: \[7\]](#)



**Table 8-418. ISP5\_MPSR**

<b>Address Offset</b>	0x0000 007C
<b>Physical Address</b>	See <a href="#">Table 8-389</a>
<b>Instance</b>	ISS_ISP5_SYS1
<b>Description</b>	ISP memory access register. One need to pay attention when setting the bit fields in this register such that there is no conflict between the CPU and module accesses. Usually, the ISP modules must have access to the memories and it is only when the ISP is idle (vertical blanking period or module disabled that the CPU can access the memories.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
RESERVED								IPIPE_GAMMA_RGB_COPY	RESERVED								IPIPE_BSC_TB1	IPIPE_BSC_TB0	IPIPE_HST_TB3	IPIPE_HST_TB2	IPIPE_HST_TB1	IPIPE_HST_TB0	IPIPE_D3L_TB3	IPIPE_D3L_TB2	IPIPE_D3L_TB1	IPIPE_D3L_TB0	IPIPE_GBC_TB	IPIPE_YEE_TB	IPIPE_GMM_TBR	IPIPE_GMM_TBG	IPIPE_GMM_TBB	IPIPE_DPC_TB	ISIF_DCLAMP	ISIF_LSC_TB1	ISIF_LSC_TB0	ISIF_LIN_TB	RESERVED

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	IPIPE_GAMMA_RGB_COPY	GAMMA table RGB Copy This bit must be enable when one wants to use the same Gamma table for the R, G and B color components. When the CPU writes the R table, it is automatically copied to the G and B tables if this bit is set.  0x0: Copy disable Independent RGB gamma table 0x1: Copy enable Common RGB Gamma table	RW	0
23:21	RESERVED		R	0x0
20	IPIPE_BSC_TB1	IPIPE BSC TB1 memory access priority This memory is expected to be read by the CPU or the DMA to get BSC information during vertical blanking period.  0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
19	IPIPE_BSC_TB0	IPIPE BSC TB0 memory access priority This memory is expected to be read by the CPU or the DMA to get BSC information during vertical blanking period.  0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
18	IPIPE_HST_TB3	IPIPE histogram memory #3 access priority This memory is expected to be read by the CPU or the DMA to get HST information during vertical blanking period.  0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
17	IPIPE_HST_TB2	IPIPE histogram memory #2 access priority This memory is expected to be read by the CPU or the DMA to get HST information during vertical blanking period.  0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0

Bits	Field Name	Description	Type	Reset
16	IPIPE_HST_TB1	IPIPE histogram memory #1 access priority This memory is expected to be read by the CPU or the DMA to get HST information during vertical blanking period. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
15	IPIPE_HST_TB0	IPIPE histogram memory #0 access priority This memory is expected to be read by the CPU or the DMA to get HST information during vertical blanking period. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
14	IPIPE_D3L_TB3	D3L TB3 memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
13	IPIPE_D3L_TB2	D3L TB2 memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
12	IPIPE_D3L_TB1	D3L TB1 memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
11	IPIPE_D3L_TB0	D3L TB0 memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
10	IPIPE_GBC_TB	IPIPE GBC TB memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
9	IPIPE_YEE_TB	YEE TB memory access priority This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0
8	IPIPE_GMM_TBR	IPIPE Gamma LUT R memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods. 0x0: MODULE access has higher priority 0x1: CPU access has higher priority.	RW	0

Bits	Field Name	Description	Type	Reset
7	IPIPE_GMM_TBG	<p>IPIPE Gamma LUT G memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods.</p> <p>0x0: MODULE access has higher priority 0x1: CPU access has higher priority.</p>	RW	0
6	IPIPE_GMM_TBB	<p>IPIPE Gamma LUT B memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods.</p> <p>0x0: MODULE access has higher priority 0x1: CPU access has higher priority.</p>	RW	0
5	IPIPE_DPC_TB	<p>IPIPE defect pixel memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods.</p> <p>0x0: MODULE access has higher priority 0x1: CPU access has higher priority.</p>	RW	0
4	ISIF_DCLAMP	<p>ISIF DC accumulation memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods.</p> <p>0x0: MODULE access has higher priority 0x1: CPU access has higher priority.</p>	RW	0
3	ISIF_LSC_TB1	<p>ISIF LSC memory 1 access This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods.</p> <p>0x0: MODULE has memory access When the module has memory access, the potential concurrent CPU accesses (on the ISP memory map) to read the memory are stalled. The CPU will eventually get back the data during the horizontal or vertical blanking periods when the module is not making access anymore.</p> <p>0x1: CPU has memory access When the CPU has memory access (read or write), it will cause data corruption if the module tries to perform concurrent memory accesses. The module cannot know that the read or write access has not taken place because of CPU accesses.</p>	RW	0
2	ISIF_LSC_TB0	<p>ISIF LSC memory 0 access This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods.</p> <p>0x0: MODULE has memory access. When the module has memory access, the potential concurrent CPU accesses (on the ISP memory map) to read the memory are stalled. The CPU will eventually get back the data during the horizontal or vertical blanking periods when the module is not making access anymore.</p> <p>0x1: CPU has memory access When the CPU has memory access (read or write), it will cause data corruption if the module tries to perform concurrent memory accesses. The module cannot know that the read or write access has not taken place because of CPU accesses.</p>	RW	0
1	ISIF_LIN_TB	<p>ISIF linearity compensation memory arbitration This memory is expected to be written during ISP initialization and potentially updated during vertical blanking periods.</p> <p>0x0: MODULE access has higher priority 0x1: CPU access has higher priority.</p>	RW	0

Bits	Field Name	Description	Type	Reset
0	RESERVED		R	0

**Table 8-419. Register Call Summary for Register ISP5\_MPSR**

ISS ISP

- [ISS ISP BL Programming Model: \[0\]](#)
- [ISS ISP5 SYS1 Register Summary: \[1\]](#)

**Table 8-420. ISP5\_BL\_MTC\_1**

<b>Address Offset</b>	0x0000 0080	<b>Instance</b>	ISS_ISP5_SYS1
<b>Physical Address</b>	See <a href="#">Table 8-389</a>		
<b>Description</b>	MEMORY REQUEST MINIMUM INTERVAL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISIF_R																IPIPEIF_R															

Bits	Field Name	Description	Type	Reset
31:16	ISIF_R	Sets the minimum interval btw two consecutive memory requests for the ISIF-Read port. Specified in number of interface clock cycles.	RW	0x0000
15:0	IPIPEIF_R	Sets the minimum interval btw two consecutive memory requests for the IPIPEIF-Read port. Specified in number of interface clock cycles.	RW	0x0000

**Table 8-421. Register Call Summary for Register ISP5\_BL\_MTC\_1**

ISS ISP

- [ISS ISP ISIF Read Port: \[0\]](#)
- [ISS ISP BL Peak Memory Bandwidth Reduction: \[1\]](#)
- [ISS ISP BL Programming Model: \[2\] \[3\] \[4\]](#)
- [ISS ISP5 SYS1 Register Summary: \[5\]](#)

**Table 8-422. ISP5\_BL\_MTC\_2**

<b>Address Offset</b>	0x0000 0084	<b>Instance</b>	ISS_ISP5_SYS1
<b>Physical Address</b>	See <a href="#">Table 8-389</a>		
<b>Description</b>	MEMORY REQUEST MINIMUM INTERVAL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H3A_W																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	H3A_W	Sets the minimum interval btw two consecutive memory requests for the H3A-Write port. Specified in number of interface clock cycles.	RW	0x0000
15:0	RESERVED		R	0x0000

**Table 8-423. Register Call Summary for Register ISP5\_BL\_MTC\_2**

ISS ISP

- [ISS ISP BL Peak Memory Bandwidth Reduction: \[0\]](#)
- [ISS ISP5 SYS1 Register Summary: \[1\]](#)

**Table 8-424. ISP5\_BL\_VBUSM**

<b>Address Offset</b>	0x0000 0088
<b>Physical Address</b>	See <a href="#">Table 8-389</a>
<b>Description</b>	<p>Instance ISS_ISP5_SYS1</p> <p>BL VBUSM TUNING REGISTER</p> <p>The settings in the register are static and not expected to be modified dynamically.</p>
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MFLAG_THRES		LASTCMD_DLY													

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5	MFLAG_THRES	<p>MFLAG Threshold value</p> <p>The value of this bit field is a threshold which is compared to the MFlag output of the ISP5. If the BL MFlag signal is greater or equal to this threshold the last beat of the VBUSM command is delayed by <a href="#">ISP5_BL_VBUSM[4:0] LASTCMD_DLY</a> cycles. Only values 0, 1 are valid, the least significant bit is tied off to 1 to make a 2-bit field.</p> <p>0x0: Thres = 1</p> <p>0x1: Thres = 3</p>	RW	1
4:0	LASTCMD_DLY	<p>The value of this bit field represents a delay expressed in cycles (L3 clock). This value is used to delay the last beat of the VBUSM command such that the ISP does not loose arbitration at the ISS level because the BL does not generate back to back requests by default. The last beat is delayed until the counter expires or the new request is accepted.</p> <p>This delay is used when the MFlag output of the ISP is greater or equal to <a href="#">ISP5_BL_VBUSM[5] MFLAG_THRES</a>. One can set this value to 0 to disable the last command beat delay.</p>	RW	0x04

**Table 8-425. Register Call Summary for Register ISP5\_BL\_VBUSM**

- ISS ISP
- [ISS ISP BL VBUSM2OCP Last Beat Command Delay: \[0\] \[1\] \[2\] \[3\]](#)
  - [ISS ISP5 SYS1 Register Summary: \[4\]](#)
  - [ISS ISP5 SYS1 Register Description: \[5\] \[6\]](#)

**8.3.5.3 ISS ISP5 SYS2 Registers**

**CAUTION**

The ISS ISP5 SYS2 registers are limited to 32 bit and 16 bit data accesses; 8bit data access is not allowed and can corrupt register content.

**8.3.5.3.1 ISS ISP5 SYS2 Register Summary**

**Table 8-426. ISS ISP5 SYS2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISP5_SYS2 Base Address
ISP5_KEY_EN1	R	32	0x0000 0000	0x5201 00A0
ISP5_KEY_EN2	R	32	0x0000 0004	0x5201 00A4
ISP5_KEY_EN3	R	32	0x0000 0008	0x5201 00A8
ISP5_KEY_EN4	R	32	0x0000 000C	0x5201 00AC
ISP5_KEY_EN5	R	32	0x0000 0010	0x5201 00B0
ISP5_KEY_EN6	R	32	0x0000 0014	0x5201 00B4
ISP5_IRQSTATUS_RA W2_j <sup>(1)</sup>	RW	32	0x0000 0018 + (0x10 * i)	0x5201 00B8 + (0x10 * i)
ISP5_IRQSTATUS2_j <sup>(1)</sup>	RW	32	0x0000 001C + (0x10 * i)	0x5201 00BC + (0x10 * i)
ISP5_IRQENABLE_SET 2_j <sup>(1)</sup>	RW	32	0x0000 0020 + (0x10 * i)	0x5201 00C0 + (0x10 * i)
ISP5_IRQENABLE_CLR 2_j <sup>(1)</sup>	RW	32	0x0000 0024 + (0x10 * i)	0x5201 00C4 + (0x10 * i)

<sup>(1)</sup> i = 0 to 3**8.3.5.3.2 ISS ISP5 SYS2 Register Description****Table 8-427. ISP5\_KEY\_EN1**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	ISS_ISP5_SYS2
<b>Physical Address</b>	See <a href="#">Table 8-426</a>		
<b>Description</b>	IPIPE eFuse enable.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	KEY1_EN														

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY1_EN	EFUSE enable Equals 1 when ISP5_EFUSE3_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0

**Table 8-428. Register Call Summary for Register ISP5\_KEY\_EN1**

ISS ISP

- [ISS ISP5 SYS2 Register Summary: \[0\]](#)

**Table 8-429. ISP5\_KEY\_EN2**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	ISS_ISP5_SYS2
<b>Physical Address</b>	See <a href="#">Table 8-426</a>		
<b>Description</b>	ISIF eFuse enable.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												KEY1_EN			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY1_EN	EFUSE enable Equals 1 when ISP5_EFUSE1_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	1

**Table 8-430. Register Call Summary for Register ISP5\_KEY\_EN2**

ISS ISP

- [ISS ISP5 SYS2 Register Summary: \[0\]](#)

**Table 8-431. ISP5\_KEY\_EN3**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	ISS_ISP5_SYS2
<b>Physical Address</b>	See <a href="#">Table 8-426</a>		
<b>Description</b>	ISIF eFuse enable.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												KEY_EN			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY_EN	EFUSE enable Equals 1 when ISP5_EFUSE3_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0

**Table 8-432. Register Call Summary for Register ISP5\_KEY\_EN3**

ISS ISP

- [ISS ISP5 SYS2 Register Summary: \[0\]](#)

**Table 8-433. ISP5\_KEY\_EN4**

<b>Address Offset</b>	0x0000 000C		
<b>Physical Address</b>	See <a href="#">Table 8-426</a>	<b>Instance</b>	ISS_ISP5_SYS2
<b>Description</b>	IPIPEIF eFuse enable.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											KEY2_EN	KEY1_EN			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	KEY2_EN	EFUSE enable Equals 1 when ISP5_EFUSE4_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0
0	KEY1_EN	EFUSE enable Equals 1 when ISP5_EFUSE1_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	1

**Table 8-434. Register Call Summary for Register ISP5\_KEY\_EN4**

ISS ISP

- [ISS ISP5 SYS2 Register Summary: \[0\]](#)

**Table 8-435. ISP5\_KEY\_EN5**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 8-426</a>	<b>Instance</b>	ISS_ISP5_SYS2
<b>Description</b>	H3A eFuse enable.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											KEY_EN				

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY_EN	EFUSE enable Equals 1 when ISP5_EFUSE2_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0

**Table 8-436. Register Call Summary for Register ISP5\_KEY\_EN5**

ISS ISP

- [ISS ISP5 SYS2 Register Summary: \[0\]](#)



**Table 8-437. ISP5\_KEY\_EN6**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	ISS_ISP5_SYS2
<b>Physical Address</b>	See <a href="#">Table 8-426</a>		
<b>Description</b>	H3A eFuse enable.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												KEY_EN			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	KEY_EN	EFUSE enable Equals 1 when ISP5_EFUSE3_EN = 1 or 0 otherwise. Read 0x1: Enable Read 0x0: Disable	R	0

**Table 8-438. Register Call Summary for Register ISP5\_KEY\_EN6**

ISS ISP

- [ISS ISP5 SYS2 Register Summary: \[0\]](#)

**Table 8-439. ISP5\_IRQSTATUS\_RAW2\_i**

<b>Address Offset</b>	0x0000 0018 + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-426</a>	<b>Instance</b>	ISS_ISP5_SYS2
<b>Description</b>	Per-event raw interrupt status vector. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug. Overflow / underflow errors are not recoverable at ISP level, a software reset is required at ISS level. The ISP outputs 4 interrupt lines ISP5_IRQ0 to ISP5_IRQ3. Any internal ISP event can be merged on the 4 lines. A same event must be enabled on only one interrupt line.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																												RESERVED	IPIPE_HST_ERR	ISIF_OVF	IPIPE_BOXCAR_OVF	IPIPEIF_UDF	H3A_OVF

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5	RESERVED		R	0x0
4	IPIPE_HST_ERR	<p>IPIPE HISTOGRAM memory read error This error will happen when the histogram data is not read fast enough by the MPU or the DMA. When the data is read with the MPU, one need to pay attention to clear the <a href="#">ISP5_CTRL[26]</a> HST_RD_CHK bit immediately after reading the last data, else this event will be set.</p> <p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
3	ISIF_OVF	<p>ISIF module overflow</p> <p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
2	IPIPE_BOXCAR_OVF	<p>IPIPE BOXCAR module overflow</p> <p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
1	IPIPEIF_UDF	<p>IPIPEIF module underflow interrupt</p> <p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0
0	H3A_OVF	<p>H3A module overflow interrupt.</p> <p>Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending Read 0x0: No event pending</p>	RW W1toSet	0

**Table 8-440. Register Call Summary for Register ISP5\_IRQSTATUS\_RAW2\_i**

## ISS ISP

- [ISS ISP IPIPEIF Module Events and Status Checking: \[0\]](#)
- [ISS ISP IPIPEIF Interframe Operations: \[1\]](#)
- [ISS ISP5 SYS2 Register Summary: \[2\]](#)

**Table 8-441. ISP5\_IRQSTATUS2\_i**

<b>Address Offset</b>	0x0000 001C + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-426</a>	<b>Instance</b>	ISS_ISP5_SYS2
<b>Description</b>	<p>Per-event "enabled" interrupt status vector. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). Overflow / underflow errors are not recoverable at ISP level, a software reset is required at ISS level. The ISP outputs 4 interrupt lines ISP5_IRQ0 to ISP5_IRQ3. Any internal ISP event can be merged on the 4 lines. A same event must be enabled on only one interrupt line.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		IPIPE_HST_ERR	ISIF_OVF	IPIPE_BOXCAR_OVF	IPIPEIF_UDF	H3A_OVF									

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5	RESERVED		R	0x0
4	IPIPE_HST_ERR	IPIPE HISTOGRAM memory read error This error will happen when the histogram data is not read fast enough by either the MPU or the DMA. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
3	ISIF_OVF	ISIF module overflow Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
2	IPIPE_BOXCAR_OVF	IPIPE BOXCAR module overflow Overflow errors are not recoverable at ISP level, a software reset is required at ISS level. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
1	IPIPEIF_UDF	IPIPEIF module underflow interrupt Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0
0	H3A_OVF	H3A module overflow interrupt. Overflow errors are not recoverable at ISP level, a software reset is required at ISS level. Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending Read 0x0: No (enabled) event pending	RW W1toClr	0

**Table 8-442. Register Call Summary for Register ISP5\_IRQSTATUS2\_i**

ISS ISP

- [ISS ISP DMA Requests: \[1\]](#)
- [ISS ISP5 SYS1 Register Description: \[4\] \[5\]](#)
- [ISS ISP5 SYS2 Register Summary: \[6\]](#)

**Table 8-443. ISP5\_IRQENABLE\_SET2\_i**

<b>Address Offset</b>	0x0000 0020 + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-426</a>	<b>Instance</b>	ISS_ISP5_SYS2
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register. Overflow / underflow errors are not recoverable at ISP level, a software reset is required at ISS level. The ISP outputs 4 interrupt lines ISP5_IRQ0 to ISP5_IRQ3. Any internal ISP event can be merged on the 4 lines. A same event must be enabled on only one interrupt line.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		PIPE_HST_ERR	ISIF_OVF	PIPE_BOXCAR_OVF	PIPEIF_UDF	H3A_OVF									

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5	RESERVED		R	0x0
4	IPIPE_HST_ERR	IPIPE HISTOGRAM memory read error This error will happen when the histogram data is not read fast enough by either the MPU or the DMA.	RW W1toSet	0
3	ISIF_OVF	ISIF module overflow	RW W1toSet	0
2	IPIPE_BOXCAR_OVF	IPIPE BOXCAR module overflow	RW W1toSet	0
1	PIPEIF_UDF	PIPEIF module underflow interrupt	RW W1toSet	0
0	H3A_OVF	H3A module overflow interrupt. Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled Read 0x0: Interrupt disabled	RW W1toSet	0

**Table 8-444. Register Call Summary for Register ISP5\_IRQENABLE\_SET2\_i**

## ISS ISP

- [ISS ISP Interrupt Tree: \[0\] \[2\] \[3\] \[4\] \[5\]](#)
- [ISS ISP PIPEIF Interrupts: \[6\]](#)
- [ISS ISP PIPE Interrupts: \[7\]](#)
- [ISS ISP H3A Interrupts: \[8\]](#)
- [ISS ISP ISIF Interrupts: \[9\]](#)
- [ISS ISP5 SYS2 Register Summary: \[10\]](#)

**Table 8-445. ISP5\_IRQENABLE\_CLR2\_i**

<b>Address Offset</b>	0x0000 0024 + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 8-426</a>	<b>Instance</b>	ISS_ISP5_SYS2
<b>Description</b>	<p>Per-event interrupt enable bit vector.          Write 1 to clear (disable interrupt).          Readout equal to corresponding _SET register.          Overflow / underflow errors are not recoverable at ISP level, a software reset is required at ISS level.          The ISP outputs 4 interrupt lines ISP5_IRQ0 to ISP5_IRQ3. Any internal ISP event can be merged on the 4 lines. A same event must be enabled on only one interrupt line.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		IPIPE_HST_ERR	ISIF_OVF	IPIPE_BOXCAR_OVF	IPIPEIF_UDF	H3A_OVF									

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5	RESERVED		R	0x0
4	IPIPE_HST_ERR	<p>IPIPE HISTOGRAM memory read error            This error will happen when the histogram data is not read fast enough by either the MPU or the DMA.            Write 0x0: No action            Write 0x1: Disable Interrupt            Read 0x1: Interrupt enabled            Read 0x0: Interrupt disabled</p>	RW W1toClr	0
3	ISIF_OVF	<p>ISIF module overflow            Write 0x0: No action            Write 0x1: Disable Interrupt            Read 0x1: Interrupt enabled            Read 0x0: Interrupt disabled</p>	RW W1toClr	0
2	IPIPE_BOXCAR_OVF	<p>IPIPE BOXCAR module overflow            Write 0x0: No action            Write 0x1: Disable Interrupt            Read 0x1: Interrupt enabled            Read 0x0: Interrupt disabled</p>	RW W1toClr	0
1	IPIPEIF_UDF	<p>IPIPEIF module underflow interrupt            Write 0x0: No action            Write 0x1: Disable Interrupt            Read 0x1: Interrupt enabled            Read 0x0: Interrupt disabled</p>	RW W1toClr	0
0	H3A_OVF	<p>H3A module overflow interrupt.            Write 0x0: No action            Write 0x1: Disable Interrupt            Read 0x1: Interrupt enabled            Read 0x0: Interrupt disabled</p>	RW W1toClr	0

**Table 8-446. Register Call Summary for Register ISP5\_IRQENABLE\_CLR2\_i**

ISS ISP

- [ISS ISP5 SYS2 Register Summary: \[0\]](#)

### 8.3.5.4 ISS RESIZER Registers

#### 8.3.5.4.1 ISS RESIZER Register Summary

**Table 8-447. ISS RESIZER Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_RESIZER Base Address
<a href="#">RSZ_REVISION</a>	R	32	0x0000 0000	0x5201 0400
<a href="#">RSZ_SYSCONFIG</a>	RW	32	0x0000 0004	0x5201 0404
RESERVED	R	32	0x0000 0008	0x5201 0408
<a href="#">RSZ_IN_FIFO_CTRL</a>	RW	32	0x0000 000C	0x5201 040C
<a href="#">RSZ_GNC</a>	R	32	0x0000 0010	0x5201 0410
<a href="#">RSZ_FRACDIV</a>	RW	32	0x0000 0014	0x5201 0414
<a href="#">RSZ_SRC_EN</a>	RW	32	0x0000 0020	0x5201 0420
<a href="#">RSZ_SRC_MODE</a>	RW	32	0x0000 0024	0x5201 0424
<a href="#">RSZ_SRC_FMT0</a>	RW	32	0x0000 0028	0x5201 0428
<a href="#">RSZ_SRC_FMT1</a>	RW	32	0x0000 002C	0x5201 042C
<a href="#">RSZ_SRC_VPS</a>	RW	32	0x0000 0030	0x5201 0430
<a href="#">RSZ_SRC_VSZ</a>	RW	32	0x0000 0034	0x5201 0434
<a href="#">RSZ_SRC_HPS</a>	RW	32	0x0000 0038	0x5201 0438
<a href="#">RSZ_SRC_HSZ</a>	RW	32	0x0000 003C	0x5201 043C
<a href="#">RSZ_DMA_RZA</a>	RW	32	0x0000 0040	0x5201 0440
<a href="#">RSZ_DMA_RZB</a>	RW	32	0x0000 0044	0x5201 0444
<a href="#">RSZ_DMA_STA</a>	R	32	0x0000 0048	0x5201 0448
<a href="#">RSZ_GCK_MMR</a>	RW	32	0x0000 004C	0x5201 044C
RESERVED	R	32	0x0000 0050	0x5201 0450
<a href="#">RSZ_GCK_SDR</a>	RW	32	0x0000 0054	0x5201 0454
<a href="#">RSZ_IRQ_RZA</a>	RW	32	0x0000 0058	0x5201 0458
<a href="#">RSZ_IRQ_RZB</a>	RW	32	0x0000 005C	0x5201 045C
<a href="#">RSZ_YUV_Y_MIN</a>	RW	32	0x0000 0060	0x5201 0460
<a href="#">RSZ_YUV_Y_MAX</a>	RW	32	0x0000 0064	0x5201 0464
<a href="#">RSZ_YUV_C_MIN</a>	RW	32	0x0000 0068	0x5201 0468
<a href="#">RSZ_YUV_C_MAX</a>	RW	32	0x0000 006C	0x5201 046C
<a href="#">RSZ_YUV_PHS</a>	RW	32	0x0000 0070	0x5201 0470
<a href="#">RSZ_SEQ</a>	RW	32	0x0000 0074	0x5201 0474
<a href="#">RZA_EN</a>	RW	32	0x0000 0078	0x5201 0478
<a href="#">RZA_MODE</a>	RW	32	0x0000 007C	0x5201 047C
<a href="#">RZA_420</a>	RW	32	0x0000 0080	0x5201 0480
<a href="#">RZA_I_VPS</a>	RW	32	0x0000 0084	0x5201 0484
<a href="#">RZA_I_HPS</a>	RW	32	0x0000 0088	0x5201 0488
<a href="#">RZA_O_VSZ</a>	RW	32	0x0000 008C	0x5201 048C
<a href="#">RZA_O_HSZ</a>	RW	32	0x0000 0090	0x5201 0490
<a href="#">RZA_V_PHS_Y</a>	RW	32	0x0000 0094	0x5201 0494
<a href="#">RZA_V_PHS_C</a>	RW	32	0x0000 0098	0x5201 0498
<a href="#">RZA_V_DIF</a>	RW	32	0x0000 009C	0x5201 049C
<a href="#">RZA_V_TYP</a>	RW	32	0x0000 00A0	0x5201 04A0

**Table 8-447. ISS RESIZER Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_RESIZER Base Address
RZA_V_LPF	RW	32	0x0000 00A4	0x5201 04A4
RZA_H_PHS	RW	32	0x0000 00A8	0x5201 04A8
RZA_H_PHS_ADJ	RW	32	0x0000 00AC	0x5201 04AC
RZA_H_DIF	RW	32	0x0000 00B0	0x5201 04B0
RZA_H_TYP	RW	32	0x0000 00B4	0x5201 04B4
RZA_H_LPF	RW	32	0x0000 00B8	0x5201 04B8
RZA_DWN_EN	RW	32	0x0000 00BC	0x5201 04BC
RZA_DWN_AV	RW	32	0x0000 00C0	0x5201 04C0
RZA_RGB_EN	RW	32	0x0000 00C4	0x5201 04C4
RZA_RGB_TYP	RW	32	0x0000 00C8	0x5201 04C8
RZA_RGB_BLD	RW	32	0x0000 00CC	0x5201 04CC
RZA_SDR_Y_BAD_H	RW	32	0x0000 00D0	0x5201 04D0
RZA_SDR_Y_BAD_L	RW	32	0x0000 00D4	0x5201 04D4
RZA_SDR_Y_SAD_H	RW	32	0x0000 00D8	0x5201 04D8
RZA_SDR_Y_SAD_L	RW	32	0x0000 00DC	0x5201 04DC
RZA_SDR_Y_OFT	RW	32	0x0000 00E0	0x5201 04E0
RZA_SDR_Y_PTR_S	RW	32	0x0000 00E4	0x5201 04E4
RZA_SDR_Y_PTR_E	RW	32	0x0000 00E8	0x5201 04E8
RZA_SDR_C_BAD_H	RW	32	0x0000 00EC	0x5201 04EC
RZA_SDR_C_BAD_L	RW	32	0x0000 00F0	0x5201 04F0
RZA_SDR_C_SAD_H	RW	32	0x0000 00F4	0x5201 04F4
RZA_SDR_C_SAD_L	RW	32	0x0000 00F8	0x5201 04F8
RZA_SDR_C_OFT	RW	32	0x0000 00FC	0x5201 04FC
RZA_SDR_C_PTR_S	RW	32	0x0000 0100	0x5201 0500
RZA_SDR_C_PTR_E	RW	32	0x0000 0104	0x5201 0504
RZB_EN	RW	32	0x0000 0108	0x5201 0508
RZB_MODE	RW	32	0x0000 010C	0x5201 050C
RZB_420	RW	32	0x0000 0110	0x5201 0510
RZB_I_VPS	RW	32	0x0000 0114	0x5201 0514
RZB_I_HPS	RW	32	0x0000 0118	0x5201 0518
RZB_O_VSZ	RW	32	0x0000 011C	0x5201 051C
RZB_O_HSZ	RW	32	0x0000 0120	0x5201 0520
RZB_V_PHS_Y	RW	32	0x0000 0124	0x5201 0524
RZB_V_PHS_C	RW	32	0x0000 0128	0x5201 0528
RZB_V_DIF	RW	32	0x0000 012C	0x5201 052C
RZB_V_TYP	RW	32	0x0000 0130	0x5201 0530
RZB_V_LPF	RW	32	0x0000 0134	0x5201 0534
RZB_H_PHS	RW	32	0x0000 0138	0x5201 0538
RZB_H_PHS_ADJ	RW	32	0x0000 013C	0x5201 053C
RZB_H_DIF	RW	32	0x0000 0140	0x5201 0540
RZB_H_TYP	RW	32	0x0000 0144	0x5201 0544
RZB_H_LPF	RW	32	0x0000 0148	0x5201 0548
RZB_DWN_EN	RW	32	0x0000 014C	0x5201 054C
RZB_DWN_AV	RW	32	0x0000 0150	0x5201 0550
RZB_RGB_EN	RW	32	0x0000 0154	0x5201 0554
RZB_RGB_TYP	RW	32	0x0000 0158	0x5201 0558
RZB_RGB_BLD	RW	32	0x0000 015C	0x5201 055C

**Table 8-447. ISS RESIZER Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_RESIZER Base Address
RZB_SDR_Y_BAD_H	RW	32	0x0000 0160	0x5201 0560
RZB_SDR_Y_BAD_L	RW	32	0x0000 0164	0x5201 0564
RZB_SDR_Y_SAD_H	RW	32	0x0000 0168	0x5201 0568
RZB_SDR_Y_SAD_L	RW	32	0x0000 016C	0x5201 056C
RZB_SDR_Y_OFT	RW	32	0x0000 0170	0x5201 0570
RZB_SDR_Y_PTR_S	RW	32	0x0000 0174	0x5201 0574
RZB_SDR_Y_PTR_E	RW	32	0x0000 0178	0x5201 0578
RZB_SDR_C_BAD_H	RW	32	0x0000 017C	0x5201 057C
RZB_SDR_C_BAD_L	RW	32	0x0000 0180	0x5201 0580
RZB_SDR_C_SAD_H	RW	32	0x0000 0184	0x5201 0584
RZB_SDR_C_SAD_L	RW	32	0x0000 0188	0x5201 0588
RZB_SDR_C_OFT	RW	32	0x0000 018C	0x5201 058C
RZB_SDR_C_PTR_S	RW	32	0x0000 0190	0x5201 0590
RZB_SDR_C_PTR_E	RW	32	0x0000 0194	0x5201 0594

**8.3.5.4.2 ISS RESIZER Register Description****Table 8-448. RSZ\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data**Table 8-449. Register Call Summary for Register RSZ\_REVISION**

ISS ISP

- [ISS RESIZER Register Summary: \[0\]](#)

**Table 8-450. RSZ\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	SYSTEM CONFIGURATION REGISTER This register is not shadowed. There is no standalone software reset for the resizer module.		
<b>Type</b>	RW		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RSZB_CLK_EN		RSZA_CLK_EN		RESERVED						AUTOGATING					

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	RSZB_CLK_EN	Resizer B clock enable This bit enable to enable / disable the RESIZER B clock. Note that it is a second level clock enable. This bit has effect only if <a href="#">RSZ_GCK_SDR</a> is set to 1. 0x0: off 0x1: on	RW	0
8	RSZA_CLK_EN	Resizer A clock enable This bit enable to enable / disable the RESIZER A clock. Note that it is a second level clock enable. This bit has effect only if <a href="#">RSZ_GCK_SDR</a> is set to 1. 0x0: off 0x1: on	RW	0
7:1	RESERVED		R	0x00
0	AUTOGATING	Internal Clock Gating Strategy Enables or disables auto clock gating.	RW	1

**NOTE:** This is only for debug purposes. When this bit is set to "0" autogating is not performed on any of the clocks, thus the clocks stay free running. Though, they are still controlled by the clock enable bit fields [RSZA\\_CLK\\_EN](#) and [RSZB\\_CLK\\_EN](#).

0x0: Clocks are free running  
0x1: Automatic clock gating strategy.

**Table 8-451. Register Call Summary for Register RSZ\_SYSCONFIG**

ISS ISP

- [ISS ISP RSZ Functional Description: \[0\]](#)
- [ISS ISP RSZ Initial Register Setup: \[1\] \[2\]](#)
- [ISS ISP RSZ Programming Constraints: \[3\]](#)
- [ISS RESIZER Register Summary: \[4\]](#)

**Table 8-452. RSZ\_IN\_FIFO\_CTRL**

<b>Address Offset</b>	0x0000 000C		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	INPUT DATA BUFFER CONTROL REGISTER This register is not shadowed		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				THRLD_LOW												RESERVED				THRLD_HIGH											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	THRLD_LOW	When <a href="#">RSZ_IN_FIFO_CTRL.THRLD_HIGH</a> = <a href="#">RSZ_IN_FIFO_CTRL.THRLD_LOW</a> , the <code>rsz_stall_input</code> is not asserted. The only purpose of the <a href="#">RSZ_IN_FIFO_CTRL.THRLD_LOW</a> register is to prevent <code>rsz_stall_input</code> signal assertion.	RW	0x0000
15:13	RESERVED		R	0x0
12:0	THRLD_HIGH	High threshold value. The <code>rsz_stall_input</code> signal is asserted if 2 lines of circular buffer are full and the third line has more pixels than <a href="#">RSZ_IN_FIFO_CTRL.THRLD_HIGH</a> . The <code>rsz_stall_input</code> signal stays high as long as one full line is not free for receiving further data. <code>THRLD_HIGH</code> is in terms of line size and can at max be programmed equal to the input line size ( <a href="#">RSZ_SRC_HSZ</a> ).	RW	0x0000

**Table 8-453. Register Call Summary for Register RSZ\_IN\_FIFO\_CTRL**

## ISS ISP

- [ISS ISP RSZ Functional Description: \[0\]](#)
- [ISS ISP RSZ Initial Register Setup: \[3\] \[4\]](#)
- [ISS ISP RSZ Programming Constraints: \[5\]](#)
- [ISS RESIZER Register Summary: \[6\]](#)
- [ISS RESIZER Register Description: \[7\] \[8\] \[9\] \[10\]](#)

**Table 8-454. RSZ\_GNC**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	GENERIC PARAMETER REGISTER		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RSZB_MEM_LINE_SIZE												RESERVED				RSZA_MEM_LINE_SIZE											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	RSZB_MEM_LINE_SIZE	Resizer #B memory line size (pixels). The output image cannot exceed this size.	R	0x0920

Bits	Field Name	Description	Type	Reset
15:13	RESERVED		R	0x0
12:0	RSZA_MEM_LINE_SIZE	Resizer #A memory line size (pixels). The output image cannot exceed this size.	R	0x1500

**Table 8-455. Register Call Summary for Register RSZ\_GNC**

ISS ISP

- [ISS ISP RSZ Operating Modes: \[0\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-456. RSZ\_FRACDIV**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	Fractional clock divider settings		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	RSZ_FRACDIV	Fractional clock divider value. The fractional clock divider gates the read requests made to the input data buffer such that the input data buffer is read at an average frequency equal to FFCLK instead of FCLK. The value of FFCLK depends upon the upscaling ratios as well as the input pixel clock: see the functional spec. We have $FFCLK = FCLK / FRACDIV$ MHz and $RSZ\_FRACDIV = 65536 / FRACDIV$ . When $RSZ\_FRACDIV = 65536$ , we have: $FFCLK = FCLK$ .	RW	0xFFFF

**Table 8-457. Register Call Summary for Register RSZ\_FRACDIV**

ISS ISP

- [ISS ISP RSZ Initial Register Setup: \[0\] \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[2\] \[3\]](#)
- [ISS RESIZER Register Summary: \[4\]](#)
- [ISS RESIZER Register Description: \[5\] \[6\]](#)

**Table 8-458. RSZ\_SRC\_EN**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER ENABLE REGISTER This register is not shadowed		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												Z			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	EN	Resizer module enable The start flag of the RESIZER module. When EN is set to 1, the RESIZER module starts the processing from the next rising edge of the VD pulse. If the processing mode of the RESIZER module is set to "one shot", the EN bit is cleared to 0 after the end of the processing. One has to pay attention that when this bit is enabled and  0x0: Disable 0x1: Enable	RW	0

**Table 8-459. Register Call Summary for Register RSZ\_SRC\_EN**

ISS ISP

- [ISS ISP RSZ Functional Description: \[0\]](#)
- [ISS ISP RSZ Initial Register Setup: \[1\]](#)
- [ISS ISP RSZ Reset Behavior: \[2\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[3\]](#)
- [ISS ISP RSZ Programming Constraints: \[4\]](#)
- [ISS RESIZER Register Summary: \[5\]](#)

**Table 8-460. RSZ\_SRC\_MODE**

<b>Address Offset</b>	0x0000 0024																														
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER																													
<b>Description</b>	This register is not shadowed																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WRT	OST														
Bits	Field Name	Description	Type	Reset																											
31:2	RESERVED		R	0x0000 0000																											
1	WRT	Video port WEN signal selection This bit selects whether the WEN signal which is present on the IPIPE and IPIPEIF video port is used or not to select the input data. If WRT is 0, the RESIZER module ignores the WEN signal and processes all image frame while RESIZER is enabled. If WRT is 1, the RESIZER module only processes the lines that arrived while the WEN is high. HD is used to sample the WEN signal.  0x0: Disable 0x1: Enable	RW	0																											
0	OST	The processing mode selection of the RESIZER module. Value 0 indicates the mode of free run, value 1 indicates the mode of one shot.  0x0: Free running 0x1: One shot	RW	0																											

**Table 8-461. Register Call Summary for Register RSZ\_SRC\_MODE**

ISS ISP

- [ISS ISP RSZ Video Port Interfaces: \[0\]](#)
- [ISS ISP RSZ Functional Description: \[1\]](#)
- [ISS ISP RSZ Initial Register Setup: \[2\] \[3\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[4\]](#)
- [ISS ISP RSZ Programming Constraints: \[5\]](#)
- [ISS RESIZER Register Summary: \[6\]](#)

**Table 8-462. RSZ\_SRC\_FMT0**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	This register is not shadowed		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BYPASS		SEL													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	BYPASS	Pass Through This bit enables or disables the RESIZER module pass through mode. The pass through mode can transfer images which are 8K pixel wide. When it is enabled, the input data buffer and the resizer engines are bypassed. 0x0: Pass through off = normal output mode, the input data buffer is used. 0x1: Pass through on = normal output mode, the input data buffer is bypassed.	RW	0
0	SEL	Input selection This bit selects which of the two video port is selected to push data through the RESIZER module. 0x0: IPIPE 0x1: IPIPEIF	RW	0

**Table 8-463. Register Call Summary for Register RSZ\_SRC\_FMT0**

ISS ISP

- [ISS ISP RSZ Functional Description: \[0\] \[1\] \[2\]](#)
- [ISS ISP RSZ Initial Register Setup: \[3\] \[4\]](#)
- [ISS ISP RSZ Programming Constraints: \[5\]](#)
- [ISS RESIZER Register Summary: \[6\]](#)
- [ISS RESIZER Register Description: \[7\] \[8\] \[9\] \[10\] \[11\]](#)

**Table 8-464. RSZ\_SRC\_FMT1**

<b>Address Offset</b>	0x0000 002C	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RW	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																COL	IN420	RAW													

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	COL	Y/C selection This bit is valid only if the input data is YUV4:2:0 (IN420 = 1). It enables to specify where the data which is input to the RESIZER module is luma or chroma data. 0x0: Y data is input 0x1: Chroma data is input	RW	0
1	IN420	Chroma Format Selection This bit sets the chroma undersampling when YUV data is input to the RESIZER module. 0x0: YUV4:2:2 is input 0x1: YUV4:2:0 is input	RW	0
0	RAW	Pass-through mode input data format selection This bit affects the horizontal reversal (flipping) process. 0x0: Flipping preserves YCbCr format 0x1: Flipping preserves RAW format	RW	0

**Table 8-465. Register Call Summary for Register RSZ\_SRC\_FMT1**

## ISS ISP

- [ISS ISP RSZ Functional Description: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[4\] \[5\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1: \[6\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2: \[7\]](#)
- [ISS RESIZER Register Summary: \[8\]](#)

**Table 8-466. RSZ\_SRC\_VPS**

<b>Address Offset</b>	0x0000 0030	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	VERTICAL POSITION REGISTER This register is not shadowed	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VPS															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	VPS	Vertical Start Position Sets the vertical position of the global frame from the rising edge of the VD. The RSZ module will start the image processing from the VPS'th line. This value can be odd or even whatever the input data format.	RW	0x0000

**Table 8-467. Register Call Summary for Register RSZ\_SRC\_VPS**

ISS ISP

- [ISS ISP RSZ Input Data Cropper: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[1\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[2\]](#)
- [ISS ISP RSZ Programming Constraints: \[3\]](#)
- [ISS RESIZER Register Summary: \[4\]](#)

**Table 8-468. RSZ\_SRC\_VSZ**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	VERTICAL SIZER REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VSZ															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VSZ	Vertical Processing Size Sets the vertical size of the processing area. The RSZ module will process (VSZ+1) lines. This value can be odd or even whatever the input data format.	RW	0x0000

**Table 8-469. Register Call Summary for Register RSZ\_SRC\_VSZ**

ISS ISP

- [ISS ISP RSZ Input Data Cropper: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[1\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-470. RSZ\_SRC\_HPS**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	HORIZONTAL POSITION REGISTER This register is not shadowed		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HPS															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	HPS	<p>Horizontal Start Position</p> <p>The <a href="#">RSZ_SRC_HPS</a> register has two functions:</p> <p>The first function is to compensate for possible delay between the HD pulse and the first valid data. It is possible for this delay to be different than 0 when the RESIZER module gets its input data from the VP connected to the IPIPEIF module (the offset value can be odd or even). When data are coming from the IPIPE module, it is not required to resynchronize HD and the first valid data.</p> <p>The second function is to crop the data in the horizontal direction. When used for cropping, only <a href="#">RSZ_SRC_HPS</a> must be even or null.</p>	RW	0x0000

**Table 8-471. Register Call Summary for Register RSZ\_SRC\_HPS**

## ISS ISP

- [ISS ISP RSZ Input Data Cropper: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[1\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[2\]](#)
- [ISS ISP RSZ Programming Constraints: \[3\]](#)
- [ISS RESIZER Register Summary: \[4\]](#)
- [ISS RESIZER Register Description: \[5\] \[6\]](#)

**Table 8-472. RSZ\_SRC\_HSZ**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	HORIZONTAL SIZE REGISTER The HSZ value is given by HSZ concatenated with HSZ_LSB		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSZ															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	HSZ	<p>Horizontal size</p> <p>Sets the horizontal size of the processing area. The RSZ module processes (HSZ+1) pixels. (HSZ+1) must be even for YUV4:2:2 and RAW data. The valid available values for HSZ are 1~xxxx.</p>	RW	0x0000

**Table 8-473. Register Call Summary for Register RSZ\_SRC\_HSZ**

## ISS ISP

- [ISS ISP RSZ Input Data Cropper: \[0\] \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[2\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[3\]](#)
- [ISS RESIZER Register Summary: \[4\]](#)
- [ISS RESIZER Register Description: \[5\]](#)



**Table 8-474. RSZ\_DMA\_RZA**

<b>Address Offset</b>	0x0000 0040	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER A - MEMORY REQUEST MINIMUM INTERVAL REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RZA															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	RZA	Sets the minimum interval btw two consecutive memory request for resizer #A. Specified in number of interface clock cycles. Values of 0, 1, and 2 are used as a condition to keep the bandwidth limiter off. When this function is enabled, this value must be greater than 10 cycles.	RW	0x0000

**Table 8-475. Register Call Summary for Register RSZ\_DMA\_RZA**

ISS ISP

- [ISS ISP RSZ MTC interfaces: \[0\]](#)
- [ISS ISP BL Peak Memory Bandwidth Reduction: \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-476. RSZ\_DMA\_RZB**

<b>Address Offset</b>	0x0000 0044	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - MEMORY REQUEST MINIMUM INTERVAL REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RZB															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	RZB	Sets the minimum interval btw two consecutive memory request for resizer #B. Specified in number of interface clock cycles. Values of 0, 1, and 2 are used as a condition to keep the bandwidth limiter off. When this function is enabled, this value must be greater than 10 cycles.	RW	0x0000

**Table 8-477. Register Call Summary for Register RSZ\_DMA\_RZB**

ISS ISP

- [ISS ISP RSZ MTC interfaces: \[0\]](#)
- [ISS ISP BL Peak Memory Bandwidth Reduction: \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-478. RSZ\_DMA\_STA**

<b>Address Offset</b>	0x0000 0048		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER STATUS REGISTER		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STATUS															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STATUS	Resizer process status This bit is set in the time window from rsz_int_reg to rsz_int_dma. Read 0x1: Active Read 0x0: Not active	R	0

**Table 8-479. Register Call Summary for Register RSZ\_DMA\_STA**

ISS ISP

- [ISS ISP RSZ MTC interfaces: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-480. RSZ\_GCK\_MMR**

<b>Address Offset</b>	0x0000 004C		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	MMR CLOCK CONTROL REGISTER This register is not shadowed		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MMR															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MMR	The on/off selection of the MMR interface clock which is used for MMR register access. 0x0: Off 0x1: On	RW	0

**Table 8-481. Register Call Summary for Register RSZ\_GCK\_MMR**

ISS ISP

- [ISS ISP RSZ VBUSP interface: \[0\]](#)
- [ISS ISP RSZ Initial Register Setup: \[1\]](#)
- [ISS ISP RSZ Programming Constraints: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-482. RSZ\_GCK\_SDR**

<b>Address Offset</b>	0x0000 0054		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	CORE CLOCK CONTROL REGISTER This register is not shadowed		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												CORE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CORE	<p>RSZ Core Clock Enable. This bit enables or disables the resizer core functional clock. When this bit is off, the resizer core (interpolator) is automatically bypassed (resizer-bypass mode of pass-through mode is selected depending on <a href="#">RSZ_SRC_FMT0</a> value). In resizer-bypass mode or pass-through mode, no up-scaling or downscaling process is operated.</p> <p>0x0: Resizer core clock disabled. Resizer in bypass mode if <a href="#">RSZ_SRC_FMT0.BYPASS</a> = 0 Resizer in pass-through if <a href="#">RSZ_SRC_FMT0.BYPASS</a> = 1</p> <p>0x1: Resizer core clock enabled. Resizer in rescaling mode if <a href="#">RSZ_SRC_FMT0.BYPASS</a> = 0 Resizer in pass-through if <a href="#">RSZ_SRC_FMT0.BYPASS</a> = 1</p>	RW	0

**Table 8-483. Register Call Summary for Register RSZ\_GCK\_SDR**

ISS ISP

- [ISS ISP RSZ Functional Description: \[0\] \[1\]](#)
- [ISS ISP RSZ Initial Register Setup: \[2\] \[3\] \[4\] \[5\]](#)
- [ISS ISP RSZ Programming Constraints: \[6\]](#)
- [ISS RESIZER Register Summary: \[7\]](#)
- [ISS RESIZER Register Description: \[8\] \[9\]](#)

**Table 8-484. RSZ\_IRQ\_RZA**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER A - CIRCULAR BUFFER INTERRUPT INTERVAL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RZA															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	RZA	<p>Resizer A circular buffer interval Sets the circular buffer interval for Resizer A. The interrupt is triggered every time (RZA+1) lines are written to the circular buffer (Y buffer). The range goes from 1 to 8192 lines. Usually, the circular buffer vertical size should be a multiple of RZA.</p>	RW	0x0000

**Table 8-485. Register Call Summary for Register RSZ\_IRQ\_RZA**

ISS ISP

- [ISS ISP RSZ Interrupts: \[0\] \[1\] \[2\]](#)
- [ISS ISP RSZ Initial Register Setup: \[3\]](#)
- [ISS ISP5 SYS1 Register Description: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [ISS RESIZER Register Summary: \[12\]](#)

**Table 8-486. RSZ\_IRQ\_RZB**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B - CIRCULAR BUFFER INTERRUPT INTERVAL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RZB															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	RZB	Resizer B circular buffer interval Sets the circular buffer interval for Resizer B. The interrupt is triggered every time (RZB+1) lines are written to the circular buffer (Y buffer). The range goes from 1 to 8192 lines. Usually, the circular buffer vertical size should be a multiple of RZB.	RW	0x0000

**Table 8-487. Register Call Summary for Register RSZ\_IRQ\_RZB**

ISS ISP

- [ISS ISP RSZ Interrupts: \[0\]](#)
- [ISS ISP RSZ Initial Register Setup: \[1\]](#)
- [ISS ISP5 SYS1 Register Description: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [ISS RESIZER Register Summary: \[10\]](#)

**Table 8-488. RSZ\_YUV\_Y\_MIN**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	LUMINANCE SATURATION REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MIN															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	MIN	The minimum value of Luminance (8bits unsigned). If the value of the Luminance is smaller than VAL, it will be clipped to VAL. This bit field must be set to its default values when the resizer is set in pass-through mode.	RW	0x00

**Table 8-489. Register Call Summary for Register RSZ\_YUV\_Y\_MIN**

ISS ISP

- [ISS ISP RSZ Data Saturator: \[0\] \[1\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-490. RSZ\_YUV\_Y\_MAX**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	LUMINANCE SATURATION REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MAX															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	MAX	The maximum value of Luminance (8bits unsigned). If the value of the Luminance is larger than VAL, it will be clipped to VAL. This bit field must be set to its default values when the resizer is set in pass-through mode.	RW	0xFF

**Table 8-491. Register Call Summary for Register RSZ\_YUV\_Y\_MAX**

ISS ISP

- [ISS ISP RSZ Data Saturator: \[0\] \[1\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-492. RSZ\_YUV\_C\_MIN**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	CHROMINANCE SATURATION REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MIN															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	MIN	The minimum value of Chrominance (8bits unsigned). If the value of the Chrominance is smaller than VAL, it will be clipped to VAL. This bit field must be set to its default values when the resizer is set in pass-through mode.	RW	0x00

**Table 8-493. Register Call Summary for Register RSZ\_YUV\_C\_MIN**

ISS ISP

- [ISS ISP RSZ Data Saturator: \[0\] \[1\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-494. RSZ\_YUV\_C\_MAX**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	CHROMINANCE SATURATION REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MAX															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	MAX	The maximum value of Chrominance (8bits unsigned). If the value of the Chrominance is larger than VAL, it will be clipped to VAL. This bit field must be set to its default values when the resizer is set in pass-through mode.	RW	0xFF

**Table 8-495. Register Call Summary for Register RSZ\_YUV\_C\_MAX**

ISS ISP

- [ISS ISP RSZ Data Saturator: \[0\] \[1\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-496. RSZ\_YUV\_PHS**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	The phase position of the output of the Chrominance		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																POS															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	POS	The phase position of the output of the chrominance. The RESIZER module does not change the relative position of the chroma samples vs. the luma samples between the input and output and the chroma position at the output of the IPIPE module and at the output of the RESIZER module must be identical. In other words, we must have <a href="#">RSZ_YUV_PHS.POS</a> = <a href="#">IPIPE_YUV_PHS.POS</a> .  0x0: Same position with Luminance: cosited 0x1: The middle of the luminance: centered	RW	0

**Table 8-497. Register Call Summary for Register RSZ\_YUV\_PHS**

ISS ISP

- [ISS ISP RSZ Functional Description: \[0\] \[1\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)
- [ISS RESIZER Register Description: \[4\]](#)
- [ISS IPIPE Register Description: \[5\]](#)

**Table 8-498. RSZ\_SEQ**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																												CRV	VRVB	HRVB	VRVA	HRVA

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x00000000
4	CRV	Chroma sampling point change 0x0: Chroma sampling point is not changed 0x1: Chroma sampling point is changed from odd-numbered pixels to even-number pixels. The pixel at the left end is removed and the pixel at the right end is duplicated.	RW	0
3	VRVB	Resizer B - Vertical reversal of output image 0x0: Processed pixels are output in the order of input (normal operation) in vertical direction. 0x1: The order of output data is flipped top to bottom.	RW	0
2	HRVB	Resizer B -Horizontal reversal of output image 0x0: Processed pixels are output in the order of input (normal operation) in horizontal direction. 0x1: The order of output data is flipped left to right.	RW	0
1	VRVA	Resizer A - Vertical reversal of output image 0x0: Processed pixels are output in the order of input (normal operation) in vertical direction. 0x1: The order of output data is flipped top to bottom.	RW	0
0	HRVA	Resizer A - Horizontal reversal of output image 0x0: Processed pixels are output in the order of input (normal operation) in horizontal direction. 0x1: The order of output data is flipped left to right.	RW	0

**Table 8-499. Register Call Summary for Register RSZ\_SEQ**

ISS ISP

- [ISS ISP RSZ Functional Description: \[0\] \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[2\] \[3\] \[4\] \[5\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 2: \[6\] \[7\]](#)
- [ISS RESIZER Register Summary: \[8\]](#)

**Table 8-500. RZA\_EN**

<b>Address Offset</b>	0x0000 0078		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER A - ENABLE REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												EN			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	EN	Enable resizer #A This bit is latched on video port VD input. The reason is that the resizer must only start the processing on a clean frame boundary. In one-shot mode, this bit is negated on VD.  0x0: Disable 0x1: Enable	RW	0

**Table 8-501. Register Call Summary for Register RZA\_EN**

ISS ISP

- [ISS ISP RSZ Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP RSZ Initial Register Setup: \[5\]](#)
- [ISS RESIZER Register Summary: \[6\]](#)

**Table 8-502. RZA\_MODE**

<b>Address Offset</b>	0x0000 007C		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER #A MODE REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MODE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MODE	Select "Free Run mode" or "One Shot Mode"  0x0: Free run 0x1: One shot	RW	0

**Table 8-503. Register Call Summary for Register RZA\_MODE**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)



**Table 8-504. RZA\_420**

<b>Address Offset</b>	0x0000 0080		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	YEN/CEN: 0/0: in = YUV4:2:2 input, out = YUV4:2:2 output 0/1: in = YUV4:2:2 input, out = Chrominance of YUV4:2:0 output 1/0: in = YUV4:2:2 input, out = Luminance of YUV4:2:0 output 1/1: in = YUV4:2:2 input, out = YUV4:2:0 output		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											CEN	YEN			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	CEN	Output Enable for Luminance This bit is valid in 422 input mode. When CEN=0 and YEN=0, output is 422  0x0: C output disable 0x1: C output enable and 422to420 conversion enabled	RW	0
0	YEN	Output Enable for Chrominance. This bit is valid in 422 input mode. When CEN=0 and YEN=0, output is 422  0x0: Y output disable 0x1: Y output enable and 422to420 conversion enabled	RW	0

**Table 8-505. Register Call Summary for Register RZA\_420**

ISS ISP

- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[0\] \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 2: \[2\] \[3\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[4\] \[5\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1: \[6\] \[7\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2: \[8\] \[9\]](#)
- [ISS RESIZER Register Summary: \[10\]](#)
- [ISS RESIZER Register Description: \[11\] \[12\]](#)

**Table 8-506. RZA\_I\_VPS**

<b>Address Offset</b>	0x0000 0084		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER A - INPUT VERTICAL START REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VPS															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VPS	Input Vertical Position Sets the vertical start position of the input image within the global frame. It enables to crop data into the global frame. After SRC_VPS, the Vps'th line is processed as the first line in each image.	RW	0x0000

**Table 8-507. Register Call Summary for Register RZA\_I\_VPS**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-508. RZA\_I\_HPS**

<b>Address Offset</b>	0x0000 0088	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER A - INPUT HORIZONTAL START REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HPS															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	HPS	Input Horizontal Position Sets the horizontal position of the first pixel for each line within the global frame. After SRC_HPS, the pixel at the VAL'th position is processed as the first pixel. This value must be even.	RW	0x0000

**Table 8-509. Register Call Summary for Register RZA\_I\_HPS**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-510. RZA\_O\_VSZ**

<b>Address Offset</b>	0x0000 008C	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER A - OUTPUT VERTICAL SIZER REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VSZ															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VSZ	The target output size of the resized image. The number of output lines is (VSZ+1). Set 479, when 480 lines of output is required.	RW	0x0000

**Table 8-511. Register Call Summary for Register RZA\_O\_VSZ**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-512. RZA\_O\_HSZ**

<b>Address Offset</b>	0x0000 0090		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER A - OUTPUT HORIZONTAL SIZE REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSZ												HSZ_LSB			

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:1	HSZ	The horizontal size of output image. The number of pixel in each line is (HSZ+1). Set 479, when 480 pixels are required. This value must be lower than the max memory line size supported by the resizer engine, except in RAW pass through mode. Note that the LSB of the 13-bit HSZ value is fixed to 1 such that the horizontal size is always even.	RW	0x000
0	HSZ_LSB	The least significant bit of HSZ is forced to 1.	R	1

**Table 8-513. Register Call Summary for Register RZA\_O\_HSZ**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-514. RZA\_V\_PHS\_Y**

<b>Address Offset</b>	0x0000 0094		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER A - INITIAL LUMINANCE PHASE OF VERTICAL RESIZING PROCESS When YUV4:2:2 data are output, the phase values for luma and chroma should typically be equal, that is, RZX_V_PHS_Y= RZX_V_PHS_C. The following constraints apply when setting the initial vertical phases ABS(RZX_V_PHS_Y - RZX_V_PHS_C) RZX_V_DIF. This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two luma pixels. Note that the absolute value is used, hence, the initial luma phase can be greater than the initial chroma phase or the other way around. As a reminder, the distance between two output pixels for luma is given by RZX_V_DIF.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	Y	The initial value for the luma phase in vertical resizing process. This value is in U14Q8 fractional format.	RW	0x0000

**Table 8-515. Register Call Summary for Register RZA\_V\_PHS\_Y**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-516. RZA\_V\_PHS\_C**

<b>Address Offset</b>	0x0000 0098	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - INITIAL CHROMINANCE PHASE OF VERTICAL RESIZING PROCESS When YUV4:2:2 data are output, the phase values for luma and chroma should typically be equal, that is, RZX_V_PHS_Y= RZX_V_PHS_C. The following constraints apply when setting the initial vertical phases ABS(RZX_V_PHS_Y - RZX_V_PHS_C) RZX_V_DIF. This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two luma pixels. Note that the absolute value is used, hence, the initial luma phase can be greater than the initial chroma phase or the other way around. As a reminder, the distance between two output pixels for luma is given by RZX_V_DIF.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	C	The initial value for the chroma phase in vertical resizing process. This value is in U14Q8 fractional format.	RW	0x0000

**Table 8-517. Register Call Summary for Register RZA\_V\_PHS\_C**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-518. RZA\_V\_DIF**

<b>Address Offset</b>	0x0000 009C	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - VERTICAL RESIZER REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																V															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	V	The parameter for vertical resize. The actual resizing ratio is 256/RZA_V_DIF. In normal mode: 16 <= RZA_V_DIF <= 4096. In down-scale mode: 256 <= RZA_V_DIF <= 4096.	RW	0x0000

**Table 8-519. Register Call Summary for Register RZA\_V\_DIF**

- ISS ISP
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
  - [ISS RESIZER Register Summary: \[1\]](#)
  - [ISS RESIZER Register Description: \[2\] \[3\] \[4\]](#)

**Table 8-520. RZA\_V\_TYP**

<b>Address Offset</b>	0x0000 00A0	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - INTERPOLATION METHOD FOR VERTICAL RESIZING		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											C	Y			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	C	Selection of resizing method for chrominance: vertical 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0
0	Y	Selection of resizing method for luminance: vertical 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0

**Table 8-521. Register Call Summary for Register RZA\_V\_TYP**

- ISS ISP
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
  - [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-522. RZA\_V\_LPF**

<b>Address Offset</b>	0x0000 00A4	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - VERTICAL LPF INTENSITY REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					C			Y							

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x000000
11:6	C	The intensity parameter for chroma vertical low pass filtering.	RW	0x00
5:0	Y	The intensity parameter for luma vertical low pass filtering.	RW	0x00

**Table 8-523. Register Call Summary for Register RZA\_V\_LPF**

- ISS ISP
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
  - [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-524. RZA\_H\_PHS**

<b>Address Offset</b>	0x0000 00A8	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - INITIAL PHASE OF HORIZONTAL RESIZING PROCESS		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PHS															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	PHS	Initial value for the phase in horizontal resizing process, that is, the sampling position is shifted. This value is in U14Q8 fractional format. Example: If RZX_H_PHS = 128, the first output pixel is sampled at the center of the first two valid input pixels. If RZX_I_HPS=100 and RZX_H_PHS=128, the first output pixel is resampled at the center of the 100-th and the 101-st input pixels.	RW	0x0000

**Table 8-525. Register Call Summary for Register RZA\_H\_PHS**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-526. RZA\_H\_PHS\_ADJ**

<b>Address Offset</b>	0x0000 00AC	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - LUMINANCE HORIZONTAL PHASE ADJUSTMENT The <a href="#">RZA_H_PHS_ADJ</a> register enables to adjust the horizontal phase for the luma component when averaging is enabled (the horizontal averaging disrupts the relative sampling point between luminance and chrominance when YUV4:2:2 cosited data is input), that is, the relative phase between luma and chroma is different before and after the horizontal averager.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADJ															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0000000
8:0	ADJ	Horizontal phase adjustment value. This value is in U9Q8 fractional format. This value is expected to be equal to zero if the averager is disabled or if input chroma is centered.	RW	0x000

**Table 8-527. Register Call Summary for Register RZA\_H\_PHS\_ADJ**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)
- [ISS RESIZER Register Description: \[2\] \[3\]](#)
- [ISS IPIPE Register Description: \[4\]](#)

**Table 8-528. RZA\_H\_DIF**

<b>Address Offset</b>	0x0000 00B0	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - HORIZONTAL RESIZER REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																H															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	H	The parameter for horizontal resizing process. The actual resizing ratio is 256/VAL. In normal mode 16<= RSZ_RZA_H_DIF<=4096 In down-scale mode 256<=RSZ_RZA_H_DIF<=4096	RW	0x0000

**Table 8-529. Register Call Summary for Register RZA\_H\_DIF**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)
- [ISS RESIZER Register Description: \[2\] \[3\] \[4\] \[5\]](#)

**Table 8-530. RZA\_H\_TYP**

<b>Address Offset</b>	0x0000 00B4	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	Resize-A		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C	Y														

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	C	Selection of resizing method for chrominance: horizontal 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0
0	Y	Selection of resizing method for luminance: horizontal 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0

**Table 8-531. Register Call Summary for Register RZA\_H\_TYP**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-532. RZA\_H\_LPF**

<b>Address Offset</b>	0x0000 00B8		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER A - HORIZONTAL LPF INTENSITY REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C								Y							

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:6	C	Horizontal LPF Intensity for Chrominance	RW	0x00
5:0	Y	Selection of resizing method for Luminance in horizontal direction	RW	0x00

**Table 8-533. Register Call Summary for Register RZA\_H\_LPF**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-534. RZA\_DWN\_EN**

<b>Address Offset</b>	0x0000 00BC		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER #A - DOWNSCALE ENABLE REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DWN_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	DWN_EN	Resizer downscale enable 0x0: Off. Normal operation: upscale and downscale are allowed. 0x1: On. Downscale mode.	RW	0

**Table 8-535. Register Call Summary for Register RZA\_DWN\_EN**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)



**Table 8-536. RZA\_DWN\_AV**

<b>Address Offset</b>	0x0000 00C0	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	Resize-A	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							V	H							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5:3	V	Vertical averaging size : $1/2^{(VWT+1)}$ The range goes from 1/2 to 1/256 in power of two. 0: _DIV2 1/2 down scale 1: _DIV4 1/4 down scale 2: _DIV8 1/8 down scale 3: _DIV16 1/16 down scale 4: _DIV32 1/32 down scale 5: _DIV64 1/64 down scale 6: _DIV128 1/128 down scale 7: _DIV256 1/256 down scale	RW	0x0
2:0	H	Horizontal averaging size : $1/2^{(HWT+1)}$ The range goes from 1/2 to 1/256 in power of two. 0: _DIV2 1/2 down scale 1: _DIV4 1/4 down scale 2: _DIV8 1/8 down scale 3: _DIV16 1/16 down scale 4: _DIV32 1/32 down scale 5: _DIV64 1/64 down scale 6: _DIV128 1/128 down scale 7: _DIV256 1/256 down scale	RW	0x0

**Table 8-537. Register Call Summary for Register RZA\_DWN\_AV**

ISS ISP

- [ISS ISP RSZ Border Conditions](#): [0] [1]
- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 1](#): [2] [3]
- [ISS RESIZER Register Summary](#): [4]

**Table 8-538. RZA\_RGB\_EN**

<b>Address Offset</b>	0x0000 00C4	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER #A - RGB OUTPUT ENABLE	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RGB_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	RGB_EN	Enable of RGB output In pass through mode, this register must be 0. This bit can only be set to 1 when YUV4:2:2 data are output. YUV4:2:2 data output is selected when SRC_FMT1.IN420 = 0 and RZA_420.YEN = RZA_420.CEN = 0  0x0: Off (YCbCr output) 0x1: On (RGB output)	RW	0

**Table 8-539. Register Call Summary for Register RZA\_RGB\_EN**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-540. RZA\_RGB\_TYP**

<b>Address Offset</b>	0x0000 00C8	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - RGB OUTPUT CONTROL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MSK1	MSK0	TYP													

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	MSK1	Enables masking of the last 2 pixels This bit is used to mask the 2 last pixels at the image boundary which are affected by the YUV4:2:2 to YUV4:4:4 conversion.  0x0: output the last 2 pixels 0x1: mask the last 2 pixels (Resizer do not output them.)	RW	0
1	MSK0	Enables masking of the first 2 pixels This bit is used to mask the 2 first pixels at the image boundary which are affected by the YUV4:2:2 to YUV4:4:4 conversion.  0x0: output the first 2 pixels 0x1: mask the first 2 pixels (Resizer do not output them.)	RW	0
0	TYP	16bit/32bit output selection  0x0: 32-bit output: alpha + R + G + B (8 bit each) This mode comes with performance degradation. The maximum input frequency in this mode is 160 MHz. This due to the fact that the output is 4 bytes / pixel. 0x1: 16-bit output: R(5 bit) + G (6 bit) + B (5 bit)	RW	0

**Table 8-541. Register Call Summary for Register RZA\_RGB\_TYP**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2: \[0\] \[1\] \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-542. RZA\_RGB\_BLD**

<b>Address Offset</b>	0x0000 00CC	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER A - RGB BLEND REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BLD															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	BLD	The alpha value used in 32-bit RGBA output mode	RW	0x00

**Table 8-543. Register Call Summary for Register RZA\_RGB\_BLD**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings – Subsequence 2: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-544. RZA\_SDR\_Y\_BAD\_H**

<b>Address Offset</b>	0x0000 00D0	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER A - OUTPUT MEMORY BASE ADDRESS REGISTER (HIGH) This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_BAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_BAD_H	Memory Base Address Sets the 16 upper bits of the 32-bit base address of the circular buffer in memory.	RW	0x0000

**Table 8-545. Register Call Summary for Register RZA\_SDR\_Y\_BAD\_H**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-546. RZA\_SDR\_Y\_BAD\_L**

<b>Address Offset</b>	0x0000 00D4		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER A - OUTPUT MEMORY BASE ADDRESS REGISTER (LOW) This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_BAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_BAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit base address of the circular buffer in memory. It is a byte address. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RAW, RGB5:6:5 and YUV4:2:2 formats (output data on 16 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RGBA format (output data on 32 bits): The three least significant bits must be set to "000" when horizontal reversal mode is off. The three least significant bits must be set to "111" when horizontal reversal mode is on.	RW	0x0000

**Table 8-547. Register Call Summary for Register RZA\_SDR\_Y\_BAD\_L**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-548. RZA\_SDR\_Y\_SAD\_H**

<b>Address Offset</b>	0x0000 00D8		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER (HIGH) This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_SAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_SAD_H	Memory Start Address Sets the 16 upper bits of the 32-bit start address in memory.	RW	0x0000

**Table 8-549. Register Call Summary for Register RZA\_SDR\_Y\_SAD\_H**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-550. RZA\_SDR\_Y\_SAD\_L**

<b>Address Offset</b>	0x0000 00DC		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER (LOW) This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_SAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_SAD_L	<p>Memory Start Address</p> <p>Sets 16 lower bits of the 32-bit start address in memory. It is a byte address. For every frame, the first line of data will be written to this address.</p> <p>We have: <math>SAD = BAD + (PTR\_S \times OFT)</math> and <math>PTR\_S &lt; PTR\_E</math></p> <p>If the first line must be written at the beginning of the circular buffer memory then <math>SAD = BAD</math> and <math>PTR\_S = 0</math>.</p> <p>YUV4:2:0 format (output data on 8 bits):</p> <p>The two least significant bits must be set to 00 when horizontal reversal mode is off.</p> <p>The two least significant bits must be set to 11 when horizontal reversal mode is on.</p> <p>RAW, RGB5:6:5 and YUV4:2:2 formats (output data on 16 bits):</p> <p>The two least significant bits must be set to 00 when horizontal reversal mode is off.</p> <p>The two least significant bits must be set to 11 when horizontal reversal mode is on.</p> <p>RGBA format (output data on 32 bits):</p> <p>The three least significant bits must be set to "000" when horizontal reversal mode is off.</p> <p>The three least significant bits must be set to "111" when horizontal reversal mode is on.</p>	RW	0x0000

**Table 8-551. Register Call Summary for Register RZA\_SDR\_Y\_SAD\_L**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 2: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-552. RZA\_SDR\_Y\_OFT**

<b>Address Offset</b>	0x0000 00E0	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY OFFSET REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_OFT															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:0	Y_OFT	<p>Memory Line Offset</p> <p>Sets the size of each line in the circular buffer. It is expressed in bytes and unsigned. Note that OFT does not necessary corresponds to the size of a line in a frame, it can be much bigger. The line offset must be a multiple of 128 bytes (bits [6:0] of RZX_SDR_Y_OFT and RZX_SDR_C_OFT must be set to 0).</p> <p>Example:</p> <p>line 0 address = SAD</p> <p>line 1 address = SAD + 1 x OFT</p> <p>line 2 address = SAD + 2 x OFT</p>	RW	0x00000

**Table 8-553. Register Call Summary for Register RZA\_SDR\_Y\_OFT**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-554. RZA\_SDR\_Y\_PTR\_S**

<b>Address Offset</b>	0x0000 00E4	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_PTR_S															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	Y_PTR_S	Start Line of Memory Pointer Sets the vertical position of the first output line in the output memory space. This value is expressed in number of lines. The hardware uses it to set up the initial value of the circular buffer. It must be set up such as PTR_S = (SAD - BAD) / OFT. This value must be set to 0 when RSZ_RZA_SDR_Y_BAD = RSZ_RZA_SDR_Y_SAD.	RW	0x0000

**Table 8-555. Register Call Summary for Register RZA\_SDR\_Y\_PTR\_S**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-556. RZA\_SDR\_Y\_PTR\_E**

<b>Address Offset</b>	0x0000 00E8	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY END ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_PTR_E															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	Y_PTR_E	End Line of Memory Pointer Sets the maximum number of lines to be stored in the output memory space. This value is expressed in number of lines. When the number of output lines exceeds this value, the address restarts from the first address in the output memory space (BAD).	RW	0x0000

**Table 8-557. Register Call Summary for Register RZA\_SDR\_Y\_PTR\_E**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS ISP5 SYS1 Register Description: \[2\] \[3\] \[4\] \[5\]](#)
- [ISS RESIZER Register Summary: \[6\]](#)

**Table 8-558. RZA\_SDR\_C\_BAD\_H**

<b>Address Offset</b>	0x0000 00EC	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY BASE ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_BAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_BAD_H	Memory Base Address Sets the 16 higher bits of the 32-bit base address of the circular buffer in memory.	RW	0x0000

**Table 8-559. Register Call Summary for Register RZA\_SDR\_C\_BAD\_H**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-560. RZA\_SDR\_C\_BAD\_L**

<b>Address Offset</b>	0x0000 00F0	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY BASE ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_BAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_BAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit base address of the circular buffer in memory. It is a byte address. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on.	RW	0x0000



**Table 8-561. Register Call Summary for Register RZA\_SDR\_C\_BAD\_L**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-562. RZA\_SDR\_C\_SAD\_H**

<b>Address Offset</b>	0x0000 00F4	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_SAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_SAD_H	Memory Base Address Sets the 16 higher bits of the 32-bit start address in memory.	RW	0x0000

**Table 8-563. Register Call Summary for Register RZA\_SDR\_C\_SAD\_H**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-564. RZA\_SDR\_C\_SAD\_L**

<b>Address Offset</b>	0x0000 00F8	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_SAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_SAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit start address in memory. It is a byte address. For every frame, the first line of data will be written to this address. We have: SAD = BAD + (PTR_S x OFT) and PTR_S < PTR_E If the first line must be written at the beginning of the circular buffer memory then SAD = BAD and PTR_S = 0. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on.	RW	0x0000

**Table 8-565. Register Call Summary for Register RZA\_SDR\_C\_SAD\_L**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 2: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-566. RZA\_SDR\_C\_OFT**

<b>Address Offset</b>	0x0000 00FC	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY OFFSET REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_OFT															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:0	C_OFT	Memory Line Offset Sets the size of each line in the circular buffer. It is expressed in bytes and unsigned. Note that OFT does not necessary corresponds to the size of a line in a frame, it can be much bigger. The line offset must be a multiple of 128 bytes (bits [6:0] of RZX_SDR_Y_OFT and RZX_SDR_C_OFT must be set to 0). Example: line 0 address = SAD line 1 address = SAD + 1 x OFT line 2 address = SAD + 2 x OFT	RW	0x00000

**Table 8-567. Register Call Summary for Register RZA\_SDR\_C\_OFT**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-568. RZA\_SDR\_C\_PTR\_S**

<b>Address Offset</b>	0x0000 0100	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_PTR_S															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	C_PTR_S	Start Line of Memory Pointer Sets the vertical position of the first output line in the output memory space. This value is expressed in number of lines. The hardware uses it to set up the initial value of the circular buffer. It must be set up such as PTR_S = (SAD - BAD) / OFT. This value must be set to 0 when RSZ_RZA_SDR_C_BAD = RSZ_RZA_SDR_C_SAD.	RW	0x0000

**Table 8-569. Register Call Summary for Register RZA\_SDR\_C\_PTR\_S**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-570. RZA\_SDR\_C\_PTR\_E**

<b>Address Offset</b>	0x0000 0104	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER A - OUTPUT MEMORY END ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_PTR_E															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	C_PTR_E	End Line of Memory Pointer Sets the maximum number of lines to be stored in the output memory space. This value is expressed in number of lines. When the number of output lines exceeds this value, the address restarts from the first address in the output memory space (BAD).	RW	0x0000

**Table 8-571. Register Call Summary for Register RZA\_SDR\_C\_PTR\_E**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-572. RZB\_EN**

<b>Address Offset</b>	0x0000 0108	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B - ENABLE REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Z															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	EN	Enable resizer #A This bit is latched on the video port VD input signal. The reason is that the resizer must only starts the processing on a clean frame boundary. In one-shot mode, this bit is negated on VD.  0x0: Disable 0x1: Enable	RW	0

**Table 8-573. Register Call Summary for Register RZB\_EN**

ISS ISP

- [ISS ISP RSZ Functional Description: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP RSZ Initial Register Setup: \[5\]](#)
- [ISS RESIZER Register Summary: \[6\]](#)

**Table 8-574. RZB\_MODE**

<b>Address Offset</b>	0x0000 010C	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B MODE REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MODE	Select "Free Run mode" or "One Shot Mode"  0x0: Free run 0x1: One shot	RW	0

**Table 8-575. Register Call Summary for Register RZB\_MODE**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-576. RZB\_420**

<b>Address Offset</b>	0x0000 0110	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	YEN/CEN: 0/0: in = YUV4:2:2 input, out = YUV4:2:2 output 0/1: in = YUV4:2:2 input, out = Chrominance of YUV4:2:0 output 1/0: in = YUV4:2:2 input, out = Luminance of YUV4:2:0 output 1/1: in = YUV4:2:2 input, out = YUV4:2:0 output		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CEN	YEN														

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	CEN	Output Enable for Luminance This bit is valid in 422 input mode. When CEN=0 and YEN=0, output is 422  0x0: C output disable 0x1: C output enable and 422to420 conversion enabled	RW	0
0	YEN	Output Enable for Chrominance. This bit is valid in 422 input mode. When CEN=0 and YEN=0, output is 422  0x0: Y output disable 0x1: Y output enable and 422to420 conversion enabled	RW	0

**Table 8-577. Register Call Summary for Register RZB\_420**

ISS ISP

- [ISS ISP RSZ Global Image Processing Settings – Subsequence 1: \[0\] \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 2: \[2\] \[3\]](#)
- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[4\] \[5\]](#)
- [ISS RESIZER Register Summary: \[6\]](#)
- [ISS RESIZER Register Description: \[7\] \[8\]](#)

**Table 8-578. RZB\_I\_VPS**

<b>Address Offset</b>	0x0000 0114	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B - INPUT VERTICAL START REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VPS															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VPS	Input Vertical Position Sets the vertical start position of the input image within the global frame. It enables to crop data into the global frame. After SRC_VPS, the Vps'th line is processed as the first line in each image.	RW	0x0000

**Table 8-579. Register Call Summary for Register RZB\_I\_VPS**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-580. RZB\_I\_HPS**

<b>Address Offset</b>	0x0000 0118	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - INPUT HORIZONTAL START REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HPS															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	HPS	Input Horizontal Position Sets the horizontal position of the first pixel for each line within the global frame. After SRC_HPS, the pixel at the VAL'th position is processed as the first pixel. This value must be even.	RW	0x0000

**Table 8-581. Register Call Summary for Register RZB\_I\_HPS**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-582. RZB\_O\_VSZ**

<b>Address Offset</b>	0x0000 011C	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT VERTICAL SIZER REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VSZ															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VSZ	The target output size of the resized image. The number of output lines is (VSZ+1). Set 479, when 480 lines of output is required.	RW	0x0000

**Table 8-583. Register Call Summary for Register RZB\_O\_VSZ**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-584. RZB\_O\_HSZ**

<b>Address Offset</b>	0x0000 0120		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT HORIZONTAL SIZE REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HSZ												HSZ_LSB			

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:1	HSZ	The horizontal size of output image. The number of pixel in each line is (HSZ+1). Set 479, when 480 pixels are required. This value must be lower than the max memory line size supported by the resizer engine, except in RAW pass through mode. Note that the LSB of the 13-bit HSZ value is fixed to 1 such that the horizontal size is always even.	RW	0x000
0	HSZ_LSB	The least significant bit of HSZ is forced to 1.	R	1

**Table 8-585. Register Call Summary for Register RZB\_O\_HSZ**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-586. RZB\_V\_PHS\_Y**

<b>Address Offset</b>	0x0000 0124		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - INITIAL LUMINANCE PHASE OF VERTICAL RESIZING PROCESS When YUV4:2:2 data are output, the phase values for luma and chroma should typically be equal, that is, RZX_V_PHS_Y= RZX_V_PHS_C. The following constraints apply when setting the initial vertical phases ABS(RZX_V_PHS_Y - RZX_V_PHS_C) RZX_V_DIF. This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two luma pixels. Note that the absolute value is used, hence, the initial luma phase can be greater than the initial chroma phase or the other way around. As a reminder, the distance between two output pixels for luma is given by RZX_V_DIF.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	Y	The initial value for the luma phase in vertical resizing process. This value is in U14Q8 fractional format.	RW	0x0000

**Table 8-587. Register Call Summary for Register RZB\_V\_PHS\_Y**

ISS ISP

- [ISS RESIZER Register Summary: \[0\]](#)

**Table 8-588. RZB\_V\_PHS\_C**

<b>Address Offset</b>	0x0000 0128	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - INITIAL CHROMINANCE PHASE OF VERTICAL RESIZING PROCESS When YUV4:2:2 data are output, the phase values for luma and chroma should typically be equal, that is, RZX_V_PHS_Y= RZX_V_PHS_C. The following constraints apply when setting the initial vertical phases ABS(RZX_V_PHS_Y - RZX_V_PHS_C) RZX_V_DIF. This constraint means that at most the distance between the initial phases for luminance and chrominance is not expected to exceed the distance between two luma pixels. Note that the absolute value is used, hence, the initial luma phase can be greater than the initial chroma phase or the other way around. As a reminder, the distance between two output pixels for luma is given by RZX_V_DIF.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	C	The initial value for the chroma phase in vertical resizing process. This value is in U14Q8 fractional format.	RW	0x0000

**Table 8-589. Register Call Summary for Register RZB\_V\_PHS\_C**

ISS ISP

- [ISS RESIZER Register Summary: \[0\]](#)

**Table 8-590. RZB\_V\_DIF**

<b>Address Offset</b>	0x0000 012C	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - VERTICAL RESIZER REGISTERR	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																V															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	V	The parameter for vertical resize. The actual resizing ratio is 256/RZB_V_DIF. In normal mode: 16 <= RZB_V_DIF <= 4096. In down-scale mode: 256 <= RZB_V_DIF <= 4096.	RW	0x0000

**Table 8-591. Register Call Summary for Register RZB\_V\_DIF**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)
- [ISS RESIZER Register Description: \[2\] \[3\] \[4\]](#)



**Table 8-592. RZB\_V\_TYP**

<b>Address Offset</b>	0x0000 0130	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - INTERPOLATION METHOD FOR VERTICAL RESIZING	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											C	Y			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	C	Selection of resizing method for chrominance: vertical 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0
0	Y	Selection of resizing method for luminance: vertical 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0

**Table 8-593. Register Call Summary for Register RZB\_V\_TYP**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-594. RZB\_V\_LPF**

<b>Address Offset</b>	0x0000 0134	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - VERTICAL LPF INTENSITY REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				C			Y								

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:6	C	The intensity parameter for chroma vertical low pass filtering.	RW	0x00
5:0	Y	The intensity parameter for luma vertical low pass filtering.	RW	0x00

**Table 8-595. Register Call Summary for Register RZB\_V\_LPF**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-596. RZB\_H\_PHS**

<b>Address Offset</b>	0x0000 0138
<b>Physical Address</b>	See <a href="#">Table 8-447</a>
<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - INITIAL PHASE OF HORIZONTAL RESIZING PROCESS
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PHS															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	PHS	Initial value for the phase in horizontal resizing process, that is, the sampling position is shifted. This value is in U14Q8 fractional format. Example: If RZX_H_PHS = 128, the first output pixel is sampled at the center of the first two valid input pixels. If RZX_L_HPS=100 and RZX_H_PHS=128, the first output pixel is resampled at the center of the 100-th and the 101-st input pixels.	RW	0x0000

**Table 8-597. Register Call Summary for Register RZB\_H\_PHS**

ISS ISP

- [ISS RESIZER Register Summary: \[0\]](#)

**Table 8-598. RZB\_H\_PHS\_ADJ**

<b>Address Offset</b>	0x0000 013C
<b>Physical Address</b>	See <a href="#">Table 8-447</a>
<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - LUMINANCE HORIZONTAL PHASE ADJUSTMENT The <a href="#">RZA_H_PHS_ADJ</a> register enables to adjust the horizontal phase for the luma component when averaging is enabled (the horizontal averaging disrupts the relative sampling point between luminance and chrominance when YUV4:2:2 cosited data is input), that is, the relative phase between luma and chroma is different before and after the horizontal averager.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADJ															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:0	ADJ	Horizontal phase adjustment value. This value is in U9Q8 fractional format. This value is expected to be equal to zero if the averager is disabled or if input chroma is centered.	RW	0x000

**Table 8-599. Register Call Summary for Register RZB\_H\_PHS\_ADJ**

ISS ISP

- [ISS RESIZER Register Summary: \[0\]](#)
- [ISS IPIPE Register Description: \[1\]](#)

**Table 8-600. RZB\_H\_DIF**

<b>Address Offset</b>	0x0000 0140		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - HORIZONTAL RESIZER REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																H															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	H	The parameter for horizontal resizing process. The actual resizing ratio is 256/VAL. In normal mode 16<= RSZ_RZA_H_DIF<=4096 In down-scale mode 256<=RSZ_RZA_H_DIF<=4096	RW	0x0000

**Table 8-601. Register Call Summary for Register RZB\_H\_DIF**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-602. RZB\_H\_TYP**

<b>Address Offset</b>	0x0000 0144		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C	Y														

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	C	Selection of resizing method for chrominance: horizontal 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0
0	Y	Selection of resizing method for luminance: horizontal 0x0: 4-tap cubic convolution (default) 0x1: 2-tap linear interpolation	RW	0

**Table 8-603. Register Call Summary for Register RZB\_H\_TYP**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-604. RZB\_H\_LPF**

<b>Address Offset</b>	0x0000 0148		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - HORIZONTAL LPF INTENSITY REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C								Y							

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:6	C	Horizontal LPF Intensity for Chrominance	RW	0x00
5:0	Y	Selection of resizing method for Luminance in horizontal direction	RW	0x00

**Table 8-605. Register Call Summary for Register RZB\_H\_LPF**

ISS ISP

- [ISS ISP RSZ Engines Interframe Image Processing Settings: \[0\] \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-606. RZB\_DWN\_EN**

<b>Address Offset</b>	0x0000 014C		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - DOWNSCALE ENABLE REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DWN_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	DWN_EN	Resizer downscale enable 0x0: Off. Normal operation: upscale and downscale are allowed. 0x1: On. Downscale mode.	RW	0

**Table 8-607. Register Call Summary for Register RZB\_DWN\_EN**

ISS ISP

- [ISS RESIZER Register Summary: \[0\]](#)

**Table 8-608. RZB\_DWN\_AV**

<b>Address Offset</b>	0x0000 0150	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							V	H							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5:3	V	Vertical averaging size : $1/2^{(VWT+1)}$ The range goes from 1/2 to 1/256 in power of two. 0: _DIV2 1/2 down scale 1: _DIV4 1/4 down scale 2: _DIV8 1/8 down scale 3: _DIV16 1/16 down scale 4: _DIV32 1/32 down scale 5: _DIV64 1/64 down scale 6: _DIV128 1/128 down scale 7: _DIV256 1/256 down scale	RW	0x0
2:0	H	Horizontal averaging size : $1/2^{(HWT+1)}$ The range goes from 1/2 to 1/256 in power of two. 0: _DIV2 1/2 down scale 1: _DIV4 1/4 down scale 2: _DIV8 1/8 down scale 3: _DIV16 1/16 down scale 4: _DIV32 1/32 down scale 5: _DIV64 1/64 down scale 6: _DIV128 1/128 down scale 7: _DIV256 1/256 down scale	RW	0x0

**Table 8-609. Register Call Summary for Register RZB\_DWN\_AV**

ISS ISP

- [ISS RESIZER Register Summary: \[0\]](#)

**Table 8-610. RZB\_RGB\_EN**

<b>Address Offset</b>	0x0000 0154	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - RGB OUTPUT ENABLE	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											RGB_EN				

ISS ISP

www.ti.com

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	RGB_EN	Enable of RGB output In pass through mode, this register must be 0. This bit can only be set to 1 when YUV4:2:2 data are output. YUV4:2:2 data output is selected when SRC_FMT1.IN420 = 0 and RZB_420.YEN = RZB_420.CEN = 0  0x0: Off (YCbCr output) 0x1: On (RGB output)	RW	0

**Table 8-611. Register Call Summary for Register RZB\_RGB\_EN**

ISS ISP

- [ISS ISP RSZ Functional Description: \[0\]](#)
- [ISS RESIZER Register Summary: \[1\]](#)

**Table 8-612. RZB\_RGB\_TYP**

<b>Address Offset</b>	0x0000 0158	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B - RGB OUTPUT CONTROL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MSK1	MSK0	TYP													

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	MSK1	Enables masking of the last 2 pixels This bit is used to mask the 2 last pixels at the image boundary which are affected by the YUV4:2:2 to YUV4:4:4 conversion.  0x0: output the last 2 pixels 0x1: mask the last 2 pixels (Resizer do not output them.)	RW	0
1	MSK0	Enables masking of the first 2 pixels This bit is used to mask the 2 first pixels at the image boundary which are affected by the YUV4:2:2 to YUV4:4:4 conversion.  0x0: output the first 2 pixels 0x1: mask the first 2 pixels (Resizer do not output them.)	RW	0
0	TYP	16bit/32bit output selection  0x0: 32-bit output: alpha + R + G + B (8 bit each) This mode comes with performance degradation. The maximum input frequency in this mode is 160 MHz. This due to the fact that the output is 4 bytes / pixel. 0x1: 16-bit output: R(5 bit) + G (6 bit) + B (5 bit)	RW	0

**Table 8-613. Register Call Summary for Register RZB\_RGB\_TYP**

ISS ISP

- [ISS RESIZER Register Summary: \[0\]](#)

**Table 8-614. RZB\_RGB\_BLD**

<b>Address Offset</b>	0x0000 015C	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - RGB BLEND REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BLD															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	BLD	The alpha value used in 32-bit RGBA output mode	RW	0x00

**Table 8-615. Register Call Summary for Register RZB\_RGB\_BLD**

ISS ISP

- [ISS RESIZER Register Summary: \[0\]](#)

**Table 8-616. RZB\_SDR\_Y\_BAD\_H**

<b>Address Offset</b>	0x0000 0160	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT MEMORY BASE ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_BAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_BAD_H	Memory Base Address Sets 16 upper bits of the 32-bit base address of the circular buffer in memory.	RW	0x0000

**Table 8-617. Register Call Summary for Register RZB\_SDR\_Y\_BAD\_H**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-618. RZB\_SDR\_Y\_BAD\_L**

<b>Address Offset</b>	0x0000 0164		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT MEMORY BASE ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_BAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_BAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit base address of the circular buffer in memory. It is a byte address. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RAW, RGB5:6:5 and YUV4:2:2 formats (output data on 16 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on. RGBA format (output data on 32 bits): The three least significant bits must be set to "000" when horizontal reversal mode is off. The three least significant bits must be set to "111" when horizontal reversal mode is on.	RW	0x0000

**Table 8-619. Register Call Summary for Register RZB\_SDR\_Y\_BAD\_L**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)



**Table 8-620. RZB\_SDR\_Y\_SAD\_H**

<b>Address Offset</b>	0x0000 0168		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_SAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_SAD_H	Memory Start Address Sets 16 upper bits of the 32-bit start address in memory.	RW	0x0000

**Table 8-621. Register Call Summary for Register RZB\_SDR\_Y\_SAD\_H**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-622. RZB\_SDR\_Y\_SAD\_L**

<b>Address Offset</b>	0x0000 016C		
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b>	ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0, RGB5:6:5, RGBA. RAW: RAW data is written to this address YUV4:2:2: YUV data is written to this address YUV4:2:0: Y data is written to this address RGB5:6:5: 16-bit RGB data is written to this address RGBA: 32-bit RGBA data is written to this address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_SAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	Y_SAD_L	<p>Memory Start Address</p> <p>Sets the 16 lower bits of the 32-bit start address in memory. It is a byte address. For every frame, the first line of data will be written to this address.</p> <p>We have: <math>SAD = BAD + (PTR\_S \times OFT)</math> and <math>PTR\_S &lt; PTR\_E</math></p> <p>If the first line must be written at the beginning of the circular buffer memory then <math>SAD = BAD</math> and <math>PTR\_S = 0</math>.</p> <p>YUV4:2:0 format (output data on 8 bits):</p> <p>The two least significant bits must be set to 00 when horizontal reversal mode is off.</p> <p>The two least significant bits must be set to 11 when horizontal reversal mode is on.</p> <p>RAW, RGB5:6:5 and YUV4:2:2 formats (output data on 16 bits):</p> <p>The two least significant bits must be set to 00 when horizontal reversal mode is off.</p> <p>The two least significant bits must be set to 11 when horizontal reversal mode is on.</p> <p>RGBA format (output data on 32 bits):</p> <p>The three least significant bits must be set to "000" when horizontal reversal mode is off.</p> <p>The three least significant bits must be set to "111" when horizontal reversal mode is on.</p>	RW	0x0000

**Table 8-623. Register Call Summary for Register RZB\_SDR\_Y\_SAD\_L**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 2: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-624. RZB\_SDR\_Y\_OFT**

<b>Address Offset</b>	0x0000 0170																																																													
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER																																																												
<b>Description</b>	RESIZER B - OUTPUT MEMORY OFFSET REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width:100%; text-align:center; border-collapse: collapse;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="12">Y_OFT</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																Y_OFT											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
RESERVED																Y_OFT																																														
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																										
31:17	RESERVED		R	0x0000																																																										
16:0	Y_OFT	<p>Memory Line Offset</p> <p>Sets the size of each line in the circular buffer. It is expressed in bytes and unsigned. Note that OFT does not necessary corresponds to the size of a line in a frame, it can be much bigger. The line offset must be a multiple of 128 bytes (bits [6:0] of RZX_SDR_Y_OFT and RZX_SDR_C_OFT must be set to 0).</p> <p>Example:</p> <p>line 0 address = SAD</p> <p>line 1 address = SAD + 1 x OFT</p> <p>line 2 address = SAD + 2 x OFT</p>	RW	0x00000																																																										

**Table 8-625. Register Call Summary for Register RZB\_SDR\_Y\_OFT**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-626. RZB\_SDR\_Y\_PTR\_S**

<b>Address Offset</b>	0x0000 0174	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_PTR_S															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	Y_PTR_S	Start Line of Memory Pointer Sets the vertical position of the first output line in the output memory space. This value is expressed in number of lines. The hardware uses it to set up the initial value of the circular buffer. It must be set up such as PTR_S = (SAD - BAD) / OFT. This value must be set to 0 when RSZ_RZA_SDR_Y_BAD = RSZ_RZA_SDR_Y_SAD.	RW	0x0000

**Table 8-627. Register Call Summary for Register RZB\_SDR\_Y\_PTR\_S**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-628. RZB\_SDR\_Y\_PTR\_E**

<b>Address Offset</b>	0x0000 0178	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B - OUTPUT MEMORY END ADDRESS REGISTER This register is used if the output data format is one of the following: RAW, YUV4:2:2, YUV4:2:0 or RGBA.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Y_PTR_E															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	Y_PTR_E	End Line of Memory Pointer Sets the maximum number of lines to be stored in the output memory space. This value is expressed in number of lines. When the number of output lines exceeds this value, the address restarts from the first address in the output memory space (BAD).	RW	0x0000

**Table 8-629. Register Call Summary for Register RZB\_SDR\_Y\_PTR\_E**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS ISP5 SYS1 Register Description: \[2\] \[3\] \[4\] \[5\]](#)
- [ISS RESIZER Register Summary: \[6\]](#)

**Table 8-630. RZB\_SDR\_C\_BAD\_H**

<b>Address Offset</b>	0x0000 017C	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT MEMORY BASE ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_BAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_BAD_H	Memory Base Address Sets the 16 upper bits of the 32-bit base address of the circular buffer in memory.	RW	0x0000

**Table 8-631. Register Call Summary for Register RZB\_SDR\_C\_BAD\_H**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-632. RZB\_SDR\_C\_BAD\_L**

<b>Address Offset</b>	0x0000 0180	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT MEMORY BASE ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_BAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_BAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit base address of the circular buffer in memory. It is a byte address. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on.	RW	0x0000

**Table 8-633. Register Call Summary for Register RZB\_SDR\_C\_BAD\_L**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-634. RZB\_SDR\_C\_SAD\_H**

<b>Address Offset</b>	0x0000 0184	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_SAD_H															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_SAD_H	Memory Base Address Sets the 16 upper bits of the 32-bit start address in memory.	RW	0x0000

**Table 8-635. Register Call Summary for Register RZB\_SDR\_C\_SAD\_H**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-636. RZB\_SDR\_C\_SAD\_L**

<b>Address Offset</b>	0x0000 0188	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0. U and V data are written into this buffer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_SAD_L															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	C_SAD_L	Memory Base Address Sets the 16 lower bits of the 32-bit start address in memory. It is a byte address. For every frame, the first line of data will be written to this address (C_SAD_H/C_SAD_L). We have: SAD = BAD + (PTR_S x OFT) and PTR_S < PTR_E If the first line must be written at the beginning of the circular buffer memory then SAD = BAD and PTR_S = 0. YUV4:2:0 format (output data on 8 bits): The two least significant bits must be set to 00 when horizontal reversal mode is off. The two least significant bits must be set to 11 when horizontal reversal mode is on.	RW	0x0000

**Table 8-637. Register Call Summary for Register RZB\_SDR\_C\_SAD\_L**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS ISP RSZ Global Image Processing Settings – Subsequence 2: \[2\]](#)
- [ISS RESIZER Register Summary: \[3\]](#)

**Table 8-638. RZB\_SDR\_C\_OFT**

<b>Address Offset</b>	0x0000 018C	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT MEMORY OFFSET REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_OFT															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:0	C_OFT	Memory Line Offset Sets the size of each line in the circular buffer. It is expressed in bytes and unsigned. Note that OFT does not necessary corresponds to the size of a line in a frame, it can be much bigger. The line offset must be a multiple of 128 bytes (bits [6:0] of RZX_SDR_Y_OFT and RZX_SDR_C_OFT must be set to 0). Example: line 0 address = SAD line 1 address = SAD + 1 x OFT line 2 address = SAD + 2 x OFT	RW	0x00000

**Table 8-639. Register Call Summary for Register RZB\_SDR\_C\_OFT**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-640. RZB\_SDR\_C\_PTR\_S**

<b>Address Offset</b>	0x0000 0190	
<b>Physical Address</b>	See <a href="#">Table 8-447</a>	<b>Instance</b> ISS_RESIZER
<b>Description</b>	RESIZER B - OUTPUT MEMORY START ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_PTR_S															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	C_PTR_S	Start Line of Memory Pointer Sets the vertical position of the first output line in the output memory space. This value is expressed in number of lines. The hardware uses it to set up the initial value of the circular buffer. It must be set up such as PTR_S = (SAD - BAD) / OFT. This value must be set to 0 when RSZ_RZA_SDR_C_BAD = RSZ_RZA_SDR_C_SAD.	RW	0x0000

**Table 8-641. Register Call Summary for Register RZB\_SDR\_C\_PTR\_S**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

**Table 8-642. RZB\_SDR\_C\_PTR\_E**

<b>Address Offset</b>	0x0000 0194	<b>Instance</b>	ISS_RESIZER
<b>Physical Address</b>	See <a href="#">Table 8-447</a>		
<b>Description</b>	RESIZER B - OUTPUT MEMORY END ADDRESS REGISTER FOR CHROMA DATA (YUV4:2:0) This register is used if the output data format is YUV4:2:0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																C_PTR_E															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	C_PTR_E	End Line of Memory Pointer Sets the maximum number of lines to be stored in the output memory space. This value is expressed in number of lines. When the number of output lines exceeds this value, the address restarts from the first address in the output memory space (BAD).	RW	0x0000

**Table 8-643. Register Call Summary for Register RZB\_SDR\_C\_PTR\_E**

ISS ISP

- [ISS ISP RSZ Circular Buffer: \[0\]](#)
- [ISS ISP RSZ Global Image Processing Settings: \[1\]](#)
- [ISS RESIZER Register Summary: \[2\]](#)

### 8.3.5.5 ISS IPIPE registers

#### 8.3.5.5.1 ISS IPIPE Register Summary

**Table 8-644. ISS IPIPE Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address
<a href="#">IPIPE_SRC_EN</a>	RW	32	0x0000 0000	0x5201 0800
<a href="#">IPIPE_SRC_MODE</a>	RW	32	0x0000 0004	0x5201 0804
<a href="#">IPIPE_SRC_FMT</a>	RW	32	0x0000 0008	0x5201 0808
<a href="#">IPIPE_SRC_COL</a>	RW	32	0x0000 000C	0x5201 080C

Table 8-644. ISS IPIPE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address
IPIPE_SRC_VPS	RW	32	0x0000 0010	0x5201 0810
IPIPE_SRC_VSZ	RW	32	0x0000 0014	0x5201 0814
IPIPE_SRC_HPS	RW	32	0x0000 0018	0x5201 0818
IPIPE_SRC_HSZ	RW	32	0x0000 001C	0x5201 081C
IPIPE_SEL_SBU	RW	32	0x0000 0020	0x5201 0820
IPIPE_SRC_STA	R	32	0x0000 0024	0x5201 0824
IPIPE_GCK_MMR	RW	32	0x0000 0028	0x5201 0828
IPIPE_GCK_PIX	RW	32	0x0000 002C	0x5201 082C
RESERVED	R	32	0x0000 0030	0x5201 0830
IPIPE_DPC_LUT_EN	RW	32	0x0000 0034	0x5201 0834
IPIPE_DPC_LUT_SEL	RW	32	0x0000 0038	0x5201 0838
IPIPE_DPC_LUT_ADR	RW	32	0x0000 003C	0x5201 083C
IPIPE_DPC_LUT_SIZ	RW	32	0x0000 0040	0x5201 0840
RESERVEDa <sup>(1)</sup>	RW	32	0x0000 0044 + (b * 4)	0x5201 0844 + (b * 4)
IPIPE_LSC_VOFT	RW	32	0x0000 0090	0x5201 0890
IPIPE_LSC_VA2	RW	32	0x0000 0094	0x5201 0894
IPIPE_LSC_VA1	RW	32	0x0000 0098	0x5201 0898
IPIPE_LSC_VS	RW	32	0x0000 009C	0x5201 089C
IPIPE_LSC_HOFT	RW	32	0x0000 00A0	0x5201 08A0
IPIPE_LSC_HA2	RW	32	0x0000 00A4	0x5201 08A4
IPIPE_LSC_HA1	RW	32	0x0000 00A8	0x5201 08A8
IPIPE_LSC_HS	RW	32	0x0000 00AC	0x5201 08AC
IPIPE_LSC_GAN_R	RW	32	0x0000 00B0	0x5201 08B0
IPIPE_LSC_GAN_GR	RW	32	0x0000 00B4	0x5201 08B4
IPIPE_LSC_GAN_GB	RW	32	0x0000 00B8	0x5201 08B8
IPIPE_LSC_GAN_B	RW	32	0x0000 00BC	0x5201 08BC
IPIPE_LSC_OFT_R	RW	32	0x0000 00C0	0x5201 08C0
IPIPE_LSC_OFT_GR	RW	32	0x0000 00C4	0x5201 08C4
IPIPE_LSC_OFT_GB	RW	32	0x0000 00C8	0x5201 08C8
IPIPE_LSC_OFT_B	RW	32	0x0000 00CC	0x5201 08CC
IPIPE_LSC_SHF	RW	32	0x0000 00D0	0x5201 08D0
IPIPE_LSC_MAX	RW	32	0x0000 00D4	0x5201 08D4
RESERVEDb <sup>(1)</sup>	RW	32	0x0000 00D8 + (b * 0x4)	0x5201 08D8 + (b * 0x4)
RESERVEDc <sup>(1)</sup>	RW	32	0x0000 01B8 + (c * 0x4)	0x5201 09B8 + (c * 0x4)
IPIPE_WB2_OFT_R	RW	32	0x0000 01D0	0x5201 09D0
IPIPE_WB2_OFT_GR	RW	32	0x0000 01D4	0x5201 09D4
IPIPE_WB2_OFT_GB	RW	32	0x0000 01D8	0x5201 09D8
IPIPE_WB2_OFT_B	RW	32	0x0000 01DC	0x5201 09DC
IPIPE_WB2_WGN_R	RW	32	0x0000 01E0	0x5201 09E0
IPIPE_WB2_WGN_GR	RW	32	0x0000 01E4	0x5201 09E4
IPIPE_WB2_WGN_GB	RW	32	0x0000 01E8	0x5201 09E8
IPIPE_WB2_WGN_B	RW	32	0x0000 01EC	0x5201 09EC

- <sup>(1)</sup> a = 0 to 18  
b = 0 to 55  
c = 0 to 5  
d = 0 to 14  
e = 0 to 1  
f = 0 to 24  
g = 0 to 18



**Table 8-644. ISS IPIPE Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address
RESERVEDd <sup>(1)</sup>	RW	32	0x0000 01F0 + (d * 0x4)	0x5201 09F0 + (d * 0x4)
IPIPE_RGB1_MUL_RR	RW	32	0x0000 022C	0x5201 0A2C
IPIPE_RGB1_MUL_GR	RW	32	0x0000 0230	0x5201 0A30
IPIPE_RGB1_MUL_BR	RW	32	0x0000 0234	0x5201 0A34
IPIPE_RGB1_MUL_RG	RW	32	0x0000 0238	0x5201 0A38
IPIPE_RGB1_MUL_GG	RW	32	0x0000 023C	0x5201 0A3C
IPIPE_RGB1_MUL_BG	RW	32	0x0000 0240	0x5201 0A40
IPIPE_RGB1_MUL_RB	RW	32	0x0000 0244	0x5201 0A44
IPIPE_RGB1_MUL_GB	RW	32	0x0000 0248	0x5201 0A48
IPIPE_RGB1_MUL_BB	RW	32	0x0000 024C	0x5201 0A4C
IPIPE_RGB1_OFT_OR	RW	32	0x0000 0250	0x5201 0A50
IPIPE_RGB1_OFT_OG	RW	32	0x0000 0254	0x5201 0A54
IPIPE_RGB1_OFT_OB	RW	32	0x0000 0258	0x5201 0A58
IPIPE_GMM_CFG	RW	32	0x0000 025C	0x5201 0A5C
IPIPE_RGB2_MUL_RR	RW	32	0x0000 0260	0x5201 0A60
IPIPE_RGB2_MUL_GR	RW	32	0x0000 0264	0x5201 0A64
IPIPE_RGB2_MUL_BR	RW	32	0x0000 0268	0x5201 0A68
IPIPE_RGB2_MUL_RG	RW	32	0x0000 026C	0x5201 0A6C
IPIPE_RGB2_MUL_GG	RW	32	0x0000 0270	0x5201 0A70
IPIPE_RGB2_MUL_BG	RW	32	0x0000 0274	0x5201 0A74
IPIPE_RGB2_MUL_RB	RW	32	0x0000 0278	0x5201 0A78
IPIPE_RGB2_MUL_GB	RW	32	0x0000 027C	0x5201 0A7C
IPIPE_RGB2_MUL_BB	RW	32	0x0000 0280	0x5201 0A80
IPIPE_RGB2_OFT_OR	RW	32	0x0000 0284	0x5201 0A84
IPIPE_RGB2_OFT_OG	RW	32	0x0000 0288	0x5201 0A88
IPIPE_RGB2_OFT_OB	RW	32	0x0000 028C	0x5201 0A8C
RESERVED	RW	32	0x0000 0290	0x5201 0A90
IPIPE_YUV_ADJ	RW	32	0x0000 0294	0x5201 0A94
IPIPE_YUV_MUL_RY	RW	32	0x0000 0298	0x5201 0A98
IPIPE_YUV_MUL_GY	RW	32	0x0000 029C	0x5201 0A9C
IPIPE_YUV_MUL_BY	RW	32	0x0000 02A0	0x5201 0AA0
IPIPE_YUV_MUL_RCB	RW	32	0x0000 02A4	0x5201 0AA4
IPIPE_YUV_MUL_GCB	RW	32	0x0000 02A8	0x5201 0AA8
IPIPE_YUV_MUL_BCB	RW	32	0x0000 02AC	0x5201 0AAC
IPIPE_YUV_MUL_RCR	RW	32	0x0000 02B0	0x5201 0AB0
IPIPE_YUV_MUL_GCR	RW	32	0x0000 02B4	0x5201 0AB4
IPIPE_YUV_MUL_BCR	RW	32	0x0000 02B8	0x5201 0AB8
IPIPE_YUV_OFT_Y	RW	32	0x0000 02BC	0x5201 0ABC
IPIPE_YUV_OFT_CB	RW	32	0x0000 02C0	0x5201 0AC0
IPIPE_YUV_OFT_CR	RW	32	0x0000 02C4	0x5201 0AC4
IPIPE_YUV_PHS	RW	32	0x0000 02C8	0x5201 0AC8

Table 8-644. ISS IPIPE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address
RESERVED <sup>e(2)</sup>	RW	32	0x0000 02CC + (e * 0x4)	0x5201 0ACC + (e * 0x4)
IPIPE_YEE_EN	RW	32	0x0000 02D4	0x5201 0AD4
IPIPE_YEE_TYP	RW	32	0x0000 02D8	0x5201 0AD8
IPIPE_YEE_SHF	RW	32	0x0000 02DC	0x5201 0ADC
IPIPE_YEE_MUL_00	RW	32	0x0000 02E0	0x5201 0AE0
IPIPE_YEE_MUL_01	RW	32	0x0000 02E4	0x5201 0AE4
IPIPE_YEE_MUL_02	RW	32	0x0000 02E8	0x5201 0AE8
IPIPE_YEE_MUL_10	RW	32	0x0000 02EC	0x5201 0AEC
IPIPE_YEE_MUL_11	RW	32	0x0000 02F0	0x5201 0AF0
IPIPE_YEE_MUL_12	RW	32	0x0000 02F4	0x5201 0AF4
IPIPE_YEE_MUL_20	RW	32	0x0000 02F8	0x5201 0AF8
IPIPE_YEE_MUL_21	RW	32	0x0000 02FC	0x5201 0AFC
IPIPE_YEE_MUL_22	RW	32	0x0000 0300	0x5201 0B00
IPIPE_YEE_THR	RW	32	0x0000 0304	0x5201 0B04
IPIPE_YEE_E_GAN	RW	32	0x0000 0308	0x5201 0B08
IPIPE_YEE_E_THR_1	RW	32	0x0000 030C	0x5201 0B0C
IPIPE_YEE_E_THR_2	RW	32	0x0000 0310	0x5201 0B10
IPIPE_YEE_G_GAN	RW	32	0x0000 0314	0x5201 0B14
IPIPE_YEE_G_OFT	RW	32	0x0000 0318	0x5201 0B18
RESERVED <sup>f(3)</sup>	RW	32	0x0000 031C + (f * 0x4)	0x5201 0B1C + (f * 0x4)
IPIPE_BOX_EN	RW	32	0x0000 0380	0x5201 0B80
IPIPE_BOX_MODE	RW	32	0x0000 0384	0x5201 0B84
IPIPE_BOX_TYP	RW	32	0x0000 0388	0x5201 0B88
IPIPE_BOX_SHF	RW	32	0x0000 038C	0x5201 0B8C
IPIPE_BOX_SDR_SAD_H	RW	32	0x0000 0390	0x5201 0B90
IPIPE_BOX_SDR_SAD_L	RW	32	0x0000 0394	0x5201 0B94
RESERVED	R	32	0x0000 0398	0x5201 0B98
IPIPE_HST_EN	RW	32	0x0000 039C	0x5201 0B9C
IPIPE_HST_MODE	RW	32	0x0000 03A0	0x5201 0BA0
IPIPE_HST_SEL	RW	32	0x0000 03A4	0x5201 0BA4
IPIPE_HST_PARA	RW	32	0x0000 03A8	0x5201 0BA8
IPIPE_HST_0_VPS	RW	32	0x0000 03AC	0x5201 0BAC
IPIPE_HST_0_VSZ	RW	32	0x0000 03B0	0x5201 0BB0
IPIPE_HST_0_HPS	RW	32	0x0000 03B4	0x5201 0BB4
IPIPE_HST_0_HSZ	RW	32	0x0000 03B8	0x5201 0BB8

<sup>(2)</sup> a = 0 to 18  
b = 0 to 55  
c = 0 to 5  
d = 0 to 14  
e = 0 to 1  
f = 0 to 24  
g = 0 to 18

<sup>(3)</sup> a = 0 to 18  
b = 0 to 55  
c = 0 to 5  
d = 0 to 14  
e = 0 to 1  
f = 0 to 24  
g = 0 to 18

**Table 8-644. ISS IPIPE Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPE Base Address
IPIPE_HST_1_VPS	RW	32	0x0000 03BC	0x5201 0BBC
IPIPE_HST_1_VSZ	RW	32	0x0000 03C0	0x5201 0BC0
IPIPE_HST_1_HPS	RW	32	0x0000 03C4	0x5201 0BC4
IPIPE_HST_1_HSZ	RW	32	0x0000 03C8	0x5201 0BC8
IPIPE_HST_2_VPS	RW	32	0x0000 03CC	0x5201 0BCC
IPIPE_HST_2_VSZ	RW	32	0x0000 03D0	0x5201 0BD0
IPIPE_HST_2_HPS	RW	32	0x0000 03D4	0x5201 0BD4
IPIPE_HST_2_HSZ	RW	32	0x0000 03D8	0x5201 0BD8
IPIPE_HST_3_VPS	RW	32	0x0000 03DC	0x5201 0BDC
IPIPE_HST_3_VSZ	RW	32	0x0000 03E0	0x5201 0BE0
IPIPE_HST_3_HPS	RW	32	0x0000 03E4	0x5201 0BE4
IPIPE_HST_3_HSZ	RW	32	0x0000 03E8	0x5201 0BE8
IPIPE_HST_TBL	RW	32	0x0000 03EC	0x5201 0BEC
IPIPE_HST_MUL_R	RW	32	0x0000 03F0	0x5201 0BF0
IPIPE_HST_MUL_GR	RW	32	0x0000 03F4	0x5201 0BF4
IPIPE_HST_MUL_GB	RW	32	0x0000 03F8	0x5201 0BF8
IPIPE_HST_MUL_B	RW	32	0x0000 03FC	0x5201 0BFC
RESERVEDg <sup>(3)</sup>	RW	32	0x0000 0400 + (g * 0x4)	0x5201 0C00 + (g * 0x4)

**8.3.5.5.2 ISS IPIPE Register Description**

**Table 8-645. IPIPE\_SRC\_EN**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	This register is not shadowed	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											Z				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	The start flag of the IPIPE module. When EN is 1, the IPIPE module starts a processing from the next rising edge of the VD. If the processing mode of the IPIPE module is one shot, the EN is cleared to 0 immediately after the processing has started.  0x0: waiting 0x1: start/busy	RW	0

**Table 8-646. Register Call Summary for Register IPIPE\_SRC\_EN**

## ISS ISP

- ISS ISP IPIPE Input Interface: [0]
- Main Sequence – ISS ISP IPIPE Global Initialization: [1]
- ISS ISP IPIPE Processing Path: Case 1 Configuration: [2]
- ISS ISP IPIPE Processing Path: Case 2 Configuration: [3]
- ISS ISP IPIPE Processing Path: Case 3 Configuration: [4]
- ISS ISP IPIPE Processing Path: Case 4 Configuration: [5]
- ISS IPIPE Register Summary: [6]

**Table 8-647. IPIPE\_SRC\_MODE**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												WRT	OST		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	WRT	The mode selection of the ipipeif_wrt which is an input port of the IPIPE module. If WRT is 0, the IPIPE module does not use the ipipeif_wrt. Else the IPIPE module uses it. 0x0: Disable 0x1: Enable	RW	0
0	OST	The processing mode selection of the IPIPE module. Value 0 indicates the mode of free run, value 1 indicates the mode of one shot. 0x0: Free run 0x1: One shot	RW	0

**Table 8-648. Register Call Summary for Register IPIPE\_SRC\_MODE**

## ISS ISP

- ISS ISP IPIPE Input Interface: [0] [1] [2] [3]
- Main Sequence – ISS ISP IPIPE Global Initialization: [4] [5]
- ISS ISP IPIPE Processing Path: Case 1 Configuration: [6] [7]
- ISS ISP IPIPE Processing Path: Case 2 Configuration: [8] [9]
- ISS ISP IPIPE Processing Path: Case 3 Configuration: [10] [11]
- ISS ISP IPIPE Processing Path: Case 4 Configuration: [12] [13]
- ISS IPIPE Register Summary: [14]

**Table 8-649. IPIPE\_SRC\_FMT**

<b>Address Offset</b>	0x0000 0008																														
<b>Physical Address</b>	See <a href="#">Table 8-644</a>								<b>Instance</b>	ISS_IPIPE																					
<b>Description</b>	RW																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								FMT															
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>														<b>Type</b>	<b>Reset</b>														
31:16	RESERVED															R	0x0000														
15:2	RESERVED															R	0x0000														
1:0	FMT	IPIPE module data path selection														RW	0x0														
		0x0: IN: RAW BAYER OUT: YUV4:2:2 Note that the IPIPE YUV4:2:2 output goes to the RESIZER module where it can be further be converted in YUV4:2:0 or RGB format.																													
		0x1: IN: RAW BAYER OUT: RAW BAYER The data are output after the White Balance module. It enables to bypass a large part of the IPIPE module.																													
		0x3: IN: YUV4:2:2 OUT: YUV4:2:2 Note that the IPIPE YUV4:2:2 output goes to the RESIZER module where it can be further be converted in YUV4:2:0 or RGB format.																													
		0x2: IN: RAW BAYER OUT: DISABLED The data are only going to BOXCAR and HISTOGRAM modules.																													

**Table 8-650. Register Call Summary for Register IPIPE\_SRC\_FMT**

ISS ISP

- [ISS ISP IPIPE Input Interface: \[0\] \[1\]](#)
- [Main Sequence – ISS ISP IPIPE Global Initialization: \[2\]](#)
- [Subsequence – ISS ISP IPIPE 4:2:2 Conversion Module Initialization: \[3\]](#)
- [ISS ISP IPIPE Processing Path: Case 1 Configuration: \[4\]](#)
- [ISS ISP IPIPE Processing Path: Case 2 Configuration: \[5\]](#)
- [ISS ISP IPIPE Processing Path: Case 3 Configuration: \[6\]](#)
- [ISS ISP IPIPE Processing Path: Case 4 Configuration: \[7\]](#)
- [ISS IPIPE Register Summary: \[8\]](#)
- [ISS IPIPE Register Description: \[9\] \[10\] \[11\]](#)

**Table 8-651. IPIPE\_SRC\_COL**

<b>Address Offset</b>	0x0000 000C																														
<b>Physical Address</b>	See <a href="#">Table 8-644</a>								<b>Instance</b>	ISS_IPIPE																					
<b>Description</b>	RW																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								OO	OE	EO	EE												

ISS ISP

www.ti.com

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:6	OO	The color pattern of the odd line and odd pixel. This parameter is valid when IPIPE_SRC[FMT] is 0,1,2. 0x0: R 0x1: Gr 0x3: B 0x2: Gb	RW	0x3
5:4	OE	The color pattern of the odd line and even pixel. This parameter is valid when IPIPE_SRC[FMT] is 0,1,2. 0x0: R 0x1: Gr 0x3: B 0x2: Gb	RW	0x2
3:2	EO	The color pattern of the even line and odd pixel. This parameter is valid when IPIPE_SRC[FMT] is 0,1,2. 0x0: R 0x1: Gr 0x3: B 0x2: Gb	RW	0x1
1:0	EE	The color pattern of the even line and even pixel. This parameter is valid when IPIPE_SRC[FMT] is 0,1,2. 0x0: R 0x1: Gr 0x3: B 0x2: Gb	RW	0x0

**Table 8-652. Register Call Summary for Register IPIPE\_SRC\_COL**

ISS ISP

- [ISS IPIPE Register Summary: \[0\]](#)

**Table 8-653. IPIPE\_SRC\_VPS**

<b>Address Offset</b>	0x0000 0010																														
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE																													
<b>Description</b>																															
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL															
Bits	Field Name	Description	Type	Reset																											
31:16	RESERVED		R	0x0000																											
15:0	VAL	The vertical position of the global frame from the rising edge of the VD. The IPIPE module will start an image processing from VAL line.	RW	0x0000																											

**Table 8-654. Register Call Summary for Register IPIPE\_SRC\_VPS**

ISS ISP

- [ISS ISP IPIPE Input Interface: \[0\]](#)
- [Main Sequence – ISS ISP IPIPE Global Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)
- [ISS IPIPE Register Description: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

**Table 8-655. IPIPE\_SRC\_VSZ**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The vertical size of the processing area. The VAL0 can not be written. The IPIPE module will process (VAL+1) lines.	RW	0x0000

**Table 8-656. Register Call Summary for Register IPIPE\_SRC\_VSZ**

ISS ISP

- [ISS ISP IPIPE Input Interface: \[0\]](#)
- [Main Sequence – ISS ISP IPIPE Global Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-657. IPIPE\_SRC\_HPS**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	VAL	The horizontal position of the global frame from the rising edge of the HD. The IPIPE module will start an image processing from VAL clock.	RW	0x0000

**Table 8-658. Register Call Summary for Register IPIPE\_SRC\_HPS**

ISS ISP

- [ISS ISP IPIPE Input Interface: \[0\]](#)
- [Main Sequence – ISS ISP IPIPE Global Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)
- [ISS IPIPE Register Description: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

**Table 8-659. IPIPE\_SRC\_HSZ**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL											VAL_0				

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:1	VAL	The horizontal size of the processing area. The VAL0 is fixed. The IPIPE module processes (VAL+1) clocks.	RW	0x000
0	VAL_0	This is the LSB of the VAL[12:0]. This bit is read only.	R	1

**Table 8-660. Register Call Summary for Register IPIPE\_SRC\_HSZ**

ISS ISP

- [ISS ISP IPIPE Input Interface: \[0\]](#)
- [Main Sequence – ISS ISP IPIPE Global Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-661. IPIPE\_SEL\_SBU**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											EDOF				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EDOF	EDOF port selection This bit must not be enabled since the EDOF module is not implemented. This is a provision for a future revision of the IP.  0x0: Not used 0x1: Used	RW	0



**Table 8-662. Register Call Summary for Register IPIPE\_SEL\_SBU**

ISS ISP

- [ISS IPIPE Register Summary: \[0\]](#)

**Table 8-663. IPIPE\_SRC\_STA**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	IPIPE STATUS REGISTER		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											VAL4	VAL3	VAL2	VAL1	VAL0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x0000
4	VAL4	Status of Histogram Process (busy status).	R	0
3	VAL3	Status of Histogram bank select.	R	0
2	VAL2	Status of BSC process (busy status).	R	0
1	VAL1	Status of Boxcar process (busy status).	R	0
0	VAL0	Status of Boxcar process (error status). This bit will be triggered when an overflow happens while transferring the boxcar data to memory. Instead of polling for this register, it is preferable to use the IPIPE_BOXCAR_OVF interrupt. Overflow errors are non recoverable at ISP level and require a software reset at ISS level.	R	0

**Table 8-664. Register Call Summary for Register IPIPE\_SRC\_STA**

ISS ISP

- [ISS ISP IPIPE Interrupts: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-665. IPIPE\_GCK\_MMR**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											REG				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	REG	The on/off selection of the clk_arm_g0 which is used for some ARM register access. 0x0: Off 0x1: On	RW	0

**Table 8-666. Register Call Summary for Register IPIPE\_GCK\_MMR**

ISS ISP

- [Main Sequence – ISS ISP IPIPE Global Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-667. IPIPE\_GCK\_PIX**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	This register is not shadowed		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x0000
3	G3	The on/off selection of the clk_pix_g3 which is use for the IPIPE processes of EE and "CAR". 0x0: Disable 0x1: Enable	RW	0
2	G2	The on/off selection of the clk_pix_g2 which is use for the IPIPE processes of CFA to "422", "Histogram(YCbCr input)", and "Boundary Signal Calculator". 0x0: Disable 0x1: Enable	RW	0
1	G1	The on/off selection of the clk_pix_g1 which is used for the IPIPE processes of "DefectCorrection" to "WhiteBalance", and "Histogram(RAW input)". 0x0: Disable 0x1: Enable	RW	0
0	G0	The on/off selection of the clk_pix_g0 which is used for the IPIPE processing of "Boxcar". 0x0: Disable 0x1: Enable	RW	0

**Table 8-668. Register Call Summary for Register IPIPE\_GCK\_PIX**

ISS ISP

- [Main Sequence – ISS ISP IPIPE Global Initialization: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS IPIPE Register Summary: \[4\]](#)

**Table 8-669. IPIPE\_DPC\_LUT\_EN**

<b>Address Offset</b>	0x0000 0034	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											Z	EN			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	Enable of LUT defect pixel correction. 0x0: Off 0x1: On	RW	0

**Table 8-670. Register Call Summary for Register IPIPE\_DPC\_LUT\_EN**

ISS ISP

- [ISS ISP IPIPE LUT Defect Pixel Correction \(LUT DPC\): \[0\]](#)
- [Subsequence – ISS ISP IPIPE LUT Defect Pixel Correction \(LUT DPC\): \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-671. IPIPE\_DPC\_LUT\_SEL**

<b>Address Offset</b>	0x0000 0038	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											TBL	DOT			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	TBL	LUT table type selection. 0x0: Up to 1024 entries. (use <a href="#">IPIPE_DPC_LUT_SIZ</a> ) 0x1: infinity number of entries. (not use <a href="#">IPIPE_DPC_LUT_SIZ</a> )	RW	0
0	DOT	Replace dot selection on processing method 0. 0x0: Replace with black dot 0x1: Replace with white dot	RW	0

**Table 8-672. Register Call Summary for Register IPIPE\_DPC\_LUT\_SEL**

ISS ISP

- [ISS ISP IPIPE LUT Defect Pixel Correction \(LUT DPC\): \[0\] \[1\] \[2\]](#)
- [Subsequence – ISS ISP IPIPE LUT Defect Pixel Correction \(LUT DPC\): \[3\] \[4\]](#)
- [ISS IPIPE Register Summary: \[5\]](#)

**Table 8-673. IPIPE\_DPC\_LUT\_ADR**

<b>Address Offset</b>	0x0000 003C	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	RW	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								ADR															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	ADR	The address of the first valid data in look-up-table	RW	0x000

**Table 8-674. Register Call Summary for Register IPIPE\_DPC\_LUT\_ADR**

ISS ISP

- [ISS ISP IPIPE LUT Defect Pixel Correction \(LUT DPC\): \[0\]](#)
- [Subsequence – ISS ISP IPIPE LUT Defect Pixel Correction \(LUT DPC\): \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-675. IPIPE\_DPC\_LUT\_SIZ**

<b>Address Offset</b>	0x0000 0040	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	RW	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								SIZ															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	SIZ	The number of valid data in look-up-table. (SIZ+1)	RW	0x000

**Table 8-676. Register Call Summary for Register IPIPE\_DPC\_LUT\_SIZ**

ISS ISP

- [ISS ISP IPIPE LUT Defect Pixel Correction \(LUT DPC\): \[0\]](#)
- [Subsequence – ISS ISP IPIPE LUT Defect Pixel Correction \(LUT DPC\): \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)
- [ISS IPIPE Register Description: \[3\] \[4\]](#)

**Table 8-677. IPIPE\_LSC\_VOFT**

<b>Address Offset</b>	0x0000 0090	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	LSC VOFT		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			LSC_VOFT												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	LSC_VOFT		RW	0x0000

**Table 8-678. Register Call Summary for Register IPIPE\_LSC\_VOFT**

ISS ISP

- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-679. IPIPE\_LSC\_VA2**

<b>Address Offset</b>	0x0000 0094	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC VA2	RW	0x0000

**Table 8-680. Register Call Summary for Register IPIPE\_LSC\_VA2**

ISS ISP

- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-681. IPIPE\_LSC\_VA1**

<b>Address Offset</b>	0x0000 0098	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC VA1	RW	0x0000

**Table 8-682. Register Call Summary for Register IPIPE\_LSC\_VA1**

ISS ISP

- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-683. IPIPE\_LSC\_VS**

<b>Address Offset</b>	0x0000 009C	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VS2		VS1							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:4	VS2	LSC VS1	RW	0x0
3:0	VS1	LSC VS1	RW	0x0

**Table 8-684. Register Call Summary for Register IPIPE\_LSC\_VS**

ISS ISP

- [ISS IPIPE Register Summary: \[3\]](#)

**Table 8-685. IPIPE\_LSC\_HOFT**

<b>Address Offset</b>	0x0000 00A0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC HOFT	RW	0x0000

**Table 8-686. Register Call Summary for Register IPIPE\_LSC\_HOFT**

ISS ISP

- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-687. IPIPE\_LSC\_HA2**

<b>Address Offset</b>	0x0000 00A4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC HA2	RW	0x0000

**Table 8-688. Register Call Summary for Register IPIPE\_LSC\_HA2**

ISS ISP

- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-689. IPIPE\_LSC\_HA1**

<b>Address Offset</b>	0x0000 00A8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	LSC HA1	RW	0x0000

**Table 8-690. Register Call Summary for Register IPIPE\_LSC\_HA1**

ISS ISP

- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-691. IPIPE\_LSC\_HS**

<b>Address Offset</b>	0x0000 00AC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						HS2		HS1							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:4	HS2	LSC HS1	RW	0x0
3:0	HS1	LSC HS1	RW	0x0

**Table 8-692. Register Call Summary for Register IPIPE\_LSC\_HS**

ISS ISP

- [ISS IPIPE Register Summary: \[3\]](#)



**Table 8-693. IPIPE\_LSC\_GAN\_R**

<b>Address Offset</b>	0x0000 00B0	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	GAN R	RW	0x00

**Table 8-694. Register Call Summary for Register IPIPE\_LSC\_GAN\_R**

ISS ISP

- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-695. IPIPE\_LSC\_GAN\_GR**

<b>Address Offset</b>	0x0000 00B4	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	GAN GR	RW	0x00

**Table 8-696. Register Call Summary for Register IPIPE\_LSC\_GAN\_GR**

ISS ISP

- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-697. IPIPE\_LSC\_GAN\_GB**

<b>Address Offset</b>	0x0000 00B8	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

ISS ISP

www.ti.com

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	GAN GB	RW	0x00

**Table 8-698. Register Call Summary for Register IPIPE\_LSC\_GAN\_GB**

ISS ISP

- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-699. IPIPE\_LSC\_GAN\_B**

<b>Address Offset</b>	0x0000 00BC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	GAN B	RW	0x00

**Table 8-700. Register Call Summary for Register IPIPE\_LSC\_GAN\_B**

ISS ISP

- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-701. IPIPE\_LSC\_OFT\_R**

<b>Address Offset</b>	0x0000 00C0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	LSC OFT R	RW	0x00

**Table 8-702. Register Call Summary for Register IPIPE\_LSC\_OFT\_R**

ISS ISP

- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-703. IPIPE\_LSC\_OFT\_GR**

<b>Address Offset</b>	0x0000 00C4	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	LSC OFT GR	RW	0x00

**Table 8-704. Register Call Summary for Register IPIPE\_LSC\_OFT\_GR**

ISS ISP

- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-705. IPIPE\_LSC\_OFT\_GB**

<b>Address Offset</b>	0x0000 00C8	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	LSC OFT GB	RW	0x00

**Table 8-706. Register Call Summary for Register IPIPE\_LSC\_OFT\_GB**

ISS ISP

- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-707. IPIPE\_LSC\_OFT\_B**

<b>Address Offset</b>	0x0000 00CC	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

ISS ISP

www.ti.com

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	LSC OFT B	RW	0x00

**Table 8-708. Register Call Summary for Register IPIPE\_LSC\_OFT\_B**

ISS ISP

- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-709. IPIPE\_LSC\_SHF**

<b>Address Offset</b>	0x0000 00D0	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPPIPE
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x000
3:0	VAL	LSC SHV	RW	0x0

**Table 8-710. Register Call Summary for Register IPIPE\_LSC\_SHF**

ISS ISP

- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-711. IPIPE\_LSC\_MAX**

<b>Address Offset</b>	0x0000 00D4	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPPIPE
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:9	RESERVED		R	0x00
8:0	VAL	LSC MAX	RW	0x000

**Table 8-712. Register Call Summary for Register IPIPE\_LSC\_MAX**

ISS ISP

- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-713. IPIPE\_WB2\_OFT\_R**

<b>Address Offset</b>	0x0000 01D0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	White Balance Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RESERVED				VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Offset before white balance (S12) -2048 to +2047	RW	0x000

**Table 8-714. Register Call Summary for Register IPIPE\_WB2\_OFT\_R**

ISS ISP

- [ISS ISP IPIPE White Balance: \[0\]](#)
- [Subsequence – ISS ISP IPIPE White Balance Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-715. IPIPE\_WB2\_OFT\_GR**

<b>Address Offset</b>	0x0000 01D4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	White Balance Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RESERVED				VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Offset before white balance (S12) -2048 to +2047	RW	0x000

**Table 8-716. Register Call Summary for Register IPIPE\_WB2\_OFT\_GR**

ISS ISP

- [Subsequence – ISS ISP IPIPE White Balance Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-717. IPIPE\_WB2\_OFT\_GB**

<b>Address Offset</b>	0x0000 01D8		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	White Balance Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Offset before white balance (S12) -2048 to +2047	RW	0x000

**Table 8-718. Register Call Summary for Register IPIPE\_WB2\_OFT\_GB**

ISS ISP

- [Subsequence – ISS ISP IPIPE White Balance Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-719. IPIPE\_WB2\_OFT\_B**

<b>Address Offset</b>	0x0000 01DC		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	White Balance Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Offset before white balance (S12) -2048 to +2047	RW	0x000

**Table 8-720. Register Call Summary for Register IPIPE\_WB2\_OFT\_B**

ISS ISP

- [Subsequence – ISS ISP IPIPE White Balance Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-721. IPIPE\_WB2\_WGN\_R**

<b>Address Offset</b>	0x0000 01E0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	White Balance Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	White balance gain for R in U4.9 format 0 to +15.998	RW	0x0200

**Table 8-722. Register Call Summary for Register IPIPE\_WB2\_WGN\_R**

ISS ISP

- [ISS ISP IPIPE White Balance: \[0\]](#)
- [Subsequence – ISS ISP IPIPE White Balance Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-723. IPIPE\_WB2\_WGN\_GR**

<b>Address Offset</b>	0x0000 01E4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	White Balance Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	White balance gain for Gr in U4.9 format 0 to +15.998	RW	0x0200

**Table 8-724. Register Call Summary for Register IPIPE\_WB2\_WGN\_GR**

ISS ISP

- [Subsequence – ISS ISP IPIPE White Balance Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-725. IPIPE\_WB2\_WGN\_GB**

<b>Address Offset</b>	0x0000 01E8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	White Balance Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	White balance gain for Gb in U4.9 format 0 to +15.998	RW	0x0200

**Table 8-726. Register Call Summary for Register IPIPE\_WB2\_WGN\_GB**

ISS ISP

- [Subsequence – ISS ISP IPIPE White Balance Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-727. IPIPE\_WB2\_WGN\_B**

<b>Address Offset</b>	0x0000 01EC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	White Balance Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	White balance gain for B in U4.9 format 0 to +15.998	RW	0x0200

**Table 8-728. Register Call Summary for Register IPIPE\_WB2\_WGN\_B**

ISS ISP

- [Subsequence – ISS ISP IPIPE White Balance Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)



**Table 8-729. IPIPE\_RGB1\_MUL\_RR**

<b>Address Offset</b>	0x0000 022C	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient. 011111111111 = 2047/256 = 7.99609375 011111111110 = 2046/256 [...] 000011111111 = 255/256 000100000000 = 256/256 = 1 000100000001 = 257/256 [...] 000000000001 = 1/256 000000000000 = 0/256 = 0 111111111111 = -1/256 = -0.00390625 111111111110 = -2/256 [...] 100000000001 = -2047/256 100000000000 = -2048/256 = -8.	RW	0x100

**Table 8-730. Register Call Summary for Register IPIPE\_RGB1\_MUL\_RR**

ISS ISP

- [ISS ISP IPIPE RGB2RGB Blending Module: \[0\]](#)
- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-731. IPIPE\_RGB1\_MUL\_GR**

<b>Address Offset</b>	0x0000 0230	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-732. Register Call Summary for Register IPIPE\_RGB1\_MUL\_GR**

ISS ISP

- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-733. IPIPE\_RGB1\_MUL\_BR**

<b>Address Offset</b>	0x0000 0234	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-734. Register Call Summary for Register IPIPE\_RGB1\_MUL\_BR**

ISS ISP

- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-735. IPIPE\_RGB1\_MUL\_RG**

<b>Address Offset</b>	0x0000 0238	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-736. Register Call Summary for Register IPIPE\_RGB1\_MUL\_RG**

ISS ISP

- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-737. IPIPE\_RGB1\_MUL\_GG**

<b>Address Offset</b>	0x0000 023C	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x100

**Table 8-738. Register Call Summary for Register IPIPE\_RGB1\_MUL\_GG**

ISS ISP

- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-739. IPIPE\_RGB1\_MUL\_BG**

<b>Address Offset</b>	0x0000 0240	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-740. Register Call Summary for Register IPIPE\_RGB1\_MUL\_BG**

ISS ISP

- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-741. IPIPE\_RGB1\_MUL\_RB**

<b>Address Offset</b>	0x0000 0244	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-742. Register Call Summary for Register IPIPE\_RGB1\_MUL\_RB**

ISS ISP

- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-743. IPIPE\_RGB1\_MUL\_GB**

<b>Address Offset</b>	0x0000 0248		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RESERVED					VAL														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-744. Register Call Summary for Register IPIPE\_RGB1\_MUL\_GB**

ISS ISP

- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-745. IPIPE\_RGB1\_MUL\_BB**

<b>Address Offset</b>	0x0000 024C		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RESERVED					VAL														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x100

**Table 8-746. Register Call Summary for Register IPIPE\_RGB1\_MUL\_BB**

ISS ISP

- [ISS ISP IPIPE RGB2RGB Blending Module: \[0\]](#)
- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-747. IPIPE\_RGB1\_OFT\_OR**

<b>Address Offset</b>	0x0000 0250	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The output offset value for R. (s13) -4096 to +4095	RW	0x0000

**Table 8-748. Register Call Summary for Register IPIPE\_RGB1\_OFT\_OR**

ISS ISP

- [ISS ISP IPIPE RGB2RGB Blending Module: \[0\]](#)
- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-749. IPIPE\_RGB1\_OFT\_OG**

<b>Address Offset</b>	0x0000 0254	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				VAL											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The output offset value for G. (s13) -4096 to +4095	RW	0x0000

**Table 8-750. Register Call Summary for Register IPIPE\_RGB1\_OFT\_OG**

ISS ISP

- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-751. IPIPE\_RGB1\_OFT\_OB**

<b>Address Offset</b>	0x0000 0258		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	VAL	The output offset value for B. (s13) -4096 to +4095	RW	0x0000

**Table 8-752. Register Call Summary for Register IPIPE\_RGB1\_OFT\_OB**

ISS ISP

- [ISS ISP IPIPE RGB2RGB Blending Module: \[0\]](#)
- [Subsequence - ISS ISP IPIPE RGB2RGB Blending Module Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-753. IPIPE\_GMM\_CFG**

<b>Address Offset</b>	0x0000 025C		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to RGB Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						SIZ	TBL	RESERVED	BYPB	BYPG	BYPR				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:7	RESERVED		R	0x000
6:5	SIZ	The size of the gamma table. 0x0: 64 words 0x1: 128 words 0x3: 512 words 0x2: 256 words	RW	0x3
4	TBL	Selection of Gamma table. 0x0: RAM 0x1: ROM	RW	0
3	RESERVED		RW	0

Bits	Field Name	Description	Type	Reset
2	BYPB	Gamma correction mode for B 0x0: Not bypassed 0x1: Bypassed	RW	1
1	BYPG	Gamma correction mode for G 0x0: Not bypassed 0x1: Bypassed	RW	1
0	BYPR	Gamma correction mode for R 0x0: Not bypassed 0x1: Bypassed	RW	1

**Table 8-754. Register Call Summary for Register IPIPE\_GMM\_CFG**

ISS ISP

- [ISS ISP IPIPE Gamma Correction Module: \[0\] \[1\] \[2\] \[3\]](#)
- [Subsequence – ISS ISP IPIPE Gamma Correction Module Initialization: \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [ISS IPIPE Register Summary: \[9\]](#)

**Table 8-755. IPIPE\_RGB2\_MUL\_RR**

<b>Address Offset</b>	0x0000 0260	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient. 011111111111 = 2047/256 = 7.99609375 011111111110 = 2046/256 000011111111 = 255/256 000100000000 = 256/256 = 1 000100000001 = 257/256 000000000001 = 1/256 000000000000 = 0/256 = 0 111111111111 = -1/256 = -0.00390625 111111111110 = -2/256 100000000001 = -2047/256 100000000000 = -2048/256 = -8.	RW	0x100

**Table 8-756. Register Call Summary for Register IPIPE\_RGB2\_MUL\_RR**

ISS ISP

- [ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix: \[0\]](#)
- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-757. IPIPE\_RGB2\_MUL\_GR**

<b>Address Offset</b>	0x0000 0264	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-758. Register Call Summary for Register IPIPE\_RGB2\_MUL\_GR**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-759. IPIPE\_RGB2\_MUL\_BR**

<b>Address Offset</b>	0x0000 0268	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-760. Register Call Summary for Register IPIPE\_RGB2\_MUL\_BR**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-761. IPIPE\_RGB2\_MUL\_RG**

<b>Address Offset</b>	0x0000 026C	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-762. Register Call Summary for Register IPIPE\_RGB2\_MUL\_RG**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-763. IPIPE\_RGB2\_MUL\_GG**

<b>Address Offset</b>	0x0000 0270	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x100

**Table 8-764. Register Call Summary for Register IPIPE\_RGB2\_MUL\_GG**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-765. IPIPE\_RGB2\_MUL\_BG**

<b>Address Offset</b>	0x0000 0274	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-766. Register Call Summary for Register IPIPE\_RGB2\_MUL\_BG**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-767. IPIPE\_RGB2\_MUL\_RB**

<b>Address Offset</b>	0x0000 0278		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-768. Register Call Summary for Register IPIPE\_RGB2\_MUL\_RB**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-769. IPIPE\_RGB2\_MUL\_GB**

<b>Address Offset</b>	0x0000 027C		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x000

**Table 8-770. Register Call Summary for Register IPIPE\_RGB2\_MUL\_GB**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-771. IPIPE\_RGB2\_MUL\_BB**

<b>Address Offset</b>	0x0000 0280		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The matrix coefficient.	RW	0x100

**Table 8-772. Register Call Summary for Register IPIPE\_RGB2\_MUL\_BB**

ISS ISP

- [ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix: \[0\]](#)
- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-773. IPIPE\_RGB2\_OFT\_OR**

<b>Address Offset</b>	0x0000 0284	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for R S10 number: -1024 to + 1023	RW	0x000

**Table 8-774. Register Call Summary for Register IPIPE\_RGB2\_OFT\_OR**

ISS ISP

- [ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix: \[0\]](#)
- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-775. IPIPE\_RGB2\_OFT\_OG**

<b>Address Offset</b>	0x0000 0288	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for G S10 number: -1024 to + 1023	RW	0x000

**Table 8-776. Register Call Summary for Register IPIPE\_RGB2\_OFT\_OG**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-777. IPIPE\_RGB2\_OFT\_OB**

<b>Address Offset</b>	0x0000 028C		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to RGB conversion after gamma		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for B S10 number: -1024 to + 1023	RW	0x000

**Table 8-778. Register Call Summary for Register IPIPE\_RGB2\_OFT\_OB**

ISS ISP

- [ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix: \[0\]](#)
- [Subsequence – ISS ISP IPIPE 2nd RGB2RGB Conversion Matrix Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-779. IPIPE\_YUV\_ADJ**

<b>Address Offset</b>	0x0000 0294		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BRT				CRT																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	BRT	The offset value for brightness control.	RW	0x00
7:0	CRT	The multiplier coefficient value for contrast control. 00000000 = 0/16 = 0 00000001 = 1/16 00001111 = 15/16 00010000 = 16/16 = 1 00010001 = 17/16 11111110 = 254/16 11111111 = 255/16 = 15.9375	RW	0x10

**Table 8-780. Register Call Summary for Register IPIPE\_YUV\_ADJ**

ISS ISP

- [ISS ISP IPIPE RGB2YCbCr Conversion Matrix: \[0\] \[1\]](#)
- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[2\] \[3\]](#)
- [ISS IPIPE Register Summary: \[4\]](#)

**Table 8-781. IPIPE\_YUV\_MUL\_RY**

<b>Address Offset</b>	0x0000 0298		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RESERVED					VAL														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Matrix Coefficient for RY (S4.8 = -8 - +7.996)	RW	0x04D

**Table 8-782. Register Call Summary for Register IPIPE\_YUV\_MUL\_RY**

ISS ISP

- [ISS ISP IPIPE RGB2YCbCr Conversion Matrix: \[0\]](#)
- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-783. IPIPE\_YUV\_MUL\_GY**

<b>Address Offset</b>	0x0000 029C		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RESERVED					VAL														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Matrix Coefficient for GY (S4.8 = -8 - +7.996)	RW	0x096

**Table 8-784. Register Call Summary for Register IPIPE\_YUV\_MUL\_GY**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-785. IPIPE\_YUV\_MUL\_BY**

<b>Address Offset</b>	0x0000 02A0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Matrix Coefficient for BY (S4.8 = -8 - +7.996)	RW	0x01D

**Table 8-786. Register Call Summary for Register IPIPE\_YUV\_MUL\_BY**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-787. IPIPE\_YUV\_MUL\_RCB**

<b>Address Offset</b>	0x0000 02A4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0xFD5

**Table 8-788. Register Call Summary for Register IPIPE\_YUV\_MUL\_RCB**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-789. IPIPE\_YUV\_MUL\_GCB**

<b>Address Offset</b>	0x0000 02A8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0xFAB

**Table 8-790. Register Call Summary for Register IPIPE\_YUV\_MUL\_GCB**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-791. IPIPE\_YUV\_MUL\_BCB**

<b>Address Offset</b>	0x0000 02AC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x080

**Table 8-792. Register Call Summary for Register IPIPE\_YUV\_MUL\_BCB**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-793. IPIPE\_YUV\_MUL\_RCR**

<b>Address Offset</b>	0x0000 02B0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VAL									

ISS ISP

www.ti.com

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0x080

**Table 8-794. Register Call Summary for Register IPIPE\_YUV\_MUL\_RCR**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-795. IPIPE\_YUV\_MUL\_GCR**

<b>Address Offset</b>	0x0000 02B4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0xF95

**Table 8-796. Register Call Summary for Register IPIPE\_YUV\_MUL\_GCR**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-797. IPIPE\_YUV\_MUL\_BCR**

<b>Address Offset</b>	0x0000 02B8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	The matrix coefficient.	RW	0xFEB

**Table 8-798. Register Call Summary for Register IPIPE\_YUV\_MUL\_BCR**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)



**Table 8-799. IPIPE\_YUV\_OFT\_Y**

<b>Address Offset</b>	0x0000 02BC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for Y	RW	0x000

**Table 8-800. Register Call Summary for Register IPIPE\_YUV\_OFT\_Y**

ISS ISP

- [ISS ISP IPIPE RGB2YCbCr Conversion Matrix: \[0\]](#)
- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-801. IPIPE\_YUV\_OFT\_CB**

<b>Address Offset</b>	0x0000 02C0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for Cb For Cb/Cr, set (0x80 + offset value) here. (0x80 for zero offset.)	RW	0x080

**Table 8-802. Register Call Summary for Register IPIPE\_YUV\_OFT\_CB**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-803. IPIPE\_YUV\_OFT\_CR**

<b>Address Offset</b>	0x0000 02C4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	RGB to YUV Conversion Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	VAL	The output offset value for Cr For Cb/Cr, set (0x80 + offset value) here. (0x80 for zero offset.)	RW	0x080

**Table 8-804. Register Call Summary for Register IPIPE\_YUV\_OFT\_CR**

ISS ISP

- [Subsequence – ISS ISP IPIPE RGB2YCbCr Conversion Matrix Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-805. IPIPE\_YUV\_PHS**

<b>Address Offset</b>	0x0000 02C8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	YUV4:2:2 down sampling register. This register controls the YUV4:4:4 to YUV4:2:2 chroma downsampling. This register is valid if <a href="#">IPIPE_SRC_FMT.FMT = 0</a> (RAW input and YUV output). <a href="#">IPIPE_YUV_PHS = 0</a> leads to pure subsampling, no filtering, cosited chroma output. <a href="#">IPIPE_YUV_PHS = 1</a> leads to (1, 1) >> 1 filtering, centered chroma output. <a href="#">IPIPE_YUV_PHS = 2</a> leads to (1, 2, 1) >> 1 filtering, cosited chroma output. <a href="#">IPIPE_YUV_PHS = 3</a> leads to (1, 3, 3, 1) >> 3 filtering, centered chroma output. When the chroma output is cosited, and that downsampling is enabled in the RESIZER module, one need to take care that the averager disrupts the relative phase for luma and chroma color components. The <a href="#">RZA_H_PHS_ADJ</a> and <a href="#">RZB_H_PHS_ADJ</a> registers need to be used to fix the disruption.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED																LPF	POS						

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	LPF	121-LPF enable for chrominance samples. This register is valid if <a href="#">IPIPE_SRC_FMT.FMT = 0</a> (RAW input and YUV output).  0x0: off 0x1: on	RW	0

Bits	Field Name	Description	Type	Reset
0	POS	<p>This bit sets the output position of the chrominance sample with regards to the luma sample positions. One can choose between centered and cosited.</p> <p>This register is valid if <code>IPIPE_SRC_FMT.FMT = 0</code> (RAW input and YUV output).</p> <p>The RESIZER module does not change the relative position of the chroma samples vs. the luma samples between the input and output and the chroma position at the output of the IPIPE module and at the output of the RESIZER module must be identical. In other words, we must have <code>RSZ_YUV_PHS.POS = IPIPE_YUV_PHS.POS</code>.</p> <p>0x0: Cosited = same position with luminance 0x1: Centered = middle of the luminance</p>	RW	0

**Table 8-806. Register Call Summary for Register IPIPE\_YUV\_PHS**

ISS ISP

- [ISS ISP IPIPE 4:2:2 Conversion Module: \[0\] \[1\]](#)
- [ISS ISP RSZ Functional Description: \[2\]](#)
- [Subsequence – ISS ISP IPIPE 4:2:2 Conversion Module Initialization: \[3\] \[4\]](#)
- [ISS RESIZER Register Description: \[5\]](#)
- [ISS IPIPE Register Summary: \[6\]](#)
- [ISS IPIPE Register Description: \[7\] \[8\] \[9\] \[10\] \[11\]](#)

**Table 8-807. IPIPE\_YEE\_EN**

<b>Address Offset</b>	0x0000 02D4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											Z				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	<p>The on/off selection of the Edge enhancer.</p> <p>0x0: Disable 0x1: Enable</p>	RW	0

**Table 8-808. Register Call Summary for Register IPIPE\_YEE\_EN**

ISS ISP

- [ISS ISP IPIPE 2D Edge Enhancer: \[0\]](#)
- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-809. IPIPE\_YEE\_TYP**

<b>Address Offset</b>	0x0000 02D8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												HAL	SEL		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	HAL	Halo reduction in Edge Sharpener module	RW	0
0	SEL	Merging method between Edge Enhancer and Edge Sharpener 0x0: Max (EE, ES) 0x1: EE + ES	RW	0

**Table 8-810. Register Call Summary for Register IPIPE\_YEE\_TYP**

ISS ISP

- [ISS ISP IPIPE 2D Edge Enhancer: \[0\] \[1\]](#)
- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[2\]](#)
- [ISS IPIPE Register Summary: \[3\]](#)

**Table 8-811. IPIPE\_YEE\_SHF**

<b>Address Offset</b>	0x0000 02DC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												SHF			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x000
3:0	SHF	Down shift length of high pass filter (HPF) in edge enhancer.	RW	0x0

**Table 8-812. Register Call Summary for Register IPIPE\_YEE\_SHF**

ISS ISP

- [ISS ISP IPIPE 2D Edge Enhancer: \[0\]](#)
- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-813. IPIPE\_YEE\_MUL\_00**

<b>Address Offset</b>	0x0000 02E0	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	Edge Enhancer Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF. 0111111111 = 511 0111111110 = 510 0000000001 = 1 0000000000 = 0 1111111111 = -1 1000000001 = -511 1000000000 = -512	RW	0x000

**Table 8-814. Register Call Summary for Register IPIPE\_YEE\_MUL\_00**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-815. IPIPE\_YEE\_MUL\_01**

<b>Address Offset</b>	0x0000 02E4	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	Edge Enhancer Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

**Table 8-816. Register Call Summary for Register IPIPE\_YEE\_MUL\_01**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-817. IPIPE\_YEE\_MUL\_02**

<b>Address Offset</b>	0x0000 02E8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED					VAL																		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

**Table 8-818. Register Call Summary for Register IPIPE\_YEE\_MUL\_02**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-819. IPIPE\_YEE\_MUL\_10**

<b>Address Offset</b>	0x0000 02EC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED					VAL																		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

**Table 8-820. Register Call Summary for Register IPIPE\_YEE\_MUL\_10**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-821. IPIPE\_YEE\_MUL\_11**

<b>Address Offset</b>	0x0000 02F0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED					VAL																		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

**Table 8-822. Register Call Summary for Register IPIPE\_YEE\_MUL\_11**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-823. IPIPE\_YEE\_MUL\_12**

<b>Address Offset</b>	0x0000 02F4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED					VAL																		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

**Table 8-824. Register Call Summary for Register IPIPE\_YEE\_MUL\_12**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-825. IPIPE\_YEE\_MUL\_20**

<b>Address Offset</b>	0x0000 02F8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED					VAL																		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

**Table 8-826. Register Call Summary for Register IPIPE\_YEE\_MUL\_20**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-827. IPIPE\_YEE\_MUL\_21**

<b>Address Offset</b>	0x0000 02FC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

**Table 8-828. Register Call Summary for Register IPIPE\_YEE\_MUL\_21**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-829. IPIPE\_YEE\_MUL\_22**

<b>Address Offset</b>	0x0000 0300	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9:0	VAL	Multiplier coefficient in HPF.	RW	0x000

**Table 8-830. Register Call Summary for Register IPIPE\_YEE\_MUL\_22**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-831. IPIPE\_YEE\_THR**

<b>Address Offset</b>	0x0000 0304	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	Edge Enhancer lower threshold before referring to LUT. If HPF < -IPIPE_YEE_THR -> output is HPF + IPIPE_YEE_THR If HPF > IPIPE_YEE_THR -> output is HPF - IPIPE_YEE_THR Otherwise, output is zero.	RW	0x00

**Table 8-832. Register Call Summary for Register IPIPE\_YEE\_THR**

ISS ISP

- ISS ISP IPIPE 2D Edge Enhancer: [0]
- Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: [1]
- ISS IPIPE Register Summary: [2]
- ISS IPIPE Register Description: [3] [4] [5] [6]

**Table 8-833. IPIPE\_YEE\_E\_GAN**

<b>Address Offset</b>	0x0000 0308	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See Table 8-644		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Edge sharpener gain	RW	0x000

**Table 8-834. Register Call Summary for Register IPIPE\_YEE\_E\_GAN**

ISS ISP

- ISS ISP IPIPE 2D Edge Enhancer: [0]
- Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: [1]
- ISS IPIPE Register Summary: [2]

**Table 8-835. IPIPE\_YEE\_E\_THR\_1**

<b>Address Offset</b>	0x0000 030C	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See Table 8-644		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VAL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VAL	Edge sharpener HPF value lower limit	RW	0x000

**Table 8-836. Register Call Summary for Register IPIPE\_YEE\_E\_THR\_1**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-837. IPIPE\_YEE\_E\_THR\_2**

<b>Address Offset</b>	0x0000 0310	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	Edge sharpener HPF value upper limit (after 6 bit right shift)	RW	0x00

**Table 8-838. Register Call Summary for Register IPIPE\_YEE\_E\_THR\_2**

ISS ISP

- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-839. IPIPE\_YEE\_G\_GAN**

<b>Address Offset</b>	0x0000 0314	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	Edge sharpener, gain value on gradient	RW	0x00

**Table 8-840. Register Call Summary for Register IPIPE\_YEE\_G\_GAN**

ISS ISP

- [ISS ISP IPIPE 2D Edge Enhancer: \[0\]](#)
- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-841. IPIPE\_YEE\_G\_OFT**

<b>Address Offset</b>	0x0000 0318		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	Edge Enhancer Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:6	RESERVED		R	0x000
5:0	VAL	Edge sharpener, offset value on gradient	RW	0x00

**Table 8-842. Register Call Summary for Register IPIPE\_YEE\_G\_OFT**

ISS ISP

- [ISS ISP IPIPE 2D Edge Enhancer: \[0\]](#)
- [Subsequence – ISS ISP IPIPE 2D Edge Enhancer Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	VAL	The threshold of the gain function for HPF value	RW	0x00

**Table 8-843. IPIPE\_BOX\_EN**

<b>Address Offset</b>	0x0000 0380		
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b>	ISS_IPIPE
<b>Description</b>	Boxcar Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								Z															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	This bit enables or disables the BOXCAR functionality. The BOXCAR output is written to SDRAM. One need to set the <a href="#">IPIPE_BOX_SDR_SAD_H</a> and <a href="#">IPIPE_BOX_SDR_SAD_L</a> registers with the appropriate address. 0x0: Disable 0x1: Enable	RW	0

**Table 8-844. Register Call Summary for Register IPIPE\_BOX\_EN**

ISS ISP

- [ISS ISP IPIPE Boxcar: \[0\]](#)
- [Subsequence – ISS ISP IPIPE Boxcar Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-845. IPIPE\_BOX\_MODE**

<b>Address Offset</b>	0x0000 0384	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Boxcar Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											OST				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	OST	The processing mode selection of the Boxcar function. A 0 indicates the mode of the free run, a 1 indicates the mode of the one shot.  0x0: Free run 0x1: One shot	RW	0

**Table 8-846. Register Call Summary for Register IPIPE\_BOX\_MODE**

ISS ISP

- [ISS ISP IPIPE Boxcar: \[0\]](#)
- [Subsequence – ISS ISP IPIPE Boxcar Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)

**Table 8-847. IPIPE\_BOX\_TYP**

<b>Address Offset</b>	0x0000 0388	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Boxcar Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											SEL				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	SEL	Block size in boxcar sampling  0x0: 8x8 0x1: 16x16	RW	0

**Table 8-848. Register Call Summary for Register IPIPE\_BOX\_TYP**

ISS ISP

- [Subsequence – ISS ISP IPIPE Boxcar Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-849. IPIPE\_BOX\_SHF**

<b>Address Offset</b>	0x0000 038C	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Boxcar Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												VAL			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2:0	VAL	The down shift value applied to the boxcar computation result. R out = SUM (Rij) >> SHF G out = (SUM (Gr ij)/2 + SUM (Gr ij)/2) >> SHF B out = SUM (Gij) >> SHF	RW	0x0

**Table 8-850. Register Call Summary for Register IPIPE\_BOX\_SHF**

ISS ISP

- [ISS ISP IPIPE Boxcar: \[0\] \[1\]](#)
- [Subsequence – ISS ISP IPIPE Boxcar Initialization: \[2\]](#)
- [ISS IPIPE Register Summary: \[3\]](#)

**Table 8-851. IPIPE\_BOX\_SDR\_SAD\_H**

<b>Address Offset</b>	0x0000 0390	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Boxcar Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	VAL	The higher 11 bits of the first address of output in memory.	RW	0x0000

**Table 8-852. Register Call Summary for Register IPIPE\_BOX\_SDR\_SAD\_H**

ISS ISP

- [ISS ISP IPIPE Boxcar: \[0\]](#)
- [Subsequence – ISS ISP IPIPE Boxcar Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)
- [ISS IPIPE Register Description: \[3\]](#)

**Table 8-853. IPIPE\_BOX\_SDR\_SAD\_L**

<b>Address Offset</b>	0x0000 0394	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	Boxcar Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL											VAL_RESERVED				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	VAL	The lower 16 bits of the first address of output in memory.	RW	0x000
4:0	VAL_RESERVED	Ensures 32-byte alignment.	R	0x00

**Table 8-854. Register Call Summary for Register IPIPE\_BOX\_SDR\_SAD\_L**

ISS ISP

- [ISS ISP IPIPE Boxcar: \[0\]](#)
- [Subsequence – ISS ISP IPIPE Boxcar Initialization: \[1\]](#)
- [ISS IPIPE Register Summary: \[2\]](#)
- [ISS IPIPE Register Description: \[3\]](#)

**Table 8-855. IPIPE\_HST\_EN**

<b>Address Offset</b>	0x0000 039C	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	Histogram	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											Z				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	EN	This bit enables or disables the HISTOGRAM functionality. When enabled, the HISTOGRAM computation will start the processing from the next rising edge of the VD pulse. If the processing mode of the HISTOGRAM is one shot, the enable bit will be cleared to 0 immediately after the processing has started.  0x0: disable 0x1: start/busy	RW	0

**Table 8-856. Register Call Summary for Register IPIPE\_HST\_EN**

ISS ISP

- [ISS ISP IPIPE Histogram: \[0\] \[1\]](#)
- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[2\]](#)
- [ISS IPIPE Register Summary: \[3\]](#)

**Table 8-857. IPIPE\_HST\_MODE**

<b>Address Offset</b>	0x0000 03A0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												OST			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	OST	The processing mode selection of the Histogram module. A 0 indicates the mode of the free run, a 1 indicates the mode of the one shot. 0x0: Free run 0x1: One shot	RW	0

**Table 8-858. Register Call Summary for Register IPIPE\_HST\_MODE**

ISS ISP

- [ISS ISP IPIPE Histogram: \[0\] \[1\]](#)
- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[2\]](#)
- [ISS IPIPE Register Summary: \[3\]](#)

**Table 8-859. IPIPE\_HST\_SEL**

<b>Address Offset</b>	0x0000 03A4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												TES	TYP		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:3	RESERVED		R	0x0000
2	SEL	Input selection. When SEL0=0, RGBY are sampled from the output of the line buffer in noise filter-2. When SEL0=1, YCbCr are sampled at the output of RGB2YCbCr module. Y is sampled twice. 0x0: From noise filter input 0x1: From RGBtoYUV	RW	0
1:0	TYP	G selection in Bayer mode (SEL0=0) 0x0: Gb 0x1: Gr 0x3: Reserved 0x2: (Gb+Gr)/2	RW	0x0

**Table 8-860. Register Call Summary for Register IPIPE\_HST\_SEL**

ISS ISP

- [ISS ISP IPIPE Histogram: \[0\] \[1\]](#)
- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[2\] \[3\] \[4\]](#)
- [ISS IPIPE Register Summary: \[5\]](#)

**Table 8-861. IPIPE\_HST\_PARA**

<b>Address Offset</b>	0x0000 03A8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram COL0, COL1, COL2, and COL3 should be set to 1.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	BIN	SHF	COL3	COL2	COL1	COL0	RGN3	RGN2	RGN1	RGN0					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:12	BIN	The number of the bins. 0x0: 32 0x1: 64 0x3: 256 0x2: 128	RW	0x0
11:8	SHF	The shift length of the input data. data = (INPUT >> SHF)	RW	0x0
7	COL3	The on/off selection of the color pattern 3 (Y). 0x0: Disable 0x1: Enable	RW	0
6	COL2	The on/off selection of the color pattern 2 (B). 0x0: Disable 0x1: Enable	RW	0
5	COL1	The on/off selection of the color pattern 1 (G). 0x0: Disable 0x1: Enable	RW	0
4	COL0	The on/off selection of the color pattern 0 (R). 0x0: Disable 0x1: Enable	RW	0
3	RGN3	The on/off selection of the region 3. 0x0: Disable 0x1: Enable	RW	0
2	RGN2	The on/off selection of the region 2. 0x0: Disable 0x1: Enable	RW	0
1	RGN1	The on/off selection of the region 1. 0x0: Disable 0x1: Enable	RW	0



Bits	Field Name	Description	Type	Reset
0	RGN0	The on/off selection of the region 0. 0x0: Disable 0x1: Enable	RW	0

**Table 8-862. Register Call Summary for Register IPIPE\_HST\_PARA**

ISS ISP

- [ISS ISP IPIPE Histogram: \[0\] \[1\] \[2\] \[3\]](#)
- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [ISS IPIPE Register Summary: \[14\]](#)

**Table 8-863. IPIPE\_HST\_0\_VPS**

<b>Address Offset</b>	0x0000 03AC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL										VAL_RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical position of the region 0 from the <a href="#">IPIPE_SRC_VPS</a> . The region 0 will start the Histogram processing from VAL line. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The vertical position of the region 0 from the <a href="#">IPIPE_SRC_VPS</a> . The region 0 will start the Histogram processing from VAL line. VAL[0] can not be written.	R	0

**Table 8-864. Register Call Summary for Register IPIPE\_HST\_0\_VPS**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-865. IPIPE\_HST\_0\_VSZ**

<b>Address Offset</b>	0x0000 03B0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL											VAL_RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical size of the region 0. The Histogram processing of the region 0 will process (VAL+1) lines.	RW	0x000
0	VAL_RESERVED	The vertical size of the region 0. The Histogram processing of the region 0 will process (VAL+1) lines. VAL[0] cannot be written.	R Rreturns 1s	1

**Table 8-866. Register Call Summary for Register IPIPE\_HST\_0\_VSZ**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-867. IPIPE\_HST\_0\_HPS**

<b>Address Offset</b>	0x0000 03B4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL											VAL_RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal position of the region 0 from the <a href="#">IPIPE_SRC_HPS</a> . The region 0 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	RW	0x000

Bits	Field Name	Description	Type	Reset
0	VAL_RESERVED	The horizontal position of the region 0 from the <a href="#">IPIPE_SRC_HPS</a> . The region 0 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	R	0

**Table 8-868. Register Call Summary for Register IPIPE\_HST\_0\_HPS**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS IPIPE Register Summary: \[4\]](#)

**Table 8-869. IPIPE\_HST\_0\_HSZ**

<b>Address Offset</b>	0x0000 03B8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL								VAL_RESERVED					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal size of the region 0. The Histogram processing of the region 0 will process (VAL+1) clocks. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The horizontal size of the region 0. The Histogram processing of the region 0 will process (VAL+1) clocks. VAL[0] cannot be written.	R Returns 1s	1

**Table 8-870. Register Call Summary for Register IPIPE\_HST\_0\_HSZ**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-871. IPIPE\_HST\_1\_VPS**

<b>Address Offset</b>	0x0000 03BC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

ISS ISP

www.ti.com

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL										VAL_RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical position of the region 0 from the <a href="#">IPIPE_SRC_VPS</a> . The region 1 will start the Histogram processing from VAL line. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The vertical position of the region 0 from the <a href="#">IPIPE_SRC_VPS</a> . The region 1 will start the Histogram processing from VAL line. VAL[0] can not be written.	R	0

**Table 8-872. Register Call Summary for Register IPIPE\_HST\_1\_VPS**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-873. IPIPE\_HST\_1\_VSZ**

<b>Address Offset</b>	0x0000 03C0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL										VAL_RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical size of the region 1. The Histogram processing of the region 1 will process (VAL+1) lines. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The vertical size of the region 1. The Histogram processing of the region 1 will process (VAL+1) lines. VAL[0] cannot be written.	R Returns 1s	1

**Table 8-874. Register Call Summary for Register IPIPE\_HST\_1\_VSZ**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-875. IPIPE\_HST\_1\_HPS**

<b>Address Offset</b>	0x0000 03C4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL											VAL_RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal position of the region 0 from the <a href="#">IPIPE_SRC_HPS</a> . The region 1 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The horizontal position of the region 0 from the <a href="#">IPIPE_SRC_HPS</a> . The region 1 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	R	0

**Table 8-876. Register Call Summary for Register IPIPE\_HST\_1\_HPS**

ISS ISP

- [ISS IPIPE Register Summary: \[0\]](#)

**Table 8-877. IPIPE\_HST\_1\_HSZ**

<b>Address Offset</b>	0x0000 03C8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL											VAL_RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal size of the region 1. The Histogram processing of the region 1 will process (VAL+1) clocks. VAL[0] cannot be written.	RW	0x000

Bits	Field Name	Description	Type	Reset
0	VAL_RESERVED	The horizontal size of the region 1. The Histogram processing of the region 1 will process (VAL+1) clocks. VAL[0] cannot be written.	R Rreturns 1s	1

**Table 8-878. Register Call Summary for Register IPIPE\_HST\_1\_HSZ**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-879. IPIPE\_HST\_2\_VPS**

<b>Address Offset</b>	0x0000 03CC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL										VAL_RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical position of the region 0 from the <a href="#">IPIPE_SRC_VPS</a> . The region 2 will start the Histogram processing from VAL line. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The vertical position of the region 0 from the <a href="#">IPIPE_SRC_VPS</a> . The region 2 will start the Histogram processing from VAL line. VAL[0] can not be written.	R	0

**Table 8-880. Register Call Summary for Register IPIPE\_HST\_2\_VPS**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-881. IPIPE\_HST\_2\_VSZ**

<b>Address Offset</b>	0x0000 03D0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL										VAL_RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical size of the region 2. The Histogram processing of the region 2 will process (VAL+1) lines. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The vertical size of the region 2. The Histogram processing of the region 2 will process (VAL+1) lines. VAL[0] cannot be written.	R Rreturns 1s	1

**Table 8-882. Register Call Summary for Register IPIPE\_HST\_2\_VSZ**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-883. IPIPE\_HST\_2\_HPS**

<b>Address Offset</b>	0x0000 03D4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL										VAL_RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal position of the region 0 from the <a href="#">IPIPE_SRC_HPS</a> . The region 2 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The horizontal position of the region 0 from the <a href="#">IPIPE_SRC_HPS</a> . The region 2 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	R	0

**Table 8-884. Register Call Summary for Register IPIPE\_HST\_2\_HPS**

ISS ISP

- [ISS IPIPE Register Summary: \[0\]](#)

**Table 8-885. IPIPE\_HST\_2\_HSZ**

<b>Address Offset</b>	0x0000 03D8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL											VAL_RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal size of the region 2. The Histogram processing of the region 2 will process (VAL+1) clocks. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The horizontal size of the region 2. The Histogram processing of the region 2 will process (VAL+1) clocks. VAL[0] cannot be written.	R Returns 1s	1

**Table 8-886. Register Call Summary for Register IPIPE\_HST\_2\_HSZ**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-887. IPIPE\_HST\_3\_VPS**

<b>Address Offset</b>	0x0000 03DC	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VAL											VAL_RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical position of the region 0 from the <a href="#">IPIPE_SRC_VPS</a> . The region 3 will start the Histogram processing from VAL line. VAL[0] can not be written.	RW	0x000



Bits	Field Name	Description	Type	Reset
0	VAL_RESERVED	The vertical position of the region 0 from the <a href="#">IPIPE_SRC_VPS</a> . The region 3 will start the Histogram processing from VAL line. VAL[0] can not be written.	R	0

**Table 8-888. Register Call Summary for Register IPIPE\_HST\_3\_VPS**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-889. IPIPE\_HST\_3\_VSZ**

<b>Address Offset</b>	0x0000 03E0	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL										VAL_RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The vertical size of the region 3. The Histogram processing of the region 3 will process (VAL+1) lines. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The vertical size of the region 3. The Histogram processing of the region 3 will process (VAL+1) lines. VAL[0] cannot be written.	R Returns 1s	1

**Table 8-890. Register Call Summary for Register IPIPE\_HST\_3\_VSZ**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-891. IPIPE\_HST\_3\_HPS**

<b>Address Offset</b>	0x0000 03E4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

ISS ISP

www.ti.com

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL										VAL_RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal position of the region 0 from the <a href="#">IPIPE_SRC_HPS</a> . The region 3 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	RW	0x000
0	VAL_RESERVED	The horizontal position of the region 0 from the <a href="#">IPIPE_SRC_HPS</a> . The region 3 will start the Histogram processing from VAL clocks. VAL[0] can not be written.	R	0

**Table 8-892. Register Call Summary for Register IPIPE\_HST\_3\_HPS**

ISS ISP

- [ISS IPIPE Register Summary: \[0\]](#)

**Table 8-893. IPIPE\_HST\_3\_HSZ**

<b>Address Offset</b>	0x0000 03E8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VAL										VAL_RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:1	VAL	The horizontal size of the region 3. The Histogram processing of the region 3 will process (VAL+1) clocks. VAL[0] cannot be written.	RW	0x000
0	VAL_RESERVED	The horizontal size of the region 3. The Histogram processing of the region 3 will process (VAL+1) clocks. VAL[0] cannot be written.	R Rreturns 1s	1

**Table 8-894. Register Call Summary for Register IPIPE\_HST\_3\_HSZ**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-895. IPIPE\_HST\_TBL**

<b>Address Offset</b>	0x0000 03EC	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	Histogram	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												CLR	SEL		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	CLR	<p>Histogram memory clear.</p> <p>The histogram can be cleared before the start of operations. However, the clear takes 512 cycles and therefore:</p> <ul style="list-style-type: none"> <li>+ if line size &gt; 512, the first line must not be used for histogram computation.</li> <li>+ if line size &lt; 512, ceil (512/line size) lines must not be used for histogram computation.</li> </ul> <p>It's the programmer's responsibility to set the histogram computation area outside the "clear" area.</p> <p>0x0: Disable 0x1: Enable</p>	RW	0
0	SEL	<p>This bit must be used to select which memory is used to store the histogram data. By selecting alternatively one or the other bit, one can double buffer the histogram output buffer. The 4 KB memory can either be read by the CPU or a DMA request.</p> <p>0x0: Use Table 0 and 1 = 4KB in the memory ISP map. 0x1: Use Table 2 and 3 = 4KB in the memory ISP map.</p>	RW	0

**Table 8-896. Register Call Summary for Register IPIPE\_HST\_TBL**

ISS ISP

- [ISS ISP DMA Requests: \[0\] \[1\]](#)
- [ISS ISP IPIPE Histogram: \[2\] \[3\]](#)
- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[4\] \[5\]](#)
- [ISS IPIPE Register Summary: \[6\]](#)

**Table 8-897. IPIPE\_HST\_MUL\_R**

<b>Address Offset</b>	0x0000 03F0	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	Histogram	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED																RESERVED												GAIN							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	GAIN	Gain	RW	0x00

**Table 8-898. Register Call Summary for Register IPIPE\_HST\_MUL\_R**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-899. IPIPE\_HST\_MUL\_GR**

<b>Address Offset</b>	0x0000 03F4	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								GAIN															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	GAIN	Gain	RW	0x00

**Table 8-900. Register Call Summary for Register IPIPE\_HST\_MUL\_GR**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-901. IPIPE\_HST\_MUL\_GB**

<b>Address Offset</b>	0x0000 03F8	<b>Instance</b>	ISS_IPIPE
<b>Physical Address</b>	See <a href="#">Table 8-644</a>		
<b>Description</b>	Histogram		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								GAIN															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	GAIN	Gain	RW	0x00

**Table 8-902. Register Call Summary for Register IPIPE\_HST\_MUL\_GB**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**Table 8-903. IPIPE\_HST\_MUL\_B**

<b>Address Offset</b>	0x0000 03FC	
<b>Physical Address</b>	See <a href="#">Table 8-644</a>	<b>Instance</b> ISS_IPIPE
<b>Description</b>	Histogram	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								GAIN															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	GAIN	Gain	RW	0x00

**Table 8-904. Register Call Summary for Register IPIPE\_HST\_MUL\_B**

ISS ISP

- [Subsequence – ISS ISP IPIPE Histogram Initialization: \[0\]](#)
- [ISS IPIPE Register Summary: \[1\]](#)

**8.3.5.6 ISS ISIF Registers**

**CAUTION**

The ISS ISIF registers are limited to 32 bit and 16 bit data accesses; 8bit data access is not allowed and can corrupt register content.

**8.3.5.6.1 ISS ISIF Register Summary**

**Table 8-905. ISS ISIF Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISIF Base Address
<a href="#">ISIF_SYNCEN</a>	RW	32	0x0000 0000	0x5201 1000
<a href="#">ISIF_MODESET</a>	RW	32	0x0000 0004	0x5201 1004
<a href="#">ISIF_HDW</a>	RW	32	0x0000 0008	0x5201 1008
<a href="#">ISIF_VDW</a>	RW	32	0x0000 000C	0x5201 100C
<a href="#">ISIF_PPLN</a>	RW	32	0x0000 0010	0x5201 1010
<a href="#">ISIF_LPFR</a>	RW	32	0x0000 0014	0x5201 1014
<a href="#">ISIF_SPH</a>	RW	32	0x0000 0018	0x5201 1018
<a href="#">ISIF_LNH</a>	RW	32	0x0000 001C	0x5201 101C
RESERVED	RW	32	0x0000 0020	0x5201 1020
RESERVED	RW	32	0x0000 0024	0x5201 1024
<a href="#">ISIF_LNV</a>	RW	32	0x0000 0028	0x5201 1028
<a href="#">ISIF_CULH</a>	RW	32	0x0000 002C	0x5201 102C
<a href="#">ISIF_CULV</a>	RW	32	0x0000 0030	0x5201 1030
<a href="#">ISIF_HSIZE</a>	RW	32	0x0000 0034	0x5201 1034
RESERVED	RW	32	0x0000 0038	0x5201 1038
<a href="#">ISIF_CADU</a>	RW	32	0x0000 003C	0x5201 103C
<a href="#">ISIF_CADL</a>	RW	32	0x0000 0040	0x5201 1040
<a href="#">ISIF_LINCFG0</a>	RW	32	0x0000 0044	0x5201 1044
<a href="#">ISIF_LINCFG1</a>	RW	32	0x0000 0048	0x5201 1048

**Table 8-905. ISS ISIF Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISIF Base Address
ISIF_CCOLP	RW	32	0x0000 004C	0x5201 104C
ISIF_CRGAIN	RW	32	0x0000 0050	0x5201 1050
ISIF_CGRGAIN	RW	32	0x0000 0054	0x5201 1054
ISIF_CGBGAIN	RW	32	0x0000 0058	0x5201 1058
ISIF_CBGAIN	RW	32	0x0000 005C	0x5201 105C
ISIF_COFSTA	RW	32	0x0000 0060	0x5201 1060
ISIF_VDINT0	RW	32	0x0000 0070	0x5201 1070
ISIF_VDINT1	RW	32	0x0000 0074	0x5201 1074
ISIF_VDINT2	RW	32	0x0000 0078	0x5201 1078
ISIF_MISC	RW	32	0x0000 007C	0x5201 107C
ISIF_CGAMMAWD	RW	32	0x0000 0080	0x5201 1080
ISIF_REC656IF	RW	32	0x0000 0084	0x5201 1084
ISIF_CCDCFG	RW	32	0x0000 0088	0x5201 1088
ISIF_DFCCTL	RW	32	0x0000 008C	0x5201 108C
ISIF_VDFSATLV	RW	32	0x0000 0090	0x5201 1090
ISIF_DFCMEMCTL	RW	32	0x0000 0094	0x5201 1094
ISIF_DFCMEM0	RW	32	0x0000 0098	0x5201 1098
ISIF_DFCMEM1	RW	32	0x0000 009C	0x5201 109C
ISIF_DFCMEM2	RW	32	0x0000 00A0	0x5201 10A0
ISIF_DFCMEM3	RW	32	0x0000 00A4	0x5201 10A4
ISIF_DFCMEM4	RW	32	0x0000 00A8	0x5201 10A8
ISIF_CLAMPCFG	RW	32	0x0000 00AC	0x5201 10AC
ISIF_CLDCOFST	RW	32	0x0000 00B0	0x5201 10B0
ISIF_CLSV	RW	32	0x0000 00B4	0x5201 10B4
ISIF_CLHWIN0	RW	32	0x0000 00B8	0x5201 10B8
ISIF_CLHWIN1	RW	32	0x0000 00BC	0x5201 10BC
ISIF_CLHWIN2	RW	32	0x0000 00C0	0x5201 10C0
ISIF_CLVRV	RW	32	0x0000 00C4	0x5201 10C4
ISIF_CLVWIN0	RW	32	0x0000 00C8	0x5201 10C8
ISIF_CLVWIN1	RW	32	0x0000 00CC	0x5201 10CC
ISIF_CLVWIN2	RW	32	0x0000 00D0	0x5201 10D0
ISIF_CLVWIN3	RW	32	0x0000 00D4	0x5201 10D4
ISIF_LSCHOFST	RW	32	0x0000 00D8	0x5201 10D8
ISIF_LSCVOFST	RW	32	0x0000 00DC	0x5201 10DC
ISIF_LSCHVAL	RW	32	0x0000 00E0	0x5201 10E0
ISIF_LSCVVAL	RW	32	0x0000 00E4	0x5201 10E4
ISIF_2DLSCCFG	RW	32	0x0000 00E8	0x5201 10E8
ISIF_2DLSCOFST	RW	32	0x0000 00EC	0x5201 10EC
ISIF_2DLSCINI	RW	32	0x0000 00F0	0x5201 10F0
ISIF_2DLSCGRBU	RW	32	0x0000 00F4	0x5201 10F4
ISIF_2DLSCGRBL	RW	32	0x0000 00F8	0x5201 10F8
ISIF_2DLSCGROF	RW	32	0x0000 00FC	0x5201 10FC
ISIF_2DLSCORBU	RW	32	0x0000 0100	0x5201 1100
ISIF_2DLSCORBL	RW	32	0x0000 0104	0x5201 1104
ISIF_2DLSCOROF	RW	32	0x0000 0108	0x5201 1108
ISIF_2DLSCIRQEN	RW	32	0x0000 010C	0x5201 110C
ISIF_2DLSCIRQST	RW	32	0x0000 0110	0x5201 1110

**Table 8-905. ISS ISIF Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_ISIF Base Address
ISIF_FMTCFG	RW	32	0x0000 0114	0x5201 1114
ISIF_FMTPLEN	RW	32	0x0000 0118	0x5201 1118
ISIF_FMTSPH	RW	32	0x0000 011C	0x5201 111C
ISIF_FMTLNH	RW	32	0x0000 0120	0x5201 1120
ISIF_FMTLSV	RW	32	0x0000 0124	0x5201 1124
ISIF_FMTLNV	RW	32	0x0000 0128	0x5201 1128
ISIF_FMTRLN	RW	32	0x0000 012C	0x5201 112C
ISIF_FMTHCNT	RW	32	0x0000 0130	0x5201 1130
ISIF_FMTAPTR0	RW	32	0x0000 0134	0x5201 1134
ISIF_FMTAPTR1	RW	32	0x0000 0138	0x5201 1138
ISIF_FMTAPTR2	RW	32	0x0000 013C	0x5201 113C
ISIF_FMTAPTR3	RW	32	0x0000 0140	0x5201 1140
ISIF_FMTAPTR4	RW	32	0x0000 0144	0x5201 1144
ISIF_FMTAPTR5	RW	32	0x0000 0148	0x5201 1148
ISIF_FMTAPTR6	RW	32	0x0000 014C	0x5201 114C
ISIF_FMTAPTR7	RW	32	0x0000 0150	0x5201 1150
ISIF_FMTAPTR8	RW	32	0x0000 0154	0x5201 1154
ISIF_FMTAPTR9	RW	32	0x0000 0158	0x5201 1158
ISIF_FMTAPTR10	RW	32	0x0000 015C	0x5201 115C
ISIF_FMTAPTR11	RW	32	0x0000 0160	0x5201 1160
ISIF_FMTAPTR12	RW	32	0x0000 0164	0x5201 1164
ISIF_FMTAPTR13	RW	32	0x0000 0168	0x5201 1168
ISIF_FMTAPTR14	RW	32	0x0000 016C	0x5201 116C
ISIF_FMTAPTR15	RW	32	0x0000 0170	0x5201 1170
ISIF_FMTPGMV0	RW	32	0x0000 0174	0x5201 1174
ISIF_FMTPGMV1	RW	32	0x0000 0178	0x5201 1178
ISIF_FMTPGMAPU0	RW	32	0x0000 017C	0x5201 117C
ISIF_FMTPGMAPU1	RW	32	0x0000 0180	0x5201 1180
ISIF_FMTPGMAPS0	RW	32	0x0000 0184	0x5201 1184
ISIF_FMTPGMAPS1	RW	32	0x0000 0188	0x5201 1188
ISIF_FMTPGMAPS2	RW	32	0x0000 018C	0x5201 118C
ISIF_FMTPGMAPS3	RW	32	0x0000 0190	0x5201 1190
ISIF_FMTPGMAPS4	RW	32	0x0000 0194	0x5201 1194
ISIF_FMTPGMAPS5	RW	32	0x0000 0198	0x5201 1198
ISIF_FMTPGMAPS6	RW	32	0x0000 019C	0x5201 119C
ISIF_FMTPGMAPS7	RW	32	0x0000 01A0	0x5201 11A0
ISIF_CSCCTL	RW	32	0x0000 01A4	0x5201 11A4
ISIF_CSCM0	RW	32	0x0000 01A8	0x5201 11A8
ISIF_CSCM1	RW	32	0x0000 01AC	0x5201 11AC
ISIF_CSCM2	RW	32	0x0000 01B0	0x5201 11B0
ISIF_CSCM3	RW	32	0x0000 01B4	0x5201 11B4
ISIF_CSCM4	RW	32	0x0000 01B8	0x5201 11B8
ISIF_CSCM5	RW	32	0x0000 01BC	0x5201 11BC
ISIF_CSCM6	RW	32	0x0000 01C0	0x5201 11C0
ISIF_CSCM7	RW	32	0x0000 01C4	0x5201 11C4
ISIF_CLKCTL	RW	32	0x0000 01F8	0x5201 11F8

### 8.3.5.6.2 ISS ISIF Register Description

**Table 8-906. ISIF\_SYNCEN**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											DWEN	RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	DWEN	Controls the storage of image sensor RAW data in memory. This bit is loaded with the timing of the internal VD signal: it becomes active starting at the lead of the VD signal that comes after 1 is written in this bit. 0x0: Disable 0x1: Enable	RW	0
0	RESERVED	Read returns reset value	RW	0

**Table 8-907. Register Call Summary for Register ISIF\_SYNCEN**

## ISS ISP

- [ISS ISP ISIF Write Port: \[0\]](#)
- [ISS ISP ISIF Events and Status Checking: \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\] \[3\]](#)
- [ISS ISP ISIF Enable/Disable Hardware: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[10\]](#)
- [ISS ISP ISIF Interframe Operations: \[11\]](#)
- [ISS ISP ISIF Summary of Constraints: \[12\] \[13\]](#)
- [ISS ISIF Register Summary: \[14\]](#)

**Table 8-908. ISIF\_MODESET**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	HLPF	INPMOD	OVF	CCDW	CCDMD	DPOL	SWEN	RESERVED	HPOL	VPOL	FIDD	HDVDD			



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED	Read returns reset value	R	0
14	HLPF	Low pass filter enable. When this bit is enabled, a 3-tap ( $1/4 + 1/2 Z^{-2} + 1/4 Z^{-4}$ ) filtering process is performed on the sensor data.  0x0: Disable 0x1: Enable	RW	0
13:12	INPMOD	Data input mode:  0x0: RAW data 0x1: YCbCr 16bit 0x3: Reserved 0x2: YCbCr 8bit	RW	0x2
11	OVF	ISIF module write port overflow status bit If the write port of the ISIF module overflows when writing data to SDRAM, this bit will toggle.  0x0: No overflow pending (r) No action (w)  0x1: Overflow pending (r) Clear overflow (w)	RW	0
10:8	CCDW	This bit enables to shift right (divide) the up-to-12-bit RAW data value when writing out to SDRAM. The effect is that the dynamic of the output signal is decreased. The <a href="#">ISIF_MODESET.CCDW</a> , <a href="#">ISIF_HSIZE.ADCR</a> , <a href="#">ISIF_HSIZE.HSIZE</a> , <a href="#">ISIF_CCDCFG.BSWD</a> , <a href="#">ISIF_CCDCFG.MSBINV</a> , <a href="#">ISIF_CCDCFG.SDRPACK</a> bit fields control how pixel data are stored to SDRAM.  0x6: Reserved  0x1: 1-bit right shift out[15:0] = 00000 & data[11:1]  0x7: Reserved  0x0: No shift out[15:0] = 0000 & data[11:0]  0x2: 2-bit right shift out[15:0] = 000000 & data[11:2]  0x4: 4-bit right shift out[15:0] = 00000000 & data[11:4]  0x5: Reserved  0x3: 3-bit right shift out[15:0] = 0000000 & data[11:3]	RW	0x0
7	CCDMD	Field mode: This bit selects the type of image sensor: interlaced or progressive  0x0: Progressive image sensor 0x1: Interlaced image sensor	RW	0
6	DPOL	Image sensor input data polarity  0x0: No change 0x1: One's complement	RW	0
5	SWEN	External WEN selection In case this bit and SYNCEN.DWEN are set to 1, the external WEN signal is used to store image sensor data to memory.  0x0: WEN not used 0x1: Use external WEN	RW	0
4	RESERVED	Read returns reset value	RW	0

Bits	Field Name	Description	Type	Reset
3	HDPOL	HD Sync Signal Polarity 0x0: Positive 0x1: Negative	RW	0
2	VDPOL	VD Sync Signal Polarity 0x0: Positive 0x1: Negative	RW	0
1	FIDD	FLD Signal Direction. There must be at least three clock cycles between the time this bit is modified and the HD/VD pulse for the start of frame comes. 0x0: Input 0x1: Output	RW	0
0	HDVDD	VD,HD Sync Signal Direction. There must be at least three clock cycles between the time this bit is modified and the HD/VD pulse for the start of frame comes. 0x0: Input 0x1: Output	RW	0

**Table 8-909. Register Call Summary for Register ISIF\_MODESET**

## ISS ISP

- [ISS ISP ISIF Interrupts: \[0\]](#)
- [ISS ISP VP Pixel Clock Inversion: \[1\] \[2\]](#)
- [ISS ISP ISIF Input Interface: \[3\] \[4\]](#)
- [ISS ISP ISIF Interface: \[5\] \[6\]](#)
- [ISS ISP ISIF Low-Pass Filter \(LPF\): \[7\] \[8\]](#)
- [ISP ISIF Storage Formatter: \[9\] \[10\]](#)
- [ISS ISP ISIF VDINT0, VDINT1, and VDINT2 Interrupts: \[11\] \[12\] \[13\]](#)
- [ISS ISP ISIF Status Checking: \[14\]](#)
- [ISS ISP ISIF Register Setup: \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\]](#)
- [ISS ISP ISIF Enable/Disable Hardware: \[26\] \[27\] \[28\] \[29\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[30\]](#)
- [ISS ISIF Register Summary: \[31\]](#)
- [ISS ISIF Register Description: \[32\] \[33\] \[34\] \[35\] \[36\]](#)

**Table 8-910. ISIF\_HDW**

<b>Address Offset</b>	0x0000 0008		<b>Instance</b>	ISS_ISIF																											
<b>Physical Address</b>	See <a href="#">Table 8-905</a>																														
<b>Description</b>																															
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			HDW												
<b>Bits</b>	<b>Field Name</b>		<b>Description</b>													<b>Type</b>	<b>Reset</b>														
31:16	RESERVED															R	0x0000														
15:12	RESERVED															R	0x0														
11:0	HDW		HD width: Sets width of HD. HD width = HDW + 1 clock													RW	0x000														

**Table 8-911. Register Call Summary for Register ISIF\_HDW**

ISS ISP

- [ISS ISP ISIF Clamp Value for Horizontal Direction: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-912. ISIF\_VDW**

<b>Address Offset</b>	0x0000 000C	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						VDW									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VDW	VD width : Sets width of VD. VD width = VDW + 1 line	RW	0x000

**Table 8-913. Register Call Summary for Register ISIF\_VDW**

ISS ISP

- [ISS ISP ISIF Clamp Value for Vertical Direction: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-914. ISIF\_PPLN**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PPLN															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	PPLN	Pixels per line Number of pixel clock periods in one line HD period = PPLN+1 pixel clocks. PPLN is not used when HD and VD are inputs, that is, when VDHDOUT in MODESET is cleared to 0. *This bit field is latched by VD.	RW	0x0000

**Table 8-915. Register Call Summary for Register ISIF\_PPLN**

ISS ISP

- [ISS ISP ISIF Black Clamp: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[3\]](#)
- [ISS ISIF Register Summary: \[4\]](#)

**Table 8-916. ISIF\_LPFR**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	Line per Frame/Field		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LPFR															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	LPFR	Half lines per filed or frame Sets number of half lines per frame or field. VD period = (LPFR+1)/2 lines. LPFR is not used when HD and are inputs, that is, when VDHDOUT in MODESET is cleared to 0. *This bit field is latched by VD.	RW	0x0000

**Table 8-917. Register Call Summary for Register ISIF\_LPFR**

ISS ISP

- [ISS ISP ISIF Vertical Line Defect Correction \(VDFC\): \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-918. ISIF\_SPH**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	Start Pixel Horizontal		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	SPH														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	SPH	The first pixel in a line to be stored to memory.	RW	0x0000

**Table 8-919. Register Call Summary for Register ISIF\_SPH**

ISS ISP

- [ISS ISP ISIF Formatter Area Settings: \[0\]](#)
- [ISP ISIF Storage Formatter: \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[3\]](#)
- [ISS ISP ISIF Summary of Constraints: \[4\] \[5\]](#)
- [ISS ISIF Register Summary: \[6\]](#)

**Table 8-920. ISIF\_LNH**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	LNH														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	LNH	Number of pixels in an line to be stored to memory. Number of pixels = LNH + 1.	RW	0x0000

**Table 8-921. Register Call Summary for Register ISIF\_LNH**

ISS ISP

- [ISS ISP ISIF Formatter Area Settings: \[0\]](#)
- [ISP ISIF Storage Formatter: \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[3\]](#)
- [ISS ISP ISIF Summary of Constraints: \[4\] \[5\] \[6\]](#)
- [ISS ISIF Register Summary: \[7\]](#)

**Table 8-922. ISIF\_LNV**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	LNV														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED	The number of lines to be stored to SDRAM.	R	0
14:0	LNV	The number of lines to be stored to memory. Number of lines = LNV + 1	RW	0x0000

**Table 8-923. Register Call Summary for Register ISIF\_LNV**

## ISS ISP

- [ISS ISP ISIF Formatter Area Settings: \[0\]](#)
- [ISP ISIF Storage Formatter: \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISP ISIF Summary of Constraints: \[3\] \[4\]](#)
- [ISS ISIF Register Summary: \[5\]](#)

**Table 8-924. ISIF\_CULH**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLHO								CLHE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CLHO	Culling Pattern in ODD Line: Sets culling pattern when data is loaded into memory (odd lines). Example: 0xAA: 1 / 2 horizontal direction culling. LSB becomes left side on screen.  0x0: Pixel invalid 0x1: Pixel valid	RW	0xFF
7:0	CLHE	Culling Pattern in Even Line: Sets culling pattern when data is loaded into memory (even lines).  0x0: Pixel invalid 0x1: Pixel valid	RW	0xFF

**Table 8-925. Register Call Summary for Register ISIF\_CULH**

## ISS ISP

- [ISS ISP ISIF Culling: \[0\] \[1\] \[2\]](#)
- [ISS ISP ISIF Register Setup: \[3\] \[4\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[5\]](#)
- [ISS ISIF Register Summary: \[6\]](#)

**Table 8-926. ISIF\_CULV**

<b>Address Offset</b>	0x0000 0030	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	RW	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								CULV															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	CULV	Culling Pattern in Vertical Line Example: 0x88: 1/4 vertical direction culling. LSB becomes top side on screen.  0x0: Pixel invalid 0x1: Pixel valid	RW	0xFF

**Table 8-927. Register Call Summary for Register ISIF\_CULV**

ISS ISP

- [ISS ISP ISIF Culling: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-928. ISIF\_HSIZE**

<b>Address Offset</b>	0x0000 0034	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	SDRAM OUTPUT CTRL REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED	ADCR	HSIZE																					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12	ADCR	SDRAM address decrement. By setting this bit, memory address in a line is automatically decreased so that a line can be horizontally flipped in memory. The <a href="#">ISIF_MODESET.CCDW</a> , <a href="#">ISIF_HSIZE.ADCR</a> , <a href="#">ISIF_HSIZE.HSIZE</a> , <a href="#">ISIF_CCDCFG.BSWD</a> , <a href="#">ISIF_CCDCFG.MSBINV</a> , <a href="#">ISIF_CCDCFG.SDRPACK</a> bit fields control how pixel data are stored to SDRAM.  0x0: Address increment. 0x1: Address decrement.	RW	0
11:0	HSIZE	Memory address offset between the lines. Specify the offset in 32-byte units.	RW	0x000

**Table 8-929. Register Call Summary for Register ISIF\_HSIZE**

ISS ISP

- [ISP ISIF Storage Formatter: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\] \[3\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[4\]](#)
- [ISS ISIF Register Summary: \[5\]](#)
- [ISS ISIF Register Description: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

**Table 8-930. ISIF\_CADU**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	SDRAM OUTPUT CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				CADU																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	CADU	Memory Address (Upper 11-bits): Specifies the memory destination address. The actual address is the value set here multiplied by 32bytes.	RW	0x000

**Table 8-931. Register Call Summary for Register ISIF\_CADU**

ISS ISP

- [ISP ISIF Storage Formatter: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Enable/Disable Hardware: \[2\] \[3\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[4\]](#)
- [ISS ISIF Register Summary: \[5\]](#)

**Table 8-932. ISIF\_CADL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	SDRAM OUTPUT CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CADL																							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	CADL	Memory Address (Lower 16-bits): Specifies the memory destination address. The actual address is the value set here multiplied by 32bytes.	RW	0x0000



**Table 8-933. Register Call Summary for Register ISIF\_CADL**

ISS ISP

- [ISP ISIF Storage Formatter: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Enable/Disable Hardware: \[2\] \[3\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[4\]](#)
- [ISS ISIF Register Summary: \[5\]](#)

**Table 8-934. ISIF\_LINCFG0**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	INPUT LINEARIZATION CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED						CORRSFT		RESERVED	LINMD	LINEN					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:7	RESERVED		R	0x000
6:4	CORRSFT	Shift up value for the correction value (S10). 0x6: 6-bit left shift 0x1: 1-bit left shift 0x7: Reserved 0x0: No shift 0x2: 2-bit left shift 0x4: 4-bit left shift 0x5: 5-bit left shift 0x3: 3-bit left shift	RW	0x0
3:2	RESERVED		R	0x0
1	LINMD	Linearization Mode: 0x0: Uniform sampling 0x1: Non-uniform sampling	RW	0
0	LINEN	Linearization Enable: 0x0: Disable 0x1: Enable	RW	0

**Table 8-935. Register Call Summary for Register ISIF\_LINCFG0**

ISS ISP

- [ISS ISP ISIF Sensor Linearization: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS ISP ISIF Register Setup: \[4\] \[5\] \[6\]](#)
- [ISS ISIF Register Summary: \[7\]](#)

**Table 8-936. ISIF\_LINCFG1**

<b>Address Offset</b>	0x0000 0048																				
<b>Physical Address</b>	See <a href="#">Table 8-905</a>											<b>Instance</b>					ISS_ISIF				
<b>Description</b>	INPUT LINEARIZATION CTRL REGISTER																				
<b>Type</b>	RW																				

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				LUTSCL																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:0	LUTSCL	Scale factor (U11Q10) for LUT input. Range: 0 - 1+1023/1024 It is applied to the Input Data before looking up the correction factor. The scale factor is only applied to the table input. It is not applied when using the input value to compute the output.	RW	0x400

**Table 8-937. Register Call Summary for Register ISIF\_LINCFG1**

ISS ISP

- [ISS ISP ISIF Sensor Linearization: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP Register Summary: \[2\]](#)

**Table 8-938. ISIF\_CCOLP**

<b>Address Offset</b>	0x0000 004C																				
<b>Physical Address</b>	See <a href="#">Table 8-905</a>											<b>Instance</b>					ISS_ISIF				
<b>Description</b>																					
<b>Type</b>	RW																				

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED								CP0_F1				CP1_F1				CP2_F1				CP3_F1				CP0_F0				CP1_F0				CP2_F0				CP3_F0			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	CP0_F1	Specifies color pattern for pixel position 0 (Field 1) Pixel position 0 corresponds to pixel count=0 at even line in case of CFAP= 0, and to pixel count=0 in case of CFAP= 1.  0x0: R / Ye 0x1: Gr / Cy 0x3: B / Mg 0x2: Gb / G	RW	0x0

Bits	Field Name	Description	Type	Reset
13:12	CP1_F1	Specifies color pattern for pixel position 1 (Field 1) Pixel position 1 corresponds to pixel count=1 at even line in case of CFAP= 0, and to pixel count=1 in case of CFAP= 1.  0x0: R / Ye 0x1: Gr / Cy 0x3: B / Mg 0x2: Gb / G	RW	0x0
11:10	CP2_F1	Specifies color pattern for pixel position 2 (Field 1) Pixel position 2 corresponds to pixel count=0 at odd line in case of CFAP= 0, and to pixel count=2 in case of CFAP= 1.  0x0: R / Ye 0x1: Gr / Cy 0x3: B / Mg 0x2: Gb / G	RW	0x0
9:8	CP3_F1	Specifies color pattern for pixel position 3 (Field 1) Pixel position 3 corresponds to pixel count=1 at odd line in case of CFAP= 0. Not applicable for CFAP= 1.  0x0: R / Ye 0x1: Gr / Cy 0x3: B / Mg 0x2: Gb / G	RW	0x0
7:6	CP0_F0	Specifies color pattern for pixel position 0 (Field 0) Pixel position 0 corresponds to pixel count=0 at even line in case of CFAP= 0, and to pixel count=0 in case of CFAP= 1.  0x0: R / Ye 0x1: Gr / Cy 0x3: B / Mg 0x2: Gb / G	RW	0x0
5:4	CP1_F0	Specifies color pattern for pixel position 1 (Field 0) Pixel position 1 corresponds to pixel count=1 at even line in case of CFAP= 0, and to pixel count=1 in case of CFAP= 1.  0x0: R / Ye 0x1: Gr / Cy 0x3: B / Mg 0x2: Gb / G	RW	0x0
3:2	CP2_F0	Specifies color pattern for pixel position 2 (Field 0) Pixel position 2 corresponds to pixel count=0 at odd line in case of CFAP= 0, and to pixel count=2 in case of CFAP= 1.  0x0: R / Ye 0x1: Gr / Cy 0x3: B / Mg 0x2: Gb / G	RW	0x0
1:0	CP3_F0	Specifies color pattern for pixel position 3 (Field 0) Pixel position 3 corresponds to pixel count=1 at odd line in case of CFAP= 0. Not applicable for CFAP= 1.  0x0: R / Ye 0x1: Gr / Cy 0x3: B / Mg 0x2: Gb / G	RW	0x0

**Table 8-939. Register Call Summary for Register ISIF\_CCOLP**

ISS ISP

- ISS ISP ISIF White Balance: [0]
- ISS ISP ISIF Register Setup: [1] [2] [3] [4]
- ISS ISP ISIF Register Accessibility During Frame Processing: [5]
- ISS ISIF Register Summary: [6]

**Table 8-940. ISIF\_CRGAIN**

<b>Address Offset</b>	0x0000 0050		<b>Instance</b>	ISS_ISIF																											
<b>Physical Address</b>	See <a href="#">Table 8-905</a>																														
<b>Description</b>																															
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				CGR																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>														<b>Type</b>	<b>Reset</b>														
31:16	RESERVED															R	0x0000														
15:12	RESERVED															R	0x0														
11:0	CGR	R/Ye gain: Performs gain adjustment on image sensor data. U12Q9. Range: 0 - 7+511/512														RW	0x200														

**Table 8-941. Register Call Summary for Register ISIF\_CRGAIN**

ISS ISP

- ISS ISP ISIF White Balance: [0]
- ISS ISP ISIF Register Setup: [1]
- ISS ISP ISIF Register Accessibility During Frame Processing: [2]
- ISS ISIF Register Summary: [3]

**Table 8-942. ISIF\_CGRGAIN**

<b>Address Offset</b>	0x0000 0054		<b>Instance</b>	ISS_ISIF																											
<b>Physical Address</b>	See <a href="#">Table 8-905</a>																														
<b>Description</b>																															
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				CGGR																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>														<b>Type</b>	<b>Reset</b>														
31:16	RESERVED															R	0x0000														
15:12	RESERVED															R	0x0														
11:0	CGGR	Gr/Cy gain: Performs gain adjustment on image sensor data. U12Q9. Range: 0 - 7+511/512														RW	0x200														

**Table 8-943. Register Call Summary for Register ISIF\_CGRGAIN**

ISS ISP

- [ISS ISP ISIF White Balance: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-944. ISIF\_CGBGAIN**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				CGGB																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	CGGB	Gb/Cy gain: Performs gain adjustment on image sensor data. U12Q9. Range: 0 - 7+511/512	RW	0x200

**Table 8-945. Register Call Summary for Register ISIF\_CGBGAIN**

ISS ISP

- [ISS ISP ISIF White Balance: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-946. ISIF\_CBGAIN**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				CGB																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	CGB	B/Mg gain: Performs gain adjustment on image sensor data. U12Q9. Range: 0 - 7+511/512	RW	0x200

**Table 8-947. Register Call Summary for Register ISIF\_CBGAIN**

ISS ISP

- [ISS ISP ISIF White Balance: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-948. ISIF\_COFSTA**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				COFT											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	COFT	Image sensor offset: Performs offset value adjustment on image sensor data (0~4095).	RW	0x000

**Table 8-949. Register Call Summary for Register ISIF\_COFSTA**

ISS ISP

- [ISS ISP ISIF White Balance: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-950. ISIF\_VDINT0**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	CVD0														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	CVD0	VD0 Interrupt timing in a field (line number).	RW	0x0000

**Table 8-951. Register Call Summary for Register ISIF\_VDINT0**

ISS ISP

- [ISS ISP ISIF VDINT0, VDINT1, and VDINT2 Interrupts: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-952. ISIF\_VDINT1**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	CVD1														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	CVD1	VD1 Interrupt timing in a field (line number).	RW	0x0000

**Table 8-953. Register Call Summary for Register ISIF\_VDINT1**

ISS ISP

- [ISS ISP ISIF VDINT0, VDINT1, and VDINT2 Interrupts: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-954. ISIF\_VDINT2**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	CVD2														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:0	CVD2	VD2 Interrupt timing in a field (line number).	RW	0x0000

**Table 8-955. Register Call Summary for Register ISIF\_VDINT2**

ISS ISP

- [ISS ISP ISIF VDINT0, VDINT1, and VDINT2 Interrupts: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-956. ISIF\_MISC**

<b>Address Offset</b>	0x0000 007C	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	DPCMPRE	DPCMEN	RESERVED										RESERVED		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13	DPCMPRE	Selects Predictor for DPCM Encoder (12-8) 0x0: Predictor 1 0x1: Predictor 2	RW	0
12	DPCMEN	Enables DPCM Encoding (12-8) 0x0: Disable 0x1: Enable	RW	0
11:1	RESERVED		R	0x000
0	RESERVED		RW	0

**Table 8-957. Register Call Summary for Register ISIF\_MISC**

## ISS ISP

- [ISS ISP ISIF 12-to-8 bit DPCM Compression Block: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\] \[3\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[4\]](#)
- [ISS ISIF Register Summary: \[5\]](#)

**Table 8-958. ISIF\_CGAMMAWD**

<b>Address Offset</b>	0x0000 0080	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	WBEN2	WBEN1	WBEN0	RESERVED	OFSTEN2	OFSTEN1	OFSTEN0	RESERVED	CFAP	GWDI			CCDTBL		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14	WBEN2	White Balance Enable for H3A 0x0: Disable 0x1: Enable	RW	0



Bits	Field Name	Description	Type	Reset
13	WBEN1	White Balance Enable for IPIPE 0x0: Disable 0x1: Enable	RW	0
12	WBEN0	White Balance Enable for memory capture 0x0: Disable 0x1: Enable	RW	0
11	RESERVED		R	0
10	OFSTEN2	Offset control Enable for H3A 0x0: Disable 0x1: Enable	RW	0
9	OFSTEN1	Offset control Enable for IPIPE 0x0: Disable 0x1: Enable	RW	0
8	OFSTEN0	Offset control Enable for SDRAM capture 0x0: Disable 0x1: Enable	RW	0
7:6	RESERVED		R	0x0
5	CFAP	Selects CFA pattern 0x0: Mosaic color pattern. It should look like this. G R G R G R G R . . . B G B G B G B G . . . G R G R G R G R . . . . . . . . 0x1: Stripe color pattern. It should look like this. R G B R G B R G B . . . R G B R G B R G B . . . R G B R G B R G B . . . . . . . . . . . . .	RW	0
4:1	GWDI	Selects MSB position of Input Data 0x6: bit 9 0x1: bit 14 0xA: Reserved 0x7: bit 8 0xD: Reserved 0x0: bit 15 0x2: bit 13 0x8: bit 7 0x9: Reserved 0xB: Reserved 0x4: bit 11 0x5: bit 10 0xF: Reserved 0xC: Reserved 0x3: bit 12 0xE: Reserved	RW	0x0
0	CCDTBL	On/Off control of A-law table for SDRAM capture 0x0: Disable 0x1: Enable	RW	0

**Table 8-959. Register Call Summary for Register ISIF\_CGAMMAWD**

ISS ISP

- [ISS ISP ISIF Interface: \[0\] \[1\]](#)
- [ISS ISP ISIF White Balance: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [ISS ISP ISIF A-Law Compression: \[9\] \[10\] \[11\]](#)
- [ISS ISP ISIF Register Setup: \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[21\]](#)
- [ISS ISIF Register Summary: \[22\]](#)

**Table 8-960. ISIF\_REC656IF**

<b>Address Offset</b>	0x0000 0084	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	INPUT CONFIG REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												R656ON	ECCFVH		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	R656ON	CCIR Rec.656 interface mode 0x0: Disable 0x1: Enable	RW	0
0	ECCFVH	Error correction of FVH code 0x0: Disable 0x1: Enable	RW	0

**Table 8-961. Register Call Summary for Register ISIF\_REC656IF**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\] \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)
- [ISS ISIF Register Description: \[3\]](#)

**Table 8-962. ISIF\_CCDCFG**

<b>Address Offset</b>	0x0000 0088	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VLDC	RESERVED	MSBINVI	BSWD	Y8POS	EXTRG	TRGSEL	WENLOG	FIDMD	BT656	YCINSWP	RESERVED	RESERVED	SDRPACK		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	VLDC	On/off control of CPU registers resynchronize function by VSYNC. All the others are shadowed registers, where register values are updated at V-sync timing by default. If VDLC=1, ISIF register values are updated immediately after register write just like non-shadowed registers. 0x0: Enable 0x1: Disable	RW	0
14	RESERVED	Reserved. must always be set to 0.	RW	0
13	MSBINVI	MSB inverse of CIN port when the data are captured to SDRAM. The <a href="#">ISIF_MODESET.CCDW</a> , <a href="#">ISIF_HSIZE.ADCR</a> , <a href="#">ISIF_HSIZE.HSIZE</a> , <a href="#">ISIF_CCDCFG.BSWD</a> , <a href="#">ISIF_CCDCFG.MSBINV</a> , <a href="#">ISIF_CCDCFG.SDRPACK</a> bit fields control how pixel data are stored to SDRAM. 0x0: Disable 0x1: Enable	RW	0
12	BSWD	On/off control of Byte SWAP function when SDRAM capturing. The <a href="#">ISIF_MODESET.CCDW</a> , <a href="#">ISIF_HSIZE.ADCR</a> , <a href="#">ISIF_HSIZE.HSIZE</a> , <a href="#">ISIF_CCDCFG.BSWD</a> , <a href="#">ISIF_CCDCFG.MSBINV</a> , <a href="#">ISIF_CCDCFG.SDRPACK</a> bit fields control how pixel data are stored to SDRAM. 0x0: Disable 0x1: Enable (swap)	RW	0
11	Y8POS	Selects Y signal position when in 8bit input mode 0x0: even pixel 0x1: odd pixel	RW	0
10	EXTRG	Setting 1 to this register, the SDRAM address is initialized at the rising edge of FID input signal or DWEN register.	RW	0
9	TRGSEL	Select trigger source signal of SDRAM address initializing in case EXTRG=1. 0x0: DWEN register 0x1: FID input port	RW	0
8	WENLOG	Specifies the CCD valid area. 0x0: internal valid signal and WEN signal is ANDed logically. 0x1: internal valid signal and WEN signal is ORed logically.	RW	0
7:6	FIDMD	Specifies FID detection mode 0x0: latch the FID at the VSYNC timing 0x1: no latch the FID 0x3: Reserved 0x2: Reserved	RW	0x0
5	BT656	Selects bit width of CCIR656. This bit applies only if <a href="#">ISIF_REC656IF.R656ON</a> = 1. 0x0: 8 bits 0x1: 10 bits	RW	0

Bits	Field Name	Description	Type	Reset
4	YCINSWP	The ISIF module has a 16-bit interface. When 16-bit YUV data are input, the luma data (YIN7-0) are expected to be on the 8 MS bits and the chroma (CIN7-0) data are expected to be on the LS bits. This bit enables to swap the 8 MS bits with the 8 LS bits of the interface in case the luma and chroma do not come in the correct order. See the top-level ISIF block diagram.  0x0: YIN7-0 = Y signal / CIN7-0 = C signal 0x1: YIN7-0 = C signal / CIN7-0 = Y signal	RW	0
3	RESERVED	Reserved. must always be set to 0.	RW	0
2	RESERVED	Reserved. must always be set to 0.	RW	0
1:0	SDRPACK	This bit field selects how the data are stored to SDRAM. There can be 8, 12 or 16 bits per pixel. The <a href="#">ISIF_MODESET.CCDW</a> , <a href="#">ISIF_HSIZE.ADCR</a> , <a href="#">ISIF_HSIZE.HSIZE</a> , <a href="#">ISIF_CCDCFG.BSWD</a> , <a href="#">ISIF_CCDCFG.MSBINV</a> , <a href="#">ISIF_CCDCFG.SDRPACK</a> bit fields control how pixel data are stored to SDRAM.  0x0: 16 bits / pixel 0x1: 12 bits / pixel 0x3: Reserved 0x2: 8 bits / pixel	RW	0x0

**Table 8-963. Register Call Summary for Register ISIF\_CCDCFG**

## ISS ISP

- [ISS ISP ISIF Interface: \[0\] \[1\] \[2\]](#)
- [ISS ISP ISIF Write Port: \[3\]](#)
- [ISS ISP ISIF Register Setup: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[14\] \[15\]](#)
- [ISS ISP ISIF Summary of Constraints: \[16\]](#)
- [ISS ISIF Register Summary: \[17\]](#)
- [ISS ISIF Register Description: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\]](#)

**Table 8-964. ISIF\_DFCCTL**

<b>Address Offset</b>	0x0000 008C	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	VERTICAL LINE DEFECT CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			VDFLSFT	VDFCUDA	VDFCSL	VDFCEN	RESERVED								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:11	RESERVED		R	0x00
10:8	VDFLSFT	Vertical line Defect level shift value Defect Level (value to be subtracted from the data) is 8bit width, but can be up-shifted up to 6bits by VDFLSFT. Left shift value = VDFLSFT (Range: 0-6) Setting 7 to VDFLSFT is not allowed.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	VDFCUDA	Vertical line Defect Correction upper pixels disable. 0x0: The whole line is corrected. 0x1: Pixels upper than the defect are not corrected.	RW	0
6:5	VDFCSL	Vertical line Defect Correction mode select. 0x0: Defect level subtraction. Just fed through if data are saturating. 0x1: Defect level subtraction. Horizontal interpolation ((i-2)+(i+2))/2 if data are saturating. 0x3: Reserved 0x2: Horizontal interpolation ((i-2)+(i+2))/2.	RW	0x0
4	VDFCEN	Vertical line Defect Correction enable. This bit field is latched by VD. 0x0: Disable 0x1: Enable	RW	0
3:0	RESERVED		R	0x0

**Table 8-965. Register Call Summary for Register ISIF\_DFCCTL**

ISS ISP

- [ISS ISP ISIF Vertical Line Defect Correction \(VDFC\): \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [ISS ISP ISIF Vertical Line Defect Table Update Procedure: \[9\] \[10\]](#)
- [ISS ISP ISIF Register Setup: \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[16\]](#)
- [ISS ISIF Register Summary: \[17\]](#)

**Table 8-966. ISIF\_VDFSATLV**

<b>Address Offset</b>	0x0000 0090	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	VERTICAL LINE DEFCT CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				VDFSALV																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	VDFSALV	Vertical line Defect Correction saturation level. VDFSALV is U12 (Range: 0 - 4,095).	RW	0x000

**Table 8-967. Register Call Summary for Register ISIF\_VDFSATLV**

ISS ISP

- [ISS ISP ISIF Vertical Line Defect Correction \(VDFC\): \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-968. ISIF\_DFCMEMCTL**

<b>Address Offset</b>	0x0000 0094	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	VERTICAL LINE DEFCT CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																RESERVED												DFCMCLR	RESERVED	DFCMARST	DFCMRD	DFCMWR

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4	DFCMCLR	Defect correction. Memory clear. Writing 1 to this bit clears the memory contents to all zero. It will be automatically cleared to '0' when the memory clear is completed.	RW	0
3	RESERVED		R	0
2	DFCMARST	Defect correction. Memory address reset. Setting DFCMWR or DFCMRD with LSCMARST set starts memory access to address offset 0. DFCMARST is automatically cleared if data transfer completes. Setting DFCMWR or DFCMRD with LSCMARST cleared starts memory access to the next address. 0x0: Increment the memory address 0x1: Clear the memory address to offset 0	RW	0
1	DFCMRD	Defect correction. Memory read [for debug purpose] Writing 1 to this bit starts reading from the memory. It will be automatically cleared when the data transfer is completed, and the data can be read from DFCMEM4-0.	RW	0
0	DFCMWR	Defect correction. Memory write Writing 1 to this bit starts writing to the memory. It will be automatically cleared when the data transfer is completed. DFCMEM4-0 should be set prior to the memory access.	RW	0

**Table 8-969. Register Call Summary for Register ISIF\_DFCMEMCTL**

ISS ISP

- [ISS ISP ISIF Vertical Line Defect Table Update Procedure: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [ISS ISP ISIF Register Setup: \[8\] \[9\] \[10\]](#)
- [ISS ISIF Register Summary: \[11\]](#)

**Table 8-970. ISIF\_DFCMEM0**

<b>Address Offset</b>	0x0000 0098	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	Defect correction memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		DFCMEM0													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	DFCMEM0	Defect correction memory 0 Sets V position of the defects.	RW	0x0000

**Table 8-971. Register Call Summary for Register ISIF\_DFCMEM0**

ISS ISP

- [ISS ISP ISIF Vertical Line Defect Correction \(VDLC\): \[0\]](#)
- [ISS ISP ISIF Vertical Line Defect Table Update Procedure: \[1\] \[2\] \[3\]](#)
- [ISS ISP ISIF Register Setup: \[4\]](#)
- [ISS ISIF Register Summary: \[5\]](#)

**Table 8-972. ISIF\_DFCMEM1**

<b>Address Offset</b>	0x0000 009C	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	Defect correction memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		DFCMEM1													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	DFCMEM1	Defect correction memory 1 Sets H position of the defects.	RW	0x0000

**Table 8-973. Register Call Summary for Register ISIF\_DFCMEM1**

ISS ISP

- [ISS ISP ISIF Vertical Line Defect Correction \(VDLC\): \[0\]](#)
- [ISS ISP ISIF Vertical Line Defect Table Update Procedure: \[1\] \[2\] \[3\]](#)
- [ISS ISP ISIF Register Setup: \[4\]](#)
- [ISS ISIF Register Summary: \[5\]](#)

**Table 8-974. ISIF\_DFCMEM2**

<b>Address Offset</b>	0x0000 00A0	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Defect correction memory	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								DFCMEM2															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	DFCMEM2	Defect correction Memory 2 Set SUB1: Defect level of the Vertical line defect position (V = Vdefect). DFCMEM2 can be up shifted according to VDFLSFT, and subtracted from the data for Vertical line defect correction.	RW	0x00

**Table 8-975. Register Call Summary for Register ISIF\_DFCMEM2**

ISS ISP

- [ISS ISP ISIF Vertical Line Defect Correction \(VDFC\): \[0\]](#)
- [ISS ISP ISIF Vertical Line Defect Table Update Procedure: \[1\] \[2\]](#)
- [ISS ISP ISIF Register Setup: \[3\]](#)
- [ISS ISIF Register Summary: \[4\]](#)

**Table 8-976. ISIF\_DFCMEM3**

<b>Address Offset</b>	0x0000 00A4	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Defect correction memory	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								DFCMEM3															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	DFCMEM3	<Defect correction> Memory 3 Set SUB2: Defect level of the pixels upper than the Vertical line defect (V < Vdefect). DFCMEM3 can be up shifted according to VDFLSFT, and subtracted from the data for Vertical line defect correction.	RW	0x00

**Table 8-977. Register Call Summary for Register ISIF\_DFCMEM3**

ISS ISP

- [ISS ISP ISIF Vertical Line Defect Correction \(VDFC\): \[0\]](#)
- [ISS ISP ISIF Vertical Line Defect Table Update Procedure: \[1\] \[2\]](#)
- [ISS ISP ISIF Register Setup: \[3\]](#)
- [ISS ISIF Register Summary: \[4\]](#)



**Table 8-978. ISIF\_DFCMEM4**

<b>Address Offset</b>	0x0000 00A8	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Defect correction memory	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								DFCMEM4															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	RESERVED		R	0x00
7:0	DFCMEM4	Memory 4 Set SUB3: Defect level of the pixels lower than the Vertical line defect ( $V > V_{defect}$ ). DFCMEM4 can be up shifted according to VDFLSFT, and subtracted from the data for Vertical line defect correction.	RW	0x00

**Table 8-979. Register Call Summary for Register ISIF\_DFCMEM4**

- ISS ISP
- [ISS ISP ISIF Vertical Line Defect Correction \(VDFC\): \[0\]](#)
  - [ISS ISP ISIF Vertical Line Defect Table Update Procedure: \[1\] \[2\]](#)
  - [ISS ISP ISIF Register Setup: \[3\]](#)
  - [ISS ISIF Register Summary: \[4\]](#)

**Table 8-980. ISIF\_CLAMPCFG**

<b>Address Offset</b>	0x0000 00AC	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	BLACK CLAMP CTRL REGISTER	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								CLMD	RESERVED	CLHMD	CLEN												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:5	RESERVED		R	0x000
4	CLMD	Black clamp mode Clamp value can be calculated regardless of the color or can be calculated separately for each 4 colors. 0x0: Clamp value calculated regardless of the pixel color. 0x1: Clamp value calculated separately for each 4 colors.	RW	0
3	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
2:1	CLHMD	Horizontal Clamp mode  0x0: Horizontal clamp disabled. Only the Vertical clamp value is subtracted from the Image data.  0x1: Horizontal clamp value calculation enabled. The calculated Horizontal clamp value is subtracted from the Image data along with the Vertical clamp value.  0x3: Reserved  0x2: Horizontal clamp value not updated. The Horizontal clamp value used for the previous image is subtracted from the Image data along with the Vertical clamp value.	RW	0x0
0	CLEN	Black Clamp Enable Enables clamp value to be subtracted from Image data.  0x0: Disable 0x1: Enable	RW	0

**Table 8-981. Register Call Summary for Register ISIF\_CLAMPCFG**

## ISS ISP

- [ISS ISP ISIF Black Clamp: \[0\] \[1\]](#)
- [ISS ISP ISIF Clamp Value for Horizontal Direction: \[2\] \[3\] \[4\]](#)
- [ISS ISP ISIF Register Setup: \[5\] \[6\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[7\]](#)
- [ISS ISIF Register Summary: \[8\]](#)
- [ISS ISIF Register Description: \[9\] \[10\]](#)

**Table 8-982. ISIF\_CLDCOFST**

<b>Address Offset</b>	0x0000 00B0	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		CLDC													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLDC	DC offset for black clamp This value is added to the incoming pixels regardless whether optical black clamp is enabled ( <a href="#">ISIF_CLAMPCFG.CLEN</a> ). This value is in S13Q0 format.	RW	0x0000

**Table 8-983. Register Call Summary for Register ISIF\_CLDCOFST**

## ISS ISP

- [ISS ISP ISIF Black Clamp: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\] \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-984. ISIF\_CLSV**

<b>Address Offset</b>	0x0000 00B4		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		CLSV													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLSV	Black Clamp Start position (V). Sets the line number where clamp value subtraction starts. Range: 0 - 8191	RW	0x0000

**Table 8-985. Register Call Summary for Register ISIF\_CLSV**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-986. ISIF\_CLHWIN0**

<b>Address Offset</b>	0x0000 00B8		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																RESERVED	CLHWN	RESERVED	CLHWM	RESERVED	CLHLMT	CLHWBS	CLHWC									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:12	CLHWN	Horizontal Black clamp - Vertical dimension of a Window (2 <sup>N</sup> ). 0x0: Window is 2 pixels tall (N=1) 0x1: Window is 4 pixels tall (N=2) 0x3: Window is 16 pixels tall (N=4) 0x2: Window is 8 pixels tall (N=3)	RW	0x0
11:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:8	CLHWM	Horizontal Black clamp - Horizontal dimension of a Window ( $2^M$ ). 0x0: Window is 32 pixels wide (M=5) 0x1: Window is 64 pixels wide (M=6) 0x3: Window is 256 pixels wide (M=8) 0x2: Window is 128 pixels wide (M=7)	RW	0x0
7	RESERVED		R	0
6	CLHLMT	Horizontal Black clamp - Pixel value limitation for the Horizontal clamp value calculation. If this bit is set, the maximum pixel value to be used for the clamp value calculation would be limited to 1023. By setting this bit, the pixel value greater than 1023 will be replaced by the last pixel value which was equal to or less than 1023. In case <code>ISIF_CLAMPCFG.CLMD=1</code> (4-color mode), the pixel value greater than 1023 will be replaced by the last pixel value of the same color which was equal to or less than 1023. 0x0: Limitation disabled 0x1: Limitation enabled	RW	0
5	CLHWBS	Horizontal Black clamp - Base Window select 0x0: The most left window 0x1: The most right window	RW	0
4:0	CLHWC	Horizontal Black clamp - Window count per color Window count = CLHWC+1 Range: 1 - 32	RW	0x00

**Table 8-987. Register Call Summary for Register ISIF\_CLHWIN0**

## ISS ISP

- ISS ISP ISIF Clamp Value for Horizontal Direction: [\[0\]](#) [\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#) [\[5\]](#)
- ISS ISP ISIF Clamp Value for Vertical Direction: [\[6\]](#)
- ISS ISP ISIF Register Setup: [\[7\]](#) [\[8\]](#) [\[9\]](#) [\[10\]](#) [\[11\]](#)
- ISS ISIF Register Summary: [\[12\]](#)

**Table 8-988. ISIF\_CLHWIN1**

<b>Address Offset</b>	0x0000 00BC	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		CLHSH													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLHSH	Horizontal black clamp. Window Start position (H). Range: 0 - 8191	RW	0x0000

**Table 8-989. Register Call Summary for Register ISIF\_CLHWIN1**

ISS ISP

- [ISS ISP ISIF Clamp Value for Horizontal Direction: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-990. ISIF\_CLHWIN2**

<b>Address Offset</b>	0x0000 00C0	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				CLHSV											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLHSV	Horizontal black clamp. Window Start position (V). Range: 0 - 8191	RW	0x0000

**Table 8-991. Register Call Summary for Register ISIF\_CLHWIN2**

ISS ISP

- [ISS ISP ISIF Clamp Value for Horizontal Direction: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-992. ISIF\_CLVRV**

<b>Address Offset</b>	0x0000 00C4	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				CLVRV											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	RESERVED		R	0x0
11:0	CLVRV	Vertical black clamp reset value. (U12) Range: 0 to 4095	RW	0x000

**Table 8-993. Register Call Summary for Register ISIF\_CLVRV**

ISS ISP

- [ISS ISP ISIF Clamp Value for Vertical Direction: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-994. ISIF\_CLVWIN0**

<b>Address Offset</b>	0x0000 00C8	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLVCOEF								RESERVED	CLVRVSL	RESERVED	CLVOBH				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CLVCOEF	Vertical Black clamp - Line average coefficient (k). Set a coefficient which is applied to the line average for clamp value calculation. (1-k) is applied to the clamp value of the previous line. Value in the U8Q8 format, the range is 0 to 255/256.	RW	0x00
7:6	RESERVED		R	0x0
5:4	CLVRVSL	Vertical Black clamp - reset value selection Select the reset value for the clamp value of the previous line  0x0: The base value calculated for Horizontal direction 0x1: Value set via the configuration register 0x3: Reserved 0x2: No update (same as the previous image)	RW	0x0
3	RESERVED		R	0
2:0	CLVOBH	Vertical Black clamp - Optical Black H valid (2 <sup>L</sup> ).  0x6: Reserved 0x1: 4 pixels wide (L=2) 0x7: Reserved 0x0: 2 pixels wide (L=1) 0x2: 8 pixels wide (L=3) 0x4: 32 pixels wide (L=5) 0x5: 64 pixels wide (L=6) 0x3: 16 pixels wide (L=4)	RW	0x0

**Table 8-995. Register Call Summary for Register ISIF\_CLVWIN0**

ISS ISP

- [ISS ISP ISIF Clamp Value for Vertical Direction: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP ISIF Register Setup: \[5\] \[6\] \[7\]](#)
- [ISS ISIF Register Summary: \[8\]](#)

**Table 8-996. ISIF\_CLVWIN1**

<b>Address Offset</b>	0x0000 00CC		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				CLVSH											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLVSH	Vertical black clamp. Window Start position (H). Range: 0 - 8191	RW	0x0000

**Table 8-997. Register Call Summary for Register ISIF\_CLVWIN1**

ISS ISP

- [ISS ISP ISIF Clamp Value for Vertical Direction: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-998. ISIF\_CLVWIN2**

<b>Address Offset</b>	0x0000 00D0		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				CLVSV											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLVSV	Vertical black clamp. Window Start position (V). Range: 0 - 8191	RW	0x0000

**Table 8-999. Register Call Summary for Register ISIF\_CLVWIN2**

ISS ISP

- [ISS ISP ISIF Clamp Value for Vertical Direction: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1000. ISIF\_CLVWIN3**

<b>Address Offset</b>	0x0000 00D4		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	BLACK CLAMP CTRL REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		CLVOBV													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	CLVOBV	Vertical black clamp. Optical black V valid (V). Range: 0 - 8191	RW	0x0000

**Table 8-1001. Register Call Summary for Register ISIF\_CLVWIN3**

ISS ISP

- [ISS ISP ISIF Clamp Value for Vertical Direction: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1002. ISIF\_LSCHOFST**

<b>Address Offset</b>	0x0000 00D8		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		HOFST													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:0	HOFST	H direction Data offset for Lens Shading Correction. Range: 0-16,383 Not valid if the Formatter is enabled.	RW	0x0000

**Table 8-1003. Register Call Summary for Register ISIF\_LSCHOFST**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)



**Table 8-1004. ISIF\_LSCVOFST**

<b>Address Offset</b>	0x0000 00DC		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		VOFST													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:0	VOFST	V direction Data offset for Lens Shading Correction. Range: 0-16,383	RW	0x0000

**Table 8-1005. Register Call Summary for Register ISIF\_LSCVOFST**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1006. ISIF\_LSCHVAL**

<b>Address Offset</b>	0x0000 00E0		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		HVAL													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:0	HVAL	Number of valid pixels in H direction. HVAL is for LSC. Number of valid pixels = HVAL+ 1	RW	0x0000

**Table 8-1007. Register Call Summary for Register ISIF\_LSCHVAL**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1008. ISIF\_LSCVVAL**

<b>Address Offset</b>	0x0000 00E4	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	VVAL														

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	RESERVED		R	0x0
13:0	VVAL	Number of valid lines in V direction. VVAL is for LSC. Number of valid lines = VVAL+ 1	RW	0x0000

**Table 8-1009. Register Call Summary for Register ISIF\_LSCVVAL**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1010. ISIF\_2DLSCCFG**

<b>Address Offset</b>	0x0000 00E8	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	GAIN_MODE_M	RESERVED	GAIN_MODE_N	BUSY	RESERVED	GAIN_FORMAT	ENABLE								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:12	GAIN_MODE_M	Define the horizontal dimension of a paxel. Possible values are listed below. 0x6: Paxel is 64 pixels tall (M=64) 0x1: Reserved 0x7: Paxel is 128 pixels tall (M=128) 0x0: Reserved 0x2: Reserved 0x4: Paxel is 16 pixels tall (M=16) 0x5: Paxel is 32 pixels tall (M=32) 0x3: Paxel is 8 pixels tall (M=8)	RW	0x6
11	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
10:8	GAIN_MODE_N	Define the vertical dimension of a paxel. Possible values are listed below. 0x6: Paxel is 64 pixels tall (N=64) 0x1: Reserved 0x7: Paxel is 128 pixels tall (N=128) 0x0: Reserved 0x2: Reserved 0x4: Paxel is 16 pixels tall (N=16) 0x5: Paxel is 32 pixels tall (N=32) 0x3: Paxel is 8 pixels tall (N=8)	RW	0x6
7	BUSY	Busy bit Read 0x1: Busy Read 0x0: Idle	R	0
6:4	RESERVED		R	0x0
3:1	GAIN_FORMAT	Sets gain table format 0x6: Coded as 3-bit integer, 5-bit fraction Range from 0 to 7+31/32 0x1: Coded as 8-bit fraction + 1.0 of base Range from 1 to 1+255/256 0x7: Coded as 3-bit integer, 5-bit fraction + 1.0 Range from 1 to 8+31/32 0x0: Coded as 8-bit fraction Range from 0 to 255/256 0x2: Coded as 1-bit integer, 7-bit fraction Range from 0 to 1+127/128 0x4: Coded as 2-bit integer, 6-bit fraction Range from 0 to 3+63/64 0x5: Coded as 2-bit integer, 6-bit fraction + 1.0 Range from 1 to 4+63/64 0x3: Coded as 1-bit integer, 7-bit fraction + 1.0 Range from 1 to 2+127/128	RW	0x0
0	ENABLE	Enables/disables LSC 0x0: Disables the module at the end of the current frame. 0x1: Enables the module.	RW	0

**Table 8-1011. Register Call Summary for Register ISIF\_2DLSCCFG**

## ISS ISP

- [ISS ISP ISIF Lens Shading Correction \(2D-LSC\): \[0\]](#)
- [ISS ISP ISIF 2D-LSC Gain and Offset Tables: \[1\] \[2\] \[3\]](#)
- [ISS ISP ISIF Enabling/ Disabling the 2D-LSC Module: \[4\]](#)
- [ISS ISP ISIF Supported On-the-Fly 2D-LSC Configurations: \[5\]](#)
- [ISS ISP ISIF Read Port: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [ISS ISP ISIF Register Setup: \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [ISS ISIF Register Summary: \[19\]](#)

**Table 8-1012. ISIF\_2DLSCOFST**

<b>Address Offset</b>	0x0000 00EC		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OFSTSF								RESERVED	OFSTSF				RESERVED	OFSTEN	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	OFSTSF	Scaling factor for Offsets (U8Q7) Range: 0 to 1+127/128	RW	0x80
7	RESERVED		R	0
6:4	OFSTSF	Shift up value for Offsets (S8Q0) 0x6: Reserved 0x1: 1bit left shift 0x7: Reserved 0x0: No shift 0x2: 2bits left shift 0x4: 4bits left shift 0x5: 5bits left shift 0x3: 3bits left shift	RW	0x0
3:1	RESERVED		R	0x0
0	OFSTEN	Enables/disables Offset control in LSC 0x0: Disable 0x1: Enable	RW	0

**Table 8-1013. Register Call Summary for Register ISIF\_2DLSCOFST**

## ISS ISP

- [ISS ISP ISIF 2D-LSC Gain and Offset Tables: \[0\] \[1\] \[2\]](#)
- [ISS ISP ISIF Application of Gain and Offset to Image Pixels: \[3\] \[4\]](#)
- [ISS ISP ISIF Read Port: \[5\]](#)
- [ISS ISP ISIF Register Setup: \[6\] \[7\] \[8\]](#)
- [ISS ISIF Register Summary: \[9\]](#)

**Table 8-1014. ISIF\_2DLSCINI**

<b>Address Offset</b>	0x0000 00F0		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	Y								RESERVED	X					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	RESERVED		R	0
14:8	Y	Initial Y Y position, in pixels, of the first active pixel in reference to the first active pixel. Must be an even number.	RW	0x00
7	RESERVED		R	0
6:0	X	Initial X X position, in pixels, of the first active pixel in reference to the first active pixel. Must be an even number.	RW	0x00

**Table 8-1015. Register Call Summary for Register ISIF\_2DLSCINI**

ISS ISP

- [ISS ISP ISIF 2D-LSC Gain and Offset Tables: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\] \[3\]](#)
- [ISS ISP ISIF Summary of Constraints: \[4\]](#)
- [ISS ISIF Register Summary: \[5\]](#)

**Table 8-1016. ISIF\_2DLSCGRBU**

<b>Address Offset</b>	0x0000 00F4	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE31_16															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BASE31_16	Gain Table address base (Upper 16-bits) Table address in bytes. Table is 32-bit aligned so this register must be a multiple of 4. This bit field sets the address of the gain table in memory.	RW	0x0000

**Table 8-1017. Register Call Summary for Register ISIF\_2DLSCGRBU**

ISS ISP

- [ISS ISP ISIF 2D-LSC Gain and Offset Tables: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1018. ISIF\_2DLSCGRBL**

<b>Address Offset</b>	0x0000 00F8	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE15_0															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BASE15_0	Gain Table address base (Lower 16-bits) Table address in bytes. Table is 32-bit aligned so this register must be a multiple of 4. This bit field sets the address of the gain table in memory.	RW	0x0000

**Table 8-1019. Register Call Summary for Register ISIF\_2DLSCGRBL**

ISS ISP

- [ISS ISP ISIF 2D-LSC Gain and Offset Tables: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1020. ISIF\_2DLSCGROF**

<b>Address Offset</b>	0x0000 00FC	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OFFSET															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	OFFSET	Gain Table offset Defines the length, in bytes, of one row of the table. Table is 32-bit aligned, so this value must be a multiple of 4. Note that the row in memory could be longer than what LSC uses.	RW	0x0000

**Table 8-1021. Register Call Summary for Register ISIF\_2DLSCGROF**

ISS ISP

- [ISS ISP ISIF 2D-LSC Gain and Offset Tables: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1022. ISIF\_2DLSCORBU**

<b>Address Offset</b>	0x0000 0100	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BASE	Offset Table address base (Upper 16-bits) Table address in bytes. Table is 32-bit aligned so this register must be a multiple of 4. This bit field sets the address of the gain table in memory.	RW	0x0000

**Table 8-1023. Register Call Summary for Register ISIF\_2DLSCORBU**

ISS ISP

- [ISS ISP ISIF 2D-LSC Gain and Offset Tables: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1024. ISIF\_2DLSCORBL**

<b>Address Offset</b>	0x0000 0104	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	2D Lens Shading Correction Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BASE	Offset Table address base (Lower 16-bits) Table address in bytes. Table is 32-bit aligned so this register must be a multiple of 4. This bit field sets the address of the gain table in memory.	RW	0x0000

**Table 8-1025. Register Call Summary for Register ISIF\_2DLSCORBL**

ISS ISP

- [ISS ISP ISIF 2D-LSC Gain and Offset Tables: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1026. ISIF\_2DLSCOROF**

<b>Address Offset</b>	0x0000 0108	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	2D Lens Shading Correction Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OFFSET															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	OFFSET	Offset Table offset Defines the length, in bytes, of one row of the table. Table is 32-bit aligned, so this value must be a multiple of 4. Note that the row in memory could be longer than what LSC uses.	RW	0x0000

**Table 8-1027. Register Call Summary for Register ISIF\_2DLSCOROF**

ISS ISP

- [ISS ISP ISIF 2D-LSC Gain and Offset Tables: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1028. ISIF\_2DLSCIRQEN**

<b>Address Offset</b>	0x0000 010C	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												SOF	PREFETCH_COMPLETED	PREFETCH_ERROR	DONE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x0000
3	SOF	Interrupt status for LSC SOF Indicates the start of the LSC valid region. LSC configuration registers can be updated after LSC SOF for the next frame.  0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
2	PREFETCH_COMPLETED	Interrupt enable for Prefetch Complete Indicates current state of the prefetch buffer. Could be used to start sending the data once the buffer is full to minimize the risk of an underflow. This event is triggered when the buffer contains 3 full paxel rows.  0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
1	PREFETCH_ERROR	Interrupt enable for Prefetch Error The prefetch error indicates when the gain table was read to slowly from SDRAM. When this event is pending the module goes into transparent mode (output=input). Normal operation can be resumed at the start of the next frame after 1) clearing this event 2) disabling the LSC module 3) enabling it  0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0
0	DONE	Interrupt enable for LSC Done The event is triggered when the internal state of LSC toggles from BUSY to IDLE.  0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0

**Table 8-1029. Register Call Summary for Register ISIF\_2DLSCIRQEN**

## ISS ISP

- [ISS ISP ISIF 2D-LSC Events and Status Checking: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISIF Register Summary: \[5\]](#)



**Table 8-1030. ISIF\_2DLSCIRQST**

<b>Address Offset</b>	0x0000 0110		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	2D Lens Shading Correction Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											SOF	PREFETCH_COMPLETED	PREFETCH_ERROR	DONE	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:4	RESERVED		R	0x0000
3	SOF	<p>Interrupt status for LSC SOF Indicates the start of the LSC valid region. LSC configuration registers can be updated after LSC SOF for the next frame.</p> <p>0x0: Event is not pending (r) Bit remains unchanged (w)</p> <p>0x1: Event is pending (r) Event is cleared (w)</p>	RW	0
2	PREFETCH_COMPLETED	<p>Interrupt status for Prefetch Complete Indicates current state of the prefetch buffer. Could be used to start sending the data once the buffer is full to minimize the risk of an underflow. This event is triggered when the buffer contains 3 full paxel rows. It could be used to minimize buffer underflow risks.</p> <p>0x0: Event is not pending (r) Bit remains unchanged (w)</p> <p>0x1: Event is pending (r) Event is cleared (w)</p>	RW	0
1	PREFETCH_ERROR	<p>Interrupt status for Prefetch Error The prefetch error indicates when the gain table was read to slowly from SDRAM. When this event is pending the module goes into transparent mode (output=input). Normal operation can be resumed at the start of the next frame after</p> <ol style="list-style-type: none"> <li>clearing this event</li> <li>disabling the LSC module</li> <li>enabling it</li> </ol> <p>0x0: Event is not pending (r) Bit remains unchanged (w)</p> <p>0x1: Event is pending (r) Event is cleared (w)</p>	RW	0
0	DONE	<p>Interrupt status for LSC Done The event is triggered when the internal state of LSC toggles from BUSY to IDLE.</p> <p>0x0: Event is not pending (r) Bit remains unchanged (w)</p> <p>0x1: Event is pending (r) Event is cleared (w)</p>	RW	0

**Table 8-1031. Register Call Summary for Register ISIF\_2DLSCIRQST**

ISS ISP

- [ISS ISP ISIF Enabling/ Disabling the 2D-LSC Module: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [ISS ISP ISIF 2D-LSC Events and Status Checking: \[9\]](#)
- [ISS ISIF Register Summary: \[10\]](#)

**Table 8-1032. ISIF\_FMTCFG**

<b>Address Offset</b>	0x0000 0114	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTAINC				RESERVED	LNUM		RESERVED	LNALT	FMTCBL	FMTEN					

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:8	FMTAINC	Address increment Address increment = (FMTAINC + 1) Range (1-16) *This bit is latched by VD.	RW	0x0
7:6	RESERVED		R	0x0
5:4	LNUM	Split/Combine number of lines *This bit is latched by VD.  0x0: 1 output line  0x1: 1 input line -> 2 output lines (FMTCBL=0) 2 input lines -> 1 output line (FMTCBL=1)  0x3: 1 input line -> 4 output lines (FMTCBL=0) 4 input lines -> 1 output line (FMTCBL=1)  0x2: 1 input line -> 3 output lines (FMTCBL=0) 3 input lines -> 1 output line (FMTCBL=1)	RW	0x0
3	RESERVED		R	0
2	LNALT	Line alternating *This bit is latched by VD.  0x0: Normal mode 0x1: Line alternative mode	RW	0
1	FMTCBL	Combine Input lines *This bit is latched by VD.  0x0: Split 1 input line into multiple output lines 0x1: Combine multiple input lines into 1 output line	RW	0
0	FMTEN	CCD Formatter enable *This bit is latched by VD.  0x0: Disable 0x1: Enable	RW	0

**Table 8-1033. Register Call Summary for Register ISIF\_FMTCFG**

ISS ISP

- [ISS ISP ISIF Input Data Formatter: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP ISIF Formatter Programming: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\]](#)
- [ISS ISP ISIF Register Setup: \[21\] \[22\] \[23\] \[24\] \[25\]](#)
- [ISS ISIF Register Summary: \[26\]](#)

**Table 8-1034. ISIF\_FMTPLEN**

<b>Address Offset</b>	0x0000 0118	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTPLEN3		RESERVED	FMTPLEN2		FMTPLEN1			FMTPLEN0							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:12	FMTPLEN3	Number of program entries for SET3 Number of entries = (FMTPLEN3 + 1) Range: 1-8 Valid only if FMTCBL is set *This bit is latched by VD.	RW	0x0
11	RESERVED		R	0
10:8	FMTPLEN2	Number of program entries for SET2 Number of entries = (FMTPLEN2 + 1) Range: 1-8 Valid only if FMTCBL is set *This bit is latched by VD.	RW	0x0
7:4	FMTPLEN1	Number of program entries for SET1 Number of entries = (FMTPLEN1 + 1) Range: 1-16 (FMTCBL = 0) 1-8 (FMTCBL = 1) Setting a value greater than 7 to FMTPLEN1 is not allowed if FMTCBL is set *This bit is latched by VD.	RW	0x0
3:0	FMTPLEN0	Number of program entries for SET0 Number of entries = (PLEN0 + 1) Range: 1-16 (FMTCBL = 0) 1-8 (FMTCBL = 1) Setting a value greater than 7 to FMTPLEN1 is not allowed if FMTCBL is set *This bit is latched by VD.	RW	0x0

**Table 8-1035. Register Call Summary for Register ISIF\_FMTPLEN**

ISS ISP

- [ISS ISP ISIF Formatter Programming: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[5\] \[6\] \[7\]](#)
- [ISS ISP ISIF Register Setup: \[8\] \[9\] \[10\] \[11\]](#)
- [ISS ISIF Register Summary: \[12\]](#)

**Table 8-1036. ISIF\_FMTSPH**

<b>Address Offset</b>	0x0000 011C	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTSPH															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTSPH	The first pixel in a line fed into the formatter	RW	0x0000

**Table 8-1037. Register Call Summary for Register ISIF\_FMTSPH**

## ISS ISP

- [ISS ISP ISIF Formatter Area Settings: \[0\] \[1\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[2\]](#)
- [ISS ISP ISIF Color Space Converter: \[3\]](#)
- [ISS ISP ISIF Register Setup: \[4\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[5\]](#)
- [ISS ISIF Register Summary: \[6\]](#)

**Table 8-1038. ISIF\_FMTLNH**

<b>Address Offset</b>	0x0000 0120	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTLNH															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTLNH	Number of pixels in a line fed to the formatter. Number of pixels = FMTLNH + 1	RW	0x0000

**Table 8-1039. Register Call Summary for Register ISIF\_FMTLNH**

## ISS ISP

- [ISS ISP ISIF Formatter Area Settings: \[0\] \[1\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[2\]](#)
- [ISS ISP ISIF Color Space Converter: \[3\]](#)
- [ISS ISP ISIF Register Setup: \[4\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[5\]](#)
- [ISS ISIF Register Summary: \[6\]](#)

**Table 8-1040. ISIF\_FMTLSV**

<b>Address Offset</b>	0x0000 0124	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTSLV															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTSLV	Start line vertical	RW	0x0000

**Table 8-1041. Register Call Summary for Register ISIF\_FMTLSV**

ISS ISP

- [ISS ISP ISIF Formatter Area Settings: \[0\] \[1\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[2\]](#)
- [ISS ISP ISIF Color Space Converter: \[3\]](#)
- [ISS ISP ISIF Register Setup: \[4\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[5\]](#)
- [ISS ISIF Register Summary: \[6\]](#)

**Table 8-1042. ISIF\_FMTLNV**

<b>Address Offset</b>	0x0000 0128	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTLNV															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:0	FMTLNV	Number of lines in vertical Number of lines = FMTLNV + 1	RW	0x0000

**Table 8-1043. Register Call Summary for Register ISIF\_FMTLNV**

ISS ISP

- [ISS ISP ISIF Formatter Area Settings: \[0\] \[1\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[2\]](#)
- [ISS ISP ISIF Color Space Converter: \[3\]](#)
- [ISS ISP ISIF Register Setup: \[4\]](#)
- [ISS ISP ISIF Register Accessibility During Frame Processing: \[5\]](#)
- [ISS ISIF Register Summary: \[6\]](#)

**Table 8-1044. ISIF\_FMTRLEN**

<b>Address Offset</b>	0x0000 012C	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTRLEN															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTRLEN	Number of pixels in an output line Maximum value = 4480	RW	0x0000

**Table 8-1045. Register Call Summary for Register ISIF\_FMTRLEN**

ISS ISP

- [ISS ISP ISIF Formatter Area Settings: \[0\] \[1\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[2\]](#)
- [ISS ISP ISIF Register Setup: \[3\]](#)
- [ISS ISIF Register Summary: \[4\]](#)

**Table 8-1046. ISIF\_FMTHCNT**

<b>Address Offset</b>	0x0000 0130	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FMTHCNT															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	FMTHCNT	HD interval for output lines Set all 0 to this register if combining multiple lines into a single line	RW	0x0000

**Table 8-1047. Register Call Summary for Register ISIF\_FMTHCNT**

ISS ISP

- [ISS ISP ISIF Formatter Area Settings: \[0\] \[1\] \[2\]](#)
- [ISS ISP ISIF Register Setup: \[3\]](#)
- [ISS ISIF Register Summary: \[4\]](#)

**Table 8-1048. ISIF\_FMTAPTR0**

<b>Address Offset</b>	0x0000 0134		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE		INIT													

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 0 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1049. Register Call Summary for Register ISIF\_FMTAPTR0**

ISS ISP

- [ISS ISP ISIF Formatter Programming: \[0\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1050. ISIF\_FMTAPTR1**

<b>Address Offset</b>	0x0000 0138		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE		INIT													

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 1 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1051. Register Call Summary for Register ISIF\_FMTAPTR1**

ISS ISP

- [ISS ISP ISIF Combine the Divided Input Lines: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1052. ISIF\_FMTAPTR2**

<b>Address Offset</b>	0x0000 013C	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE		INIT													

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 2 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1053. Register Call Summary for Register ISIF\_FMTAPTR2**

ISS ISP

- [ISS ISP ISIF Combine the Divided Input Lines: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1054. ISIF\_FMTAPTR3**

<b>Address Offset</b>	0x0000 0140	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE		INIT													

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 3 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1055. Register Call Summary for Register ISIF\_FMTAPTR3**

ISS ISP

- [ISS ISP ISIF Combine the Divided Input Lines: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)



**Table 8-1056. ISIF\_FMTAPTR4**

<b>Address Offset</b>	0x0000 0144		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE		INIT													

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 4 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1057. Register Call Summary for Register ISIF\_FMTAPTR4**

ISS ISP

- [ISS ISP ISIF Combine the Divided Input Lines: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1058. ISIF\_FMTAPTR5**

<b>Address Offset</b>	0x0000 0148		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE		INIT													

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 5 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1059. Register Call Summary for Register ISIF\_FMTAPTR5**

ISS ISP

- [ISS ISP ISIF Combine the Divided Input Lines: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1060. ISIF\_FMTAPTR6**

<b>Address Offset</b>	0x0000 014C	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LINE		INIT															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 6 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1061. Register Call Summary for Register ISIF\_FMTAPTR6**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

**Table 8-1062. ISIF\_FMTAPTR7**

<b>Address Offset</b>	0x0000 0150	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LINE		INIT															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 7 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1063. Register Call Summary for Register ISIF\_FMTAPTR7**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

**Table 8-1064. ISIF\_FMTAPTR8**

<b>Address Offset</b>	0x0000 0154	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE		INIT													

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 8 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1065. Register Call Summary for Register ISIF\_FMTAPTR8**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

**Table 8-1066. ISIF\_FMTAPTR9**

<b>Address Offset</b>	0x0000 0158	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE		INIT													

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 9 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1067. Register Call Summary for Register ISIF\_FMTAPTR9**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

**Table 8-1068. ISIF\_FMTAPTR10**

<b>Address Offset</b>	0x0000 015C		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LINE		INIT															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 10 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1069. Register Call Summary for Register ISIF\_FMTAPTR10**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

**Table 8-1070. ISIF\_FMTAPTR11**

<b>Address Offset</b>	0x0000 0160		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LINE		INIT															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 11 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1071. Register Call Summary for Register ISIF\_FMTAPTR11**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

**Table 8-1072. ISIF\_FMTAPTR12**

<b>Address Offset</b>	0x0000 0164	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LINE		INIT															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 12 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1073. Register Call Summary for Register ISIF\_FMTAPTR12**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

**Table 8-1074. ISIF\_FMTAPTR13**

<b>Address Offset</b>	0x0000 0168	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LINE		INIT															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 13 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1075. Register Call Summary for Register ISIF\_FMTAPTR13**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

**Table 8-1076. ISIF\_FMTAPTR14**

<b>Address Offset</b>	0x0000 016C	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LINE		INIT															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 14 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1077. Register Call Summary for Register ISIF\_FMTAPTR14**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

**Table 8-1078. ISIF\_FMTAPTR15**

<b>Address Offset</b>	0x0000 0170	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Input Data Formatter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LINE		INIT															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x00000
14:13	LINE	The output line the address belongs to Valid only if FMTCBL is cleared  0x0: 1st line 0x1: 2nd line 0x3: 4th line 0x2: 3rd line	RW	0x0
12:0	INIT	Initial address value for address pointer 15 This address can not exceed FMTRLEN - 1	RW	0x0000

**Table 8-1079. Register Call Summary for Register ISIF\_FMTAPTR15**

ISS ISP

- [ISS ISP ISIF Formatter Programming: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1080. ISIF\_FMTPGMVFO**

<b>Address Offset</b>	0x0000 0174	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM15EN	PGM14EN	PGM13EN	PGM12EN	PGM11EN	PGM10EN	PGM09EN	PGM08EN	PGM07EN	PGM06EN	PGM05EN	PGM04EN	PGM03EN	PGM02EN	PGM01EN	PGM00EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PGM15EN	Program 15 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
14	PGM14EN	Program 14 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
13	PGM13EN	Program 13 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
12	PGM12EN	Program 12 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
11	PGM11EN	Program 11 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
10	PGM10EN	Program 10 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
9	PGM09EN	Program 9 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
8	PGM08EN	Program 8 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
7	PGM07EN	Program 7 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
6	PGM06EN	Program 6 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
5	PGM05EN	Program 5 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
4	PGM04EN	Program 4 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0

Bits	Field Name	Description	Type	Reset
3	PGM03EN	Program 3 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
2	PGM02EN	Program 2 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
1	PGM01EN	Program 1 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
0	PGM00EN	Program 0 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0

**Table 8-1081. Register Call Summary for Register ISIF\_FMTPGMVF0**

## ISS ISP

- [ISS ISP ISIF Formatter Programming: \[0\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP ISIF Register Setup: \[5\]](#)
- [ISS ISIF Register Summary: \[6\]](#)

**Table 8-1082. ISIF\_FMTPGMVF1**

<b>Address Offset</b>	0x0000 0178	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM31EN	PGM30EN	PGM29EN	PGM28EN	PGM27EN	PGM26EN	PGM25EN	PGM24EN	PGM23EN	PGM22EN	PGM21EN	PGM20EN	PGM19EN	PGM18EN	PGM17EN	PGM16EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PGM31EN	Program 31 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
14	PGM30EN	Program 30 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
13	PGM29EN	Program 29 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
12	PGM28EN	Program 28 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
11	PGM27EN	Program 27 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0



Bits	Field Name	Description	Type	Reset
10	PGM26EN	Program 26 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
9	PGM25EN	Program 25 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
8	PGM24EN	Program 24 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
7	PGM23EN	Program 23 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
6	PGM22EN	Program 22 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
5	PGM21EN	Program 21 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
4	PGM20EN	Program 20 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
3	PGM19EN	Program 19 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
2	PGM18EN	Program 18 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
1	PGM17EN	Program 17 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0
0	PGM16EN	Program 16 Valid Flag 0x0: Skip this pixel 0x1: This pixel is valid	RW	0

**Table 8-1083. Register Call Summary for Register ISIF\_FMTPGMVF1**

## ISS ISP

- [ISS ISP ISIF Formatter Programming: \[0\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[1\] \[2\]](#)
- [ISS ISP ISIF Register Setup: \[3\]](#)
- [ISS ISIF Register Summary: \[4\]](#)

**Table 8-1084. ISIF\_FMTPGMAPU0**

<b>Address Offset</b>	0x0000 017C		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM15UPDT	PGM14UPDT	PGM13UPDT	PGM12UPDT	PGM11UPDT	PGM10UPDT	PGM9UPDT	PGM8UPDT	PGM7UPDT	PGM6UPDT	PGM5UPDT	PGM4UPDT	PGM3UPDT	PGM2UPDT	PGM1UPDT	PGM0UPDT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PGM15UPDT	Program 15 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
14	PGM14UPDT	Program 14 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
13	PGM13UPDT	Program 13 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
12	PGM12UPDT	Program 12 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
11	PGM11UPDT	Program 11 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
10	PGM10UPDT	Program 10 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
9	PGM9UPDT	Program 9 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
8	PGM8UPDT	Program 8 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
7	PGM7UPDT	Program 7 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
6	PGM6UPDT	Program 6 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
5	PGM5UPDT	Program 5 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0

Bits	Field Name	Description	Type	Reset
4	PGM4UPDT	Program 4 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
3	PGM3UPDT	Program 3 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
2	PGM2UPDT	Program 2 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
1	PGM1UPDT	Program 1 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
0	PGM0UPDT	Program 0 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0

**Table 8-1085. Register Call Summary for Register ISIF\_FMTPGMAPU0**

ISS ISP

- [ISS ISP ISIF Formatter Programming: \[0\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP ISIF Register Setup: \[5\]](#)
- [ISS ISIF Register Summary: \[6\]](#)

**Table 8-1086. ISIF\_FMTPGMAPU1**

<b>Address Offset</b>	0x0000 0180	<b>Instance</b>	ISS_ISIF
<b>Physical Address</b>	See <a href="#">Table 8-905</a>		
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM31UPDT	PGM30UPDT	PGM29UPDT	PGM28UPDT	PGM27UPDT	PGM26UPDT	PGM25UPDT	PGM24UPDT	PGM23UPDT	PGM22UPDT	PGM21UPDT	PGM20UPDT	PGM19UPDT	PGM18UPDT	PGM17UPDT	PGM16UPDT

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	PGM31UPDT	Program 31 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
14	PGM30UPDT	Program 30 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
13	PGM29UPDT	Program 29 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
12	PGM28UPDT	Program 28 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0

Bits	Field Name	Description	Type	Reset
11	PGM27UPDT	Program 27 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
10	PGM26UPDT	Program 26 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
9	PGM25UPDT	Program 25 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
8	PGM24UPDT	Program 24 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
7	PGM23UPDT	Program 23 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
6	PGM22UPDT	Program 22 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
5	PGM21UPDT	Program 21 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
4	PGM20UPDT	Program 20 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
3	PGM19UPDT	Program 19 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
2	PGM18UPDT	Program 18 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
1	PGM17UPDT	Program 17 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0
0	PGM16UPDT	Program 16 Address Pointer Update 0x0: APTR* + N (Auto increment) 0x1: APTR* - N (Auto decrement)	RW	0

**Table 8-1087. Register Call Summary for Register ISIF\_FMTPGMAPU1**

## ISS ISP

- [ISS ISP ISIF Formatter Programming: \[0\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[1\] \[2\]](#)
- [ISS ISP ISIF Register Setup: \[3\]](#)
- [ISS ISIF Register Summary: \[4\]](#)

**Table 8-1088. ISIF\_FMTPGMAPS0**

<b>Address Offset</b>	0x0000 0184		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM3APTR			PGM2APTR			PGM1APTR			PGM0APTR						

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM3APTR	Program 3 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM2APTR	Program 2 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM1APTR	Program 1 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM0APTR	Program 0 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

**Table 8-1089. Register Call Summary for Register ISIF\_FMTPGMAPS0**

ISS ISP

- [ISS ISP ISIF Formatter Programming: \[0\]](#)
- [ISS ISP ISIF Combine the Divided Input Lines: \[1\] \[2\]](#)
- [ISS ISP ISIF Register Setup: \[3\]](#)
- [ISS ISIF Register Summary: \[4\]](#)

**Table 8-1090. ISIF\_FMTPGMAPS1**

<b>Address Offset</b>	0x0000 0188		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM7APTR			PGM6APTR			PGM5APTR			PGM4APTR						

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM7APTR	Program 7 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM6APTR	Program 6 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM5APTR	Program 5 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM4APTR	Program 0 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

**Table 8-1091. Register Call Summary for Register ISIF\_FMTPGMAPS1**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1092. ISIF\_FMTPGMAPS2**

<b>Address Offset</b>	0x0000 018C		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM11APTR		PGM10APTR		PGM9APTR			PGM8APTR								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM11APTR	Program 11 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM10APTR	Program 10 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM9APTR	Program 9 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM8APTR	Program 8 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

**Table 8-1093. Register Call Summary for Register ISIF\_FMTPGMAPS2**

ISS ISP

- [ISS ISP ISIF Combine the Divided Input Lines: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1094. ISIF\_FMTPGMAPS3**

<b>Address Offset</b>	0x0000 0190		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM15APTR		PGM14APTR		PGM13APTR			PGM12APTR								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM15APTR	Program 15 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM14APTR	Program 14 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM13APTR	Program 13 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM12APTR	Program 12 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

**Table 8-1095. Register Call Summary for Register ISIF\_FMTPGMAPS3**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1096. ISIF\_FMTPGMAPS4**

<b>Address Offset</b>	0x0000 0194		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM19APTR		PGM18APTR		PGM17APTR			PGM16APTR								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM19APTR	Program 19 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM18APTR	Program 18 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM17APTR	Program 17 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM16APTR	Program 16 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

**Table 8-1097. Register Call Summary for Register ISIF\_FMTPGMAPS4**

ISS ISP

- [ISS ISP ISIF Combine the Divided Input Lines: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1098. ISIF\_FMTPGMAPS5**

<b>Address Offset</b>	0x0000 0198		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM23APTR		PGM22APTR		PGM21APTR			PGM20APTR								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM23APTR	Program 23 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM22APTR	Program 22 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM21APTR	Program 21 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM20APTR	Program 20 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

**Table 8-1099. Register Call Summary for Register ISIF\_FMTPGMAPS5**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1100. ISIF\_FMTPGMAPS6**

<b>Address Offset</b>	0x0000 019C		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM27APTR		PGM26APTR		PGM25APTR			PGM24APTR								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM27APTR	Program 27 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM26APTR	Program 26 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM25APTR	Program 25 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM24APTR	Program 24 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

**Table 8-1101. Register Call Summary for Register ISIF\_FMTPGMAPS6**

ISS ISP

- [ISS ISP ISIF Register Setup: \[0\]](#)
- [ISS ISIF Register Summary: \[1\]](#)

**Table 8-1102. ISIF\_FMTPGMAPS7**

<b>Address Offset</b>	0x0000 01A0		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Input Data Formatter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PGM31APTR		PGM30APTR		PGM29APTR			PGM28APTR								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PGM31APTR	Program 31 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
11:8	PGM30APTR	Program 30 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
7:4	PGM29APTR	Program 29 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0
3:0	PGM28APTR	Program 28 Address Pointer Select n: APTRn n:0-15 (APTR0 - APR15)	RW	0x0

**Table 8-1103. Register Call Summary for Register ISIF\_FMTPGMAPS7**

ISS ISP

- [ISS ISP ISIF Formatter Programming: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)



**Table 8-1104. ISIF\_CSCCTL**

<b>Address Offset</b>	0x0000 01A4	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Color Space Converter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											CSCEN				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	CSCEN	Controls ON/OFF of Color Space converter. 0x0: Disable 0x1: Enable	RW	0

**Table 8-1105. Register Call Summary for Register ISIF\_CSCCTL**

ISS ISP

- [ISS ISP ISIF Color Space Converter: \[0\]](#)
- [ISS ISP ISIF Register Setup: \[1\]](#)
- [ISS ISIF Register Summary: \[2\]](#)

**Table 8-1106. ISIF\_CSCM0**

<b>Address Offset</b>	0x0000 01A8	
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b> ISS_ISIF
<b>Description</b>	Color Space Converter Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CSCM01						CSCM00									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM01	Color Space convert coefficient value M01: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM00	Color Space convert coefficient value M00: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

**Table 8-1107. Register Call Summary for Register ISIF\_CSCM0**

ISS ISP

- [ISS ISP ISIF Color Space Converter: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1108. ISIF\_CSCM1**

<b>Address Offset</b>	0x0000 01AC		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Color Space Converter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSCM03								CSCM02															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM03	Color Space convert coefficient value M03: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM02	Color Space convert coefficient value M02: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

**Table 8-1109. Register Call Summary for Register ISIF\_CSCM1**

ISS ISP

- [ISS ISP ISIF Color Space Converter: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1110. ISIF\_CSCM2**

<b>Address Offset</b>	0x0000 01B0		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Color Space Converter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSCM11								CSCM10															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM11	Color Space convert coefficient value M11: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM10	Color Space convert coefficient value M10: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

**Table 8-1111. Register Call Summary for Register ISIF\_CSCM2**

ISS ISP

- [ISS ISP ISIF Color Space Converter: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1112. ISIF\_CSCM3**

<b>Address Offset</b>	0x0000 01B4		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Color Space Converter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSCM13								CSCM12															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM13	Color Space convert coefficient value M13: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM12	Color Space convert coefficient value M12: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

**Table 8-1113. Register Call Summary for Register ISIF\_CSCM3**

ISS ISP

- [ISS ISP ISIF Color Space Converter: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1114. ISIF\_CSCM4**

<b>Address Offset</b>	0x0000 01B8		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Color Space Converter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSCM21								CSCM20															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM21	Color Space convert coefficient value M21: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM20	Color Space convert coefficient value M20: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

**Table 8-1115. Register Call Summary for Register ISIF\_CSCM4**

ISS ISP

- [ISS ISP ISIF Color Space Converter: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1116. ISIF\_CSCM5**

<b>Address Offset</b>	0x0000 01BC		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Color Space Converter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSCM23								CSCM22															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM23	Color Space convert coefficient value M23: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM22	Color Space convert coefficient value M22: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

**Table 8-1117. Register Call Summary for Register ISIF\_CSCM5**

ISS ISP

- [ISS ISP ISIF Color Space Converter: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1118. ISIF\_CSCM6**

<b>Address Offset</b>	0x0000 01C0		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Color Space Converter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSCM31								CSCM30															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM31	Color Space convert coefficient value M31: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM30	Color Space convert coefficient value M30: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

**Table 8-1119. Register Call Summary for Register ISIF\_CSCM6**

ISS ISP

- [ISS ISP ISIF Color Space Converter: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1120. ISIF\_CSCM7**

<b>Address Offset</b>	0x0000 01C4		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>	Color Space Converter Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSCM33								CSCM32															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	CSCM33	Color Space convert coefficient value M33: This value is signed 8-bit with the 5-bits decimal.	RW	0x00
7:0	CSCM32	Color Space convert coefficient value M32: This value is signed 8-bit with the 5-bits decimal.	RW	0x00

**Table 8-1121. Register Call Summary for Register ISIF\_CSCM7**

ISS ISP

- [ISS ISP ISIF Color Space Converter: \[0\] \[1\]](#)
- [ISS ISP ISIF Register Setup: \[2\]](#)
- [ISS ISIF Register Summary: \[3\]](#)

**Table 8-1122. ISIF\_CLKCTL**

<b>Address Offset</b>	0x0000 01F8		
<b>Physical Address</b>	See <a href="#">Table 8-905</a>	<b>Instance</b>	ISS_ISIF
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED										CLKEN1	CLKEN2				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:2	RESERVED		R	0x0000
1	CLKEN1	Forces isif_clken1 to be active. (Test mode) 0x0: normal mode 0x1: force isif_clken1 to be active	RW	0
0	CLKEN2	Forces isif_clken2 to be active. (Test mode) 0x0: normal mode 0x1: force isif_clken2 to be active	RW	0

**Table 8-1123. Register Call Summary for Register ISIF\_CLKCTL**

ISS ISP

- [ISS ISIF Register Summary: \[0\]](#)

### 8.3.5.7 ISS IPIPEIF Registers

#### 8.3.5.7.1 ISS IPIPEIF Register Summary

**Table 8-1124. ISS IPIPEIF Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_IPIPEIF Base Address
IPIPEIF_ENABLE	RW	32	0x0000 0000	0x5201 1200
IPIPEIF_CFG1	RW	32	0x0000 0004	0x5201 1204
IPIPEIF_PPLN	RW	32	0x0000 0008	0x5201 1208
IPIPEIF_LPFR	RW	32	0x0000 000C	0x5201 120C
IPIPEIF_HNUM	RW	32	0x0000 0010	0x5201 1210
IPIPEIF_VNUM	RW	32	0x0000 0014	0x5201 1214
IPIPEIF_ADDRU	RW	32	0x0000 0018	0x5201 1218
IPIPEIF_ADDRL	RW	32	0x0000 001C	0x5201 121C
IPIPEIF_ADOFS	RW	32	0x0000 0020	0x5201 1220
IPIPEIF_RSZ	RW	32	0x0000 0024	0x5201 1224
IPIPEIF_GAIN	RW	32	0x0000 0028	0x5201 1228
IPIPEIF_DPCM	RW	32	0x0000 002C	0x5201 122C
IPIPEIF_CFG2	RW	32	0x0000 0030	0x5201 1230
IPIPEIF_INIRSZ	RW	32	0x0000 0034	0x5201 1234
IPIPEIF_OCLIP	RW	32	0x0000 0038	0x5201 1238
IPIPEIF_DTUDF	RW	32	0x0000 003C	0x5201 123C
IPIPEIF_CLKDIV	RW	32	0x0000 0040	0x5201 1240
IPIPEIF_DPC1	RW	32	0x0000 0044	0x5201 1244
IPIPEIF_DPC2	RW	32	0x0000 0048	0x5201 1248
IPIPEIF_RSZ3A	RW	32	0x0000 0054	0x5201 1254
IPIPEIF_INIRSZ3A	RW	32	0x0000 0058	0x5201 1258

#### 8.3.5.7.2 ISS IPIPEIF Register Description

**Table 8-1125. IPIPEIF\_ENABLE**

<b>Address Offset</b>	0x0000 0000																																																										
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>	<b>Instance</b> ISS_IPIPEIF																																																									
<b>Description</b>	IPIPEIF Enable.																																																										
<b>Type</b>	RW																																																										
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> </thead> <tbody> <tr> <td colspan="23">RESERVED</td> <td>SYNCOFF</td> <td>ENABLE</td> </tr> </tbody> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																							SYNCOFF	ENABLE
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																												
RESERVED																							SYNCOFF	ENABLE																																			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	SYNCOFF	<p>VD output mask</p> <p>This register masks the VD output to the IPIPE module. This can be useful when one wants to read data from SDRAM which are stored in a double buffer. If the VD is not masked each time we start the module a new VD will be generated to the IPIPEIF module.</p> <p>Let's consider two buffers A and B of N lines each.</p> <p>*This bit field is latched by VD.</p> <p>0x0: VD output mask is disabled.</p> <p>0x1: VD output mask is enabled.</p>	RW	0
0	ENABLE	<p>IPIPE I/F Enable</p> <p>This register is used to start the operation of SDRAM buffer memory read and generates SYNC signals. This register is available when INPSRC1 or INPSCR2 = 1, 2 or 3.</p> <p>0x0: disable</p> <p>0x1: enable</p>	RW	0

**Table 8-1126. Register Call Summary for Register IPIPEIF\_ENABLE**

ISS ISP

- [ISS ISP IPIPEIF Double-Buffer Input Function When Reading From BL: \[0\]](#)
- [ISS ISP IPIPEIF Timing Generation: \[1\]](#)
- [ISS ISP IPIPEIF Register Setup: \[2\]](#)
- [ISS ISP IPIPEIF Enable/Disable Hardware: \[3\] \[4\] \[5\] \[6\]](#)
- [ISS ISP IPIPEIF Register Accessibility During Frame Processing: \[7\] \[8\] \[9\]](#)
- [ISS IPIPEIF Register Summary: \[10\]](#)

**Table 8-1127. IPIPEIF\_CFG1**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Configuration #1		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																INPSRC1	DATASFT				CLKSEL	UNPACK	AVGFILT	RESERVED	INPSRC2	DECIM	ONESHOT				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:14	INPSRC1	<p>Selects the source for the mux (VPORT / ISIF / SDRAM) as well as the data format type.</p> <p>0x0: VPORT_RAW</p> <p>0x1: SDRAM_RAW</p> <p>0x3: SDRAM_YUV</p> <p>0x2: ISIF_DARKFM</p> <p>Input ports to DFS submodule are ISIF and SDRAM.</p>	RW	0x0

Bits	Field Name	Description	Type	Reset
13:11	DATASFT	SDRAM read data shift This register is available when INPSRC1 or INPSRC2 = 1 or 2, that is, when data are read from SDRAM. 0x6: Output data[11:0] = (read data[15:4] >> 4) & 0x0FFF 0x1: Output data[11:0] = (read data[11:0] << 1) & 0x0FFF 0x7: Output data[11:0] = (read data[15:4] >> 4) & 0x0FFF 0x0: Output data[11:0] = (read data[11:0]) & 0x0FFF 0x2: Output data[11:0] = (read data[11:0] << 2) & 0x0FFF 0x4: Output data[11:0] = (read data[11:0] << 4) & 0x0FFF 0x5: Output data[11:0] = (read data[15:4] >> 4) & 0x0FFF 0x3: Output data[11:0] = (read data[11:0] << 3) & 0x0FFF	RW	0x0
10	CLKSEL	IPIPEIF & IPIPE module pixel clock selection. This register must be set to 1 when INPSRC1 or INPSRC2 = 1 or 3, that is, data are solely read from SDRAM (VPORT inactive). 0x0: Selects the pixel clock from the VPORT. 0x1: Selects the pixel clock from the fractional clock divider. The fractional clock divider value is setup with the IPIPEIF_CLKDIV register.	RW	0
9:8	UNPACK	8-Bit, 12-bit Packed Mode When sensor raw data are stored in 8-bit packed mode or 12-bit packed mode, this register should code 1 or 3. This register is effective when INPSRC = 1 or 2. 0x0: 16 bits / pixel 0x1: 8 bits / pixel 0x3: 12 bits / pixel 0x2: 8 bits / pixel + inverse A law (8 bits to 10 bits)	RW	0x0
7	AVGFILT	Averaging Filter It applies (1,2,1) filter for the RGB/YCbCr data. *This bit field is latched by VD. 0x0: disable 0x1: enable	RW	0
6:4	RESERVED		R	0x0
3:2	INPSRC2	Selects the source for the mux (ISIF / SDRAM) as well as the data format type. 0x0: ISIF 0x1: SDRAM_RAW 0x3: SDRAM_YUV 0x2: ISIF_DARKFM Input ports to DFS submodule are VPORT and SDRAM.	RW	0x0
1	DECIM	Pixel Decimation The decimation rate defined by RSZ register. *This bit field is latched by VD. 0x0: No decimation 0x1: Decimation	RW	0
0	ONESHOT	One Shot Mode This register is available when INPSRC = 1 or 3. 0x0: Continuous mode 0x1: One shot mode	RW	0



**Table 8-1128. Register Call Summary for Register IPIPEIF\_CFG1**

ISS ISP

- ISS ISP IPIPEIF Data Path Selection: [0] [1] [2] [3]
- ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 0: [4] [5]
- ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 1: [6] [7]
- ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 2: [8] [9]
- ISS ISP IPIPEIF INPSRC1= 0 and INPSRC2 = 3: [10] [11]
- ISS ISP IPIPEIF INPSRC1= 1 and INPSRC2 = 0: [12] [13]
- ISS ISP IPIPEIF INPSRC1= 2 and INPSRC2 = 0: [14] [15]
- ISS ISP IPIPEIF INPSRC1= 3 and INPSRC2 = 0: [16] [17]
- ISS ISP IPIPEIF Timing Generation: [18] [19] [20] [21] [22] [23] [24] [25]
- ISS ISP IPIPEIF Fractional Clock Divider: [26] [27] [28]
- ISS ISP IPIPEIF Decompression (DCPM) Subblock: Unpack and Decompression Function: [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41]
- ISS ISP IPIPEIF Dark-Frame Subtraction Functionality: [42] [43] [44] [45] [46]
- ISS ISP IPIPEIF (1, 2, 1) Averaging Filter for IPIPE Data Path: [47] [48] [49] [50] [51]
- ISS ISP IPIPEIF Horizontal Pixel Decimator (Downsizer) for IPIPE Data Path: [52]
- ISS ISP IPIPEIF YUV422 8-bits packed Data Input Coming from ISIF Module: [53]
- ISS ISP IPIPEIF YUV4:2:0 Data Input for Memory-to-Memory Resize Operations: [54] [55] [56] [57] [58] [59] [60] [61]
- ISS ISP IPIPEIF Register Setup: [62] [63] [64] [65] [66] [67] [68] [69] [70] [71]
- ISS ISP IPIPEIF Enable/Disable Hardware: [72] [73] [74] [75] [76]
- ISS ISP IPIPEIF Register Accessibility During Frame Processing: [77] [78]
- ISS ISP IPIPEIF Summary of Constraints: [79] [80]
- ISS IPIPEIF Register Summary: [81]
- ISS IPIPEIF Register Description: [82] [83] [84] [85]

**Table 8-1129. IPIPEIF\_PPLN**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Interval of HD / Start pixel in HD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		PPLN													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	PPLN	Case-1: Interval of Horizontal Sync (HD) Specifies the interval of horizontal sync. This register is available when INPSRC = 1 or 3. Case-2: Start Pixel in Horizontal Sync (HD) Specifies the start pixel in horizontal sync. This register is available when INPSRC = 2 *This bit field is latched by VD.	RW	0x0000

**Table 8-1130. Register Call Summary for Register IPIPEIF\_PPLN**

## ISS ISP

- [ISS ISP IPIPEIF Input From BL: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS ISP IPIPEIF Register Setup: \[4\]](#)
- [ISS ISP IPIPEIF Register Accessibility During Frame Processing: \[5\]](#)
- [ISS ISP IPIPEIF Summary of Constraints: \[6\]](#)
- [ISS IPIPEIF Register Summary: \[7\]](#)

**Table 8-1131. IPIPEIF\_LPFR**

<b>Address Offset</b>	0x0000 000C	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Interval of VD / Start line in VD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		LPFR													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	LPFR	Case-1: Interval of Vertical Sync (VD) Specifies the interval of vertical sync. This register is available when INPSRC = 1 or 3. Case-2: Start Pixel in Vertical Sync (VD) Specifies the start line in vertical sync. This register is available when INPSRC = 2 *This bit field is latched by VD.	RW	0x0000

**Table 8-1132. Register Call Summary for Register IPIPEIF\_LPFR**

## ISS ISP

- [ISS ISP IPIPEIF Input From BL: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP IPIPEIF Register Setup: \[5\]](#)
- [ISS ISP IPIPEIF Summary of Constraints: \[6\] \[7\]](#)
- [ISS IPIPEIF Register Summary: \[8\]](#)

**Table 8-1133. IPIPEIF\_HNUM**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Number of valid pixels per line		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		HNUM													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12:0	HNUM	The Number of Valid Pixels in a Line Specifies the number of valid pixels in a horizontal line. This register is available when INPSRC = 1, 2 or 3 *This bit field is latched by VD.	RW	0x0000

**Table 8-1134. Register Call Summary for Register IPIPEIF\_HNUM**

ISS ISP

- ISS ISP IPIPEIF Input From BL: [0] [1] [2] [3]
- ISS ISP IPIPEIF Register Setup: [4]
- ISS ISP IPIPEIF Register Accessibility During Frame Processing: [5]
- ISS ISP IPIPEIF Summary of Constraints: [6]
- ISS IPIPEIF Register Summary: [7]

**Table 8-1135. IPIPEIF\_VNUM**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Number of valid lines per frame		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VNUM															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:0	VNUM	The Number of Valid Line in a Vertical Specifies the number of valid line in a vertical. This register is available when INPSRC = 1, 2 or 3 *This bit field is latched by VD.	RW	0x0000

**Table 8-1136. Register Call Summary for Register IPIPEIF\_VNUM**

ISS ISP

- ISS ISP IPIPEIF Input From BL: [0] [1] [2] [3]
- ISS ISP IPIPEIF Register Setup: [4]
- ISS ISP IPIPEIF Register Accessibility During Frame Processing: [5]
- ISS ISP IPIPEIF Summary of Constraints: [6]
- ISS IPIPEIF Register Summary: [7]

**Table 8-1137. IPIPEIF\_ADDRU**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Memory Address (Upper)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDRU															

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10:0	ADDRU	Memory Address - Upper Memory address upper 11-bits are specified in units of 32-bytes This register is available when INPSRC = 1, 2 or 3. *This bit field is latched by VD.	RW	0x000

**Table 8-1138. Register Call Summary for Register IPIPEIF\_ADDRU**

ISS ISP

- [ISS ISP IPIPEIF Input From BL: \[0\]](#)
- [ISS ISP IPIPEIF Register Setup: \[1\]](#)
- [ISS ISP IPIPEIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS IPIPEIF Register Summary: \[3\]](#)

**Table 8-1139. IPIPEIF\_ADDRLL**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Memory Address (Lower)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDRLL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ADDRLL	Memory Address - Lower Memory address lower 16-bits are specified in units of 32-bytes. This register is available when INPSRC = 1, 2 or 3. *This bit field is latched by VD.	RW	0x0000

**Table 8-1140. Register Call Summary for Register IPIPEIF\_ADDRLL**

ISS ISP

- [ISS ISP IPIPEIF Input From BL: \[0\]](#)
- [ISS ISP IPIPEIF Register Setup: \[1\]](#)
- [ISS IPIPEIF Register Summary: \[2\]](#)

**Table 8-1141. IPIPEIF\_ADOFS**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Address offset		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADOFS															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	ADOF5	Specifies the SDRAM stride for each line in units of 32-bytes. This register is available when reading data from SDRAM: INPSRC1 or INPSRC2 = 1, 2 or 3. Assuming that the first line is at position ADDR, the second line is at address ADDR+ ADOFS, etc. *This bit field is latched by VD.	RW	0x000

**Table 8-1142. Register Call Summary for Register IPIPEIF\_ADOFS**

ISS ISP

- [ISS ISP IPIPEIF Input From BL: \[0\]](#)
- [ISS ISP IPIPEIF Register Setup: \[1\]](#)
- [ISS ISP IPIPEIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS IPIPEIF Register Summary: \[3\]](#)

**Table 8-1143. IPIPEIF\_RSZ**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Horizontal Resizing Parameter on IPIPE data path		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RSZ															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6:0	RSZ	Horizontal Resizing Parameter for IPIPE data path Specifies the horizontal resizing parameter. The RSZ register can be configured within 16 to 112 range. This resizing ratio is determined by 16/RSZ (= 1/1 to 1/7) *This bit field is latched by VD.	RW	0x10

**Table 8-1144. Register Call Summary for Register IPIPEIF\_RSZ**

ISS ISP

- [ISS ISP IPIPEIF Horizontal Pixel Decimator \(Downsizer\) for IPIPE Data Path: \[0\]](#)
- [ISS ISP IPIPEIF Register Setup: \[1\]](#)
- [ISS ISP IPIPEIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS IPIPEIF Register Summary: \[3\]](#)

**Table 8-1145. IPIPEIF\_GAIN**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Gain Parameter		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																GAIN															

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9:0	GAIN	Gain Parameter Specifies the gain applied to RAW data before it is forwarded to the IPIPE module. The gain value is expressed using the U10Q9 fractional format. The range is from 0.00195 (1/512) to 1.99805(1023/512). By default the unity gain is applied, that is, <code>IPIPEIF_GAIN.GAIN = 0x200</code> . The gain is applied to RAW data only ( <code>IPIPEIF_CFG1.INPSRC2 != 3</code> ): the gain is not applied if the input data is YCbCr. *This bit field is latched by VD.	RW	0x200

**Table 8-1146. Register Call Summary for Register IPIPEIF\_GAIN**

ISS ISP

- [ISS ISP IPIPEIF RAW Data Gain for IPIPE Data Path: \[0\]](#)
- [ISS ISP IPIPEIF Register Setup: \[1\]](#)
- [ISS ISP IPIPEIF Register Accessibility During Frame Processing: \[2\]](#)
- [ISS IPIPEIF Register Summary: \[3\]](#)
- [ISS IPIPEIF Register Description: \[4\]](#)

**Table 8-1147. IPIPEIF\_DPCM**

<b>Address Offset</b>	0x0000 002C													
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>	<b>Instance</b> ISS_IPIPEIF												
<b>Description</b>	IPIPEIF DPCM configuration This register applies only if <code>IPIPEIF_CFG1.UNPACK = 1</code> , that is, RAW8 data is read from SDRAM.													
<b>Type</b>	RW													
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0												
	RESERVED											BITS	PRED	ENA

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	BITS	DPCM bit mode for SDRAM data 0x0: 8bit to 10bit DPCM decompression 0x1: 8bit to 12bit DPCM decompression	RW	0
1	PRED	DPCM prediction mode for SDRAM data 0x0: Simple predictor 0x1: Advanced predictor	RW	0
0	ENA	DPCM decompression enable for SDRAM data. 0x0: DPCM off (no decompression) 0x1: DPCM on	RW	0

**Table 8-1148. Register Call Summary for Register IPIPEIF\_DPCM**

ISS ISP

- [ISS ISP IPIPEIF Decompression \(DCPM\) Subblock: Unpack and Decompression Function: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [ISS ISP IPIPEIF Register Setup: \[7\] \[8\] \[9\]](#)
- [ISS IPIPEIF Register Summary: \[10\]](#)

**Table 8-1149. IPIPEIF\_CFG2**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF Configuration #2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								YUV8P	YUV8	DFSDIR	RESERVED	YUV16	VDPOL	HDPOL	INTSW

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	YUV8P	8-bit YUV data unpacking to 16 bits When <a href="#">IPIPEIF_CFG1.INPSRC2</a> = 0 and <a href="#">IPIPEIF_CFG2.YUV16</a> = 1, the 8-bit YUV data are transformed into 16-bit YUV data. The way the data are unpacked from 8 bits to 16 bits is controlled by the <a href="#">IPIPEIF_CFG2.YUV8P</a> register. The upper 8 bits of the 16-bit output are set to 0.  0x0: Y output on even pixels C output on odd pixels  0x1: C output on even pixels Y output on odd pixels	RW	0
6	YUV8	YUV 8bit mode When <a href="#">ISIF_CFG1.INPSRC2</a> = 0 and <a href="#">YUV16</a> = 1, setting this bit to 1 enables the conversion from 8-bit YUV input to 16-bit YUV. This register is used when the input data from the ISIF module is 8-bit YUV data.  0x0: YUV16 0x1: YUV8 to 16	RW	0
5	DFSDIR	DFS direction Selects the direction of dark frame subtraction.  0x0: VPORT IF(capture frame) - SDRAM (dark frame) 0x1: SDRAM (capture frame) - VPORT IF(dark frame)	RW	0
4	RESERVED	Read returns reset value	RW	0
3	YUV16	Data type selection. The behavior of this bit field depends upon other register settings. The functionality is best explained with the following pseudo code: if (( <a href="#">CFG1.INPSRC2</a> ==0 && <a href="#">CFG2.YUV16</a> )    <a href="#">CFG1.INPSRC2</a> ==3) { data_out[15:0] = yuv[15:0] } else if ( <a href="#">CFG1.INPSRC2</a> ==1 && <a href="#">CFG2.YUV16</a> && <a href="#">CFG1.UNPACK</a> =1) { data_out[15:8] = gain_clip[7:0]; data_out[ 7:0] = 0; } else { data_out[15:12] = 0; data_out[11: 0] = gain_clip[11:0]; } where: o data_out[15:0] = 16-bit YUV or 12-bit RAW data to ipipe o yuv[15:0] = 16-bit YUV data from "horizontal pixel decimator" block. o gain_clip[11:0] = 12-bit RAW data from "gain" block.  0x0: 12-bit RAW data 0x1: 16-bit YUV data	RW	0

Bits	Field Name	Description	Type	Reset
2	VDPOL	VD Sync Polarity When input VD is active low SYNC pulse, this bit needs to be set to 1.  0x0: Positive 0x1: Negative	RW	0
1	HDPOL	HD Sync Polarity When input HD is active low SYNC pulse, this bit needs to be set to 1.  0x0: Positive 0x1: Negative	RW	0
0	INTSW	IPIPEIF interrupt source selection. This register select the interrupt source.  0x0: Start position of VD from VPORT interface 0x1: Start position of VD from ISIF module	RW	0

**Table 8-1150. Register Call Summary for Register IPIPEIF\_CFG2**

## ISS ISP

- [ISS ISP IPIPEIF DFS Subtraction Direction: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [ISS ISP IPIPEIF \(1, 2, 1\) Averaging Filter for IPIPE Data Path: \[6\]](#)
- [ISS ISP IPIPEIF YUV422 8-bits packed Data Input Coming from ISIF Module: \[7\] \[8\] \[9\]](#)
- [ISS ISP IPIPEIF YUV4:2:0 Data Input for Memory-to-Memory Resize Operations: \[10\] \[11\]](#)
- [ISS ISP IPIPEIF Module Events and Status Checking: \[12\]](#)
- [ISS ISP IPIPEIF Register Setup: \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\]](#)
- [ISS IPIPEIF Register Summary: \[21\]](#)
- [ISS IPIPEIF Register Description: \[22\] \[23\]](#)

**Table 8-1151. IPIPEIF\_INIRSZ**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF resize initial position - IPIPE data path.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ALNSYNC	INIRSZ														

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13	ALNSYNC	Align the HSYNC, VSYNC to initial position defined by INIRSZ.  0x0: Disable 0x1: Enable	RW	0
12:0	INIRSZ	Offset used to re-initialize the HD/VD position after resizer. From 0 to 8191 PCLK cycles. Skips INIRSZ pixels for every line.	RW	0x0000



**Table 8-1152. Register Call Summary for Register IPIPEIF\_INIRSZ**

ISS ISP

- ISS ISP IPIPEIF (1, 2, 1) Averaging Filter for IPIPE Data Path: [0]
- ISS ISP IPIPEIF Horizontal Pixel Decimator (Downsizer) for IPIPE Data Path: [1] [2]
- ISS ISP IPIPEIF Register Setup: [3] [4]
- ISS IPIPEIF Register Summary: [5]

**Table 8-1153. IPIPEIF\_OCLIP**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF output clipping value		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OCLIP															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	OCLIP	Output clipping value after gain control on IPIPE data path. This value is in U12Q0 data format.	RW	0xFF

**Table 8-1154. Register Call Summary for Register IPIPEIF\_OCLIP**

ISS ISP

- ISS ISP IPIPEIF RAW Data Gain for IPIPE Data Path: [0]
- ISS ISP IPIPEIF Register Setup: [1]
- ISS IPIPEIF Register Summary: [2]

**Table 8-1155. IPIPEIF\_DTUDF**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF data underflow detection		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											DTUDF				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:1	RESERVED		R	0x0000
0	DTUDF	Data under flow error status register. Reading 1 shows there is data under flow and at least one data is corrupted while reading from SDRAM. Writing 1 to this register clears (=0) the error (=1) status. Underflow errors are non recoverable at ISP level, need to do a soft reset at ISS level. The IPIPEIF_UDF interrupt is generated when an underflow happens. The interrupt avoids polling this register for errors.	RW	0

**Table 8-1156. Register Call Summary for Register IPIPEIF\_DTUDF**

ISS ISP

- [ISS IPIPEIF Register Summary: \[0\]](#)

**Table 8-1157. IPIPEIF\_CLKDIV**

<b>Address Offset</b>	0x0000 0040		<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>			
<b>Description</b>	IPIPEIF CLOCK DIVIDER			
<b>Type</b>	RW			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKDIV															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	CLKDIV	IPIPEIF clock rate configuration IPIPE/IPIPEIF clock frequency = M/N x clk_vpss clock frequency. We have M = CLKDIV[15:8] + 1 and N = CLKDIV[7:0] + 1 When clk_vpss > 121.5 MHz, M/N must not be greater than 1/2 value This register is available when <a href="#">IPIPEIF_CFG1.CLKSEL</a> = 1.	RW	0x0001

**Table 8-1158. Register Call Summary for Register IPIPEIF\_CLKDIV**

ISS ISP

- [ISS ISP IPIPEIF Timing Generation: \[0\]](#)
- [ISS ISP IPIPEIF Fractional Clock Divider: \[1\] \[2\]](#)
- [ISS IPIPEIF Register Summary: \[3\]](#)
- [ISS IPIPEIF Register Description: \[4\]](#)

**Table 8-1159. IPIPEIF\_DPC1**

<b>Address Offset</b>	0x0000 0044		<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>			
<b>Description</b>	IPIPEIF defect pixel correction #1			
<b>Type</b>	RW			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	ENA	TH													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:13	RESERVED		R	0x0
12	ENA	DPC enable. Applies DPC for video port data, ISIF input path. 0x0: Disable 0x1: Enable	RW	0
11:0	TH	DPC threshold value	RW	0x000

**Table 8-1160. Register Call Summary for Register IPIPEIF\_DPC1**

ISS ISP

- [ISS ISP IPIPEIF Defect Pixel Correction: \[0\] \[1\]](#)
- [ISS ISP IPIPEIF Register Setup: \[2\] \[3\]](#)
- [ISS IPIPEIF Register Summary: \[4\]](#)

**Table 8-1161. IPIPEIF\_DPC2**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF defect pixel correction #2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENA	TH														

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	ENA	DPC enable. Applies DPC for SDRAM input path. 0x0: Disable 0x1: Enable	RW	0
11:0	TH	DPC threshold value	RW	0x000

**Table 8-1162. Register Call Summary for Register IPIPEIF\_DPC2**

ISS ISP

- [ISS ISP IPIPEIF Defect Pixel Correction: \[0\] \[1\]](#)
- [ISS ISP IPIPEIF Register Setup: \[2\] \[3\]](#)
- [ISS IPIPEIF Register Summary: \[4\]](#)

**Table 8-1163. IPIPEIF\_RSZ3A**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	ISS_IPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF HORIZONTAL RESIZING PARAMETER FOR H3A		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																RESERVED				DECIM	AVGFILT	RESERVED	RSZ										

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:10	RESERVED		R	0x00
9	DECIM	Pixel Decimation Enable The decimation rate defined by the RSZ bit field. *This bit field is latched by VD. 0x0: No Decimation 0x1: Decimate	RW	0

Bits	Field Name	Description	Type	Reset
8	AVGFILT	Averaging Filter It applies a (1, 2, 1) filter for the RGB/YCbCr data. *This bit field is latched by VD.  0x0: Disable 0x1: Enable	RW	0
7	RESERVED		R	0
6:0	RSZ	Horizontal Resizing Parameter for H3A data path Specifies the horizontal resizing parameter. The RSZ register can be configured within 16 to 112 range. This resizing ratio is determined by 16/RSZ (= 1/1 to 1/7) *This bit field is latched by VD.	RW	0x10

**Table 8-1164. Register Call Summary for Register IPIPEIF\_RSZ3A**

ISS ISP

- [ISS ISP IPIPEIF \(1, 2, 1\) Averaging Filter for H3A Data Path: \[0\] \[1\]](#)
- [ISS ISP IPIPEIF Horizontal Pixel Decimator \(Downsizer\) for H3A Data Path: \[2\] \[3\]](#)
- [ISS ISP IPIPEIF Register Setup: \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [ISS ISP IPIPEIF Register Accessibility During Frame Processing: \[9\] \[10\] \[11\]](#)
- [ISS IPIPEIF Register Summary: \[12\]](#)

**Table 8-1165. IPIPEIF\_INIRSZ3A**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	ISS_IPPIPEIF
<b>Physical Address</b>	See <a href="#">Table 8-1124</a>		
<b>Description</b>	IPIPEIF resize initial position - H3A data path.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ALNSYNC	INIRSZ														

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13	ALNSYNC	Align the HD, VD to initial position defined by the INIRSZ bit field. It means that HD and VD are effectively shifted by INIRSZ pixel clock cycles.  0x0: Disable 0x1: Enable	RW	0
12:0	INIRSZ	Offset used to re-initialize the HD/VD position after resizer. From 0 to 8191 PCLK cycles. Skips INIRSZ pixels for every line.	RW	0x0000

**Table 8-1166. Register Call Summary for Register IPIPEIF\_INIRSZ3A**

ISS ISP

- [ISS ISP IPIPEIF Horizontal Pixel Decimator \(Downsizer\) for H3A Data Path: \[0\] \[1\]](#)
- [ISS IPIPEIF Register Summary: \[2\]](#)

### 8.3.5.8 ISS H3A Registers

#### 8.3.5.8.1 ISS H3A Register Summary

**Table 8-1167. ISS H3A Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ISS_H3A Base Address
H3A_PID	R	32	0x0000 0000	0x5201 1400
H3A_PCR	RW	32	0x0000 0004	0x5201 1404
H3A_AFPAX1	RW	32	0x0000 0008	0x5201 1408
H3A_AFPAX2	RW	32	0x0000 000C	0x5201 140C
H3A_AFPAXSTART	RW	32	0x0000 0010	0x5201 1410
H3A_AFIIRSH	RW	32	0x0000 0014	0x5201 1414
H3A_AFBUFST	RW	32	0x0000 0018	0x5201 1418
H3A_AFCOEF010	RW	32	0x0000 001C	0x5201 141C
H3A_AFCOEF032	RW	32	0x0000 0020	0x5201 1420
H3A_AFCOEF054	RW	32	0x0000 0024	0x5201 1424
H3A_AFCOEF076	RW	32	0x0000 0028	0x5201 1428
H3A_AFCOEF098	RW	32	0x0000 002C	0x5201 142C
H3A_AFCOEF0010	RW	32	0x0000 0030	0x5201 1430
H3A_AFCOEF110	RW	32	0x0000 0034	0x5201 1434
H3A_AFCOEF132	RW	32	0x0000 0038	0x5201 1438
H3A_AFCOEF154	RW	32	0x0000 003C	0x5201 143C
H3A_AFCOEF176	RW	32	0x0000 0040	0x5201 1440
H3A_AFCOEF198	RW	32	0x0000 0044	0x5201 1444
H3A_AFCOEF1010	RW	32	0x0000 0048	0x5201 1448
H3A_AEWWIN1	RW	32	0x0000 004C	0x5201 144C
H3A_AEWINSTART	RW	32	0x0000 0050	0x5201 1450
H3A_AEWINBLK	RW	32	0x0000 0054	0x5201 1454
H3A_AEWSUBWIN	RW	32	0x0000 0058	0x5201 1458
H3A_AEWBUFST	RW	32	0x0000 005C	0x5201 145C
H3A_AEWCFG	RW	32	0x0000 0060	0x5201 1460
H3A_LINE_START	RW	32	0x0000 0064	0x5201 1464
H3A_VFV_CFG1	RW	32	0x0000 0068	0x5201 1468
H3A_VFV_CFG2	RW	32	0x0000 006C	0x5201 146C
H3A_VFV_CFG3	RW	32	0x0000 0070	0x5201 1470
H3A_VFV_CFG4	RW	32	0x0000 0074	0x5201 1474
H3A_HVF_THR	RW	32	0x0000 0078	0x5201 1478
H3A_ADVANCED	RW	32	0x0000 007C	0x5201 147C

**8.3.5.8.2 ISS H3A Register Description**

**Table 8-1168. H3A\_PID**

<b>Address Offset</b>	0x0000 0000																																
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>																<b>Instance</b>	ISS_H3A															
<b>Description</b>	Peripheral Revision and Class Information																																
<b>Type</b>	R																																

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME		RESERVED		FUNC												RTL				MAJOR				RESERVED		MINOR					

Bits	Field Name	Description	Type	Reset
31:30	SCHEME		R	0x1
29:28	RESERVED		R	0x0
27:16	FUNC		R	0xD01
15:11	RTL		R	0x00
10:8	MAJOR		R	0x0
7:6	RESERVED		R	0x0
5:0	MINOR		R	0x00

**Table 8-1169. Register Call Summary for Register H3A\_PID**

ISS ISP

- [ISS H3A Register Summary: \[0\]](#)

**Table 8-1170. H3A\_PCR**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	Peripheral Control Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AVE2LMT								OVF		AF_VF_EN	AEW_MED_EN	BUSYAEAWB	AEW_ALAW_EN	AEW_EN	BUSYAF	FVMODE	RGBPOS				MED_TH				AF_MED_EN	AF_ALAW_EN	AF_EN				

Bits	Field Name	Description	Type	Reset
31:22	AVE2LMT	AE/AWB Saturation Limit This is the value that all sub sampled pixels in the AE/AWB engine are compared to. If the data is greater or equal to this data then the block is considered saturated.	RW	0x3FF
21	OVF	H3A module overflow status bit. If the H3A module overflows it will keep sending data. The software can read this status bit during vertical blanking period to ensure that no overflow happened while writing out the data to SDRAM. There is also an interrupt at ISP level (H3A_OVF) which can be used to monitor this.  0x0: Read 0: No overflow pending Write 0: Status bit unchanged  0x1: Read 1: Overflow happened while writing out the data. Output data likely to be corrupted. Write 1: Clear the status bit.	RW	0
20	AF_VF_EN	AF Vertical Focus Enable  0x0: 4 Color Horizontal FV only 0x1: 1 Color Horizontal FV and 1 Color Vertical FV	RW	0
19	AEW_MED_EN	AE/AWB Median filter Enable If the median filter is enabled, then the 1st 2 and last 2 pixels in the frame are not filtered.  0x0: Disable Auto Focus median filter 0x1: Enable Auto Focus median filter	RW	0
18	BUSYAEAWB	Busy bit for AE/AWB	R	0

Bits	Field Name	Description	Type	Reset
17	AEW_ALAW_EN	AE/AWB A-law Enable 0x0: Disable Auto exposure/white balance A-law table 0x1: Enable Auto exposure/white balance A-law table.	RW	0
16	AEW_EN	AE/AWB enable 0x0: Disable Auto exposure/white balance 0x1: Enable Auto exposure/white balance	RW	0
15	BUSYAF	Busy bit for AF.	R	0
14	FVMODE	Focus Value Accumulation Mode 0x0: Sum mode. 0x1: Peak mode.	RW	0
13:11	RGBPOS	Red, Green, and blue pixel location in the AF windows RGBPOS(0): GR and GB as Bayer pattern RGBPOS(1): RG and GB as Bayer pattern RGBPOS(2): GR and BG as Bayer pattern RGBPOS(3): RG and BG as Bayer pattern RGBPOS(4): GG and RB as custom pattern RGBPOS(5): RB and GG as custom pattern 6 and 7 are reserved This Value is only used if VF is disabled	RW	0x0
10:3	MED_TH	Median filter threshold.	RW	0xFF
2	AF_MED_EN	Auto Focus Median filter Enable If the median filter is enabled, then the 1st 2 and last 2 pixels in the frame are not in the valid region. Therefore the pixel start/end and IIR filter start positions should not be set within the 1st and last 2 pixels. 0x0: Disable AF median filter. 0x1: Enable AF median filter.	RW	0
1	AF_ALAW_EN	AF A-law table enable 0x0: Disable Auto Focus A-law table 0x1: Enable Auto Focus A-law table	RW	0
0	AF_EN	AF enable 0x0: Disable Auto Focus Engine 0x1: Enable Auto Focus Engine	RW	0

**Table 8-1171. Register Call Summary for Register H3A\_PCR**

## ISS ISP

- [ISS ISP H3A Interrupts: \[0\]](#)
- [ISS ISP H3A Optional Preprocessing: \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP H3A Pixel Extraction: \[5\] \[6\]](#)
- [ISS ISP H3A HFV Accumulator: \[7\] \[8\]](#)
- [ISS ISP H3A Saturation Check: \[9\]](#)
- [ISS ISP H3A Events and Status Checking: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [ISS ISP H3A AF Engine: \[16\] \[17\] \[18\] \[19\] \[20\] \[21\]](#)
- [ISS ISP H3A AEW Engine: \[22\] \[23\] \[24\] \[25\]](#)
- [ISS ISP H3A Enable/Disable Hardware: \[26\] \[27\] \[28\] \[29\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\]](#)
- [ISS ISP H3A Interframe Operations: \[37\]](#)
- [ISS H3A Register Summary: \[38\]](#)

**Table 8-1172. H3A\_AFPAX1**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	Setup for the AF Engine Poxel Configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PAXW								RESERVED								PAXH							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	PAXW	AF Engine Poxel Width The width of the poxel is the value of this register plus 1 multiplied by 2. The minimum width is expected to be 8 pixels. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00
15:8	RESERVED		R	0x00
7:0	PAXH	AF Engine Poxel Height The height of the poxel is the value of this register plus 1 multiplied by 2 with a final value of 2-256 (even) * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00

**Table 8-1173. Register Call Summary for Register H3A\_AFPAX1**

## ISS ISP

- [ISS ISP H3A Poxel Extraction: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP H3A AF Engine: \[5\] \[6\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[7\]](#)
- [ISS ISP H3A Summary of Constraints: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [ISS H3A Register Summary: \[18\]](#)

**Table 8-1174. H3A\_AFPAX2**

<b>Address Offset</b>	0x0000 000C		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	Setup for the AF Engine Poxel Configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								AFINCH				AFINCV				PAXVC				PAXHC											

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20:17	AFINCH	AF Engine Column Increments Number of columns to increment in a poxel plus 1 multiplied by 2. Thus, the number of columns that can be skipped between two processed line pairs is 2-32 (even). The starting two columns in a poxel are first processed before this field is applied. This must be set so that there are at least 4 samples on a line when combined with the number of horizontal paxels. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x0



Bits	Field Name	Description	Type	Reset
16:13	AFINCV	AF Engine Line Increments Number of lines to increment in a Poxel plus 1 multiplied by 2. Incrementing the line in a paxel is always done on a line pair due to the fact that the RGB pattern falls in two lines. If all the lines are to be processed, this field should be set to zero, and thus line count is incremented by 2 following a line pair. Thus, the number of lines that can be skipped between two processed line pairs is 0-30 (even). The starting two lines in a paxel are first processed before this field is applied. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x0
12:6	PAXVC	AF Engine Vertical Poxel Count The number of paxels in the vertical direction plus 1. The maximum number of vertical paxels in a frame should not exceed 128. The value should be set to ensure that the bandwidth requirements and buffer size are not exceeded. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00
5:0	PAXHC	AF Engine Horizontal Poxel Count The number of paxels in the horizontal direction plus 1. It is illegal to set a number that is greater than 35 (total of 36 paxels in the horizontal direction). The minimum number of paxels should be 2 (valid range for the field is 1-35). * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00

**Table 8-1175. Register Call Summary for Register H3A\_AFPAX2**

ISS ISP

- ISS ISP H3A Poxel Extraction: [0] [1] [2] [3] [4] [5] [6]
- ISS ISP H3A AF Engine: [7] [8] [9] [10]
- ISS ISP H3A Register Accessibility During Frame Processing: [11]
- ISS ISP H3A Summary of Constraints: [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28]
- ISS H3A Register Summary: [29]

**Table 8-1176. H3A\_AFPAXSTART**

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b> ISS_H3A
<b>Description</b>	Start Position for AF Engine Paxels	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PAXSH								RESERVED				PAXSV											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	PAXSH	AF Engine Poxel Horizontal start position Range: 2-4094 PAXSH must be equal to or greater than (IIRSH + 2) This value must be even if Vertical mode is not enabled. If Vertical mode is enabled then the lower bit of PAXSH and IIRSH must be equal. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000
15:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11:0	PAXSV	AF Engine Poxel Vertical start position Range: 0-4095 Sets the vertical line for the first poxel. This value must be greater then or equal to 8 if the vertical mode is enabled. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000

**Table 8-1177. Register Call Summary for Register H3A\_AFPAXSTART**

ISS ISP

- [ISS ISP H3A Poxel Extraction: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [ISS ISP H3A AF Engine: \[5\] \[6\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[7\]](#)
- [ISS ISP H3A Summary of Constraints: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [ISS H3A Register Summary: \[15\]](#)

**Table 8-1178. H3A\_AFIIRSH**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	Start Position for IIRSH		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IIRSH															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	IIRSH	AF Engine IIR Horizontal Start Position Range from 0-4094. When the horizontal position of a line equals this value the shift registers are cleared on the next pixel. This value must be even if Vertical mode is not enabled. If vertical mode is enabled then the lower bit must match the poxel horizontal start position. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000

**Table 8-1179. Register Call Summary for Register H3A\_AFIIRSH**

ISS ISP

- [ISS ISP H3A Poxel Extraction: \[0\] \[1\]](#)
- [ISS ISP H3A Horizontal FV Calculator: \[2\]](#)
- [ISS ISP H3A AF Engine: \[3\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[4\]](#)
- [ISS ISP H3A Summary of Constraints: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [ISS H3A Register Summary: \[15\]](#)

**Table 8-1180. H3A\_AFBUFST**

<b>Address Offset</b>	0x0000 0018	
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b> ISS_H3A
<b>Description</b>	SDRAM destination address for AF engine statistics	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFBUFST																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	AFBUFST	SDRAM destination address for AF engine statistics The SDRAM destination address for the AF statistics. The 6 LSBs are ignored, address must be on a 64-byte boundary. This field can be altered even when the AF is busy. Change will take place only for the next frame. However, note that reading this register will always give the latest value.	RW	0x0000000
4:0	RESERVED		R	0x00

**Table 8-1181. Register Call Summary for Register H3A\_AFBUFST**

ISS ISP

- [ISS ISP H3A DMA Interface: \[0\] \[1\] \[2\]](#)
- [ISS ISP H3A AF Engine: \[3\]](#)
- [ISS ISP H3A Summary of Constraints: \[4\]](#)
- [ISS H3A Register Summary: \[5\]](#)

**Table 8-1182. H3A\_AFCOEF010**

<b>Address Offset</b>	0x0000 001C	
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b> ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 0.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				COEFF1								RESERVED				COEFF0															

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF1	AF Engine IIR filter Coefficient #1 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF0	AF Engine IIR filter Coefficient #0 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1183. Register Call Summary for Register H3A\_AFCOEF010**

ISS ISP

- [ISS ISP H3A Horizontal FV Calculator: \[0\]](#)
- [ISS ISP H3A AF Engine: \[1\]](#)
- [ISS H3A Register Summary: \[2\]](#)

**Table 8-1184. H3A\_AFCOEF032**

<b>Address Offset</b>	0x0000 0020		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 0.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF3								RESERVED								COEFF2							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF3	AF Engine IIR filter Coefficient #3 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF2	AF Engine IIR filter Coefficient #2 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1185. Register Call Summary for Register H3A\_AFCOEF032**

ISS ISP

- [ISS ISP H3A AF Engine: \[0\]](#)
- [ISS H3A Register Summary: \[1\]](#)

**Table 8-1186. H3A\_AFCOEF054**

<b>Address Offset</b>	0x0000 0024		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 0.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF5								RESERVED								COEFF4							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF5	AF Engine IIR filter Coefficient #5 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF4	AF Engine IIR filter Coefficient #4 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1187. Register Call Summary for Register H3A\_AFCOEF054**

ISS ISP

- [ISS ISP H3A AF Engine: \[0\]](#)
- [ISS H3A Register Summary: \[1\]](#)

**Table 8-1188. H3A\_AFCEOF076**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 0.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF7								RESERVED								COEFF6							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF7	AF Engine IIR filter Coefficient #7 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF6	AF Engine IIR filter Coefficient #6 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1189. Register Call Summary for Register H3A\_AFCEOF076**

ISS ISP

- [ISS ISP H3A AF Engine: \[0\]](#)
- [ISS H3A Register Summary: \[1\]](#)

**Table 8-1190. H3A\_AFCEOF098**

<b>Address Offset</b>	0x0000 002C		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 0.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF9								RESERVED								COEFF8							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF9	AF Engine IIR filter Coefficient #9 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF8	AF Engine IIR filter Coefficient #8 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1191. Register Call Summary for Register H3A\_AFCEOF098**

ISS ISP

- [ISS ISP H3A AF Engine: \[0\]](#)
- [ISS H3A Register Summary: \[1\]](#)

**Table 8-1192. H3A\_AFCOEF0010**

<b>Address Offset</b>	0x0000 0030	
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b> ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 0.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																COEFF10															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	COEFF10	AF Engine IIR filter Coefficient #10 (Set 0) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1193. Register Call Summary for Register H3A\_AFCOEF0010**

ISS ISP

- [ISS ISP H3A Horizontal FV Calculator: \[0\]](#)
- [ISS ISP H3A AF Engine: \[1\]](#)
- [ISS H3A Register Summary: \[2\]](#)

**Table 8-1194. H3A\_AFCOEF110**

<b>Address Offset</b>	0x0000 0034	
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b> ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 1.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF1								RESERVED								COEFF0							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF1	AF Engine IIR filter Coefficient #1 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF0	AF Engine IIR filter Coefficient #0 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1195. Register Call Summary for Register H3A\_AFCOEF110**

ISS ISP

- [ISS ISP H3A Horizontal FV Calculator: \[0\]](#)
- [ISS ISP H3A AF Engine: \[1\]](#)
- [ISS H3A Register Summary: \[2\]](#)

**Table 8-1196. H3A\_AFCEOF132**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 1.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF3								RESERVED								COEFF2							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF3	AF Engine IIR filter Coefficient #3 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF2	AF Engine IIR filter Coefficient #2 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1197. Register Call Summary for Register H3A\_AFCEOF132**

ISS ISP

- [ISS ISP H3A AF Engine: \[0\]](#)
- [ISS H3A Register Summary: \[1\]](#)

**Table 8-1198. H3A\_AFCEOF154**

<b>Address Offset</b>	0x0000 003C		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 1.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF5								RESERVED								COEFF4							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF5	AF Engine IIR filter Coefficient #5 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF4	AF Engine IIR filter Coefficient #4 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1199. Register Call Summary for Register H3A\_AFCEOF154**

ISS ISP

- [ISS ISP H3A AF Engine: \[0\]](#)
- [ISS H3A Register Summary: \[1\]](#)

**Table 8-1200. H3A\_AFCEOF176**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 1.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF7								RESERVED								COEFF6							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF7	AF Engine IIR filter Coefficient #7 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF6	AF Engine IIR filter Coefficient #6 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1201. Register Call Summary for Register H3A\_AFCEOF176**

ISS ISP

- [ISS ISP H3A AF Engine: \[0\]](#)
- [ISS H3A Register Summary: \[1\]](#)

**Table 8-1202. H3A\_AFCEOF198**

<b>Address Offset</b>	0x0000 0044		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 1.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COEFF9								RESERVED								COEFF8							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	COEFF9	AF Engine IIR filter Coefficient #9 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000
15:12	RESERVED		R	0x0
11:0	COEFF8	AF Engine IIR filter Coefficient #8 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1203. Register Call Summary for Register H3A\_AFCEOF198**

ISS ISP

- [ISS ISP H3A AF Engine: \[0\]](#)
- [ISS H3A Register Summary: \[1\]](#)



**Table 8-1204. H3A\_AFCOEF1010**

<b>Address Offset</b>	0x0000 0048		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	IIR filter coefficient data for SET 1.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COEFF10																			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	COEFF10	AF Engine IIR filter Coefficient #10 (Set 1) The range is signed -32 <= value <= 31 +63/64	RW	0x000

**Table 8-1205. Register Call Summary for Register H3A\_AFCOEF1010**

ISS ISP

- [ISS ISP H3A Horizontal FV Calculator: \[0\]](#)
- [ISS ISP H3A AF Engine: \[1\]](#)
- [ISS H3A Register Summary: \[2\]](#)

**Table 8-1206. H3A\_AEWWIN1**

<b>Address Offset</b>	0x0000 004C		
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b>	ISS_H3A
<b>Description</b>	Configuration for AE/AWB Windows.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WINH								RESERVED	WINW								WINVC				WINHC										

Bits	Field Name	Description	Type	Reset
31:24	WINH	AE/AWB Engine Window Height This specifies the window height in an even number of pixels, the window height is the value plus 1 multiplied by 2. The final value can be from 2-512 (even) * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00
23:21	RESERVED		R	0x0
20:13	WINW	AE/AWB Engine Window Width This specifies the window width in an even number of pixels, the window width is the value plus 1 multiplied by 2. The minimum width is expected to be 8 pixels. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00
12:6	WINVC	AE/AWB Engine Vertical Window Count The number of windows in the vertical direction plus 1. The maximum number of vertical windows in a frame should not exceed 128. The value should be set to ensure that the bandwidth requirements and buffer size are not exceeded * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00

Bits	Field Name	Description	Type	Reset
5:0	WINHC	AE/AWB Engine Horizontal Window Count The number of horizontal windows plus 1. The maximum number of horizontal windows is 35 plus 1 (36). The minimum number of windows should be 2 (valid range for the field is 1-35). * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00

**Table 8-1207. Register Call Summary for Register H3A\_AEWIN1**

ISS ISP

- [ISS ISP H3A Sub-sampler: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS ISP H3A Additional Black Row of AE/AWB Windows: \[4\] \[5\] \[6\] \[7\]](#)
- [ISS ISP H3A AEW Engine: \[8\] \[9\] \[10\] \[11\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[12\]](#)
- [ISS ISP H3A Summary of Constraints: \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\]](#)
- [ISS H3A Register Summary: \[23\]](#)

**Table 8-1208. H3A\_AEWINSTART**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	Start position for AE/AWB Windows.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WINSV								RESERVED								WINSH							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	WINSV	AE/AWB Engine Vertical Window Start Position Sets the first line for the first window. Range 0-4095 * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000
15:12	RESERVED		R	0x0
11:0	WINSH	AE/AWB Engine Horizontal Window Start Position Sets the horizontal position for the first window on each line. Range 0-4095 * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x000

**Table 8-1209. Register Call Summary for Register H3A\_AEWINSTART**

ISS ISP

- [ISS ISP H3A Sub-sampler: \[0\] \[1\]](#)
- [ISS ISP H3A Additional Black Row of AE/AWB Windows: \[2\] \[3\]](#)
- [ISS ISP H3A AEW Engine: \[4\] \[5\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[6\]](#)
- [ISS H3A Register Summary: \[7\]](#)

**Table 8-1210. H3A\_AEWINBLK**

<b>Address Offset</b>	0x0000 0054	
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b> ISS_H3A
<b>Description</b>	Start position and height for black line of AE/AWB Windows	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WINSV								RESERVED								WINH							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	WINSV	AE/AWB Engine Vertical Window Start Position for single black line of windows Sets the first line for the single black line of windows. * This value is shadowed and latched on the rising edge of VSYNC. Range 0-4095 Note that the horizontal start and the horizontal number of windows will be similar to the regular windows	RW	0x000
15:7	RESERVED		R	0x000
6:0	WINH	AE/AWB Engine Window Height for the single black line of windows This specifies the window height in an even number of pixels, the window height is the value plus 1 multiplied by 2. The final value can be from 2-256 (even) * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x00

**Table 8-1211. Register Call Summary for Register H3A\_AEWINBLK**

ISS ISP

- [ISS ISP H3A Additional Black Row of AE/AWB Windows: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS ISP H3A AEW Engine: \[4\] \[5\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[6\]](#)
- [ISS ISP H3A Summary of Constraints: \[7\] \[8\]](#)
- [ISS H3A Register Summary: \[9\]](#)

**Table 8-1212. H3A\_AEWSUBWIN**

<b>Address Offset</b>	0x0000 0058	
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b> ISS_H3A
<b>Description</b>	Configuration for subsample data in AE/AWB window.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AEWINCV				RESERVED				AEWINCH							

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:8	AEWINCV	AE/AWB Engine Vertical Sampling Point Increment Sets vertical distance between sub-samples within a window plus 1 multiplied by 2. The final range is 2-32. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x0
7:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	AEWINCH	AE/AWB Engine Horizontal Sampling Point Increment Sets horizontal distance between sub-samples within a window plus 1 multiplied by 2. The final range is 2-32. * This value is shadowed and latched on the rising edge of VSYNC.	RW	0x0

**Table 8-1213. Register Call Summary for Register H3A\_AEWSUBWIN**

## ISS ISP

- [ISS ISP H3A Sub-sampler: \[0\] \[1\]](#)
- [ISS ISP H3A Additional Black Row of AE/AWB Windows: \[2\] \[3\]](#)
- [ISS ISP H3A AEW Engine: \[4\] \[5\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[6\]](#)
- [ISS ISP H3A Summary of Constraints: \[7\]](#)
- [ISS H3A Register Summary: \[8\]](#)

**Table 8-1214. H3A\_AEWBUFST**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	SDRAM destination address for AE/AWB engine statistics		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AEWBUFST																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	AEWBUFST	SDRAM destination address for AE/AWB engine statistics The start location in SDRAM for the AE/AWB statistics. The 6 LSB are ignored, address should be on a 64-byte boundary This field can be altered even when the AE/AWB is busy. Change will take place only for the next frame. However, note that reading this register will always give the latest value.	RW	0x0000000
4:0	RESERVED		R	0x00

**Table 8-1215. Register Call Summary for Register H3A\_AEWBUFST**

## ISS ISP

- [ISS ISP H3A DMA Interface: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\]](#)
- [ISS ISP H3A AEW Engine: \[28\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[29\]](#)
- [ISS H3A Register Summary: \[30\]](#)

**Table 8-1216. H3A\_AEWCFG**

<b>Address Offset</b>	0x0000 0060	
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b> ISS_H3A
<b>Description</b>	Configuration for AE/AWB	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AEFMT	RESERVED				SUMSHFT										

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9:8	AEFMT	AE/AWB output format 0 = sum of squares 1 = min/max 2 = sum only; no sum of squares or min/max * This value is shadowed and latched on the rising edge of VSYNC	RW	0x0
7:4	RESERVED		R	0x0
3:0	SUMSHFT	AE/AWB engine shift value for the accumulation of pixel values This bit field sets the right shift value which is applied on the result of the pixel accumulation before it is stored in the packet. The accumulation takes place on 26 bits which is enough for 10-bit data and a maximum window size of 512 x 512 which results into the accumulation of 256 x 256 pixels of the same color. The shift value must be set such that the result fits on 16 bits. SUMSHFT = right shift value. Range: 0 -15 * This value is shadowed and latched on the rising edge of VSYNC	RW	0x0

**Table 8-1217. Register Call Summary for Register H3A\_AEWCFG**

ISS ISP

- [ISS ISP H3A AE/AWB Engine: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS ISP H3A AE/AWB Accumulators: \[4\]](#)
- [ISS ISP H3A DMA Interface: \[5\] \[6\] \[7\]](#)
- [ISS ISP H3A AEW Engine: \[8\] \[9\]](#)
- [ISS ISP H3A Register Accessibility During Frame Processing: \[10\]](#)
- [ISS H3A Register Summary: \[11\]](#)

**Table 8-1218. H3A\_LINE\_START**

<b>Address Offset</b>	0x0000 0064	
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>	<b>Instance</b> ISS_H3A
<b>Description</b>	Line Framing Logic Register In certain cases the number of clock cycles between HD pulses will be greater than the line buffer included in the H3A module. The framing module prior to the line buffer enables to control the data which is input to the line buffer.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLV																LINE_START															

Bits	Field Name	Description	Type	Reset
31:16	SLV	Start Line Vertical Specifies how many lines after the VD rising edge the real frame starts.	W	0x0000
15:0	LINE_START	Line Start The framing module uses the LINE_START bit field to find the position of the first pixel to place into the line buffer. Range: 0-65535	RW	0x0000

**Table 8-1219. Register Call Summary for Register H3A\_LINE\_START**

ISS ISP

- [ISS ISP H3A Line Framing Logic: \[0\] \[1\] \[2\] \[3\]](#)
- [ISS ISP H3A Poxel Extraction: \[4\]](#)
- [ISS ISP H3A AF Engine: \[5\] \[6\]](#)
- [ISS ISP H3A AEW Engine: \[7\] \[8\]](#)
- [ISS H3A Register Summary: \[9\]](#)

**Table 8-1220. H3A\_VFV\_CFG1**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	Vertical focus value configuration 1.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCOEF1_3								VCOEF1_2								VCOEF1_1								VCOEF1_0							

Bits	Field Name	Description	Type	Reset
31:24	VCOEF1_3	Vertical FV FIR 1 coefficient 3	RW	0x00
23:16	VCOEF1_2	Vertical FV FIR 1 coefficient 2	RW	0x00
15:8	VCOEF1_1	Vertical FV FIR 1 coefficient 1	RW	0x00
7:0	VCOEF1_0	Vertical FV FIR 1 coefficient 0	RW	0x00

**Table 8-1221. Register Call Summary for Register H3A\_VFV\_CFG1**

ISS ISP

- [ISS ISP H3A VFV Calculator: \[0\]](#)
- [ISS ISP H3A AF Engine: \[1\]](#)
- [ISS H3A Register Summary: \[2\]](#)

**Table 8-1222. H3A\_VFV\_CFG2**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	Vertical focus value configuration 2.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTNR1								RESERVED								VCOEF1_4															

Bits	Field Name	Description	Type	Reset
31:16	VTHR1	Threshold for vertical FV FIR 1	RW	0x0000
15:8	RESERVED		R	0x00
7:0	VCOEF1_4	Vertical FV FIR 1 coefficient 4	RW	0x00

**Table 8-1223. Register Call Summary for Register H3A\_VFV\_CFG2**

ISS ISP

- [ISS ISP H3A VFV Calculator: \[0\] \[1\]](#)
- [ISS ISP H3A VFV Accumulator: \[2\]](#)
- [ISS ISP H3A AF Engine: \[3\]](#)
- [ISS H3A Register Summary: \[4\]](#)

**Table 8-1224. H3A\_VFV\_CFG3**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	Vertical focus value configuration 4.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VCOEF2_3								VCOEF2_2								VCOEF2_1								VCOEF2_0							

Bits	Field Name	Description	Type	Reset
31:24	VCOEF2_3	Vertical FV FIR 2 coefficient 3	RW	0x00
23:16	VCOEF2_2	Vertical FV FIR 2 coefficient 2	RW	0x00
15:8	VCOEF2_1	Vertical FV FIR 2 coefficient 1	RW	0x00
7:0	VCOEF2_0	Vertical FV FIR 2 coefficient 0	RW	0x00

**Table 8-1225. Register Call Summary for Register H3A\_VFV\_CFG3**

ISS ISP

- [ISS ISP H3A VFV Calculator: \[0\]](#)
- [ISS ISP H3A AF Engine: \[1\]](#)
- [ISS H3A Register Summary: \[2\]](#)

**Table 8-1226. H3A\_VFV\_CFG4**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	Vertical focus value configuration 4.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTHR2								RESERVED								VCOEF2_4															

Bits	Field Name	Description	Type	Reset
31:16	VTHR2	Threshold for vertical FV FIR 2	RW	0x0000
15:8	RESERVED		R	0x00
7:0	VCOEF2_4	Vertical FV FIR 2 coefficient 4	RW	0x00

**Table 8-1227. Register Call Summary for Register H3A\_VFV\_CFG4**

ISS ISP

- [ISS ISP H3A VFV Calculator: \[0\] \[1\]](#)
- [ISS ISP H3A VFV Accumulator: \[2\]](#)
- [ISS ISP H3A AF Engine: \[3\]](#)
- [ISS H3A Register Summary: \[4\]](#)

**Table 8-1228. H3A\_HVF\_THR**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	Horizontal Focus Value Threshold		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTHR2																HTHR1															

Bits	Field Name	Description	Type	Reset
31:16	HTHR2	Threshold for horizontal FV IIR 2	RW	0x0000
15:0	HTHR1	Threshold for horizontal FV IIR 1	RW	0x0000

**Table 8-1229. Register Call Summary for Register H3A\_HVF\_THR**

ISS ISP

- [ISS ISP H3A HFV Accumulator: \[0\] \[1\]](#)
- [ISS ISP H3A AF Engine: \[2\] \[3\]](#)
- [ISS H3A Register Summary: \[4\]](#)

**Table 8-1230. H3A\_ADVANCED**

<b>Address Offset</b>	0x0000 007C	<b>Instance</b>	ISS_H3A
<b>Physical Address</b>	See <a href="#">Table 8-1167</a>		
<b>Description</b>	Normal and Advanced AF stats collection mode		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																RESERVED											AF_MODE				

Bits	Field Name	Description	Type	Reset
31:15	ID	0x0: This bit field must be set to 0xCA00 to enable AF advanced mode.	RW	0x00000
14:1	RESERVED		R	0x0000
0	AF_MODE	AF engine mode. 0x0: Normal Mode 0x1: Advanced mode. <a href="#">H3A_ADVANCED.ID</a> must be set to 0xCA00 to enable this functionality.	RW	0



---

**Table 8-1231. Register Call Summary for Register H3A\_ADVANCED**

---

ISS ISP

- [ISS ISP H3A Poxel Extraction: \[0\] \[1\]](#)
  - [ISS H3A Register Summary: \[2\]](#)
  - [ISS H3A Register Description: \[3\]](#)
- 

PRELIMINARY

PRELIMINARY

## 8.4 ISS Still Image Coprocessor

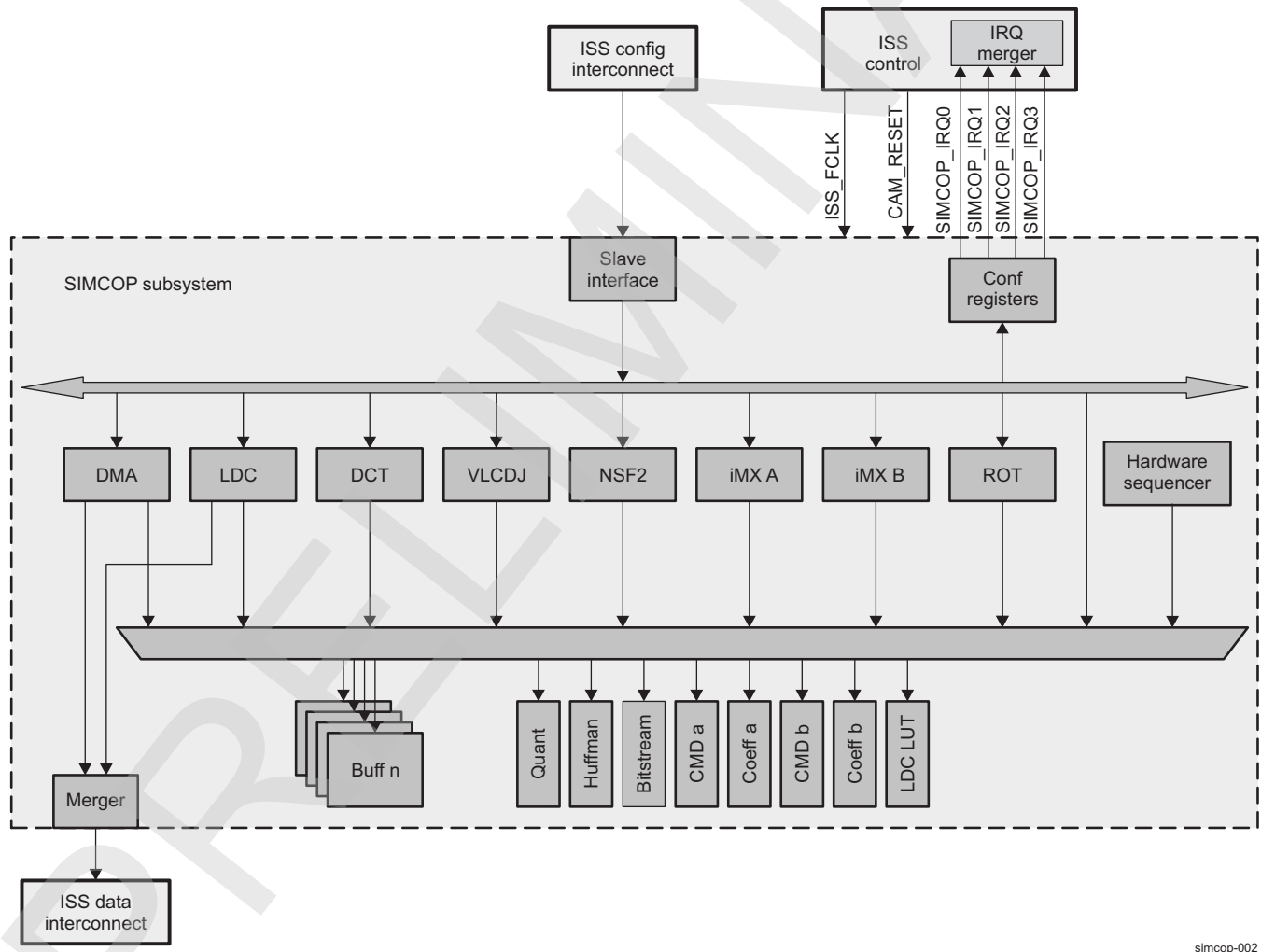
This section gives a high-level description of the ISS still iMage coprocessor (SIMCOP) in the device. For a high-level description of ISS level, see [Section 8.1, ISS: Overview](#).

### 8.4.1 ISS SIMCOP Overview

The SIMCOP subsystem is designed to encode, decode, and process image data. The SIMCOP is a macroblock-based memory-to-memory processing engine. It fetches macroblocks from system memory and stores them into local memories. Different accelerators take the fetched data, perform processing, and send the processing outcome back to local memories. From there the data could be further processed by other accelerators or be sent back to system memory. The SIMCOP needs an external central processing unit (CPU) to perform high-level control tasks and configurations. The SIMCOP is closely coupled to a CPU for that purpose.

Figure 8-238 shows a block diagram of the SIMCOP subsystem.

Figure 8-238. SIMCOP Subsystem Overview



simcop-002

The ISS SIMCOP subsystem consists of:

- Noise filter (NSF2) for advanced noise filtering and an edge-enhancement
- Two image and video processing engines (iMXs) supplement various fixed-function processing blocks
- Variable-length coder/decoder for JPEG (VLCDJ) module handles quantization and variable length coding, decoding and inverse quantization
- Discrete cosine transform (DCT) module implements method in the compression flow to transform

blocks of image

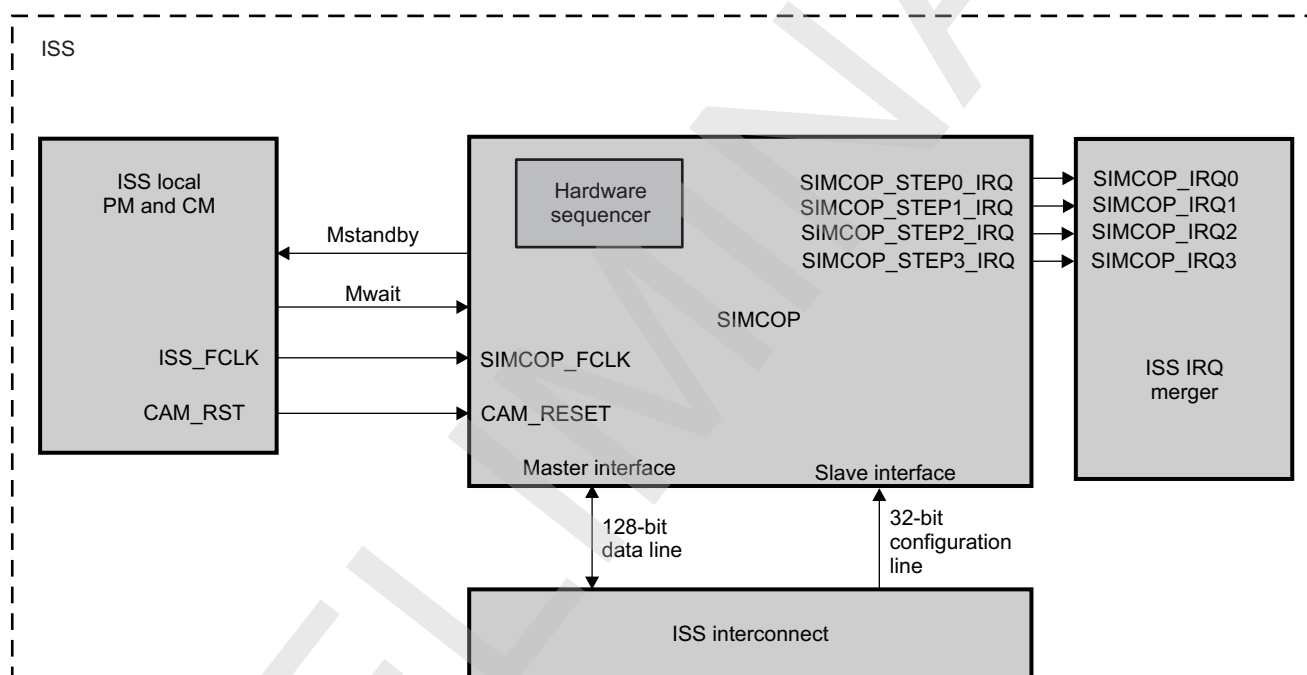
- Lens distortion correction (LDC) module deals with lens geometric distortion issues in the camera subsystem
- Rotation accelerator (ROT) engine blocks data rotation and shifting from a SIMCOP local memory to another SIMCOP local memory
- Hardware sequencer and buffers control SIMCOP modules/buffers to offload an external processor to perform low-level sequencing tasks
- Direct memory access (DMA) controller data transfers from SIMCOP memories to system memory or from system memory to SIMCOP memories

#### 8.4.1.1 ISS SIMCOP Integration

The SIMCOP subsystem is connected to the rest of the system through the ISS local interconnect.

Figure 8-239 shows the integration of the SIMCOP subsystem in the ISS.

**Figure 8-239. SIMCOP Integration**



simcop-001

**NOTE:** For more information about the IDLE hardware handshake and the wake-up request, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

This section gives an overview of typical uses of the module. See [Chapter 3, Power, Reset, and Clock Management](#), for more information and settings of the PRCM relationship to ISS clocks and resets.

Table 8-1232 lists the integration attributes.

**Table 8-1232. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
SIMCOP	PD_CAM	

Table 8-1233 lists the clocks and resets values.

**Table 8-1233. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SIMCOP	SIMCOP_FCLK	ISS_FCLK	PRCM	Functional clock provided by ISS_CLK from the PRCM module. It is used by all ISS submodules and ISS top-level resources.
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SIMCOP	CAM_RESET	CAM_RST	PRCM	ISS and SIMCOP global reset

Table 8-1234 lists the hardware resets.

**Table 8-1234. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
SIMCOP	SIMCOP_STEP_0_IRQ	SIMCOP_IRQ0	SIMCOP IRQ merger	Event triggered when a SIMCOP context is activated by the hardware sequencer
SIMCOP	SIMCOP_STEP_1_IRQ	SIMCOP_IRQ1	SIMCOP IRQ merger	Event triggered when a SIMCOP context is activated by the hardware sequencer
SIMCOP	SIMCOP_STEP_2_IRQ	SIMCOP_IRQ2	SIMCOP IRQ merger	Event triggered when a SIMCOP context is activated by the hardware sequencer
SIMCOP	SIMCOP_STEP_3_IRQ	SIMCOP_IRQ3	SIMCOP IRQ merger	Event triggered when a SIMCOP context is activated by the hardware sequencer

For more information about interrupt requests, see [Section 8.4.1.2.3, Interrupt Merger](#).

## 8.4.1.2 ISS SIMCOP Functional Description

### 8.4.1.2.1 ISS SIMCOP Local Power and Clock Management

#### 8.4.1.2.1.1 ISS SIMCOP Local Clock Management

ISS has five asynchronous clock domains. The SIMCOP belongs to the ISS\_FCLK clock domain.

An externally provided clock-enable signal is used. Multiple subdomains exist internally to reduce dynamic power consumption. This feature does not require intervention by the PRCM module. For high-level description of the SIMCOP clock domain and configuration inside ISS, see [Section 8.1.2.2, ISS Clocks](#).

Moreover, at high level the functional clock of ISS SIMCOP submodules can be cut by software to reduce power consumption by cutting off or turning on the modules from the [SIMCOP\\_CLKCTRL](#) register (see [Table 8-1235](#)).

When software wants to enable a submodule:

- Software sets the adequate bit in the [SIMCOP\\_CLKCTRL](#) register.
- Hardware enables the submodule functional and interface clocks.
- Hardware connects the configuration port of the submodule to the coprocessor bus.
- Hardware sets the appropriate bit in the [SIMCOP\\_CLKCTRL](#) register. Software must check if this bit is set before accessing the submodule.

[Table 8-1235](#) lists the ISS SIMCOP clock control register settings.

**Table 8-1235. ISS SIMCOP Clock Control Register Settings**

Module Name	Bit Field Name	Description
ROT	<a href="#">SIMCOP_CLKCTRL</a> [7] ROT_A	Writing 0x1 enables the module
iMX B	<a href="#">SIMCOP_CLKCTRL</a> [6] IMX_B	Writing 0x1 enables the module
iMX A	<a href="#">SIMCOP_CLKCTRL</a> [5] IMX_A	Writing 0x1 enables the module
NSF2	<a href="#">SIMCOP_CLKCTRL</a> [4] NSF2	Writing 0x1 enables the module
VLCDJ	<a href="#">SIMCOP_CLKCTRL</a> [3] VLCDJ	Writing 0x1 enables the module
DCT	<a href="#">SIMCOP_CLKCTRL</a> [2] DCT	Writing 0x1 enables the module
LDC	<a href="#">SIMCOP_CLKCTRL</a> [1] LDC	Writing 0x1 enables the module
DMA	<a href="#">SIMCOP_CLKCTRL</a> [0] DMA	Writing 0x1 enables the module

When software wants to shutdown a submodule:

- Software ensures that the submodule is idle. Mainly:
  - The submodule must not generate new events.
  - The submodule must not have any pending events.
  - For initiators: the submodule must stop interconnect transaction generation.
- Software clears the appropriate bit in the [SIMCOP\\_CLKCTRL](#) register.
- Hardware disconnects the configuration port of the module from the coprocessor bus: no more accesses are routed to the submodule. Ongoing transactions are finished (hardware stays in this state as long as required).
- For initiators: hardware waits until there are no more outstanding transactions:
  - SIMCOP DMA provides an MStandBy signal for that purpose.
  - The LDC does not provide this information. Software must ensure that the LDC is idle. SIMCOP hardware ensures that the internal MTC2OCP bridges have no outstanding transactions.
- Hardware cuts the submodules clock.
- Hardware clears the appropriate [SIMCOP\\_CLKCTRL](#) registers.

SIMCOP submodules support autogating. They can gate their functional and interface clock internally based on functional requirements. This feature could be disabled for debug or power consumption measurement purposes.

#### 8.4.1.2.1.2 Local Clock Autogating

The SIMCOP subsystem modules support autogating. They can gate their functional and interface clock internally based on functional requirements. This feature can be disabled. [Table 8-1236](#) summarizes which modules support autogating control and which control bit is used.

**Table 8-1236. Autoclock Gating Bit Control**

SIMCOP Resource	Autogating Override
SIMCOP top-level resources	N/A
SIMCOP DMA	N/A
LDC	N/A
DCT	DCT_CFG[5] AUTOGATING
VLCDJ	VLCDJ_CTRL[3] AUTOGATING
NSF2	N/A
iMX A	iMXa.IMX_CLKCNTRL[0] CLKCNTL
iMX B	iMXb.IMX_CLKCNTRL[0] CLKCNTL
ROT	ROT_CFG[9] AUTOGATING

#### 8.4.1.2.1.3 ISS SIMCOP Power Management

The SIMCOP supports the STANDBY power-management protocol. It is used by the SIMCOP to indicate when it has no more interconnect transactions to perform. This information is used at ISS level to control the SIMCOP and ISS shut-down sequence. The SIMCOP does not have any internal wake-up events. DISCONNECT and IDLE protocols are not supported.

The typical SIMCOP shut-down sequence is:

1. Software disables all SIMCOP submodules. This ensures the SIMCOP does not:
  - (a) Generate any traffic on its master port: SIMCOP asserts the MStandby signal.
  - (b) Generate new events (IRQs).
2. Software clears all pending events at SIMCOP level: All interrupt lines become inactive.
3. Software initiates the SIMCOP shut-down sequence by writing the appropriate bit at ISS level.
4. ISS hardware:
  - (a) Disconnects the SIMCOP configuration port. Any access attempts to SIMCOP are blocked at ISS level. The DISCONNECT protocol is handled at ISS level: SIMCOP does not provide any handshake signals to support it.
  - (b) Waits until MStandby from SIMCOP is asserted (always true in a normal case because SIMCOP submodules are inactive)
  - (c) Asserts the SWait input of SIMCOP
  - (d) Cuts the SIMCOP clock

The typical SIMCOP enable sequence is:

1. Software initiates the SIMCOP enable sequence by writing the appropriate bit at ISS level.
2. ISS hardware:
  - (a) Enables the SIMCOP clock
  - (b) Deasserts the SWait input of the SIMCOP
  - (c) Connects the SIMCOP configuration port

Three modes for MStandBy control are supported. The mode is selected using the [SIMCOP\\_HL\\_SYSCONFIG\[5:4\] STANDBYMODE](#) bit field.

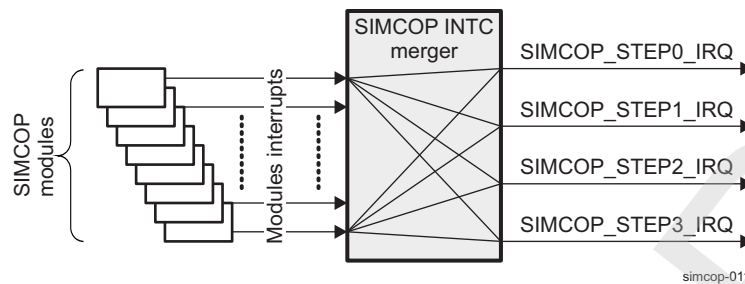
- Smart-standby: This is the normal mode to be used. When in this mode, the SIMCOP asserts the MStandBy signal when the MStandBy output of SIMCOP DMA is asserted, the LDC is effectively disabled ([SIMCOP\\_CLKCTRL\[1\] LDC =0](#)), and the MTCR2OCP bridge and the merger have no more outstanding transactions.
- Force-standby: This is a backup mode intended to be used only if smart-idle mode is bugged. When in this mode, the SIMCOP asserts MStandBy unconditionally. Software must ensure that the SIMCOP is in a correct quiet state before programming this mode.
- No-standby: This is a backup mode intended to be used only if smart-idle mode is bugged. When in this mode, the SIMCOP never asserts the MStandBy signal.

#### 8.4.1.2.2 Software Reset

Software reset is intended to return the SIMCOP into a known state without requiring a complete device reset. A software reset at SIMCOP level is seen as a hardware reset for SIMCOP submodules. To perform a software reset, sets the [SIMCOP\\_HL\\_SYSCONFIG\[0\] SOFTRESET](#) bit to 1. The [SIMCOP\\_HL\\_SYSCONFIG\[0\] SOFTRESET](#) bit indicates that the software reset is ongoing when its value is 1. When the software reset completes, the [SIMCOP\\_HL\\_SYSCONFIG\[1\] SOFTRESET](#) bit is automatically reset. Software must ensure that the software reset completes before configuring the hardware sequencer or buffers.

#### 8.4.1.2.3 Interrupt Merger

SIMCOP interrupts are merged at the SIMCOP subsystem level into four interrupts, as shown in [Figure 8-240](#).

**Figure 8-240. SIMCOP Interrupt Merger Overview**

These four outputs interrupts can be configured through [SIMCOP\\_HL\\_IRQENABLE\\_SET\\_i](#) and [SIMCOP\\_HL\\_IRQENABLE\\_CLR\\_i](#) ( $i = 0$  to 3, corresponding to output interrupt) to enable or disable interrupt mask for each interrupt output. Software can poll the value of interrupt before masking ([SIMCOP\\_HL\\_IRQSTATUS\\_RAW\\_i](#)) and after masking ([SIMCOP\\_HL\\_IRQSTATUS\\_i](#)).

Software can configure how outputs interrupts are generated through [SIMCOP\\_CTRL\[i\] IRQi\\_MODE](#) ( $i = 0$  to 3):

- 0x0: OR, the interrupt is asserted when one of the selected events has occurred
- 0x1: AND, the interrupt is asserted when all selected events have occurred

[Table 8-1237](#) shows how interrupts are mapped on the [SIMCOP\\_HL\\_IRQSTATUS\\_RAW\\_i](#), [SIMCOP\\_HL\\_IRQSTATUS\\_i](#), [SIMCOP\\_HL\\_IRQENABLE\\_SET\\_i](#), and [SIMCOP\\_HL\\_IRQENABLE\\_CLR\\_i](#) registers.

**Table 8-1237. SIMCOP Interrupts**

Bit	Name	Description
0	SIMCOP_DMA_IRQ0	Interrupt triggered by SIMCOP DMA
1	LDC_FRAME_IRQ	A full frame has been processed by LDC2
2	DCT_IRQ	DCT operation has been completed (configured number of MCUs for YUV4:2:0/4:2:2 mode, or number of blocks for sequential block mode)
3	VLCDJ_BLOCK_IRQ	A macroblock has been processed (encode/decode)
4	NSF_IRQ	Event triggered by the NSF2 imaging accelerator when processing of a block is done.
5	IMX_A_IRQ	Event triggered when iMX has executed a SLEEP instruction.
6	IMX_B_IRQ	
7	ROT_A_IRQ	Event triggered by the ROT engine.
8	RESERVED	Reserved
9	LDC_BLOCK_IRQ	A macroblock has been processed by LDC2
10	SIMCOP_STEP0_IRQ	Event triggered when a SIMCOP context is activated by the hardware sequencer
11	SIMCOP_STEP1_IRQ	
12	SIMCOP_STEP2_IRQ	
13	SIMCOP_STEP3_IRQ	
14	DONE_IRQ	Event triggered when the hardware sequencer finishes the sequence: - the sequence step counter has reached the limit - All accelerator and DMA events for the last sequence step have been received.
15	VLCDJ_DECODE_ERR_IRQ	A decode error has been signaled by the VLCDJ
16	ICNT_ERR_IRQ	An error has been received on the SIMCOP master port on the L3 interconnect.
17	LDC2BRIDGE_ERR_IRQ	The LDC2 bridge has generated an error. This event must not trigger in a normal use case.
18	SIMCOP_DMA_IRQ1	Interrupt triggered by SIMCOP DMA
19	CPU_PROC_START_IRQ	This interrupt is used when CPU data processing is used in a macroblock processing pipeline. When the CPU receives this IRQ, data is ready to be processed. When CPU is done with processing it acknowledges by setting the SIMCOP_HWSEQ_CTRL.CPU_PROC_DONE bit. The interrupt is cleared as usual.



#### 8.4.1.2.4 ISS SIMCOP Modules Description

This section gives links to each subsection for each submodule in the ISS SIMCOP module.

The ISS module provides unique ISS\_FCLK for all the submodules inside the SIMCOP (see [Section 8.4.1.2.1.1, ISS SIMCOP Local Clock Management](#)).

For information about hardware and software resets for each module see [Table 8-1237](#).

For interrupt request information for each module, see [Section 8.4.1.2.3, Interrupt Merger](#).

ISS SIMCOP consists of the following submodules:

- Hardware sequencer and buffers, for more information see [Section 8.4.2, ISS SIMCOP Hardware Sequencer and Buffers](#).
- DMA, for more information, see [Section 8.4.3, ISS SIMCOP DMA Module](#).
- ROT, for more information, see [Section 8.4.7, ISS SIMCOP Rotation Accelerator \(ROT\) Module](#).
- iMX
- NSF2
- VLCDJ, for more information, see [Section 8.4.6, ISS SIMCOP Variable Length Coder/Decoder for JPEG \(VLCDJ\) Module](#).
- DCT, for more information, see [Section 8.4.5, ISS SIMCOP Discrete Cosine Transform \(DCT\) Module](#).
- LDC, for more information, see [Section 8.4.4, ISS SIMCOP LDC Module](#).

#### 8.4.1.3 ISS SIMCOP Programming Models

##### 8.4.1.3.1 Global Initialization

###### 8.4.1.3.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the ISS SIMCOP module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the ISS SIMCOP.

**Table 8-1238. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
ISS local PM and CM	Module interface and functional clocks must be enabled. For more information, see <a href="#">Section 8.4.1.2.1, ISS SIMCOP Local Power and Clock Management</a> .

###### 8.4.1.3.1.2 ISS SIMCOP Module Global Initialization

This procedure initializes the ISS SIMCOP module after a poweron or software reset.

**Table 8-1239. ISS SIMCOP Global Initialization**

Step	Register/ Bit Field / Programming Model	Value
Execute software reset.	SIMCOP_HL_SYSCONFIG[0] SOFTRESET	0x1
Wait until reset completed?	SIMCOP_HL_SYSCONFIG[0] SOFTRESET	0x0
Configure STANDBYMODE.	SIMCOP_HL_SYSCONFIG[5:4] STANDBYMODE	0x2
Configure submodules interface and functional clock.	SIMCOP_CLKCTRL[7:0]	xxx
<b>Interrupt configuration</b>		
Set interrupt enable bit for the submodules.	SIMCOP_HL_IRQENABLE_SET_i[7:0]	xxx

### 8.4.1.3.2 ISS SIMCOP Operational Modes Configuration

#### 8.4.1.3.2.1 Interrupts

To unmask an interrupt for generation of SIMCOP\_IRQ<sub>i</sub> output interrupts, software must set the corresponding bit of `SIMCOP_HL_IRQENABLE_SET_i` ( $i = 0$  to 3) to 1.

Example 1, SIMCOP\_IRQ1 is generated when DCT\_IRQ or VLCDJ\_BLOCK\_IRQ are generated:

- `SIMCOP_HL_IRQENABLE_SET_i[2]` DCT\_IRQ = 0x1 ( $i = 1$ )
- `SIMCOP_HL_IRQENABLE_SET_i[3]` VLCDJ\_BLOCK\_IRQ = 0x1 ( $i = 1$ )
- `SIMCOP_CTRL[1]` IRQ1\_MODE = 0x0

Example 2, SIMCOP\_IRQ3 is generated when IMX\_A\_IRQ and LDC\_BLOCK\_IRQ are generated:

- `SIMCOP_HL_IRQENABLE_SET_i[5]` IMX\_A\_IRQ = 0x1 ( $i = 3$ )
- `SIMCOP_HL_IRQENABLE_SET_i[9]` LDC\_BLOCK\_IRQ = 0x1 ( $i = 3$ )
- `SIMCOP_CTRL[3]` IRQ3\_MODE = 0x1

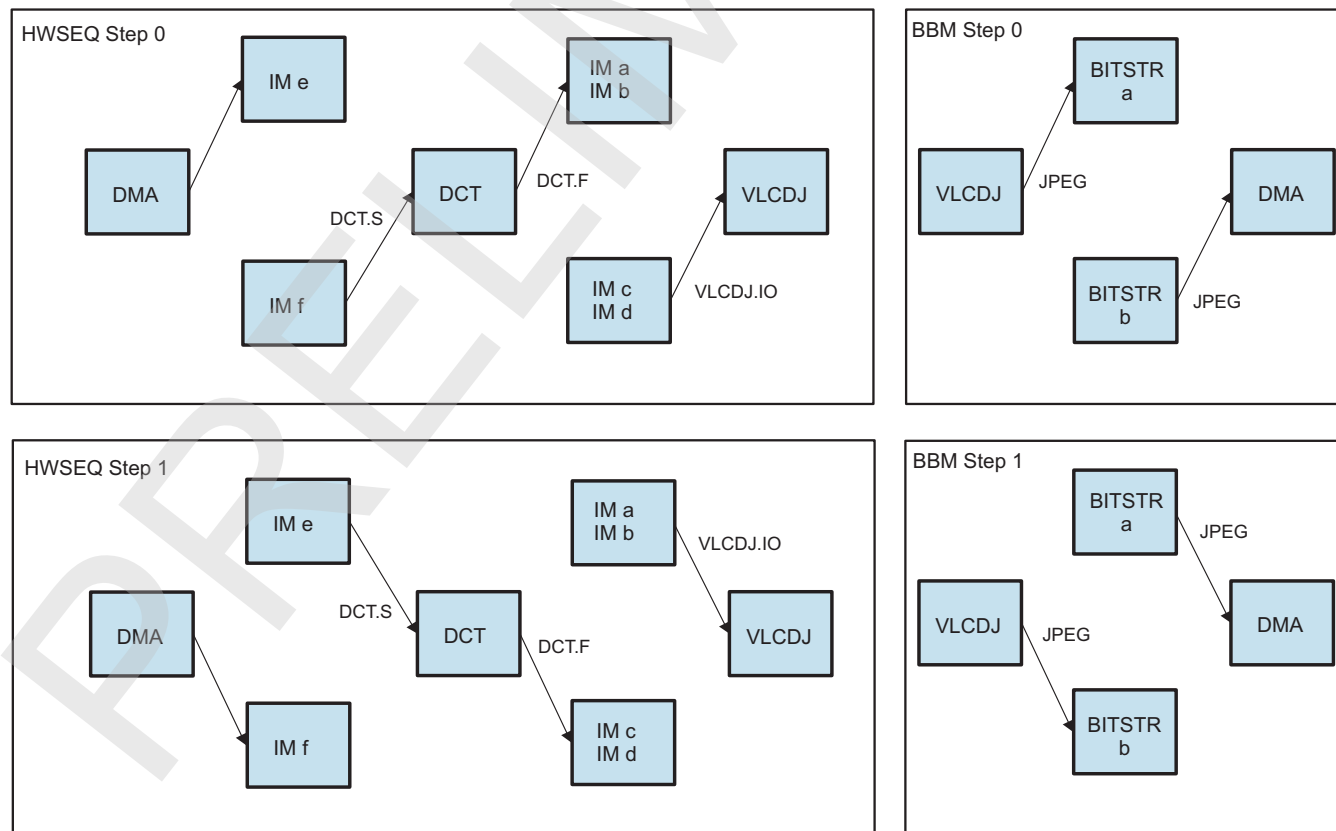
To mask an interrupt for generation of SIMCOP\_IRQ<sub>i</sub> output interrupts, software must clear the corresponding bit of `SIMCOP_HL_IRQENABLE_CLEAR_i` ( $i = 0$  to 3) to 0.

When an event occurs, the corresponding bit in the `SIMCOP_HL_IRQSTATUS_RAW_i` ( $i = 0$  to 3) register is set regardless if the event has been enabled or not. Bits in the `SIMCOP_HL_IRQSTATUS_i` ( $i = 0$  to 3) registers are only set when an enabled event occurs. Software can clear a pending event by setting the adequate bit in the `SIMCOP_HL_IRQSTATUS_i` ( $i = 0$  to 3) register.

#### 8.4.1.3.2.2 JPEG Encode Operational Mode Configuration

Figure 8-241 shows a typical JPEG encode macroblock pipeline.

Figure 8-241. JPEG Encode Pipeline



simcop-012

**Table 8-1240. JPEG Encode Pipeline**

Step	Register/ Bit Field / Programming Model	Value
Set SIMCOP DMA to fetch data from SDRAM and stored into an image buffer (IM e or IM f).	See DMA programming model <a href="#">Section 8.4.3.4</a>	xxx
Set DCT to reads data from an image buffer (IM e or IM f).	DCT_SPTR[12:5] ADDR	xxx
Set DCT to store data into sets of two image buffers (IM a + IM b or IM c + IM d).	DCT_FPTR[12:4] ADDR	xxx
Set VLCDJ to read data from image buffers.	See VLCDJ programming model <a href="#">Section 8.4.6.4</a>	xxx
Set VLCDJ to write the data to the bitstream buffer.	See VLCDJ programming model <a href="#">Section 8.4.6.4</a>	xxx
Set the SIMCOP DMA to send the data to SDRAM.	See DMA programming model <a href="#">Section 8.4.3.4</a>	xxx

### 8.4.1.4 ISS SIMCOP Register Manual

#### 8.4.1.4.1 SIMCOP Instance Summary

Table 8-1241 is the SIMCOP top level instance summary.

**Table 8-1241. SIMCOP Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-M3 Private Access	Size
SIMCOP_CONTROL	0x5202 0000	0x5506 0000	256 bytes

**NOTE:** Private Access is an access which is not using the L3/L4 interconnects.

#### 8.4.1.4.2 SIMCOP Registers

##### 8.4.1.4.2.1 SIMCOP Register Summary

Table 8-1242 is the SIMCOP\_CONTROL register mapping summary.

**Table 8-1242. SIMCOP\_CONTROL Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-M3 Private Access
SIMCOP_HL_REVISION	R	32	0x0000 0000	0x5202 0000	0x5506 0000
SIMCOP_HL_HWINFO	R	32	0x0000 0004	0x5202 0004	0x5506 0004
SIMCOP_HL_SYSCONFIG	RW	32	0x0000 0010	0x5202 0010	0x5506 0010
RESERVED	RW	32	0x0000 001C	0x5202 001C	0x5506 001C
SIMCOP_HL_IRQSTATUS_RAW_i <sup>(1)</sup>	RW	32	0x0000 0020 + (0x10 * i)	0x5202 0020 + (0x10 * i)	0x5506 0020 + (0x10 * i)
SIMCOP_HL_IRQSTATUS_i <sup>(1)</sup>	RW	32	0x0000 0024 + (0x10 * i)	0x5202 0024 + (0x10 * i)	0x5506 0024 + (0x10 * i)
SIMCOP_HL_IRQENABLE_SET_i <sup>(1)</sup>	RW	32	0x0000 0028 + (0x10 * i)	0x5202 0028 + (0x10 * i)	0x5506 0028 + (0x10 * i)
SIMCOP_HL_IRQENABLE_CLR_i <sup>(1)</sup>	RW	32	0x0000 002C + (0x10 * i)	0x5202 002C + (0x10 * i)	0x5506 002C + (0x10 * i)
SIMCOP_CTRL	RW	32	0x0000 0060	0x5202 0060	0x5506 0060
SIMCOP_CLKCTRL	RW	32	0x0000 0064	0x5202 0064	0x5506 0064

<sup>(1)</sup> i = 0 to 3

##### 8.4.1.4.2.2 SIMCOP\_CONTROL Register Descriptions

**Table 8-1243. SIMCOP\_HL\_REVISION**

Address Offset	0x0000 0000	Instance	SIMCOP_CONTROL
Physical Address	See Table 8-1242.		
Description	MODULE REVISION This register contains the IP revision code in binary coded digital. For example, we have: 0x01 = revision 0.1 and 0x21 = revision 2.1		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
																	REV																			

Bits	Field Name	Description	Type	Reset
31:0	REV	Revision ID	R	_(1)

(1) TI internal data

**Table 8-1244. Register Call Summary for Register SIMCOP\_HL\_REVISION**

ISS SIMCOP Registers Manual

- [SIMCOP Register Summary: \[0\]](#)

**Table 8-1245. SIMCOP\_HL\_HWINFO**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	SIMCOP_CONTROL
<b>Physical Address</b>	See <a href="#">Table 8-1242</a> .		
<b>Description</b>	Information about the IP module's hardware configuration. It provides information about the generic parameters.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LDCR_RESP_FIFO		IMAGE_BUFFERS		RESERVED	ROT_A_ENABLE	IMX_B_ENABLE	IMX_A_ENABLE	NSF_ENABLE	VLCDJ_ENABLE	DCT_ENABLE	LDC_ENABLE				

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Read returns 0.	R	0x00000
12:10	LDCR_RESP_FIFO	Defines the size of the LDC read master response FIFO in words of 128-bits. Read 0x2: 8x128 bits Read 0x3: 16x128 bits Read 0x4: 32x128 bits Read 0x5: 64x128 bits Read 0x6: 128x128 bits Read 0x7: 256x256 bits	R	0x4
9:8	IMAGE_BUFFERS	This parameter defines the image buffer count. Read 0x0: 4 Image buffers (e, f, g, h) Read 0x1: 8 Image buffers	R	0x1
7	RESERVED	Read returns 0.	R	0
6	ROT_A_ENABLE	The ROT a is present when this parameter is set. Read 0x0: Disabled at design time Read 0x1: Enabled at design time	R	1
5	IMX_B_ENABLE	The iMX B module and the CMD b, COEFF b memories are present when this parameter is set. Read 0x0: Disabled at design time Read 0x1: Enabled at design time	R	1
4	IMX_A_ENABLE	The iMX A module and the CMD a, COEFF a memories are present when this parameter is set. Read 0x0: Disabled at design time Read 0x1: Enabled at design time	R	1
3	NSF_ENABLE	The NSF2 is present when this parameter is set. Read 0x0: Disabled at design time Read 0x1: Enabled at design time	R	1

Bits	Field Name	Description	Type	Reset
2	VLCDJ_ENABLE	The VLCD module and the QUANT, HUFFMAN, BITSTREAM memories are present when this parameter is set. Read 0x0: Disabled at design time Read 0x1: Enabled at design time	R	1
1	DCT_ENABLE	The DCT module is present when this parameter is set. Read 0x0: Disabled at design time Read 0x1: Enabled at design time	R	1
0	LDC_ENABLE	The LDC module and the LDC LUT are present when this parameter is set. Read 0x0: Disabled at design time Read 0x1: Enabled at design time	R	1

**Table 8-1246. Register Call Summary for Register SIMCOP\_HL\_HWINFO**

ISS SIMCOP Registers Manual

- [SIMCOP Register Summary: \[0\]](#)

**Table 8-1247. SIMCOP\_HL\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	SIMCOP_CONTROL
<b>Physical Address</b>	See <a href="#">Table 8-1242</a> .		
<b>Description</b>	This register controls the various parameters of the OCP interface		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STANDBYMODE			RESERVED		SOFTRESET										

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Read returns 0.	R	0x00000000
5:4	STANDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. 0x0: Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only. 0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only. 0x2: Smart-standby mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator.	RW	0x0
3:1	RESERVED	Read returns 0.	R	0x0
0	SOFTRESET	Software reset Read 0x0: Reset done, no pending action Write 0x0: No action Write 0x1: Initiate software reset Read 0x1: Reset (software or other) ongoing	RW	0

**Table 8-1248. Register Call Summary for Register SIMCOP\_HL\_SYSCONFIG**

ISS SIMCOP Functional Description

- [ISS SIMCOP Power Management: \[0\]](#)
- [Software Reset: \[1\] \[2\] \[3\]](#)

ISS SIMCOP Programming Models

- [ISS SIMCOP Module Global Initialization: \[4\] \[5\] \[6\]](#)

ISS SIMCOP Registers Manual

- [SIMCOP Register Summary: \[7\]](#)

**Table 8-1249. SIMCOP\_HL\_IRQSTATUS\_RAW\_i**

<b>Address Offset</b>	0x0000 0020 + (0x10 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1242</a> .	<b>Instance</b>	SIMCOP_CONTROL
<b>Description</b>	Per-event raw interrupt status vector Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CPU_PROC_START_IRQ	SIMCOP_DMA_IRQ1	LDC2BRIDGE_ERR_IRQ	ICNT_ERR_IRQ	VLCDJ_DECODE_ERR_IRQ	DONE_IRQ	STEP3_IRQ	STEP2_IRQ	STEP1_IRQ	STEP0_IRQ	LDC_BLOCK_IRQ	RESERVED	ROT_A	IMX_B_IRQ	IMX_A_IRQ	NSF_IRQ_IRQ	VLCDJ_BLOC_IRQ	DCT_IRQ	LDC_FRAME_IRQ	SIMCOP_DMA_IRQ0				

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Read returns 0.	R	0x000
19	CPU_PROC_START_IRQ	Event triggered by the hardware sequencer to instruct the CPU to process a macroblock  Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
18	SIMCOP_DMA_IRQ1	Event triggered by the SIMCOP DMA. This event is automatically cleared at SIMCOP level when it is cleared at SIMCOP DMA level. Check SIMCOP DMA IRQ registers.  Read 0x0: No event pending Read 0x1: Event pending	R	0
17	LDC2BRIDGE_ERR_IRQ	The LDC2 bridge has generated an error.  Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
16	ICNT_ERR_IRQ	An error has been received on the SIMCOP master port.  Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
15	VLCDJ_DECODE_ERR_IRQ	A decode error has been signaled by the VLCDJ module Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
14	DONE_IRQ	Event triggered when the hardware sequencer finishes the sequence: - the sequence step counter has reached the limit - all accelerator and DMA events for the last sequence step have been received. Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
13	STEP3_IRQ	Event triggered when Step 3 is activated by the hardware sequencer Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
12	STEP2_IRQ	Event triggered when Step 2 is activated by the hardware sequencer Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
11	STEP1_IRQ	Event triggered when Step 1 is activated by the hardware sequencer Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
10	STEP0_IRQ	Event triggered when Step 0 is activated by the hardware sequencer Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
9	LDC_BLOCK_IRQ	This event is triggered by LDC when a macroblock has been processed Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
8	RESERVED	Read returns 0.	R	0
7	ROT_A	Event triggered by the ROT a engine Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0



Bits	Field Name	Description	Type	Reset
6	IMX_B_IRQ	Event triggered when iMX has executed a SLEEP instruction. Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
5	IMX_A_IRQ	Event triggered when iMX has executed a SLEEP instruction. Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
4	NSF_IRQ_IRQ	Event triggered by the NSF2 imaging accelerator when processing of a block is done. Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
3	VLCDJ_BLOC_IRQ	This event is triggered by VLCDJ when a macroblock has been processed (encode/decode) Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
2	DCT_IRQ	Event triggered when a block has been processed by the DCT module and the filter outcome has been stored to an image buffer. Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
1	LDC_FRAME_IRQ	This event is triggered by LDC when a full frame has been processed Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
0	SIMCOP_DMA_IRQ0	Event triggered by the SIMCOP DMA. This event is automatically cleared at SIMCOP level when it is cleared at SIMCOP DMA level. Check SIMCOP DMA IRQ registers. Read 0x0: No event pending Read 0x1: Event pending	R	0

**Table 8-1250. Register Call Summary for Register SIMCOP\_HL\_IRQSTATUS\_RAW\_i**

ISS SIMCOP Functional Description

- [Interrupt Merger: \[0\] \[1\]](#)

ISS SIMCOP Programming Models

- [Interrupts: \[2\]](#)

ISS SIMCOP Registers Manual

- [SIMCOP Register Summary: \[3\]](#)

**Table 8-1251. SIMCOP\_HL\_IRQSTATUS\_i**

<b>Address Offset</b>	0x0000 0024 + (0x10 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1242</a> .	<b>Instance</b>	SIMCOP_CONTROL
<b>Description</b>	Per-event "enabled" interrupt status vector. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CPU_PROC_START_IRQ	SIMCOP_DMA_IRQ1	LDC2BRIDGE_ERR_IRQ	ICNT_ERR_IRQ	VLCDJ_DECODE_ERR_IRQ	DONE_IRQ	STEP3_IRQ	STEP2_IRQ	STEP1_IRQ	STEP0_IRQ	LDC_BLOCK_IRQ	RESERVED	ROT_A	IMX_B_IRQ	IMX_A_IRQ	NSF_IRQ_IRQ	VLCDJ_BLOC_IRQ	DCT_IRQ	LDC_FRAME_IRQ	SIMCOP_DMA_IRQ0				

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Read returns 0.	R	0x000
19	CPU_PROC_START_IRQ	Event triggered by the hardware sequencer to instruct the CPU to process a macroblock Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
18	SIMCOP_DMA_IRQ1	Event triggered by the SIMCOP DMA. This event is automatically cleared at SIMCOP level when it is cleared at SIMCOP DMA level. Check SIMCOP DMA IRQ registers. Read 0x0: No (enabled) event pending Read 0x1: Event pending	R	0
17	LDC2BRIDGE_ERR_IRQ	The LDC2 bridge has generated an error. Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
16	ICNT_ERR_IRQ	An error has been received on the SIMCOP master port. Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
15	VLCDJ_DECODE_ERR_IRQ	A decode error has been signaled by the VLCDJ module Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
14	DONE_IRQ	Event triggered when the hardware sequencer finishes the sequence: -The sequence step counter has reached the limit. All accelerator and DMA events for the last sequence step have been received. Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
13	STEP3_IRQ	Event triggered when Step 3 is activated by the hardware sequencer Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
12	STEP2_IRQ	Event triggered when Step 2 is activated by the hardware sequencer Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
11	STEP1_IRQ	Event triggered when Step 1 is activated by the hardware sequencer Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
10	STEP0_IRQ	Event triggered when Step 0 is activated by the hardware sequencer Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
9	LDC_BLOCK_IRQ	This event is triggered by LDC when a macroblock has been processed Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
8	RESERVED	Read returns 0.	R	0
7	ROT_A	Event triggered by the ROT a engine Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
6	IMX_B_IRQ	Event triggered when iMX has executed a SLEEP instruction. Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
5	IMX_A_IRQ	Event triggered when iMX has executed a SLEEP instruction. Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
4	NSF_IRQ_IRQ	Event triggered by the NSF2 imaging accelerator when processing of a block is done. Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
3	VLCDJ_BLOC_IRQ	This event is triggered by VLCDJ when a macroblock has been processed (encode/decode) Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
2	DCT_IRQ	Event triggered when a block has been processed by the DCT module and the filter outcome has been stored to an image buffer. Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
1	LDC_FRAME_IRQ	This event is triggered by LDC when a full frame has been processed Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
0	SIMCOP_DMA_IRQ0	Event triggered by the SIMCOP DMA. This event is automatically cleared at SIMCOP level when it is cleared at SIMCOP DMA level. Check SIMCOP DMA IRQ registers. Read 0x0: No (enabled) event pending Read 0x1: Event pending	R	0

**Table 8-1252. Register Call Summary for Register SIMCOP\_HL\_IRQSTATUS\_i**

ISS SIMCOP Functional Description

- [Interrupt Merger: \[0\] \[1\]](#)

ISS SIMCOP Programming Models

- [Interrupts: \[2\] \[3\]](#)

ISS SIMCOP Registers Manual

- [SIMCOP Register Summary: \[4\]](#)

**Table 8-1253. SIMCOP\_HL\_IRQENABLE\_SET\_i**

<b>Address Offset</b>	0x0000 0028 + (0x10 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1242</a> .	<b>Instance</b>	SIMCOP_CONTROL
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CPU_PROC_START_IRQ	SIMCOP_DMA_IRQ1	LDC2BRIDGE_ERR_IRQ	ICNT_ERR_IRQ	VLCDJ_DECODE_ERR_IRQ	DONE_IRQ	STEP3_IRQ	STEP2_IRQ	STEP1_IRQ	STEP0_IRQ	LDC_BLOCK_IRQ	RESERVED	ROT_A	IMX_B_IRQ	IMX_A_IRQ	NSF_IRQ_IRQ	VLCDJ_BLOC_IRQ	DCT_IRQ	LDC_FRAME_IRQ	SIMCOP_DMA_IRQ0				

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Read returns 0.	R	0x000
19	CPU_PROC_START_IRQ	Event triggered by the hardware sequencer to instruct the CPU to process a macroblock Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
18	SIMCOP_DMA_IRQ1	Event triggered by the SIMCOP DMA. Check SIMCOP DMA IRQ registers. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
17	LDC2BRIDGE_ERR_IRQ	The LDC2 bridge has generated an error. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
16	ICNT_ERR_IRQ	An error has been received on the SIMCOP master port. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
15	VLCDJ_DECODE_ERR_IRQ	A decode error has been signaled by the VLCDJ module Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
14	DONE_IRQ	Event triggered when the hardware sequencer finishes the sequence: The sequence step counter has reached the limit. All accelerator and DMA events for the last sequence step have been received. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
13	STEP3_IRQ	Event triggered when Step 3 is activated by the hardware sequencer Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
12	STEP2_IRQ	Event triggered when Step 2 is activated by the hardware sequencer Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
11	STEP1_IRQ	Event triggered when Step 1 is activated by the hardware sequencer Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
10	STEP0_IRQ	Event triggered when Step 0 is activated by the hardware sequencer Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
9	LDC_BLOCK_IRQ	This event is triggered by LDC when a macroblock has been processed Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
8	RESERVED	Read returns 0.	R	0
7	ROT_A	Event triggered by the ROT a engine Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
6	IMX_B_IRQ	Event triggered when iMX has executed a SLEEP instruction. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
5	IMX_A_IRQ	Event triggered when iMX has executed a SLEEP instruction. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
4	NSF_IRQ_IRQ	Event triggered by the NSF2 imaging accelerator when processing of a block is done. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
3	VLCDJ_BLOC_IRQ	This event is triggered by VLCDJ when a macroblock has been processed (encode/decode) Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
2	DCT_IRQ	Event triggered when a block has been processed by the DCT module and the filter outcome has been stored to an image buffer. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
1	LDC_FRAME_IRQ	This event is triggered by LDC when a full frame has been processed Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
0	SIMCOP_DMA_IRQ0	Event triggered by the SIMCOP DMA. Check SIMCOP DMA IRQ registers. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0

**Table 8-1254. Register Call Summary for Register SIMCOP\_HL\_IRQENABLE\_SET\_i**

ISS SIMCOP Functional Description

- [Interrupt Merger: \[0\] \[1\]](#)

ISS SIMCOP Programming Models

- [ISS SIMCOP Module Global Initialization: \[2\]](#)
- [Interrupts: \[3\] \[4\] \[5\] \[6\] \[7\]](#)

ISS SIMCOP Registers Manual

- [SIMCOP Register Summary: \[8\]](#)

**Table 8-1255. SIMCOP\_HL\_IRQENABLE\_CLR\_i**

<b>Address Offset</b>	0x0000 002C + (0x10 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1242</a> .	<b>Instance</b>	SIMCOP_CONTROL
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CPU_PROC_START_IRQ	SIMCOP_DMA_IRQ1	LDC2BRIDGE_ERR_IRQ	ICNT_ERR_IRQ	VLCDJ_DECODE_ERR_IRQ	DONE_IRQ	STEP3_IRQ	STEP2_IRQ	STEP1_IRQ	STEP0_IRQ	LDC_BLOCK_IRQ	RESERVED	ROT_A	IMX_B_IRQ	IMX_A_IRQ	NSF_IRQ_IRQ	VLCDJ_BLOC_IRQ	DCT_IRQ	LDC_FRAME_IRQ	SIMCOP_DMA_IRQ0				

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Read returns 0.	R	0x000
19	CPU_PROC_START_IRQ	Event triggered by the hardware sequencer to instruct the CPU to process a macroblock Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
18	SIMCOP_DMA_IRQ1	Event triggered by the SIMCOP DMA. Check SIMCOP DMA IRQ registers. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
17	LDC2BRIDGE_ERR_IRQ	The LDC2 bridge has generated an error. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
16	ICNT_ERR_IRQ	An error has been received on the SIMCOP master port. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
15	VLCDJ_DECODE_ERR_IRQ	A decode error has been signaled by the VLCDJ module Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0



Bits	Field Name	Description	Type	Reset
14	DONE_IRQ	Event triggered when the hardware sequencer finishes the sequence: The sequence step counter has reached the limit. All accelerator and DMA events for the last sequence step have been received. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
13	STEP3_IRQ	Event triggered when Step 3 is activated by the hardware sequencer Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
12	STEP2_IRQ	Event triggered when Step 2 is activated by the hardware sequencer Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
11	STEP1_IRQ	Event triggered when Step 1 is activated by the hardware sequencer Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
10	STEP0_IRQ	Event triggered when Step 0 is activated by the hardware sequencer Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
9	LDC_BLOCK_IRQ	This event is triggered by LDC when a macroblock has been processed Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
8	RESERVED	Read returns 0.	R	0
7	ROT_A	Event triggered by the ROT a engine Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
6	IMX_B_IRQ	Event triggered when iMX has executed a SLEEP instruction. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
5	IMX_A_IRQ	Event triggered when iMX has executed a SLEEP instruction. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
4	NSF_IRQ_IRQ	Event triggered by the NSF2 imaging accelerator when processing of a block is done. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
3	VLCDJ_BLOC_IRQ	This event is triggered by VLCDJ when a macroblock has been processed (encode/decode) Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
2	DCT_IRQ	Event triggered when a block has been processed by the DCT module and the filter outcome has been stored to an image buffer. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
1	LDC_FRAME_IRQ	This event is triggered by LDC when a full frame has been processed Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
0	SIMCOP_DMA_IRQ0	Event triggered by the SIMCOP DMA. Check SIMCOP DMA IRQ registers. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0

**Table 8-1256. Register Call Summary for Register SIMCOP\_HL\_IRQENABLE\_CLR\_i**

ISS SIMCOP Functional Description

- [Interrupt Merger: \[0\] \[1\]](#)

ISS SIMCOP Registers Manual

- [SIMCOP Register Summary: \[2\]](#)

**Table 8-1257. SIMCOP\_CTRL**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	SIMCOP_CONTROL
<b>Physical Address</b>	See <a href="#">Table 8-1242</a> .		
<b>Description</b>	SIMCOP control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				LDC_R_BURST_BREAK	RESERVED			LDC_R_TAG_CNT	RESERVED			LDC_R_TAG_OFST	RESERVED		IMX_B_CMD	IMX_A_CMD	HUFF	QUANT	RESERVED		LDC_LUT	LDC_INPUT	NSF_WMEM	IRQ3_MODE	IRQ2_MODE	IRQ1_MODE	IRQ0_MODE				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Read returns 0.	R	0x0
28	LDC_R_BURST_BREAK	Controls if bursts issued by the LDC2 bridge could cross burst length boundaries. When this register is set, the LDC2 bridge only issues aligned bursts. Register can only be used when LDC_R_MAX_BURST_LENGTH is 32, 64, or 128 bytes.  0x0: Yes. 0x1: No. OCP transactions must be split	RW	0
27:26	LDC_R_MAX_BURST_LENGTH	Limits the maximum burst length that could be used by the LDC2 bridge  0x0: 8 x 128 0x1: 6 x 128 0x2: 4 x 128 0x3: 2 x 128	RW	0x0
25	RESERVED	Read returns 0.	R	0
24:21	LDC_R_TAG_CNT	Limits the maximum number of outstanding requests to LDC_R_TAG_CNT+1	RW	0x3
20	RESERVED	Read returns 0.	R	0
19:16	LDC_R_TAG_OFST	First OCP tag ID that can be used by LDC reads. Software must prevent overlap with tags generated by the SIMCOP DMA. Typically this value should be equal to SIMCOP_DMA_CTRL.TAG_CNT+1.	RW	0xC
15	RESERVED	Read returns 0.	R	0
14	IMX_B_CMD	Switch for iMX # command memory 0x0: Coprocessor bus 0x1: iMX B instruction read / write	RW	0
13:12	IMX_A_CMD	Switch for iMX A command memory 0x0: Coprocessor bus 0x1: iMX A instruction read/write 0x2: iMX B instruction read/write	RW	0x0
11	HUFF	Switch for Huffman table 0x0: Coprocessor 0x1: VLCDJ Huffman table read	RW	0

Bits	Field Name	Description	Type	Reset
10	QUANT	Switch for quantization table 0x0: Coprocessor bus 0x1: VLCDJ quantization table read	RW	0
9	RESERVED	Read returns 0.	R	0
8	LDC_LUT	Switch for LDC LUT 0x0: Coprocessor bus 0x1: The LDC module could access the LDC LUT.	RW	0
7:6	LDC_INPUT	Selects input data buffer for LDC. Memories attached to LDC as working memories cannot be used by any other accelerators. HWSEQ or HWSEQ software override settings are ignored for those memories. 0x0: No input memory attached 0x1: Use image buffers a and b 0x2: Use image buffers a, b, c, and d 0x3: Use LDC private input memory.	RW	0x0
5:4	NSF_WMEM	Selects working memory for NSF. Memories attached to NSF as working memories cannot be used by any other accelerators. HWSEQ or HWSEQ software override settings are ignored for those memories. 0x0: No working memory attached to NSF2. NSF2 cannot be used. 0x1: iMX A coefficient memory used. 0x2: Image buffers a and b used. Those image buffers cannot be used for other purposes. This setting has higher priority than the context configuration. 0x3: Image buffers a, b, c, and d used. Those image buffers cannot be used for other purposes. This setting has higher priority than the context configuration.	RW	0x0
3	IRQ3_MODE	Interrupt generation method 0x0: The interrupt line is asserted when one of the events enabled in SIMCOP_IRQENABLE_3 is pending. 0x1: The interrupt line is asserted when all events enabled in SIMCOP_IRQENABLE_3 are pending.	RW	0
2	IRQ2_MODE	Interrupt generation method 0x0: The interrupt line is asserted when one of the events enabled in SIMCOP_IRQENABLE_2 is pending. 0x1: The interrupt line is asserted when all events enabled in SIMCOP_IRQENABLE_2 are pending.	RW	0
1	IRQ1_MODE	Interrupt generation method 0x0: The interrupt line is asserted when one of the events enabled in SIMCOP_IRQENABLE_1 is pending. 0x1: The interrupt line is asserted when all events enabled in SIMCOP_IRQENABLE_1 are pending.	RW	0
0	IRQ0_MODE	Interrupt generation method 0x0: The interrupt line is asserted when one of the events enabled in SIMCOP_IRQENABLE_0 is pending. 0x1: The interrupt line is asserted when all events enabled in SIMCOP_IRQENABLE_0 are pending.	RW	0

**Table 8-1258. Register Call Summary for Register SIMCOP\_CTRL**

ISS SIMCOP Functional Description

- [Interrupt Merger: \[0\]](#)

ISS SIMCOP Programming Models

- [Interrupts: \[1\] \[2\]](#)

**Table 8-1258. Register Call Summary for Register SIMCOP\_CTRL (continued)**

ISS SIMCOP Registers Manual

- [SIMCOP Register Summary: \[3\]](#)

**Table 8-1259. SIMCOP\_CLKCTRL**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	SIMCOP_CONTROL
<b>Physical Address</b>	See <a href="#">Table 8-1242</a> .		
<b>Description</b>	SIMCOP clock control register. Use to enable/disable the interface and functional clock of SIMCOP submodules. Disabled modules cannot be accessed		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ROT_A	IMX_B	IMX_A	NSF2	VLCDJ	DCT	LDC	DMA

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	ROT_A	<p>ROT A</p> <p>Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off.</p> <p>Read 0x0: The submodule is off</p> <p>Read 0x1: The submodule is on</p> <p>Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.</p>	RW	0
6	IMX_B	<p>iMX B</p> <p>Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off.</p> <p>Read 0x0: The submodule is off.</p> <p>Read 0x1: The submodule is on.</p> <p>Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.</p>	RW	0
5	IMX_A	<p>iMX A</p> <p>Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off.</p> <p>Read 0x0: The submodule is off.</p> <p>Read 0x1: The submodule is on.</p> <p>Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.</p>	RW	0
4	NSF2	<p>NSF2</p> <p>Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off.</p> <p>Read 0x0: The submodule is off.</p> <p>Read 0x1: The submodule is on.</p> <p>Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.</p>	RW	0
3	VLCDJ	<p>VLCDJ</p> <p>Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off.</p> <p>Read 0x0: The submodule is off.</p> <p>Read 0x1: The submodule is on.</p> <p>Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.</p>	RW	0

Bits	Field Name	Description	Type	Reset
2	DCT	<p>DCT</p> <p>Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off.</p> <p>Read 0x0: The submodule is off.</p> <p>Read 0x1: The submodule is on.</p> <p>Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.</p>	RW	0
1	LDC	<p>LDC</p> <p>Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off.</p> <p>Read 0x0: The submodule is off.</p> <p>Read 0x1: The submodule is on.</p> <p>Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.</p>	RW	0
0	DMA	<p>DMA</p> <p>Write 0x0: Request shutdown of the submodule. No effect if the submodule clock is already off.</p> <p>Read 0x0: The submodule is off.</p> <p>Read 0x1: The submodule is on.</p> <p>Write 0x1: Request enable of the submodule. No effect if the submodule clock is already off.</p>	RW	0

**Table 8-1260. Register Call Summary for Register SIMCOP\_CLKCTRL**

ISS SIMCOP Functional Description

- [ISS SIMCOP Local Clock Management: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [ISS SIMCOP Power Management: \[13\]](#)

ISS SIMCOP Programming Models

- [ISS SIMCOP Module Global Initialization: \[14\]](#)

ISS SIMCOP Registers Manual

- [SIMCOP Register Summary: \[15\]](#)

### 8.4.1.5 ISS SIMCOP Memory Mapping

Table 8-1261 shows the SIMCOP memory mapping seen by the L3 interconnect and the Cortex-M3 MPU subsystem. SIMCOP memory mapping has a 128-KB address space.

**Table 8-1261. SIMCOP Memory Mapping**

Module	L3 Address		Cortex-M3 Address		Size (Bits)
	Begin	End	Begin	End	
Hardware sequencer/buffer registers	0x5202 0000	0x5202 00FF	0x5506 0000	0x5506 00FF	256
LDC registers	0x5202 0100	0x5202 017F	0x5506 0100	0x5506 017F	128
iMX A registers	0x5202 0180	0x5202 01BF	0x5506 0180	0x5506 01BF	64
iMX B registers	0x5202 01C0	0x5202 01FF	0x5506 01C0	0x5506 01FF	64
SIMCOP DMA registers	0x5202 0200	0x5202 03FF	0x5506 0200	0x5506 03FF	512
Reserved	0x5202 0400	0x5202 05FF	0x5506 0400	0x5506 05FF	512
VLCDJ registers	0x5202 0600	0x5202 067F	0x5506 0600	0x5506 067F	128
Reserved	0x5202 0680	0x5202 06FF	0x5506 0680	0x5506 06FF	128
ROT registers	0x5202 0700	0x5202 073F	0x5506 0700	0x5506 073F	64
Reserved	0x5202 0740	0x5202 077F	0x5506 0740	0x5506 077F	64
NSF2 registers	0x5202 0780	0x5202 07FF	0x5506 0780	0x5506 07FF	128
DCT registers	0x5202 0800	0x5202 081F	0x5506 0800	0x5506 081F	32
Reserved	0x5202 0820	0x5202 0FFF	0x5506 0820	0x5506 0FFF	2016
Bitstream buffer	0x5202 1000	0x5202 1FFF	0x5506 1000	0x5506 1FFF	4096
Huffman tables	0x5202 2000	0x5202 2FFF	0x5506 2000	0x5506 2FFF	4096
Reserved	0x5202 3000	0x5202 3FFF	0x5506 3000	0x5506 3FFF	4096
iMX B command memory	0x5202 4000	0x5202 5FFF	0x5506 4000	0x5506 5FFF	8192
iMX A command memory	0x5202 6000	0x5202 6FFF	0x5506 6000	0x5506 6FFF	4096
Quantization tables	0x5202 7000	0x5202 73FF	0x5506 7000	0x5506 73FF	1024
LDC LUT	0x5202 7400	0x5202 75FF	0x5506 7400	0x5506 75FF	512
Reserved	0x5202 7600	0x5202 7FFF	0x5506 7600	0x5506 7FFF	2560
Image buffer A	0x5202 8000	0x5202 8FFF	0x5506 8000	0x5506 8FFF	4096
Image buffer B	0x5202 9000	0x5202 9FFF	0x5506 9000	0x5506 9FFF	4096
Image buffer C	0x5202 A000	0x5202 AFFF	0x5506 A000	0x5506 AFFF	4096
Image buffer D	0x5202 B000	0x5202 BFFF	0x5506 B000	0x5506 BFFF	4096
Image buffer E	0x5202 C000	0x5202 CFFF	0x5506 C000	0x5506 CFFF	4096
Image buffer F	0x5202 D000	0x5202 DFFF	0x5506 D000	0x5506 DFFF	4096
Image buffer G	0x5202 E000	0x5202 EFFF	0x5506 E000	0x5506 EFFF	4096
Image buffer H	0x5202 F000	0x5202 FFFF	0x5506 F000	0x5506 FFFF	4096
iMX A coefficients memory	0x5203 0000	0x5203 3FFF	0x5507 0000	0x5507 3FFF	16,384
iMX B coefficients memory	0x5203 4000	0x5203 7FFF	0x5507 4000	0x5507 7FFF	16,384

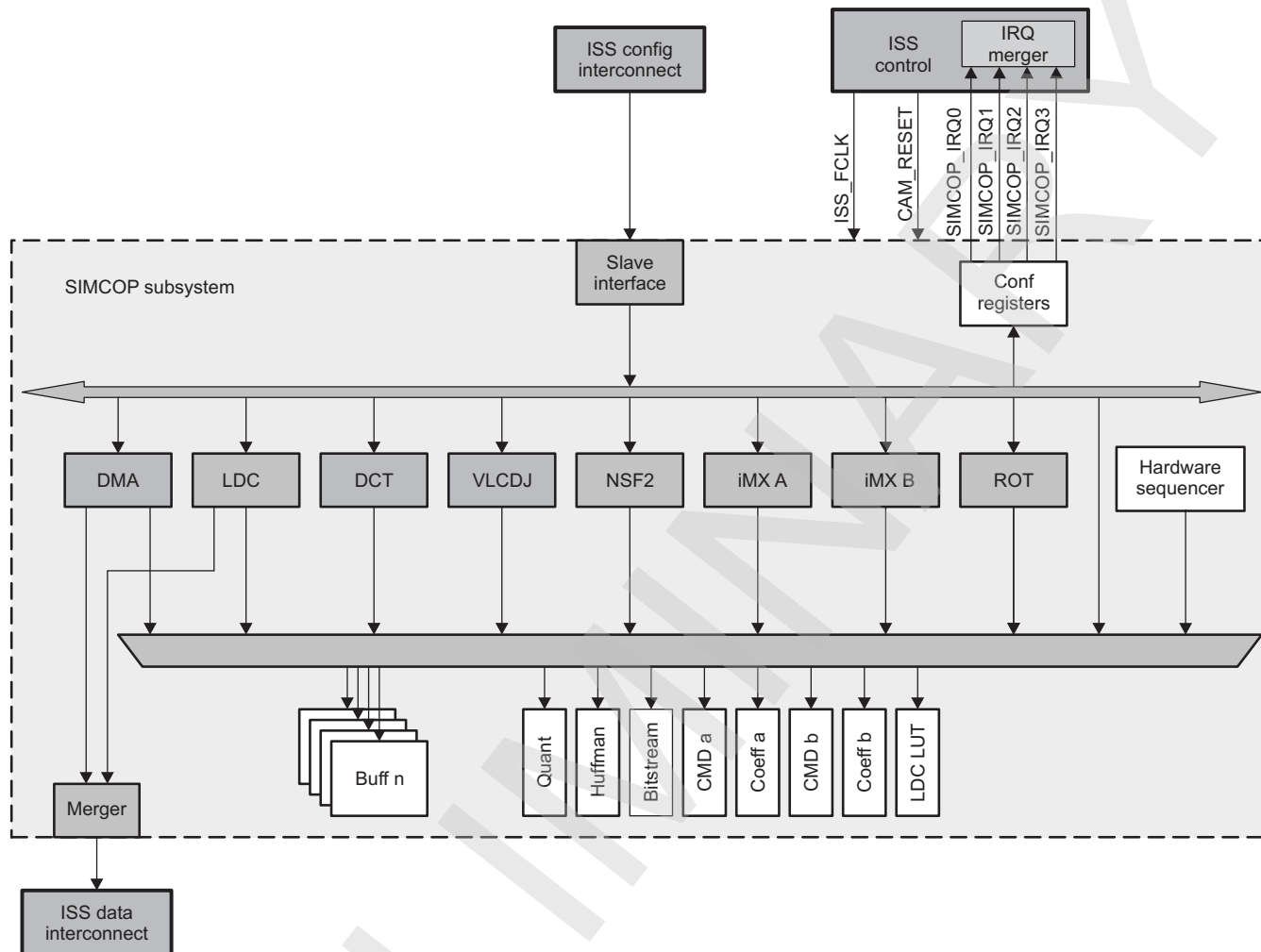
## 8.4.2 ISS SIMCOP Hardware Sequencer and Buffers

This section describes the hardware sequencer and buffers in the still image coprocessor (SIMCOP) subsystem.

### 8.4.2.1 ISS SIMCOP Hardware Sequencer and Buffers Overview

The SIMCOP hardware sequencer control all the modules included in the SIMCOP subsystem with their memories.

Figure 8-242 shows an overview of the hardware sequencer and buffers in the SIMCOP subsystem.

**Figure 8-242. Hardware Sequencer and Buffers in the SIMCOP Subsystem**

hwseq-001

- Hardware acceleration for low-level sequencing tasks:
  - Synchronizes with DMA, DCT, VLCDJ, ROT, NSF2, LDC, and iMX
  - Supports 4 memory crossbar contexts
- On-chip memories (RAM):
  - Eight image buffers, 4KB each
  - Quantization table storage: 512 bytes
  - Huffman table storage: 4KB
  - Bitstream buffer: 4KB
  - Two iMX coefficient memories: 16KB each
  - Two iMX program memories: 4KB for iMX A and 8KB for iMX B
  - LDC look-up table (LUT) storage: 256 entries
- Power-management support

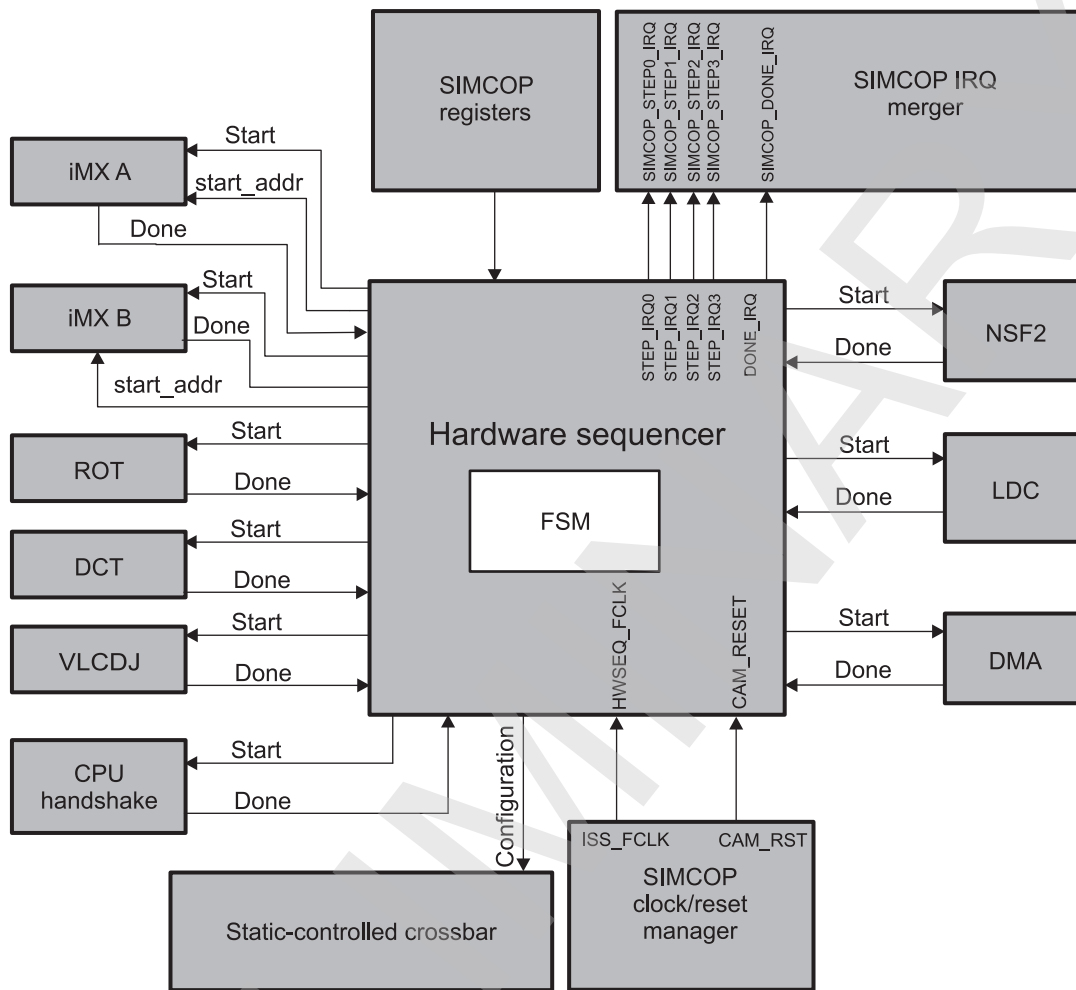
#### 8.4.2.2 ISS SIMCOP Hardware Sequencer and Buffer Integration

The hardware sequencer and buffers module is part of the SIMCOP subsystem in the imaging subsystem (ISS).

Figure 8-243 shows the integration of the hardware sequencer and buffers in the SIMCOP subsystem.



Figure 8-243. Hardware Sequencer and Buffer Integration



hwseq-011

Table 8-1262 lists the integration attributes, Table 8-1263 lists the clocks and resets, and Table 8-1264 lists the hardware requests.

Table 8-1262. Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
Hardware sequencer and buffers	PD_CAM	

Table 8-1263. Clocks and Resets

Clocks					
Module Instance	Destination Signal Name	Source Signal Name	Source	Description	
Hardware and buffers sequencer	HWSEQ_FCLK	ISS_FCLK	PRCM	Functional clock provided by ISS_CLK from PRCM . It is used by all ISS submodules and ISS top-level resources.	
Resets					
Hardware and buffers sequencer	CAM_RESET	CAM_RST	PRCM	ISS and SIMCOP global reset	

For information about clock and reset management, see [Section 8.4.1.2.1](#), *ISS SIMCOP Local Power and Clock Management*.

**Table 8-1264. Hardware Requests**

Interrupt Requests					
Module Instance		Source Signal Name	Destination Signal Name	Destination	Description
Hardware and buffers	sequencer	STEP_IRQ0	SIMCOP_STEP0_IRQ	SIMCOP IRQ merger	Event triggered when a SIMCOP context is activated by the hardware sequencer
Hardware and buffers	sequencer	STEP_IRQ1	SIMCOP_STEP1_IRQ	SIMCOP IRQ merger	Event triggered when a SIMCOP context is activated by the hardware sequencer
Hardware and buffers	sequencer	STEP_IRQ2	SIMCOP_STEP2_IRQ	SIMCOP IRQ merger	Event triggered when a SIMCOP context is activated by the hardware sequencer
Hardware and buffers	sequencer	STEP_IRQ3	SIMCOP_STEP3_IRQ	SIMCOP IRQ merger	Event triggered when a SIMCOP context is activated by the hardware sequencer
Hardware and buffers	sequencer	DONE_IRQ	SIMCOP_DONE_IRQ	SIMCOP IRQ merger	Event triggered when the hardware sequencer finishes the sequence

For more information about interrupt requests, see [Section 8.4.1.2.3](#), *Interrupt Merger*.

### 8.4.2.3 ISS SIMCOP Hardware Sequencer and Buffers Functional Description

#### 8.4.2.3.1 ISS SIMCOP Hardware Sequencer and Buffers Software Reset

For information about how to perform a software reset, see [Section 8.4.1.2.2, Software Reset](#).

#### 8.4.2.3.2 ISS SIMCOP Hardware Sequencer and Buffers Power Management

For more information about power management, see [Section 8.4.1.2.1.3, ISS SIMCOP Power Management](#).

#### 8.4.2.3.3 ISS SIMCOP Hardware Sequencer and Buffer Interrupt Requests

This module sends four interrupts to the ISS top level. For more information about interrupt management, see [Section 8.4.1.2.3, Interrupt Merger](#).

#### 8.4.2.3.4 ISS SIMCOP Hardware Sequencer Buffer Description

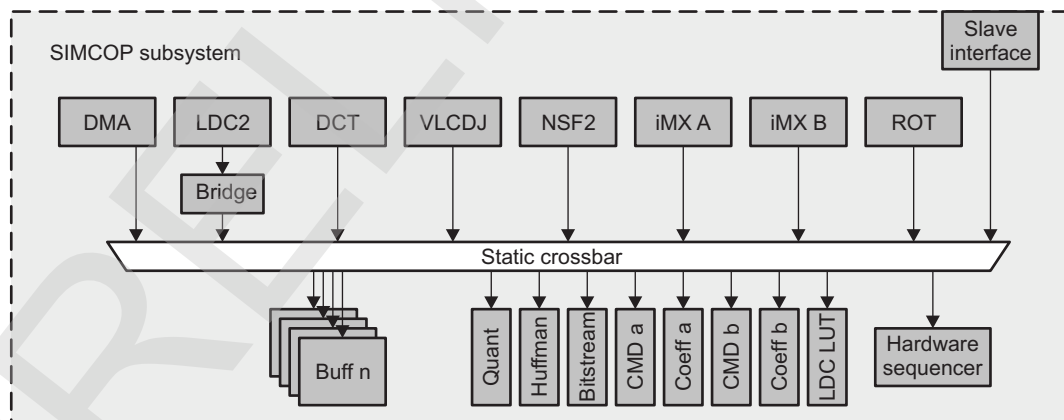
The SIMCOP subsystem includes the following memories:

- Eight image buffers, 4KB each
- Quantization table storage: 512 bytes
- Huffman table storage: 4KB
- Bitstream buffer: 4KB
- Two iMX coefficient memories: 16KB each
- Two iMX program memories: 4KB for iMX A and 8KB for iMX B
- LDC LUT storage: 256 entries
- LDC private working memory

##### 8.4.2.3.4.1 Static Crossbar

The static crossbar connects SIMCOP modules to the SIMCOP memories as shown in [Figure 8-244](#).

Figure 8-244. SIMCOP Static Crossbar Overview



The buses for QUANT, HUFF, LDC LUT, CMD a, and CMD b are controlled by an external initiator. Typically, those memories are initialized before SIMCOP starts processing, and the content remains unchanged until the full frame is processed. The amount of data transferred to those memories is low and transfer is not timing critical. [Table 8-1265](#) shows how to connect these memories to SIMCOP modules.

**NOTE:** The transfer can only be done through the slave interface (for example, CPU or system DMA [sDMA]). The SIMCOP DMA cannot access those memories.

**Table 8-1265. Static Crossbar Control Summary**

Memory	Register SIMCOP_CTRL	Value						
		Bus	iMX A inst	iMX B inst	VLCDJ HUFF	VLCDJ QUANT	LDC LUT	LDC INPUT
Quantization	[10] QUANT	0				1		
Huffman	[11] HUFF	0			1			
CMD a	[13:12] IMX_A_CMD	0	1	2				
CMD b	[14] IMX_B_CMD	0		1				
LDC LUT	[8] LDC_LUT	0					1	
LDC Priv	[7:6] LDC_INPUT							3

Image buffers and coefficient memories are used to store the data to be processed and processing results. Those memories can be accessed by the SIMCOP DMA, SIMCOP bus, and SIMCOP modules. [Table 8-1266](#) shows possible connections in the SIMCOP static crossbar: for example DCT Freq port could access image buffer A when SIMCOP\_HWSEQ\_SWITCH\_i[2:0] IMBUFF\_A=5 (i = 0, 1, 2, or 3 corresponds to the active context).

SIMCOP supports multiple contexts for the configuration of those memories. Software or the hardware sequencer can activate different contexts.

The BITSTREAM data flow is asynchronous from the macroblock pipeline. Therefore it supports dynamic buffer switching logic.

**Table 8-1266. Static Crossbar Control Summary (2/2)**

Memory	Register (SIMCOP_HWSEQ_S TEP_SWITCH_i and SIMCOP_HWSEQ_S TEP_CTRL2_i)	Value													
		Bus	DMA	iMX data A	iMX data B	VLC DJ IO	VLC DJ BS	DCT Spati al	DCT Freq	NSF2 IO	NSF2 WME M	LDC 0	LDC WME M	ROT In	ROT Out
IMBUF A	[2:0] IMBUFF A	0	1	2	3	4			5		NSF2 _WM EM = 2,3		LDC_ INPU T= 1 or 2	6	
IMBUF B	[6:4] IMBUFF B	0	1	2	3	4			5					6	
IMBUF C	[10:8] IMBUFF C	0	1	2	3	4			5		NSF2 _WM EM = 3		LDC_ INPU T= 2	6	
IMBUF D	[14:12] IMBUFF D	0	1	2	3	4			5					6	
IMBUF E	[18:16] IMBUFF E	0	1	2	3			4		5		6			7
IMBUF F	[22:20] IMBUFF F	0	1	2	3			4		5		6			7
IMBUF G	[27:24] IMBUFF G	0	1	2	3	4		5	6	8		9			7
IMBUF H	[31:28] IMBUFF H	0	1	2	3	4		5	6	8		9			7
iMX A coeff	[2:0] COEFF_A	0	1	2	3	4			5		NSF2 _WM EM = 1			6	
iMX B coeff	[6:4] COEFF_B	0	1	2	3	4			5						6
Bitstream	Handled by hardware	X	X					X							

#### 8.4.2.3.4.2 Image Buffers

Image buffers are 4 banks × 256 × 32 bits (4KB) working memories used by the SIMCOP modules to perform image processing operations. Image buffers are shared between the iMX, NSF2, DCT, LDC, VLCDJ, ROT, DMA, and SIMCOP bus accesses. Not all initiators can access the image buffers simultaneously.

Switches are controlled by the hardware sequencer or an external initiator through register configuration. Data is always organized in little-endian format.

One or multiple image buffers can be attached to a given accelerator port by configuring the adequate bits in the [SIMCOP\\_HWSEQ\\_STEP\\_SWITCH\\_i](#) registers.

The image buffer address mapping to different buses listed in [Table 8-1267](#) is controlled using the adequate [SIMCOP\\_HWSEQ\\_STEP\\_CTRL\\_i](#) register bit fields.

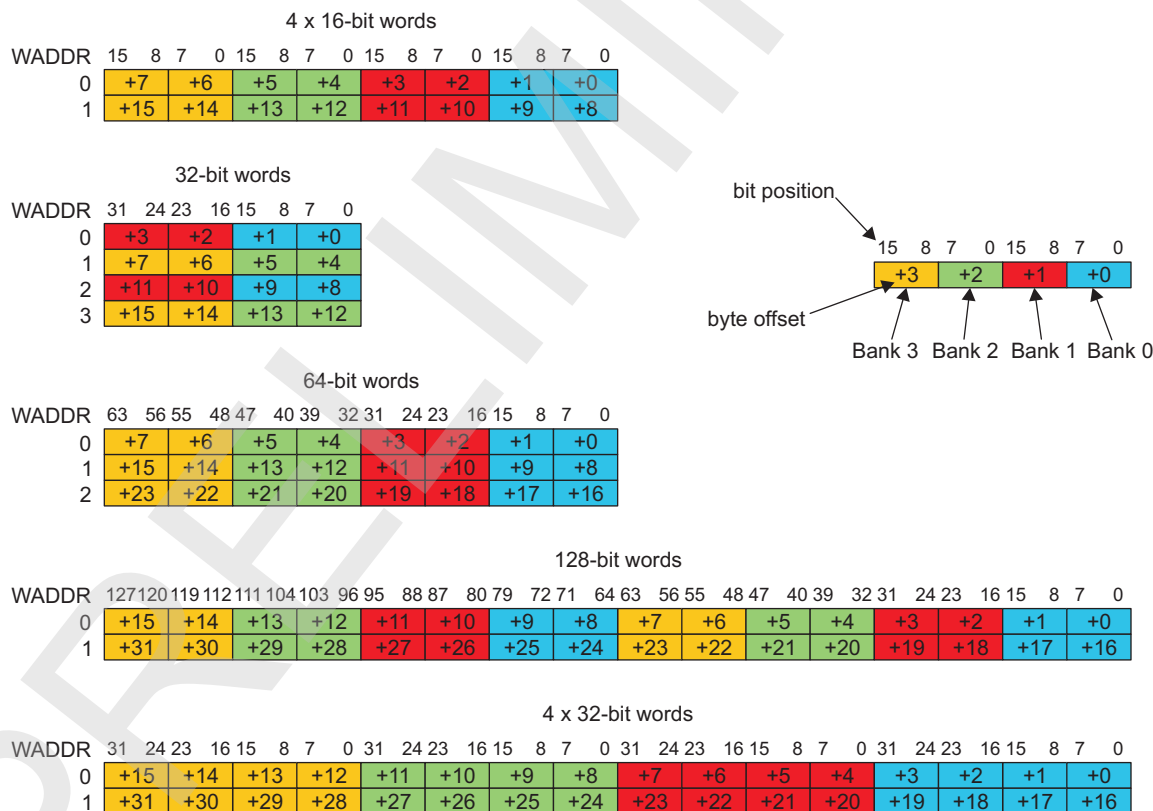
**Table 8-1267. Image Buffer Address Map**

Port	OFST	0x0000	0x1000	0x2000	0x3000	0x4000	0x5000	0x6000	0x7000
DMA	0	A	B	C	D	E	F	G	H
	1	B	C	D	E	F	G	H	A
	2	C	D	E	F	G	H	A	B
	3	D	E	F	G	H	A	B	C
	4	E	F	G	H	A	B	C	D
	5	F	G	H	A	B	C	D	E
	6	G	H	A	B	C	D	E	F
	7	H	A	B	C	D	E	F	G
iMX A data	0	A	B	C	D				
	1	C	D	E	F				
	2	E	F	G	H				
	3	G	H	A	B				
iMX B data	0	A	B	C	D				
	1	C	D	E	F				
	2	E	F	G	H				
	3	G	H	A	B				
VLCDJ I/O	0	A	B	C	D				
	1	B	C	D	G				
	2	C	D	G	H				
	3	D	G	H	A				
	4	G	H	A	B				
	5	H	A	B	C				
	6								
	7								
DCT Spatial	0	E	F						
	1	F	G						
	2	G	H						
	3	H	E						
DCT Freq	0	A	B	C	D				
	1	B	C	D	G				
	2	C	D	G	H				
	3	D	G	H	A				
	4	G	H	A	B				
	5	H	A	B	C				
	6								
	7								
NSF I/O	0	E	F						
	1	F	G						
	2	G	H						
	3	H	E						
NSF2 WMEM	N/A	A	B	C	D				

**Table 8-1267. Image Buffer Address Map (continued)**

Port	OFST	0x0000	0x1000	0x2000	0x3000	0x4000	0x5000	0x6000	0x7000
LDC O	0	E	F	G	H				
	1	F	G	H	E				
	2	G	H	E	F				
	3	H	E	F	G				
LDC W	N/A	A	B	C	D				
ROT in	0	A	B	C	D				
	1	B	C	D	A				
	2	C	D	A	B				
	3	D	A	B	C				
ROT out	0	E	F	G	H				
	1	F	G	H	E				
	2	G	H	E	F				
	3	H	E	F	G				

Figure 8-245 shows how width conversion and bank interleaving is performed. Each color corresponds to a different physical bank.

**Figure 8-245. Image Buffer Width Translation**

hwseq-003

#### 8.4.2.3.4.3 Coefficient Memories

The IMX\_COEF\_A memory is used by the NSF2 to store temporary data. The IMX\_x also uses COEFF\_A and COEFF\_B to store coefficients used during certain operations. IMX\_COEF\_A and IMX\_COEFF\_B are shared between the iMX, DMA, DCT, VLCDJ, and SIMCOP bus accesses. Only one initiator can access a coefficient memory at given time.

Switches are controlled by the hardware sequencer or an external initiator through register configuration. Data is always organized in little-endian format.

The initiators access types to the IMX\_COEFF\_x memory are summarized in Table 8-1268. Coefficient memory configuration depends on the initiators. Data is always organized little endian.

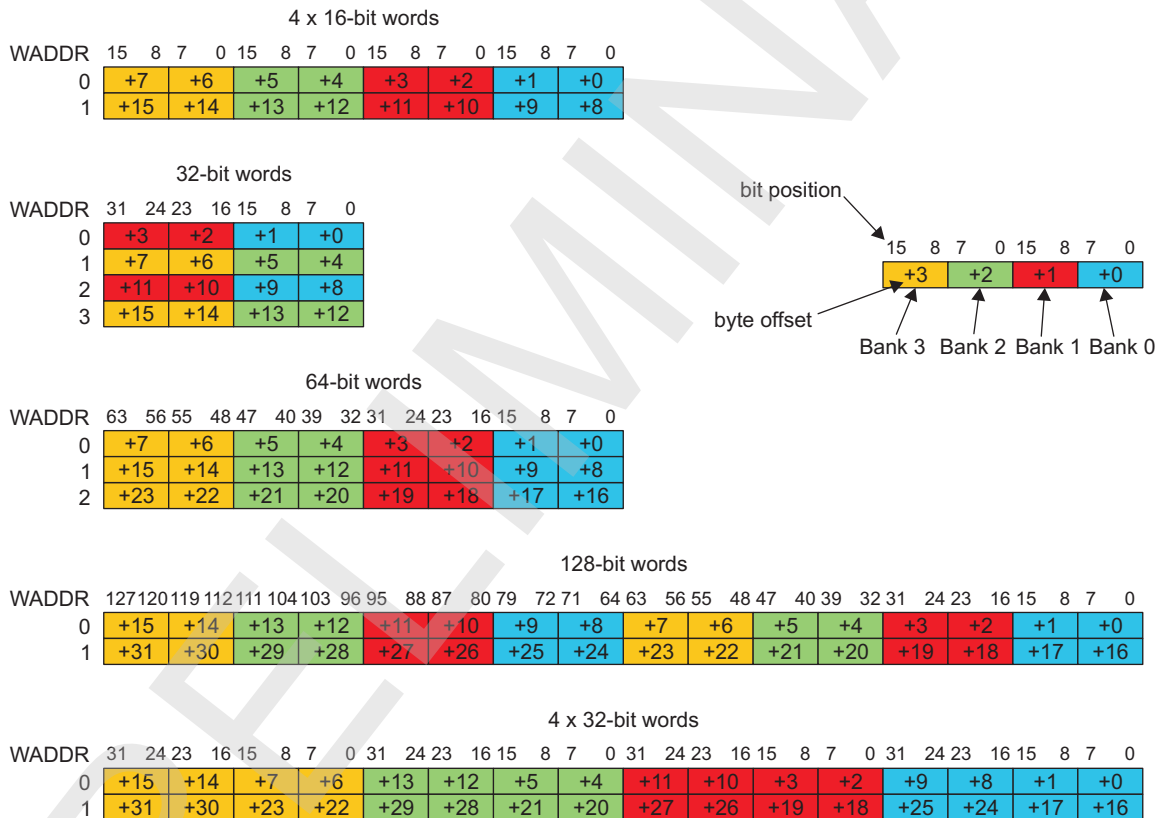
**Table 8-1268. Coefficient Buffer Organization**

IMX	Coprocessor Bus	ROT In	DMA
VLCDJ IO		ROT Out	DCT Freq
			NSF2 WMEM <sup>(1)</sup>
4 banks x 2048 words x 16 bits	4096 words x 32 bits	2048 words x 64 bits	1024 words x 128 bits

<sup>(1)</sup> IMX\_COEFF\_A only

Figure 8-246 shows how width conversion and bank interleaving is performed. Each color corresponds to a different physical bank.

**Figure 8-246. Coefficient Memory Width Translation**



hwseq-004

**8.4.2.3.4.4 Quantization Memories**

The QUANT memory is used to store the quantization and dequantization tables for the VLCDJ module.

Coefficients are coded as 16-bit values. Every color component requires 8 x 8 quantization entries, therefore the QUANT memory of SIMCOP can hold up to four tables as shown in Table 8-1269.

**Table 8-1269. Quantization Table Storage in QUANT Memory**

Table	0	1	2	3
ADDR	0x000	0x0080	0x0100	0x0180
Encode only	Y quantization	UV quantization	Unused	Unused

**Table 8-1269. Quantization Table Storage in QUANT Memory (continued)**

Table	0	1	2	3
Decode only	Unused	Unused	Y dequantization	UV dequantization
Encode and decode	Y quantization	UV quantization	Y dequantization	UV dequantization

QUANT is shared between the VLCDJ and SIMCOP bus accesses. All initiators cannot access QUANT simultaneously.

The initiators access types to the QUANT memory are summarized in [Table 8-1270](#). QUANT memory configuration depends on the initiators.

**Table 8-1270. QUANT Memory Organization**

VLCDJ	Coprocessor Bus
128 words x 32 bits	128 words x 32 bits

#### 8.4.2.3.4.5 Huffman Tables

The HUFF memory is used to store the Huffman table for the VLCDJ module. Details about the Huffman table organization could be found in the VLCDJ module description (see [Section 8.4.6, VLCDJ](#)).

HUFF memory is shared between the VLCDJ and SIMCOP bus accesses. All initiators cannot access HUFF simultaneously.

The initiators access types to the HUFF memory are summarized in [Table 8-1271](#). HUFF memory configuration depends on the initiators.

**Table 8-1271. HUFF Memory Organization**

VLCDJ	Coprocessor Bus
1024 words x 32 bits	1024 words x 32 bits

#### 8.4.2.3.4.6 Bitstream Buffer

##### 8.4.2.3.4.6.1 Overview

The bitstream buffer is used to store the VLCDJ bitstream. The bitstream buffer is shared between the VLCDJ, DMA, and SIMCOP bus accesses.

The bitstream buffer can be used in manual or automatic mode.

In manual mode, software selects if SIMCOP DMA, VLCDJ bitstream port, or SIMCOP bus could access the bitstream buffer. Manual mode is typically used to interleave JPEG encode and decode, for example, to support JPEG transcoding.

In automatic mode ([SIMCOP\\_HWSEQ\\_CTRL\[6:4\] BITSTREAM = ENCODE or DECODE](#)) the bitstream buffer is handled as two banks to prevent access collision between VLCDJ and SIMCOP DMA. A dedicated hardware engine (BBM) is used to ensure that VLCDJ and SIMCOP DMA access separate banks.

Automatic mode is disabled by writing [SIMCOP\\_HWSEQ\\_CTRL\[6:4\] BITSTREAM = COPR](#).

The bitstream buffer mapping into the SIMCOP DMA address map depends on the [SIMCOP\\_HWSEQ\\_CTRL\[6:4\] BITSTREAM](#) and [SIMCOP\\_HWSEQ\\_CTRL\[3:2\] BITSTR\\_XFER\\_SIZE](#) bit fields and the internal state of the BBM. [Table 8-1272](#) lists the mapping of the bitstream buffer for SIMCOP DMA.

**Table 8-1272. Bitstream Buffer Mapping to SIMCOP DMA**

<a href="#">SIMCOP_HWSEQ_CTRL</a>		Address Map		BBM count	Bitstream buffer	
BITSTREAM	XFER_SIZE					
0: COPRO	N/A	0x1000	0x1FFF	N/A	0x000	0xFF



**Table 8-1272. Bitstream Buffer Mapping to SIMCOP DMA (continued)**

SIMCOP_HWSEQ_CTRL		Address Map		BBM count	Bitstream buffer		
BITSTREAM	XFER_SIZE						
5: ENCODE  or  6: DECODE	2048 bytes	0x1000	0x17FF	N/A	0x000	0x7FF	
				N/A	0x800	0xFFF	
		0x1800	0x1FFF	N/A	N/A	N/A	
	1024 bytes	0x1000	0x13FF	0	0x000	0x3FF	
				1	0x400	0x7FF	
		0x1400	0x1FFF	N/A	N/A	N/A	
	512 bytes	0x1000	0x11FF	0	0x000	0x1FF	
				1	0x200	0x3FF	
				2	0x400	0x5FF	
				3	0x600	0x7FF	
			0x1200	0x1FFF	N/A	N/A	N/A
	256 bytes	0x1000	0x10FF	0	0x000	0x0FF	
1				0x100	0x1FF		
2				0x200	0x2FF		
3				0x300	0x3FF		
4				0x400	0x4FF		
5				0x500	0x5FF		
6				0x600	0x6FF		
7				0x700	0x7FF		
0x1100				0x1FFF	N/A	N/A	N/A

**8.4.2.3.4.6.2 Automatic Bitstream Buffer Management**

To enter automatic mode software must write the following values into the BITSTREAM configuration register:

- SIMCOP\_HWSEQ\_CTRL[6:4] BITSTREAM=COPR (required to reset BBM)
- SIMCOP\_HWSEQ\_CTRL[6:4] BITSTREAM=JPEG\_ENCODE or JPEG\_DECODE

**8.4.2.3.4.7 LDC Lookup Table**

The LDC\_LUT memory is used to store LUTs for the LDC module. LDC\_LUT is shared between the LDC and SIMCOP bus accesses. All initiators cannot access LDC\_LUT simultaneously.

The initiators access types to the QUANT memory are summarized in Table 8-1273. QUANT memory configuration depends on the initiators.

**Table 8-1273. LDC\_LUT Memory Organization**

LDC	Coprocessor Bus
256 words x 14 bits	128 words x 32 bits (see Table 8-1274)

**Table 8-1274. LUT Memory Format on 32-Bit Words**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESE		LDC_LUT [2a+1]														RESE		LDC_LUT [2a+0]													
RVED																RVED															

#### 8.4.2.3.4.8 LDC Private Input Memory

The LDC has a private input memory that can be used when LDC operates in YUV 4:2:0 mode. For YUV 4:2:2 and RAW operation, LDC must use image buffers as input memories.

The LDC input memory is dedicated for LDC operation. It cannot be accessed from the SIMCOP bus or by any other SIMCOP module.

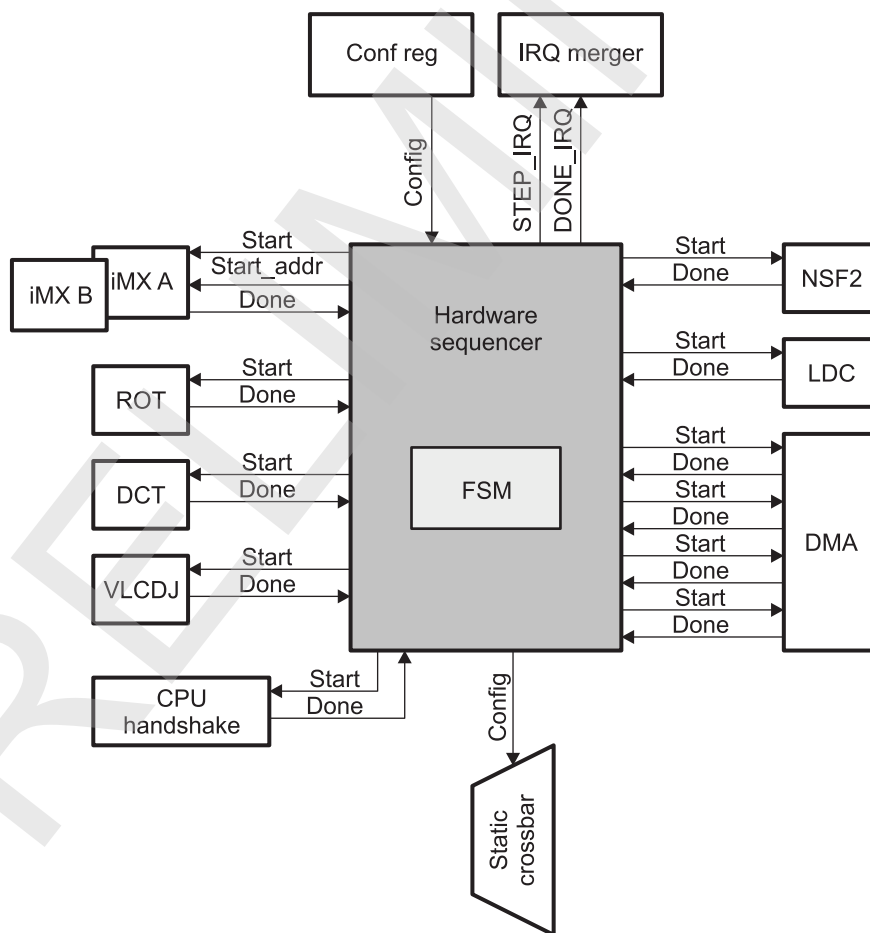
#### 8.4.2.3.5 ISS SIMCOP Hardware Sequencer

SIMCOP typically processes a full frame on a macroblock by macroblock basis. SIMCOP modules and the DMA controller trigger an interrupt when they have completed a macroblock. This event can trigger an interrupt to an external initiator (manual mode) or trigger a context switch and resume DMA/SIMCOP module processing. Typically, initiator intervention is needed to fill the macroblock pipeline (called pipe-up. For example, first n macroblocks of an image) and to flush the macroblock pipeline (called pipe-down. For example, last n macroblocks of an image). Automatic sequencing mode is typically used for the rest of the image.

The hardware sequencer is particularly useful to offload the initiator during JPEG encode/decode, ROT, LDC, NSF2, or iMX processing.

Figure 8-247 shows the SIMCOP hardware sequencer overview.

**Figure 8-247. SIMCOP Hardware Sequencer Overview**



hwseq-008

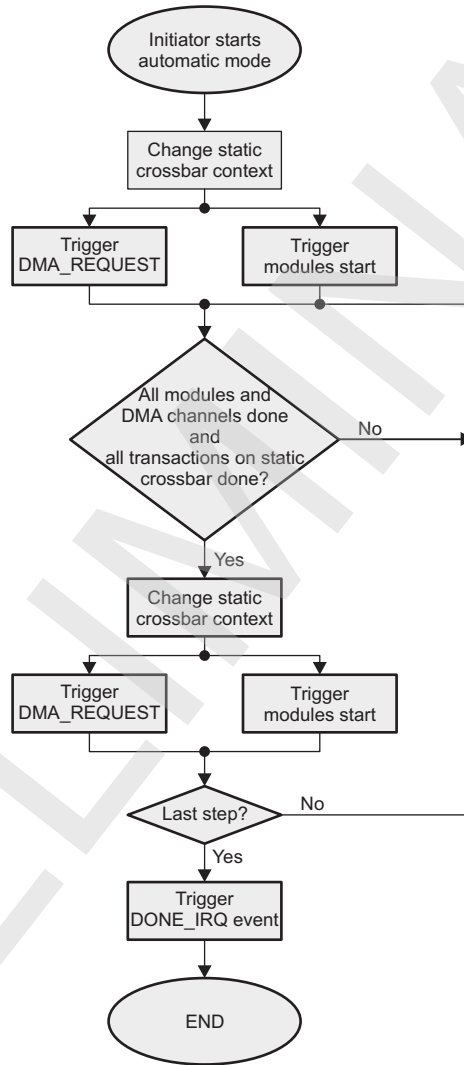
The hardware sequencer can automatically execute a predefined number of sequencing steps without

software intervention. That is called automatic operation. Automatic operation can also be used to pipeline filling and flushing with some limited software intervention. During automatic operation, the hardware sequencer controls generation of START/DONE pulses and the connections in the static-controlled crossbar. Software can take control over some resources while the hardware sequencer is running. That is called hardware sequencer override.

8.4.2.3.5.1 Automatic Operation

Figure 8-248 shows the flow diagram executed by the hardware sequencer.

Figure 8-248. Hardware Sequencing Example



hwseq-009

The external initiator configures the different SIMCOP modules and DMA channels. When this is done, the initiator enables the hardware sequencer. This resets the internal state of the hardware sequencer (pending events, context switch counter).

It starts by running sequence step [SIMCOP\\_HWSEQ\\_CTRL\[12:11\] STEP](#):

- Clear all pending completion events that have been selected for step 0 in the hardware sequencer. This does not affect the SIMCOP\_IRQSTATUS register.
- Apply step [SIMCOP\\_HWSEQ\\_CTRL\[12:11\] STEP](#) configuration to the static crossbar.
- Start DMA transfers. Multiple channels can be enabled sequentially (for example, YUV 4:2:0 data transfers or SIMCOP→SDRAM followed by SDRAM→SIMCOP) by using the SIMCOP DMA linking feature.

- Start hardware accelerators. The list of accelerators to start can be selected by software using the `SIMCOP_HWSEQ_STEP_CTRL_i[7:0] * _SYNC` register.

The hardware sequencer then waits for completion of events from selected SIMCOP modules and SIMCOP DMA channels.

The hardware sequencer also ensures that all SIMCOP memory accesses requested by accelerators and DMA channels enabled for the synchronization step are complete. This condition prevents data corruption due to early switching.

Steps can be chained to define the sequence to execute. Steps are started until the step counter reaches the limit defined by software. When the last step has completed, a `DONE_IRQ` event is triggered.

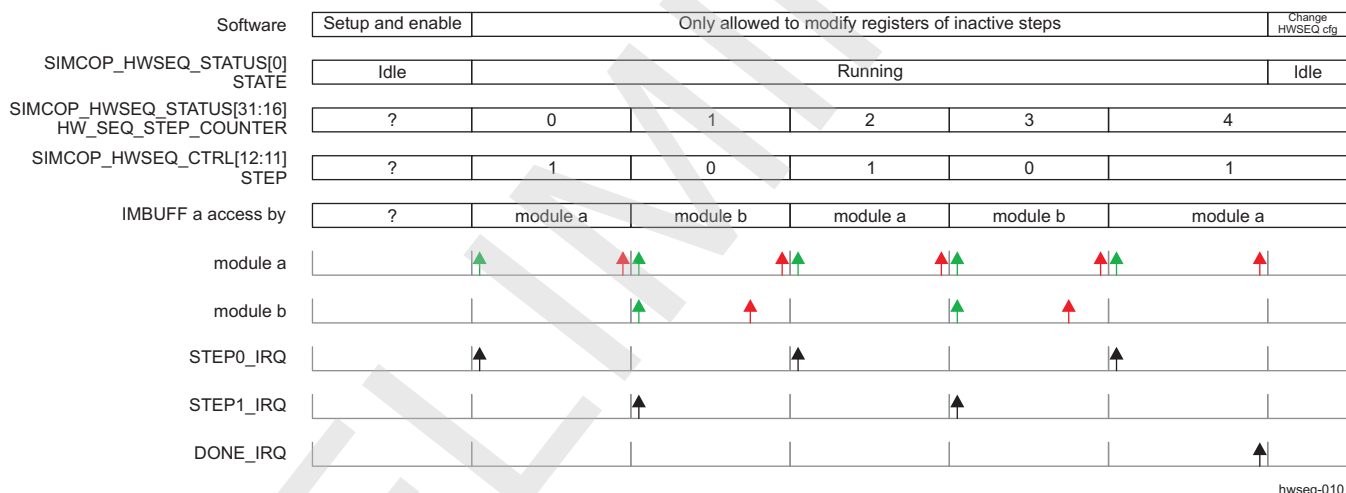
SIMCOP can execute one hardware sequence at a time. However, software can be used to execute multiple independent sequences in parallel (for example, two unrelated macroblock pipelines) using the override feature.

The hardware sequencer has no specific support for macroblock pipeline filling (pipe-up) or flushing (pipe-down). It needs some software intervention.

The hardware sequencer supports sequences composed of up to four steps. Longer sequences require software intervention.

Figure 8-248 shows hardware sequencer operation.

**Figure 8-249. Hardware Sequencer Operation Example**



The example involves use of two SIMCOP modules (or DMA channels) that are synchronized using the hardware sequencer. A total of five sequencing steps are executed (`SIMCOP_HWSEQ_CTRL[31:6] HW_SEQ_STEP_COUNTER=5`). The sequence is 2 steps long and starts with step 1. These two steps are configured using the `SIMCOP_HWSEQ_STEP_*_i` registers (where  $i = 0$  and  $1$ ).

SIMCOP modules processing is started using `START` pulses (green). When SIMCOP modules are done they return a `DONE` pulse (red). Software could choose which accelerators to run for a given step using the `SIMCOP_HWSEQ_STEP_CTRL_i[7:0] * _SYNC` registers. They do not need to finish at the same time because the hardware sequencer waits for reception of all `DONE` pulses of used accelerators before moving to the next step.

The hardware sequencer also triggers a series of interrupts that can be used by software. For example, it is possible for software to change the configuration of inactive sequencing steps. This can be particularly useful to run longer sequences than what is supported by hardware.

For example, software could setup a 4-step looping sequence: `Step 0 → 1 → 2 → 3 → 0 → 1 → ...`. When the `STEP1_IRQ` event is triggered, software could load the configuration into the `SIMCOP_HWSEQ_STEP_*_i` registers, to be used next time step 0 becomes active.

#### 8.4.2.3.5.2 Hardware Sequencer Override

By default, the hardware sequencer controls START/DONE pulses of all SIMCOP modules as well as the static-controlled crossbar. Software can take control over some resources by setting the adequate bits in the `SIMCOP_HWSEQ_OVERRIDE` register. Sequencing resources under software control are managed using the `SIMCOP_HWSEQ_STEP_CTRL_OVERRIDE` and `SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE` registers.

Changes to the `SIMCOP_HWSEQ_STEP_CTRL_OVERRIDE[27:11] *_OFST` and `SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE` registers have immediate effect. Software must ensure there is no active traffic on the impacted connection in the static-controlled crossbar.

For example, when software wants to alternatively attach image buffers a and b to SIMCOP\_DMA while preserving the buffer start address in the SIMCOP DMA address map constant it sets the following for the full application duration:

- `SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE[2:0] IMBUFF_A = 1`
- `SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE[6:4] IMBUFF_B = 1`

It then changes the `SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE[2:0] IMBUFF_A` and `SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE[6:4] IMBUFF_B` registers to attach the image buffers to the correct SIMCOP submodule.

The settings in the `SIMCOP_HWSEQ_STEP_CTRL_OVERRIDE` and `SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE` registers for resources controlled by the hardware sequencer have no effect.

Synchronization with SIMCOP modules is controlled using the `SIMCOP_HWSEQ_STEP_CTRL_OVERRIDE[7:0] *_TRIGGER` bit fields. For ROT, NSF, VLCDJ, DCT, and LDC, a START pulse is sent to the module when 1 is written into the adequate bit field. Software can trigger one or multiple start pulses at the time. The `*_TRIGGER` bit field is cleared by writing 1. It is automatically set by hardware when a DONE pulse is received from the relevant SIMCOP module.

Software can poll the status of the `SIMCOP_HWSEQ_STEP_CTRL_OVERRIDE[7:0] *_TRIGGER` bit field to detect when a SIMCOP module has completed its processing task. Alternatively, software can also use an interrupt. SIMCOP can merge one or multiple internal events into SIMCOP outgoing interrupts. Software can choose between two modes for each interrupt output (`SIMCOP_CTRL[3:0] IRQx_MODE`):

- 0x0: OR, the interrupt is asserted when one of the selected events has occurred.
- 0x1: AND, the interrupt is asserted when all selected events have occurred.

The AND mode leads to lower initiator load. In fact, the initiator is only interrupted once per sequencing step, even when events from multiple SIMCOP submodules are expected. Also, it does not need to poll the `SIMCOP_HL_IRQSTATUS_i` register to detect which events have occurred: it simply must write 0xFFFF into the `SIMCOP_HL_IRQSTATUS_i` register to clear all pending events.

The SIMCOP DMA START/DONE control is slightly different. Software can trigger one or multiple DMA channels by writing into the `SIMCOP_HWSEQ_STEP_CTRL_OVERRIDE[7:5] DMA_TRIGGER` bit field. The `DMA_TRIGGER` bit field returns the written value when all expected DONE pulses are received from SIMCOP DMA.

Alternatively, SIMCOP modules (except LDC) may be triggered by direct writes into their configuration registers. However, this approach leads to higher initiator load because multiple writes must be done.

## 8.4.2.4 ISS SIMCOP Hardware Sequencer and Buffers Basic Programming Model

### 8.4.2.4.1 ISS SIMCOP Application Programming Principle

Software must first configure SIMCOP access tag handling (SIMCOP\_CTRL[24:21] LDC\_R\_TAG\_CNT, SIMCOP\_CTRL[19:16] LDC\_R\_TAG\_OFST, and SIMCOP\_DMA\_CTRL[7:4] TAG\_CNT). Software must ensure there is no active traffic on the SIMCOP master port before changing those registers.

Then it initializes Huffman, Quantization, and LDC LUT if they are used. Those memories must first be attached to the SIMCOP bus so that they could be accessed from the slave port (SIMCOP\_CTRL[11] HUFF, SIMCOP\_CTRL[10] QUANT, and SIMCOP\_CTRL[7:6] LDC\_LUT). Then they could be preloaded using either an external CPU or an external DMA engine. SIMCOP DMA can not preload those tables.

Similarly, software could also preload data like filter coefficients into iMX coefficient memories or image buffers. This could be handled by SIMCOP DMA:

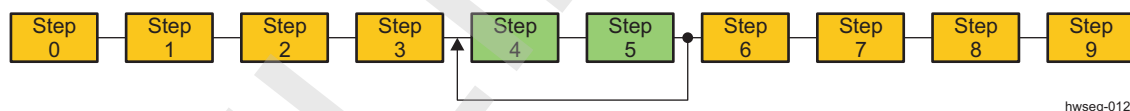
- The buffers must first be attached to SIMCOP DMA using the [SIMCOP\\_HWSEQ\\_OVERRIDE](#) and [SIMCOP\\_HWSEQ\\_STEP\\_SWITCH\\_OVERRIDE](#) registers.
- Alternatively, software could use the SIMCOP\_HWSEQ\_STEP\_\*\_i registers (where i = 0) instead of the override feature. The hardware sequencer ensures that the configuration of step 0 is applied when the hardware sequencer is disabled.
- Software then configures and starts one or multiple data transfers using the SIMCOP DMA.
- Software monitors SIMCOP DMA events to know when DMA transfers are completed. Other configuration tasks may happen concurrently, but software must ensure that transfers complete before attaching the memories to another module.

Software should then enable the modules to be used.

Typically, the first steps and last steps of the macroblock pipeline are different from the ones in the middle of the pipeline. The hardware sequencer has no specific support for pipe-up and pipe-down. However, software could cut the sequence in smaller sequences to still benefit from the hardware sequencing and to avoid software sequencing.

Figure 8-250 shows an example of a 10-step sequence involving two DMA transfers and three modules.

**Figure 8-250. Typical Hardware Sequencer Controlled Sequencer With Pipe-Up and Pipe-Down**



The macroblock pipeline is composed of a chain of three modules. Data between modules is exchanged using ping-pong buffering.

Pipe-up and pipe-down require four sequencing steps each. The main processing could be done with two steps.

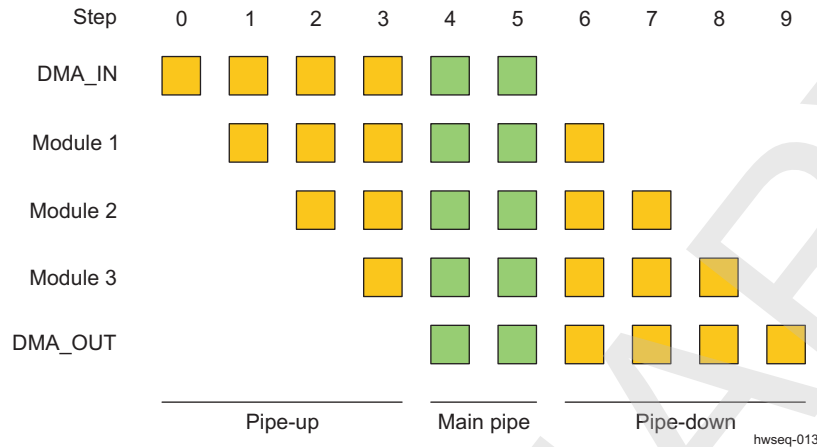
Software configures the hardware sequencer with the 4-step pipe-up sequence. Each step is executed once. When pipe-up is done, SIMCOP triggers the DONE\_IRQ event.

Software then reconfigures the hardware sequencer for the 2-step main sequence. The number of steps is defined by the amount of data to process. When the main pipe is done, SIMCOP triggers the DONE\_IRQ event.

Software finally reconfigures the hardware sequencer with the 4-step pipe-down sequence. Each step is executed once. When pipe-down is done, SIMCOP triggers the DONE\_IRQ event.

Figure 8-251 shows an example of pipe-up and pipe-down.

**Figure 8-251. Pipe-Up and Pipe-Down**



The hardware sequencer configuration determines which module could access a given SIMCOP memory for a sequence step.

**8.4.2.4.2 External CPU Use for Data Processing**

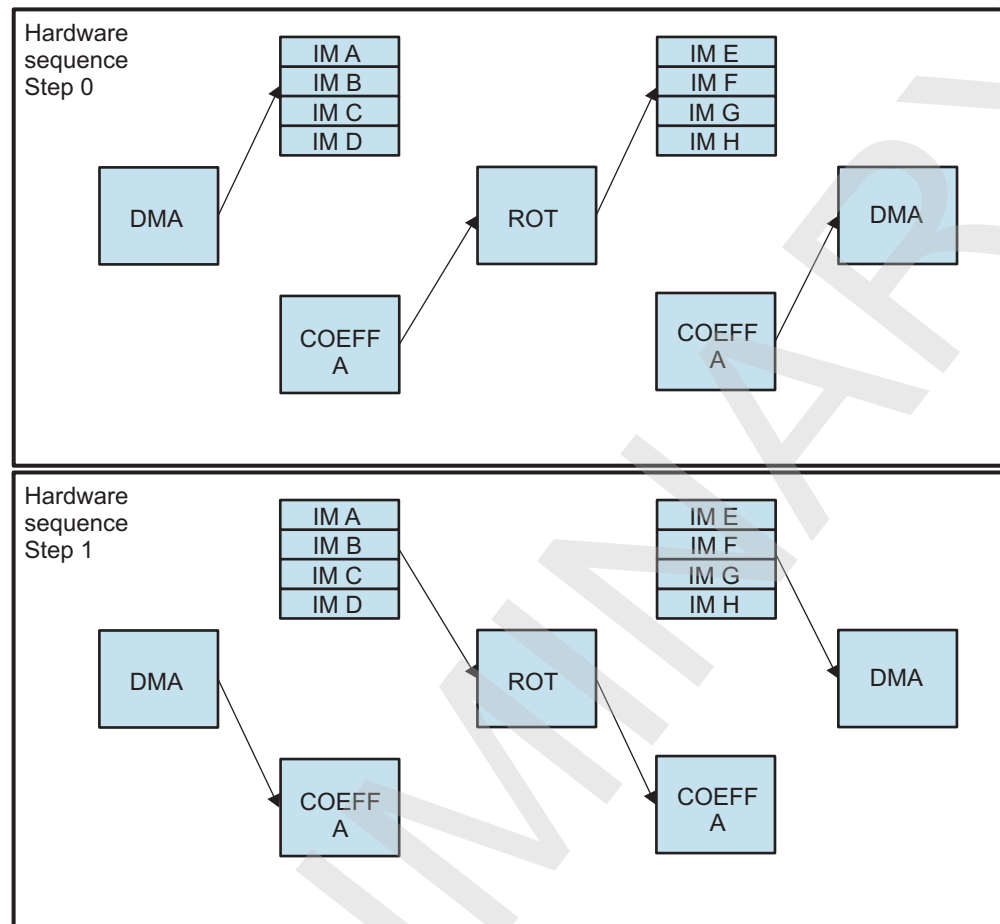
An external CPU could be used to perform some data processing as SIMCOP hardware accelerators do. A similar mechanism to the START/DONE pulse interaction with SIMCOP hardware accelerators is implemented.

The external CPU receives the request to start processing by the CPU\_PROC\_START\_IRQ interrupt merged at SIMCOP subsystem level. This interrupt is enabled and acknowledged as usual.

When processing completes, the CPU generates the DONE pulse to the hardware sequencer by setting the [SIMCOP\\_HWSEQ\\_CTRL\[10\] CPU\\_PROC\\_DONE](#) bit to 1.

**8.4.2.4.3 Rotation Operational Mode Configuration**

The macroblock pipeline is controlled by a 2-step hardware sequence:

**Figure 8-252. Rotation Macroblock Pipeline**

hwseq-014

**Table 8-1275. Rotation Macroblock Pipeline**

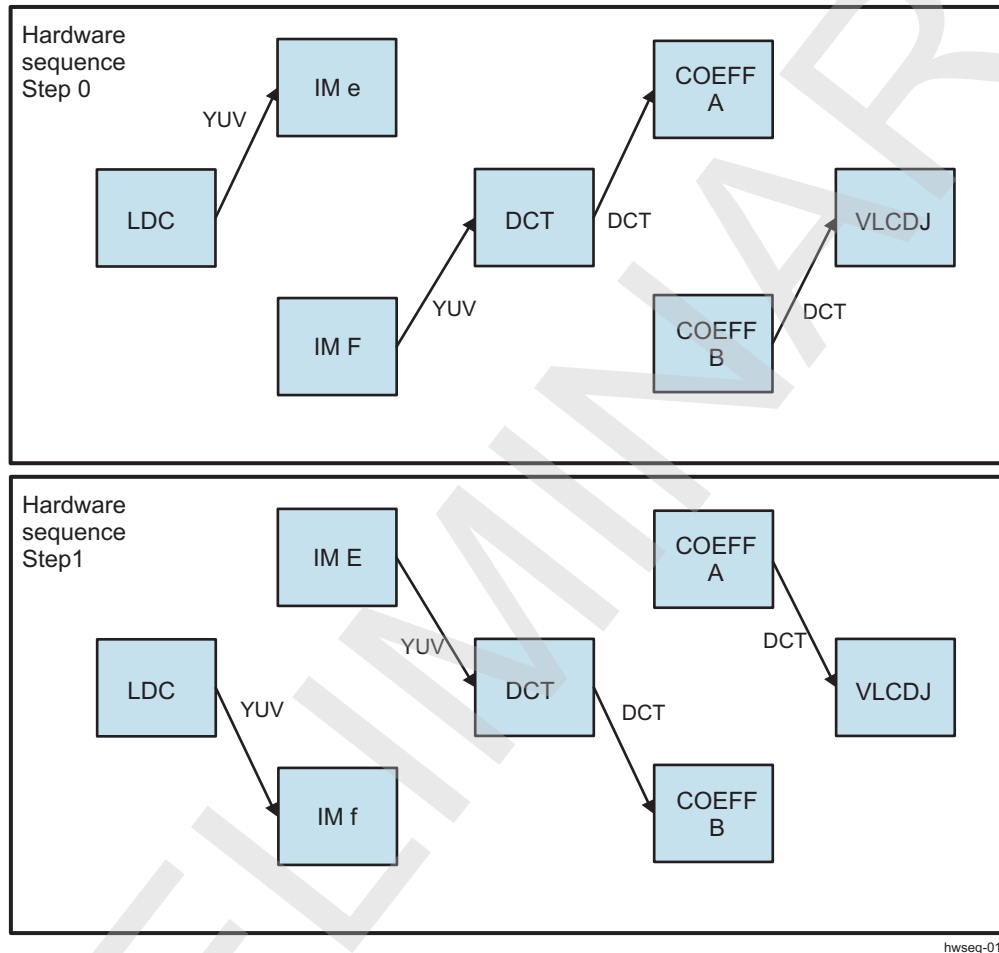
Step	Register/Bit Field/Programming Model	Value
Set DMA to access IMBUFF_A	<a href="#">SIMCOP_HWSEQ_STEP_SWITCH_i[2:0]</a> IMBUFF_A (i = 0)	0x1
Set ROT_A_O to access IMBUFF_E	<a href="#">SIMCOP_HWSEQ_STEP_SWITCH_i[18:16]</a> IMBUFF_E (i = 0)	0x7
Set ROT_A_I to access COEFF_A	<a href="#">SIMCOP_HWSEQ_STEP_CTRL2_j[2:0]</a> = COEFF_A (i = 0)	0x6
Set DMA to access COEFF_B	<a href="#">SIMCOP_HWSEQ_STEP_CTRL2_i[6:4]</a> COEFF_B (i = 0)	0x1
Set ROT_A_I to access IMBUFF_A	<a href="#">SIMCOP_HWSEQ_STEP_SWITCH_i[2:0]</a> IMBUFF_A (i = 1)	0x6
Set DMA to access IMBUFF_E	<a href="#">SIMCOP_HWSEQ_STEP_SWITCH_i[18:16]</a> IMBUFF_E (i = 1)	0x1
Set DMA to access COEFF_A	<a href="#">SIMCOP_HWSEQ_STEP_CTRL2_j[2:0]</a> COEFF_A (i = 1)	0x1
Set ROT_A_O to access COEFF_B	<a href="#">SIMCOP_HWSEQ_STEP_CTRL2_i[6:4]</a> COEFF_B (i = 1)	0x6



### 8.4.2.4.4 LDC Operational Mode Configuration

The LDC module fetches data from system memory, performs the transformation and stores results into image buffers. DCT and VLCDJ modules are used to JPEG compress this data and store the results to the bitstream buffer. SIMCOP DMA is used to transfer compressed data from bitstream buffer to system memory.

Figure 8-253. LDC Transformation Pipeline



hwseq-015

Table 8-1276. LDC Transformation

Step	Register/Bit Field/Programming Model	Value
Set IMBUFF_A and IMCUFF_B as input buffer	SIMCOP_CTRL[7:6] LDC_INPUT	0x1
Set LDC to access IMBUFF_E	SIMCOP_HWSEQ_STEP_SWITCH_i[18:16] IMBUFF_E	0x6
Set DCT to access COEFF_A	SIMCOP_HWSEQ_STEP_CTRL2_i[2:0] COEFF_A	0x5
Set VLCDJ_IO to access COEFF_B	SIMCOP_HWSEQ_STEP_CTRL2_i[6:4] COEFF_B	0x4
Set DCT to access IMBUFF_E	SIMCOP_HWSEQ_STEP_SWITCH_i[18:16] IMBUFF_E	0x4
Set LDC to access IMBUFF_F	SIMCOP_HWSEQ_STEP_SWITCH_i[18:16] IMBUFF_F	0x6
Set VLCDJ_IO to access COEFF_A	SIMCOP_HWSEQ_STEP_CTRL2_i[2:0] COEFF_A	0x4
Set DCT to access COEFF_B	SIMCOP_HWSEQ_STEP_CTRL2_i[6:4] COEFF_B	0x5

#### 8.4.2.4.5 Concurrent Software and Hardware Sequencing

Software could perform sequencing tasks concurrently to hardware sequencing. This is supported through the hardware sequencer override registers:

- [SIMCOP\\_HWSEQ\\_OVERRIDE](#)
- [SIMCOP\\_HWSEQ\\_STEP\\_CTRL\\_OVERRIDE](#)
- [SIMCOP\\_HWSEQ\\_STEP\\_SWITCH\\_OVERRIDE](#)
- [SIMCOP\\_HWSEQ\\_STEP\\_CTRL2\\_OVERRIDE](#)

Software could select what hardware resources are controlled by hardware sequencer and what resources are controlled by software. Configuration parameters for software controlled resources are provided using those registers as well.

**Table 8-1277. Concurrent Software and Hardware Sequencing**

Step	Register/Bit Field/Programming Model	Value
Set software to control IMBUFF_E	<a href="#">SIMCOP_HWSEQ_OVERRIDE</a> [13] IMBUFF_E	0x1
Set DMA to access IMBUFF_E	<a href="#">SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE</a> [1:8:16] IMBUFF_E	0x1
Set software to control IMBUFF_F	<a href="#">SIMCOP_HWSEQ_OVERRIDE</a> [14] IMBUFF_F	0x1
Set DCT_S to access IMBUFF_F	<a href="#">SIMCOP_HWSEQ_STEP_SWITCH_OVERRIDE</a> [2:2:20] IMBUFF_F	0x4
Set software to control COEFF_A	<a href="#">SIMCOP_HWSEQ_OVERRIDE</a> [17] COEFF_A	0x1
Set iMX A to access the COEFF_A	<a href="#">SIMCOP_HWSEQ_STEP_CTRL2_OVERRIDE</a> [2:0] COEFF_A	0x2
Set hardware sequencer to control COEFF_B	<a href="#">SIMCOP_HWSEQ_OVERRIDE</a> [18] COEFF_B	0x0
Set iMX B to access the COEFF_B	<a href="#">SIMCOP_HWSEQ_STEP_CTRL2_i</a> [6:4] COEFF_B	0x3

### 8.4.2.5 ISS SIMCOP Hardware Sequencer and Buffers Register Manual

#### 8.4.2.5.1 Hardware Sequencer Instance Summary

Table 8-1278 summarizes the hardware sequencer instance.

**Table 8-1278. Hardware Sequencer Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-M3 Private Access	Size
HWSEQ	0x5202 0000	0x5506 0000	256 bytes

**NOTE:** Private Access is an access which is not using the L3/L4 interconnects.

#### 8.4.2.5.2 Hardware Sequencer Registers

##### 8.4.2.5.2.1 Hardware Sequencer Register Summary

Table 8-1279 lists the HWSEQ registers mapping.

**Table 8-1279. HWSEQ Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-M3 Private Access
SIMCOP_HWSEQ_CTRL	RW	32	0x0000 0068	0x5202 0068	0x5506 0068
SIMCOP_HWSEQ_STATUS	R	32	0x0000 006C	0x5202 006C	0x5506 006C
SIMCOP_HWSEQ_OVERRIDE	RW	32	0x0000 0070	0x5202 0070	0x5506 0070
SIMCOP_HWSEQ_STEP_CTR L_OVERRIDE	RW	32	0x0000 0074	0x5202 0074	0x5506 0074
SIMCOP_HWSEQ_STEP_SWIT CH_OVERRIDE	RW	32	0x0000 0078	0x5202 0078	0x5506 0078
SIMCOP_HWSEQ_STEP_CTR L2_OVERRIDE	RW	32	0x0000 007C	0x5202 007C	0x5506 007C
SIMCOP_HWSEQ_STEP_CTR L <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0080 + (0x10 * i)	0x5202 0080 + (0x10 * i)	0x5506 0080 + (0x10 * i)
SIMCOP_HWSEQ_STEP_SWIT CH <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0084 + (0x10 * i)	0x5202 0084 + (0x10 * i)	0x5506 0084 + (0x10 * i)
SIMCOP_HWSEQ_STEP_IMX_CTRL <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0088 + (0x10 * i)	0x5202 0088 + (0x10 * i)	0x5506 0088 + (0x10 * i)
SIMCOP_HWSEQ_STEP_CTR L2 <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 008C + (0x10 * i)	0x5202 008C + (0x10 * i)	0x5506 008C + (0x10 * i)

<sup>(1)</sup> i = 0 to 3

##### 8.4.2.5.2.2 Hardware Sequencer Register Descriptions

Table 8-1280 through Table 8-1298 describe the registers in detail.

**Table 8-1280. SIMCOP\_HWSEQ\_CTRL**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	HWSEQ
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .		
<b>Description</b>	SIMCOP hardware sequencer control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HW_SEQ_STEP_COUNTER																RESERVED		STEP		CPU_PROC_DONE	BBM_SYNC_CHAN	BBM_STATUS	BITSTREAM		BITSTR_XFER_SIZE	HW_SEQ_STOP	HW_SEQ_START				

Bits	Field Name	Description	Type	Reset
31:16	HW_SEQ_STEP_COUNTER	Number of steps executed by the hardware sequencer. HW_SEQ_STEP_COUNTER=0 corresponds to manual sequencing.	RW	0x0000
15:13	RESERVED	Read returns 0.	R	0x0
12:11	STEP	This register is automatically updated by the hardware sequencer when it is active. Otherwise, software could use it to activate the content of a given set of step registers (SIMCOP_HWSEQ_STEP_i).	RW	0x0
10	CPU_PROC_DONE	Used by the CPU to tell that it has completed data processing. This feature should be used together with the CPU_PROC_START_IRQ event Read's always return 0. Write 0x0: No effect. Write 0x1: CPU processing completed.	W	0
9:8	BBM_SYNC_CHAN	Defines the SIMCOP DMA hardware synchronization channel to be used for BBM. This register is only used when BITSTREAM=ENCODE or DECODE. Software must ensure that the same DMA hardware synchronization channel is not used by the hardware sequencer.	RW	0x0
7	BBM_STATUS	Status of the Bitstream buffer management hardware. Used only during automatic mode [BITSTREAM=5 or 6]. Equals 0 (IDLE) in manual mode [BITSTREAM=0..4]. Set when automatic mode is entered. Automatic encode mode: used to detect when all banks have been flushed after the processing has completed (i.e. but request bank signals have been de-asserted by BBM). Automatic decode mode (BITSTREAM=DECODE): returns to 0 (IDLE) when automatic mode is left (BITSTREAM=COPR). Read 0x1: BBM is busy. Read 0x0: BBM is idle	R	0

Bits	Field Name	Description	Type	Reset
6:4	BITSTREAM	<p>Bitstream buffer access control</p> <p>0x0: Bank 0: coprocessor bus (0x1000-0x17FF) Bank 1: coprocessor bus (0x1800-0x1FFF)</p> <p>0x1: Bank 0: DMA (0x1000-0x17FF) Bank 1: DMA (0x1800-0x1FFF)</p> <p>0x2: Bank 0: VLCDJ.B (0x000-0x7FF) Bank 1: VLCDJ.B (0x800-0xFFFF)</p> <p>0x3: Bank 0: DMA (0x1000-0x17FF) Bank 1: VLCDJ.B (0x800-0xFFFF)</p> <p>0x4: Bank 0: VLCDJ.B (0x000-0x7FF) Bank 1: DMA (0x1800-0x1FFF)</p> <p>0x5: The bitstream buffer is managed by hardware as a PING/PONG buffer to support JPEG encode use case. It could be accessed by the SIMCOP DMA or the VLCDJ module. The BITSTREAM hardware sequence is reset when the mode is changed to COPR, VLCDJ or DMA.</p> <p>0x6: The bitstream buffer is managed by hardware as a PING/PONG buffer to support JPEG decode use case. It could be accessed by the SIMCOP DMA or the VLCDJ module. The BITSTREAM hardware sequence is reset when the mode is changed to COPR, VLCDJ or DMA.</p>	RW	0x0
3:2	BITSTR_XFER_SIZE	<p>Defines the amount of data to be transferred per hardware request to the SIMCOP DMA. Bigger sizes lead to better SDRAM efficiency but prevents fine grained DMA transfer arbitration. This register is only used by hardware when BITSTREAM=ENCODE or BITSTREAM=DECODE.</p> <p>0x0: 2048 bytes</p> <p>0x1: 1024 bytes</p> <p>0x2: 512 bytes</p> <p>0x3: 256 bytes</p>	RW	0x0
1	HW_SEQ_STOP	<p>Stop the hardware sequencer. This feature is typically used to recover from an error condition. Read's always return 0.</p> <p>Write 0x0: No effect.</p> <p>Write 0x1: Stop the hardware sequence immediately (don't wait for expected DONE events). Setting this bit while the sequencer is idle has no effect.</p>	W	0
0	HW_SEQ_START	<p>Start the hardware sequencer. Read's always return 0.</p> <p>Write 0x0: No effect.</p> <p>Write 0x1: Start STEP0 of the hardware sequence. Setting this bit while the sequencer is running has no effect.</p>	W	0

**Table 8-1281. Register Call Summary for Register SIMCOP\_HWSEQ\_CTRL**

ISS SIMCOP Hardware Sequencer and Buffers Functional Description

- [Overview: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Automatic BITSTREAM Buffer Management: \[5\] \[6\]](#)
- [Automatic Operation: \[7\] \[8\] \[9\]](#)

ISS SIMCOP Hardware Sequencer and Buffers Basic Programming Model

- [External CPU Use for Data Processing: \[10\]](#)

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[11\]](#)

**Table 8-1282. SIMCOP\_HWSEQ\_STATUS**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	HWSEQ
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .		
<b>Description</b>	Hardware sequencer status register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HW_SEQ_STEP_COUNTER												RESERVED												STATE							

Bits	Field Name	Description	Type	Reset
31:16	HW_SEQ_STEP_COUNTER	Current step number	R	0x0000
15:1	RESERVED	Read returns 0.	R	0x0000
0	STATE	Current state Read 0x0: Idle Read 0x1: Running	R	0

**Table 8-1283. Register Call Summary for Register SIMCOP\_HWSEQ\_STATUS**

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[0\]](#)

**Table 8-1284. SIMCOP\_HWSEQ\_OVERRIDE**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	HWSEQ
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .		
<b>Description</b>	Hardware sequencer override control register. Bits in this register select what configuration register control a resource. 0: Resource controlled by hardware sequencer. Hardware uses the value from SIMCOP_HWSEQ_STEP_xx registers for the chosen resource 1: Resource controlled by software. Hardware uses the value from SIMCOP_HWSEQ_STEP_x_OVERRIDE registers for the chosen resource The bit field name matches the one of the resource. For example, IMX_A_D_OFST_OVR selects if <a href="#">SIMCOP_HWSEQ_STEP_CTRL_[12:11]</a> IMX_A_D_OFST or <a href="#">SIMCOP_HWSEQ_STEP_CTRL_OVERRIDE[12:11]</a> IMX_A_D_OFST controls how image buffers are arranged in the IMX A address map.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												COEFF_B	COEFF_A	IMBUFF_H	IMBUFF_G	IMBUFF_F	IMBUFF_E	IMBUFF_D	IMBUFF_C	IMBUFF_B	IMBUFF_A	LDC_O_OFST_OVR	ROT_O_OFST_OVR	ROT_I_OFST_OVR	NSF_IO_OFST_OVR	DCT_F_OFST_OVR	DCT_S_OFST_OVR	VLCDJ_IO_OFST_OVR	IMX_B_D_OFST_OVR	IMX_A_D_OFST_OVR	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Read returns 0.	R	0x0000
18	COEFF_B	See register description	RW	0
17	COEFF_A	See register description	RW	0
16	IMBUFF_H	See register description	RW	0
15	IMBUFF_G	See register description	RW	0
14	IMBUFF_F	See register description	RW	0

Bits	Field Name	Description	Type	Reset
13	IMBUFF_E	See register description	RW	0
12	IMBUFF_D	See register description	RW	0
11	IMBUFF_C	See register description	RW	0
10	IMBUFF_B	See register description	RW	0
9	IMBUFF_A	See register description	RW	0
8	LDC_O_OFST_OVR	See register description	RW	0
7	ROT_O_OFST_OVR	See register description	RW	0
6	ROT_I_OFST_OVR	See register description	RW	0
5	NSF_IO_OFST_OVR	See register description	RW	0
4	DCT_F_OFST_OVR	See register description	RW	0
3	DCT_S_OFST_OVR	See register description	RW	0
2	VLCDJ_IO_OFST_OVR	See register description	RW	0
1	IMX_B_D_OFST_OVR	See register description	RW	0
0	IMX_A_D_OFST_OVR	See register description	RW	0

**Table 8-1285. Register Call Summary for Register SIMCOP\_HWSEQ\_OVERRIDE**

ISS SIMCOP Hardware Sequencer and Buffers Functional Description

- [Hardware Sequencer Override: \[0\]](#)

ISS SIMCOP Hardware Sequencer and Buffers Basic Programming Model

- [ISS SIMCOP Application Programming Principle: \[1\]](#)
- [Concurrent SW and HW sequencing: \[2\] \[3\] \[4\] \[5\] \[6\]](#)

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[7\]](#)

**Table 8-1286. SIMCOP\_HWSEQ\_STEP\_CTRL\_OVERRIDE**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	HWSEQ
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .		
<b>Description</b>	Hardware sequencer override register. Used to execute software sequences in parallel to hardware sequencing steps		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED								ROT_O_OFST	ROT_I_OFST	RESERVED	DCT_F_OFST	DCT_S_OFST	VLCDJ_IO_OFST	IMX_B_D_OFST	IMX_A_D_OFST	RESERVED	DMA_TRIGGER	ROT_A_TRIGGER	NSF_TRIGGER	VLCDJ_TRIGGER	DCT_TRIGGER	LDC_TRIGGER											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:26	ROT_O_OFST	Controls ROT.O bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x0: EFGH 0x1: FGHE 0x2: GHEF 0x3: HEFG	RW	0x0

Bits	Field Name	Description	Type	Reset
25:24	ROT_I_OFST	Controls ROT_I bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000  0x0: ABCD 0x1: BCDA 0x2: CDAB 0x3: DABC	RW	0x0
23	RESERVED	Read returns 0.	R	0
22:20	DCT_F_OFST	Controls DCT.F bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000  0x0: ABCD 0x1: BCDG 0x2: CDGH 0x3: DGHA 0x4: GHAB 0x5: HABC	RW	0x0
19:18	DCT_S_OFST	Controls DCT.S bus mapping to image buffers: 0x0000 0x1000  0x0: EF 0x1: FG 0x2: GH 0x3: HE	RW	0x0
17:15	VLCDJ_IO_OFST	Controls VLCDJ IO data bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000  0x0: ABCD 0x1: BCDG 0x2: CDGH 0x3: DGHA 0x4: GHAB 0x5: HABC	RW	0x0
14:13	IMX_B_D_OFST	Controls iMX B data bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000  0x0: ABCD 0x1: CDEF 0x2: EFGH 0x3: GHAB	RW	0x0
12:11	IMX_A_D_OFST	Controls iMX A data bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000  0x0: ABCD 0x1: CDEF 0x2: EFGH 0x3: GHAB	RW	0x0
10:8	RESERVED	Read returns 0.	R	0x0



Bits	Field Name	Description	Type	Reset
7:5	DMA_TRIGGER	<p>Software controlled START/DONE synchronization</p> <p>Write 0x0: No effect</p> <p>Read 0x0: No done pulse have been received since last non zero write into the DMA_TRIGGER register</p> <p>Read 0x1: DONE pulses for channel 0 and 1 have been received</p> <p>Write 0x1: Trigger channel 0 and 1. Clears all memorized done pulses for DMA.</p> <p>Write 0x2: Trigger channel 0, 1, 2. Clears all memorized done pulses for DMA.</p> <p>Read 0x2: DONE pulses for channel 0, 1 and 2 have been received</p> <p>Read 0x3: DONE pulses for channel 0, 1, 2 and 3 have been received.</p> <p>Write 0x3: Trigger channel 0, 1, 2 and 3. Clears all memorized done pulses for DMA.</p> <p>Read 0x4: DONE pulse for channel 0 has been received</p> <p>Write 0x4: Trigger channel 0. Clears all memorized done pulses for DMA.</p> <p>Write 0x5: Trigger channel 1. Clears all memorized done pulses for DMA.</p> <p>Read 0x5: DONE pulse for channel 1 has been received</p> <p>Read 0x6: DONE pulse for channel 2 has been received</p> <p>Write 0x6: Trigger channel 2. Clears all memorized done pulses for DMA.</p> <p>Write 0x7: Trigger channel 3. Clears all memorized done pulses for DMA.</p> <p>Read 0x7: DONE pulse for channel 3 has been received</p>	RW	0x0
4	ROT_A_TRIGGER	<p>Software controlled START/DONE synchronization</p> <p>Write 0x0: No Effect</p> <p>Read 0x0: No DONE pulse received since the last START pulse has been sent</p> <p>Read 0x1: DONE pulse received</p> <p>Write 0x1: Send a start pulse and clears the memorized done pulse</p>	RW	0
3	NSF_TRIGGER	<p>Software controlled START/DONE synchronization</p> <p>Write 0x0: No Effect</p> <p>Read 0x0: No DONE pulse received since the last START pulse has been sent</p> <p>Read 0x1: DONE pulse received</p> <p>Write 0x1: Send a start pulse and clears the memorized done pulse</p>	RW	0
2	VLCDJ_TRIGGER	<p>Software controlled START/DONE synchronization</p> <p>Write 0x0: No Effect</p> <p>Read 0x0: No DONE pulse received since the last START pulse has been sent</p> <p>Read 0x1: DONE pulse received</p> <p>Write 0x1: Send a start pulse and clears the memorized done pulse</p>	RW	0

Bits	Field Name	Description	Type	Reset
1	DCT_TRIGGER	Software controlled START/DONE synchronization Write 0x0: No Effect Read 0x0: No DONE pulse received since the last START pulse has been sent Read 0x1: DONE pulse received Write 0x1: Send a start pulse and clears the memorized done pulse	RW	0
0	LDC_TRIGGER	Software controlled START/DONE synchronization Write 0x0: No Effect Read 0x0: No DONE pulse received since the last START pulse has been sent Read 0x1: DONE pulse received Write 0x1: Send a start pulse and clears the memorized done pulse	RW	0

**Table 8-1287. Register Call Summary for Register SIMCOP\_HWSEQ\_STEP\_CTRL\_OVERRIDE**

ISS SIMCOP Hardware Sequencer and Buffers Functional Description

- [Hardware Sequencer Override: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)

ISS SIMCOP Hardware Sequencer and Buffers Basic Programming Model

- [Concurrent SW and HW sequencing: \[6\]](#)

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[7\]](#)
- [SIMCOP Register Description: \[8\]](#)

**Table 8-1288. SIMCOP\_HWSEQ\_STEP\_SWITCH\_OVERRIDE**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	HWSEQ
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .		
<b>Description</b>	Hardware sequencer override register. Used to execute software sequences in parallel to hardware sequencing steps		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMBUFF_H				IMBUFF_G				RESERVED	IMBUFF_F			RESERVED	IMBUFF_E			RESERVED	IMBUFF_D			RESERVED	IMBUFF_C			RESERVED	IMBUFF_B			RESERVED	IMBUFF_A		

Bits	Field Name	Description	Type	Reset
31:28	IMBUFF_H	Switch for image buffer h 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_S 0x6: DCT_F 0x7: ROT_A_O 0x8: NSF_IO 0x9: LDC_O	RW	0x0
27:24	IMBUFF_G	Switch for image buffer g	RW	0x0

Bits	Field Name	Description	Type	Reset
		0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_S 0x6: DCT_F 0x7: ROT_A_O 0x8: NSF_IO 0x9: LDC_O		
23	RESERVED	Read returns 0.	R	0
22:20	IMBUFF_F	Switch for image buffer f 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: DCT_S 0x5: NSF_IO 0x6: LDC_O 0x7: ROT_A_O	RW	0x0
19	RESERVED	Read returns 0.	R	0
18:16	IMBUFF_E	Switch for image buffer e 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: DCT_S 0x5: NSF_IO 0x6: LDC_O 0x7: ROT_A_O	RW	0x0
15	RESERVED	Read returns 0.	R	0
14:12	IMBUFF_D	Switch for image buffer d 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT_A_I 0x7: Reserved	RW	0x0
11	RESERVED	Read returns 0.	R	0
10:8	IMBUFF_C	Switch for image buffer c. 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT_A_I	RW	0x0

Bits	Field Name	Description	Type	Reset
		0x7: Reserved		
7	RESERVED	Read returns 0.	R	0
6:4	IMBUFF_B	Switch for image buffer b. 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A IMBUFF 0x3: iMX B IMBUFF 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT_A_I 0x7: Reserved	RW	0x0
3	RESERVED	Read returns 0.	R	0
2:0	IMBUFF_A	Switch for image buffer a 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A IMBUFF 0x3: iMX B IMBUFF 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT_A_I 0x7: Reserved	RW	0x0

**Table 8-1289. Register Call Summary for Register SIMCOP\_HWSEQ\_STEP\_SWITCH\_OVERRIDE**

ISS SIMCOP Hardware Sequencer and Buffers Functional Description

- [Hardware Sequencer Override: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

ISS SIMCOP Hardware Sequencer and Buffers Basic Programming Model

- [ISS SIMCOP Application Programming Principle: \[7\]](#)
- [Concurrent SW and HW sequencing: \[8\] \[9\] \[10\]](#)

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[11\]](#)

**Table 8-1290. SIMCOP\_HWSEQ\_STEP\_CTRL2\_OVERRIDE**

<b>Address Offset</b>	0x0000 007C															
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .	<b>Instance</b> HWSEQ														
<b>Description</b>	Hardware sequencer override register. Used to execute software sequences in parallel to hardware sequencing steps															
<b>Type</b>	RW															
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0														
RESERVED											NSF2_IO_OFST	LDC_O_OFST	RESERVED	COEFF_B	RESERVED	COEFF_A

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Read returns 0.	R	0x00000
11:10	NSF2_IO_OFST	Controls NSF_IO bus mapping to image buffers: 0x0000 0x1000 0x0: EF 0x1: FG 0x2: GH 0x3: HE	RW	0x0
9:8	LDC_O_OFST	Controls LDC.O bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x0: EFGH 0x1: FGHE 0x2: GHEF 0x3: HEFG	RW	0x0
7	RESERVED	Read returns 0.	R	0
6:4	COEFF_B	Coefficient buffer b switch 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT A O 0x7: Reserved	RW	0x0
3	RESERVED	Read returns 0.	R	0
2:0	COEFF_A	Coefficient buffer a switch 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT A I 0x7: Reserved	RW	0x0

**Table 8-1291. Register Call Summary for Register SIMCOP\_HWSEQ\_STEP\_CTRL2\_OVERRIDE**

ISS SIMCOP Hardware Sequencer and Buffers Basic Programming Model

- [Concurrent SW and HW sequencing: \[0\] \[1\]](#)

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[2\]](#)

**Table 8-1292. SIMCOP\_HWSEQ\_STEP\_CTRL\_i**

<b>Address Offset</b>	0x0000 0080 + (0x10 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .	<b>Instance</b>	HWSEQ
<b>Description</b>	Hardware sequencer step control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU_SYNC	DMA_OFST		ROT_O_OFST	ROT_I_OFST	RESERVED	DCT_F_OFST		DCT_S_OFST		VLCDJ_IO_OFST		IMX_B_D_OFST		IMX_A_D_OFST		NEXT		RESERVED	DMA_SYNC		ROT_A_SYNC	NSF_SYNC	VLCDJ_SYNC	DCT_SYNC	LDC_SYNC						

Bits	Field Name	Description	Type	Reset
31	CPU_SYNC	Enable hardware synchronization with the CPU so that it could be used for some processing on in the macroblock pipeline. 0x0: Disabled 0x1: Enabled.	RW	0
30:28	DMA_OFST	Controls DMA bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x4000 0x5000 0x6000 0x7000 0x0: ABCDEFGH 0x1: BCDEFGHA 0x2: CDEFGHAB 0x3: DEFGHABC 0x4: EFGHABCD 0x5: FGHABCDE 0x6: GHABCDEF 0x7: HABCDEFG	RW	0x0
27:26	ROT_O_OFST	Controls ROT.O bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x0: EFGH 0x1: FGHE 0x2: GHEF 0x3: HEFG	RW	0x0
25:24	ROT_I_OFST	Controls ROT.I bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x0: ABCD 0x1: BCDA 0x2: CDAB 0x3: DABC	RW	0x0
23	RESERVED	Read returns 0.	R	0
22:20	DCT_F_OFST	Controls DCT.F bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x0: ABCD 0x1: BCDG 0x2: CDGH 0x3: DGHA 0x4: GHAB 0x5: HABC	RW	0x0

Bits	Field Name	Description	Type	Reset
19:18	DCT_S_OFST	Controls DCT.S bus mapping to image buffers: 0x0000 0x1000 0x0: EF 0x1: FG 0x2: GH 0x3: HE	RW	0x0
17:15	VLCDJ_IO_OFST	Controls VLCDJ IO data bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x0: ABCD 0x1: BCDG 0x2: CDGH 0x3: DGHA 0x4: GHAB 0x5: HABC	RW	0x0
14:13	IMX_B_D_OFST	Controls iMX B data bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x0: ABCD 0x1: CDEF 0x2: EFGH 0x3: GHAB	RW	0x0
12:11	IMX_A_D_OFST	Controls iMX A data bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x0: ABCD 0x1: CDEF 0x2: EFGH 0x3: GHAB	RW	0x0
10:9	NEXT	Next channel in the sync chain 0x0: Step 0 0x1: Step 1 0x2: Step 2 0x3: Step 3	RW	0x0
8	RESERVED	Read returns 0.	R	0
7:5	DMA_SYNC	Enable hardware synchronization with the SIMCOP DMA 0x0: Disabled 0x1: Channel 0 and 1 0x2: Channel 0, 1, 2 0x3: Channel 0, 1, 2 and 3 0x4: Channel 0 0x5: Channel 1 0x6: Channel 2 0x7: Channel 3	RW	0x0
4	ROT_A_SYNC	Enable hardware synchronization with the ROT #a module 0x0: Disabled 0x1: Enabled.	RW	0
3	NSF_SYNC	Enable hardware synchronization with the NSF module 0x0: Disabled 0x1: Enabled	RW	0
2	VLCDJ_SYNC	Enable hardware synchronization with the VLCDJ module 0x0: Disabled 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
1	DCT_SYNC	Enable hardware synchronization with the DCT module 0x0: Disabled 0x1: Enabled	RW	0
0	LDC_SYNC	Enable hardware synchronization with the LDC module 0x0: Disabled 0x1: Enabled	RW	0

**Table 8-1293. Register Call Summary for Register SIMCOP\_HWSEQ\_STEP\_CTRL\_i**

ISS SIMCOP Hardware Sequencer and Buffers Functional Description

- [Image Buffers: \[0\]](#)
- [Automatic Operation: \[1\] \[2\]](#)

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[3\]](#)
- [SIMCOP Register Description: \[4\]](#)

**Table 8-1294. SIMCOP\_HWSEQ\_STEP\_SWITCH\_i**

<b>Address Offset</b>	0x0000 0084 + (0x10 * i)	<b>Instance</b>	HWSEQ
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .		
<b>Description</b>	Image buffer switch control. The configuration of step #0 is used when hardware sequencer is idle.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
IMBUFF_H				IMBUFF_G				RESERVED	IMBUFF_F				RESERVED	IMBUFF_E				RESERVED	IMBUFF_D				RESERVED	IMBUFF_C				RESERVED	IMBUFF_B				RESERVED	IMBUFF_A			

Bits	Field Name	Description	Type	Reset
31:28	IMBUFF_H	Switch for image buffer h 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_S 0x6: DCT_F 0x7: ROT_A_O 0x8: NSF_IO 0x9: LDC_O	RW	0x0
27:24	IMBUFF_G	Switch for image buffer g 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_S 0x6: DCT_F 0x7: ROT_A_O 0x8: NSF_IO	RW	0x0



Bits	Field Name	Description	Type	Reset
		0x9: LDC_O		
23	RESERVED	Read returns 0.	R	0
22:20	IMBUFF_F	Switch for image buffer f 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: DCT_S 0x5: NSF_IO 0x6: LDC_O 0x7: ROT_A_O	RW	0x0
19	RESERVED	Read returns 0.	R	0
18:16	IMBUFF_E	Switch for image buffer e 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: DCT_S 0x5: NSF_IO 0x6: LDC_O 0x7: ROT_A_O	RW	0x0
15	RESERVED	Read returns 0.	R	0
14:12	IMBUFF_D	Switch for image buffer d 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT_A_I 0x7: Reserved	RW	0x0
11	RESERVED	Read returns 0.	R	0
10:8	IMBUFF_C	Switch for image buffer c. 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT_A_I 0x7: Reserved	RW	0x0
7	RESERVED	Read returns 0.	R	0
6:4	IMBUFF_B	Switch for image buffer b. 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A IMBUFF 0x3: iMX B IMBUFF 0x4: VLCDJ_IO 0x5: DCT_F	RW	0x0

Bits	Field Name	Description	Type	Reset
		0x6: ROT_A_I		
		0x7: Reserved		
3	RESERVED	Read returns 0.	R	0
2:0	IMBUFF_A	Switch for image buffer a	RW	0x0
		0x0: Coprocessor bus		
		0x1: SIMCOP DMA		
		0x2: iMX A IMBUFF		
		0x3: iMX B IMBUFF		
		0x4: VLCDJ_IO		
		0x5: DCT_F		
		0x6: ROT_A_I		
		0x7: Reserved		

**Table 8-1295. Register Call Summary for Register SIMCOP\_HWSEQ\_STEP\_SWITCH\_i**

ISS SIMCOP Hardware Sequencer and Buffers Functional Description

- [Static Crossbar: \[0\]](#)
- [Image Buffers: \[1\]](#)

ISS SIMCOP Hardware Sequencer and Buffers Basic Programming Model

- [Rotation Operational Mode Configuration: \[2\] \[3\] \[4\] \[5\]](#)
- [LDC Operational Mode Configuration: \[6\] \[7\] \[8\]](#)

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[9\]](#)

**Table 8-1296. SIMCOP\_HWSEQ\_STEP\_IMX\_CTRL\_i**

<b>Address Offset</b>	0x0000 0088 + (0x10 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .	<b>Instance</b>	HWSEQ
<b>Description</b>	Hardware sequencer step control register The configuration of step #0 is used when hardware sequencer is idle.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
IMX_B_SYNC	RESERVED															IMX_A_SYNC	RESERVED															

Bits	Field Name	Description	Type	Reset
31	IMX_B_SYNC	Enable hardware synchronization with the iMX B module 0x0: Disabled 0x1: Enabled.	RW	0
30:29	RESERVED	Read returns 0.	R	0x0
28:16	IMX_B_START	This register is only used when <a href="#">SIMCOP_HWSEQ_STEP_IMX_CTRL_i</a> [31] IMX_B_SYNC=1. It contains the address, in 16-bit words, of the first instruction iMX will execute when it gets started.	RW	0x0000
15	IMX_A_SYNC	Enable hardware synchronization with the iMX A module 0x0: Disabled 0x1: Enabled.	RW	0
14:13	RESERVED	Read returns 0.	R	0x0

Bits	Field Name	Description	Type	Reset
12:0	IMX_A_START	This register is only used when <a href="#">SIMCOP_HWSEQ_STEP_IMX_CTRL_i</a> [15] IMX_A_SYNC=1. It contains the address, in 16-bit words, of the first instruction iMX will execute when it gets started.	RW	0x0000

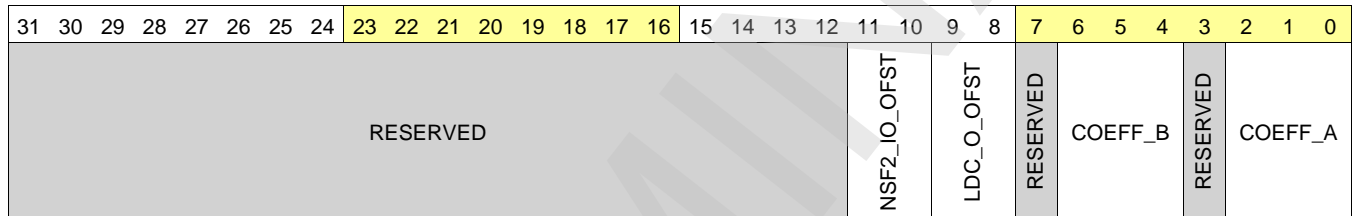
**Table 8-1297. Register Call Summary for Register SIMCOP\_HWSEQ\_STEP\_IMX\_CTRL\_i**

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[0\]](#)
- [SIMCOP Register Description: \[1\] \[2\]](#)

**Table 8-1298. SIMCOP\_HWSEQ\_STEP\_CTRL2\_i**

<b>Address Offset</b>	0x0000 008C + (0x10 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1279</a> .	<b>Instance</b>	HWSEQ
<b>Description</b>	Hardware sequencer step control register		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Read returns 0.	R	0x00000
11:10	NSF2_IO_OFST	Controls NSF_IO bus mapping to image buffers: 0x0000 0x1000 0x0: EF 0x1: FG 0x2: GH 0x3: HE	RW	0x0
9:8	LDC_O_OFST	Controls LDC.O bus mapping to image buffers: 0x0000 0x1000 0x2000 0x3000 0x0: EFGH 0x1: FGHE 0x2: GHEF 0x3: HEFG	RW	0x0
7	RESERVED	Read returns 0.	R	0
6:4	COEFF_B	Coefficient buffer b switch 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT A O 0x7: Reserved	RW	0x0
3	RESERVED	Read returns 0.	R	0

Bits	Field Name	Description	Type	Reset
2:0	COEFF_A	Coefficient buffer a switch 0x0: Coprocessor bus 0x1: SIMCOP DMA 0x2: iMX A 0x3: iMX B 0x4: VLCDJ_IO 0x5: DCT_F 0x6: ROT A I 0x7: Reserved	RW	0x0

**Table 8-1299. Register Call Summary for Register SIMCOP\_HWSEQ\_STEP\_CTRL2\_i**

ISS SIMCOP Hardware Sequencer and Buffers Functional Description

- [Static Crossbar: \[0\]](#)

ISS SIMCOP Hardware Sequencer and Buffers Basic Programming Model

- [Rotation Operational Mode Configuration: \[1\] \[2\] \[3\] \[4\]](#)
- [LDC Operational Mode Configuration: \[5\] \[6\] \[7\] \[8\]](#)
- [Concurrent SW and HW sequencing: \[9\]](#)

ISS SIMCOP Hardware Sequencer and Buffer Registers Manual

- [SIMCOP Register Summary: \[10\]](#)

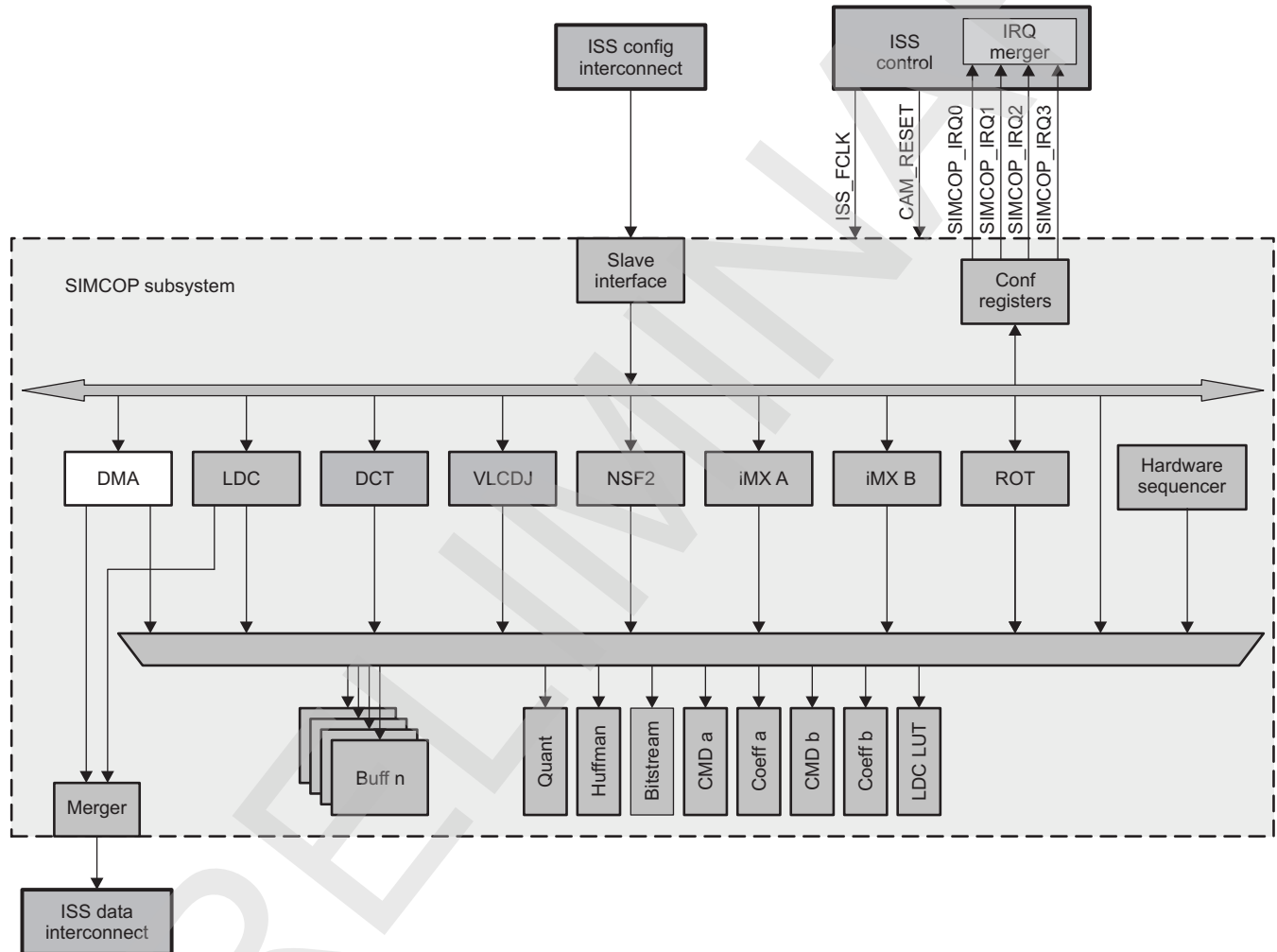
### 8.4.3 ISS SIMCOP DMA Module

This section describes the direct memory access (DMA) module in the still image coprocessor (SIMCOP) subsystem.

#### 8.4.3.1 ISS SIMCOP DMA Overview

The direct memory access (DMA) module transfers data from system memory to the still image coprocessor (SIMCOP) memories and from SIMCOP memories. Figure 8-254 is an overview of the DMA module in the SIMCOP subsystem.

Figure 8-254. DMA in the SIMCOP Subsystem



scpdma-001

The SIMCOP DMA supports the following features:

- Configuration interface, for register access
- Imaging subsystem (ISS) data interconnect master interface, for system memory access
  - Supports 1-dimensional (1D) and 2-dimensional (2D) burst
- SIMCOP crossbar master interface, for SIMCOP memories access
- Eight logical channels supported
- Hardware synchronization support
- 4-dimensional (4D) addressing modes
- 2D data block transfer (base address, stride, height, width)
- Array of 2D block transfer support

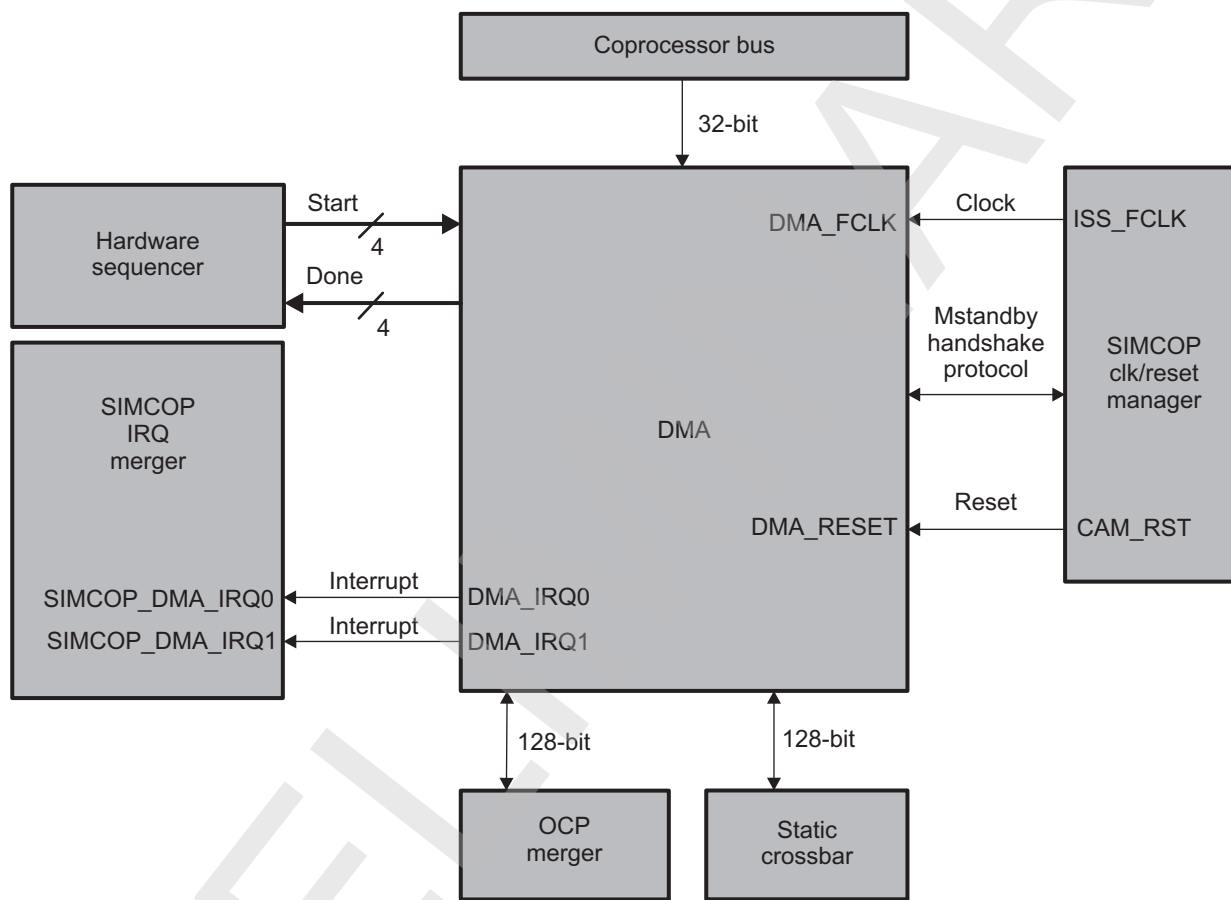
- Performance: Software-controllable bandwidth limiter

### 8.4.3.2 ISS SIMCOP DMA Integration

The DMA module is part of the SIMCOP subsystem in the ISS. Figure 8-255 shows the integration of the DMA in the SIMCOP subsystem.

Table 8-1300, Table 8-1301, and Table 8-1302 list the SIMCOP DMA integration attributes, clocks and resets, and hardware requests, respectively.

Figure 8-255. DMA Engine Integration



scpdma-004

Table 8-1300. Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
DMA	PD_CAM	

Table 8-1301. Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DMA	DMA_FCLK	ISS_FCLK	PRCM	Functional clock provided by ISS_CLK from the PRCM module. It is used by all ISS submodules and ISS top-level resources.
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description

**Table 8-1301. Clocks and Resets (continued)**

DMA	DMA_RESET	CAM_RST	PRCM	ISS and SIMCOP global reset
-----	-----------	---------	------	-----------------------------

For information about clock and reset management, see [Section 8.4.1.2.1](#), *ISS SIMCOP Local Power and Clock Management*.

**Table 8-1302. Hardware Requests**

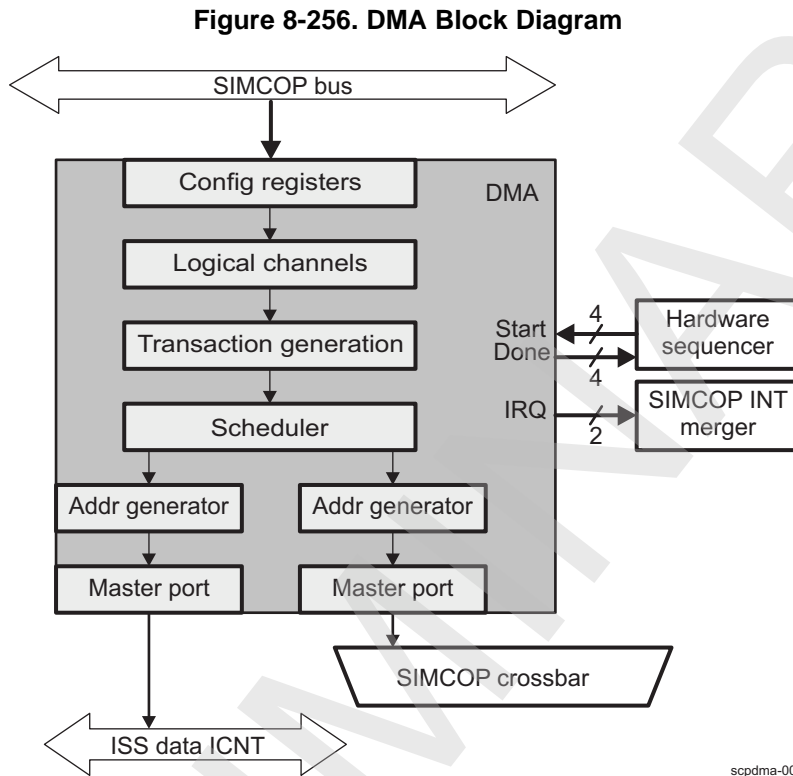
Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
DMA	DMA_IRQ0	SIMCOP_DMA_I RQ0	SIMCOP IRQ merger	Interrupt triggered by SIMCOP DMA
DMA	DMA_IRQ1	SIMCOP_DMA_I RQ1	SIMCOP IRQ merger	Interrupt triggered by SIMCOP DMA

For more information about interrupt requests, see [Section 8.4.1.2.3](#), *Interrupt Merger*.

### 8.4.3.3 ISS SIMCOP DMA Functional Description

#### 8.4.3.3.1 ISS SIMCOP DMA Block Diagram

Figure 8-256 is a block diagram of the SIMCOP DMA.



An external initiator sets up logical channels using the configuration interface.

When a logical channel becomes active, a transfer must be executed. A transfer corresponds to a 2D block of data copied from system memory to SIMCOP memory and vice versa.

A transfer is decomposed into one or multiple transactions by the transaction generation unit.

The scheduler issues transactions when tags are available. There can be one outstanding command per tag on the ISS data interconnect.

#### 8.4.3.3.2 ISS SIMCOP DMA Power Management

Table 8-1303 lists the power-management features available in the DMA module.

**Table 8-1303. Local Power-Management Features**

Feature	Registers	Description
Master standby modes	<a href="#">SIMCOP_DMA_SYSCONFIG</a> [5:4] STANDBYMODE	Force-standby, no-standby, and smart-standby modes are available.

#### 8.4.3.3.3 ISS SIMCOP DMA Interrupt Requests

The SIMCOP DMA module can generate two interrupts, SIMCOP\_DMA\_IRQ0 and SIMCOP\_DMA\_IRQ1. These two interrupts are controlled by four registers:

- [SIMCOP\\_DMA\\_IRQSTATUS\\_RAW\\_j](#): The status of the interrupt before masking
- [SIMCOP\\_DMA\\_IRQSTATUS\\_j](#): The status of the interrupt after masking
- [SIMCOP\\_DMA\\_IRQENABLE\\_SET\\_j](#): Enable an interrupt to propagate to SIMCOP\_DMA\_IRQi signal (unmask)



- [SIMCOP\\_DMA\\_IRQENABLE\\_CLR\\_j](#): Disable an interrupt to propagate to [SIMCOP\\_DMA\\_IRQi](#) signal (mask)

To clear an interrupt, software must write 1 in the corresponding bit of the [SIMCOP\\_DMA\\_IRQSTATUS\\_j](#) register even if the event is not enabled ([SIMCOP\\_DMA\\_IRQSTATUS\\_RAW\\_j](#) is also cleared).

[Table 8-1304](#) lists all the SIMCOP DMA events that can lead to an interrupt generation.

**Table 8-1304. SIMCOP DMA Events List**

Bit Name	Bit	Event
BUS_ERR	0	Error occurred on bus
CHAN0_BLOCK_DONE_IRQ	16	Channel 0 has completed transfer of one 2D block.
CHAN1_BLOCK_DONE_IRQ	17	Channel 1 has completed transfer of one 2D block.
CHAN2_BLOCK_DONE_IRQ	18	Channel 2 has completed transfer of one 2D block.
CHAN3_BLOCK_DONE_IRQ	19	Channel 3 has completed transfer of one 2D block.
CHAN4_BLOCK_DONE_IRQ	20	Channel 4 has completed transfer of one 2D block.
CHAN5_BLOCK_DONE_IRQ	21	Channel 5 has completed transfer of one 2D block.
CHAN6_BLOCK_DONE_IRQ	22	Channel 6 has completed transfer of one 2D block.
CHAN7_BLOCK_DONE_IRQ	23	Channel 7 has completed transfer of one 2D block.
CHAN0_FRAME_DONE_IRQ	24	Channel 0 has completed transfer of the full frame.
CHAN1_FRAME_DONE_IRQ	25	Channel 1 has completed transfer of the full frame.
CHAN2_FRAME_DONE_IRQ	26	Channel 2 has completed transfer of the full frame.
CHAN3_FRAME_DONE_IRQ	27	Channel 3 has completed transfer of the full frame.
CHAN4_FRAME_DONE_IRQ	28	Channel 4 has completed transfer of the full frame.
CHAN5_FRAME_DONE_IRQ	29	Channel 5 has completed transfer of the full frame.
CHAN6_FRAME_DONE_IRQ	30	Channel 6 has completed transfer of the full frame.
CHAN7_FRAME_DONE_IRQ	31	Channel 7 has completed transfer of the full frame.

#### 8.4.3.3.4 ISS SIMCOP DMA Logical Channels

In the context of SIMCOP, multiple logical channels are required to offload the external initiator. Typical uses of multiple context are:

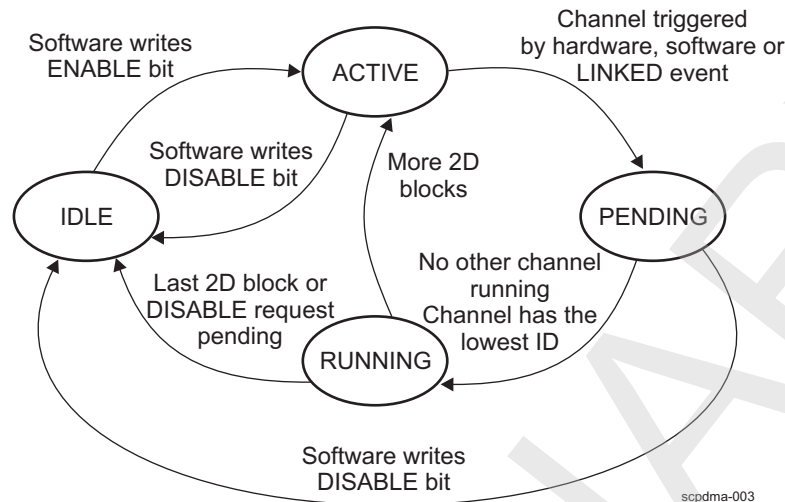
- Run a read transfer followed by a write transfer. That is the required mode to operate the NSF2 module. When the NSF2 completes processing, filtering outcome is first copied from the image buffers to system memory. When this transfer completes, new input data is fetched from system memory and stored in the image buffers.
- Read or write YUV4:2:0 data from system memory; YUV4:2:0 data is stored in two different system memory buffers.

##### 8.4.3.3.4.1 Logical Channel States

The SIMCOP DMA supports up to eight (([SIMCOP\\_DMA\\_HWINFO\[2\]](#) CHAN+1)\*4) logical channels. Each logical channel can be configured and triggered separately.

Idle logical channels must be configured and enabled ([SIMCOP\\_DMA\\_CHAN\\_CTRL\\_i\[0\]](#) ENABLE = 1) to be used.

[Figure 8-257](#) shows the different states of a logical channel. Software can poll the current state from the [SIMCOP\\_DMA\\_CHAN\\_CTRL\\_i\[4:3\]](#) STATUS bit field.

**Figure 8-257. DMA Logical Channel States**

When a trigger event is received by a channel in the ACTIVE state, it moves to the PENDING state. Trigger events received when the channel is in the IDLE, PENDING, or RUNNING state are discarded.

A logical channel is automatically disabled by hardware when a complete frame (all 2D blocks) is transferred. It can be disabled earlier by software by writing the `SIMCOP_DMA_CHAN_CTRL_i[1]` DISABLE bit. When a disable request occurs in the RUNNING state, it is memorized and executed when the channel leaves the RUNNING state. Pending disable requests have higher priority than channel trigger events.

When multiple logical channels are pending simultaneously, the following rules apply:

- An ongoing transfer of a 2D block is never stopped. It must complete before the next one starts.
- The logical channel with the lowest ID wins the arbitration (that is, CHAN0 has higher priority than CHAN1). Software can give higher priority to a channel by choosing a lower ID.

This situation can, for example, occur when software triggers a channel while a transfer is ongoing for another channel. Two types of interrupts can be generated:

- `CHANx_BLOCK_DONE_IRQ` (where  $x$  = channel number) is generated when the channel leaves the RUNNING state (RUNNING  $\rightarrow$  ACTIVE or RUNNING  $\rightarrow$  IDLE).
- `CHANx_FRAME_DONE_IRQ` (where  $x$  = channel number) is generated when the channel enters the IDLE state.

#### 8.4.3.3.4.2 Logical Channel Chaining, Trigger, and Hardware Synchronization

Active logical channels can be triggered:

- By software. Writes the `SIMCOP_DMA_CHAN_CTRL_i[2]` SWTRIGGER bit
- By a hardware DMA request. A channel can be made sensitive to a pulse on the `START[y]` input when `SIMCOP_DMA_CHAN_CTRL_i[19:17]` `HWSTART = y` has been configured by software.
- When the previous channel in the linked list finishes. This feature is enabled through the `SIMCOP_DMA_CHAN_CTRL_i[16:12]` LINKED bit field of the preceding channel in the chain.

When a channel completes, it can optionally send a pulse to the `DONE[y]` output. Software enables this feature by configuring `SIMCOP_DMA_CHAN_CTRL_i[22:20]` `HWSTOP = y`.

Typically, the START + DONE synchronization mechanism is used together with the SIMCOP hardware sequencer. The hardware sequencer sends a pulse to `START[y]` a data transfer and the SIMCOP DMA responds with a `DONE[y]` pulse when the transfer completes. A transfer can be composed of one or multiple logical channels. For example, a YUV4:2:0-NV12 block copy from system memory to SIMCOP local memories requires two chained logical channels.

**CAUTION**

Software must ensure that only one trigger event source is selected at a given time for a logical channel; otherwise, unpredictable behavior may occur.

Typically, the following combinations can be used:

- Single channel triggered by software: Hardware synchronization and linking are disabled. Software writes the `SIMCOP_DMA_CHAN_CTRL_i[2]` SWTRIGGER bit to start the transfer of one 2D block. Software must not trigger the channel again before the `CHANx_BLOCK_DONE_IRQ` (where x = channel number) event is generated by the DMA.
- Channels chain-triggered by software: Multiple channels are linked together using the `SIMCOP_DMA_CHAN_CTRL_i[16:12]` LINKED bit field. Software triggers the first one in the chain and monitors the `CHANx_BLOCK_DONE_IRQ` (where x = channel number) event to know when the chain can be triggered again.
- Single channel triggered by hardware synchronization: Sensitivity on START pulses is activated through the `SIMCOP_DMA_CHAN_CTRL_i[19:17]` HWSTART bit field. DONE pulses are activated using the `SIMCOP_DMA_CHAN_CTRL_i[22:20]` HWSTOP bit field.
- Channels chain-triggered by hardware: Multiple channels are linked together using the `SIMCOP_DMA_CHAN_CTRL_i[16:12]` LINKED bit field. Sensitivity on START pulses is activated for the first channel in the chain through the `SIMCOP_DMA_CHAN_CTRL_i[19:17]` HWSTART bit field. DONE pulses for the last channel in the chain are activated using the `SIMCOP_DMA_CHAN_CTRL_i[22:20]` HWSTOP bit field.

**8.4.3.3.4.3 Logical Channel Data Transfer**

Each logical channel can handle 4D transfers from system memory to SIMCOP memory and vice versa.

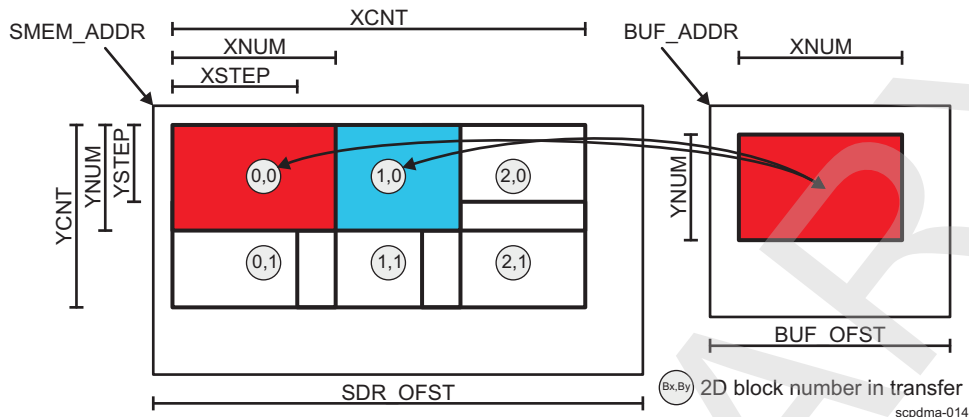
A frame is composed of `SIMCOP_DMA_CHAN_FRAME_i[9:0]` XCNT \* `SIMCOP_DMA_CHAN_FRAME_i[25:16]` YCNT 2D blocks of data. One 2D block of data is transferred each time a logical channel is in the RUNNING state.

The size of a 2D block is `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[13:4]` XNUM \* `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[28:16]` YNUM words of 128 bits. The SIMCOP DMA can transfer only full 128-bit words.

2D blocks may overlap in system memory. This feature is typically used for source data fetches from system memory to handle filter dependencies. If this feature is not used, software must set:

- `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[13:4]` XNUM = `SIMCOP_DMA_CHAN_BLOCK_STEP_i[14:4]` XSTEP
- `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[28:16]` YNUM = `SIMCOP_DMA_CHAN_BLOCK_STEP_i[29:16]` YSTEP

Figure 8-258 shows DMA addressing.

**Figure 8-258. DMA Addressing**

The base position of 2D blocks in system memory is automatically computed by the SIMCOP DMA. The base address is computed using the following formula:

$$\begin{aligned} \text{SMEM\_BASE} = & \text{SIMCOP\_DMA\_CHAN\_SMEM\_ADDR\_i}[31:4] \text{ ADDR} + \\ & \text{Bx} * \text{SIMCOP\_DMA\_CHAN\_BLOCK\_STEP\_i}[14:4] \text{ XSTEP} + \\ & \text{By} * \text{SIMCOP\_DMA\_CHAN\_BLOCK\_STEP\_i}[29:16] \text{ YSTEP} * \\ & \text{SIMCOP\_DMA\_CHAN\_SMEM\_OFST\_i}[19:4] \text{ OFST} \end{aligned}$$

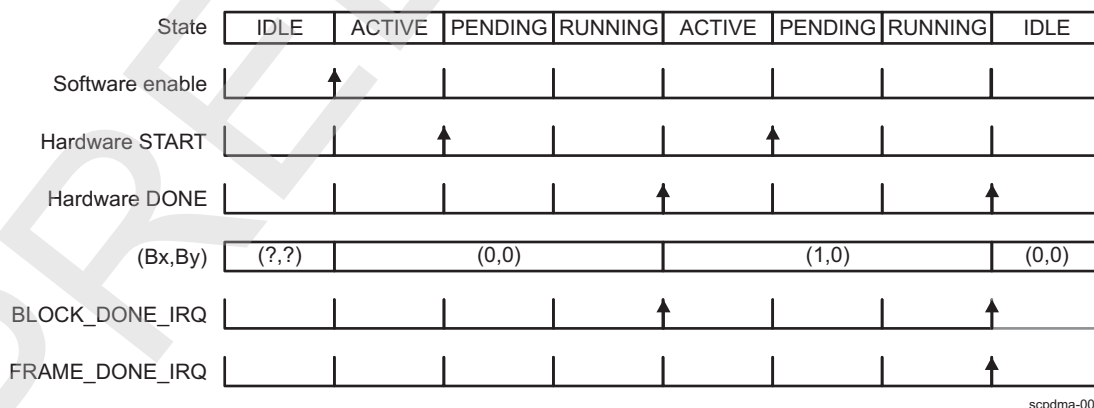
All 2D blocks of a logical channel have the same location in SIMCOP local memories:

$$\text{BUF\_BASE} = \text{SIMCOP\_DMA\_CHAN\_BUF\_ADDR\_i}[23:4] \text{ ADDR}$$

The same functionality can also be used to read or write sparse data from synchronous dynamic random access memory (SDRAM). Typically, that is useful to split processing between two SIMCOP accelerators (that is, iMX).

Software can poll the status of a channel using the `SIMCOP_DMA_CHAN_CURRENT_BLOCK_i` register. It holds the Bx, By values of the last-transferred 2D block. The Bx, By values are reset when a channel is enabled (set the `SIMCOP_DMA_CHAN_CTRL_i[0]` ENABLE bit) and they are updated when a channel enters the RUNNING state.

Figure 8-259 shows an example of a 2-block large transfer.

**Figure 8-259. 2-Block Large Transfer Example**

#### 8.4.3.3.5 Transaction Generation

Each transfer is decomposed into one or multiple transactions by the DMA engine. The firmware can define:

- Burst generation for regular or tiled accesses ([SIMCOP\\_DMA\\_CHAN\\_CTRL\\_i\[6\]](#) TILERMODE). Additional parameters must be defined for tiled accesses. See [Section 8.4.3.3.5.2, Block Bursts for Tiled Transfers](#).
- The maximum burst size ([SIMCOP\\_DMA\\_CTRL\[1:0\]](#) MAX\_BURST\_SIZE). It must be aligned with the interconnect configuration.
- Use of posted or nonposted writes ([SIMCOP\\_DMA\\_CTRL\[3\]](#) POSTED\_WRITES). It must be aligned with the interconnect configuration. Mainly, in a multichannel system memory, system bursts must be issued nonposted.

#### 8.4.3.3.5.1 Incrementing Bursts for Regular Transfers

Lines of 2D blocks are processed sequentially. The following transaction process is used for a given line:

1. Perform a single nonaligned burst from the start address to the next [SIMCOP\\_DMA\\_CTRL\[1:0\]](#) MAX\_BURST\_SIZE boundary (or less if the transfer is smaller).
2. Perform "n" aligned bursts (n can be 0).
3. Perform a single nonaligned burst up to the end address (if needed).

A transaction is aligned when the start address modulo burst length is 0.

#### 8.4.3.3.5.2 Block Bursts for Tiled Transfers

The line stride is configured using the [SIMCOP\\_DMA\\_CHAN\\_SMEM\\_OFST\\_i\[19:4\]](#) OFST bit field. The full register content is used for internal address computation. The [SIMCOP\\_DMA\\_CHAN\\_SMEM\\_OFST\\_i\[19:4\]](#) OFST bit field must be a multiple of the [SIMCOP\\_DMA\\_CTRL\[1:0\]](#) MAX\_BURST\_SIZE bit field.

The SIMCOP DMA has the following alignment constraints:

- [SIMCOP\\_DMA\\_CHAN\\_SMEM\\_ADDR\\_i\[31:4\]](#) ADDR is 128 bit-aligned.
- [SIMCOP\\_DMA\\_CHAN\\_BLOCK\\_SIZE\\_i\[13:4\]](#) XNUM is 128 bit-aligned.

Only the values in [Table 8-1305](#) are allowed for the [SIMCOP\\_DMA\\_CTRL\[1:0\]](#) MAX\_BURST\_SIZE bit field. Other values lead to undefined behavior.

**Table 8-1305. Allowed Maximum Burst Size**

Minimum Burst Height	MAX_BURST_SIZE
2	2, 4, or 8 x 128 bits
4	4 or 8 x 128 bits

### 8.4.3.4 ISS SIMCOP DMA Basic Programming Model

#### 8.4.3.4.1 Initialization of Surrounding Modules

To initialize the DMA, the surrounding modules must be initialized. [Table 8-1306](#) lists these modules.

**Table 8-1306. Initialization of Surrounding Modules**

Module	Minimum Required Setting	Optional Settings
CLKC	Enable SIMCOP clock.	
Buffers	Configure DMA access to image buffer.	
SIMCOP_CTRL		Configure SIMCOP_CTRL to handle DMA interrupts.

#### 8.4.3.4.2 ISS SIMCOP DMA Channel Configuration and Hardware Synchronization

Software must fully configure a DMA channel before enabling it. It then configures channel linking and the used synchronization method.

The following example shows four logical channels chained with hardware synchronization used to copy YUV4:2:0-NV12 data from SIMCOP local memories to system memory, and then to copy YUV4:2:0-NV12 data from system memory to SIMCOP local memories.

YUV4:2:0-NV12 data is stored in two separate buffers in SDRAM. Therefore, two channels are used for the SIMCOP -> system copy, and two channels are used for the system -> SIMCOP copy.

In this example, the intent is to transfer a 2 x 2 array of 64 x 32 pixel blocks. It is assumed that input data is stored contiguously in system memory and that the output buffer size in system memory is 256 x 32 pixels. In SIMCOP local memories, data is contiguously stored starting at address 0.

The following configuration is used:

- Channel  $i = 0$ , input Y buffer:
  - `SIMCOP_DMA_CHAN_SMEM_ADDR_i[31:4]` ADDR = Start address of Y input buffer
  - `SIMCOP_DMA_CHAN_SMEM_OFST_i[19:4]` OFST = 16 (= 256 bytes)
  - `SIMCOP_DMA_CHAN_BUF_ADDR_i[23:4]` ADDR = 0
  - `SIMCOP_DMA_CHAN_BUF_OFST_i[23:4]` OFST = 8 (= 128 bytes)
  - `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[13:4]` XNUM = 4 (= 64 bytes)
  - `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[28:16]` YNUM = 32
  - `SIMCOP_DMA_CHAN_FRAME_i[9:0]` XCNT = 2
  - `SIMCOP_DMA_CHAN_FRAME_i[25:16]` YCNT = 2
  - `SIMCOP_DMA_CHAN_BLOCK_STEP_i[14:4]` XSTEP = 4 (= 64 bytes)
  - `SIMCOP_DMA_CHAN_BLOCK_STEP_i[29:16]` YSTEP = 32
- Channel  $i = 1$ , input UV buffer:
  - `SIMCOP_DMA_CHAN_SMEM_ADDR_i[31:4]` ADDR = Start address of UV input buffer
  - `SIMCOP_DMA_CHAN_SMEM_OFST_i[19:4]` OFST = 16 (= 256 bytes)
  - `SIMCOP_DMA_CHAN_BUF_ADDR_i[23:4]` ADDR = 128 (2KB)
  - `SIMCOP_DMA_CHAN_BUF_OFST_i[23:4]` OFST = 8 (= 128 bytes)
  - `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[13:4]` XNUM = 4 (= 64 bytes)
  - `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[28:16]` YNUM = 16
  - `SIMCOP_DMA_CHAN_FRAME_i[9:0]` XCNT = 2
  - `SIMCOP_DMA_CHAN_FRAME_i[25:16]` YCNT = 2
  - `SIMCOP_DMA_CHAN_BLOCK_STEP_i[14:4]` XSTEP = 4 (= 64 bytes)
  - `SIMCOP_DMA_CHAN_BLOCK_STEP_i[29:16]` YSTEP = 16
- Channel  $i = 2$ , output Y buffer:
  - `SIMCOP_DMA_CHAN_SMEM_ADDR_i[31:4]` ADDR = Start address of Y output buffer
  - `SIMCOP_DMA_CHAN_SMEM_OFST_i[19:4]` OFST = 16 (= 256 bytes)
  - `SIMCOP_DMA_CHAN_BUF_ADDR_i[23:4]` ADDR = 0
  - `SIMCOP_DMA_CHAN_BUF_OFST_i[23:4]` OFST = 8 (= 128 bytes)
  - `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[13:4]` XNUM = 4 (= 64 bytes)
  - `SIMCOP_DMA_CHAN_BLOCK_SIZE_i[28:16]` YNUM = 32
  - `SIMCOP_DMA_CHAN_FRAME_i[9:0]` XCNT = 2
  - `SIMCOP_DMA_CHAN_FRAME_i[25:16]` YCNT = 2
  - `SIMCOP_DMA_CHAN_BLOCK_STEP_i[14:4]` XSTEP = 4 (= 64 bytes)
  - `SIMCOP_DMA_CHAN_BLOCK_STEP_i[29:16]` YSTEP = 32
- Channel  $i = 3$ , output UV buffer:
  - `SIMCOP_DMA_CHAN_SMEM_ADDR_i[31:4]` ADDR = Start address of UV output buffer
  - `SIMCOP_DMA_CHAN_SMEM_OFST_i[19:4]` OFST = 16 (= 256 bytes)
  - `SIMCOP_DMA_CHAN_BUF_ADDR_i[23:4]` ADDR = 128 (= 2KB)
  - `SIMCOP_DMA_CHAN_BUF_OFST_i[23:4]` OFST = 8 (= 128 bytes)

- `SIMCOP_DMA_CHAN_BLOCK_SIZE_i`[13:4] XNUM = 4 (= 64 bytes)
- `SIMCOP_DMA_CHAN_BLOCK_SIZE_i`[28:16] YNUM = 16
- `SIMCOP_DMA_CHAN_FRAME_i`[9:0] XCNT = 2
- `SIMCOP_DMA_CHAN_FRAME_i`[25:16] YCNT = 2
- `SIMCOP_DMA_CHAN_BLOCK_STEP_i`[14:4] XSTEP = 4 (= 64 bytes)
- `SIMCOP_DMA_CHAN_BLOCK_STEP_i`[29:16] YSTEP = 16

Figure 8-260 shows the channel linking chain used. All four channels are executed sequentially. Doing so prevents access collision in SIMCOP local memories or system memories. Channel 0 is started by a START[0] pulse from SIMCOP. When Channel 3 completes the last transfer, a pulse to DONE[0] is sent back to SIMCOP.

Figure 8-261 shows the temporal channel sequence.

Figure 8-260. YUV4:2:0-NV12 Read/Write Chain

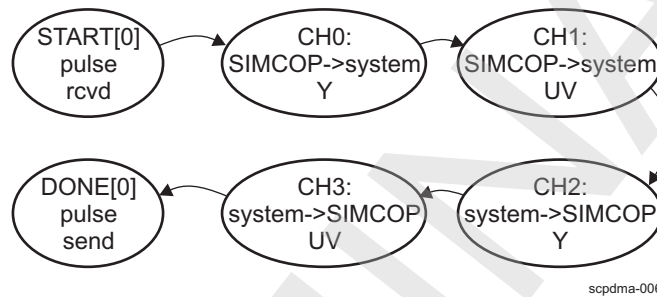
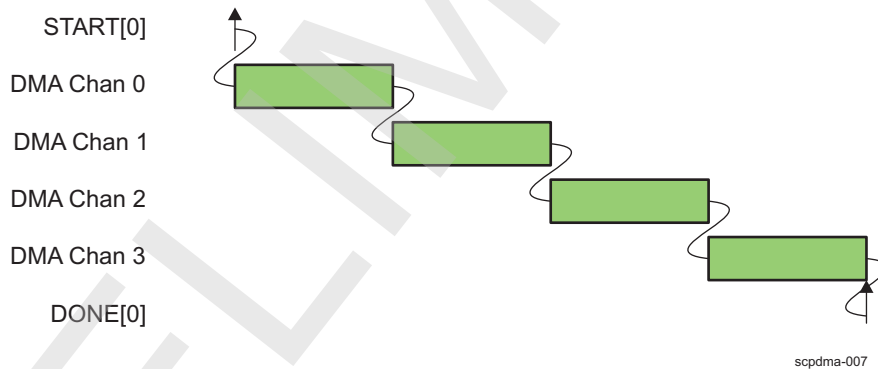


Figure 8-261. Temporal Channel Sequence



The following configuration is used for each channel:

- Channel  $i = 0$ , input Y buffer, first in sequence:
  - `SIMCOP_DMA_CHAN_CTRL_i`[16:12] LINKED = 0x11 (= CHAN1)
  - `SIMCOP_DMA_CHAN_CTRL_i`[19:17] HWSTART = 0x4 (= CHAN0)
  - `SIMCOP_DMA_CHAN_CTRL_i`[5] DIR = 1 (=SIMCOP buffers -> system memory)
  - `SIMCOP_DMA_CHAN_CTRL_i`[0] ENABLE = 1
- Channel  $i = 1$ , input UV buffer, second in sequence:
  - `SIMCOP_DMA_CHAN_CTRL_i`[16:12] LINKED = 0x12 (= CHAN2)
  - `SIMCOP_DMA_CHAN_CTRL_i`[5] DIR = 1 (= SIMCOP buffers -> system memory)
  - `SIMCOP_DMA_CHAN_CTRL_i`[0] ENABLE = 1
- Channel  $i = 2$ , output Y buffer, third in sequence:
  - `SIMCOP_DMA_CHAN_CTRL_i`[16:12] LINKED = 0x13 (= CHAN3)
  - `SIMCOP_DMA_CHAN_CTRL_i`[5] DIR = 0 (= system memory -> SIMCOP buffers)
  - `SIMCOP_DMA_CHAN_CTRL_i`[0] ENABLE = 1
- Channel  $i = 3$ , output UV buffer, last in sequence:

- `SIMCOP_DMA_CHAN_CTRL_j[16:12]` LINKED = 0x00 (no link)
- `SIMCOP_DMA_CHAN_CTRL_j[22:20]` HWSTART = 0x4 (= CHAN0)
- `SIMCOP_DMA_CHAN_CTRL_j[5]` DIR = 0 (= system memory -> SIMCOP buffers)
- `SIMCOP_DMA_CHAN_CTRL_j[0]` ENABLE = 1

Software can read the status of a logical channel at any time. However, logical channel configuration registers can be changed only when the channel is in the IDLE state.

Software can disable a logical channel by writing the `SIMCOP_DMA_CHAN_CTRL_j[1]` DISABLE bit. A channel is effectively disabled when `SIMCOP_DMA_CHAN_CTRL_j[4:3]` STATUS = IDLE. In case of linked channels, software must disable all channels in the chain and wait until all of them are back in the IDLE state before channels can be reprogrammed.

#### 8.4.3.4.3 Software Synchronization

Software performs the following sequence:

1. Configures the channel
2. Enables the channel
3. Triggers the channel by writing the `SIMCOP_DMA_CHAN_CTRL_j[2]` SWTRIGGER bit
4. Waits for BLOCK\_DONE\_IRQ

Step 2 and Step 4 are repeated until all 2D blocks have been transferred.



### 8.4.3.5 ISS SIMCOP DMA Register Manual

#### 8.4.3.5.1 ISS SIMCOP DMA Instance Summary

Table 8-1307 summarizes the DMA instance.

**Table 8-1307. SIMCOP DMA Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-M3 Private Access	Size
DMA	0x5202 0200	0x5506 0200	512 bytes

**NOTE:** Private Access is an access which is not using the L3/L4 interconnects.

#### 8.4.3.5.2 ISS SIMCOP DMA Registers

##### 8.4.3.5.2.1 ISS SIMCOP DMA Register Summary

Table 8-1308 summarizes the SIMCOP DMA register mapping.

**Table 8-1308. DMA Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-M3 Private Access
SIMCOP_DMA_REVISION	R	32	0x0000 0000	0x5202 0200	0x5506 0200
SIMCOP_DMA_HWINFO	R	32	0x0000 0004	0x5202 0204	0x5506 0204
SIMCOP_DMA_SYSCONFIG	RW	32	0x0000 0010	0x5202 0210	0x5506 0210
RESERVED	R	32	0x0000 0018	0x5202 0218	0x5506 0218
SIMCOP_DMA_CTRL	RW	32	0x0000 001C	0x5202 021C	0x5506 0218
SIMCOP_DMA_IRQSTATUS_RAW_j <sup>(1)</sup>	RW	32	0x0000 0020 + (0x10 * j)	0x5202 0220 + (0x10 * j)	0x5506 0220 + (0x10 * j)
SIMCOP_DMA_IRQSTATUS_j <sup>(1)</sup>	RW	32	0x0000 0024 + (0x10 * j)	0x5202 0224 + (0x10 * j)	0x5506 0224 + (0x10 * j)
SIMCOP_DMA_IRQENABLE_SET_j <sup>(1)</sup>	RW	32	0x0000 0028 + (0x10 * j)	0x5202 0228 + (0x10 * j)	0x5506 0228 + (0x10 * j)
SIMCOP_DMA_IRQENABLE_CLR_j <sup>(1)</sup>	RW	32	0x0000 002C + (0x10 * j)	0x5202 022C + (0x10 * j)	0x5506 022C + (0x10 * j)
SIMCOP_DMA_CHAN_CTRL_i <sup>(2)</sup>	RW	32	0x0000 0080 + (0x30 * i)	0x5202 0280 + (0x30 * i)	0x5506 0280 + (0x30 * i)
SIMCOP_DMA_CHAN_SMEM_ADD_R_j <sup>(2)</sup>	RW	32	0x0000 0084 + (0x30 * i)	0x5202 0284 + (0x30 * i)	0x5506 0284 + (0x30 * i)
SIMCOP_DMA_CHAN_SMEM_OFS_T_j <sup>(2)</sup>	RW	32	0x0000 0088 + (0x30 * i)	0x5202 0288 + (0x30 * i)	0x5506 0288 + (0x30 * i)
SIMCOP_DMA_CHAN_BUF_OFST_j <sup>(2)</sup>	RW	32	0x0000 008C + (0x30 * i)	0x5202 028C + (0x30 * i)	0x5506 028C + (0x30 * i)
SIMCOP_DMA_CHAN_BUF_ADDR_j <sup>(2)</sup>	RW	32	0x0000 0090 + (0x30 * i)	0x5202 0290 + (0x30 * i)	0x5506 0290 + (0x30 * i)
SIMCOP_DMA_CHAN_BLOCK_SIZE_j <sup>(2)</sup>	RW	32	0x0000 0094 + (0x30 * i)	0x5202 0294 + (0x30 * i)	0x5506 0294 + (0x30 * i)
SIMCOP_DMA_CHAN_FRAME_j <sup>(2)</sup>	RW	32	0x0000 0098 + (0x30 * i)	0x5202 0298 + (0x30 * i)	0x5506 0298 + (0x30 * i)
SIMCOP_DMA_CHAN_CURRENT_BLOCK_j <sup>(2)</sup>	R	32	0x0000 00A0 + (0x30 * i)	0x5202 02A0 + (0x30 * i)	0x5506 02A0 + (0x30 * i)
SIMCOP_DMA_CHAN_BLOCK_STEP_j <sup>(2)</sup>	RW	32	0x0000 00A4 + (0x30 * i)	0x5202 02A4 + (0x30 * i)	0x5506 02A4 + (0x30 * i)

<sup>(1)</sup> j = 0 to 7

<sup>(2)</sup> i = 0 to 1

### 8.4.3.5.2.2 ISS SIMCOP DMA Register Descriptions

Table 8-1309 through Table 8-1341 describe the registers in details.

**Table 8-1309. SIMCOP\_DMA\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															

Bits	Field Name	Description	Type	Reset
31:0	REV	Revision ID	R	_( <sup>(1)</sup> )

(<sup>1</sup>) TI internal data

**Table 8-1310. Register Call Summary for Register SIMCOP\_DMA\_REVISION**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Registers: \[0\]](#)

**Table 8-1311. SIMCOP\_DMA\_HWINFO**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	Information about the IP module's hardware configuration, that is, typically the module's HDL generics.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CHAN	CONTEXT														

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	CHAN	Logical channels Read 0x0: 4 Read 0x1: 8	R	1
1:0	CONTEXT	Maximum outstanding OCP transactions Read 0x0: 4 Read 0x1: 8 Read 0x2: 16	R	0x2

**Table 8-1312. Register Call Summary for Register SIMCOP\_DMA\_HWINFO**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Logical Channels: \[0\]](#)
- [ISS SIMCOP DMA Registers: \[1\]](#)

**Table 8-1313. SIMCOP\_DMA\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	DMA
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .		
<b>Description</b>	Clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STANDBYMODE			RESERVED												

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5:4	STANDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state.  0x0: Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only.  0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only.  0x2: Smart-standby mode.  0x3: Smart-standby mode.	RW	0x2
3:0	RESERVED		R	0x0

**Table 8-1314. Register Call Summary for Register SIMCOP\_DMA\_SYSCONFIG**

- ISS SIMCOP DMA Module
- [ISS SIMCOP DMA Power Management: \[0\]](#)
  - [ISS SIMCOP DMA Registers: \[1\]](#)

**Table 8-1315. SIMCOP\_DMA\_CTRL**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	DMA
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BW_LIMITER																RESERVED						TAG_CNT			POSTED_WRITES	RESERVED	MAX_BURST_SIZE				

Bits	Field Name	Description	Type	Reset
31:16	BW_LIMITER	SIMCOP DMA guarantees that there are at least BW_LIMITER functional clock cycles between two OCP requests. No IDLE cycles are inserted during an OCP transaction. This parameter could be used to reduce traffic generated by the SIMCOP DMA for non timing critical applications. Doing so leaves more BW for other system initiators. Default value corresponds to maximum performance.	RW	0x0000
15:8	RESERVED		R	0x00
7:4	TAG_CNT	Limits the outstanding transactions count. Only tags 0 - TAG_CNT will be used by SIMCOP DMA. The maximum allowed value is $2^{(SIMCOP\_DMA\_GNC\_CONTEXT+2)-1}$	RW	0x3
3	POSTED_WRITES	Select write type. Setting depend on the used interconnect 0x0: Only nonposted writes are generated 0x1: Only posted writes are generated	RW	0
2	RESERVED		R	0
1:0	MAX_BURST_SIZE	Defines the maximum burst length for INCR bursts. In case of 2D bursts, length x height is less or equal to this value. 0x0: Single requests only 0x1: less or equal to 2 0x2: less or equal to 4 0x3: less or equal to 8	RW	0x0

**Table 8-1316. Register Call Summary for Register SIMCOP\_DMA\_CTRL**

ISS SIMCOP DMA Module

- [Transaction Generation: \[0\] \[1\] \[2\] \[3\] \[5\]](#)
- [ISS SIMCOP DMA Registers: \[6\]](#)

**Table 8-1317. SIMCOP\_DMA\_IRQSTATUS\_RAW\_j**

<b>Address Offset</b>	0x0000 0020 + (0x10 * j)
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .
<b>Description</b>	Per-event raw interrupt status vector Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHAN7_FRAME_DONE_IRQ	CHAN6_FRAME_DONE_IRQ	CHAN5_FRAME_DONE_IRQ	CHAN4_FRAME_DONE_IRQ	CHAN3_FRAME_DONE_IRQ	CHAN2_FRAME_DONE_IRQ	CHAN1_FRAME_DONE_IRQ	CHAN0_FRAME_DONE_IRQ	CHAN7_BLOCK_DONE_IRQ	CHAN6_BLOCK_DONE_IRQ	CHAN5_BLOCK_DONE_IRQ	CHAN4_BLOCK_DONE_IRQ	CHAN3_BLOCK_DONE_IRQ	CHAN2_BLOCK_DONE_IRQ	CHAN1_BLOCK_DONE_IRQ	CHAN0_BLOCK_DONE_IRQ	RESERVED											OCP_ERR				

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
31	CHAN7_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
30	CHAN6_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
29	CHAN5_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
28	CHAN4_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
27	CHAN3_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
26	CHAN2_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
25	CHAN1_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
24	CHAN0_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
23	CHAN7_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
22	CHAN6_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
21	CHAN5_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
20	CHAN4_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
19	CHAN3_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
18	CHAN2_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
17	CHAN1_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
16	CHAN0_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW	0
15:1	RESERVED		R	0x0000
0	OCP_ERR	OCP error Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0

**Table 8-1318. Register Call Summary for Register SIMCOP\_DMA\_IRQSTATUS\_RAW\_j**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Interrupt Requests: \[0\] \[1\]](#)
- [ISS SIMCOP DMA Registers: \[2\]](#)

**Table 8-1319. SIMCOP\_DMA\_IRQSTATUS\_j**

<b>Address Offset</b>	0x0000 0024 + (0x10 * j)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	Per-event "enabled" interrupt status vector Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHAN7_FRAME_DONE_IRQ	CHAN6_FRAME_DONE_IRQ	CHAN5_FRAME_DONE_IRQ	CHAN4_FRAME_DONE_IRQ	CHAN3_FRAME_DONE_IRQ	CHAN2_FRAME_DONE_IRQ	CHAN1_FRAME_DONE_IRQ	CHAN0_FRAME_DONE_IRQ	CHAN7_BLOCK_DONE_IRQ	CHAN6_BLOCK_DONE_IRQ	CHAN5_BLOCK_DONE_IRQ	CHAN4_BLOCK_DONE_IRQ	CHAN3_BLOCK_DONE_IRQ	CHAN2_BLOCK_DONE_IRQ	CHAN1_BLOCK_DONE_IRQ	CHAN0_BLOCK_DONE_IRQ	RESERVED											BUS_ERR				

Bits	Field Name	Description	Type	Reset
31	CHAN7_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
30	CHAN6_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
29	CHAN5_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
28	CHAN4_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
27	CHAN3_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
26	CHAN2_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
25	CHAN1_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
24	CHAN0_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
23	CHAN7_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
22	CHAN6_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
21	CHAN5_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
20	CHAN4_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
19	CHAN3_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
18	CHAN2_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
17	CHAN1_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0



Bits	Field Name	Description	Type	Reset
16	CHAN0_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
15:1	RESERVED		R	0x0000
0	BUS_ERR	BUS error Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0

**Table 8-1320. Register Call Summary for Register SIMCOP\_DMA\_IRQSTATUS\_j**

ISS SIMCOP DMA Module

- ISS SIMCOP DMA Interrupt Requests: [0] [1]
- ISS SIMCOP DMA Registers: [2]

**Table 8-1321. SIMCOP\_DMA\_IRQENABLE\_SET\_j**

<b>Address Offset</b>	0x0000 0028 + (0x10 * j)
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .
<b>Description</b>	Per-event interrupt enable bit vector Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHAN7_FRAME_DONE_IRQ	CHAN6_FRAME_DONE_IRQ	CHAN5_FRAME_DONE_IRQ	CHAN4_FRAME_DONE_IRQ	CHAN3_FRAME_DONE_IRQ	CHAN2_FRAME_DONE_IRQ	CHAN1_FRAME_DONE_IRQ	CHAN0_FRAME_DONE_IRQ	CHAN7_BLOCK_DONE_IRQ	CHAN6_BLOCK_DONE_IRQ	CHAN5_BLOCK_DONE_IRQ	CHAN4_BLOCK_DONE_IRQ	CHAN3_BLOCK_DONE_IRQ	CHAN2_BLOCK_DONE_IRQ	CHAN1_BLOCK_DONE_IRQ	CHAN0_BLOCK_DONE_IRQ	RESERVED											0CP_ERR				

Bits	Field Name	Description	Type	Reset
31	CHAN7_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
30	CHAN6_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
29	CHAN5_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
28	CHAN4_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
27	CHAN3_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
26	CHAN2_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
25	CHAN1_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
24	CHAN0_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
23	CHAN7_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
22	CHAN6_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
21	CHAN5_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
20	CHAN4_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
19	CHAN3_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
18	CHAN2_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
17	CHAN1_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
16	CHAN0_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
15:1	RESERVED		R	0x0000
0	OCP_ERR	OCP error Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0

**Table 8-1322. Register Call Summary for Register SIMCOP\_DMA\_IRQENABLE\_SET\_j**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Interrupt Requests: \[0\]](#)
- [ISS SIMCOP DMA Registers: \[1\]](#)

**Table 8-1323. SIMCOP\_DMA\_IRQENABLE\_CLR\_j**

<b>Address Offset</b>	0x0000 002C + (0x10 * j)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	Per-event interrupt enable bit vector Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHAN7_FRAME_DONE_IRQ	CHAN6_FRAME_DONE_IRQ	CHAN5_FRAME_DONE_IRQ	CHAN4_FRAME_DONE_IRQ	CHAN3_FRAME_DONE_IRQ	CHAN2_FRAME_DONE_IRQ	CHAN1_FRAME_DONE_IRQ	CHAN0_FRAME_DONE_IRQ	CHAN7_BLOCK_DONE_IRQ	CHAN6_BLOCK_DONE_IRQ	CHAN5_BLOCK_DONE_IRQ	CHAN4_BLOCK_DONE_IRQ	CHAN3_BLOCK_DONE_IRQ	CHAN2_BLOCK_DONE_IRQ	CHAN1_BLOCK_DONE_IRQ	CHAN0_BLOCK_DONE_IRQ	RESERVED											OCP_ERR				

Bits	Field Name	Description	Type	Reset
31	CHAN7_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
30	CHAN6_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
29	CHAN5_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
28	CHAN4_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
27	CHAN3_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
26	CHAN2_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
25	CHAN1_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
24	CHAN0_FRAME_DONE_IRQ	Channel has completed transfer of the full frame Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
23	CHAN7_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
22	CHAN6_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
21	CHAN5_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
20	CHAN4_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
19	CHAN3_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
18	CHAN2_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
17	CHAN1_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
16	CHAN0_BLOCK_DONE_IRQ	Channel has completed transfer of one 2D block Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
15:1	RESERVED		R	0x0000

Bits	Field Name	Description	Type	Reset
0	OCP_ERR	OCP error Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0

**Table 8-1324. Register Call Summary for Register SIMCOP\_DMA\_IRQENABLE\_CLR\_j**

ISS SIMCOP DMA Module

- ISS SIMCOP DMA Interrupt Requests: [0]
- ISS SIMCOP DMA Registers: [1]

**Table 8-1325. SIMCOP\_DMA\_CHAN\_CTRL\_i**

<b>Address Offset</b>	0x0000 0080 + (0x30 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	Logical channel control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HWSTOP				HWSTART				LINKED				RESERVED				TILERMODE	DIR	STATUS	SWTRIGGER	DISABLE	ENABLE		

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x000
22:20	HWSTOP	DMA logical channel hardware synchronization. Controls generation of the DONE pulse for the logical channel. Only the values listed below are allowed. Other values lead to undefined behavior. 0x0: Disabled. 0x4: Use hardware synchronization channel 0 0x5: Use hardware synchronization channel 1 0x6: Use hardware synchronization channel 2 0x7: Use hardware synchronization channel 3	RW	0x0
19:17	HWSTART	DMA logical channel hardware synchronization. Controls sensitivity of the logical channel on a START pulse. Only the values listed below are allowed. Other values lead to undefined behavior. 0x0: Disabled. 0x4: Use hardware synchronization channel 0 0x5: Use hardware synchronization channel 1 0x6: Use hardware synchronization channel 2 0x7: Use hardware synchronization channel 3	RW	0x0

Bits	Field Name	Description	Type	Reset
16:12	LINKED	DMA logical channel linking. Only the values listed below are allowed. Other values lead to undefined behavior. 0x0: Disabled. 0x10: Start channel 0 when this channel has completed transfer of one 2D block 0x11: Start channel 1 when this channel has completed transfer of one 2D block 0x12: Start channel 2 when this channel has completed transfer of one 2D block 0x13: Start channel 3 when this channel has completed transfer of one 2D block 0x14: Start channel 4 when this channel has completed transfer of one 2D block 0x15: Start channel 5 when this channel has completed transfer of one 2D block 0x16: Start channel 6 when this channel has completed transfer of one 2D block 0x17: Start channel 7 when this channel has completed transfer of one 2D block	RW	0x00
11:7	RESERVED		R	0x0
6	TILERMODE	Selects OCP transaction breakdown algorithm 0x0: Regular mode. INCR burst are used. ADDR[32]=0 for OCP transactions 0x1: Tiler mode. BLCK burst are used. ADDR[32]=1 for OCP transactions	RW	-
5	DIR	Transfer direction 0x0: System memory -> SIMCOP buffers 0x1: SIMCOP buffers -> system memory	RW	0
4:3	STATUS	SW could poll this bit to know the state of the channel Read 0x0: Idle Read 0x1: Active Read 0x2: Pending Read 0x3: Running	R	0x0
2	SWTRIGGER	Software trigger of the DMA channel. Read of this register always returns 0. Write 0x0: No effect Write 0x1: Change the logical channel state to PENDING if it is in ACTIVE state. No effect if the channel is in RUNNING, PENDING or IDLE state	W	0
1	DISABLE	Disable control of the logical channel. Read of this register always returns 0. Write 0x0: No effect. Write 0x1: Disable the channel. Changes the logical channel state to IDLE when it is in ACTIVE state. Memorize a disable request when the channel is in RUNNING or PENDING state.	W	0
0	ENABLE	Enable control of the logical channel. Read of this register always returns 0. Write 0x0: No effect Write 0x1: Enable the channel. Changes the state of the logical channel from IDLE to ACTIVE.	W	0

**Table 8-1326. Register Call Summary for Register SIMCOP\_DMA\_CHAN\_CTRL\_i**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Logical Channels: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [Transaction Generation: \[15\]](#)
- [ISS SIMCOP DMA Registers: \[18\]](#)
- [ISS SIMCOP DMA Channel Configuration and Hardware Synchronization: \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\]](#)
- [Software Synchronization: \[35\]](#)

**Table 8-1327. SIMCOP\_DMA\_CHAN\_SMEM\_ADDR\_i**

<b>Address Offset</b>	0x0000 0084 + (0x30 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	System memory address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Address in 128-bit words	RW	0x-----
3:0	RESERVED		R	0x0

**Table 8-1328. Register Call Summary for Register SIMCOP\_DMA\_CHAN\_SMEM\_ADDR\_i**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Logical Channels: \[0\]](#)
- [Transaction Generation: \[1\]](#)
- [ISS SIMCOP DMA Registers: \[2\]](#)
- [ISS SIMCOP DMA Channel Configuration and Hardware Synchronization: \[3\] \[4\] \[5\] \[6\]](#)

**Table 8-1329. SIMCOP\_DMA\_CHAN\_SMEM\_OFST\_i**

<b>Address Offset</b>	0x0000 0088 + (0x30 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	System memory line offset in 128-bit words. Maximum stride = 1MB		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												OFST												RESERVED							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:4	OFST	Line offset. In 128-bit words.	RW	0x----
3:0	RESERVED		R	0x0

**Table 8-1330. Register Call Summary for Register SIMCOP\_DMA\_CHAN\_SMEM\_OFST\_i**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Logical Channels: \[0\]](#)
- [Transaction Generation: \[1\] \[2\]](#)
- [ISS SIMCOP DMA Registers: \[3\]](#)
- [ISS SIMCOP DMA Channel Configuration and Hardware Synchronization: \[4\] \[5\] \[6\] \[7\]](#)



**Table 8-1331. SIMCOP\_DMA\_CHAN\_BUF\_OFST\_i**

<b>Address Offset</b>	0x0000 008C + (0x30 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	SIMCOP memory line offset		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OFST																RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:4	OFST	Line offset. In 128-bit words.	RW	0x----
3:0	RESERVED		R	0x0

**Table 8-1332. Register Call Summary for Register SIMCOP\_DMA\_CHAN\_BUF\_OFST\_i**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Registers: \[0\]](#)
- [ISS SIMCOP DMA Channel Configuration and Hardware Synchronization: \[1\] \[2\] \[3\] \[4\]](#)

**Table 8-1333. SIMCOP\_DMA\_CHAN\_BUF\_ADDR\_i**

<b>Address Offset</b>	0x0000 0090 + (0x30 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	SIMCOP memory address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ADDR																RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:4	ADDR	Address in 128-bit words.	RW	0x----
3:0	RESERVED		R	0x0

**Table 8-1334. Register Call Summary for Register SIMCOP\_DMA\_CHAN\_BUF\_ADDR\_i**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Logical Channels: \[0\]](#)
- [ISS SIMCOP DMA Registers: \[1\]](#)
- [ISS SIMCOP DMA Channel Configuration and Hardware Synchronization: \[2\] \[3\] \[4\] \[5\]](#)

**Table 8-1335. SIMCOP\_DMA\_CHAN\_BLOCK\_SIZE\_i**

<b>Address Offset</b>	0x0000 0094 + (0x30 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	2D block size		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								YNUM								RESERVED				XNUM								RESERVED			

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	YNUM	Height, in lines, per 2D block. Valid values are 1- 8191.	RW	0bxxxxxxxxxxxx
15:14	RESERVED		R	0x0
13:4	XNUM	Width, in 128-bit words, per 2D block. Valid values are 1-1023, that corresponds to 16 bytes to 16KB.	RW	0bxxxxxxxx
3:0	RESERVED		R	0x0

**Table 8-1336. Register Call Summary for Register SIMCOP\_DMA\_CHAN\_BLOCK\_SIZE\_i**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Logical Channels: \[0\] \[1\] \[2\] \[3\]](#)
- [Transaction Generation: \[4\]](#)
- [ISS SIMCOP DMA Registers: \[5\]](#)
- [ISS SIMCOP DMA Channel Configuration and Hardware Synchronization: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)

**Table 8-1337. SIMCOP\_DMA\_CHAN\_FRAME\_i**

<b>Address Offset</b>	0x0000 0098 + (0x30 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	Defines a frame. A frame is composed of 2D blocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								YCNT								RESERVED				XCNT											

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved.	R	0x00
25:16	YCNT	Vertical count of 2D blocks per frame. Valid values are 1-1023	RW	0bxxxxxxxx
15:10	RESERVED	Reserved.	R	0x00
9:0	XCNT	Horizontal count of 2D blocks per frame. Valid values are 1-1023	RW	0bxxxxxxxx

**Table 8-1338. Register Call Summary for Register SIMCOP\_DMA\_CHAN\_FRAME\_i**

ISS SIMCOP DMA Module

- [ISS SIMCOP DMA Logical Channels: \[0\] \[1\]](#)
- [ISS SIMCOP DMA Registers: \[2\]](#)
- [ISS SIMCOP DMA Channel Configuration and Hardware Synchronization: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

**Table 8-1339. SIMCOP\_DMA\_CHAN\_CURRENT\_BLOCK\_i**

<b>Address Offset</b>	0x0000 00A0 + (0x30 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	SW could read the coordinates of the last transferred block. The status is reset when the channel is enabled (change the state of CTRL.ENABLE from 0 to 1).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BY								RESERVED								BX							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved.	R	0x00
25:16	BY	Vertical position of the last transferred 2D block in the frame.	R	0bxxxxxxxxxx
15:10	RESERVED	Reserved.	R	0x00
9:0	BX	Horizontal position of the last transferred 2D block in the frame.	R	0bxxxxxxxxxx

**Table 8-1340. Register Call Summary for Register SIMCOP\_DMA\_CHAN\_CURRENT\_BLOCK\_i**

- ISS SIMCOP DMA Module
- [ISS SIMCOP DMA Logical Channels: \[0\]](#)
  - [ISS SIMCOP DMA Registers: \[1\]](#)

**Table 8-1341. SIMCOP\_DMA\_CHAN\_BLOCK\_STEP\_i**

<b>Address Offset</b>	0x0000 00A4 + (0x30 * i)		
<b>Physical Address</b>	See <a href="#">Table 8-1308</a> .	<b>Instance</b>	DMA
<b>Description</b>	Offset between 2D blocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED								YSTEP								RESERVED								XSTEP								RESERVED							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	YSTEP	Vertical offset, in lines, between rows of 2D blocks. For contiguous 2D blocks YSTEP=YNUM Valid values are -8192 to +8191.	RW	0b0xxxxxxxxxxxxx
15	RESERVED		R	0
14:4	XSTEP	Horizontal offset, in 128-bit words, between 2D block columns. For contiguous 2D blocks XSTEP=XNUM Valid values are -1024 to +1023, that corresponds to 16 bytes to 16KB.	RW	0bxxxxxxxxxxxxx
3:0	RESERVED		R	0x0

**Table 8-1342. Register Call Summary for Register SIMCOP\_DMA\_CHAN\_BLOCK\_STEP\_i**

- ISS SIMCOP DMA Module
- [ISS SIMCOP DMA Logical Channels: \[0\] \[1\] \[2\] \[3\]](#)
  - [ISS SIMCOP DMA Registers: \[4\]](#)
  - [ISS SIMCOP DMA Channel Configuration and Hardware Synchronization: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)

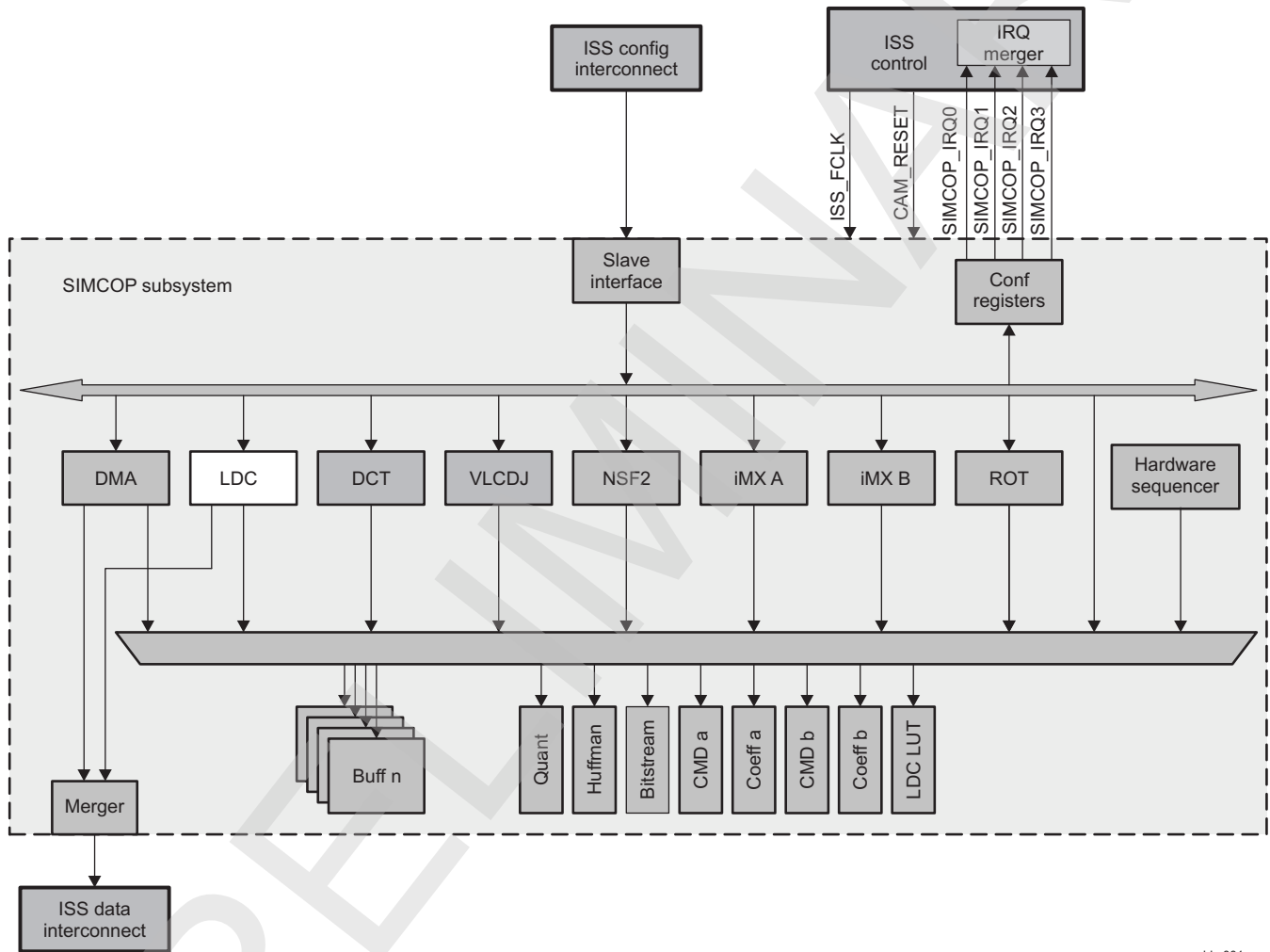
### 8.4.4 ISS SIMCOP LDC Module

This chapter describes the lens distortion correction (LDC) module in the still image coprocessor (SIMCOP) subsystem.

#### 8.4.4.1 ISS SIMCOP LDC Overview

The LDC deals with lens geometric distortion issues, chromatic aberration, and affine transformation for camera images. [Figure 8-262](#) is an overview of the LDC in the SIMCOP subsystem.

**Figure 8-262. LDC in the SIMCOP Subsystem**



ldc-001

For still image or video applications the LDC supports:

- Lens distortion correction:
  - YCbCr 4:2:2/4:2:0 data format input/output (I/O) for post-image-pipe correction
  - Correct barrel distortion and pin-cushion distortion
  - Radius-to-magnification-factor table to accommodate various distortion functions through programming
  - Configurable center point and horizontal/vertical adjustment
  - One 256-entry look-up table (LUT) for Y, Cb, and Cr
  - 8 bits for YCbCr
  - Bicubic interpolation for Y and bilinear interpolation for Cb/Cr
  - Bilinear interpolation mode for Y to offer faster processing
  - Up to 16,383 x 16,383 image dimension

- Format restriction:
  - YCbCr 4:2:2 format: Tile width must be a multiple of 16.
  - YCbCr 4:2:0 format: Tile width must be a multiple of 32.
- Image warping, scaling, and rotation:
  - Affine transformation for image warping, scaling, and rotation
  - 8 bits for YCbCr
  - Bicubic interpolation for Y and bilinear interpolation for Cb/Cr
  - Bilinear interpolation mode for Y to offer double throughput
  - Format restriction:
    - YCbCr 4:2:2 format: Tile width must be a multiple of 16.
    - YCbCr 4:2:0 format: Tile width must be a multiple of 32.
- Chromatic aberration correction:
  - Supports Bayer data in unpacked 12-bit per pixel, 8-bit per pixel A-Law, 8-bit per pixel unpacked, and 12-bit per pixel packed formats
  - Radius-to-magnification-factor table to accommodate various distortion functions through programming
  - Configurable center point and horizontal/vertical adjustment
  - Red and blue distortion only. Green pixels are copied to output.
  - Independent 128-entry LUTs for red and blue
  - Bilinear interpolation
  - Up to 16,383 x 16,383 image dimension
  - Format restriction:
    - Unpacked 12-bit format: Tile width must be a multiple of 16.
    - A-Law 8-bit format: Tile width must be a multiple of 32.
    - Unpacked 8-bit format: Tile width must be a multiple of 32.
    - Packed 12-bit format: Tile width must be a multiple of 64.

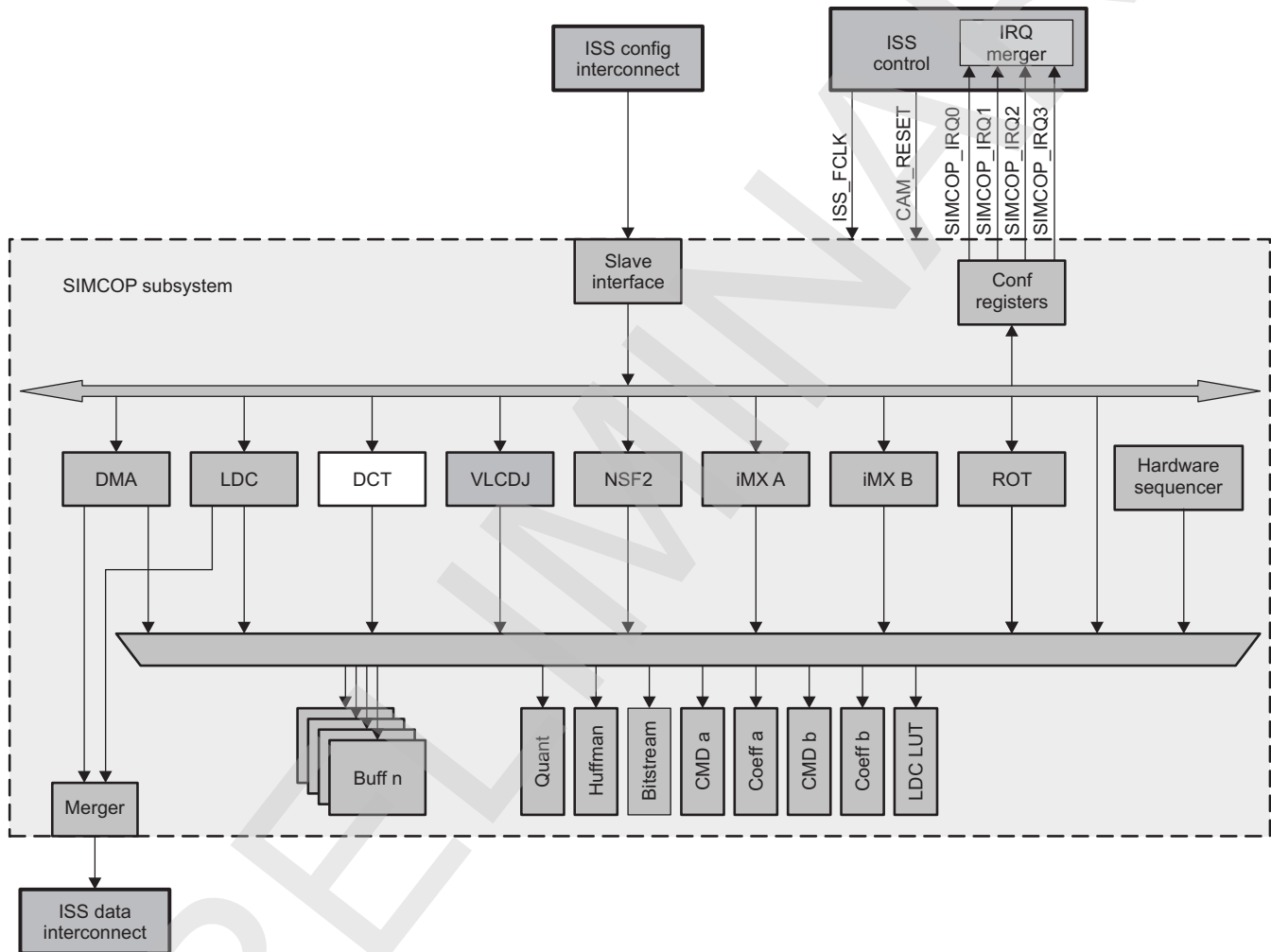
### 8.4.5 ISS SIMCOP Discrete Cosine Transform Module

This section describes the discrete cosine transform (DCT) module in the still-image coprocessor (SIMCOP) subsystem.

#### 8.4.5.1 ISS SIMCOP DCT Overview

The DCT module performs DCT and inverse DCT (IDCT) operations required for still-image coder/decoder (codec) applications. [Figure 8-263](#) is an overview of the DCT in the SIMCOP subsystem.

**Figure 8-263. DCT in the SIMCOP Subsystem**



dct-001

For still-image or video applications, the DCT supports:

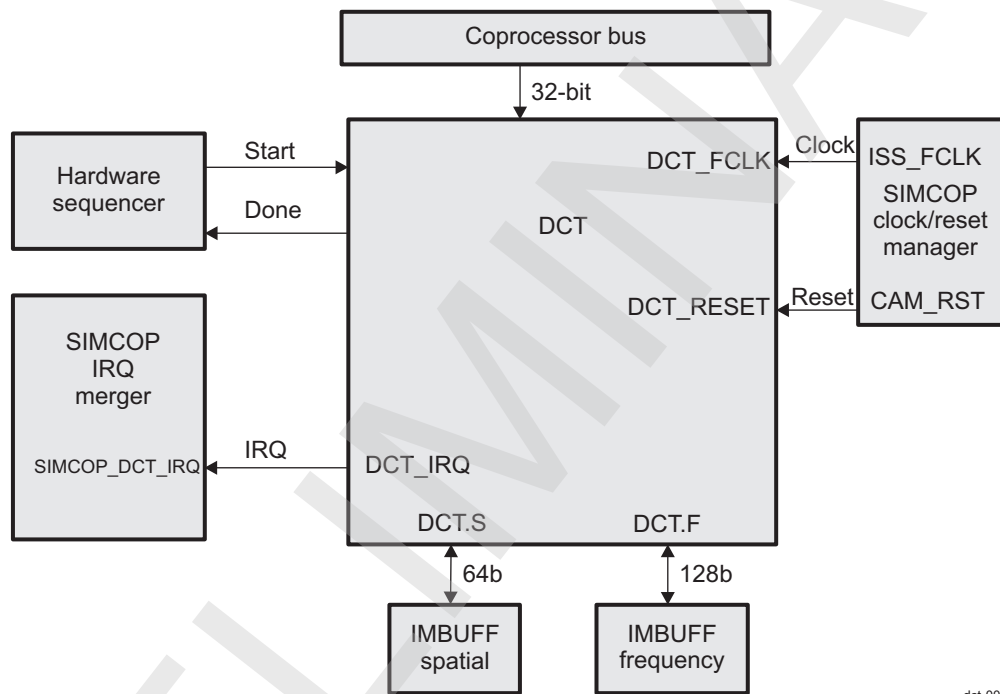
- 8-bit data in the spatial domain (input to DCT or output of IDCT) with  $-128$  offset on color components
- 16-bit data in the frequency domain
- Exact input/output (I/O) range:
  - DCT expected input range  $[0, 255]$ , subtracting 128 offset before transform, output saturated to  $[-1024, 1023]$
  - IDCT expected input range  $[-1280, 1079]$ , adding 128 to inverse transform outcome and saturating to  $[0, 255]$
  - IDCT input range is from DCT output range plus  $[-256, 256]$  worst-case quantization error
- Configurable to process up to 32 microcontroller units (MCUs) at a time (21 MCUs for YUV4:2:0, and 32 MCUs for YUV4:2:2, because of memory size limitations)
- The following formats are supported by the DCT:

- 16 x 16-pixel block MCU for YUV4:2:0 format, spatial data storage follows NV12
- 16 x 8-pixel block MCU for YUV4:2:2 format, spatial data storage follows UYVY
- Sequential 8 x 8 block mode, other than YUV4:2:0 and YUV4:2:2, a color-neutral format to allow flexibility in color component configuration in MCU, processes up to 64 blocks per task
- JPEG baseline sequential is supported by sequential 8 x 8 block mode. Some post-processing to reorder data and color space conversion may be required.
- All frequency domain data are stored sequentially for each 8 x 8 block, 16 bits per data point.

### 8.4.5.2 ISS SIMCOP DCT Integration

The DCT is part of the SIMCOP subsystem in the imaging subsystem (ISS). [Figure 8-264](#) shows the integration of the DCT in the SIMCOP subsystem.

**Figure 8-264. DCT Engine Integration**



[Table 8-1343](#) through [Table 8-1345](#) list the integration attributes, clocks and resets, and hardware requests, respectively, for the DCT.

**Table 8-1343. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
DCT	PD_CAM	

**Table 8-1344. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DCT	DCT_FCLK	ISS_FCLK	PRCM	Functional clock provided by ISS_CLK from the power, reset, and clock management (PRCM) module. It is used by all ISS submodules and ISS top-level resources.
Resets				
DCT	DCT_RESET	CAM_RST	PRCM	ISS and SIMCOP global reset

For information about clock and reset management, see [Section 8.4.1.2.1, ISS SIMCOP Local Power and Clock Management](#).

**Table 8-1345. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
DCT	DCT_IRQ	SIMCOP_DCT_IRQ	SIMCOP IRQ merger	DCT operation is complete.

For more information about interrupt requests, see [Section 8.4.1.2.3, Interrupt Merger](#).



### 8.4.5.3 ISS SIMCOP DCT Functional Description

#### 8.4.5.3.1 ISS SIMCOP DCT Block Diagram

The DCT implements 8 × 8, 2-dimensional (2D) DCT and IDCT specified in JPEG image coding standards:

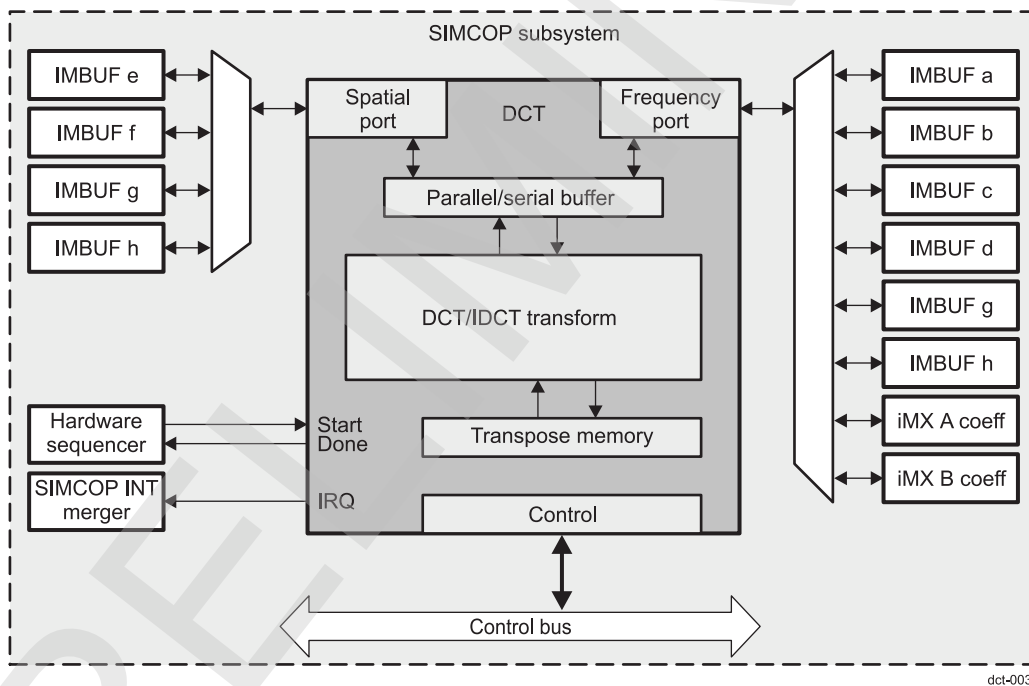
- In 4:2:0 mode, spatial domain data is organized in NV12 format. The frequency domain is organized as six blocks of 8 × 8.
- In 4:2:2 mode, spatial domain data is organized in UYVY format. The frequency domain is organized as four blocks of 8 × 8.
- In sequential 8 × 8 block mode, spatial and frequency domain data are organized as N blocks of 8 × 8 blocks.

**NOTE:**

- The spatial domain is the DCT input and IDCT output.
- The frequency domain is the DCT output and IDCT input.

Figure 8-265 is a block diagram of the DCT.

**Figure 8-265. DCT Block Diagram**



#### 8.4.5.3.2 ISS SIMCOP DCT Power Management

Table 8-1346 describes the power-management features available for the DCT.

**Table 8-1346. Local Power-Management Features**

Feature	Registers	Description
Clock autogating	DCT_CFG[5] AUTOGATING	Free-running or autogating modes are available.

#### 8.4.5.3.3 ISS SIMCOP DCT Interrupt Requests

Table 8-1347 lists the event flags, and their mask, that can cause module interrupts.

**Table 8-1347. Events**

Event Flag	Event Mask	Map to	Description
N/A	<a href="#">DCT_CFG[3]</a> INTEN	DCT_IRQ	End of processing

#### 8.4.5.3.4 Control and Status

The DCT is controlled through the control registers accessed through the coprocessor bus by an external initiator. Before starting the DCT, the buffer memory must be configured for DCT access and initialized, and the control register must be set (for more information about DCT/IDCT processing, see [Section 8.4.5.3.4.1, DCT Configuration](#)).

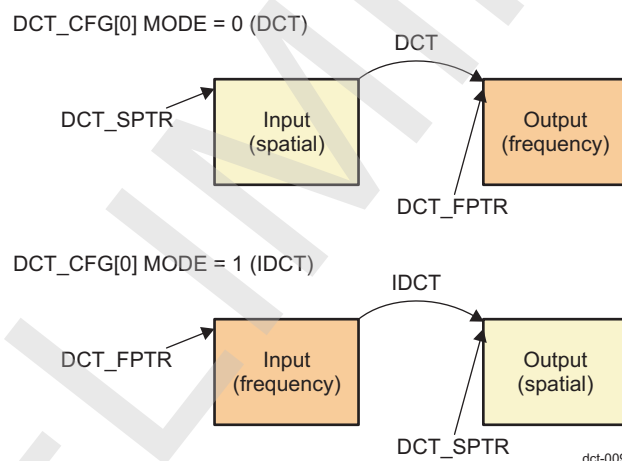
The [DCT\\_CFG\[4\]](#) TRIG\_SRC bit is used to select the DCT start source:

- If TRIG\_SRC = 0, the DCT is started through the [DCT\\_CTRL\[0\]](#) EN bit.
- If TRIG\_SRC = 1, the DCT is started by a hardware-start signal generated by the SIMCOP hardware sequencer.

When the DCT is configured for one trigger, the other trigger has no effect on the DCT.

An external initiator can read the state of the DCT (running or not running) by reading the [DCT\\_CTRL\[15\]](#) BUSY bit. When the BUSY bit goes back to 0, processing is complete (TRIG\_SRC = 0 or 1).

Before starting the DCT, the [DCT\\_SPTR](#) and [DCT\\_FPTR](#) registers must be configured with spatial and frequency addresses. [Figure 8-266](#) shows the use of spatial and frequency addresses.

**Figure 8-266. DCT I/O Address Configuration**

An interrupt can be generated when DCT operation completes (the configured number of MCUs for YUV4:2:0/4:2:2 mode, or the number of blocks for sequential block mode). To enable interrupt generation, set the [DCT\\_CFG\[3\]](#) INTEN bit to 1.

When the [DCT\\_CFG\[3\]](#) INTEN bit is set to 0, no interrupt generation is performed.

#### 8.4.5.3.4.1 DCT Configuration

The following features can be configured in the DCT register:

- DCT module operation: The [DCT\\_CFG\[0\] MODE](#) bit configures the processing:
  - MODE = 0: DCT processing
  - MODE = 1: IDCT processing
- Number of MCUs to process: The number of MCUs to process –1 is configured in the [DCT\\_CFG\[13:8\]](#) NMCUS bit field:
  - YUV4:2:0 maximum MCUs = 21 (memory limitation, 126 blocks)
  - YUV4:2:2 maximum MCUs = 32 (128 blocks)
  - Sequential maximum MCUs = 64 (64 blocks)

### 8.4.5.3.5 Data Format

The DCT reads and writes data from and to image buffers. I/O format can be configured in the `DCT_CFG[2:1]` FMT bit field. The following formats can be configured:

- YUV4:2:0
- YUV4:2:2
- Sequential

#### 8.4.5.3.5.1 YUV4:2:0 Format

In JPEG YUV4:2:0 format, each MCU is  $16 \times 16$  pixels, and has  $16 \times 16$  Y data points,  $8 \times 8$  U data points, and  $8 \times 8$  V data points. Each square array of  $8 \times 8$  data points is called a block. Thus, an MCU with a YUV4:2:0 format has four Y blocks, one U block, and one V block.

This module follows the NV12 data storage format for YUV4:2:0 spatial data. For YUV4:2:0 format:

- `line_offset_in_bytes` = `(DCT_CFG[13:8] NMCUS + 1) 16`
- `starting_addr_uv` = `DCT_SPTR + (DCT_CFG[13:8] NMCUS + 1) 256`

Figure 8-267 shows the organization of the data for two MCUs in the spatial domain, when the DCT is configured to process two MCUs. To select the YUV4:2:0 format, write 0x0 in the `DCT_CFG[2:1]` FMT bit field. YUV4:2:0 processes six blocks (four Y, one U, and one V) during each MCU DCT processing.

**Figure 8-267. YUV4:2:0 Format Data Order**

		Byte offset					
		0	1	2	3	...	31
0		Y(0,0)	Y(1,0)	Y(2,0)	Y(3,0)	...	Y(15,0)
32		Y(0,1)	Y(1,1)	Y(2,1)	Y(3,1)	...	Y(15,1)
64		Y(0,2)	Y(1,2)	Y(2,2)	Y(3,2)	...	Y(15,2)
	⋮						
480		Y(0,15)	Y(1,15)	Y(2,15)	Y(3,15)	...	Y(15,15)
512		U(0,0)	V(0,0)	U(2,0)	V(2,0)	...	V(14,0)
544		U(0,2)	V(0,2)	U(2,2)	V(2,2)	...	V(14,2)
736		U(0,14)	V(0,14)	U(2,14)	V(2,14)	...	V(14,2)

dct-005

Each YUV4:2:0 has four Y blocks, one U block, and one V block. For the first MCU in Figure 8-267, the blocks are:

- Y\_Block 0:
  - Y(0, 0), Y(1, 0), ..., Y(7, 0)
  - Y(0, 1), Y(1, 1), ..., Y(7, 1)
  - ....
  - Y(0, 7), Y(1, 7), ..., Y(7, 7)
- Y\_Block 1:
  - Y(8, 0), Y(9, 0), ..., Y(15, 0)
  - Y(8, 1), Y(9, 1), ..., Y(15, 1)
  - ....
  - Y(8, 7), Y(9, 7), ..., Y(15, 7)
- Y\_Block 2:
  - Y(0, 8), Y(1, 8), ..., Y(7, 8)
  - Y(0, 9), Y(1, 9), ..., Y(7, 9)
  - ....
  - Y(0, 15), Y(1, 15), ..., Y(7, 15)
- Y\_Block 3:

- Y(8, 8), Y(9, 8), ..., Y(15, 8)
- Y(8, 9), Y(9, 9), ..., Y(15, 9)
- ....
- Y(8, 15), Y(9, 15), ..., Y(15, 15)
- U\_Block:
  - U(0, 0), U(2, 0), ..., U(14, 0)
  - U(0, 2), U(2, 2), ..., U(14, 2)
  - ....
  - U(0, 14), U(2, 14), ..., U(14, 14)
- V\_Block:
  - V(0, 0), V(2, 0), ..., V(14, 0)
  - V(0, 2), V(2, 2), ..., V(14, 2)
  - ....
  - V(0, 14), V(2, 14), ..., V(14, 14)

#### 8.4.5.3.5.2 YUV4:2:2 Format

In JPEG YUV4:2:2 format, each MCU is  $16 \times 8$  pixels and has  $16 \times 8$  Y data points,  $8 \times 8$  U data points, and  $8 \times 8$  V data points. Thus, an MCU with a YUV4:2:2 format has two Y blocks, one U block, and one V block.

This module follows the UYVY data storage format for YUV4:2:2 spatial data. For YUV4:2:2 format:  $\text{line\_offset\_in\_bytes} = (\text{DCT\_CFG}[13:8] \text{ NMCUS} + 1) \cdot 32$

Figure 8-268 shows the organization of the data for two MCUs in the spatial domain, when the DCT is configured to process two MCUs. To select the YUV4:2:2 format, write 0x1 in the `DCT_CFG[2:1]` FMT bit field. YUV4:2:2 processes four blocks (two Y, one U, and one V) during each MCU DCT processing.

**Figure 8-268. YUV4:2:2 Format Data Order**

Byte offset	0	1	2	3	4	5	6	7		31	32	33	34		63
0	U(0,0)	Y(0,0)	V(0,0)	Y(1,0)	U(2,0)	Y(2,0)	V(2,0)	Y(3,0)	...	Y(15,0)	U(16,0)	Y(16,0)	V(16,0)	...	Y(31,0)
64	U(0,1)	Y(0,1)	V(0,1)	Y(1,1)	U(2,1)	Y(2,1)	V(2,1)	Y(3,1)	...	Y(15,1)	U(16,1)	Y(16,1)	V(16,1)	...	Y(31,1)
128	U(0,2)	Y(0,2)	V(0,2)	Y(1,2)	U(2,2)	Y(2,2)	V(2,2)	Y(3,2)	...	Y(15,2)	U(16,2)	Y(16,2)	V(16,2)	...	Y(31,2)
	⋮								...					...	
448	U(0,7)	Y(0,7)	V(0,7)	Y(1,7)	U(2,7)	Y(2,7)	V(2,7)	Y(3,7)	...	Y(15,7)	U(16,7)	Y(16,7)	V(16,7)	...	Y(31,7)

dct-011

Each YUV4:2:2 has two Y blocks, one U block, and one V block. The four blocks of the first MCU are:

- Y\_Block 0:
  - Y(0, 0), Y(1, 0), ..., Y(7, 0)
  - Y(0, 1), Y(1, 1), ..., Y(7, 1)
  - ....
  - Y(0, 7), Y(1, 7), ..., Y(7, 7)
- Y\_Block 1:
  - Y(8, 0), Y(9, 0), ..., Y(15, 0)
  - Y(8, 1), Y(9, 1), ..., Y(15, 1)
  - ....
  - Y(8, 7), Y(9, 7), ..., Y(15, 7)

#### **8.4.5.3.5.3 Sequential Block Format**

The sequential block format supports any color format other than YUV4:2:0 or YUV4:2:2. There is no notion of colors, and each 8 × 8 block is ordered sequentially. The DCT can be configured to process N blocks (versus N MCUs for YUV4:2:0 and YUV4:2:2, N is up to 64). In this format:

- Block i =
  - $X(8*i, 0), X(8*i+1, 0), \dots, X(8*i+7, 0)$
  - $X(8*i, 1), X(8*i+1, 1), \dots, X(8*i+7, 1)$
  - ....
  - $X(8*i, 7), X(8*i+1, 7), \dots, X(8*i+7, 7)$

To select sequential block format, write 0x2 in the `DCT_CFG[2:1]` FMT bit field.

### 8.4.5.4 ISS SIMCOP DCT Basic Programming Model

#### 8.4.5.4.1 Initialization of Surrounding Modules

To initialize the DCT, the surrounding modules must be initialized. [Table 8-1348](#) lists these modules.

**Table 8-1348. Initialization of Surrounding Modules**

Module	Minimum Required Setting	Optional Settings
CLKC	Enable the SIMCOP clock.	
Buffers	Configure DCT access to the image buffer.	
SIMCOP CTRL		Configure SIMCOP CTRL to handle DCT interrupts.

#### 8.4.5.4.2 Using the DCT for DCT Processing

To configure the DCT for DCT processing, software must follow the procedure listed in [Table 8-1349](#).

**Table 8-1349. DCT Processing Procedure**

Step	Register/Bit Field	Value
Prepare data in buffer memory.	SIMCOP buffers	
Switch buffer access to the DCT.	SIMCOP buffers	
Configure the start of the DCT input address (address must be in the memory buffer).	<a href="#">DCT_SPTR</a> [12:5] ADDR	Start address (spatial domain)
Configure the start of the DCT output address (address must be in the memory buffer).	<a href="#">DCT_FPTR</a> [13:4] ADDR	Start address (frequency domain)
Set the <a href="#">DCT_CFG</a> register.	<a href="#">DCT_CFG</a> [0] MODE	0 (DCT mode)
	<a href="#">DCT_CFG</a> [2:1] FMT	Depends on data format
	<a href="#">DCT_CFG</a> [3] INTEN	0: No interrupt 1: Interrupt at end of processing
	<a href="#">DCT_CFG</a> [4] TRIG_SRC	1: Start controlled by hardware signal 0: Start controlled by a register
	<a href="#">DCT_CFG</a> [13:8] NMCUS	Number of MCUs to process
<b>IF:</b> <a href="#">DCT_CFG</a> [4] TRIG_SRC = 0		
Start DCT through the register.	<a href="#">DCT_CTRL</a> [0] EN	1
<b>ELSE</b>		
Start the DCT using the hardware sequencer.	SIMCOP hardware sequencer	
<b>ENDIF</b>		

The DCT is now processing the blocks. To detect the end of DCT processing:

- If [DCT\\_CFG](#)[3] INTEN = 1, wait for the interrupt signal.
- If [DCT\\_CFG](#)[3] INTEN = 0, the processor must poll the [DCT\\_CTRL](#)[15] BUSY bit. When the BUSY bit is 0, processing is complete.

When DCT processing completes, the frequency output result is stored in the memory buffer at the address given by the [DCT\\_FPTR](#)[13:4] ADDR bit field.

#### 8.4.5.4.3 Using the DCT for IDCT Processing

To configure the DCT for IDCT processing, software must follow the procedure listed in [Table 8-1350](#).

**Table 8-1350. IDCT Processing Procedure**

Step	Register/Bit Field	Value
Prepare data in buffer memory.	SIMCOP buffers	
Switch buffer access to the DCT.	SIMCOP buffers	
Configure the start of the DCT output address (address must be in the memory buffer).	<a href="#">DCT_SPTR</a> [12:5] ADDR	Start address (spatial domain)

**Table 8-1350. IDCT Processing Procedure (continued)**

Step	Register/Bit Field	Value
Configure the start of the DCT input address (address must be in the memory buffer).	DCT_FPTR[13:4] ADDR	Start address (frequency domain)
Set the DCT_CFG register.	DCT_CFG[0] MODE	1 (IDCT mode)
	DCT_CFG[2:1] FMT	Depends on data format
	DCT_CFG[3] INTEN	0: No interrupt 1: Interrupt at end of processing
	DCT_CFG[4] TRIG_SRC	1: Start controlled by a hardware signal 0: Start controlled by a register
	DCT_CFG[13:8] NMCUS	Number of MCUs to process
<b>IF:</b> DCT_CFG[4] TRIG_SRC = 0		
Start the DCT through a register.	DCT_CTRL[0] EN	1
<b>ELSE</b>		
Start the DCT using hardware sequencer.	SIMCOP hardware sequencer	
<b>ENDIF</b>		

The DCT is now processing the blocks. To detect the end of IDCT processing:

- If DCT\_CFG[3] INTEN = 1, wait for the interrupt signal.
- If DCT\_CFG[3] INTEN = 0, the processor must poll the DCT\_CTRL[15] BUSY bit. When the BUSY bit is 0, processing is complete.

When IDCT processing completes, the spatial output result is stored in the memory buffer at the address given by the DCT\_SPTR[12:5] ADDR bit field.

### 8.4.5.5 ISS SIMCOP DCT Register Manual

#### 8.4.5.5.1 ISS SIMCOP DCT Instance Summary

Table 8-1351 summarizes the DCT instance.

**Table 8-1351. ISS SIMCOP DCT Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-M3 Private Access	Size
DCT	0x5202 0800	0x5506 0800	32 bytes

**NOTE:** Private Access is an access which is not using the L3/L4 interconnects.

#### 8.4.5.5.2 DCT Registers

##### 8.4.5.5.2.1 DCT Register Summary

Table 8-1352 summarizes the DCT register mapping.

**Table 8-1352. DCT Register Mapping Summary**

Register	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-M3 Private Access
<a href="#">DCT_VERSION</a>	R	32	0x0000 0000	0x5202 0800	0x5506 0800
<a href="#">DCT_CTRL</a>	RW	32	0x0000 0004	0x5202 0804	0x5506 0804
<a href="#">DCT_CFG</a>	RW	32	0x0000 0008	0x5202 0808	0x5506 0808
<a href="#">DCT_SPTR</a>	RW	32	0x0000 000C	0x5202 080C	0x5506 080C
<a href="#">DCT_FPTR</a>	RW	32	0x0000 0010	0x5202 0810	0x5506 0810



8.4.5.5.2.2 DCT Register Descriptions

Table 8-1353 through Table 8-1361 describe the registers in detail.

Table 8-1353. DCT\_VERSION

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See Table 8-1352.	<b>Instance</b>	DCT
<b>Description</b>	IP Revision		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

Table 8-1354. Register Call Summary for Register DCT\_VERSION

- ISS SIMCOP DCT Register Manual
- [ISS SIMCOP DCT Register Summary: \[0\]](#)

Table 8-1355. DCT\_CTRL

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See Table 8-1352.	<b>Instance</b>	DCT
<b>Description</b>	DCT control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUSY	RESERVED										EN				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Read returns 0.	R	0x0000
15	BUSY	IDCT/busy status 0: Idle 1: Busy	R	0x0
14:1	RESERVED	Read returns 0.	R	0x0000
0	EN	Write 1 when <a href="#">DCT_CFG[4]</a> TRIG_SRC = 0 to start module operation. Read returns 0.	W	0x0

Table 8-1356. Register Call Summary for Register DCT\_CTRL

- ISS SIMCOP DCT Functional Description
- [Control and Status: \[0\] \[1\]](#)
- ISS SIMCOP DCT Basic Programming Model
- [Using the DCT for DCT Processing: \[2\] \[3\]](#)
  - [Using the DCT for IDCT Processing: \[4\] \[5\]](#)
- ISS SIMCOP DCT Register Manual
- [ISS SIMCOP DCT Register Summary: \[6\]](#)

**Table 8-1357. DCT\_CFG**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	DCT
<b>Physical Address</b>	See <a href="#">Table 8-1352</a> .		
<b>Description</b>	DCT configuration register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NMCUS						RESERVED	AUTOGATING	TRIG_SRC	INTEN	FMT	MODE				

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Read returns 0.	R	0x00000
13:8	NMCUS	Number of MCUs (for FMT = 0, 1) or blocks (for FMT = 2) 0 = 1 MCU or block 1 = 2 MCUs or blocks ... 63 = 64 MCUs or blocks	RW	0x00
7:6	RESERVED	Read returns 0.	R	0x0
5	AUTOGATING	Internal clock gating on interface and functional clocks 0: Clocks are free-running 1: Clocks are gated off in subblocks that are not required for operation.	RW	0x0
4	TRIG_SRC	Trigger source 0: Memory mapper register 1: Hardware start signal	RW	0x0
3	INTEN	0: Interrupt disabled 1: Interrupt enabled	RW	0x0
2:1	FMT	Data format 0: YUV4:2:0 format 1: YUV4:2:2 format 2: Sequential blocks format 3: Reserved	RW	0x0
0	MODE	0: DCT 1: IDCT	RW	0x0

**Table 8-1358. Register Call Summary for Register DCT\_CFG**

## ISS SIMCOP DCT Functional Description

- [ISS SIMCOP DCT Power Management: \[0\]](#)
- [ISS SIMCOP DCT Interrupt Requests: \[1\]](#)
- [Control and Status: \[2\] \[3\] \[4\]](#)
- [Configuration: \[5\] \[6\]](#)
- [Data Format: \[7\]](#)
- [YUV4:2:0 Format: \[8\] \[9\] \[10\]](#)
- [YUV4:2:2 Format: \[11\] \[12\]](#)
- [Sequential Block Format: \[13\]](#)

## ISS SIMCOP DCT Basic Programming Model

- [Using the DCT for DCT Processing: \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\]](#)
- [Using the DCT for IDCT Processing: \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)

## ISS SIMCOP DCT Register Manual

- [ISS SIMCOP DCT Register Summary: \[32\]](#)
- [ISS SIMCOP DCT Register Manual: \[33\]](#)

**Table 8-1359. DCT\_SPTR**

<b>Address Offset</b>	0x0000 000C		
<b>Physical Address</b>	See <a href="#">Table 8-1352</a> .	<b>Instance</b>	DCT
<b>Description</b>	Spatial-domain data pointer, byte address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDR								RESERVED							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Read returns 0.	R	0x00000
12:5	ADDR	Address in 256-bit words Intention is that software write a byte address into the register. Hardware ignores the lowest 5 bits and bits 12..5 specifies the 256-bit/word memory address..	RW	0x00
4:0	RESERVED	Read returns 0.	R	0x00

**Table 8-1360. Register Call Summary for Register DCT\_SPTR**

ISS SIMCOP DCT Functional Description

- [Control and Status: \[0\]](#)
- [YUV4:2:0 Format: \[1\]](#)

ISS SIMCOP DCT Basic Programming Model

- [Using the DCT for DCT Processing: \[2\]](#)
- [Using the DCT for IDCT Processing: \[3\] \[4\]](#)

ISS SIMCOP DCT Register Manual

- [ISS SIMCOP DCT Register Summary: \[5\]](#)

**Table 8-1361. DCT\_FPTR**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 8-1352</a> .	<b>Instance</b>	DCT
<b>Description</b>	Frequency-domain data pointer, byte address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDR								RESERVED							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Read returns 0.	R	0x000000
13:4	ADDR	Address in 128-bit words. Intention is that software write a byte address into the register. Hardware ignores the lowest 4 bits and bits 13..4 specifies the 128-bit/word memory address.	RW	0x000
3:0	RESERVED	Read returns 0.	R	0x0

**Table 8-1362. Register Call Summary for Register DCT\_FPTR**

ISS SIMCOP DCT Functional Description

- [Control and Status: \[0\]](#)

ISS SIMCOP DCT Basic Programming Model

- [Using the DCT for DCT Processing: \[1\] \[2\]](#)
- [Using the DCT for IDCT Processing: \[3\]](#)

---

**Table 8-1362. Register Call Summary for Register DCT\_FPTR (continued)**

---

ISS SIMCOP DCT Register Manual

- [ISS SIMCOP DCT Register Summary: \[4\]](#)
- 

PRELIMINARY

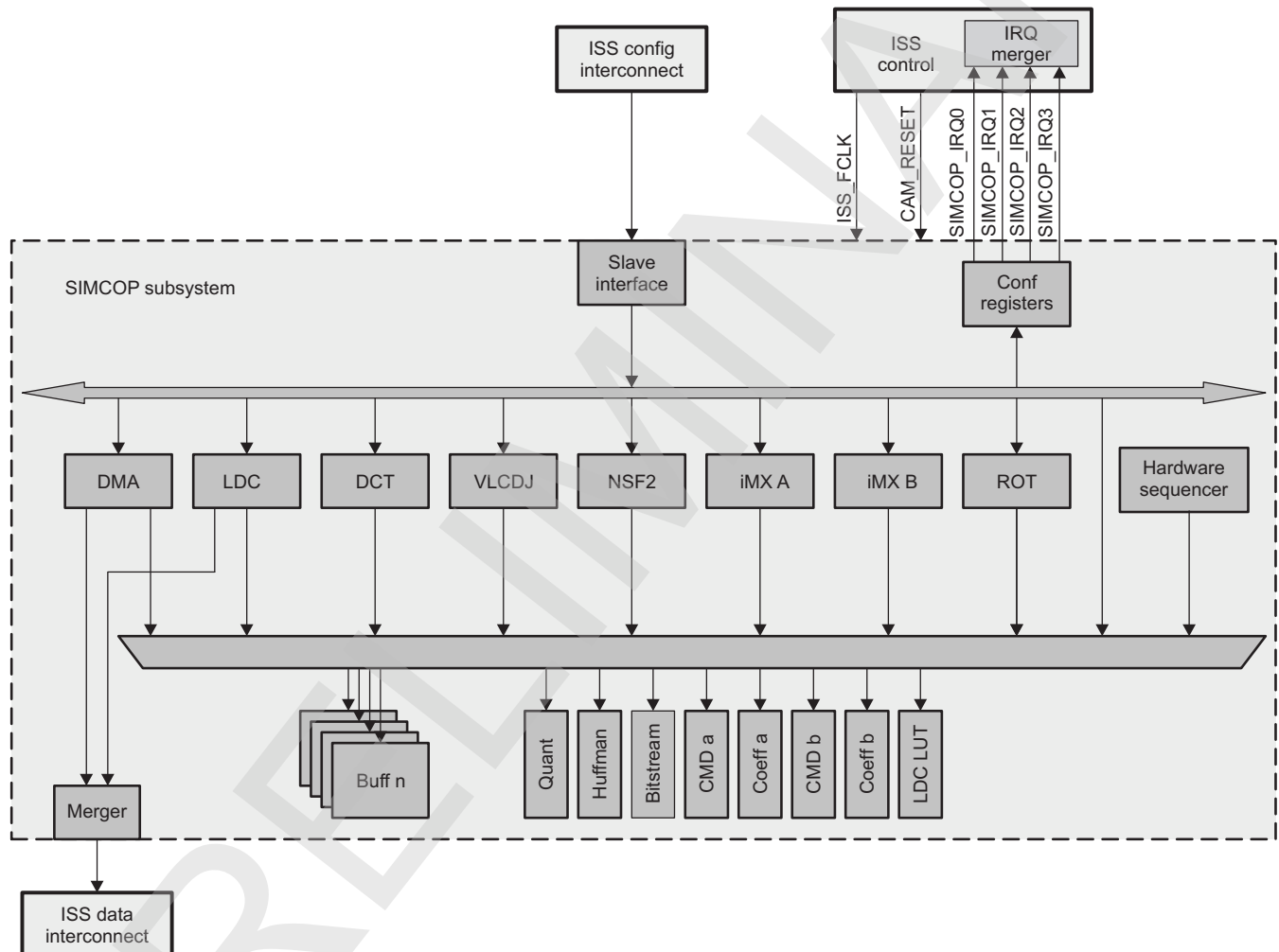
### 8.4.6 ISS SIMCOP Variable Length Coder/Decoder for JPEG Module

This section describes the variable-length coder/decoder for JPEG (VLCDJ) module in the still-image coprocessor (SIMCOP) subsystem.

#### 8.4.6.1 ISS SIMCOP VLCDJ Overview

The VLCDJ module handles quantization (Q) and variable-length coding (VLC) in JPEG encoding, and variable-length decoding (VLD) and inverse quantization (IQ) in JPEG decoding. These operations are required for JPEG image compression and decompression. Figure 8-269 is an overview of the VLCDJ in the SIMCOP subsystem.

Figure 8-269. VLCDJ in the SIMCOP Subsystem



vlcdj-001

The VLCDJ supports the following operations:

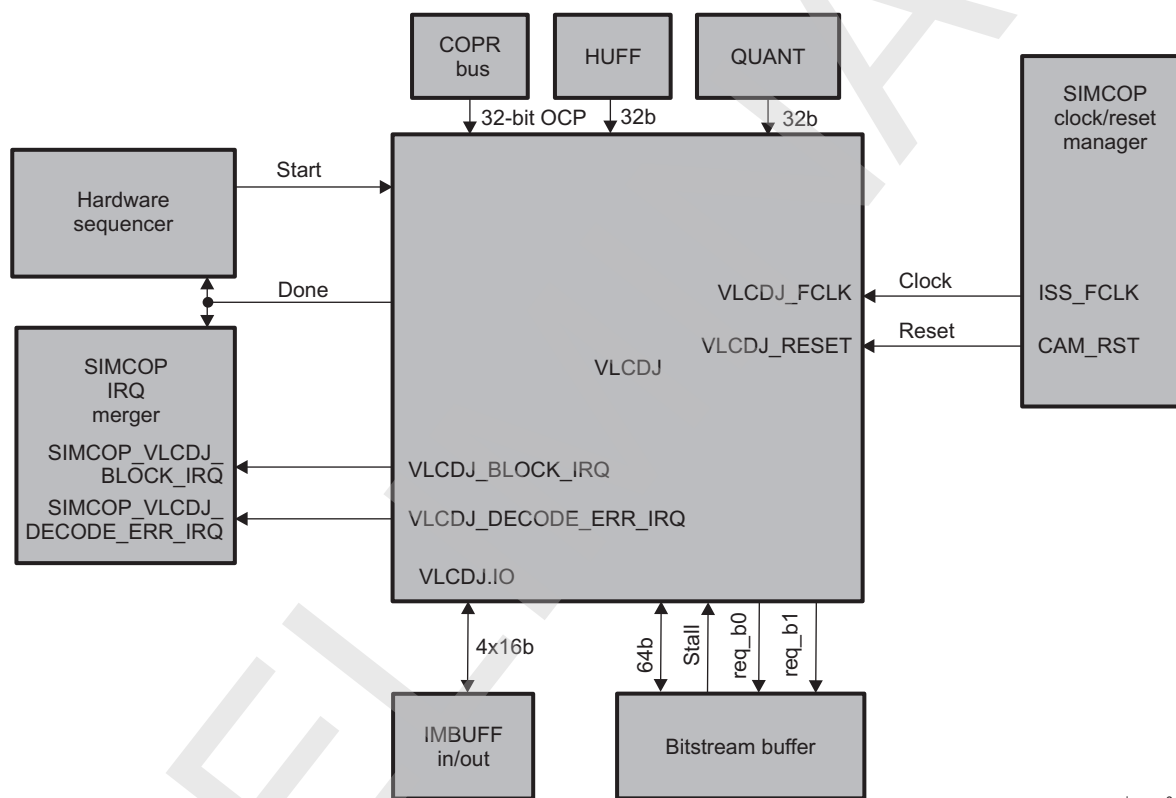
- Q and VLC for JPEG encoding
- VLD and IQ for JPEG decoding
- Data formats:
  - YUV4:2:0 format in 16 × 16-pixel microcontroller unit (MCU)
  - YUV4:2:2 format in 16 × 8-pixel MCU
  - Sequential block mode to support arbitrary color component configuration within JPEG baseline sequential (software intervention required to context-switch among color components)
- Configurable for number of MCUs per initiation (up to 21 MCUs for YUV4:2:0, 32 MCUs for YUV4:2:2, 64 8 × 8 blocks for sequential block mode)

- Discrete cosine transform (DCT) coefficients are stored sequentially for each 8 × 8 block, 16 bits per data point
- Stall processing for interface with bitstream data overrun/underrun protection
- Insertion of restart markers during encoding (YUV4:2:0, YUV4:2:2, sequential block modes)
- Record of restart marker locations during encoding (YUV4:2:0, YUV4:2:2 only)
- Processing of restart markers on the MCU boundary during decoding, reset of direct coefficient (DC) predictors as required by the JPEG standard (YUV4:2:0, YUV4:2:2, sequential block modes)

### 8.4.6.2 ISS SIMCOP VLCDJ Integration

The VLCDJ is part of the SIMCOP subsystem in the imaging subsystem (ISS). Figure 8-270 shows the integration of the VLCDJ in the SIMCOP subsystem.

Figure 8-270. VLCDJ Engine Integration



simcop-013

Table 8-1363 through Table 8-1365 list the integration attributes, clocks and resets, and hardware requests, respectively, for the VLCDJ.

Table 8-1363. Integration Attributes

Module Instance	Attributes
	Power Domain
VLCDJ	PD_CAM

Table 8-1364. Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description

**Table 8-1364. Clocks and Resets (continued)**

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
VLCDJ	VLCDJ_FCLK	ISS_FCLK	PRCM	Functional clock provided by ISS_CLK from the power, reset, and clock management (PRCM) module. It is used by all ISS submodules and ISS top-level resources.
<b>Resets</b>				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
VLCDJ	VLCDJ_RESET	CAM_RST	PRCM	ISS and SIMCOP global reset

For information about clock and reset management, see [Section 8.4.1.2.1](#), *ISS SIMCOP Local Power and Clock Management*.

**Table 8-1365. Hardware Requests**

<b>Interrupt Requests</b>				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
VLCDJ	VLCDJ_BLOCK_IRQ	SIMCOP_VLCDJ_BLOCK_IRQ	SIMCOP IRQ merger	A macroblock has been processed.
VLCDJ	VLCDJ_DECODE_ERROR_IRQ	SIMCOP_VLCDJ_DECODE_ERROR_IRQ	SIMCOP IRQ merger	A decode error has been signaled by the VLCDJ.

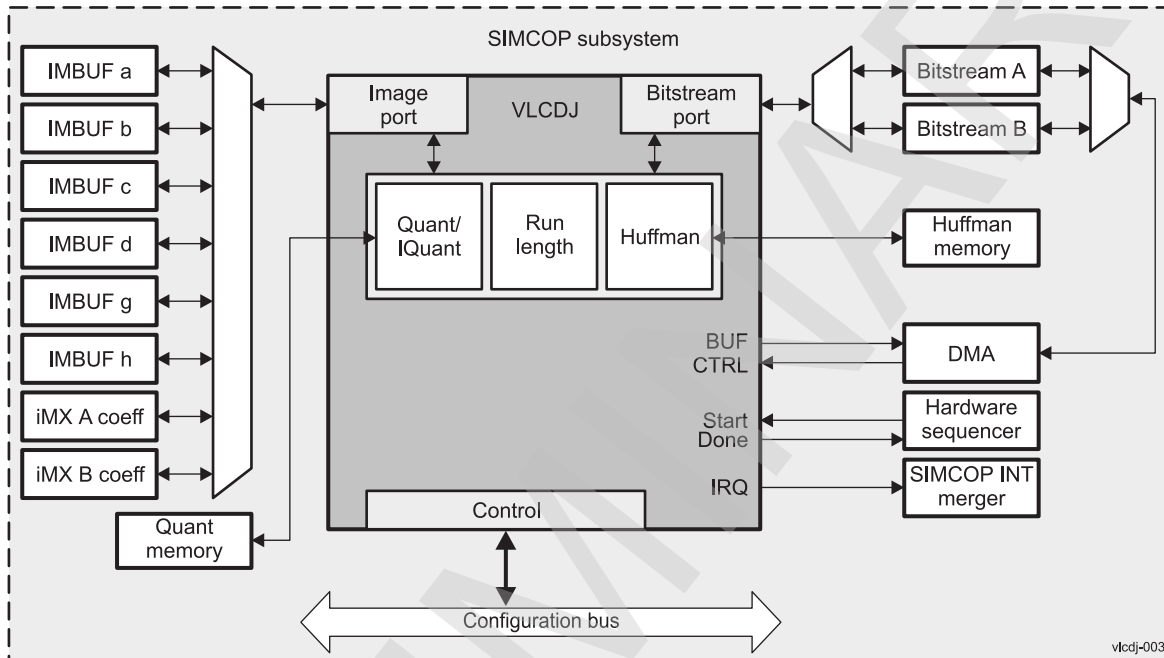
For more information about interrupt requests, see [Section 8.4.1.2.3](#), *Interrupt Merger*.

### 8.4.6.3 ISS SIMCOP VLCDJ Functional Description

#### 8.4.6.3.1 ISS SIMCOP VLCDJ Block Diagram

The VLCDJ implements quantization, run-length encode, and Huffman encode stages of JPEG compression, and Huffman decode, run-length expansion, and dequantization stages of JPEG decompression. [Figure 8-271](#) is a block diagram of the VLCDJ with connected memories and modules.

**Figure 8-271. VLCDJ Block Diagram**



#### 8.4.6.3.2 ISS SIMCOP VLCDJ Power Management

The VLCDJ uses clock autogating to conserve power. When any substantial subblocks are not in use, the corresponding clock trees are gated off. [Table 8-1366](#) describes the power-management feature available for the VLCDJ.

**Table 8-1366. Local Power-Management Feature**

Feature	Registers	Description
Clock autogating	<a href="#">VLCDJ_CTRL[3]</a> AUTOGATING	Free-running or autogating modes are available.

#### 8.4.6.3.3 ISS SIMCOP VLCDJ Interrupt Requests

[Table 8-1367](#) lists the event flags and their masks that can cause module interrupts.

**Table 8-1367. Events**

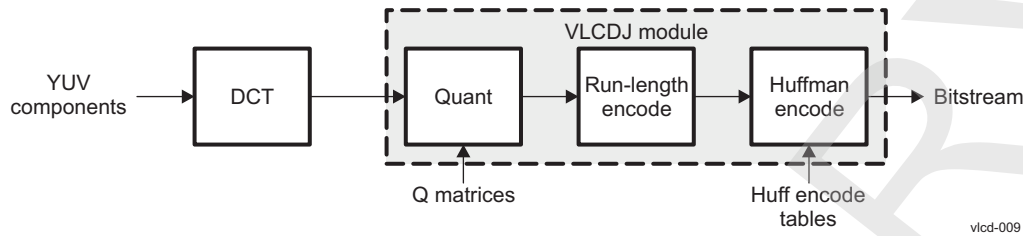
Event Flag	Event Mask	Map to	Description
N/A	<a href="#">VLCDJ_CTRL[4]</a> INTEN_DONE	VLCDJ_BLOCK_IRQ	End of block processing
N/A	<a href="#">VLCDJ_CTRL[5]</a> INTEN_ERR	VLCDJ_DECODE_ER R_IRQ	Error occurred



**8.4.6.3.4 ISS SIMCOP VLCDJ in JPEG Encoding Flow**

Figure 8-272 shows the JPEG compression or encoding flow.

**Figure 8-272. JPEG Encoding Flow**



The DCT coefficients are quantized by dividing each data point with a quantizer. The quantizers form an 8 × 8 matrix for Luma Y, and another 8 × 8 matrix for U and V, and the matrices are fixed in each image. The quantizer matrices can vary between images, and are sometimes fine-tuned in a multiple-pass process to regulate the compressed JPEG file size.

For quantization, the VLCDJ reads a reciprocal of the quantizer values (16 bits per entry carrying  $2^{15}/\text{quant}$ ) so that hardware performs multiplications rather than divisions. The result is rounded to the nearest integer.

After quantization, the DC terms (the first term in each 8 × 8 block of DCT coefficients) are differentially coded within each color. Thus, three predictors are used for the common formats.

Next, the coefficients are scanned in a zigzag order (the same order for all colors) to form DC term + zero-run and nonzero level pairs for the alternative coefficient (AC) terms. The run-level pairs are then coded with a Huffman table.

The DC level is first categorized into size groups, basically the number of bits that represents the absolute value of the level. This size is translated to a bit segment using the Huffman table. Then, additional bits are appended to represent the exact DC level.

For AC run-level pairs, the process is similar. Each AC level is first categorized into size groups, basically the number of bits that represents the absolute value of the level. The size and the run are combined and translated to a bit segment using the Huffman table. Then, additional bits are appended to represent the exact AC level.

The number of additional bits to encode after the Huffman-coded segment, for DC or AC level, equals the size value. For example, coefficient values -3, -2, and 3 are of size 2, because 2 bits are required to represent the absolute values 2 and 3. There are four coefficient values in the size -2 category, so 2 additional bits are required to uniquely identify each value.

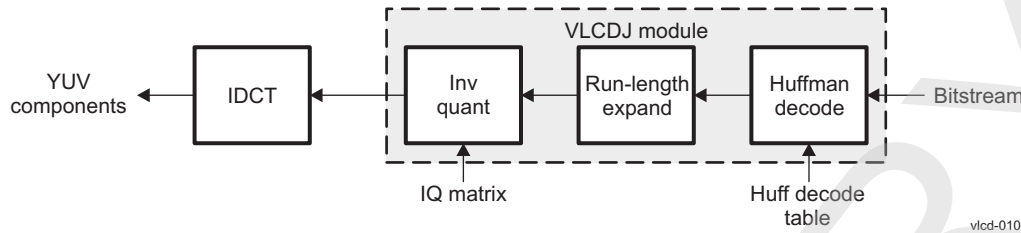
Some special symbols go with the AC run/size pairs. The ZRL symbol represents a run of 15 zeros, and is used to code unusually long zero runs. The EOB symbol represents end of block, meaning that there are only 0 AC coefficients at the end.

In the bitstream management level, after each byte of 0xFF, a byte of 0 is inserted to prevent the decoder from detecting special markers in the bitstream.

Because the VLCDJ handles multiple MCUs and allows restart marker insertion in MCU boundaries inside a task, the inserted flags are flagged to prevent the insertion of a 0 byte after 0xFF.

**8.4.6.3.5 ISS SIMCOP VLCDJ in JPEG Decoding Flow**

Decoding or decompression operates in the reverse order, as shown in Figure 8-273.

**Figure 8-273. JPEG Decoding Flow**

1. In decoding, in the MCU data section, when the VLCDJ recognizes the 0xFF byte in the bitstream, it detects whether the next byte is a 0 byte. If it is, the 0 byte is removed and decoding continues as if the 0 byte is not there (the 0xFF byte is kept). If the next byte is not a 0 byte, the VLCDJ flags this as a JPEG data error. This can be caused by an earlier bitstream error that causes the module to parse outside the MCU entropy data and into markers that are not supported by the module.
2. Because the VLCDJ decodes for multiple MCUs, restart markers can occur in MCU boundaries. Restart marker detection and processing work with the 0 byte removal logic to ensure that restart markers are detected and removed correctly. When the VLCDJ detects a restart marker, it resets the DC predictors in the IQ stage.
3. The VLCDJ looks up the Huffman table to convert a variable-length bit segment into a DC size or an AC run/size pair. The number of additional bits equal to the size is extracted and the DC/AC coefficient is reconstructed. Then, zero runs are expanded.
4. IQ is performed, which is simply multiplication of the coefficients with values in the quantizer matrices.

#### 8.4.6.3.6 Memory Interfaces

The VLCDJ has four memory interfaces (all memories are inside the SIMCOP subsystem):

- Image buffer (input DCT coefficients for encoding and output DCT coefficients for decoding)
- Quantizer memory
- Huffman memory
- Bitstream memory (output for encoding and input for decoding)

##### 8.4.6.3.6.1 Image Buffer

An image buffer stores DCT data in the context of the VLCDJ. This is an input in encode mode and an output in decode mode.

When multiple MCUs are configured, DCT coefficients for the MCUs are stored sequentially (MCU 0 is stored first, followed by MCU 1, and so on).

Blocks in an MCU are stored sequentially; that is, block 0 first, block 1 next, and so on. There are six blocks per MCU in YUV4:2:0 format, {Y0, Y1, Y2, Y3, U, V}, and four blocks per MCU in YUV4:2:2 format, {Y0, Y1, U, V}.

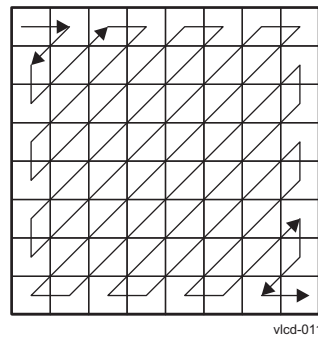
The 64 DCT coefficients for each block are stored in an 8 × 8 transposed format (the companion DCT module handles the transposed format). Each coefficient, up to [−2047, 2047] in range is stored on a 16-bit word. [Table 8-1368](#) lists the data format in each 8 × 8 block of coefficients.

**Table 8-1368. Image Buffer Block Format**

Byte Offset								
0	1	2	3	4	5	...	14	15
c(0, 0)		c(0, 1)		c(0, 2)		...		c(0, 7)
c(1, 0)		c(1, 1)		c(1, 2)		...		c(1, 7)
c(2, 0)		c(2, 1)		c(2, 2)		...		c(2, 7)
:								
c(7, 0)		c(7, 1)		c(7, 2)		...		c(7, 7)

The DCT coefficients are processed in zigzag order (see [Figure 8-274](#)), as required by JPEG.

**Figure 8-274. JPEG Zigzag Order**



**8.4.6.3.6.2 Quantizer Memory**

The quantization memory stores the reciprocal of quantizers,  $2^{15}/qmat[i][j]$  so that hardware performs multiplications rather than divisions. Each data point is stored in 16 bits.

The quantizer matrices have a fixed size, 64 16-bit elements each. The quantizer memory (QMEM) is sized to hold up to four matrices ([Table 8-1369](#) lists the recommended allocation for YUV4:2:0 and YUV4:2:2).

**Table 8-1369. Recommended QMEM Allocation for YUV4:2:0 and YUV4:2:2**

Quadrant of QMEM	Address Offset (Bytes)	Contents
00	0x000	(Encode) Quantizer reciprocal for Y ( $2^{15}/qmat[i][j]$ )
01	0x080	(Encode) Quantizer reciprocal for UV ( $2^{15}/qmat[i][j]$ )
10	0x100	(Decode) Quantizer for Y ( $qmat[i][j]$ )
11	0x180	(Decode) Quantizer for UV ( $qmat[i][j]$ )

Each matrix is stored in zigzag order to make quantization and inverse quantization more efficient.

**8.4.6.3.6.3 Huffman Memory**

Huffman memories store Huffman encode tables and Huffman control/decode tables. For information about Huffman tables, see [Section 8.4.6.3.7, Huffman Table Organization](#).

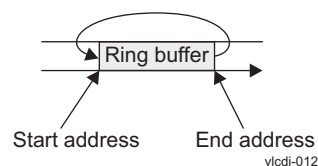
**8.4.6.3.6.4 Bitstream Memory**

Bitstream memory is for input/output (I/O) bitstream. Bitstream data is stored in little-endian format in the memory. Circular buffer (CBUFF) start and end addresses are provided in registers to facilitate bitstream data management:

- [VLCDJE\\_CBUF\[11:10\]](#) START, start of ring buffer for encode mode (address must be 1KB aligned)
- [VLCDJE\\_CBUF\[27:26\]](#) END, end of ring buffer for encode mode (address must be 1KB aligned)
- [VLCDJD\\_CBUF\[11:10\]](#) START, start of ring buffer for decode mode (address must be 1KB aligned)
- [VLCDJD\\_CBUF\[27:26\]](#) END, end of ring buffer for decode mode (address must be 1KB aligned)

[Figure 8-275](#) shows the start and end addresses of a ring buffer.

**Figure 8-275. VLCDJ Ring Buffer**



In encoding and decoding, a bit-level boundary defines bitstream data written (for encoding) or parsed (for decoding). To point to that boundary, bit and byte pointers are provided in the registers:

- `VLCDJE_BSPTR[19:16]` BITPTR (encoding)
- `VLCDJE_BSPTR[11:0]` BYTEPTR (encoding)
- `VLCDJD_BSPTR[19:16]` BITPTR (decoding)
- `VLCDJD_BSPTR[11:0]` BYTEPTR (decoding)

BYTEPTR (byte pointer) points to the byte where the boundary lies, and BITPTR (bit pointer) specifies the number of available vacant data bits (for encoding) or remaining data bits (for decoding). BITPTR is always in the range of 1 to 8, where 8 indicates that an entire byte is available for writing or reading.

For example, for encoding, if BYTEPTR = 0 and BITPTR = 8, after the hardware encodes a specified number of MCUs that together generate 30 bits, or 3 full bytes and 6 bits, BYTEPTR = 3 and BITPTR = 2. The JPEG standard requires that at the end of MCU data, any remaining bits of the last byte are padded with 1s. For example, if the last byte has 4 bits written 0101, the last byte of MCU data in the JPEG data is 0x5F.

### 8.4.6.3.7 Huffman Table Organization

#### 8.4.6.3.7.1 Huffman Table for Encode Mode

Because the Huffman encode tables are compact, they are fixed relative to a common pointer:

- The Luma DC table starts at `VLCDJE_VLCTBL[11:2] ADDR` (12 entries  $\times$  4 = 48 bytes)
- The Luma AC table starts at `VLCDJE_VLCTBL[11:2] ADDR + 48 bytes` (11  $\times$  16  $\times$  4 = 704 bytes)
- The Chroma DC table starts at `VLCDJE_VLCTBL[11:2] ADDR + 752 bytes` (12 entries  $\times$  4 = 48 bytes)
- The Chroma AC table starts at `VLCDJE_VLCTBL[11:2] ADDR + 800 bytes` (11  $\times$  16  $\times$  4 = 704 bytes)

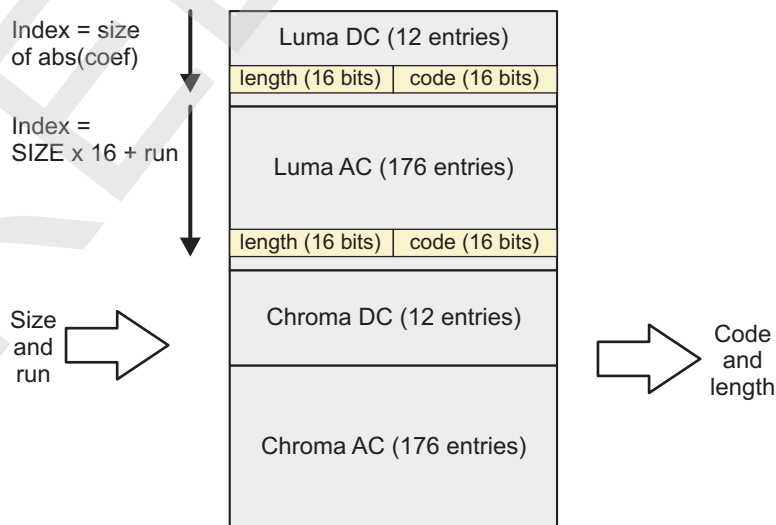
For DC tables, the index of each table corresponds to the size of the absolute value of the DC coefficient.

For AC tables, the index of each table corresponds to a formula including the size (of absolute value of nonzero AC coefficients) and the zero run:  $\text{Index} = \text{SIZE} \times 16 + \text{run}$ . Because the size cannot be 0, the first 16 entries hold special symbols. EOB is at entry 0, and ZRL is at entry 15.

The size of a nonnegative value is the number of bits required to represent the value (for example, a size of 4..7 = 3).

Figure 8-276 shows the organization of the Huffman encode table.

**Figure 8-276. Huffman Encode Table Organization**



vlcdj-002

### **8.4.6.3.8 Restart Marker**

#### **8.4.6.3.8.1 Restart Marker Insertion for Encode**

As specified in the JPEG standard, the VLCDJ can insert a restart marker at a specified MCU interval (for information about restart markers, see [Section 8.4.6.3.5, ISS SIMCOP VLDCJ in JPEG Decoding Flow](#)). Any remaining bits in the last partial byte are padded with 1s.

When enabled, the VLCDJ inserts a restart marker at the end of a group of MCUs. A partial byte is automatically padded with 1s.

#### **8.4.6.3.8.2 Restart Marker Location Recording**

To facilitate reordering of compressed JPEG data sections to achieve JPEG rotation, the VLCDJ also records the restart marker locations.

#### **8.4.6.3.8.3 Restart Marker Handling for Decoding**

Restart markers are handled automatically during MCU data decoding according to the JPEG standard. The DC predictors are reset at the MCU boundary where restart markers are present.

Because the handling of restart markers is complex, there is no check of the consistency of restart marker value or existence/nonexistence with the restart interval (which is not part of the decoder register setting). Correct JPEG files are decoded correctly. Incorrect JPEG files can escape error detection.

To decode correctly, in spite of the presence of restart markers, software must configure the VLCDJ with the correct byte pointer for the VLCDJ. To reset DC predictors to occur before an MCU, the byte pointer must point to the restart marker before the MCU data in the bitstream data. In other words, the VLCDJ does not search for the correct marker and resume MCU decoding after the marker. The VLCDJ detects the presence of the marker only at the MCU boundaries.

#### **8.4.6.3.9 Encode/Decode Context Switching**

The bitstream memory is organized as two banks of memory. When the VLCDJ performs only encoding or only decoding, the two banks are used to achieve VLCDJ and DMA concurrence, and automatic bitstream data DMA is supported.

When the VLCDJ performs one encode and one decode concurrently, both contexts can coexist in the configuration registers, because encode and decode configurations are in separate registers. The bitstream memory must be partitioned half-and-half between encode and decode. For example:

- `VLCDJE_CBUF[11:10] START = 0x0`
- `VLCDJE_CBUF[27:26] END = 0x7FF`
- `VLCDJD_CBUF[11:10] START = 0x800`
- `VLCDJD_CBUF[27:26] END = 0xFFF`

The byte pointers and the bitstream data DMA must be set up accordingly.

When the VLCDJ is encoding, the decode half of bitstream memory can be refilled by DMA. When the VLCDJ is decoding, the encode half of bitstream memory can be transferred out by DMA. Hardware does not support automatic triggering of DMA; software must submit DMA based on the byte pointer value of each context. For good SDRAM efficiency, the DMA block must be aligned correctly.

#### **8.4.6.3.10 Using Sequential Block Mode for YUV4:4:4 Encoding/Decoding**

In sequential block mode, because it deals with only one color component, the VLCDJ does the following:

- Uses DCPREDY for the DC predictor
- Uses the first quantizer matrix pointed by QM/QMR
- Starts from the first block pointed to by DCT
- Uses the first VLC/VLD table

Thus, to encode/decode YUV4:4:4 format using sequential block mode, software must switch context among Y, U, and V between initiations of the VLCDJ. Memory organization and registers are designed so that QM/QMR, DCT, and VLC/VLD tables can be swapped by changing the corresponding pointers in the registers. The DC predictor must be saved and restored.

For more information, see [Section 8.4.6.4.6](#), *Example of YUV4:4:4 Encoding*.

#### 8.4.6.3.11 Error Reporting

The VLCDJ can report the following decode errors:

- Invalid Huffman code
- Block exceeding 64 coefficients

---

**NOTE:** Decoding of an 0xFF byte followed by a nonzero byte is not reported as an error (for more information, see [Section 8.4.6.3.5](#), *ISS SIMCOP VLCDJ in JPEG Decoding Flow*).

---

When there is a decode error, VLCDJ operation is suspended immediately. Bit and byte pointers to the bitstream near where the error occurs are reflected in the [VLCDJD\\_BSPTR\[19:16\]](#) BITPTR and [VLCDJD\\_BSPTR\[11:0\]](#) BYTEPTR bit fields. The DCT coefficient index, in zigzag order of the task, near where the error occurs, is also reflected in a read-only bit field, [VLCDJD\\_DCTERR\[13:0\]](#) ERRPTR. For example, if decoding fails at the fifth coefficient of the third block of the VLCDJ task (which can be multiple MCUs), ERRPTR returns  $2 \times 64 + 4 = 195$ .

The IQ stage of the hardware does not stop and runs to completion. When there is a decode error, the DC predictor and decoded DCT coefficient outcomes are indeterminate.

Because the VLCDJ does not start in the middle of an MCU, software must recover from the error and configure the VLCDJ from an MCU boundary.

### 8.4.6.4 ISS SIMCOP VLCDJ Basic Programming Model

#### 8.4.6.4.1 Initialization of Surrounding Modules

To initialize the VLCDJ, surrounding modules must be initialized. [Table 8-1370](#) lists these modules.

**Table 8-1370. Initialization of Surrounding Modules**

Module	Minimum Required Setting	Optional Settings
CLKC	Enable the SIMCOP clock.	
Buffers	Configure VLCDJ access to the image buffer.	
SIMCOP CTRL		Configure SIMCOP CTRL to handle VLCDJ interrupts.

#### 8.4.6.4.2 ISS SIMCOP VLCDJ Start and Register Modification During Processing

The module operation can be started with two mechanisms, depending on the status of the [VLCDJ\\_CTRL\[2\] TRIG\\_SRC](#) bit.

When the [VLCDJ\\_CTRL\[2\] TRIG\\_SRC](#) bit is set to 0 and the module is idle (the [VLCDJ\\_CTRL\[16\] BUSY](#) bit is set to 0), the module is started by setting the [VLCDJ\\_CTRL\[0\] EN](#) bit to 1. Any hardware start pulses are ignored. To simplify software control, in the same register write, changing the [VLCDJ\\_CTRL\[2\] TRIG\\_SRC](#) bit from 1 to 0 and setting the [VLCDJ\\_CTRL\[0\] EN](#) bit to 1 is recognized as a valid starting condition.

When the [VLCDJ\\_CTRL\[2\] TRIG\\_SRC](#) bit is set to 1 and the module is idle (the [VLCDJ\\_CTRL\[16\] BUSY](#) bit is set to 0), the module is started by the START signal going active. Any software write to the [VLCDJ\\_CTRL\[0\] EN](#) bit is ignored.

During encoding, encode configuration registers must not be modified by software, while decode configuration registers can be freely modified.

During decoding, decode configuration registers must not be modified by software, while encode configuration registers can be freely modified.

#### 8.4.6.4.3 JPEG Encoding

For JPEG encoding of a YUV4:2:0 image consisting of eight MCUs per task, software first composes the JPEG header, which includes image dimension, YUV4:2:0 format, Huffman tables, quantizer matrices, and the start-of-scan marker and scan header, all before the MCU data. Software then sets up the encoding task, once per image, with the procedure listed in [Table 8-1371](#).

**Table 8-1371. JPEG Encoding Procedure**

Step	Register/Bit Field	Value
Configure the encode configuration register.	<a href="#">VLCDJE_CFG[1:0] FMT</a>	0 (YUV4:2:0 mode)
	<a href="#">VLCDJE_CFG[13:8] NMCUS</a>	7 (eight MCUs)
	<a href="#">VLCDJE_CFG[2] RSTEN</a>	0 (no restart marker)
Configure the DC predictor.	<a href="#">VLCDJE_DCPREDY[11:0] PREDY</a>	0 (reset DC predictor for Y)
	<a href="#">VLCDJE_DCPREDU[11:0] PREDU</a>	0 (reset DC predictors for U)
	<a href="#">VLCDJE_DCPREDV[27:16] PREDV</a>	0 (reset DC predictors for V)
Configure the input and output memories.	<a href="#">VLCDJE_BSPTR[11:0] BYTEPTR</a>	0 (beginning of bitstream memory)
	<a href="#">VLCDJE_BSPTR[19:16] BITPTR</a>	8 (initially empty byte, thus 8 free bits)
	<a href="#">VLCDJE_CBUF[11:10] START</a>	0 (bitstream starts from beginning of first quadrant of bitstream memory)
	<a href="#">VLCDJE_CBUF[27:26] END</a>	3 (bitstream ends at end of fourth quadrant of bitstream memory)
	<a href="#">VLCDJE_DCTQM[13:4] DCT</a>	0 (DCT coefficients at beginning of image buffer)

**Table 8-1371. JPEG Encoding Procedure (continued)**

Step	Register/Bit Field	Value
Configure Huffman and quantizer memory pointers.	<a href="#">VLCDJE_DCTQM</a> [24:23] QMR	0 (quantizer matrix reciprocal at beginning of QMEM)
	<a href="#">VLCDJE_VLCTBL</a> [11:2] ADDR	0 (VLC table at beginning of HUFMEM)
Start encode with automatic ring buffer management.	<a href="#">VLCDJ_CTRL</a> [1] MODE	0 (encode mode)
	<a href="#">VLCDJ_CTRL</a> [0] EN	1 (start VLCDJ operation)

For each group of eight MCUs (the VLCDJ does not care if these MCUs cross multiple MCU rows), software sets the [VLCDJ\\_CTRL](#)[0] EN bit to 1 (restart VLCDJ operation).

Alternately, the hardware sequencer can start the VLCDJ. Usually, DMA is also configured to bring in YUV4:2:0 image data as input to DCT, and DMA, DCT, and VLCDJ are all put under control of the hardware sequencer.

Bitstream data is managed automatically between the VLCDJ and DMA throughout JPEG encoding of MCUs.

**NOTE:** Many registers change during the encoding of a frame, but software is not required to access these registers, because the VLCDJ manages the status information between tasks.

The last hardware task may require a smaller number of MCUs (the [VLCDJE\\_CFG](#)[13:8] NMCUS bit field) to adjust for the number of MCUs in the last group.

After the last VLCDJ task for the image, software must disable the automatic bitstream buffer management and prepare for the next image by setting the following values (these can be issued as a single write):

- [VLCDJ\\_CTRL](#)[6] RBEN = 0 (disable automatic DMA interaction)
- [VLCDJ\\_CTRL](#)[7] CLRRB = 1

Finally, software must pad the last data byte with bit 1s, then place an end-of-image marker after the last data byte. The [VLCDJE\\_BSPTR](#)[11:0] BYTEPTR and [VLCDJE\\_BSPTR](#)[19:16] BITPTR bit fields indicate the locations of the last data byte and bit, respectively. Software must then set up DMA to transfer out the remaining data in bitstream memory.

The number of MCUs processed by hardware per task must be set according to image buffer allocation. For example, with 8KB available for DCT coefficient memory, and YUV4:2:0 format, it is possible to process  $8K/(16 \times 16 \times 1.5 \times 2) = 10$  MCUs. Because the bitstream DMA is automatic, bitstream data does not overflow bitstream memory for this many MCUs in densely coded JPEG.

#### 8.4.6.4.4 JPEG Decoding

For JPEG decoding of an image, software first parses the JPEG header, which includes image dimension, YUV sampling, Huffman tables, quantizer matrices, and start-of-scan marker and scan header, before the MCU data. Software then sets up DMA to load bitstream data into the first half of bitstream memory.

The Huffman tables must be processed to generate hardware control tables and decode tables for a particular JPEG file, unless the Huffman tables match previously processed tables for which the hardware control and decode tables are known. The hardware control and decode tables must be transferred to HUFMEM.

If the JPEG file is YUV4:2:0, and image buffer memory is sufficient to process eight MCUs per hardware task, software must set up the decode task, once per image, with the procedure listed in [Table 8-1372](#).

**Table 8-1372. JPEG Decoding Procedure**

Step	Register/Bit Field	Value
Configure the decode configuration register.	<a href="#">VLCDJD_CFG</a> [1:0] FMT	0 (YUV4:2:0 mode)
	<a href="#">VLCDJD_CFG</a> [13:8] NMCUS	7 (eight MCUs)



**Table 8-1372. JPEG Decoding Procedure (continued)**

Step	Register/Bit Field	Value	
Configure the DC predictor.	<a href="#">VLCDJD_CFG</a> [2] RSTEN	0 (no restart marker)	
	<a href="#">VLCDJD_DCPREDY</a> [11:0] PREDY	0 (reset DC predictor for Y)	
	<a href="#">VLCDJD_DCPREDU</a> [11:0] PREDU	0 (reset DC predictors for U)	
	<a href="#">VLCDJD_DCPREDV</a> [27:16] PREDV	0 (reset DC predictors for V)	
Configure the input and output memories.	<a href="#">VLCDJD_BSPTR</a> [11:0] BYTEPTR	0 (beginning of bitstream memory)	
	<a href="#">VLCDJD_BSPTR</a> [19:16] BITPTR	8 (initially empty byte, thus 8 free bits)	
	<a href="#">VLCDJD_CBUF</a> [11:10] START	0 (bitstream starts from beginning of first quadrant of bitstream memory)	
	<a href="#">VLCDJD_CBUF</a> [27:26] END	3 (bitstream ends at end of fourth quadrant of bitstream memory)	
	<a href="#">VLCDJD_DCTQM</a> [13:4] DCT	0 (DCT coefficients at the beginning of the image buffer)	
	Configure Huffman and quantizer memory pointers.	<a href="#">VLCDJD_DCTQM</a> [24:23] QMR	0 (quantizer matrix at beginning of QMEM)
		<a href="#">VLCDJD_CTRLTBL</a> [11:2] ADDR	0 (typically, control table is placed at the beginning of HUFMEM)
<a href="#">VLCDJD_DCDTBL0</a> [11:2] TBL0		30 (typically, the first decode table follows the control table, 60 shorts = 120 bytes = 30 words)	
<a href="#">VLCDJD_DCDTBL0</a> [27:18] TBL1		? (where the second decode table starts)	
<a href="#">VLCDJD_DCDTBL23</a> [11:2] TBL2		? (where the third decode table starts)	
<a href="#">VLCDJD_DCDTBL23</a> [27:18] TBL3		? (where the fourth decode table starts)	
Start decode with automatic ring buffer management.	<a href="#">VLCDJ_CTRL</a> [1] MODE	1 (decode mode)	
	<a href="#">VLCDJ_CTRL</a> [0] EN	1 (start VLCDJ operation)	

For each group of eight MCUs (the VLCDJ does not care if these MCUs cross multiple MCU rows), software sets the [VLCDJ\\_CTRL](#)[0] EN bit to 1 (restart VLCDJ operation).

Alternately, the hardware sequencer can start the VLCDJ, as well as control buffer switches and coordinate among the VLCDJ, DCT, and DMA.

Bitstream data is managed automatically between the VLCDJ and DMA throughout JPEG encoding of MCUs.

**NOTE:** Many registers change during the decoding of a frame, but software is not required to access these registers, because the VLCDJ manages the status information between tasks.

The last hardware task may require a smaller number of MCUs (the [VLCDJD\\_CFG](#)[13:8] NMCUS bit field) to adjust for the number of MCUs in the last group.

After the last VLCDJ task for the image, software must disable the automatic bitstream buffer management and prepare for the next image by setting the following values (these can be issued as a single write):

- [VLCDJ\\_CTRL](#)[6] RBEN = 0 (disable automatic DMA interaction)
- [VLCDJ\\_CTRL](#)[7] CLRRB = 1

Before the VLCDJ processes the next image decode, new bitstream data must be brought in, thereby overwriting the old data, and bit/byte pointers must be reset to the beginning of bitstream memory. As a result, any remaining bitstream data in bitstream memory afterward is harmless.

The number of MCUs processed by hardware per task must be set according to image buffer allocation. For example, with 16KB available for DCT coefficient memory, and YUV4:2:0 format, it is possible to process  $16K / (16 \times 16 \times 1.5 \times 2) = 21$  MCUs. Because the bitstream DMA is automatic, there is no bitstream data underrun for this many MCUs.

#### 8.4.6.4.5 Concurrent JPEG Encoding and Decoding

For concurrent decoding and encoding, the two operations are configured separately but similarly.

Half of bitstream memory is used for encoding (for example, the first half), and the rest is used for decoding. Automatic bitstream DMA does not work in this case, so the number of MCUs must accommodate the worst-case bitstream size per MCU.

Theoretically, the worst case is 16 bits of Huffman code plus 12 additional bits per coefficient, or 28 bits per coefficient. With natural images, there are typically about 8 bits per pixel when quantizers are minimal (that is, all 1s). With 16 bits per coefficient, or 768 bytes per MCU in YUV4:2:0 (this is 24 bits per pixel and three times the typical worst case), half of bitstream memory is 2KB, which can safely hold two MCUs.

In this case, half of QMEM is used for encoding and half is used for decoding. Encode and decode Huffman tables are placed so that they coexist in HUFFMEM.

The Huffman tables for encoding, decoding, and quantizer matrices must be placed accordingly in HUFMEM and QMEM. Bitstream data is filled at the second half of bitstream memory for decoding.

Register fields are configured as listed in [Table 8-1373](#).

**Table 8-1373. Concurrent JPEG Encoding and Decoding**

Step	Register/Bit Field	Value
Configure the encode register.	<a href="#">VLCDJE_CFG</a> [1:0] FMT	0 (YUV4:2:0 mode)
	<a href="#">VLCDJE_CFG</a> [13:8] NMCUS	1 (two MCUs)
	<a href="#">VLCDJD_CFG</a> [2] RSTEN	0 (no restart marker)
	<a href="#">VLCDJE_DCPREDY</a> [11:0] PREDY	0 (reset DC predictor for Y)
	<a href="#">VLCDJE_DCPREDUV</a> [11:0] PREDU	0 (reset DC predictors for U)
	<a href="#">VLCDJE_DCPREDUV</a> [27:16] PREDV	0 (reset DC predictors for V)
	<a href="#">VLCDJE_BSPTR</a> [19:16] BITPTR	8 (initially empty byte, thus 8 free bits)
	<a href="#">VLCDJE_DCTQM</a> [13:4] DCT	0 (DCT coefficients at beginning of image buffer)
Configure the decode register.	<a href="#">VLCDJD_CFG</a> [1:0] FMT	0 (YUV4:2:0 mode)
	<a href="#">VLCDJD_CFG</a> [13:8] NMCUS	1 (two MCUs)
	<a href="#">VLCDJD_CFG</a> [2] RSTEN	0 (no restart marker)
	<a href="#">VLCDJD_DCPREDY</a> [11:0] PREDY	0 (reset DC predictor for Y)
	<a href="#">VLCDJD_DCPREDUV</a> [11:0] PREDU	0 (reset DC predictors for U)
	<a href="#">VLCDJD_DCPREDUV</a> [27:16] PREDV	0 (reset DC predictors for V)
	<a href="#">VLCDJD_BSPTR</a> [19:16] BITPTR	8 (initially empty byte, thus 8 free bits)
	<a href="#">VLCDJD_DCTQM</a> [13:4] DCT	0 (DCT coefficients at the beginning of the image buffer)
Configure the byte pointer to bitstream memory.	<a href="#">VLCDJE_BSPTR</a> [11:0] BYTEPTR	0 (beginning of bitstream memory)
	<a href="#">VLCDJD_BSPTR</a> [11:0] BYTEPTR	0x800 (beginning of bitstream memory second half)
Configure the bitstream memory partition.	<a href="#">VLCDJE_CBUF</a> [11:10] START	0 (bitstream starts at beginning of first quadrant of bitstream memory)
	<a href="#">VLCDJE_CBUF</a> [27:26] END	1 (bitstream ends at end of second quadrant of bitstream memory)
	<a href="#">VLCDJD_CBUF</a> [11:10] START	2 (bitstream starts at beginning of third quadrant of bitstream memory)
	<a href="#">VLCDJD_CBUF</a> [27:26] END	3 (bitstream ends at end of fourth quadrant of bitstream memory)
Configure the quantizer matrix pointer.	<a href="#">VLCDJE_DCTQM</a> [24:23] QMR	0 (quantizer matrix at beginning of QMEM)
	<a href="#">VLCDJD_DCTQM</a> [24:23] QMR	2 (quantizer matrix at second half of QMEM)
Configure Huffman tables.	<a href="#">VLCDJE_VLCTBL</a> [11:2] ADDR	0 (VLC table at beginning of HUFMEM)
	<a href="#">VLCDJD_CTRLTBL</a> [11:2] ADDR	376 (after VLC table, 2 × (24 + 352) shorts = 376 words)

**Table 8-1373. Concurrent JPEG Encoding and Decoding (continued)**

Step	Register/Bit Field	Value
	VLCDJD_DCDTBL0[11:2] TBL0	406 (after control table of 30 words)
	VLCDJD_DCDTBL0[27:18] TBL1	? (where the second decode table starts)
	VLCDJD_DCDTBL23[11:2] TBL2	? (where the third decode table starts)
	VLCDJD_DCDTBL23[27:18] TBL3	? (where the fourth decode table starts)

Software sets the following values to start encoding:

- VLCDJ\_CTRL[1] MODE = 0 (encode mode)
- VLCDJ\_CTRL[0] EN = 1 (start VLCDJ operation)

When the VLCDJ completion interrupt is detected by software, encoding is complete and software must service the encoded bitstream data in the first half of bitstream memory, letting hardware access the second half of the bitstream for decoding. Software sets up bitstream writes to transfer the encoded bitstream from bitstream memory[0] to bitstream memory[128 × (VLCDJE\_BSPTR[11:0] BYTEPTR/128)] (rounding down to 128-byte alignment improves SDRAM efficiency). Software must identify the bitstream data already transferred out, for the next round of servicing encoded bitstream data. Software must also service to-encode input data to the DCT.

After kicking off encode DMAs, without waiting for DMA completion, software must set the following values to start VLCDJ decoding in parallel with the DMA:

- VLCDJ\_CTRL[1] MODE = 1 (decode mode)
- VLCDJ\_CTRL[0] EN = 1 (start VLCDJ operation)

When the VLCDJ completion interrupt is detected by software, decoding is complete and software must service the decoded bitstream data in the second half of bitstream memory, letting hardware access the first half of the bitstream for encoding. Software sets up bitstream writes to transfer the bitstream to be decoded from bitstream memory[0] to bitstream memory[128 × (VLCDJD\_BSPTR[11:0] BYTEPTR/128)] (rounding down to 128-byte alignment improves SDRAM efficiency). Software must identify the bitstream data already transferred in, for the next round of servicing bitstream data to be decoded. Software must also service decoded image data in the output image buffer of the DCT.

After kicking off decode DMAs, without waiting for DMA completion, software must start VLCDJ encoding in parallel with the DMA.

Software must continue to alternate between encoding and decoding until the image completes.

Figure 8-277 shows the concurrence schedule for JPEG encoding and decoding.

**Figure 8-277. Concurrence Schedule for JPEG Encoding and Decoding**

DMA in	E[0]		E[1]		E[2]		E[3]					
			D[0]		D[1]		D[2]		D[3]			
DCT		E[0]		E[1]	D[0]	E[2]	D[1]	E[3]	D[2]		D[3]	
VLCDJ			E[0]	D[0]	E[1]	D[1]	E[2]	D[2]	E[3]	D[3]		
DMA out				E[0]		E[1]		E[2]		E[3]		
					D[0]		D[1]		D[2]		D[3]	

vlcdj-007

Management of bitstream I/O and image I/O is heavy in intervals of only two MCUs. It is better to run only encoding for many MCUs, switch context, and then run decoding for many MCUs.

When running JPEG transcoding, decoding a picture and re-encoding it, to adjust quantizers and/or remove/insert restart markers, it can be more efficient to run encode and decode concurrently in VLCDJ, skipping DCT and image I/O DMA altogether. In this case, software handles only bitstream data I/O, which is more manageable.

Figure 8-278 shows the concurrence schedule for JPEG transcoding.

**Figure 8-278. Concurrency Schedule for JPEG Transcoding**

DMA in	D[0]		D[1]		D[2]		D[3]			
VLCDJ		D[0]	E[0]	D[1]	E[1]	D[2]	E[2]	D[3]	E[3]	
DMA out				E[0]		E[1]		E[2]		E[3]

vLCDJ-008

#### 8.4.6.4.5.1 DMA and Buffer Manual Management

When the VLCDJ performs one encode and one decode concurrently, both contexts can coexist in the configuration registers, because encode and decode configurations are in separate registers. The bitstream memory must be partitioned half-and-half between encode and decode:

- VLCDJE\_CBUF[11:10] START = 0 (0x0)
- VLCDJE\_CBUF[27:26] END = 1 (0x7FF)
- VLCDJD\_CBUF[11:10] START = 2 (0x800)
- VLCDJD\_CBUF[27:26] END = 3 (0xFFF)

The byte pointers and the bitstream data DMA must be set up accordingly.

When the VLCDJ is encoding, the decode half of the bitstream memory can be refilled by DMA. When the VLCDJ is decoding, the encode half of the bitstream memory can be transferred out by DMA. Hardware does not support automatic triggering of DMA; software must submit DMA based on the byte pointer value of each context. For greater SDRAM efficiency, the DMA block must be aligned correctly. For example, the pseudocode that sets up the DMA for an encode task may appear as:

```
new_dma_ptr = VLCDJ_BSPTR.BYTEPTR & ~0x3F; // align to 64-byte
if (new_dma_ptr > saved_dma_ptr)
{
    setup_dma(saved_dma_ptr, new_dma_ptr - saved_dma_ptr);
    // start DMA at saved_dma_ptr, length = new_dma_ptr - saved_dma_ptr
}
else // wraps around, need to DMA 2 blocks
{
    setup_dma(saved_dma_ptr, 0x800 - saved_dma_ptr);
    setup_dma(saved_dma_ptr, new_dma_ptr);
}
saved_dma_ptr = new_dma_ptr;
```

This DMA setup can be carried out concurrently with VLCDJ operation; thus, this code is executed with each encode or decode task. When the VLCDJ is encoding, software can access the decode bit/byte points to set up DMA for decoding, and vice versa.

Because hardware stalling does not work in this case, the number of MCUs must be configured so that in the worst case there is no bitstream data overrun or underrun.

The bitstream buffer can be allocated at a quarter of bitstream memory granularity, or 1KB. This allows at most four separate buffers to coexist in bitstream memory.

If multiple encode and/or decode threads are performed concurrently, software must save and restore the appropriate configuration registers. The register map has common registers in one group, encode registers in another group, and decode registers in a third group. Bitstream memory partitioning and corresponding DMA submission are left to software.

#### 8.4.6.4.6 Example of YUV4:4:4 Encoding

The following pseudocode shows the general sequence involved in YUV4:4:4 encoding. Decoding can be similarly derived.

```
encode_yuv444()
{
    VLCDJ_CTRL[1] MODE = 0; /* encode */
    VLCDJE_CFG[1:0] FMT = 2; /* sequential block mode */
    VLCDJE_CFG[13:8] NMCUS = 0; /* 1 block at a time */
    pred_y = 0;
    pred_u = 0;
    pred_v = 0;
```

```

for (i=0; i<num_mcus; i++)
{
VLCDJE_DCPREDY = pred_y; /* restore Y context */
VLCDJE_DCT = DCT_Y;
VLCDJE_QMR = QMR_Y;
VLCDJE_VLCTBL = VLC_Y;
VLCDJ_CTRL[0] EN = 1; /* kick off VLCDJ for Y */
pred_y = VLCDJE_DCPREDY; /* save Y context */
VLCDJE_DCPREDY = pred_u; /* restore U context */
VLCDJE_DCT = DCT_U;
VLCDJE_QMR = QMR_UV;
VLCDJE_VLCTBL = VLC_UV;
VLCDJ_CTRL[0] EN = 1; /* kick off VLCDJ for U */
pred_u = VLCDJE_DCPREDY; /* save U context */
VLCDJE_DCPREDY = pred_v; /* restore V context */
VLCDJE_DCT = DCT_V;
VLCDJ_CTRL[0] EN = 1; /* kick off VLCDJ for V */
pred_v = VLCDJE_DCPREDY; /* save V context */
}
}

```

**8.4.6.4.7 Example of Huffman Encode and Decode Tables**

**8.4.6.4.7.1 Huffman Encode Table**

Sample JPEG Luma DC encode table:

```

unsigned short jpeg_dc_y[24] =
{
0x0000,0x0002,0x0002,0x0003,0x0003,0x0003,0x0004,0x0003,
0x0005,0x0003,0x0006,0x0003,0x000e,0x0004,0x001e,0x0005,
0x003e,0x0006,0x007e,0x0007,0x00fe,0x0008,0x01fe,0x0009
};

```

Table 8-1374 corresponds to Table 17 in the JPEG standard *ISO/IEC JTC1/SC29/WG10 Standard 10918-1 (JPEG), Information Technology – Digital Compression and Coding of Continuous-Tone Still Images – Requirements and Guidelines*.

**Table 8-1374. Sample Luma DC Huffman Code Table**

Category	Code Length	Code Word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

To code a DC coefficient value of -5:

Use  $\text{abs}(-5) = 5$  to select a size of 3. Look up the encode table to find:

- $\text{jpeg\_dc\_y}[2 \times 3] = 4$  (code)
- $\text{jpeg\_dc\_y}[2 \times 3 + 1] = 3$  (length)

Thus, the VLCDJ adds 100 to the bitstream as the Huffman code of the size.

The additional bits for negative coefficients are ones complement of the absolute value, so for -5, this is 010, or 2. After the Huffman code of the size, the VLCDJ adds 010 to the bitstream.

The AC coefficient table is the same, except that the VLCDJ uses  $16 \times \text{size} + \text{run}$  to index the encode table.

#### 8.4.6.4.7.2 Huffman Decode Tables

Sample JPEG Luma DC control and decode tables:

```
unsigned short jpeg_dc_y_ctl_tbl[13]=
{
    0x0000,0x0000,0x0000,0x0000,0x1800,0x2001,0x2802,0x3003,
    0x3804,0x4005,0x4806,0x4806,0x4806
};
unsigned short jpeg_dc_y_dcd_tbl[14]=
{
    0x0000,0x480b,0x400a,0x3809,0x3008,0x2807,0x2006,0x1805,
    0x1804,0x1803,0x1802,0x1801,0x1000,0x1000
};
```

These tables contain the same information as [Table 8-1374](#), which complies with the JPEG standard.

### 8.4.6.5 ISS SIMCOP VLCDJ Register Manual

#### 8.4.6.5.1 ISS SIMCOP VLCDJ Instance Summary

Table 8-1375 summarizes the VLDCJ instance.

**Table 8-1375. ISS SIMCOP VLCDJ Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-M3 Private Access	Size
VLCDJ	0x5202 0600	0x5506 0600	128 bytes

**NOTE:** Private Access is an access which is not using the L3/L4 interconnects.

#### 8.4.6.5.2 VLCDJ Registers

##### 8.4.6.5.2.1 VLCDJ Register Summary

Table 8-1376 summarizes VLDCJ register mapping.

**Table 8-1376. VLCDJ Register Mapping Summary**

Register	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-M3 Private Access
VLCDJ_REVISION	RW	32	0x0000 0000	0x5202 0600	0x5506 0600
VLCDJ_CTRL	RW	32	0x0000 0004	0x5202 0604	0x5506 0604
VLCDJE_CFG	RW	32	0x0000 0008	0x5202 0608	0x5506 0608
VLCDJE_DCPREDY	RW	32	0x0000 000C	0x5202 060C	0x5506 060C
VLCDJE_DCPREDUV	RW	32	0x0000 0010	0x5202 0610	0x5506 0610
VLCDJE_BSPTR	RW	32	0x0000 0014	0x5202 0614	0x5506 0614
VLCDJE_CBUF	RW	32	0x0000 0018	0x5202 0618	0x5506 0618
VLCDJE_RSTCFG	RW	32	0x0000 001C	0x5202 061C	0x5506 061C
VLCDJE_DCTQM	RW	32	0x0000 0020	0x5202 0620	0x5506 0620
VLCDJE_VLCTBL	RW	32	0x0000 0024	0x5202 0624	0x5506 0624
VLCDJE_RSTPTR	RW	32	0x0000 0028	0x5202 0628	0x5506 0628
VLCDJE_RSTOFST	RW	32	0x0000 002C	0x5202 062C	0x5506 062C
VLCDJD_CFG	RW	32	0x0000 0040	0x5202 0640	0x5506 0640
VLCDJD_DCPREDY	RW	32	0x0000 0044	0x5202 0644	0x5506 0644
VLCDJD_DCPREDUV	RW	32	0x0000 0048	0x5202 0648	0x5506 0648
VLCDJD_BSPTR	RW	32	0x0000 004C	0x5202 064C	0x5506 064C
VLCDJD_CBUF	RW	32	0x0000 0050	0x5202 0650	0x5506 0650
VLCDJD_DCTQM	RW	32	0x0000 0054	0x5202 0654	0x5506 0654
VLCDJD_CTRLTBL	RW	32	0x0000 0058	0x5202 0658	0x5506 0658
VLCDJD_DCDBL01	RW	32	0x0000 005C	0x5202 065C	0x5506 065C
VLCDJD_DCDBL23	RW	32	0x0000 0060	0x5202 0660	0x5506 0660
VLCDJD_DCTERR	R	32	0x0000 0064	0x5202 0664	0x5506 0664

### 8.4.6.5.2.2 VLCDJ Register Descriptions

through describe the individual registers.

**Table 8-1377. VLCDJ\_REVISION**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .	<b>Instance</b>
<b>Description</b>	IP revision	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

**Table 8-1378. Register Call Summary for Register VLCDJ\_REVISION**

ISS Still Image Coprocessor

- [VLCDJ Registers: \[0\]](#)

**Table 8-1379. VLCDJ\_CTRL**

<b>Address Offset</b>	0x0000 0004	
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .	<b>Instance</b>
<b>Description</b>	Controls common to encoding and decoding	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RESERVED																BUSY	RESERVED										CLRRB	RBEN	INTEN_ERR	INTEN_DONE	AUTOGATING	TRIG_SRC	MODE	EN

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Read returns 0.	R	0x00
16	BUSY	Idle/busy status 0: Idle 1: Busy	R	0
15:8	RESERVED	Read returns 0.	R	0x00
7	CLRRB	Write only; read returns 0.	W	0
6	RBEN	Enable RB signaling.	RW	0
5	INTEN_ERR	Interrupt enable for decode error 0: No interrupt generated on decode error 1: Interrupt generated on decode error	RW	0
4	INTEN_DONE	Interrupt enable for task completion. DONE_VLCD is not gated by this and is always asserted at task completion. 0: No interrupt generated on task completion 1: Interrupt generated on task completion	RW	0
3	AUTOGATING	Internal clock gating on OCP clock and functional clock 0: Clocks are free-running. 1: Clocks are gated off in sub-blocks that are not required for operation.	RW	1



Bits	Field Name	Description	Type	Reset
2	TRIG_SRC	Which mechanism starts VLCDJ operation 0: MMR write to VLCDJ_CTRL.EN 1: Hardware start signal	RW	0
1	MODE	0: Encode 1: Decode	RW	0
0	EN	Module enable by software (write-only, read returns 0). When TRIG_SRC = 0 and BUSY = 0, set this field to 1 to start VLCDJ. When TRIG_SRC = 1, writes to this field are ignored. Setting TRIG_SRC = 0 and EN = 1 on the same register write is recognized.	W	0

**Table 8-1380. Register Call Summary for Register VLCDJ\_CTRL**

ISS Still Image Coprocessor

- [ISS SIMCOP VLCDJ Power Management: \[0\]](#)
- [ISS SIMCOP VLCDJ Interrupt Requests: \[1\] \[2\]](#)
- [ISS SIMCOP VLCDJ Start and Register Modification During Processing: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [JPEG Encoding: \[15\] \[17\] \[18\] \[21\] \[22\]](#)
- [JPEG Decoding: \[23\] \[25\] \[26\] \[29\] \[30\]](#)
- [Concurrent JPEG Encoding and Decoding: \[32\] \[33\] \[34\] \[35\]](#)
- [VLCDJ Registers: \[37\] \[38\]](#)

**Table 8-1381. VLCDJE\_CFG**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Encode configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NMCUS						RESERVED				RLOCEN	RSTEN	FMT			

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Read returns 0.	R	0x00000
13:8	NMCUS	Number of MCUs (FMT = 0, 1) or blocks (FMT = 2) to encode 0: 1 MCU 1: 2 MCUs, etc.	RW	0x00
7:4	RESERVED	Read returns 0.	R	0x0
3	RLOCEN	Restart marker location recording enable 0: Restart marker location recording disabled 1: Restart marker location recording enabled	RW	0
2	RSTEN	Restart marker insertion enable 0: Restart marker insertion disabled 1: Restart marker insertion enabled	RW	0
1:0	FMT	0: YUV4:2:0 1: YUV4:2:2 2 : Sequential blocks	RW	0x0

**Table 8-1382. Register Call Summary for Register VLCDJE\_CFG**

ISS Still Image Coprocessor

- [Restart Marker](#):
- [JPEG Encoding](#): [1] [2] [3] [4]
- [Concurrent JPEG Encoding and Decoding](#): [5] [6]
- [VLCDJ Registers](#): [9]

**Table 8-1383. VLCDJE\_DCPREDY**

<b>Address Offset</b>	0x0000 000C	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Encode DC predictor for Y		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PREDY																			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Read returns 0.	R	0x00000
11:0	PREDY	DC predictor for Y	RW	0x000

**Table 8-1384. Register Call Summary for Register VLCDJE\_DCPREDY**

ISS Still Image Coprocessor

- [JPEG Encoding](#): [0]
- [Concurrent JPEG Encoding and Decoding](#): [1]
- [VLCDJ Registers](#): [2]

**Table 8-1385. VLCDJE\_DCPREDUV**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Encode DC predictor for U and V		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				PREDV												RESERVED				PREDU											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:16	PREDV	DC predictor for V	RW	0x000
15:12	RESERVED	Read returns 0.	R	0x0
11:0	PREDU	DC predictor for U	RW	0x000

**Table 8-1386. Register Call Summary for Register VLCDJE\_DCPREDUV**

ISS Still Image Coprocessor

- [JPEG Encoding](#): [0] [1]
- [Concurrent JPEG Encoding and Decoding](#): [2] [3]
- [VLCDJ Registers](#): [4]

**Table 8-1387. VLCDJE\_BSPTR**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .	<b>Instance</b>	VLCDJ
<b>Description</b>	Encode bitstream pointer		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BITPTR				RESERVED				BYTEPTR															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Read returns 0.	R	0x000
19:16	BITPTR	Bit pointer, 1..8, indicates number of available bits	RW	0x8
15:12	RESERVED	Read returns 0.	R	0x0
11:0	BYTEPTR	Byte pointer (to BSMEM)	RW	0x000

**Table 8-1388. Register Call Summary for Register VLCDJE\_BSPTR**

ISS Still Image Coprocessor

- [Memory Interfaces: \[0\] \[1\]](#)
- [JPEG Encoding: \[2\] \[3\] \[4\] \[5\]](#)
- [Concurrent JPEG Encoding and Decoding: \[6\] \[7\] \[8\]](#)
- [VLCDJ Registers: \[9\]](#)

**Table 8-1389. VLCDJE\_CBUF**

<b>Address Offset</b>	0x0000 0018		
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .	<b>Instance</b>	VLCDJ
<b>Description</b>	Encode bitstream circular buffer		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				END	RESERVED								RESERVED				START	RESERVED													

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:26	END	Ending quarter (1KB each unit) of bitstream buffer. Software can write a byte address into the upper 16 bits of the register.	RW	0x3
25:16	RESERVED	Read returns 0x3FF.	R; returns 1s	0x3FF
15:12	RESERVED	Read returns 0.	R	0x0
11:10	START	Starting quarter (1KB each unit) of bitstream buffer. Software can write a byte address to the lower 16 bits of the register.	RW	0x0
9:0	RESERVED	Read returns 0.	R	0x000

**Table 8-1390. Register Call Summary for Register VLCDJE\_CBUF**

ISS Still Image Coprocessor

- [Memory Interfaces: \[0\] \[1\]](#)
- [Encode/Decode Context Switching: \[2\] \[3\]](#)
- [JPEG Encoding: \[4\] \[5\]](#)
- [Concurrent JPEG Encoding and Decoding: \[6\] \[7\] \[8\] \[9\]](#)
- [VLCDJ Registers: \[10\]](#)

**Table 8-1391. VLCDJE\_RSTCFG**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Encode restart marker configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	INC			RESERVED	INIT			RESERVED	PHASE							RESERVED	INTRVL														

Bits	Field Name	Description	Type	Reset
31	RESERVED	Read returns 0.	R	0
30:28	INC	Restart count increment value	RW	0x1
27	RESERVED	Read returns 0.	R	0
26:24	INIT	Restart marker initial count	RW	0x0
23:22	RESERVED	Read returns 0.	R	0x0
21:12	PHASE	MCU count within the interval	RW	0x000
11:10	RESERVED	Read returns 0.	R	0x0
9:0	INTRVL	Restart interval (in MCUs)	RW	0x000

**Table 8-1392. Register Call Summary for Register VLCDJE\_RSTCFG**

ISS Still Image Coprocessor

- [Restart Marker:](#)
- [VLCDJ Registers: \[13\]](#)

**Table 8-1393. VLCDJE\_DCTQM**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Encode DCT coefficient and quantizer matrix pointers		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								QMR	RESERVED							DCT							RESERVED								

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24:23	QMR	Quarter (128 bytes/unit) of quantization matrix reciprocal. Software can write a byte address to the upper 16 bits of the register.	RW	0x0
22:14	RESERVED	Read returns 0.	R	0x000

Bits	Field Name	Description	Type	Reset
13:4	DCT	128-bit/word address of DCT coefficients. Software can write a byte address to the lower 16 bits of the register.	RW	0x000
3:0	RESERVED	Read returns 0.	R	0x0

**Table 8-1394. Register Call Summary for Register VLCDJE\_DCTQM**

ISS Still Image Coprocessor

- [JPEG Encoding: \[0\] \[1\]](#)
- [Concurrent JPEG Encoding and Decoding: \[2\] \[3\]](#)
- [VLCDJ Registers: \[4\]](#)

**Table 8-1395. VLCDJE\_VLCTBL**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a>		
<b>Description</b>	Encode Huffman table pointer		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDR											RESERVED				

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Read returns 0.	R	0x00000
11:2	ADDR	Encode Huffman table pointer, 32-bit word address. Software can write a byte address into the entire register.	RW	0x000
1:0	RESERVED	Read returns 0.	R	0x0

**Table 8-1396. Register Call Summary for Register VLCDJE\_VLCTBL**

ISS Still Image Coprocessor

- [Huffman Table Organization: \[0\] \[1\] \[2\] \[3\]](#)
- [JPEG Encoding: \[4\]](#)
- [Concurrent JPEG Encoding and Decoding: \[5\]](#)
- [VLCDJ Registers: \[6\]](#)

**Table 8-1397. VLCDJE\_RSTPTR**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Encode restart marker locations		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDR											RESERVED				

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Read returns 0.	R	0x00000
13:4	ADDR	Pointer to restart marker locations in image buffer, 128-bit/word address Software can write a byte address into the entire register.	RW	0x000
3:0	RESERVED	Read returns 0.	R	0x0

**Table 8-1398. Register Call Summary for Register VLCDJE\_RSTPTR**

ISS Still Image Coprocessor

- [Restart Marker:](#)
- [VLCDJ Registers:](#) [2]

**Table 8-1399. VLCDJE\_RSTOFST**

<b>Address Offset</b>	0x0000 002C																																																													
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .	<b>Instance</b> VLCDJ																																																												
<b>Description</b>	SDRAM address to add to encode restart marker locations																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">OFFSET</td> <td colspan="12"></td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	OFFSET																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
OFFSET																																																														

Bits	Field Name	Description	Type	Reset
31:0	OFFSET	SDRAM address of bitstream buffer, to be added to the restart marker locations	RW	0x0000 0000

**Table 8-1400. Register Call Summary for Register VLCDJE\_RSTOFST**

ISS Still Image Coprocessor

- [Restart Marker:](#)
- [VLCDJ Registers:](#) [1]

**Table 8-1401. VLCDJD\_CFG**

<b>Address Offset</b>	0x0000 0040																																																							
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .	<b>Instance</b> VLCDJ																																																						
<b>Description</b>	Decode configuration																																																							
<b>Type</b>	RW																																																							
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="12">RESERVED</td> <td colspan="4">NMCUS</td> <td colspan="4">RESERVED</td> <td>RSTEN</td> <td>FMT</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED												NMCUS				RESERVED				RSTEN	FMT
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																									
RESERVED												NMCUS				RESERVED				RSTEN	FMT																																			

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Read returns 0.	R	0x00000
13:8	NMCUS	Number of MCUs (FMT = 0, 1) or blocks (FMT = 2) to decode 0: 1 MCU 1: 2 MCUs, etc.	RW	0x00
7:3	RESERVED	Read returns 0.	R	0x00
2	RSTEN	Restart marker detection/processing enable 0: Restart marker detection/processing disabled 1: Restart marker detection/processing enabled	RW	1
1:0	FMT	0: YUV4:2:0 1: YUV4:2:2 2: Sequential blocks	RW	0x0

**Table 8-1402. Register Call Summary for Register VLCDJD\_CFG**

ISS Still Image Coprocessor

- [JPEG Decoding: \[0\] \[1\] \[2\] \[3\]](#)
- [Concurrent JPEG Encoding and Decoding: \[4\] \[5\] \[6\] \[7\]](#)
- [VLCDJ Registers: \[8\]](#)

**Table 8-1403. VLCDJD\_DCPREDY**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Decode DC predictor for Y		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PREDY															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Read returns 0.	R	0x0000
11:0	PREDY	DC predictor for Y	RW	0x000

**Table 8-1404. Register Call Summary for Register VLCDJD\_DCPREDY**

ISS Still Image Coprocessor

- [JPEG Decoding: \[0\]](#)
- [Concurrent JPEG Encoding and Decoding: \[1\]](#)
- [VLCDJ Registers: \[2\]](#)

**Table 8-1405. VLCDJD\_DCPREDUV**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Decode DC predictor for U and V		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PREDV								RESERVED								PREDU							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:16	PREDV	DC predictor for V	RW	0x000
15:12	RESERVED	Read returns 0.	R	0x0
11:0	PREDU	DC predictor for U	RW	0x000

**Table 8-1406. Register Call Summary for Register VLCDJD\_DCPREDUV**

ISS Still Image Coprocessor

- [JPEG Decoding: \[0\] \[1\]](#)
- [Concurrent JPEG Encoding and Decoding: \[2\] \[3\]](#)
- [VLCDJ Registers: \[4\]](#)

**Table 8-1407. VLCDJD\_BSPTR**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Decode bitstream pointer		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BITPTR				RESERVED				BYTEPTR															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Read returns 0.	R	0x000
19:16	BITPTR	Bit pointer, 1..8, indicates number of available bits	RW	0x8
15:12	RESERVED	Read returns 0.	R	0x0
11:0	BYTEPTR	Byte pointer (to BSMEM)	RW	0x000

**Table 8-1408. Register Call Summary for Register VLCDJD\_BSPTR**

ISS Still Image Coprocessor

- [Memory Interfaces: \[0\] \[1\]](#)
- [Error Reporting: \[2\] \[3\]](#)
- [JPEG Decoding: \[4\] \[5\]](#)
- [Concurrent JPEG Encoding and Decoding: \[6\] \[7\] \[8\]](#)
- [VLCDJ Registers: \[9\]](#)

**Table 8-1409. VLCDJD\_CBUF**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Decode bitstream circular buffer		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				END	RESERVED								RESERVED				START	RESERVED													

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:26	END	Ending quarter (1KB each unit) of bitstream buffer. Software can write a byte address into the upper 16 bits of the register.	RW	0x3
25:16	RESERVED	Read returns 0x3FF.	R; returns 1s	0x3FF
15:12	RESERVED	Read returns 0.	R	0x0
11:10	START	Starting quarter (1KB each unit) of bitstream buffer. Software can write a byte address to the lower 16 bits of the register.	RW	0x0
9:0	RESERVED	Read returns 0.	R	0x000



**Table 8-1410. Register Call Summary for Register VLCDJD\_CBUF**

ISS Still Image Coprocessor

- [Memory Interfaces: \[0\] \[1\]](#)
- [Encode/Decode Context Switching: \[2\] \[3\]](#)
- [JPEG Decoding: \[4\] \[5\]](#)
- [Concurrent JPEG Encoding and Decoding: \[6\] \[7\] \[8\] \[9\]](#)
- [VLCDJ Registers: \[10\]](#)

**Table 8-1411. VLCDJD\_DCTQM**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Decode DCT coefficient and quantizer matrix pointers		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								QM	RESERVED								DCT								RESERVED						

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24:23	QM	Quarter (128 bytes/unit) of quantization matrix. Software can write a byte address to the upper 16 bits of the register.	RW	0x0
22:14	RESERVED	Read returns 0.	R	0x000
13:4	DCT	128-bit/word address of DCT coefficients. Software can write a byte address to the lower 16 bits of the register.	RW	0x000
3:0	RESERVED	Read returns 0.	R	0x0

**Table 8-1412. Register Call Summary for Register VLCDJD\_DCTQM**

ISS Still Image Coprocessor

- [JPEG Decoding: \[0\] \[1\]](#)
- [Concurrent JPEG Encoding and Decoding: \[2\] \[3\]](#)
- [VLCDJ Registers: \[4\]](#)

**Table 8-1413. VLCDJD\_CTRLTBL**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Decode control table base		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDR											RESERVED				

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Read returns 0.	R	0x00000
11:2	ADDR	Starting address of decode control table, 32-bit word. Software can write a byte address into the entire register.	RW	0x000
1:0	RESERVED	Read returns 0.	R	0x0

**Table 8-1414. Register Call Summary for Register VLCDJD\_CTRLTBL**

ISS Still Image Coprocessor

- [JPEG Decoding: \[0\]](#)
- [Concurrent JPEG Encoding and Decoding: \[1\]](#)
- [VLCDJ Registers: \[2\]](#)

**Table 8-1415. VLCDJD\_DCDTBL01**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Decode Huffman tables 0 and 1		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED								DCDTBL1								RESERVED								DCDTBL0								RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:18	DCDTBL1	Starting byte address of decode table 1, 32-bit word. Software can write a byte address into the lower 16 bits of the register. This is for Luma AC.	RW	0x000
17:12	RESERVED	Read returns 0.	R	0x00
11:2	DCDTBL0	Starting byte address of decode table 0, 32-bit word. Software can write a byte address into the lower 16 bits of the register. This is for Luma DC.	RW	0x000
1:0	RESERVED	Read returns 0.	R	0x0

**Table 8-1416. Register Call Summary for Register VLCDJD\_DCDTBL01**

ISS Still Image Coprocessor

- [Huffman Table Organization:](#)
- [JPEG Decoding: \[2\] \[3\]](#)
- [Concurrent JPEG Encoding and Decoding: \[4\] \[5\]](#)
- [VLCDJ Registers: \[6\]](#)

**Table 8-1417. VLCDJD\_DCDTBL23**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>	Decode Huffman tables 2 and 3		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED								DCDTBL3								RESERVED								DCDTBL2								RESERVED

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:18	DCDTBL3	Starting byte address of decode table 3, 32-bit word. Software can write a byte address into the lower 16 bits of the register. This is for Chroma AC.	RW	0x000
17:12	RESERVED	Read returns 0.	R	0x00
11:2	DCDTBL2	Starting byte address of decode table 2, 32-bit word. Software can write a byte address into the lower 16 bits of the register. This is for Chroma DC.	RW	0x000
1:0	RESERVED	Read returns 0.	R	0x0

**Table 8-1418. Register Call Summary for Register VLCDJD\_DCDTBL23**

ISS Still Image Coprocessor

- [Huffman Table Organization](#):
- [JPEG Decoding](#): [2] [3]
- [Concurrent JPEG Encoding and Decoding](#): [4] [5]
- [VLCDJ Registers](#): [6]

**Table 8-1419. VLCDJD\_DCTERR**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	VLCDJ
<b>Physical Address</b>	See <a href="#">Table 8-1376</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ERRPTR															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Read returns 0.	R	0x00000
13:0	ERRPTR	Byte address pointer to DCT coefficients, near where decode error occurs (read-only)	R	0x0000

**Table 8-1420. Register Call Summary for Register VLCDJD\_DCTERR**

ISS Still Image Coprocessor

- [Error Reporting](#): [0]
- [VLCDJ Registers](#): [1]

## 8.4.7 ISS SIMCOP Rotation Accelerator Module

This section describes the rotation accelerator (ROT) module in the still-image coprocessor (SIMCOP) subsystem.

### 8.4.7.1 ISS SIMCOP ROT Overview

The ROT module is intended to be used in the SIMCOP to perform block data rotation and data shifting from one SIMCOP local memory to another. The following formats and operations are supported:

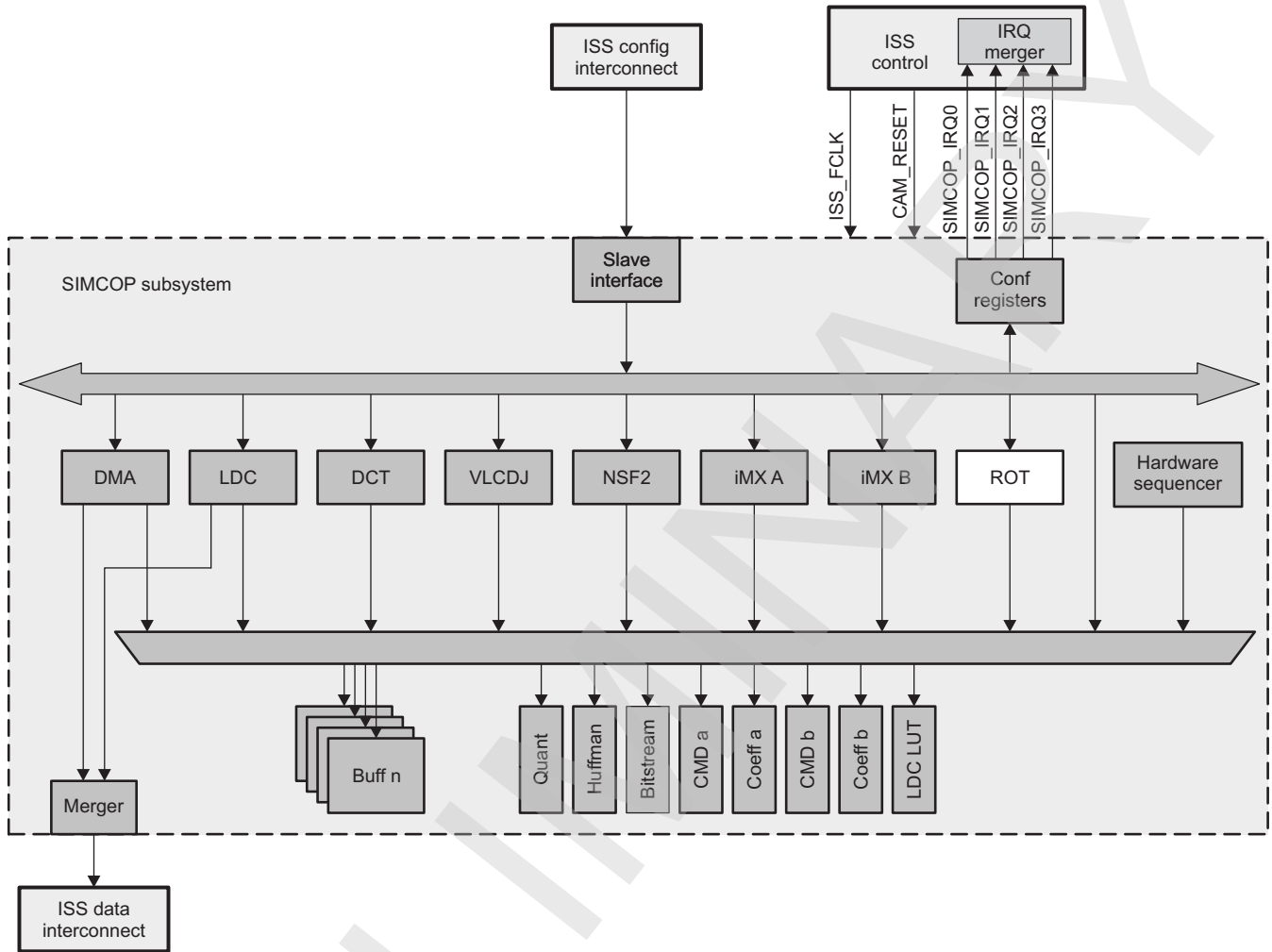
- YUV4:2:2 and generic 8-/16-/32-bit data block rotation by 90, 180, and 270 degrees
- YUV4:2:0 rotation by 90, 180, and 270 degrees
- Multiple-block rotation
- Data block shifting
- Data block horizontal circular shifting

The ROT supports a simple 1-block-per-initiation interaction, and includes a microcontroller unit (MCU) software-writable enable bit and hardware trigger signal to start the processing (see [Section 8.4.7.3.7, ISS SIMCOP ROT Synchronization](#)). It relies on external direct memory access (DMA) for transfers, and can work with the MCU or hardware sequencer to coordinate between transfer and computation.

The ROT has an input and output memory interface (see [Figure 8-281](#)). The rotation and horizontal-shifting operations read from the input memory and write to the output memory. The module connects to an input memory of up to 16KB (2048 × 64-bit), and an output memory of up to 16KB (2048 × 64-bit) using synchronous single-port memory interface.

[Figure 8-279](#) is an overview of the ROT.

Figure 8-279. ISS SIMCOP ROT Module Overview

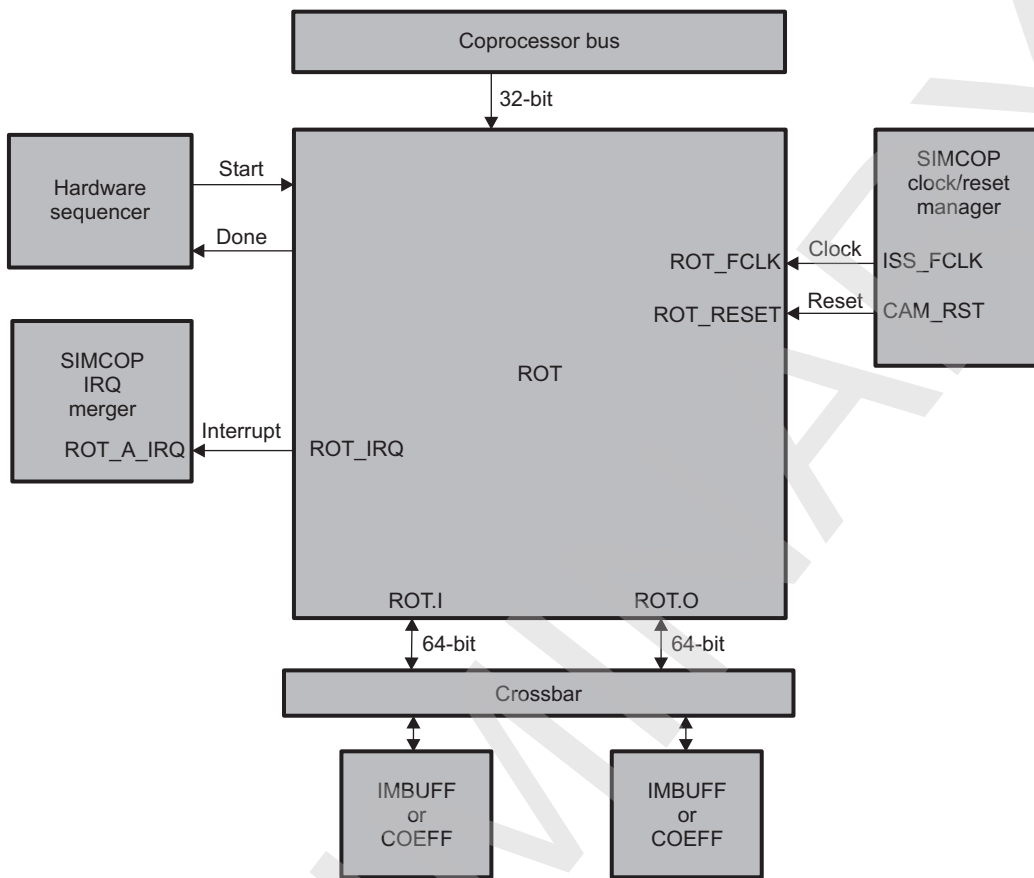


rot-001

### 8.4.7.2 ISS SIMCOP ROT Integration

The ROT is part of the SIMCOP subsystem in the imaging subsystem (ISS). [Figure 8-280](#) shows the integration of the ROT in the SIMCOP subsystem.

**Figure 8-280. ISS SIMCOP ROT Engine Integration**



rot-014

Table 8-1421 through Table 8-1423 list the integration attributes, clocks and resets, and hardware requests, respectively, of the SIMCOP ROT.

**Table 8-1421. ISS SIMCOP ROT Integration Attributes**

Module Instance	Attributes
	Power Domain
ROT	PD_CAM

**Table 8-1422. ISS SIMCOP ROT Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
ROT	ROT_FCLK	ISS_FCLK	PRCM	Functional clock provided by ISS_CLK from the power, reset, and clock management (PRCM) module. It is used by all ISS submodules and ISS top-level resources.
Resets				
ROT	ROT_RESET	CAM_RST	PRCM	ISS and SIMCOP global reset

For information about clock and reset management, see Section 8.4.1.2.1, *ISS SIMCOP Local Power and Clock Management*.

**Table 8-1423. ISS SIMCOP ROT Hardware Requests**

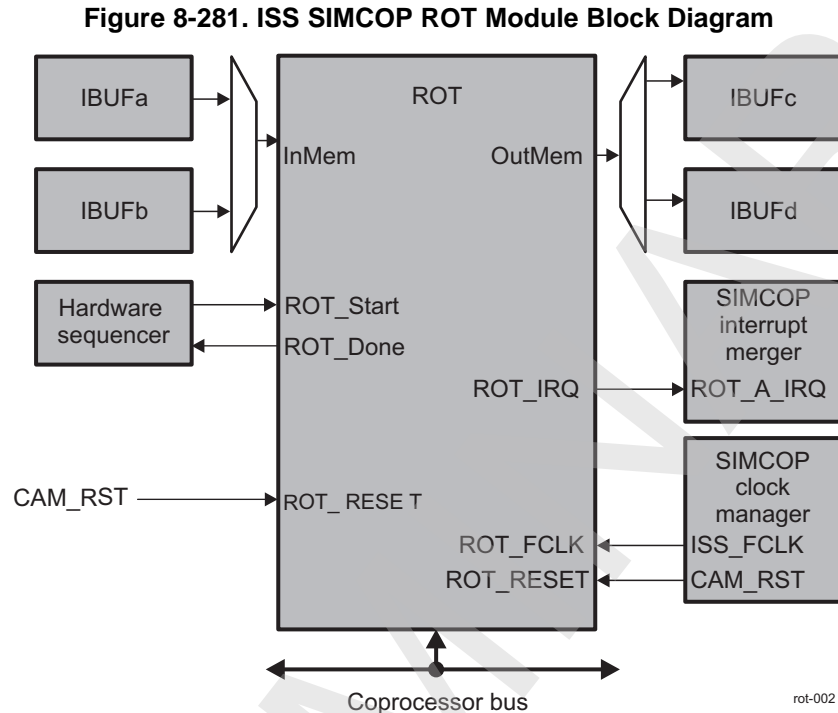
Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
ROT	ROT_IRQ	ROT_A_IRQ	SIMCOP IRQ merger	Event triggered by the ROT engine.

For more information about interrupt requests, see [Section 8.4.1.2.3](#), *Interrupt Merger*.

### 8.4.7.3 ISS SIMCOP ROT Functional Description

#### 8.4.7.3.1 ISS SIMCOP ROT Block Diagram

Figure 8-281 shows the block diagram of the ROT.



#### 8.4.7.3.2 ISS SIMCOP ROT Clock Configuration

The ROT receives a functional clock (ISS\_FCLK) from the ISS SIMCOP. The configuration clock is derived from the functional clock and interconnect clock enable signal (the [ROT\\_CFG\[9\]](#) AUTOGATING bit).

#### 8.4.7.3.3 ISS SIMCOP ROT Reset

The ROT receives an asynchronous hardware reset signal (CAM\_RST) from the ISS SIMCOP.

#### 8.4.7.3.4 ISS SIMCOP ROT Power Management

The ROT uses clock autogating to conserve power. When a subblock is not in use, the corresponding clock tree is gated. The [ROT\\_CFG\[9\]](#) AUTOGATING bit controls whether autogating is enabled.

#### 8.4.7.3.5 ISS SIMCOP ROT Interrupt Requests

One interrupt output line, ROT\_A\_IRQ, is used as a sideband signal of the interconnect slave port. This signal is connected to the SIMCOP interrupt merger. See [Section 8.4.1.2.3, Interrupt Merger](#).

#### 8.4.7.3.6 ISS SIMCOP ROT DMA Requests

The ROT has no DMA requests from the module.



### 8.4.7.3.7 ISS SIMCOP ROT Synchronization

The ROT has a simple one-block-per-task interaction with outside logic/microprocessor unit (MPU). The module can be started by setting the `ROT_CTRL[0]` EN bit to 0x1, or through the ROT\_START signal of the hardware sequencer interface. When the ROT completes the specified rotation or data shift task, it sends a pulse on the ROT\_DONE signal and, when the interrupt is enabled, sends a pulse on the ROT\_A\_IRQ interrupt output.

The `ROT_CFG[8]` TRIG\_SRC bit specifies to which source the hardware must respond:

- When the ROT is idle and the `ROT_CFG[8]` TRIG\_SRC bit is set to 0x0, hardware responds to the memory mapped register (MMR) mechanism, and any pulses on the start signal are ignored.
- When the ROT is idle and the `ROT_CFG[8]` TRIG\_SRC bit is set to 0x1, hardware responds to the hardware start signal, and any write operation to the enable bit field is ignored.

When the ROT is busy, both triggering mechanisms are ignored.

### 8.4.7.3.8 ISS SIMCOP ROT Formats and Operations

The ROT supports the formats and operations described in the following sections.

#### 8.4.7.3.8.1 ISS SIMCOP ROT YUV4:2:0 and Generic 8-/16-/32-Bit Data Block Rotation

Rotation of YUV4:2:0 and generic 8-bit, 16-bit, 32-bit data blocks is straightforward. None of the data points are altered; each data point is copied to an appropriate address location. Zero-degree rotation is included for completeness of the programming model, and involves only data copy.

When YUV4:2:0 data have separate Y and UV processing and must be separately rotated, Y data can use generic 8-bit data rotation, and UV data can use generic 16-bit data rotation.

#### 8.4.7.3.8.2 ISS SIMCOP ROT YUV4:2:2 Rotation

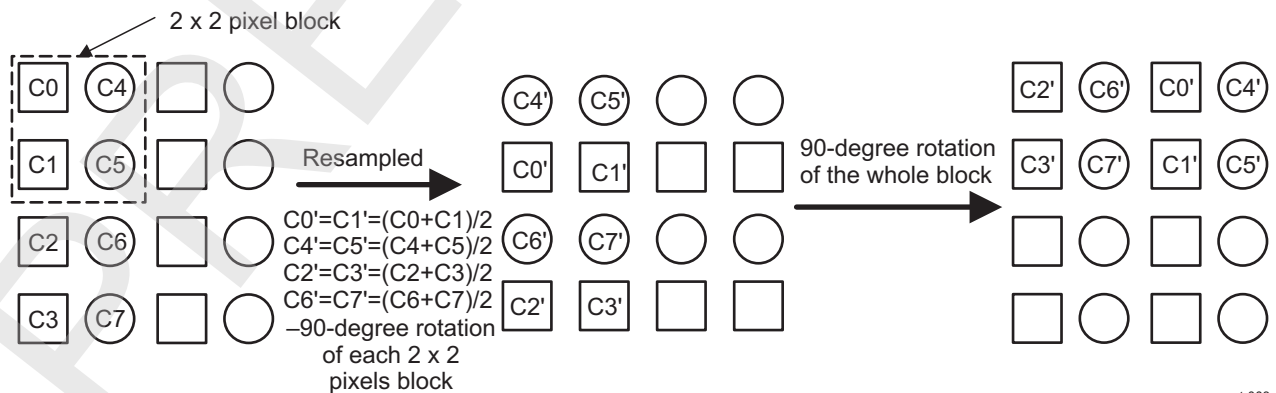
Various rotation cases of YUV4:2:2 data are described in the following and shown in Figure 8-282 through Figure 8-284 using 4 x 4-pixel block as an example.

The degree of rotation is defined as rotating clockwise. For 90-degree rotation:

- The two U samples in each 2 x 2-pixel input block are averaged and then replicated to provide the two U samples in the 2 x 2-pixel output block.
- V samples - processed in the same way.

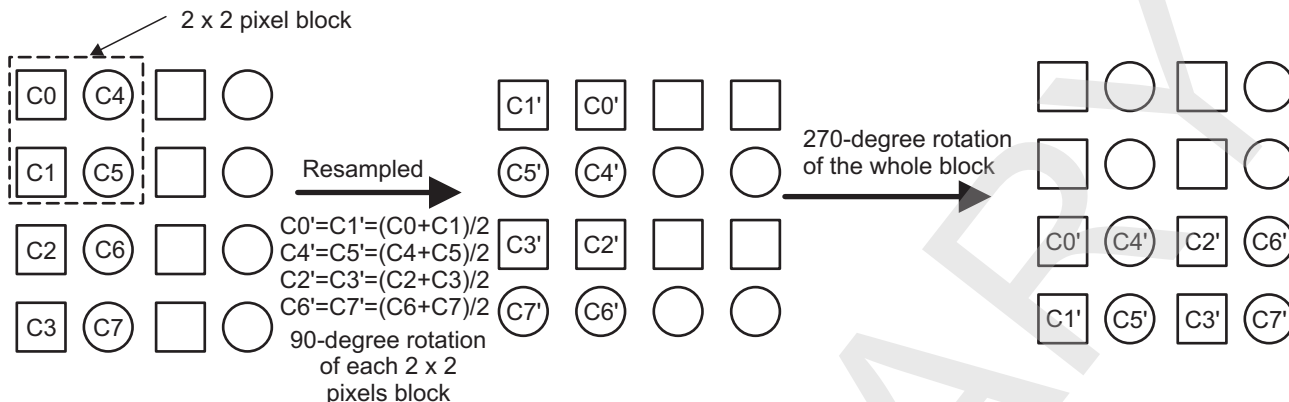
Thus, each 2N x 2N-pixel output block depends on only one 2N x 2N-pixel input block. After resampling every 2 x 2-pixel block, it is rotated -90 degrees, and then the whole 2N x 2N block is rotated 90 degrees.

Figure 8-282. ISS SIMCOP ROT 90-Degree Rotation of YUV4:2:2 Data

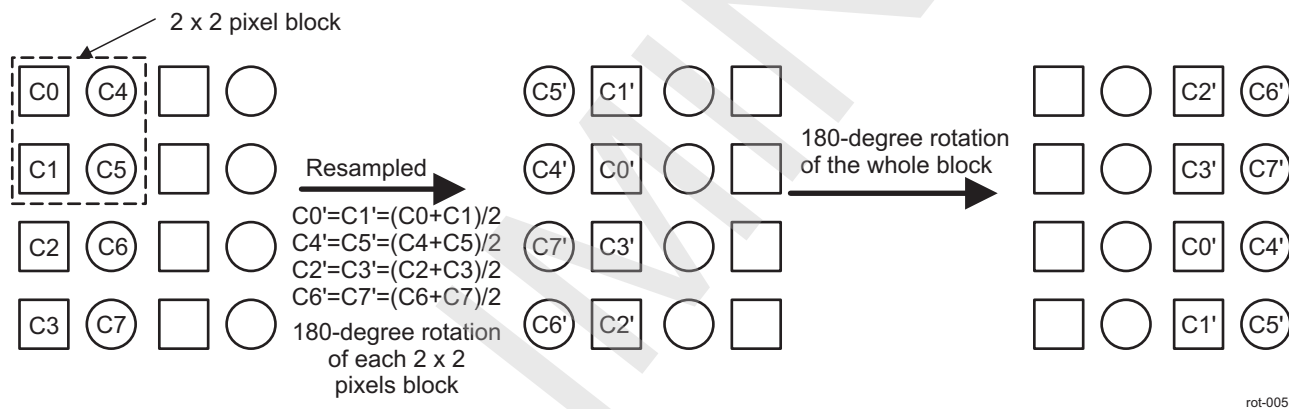


rot-003

The same algorithm is used for 270-degree rotation.

**Figure 8-283. ISS SIMCOP ROT 270-Degree Rotation of YUV4:2:2 Data**

For 180-degree rotation, there is no arithmetic operation; Chroma values are simply shifted horizontally before being rotated.

**Figure 8-284. ISS SIMCOP ROT 180-Degree Rotation of YUV4:2:2 Data**

**NOTE:** The intermediate process of 2 x 2 pixel block rotation, represented in the middle section in [Figure 8-282](#) through [Figure 8-284](#), does not involve data storage and reading into memory. This middle section is for better visualization of the process. The whole process is performed with a single memory reading and writing.

Zero-degree rotation is included for completeness of programming model, and involves only data copy.

#### 8.4.7.3.8.3 ISS SIMCOP ROT Multiple-Block Rotation

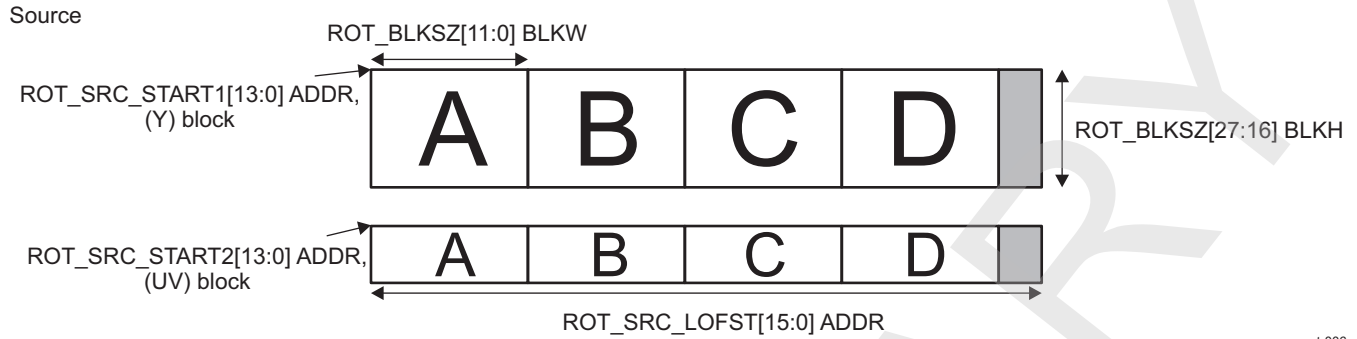
For generic 8-/16-/32-bit data rotation and YUV4:2:0/4:2:2 rotation, multiple-block operation per hardware task is supported.

[Figure 8-285](#) through [Figure 8-288](#) show the multiple-block rotation feature.

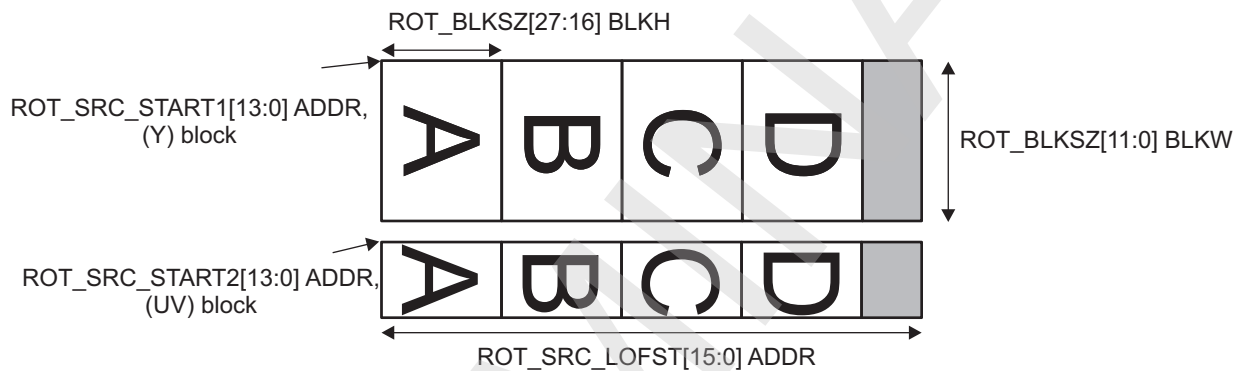
The multiple blocks are in the horizontal dimension, and the blocks are horizontally adjacent to one another. In other words, in the source data and 180-degree rotated data, address offset between blocks is the source block width times the data size.

In the 90-degree and 270-degree rotated data, address offset is the source block height times the data size. The source/destination line offsets must be sufficiently large to accommodate the multiple blocks.

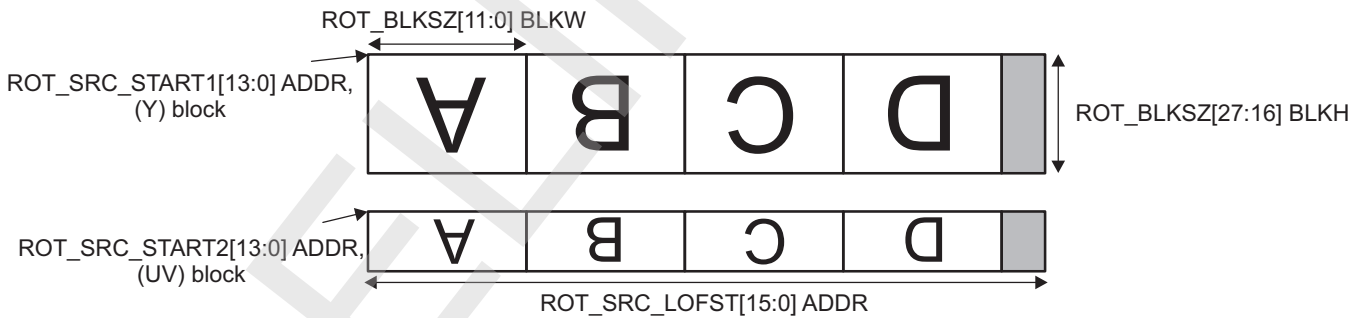
**Figure 8-285. ISS SIMCOP ROT Block Representation Before Rotation**



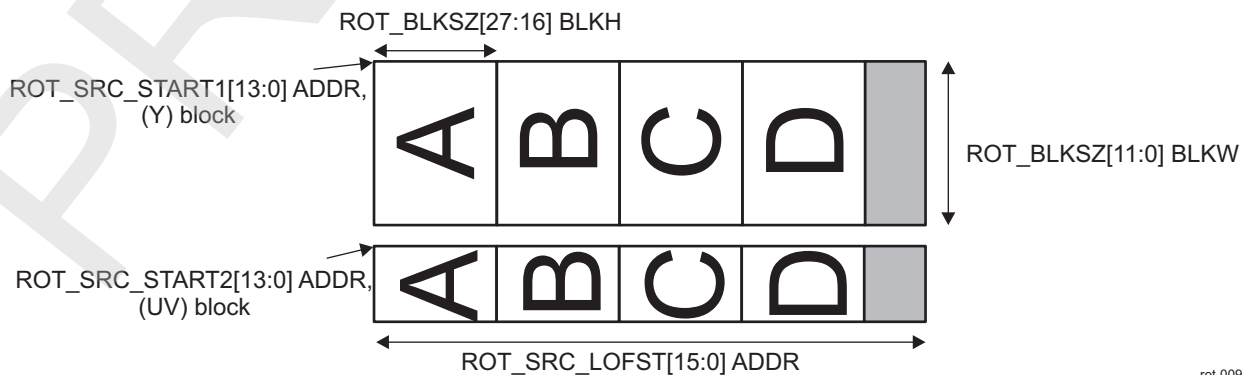
**Figure 8-286. ISS SIMCOP ROT 90-Degree Block Rotation**



**Figure 8-287. ISS SIMCOP ROT 180-Degree Block Rotation**



**Figure 8-288. ISS SIMCOP ROT 270-Degree Block Rotation**



The operation is configured with these parameters:

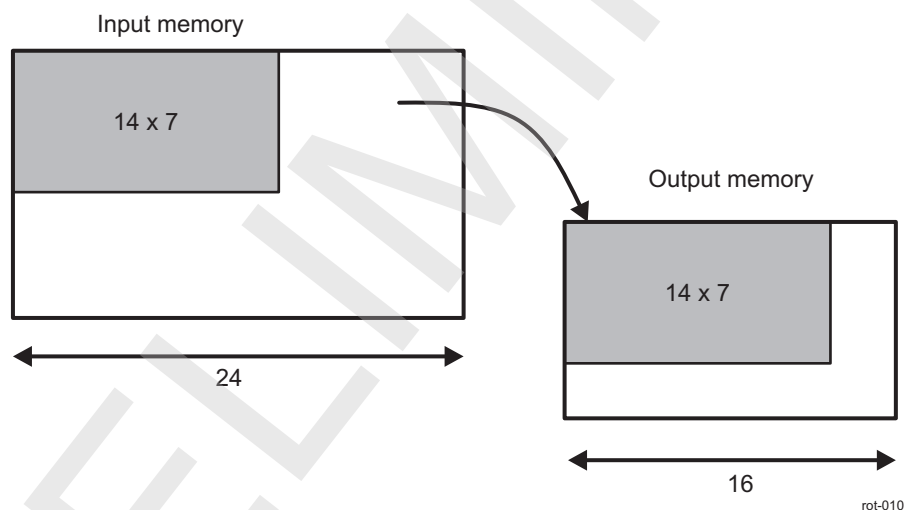
- **ROT\_CFG[2:0]** OP (rotation orientation)
- **ROT\_CFG[6:4]** FMT (data format)
- **ROT\_CFG[15:12]** NBLKS (number of blocks minus 1; register value of 0 means one block, etc.)
- **ROT\_BLKSZ[11:0]** BLKW (width of each block; all blocks have the same width/height)
- **ROT\_BLKSZ[27:16]** BLKH (height of each block)
- **ROT\_SRC\_START1[13:0]** ADDR (starting address of the first input block)
- **ROT\_SRC\_LOFST[15:0]** ADDR (line offset of input block)
- **ROT\_DST\_START1[13:0]** ADDR (starting address of the first output block)
- **ROT\_DST\_LOFST[15:0]** ADDR (line offset of output block)
- **ROT\_SRC\_START2[13:0]** ADDR (starting address of the first input UV block, if format is YUV4:2:0)
- **ROT\_DST\_START2[13:0]** ADDR (starting address of the first output UV block, if format is YUV4:2:0)

#### 8.4.7.3.8.4 ISS SIMCOP ROT Data-Block Shifting

The ROT supports shifting of 2-dimensional (2D) data blocks in 8-bit granularity. This function is intended to supplement the SIMCOP DMA module, which operates on 128-bit alignment.

Figure 8-289 is an example of a 14 × 7-byte data block copied from one location in input memory to another location in output memory.

**Figure 8-289. ISS SIMCOP ROT Data-Block Shifting Example**



The following parameters are specified:

- **ROT\_BLKSZ[11:0]** BLKW = 14 (in bytes)
- **ROT\_BLKSZ[27:16]** BLKH = 7 (in rows)
- **ROT\_SRC\_LOFST[15:0]** LOFST = 24 (in bytes)
- **ROT\_DST\_LOFST[15:0]** LOFST = 16 (in bytes)

---

**NOTE:** This function can be used to shift horizontally as well as vertically.

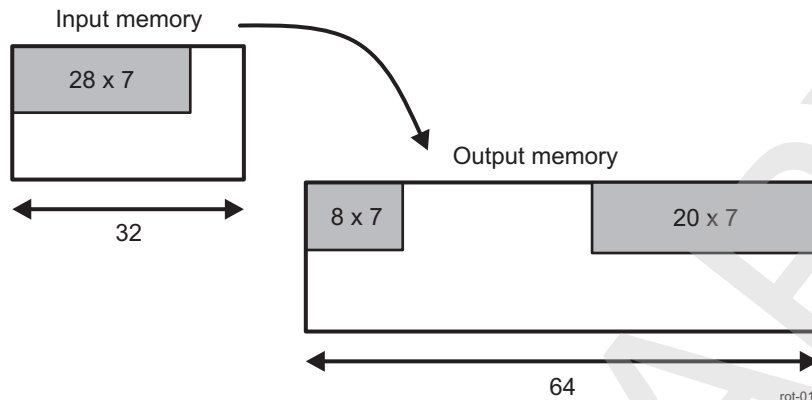
---

#### 8.4.7.3.8.5 ISS SIMCOP ROT Data-Block Horizontal Circular Shifting

The ROT supports shifting of 2D data blocks in 8-bit granularity, with horizontal circular buffer addressing. This function is intended to supplement the SIMCOP DMA module, which operates on 128-bit alignment.

Figure 8-290 is an example of a 28 × 7-byte data block copied from one location in input memory to another location in output memory, and horizontal wrap-around occurs in the output memory.

**Figure 8-290. ISS SIMCOP ROT Horizontal Circular Shifting Example**



The following parameters are specified:

- `ROT_BLKSZ[11:0]` BLKW = 28 (in bytes)
- `ROT_BLKSZ[27:16]` BLKH = 7 (in rows)
- `ROT_SRC_LOFST[15:0]` LOFST = 32 (in bytes)
- `ROT_DST_LOFST[15:0]` LOFST = 64 (in bytes)

Wrapping around is allowed in the input memory, in the output memory, or in both. For this mode of operation, the line offset of the source and destination must be a power of 2, and at least 32 bytes. Source and destination line offsets can be different. This function is used to handle misaligned SDRAM data. The following example shows that the line offset is at least the minimal transfer width, or the SDRAM burst. For target applications, 32 bytes is the minimal burst size to support.

---

**NOTE:** This function can be used to shift horizontally as well as vertically. Input and output blocks can also have different line offsets.

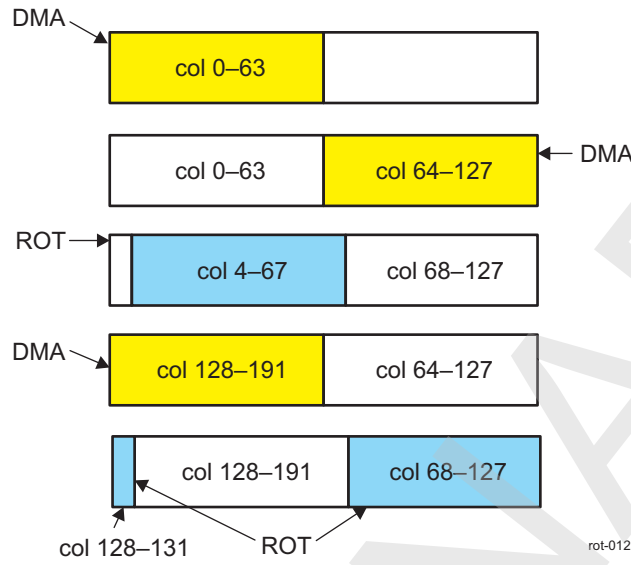
---

This mode of operation is intended to be used to manage 2D data blocks that are out of SDRAM alignment (for example, to crop out the left-most 2 pixels of the ISP YUV4:2:2 output image before JPEG compression). Assume that the ISP output is 64-byte SDRAM burst-aligned; thus, the starting address for JPEG input is 4 bytes out of alignment. Without the horizontal circular shifting function, the best solution is to use wide data blocks (for example, 128 pixels = 256 bytes wide) to reduce the bandwidth penalty of misalignment (costing five bursts to get four bursts worth of data). With horizontal circular shifting, the bandwidth penalty can be eliminated. In this example, the SIMCOP DMA is configured to transfer aligned data, and the ROT is used to perform the crop-out. A memory transaction log would be:

- The DMA writes columns 0–63, starting at byte address 0.
- The DMA writes columns 64–127, starting at byte address 64.
- The ROT module reads columns 4–67, starting at byte address 4.
- The DMA writes columns 128–191, starting at byte address 0.
- The ROT module reads columns 68–131, starting at byte address 68, and so on.

Figure 8-291 shows the data movement by the DMA and the ROT.

**Figure 8-291. ISS SIMCOP ROT Using Horizontal Circular Shifting to Manage Misaligned Data**



### 8.4.7.4 ISS SIMCOP ROT Module Programming Guide

#### 8.4.7.4.1 ISS SIMCOP ROT Low-Level Programming Models

This section describes the low-level programming sequences for configuration and use of the ROT.

##### 8.4.7.4.1.1 ISS SIMCOP ROT Global Initialization

###### 8.4.7.4.1.1.1 ISS SIMCOP ROT Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the ROT is to be used for the first time after a device reset. Initialization of surrounding modules is based on the integration and environment of the module.

Table 8-1424 describes the global initialization of the surrounding modules.

**Table 8-1424. ISS SIMCOP ROT Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
SIMCOP (clock management)	The ROT functional clock must be enabled. See <a href="#">Section 8.4.7.3.2, ISS SIMCOP ROT Clock Configuration</a> .
SIMCOP (power management)	The ROT must be enabled (SIMCOP_CLKCTRL[7] ROT_A). See <a href="#">Section 8.4.1.2.1, ISS SIMCOP Local Power and Clock Management</a> .
Buffers	Configure the ROT access to the image buffer inside the SIMCOP_HWSEQ_STEP_SWITCH_i register. See <a href="#">Section 8.4.2, ISS SIMCOP Hardware Sequencer and Buffers Module</a> .
SIMCOP IRQ merger	Configure SIMCOP to handle the ROT interrupts. See <a href="#">Section 8.4.1.2.3, Interrupt Merger</a> .
SIMCOP hardware sequencer	Configure the hardware sequencer to activate the ROT. See <a href="#">Section 8.4.2, ISS SIMCOP Hardware Sequencer and Buffers Module</a> .

###### 8.4.7.4.1.1.2 ISS SIMCOP ROT Global Initialization

Table 8-1425 describes the global initialization of the ROT.

**Table 8-1425. ISS SIMCOP ROT Global Initialization of the ROT**

Step	Register/Bit Field/Programming Model	Value
Enable the ROT.	ROT_CTRL[0] EN	0x0

###### 8.4.7.4.1.2 ISS SIMCOP ROT Operational Modes Configuration

###### 8.4.7.4.1.2.1 ISS SIMCOP ROT Rotation

The proper value for the trigger source bit (ROT\_CFG[8] TRIG\_SRC) is set depending on whether the operation is to be triggered by software, writing to MMR, or by the hardware sequencer. Table 8-1426 lists the procedure to set the value of the trigger source bit.

**Table 8-1426. ISS SIMCOP ROT Trigger Source Bit Value**

Step	Register/Bit Field/Programming Model	Value
Configure the degree of rotation.	ROT_CFG[2:0] OP	xxx
Configure the data format.	ROT_CFG[6:4] FMT	xxx
Specify the block width and height.	ROT_BLKSZ[11:0] BLKW ROT_BLKSZ[27:16] BLKH	xxx
Specify the input and output starting addresses.	ROT_SRC_START1[13:0] ADDR, ROT_DST_START1[13:0] ADDR	xxx
Specify the line offsets.	ROT_SRC_LOFST[15:0] LOFST, ROT_DST_LOFST[15:0] LOFST	xxx

**Table 8-1426. ISS SIMCOP ROT Trigger Source Bit Value (continued)**

Step	Register/Bit Field/Programming Model	Value
For UV data of YUV4:2:0, input and output second starting addresses.	<a href="#">ROT_SRC_START2</a> [13:0] ADDR, <a href="#">ROT_DST_START2</a> [13:0] ADDR	xxx

#### 8.4.7.4.1.2.2 ISS SIMCOP ROT Data-Block Shifting

[Table 8-1427](#) lists the bit field values for data-block shifting.

**Table 8-1427. ISS SIMCOP ROT Data-Block Shifting**

Step	Register/Bit Field/Programming Model	Value
Configure shifting.	<a href="#">ROT_CFG</a> [2:0] OP	xxx
Set the data shifting block.	<a href="#">ROT_CFG</a> [6:4] FMT	xxx
Specify the block width and height.	<a href="#">ROT_BLKSZ</a> [11:0] BLKW, <a href="#">ROT_BLKSZ</a> [27:16] BLKH	xxx
Specify the input and output starting addresses.	<a href="#">ROT_SRC_START1</a> [13:0] ADDR, <a href="#">ROT_DST_START1</a> [13:0] ADDR	xxx
Specify the line offsets.	<a href="#">ROT_SRC_LOFST</a> [15:0] LOFST, <a href="#">ROT_DST_LOFST</a> [15:0] LOFST	xxx

Once the input data and output buffer are available, the [ROT\\_CTRL](#)[0] EN bit is set to 1 to trigger the operation through software. Alternatively, DMA transfer(s) and the hardware sequencer are configured to trigger the ROT with the hardware start signal when appropriate.



8.4.7.5 ISS SIMCOP ROT Register Manual

8.4.7.5.1 ISS SIMCOP ROT Instance Summary

Table 8-1428 lists the ROT instance.

Table 8-1428. ISS SIMCOP ROT Instance Summary

Module Name	Base Address L3 Interconnect	Base Address Cortex-M3 Private Access	Size
ROT	0x5202 0700	0x5506 0700	64 bytes

**NOTE:** Private Access is an access which is not using the L3/L4 interconnects.

8.4.7.5.2 ISS SIMCOP ROT Registers

8.4.7.5.2.1 ISS SIMCOP ROT Register Summary

Table 8-1429 summarizes the ROT registers.

Table 8-1429. ROT Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-M3 Private Access
ROT_REVISION	R	32	0x0000 0000	0x5202 0700	0x5506 0700
ROT_CTRL	W	32	0x0000 0004	0x5202 0704	0x5506 0704
ROT_CFG	RW	32	0x0000 0008	0x5202 0708	0x5506 0708
ROT_BLKSZ	RW	32	0x0000 000C	0x5202 070C	0x5506 070C
ROT_SRC_START1	RW	32	0x0000 0010	0x5202 0710	0x5506 0710
ROT_SRC_LOFST	RW	32	0x0000 0014	0x5202 0714	0x5506 0714
ROT_DST_START1	RW	32	0x0000 0018	0x5202 0718	0x5506 0718
ROT_DST_LOFST	RW	32	0x0000 001C	0x5202 071C	0x5506 071C
ROT_SRC_START2	RW	32	0x0000 0020	0x5202 0720	0x5506 0720
ROT_DST_START2	RW	32	0x0000 0024	0x5202 0724	0x5506 0724

8.4.7.5.2.2 ISS SIMCOP ROT Register Descriptions

Table 8-1430 through Table 8-1448 describe the ROT registers.

Table 8-1430. ROT\_REVISION

Address Offset	0x0000 0000	Instance	ROT
Physical Address	See Table 8-1429.		
Description	Module revision		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	IP Revision	R	TI Internal Data

**Table 8-1431. Register Call Summary for Register ROT\_REVISION**

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[0\]](#)

**Table 8-1432. ROT\_CTRL**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	ROT
<b>Physical Address</b>	See <a href="#">Table 8-1429</a> .		
<b>Description</b>	Control		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUSY	RESERVED										EN				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	BUSY	Idle/busy status (read-only) 0 = Idle, 1 = Busy	R	0
14:1	RESERVED		R	0x0000
0	EN	Module enable, writing 1 starts the module; always reads as 0.	W	0

**Table 8-1433. Register Call Summary for Register ROT\_CTRL**

ISS SIMCOP ROT Module Functional Description

- [ISS SIMCOP ROT Synchronization: \[0\]](#)

ISS SIMCOP ROT Module Programming Guide

- [ROT Module Global Initialization: \[1\]](#)
- [Data-Block Shifting: \[2\]](#)

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[3\]](#)

**Table 8-1434. ROT\_CFG**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	ROT
<b>Physical Address</b>	See <a href="#">Table 8-1429</a> .		
<b>Description</b>	Configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NBLKS		RESERVED	AUTOGATING	TRIG_SRC	RESERVED	FMT		RESERVED	OP						

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	NBLKS	Number of blocks minus 1 0: 1 block, 1: 2 blocks, etc.	RW	0x0
11:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9	AUTOGATING	Internal interconnect and functional clock gating 0: Interconnect and functional clocks are free-running. 1: Automatic clock gating is applied, based on the interface activity for the interface clock, and on the functional activity for the functional clocks.	RW	1
8	TRIG_SRC	Trigger source 0 = MMR write 1 = Hardware start signal	RW	0
7	RESERVED		R	0
6:4	FMT	Data format 0 = 8-bit data 1 = 16-bit data 2 = 32-bit data 3 = YUV4:2:2 data 4 = YUV4:2:0 data	RW	0x0
3	RESERVED		R	0
2:0	OP	Operation 0 = Rotate 0 degree 1 = Rotate 90 degrees 2 = Rotate 180 degrees 3 = Rotate 270 degrees 4 = Data shift (FMT = 0) 5 = Horizontal circular (FMT = 0) shift	RW	0x0

**Table 8-1435. Register Call Summary for Register ROT\_CFG**

ISS SIMCOP ROT Module Functional Description

- [ISS SIMCOP ROT Module Clock Configuration: \[0\]](#)
- [ISS SIMCOP ROT Module Power Management: \[1\]](#)
- [ISS SIMCOP ROT Synchronization: \[2\] \[3\] \[4\]](#)
- [Multiple-Block Rotation: \[5\] \[6\] \[7\]](#)

ISS SIMCOP ROT Module Programming Guide

- [Rotation: \[8\] \[9\] \[10\]](#)
- [Data-Block Shifting: \[11\] \[12\]](#)

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[13\]](#)

**Table 8-1436. ROT\_BLKSZ**

<b>Address Offset</b>	0x0000 000C																																																																																						
<b>Physical Address</b>	See <a href="#">Table 8-1429</a> .																<b>Instance</b>																ROT																																																						
<b>Description</b>	Block size																																																																																						
<b>Type</b>	RW																																																																																						
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:5%;">31</td><td style="width:5%;">30</td><td style="width:5%;">29</td><td style="width:5%;">28</td><td style="width:5%;">27</td><td style="width:5%;">26</td><td style="width:5%;">25</td><td style="width:5%;">24</td><td style="width:5%; background-color: #ffffcc;">23</td><td style="width:5%; background-color: #ffffcc;">22</td><td style="width:5%; background-color: #ffffcc;">21</td><td style="width:5%; background-color: #ffffcc;">20</td><td style="width:5%; background-color: #ffffcc;">19</td><td style="width:5%; background-color: #ffffcc;">18</td><td style="width:5%; background-color: #ffffcc;">17</td><td style="width:5%; background-color: #ffffcc;">16</td><td style="width:5%;">15</td><td style="width:5%;">14</td><td style="width:5%;">13</td><td style="width:5%;">12</td><td style="width:5%;">11</td><td style="width:5%;">10</td><td style="width:5%;">9</td><td style="width:5%;">8</td><td style="width:5%; background-color: #ffffcc;">7</td><td style="width:5%; background-color: #ffffcc;">6</td><td style="width:5%; background-color: #ffffcc;">5</td><td style="width:5%; background-color: #ffffcc;">4</td><td style="width:5%; background-color: #ffffcc;">3</td><td style="width:5%; background-color: #ffffcc;">2</td><td style="width:5%; background-color: #ffffcc;">1</td><td style="width:5%; background-color: #ffffcc;">0</td> </tr> <tr> <td colspan="8">RESERVED</td><td colspan="8">BLKH</td><td colspan="8">RESERVED</td><td colspan="8">BLKW</td> </tr> </table>																								31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED								BLKH								RESERVED								BLKW							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																								
RESERVED								BLKH								RESERVED								BLKW																																																															
Bits	Field Name	Description	Type	Reset																																																																																			
31:28	RESERVED		R	0x0																																																																																			
27:16	BLKH	Block height, in pixels (YUV4:2:0/4:2:2) or number of rows (8-/16-/32-bit). Should be a multiple of 8, and at least 8, for rotation. Should be at least 1 for data shifting.	RW	0x000																																																																																			
15:12	RESERVED		R	0x0																																																																																			
11:0	BLKW	Block width, in pixels (YUV4:2:0/4:2:2) or data units (8-/16-/32-bit). Should be a multiple of 8, and at least 8, for rotation. Should be at least 4 for data shifting.	RW	0x000																																																																																			

**Table 8-1437. Register Call Summary for Register ROT\_BLKSZ**

ISS SIMCOP ROT Module Functional Description

- [Multiple-Block Rotation: \[0\] \[1\]](#)
- [Data-Block Shifting: \[2\] \[3\]](#)
- [Data-Block Horizontal Circular Shifting: \[4\] \[5\]](#)

ISS SIMCOP ROT Module Programming Guide

- [Rotation: \[6\] \[7\]](#)
- [Data-Block Shifting: \[8\] \[9\]](#)

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[10\]](#)

**Table 8-1438. ROT\_SRC\_START1**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	ROT
<b>Physical Address</b>	See <a href="#">Table 8-1429</a> .		
<b>Description</b>	Source starting address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDR															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	ADDR	Byte address Should be a multiple of 8 for rotation. No constraint for data shifting.	RW	0x0000

**Table 8-1439. Register Call Summary for Register ROT\_SRC\_START1**

ISS SIMCOP ROT Module Functional Description

- [Multiple-Block Rotation: \[0\]](#)

ISS SIMCOP ROT Module Programming Guide

- [Rotation: \[1\]](#)
- [Data-Block Shifting: \[2\]](#)

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[3\]](#)

**Table 8-1440. ROT\_SRC\_LOFST**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	ROT
<b>Physical Address</b>	See <a href="#">Table 8-1429</a> .		
<b>Description</b>	Source line offset		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOFST															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	LOFST	Line offset in bytes Should be a multiple of 8 for rotation and normal data shifting. Should be a power of 2 and at least 32 for horizontal circular shifting.	RW	0x0000

**Table 8-1441. Register Call Summary for Register ROT\_SRC\_LOFST**

ISS SIMCOP ROT Module Functional Description

- [Multiple-Block Rotation: \[0\]](#)
- [Data-Block Shifting: \[1\]](#)
- [Data-Block Horizontal Circular Shifting: \[2\]](#)

ISS SIMCOP ROT Module Programming Guide

- [Rotation: \[3\]](#)
- [Data-Block Shifting: \[4\]](#)

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[5\]](#)

**Table 8-1442. ROT\_DST\_START1**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	ROT
<b>Physical Address</b>	See <a href="#">Table 8-1429</a> .		
<b>Description</b>	Destination starting address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDR															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	ADDR	Byte address Should be a multiple of 8 for rotation. No constraint for data shifting.	RW	0x0000

**Table 8-1443. Register Call Summary for Register ROT\_DST\_START1**

ISS SIMCOP ROT Module Functional Description

- [Multiple-Block Rotation: \[0\]](#)

ISS SIMCOP ROT Module Programming Guide

- [Rotation: \[1\]](#)
- [Data-Block Shifting: \[2\]](#)

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[3\]](#)

**Table 8-1444. ROT\_DST\_LOFST**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	ROT
<b>Physical Address</b>	See <a href="#">Table 8-1429</a> .		
<b>Description</b>	Destination line offset		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOFST															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	LOFST	Line offset in bytes Should be a multiple of 8 for rotation and normal data shifting. Should be a power of 2 and at least 32 for horizontal circular shifting.	RW	0x0000

**Table 8-1445. Register Call Summary for Register ROT\_DST\_LOFST**

ISS SIMCOP ROT Module Functional Description

- [Multiple-Block Rotation: \[0\]](#)
- [Data-Block Shifting: \[1\]](#)
- [Data-Block Horizontal Circular Shifting: \[2\]](#)

ISS SIMCOP ROT Module Programming Guide

- [Rotation: \[3\]](#)
- [Data-Block Shifting: \[4\]](#)

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[5\]](#)

**Table 8-1446. ROT\_SRC\_START2**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	ROT
<b>Physical Address</b>	See <a href="#">Table 8-1429</a> .		
<b>Description</b>	Source starting address 2 (only form YUV4:2:0 FMT = 2)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDR															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	ADDR	Byte address, should be a multiple of 8.	RW	0x0000

**Table 8-1447. Register Call Summary for Register ROT\_SRC\_START2**

ISS SIMCOP ROT Module Functional Description

- [Multiple-Block Rotation: \[0\]](#)

ISS SIMCOP ROT Module Programming Guide

- [Rotation: \[1\]](#)

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[2\]](#)

**Table 8-1448. ROT\_DST\_START2**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	ROT
<b>Physical Address</b>	See <a href="#">Table 8-1429</a> .		
<b>Description</b>	Destination starting address 2 (only form YUV4:2:0 FMT = 2)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ADDR															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:0	ADDR	Byte address, should be a multiple of 8.	RW	0x0000

**Table 8-1449. Register Call Summary for Register ROT\_DST\_START2**

ISS SIMCOP ROT Module Functional Description

- [Multiple-Block Rotation: \[0\]](#)

ISS SIMCOP ROT Module Programming Guide

- [Rotation: \[1\]](#)

---

**Table 8-1449. Register Call Summary for Register ROT\_DST\_START2 (continued)**

---

ISS SIMCOP ROT Module Register Manual

- [ROT Module Register Summary: \[2\]](#)
- 

PRELIMINARY

PRELIMINARY



## Face Detect

This chapter describes the features and functions of the Face Detect (FDIF) module of the multimedia device.

Topic	Page
<b>9.1 Face Detect Overview .....</b>	<b>2042</b>
<b>9.2 FDIF Integration .....</b>	<b>2044</b>
<b>9.3 FDIF Functional Description .....</b>	<b>2046</b>
<b>9.4 FDIF Programming Guide .....</b>	<b>2050</b>
<b>9.5 FDIF Register Manual .....</b>	<b>2053</b>

## 9.1 Face Detect Overview

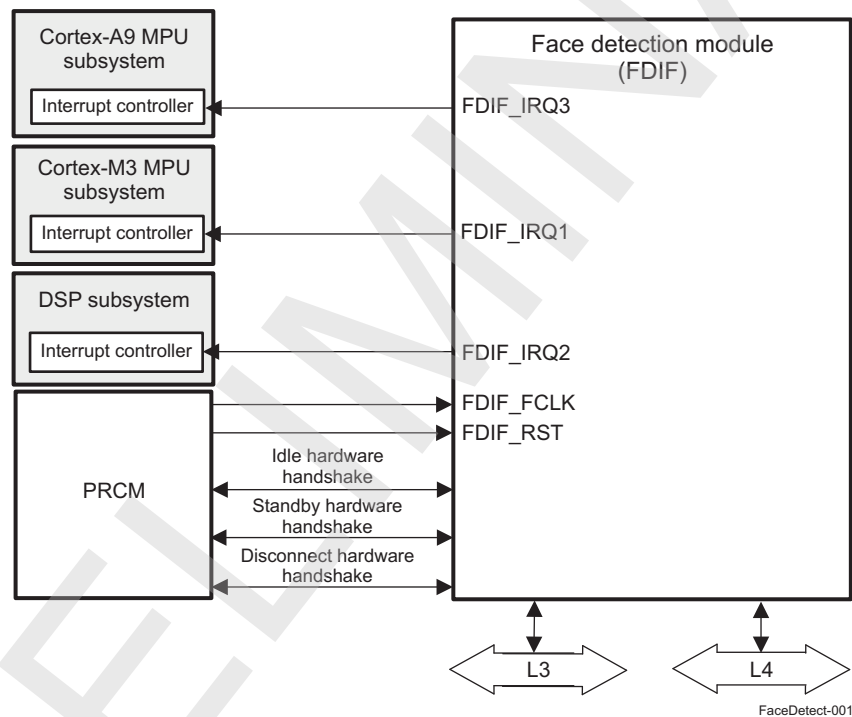
The FDIF module performs face detection within a picture stored in memory (QVGA luminance data resolution). This module is typically used for video encoding, face-based priority auto-focusing, or red-eye removal.

The face detect (FD) core is a standalone module: it embeds its own direct memory access (DMA) engine for accessing data in memory. The FD core is under microprocessor unit (MPU) control for its initialization and to start the processing operation.

The FD core supports single input resolution (QVGA) in a single format (8-bit luma). The FD core requires the input image to be stored in synchronous dynamic random access memory (SDRAM). The core also requires working memory mapped in SDRAM. The input image data (320x240) requires 75KB for the SDRAM and 51.25KB for the working memory.

Figure 9-1 is an overview of the FDIF. Three interrupt lines go to the three central processing units (CPUs) to allow control of the FD core module by different processors, but only one interrupt at a time can be active. The three interrupt lines provide flexibility for controlling the module.

**Figure 9-1. Face Detect Highlight**



### 9.1.1 Main Features

The FDIF has the following features:

- Two interconnects interface:
  - 32-bit-wide level 4 (L4) interface for register configuration
  - 32-bit-wide level 3 (L3) interface for command
- Supports the IDLE, STANDBY, and DISCONNECT protocols
- 3rd party IP features: For information about the input image features and detection capabilities, see [Table 9-1](#).

**Table 9-1. 3rd Party IP Features**

Items	Description
Input image size	QVGA input image size H x V = 320 x 240

**Table 9-1. 3rd Party IP Features (continued)**

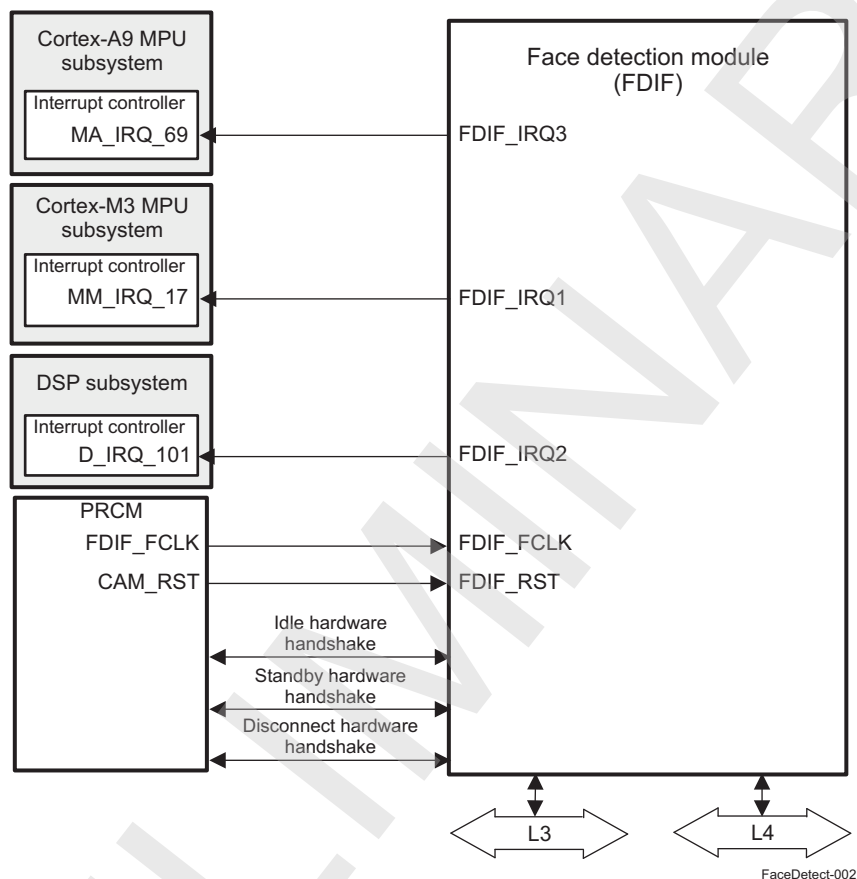
<b>Items</b>	<b>Description</b>
Input data format	8-bit gray scale data in little-endian format (color pictures must be converted before FD can be applied) 0x00 = black and 0xFF = white
Face inclination	± 45 degrees
Face direction	Up/down: ± 30 degrees Left/right: ± 60 degrees
Maximum detection count	35 faces
Detection direction	The face orientation must be selected from the following possibilities: 0 degrees: Faces are vertical. + 90 degrees: Faces are rotated right by 90 degrees. – 90 degrees: Faces are rotated left by 90 degrees.
Detection minimum face size	Four grades are available: 20 pixels 25 pixels 32 pixels 40 pixels
Detection area	Detection start position: X = 0 to 160 Y = 0 to 120 Detection area size: X = 160 to 320 Y = 120 to 240
Detection result	The following information is provided by the module for each face: Size Position Angle Confidence level

## 9.2 FDIF Integration

This section describes the integration of FDIF in the device, including information about clocks, resets, and hardware requests.

Figure 9-2 shows the integration of the module in the device.

**Figure 9-2. FDIF Integration**



**NOTE:** For more information about the IDLE, STANDBY, and DISCONNECT hardware handshakes, see [Chapter 3, Power, Reset, and Clock Management](#).

Table 9-2 through Table 9-4 summarize the integration of the module in the device. For more information about the power domain or the clock and reset signals, see [Chapter 3, Power, Reset, and Clock Management](#).

**Table 9-2. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
Face Detect	PD_CAM	No	L3

**Table 9-3. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Face detect	FDIF_FCLK	FDIF_FCLK	PRCM	For information about PRCM clock gating and management, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .

**Table 9-3. Clocks and Resets (continued)**

Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Face detect	FDIF_RST	CAM_RST	PRCM	For information about PRCM reset sources and distribution, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .

**Table 9-4. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
Face detect	FDIF_IRQ3	MA_IRQ_69	Cortex™-A9 MPU subsystem	FDIF interrupt to the Cortex-A9 MPU subsystem
Face detect	FDIF_IRQ1	MM_IRQ_17	Cortex™-M3 MPU subsystem	FDIF interrupt to the Cortex-M3 MPU subsystem
Face detect	FDIF_IRQ2	D_IRQ_101	DSP subsystem	FDIF interrupt to the digital signal processor (DSP) subsystem

**No DMA Requests**

**NOTE:** For more information about interrupt sources, see [Section 9.3.4, Interrupts and Events](#).

## 9.3 FDIF Functional Description

### 9.3.1 FDIF Block Description

The FDIF is a standalone module. It has a port connected to the L4 interconnect, which is used for configuration, and a port connected to the L3 interconnect, which is used to read and write data to the system memory.

#### CAUTION

These ports are accessible only in 32 bits.

The FDIF core contains the following submodules:

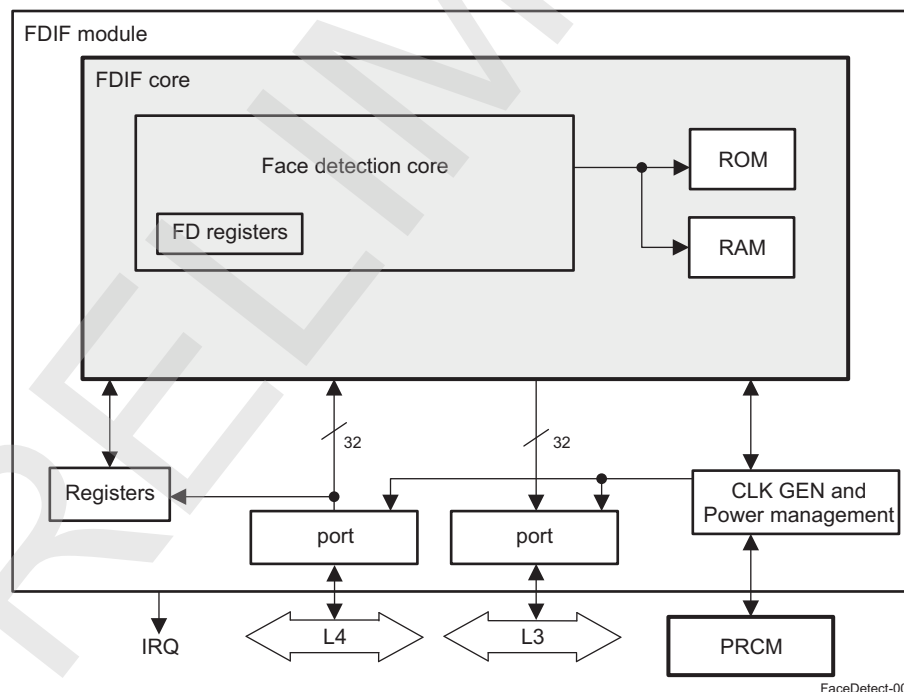
- Face detection (FD) IP core
- RAM and ROM memories

The FDIF module wraps the FD core for integration in the device. The following submodules are added:

- Clock generation
- Interrupt generation
- Power management
- L3 and L4 interface ports: Two ports required to decorrelate the FDIF clocks from other system clocks, one on the L3 port interface, and one on the L4 port interface.

Figure 9-3 shows the overall architecture of the FDIF.

**Figure 9-3. FDIF Block Diagram**



### 9.3.2 Software Reset

The module is reset by writing the `FDIF_SYSCONFIG[0]` SOFTRESET bit to 1. The bit is automatically reset by hardware. During reads, it always returns 0.

The FD core also provides a software reset setup by the `FD_CTRL[0]` SRST bit, but it is strongly recommended to use only the `FDIF_SYSCONFIG[0]` SOFTRESET bit.

### 9.3.3 System Power Management

There is one clock domain in FDIF: the L4 port domain and L3 port domain are synchronized with the module functional clock domain, FDIF\_FCLK.

Table 9-5 lists the power-management features available for FDIF.

**NOTE:**

- For more information about source clock gating, see [Chapter 3, Power, Reset, and Clock Management](#).
- For descriptions of idle mode and standby mode and the power, reset, and clock management (PRCM) power handshake, see [Chapter 3, Power, Reset, and Clock Management](#).

**Table 9-5. Local Power-Management Features**

Feature	Registers	Description
Clock autogating	No software control for it.	It is always enabled.
Slave idle modes	FDIF_SYSCONFIG[3:2] IDLEMODE	Force-idle, no-idle, and smart-idle modes are available
Clock activity	N/A	
Master standby modes	FDIF_SYSCONFIG[5:4] STANDBYMODE	Force-standby, no-standby, and smart-standby mode modes are available. See <a href="#">Section 9.3.3.2.1</a> for more information.
Global wake-up enable	N/A	
Wake-up sources enable	N/A	

#### 9.3.3.1 Autogating

The FDIF performs clock autogating whenever possible. There is no software control for it; it is always enabled.

#### 9.3.3.2 PRCM Hardware Handshake

The FDIF supports the IDLE protocol between the L4 port and the PRCM module and the STANDBY protocol between the L3 port and the PRCM module. The FDIF also supports the DISCONNECT protocol on the L4 and L3 ports.

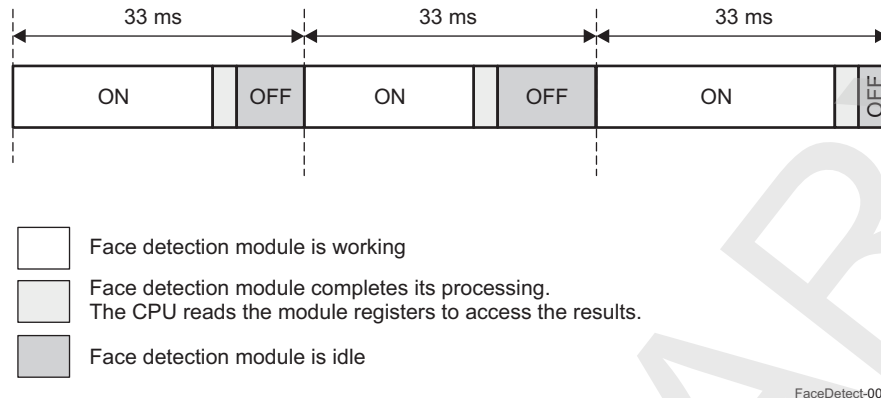
The FDIF first initiates the STANDBY protocol with the PRCM module, which in turn generates the IDLE protocol with the FDIF. The functional clock is gated by the PRCM module only when the IDLE and STANDBY power-management protocols are complete.

For more information about these protocols, see [Chapter 3, Power, Reset, and Clock Management](#).

##### 9.3.3.2.1 Protocol Transitions

The FDIF goes to IDLE as soon as it is done processing and the MPU has read the results of the computation.

Figure 9-4 shows the principle. The 33-ms period corresponds to 30 fps, which is usually the target for imaging applications. Time spent in READ state (= ON state) is always the same, and the time in OFF state depends on the time spent in ON state.

**Figure 9-4. FDIF Power-Management Transitions**

### 9.3.3.2.1.1 Normal Mode to Idle Mode

Software programs `FDIF_SYSCONFIG[5:4] STANDBYMODE = 0x2` to program the power-management mode.

When its processing is complete, the FD core generates the `FINISH_IRQ` event. At this time the `FD_CTRL[1] RUN` bit reflects that the FD core is idle.

Software running on the MPU reads the results of the FD algorithm. When it finishes reading the results, it clears the `FDIF_IRQSTATUS_RAW_j[8] FINISH_IRQ` bit and writes `FDIF_CTRL[5] MSTANDBY = 1` to initiate the MStandby signal generation. For more information about the STANDBY and IDLE hardware handshakes, see [Chapter 3, Power, Reset, and Clock Management](#).

The FDIF clock can be gated only when the STANDBY and IDLE protocols are complete.

### 9.3.3.2.1.2 Idle Mode to Normal Mode

The PRCM module deasserts a signal to the FDIF under software control so that the module can switch back to normal mode. Then, the module functional clock is ensured and stable. Software disables the MStandby signal generation by writing `FDIF_CTRL[5] MSTANDBY = 0`, and then polls for `FDIF_CTRL[6] MSTANDBY_HDSHK = 1` to ensure that the power-management handshaking is complete. For more information about the standby and idle modes, see [Chapter 3, Power, Reset, and Clock Management](#).

## 9.3.4 Interrupts and Events

The FDIF has three interrupt lines (`FDIF_IRQ1`, `FDIF_IRQ2`, and `FDIF_IRQ3`) to enable flexibility for controlling the module. Each event can be enabled to only a single interrupt line (that is, to only one interrupt controller), because only one interrupt at a time can be active.

The FDIF generates events listed in [Table 9-6](#). An interrupt can be enabled by using the `FDIF_IRQENABLE_SET_j` register and can be disabled by using the `FDIF_IRQENABLE_CLR_j` register.

**Table 9-6. Interrupts and Events Description**

Event Name	Description
<code>FDIF_IRQSTATUS_RAW_j / FDIF_IRQSTATUS_j[8] FINISH_IRQ</code>	This event is generated by the FD IP core. It flags the completion of the processing by the IP.
<code>FDIF_IRQSTATUS_RAW_j / FDIF_IRQSTATUS_j[0] ERR_IRQ</code>	This event flags that the L3 port has received an error.



The [FDIF\\_IRQSTATUS\\_RAW\\_j](#) and [FDIF\\_IRQSTATUS\\_j](#) registers give the interrupt status. The difference between them is that the [FDIF\\_IRQSTATUS\\_RAW\\_j](#) register is set even if events are not enabled.

The [FDIF\\_IRQSTATUS\\_j](#) register can be used to clear events.

### 9.3.5 Typical Use

Typically, FD processing is applied to buffers provided by the ISP module, already in the correct format.

To enable FD processing, the 32-bit address of the input image in memory must be set up in the [FDIF\\_PICADDR](#) register, and the 32-bit address of the working memory must be set up in the [FDIF\\_WKADDR](#) register.

The detection condition settings can be set with the following registers:

- [FD\\_DCOND](#)
- [FD\\_STARTX](#)
- [FD\\_STARTY](#)
- [FD\\_SIZEX](#)
- [FD\\_STARTY](#)

For more information, see [Section 9.4, FDIF Programming Guide](#).

When the [FD\\_CTRL\[1\]](#) RUN bit is set to 1, the process starts. When the process is complete ([FD\\_CTRL\[2\]](#) FINISH = 1), the results are available in the following registers:

- [FD\\_DNUM](#)
- [FD\\_CENTERX\\_j](#)
- [FD\\_CENTERY\\_j](#)
- [FD\\_CONFSIZE\\_j](#)
- [FD\\_ANGLE\\_j](#)

For more information, see [Section 9.4, FDIF Programming Guide](#).

### 9.3.6 Performance Parameters

[Table 9-7](#) lists the register settings that affect performance.

**Table 9-7. Performance Parameters**

Parameter	Comments
<a href="#">FD_DCOND[1:0]</a> MIN	Sets the minimum face size. The permitted values are 20x20, 25x25, 32x32, and 40x40. A high value leads to higher performance.
<a href="#">FD_DCOND[3:2]</a> DIR	Sets the detection direction setting. The permitted values are UP, RIGHT, and LEFT. The UP value should lead to better performance than RIGHT and LEFT.
<a href="#">FD_LHIT[3:0]</a> LHIT	Sets the detection threshold. The permitted values are in the range of 0 to 9. A low value increases the FD probability detection, but it also increases false detections.

### 9.3.7 L3 Interconnect Parameters

The FDIF is an initiator on the L3 interconnect. L3 accesses can be configured with the [FDIF\\_CTRL\[4:1\]](#) MAX\_TAGS bit field and the [FDIF\\_CTRL\[0\]](#) WRNP bit. For more information about these parameters, see [Section 13.2 L3 interconnect](#).

## 9.4 FDIF Programming Guide

### 9.4.1 FDIF Low-level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the FDIF.

#### 9.4.1.1 Global Initialization

##### 9.4.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the FDIF is to be used for the first time after a device reset. This initialization of the surrounding modules is based on the integration of the FDIF.

For more information, see [Section 9.2, FDIF Integration](#).

**Table 9-8. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	FDIF_FCLK functional clock must be enabled. See <a href="#">Section 3.6.22, CD_CAM Clock Domain</a> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Cortex-A9 MPU interrupt controller (INTC) (or Cortex-M3 MPU INTC or DSP INTC)	The Cortex-A9 MPU (or Cortex-M3 MPU or DSP) interrupt controller must be configured to enable the interrupt request generation to the Cortex-A9 MPU (or Cortex-M3 MPU or DSP) subsystem when interrupt requests are generated by the FDIF. See the respective Functional Description in <a href="#">Chapter 17, Interrupt Controller</a> , for the MPUs <a href="#">Section 17.4.1</a> , <a href="#">Section 17.4.2</a> , and the DSP interrupt request lines in <a href="#">Chapter 5, DSP Subsystem</a> , for the DSP.
Interconnect (L3 and L4)	For more information, see <a href="#">Section 13.2, L3 Interconnect</a> , and <a href="#">Section 13.3, L4 Interconnect</a> .

##### 9.4.1.1.2 FDIF Global Initialization

###### 9.4.1.1.2.1 Main Sequence – FDIF Global Initialization

[Table 9-9](#) lists the procedure to initialize the FDIF after a power-on or software reset.

**Table 9-9. FDIF Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Perform a software reset.	FDIF_SYSCONFIG[0] SOFTRESET	0x1
Wait until reset is finished.	FDIF_SYSCONFIG[0] SOFTRESET	0x0
Set the maximum interconnect tags.	FDIF_CTRL[4:1] MAX_TAGS	0xA
Enable ERROR type Interrupt.	FDIF_IRQENABLE_SET_j[0] ERR_IRQ	0x1

## 9.4.1.2 FD Operational Modes Configuration

### 9.4.1.2.1 FD Processing Modes

#### 9.4.1.2.1.1 Main Sequence – FDIF Polling Method

Step	Register/Bit Field/Programming Model	Value
Set image parameters.	See Section 9.4.1.2.1.3, Subsequence – Set Image Parameters.	
Request the processing.	FD_CTRL[1] RUN	0x1
Wait until process is finished.	FD_CTRL[2] FINISH	0x1
Read the results.	See Section 9.4.1.2.1.4	

#### 9.4.1.2.1.2 Main Sequence – FDIF Interrupt Method

Step	Register/Bit Field/Programming Model	Value
Set image parameters.	See Section 9.4.1.2.1.3, Subsequence – Set Image Parameters.	
Enable FINISH type interrupt.	FDIF_IRQENABLE_SET_j[8] FINISH IRQ	0x1
Request the processing.	FD_CTRL[1] RUN	0x1

#### WHEN INTERRUPT OCCURS:

Step	Register/Bit Field/Programming Model	Value
<b>IF:</b> This is FD Process Complete Event?	FDIF_IRQSTATUS_j[8] FINISH_IRQ	0x1
Clear (Disable) FINISH type interrupt.	FDIF_IRQENABLE_CLR_j[8] FINISH IRQ	0x1
Read the results	See Section 9.4.1.2.1.4, Subsequence – Read the Results.	
<b>ELSE</b>		
Clear (Disable) ERROR type Interrupt.	FDIF_IRQENABLE_CLR_j[0] ERR_IRQ	0x1
Perform a software reset.	FDIF_SYSCONFIG[0] SOFTRESET	0x1
For possible errors and the explanations of them, see Chapter 13, Interconnect.		
<b>ENDIF</b>		

#### 9.4.1.2.1.3 Subsequence – Set Image Parameters

Step	Register/Bit Field/Programming Model	Value
Set input image 32-bit address.	FDIF_PICADDR[31:5] ADDR	0x----
Set working memory 32-bit address.	FDIF_WKADDR[31:5] ADDR	0x----
Set the minimum face size.	FD_DCOND[1:0] MIN	0x-
Set the face direction.	FD_DCOND[3:2] DIR	0x-
Set the horizontal start position of the detection area.	FD_STARTX[7:0] STARTX	0x-
Set the vertical start position of the detection area.	FD_STARTY[6:0] STARTY	0x-
Set the horizontal start size of the detection area.	FD_SIZEX[8:0] SIZEX	0x-
Set the vertical start size of the detection area.	FD_SIZEY[7:0] SIZEY	0x-
Set the detection threshold.	FD_LHIT[3:0] LHIT	0x-

#### 9.4.1.2.1.4 Subsequence – Read the Results

Step	Register/Bit Field/Programming Model	Value
For each face detected, read the following parameters by processing loop until DNUM = 0:		
Read X <sub>i</sub> coordinate	FD_CENTERX <sub>i</sub> [8:0] CENTERX	0x-
Read Y <sub>i</sub> coordinate	FD_CENTERY <sub>i</sub> [7:0] CENTERY	0x-
Read confidence level	FD_CONF <sub>i</sub> [11:8] CONF	0x-
For each face read size <sub>i</sub>	FD_CONF <sub>i</sub> [7:0] SIZE	0x-
For each face read the angle	FD_ANGLE <sub>i</sub> [8:0] ANGLE	0x-

## 9.5 FDIF Register Manual

### 9.5.1 FDIF Instance Summary

Table 9-10 is the FDIF instance.

**Table 9-10. FDIF Instance Summary**

Module Name	Base Address	Size
FDIF	0x4A10 A000	4 KB

### 9.5.2 FDIF Registers

Table 9-11 summarizes the FDIF register mapping. Table 9-12 through Table 9-54 describe the register bits.

#### 9.5.2.1 FDIF Register Summary

**Table 9-11. FDIF Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	FDIF L4 Base Address
FDIF_REVISION	R	32	0x0000 0000	0x4A10 A000
FDIF_HWINFO	R	32	0x0000 0004	0x4A10 A004
FDIF_SYSCONFIG	RW	32	0x0000 0010	0x4A10 A010
RESERVED	R	32	0x0000 0020	0x4A10 A020
FDIF_IRQSTATUS_RA W <sub>j</sub> <sup>(1)</sup>	RW	32	0x0000 0024 + (0x10 * j)	0x4A10 A024 + (0x10 * j)
FDIF_IRQSTATUS_j <sup>(1)</sup>	RW	32	0x0000 0028 + (0x10 * j)	0x4A10 A028 + (0x10 * j)
FDIF_IRQENABLE_SET _j <sup>(1)</sup>	RW	32	0x0000 002C + (0x10 * j)	0x4A10 A02C + (0x10 * j)
FDIF_IRQENABLE_CLR _j <sup>(1)</sup>	RW	32	0x0000 0030 + (0x10 * j)	0x4A10 A030 + (0x10 * j)
FDIF_PICADDR	RW	32	0x0000 0060	0x4A10 A060
FDIF_CTRL	RW	32	0x0000 0064	0x4A10 A064
FDIF_WKADDR	RW	32	0x0000 0068	0x4A10 A068
FD_CTRL	RW	32	0x0000 0080	0x4A10 A080
FD_DNUM	R	32	0x0000 0084	0x4A10 A084
FD_DCOND	RW	32	0x0000 0088	0x4A10 A088
FD_STARTX	RW	32	0x0000 008C	0x4A10 A08C
FD_STARTY	RW	32	0x0000 0090	0x4A10 A090
FD_SIZEX	RW	32	0x0000 0094	0x4A10 A094
FD_SIZEY	RW	32	0x0000 0098	0x4A10 A098
FD_LHIT	RW	32	0x0000 009C	0x4A10 A09C
FD_CENTERX_j <sup>(2)</sup>	R	32	0x0000 0160 + (0x10 * i)	0x4A10 A160 + (0x10 * i)
FD_CENTERY_j <sup>(2)</sup>	R	32	0x0000 0164 + (0x10 * i)	0x4A10 A164 + (0x10 * i)
FD_CONFSIZE_j <sup>(2)</sup>	R	32	0x0000 0168 + (0x10 * i)	0x4A10 A168 + (0x10 * i)
FD_ANGLE_j <sup>(2)</sup>	R	32	0x0000 016C + (0x10 * i)	0x4A10 A16C + (0x10 * i)

<sup>(1)</sup> j = 0 to 2

<sup>(2)</sup> i = 0 to 34

### 9.5.2.2 FDIF Register Description

**Table 9-12. FDIF\_REVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision number	R	0x---- ---- TI Internal Data

**Table 9-13. Register Call Summary for Register FDIF\_REVISION**

FDIF Register Manual

- [FDIF Register Summary: \[0\]](#)

**Table 9-14. FDIF\_HWINFO**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Information about the IP module's hardware configuration, that is, typically the module's HDL generics (if any). Actual field format and encoding is up to the module's designer to decide.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FDIF_TAGS															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3:0	FDIF_TAGS	Hardware design value. This bit field reflects the value of the FDIF_TAG generic parameter. 0x0: 1 tag supported 0x1: 2 tags supported [...] 0xF: 16 tags supported	R	0xF

**Table 9-15. Register Call Summary for Register FDIF\_HWINFO**

FDIF Register Manual

- [FDIF Register Summary: \[0\]](#)

**Table 9-16. FDIF\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							STANDBYMODE	IDLEMODE	RESERVED	SOFTRESET					

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0000000
5:4	STANDBYMODE	<p>Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state.</p> <p>0x0: Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only.</p> <p>0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only.</p> <p>0x2: Smart-standby mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator. IP module shall not generate (initiator-related) wake-up events.</p> <p>0x3: Reserved</p>	RW	0x2
3:2	IDLEMODE	<p>Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.</p> <p>0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. Backup mode, for debug only.</p> <p>0x1: No-idle mode: local target never enters idle state. Backup mode, for debug only.</p> <p>0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wake-up events.</p> <p>0x3: Reserved</p>	RW	0x2
1	RESERVED	Reserved	R	0
0	SOFTRESET	<p>Software reset.</p> <p>Read 0x0: Reset done, no pending action</p> <p>Write 0x0: No action</p> <p>Write 0x1: Initiate software reset</p> <p>Read 0x1: Reset (software or other) ongoing</p>	RW	0

**Table 9-17. Register Call Summary for Register FDIF\_SYSCONFIG**

FDIF Functional Description

- [Software Reset: \[0\] \[1\]](#)
- [System Power Management: \[2\] \[3\]](#)
- [PRCM Hardware Handshake: \[4\]](#)

**Table 9-17. Register Call Summary for Register FDIF\_SYSCONFIG (continued)**

## FDIF Programming Guide

- [Global Initialization: \[5\] \[6\]](#)
- [FD Operational Modes Configuration: \[7\]](#)

## FDIF Register Manual

- [FDIF Register Summary: \[8\]](#)
- [FDIF Register Description: \[9\]](#)

**Table 9-18. FDIF\_IRQSTATUS\_RAW\_j**

<b>Address Offset</b>	0x0000 0024 + (0x10 * j)		
<b>Physical Address</b>	See <a href="#">Table 9-11</a>	<b>Instance</b>	FDIF
<b>Description</b>	Per-event raw interrupt status vector, line #n. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FINISH_IRQ	RESERVED								ERR_IRQ						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	FINISH_IRQ	Face detection processing done. Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0
7:1	RESERVED	Reserved	R	0x00
0	ERR_IRQ	Error received by the L3 port. Write 0x0: No action Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set event (debug)	RW W1toSet	0

**Table 9-19. Register Call Summary for Register FDIF\_IRQSTATUS\_RAW\_j**

## FDIF Functional Description

- [PRCM Hardware Handshake: \[0\]](#)
- [Interrupts and Events: \[1\] \[2\] \[3\] \[4\]](#)

## FDIF Register Manual

- [FDIF Register Summary: \[5\]](#)



**Table 9-20. FDIF\_IRQSTATUS\_j**

<b>Address Offset</b>	0x0000 0028 + (0x10 * j)		
<b>Physical Address</b>	See <a href="#">Table 9-11</a>	<b>Instance</b>	FDIF
<b>Description</b>	Per-event "enabled" interrupt status vector, line #n. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FINISH_IRQ	RESERVED								ERR_IRQ						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	FINISH_IRQ	Face detection processing done. Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0
7:1	RESERVED	Reserved	R	0x00
0	ERR_IRQ	Error received by the L3 port. Write 0x0: No action Read 0x0: No (enabled) event pending Read 0x1: Event pending Write 0x1: Clear (raw) event	RW W1toClr	0

**Table 9-21. Register Call Summary for Register FDIF\_IRQSTATUS\_j**

FDIF Functional Description

- [Interrupts and Events: \[0\] \[1\] \[2\] \[3\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[4\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[5\]](#)

**Table 9-22. FDIF\_IRQENABLE\_SET\_j**

<b>Address Offset</b>	0x0000 002C + (0x10 * j)		
<b>Physical Address</b>	See <a href="#">Table 9-11</a>	<b>Instance</b>	FDIF
<b>Description</b>	Per-event interrupt enable bit vector, line #n. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FINISH_IRQ	RESERVED								ERR_IRQ						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	FINISH_IRQ	Face detection processing done. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0
7:1	RESERVED	Reserved	R	0x00
0	ERR_IRQ	Error received by the L3 port. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW W1toSet	0

**Table 9-23. Register Call Summary for Register FDIF\_IRQENABLE\_SET\_j**

FDIF Functional Description

- [Interrupts and Events: \[0\]](#)

FDIF Programming Guide

- [Global Initialization: \[1\]](#)
- [FD Operational Modes Configuration: \[2\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[3\]](#)

**Table 9-24. FDIF\_IRQENABLE\_CLR\_j**

<b>Address Offset</b>	0x0000 0030 + (0x10 * j)	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Per-event interrupt enable bit vector, line #n. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FINISH_IRQ	RESERVED										ERR_IRQ				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	FINISH_IRQ	Face detection processing done. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0
7:1	RESERVED	Reserved	R	0x00
0	ERR_IRQ	Error received by the L3 port. Write 0x0: No action Read 0x0: Interrupt disabled (masked) Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW W1toClr	0

**Table 9-25. Register Call Summary for Register FDIF\_IRQENABLE\_CLR\_j**

FDIF Functional Description

- [Interrupts and Events: \[0\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[1\] \[2\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[3\]](#)

**Table 9-26. FDIF\_PICADDR**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Picture data store address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	ADDR	Picture data store address. The 5 least significant bits are forced to 0.	RW	0x0000000
4:0	RESERVED	Read returns 0.	R	0x00

**Table 9-27. Register Call Summary for Register FDIF\_PICADDR**

FDIF Functional Description

- [Typical Use: \[0\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[1\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[2\]](#)

**Table 9-28. FDIF\_CTRL**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MSTANDBY_HDSHK	MSTANDBY	MAX_TAGS				WRNP									

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6	MSTANDBY_HDSHK	<p>MStandby / Wait power management handshaking status bit</p> <p>The power management framework of the FDIF module is based on the handshaking of the MSTANDBY and WAIT signals. When going from a idle to normal power management transition, the software polls for <a href="#">FDIF_CTRL.MSTANDBY_HDSHK = 1</a> before starting the face detection processing.</p> <p>Read 0x0: Handshaking not completed. Do not initiate traffic on L3.</p> <p>Read 0x1: Handshaking completed. Can safely use the FDIF module.</p>	R	0
5	MSTANDBY	<p>MStandby signal generation. This bit shall be set to initiate a power management transition from NORMAL to IDLE or IDLE to NORMAL.</p> <p>0x0: Write: Clear MStandby signal. One polls <a href="#">FDIF_CTRL.MSTANDBY_HDSHK = 1</a> after writing this bit to ensure that the power management handshaking is completed.</p> <p>0x1: Write: Asserts MStandby signal</p>	RW	1
4:1	MAX_TAGS	<p>Max interconnect tags. This bit field sets the maximum number of interconnect tags that the module shall use. This number is programmable between 1 (<math>MAX\_TAGS = 0</math>) and <math>FDIF\_TAGS</math> (<math>MAX\_TAGS = FDIF\_TAGS - 1</math>). The value of <math>MAX\_TAGS</math> is reflected in the <math>FIDIF\_HWINFO</math> register setting. This register setting is expected to have impact on performance. It shall be set once at initialization. Higher value will give more bandwidth to the initiator. Lower value will give less bandwidth to the initiator. This value shall be set as low as possible such that other system initiators are not penalized.</p>	RW	0xF
0	WRNP	<p>L3 port non-posted write control. Dynamic usage of this feature is not supported. This bit shall be set at initialization and not modified hereafter until the processing is completed. When non-posted writes are used, tags shall be used for best performance (<math>MAX\_TAGS &gt; 1</math>).</p> <p>0x0: All writes are non posted</p> <p>0x1: All writes are posted</p>	RW	0

**Table 9-29. Register Call Summary for Register FDIF\_CTRL**

FDIF Functional Description

- [PRCM Hardware Handshake: \[0\] \[1\] \[2\]](#)
- [L3 Interconnect Parameters: \[3\] \[4\]](#)

FDIF Programming Guide

- [Global Initialization: \[5\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[6\]](#)
- [FDIF Register Description: \[7\] \[8\]](#)

**Table 9-30. FDIF\_WKADDR**

<b>Address Offset</b>	0x0000 0068																																																													
<b>Physical Address</b>	See <a href="#">Table 9-11</a>	<b>Instance</b> FDIF																																																												
<b>Description</b>	RW																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">ADDR</td> <td colspan="12">RESERVED</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ADDR																RESERVED											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
ADDR																RESERVED																																														
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																										
31:5	ADDR	Work area address The 5 least significant bits are forced to 0.	RW	0x0000000																																																										
4:0	RESERVED	Read returns 0.	R	0x00																																																										

**Table 9-31. Register Call Summary for Register FDIF\_WKADDR**

FDIF Functional Description

- [Typical Use: \[0\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[1\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[2\]](#)

**Table 9-32. FD\_CTRL**

<b>Address Offset</b>	0x0000 0080																																																				
<b>Physical Address</b>	See <a href="#">Table 9-11</a>	<b>Instance</b> FDIF																																																			
<b>Description</b>	Control register Don't set more than 2 bits to "1" at the same time. Otherwise, operations cannot be guaranteed.																																																				
<b>Type</b>	RW																																																				
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td>FINISH</td><td>RUN</td><td>SRST</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																FINISH	RUN	SRST
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																						
RESERVED																FINISH	RUN	SRST																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																	
31:3	RESERVED	Reserved	R	0x0000 0000																																																	
2	FINISH	Process Completion Flag Clear 0x0: Write: Disable Read: Process incomplete 0x1: Write: Process complete flag clear Read: Process complete	RW	0																																																	
1	RUN	Process Start Request 0x0: Write: Disable Read: No processing 0x1: Write: Process start request Read: Processing data	RW	0																																																	
0	SRST	Software Reset This bit shall not be used to reset the FDIF module. Instead, the <a href="#">FDIF_SYSCONFIG[0]</a> SOFTRESET bit shall be used for complete soft reset. 0x0: Write: Disable Read: Reset cancel 0x1: Write: Reset Read: Under reset	RW	0																																																	

**Table 9-33. Register Call Summary for Register FD\_CTRL**

## FDIF Functional Description

- [Software Reset: \[0\]](#)
- [PRCM Hardware Handshake: \[1\]](#)
- [Typical Use: \[2\] \[3\]](#)

## FDIF Programming Guide

- [FD Operational Modes Configuration: \[4\] \[5\] \[6\]](#)

## FDIF Register Manual

- [FDIF Register Summary: \[7\]](#)

**Table 9-34. FD\_DNUM**

<b>Address Offset</b>	0x0000 0084		<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>			
<b>Description</b>	Face Detection Result Count Register			
<b>Type</b>	R			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DNUM															

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x00000000
5:0	DNUM	Face detection result count. Up to 35 faces can be detected. Number of face(s) detected: 0x0: 0 face detected 0x1: 1 face detected 0x2: 2 faces detected [...] 0x23: 35 faces detected 0x24 to 0x3F: unused	R	0x00

**Table 9-35. Register Call Summary for Register FD\_DNUM**

## FDIF Functional Description

- [Typical Use: \[0\]](#)

## FDIF Register Manual

- [FDIF Register Summary: \[1\]](#)

**Table 9-36. FD\_DCOND**

<b>Address Offset</b>	0x0000 0088		<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>			
<b>Description</b>	Detection Condition Setting Register			
<b>Type</b>	RW			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DIR	MIN														

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3:2	DIR	Detection direction setting 0x0: Executes only for UP 0x1: Executes only for RIGHT 0x2: Executes only for LEFT 0x3: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
1:0	MIN	Reserved 0x0: Set the min face size to 20 pixels 0x1: Set the min face size to 25 pixels 0x2: Set the min face size to 32 pixels 0x3: Set the min face size to 40 pixels	RW	0x0

**Table 9-37. Register Call Summary for Register FD\_DCOND**

FDIF Functional Description

- [Typical Use: \[0\]](#)
- [Performance Parameters: \[1\] \[2\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[3\] \[4\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[5\]](#)

**Table 9-38. FD\_STARTX**

<b>Address Offset</b>	0x0000 008C	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Detection Area Setting Register: X Start Coordinate.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STARTX															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STARTX	Starting X coordinates Permitted values are 0<=STARTX<=160	RW	0x00

**Table 9-39. Register Call Summary for Register FD\_STARTX**

FDIF Functional Description

- [Typical Use: \[0\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[1\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[2\]](#)

**Table 9-40. FD\_STARTY**

<b>Address Offset</b>	0x0000 0090	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Detection Area Setting Register: Y Start Coordinate.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STARTY															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6:0	STARTY	Starting Y coordinates Permitted values are 0<=STARTY<=120	RW	0x00

**Table 9-41. Register Call Summary for Register FD\_STARTY**

FDIF Functional Description

- [Typical Use: \[0\] \[1\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[2\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[3\]](#)

**Table 9-42. FD\_SIZEX**

<b>Address Offset</b>	0x0000 0094	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Detection Area Setting Register: X Direction Size		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SIZEX															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0000000
8:0	SIZEX	X Direction Size	RW	0x140

**Table 9-43. Register Call Summary for Register FD\_SIZEX**

FDIF Functional Description

- [Typical Use: \[0\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[1\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[2\]](#)

**Table 9-44. FD\_SIZEY**

<b>Address Offset</b>	0x0000 0098	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Detection Area Setting Register: Y Direction Size		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SIZEY															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000000
7:0	SIZEY	Y Direction Size	RW	0xF0



**Table 9-45. Register Call Summary for Register FD\_SIZEY**

- FDIF Programming Guide
- [FD Operational Modes Configuration: \[0\]](#)
- FDIF Register Manual
- [FDIF Register Summary: \[1\]](#)

**Table 9-46. FD\_LHIT**

<b>Address Offset</b>	0x0000 009C	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Threshold Setting Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LHIT															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:0	LHIT	Threshold Permitted values are 0x0 to 0x9	RW	0x5

**Table 9-47. Register Call Summary for Register FD\_LHIT**

- FDIF Functional Description
- [Performance Parameters: \[0\]](#)
- FDIF Programming Guide
- [FD Operational Modes Configuration: \[1\]](#)
- FDIF Register Manual
- [FDIF Register Summary: \[2\]](#)

**Table 9-48. FD\_CENTERX\_i**

<b>Address Offset</b>	0x0000 0160 + (0x10 * i)	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Detection Result: X Coordinate. Its value is undefined after reset.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CENTERX															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved. Read returns reset value.	R	0x-
8:0	CENTERX	Face position: center X coordinate The coordinates given by ( <a href="#">FD_CENTERX_i</a> , <a href="#">FD_CENTERY_i</a> ) give the central coordinates of the face position. Permitted values are 0x0 to 0x13F	R	0x-

**Table 9-49. Register Call Summary for Register FD\_CENTERX\_i**

- FDIF Functional Description
- [Typical Use: \[0\]](#)
- FDIF Programming Guide
- [FD Operational Modes Configuration: \[1\]](#)

**Table 9-49. Register Call Summary for Register FD\_CENTERX\_i (continued)**

FDIF Register Manual

- [FDIF Register Summary: \[2\]](#)
- [FDIF Register Description: \[3\] \[4\]](#)

**Table 9-50. FD\_CENTERY\_i**

<b>Address Offset</b>	0x0000 0164 + (0x10 * i)	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Detection Result: Y Coordinate. Its value is undefined after reset.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CENTERY															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved. Read returns reset value.	R	0x-
7:0	CENTERY	Face position: center Y coordinate The coordinates given by ( <a href="#">FD_CENTERX_i</a> , <a href="#">FD_CENTERY_i</a> ) give the central coordinates of the face position. Permitted values are 0x0 to 0xEF	R	0x-

**Table 9-51. Register Call Summary for Register FD\_CENTERY\_i**

FDIF Functional Description

- [Typical Use: \[0\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[1\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[2\]](#)
- [FDIF Register Description: \[3\] \[4\]](#)

**Table 9-52. FD\_CONFSIZE\_i**

<b>Address Offset</b>	0x0000 0168 + (0x10 * i)	<b>Instance</b>	FDIF
<b>Physical Address</b>	See <a href="#">Table 9-11</a>		
<b>Description</b>	Detection Result: Confidence Level and Size. Its value is undefined after reset.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CONF						SIZE									

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved. Read returns reset value.	R	0x-
11:8	CONF	Confidence level Permitted values are 0x0 (high) to 0x9 (low)	R	0x-
7:0	SIZE	Detection result face size Permitted values 0x14 to 0xF0	R	0x-

**Table 9-53. Register Call Summary for Register FD\_CONFSIZE\_i**

FDIF Functional Description

- [Typical Use: \[0\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[1\] \[2\]](#)

**Table 9-53. Register Call Summary for Register FD\_CONFSIZE\_i (continued)**

FDIF Register Manual

- [FDIF Register Summary: \[3\]](#)

**Table 9-54. FD\_ANGLE\_i**

<b>Address Offset</b>	0x0000 016C + (0x10 * i)	
<b>Physical Address</b>	See <a href="#">Table 9-11</a>	<b>Instance</b> FDIF
<b>Description</b>	Detection Result: Angle. Its value is undefined after reset.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ANGLE															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved. Read returns reset value.	R	0x-
8:0	ANGLE	Detection result face angle Permitted values: When DIR = 0: 0x0 (0 deg) to 0x1E (30 deg) and 0x14A (330 deg) to 0x167 (359 deg) When DIR = 1: 0x03C (60 deg) to 0x078 (120 deg) When DIR = 2: 0x0F0 (240 deg) to 0x12C (300 deg)	R	0x-

**Table 9-55. Register Call Summary for Register FD\_ANGLE\_i**

FDIF Functional Description

- [Typical Use: \[0\]](#)

FDIF Programming Guide

- [FD Operational Modes Configuration: \[1\]](#)

FDIF Register Manual

- [FDIF Register Summary: \[2\]](#)

PRELIMINARY

## Display Subsystem

This chapter describes the display subsystem (DSS) for the device.

Topic	Page
<b>10.1 Display Subsystem Overview .....</b>	<b>2070</b>
<b>10.2 Display Controller .....</b>	<b>2091</b>
<b>10.3 MIPI Display Serial Interface .....</b>	<b>2329</b>
<b>10.4 High-Definition Multimedia Interface .....</b>	<b>2509</b>
<b>10.5 Remote Frame Buffer Interface .....</b>	<b>2512</b>
<b>10.6 Video Encoder .....</b>	<b>2558</b>

## 10.1 Display Subsystem Overview

The display subsystem (DSS) provides the logic to display a video frame from the memory frame buffer on a liquid-crystal display (LCD) panel or a TV set.

The DSS can display different pictures simultaneously by using the primary LCD1 and secondary LCD2 outputs, in addition to the TV output.

The primary LCD output can be one of the following:

- Display serial interface (DSI) ( MIPI® DSI)
- Remote frame buffer interface (RFBI) (MIPI DBI 2.0)

The secondary LCD output can be one of the following:

- Parallel CMOS output (MIPI DPI 1.0)
- DSI (MIPI DSI)
- RFBI (MIPI DBI 2.0)

The TV output can be one of the following:

- Composite signal
- High-definition multimedia interface (HDMI)

The modules integrated in the DSS are:

- Display controller (DISPC):
  - One direct memory access (DMA) engine
  - Two LCD outputs and one TV output, each one with a dedicated overlay manager
  - One graphics pipeline (GFX), three video pipelines, and one write-back pipeline
- RFBI:
  - 8-, 9-, 16-bit parallel interface
  - Programmable pixel memory formats
  - Programmable output formats on one or multiple cycles per pixel
- Two DSI protocol engines:
  - 4-data lane complex input/output (I/O) for DSI1 and 2-data lane complex I/O for DSI2
  - Bidirectional data link support (only one data lane is used in reverse direction in command mode)
  - Supports video mode and command mode
  - Supports data interleaving
- NTSC/PAL video encoder (VENC):
  - Composite output signal through external video digital-to-analog converter (DAC)
- HDMI protocol engine:
  - Supports HDMI 1.3
  - 30-bit RGB color
  - HDCP key protection
  - Deep color mode support (10-bit for 1080p and up to 12-bit for 1080i/720p)

The necessary phase-locked loops (PLLs) with their control modules and the physical layers (PHYs) are outside the DSS. The PHYs can be:

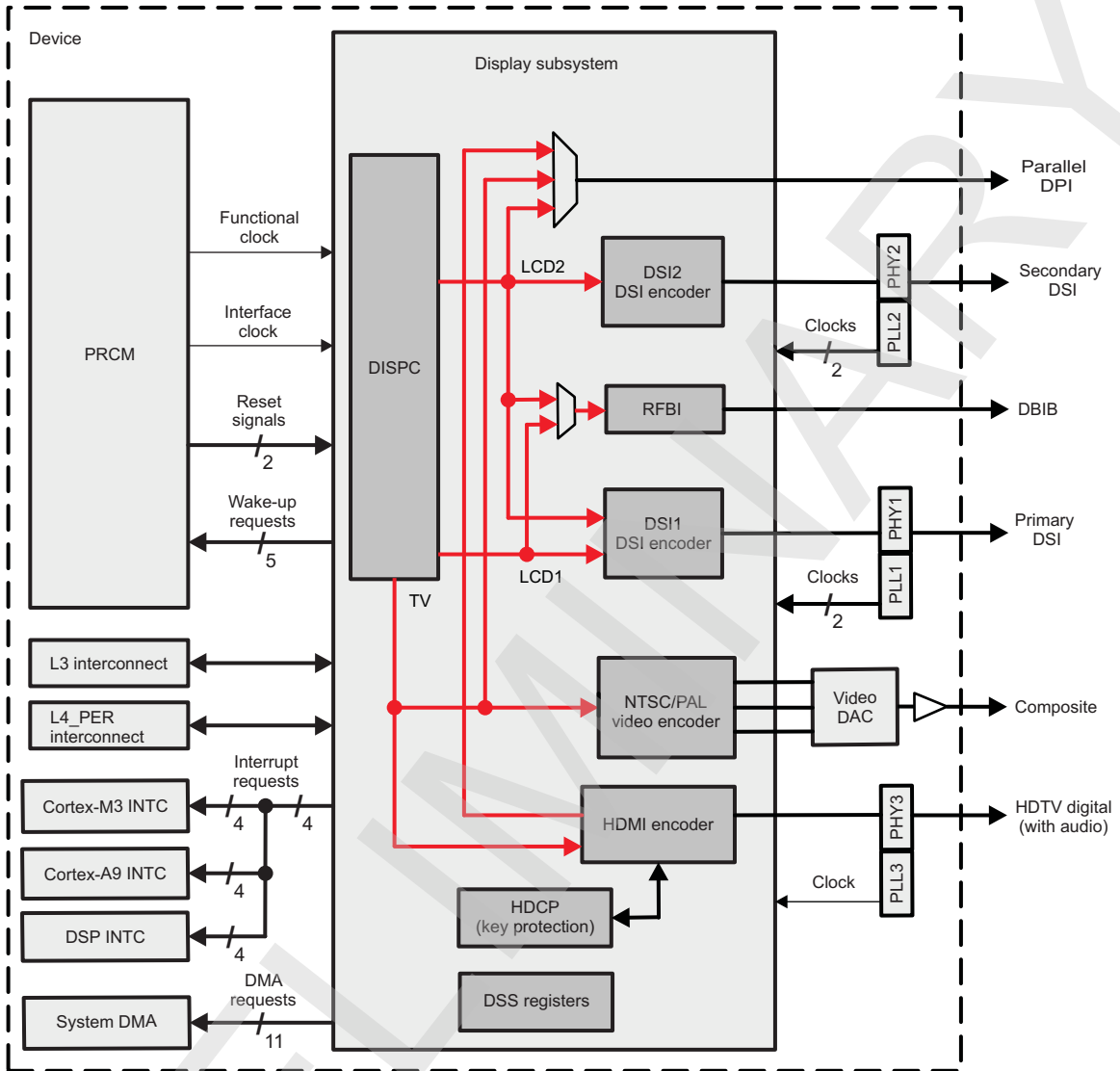
- HDMI-PHY
- VDAC
- DSI-PHY

To ensure efficient bandwidth, the DSS integrates:

- Local 32-bit interconnect with a level 4 (L4\_PER) connection to configure the DSS modules
- Connection between level 3 (L3) interconnect and the DISPC to exchange data with synchronous dynamic random access memory (SDRAM) and memory using the DISPC DMA engine

Figure 10-1 is a high-level diagram of the DSS.

Figure 10-1. DSS Overview



dss-001

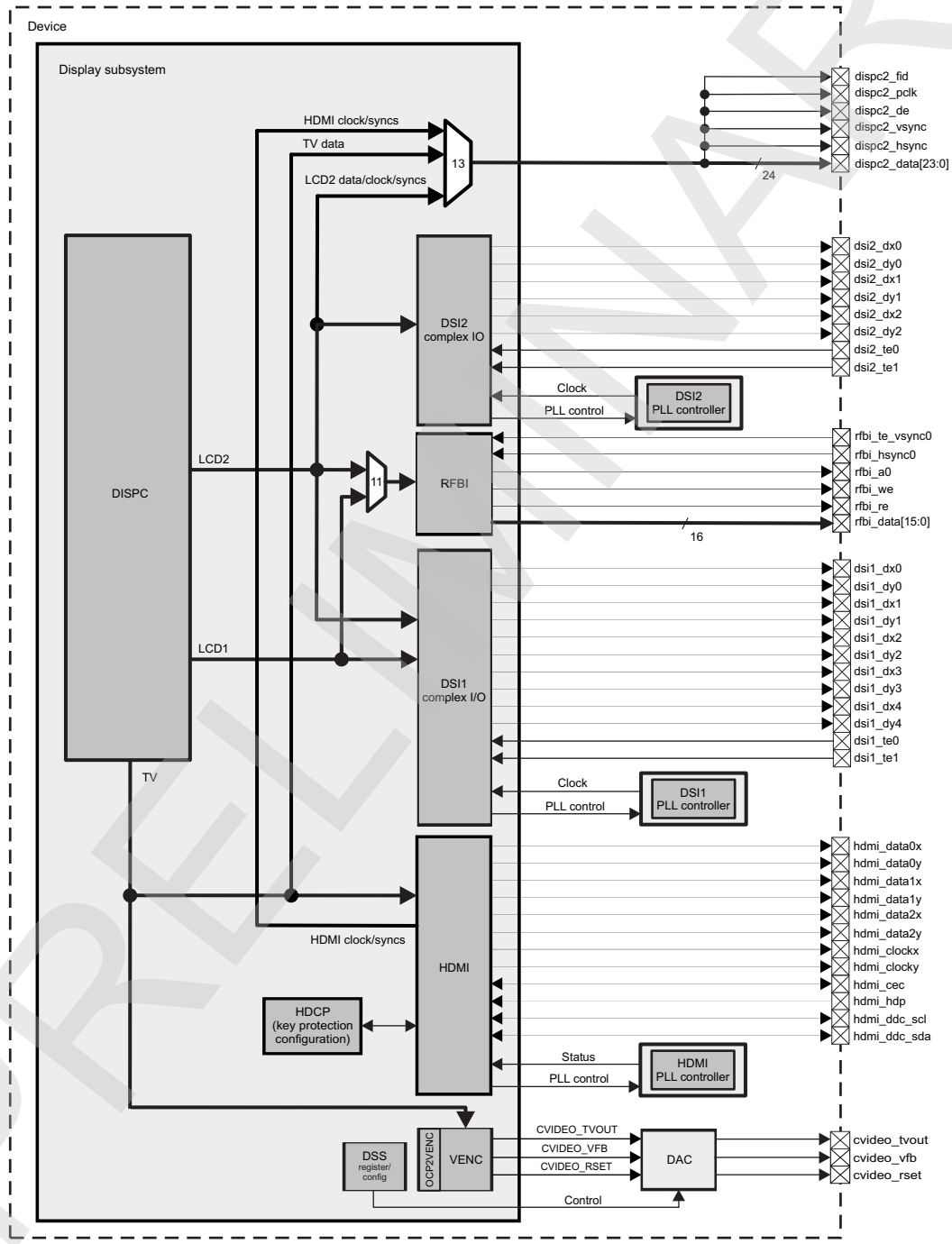
### 10.1.1 DSS Environment

This section describes the various outputs handled by the DSS:

- LCD support
- TV display support

Figure 10-2 is a diagram of the DSS environment.

Figure 10-2. DSS Environment



dss-002



### 10.1.1.1 DSS LCD Support

LCD panels must be connected to the DSS of the device using parallel and/or serial interfaces. Two modes are supported in parallel:

- DBIB
- Display parallel interface (DPI)

#### 10.1.1.1.1 Parallel Interface

In parallel interface, the modules of the DSS path are the DISPC and the RFBI.

The DISPC is connected to the memory through the L3 interconnect and uses its own DMA (with embedded FIFO) to read data from the system memory.

The remote frame buffer (RFB) of the LCD panel is connected directly to the RFBI module of the device. The RFBI controls the reads/writes from/to the RFB. The RFBI receives the output data from the DISPC and generates the signals to control the LCD panel. Through the RFBI, the microprocessor unit (MPU) can send commands or parameter/display data to the LCD panel and directly set the DISPC registers to read/write the data from/to the memory in the LCD panel.

The RFBI can be connected to either of the DISPC LCD outputs, but only one at a time can be used. The selection (multiplexer 11 in [Figure 10-2](#)) is done through the `DSS_CTRL[14]` RFBI\_SWITCH bit. It is not allowed to change the configuration, when the DISPC or RFBI modules are active.

For more details, see [Section 10.2, Display Controller](#), and [Section 10.5, Remote Frame Buffer Interface](#).

In synchronous parallel interface, the required data and control signals are provided directly to an external MIPI DPI-compatible parallel panel.

Two configuration options are available, selected through the `DSS_CTRL[17]` LCD2\_TV\_SEL bit field (multiplexer 13 in [Figure 10-2](#)):

- `LCD2_TV_SEL = 0` (default): The LCD2 channel of DISPC (pixel data, clock and synchronization signals) is output on the parallel interface.
- `LCD2_TV_SEL = 1`: The TV channel of DISPC (pixel data), together with the pixel clock and synchronization signals of HDMI, is output on the parallel interface.

, [Table 10-1](#) lists the exact DSS signals outgoing on the device boundary pads.

**Table 10-1. Display Parallel Interface Signals Mapping**

Signal names at device pads (See <a href="#">Figure 10-2</a> )	<code>DSS_CTRL[17]</code> LCD2_TV_SEL = 0 DISPC LCD2 channel out (pixel data, clock, syncs)	<code>DSS_CTRL[17]</code> LCD2_TV_SEL = 1 DISPC TV channel out (pixel data) HDMI (pixel clock, syncs)
dispc2_fid	DISPC_LCD2_FID	HDMI_M_FID
dispc2_pclk	DISPC_LCD2_PCLK	VIDEO_M_PCLK
dispc2_de	DISPC_LCD2_DE	HDMI_M_DE
dispc2_vsync	DISPC_LCD2_VSYNC	HDMI_M_VS
dispc2_hsync	DISPC_LCD2_HSYNC	HDMI_M_HS
dispc2_data[23:0]	DISPC_LCD2_DATA[23:0]	DISPC_TV_DATA[29:0]

For more details on LCD2 and TV out pixel data formats for the parallel interface, see [Section 10.2.2, Display Controller Environment](#).

#### 10.1.1.1.2 Serial Interfaces

In serial interface, the DISPC and the two DSI modules with their associated PLLs are used in the data path. The DISPC is connected to the memory through the L3 interconnect and uses its own DMA to read the data from the system memory.

The DSI module contains a PLL to multiply the pixel clock by an appropriate factor. The resulting serialized data is then transmitted on one to four differential data pairs and an additional clock pair that has a reference transition at the boundary between pixels.

For more details, see [Section 10.3, MIPI DSI](#).

## 10.1.1.2 DSS TV Display Support

### 10.1.1.2.1 Video Encoder

In a TV output configuration, the DISPC, the VENC and the associated PLL modules, and the Video DAC are used in the data path. The DISPC receives synchronization signals from the VENC and synchronously sends pixel data to the VENC with these signals. The digital output of the DISPC is always a 24-bit RGB value based on a pixel request from the VENC.

The VENC converts RGB video signals to conform to the NTSC/PAL standard for analog video.

For more details, see [Section 10.6](#), *VENC*.

### 10.1.1.2.2 HDMI

In an HDMI configuration, the DISPC, HDMI, and HDCP modules and the associated PLL modules are used in the data path. The HDMI module converts the RGB video into standard high-definition digital video format. The module has a PLL, which can multiply the pixel clock by an appropriate factor. The data is transmitted on three differential data pairs and an additional clock pair.

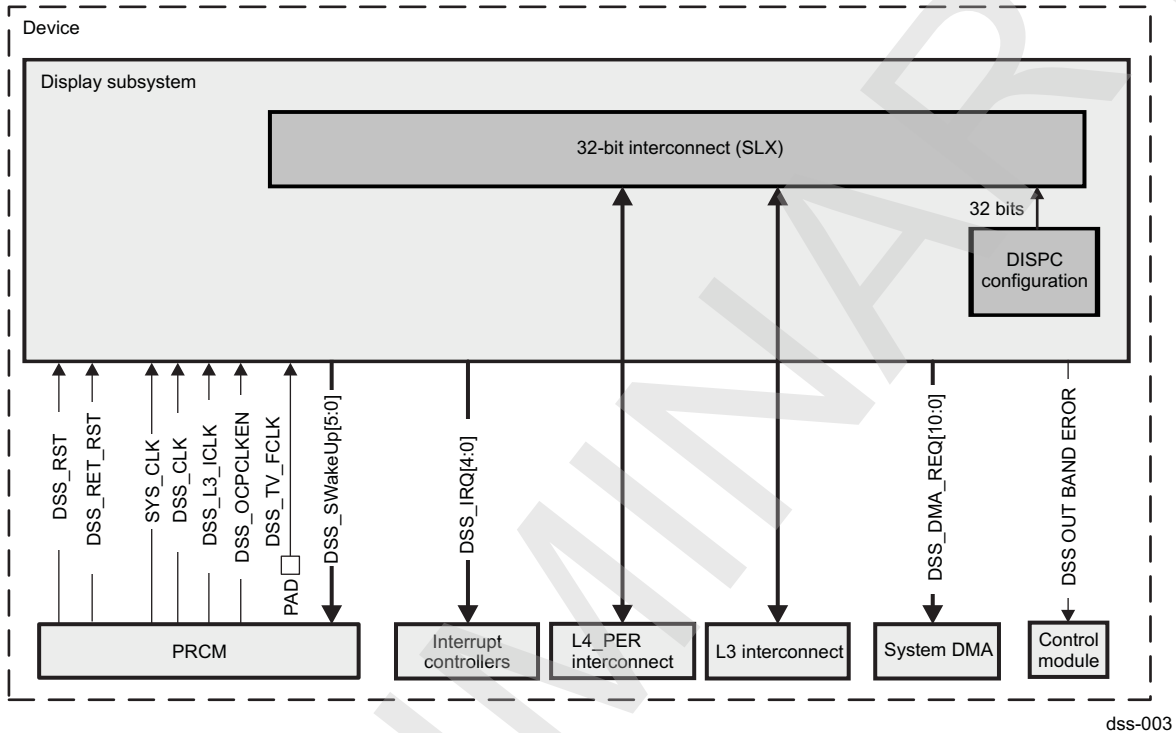
For more details, see [Section 10.4](#), *HDMI*.

### 10.1.2 DSS Integration

This section describes the integration of the DSS module in the device, including information about clocks, resets, and hardware requests.

Figure 10-3 shows the integration of the DSS in the device.

Figure 10-3. DSS Integration



dss-003

#### 10.1.2.1 DSS Clocks

The power, reset, and clock management (PRCM) module provides clock signals to the DSS.

Figure 10-4 shows the details of the DSS clock tree.

Figure 10-4. DSS Clock Tree

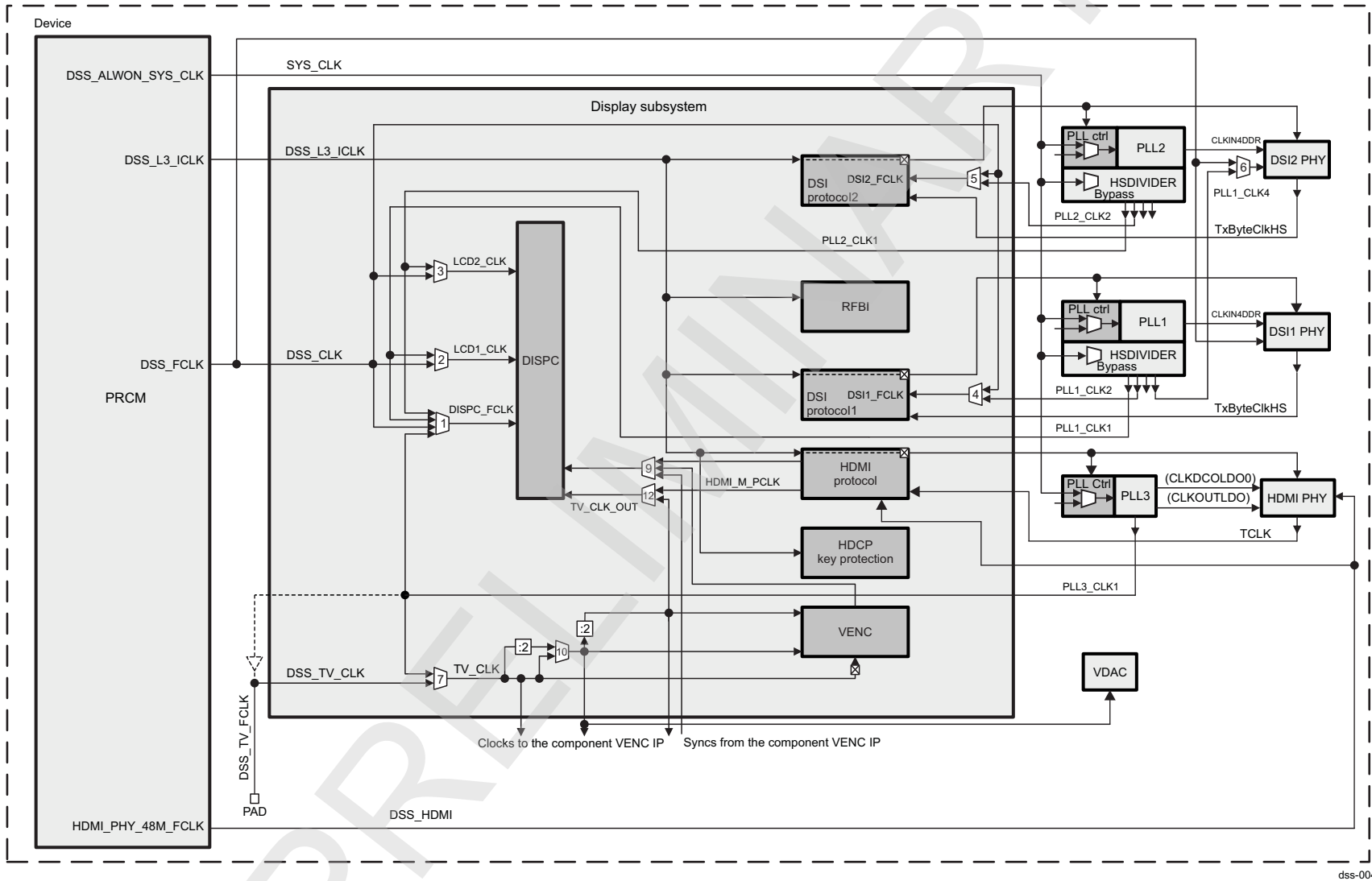


Table 10-2 lists the controls for the clock multiplexer.

**Table 10-2. DSS Clocks**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DSS	DSS_HDMI	HDMI_PHY_48M_FCLK	PRCM module	Functional clock
	DSS_CLK	DSS_FCLK	PRCM module	Functional clock
	SYS_CLK	DSS_ALWON_SYS_CLK	PRCM module	System clock
	DSS_TV_CLK	DSS_TV_FCLK	PAD	TV output interface clock
	DSS_L3_ICLK	DSS_L3_ICLK	PRCM module	Interface clock

- The clock source for the L3 interface clock is DSS\_L3\_ICLK.
- The clock sources for the DSS modules are listed in [Table 10-3](#).

**Table 10-3. DSS Modules Clock Sources**

Destination	Source Signal Name	Source	Multiplexer Number	DSS_CTRL Register Bit Field
DISPC functional clock (DISPC_FCLK)	DSS_CLK	PER_PLL	1	[9:8] FCK_CLK_SWITCH
	PLL1_CLK1	PLL1	1	
	PLL2_CLK1	PLL2	1	
	PLL3_CLK1	PLL3	1	
DSI1 functional clock (LCD1_CLK)	DSS_CLK	PER_PLL	2	[0] LCD1_CLK_SWITCH
	PLL1_CLK1	PLL1	2	
DSI2 functional clock (LCD2_CLK)	DSS_CLK	PER_PLL	3	[12] LCD2_CLK_SWITCH
	PLL2_CLK1	PLL2	3	
DSI1 protocol engine functional clock (DSI1_FCLK)	DSS_CLK	PER_PLL	4	[1] DSI1_CLK_SWITCH
	PLL1_CLK2	PLL1	4	
DSI2 protocol engine functional clock (DSI2_FCLK)	DSS_CLK	PER_PLL	5	[10] DSI2_CLK_SWITCH
	PLL2_CLK2	PLL2	5	
VENC/HDMI functional clock (TV_CLK)	DSS_TV_CLK	PAD	7	[11] TV_CLK_SWITCH
	PLL3_CLK1	PLL3	7	
DISPC external clock (TV_CLK_OUT)	TV_CLK	Multiplexer 7	12	[15] VENC_HDMI_SWITCH
	HDMI_M_PCLK	HDMI	12	

**NOTE:** Multiplexer 9 in [Figure 10-4](#) has the same control as multiplexer 12 when switched between VENC and HDMI.

Multiplexer 6 in [Figure 10-4](#) allows the DSS\_FCLK or PLL1\_CLK4 clock to be sent on the DSI2 PHY bypass input. The use of the bypassed clock allows both DSI modules to run on DSS\_FCLK or on a single PLL. The selection is done on device control module level. For more information, see [Section 10.3.4.2, DSI Clock Configuration](#) in [Section 10.3, MIPI Display Serila Interface](#).

- Clock generation  
The TV\_CLK clock is divided into three balanced clocks, depending on the clock mode selected (see [Table 10-4](#)).

**Table 10-4. Possible Digital Clock Division for the Video Encoder**

Clock Output	Clock Mode	
	Clock Mode 0	Clock Mode 1
VENC clock 4x	TV_CLK	TV_CLK or 0 (gated)

**Table 10-4. Possible Digital Clock Division for the Video Encoder (continued)**

Clock Output	Clock Mode	Clock Mode
VENC clock 2x	TV_CLK/2	TV_CLK
VENC clock 1x	TV_CLK/4	TV_CLK/2

The clock mode is defined by the [DSS\\_CTRL\[2\]](#) VENC\_CLOCK\_MODE bit:

- In case of clock mode 1, the [DSS\\_CTRL\[3\]](#) VENC\_CLOCK\_4X\_ENABLE bit field is used to control the clock gating.
- In case of clock mode 0, the VENC\_CLOCK\_4X\_ENABLE bit field must be set to 0x1 by software.

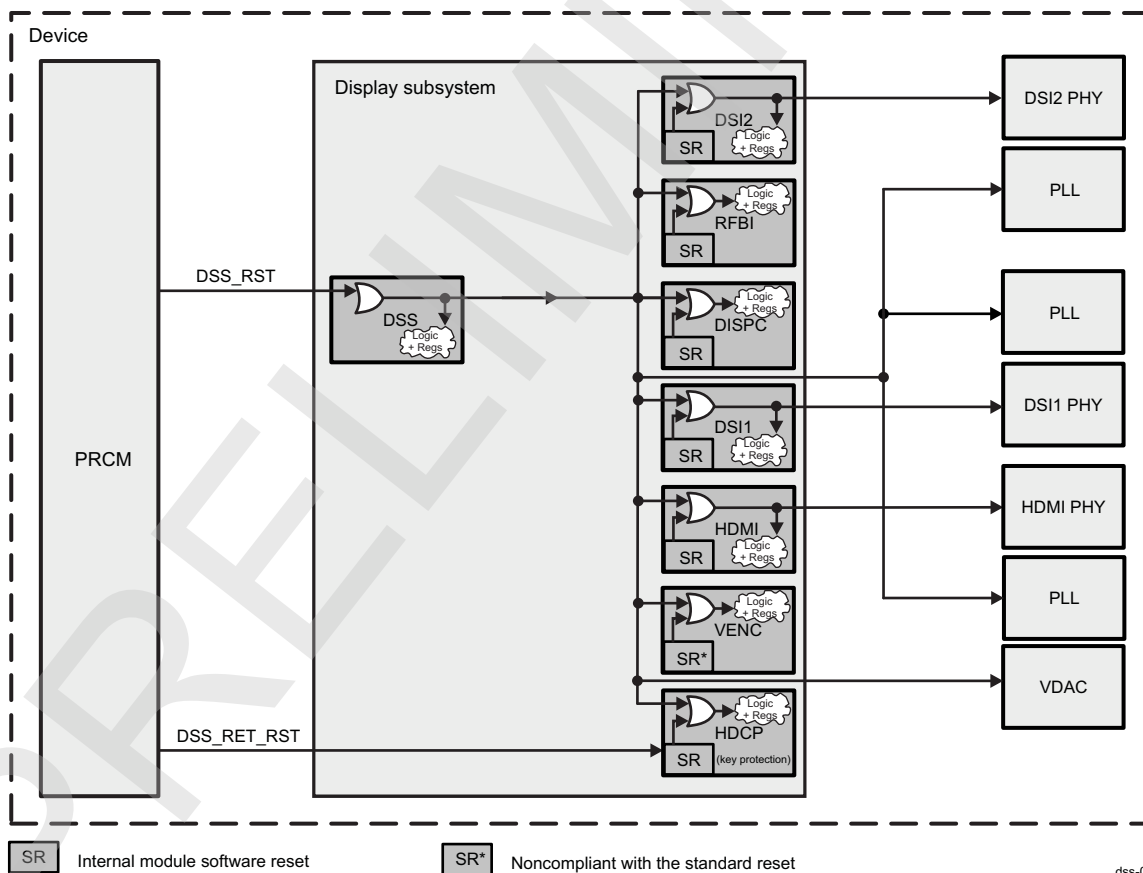
**NOTE:** After reset, clock mode 0 is selected by default, and the TV\_CLK clock is disabled.

The TV\_CLK/4 in mode 0, or the TV\_CLK/2 in mode 1, is used into the DISPC to send data to the VENC.

### 10.1.2.2 DSS Resets

The PRCM module provides two reset signals to the DSS.

[Figure 10-5](#) shows the details of the reset tree for the DSS.

**Figure 10-5. DSS Reset Scheme**

[Table 10-5](#) lists the resets for the DSS.

**Table 10-5. DSS Resets**

Module Instance	Destination Signal Name	Resets		
		Source Signal Name	Source	Description
DSS	DISPSS_RET_RST	DSS_RET_RST	PRCM module	Retention reset
	DISPSS_RST	DSS_RST	PRCM module	Nonretention reset

**10.1.2.2.1 Hardware Reset**

The DSS receives its DISPSS\_RST reset signal (the reset signal of the DSS power domain) from the PRCM module. The DISPSS\_RET\_RST is used only for the HDCP key protection module.

**10.1.2.3 DSS Power Management**

The DSS modules are in the DSS power domain, except for the dual video DACs, which are in the analog VDDADAC voltage domain.

Table 10-6 lists the DSS power domains.

**Table 10-6. DSS Power Domains**

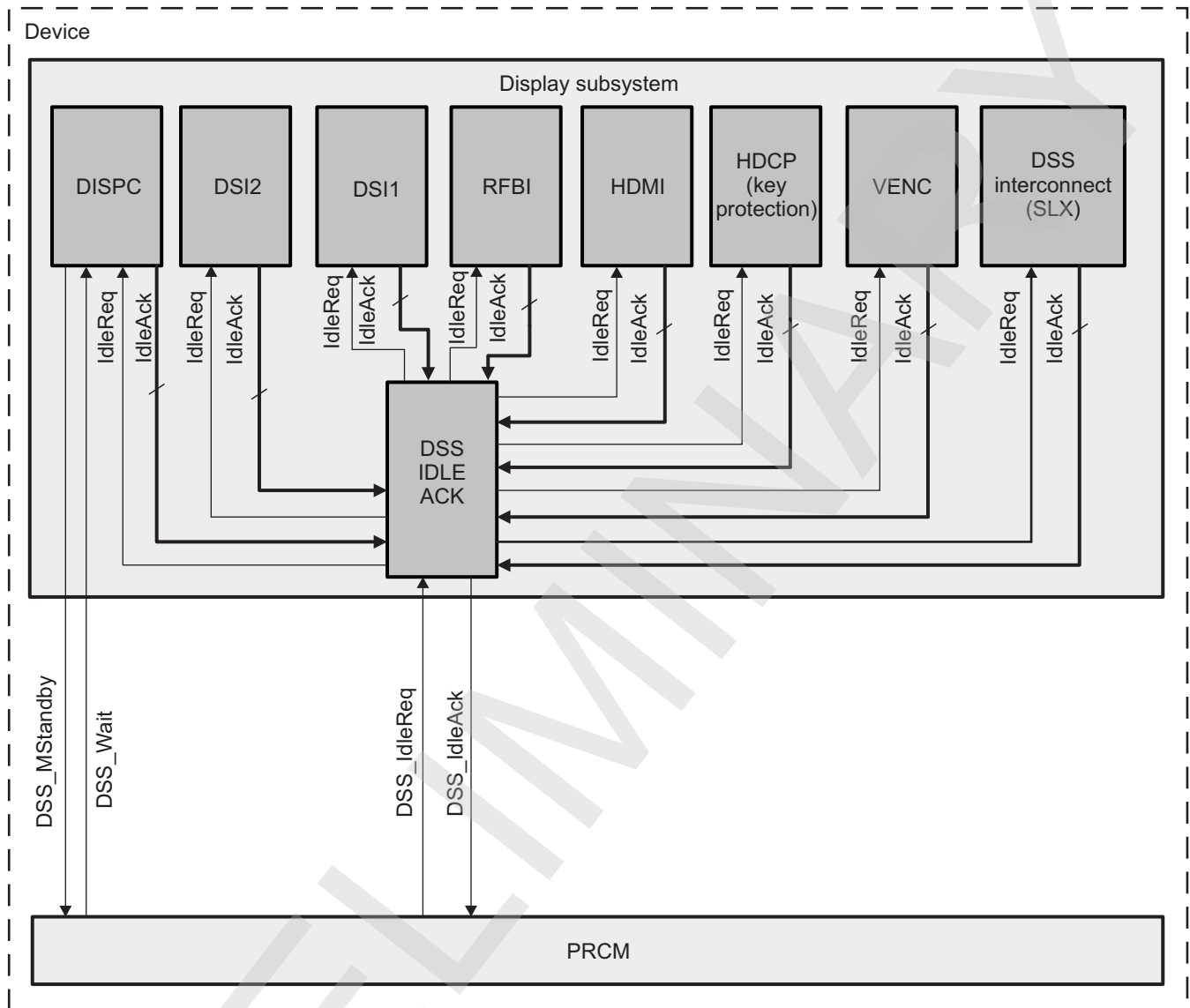
Module Instance	Attributes
	Power Domain
Display subsystem	PD_DSS
DISPC	For more details, see <a href="#">Section 10.2, Display Controller</a> .
MIPI DSI1	For more details, see <a href="#">Section 10.3, MIPI Display Serial Interface</a> .
MIPI DSI2	For more details, see <a href="#">Section 10.3, MIPI Display Serial Interface</a> .
RFBI	For more details, see <a href="#">Section 10.5, Remote Frame Buffer Interface</a> .
HDMI	For more details, see <a href="#">Section 10.4, High-Definition Multimedia Interface</a> .
Video encoder	For more details, see <a href="#">Section 10.6, Video Encoder</a> .

**10.1.2.3.1 Standby Mode**

As part of the system-wide power-management scheme, the DSS supports the MStandby/MWait and SIdleReq/SIdleAck protocols:

- MStandby/MWait
  - DISPC
- SIdleReq/SIdleAck
  - DISPC
  - DSI1
  - DSI2
  - RFBI
  - HDMI
  - HDCP (key protection)
  - VENC
  - DSS interconnect

Figure 10-6 shows the details of the DSS wake-up generation.

**Figure 10-6. DSS IdleAck/MStandby Generation**

dss-007

#### 10.1.2.3.1.1 Conditions to Exit Standby Mode

For information about the conditions that allow the subsystem to exit standby mode, see [Section 10.2, Display Controller](#).

#### 10.1.2.3.1.2 Standby Procedure Description

When the DSS initiates a standby procedure, it also initiates a standby/wait handshake protocol with the PRCM module that lets the PRCM module cut the DSS clocks.

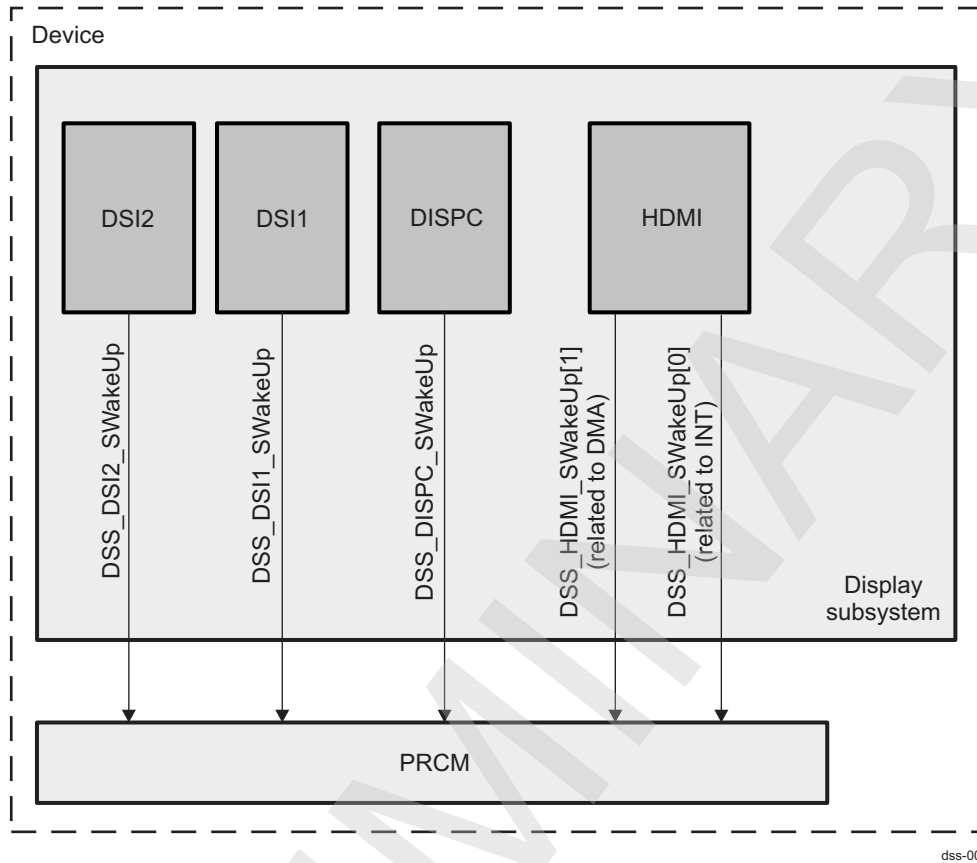
#### 10.1.2.3.2 Wake-Up Mode

The DISPC, DSI1, DSI2, and HDMI modules support the wake-up protocol. DSS\_HDMI\_SWakeUp is associated with DSS\_HMDI\_IRQ, which is generated by the HDMI module. For the events that generate an Swakeup and the description and configuration of the registers, see the DISPC, DSI, and HDMI TRM chapters.

[Figure 10-7](#) shows DSS wake-up generation.



Figure 10-7. DSS Wake-Up Generation



#### 10.1.2.4 DSS Interrupt Requests

The DSS has four interrupt lines. The interrupt signals are connected to the MPUs and DSP interrupt controller (INTC) modules. The interrupts are generated from the DISPC, DSI1 and DSI2 protocol engines, and HDMI.

Figure 10-8 shows DSS interrupt generation.

Figure 10-8. DSS Interrupt Generation

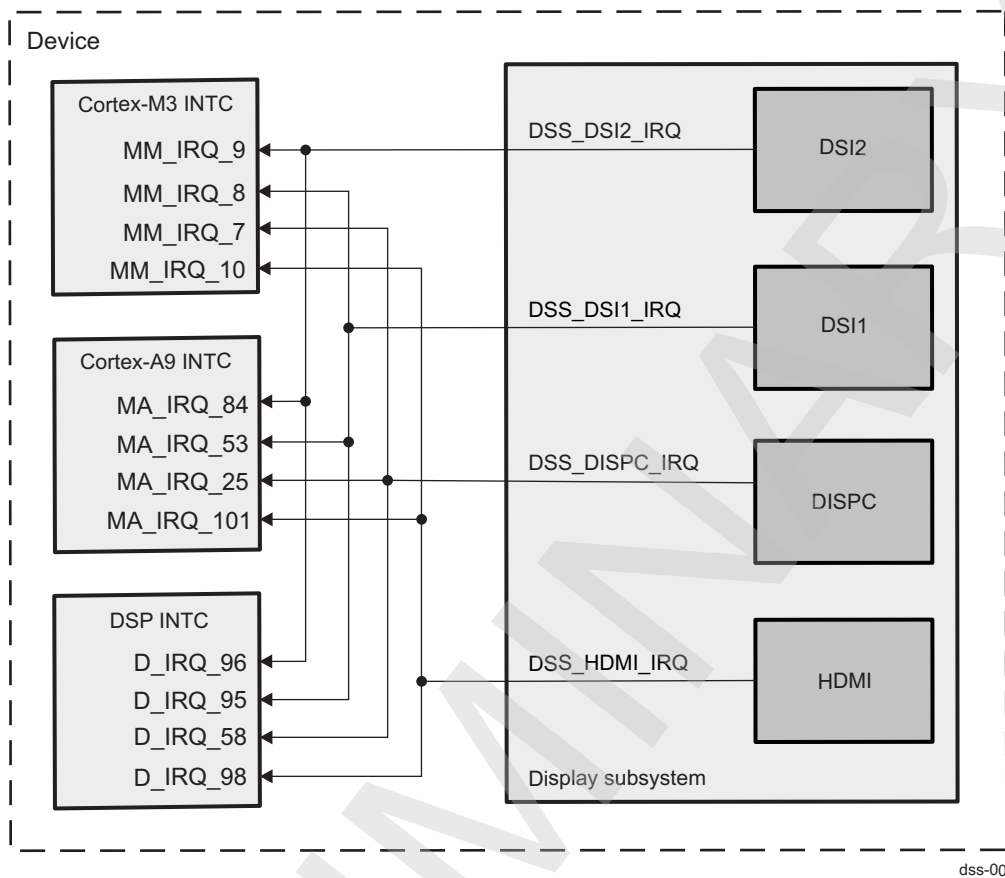


Table 10-7 lists the DSS interrupts.

Table 10-7. DSS Interrupts

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
DSS				
DISPC	DSS_DISPC_IRQ	D_IRQ_58	DSP INTC	DISPC interrupt request
		MA_IRQ_25	Cortex™-A9 INTC	DISPC interrupt request
		MM_IRQ_7	Cortex™-M3 INTC	DISPC interrupt request
MIPI DSI1	DSS_DSI1_IRQ	D_IRQ_95	DSP INTC	Display DSI1 interrupt request
		MA_IRQ_53	Cortex-A9 INTC	Display DSI1 interrupt request
		MM_IRQ_8	Cortex-M3 INTC	Display DSI1 interrupt request
MIPI DSI2	DSS_DSI2_IRQ	D_IRQ_96	DSP INTC	Display DSI2 interrupt request
		MA_IRQ_84	Cortex-A9 INTC	Display DSI2 interrupt request
		MM_IRQ_9	Cortex-M3 INTC	Display DSI2 interrupt request
HDMI	DSS_HDMI_IRQ	D_IRQ_98	DSP INTC	Display HDMI interrupt request
		MA_IRQ_101	Cortex-A9 INTC	Display HDMI interrupt request
		MM_IRQ_10	Cortex-M3 INTC	Display HDMI interrupt request

### 10.1.2.5 DSS DMA Requests

Eleven DMA requests are connected to the system DMA (sDMA).

The DMA requests are:

- Four DMA request signals from each DSI protocol engine (data traffic: from SDRAM or IVA-HD SL2 to DSI protocol engine and from DSI protocol engine to SDRAM or IVA-HD SL2)
- One DMA request signal from RFBI module (data traffic: from SDRAM to RFBI)
- One DMA request signal from HDMI module (audio traffic: from SDRAM or IVA-HD SL2 to HDMI)
- One DMA request signal from DISPC module (used for synchronization of a logical channel in the sDMA for memory-to-memory transfers)

Figure 10-9 shows the details of DSS DMA request generation.

Figure 10-9. DSS DMA Request Generation

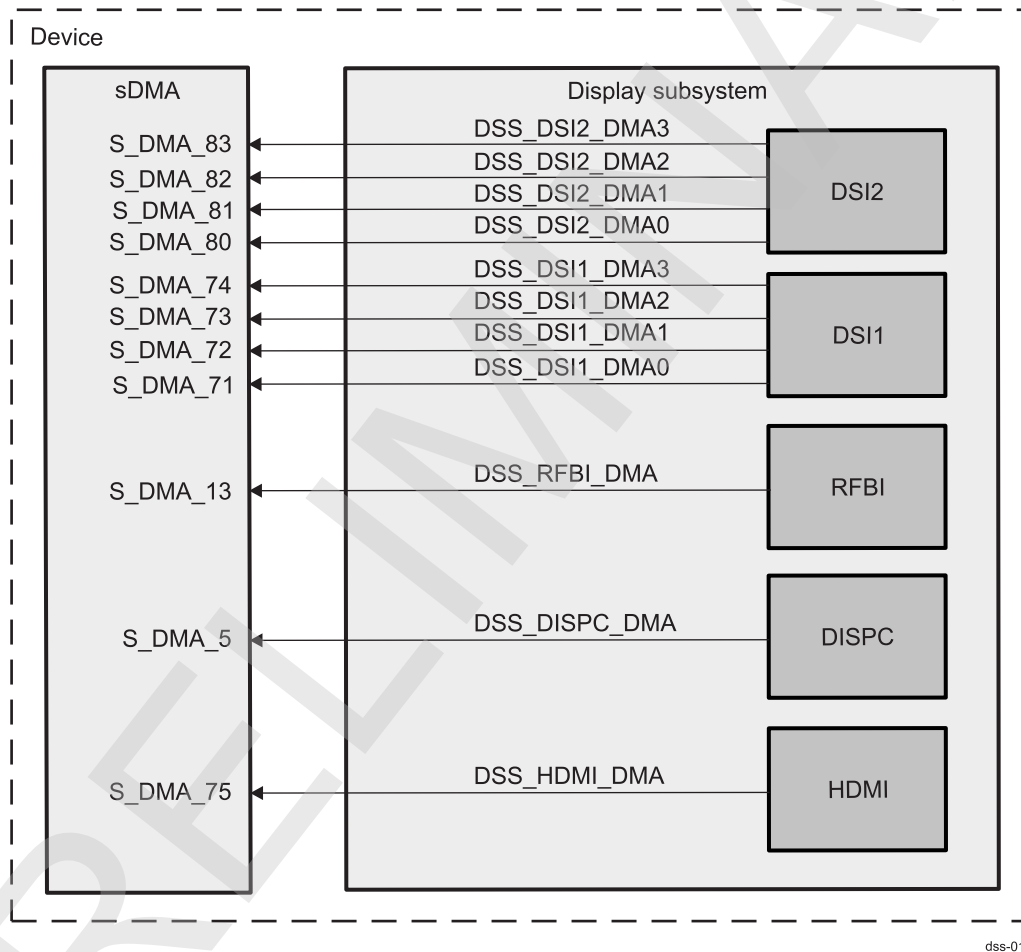


Table 10-8 lists the DSS DMA requests.

Table 10-8. DSS DMA Requests

DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
RFBI	DSS_RFBI_DMA	S_DMA_13	sDMA	DSS RFBI DMA request
DISPC	DSS_DISPC_DMA	S_DMA_5	sDMA	The line trigger signal to synchronize a memory-to-memory logical channel in the DMA4 sDMA is generated by the DISPC IP.
MIPI DSI1	DSS_DSI1_DMA0	S_DMA_71	sDMA	DSS DSI1 DMA request 0

**Table 10-8. DSS DMA Requests (continued)**

	DSS_DSI1_DMA1	S_DMA_72	sDMA	DSS DSI1 DMA request 1
	DSS_DSI1_DMA2	S_DMA_73	sDMA	DSS DSI1 DMA request 2
	DSS_DSI1_DMA3	S_DMA_74	sDMA	DSS DSI1 DMA request 3
MIPI DSI2	DSS_DSI2_DMA0	S_DMA_80	sDMA	DSS DSI2 DMA request 0
	DSS_DSI2_DMA1	S_DMA_81	sDMA	DSS DSI2 DMA request 1
	DSS_DSI2_DMA2	S_DMA_82	sDMA	DSS DSI2 DMA request 2
	DSS_DSI2_DMA3	S_DMA_83	sDMA	DSS DSI2 DMA request 3
HDMI	DSS_HDMI_DMA	S_DMA_75	sDMA	DSS HDMI audio DMA request

### 10.1.3 DSS Register Manual

**CAUTION**

The main access to all DSS registers is through the L3 interconnect.  
The access through the L4\_PER interconnect is provided for back software compatibility.

#### 10.1.3.1 DSS Instance Summary

**Table 10-9. DSS Instance Summary**

Module Name	L4_PER Base Address	L3 Base Address	Size
DSS	0x4804 0000	0x5800 0000	4KB

#### 10.1.3.2 DSS Registers

##### 10.1.3.2.1 DSS Registers Mapping Summary

Table 10-10 summarizes the DSS register mapping.

**Table 10-10. DSS Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSS L4_PER Physical Address	DSS L3 Physical Address
DSS_REVISION	R	32	0x0000 0000	0x4804 0000	0x5800 0000
RESERVED	R	32	0x0000 0010	0x4804 0010	0x5800 0010
DSS_SYSSTATUS	R	32	0x0000 0014	0x4804 0014	0x5800 0014
DSS_CTRL	RW	32	0x0000 0040	0x4804 0040	0x5800 0040
DSS_STATUS	R	32	0x0000 005C	0x4804 005C	0x5800 005C

##### 10.1.3.2.2 DSS Register Description

Table 10-11 through Table 10-17 describe the register bits.

**Table 10-11. DSS\_REVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	DSS_FAMILY
<b>Physical Address</b>	See Table 10-10.		
<b>Description</b>	This register contains the DSS revision number.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 10-12. Register Call Summary for Register DSS\_REVISION**

- DSS Register Manual
- [DSS Registers Mapping Summary: \[0\]](#)

**Table 10-13. DSS\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	DSS_FAMILY
<b>Physical Address</b>	See <a href="#">Table 10-10</a> .		
<b>Description</b>	This register provides status information about the module.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RESETDONE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is ongoing. Read 0x1: Reset complete	R	0x1

**Table 10-14. Register Call Summary for Register DSS\_SYSSTATUS**

DSS Register Manual

- [DSS Registers Mapping Summary: \[0\]](#)

**Table 10-15. DSS\_CTRL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	DSS_FAMILY
<b>Physical Address</b>	See <a href="#">Table 10-10</a> .		
<b>Description</b>	This register contains the DSS control bits.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																LCD2_TV_SEL	RESERVED	VENC_HDMI_SWITCH	RFBI_SWITCH	SYNC_SWITCH	LCD2_CLK_SWITCH	TV_CLK_SWITCH	DSI2_CLK_SWITCH	FCK_CLK_SWITCH	RESERVED	VENC_OUT_SEL	DAC_POWERDN_BGZ	DAC_DEMEN	VENC_CLOCK_4X_ENABLE	VENC_CLOCK_MODE	DSI1_CLK_SWITCH	LCD1_CLK_SWITCH

Bits	Field Name	Description	Type	Reset
31: 18	RESERVED	Reserved	R	0x0000
17	LCD2_TV_SEL	Selection between LCD2 and TV channel out on the LCD2 parallel output (multiplexer 13) 0x0: Select LCD2 channel output (default selection) 0x1: Select TV channel output	RW	0x0
16	RESERVED	Reserved	R	0x0
15	VENC_HDMI_SWITCH	Selects HDMI sync and associated clock or VENC and its associated TV_CLK 0x0: VENC selected 0x1: HDMI selected	RW	0x0

Bits	Field Name	Description	Type	Reset
14	RFBI_SWITCH	Selects the video port from DISPC between video port 1 and video port 2 (multiplexer 11). Read 0x0: Video port 1 (also named primary LCD output or LCD1) is selected (backward-compatible mode). Read 0x1: Video port 2 (also named secondary LCD output or LCD2) is selected.	RW	0x0
13	SYNC_SWITCH	Selects the sync generator for SD video (DSS VENC IP, or component VENC IP connected at the top) Read 0x0: Internal VENC IP selected Read 0x1: Component VENC IP selected	RW	0x0
12	LCD2_CLK_SWITCH	DSS_CLK/PLL2_CLK1 clock switch (multiplexer 3) Selects the clock source for the DISPC LCD2_CLK clock 0x0: DSS_CLK selected (from PRCM) 0x1: PLL2_CLK1 selected	RW	0x0
11	TV_CLK_SWITCH	DSS_TV_CLK/PLL3_CLK clock switch Selects the clock source for the VENC/HDMI functional clock 0x0: PLL3_CLK1 selected (from HDMI PLL) 0x1: DSS_TV_CLK selected (from PAD)	RW	0x0
10	DSI2_CLK_SWITCH	DSS_CLK/PLL2_CLK2 clock switch Selects the clock source for the DSI2 functional clock DSI2_CLK 0x0: DSS_CLK selected (from PRCM) 0x1: PLL2_CLK2 selected (from DSI2 PLL)	RW	0x0
9:8	FCK_CLK_SWITCH	Selects the clock source for the DISPC functional clock DISPC_FCLK 0x0: DSS_CLK selected (from PRCM) 0x1: PLL1_CLK1 selected (from DSI1 PLL) 0x2: PLL2_CLK1 selected (from DSI2 PLL) 0x3: PLL3_CLK1 selected (from HDMI PLL)	RW	0x0
7	RESERVED	Write 0's for future compatibility. Reads returns 0.	R	0x0
6	VENC_OUT_SEL	VENC mode selection for VENC output multiplexer 0x0: Composite VENC output selected 0x1: Luminance VENC output selected	RW	0x0
5	DAC_POWERDN_BGZ	DAC Power-down band gap control 0x0: DAC power-down band gap disabled 0x1: DAC power-down band gap enabled	RW	0x0
4	DAC_DEMEN	DAC Dynamic Element Matching Enable 0x0: DAC dynamic element matching disabled 0x1: DAC dynamic element matching enabled	RW	0x0
3	VENC_CLOCK_4X_ENABLE	VENC clock 4x enable 0x0: Disable 0x1: Enable	RW	0x0
2	VENC_CLOCK_MODE	VENC clock mode. See <a href="#">Table 10-4</a> . 0x0: Mode0. All three balanced clocks, derived from the TV_CLK clock, are provided to the VENC if the VENC_CLOCK_4X_ENABLE [3] bit is set to 1 by software. 0x1: Mode1. The VENC_CLOCK_4X_ENABLE [3] bit is used to control the clock gating.	RW	0x0

Bits	Field Name	Description	Type	Reset
1	DSI1_CLK_SWITCH	DSS_CLK/PLL1_CLK2 clock switch Selects the clock source for the DSI1 functional clock DSI1_CLK  0x0: DSS_CLK selected (from PRCM) 0x1: PLL1_CLK2 selected (from DSI1 PLL)	RW	0x0
0	LCD1_CLK_SWITCH	DSS_CLK/PLL1_CLK1 clock switch (multiplexer 2) Selects the clock source for the DISPC LCD1_CLK clock  0x0: DSS_CLK selected (from PRCM) 0x1: PLL1_CLK1 selected (from DSI1 PLL)	RW	0x0

**Table 10-16. Register Call Summary for Register DSS\_CTRL**

DSS Environment

- [Parallel Interface: \[0\] \[1\] \[6\] \[7\]](#)

DSS Integration

- [DSS Clocks: \[2\] \[3\] \[4\]](#)

DSS Register Manual

- [DSS Registers Mapping Summary: \[5\]](#)

**Table 10-17. DSS\_STATUS**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	DSS_FAMILY
<b>Physical Address</b>	See <a href="#">Table 10-10</a> .		
<b>Description</b>	This register contains the DSS status.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED									RFBI_STATUS		SYNC_STATUS		FCK_CLK_STATUS		TV_CLK_STATUS		LCD2_CLK_STATUS		DSI2_CLK_STATUS		DSI1_CLK_STATUS		TV_CLK_OUT_STATUS		RESERVED		LCD1_CLK_STATUS					

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21	RFBI_STATUS	Video port selection status (multiplexer 11) Indicates if video port 1 or video 2 from DISPC is used to provide data to the RFBI  Read 0x0: Video port 1 (named also primary LCD output or LCD1) used to provide data to RFBI Read 0x1: Video port 2 (named also secondary LCD output or LCD2) used to provide data to RFBI	R	0x0
20:19	SYNC_STATUS	Sync generator selection status (multiplexer 9) Indicates if DSS VENC IP or external component VENC IP is selected as syncs generator (master mode)  Read 0x0: Internal VENC IP selected Read 0x1: Component VENC IP selected Read 0x2: HDMI IP selected	R	0x0



Bits	Field Name	Description	Type	Reset
18:15	FCK_CLK_STATUS	DISPC_FCLK clock selection status (multiplexer 1), indicates which clock is used by the glitch-free multiplexer selecting the source of DISPC_FCLK. The current and the new-selected clocks must be running to be able to switch. Both clocks are used at the same time while the switching is ongoing.  Read 0x0: DSS_CLK clock switching is ongoing. Read 0x1: DSS_CLK is used by DISPC as DISPC_FCLK clock.  Read 0x2: PLL1_CLK1 is used by DISPC as DISPC_FCLK clock. Read 0x4: PLL2_CLK1 is used by DISPC as DISPC_FCLK clock. Read 0x8: PLL3_CLK1 (TV_CLK) is used by DISPC as DISPC_FCLK clock.	R	0x1
14:13	TV_CLK_STATUS	TV_CLK clock selection status (multiplexer 7), indicates which clock is used by the multiplexer selecting the source of TV_CLK. The current and the new-selected clocks must be running to be able to switch. Both clocks are used at the same time while the switching is ongoing.  Read 0x1: PLL3_CLK1 is used as TV_CLK. Read 0x2: DSS_TV_CLK is used as TV_CLK.	R	0x1
12:11	LCD2_CLK_STATUS	LCD2_CLK clock selection status (multiplexer 3), indicates which clock is used by the glitch-free multiplexer selecting the source of LCD2_CLK. The current and the new-selected clocks must be running to be able to switch. Both clocks are used at the same time while the switching is ongoing.  Read 0x0: LCD2_CLK clock switching is ongoing. Read 0x1: DSS_CLK is used as LCD2_CLK. Read 0x2: PLL2_CLK1 is used by DISPC as LCD2_CLK clock.	R	0x1
10:9	DSI2_CLK_STATUS	DSI2_CLK clock selection status (multiplexer 5), indicates which clock is used by the glitch-free multiplexer selecting the source of DSI2_CLK. The current and the new-selected clocks must be running to be able to switch. Both clocks are used at the same time while the switching is ongoing.  Read 0x0: DSI2_CLK clock switching is ongoing Read 0x1: DSS_CLK is used by DSI2 as DSI2_CLK clock. Read 0x2: PLL2_CLK2 is used by DSI2 as DSI2_CLK clock.	R	0x1
8:7	DSI1_CLK_STATUS	DSI1_CLK clock selection status (multiplexer 4), indicates which clock is used by the glitch-free multiplexer selecting the source of DSI1_CLK. The current and the new-selected clocks must be running to be able to switch. Both clocks are used at the same time while the switching is ongoing.  Read 0x0: DSI1_CLK clock switching is ongoing. Read 0x1: DSS_CLK is used by DSI1 as DSI1_CLK clock. Read 0x2: PLL1_CLK2 is used by DSI1 as DSI1_CLK clock.	R	0x1

Bits	Field Name	Description	Type	Reset
6:5	TV_CLK_OUT_STATUS	<p>TV_CLK_OUT selection status (multiplexer 12) indicates which clock is used by the multiplexer selecting the TV_CLK_OUT of the DISPC.</p> <p>The current and the new-selected clocks must be running to be able to switch.</p> <p>Both clocks are used at the same time while the switching is ongoing.</p> <p>Read 0x1: TV_CLK is used by DISPC as TV_CLK_OUT clock.</p> <p>Read 0x2: VIDEO_M_PCLK is used by DISPC as TV_CLK_OUT clock.</p>	R	0x1
4:2	RESERVED	Reserved	R	0x0
1:0	LCD1_CLK_STATUS	<p>LCD1_CLK clock selection status (multiplexer 2), indicates which clock is used by the glitch-free multiplexer selecting the source of LCD1_CLK.</p> <p>The current and the new-selected clocks must be running to be able to switch.</p> <p>Both clocks are used at the same time while the switching is ongoing.</p> <p>Read 0x0: LCD1_CLK clock switching is ongoing.</p> <p>Read 0x1: DSS_CLK is used as LCD1_CLK.</p> <p>Read 0x2: PLL1_CLK1 is used by DISPC as LCD1_CLK clock.</p>	R	0x1

**Table 10-18. Register Call Summary for Register DSS\_STATUS**

DSS Register Manual

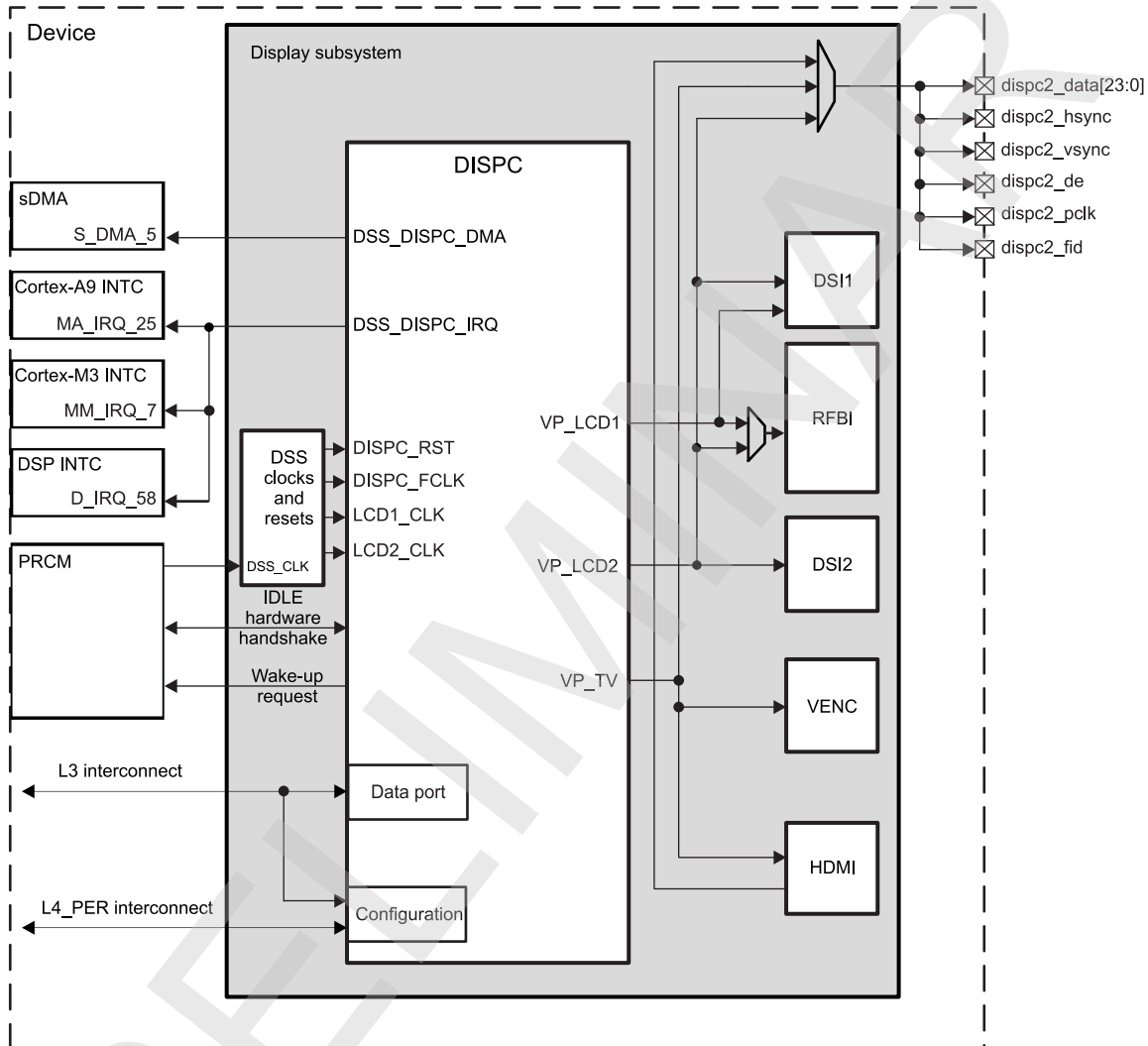
- [DSS Registers Mapping Summary: \[0\]](#)

## 10.2 Display Controller

### 10.2.1 Display Controller Overview

Figure 10-10 shows a block diagram of the display subsystem.

Figure 10-10. Display Subsystem Highlight



dispc-064

The DISPC includes the following main features:

- Five pipelines for processing:
  - One Graphics pipeline (GFX):
    - Pixel formats: ARGB16-4444, xRGB12-4444, RGBA16-4444, RGBx12-4444, RGB16-565, ARGB16-1555, xRGB16-1555, ARGB32-8888, RGBA32-8888, xRGB32-8888, RGBx32-8888, RGB24-888, and BITMAP (1-, 2-, 4-, or 8-bits per pixel) (x means that the correspond bits in the container are not used)
    - Selection of the color depth expansion from ARGB16-4444, RGBA16-4444, and ARGB16-1555 to ARGB32-8888 and from xRGB12-4444, RGBx12-4444, and xRGB16-1555 to xRGB32-8888 (replication of the MSBs or adding 0s)
    - Support for color look-up table (CLUT): 256 × 24-bit entries palette in RGB
    - Support for antiflicker on RGB pixel formats using 3-tap filter.
  - Three Video pipelines (VID1,VID2, and VID3):

- Pixel formats: ARGB16-4444, xRGB12-4444, RGBx12-4444, RGBA12-3333, RGBA16-4444, RGB16-565, ARGB16-1555, xRGB16-1555, ARGB32-8888, RGBA32-8888, xRGB32-8888, RGBx32-8888, RGB24-888, YUV422-UYYVY, YUV422-YUV2, and YUV420-NV12 (x means that the correspond bits in the container are not used)
- Selection of the color depth expansion from ARGB16-4444, RGBA16-4444, and ARGB16-1555 to ARGB32-8888 and from xRGB12-4444, RGBx12-4444, and xRGB16-1555 to xRGB32-8888 (replication of the MSBs or adding 0s)
- Programmable polyphase filter:
  - Independent horizontal and vertical resampling: upsampling (up to x8) and downsampling (down to 1/4)
  - Maximum input width of 1920 pixels
  - No limitation on the input height
  - Supported input formats are ARGB32-8888, YUV422-UYYVY, YUV422-YUV2, and YUV420-NV12
  - Alpha blending factor is rescaled like the R, G, and B color components.
- Programmable color space conversion from YUV4:2:2 (YUV4:4:4, YUV4:2:0 after Chroma upsampling through the scaler) into ARGB32-8888. Images in YUV4:2:2 format with 90/270 degree rotation are preprocessed to YUV4:4:4 before the scaler, by duplicating the missing Chroma.
- Programmable VC-1 range mapping
- One Write-back pipeline (WB): allows to take benefit of the hardware processing available inside the DISPC like color space conversion, rescaling, compositing to perform memory-to-memory transfer with data processing or capturing a displayed frame.
  - Programmable color space conversion RGB24 into YUV4:4:4 or to YUV422-UYYVY, YUV422-YUV2, or YUV420-NV12 using programmable polyphase filter.
  - Programmable color space conversion RGB24 into YUV422-UYYVY, YUV422-YUV2, or YUV420-NV12
  - Selection of the color depth reduction from RGB24 to RGB16
  - Programmable polyphase filter:
    - Independent horizontal and vertical resampling: upsampling (up to x8) and downsampling (down to 1/4)
    - Maximum input width of 1920 pixels
    - No limitation on the input height
    - Supported input formats are ARGB32-8888, YUV422-UYYVY, YUV422-YUV2, and YUV420-NV12
    - Alpha blending factor is rescaled like the R, G, and B color components.
  - Selection of the source of the data:
    - Overlay output:
      - Primary LCD
      - Secondary LCD
      - TV
    - Pipelines:
      - Graphic
      - Video 1
      - Video 2
      - Video 3

- Two LCD outputs, primary (LCD1) and secondary (LCD2):
  - Input pixel format: ARGB32-8888
  - Output pixel format: RGB24-888
  - Overlay of graphic and video 1, 2, and 3 pipelines
  - Source and destination transparency color key
  - Global and pixel alpha blending (up to 8-bit blending factor)
  - Z-order programmable (full flexibility)
  - Three types of displays are supported:
    - Passive matrix monochrome: 4-/8-bit monochrome panel interface support (15 grayscale levels supported using dithering block)
    - Passive matrix color: 8-bit color panel interface support (3375 colors supported for a color panel using dithering block)
    - Active matrix color: 12-/16-/18-/24-bit panel interface support (replicated or dithered encoded pixel values)
  - Independent programmable timing generators for LCD1 and LCD2 to support:

<b>Primary LCD</b>	Using DSI1 interface	SXGA VESA timing @ 60 FPS
	Using RFBI interface	WVGA @ 30 fps
<b>Secondary LCD</b>	Using DSI2 interface	XGA VESA timing @ 85 fps
	Using RFBI interface	WVGA @ 30 fps
	Using CMOS interface	XGA VESA timing @ 85 fps 1080i/720p CEA-861-D

- Configurable LCD output mode: progressive or interlace mode
- Selection between RGB and YUV4:2:2 output pixel format
- Remote frame buffer support through the RFBI module
- Partial display through the RFBI
- Multiple-cycle output format on 8-/9-/12-/16-bit interface time division multiplexing (TDM)
- One TV output
  - Input pixel format: ARGB40-10.10.10.10
  - Output pixel format: ARGB40-10.10.10.10
  - Overlay of graphic and video 1, 2, and 3 pipelines
  - Source and destination transparency color key
  - Global and pixel alpha blending (up to 10-bit blending factor)
  - Z-order programmable (full flexibility)
  - Slave mode support (no master mode support) with synchronization signals provided by the VENC (NTSC/PAL encoder or HDMI TX)
    - HSYNC (horizontal synchronization signal)
    - VSYNC (vertical synchronization signal)
    - RE (data request signal)
    - FID (field ID: odd/even field information)
  - RGB24-888 data bus output for connection to the NTSC/PAL VENC
  - RGB30-10.10.10 data bus output for connection to HDMI TX and extended to 36 by duplication of the MSBs
  - SD-576i and SD-480i using NTSC/PAL VEC
  - HD-1080p, HD-1080i, HD-720p, SD-480p, SD-576p, SD-576i, and SD-480i using HDMI
  - HDMI deep color mode support, 30-bit data output to HDMI encoder
  - Pixel duplication capability (from one pixel clock cycle up to eight cycles)
- Panel support with MIPI DPI protocol
  - 4-/8-bit monochrome passive matrix panel interface support (15 grayscale levels supported using dithering block)
  - 8-bit color passive matrix panel interface support (3375 colors supported for a color panel using dithering block)

- 12-/16-/18-/24-bit active matrix panel interface support (replicated or dithered encoded pixel values)
- Common
  - Rotation 0, 90, 180, and 270 using DMM-TILER
  - Synchronized buffer update
  - Hardware cursor (using the graphics pipeline or one of the video pipelines)
  - Independent gamma curve support on primary LCD output, secondary LCD output and TV output
  - Multiple-buffer support
  - Mirroring/flip-flop support (using DMM-TILER)
  - Programmable color phase rotation (CPR)
  - Alpha blending support:
    - Embedded pixel factor (ARGB and RGBA)
    - Global alpha
- DMA (internal to the DISPC)
  - Support for accessing tiled structure through the TILER inside the dynamic memory management (DMM)
  - Support for rotation, flip-flop, and mirroring through the TILER inside the DMM
  - Support for memory fragmentation through the TILER inside the DMM
  - Integrated shared buffers between DMA engine and pipelines
  - Programmable buffer thresholds
  - Bandwidth limiter on write request (insertion on idle cycles between requests)
- Advanced
  - Mode outputting data on display only from the DMA buffer (self-refresh using the DMA FIFO)
  - DMA buffer hand-check in STALL mode
  - Arbitration between high/low priority (GFX, VID1, VID2, VID3, and WB pipelines)
- Power modes
  - Low-power saving modes
  - Support on-the-fly dynamic voltage and frequency scaling (DVFS)
  - Merge capability of the DMA buffers to support greater OFF period on the L3 interconnect
    - All buffers associated to a single pipeline
    - Reallocation of the buffers of the nonactive pipelines to the active pipelines

### 10.2.2 Display Controller Environment

The DISPC provides the required control signals to interface directly to an external parallel panel for the MIPI DPI protocol.

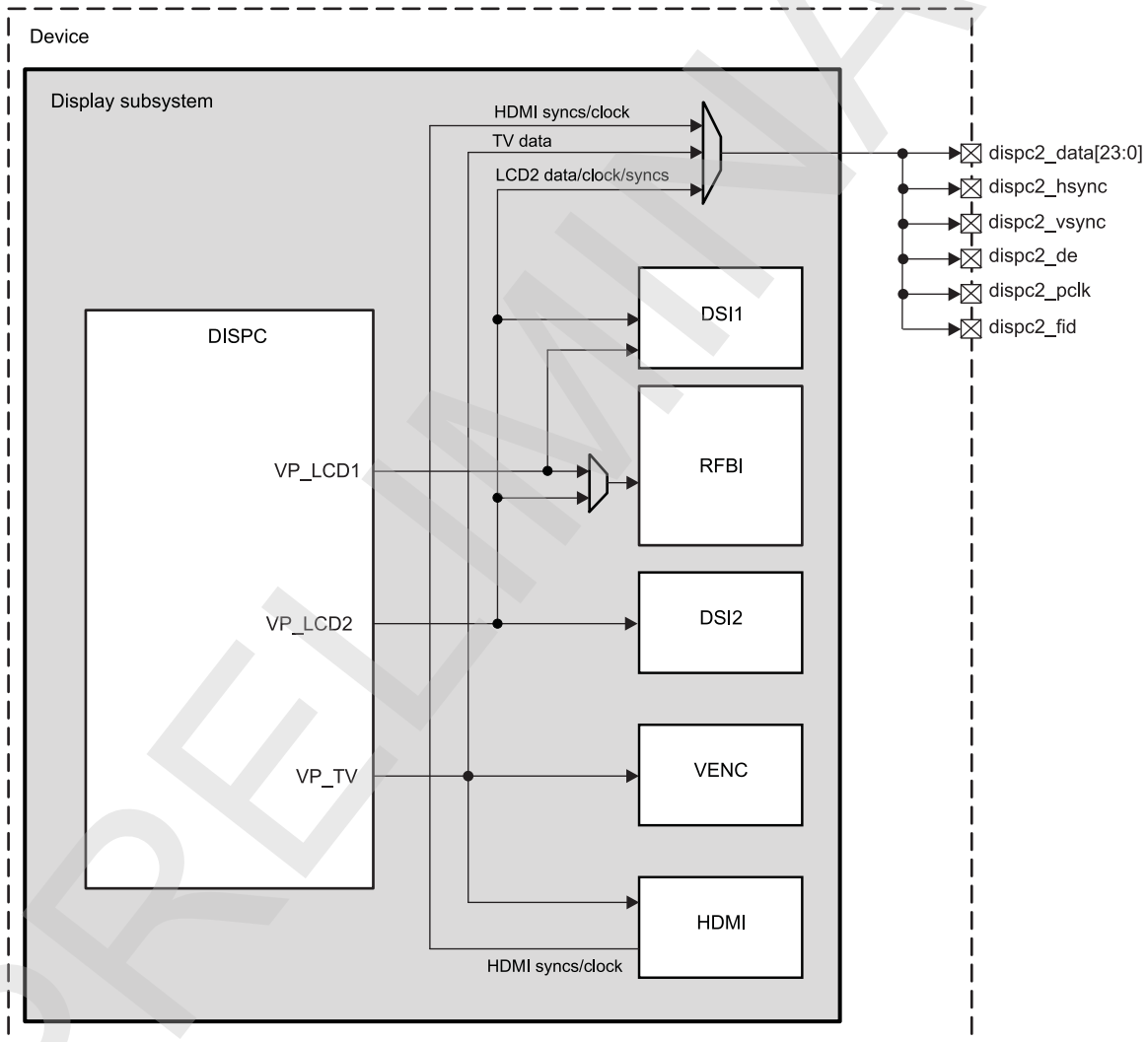
Figure 10-11 shows the LCD support parallel interface.

**NOTE:** Parallel interface is available through the LCD2 and TV outputs of the DISPC.

The LCD2 data and control signals are multiplexed with the TV output data, and control signals provided by the HDMI module.

The selection can be done at DSS top level. For further details and signals mapping, see [Section 10.1.1.1.1 Parallel Interface](#), under [Section 10.1 Display Subsystem Overview](#).

**Figure 10-11. LCD Support Parallel Interface**



dispc-083

Table 10-19 describes the interface signals to/from the LCD panel in bypass mode.

**Table 10-19. Parallel Interface Signals**

Signal Name	Type <sup>(1)</sup>	Description
dispc2_data[23:0]	O	Pixels data

<sup>(1)</sup> I = Input, O = Output, I/O = Input/Output

**Table 10-19. Parallel Interface Signals (continued)**

Signal Name	Type <sup>(1)</sup>	Description
dispc2_pclk	O	Pixel clock
dispc2_vsync	O	Vertical synchronization. The LCD frame clock (vsync) toggles after all the lines in a frame are transmitted to the LCD panel and a programmable number of line clock cycles has elapsed at the beginning and at the end of each frame.
dispc2_hsync	O	Horizontal synchronization. The LCD line clock (hsync) toggles after all pixels in a line are transmitted to the LCD panel and a programmable number of pixel clock wait-states elapse, at the beginning and the end of each line.
dispc2_de	O	<ul style="list-style-type: none"> <li>In passive matrix technology, the de signal is configured to transition each time a programmable number of line clocks occurs. To prevent a dc charge within the screen pixels, the power and ground supplies of the panel are periodically switched. The DISPC signals the panel to switch the polarity by toggling the de pin.</li> <li>In active matrix technology, the de signal acts as an output-enable signal to indicate when data must be latched using the pixel clock.</li> </ul>
dispc2_fid	O	The FID signal indicates the field identifier for the LCD output field <ul style="list-style-type: none"> <li>0 means even.</li> <li>1 means odd.</li> </ul>

### 10.2.2.1 LCD Output and Data Format for the Parallel Interface

This section describes the pixel data bus and shows timing diagrams of transactions and synchronizations.

Figure 10-12 through Figure 10-18 show the pixel data bus, depending on the use of 4-, 8-, 12-, 16-, 18-, or 24-pixel data output pins.

Table 10-20 lists the number of displayed pixels per pixel clock cycle based on the type of display panel.

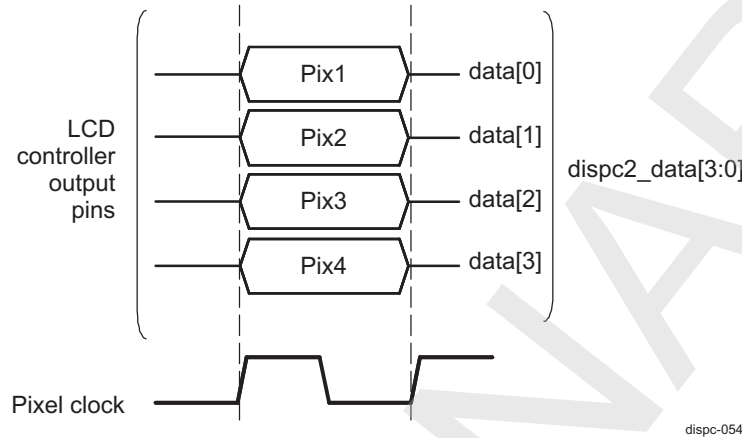
**Table 10-20. Number of Displayed Pixels per Pixel Clock Cycle Based on Display Type**

Display Panel	Number of Displayed Pixels per Pixel Clock Cycle
Monochrome 4-bit	4
Monochrome 8-bit	8
Passive matrix color	8/3
Active matrix	1

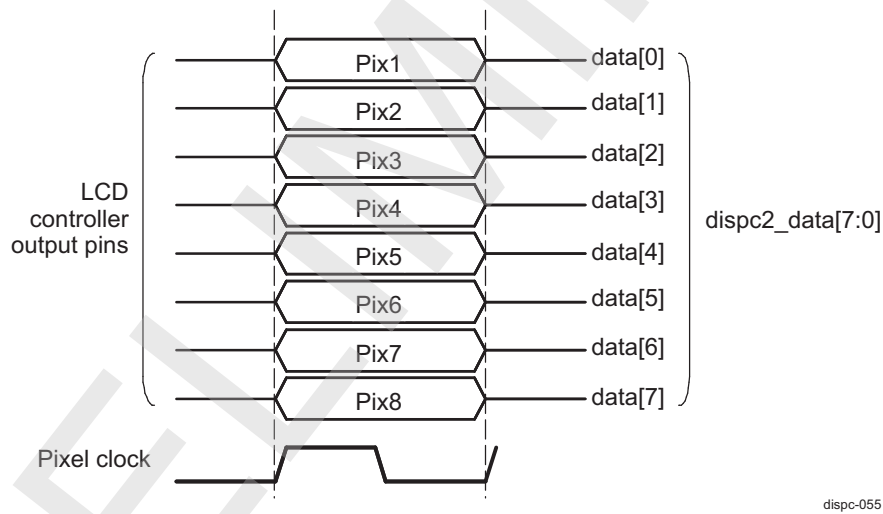


- Passive matrix technology, monochrome mode  
 Monochrome displays use a 4-bit or 8-bit interface. Each bit represents one pixel (on or off), which means that 4 or 8 pixels are sent to the LCD at each pixel clock.  
 Figure 10-12 and Figure 10-13 show 4- and 8-bit monochrome displays, respectively.

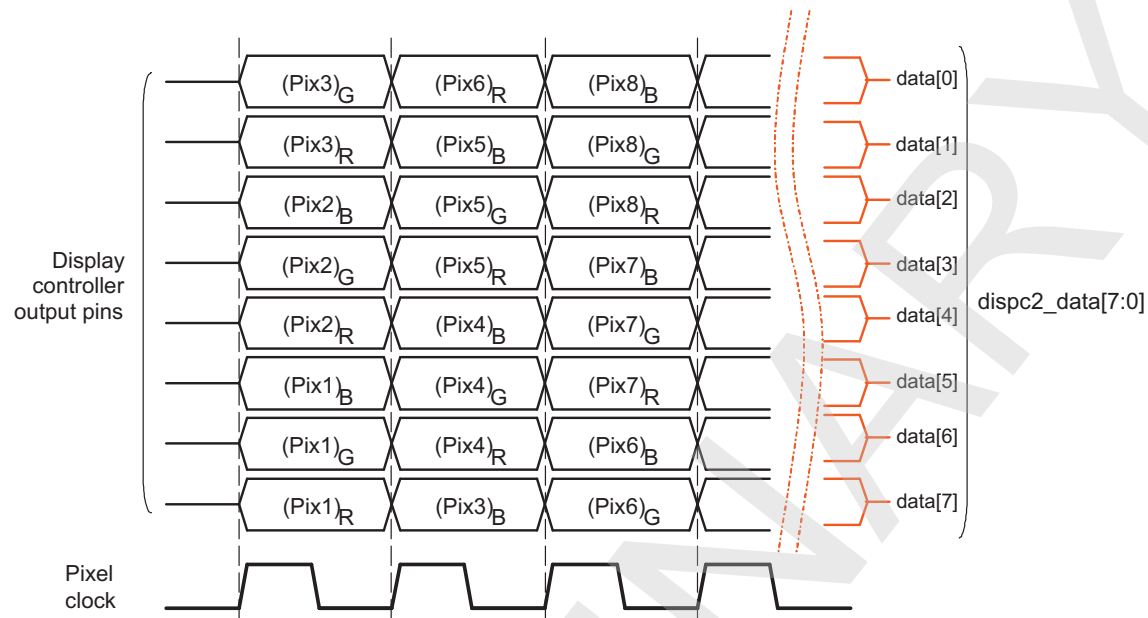
**Figure 10-12. LCD Pixel Data Monochrome4 Passive Matrix**



**Figure 10-13. LCD Pixel Data Monochrome8 Passive Matrix**

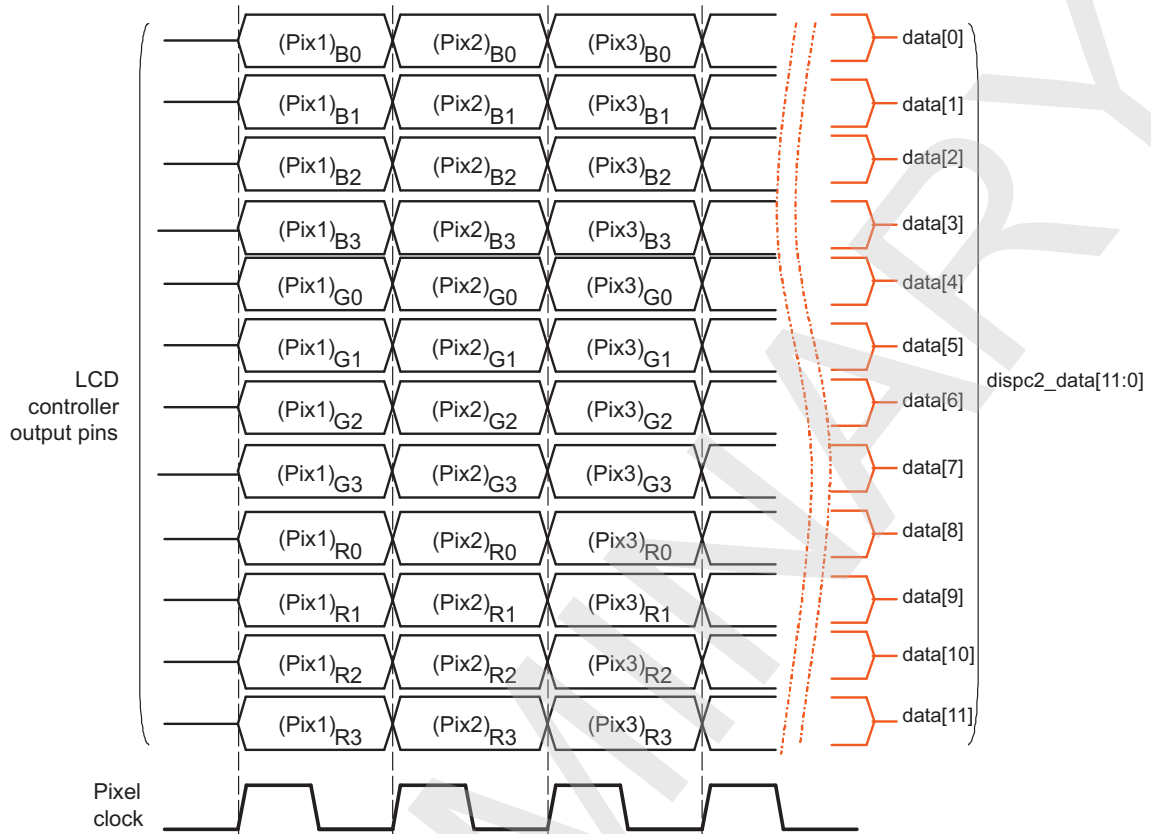


- Passive matrix technology, color mode  
 Color passive displays use 8-bit data input lines. In a given pixel clock cycle, each line represents one color component (red, green, or blue).  
 Figure 10-14 shows an 8-bit color passive matrix display.

**Figure 10-14. LCD Pixel Data 8-Bit Color Passive Matrix**

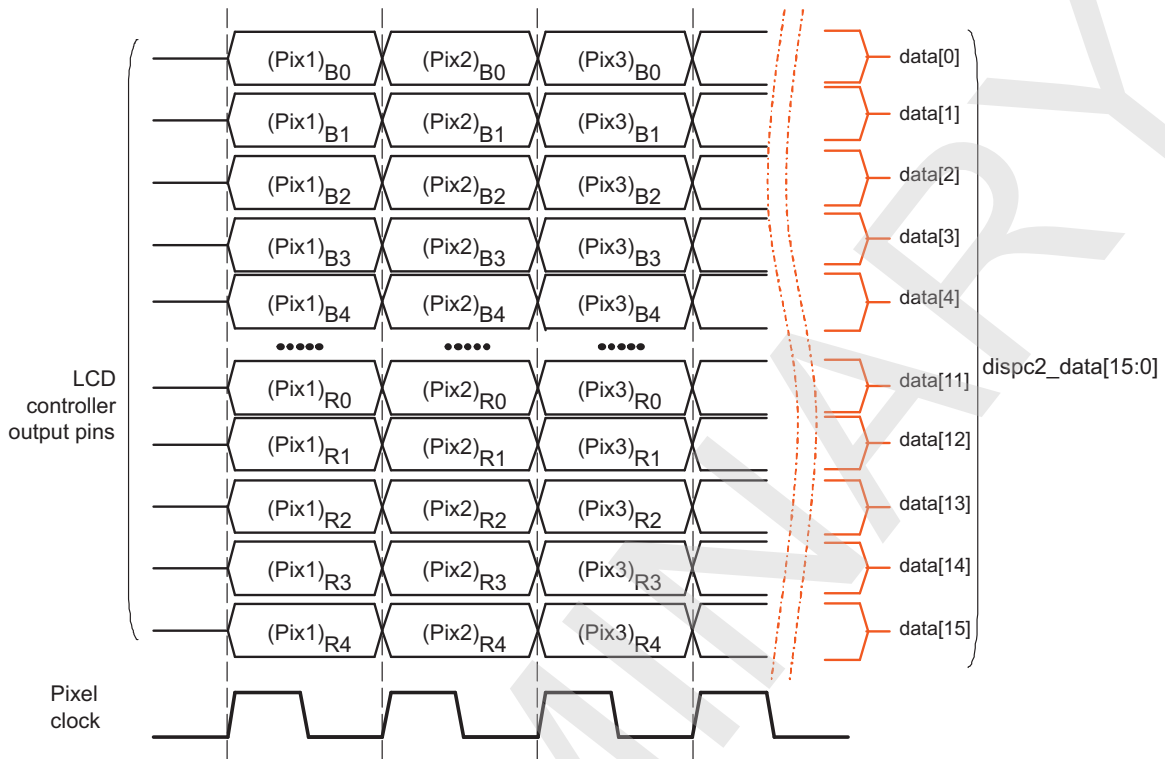
- Active matrix technology  
Active matrix displays bypass the STN dithering logic block and the output FIFO. Each line represents one pixel.  
Figure 10-15 through Figure 10-18 show 12-, 16-, 18-, and 24-active matrix displays, respectively.

Figure 10-15. LCD Pixel Data Color12 Active Matrix



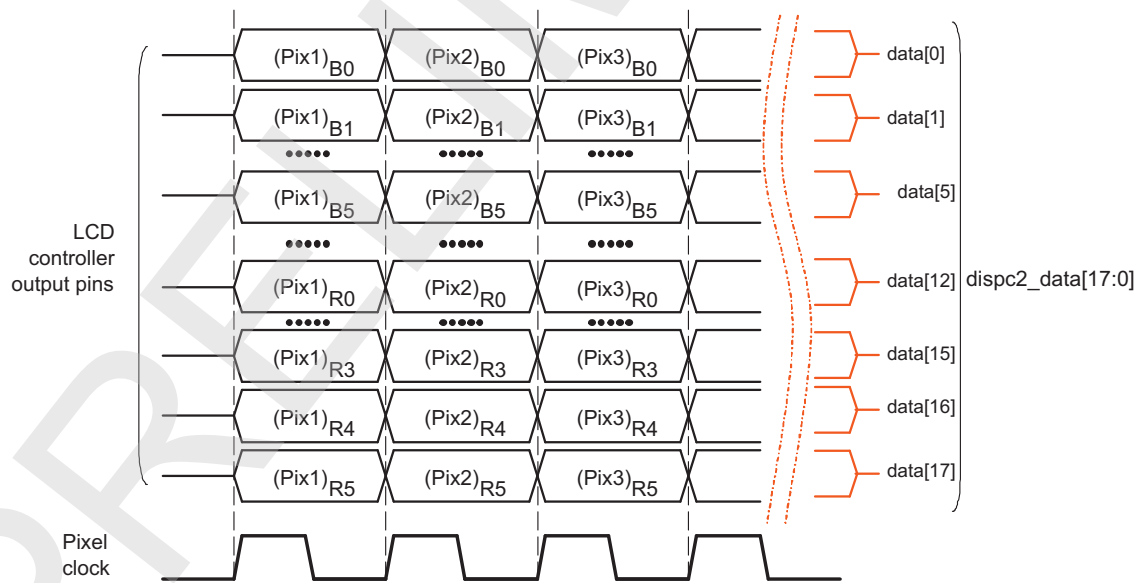
dispc-050

**Figure 10-16. LCD Pixel Data Color16 Active Matrix**



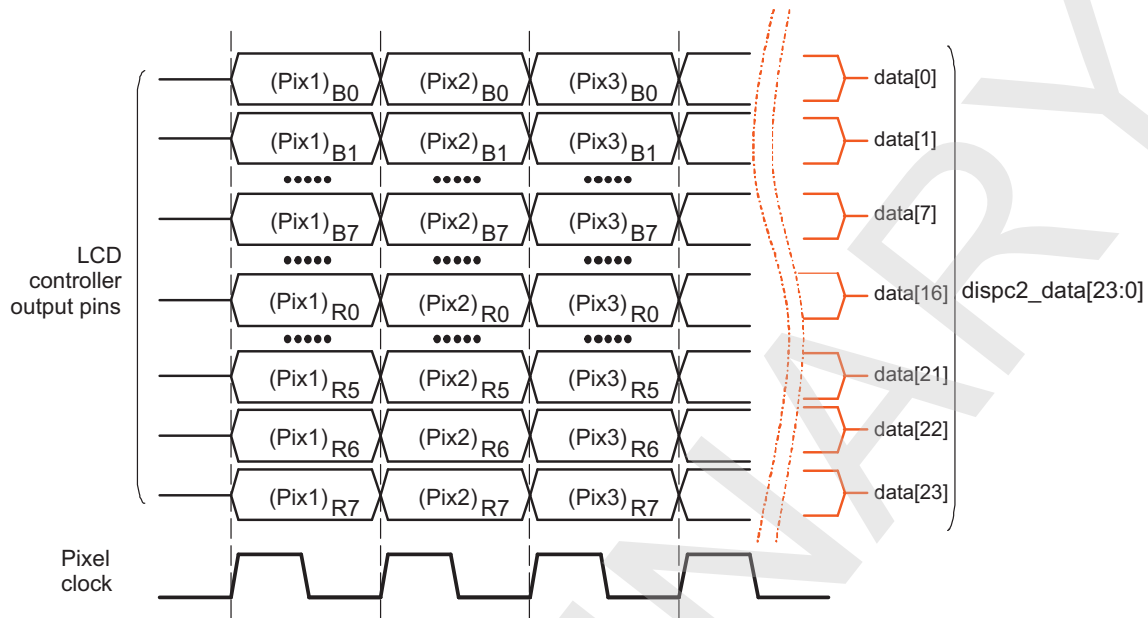
dispc-051

**Figure 10-17. LCD Pixel Data Color18 Active Matrix**



dispc-052

Figure 10-18. LCD Pixel Data Color24 Active Matrix



dispc-053

### 10.2.2.2 Transaction Timing Diagrams

Figure 10-19 through Figure 10-34 show timing diagrams of synchronization signals and pixel clock for passive matrix and active matrix panels. The DISPC directly drives these signals, which are related to the programmable fields described in Table 10-21.

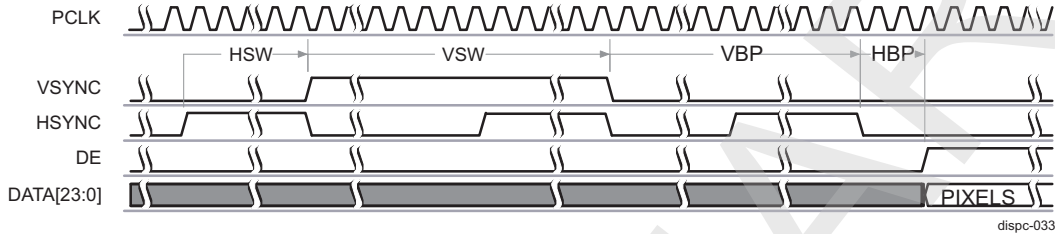
Table 10-21. Programmable Fields in Bypass Mode

Name	Register	Description
PPL	DISPC_SIZE_LCD0[10:0] PPL bit field value + 1	Pixels per line
LPP	DISPC_SIZE_LCD0[26:16] LPP bit field value + 1	Lines per panel
HBP	DISPC_TIMING_Ho[31:20] HBP bit field value + 1	Horizontal back porch
HFP	DISPC_TIMING_Ho[19:8] HFP bit field value + 1	Horizontal front porch
HSW	DISPC_TIMING_Ho[7:0] HSW bit field value + 1	Horizontal synchronization pulse width
VBP	DISPC_TIMING_Vo[31:20] VBP bit field value	Vertical back porch
VFP	DISPC_TIMING_Vo[19:8] VFP bit field value	Vertical front porch
VSW	DISPC_TIMING_Vo[7:0] VSW bit field value + 1	Vertical synchronization pulse width
ONOFF	DISPC_POL_FREQ0[17] ONOFF bit	DISPC_HSYNC and DISPC_VSYNC pixel clock control
RF	DISPC_POL_FREQ0[16] RF bit	DISPC_HSYNC and DISPC_VSYNC pixel clock edge control
IEO	DISPC_POL_FREQ0[15] IEO bit	Invert DISPC_ACBIAS
IPC	DISPC_POL_FREQ0[14] IPC bit	Invert DISPC_PCLK
IHS	DISPC_POL_FREQ0[13] IHS bit	Invert DISPC_HSYNC
IVS	DISPC_POL_FREQ0[12] IVS bit	Invert DISPC_VSYNC

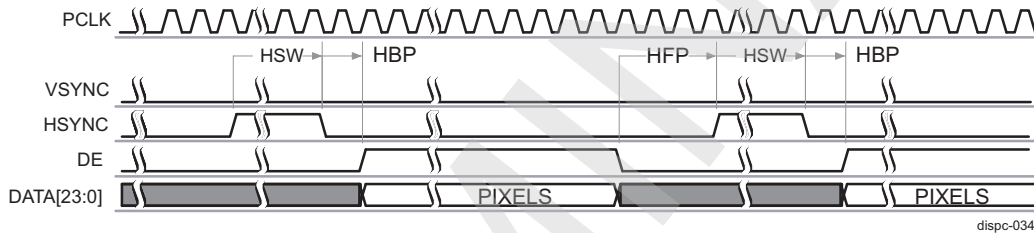
- Active matrix timing configuration 1
  - DISPC\_POL\_FREQ0[17] ONOFF bit = 0
  - DISPC\_POL\_FREQ0[16] RF bit = 0  
The HSYNC and VSYNC signals are driven on the opposite edge of PCLK from the pixel data.
  - DISPC\_POL\_FREQ0[15] IEO = 0  
The DE signal is active high.

- DISPC\_POL\_FREQo[14] IPC = 0  
The pixel data are driven on the rising edge of PCLK.
- DISPC\_POL\_FREQo[13] IHS = 0  
The HSYNC signal is active high.
- DISPC\_POL\_FREQo[12] IVS = 0  
The VSYNC signal is active high.

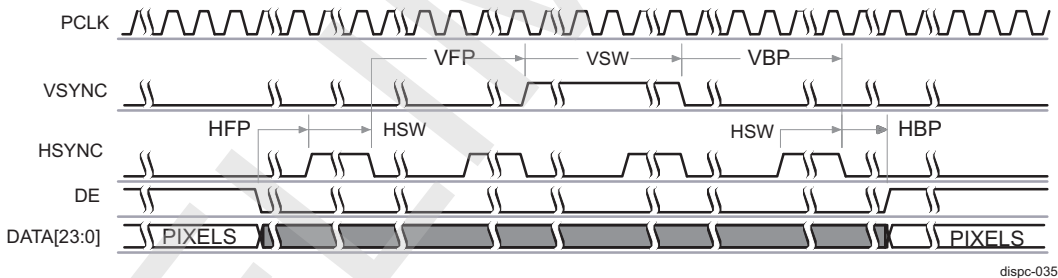
**Figure 10-19. Active Matrix Timing Diagram of Configuration 1 (Start of Frame)**



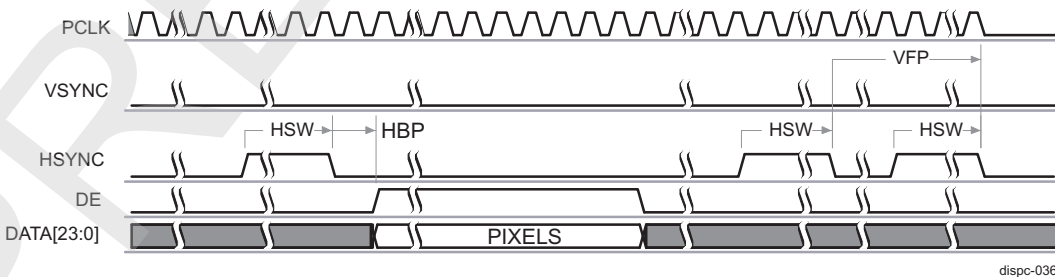
**Figure 10-20. Active Matrix Timing Diagram of Configuration 1 (Between Lines)**



**Figure 10-21. Active Matrix Timing Diagram of Configuration 1 (Between Frames)**



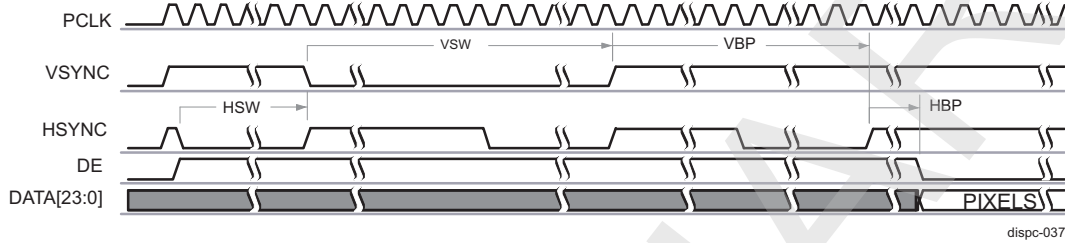
**Figure 10-22. Active Matrix Timing Diagram of Configuration 1 (End of Frame)**



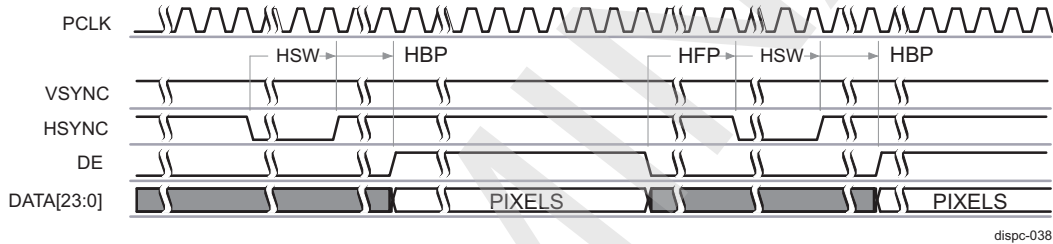
- Active matrix timing configuration 2
  - DISPC\_POL\_FREQo[17] ONOFF bit = 1
  - DISPC\_POL\_FREQo[16] RF bit = 1  
The HSYNC and VSYNC signals are driven on the rising edge of PCLK.
  - DISPC\_POL\_FREQo[15] IEO = 1  
The DE signal is active low.

- DISPC\_POL\_FREQo[14] IPC = 1  
The pixel data is driven on the falling edge of PCLK.
- DISPC\_POL\_FREQo[13] IHS = 1  
The HSYNC signal is active low.
- DISPC\_POL\_FREQo[12] IVS = 1  
The VSYNC signal is active low.

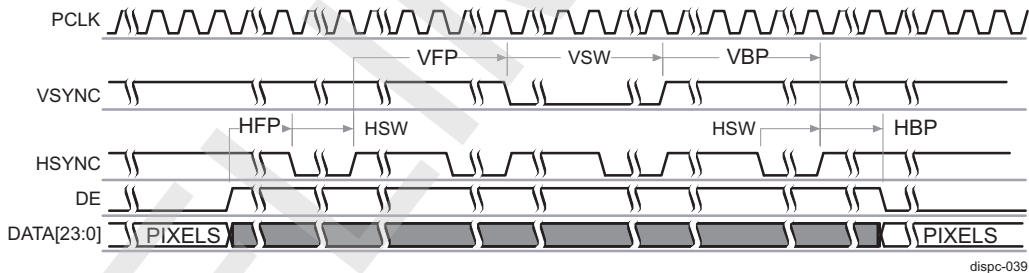
**Figure 10-23. Active Matrix Timing Diagram of Configuration 2 (Start of Frame)**



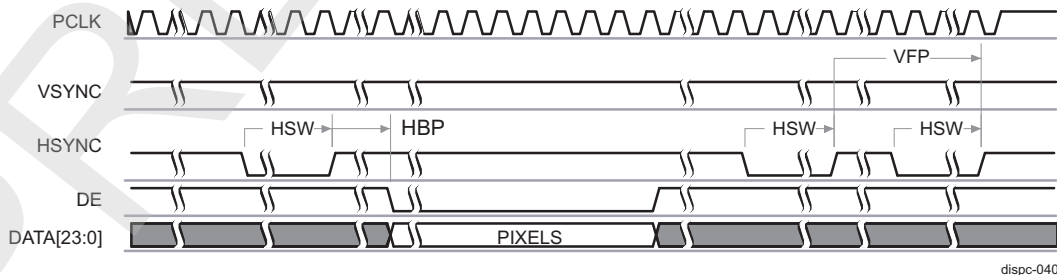
**Figure 10-24. Active Matrix Timing Diagram of Configuration 2 (Between Lines)**



**Figure 10-25. Active Matrix Timing Diagram of Configuration 2 (Between Frames)**

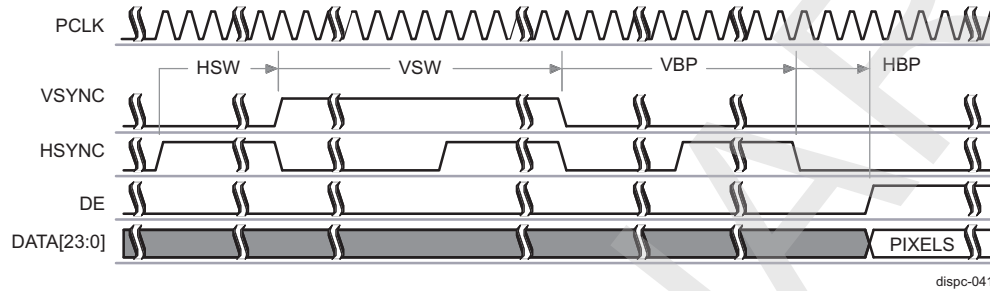
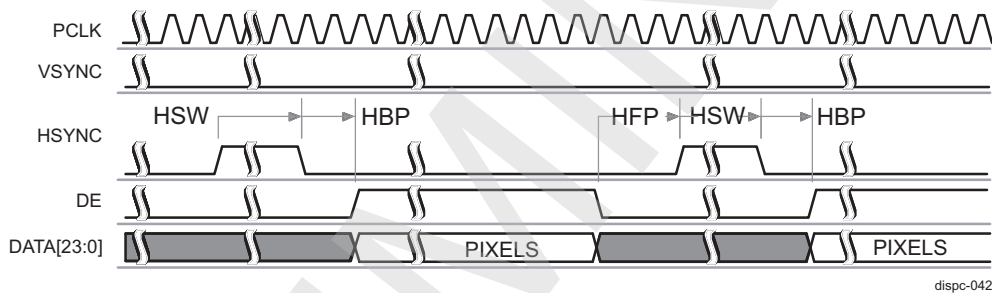
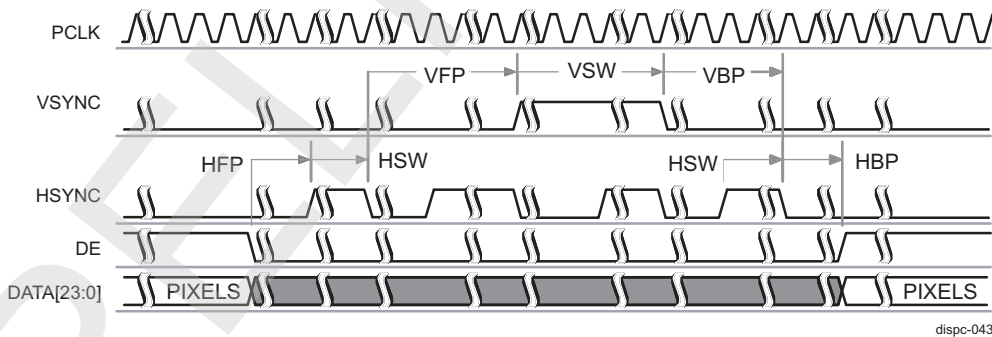


**Figure 10-26. Active Matrix Timing Diagram of Configuration 2 (End of Frame)**



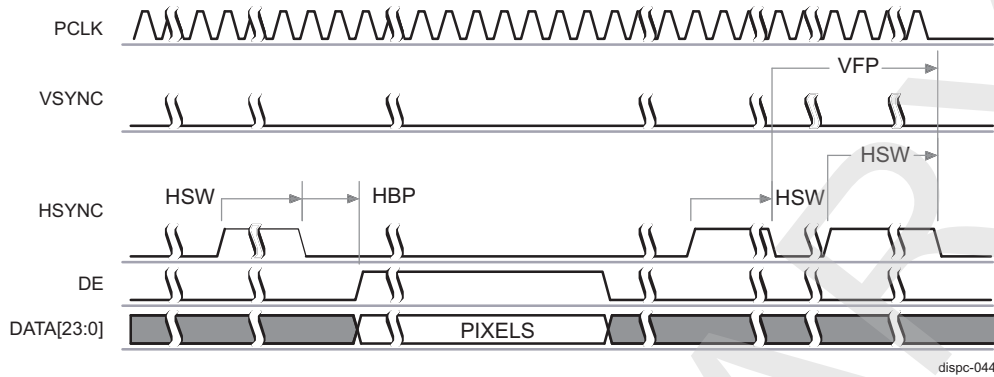
- Active matrix timing configuration 3
  - DISPC\_POL\_FREQo[17] ONOFF bit = 1
  - DISPC\_POL\_FREQo[16] RF bit = 1  
The HSYNC and VSYNC signals are driven on the rising edge of PCLK.
  - DISPC\_POL\_FREQo[15] IEO = 0  
The DE signal is active high.

- DISPC\_POL\_FREQo[14] IPC = 0  
The pixel data are driven on the rising edge of PCLK.
- DISPC\_POL\_FREQo[13] IHS = 0  
The HSYNC signal is active high.
- DISPC\_POL\_FREQo[12] IVS = 0  
The VSYNC signal is active high.

**Figure 10-27. Active Matrix Timing Diagram of Configuration 3 (Start of Frame)**

**Figure 10-28. Active Matrix Timing Diagram of Configuration 3 (Between Lines)**

**Figure 10-29. Active Matrix Timing Diagram of Configuration 3 (Between Frames)**


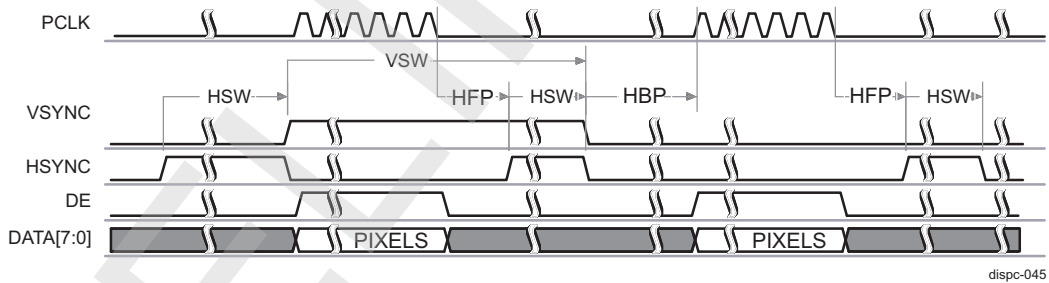


**Figure 10-30. Active Matrix Timing Diagram of Configuration 3 (End of Frame)**

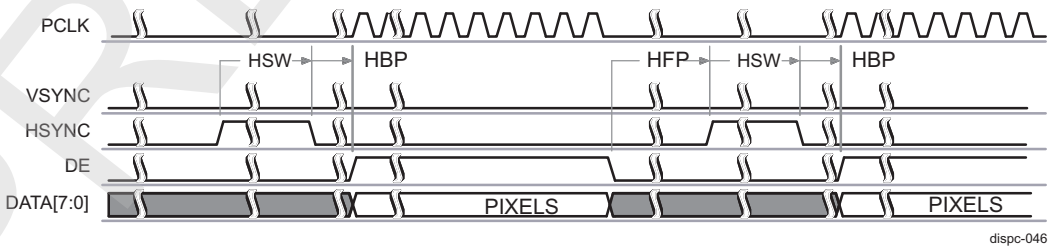


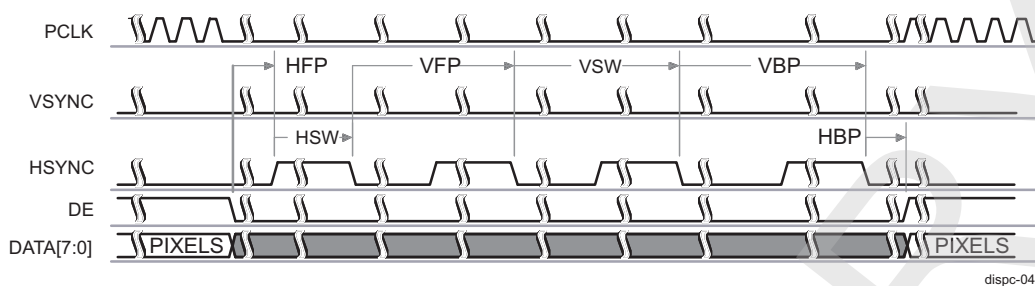
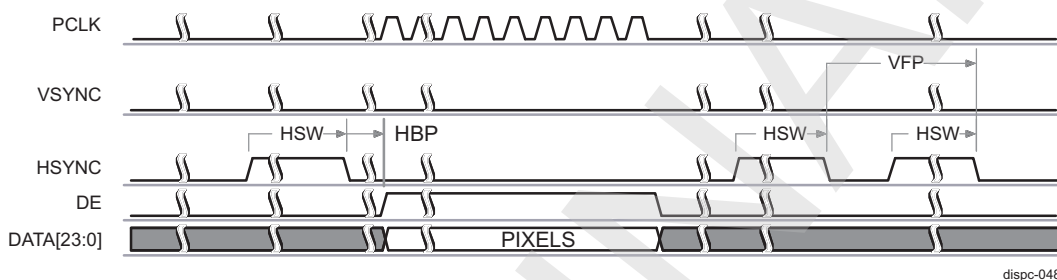
- Passive matrix timing configuration
  - DISPC\_POL\_FREQo[17] ONOFF bit = 0
  - DISPC\_POL\_FREQo[16] RF bit = 0  
The HSYNC and VSYNC signals are driven on the opposite edge of PCLK from the pixel data.
  - DISPC\_POL\_FREQo[15] IEO = 0  
The DE signal is active high.
  - DISPC\_POL\_FREQo[14] IPC = 0  
The pixel data are driven on the rising edge of PCLK.
  - DISPC\_POL\_FREQo[13] IHS = 0  
The HSYNC signal is active high.
  - DISPC\_POL\_FREQo[12] IVS = 0  
The VSYNC signal is active high.

**Figure 10-31. Passive Matrix Timing Diagram (Start of Frame)**



**Figure 10-32. Passive Matrix Timing Diagram (Between Lines)**



**Figure 10-33. Passive Matrix Timing Diagram (Between Frames)****Figure 10-34. Passive Matrix Timing Diagram (End of Frame)**

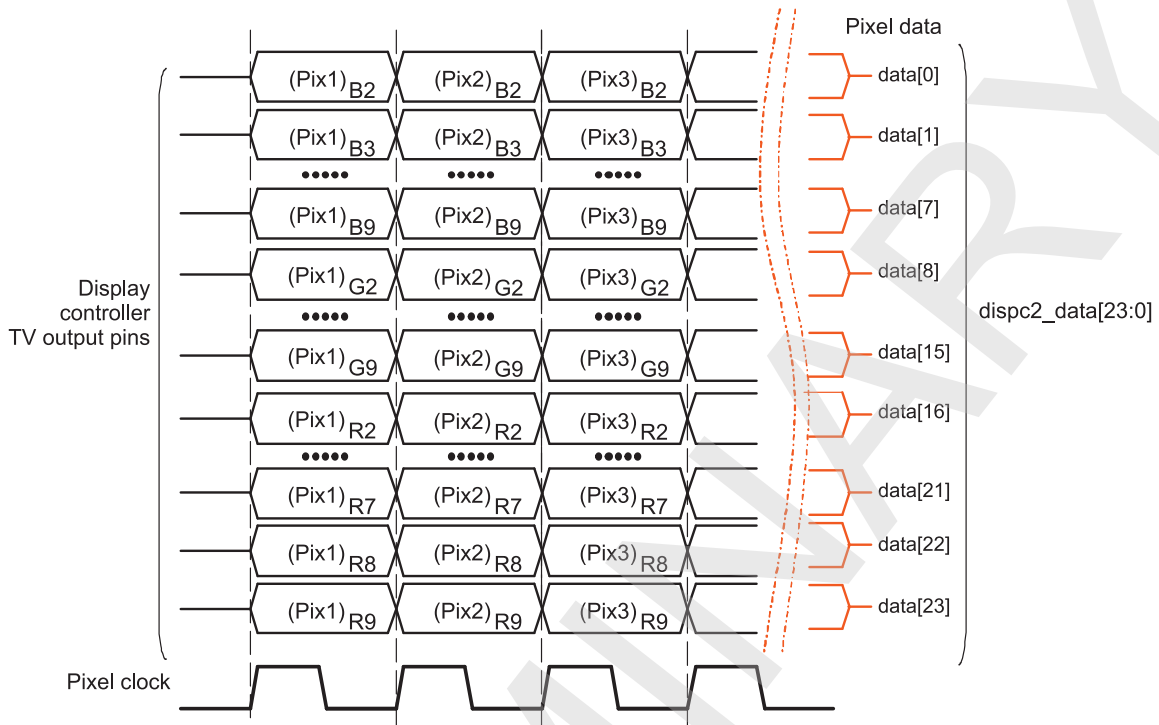
### 10.2.2.3 TV Output and Data Format for the Parallel Interface

This section describes the TV output pixel data bus for the parallel interface.

The TV pixel data interface is a 30-bit RGB interface. Only the MSB part of each color component is connected to the DSS boundary: R[9:2], G[9:2], B[9:2]. The output of the data is synchronized to the data request signal (HDMI\_M\_DE) from the HDMI encoder.

Figure 10-35 describes the TV output pixel data format.

Figure 10-35. TV Output Pixel Data

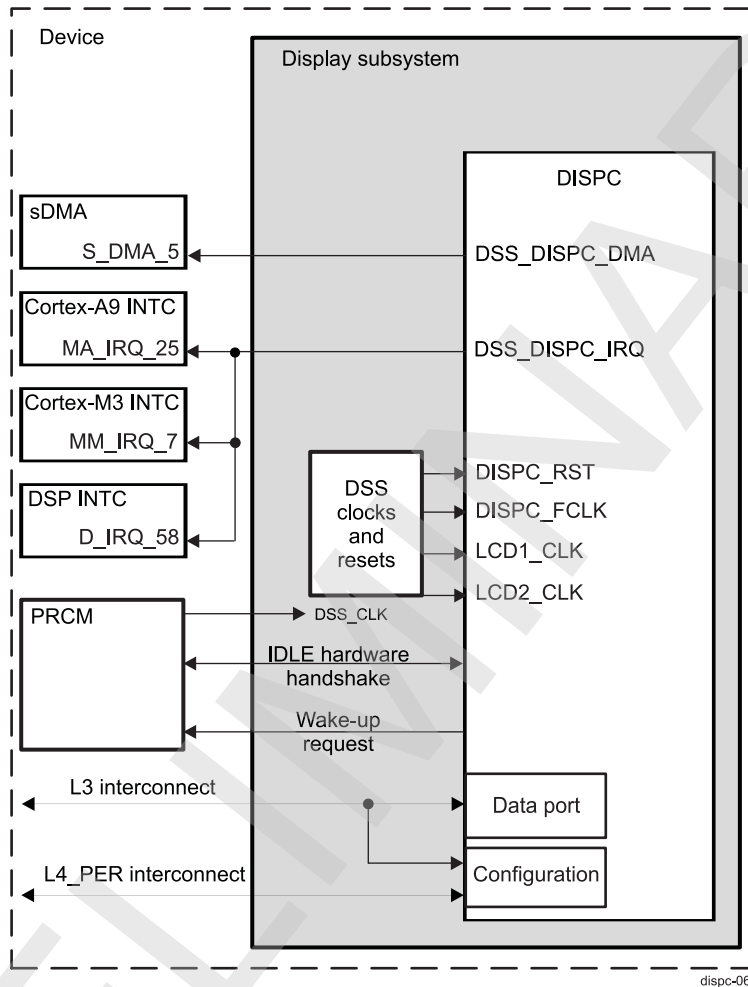


dispc-093

### 10.2.3 Display Controller Integration

This section describes the DISPC integration in the device. The complete details about clocks, resets are detailed in the [Section 10.1, Display Subsystem Overview](#).

**Figure 10-36. Display Subsystem Integration**



[Table 10-22](#) and [Table 10-23](#) list the integration attributes and clock and resets, respectively.

[Table 10-24](#) summarize the integration of the module in the device.

**Table 10-22. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
DISPC	PD_DSS	L3 for data transfer L4_PER for configuration

**Table 10-23. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DISPC	DISPC_FCLK	DSS_CLK/PLL1_CLK1/PLL2_CLK1/PLL3_CLK1	PRCM/PLLs	Functional clock for the DISPC logic. For multiplexing description, see <a href="#">Section 10.1.2, DSS Integration</a> .
	LCD1_CLK	DSS_CLK/PLL1_CLK1	PRCM/PLL1	Clock used to generate the divided pixel clock for the primary LCD interface. For multiplexing description, see <a href="#">Section 10.1.2, DSS Integration</a> .
	LCD2_CLK	DSS_CLK/PLL2_CLK1	PRCM/PLL2	Clock used to generate the divided pixel clock for the secondary LCD interface. For multiplexing description, see <a href="#">Section 10.1.2, DSS Integration</a> .
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DISPC	DISPC_RST	DISPSS_RST	PRCM/DSS	Hardware reset is coming from PRCM, or via a software reset performed at DSS level. For the tree reset description, see <a href="#">Section 10.1.2, DSS Integration</a> .

**Table 10-24. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
DISPC	DSS_DISPC_IRQ	MM_IRQ_	Cortex-M3	Display controller interrupt. For information about Cortex-M3 interrupt control, see <a href="#">Chapter 7</a> .
	DSS_DISPC_IRQ	D_IRQ_58	DSP	Display controller interrupt. For information about DSP interrupt control, see <a href="#">Chapter 5</a> .
	DSS_DISPC_IRQ	MA_IRQ_25	Cortex-A9	Display controller interrupt. For information about Cortex-A9 interrupt control, see <a href="#">Chapter 4</a> .

**Table 10-24. Hardware Requests (continued)**

DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
DISPC	DSS_DISPC_DMA	S_DMA_5	System DMA	The line trigger signal informs the sDMA that a programmable number of lines are output to the LCD, and that the system memory can be updated. For more details, see <a href="#">Section 10.2.4.5, sDMA Requests</a> .

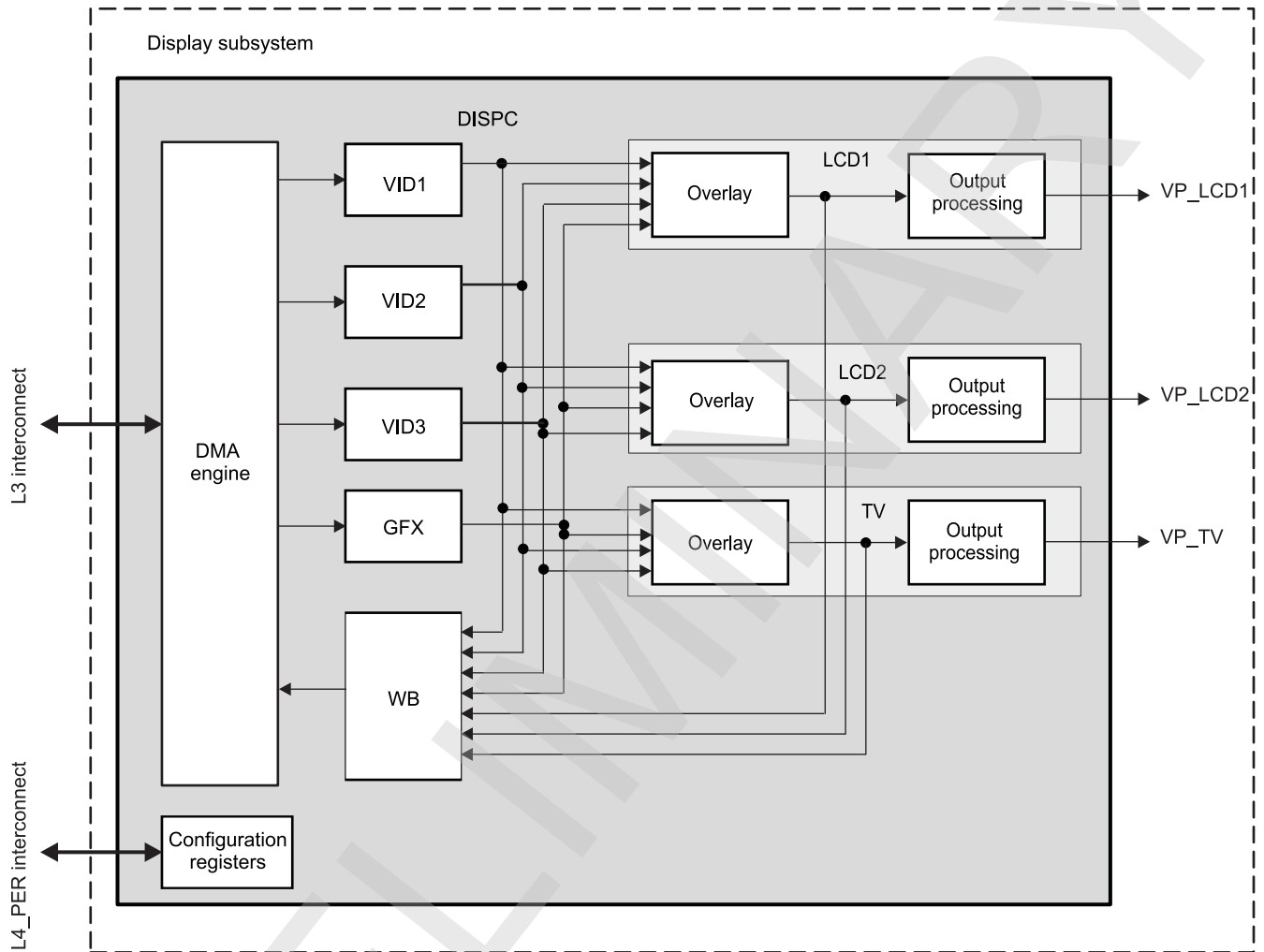
**NOTE:**

- For a description of interrupt sources, see [Section 10.2.4.4, Interrupt Requests](#).
- For a description of DMA sources, see [Section 10.2.4.5, sDMA Requests](#).

### 10.2.4 Display Controller Functional Description

The DISPC can read and display the encoded pixel data stored in memory (see Figure 10-37).

Figure 10-37. DISPC Architecture Overview



dispc-068

The DISPC can read and display the encoded pixel data stored in memory and write the output of one of the overlays or one of the pipelines into system memory.

Several processes can be configured to manage the graphics pipeline (palette, antiflicker) and video pipelines (VC-1, color space conversion, scaling, overlay, and transparency ...).

The data coming out of a pipeline is sent to one of the three outputs depending on user configuration. An overlay manager manages inputs of multiple pipelines. Timings user configuration for LCD and TV are available.

The DISPC allows capturing of one output of the pipeline or overlay manager to redirect it into the WB pipeline. It allows to take benefit of the hardware processing available inside the DISPC like color space conversion, rescaling, compositing, etc., to perform memory-to-memory transfer with data processing.

### 10.2.4.1 Clock Configuration

The PCLK frequency for each LCD output is derived from a dedicated input clock: LCD1\_CLK and LCD2\_CLK for the primary and secondary LCD outputs, respectively. Each input clock is divided by the values of the `DISPC_DIVISORo[23:16]` LCD bit field and then the `DISPC_DIVISORo[7:0]` PCD bit field independently for each LCD pixel clock (see Figure 10-38). LCD1\_PCLK and LCD2\_PCLK are independent.

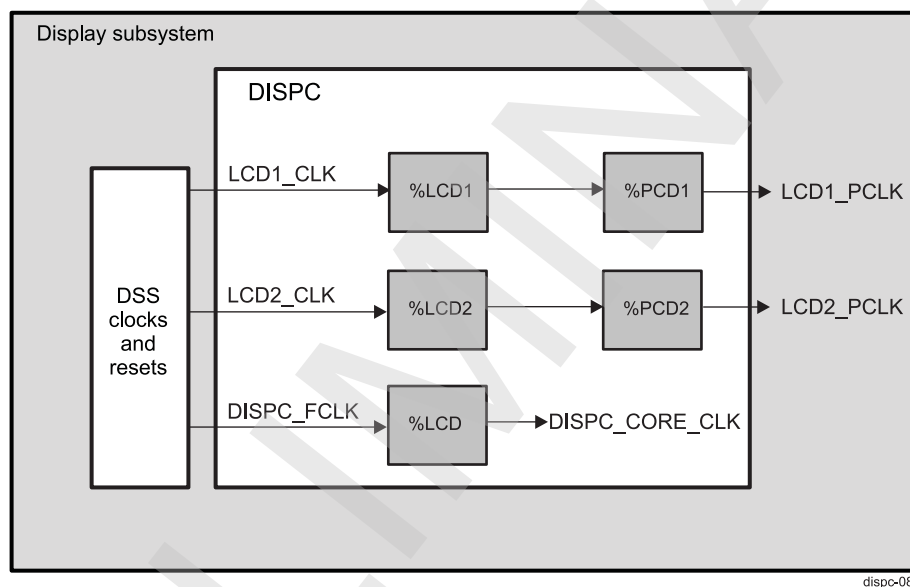
$$\text{LCD1\_PCLK} = (\text{LCD1\_CLK} / \text{LCD1}) / \text{PCD1}$$

$$\text{LCD2\_PCLK} = (\text{LCD2\_CLK} / \text{LCD2}) / \text{PCD2}$$

The functional clock of the DISPC is derived from DISPC\_FCLK by an independent divisor. The dividing value is set in the `DISPC_DIVISOR[23:16]` LCD bit field.

For backward compatibility, the divisor value LCD can be set to the value of LCD1. To enable this functionality, the `DISPC_DIVISOR[0]` ENABLE bit must be set to 1.

**Figure 10-38. DISPC Clock Tree Overview**



### 10.2.4.2 Software Reset

To perform a software reset on the DISPC, set the `DISPC_SYSCONFIG[1]` SOFTRESET bit to 0x1. The `DISPC_SYSSTATUS[0]` RESETDONE bit indicates that the software reset is complete when its value is 0x1. When the software reset completes, the `DISPC_SYSCONFIG[1]` SOFTRESET bit is automatically reset. Software must ensure that the software reset completes before doing DISPC operations.

### 10.2.4.3 Power Management

The DISPC supports the MStandby/Wait, IdleReq/SIdleAck, and Wake-up protocols as defined in PRCM chapter.

#### 10.2.4.3.1 Idle Mode

The DISPC supports no-idle mode, force-idle mode, and smart-idle mode. The mode can be selected by programming the appropriate value in the `DISPC_SYSCONFIG[4:3]` SIDLEMODE bit field.

Condition of assertion of SIdleAck signal:

- In no-idle mode: In this mode, SIdleAck is never asserted.
- In force-idle mode: In this mode, SIdleAck is asserted unconditionally with a 1-configuration port interface clock cycle delay with respect to IdleReq assertion.



**NOTE:** A proper use of force-idle mode assumes that no interrupt need to be generated.

- In smart-idle mode: In this mode, SIdleAck is asserted when all at least the following conditions are satisfied:
  - No interrupt is pending.
  - The DISPC no longer uses the interface clock for the slave port.

Once SIdleAck is asserted:

- The DISPC interface lock used by the slave port may be shut down at any time.
- Any transactions on the configuration port (L4\_PER interconnect) are ignored.

The condition of deassertion of SIdleAck signal are:

- In force-idle mode: In this mode, SIdleAck is deasserted with a 1-configuration port interface clock cycle delay with respect to IdleReq deassertion.
- In smart-idle mode: In this mode, SIdleAck is deasserted with a 1-configuration port interface clock cycle delay with respect to IdleReq deassertion.

Once SIdleAck is released, the DISPC is fully operational and a DMA request can be processed normally.

#### **10.2.4.3.2 StandBy Mode**

The DISPC supports the no-standby mode, the force-standby mode and a single smart-standby mode. The mode is set in MIDDLEMODE field [DISPC\\_SYSCONFIG\[13:12\]](#). The functional clock is always active if the module is enabled. The L3 clock can be shut down at any time independently of the status of MStandby.

The condition of assertion of MStandby signal are:

- In no-standby mode: In this mode, MStandby is never asserted.
- In force-standby mode: In this mode, MStandby is asserted when the module is disabled.
- In smart-standby: In the case of one of the following conditions:
  - GFX pipeline is disabled, or GFX pipeline is enabled but the data fetch is completed for GFX window, or GFX pipeline is enabled and the data fetch did not complete and data in the DMA buffer is greater than high threshold value.
  - VID1 pipeline is disabled, or VID1 pipeline is enabled but the data fetch is completed for VID1 window, or VID1 pipeline is enabled and the data fetch did not complete and data in the DMA buffer is greater than high threshold value.
  - VID2 pipeline is disabled, or VID2 pipeline is enabled but the data fetch is completed for VID2 window, or VID2 pipeline is enabled and the data fetch did not complete and data in the DMA buffer is greater than high threshold value.
  - VID3 pipeline is disabled, or VID3 pipeline is enabled but the data fetch is completed for VID3 window, or VID3 pipeline is enabled and the data fetch did not complete and data in the DMA buffer is greater than high threshold value.
  - WB pipeline is disabled, or WB pipeline is enabled but the data store to memory is completed for WB picture, or WB pipeline is enabled and the data storage did not complete but data in the DMA buffer is lower than low threshold value.

MStandby signal asserts whenever all five events have occurred or the DISPC is disabled. While the MStandby is asserted, the DISPC does not generate any transaction on the L3 master port.

The condition of deassertion of the MStandby signal are:

- In force-standby mode: In this mode, MStandby is deasserted only when the DISPC is enabled.
- In smart-standby mode: In the case of one of the following conditions:
  - GFX pipeline is enabled and the data fetch did not complete for GFX window and the data in the DMA buffer is less than low threshold value.
  - VID1 pipeline is enabled and the data fetch did not complete for VID1 window and the data in the DMA buffer is less than low threshold value.
  - VID2 pipeline is enabled and the data fetch did not complete for VID2 window and the data in the DMA buffer is less than low threshold value.

- VID3 pipeline is enabled and the data fetch did not complete for VID3 window and the data in the DMA buffer is less than low threshold value.
- WB pipeline is enabled and the data store did not complete for WB picture and the data in the DMA buffer more than high threshold value.

Detection of the deassertion conditions assumes that the interface clocks are active.

### 10.2.4.3.3 Wakeup

The DISPC supports wake-up signaling. The mode can be selected by programming the appropriate value in the [DISPC\\_SYSCONFIG](#)[2] ENAWAKEUP bit. The SWakeup signal is asynchronous so it does not require the interface clock.

The conditions of assertion of the SWakeup signal are:

- GFX pipeline is enabled and the data fetch did not complete for the GFX window and the data in DMA buffer are less than low threshold value.
- VID1 pipeline is enabled and the data fetch did not complete for the VID1 window and the data in DMA buffer are less than low threshold value.
- VID2 pipeline is enabled and the data fetch did not complete for the VID2 window and the data in DMA buffer are less than low threshold value.
- VID3 pipeline is enabled and the data fetch did not complete for the VID3 window and the data in DMA buffer are less than low threshold value.
- WB pipeline is enabled and the data in DMA buffer are more than high threshold value.
- The last pixel displayed into LCD1 panel if it is not the last frame
- The last pixel displayed into LCD2 panel if it is not the last frame
- The last pixel displayed into TV panel if it is not the last frame

SWakeup signal is asserted whenever any one of eight events occurs and IdleAck is asserted.

When one of the active pipelines reaches the low threshold and must refill its DMA buffer for the current frame, all other pipelines refill their own DMA buffer even if their low threshold has not been reached. The [DISPC\\_CONFIG](#)[17] BUFFERFILLING bit field is used to increase the probability that the time increases, where there is no access to the L3 interconnect.

The condition of deassertion of SWakeup signal is:

- Immediately after deassertion of IdleReq

### 10.2.4.4 Interrupt Requests

[Table 10-25](#) details the interrupts generated for the DISPC.

The interrupt line, DSS\_DISPC\_IRQ, indicates when one or more events are detected by the hardware. Each event is independently maskable by setting the [DISPC\\_IRQENABLE](#) register.

To check when a particular interrupt event occurs and to reset a particular event, the [DISPC\\_IRQSTATUS](#) register must be accessed. This register regroups all the status of the module internal events that generate an interrupt (read 0: No interrupt occurred; read 1: Interrupt occurred; write 1: Status bit reset). See *Display Subsystem Register Manual*, for more information on checking and clearing interrupt events.

[Table 10-25](#) lists the DISPC interrupt events.

**Table 10-25. DISPC Interrupts**

Interrupt Name	Description
FRAMEDONE1_IRQ	Frame Done for LCD1 output: Active frame related to the LCD1 has completed and LCD1 output of the DISPC is disabled.
FRAMEDONE2_IRQ	Frame Done for LCD2 output: Active frame related to the LCD2 has completed and LCD2 output of the DISPC is disabled.
FRAMEDONETV_IRQ	Frame Done for TV output: Active frame related to the TV output has completed and TV output of the DISPC is disabled.

**Table 10-25. DISPC Interrupts (continued)**

Interrupt Name	Description
FRAMEDONEWB_IRQ	Frame Done for WB output: Active frame related to the WB has completed. First, it is used when the WB channel is connected to one of the pipelines to determine when the memory-to-memory transfer through DISPC is completed. Second, it is used when the WB channel is connected to one of the overlay output in nonreal-time mode to determine when the memory-to-memory transfer with overlay processing is completed.
VSYNC1_IRQ	VSYNC for primary LCD output: VSYNC interrupt for the primary LCD has occurred at the end of the frame.
VSYNC2_IRQ	VSYNC for secondary LCD output: VSYNC interrupt for the secondary LCD has occurred at the end of the frame.
EVSYNC_EVEN_IRQ	VSYNC for even field: EVSYNC_EVEN interrupt has occurred at the end of the frame (EVSYNC received and the field polarity is even) (VENC/HDMI)
EVSYNC_ODD_IRQ	VSYNC for odd field: EVSYNC_ODD interrupt has occurred at the end of the frame (EVSYNC received and the field polarity is odd) (VENC/HDMI)
ACBIASCOUNTSTATUS1_IRQ	AC Bias Count Status for LCD1 output: AC Bias transition counter has decremented to 0
ACBIASCOUNTSTATUS2_IRQ	AC Bias Count Status for LCD2 output: AC Bias transition counter has decremented to 0
PROGRAMMEDLINENUMBER_IRQ	Programmed Line Number: The primary LCD has reached the user programmed line number.
VID1ENDWINDOW_IRQ	End of the VID1 window: The DMA engine has fetched all the data from memory for the VID1 for the current frame.
VID2ENDWINDOW_IRQ	End of the VID2 window: The DMA engine has fetched all the data from memory for the VID2 for the current frame.
VID3ENDWINDOW_IRQ	End of the VID3 window: The DMA engine has fetched all the data from memory for the VID3 for the current frame.
GFXENDWINDOW_IRQ	End of the graphics window: The DMA engine has fetched all the data from memory for the graphics for the current frame.
VID1BUFFERUNDERFLOW_IRQ	VID1 DMA buffer underflow: The input VID1 DMA buffer goes underflow.
VID2BUFFERUNDERFLOW_IRQ	VID2 DMA buffer underflow: The input VID2 DMA buffer goes underflow.
VID3BUFFERUNDERFLOW_IRQ	VID3 DMA buffer underflow: The input VID3 DMA buffer goes underflow.
WBBUFFEROVERFLOW_IRQ	WB DMA buffer overflow: The output WB DMA buffer goes overflow. It cannot occur when WB channel is used in memory-to-memory transfer mode but only in capture mode. In capture mode the timings are defined by the timer associated with the output. In memory-to-memory mode, there is timing constraint.
GFXBUFFERUNDERFLOW_IRQ	GFX DMA buffer underflow: The input Graphics DMA buffer goes underflow.
PALETTEGAMMALOADING_IRQ	Palette/Gamma Table Loading: The palette/Gamma table in the graphics pipeline has been loaded using the DISPC DMA engine.
WBUNCOMPLETEERROR_IRQ	The write back buffer has been flushed before been fully drained.
OCPERROR_IRQ	OCP Error: L3 interconnect has sent SResp = ERR
SYNCLOST1_IRQ	Synchronization Lost on LCD1 output: Occurs when VSYNC width/front or back porches are not wide enough to load the pipelines with data (LCD output).
SYNCLOST2_IRQ	Synchronization Lost on LCD2 output: Occurs when VSYNC width/front or back porches are not wide enough to load the pipelines with data (LCD output).
SYNCLOSTTV_IRQ	Synchronization Lost on TV output: Occurs when porches are not wide enough to load the pipelines with data (TV output connected to the video NTSC/PAL encoder).

**Table 10-25. DISPC Interrupts (continued)**

Interrupt Name	Description
WAKEUP_IRQ	WakeUp: Occurs when the SWakeUp signal is asserted.

#### 10.2.4.5 sDMA Requests

The DMA synchronization line, DSS\_DISPC\_DMA, is connected to the sDMA. This DMA request is not a classical one but rather a synchronization signal between the DISPC and sDMA. The sDMA is informed that a programmable number of lines are output to the LCD, and that a system memory can be updated. This request is related to the interrupt event PROGRAMMEDLINENUMBER\_IRQ described in [Table 10-25](#). This allows the sDMA channel to be synchronized with the display subsystem internal DMA controller.

In other words, it allows to synchronize a memory-to-memory frame buffer update based on the scan line of the frame buffer in system memory (SDRAM or SRAM) by the DISPC. The DSS\_DISPC\_DMA DMA request is generated at a programmable line number defined in the [DISPC\\_LINE\\_NUMBER\[10:0\]](#) LINENUMBER bit field. This process allows an application to use a single frame buffer and update it after a certain number of lines has been read by the DISPC.

#### 10.2.4.6 DMA Engine

The DMA engine:

- Supplies data (encoded pixel data, palette, and gamma curve) from memories to GFX, VID1, VID2, and VID3 pipelines through the interconnect based on the configuration of the DISPC and pipelines settings.
- Stores (encoded pixel data) from GFX/VID pipelines or overlays to memories through the WB pipeline and interconnect based on the configuration of the the DISPC and WB pipeline settings.

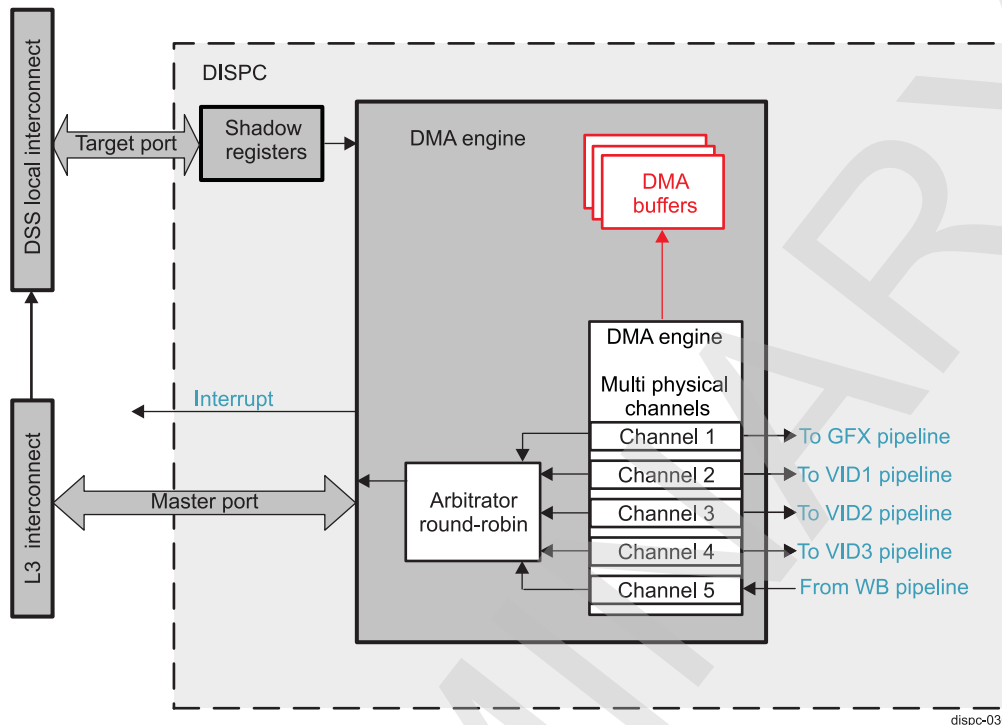
Each pipeline has a dedicated buffer and a channel with independant settings. The default size and allocation of the DMA buffer is detailed in [Table 10-26](#). Each DMA buffer is divided into two spaces, top and bottom buffer. Depending on the application, a DMA buffer space can be associated to a pipeline or merged with other spaces. The total number of spaces for each individual pipeline is from 0 (pipeline inactive) to number of pipelines x2 (in that case, all the DMA buffers are associated to a single pipeline). The sum of the number of spaces allocated for each pipeline must not be greater than the maximum number of spaces available. The correct number of spaces must be allocated to ensure no underflow. The spaces allocated to each pipeline must be greater than or equal to the minimum number of spaces required to support the throughput and the system latency. The space assignation are done in the [DISPC\\_GLOBAL\\_BUFFER](#) register.

[Figure 10-39](#) shows an overview of the DMA engine.

**Table 10-26. DMA Buffer Size**

Pipelines	DMA Buffer Size
GFX	2 lines x 640 x 128-bit
VID1	2 lines x 1024 x 128-bit
VID2	2 lines x 1024 x 128-bit
VID3	2 lines x 1024 x 128-bit
WB	2 lines x 1024 x 128-bit

Figure 10-39. DMA Engine Overview



#### 10.2.4.6.1 Addressing and Bursts

For each line to be fetched/stored, the DMA engine:

- Calculates the pixel address.
- Aligns the address.
- Defines the burst structure:
  - Type of the burst (1D or 2D burst structure).
  - Length of the burst.

The DMA engine generates scan addresses to read/write data from/to system memory. The base address defines the start address of the first pixel, then the address is incremented based on the number of pixels per line, offset between two consecutive lines and number of lines. The byte address of each pixel in the frame buffer in the system memory is determined by:

$$\text{Pixel address} = \text{Base address} + x \times \text{bpp} + (y \times ((\text{width} \times \text{bpp}) + \text{increment}))$$

where:

- Base address corresponds to the base address (for YUV-NV12 format) defined by:
  - DISPC\_GFX\_BA\_0[31:0], BA bit field
  - DISPC\_GFX\_BA\_1[31:0], BA bit field
  - DISPC\_VIDp\_BA\_0[31:0], BA bit field
  - DISPC\_VIDp\_BA\_1[31:0], BA bit field
  - DISPC\_VIDp\_BA\_UV\_0[31:0], BA bit field
  - DISPC\_VIDp\_BA\_UV\_1[31:0], BA bit field
- bpp corresponds to the number of bits per pixel defined by the DISPC\_GFX\_ATTRIBUTES[4:1] FORMAT bit field or DISPC\_VIDp\_ATTRIBUTES[4:1] FORMAT bit field.
- width corresponds to the number of pixels per line defined by DISPC\_GFX\_SIZE[10:0] SIZE+1 bit field or DISPC\_VIDp\_SIZE[10:0] SIZE+1 bit field.
- increment corresponds to the number of bytes to skip between two contiguous lines defined by DISPC\_GFX\_ROW\_INC[31:0] ROWINC -1 bit field or DISPC\_VIDp\_ROW\_INC[31:0] ROWINC -1 bit

field.

- x corresponds to the pixel position on the x-axis.
- y corresponds to the pixel position on the y-axis.

**NOTE:** For YUV format, the pixel values are defined in two separated buffers (Y and UV). The Y buffers base address are defined in the DISPC\_VIDp\_BA\_j[31:0] BA bit field. The UV buffers base address are defined in the DISPC\_VIDp\_BA\_UV\_j[31:0] BA bit field.

Table 10-27 resumes the registers settings for a simple access of a picture in system memory.

**Table 10-27. Register Settings for Accessing Image in Internal Memory**

Registers	Value
DISPC_GFX_BA_j/DISPC_VIDp_BA_j/DISPC_WB_BA_j	PBA, the physical base address of image in the memory
DISPC_VIDp_BA_UV_j/DISPC_WB_BA_UV_j	PBA, the physical base address of UV buffers image in the memory
DISPC_GFX_PIXEL_INC /DISPC_VIDp_PIXEL_INC/DISPC_WB_PIXEL_INC	1 or other in pixel incremental value
DISPC_GFX_ROW_INC /DISPC_VIDp_ROW_INC/DISPC_WB_ROW_INC	1 or other in row incremental value

An interconnect request, 128-bits, corresponds to one or several pixels depending on bits per pixels. Therefore the DMA engine determines the appropriate burst sequence to optimize each new line fetching/storing. The DMA engine shall avoid that a single burst cross two lines. The DMA engine supports burst of 1x128-bit, 2x128-bit, 4x128-bit and 8x128-bit. The default burst size at reset time is 8x128-bit. The maximum burst size can be configured for each pipeline by setting BURSTSIZE field DISPC\_GFX\_ATTRIBUTES[7:6] or DISPC\_VIDp\_ATTRIBUTES[7:6]. Since the burst size has to be aligned to the burst boundary, in case of misalignment, the DMA engine may issue single and/or smaller burst requests. Two type of burst are present:

- 1D-burst: used if the fetch/storage is linear in memory through the TILER. There is no rotation of the frame buffer. The frame buffer is not tiled.
- 2D-burst: used if the frame buffer is tiled.

Even if the DISPC\_VIDp\_ROW\_INC is not equal to 1, the 2D burst can be selected by the user. It is used when the display controller is configured to read one field of a frame by accessing only the even and odd lines.

**NOTE:** The burst size is initialized once at configuration and can be changed when the DISPC is disable.

#### 10.2.4.6.2 DMA Buffers

##### 10.2.4.6.2.1 READ DMA Buffers (GFX and VID Pipelines)

When the Vertical Front Porch (VFP) period starts after the last Horizontal Front Porch (HFP) of the last line or the external VSYNC is received, the DMA buffers are flushed according to the selected output associated with the pipeline. The DMA engine restarts fetching data from the memory through the L3 Interconnect. Enabling/disabling the display controller flushes the DMA buffers (except WB DMA buffers).

Programmable high and low thresholds, independent for each DMA buffer, are used by the DMA engine to start and stop requesting data to the L3 interconnect.

- When low threshold, set in BUFLOWTHRESHOLD field DISPC\_GFX\_BUF\_THRESHOLD[15:0], or DISPC\_VIDp\_BUF\_THRESHOLD[15:0], is reached the DMA engines start request on the L3 interconnect to fill the DMA buffer.
- When high threshold, set in BUFHIGHTHRESHOLD field DISPC\_GFX\_BUF\_THRESHOLD[31:16], or DISPC\_VIDp\_BUF\_THRESHOLD[31:16] is reached, the DMA engine stops requesting encoded pixel.

In order to avoid underflow at the beginning of a frame and have sufficient encoded pixel data to start some processing, a preloading of the DMA buffer is configurable between a fixed value of bytes or the high threshold value. The preload guarantee a minimum number of pixel present in the buffer. When the preload value is reached the associated channel shall start pulling pixel out of the DMA buffer. To enable the preload based on the value enter in PRELOAD field `DISPC_GFX_PRELOAD[11:0]` or `DISPC_VIDp_PRELOAD[11:0]`, the `BUFPRELOAD` bit `DISPC_GFX_BUF_THRESHOLD[11]`, or `DISPC_VIDp_BUF_THRESHOLD[19]` must be set to 0x0.

---

**NOTE:** When the self-refresh mode is selected, meaning the data in the DMA buffers are used for multiple frames, and at the end of each frame, the DMA buffers are not flushed.

---

#### 10.2.4.6.2.2 WRITE DMA Buffer (WB Pipeline)

Two modes are supported by the WB channel, selectable through `DISPC_WB_ATTRIBUTES[19]` `WRITEBACKMODE` bit:

- Capture mode, `WRITEBACKMODE` bit set to 0: one of the overlay output going to LCD or TV outputs is captured and at the same time the data are sent on the output. The WB timings are controlled by the LCD or TV timings.
- Memory-to-memory mode, `WRITEBACKMODE` bit set to 1: one of the overlay output or one of the pipelines is captured in order to perform memory-to-memory transfer with some processing by the DISPC (rescaling, overlaying, color space conversion,...)

**In capture mode:** The WB DMA buffers are flushed when the VFP period starts after the HFP following the last line, or when the external VSYNC is received depending on which output (LCD/TV) the WB pipeline is capturing data, if the programmed value in `DISPC_WB_ATTRIBUTES2[7:0]` `WBDELAYCOUNT` is set to zero. If the programmed value in `DISPC_WB_ATTRIBUTES2[7:0]` `WBDELAYCOUNT` is set to N (1:255), the write buffers DMA are flushed N lines later. The DMA engine starts storing data to memory through the L3-Based Interconnect as soon as enough data is available for the programmed burst size. When enabling/disabling the DISPC, the DMA buffers are flushed. The programmable thresholds low and high are used by the DMA engine to start and stop sending data to the L3 interconnect.

---

**NOTE:** If `DISPC_WB_ATTRIBUTES2[7:0]` `WBDELAYCOUNT` bit is set with zero, the WB is re-initialized just at the end of the last line of a frame at the beginning of the VFP signal. In order to let the WB completing data write to the external memory, the highest possible value compatible with the vertical blanking period shall be set.

---

**In memory-to-memory mode:** The WB pipeline is not synchronized to any internal or external timing generator. The WB pipeline is storing the output of one of the overlay outputs or one of the pipelines. When enabling/disabling the display controller, the DMA buffers are flushed. The programmable thresholds low and high are used by the DMA engine to start and stop sending data to the L3 interconnect.

Programmable high and low thresholds are used by the DMA engine to start and stop sending data to the L3 interconnect.

- When high threshold, set in `BUFHIGHTHRESHOLD` field `DISPC_WB_BUF_THRESHOLD[31:16]` is reached the DMA engines start sending data on the L3 interconnect to empty buffer.
- When low threshold, set in `BUFLOWTHRESHOLD` field `DISPC_WB_BUF_THRESHOLD[15:0]` is reached, the DMA engine stops sending encoded pixel.

At the end of the frame in order to drain the DMA buffer, some smaller burst (even single requests) shall be issued in order to completely empty the DMA buffer. To limit the number of interconnect requests from DISPC, a number of IDLE cycles between requests can be inserted. Idles can only be inserted when WB is used in memory-to-memory mode. It is ignored when WB is in capture mode.

The number of idle between requests can be activated and determined by :

- Setting to 0x0 `IDLESIZE` bit `DISPC_WB_ATTRIBUTES[27]` (default value) and entering the number of idle between request in `IDLENUMBER` field `DISPC_WB_ATTRIBUTES[31:28]`. Idle numbers vary from 0 to 15.

- Setting to 0x1 IDLESIZE bit `DISPC_WB_ATTRIBUTES[27]` which will take in consideration the size of the burst , BURSTSIZE field `DISPC_WB_ATTRIBUTES[15:14]`, to determine the number of idle.
  - if BURSTSIZE = 0x0 then idle equals IDLENUMBER (0 to 15).
  - if BURSTSIZE = 0x1 then idle equals IDLENUMBER \* 4 (0 to 60).
  - if BURSTSIZE = 0x2 then idle equals IDLENUMBER \* 8 (0 to 120).

#### 10.2.4.6.3 Predecimation

The predecimation process consist of downscaling an image by fetching only the necessary pixel in the memory. Vertical and horizontal pre decimation are possible:

- Vertical predecimation: The picture store in memory can be predecimated vertically by skipping lines. The burst mode is used to fetch the data when skipping line. Only the lines that will be used by the DISPC are fetched from memory, the other lines are skipped. The DMA engine sent requests for only the useful lines using 1D burst or 2D burst depending on the setting. The base address indicates the first valid pixel to fetch from memory. The number of lines to skip is set in ROWINC field `DISPC_GFX_ROW_INC[31:0]` or `DISPC_VIDp_ROW_INC[31:0]`
- Horizontal predecimation: When fetching data from memory, it is possible to skip one out of two pixels, up to one out of 16 pixels by setting PIXELINC field `DISPC_GFX_PIXEL_INC[7:0]` or `DISPC_VIDp_PIXEL_INC[7:0]` . The number of pixels skipped is determine by Size of a pixel in byte \* `PIXELINC +1`.

---

**NOTE:** The restriction to horizontal predecimation is that there is at least one useful pixel per 128-bit request. In that case, the DMA engine uses burst mode instead of singles to optimize, in term of latency and SDRAM efficiency, the requests.

No decimation is supported when input format is 1,2,4,8 BITMAP

---

#### 10.2.4.6.4 Arbitration

The requests (reads or writes) sent to the L3 interconnect are pipelined and are arbitrated on round-robin scheme. The default arbitration scheme shall be modified by setting the priority attribute of each pipeline as defined in ARBITRATION bit `DISPC_GFX_ATTRIBUTES[14]`, `DISPC_VIDp_ATTRIBUTES[14]` or `DISPC_WB_ATTRIBUTES[14]`.

By default all pipelines have the same priority, normal, meaning all pipelines requests are treated in A round robin order manner. If one or more pipelines require a higher number of requests going to the L3 interconnect, its priority can be moved up to a high priority. In this case, the high priority pipeline will be granted access first before any pipeline in the normal priority. If more than one active pipeline are in the high priority, the behavior is the same as all active pipelines in the normal priority. Normal active pipelines are not treated until all high active pipeline are finished. The ARBITRATION bit can not be modified during the whole frame.

This functionality balances each pipelines bandwidth depending on its constraint. It can be used to give higher priority to the pipelines with real time constraint vs. non real time pipelines. For e.g. pipelines associated to the LCD output in STALL mode shall have lower priority than pipelines associated to TV output.

#### 10.2.4.6.5 DMA Power Modes

##### 10.2.4.6.5.1 Low-Power Mode

Each DMA buffer is divided into two spaces. Each space can be associated to the pipeline or merged with other spaces. The total number of spaces for each individual pipeline is from 0 (pipeline inactive) to number of pipelines x 2 (in that case all the DMA buffers are associated to a single pipeline). The sum of the number of spaces allocated for each pipeline shall not be greater than the maximum available. Correct number of spaces shall be allocated to guarantee no underflow. The spaces allocated to each pipeline shall be greater or equal to the minimum required to support the through-put and the system latency.



---

**NOTE:** When the number of spaces is changed, the thresholds shall be reprogrammed to reflect the new DMA buffer configuration.

---

#### 10.2.4.6.5.2 *Ultralow-Power Mode*

In low-power mode, the L3 interconnect is used to fill-up the DMA buffers in order to store all the data required to display a full frame. Then the L3 interconnect is not used to fetch the new pixels for the following frames. The data in the DMA buffer are reused to display on the screen.

The setting of the mode is independent for each pipeline. One pipeline may have all the frame pixels into its DMA buffer and the others may have to refill their respective DMA buffers along the display scan since the frame buffer is too big to be stored in the DMA buffer.

The DMA buffers can be merged in order to optimize the L3 interconnect OFF time. The merge of the DMA buffers into a single one can be used at the same time to improve the Ultralow-Power mode, refer to [Section 10.2.4.6.5.1](#).

During the time where the frames is fetched in the internal DMA buffer, MStandby shall be asserted if MIDDLEMODE field [DISPC\\_SYSCONFIG\[13:12\]](#) is set to 0x2 (Smart-Standby).

Two ultra low power can be entered manually or automatically:

- Self-refresh mode: The starting of self-refresh mode is done manually by setting to 0x1 SELFREFRESH bit [DISPC\\_GFX\\_ATTRIBUTES\[15\]](#) or [DISPC\\_VIDp\\_ATTRIBUTES\[15\]](#) after capturing a frame in the DMA buffers. On the opposite, the stop selfrefresh is done by setting to 0x0 SELFREFRESH.
- Automatic self-refresh mode: By setting to 0x1 SELFREFRESHAUTO bit [DISPC\\_GFX\\_ATTRIBUTES\[17\]](#) or [DISPC\\_VIDp\\_ATTRIBUTES\[17\]](#), the transition from OFF to ON self-refresh mode is done by hardware after capturing the first frame. The hardware will reflected the status of the self-refresh mode by setting to 0x1 SELFREFRESH bit meaning that the data are read inside the DMA buffer without accessing the interconnect and system memory during the frame. Setting to 0x0 SELFREFRESH bit will update the DMA buffer.

---

**NOTE:** The WB pipeline does not support Ultralow-power mode.

---

#### 10.2.4.7 *Rotation and Mirroring*

The DISPC provides flexible mechanisms for an efficient implementation of rotation using the DISPC, its DMA engine, and the rotation engine of the TILER module. The rotation is handled only through the TILER module, which supplies the encoded pixels to the DISPC.

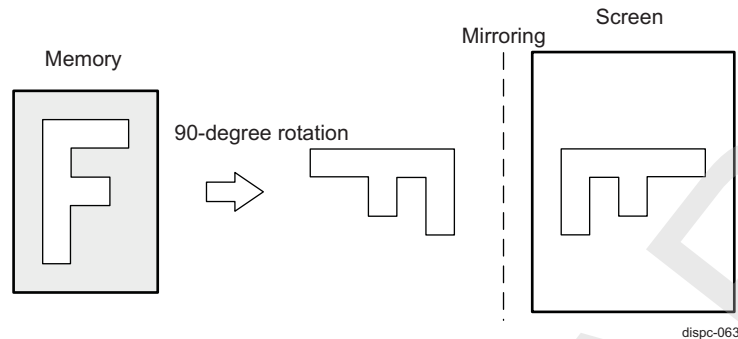
---

**NOTE:** No rotation or mirroring is supported when accessing directly internal SDRAM.

---

The TILER supports:

- Rotation 0, 90, 180, 270-degree views
- Mirroring
- Any combination of the 2 previous (e.g. Rot-90 + Mirroring, etc.)

**Figure 10-40. 90° Rotation with Mirroring**

When accessing YUV4:2:2 data, each 32-bit value loaded into the DMA buffer can represent either:

- 2 consecutive pixels on the same line (for instance, 0-/180-degree rotation)
- 2 pixels adjacent vertically (for instance, 90-/270-degree rotation)

The reading from the DMA buffer supports the extraction of the 2 pixels regardless of the rotation. When the pixels are not consecutive on the same line (90-/270-degree rotation), the chrominance sample of the first pixel of each 32-bit value is duplicated for the second pixel in the same 32-bit value.

The rotation flag `DISPC_VIDn_ATTRIBUTES[13:12] ROTATION` and the `DISPC_VIDn_ATTRIBUTES[4:1] FORMAT` bit fields define the processing to extract the pixels from YUV4:2:2 32-bit values. Software must ensure that the `ROTATION` and `FORMAT` bit field settings are coherent with the rotation/mirroring defined through the address format in the TILER-specific address map. For more information, see [Section 15.2.3.6.2, TILER Macro-Architecture](#), in [Section 15.2, Dynamic Memory Manager](#)

[Table 10-28](#) details the registers setting of the DISPC when accessing, rotating, mirroring an image via the TILER. A Physical Base Address (PBA) for each rotation is determined and set as the buffer address (BA). The row incremental is determined and set in `ROW_INC`. The pixel increment, `PIXEL_INC`, values is set to 0x1(contiguous pixels).

**Table 10-28. Register settings for rotation using TILER**

Rotation	Registers	Rotation with and without mirroring
0 degree	<a href="#">DISPC_GFX_BA_j</a> / <a href="#">DISPC_VIDp_BA_j</a> / <a href="#">DISPC_WB_BA_j</a> <a href="#">DISPC_GFX_PIXEL_INC</a> / <a href="#">DISPC_VIDp_PIXEL_INC</a> / <a href="#">DISPC_WB_PIXEL_INC</a> <a href="#">DISPC_GFX_ROW_INC</a> / <a href="#">DISPC_VIDp_ROW_INC</a> / <a href="#">DISPC_WB_ROW_INC</a>	PBA0  1 to 16  ROW0
90 degrees	<a href="#">DISPC_GFX_BA_j</a> / <a href="#">DISPC_VIDp_BA_j</a> / <a href="#">DISPC_WB_BA_j</a> <a href="#">DISPC_GFX_PIXEL_INC</a> / / <a href="#">DISPC_VIDp_PIXEL_INC</a> / <a href="#">DISPC_WB_PIXEL_INC</a> <a href="#">DISPC_GFX_ROW_INC</a> / / <a href="#">DISPC_VIDp_ROW_INC</a> / <a href="#">DISPC_WB_ROW_INC</a>	PBA90  1 to 16  ROW90
180 degrees	<a href="#">DISPC_GFX_BA_j</a> / <a href="#">DISPC_VIDp_BA_j</a> / <a href="#">DISPC_WB_BA_j</a> <a href="#">DISPC_GFX_PIXEL_INC</a> / <a href="#">DISPC_VIDp_PIXEL_INC</a> / <a href="#">DISPC_WB_PIXEL_INC</a> <a href="#">DISPC_GFX_ROW_INC</a> / / <a href="#">DISPC_VIDp_ROW_INC</a> / <a href="#">DISPC_WB_ROW_INC</a>	PBA180  1 to 16  ROW180
270 degrees	<a href="#">DISPC_GFX_BA_j</a> / <a href="#">DISPC_VIDp_BA_j</a> / <a href="#">DISPC_WB_BA_j</a> <a href="#">DISPC_GFX_PIXEL_INC</a> / <a href="#">DISPC_VIDp_PIXEL_INC</a> / <a href="#">DISPC_WB_PIXEL_INC</a> <a href="#">DISPC_GFX_ROW_INC</a> / / <a href="#">DISPC_VIDp_ROW_INC</a> / <a href="#">DISPC_WB_ROW_INC</a>	PBA270  1 to 16  ROW270

**NOTE:**

- For YUV format, in addition to DISPC\_VIDp\_BA\_j register used for the Y component, the register DISPC\_VIDp\_BA\_UV\_j shall be set in order to define the base address of the UV frame buffer in memory.

The PBA rotation is determined by:

- $PBA0 = PBA \mid (\text{mode} \ll 27) \mid (\text{orientation} \ll 29) \mid (1 \ll 32)$
- $PBA90 = PBA \mid (\text{mode} \ll 27) \mid (\text{orientation} \ll 29) \mid (1 \ll 32)$
- $PBA180 = PBA \mid (\text{mode} \ll 27) \mid (\text{orientation} \ll 29) \mid (1 \ll 32)$
- $PBA270 = PBA \mid (\text{mode} \ll 27) \mid (\text{orientation} \ll 29) \mid (1 \ll 32)$

Where PBA is the physical base address of image in the memory

The ROW rotation is determine by:

- If 8-bit per pixel
  - $ROW0 = 16384 - \text{width of the Video Picture in memory (in bytes)} + 1$
  - $ROW90 = 8192 - \text{width of the Video Picture in memory (in bytes)} + 1$
  - $ROW180 = 16384 - \text{width of the Video Picture in memory (in bytes)} + 1$
  - $ROW270 = 8192 - \text{width of the Video Picture in memory (in bytes)} + 1$
- If 16-bit per pixel
  - $ROW0 = 32768 - \text{width of the Video Picture in memory (in bytes)} + 1$
  - $ROW90 = 8192 - \text{width of the Video Picture in memory (in bytes)} + 1$
  - $ROW180 = 32768 - \text{width of the Video Picture in memory (in bytes)} + 1$
  - $ROW270 = 8192 - \text{width of the Video Picture in memory (in bytes)} + 1$
- If 32-bit per pixel
  - $ROW0 = 32768 - \text{width of the Video Picture in memory (in bytes)} + 1$
  - $ROW90 = 16384 - \text{width of the Video Picture in memory (in bytes)} + 1$
  - $ROW180 = 32768 - \text{width of the Video Picture in memory (in bytes)} + 1$
  - $ROW270 = 16384 - \text{width of the Video Picture in memory (in bytes)} + 1$

**Table 10-29. Mode Definition**

	8-bit Tiled	16-bit Tiled	32-bit Tiled	Page Tiled
Mode	0	1	2	3

**Table 10-30. Orientation Definition**

Type of Orientation	Value
0-view	0x0
180-view w/ mirroring	0x1
0-view w/ mirroring	0x2
180-view	0x3
270-view w/ mirroring	0x4
270-view	0x5
90-view	0x6
90-view w/ mirroring	0x7

**NOTE:** For YUV4:2:0 progressive pixel format, since the register DISPC\_VIDp\_ROW\_INC value is defined for the Y buffer, the bit-field DISPC\_VIDp\_ATTRIBUTES[22] DOUBLESTRIDE needs to be set to 1 when rotating the picture by 0- and 180-degree and needs to be reset to 0 when rotating the picture by 90- and 270-degree.

For YUV4:2:0 interlaced pixel format, since the register DISPC\_VIDp\_ROW\_INC value is defined for the Y buffer, the bit-field DISPC\_VIDp\_ATTRIBUTES[22] DOUBLESTRIDE needs to be set to 1 when rotating the picture by 0- and 180-degree. 90- and 270-degree are not supported with this pixel format.

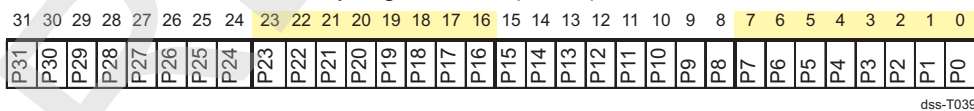
### 10.2.4.8 Memory Format

The graphic and video pipelines support various types of memory format. Table 10-31 list all supported format for each pipelines. The nibble mode is enable by setting to 0x1 NIBBLEMODE bit DISPC\_GFX\_ATTRIBUTES[9] and only applicable on BITMAP format for the GFX pipeline.

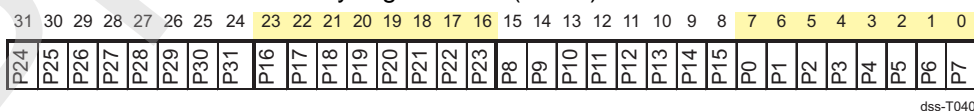
**Table 10-31. Memory Formats Supported**

Formats	GFX	VID1	VID2	VID3	WB
BITMAP 1-BPP	x				
BITMAP 2-BPP	x				
BITMAP 4-BPP	x				
BITMAP 8-BPP	x				
xRGB12-4444	x	x	x	x	x
RGBx12-4444	x	x	x	x	x
ARGB16-4444	x	x	x	x	x
RGBA16-4444	x	x	x	x	x
RGB16-565	x	x	x	x	x
xRGB16-1555	x	x	x	x	x
ARGB16-1555	x	x	x	x	x
xRGB24-8888	x	x	x	x	x
RGBx24-8888	x	x	x	x	
RGB24-888	x	x	x	x	x
ARGB32-8888	x	x	x	x	x
RGBA32-8888	x	x	x	x	x
UYUV 4:2:2		x	x	x	x
YUV2 4:2:2		x	x	x	x
YUV 4:2:0 - NV12		x	x	x	x

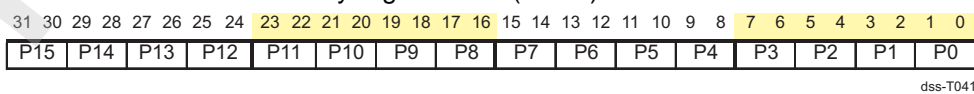
- BITMAP 1-BPP data memory organization (CLUT)



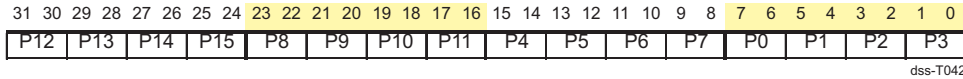
- BITMAP 1-BPP data memory organization (CLUT) in nibble mode.



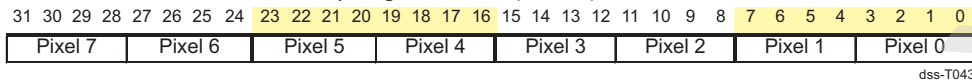
- BITMAP 2-BPP data memory organization (CLUT)



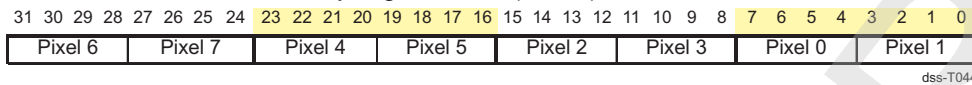
- BITMAP 2-BPP data memory organization (CLUT) in nibble mode.



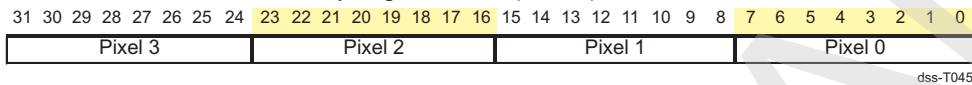
- BITMAP 4-BPP data memory organization (CLUT)



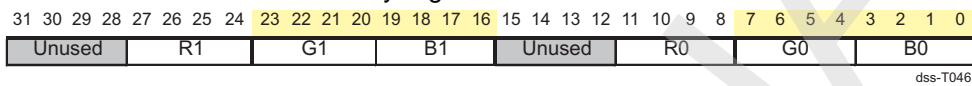
- BITMAP 4-BPP data memory organization (CLUT) in nibble mode.



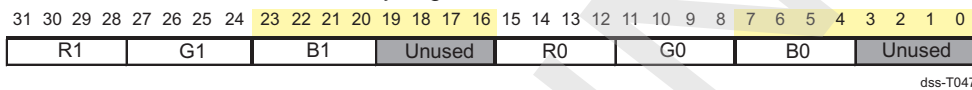
- BITMAP 8-BPP data memory organization (CLUT)



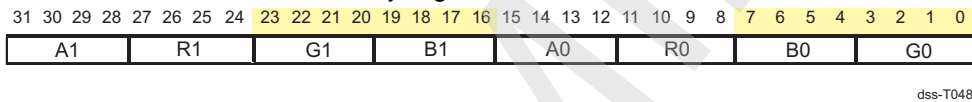
- xRGB12-4444 BPP data memory organization



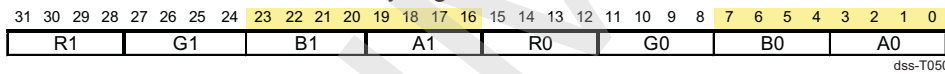
- RGBx12-4444 BPP data memory organization



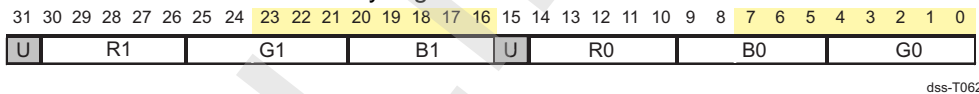
- ARGB16-4444 BPP data memory organization



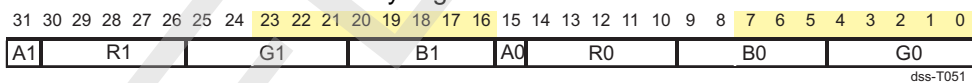
- RGBA16-4444 BPP data memory organization



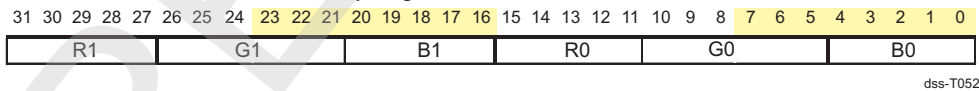
- xRGB16-555 BPP data memory organization



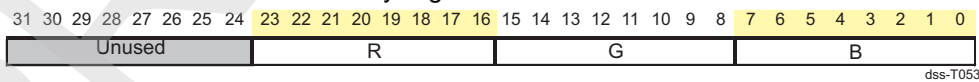
- ARGB16-1555 BPP data memory organization



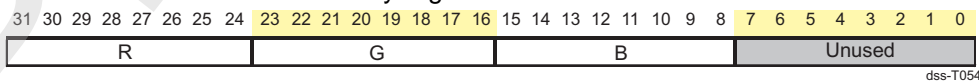
- RGB16-565 BPP data memory organization



- xRGB24-8888 BPP data memory organization



- RGBx24-8888 BPP data memory organization



- RGB24-888 BPP packed data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B1				R0				G0				B0				+0x0															
G2				B2				R1				G1				+0x4															
R3				G3				B3				R2				+0x8															

dss-T055

- ARGB32-8888 BPP data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
A								R								G								B								dss-T056

- RGBA32-8888 BPP data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R								G								B								A								dss-T057

- UYVY 4:2:2 data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y1				Cr0				Y0				Cb0				+0x0															
Y3				Cr2				Y2				Cb2				+0x4															
Y5				Cr4				Y4				Cb4				+0x8															
Y7				Cr6				Y6				Cb6				+0xC															

dss-T059

- YUV2 4:2:2 data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cr0				Y1				Cb0				Y0				+0x0															
Cr2				Y3				Cb2				Y2				+0x4															
Cr4				Y5				Cb4				Y4				+0x8															
Cr6				Y7				Cb6				Y6				+0xC															

dss-T060

- YUV4:2:0 - NV12 data memory organization

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y3				Y2				Y1				Y0				+0x0															
Y7				Y6				Y5				Y4				+0x4															
Cr1				Cb1				Cr0				Cb0				+0x0 + Offset															
Cr3				Cb3				Cr2				Cb2				+0x4 + Offset															
Cr5				Cb5				Cr4				Cb4				+0x8 + Offset															
Cr7				Cb7				Cr6				Cb6				+0xC + Offset															

dss-T061

### 10.2.4.9 Graphics Pipeline

The graphics pipeline is connected to the GFX FIFO for the input port and to the three overlay managers for the output or WB pipeline. The pixel output is directed to LCD, TV or WB path by setting CHANNELOUT bit [DISPC\\_GFX\\_ATTRIBUTES\[8\]](#) and CHANNELOUT2 field [DISPC\\_GFX\\_ATTRIBUTES\[31:30\]](#). [Table 10-50](#) resumes the bitfied settings to orient a pipeline to a LCD/TV or WB output. The default value direct GFX pipeline to LCD1.

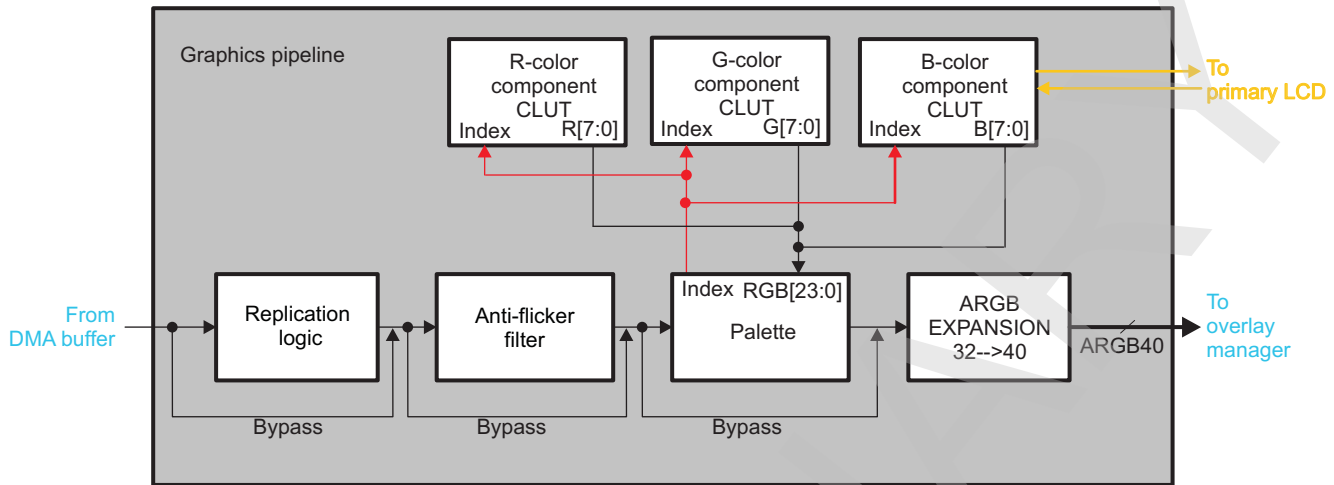
The pipeline consists of programmable replication logic, an antiflicker filter and one 256-entry Palette table. The replication logic is used to convert the RGB pixels, excluding the RGB24 format, into ARGB32 format based. The antiflicker filter processes the graphics data in RGB format in order to remove some of the vertical flicker. The 256-entry palette is used to convert bitmaps formats into RGB formats or for gamma corrections when in RGB format is inputted.

The inputs formats supported by graphic pipeline are listed in [Table 10-31](#).

The graphics pipeline is enable by setting ENABLE bit [DISPC\\_GFX\\_ATTRIBUTES\[0\]](#) to 0x1.

[Figure 10-41](#) represents the graphics pipeline.

Figure 10-41. Graphics Pipeline



dispc-001

10.2.4.9.1 Replication Logic

The replication logic increases the color depth of the graphics encoded pixels (from true color RGB 12-, and 16-BPP to 40-BPP).

- When the replication logic is enabled by setting REPLICATIONENABLE bit DISPC\_GFX\_ATTRIBUTES[5] to 0x1, the MSB bits are copied to the missing LSB. Table 10-34 indicates the remapping of the RGB pixels into ARGB 40-bit values.

Table 10-32. Replication Enabled: RGB Pixel Formats Remapping into ARGB40-10.10.10.10

Formats	A[7:0]	R[7:0]	G[7:0]	B[7:0]
	MSB LSB	MSB LSB	MS LSB	MSB LSB
xRGB12-4444	11111111	R[3:0]R[3:0]R[3:2]	G[3:0]G[3:0]G[3:2]	B[3:0]B[3:0]B[3:2]
RGBx12-4444	11111111	R[3:0]R[3:0]R[3:2]	G[3:0]G[3:0]G[3:2]	B[3:0]B[3:0]B[3:2]
RGB16-565	11111111	R[4:0]R[4:0]	G[5:0]G[5:2]	B[4:0]B[4:0]
xRGB16-1555	11111111	R[4:0]R[4:0]	G[4:0]G[4:0]	B[4:0]B[4:0]
ARGB16-1555	AAAAAAAA	R[4:0]R[4:0]	G[4:0]G[4:0]	B[4:0]B[4:0]
ARGB16-4444	A[3:0]A[3:0]A[3:2]	R[3:0]R[3:0]R[3:2]	G[3:0]G[3:0]G[3:2]	B[3:0]B[3:0]B[3:2]
RGBA16-4444	A[3:0]A[3:0]A[3:2]	R[3:0]R[3:0]R[3:2]	G[3:0]G[3:0]G[3:2]	B[3:0]B[3:0]B[3:2]

- When the replication logic is disabled by setting REPLICATIONENABLE bit DISPC\_GFX\_ATTRIBUTES[5] to 0x0, the encoded pixel values are shifted to the MSB boundary of the 24-bit format. The missing bit values are filled up with 0s. Table 10-35 indicates the remapping of the RGB pixels into ARGB 40-bit values.

Table 10-33. Replication Disabled: RGB Pixel Formats Remapping into ARGB40-10.10.10.10

Formats	A[7:0]	R[7:0]	G[7:0]	B[7:0]
	MSB LSB	MSB LSB	MSB LSB	MSB LSB
xRGB12-4444	1111111111	R[3:0]000000	G[3:0]000000	B[3:0]000000
RGBx12-4444	1111111111	R[3:0]000000	G[3:0]000000	B[3:0]000000
RGB16-565	1111111111	R[4:0]000000	G[5:0]0000	B[4:0]000000
xRGB16-1555	1111111111	R[4:0]000000	G[4:0]000000	B[4:0]000000
ARGB16-1555	AAAAAAAAAA	R[4:0]000000	G[4:0]000000	B[4:0]000000
ARGB16-4444	A[3:0]A[3:0]A[3:2]	R[3:0]000000	G[3:0]000000	B[3:0]000000
RGBA16-4444	A[3:0]A[3:0]A[3:2]	R[3:0]000000	G[3:0]000000	B[3:0]000000

### 10.2.4.9.2 Antiflicker Filter

The antiflicker filter processes the graphics data to remove some of the vertical flicker. It is based on 3-tap FIR filter with fixed coefficients. For each pixel to be output out of the graphics pipeline, the pixel above and pixel below the current line has to be read from the DMA graphics FIFO. Therefore, a 3 lines of pixels should be stored in the DMA graphics FIFO.

The antiflickering equation for A, R, G, and B components are:

$$A_{out}(x,y)=0.25*A_{in}(x,y-1) + 0.5*A_{in}(x,y) + 0.25*A_{in}(x,y+1)$$

$$R_{out}(x,y)=0.25*R_{in}(x,y-1) + 0.5*R_{in}(x,y) + 0.25*R_{in}(x,y+1)$$

$$G_{out}(x,y)=0.25*G_{in}(x,y-1) + 0.5*G_{in}(x,y) + 0.25*G_{in}(x,y+1)$$

$$B_{out}(x,y)=0.25*B_{in}(x,y-1) + 0.5*B_{in}(x,y) + 0.25*B_{in}(x,y+1)$$

For the first line of processing, since there is no pixel above, the value (x, y) is duplicated.

$$Out(x,y)=0.25*In(x,y) + 0.5*In(x,y) + 0.25*In(x,y+1)$$

For the last line of processing, since there is no pixel below, the value (x, y) is duplicated.

$$Out(x,y)=0.25*In(x,y-1) + 0.5*In(x,y) + 0.25*In(x,y)$$

---

**NOTE:** Anti flicker filtering is only supported in RGB formats and not on BITMAP formats.

Antiflickering is not supported for picture with less than 2 lines. In this case user shall disable antiflickering processing.

---

### 10.2.4.9.3 Color Look-Up Table (CLUT)

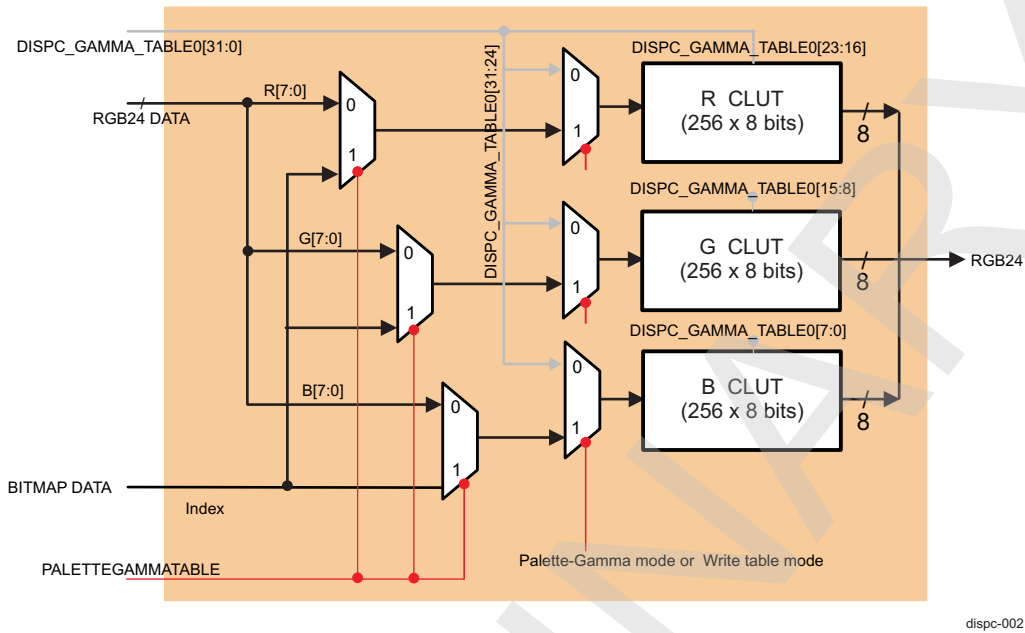
The graphics pipeline supports conversion of the BITMAP formats into RGB24 formats via a palette table. To enable the palette mode, PALETTEGAMMATALE bit [DISPC\\_CONFIG1\[3\]](#) should be set to 0x0. [Figure 10-44](#) shows the internal architecture of the color look-up/gamma table.

The palette is split into three memories of 256-bit x 8-bit entries. For bitmap (CLUT) indexes, the same value (1-, 2-, 4-, or 8-BPP) indexes the three memories. The table can be reloaded every frame, once or never (at the beginning of the frame before fetching the pixels for the graphics). The table is loaded dynamically at the beginning of the frame by the DISPC DMA engine. The base address of the table buffer is set in TABLEBA field [DISPC\\_GFX\\_TABLE\\_BA\[31:0\]](#). It is possible to load the table for every frame or only for a specific frame by setting LOAD MODE bit field [DISPC\\_CONFIG1\[2:1\]](#).

The table can be loaded without fetching datas for the graphics pipeline. The DMA buffer associated to the graphics pipeline is used to fetch the table from memory. Regardless of the mode; color or monochrome; for each entry into the table, one 32-bit value is fetched from memory.



Figure 10-42. Palette/Gamma Correction Architecture



The palette mode uses the encoded pixel values from the input graphics FIFO as pointers to index the 24-bit-wide palette: 1-BPP pixels address 2 palette entries, 2-BPP pixels address 4 palette entries, 4-BPP pixels address 16 palette entries, and 8-BPP pixels address 256 palette entries. In color mode, the value within the palette is made up of three 8-bit fields, one for each color component: red, green, and blue. For color operation, an individual frame is limited to a selection of 256 colors (the number of palette entries). In monochrome mode, only one 8-bit value is present. 256 grayscales and 16,777,216 colors are obtained after passing through the palette.

Figure 10-43. Data Memory Organization: Color Mode

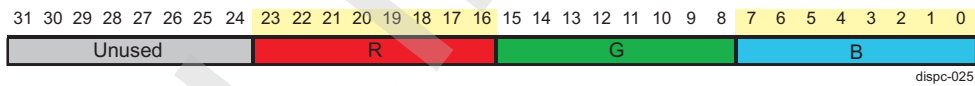
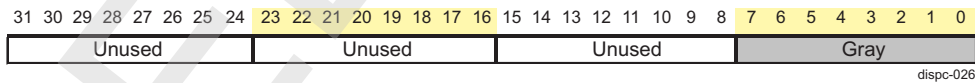


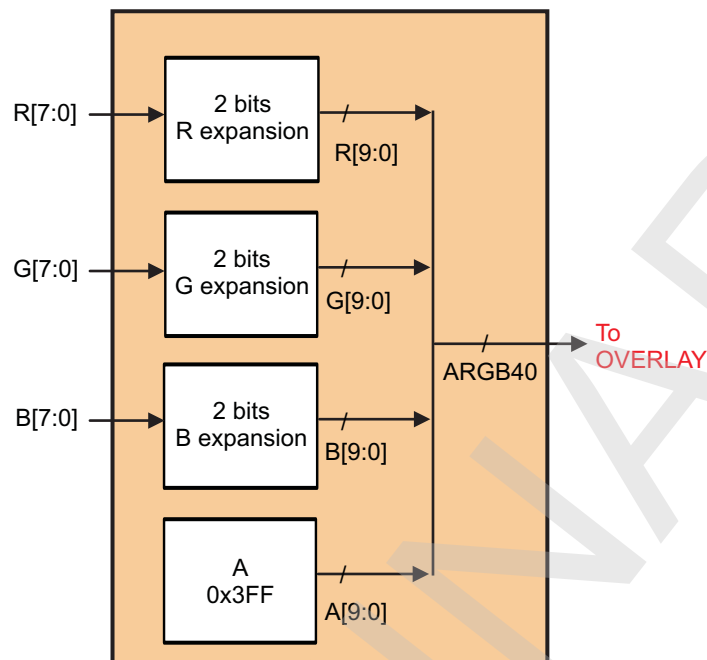
Figure 10-44. Data Memory Organization: Gray Mode



When a palette entry is selected by the encoded pixel value, both the content of the entry and the index are sent to the LCD1 overlay manager.

#### 10.2.4.9.4 Expansion to ARGB40

The expansion module takes in inputs the RGB24 pixel format outputted from the palette module. The module automatically replicates the 2 MSBs of each color component to create a ARGB40 pixel with A component set to 0x3FF.

**Figure 10-45. Expansion to ARGB40**

dispc-092

#### 10.2.4.10 Video Pipelines

Three identical video pipelines are available, VID1, VID and VID3. Each video pipeline is connected to its video FIFO controller for the input port and to the three overlay managers LCD 1, LCD2 and TV or WB pipeline. The pixel output is directed to LCD, TV or WB path by setting CHANNELOUT bit DISPC\_VIDp\_ATTRIBUTES[8] and CHANNELOUT2 field DISPC\_VIDp\_ATTRIBUTES[31:30]. [Table 10-50](#) resumes the bitfield settings to orient a pipeline to a LCD/TV or WB output. The default value direct all video pipelines to LCD1.

A video pipeline consists of a Scaler unit, a Color Space Conversion Unit, a VC-1 range mapping Unit and some programmable replication logic. The video pipeline unit order can be configured in two manners, see [Figure 10-46](#) and [Figure 10-47](#):

- Configuration 1: VC-1 range mapping Unit followed by Color Space conversion (CSC) Unit then Scaler Unit. The configuration is used to support RGB, ARGB, RGBA formats, YUV4:2:2 in backward mode for both data types. Each block can be independently bypassed.
- Configuration 2: VC-1 range mapping Unit followed by Scaler Unit then Color Space conversion Unit. The configuration is used to support RGB, ARGB, RGBA formats, YUV4:2:2 formats and YUV420-NV12 format taking advantage of the scaler in order to resample the chrominance using 5 taps horizontally and 3 or 5 taps vertically. Each block can be independently bypassed.

The DISPC\_VIDn\_ATTRIBUTES2[] YUVCHROMARESAMPLING bit controls the order of the scaler unit in the video pipeline:

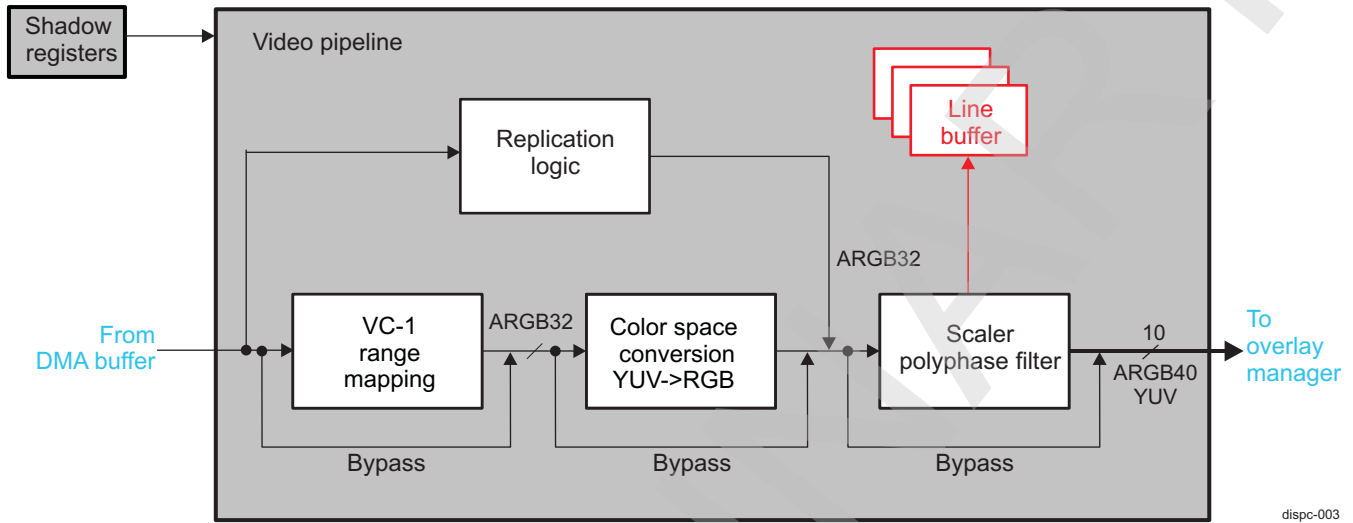
- When YUVCHROMARESAMPLING is set to 0x0, the video pipeline is in configuration 1, and the scaler is after the color space conversion unit. In case of YUV input data in 4:2:2 format, the chrominance resampling (4:2:2 to 4:4:4 format) is done by averaging the chrominance adjacent samples only for 0 degrees (zero rotation), because other rotation (90/180/270 degrees) is not supported in this mode. For more information about the supported chrominance resampling methods, see [Section 10.2.4.10.3.1, Chrominance Resampling](#).
- When YUVCHROMARESAMPLING is set to 0x1, the video pipeline is in configuration 2, and the scaler is before the color space conversion unit.
  - In case of YUV4:2:2 input data with 90-/270-degree rotation, the data is preprocessed by duplicating the missing chrominance samples.
  - In case of YUV4:2:2 input data with 0-/180-degree rotation, the chroma upsampling is performed in

the scaler unit.

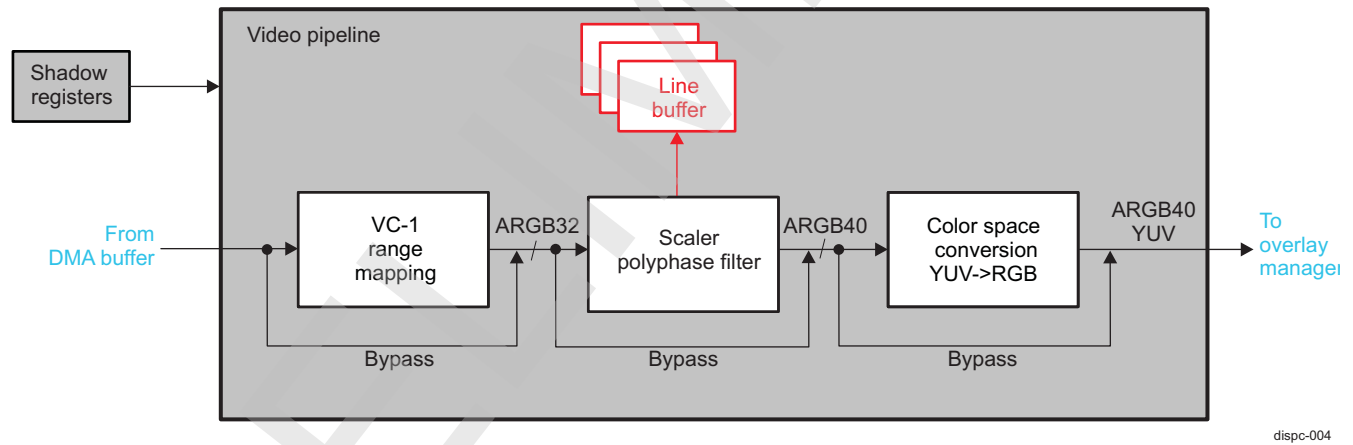
The inputs formats supported by video pipelines are listed in [Table 10-31](#).

The video pipeline is enable by setting ENABLE bit DISPC\_VIDp\_ATTRIBUTES[0] to 0x1.

**Figure 10-46. Configuration 1: Video Pipeline**



**Figure 10-47. Configuration 2: Video Pipeline**



**10.2.4.10.1 Replication Logic**

The replication logic increases the color depth of the video encoded pixels (from true color RGB 12-, and 16-BPP to 32-BPP) and only available in configuration 1.

The expansion from 8 to 10 bits color component is done by:

- The CSC unit, when the scaler is after the CSC module and when the vertical scaler is disabled.
- The vertical scaler, when the scaler is after the CSC unit and enabled.
- The replication bit module, when both the CSC unit and scaler are disabled.
- When the replication logic is enabled by setting REPLICATIONENABLE bit DISPC\_VIDp\_ATTRIBUTES[10] to 0x1, the MSB bits are copied to the LSB missing ones. [Table 10-34](#) indicates the remapping of the RGB pixels into ARGB 32-bit values.

**Table 10-34. Replication Enabled: RGB Pixel Formats Remapping into ARGB32-8888**

Formats	A[7:0]	R[7:0]	G[7:0]	B[7:0]
	MSB LSB	MSB LSB	MSB LSB	MSB LSB
xRGB12-4444	11111111	R[3:0]R[3:0]	G[3:0]G[3:0]	B[3:0]B[3:0]
RGBx12-4444	11111111	R[3:0]R[3:0]	G[3:0]G[3:0]	B[3:0]B[3:0]
RGB16-565	11111111	R[4:0]R[4:2]	G[5:0]G[5:4]	B[4:0]B[4:2]
xRGB16-1555	11111111	R[4:0]R[4:2]	G[4:0]G[4:2]	B[4:0]B[4:2]
ARGB16-1555	AAAAAAAA	R[4:0]R[4:2]	G[4:0]G[4:2]	B[4:0]B[4:2]
ARGB16-4444	A[3:0]A[3:0]	R[3:0]R[3:0]	G[3:0]G[3:0]	B[3:0]B[3:0]
RGBA16-4444	A[3:0]A[3:0]	R[3:0]R[3:0]	G[3:0]G[3:0]	B[3:0]B[3:0]

- When the replication logic is disabled by setting REPLICATIONENABLE bit DISPC\_VIDp\_ATTRIBUTES[10] to 0x0, the encoded pixel values are shifted to the MSB boundary of the 24-bit format. The missing bit values are filled up with 0s. Table 10-35 indicates the remapping of the RGB pixels into ARGB 32-bit values.

**Table 10-35. Replication Disabled: RGB Pixel Formats Remapping into ARGB32-8888**

Formats	A[7:0]	R[7:0]	G[7:0]	B[7:0]
	MSB LSB	MSB LSB	MSB LSB	MSB LSB
xRGB12-4444	11111111	R[3:0]0000	G[3:0]0000	B[3:0]0000
RGBx12-4444	11111111	R[3:0]0000	G[3:0]0000	B[3:0]0000
RGB16-565	11111111	R[4:0]000	G[5:0]00	B[4:0]000
xRGB16-1555	11111111	R[4:0]000	G[4:0]000	B[4:0]000
ARGB16-1555	AAAAAAAA	R[4:0]000	G[4:0]000	B[4:0]000
ARGB16-4444	A[3:0]A[3:0]	R[3:0]0000	G[3:0]0000	B[3:0]0000
RGBA16-4444	A[3:0]A[3:0]	R[3:0]0000	G[3:0]0000	B[3:0]0000

#### 10.2.4.10.2 VC-1 Range Mapping Unit

The VC-1 range mapping unit is used when the video frame picture is decoded using VC-1 codec by the video accelerator. It remaps the Y, Cb and Cr component. The unit can be used primarily for YUV 4:2:0-NV12 pixel format but also can be applied on YUV 4:2:2 pixel formats (YUV2 and UYVY).

The VC-1 range mapping Unit is enable by setting to 0x1 VC1ENABLE bit DISPC\_VIDp\_ATTRIBUTES2[0]. VC1\_RANGE\_Y and VC1\_RANGE\_CBCR fields DISPC\_VIDp\_ATTRIBUTES2[3:1][6:4] are two 3-bit values programmed by the user and independent for each video pipelines. The module is governed by the equations:

$$Y_{out} = \text{CLIP}(\left( (Y_{int} - 128) \times (VC1\_RANGE\_Y + 9) + 4 \right) / 8) + 128$$

$$C_b = \text{CLIP}(\left( (C_b - 128) \times (VC1\_RANGE\_CBCR + 9) + 4 \right) / 8) + 128$$

$$C_r = \text{CLIP}(\left( (C_r - 128) \times (VC1\_RANGE\_CBCR + 9) + 4 \right) / 8) + 128$$

**NOTE:** The input and output pixel values are unsigned (Y, Cr, and Cb).

The function CLIP () clips to 0 or 255 when min or max are reached respectively otherwise resulting output stay identical.

#### 10.2.4.10.3 Color Space Conversion Unit YUV to RGB

The color space conversion module converts the video encoded pixel values from YUV4:4:4 format into RGB24 or RGB30 format. The output format depends on the video pipeline configuration selected:

- Configuration 1: RGB24 output format, with 8-bit value per component: red, green, and blue.
- Configuration 2: RGB30 output format, with 10-bit value per component: red, green, and blue.

In case of YUV4:2:0 or YUV4:2:2 formats, a chrominance resampling to YUV4:4:4 is required before converting the YUV into RGB values, refer to [Section 10.2.4.10.3.1](#). YUV4:2:2 or YUV4:2:0 to YUV4:4:4 chrominance resampling are a preprocessing to the color space conversion.

[Figure 10-48](#) to [Figure 10-51](#) show the 3 × 3 11-bits coefficients used to convert from YUV4:4:4 into RGB24. The coefficients are set according to the standard used to encode the pixel data in YUV color space. [Table 10-36](#) resumes the coefficients with their respective bit fields.

**Table 10-36. Color Space Conversion YUV to RGB Bit Field Setting**

Coefficients	Bit Field Registers
R <sub>Y</sub>	RY field DISPC_VIDp_CONV_COEF0[10:0]
R <sub>Cr</sub>	RCR field DISPC_VIDp_CONV_COEF0[26:16]
R <sub>Cb</sub>	RCB field DISPC_VIDp_CONV_COEF1[10:0]
G <sub>Y</sub>	GY field DISPC_VIDp_CONV_COEF1[26:16]
G <sub>Cr</sub>	GCR field DISPC_VIDp_CONV_COEF2[10:0]
G <sub>Cb</sub>	GCB field DISPC_VIDp_CONV_COEF2[26:16]
B <sub>Y</sub>	BY field DISPC_VIDp_CONV_COEF3[10:0]
B <sub>Cr</sub>	BCR field DISPC_VIDp_CONV_COEF3[26:16]
B <sub>Cb</sub>	BCB field DISPC_VIDp_CONV_COEF4[10:0]

- For configuration 1 with a RGB24 output  
If the active range for the luminance samples (Y) is [16:235] and [16:240] for the chrominance samples (Cb and Cr), the range selection is done by setting to 0x0 FULLRANGE bit DISPC\_VIDp\_ATTRIBUTES[11]. The values of R, G, and B output components are clipped to the range [0:255].

**NOTE:** The scaling and Color Space Conversion clipping is set by the same bit, FULLRANGE bit DISPC\_VIDp\_ATTRIBUTES[11].

**Figure 10-48. YCbCr to RGB Registers (FULLRANGE=0), 8 Bits Outputs**

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{Cr} & R_{Cb} \\ G_Y & G_{Cr} & G_{Cb} \\ B_Y & B_{Cr} & B_{Cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} - 16 \\ Cr_{IN} - 128 \\ Cb_{IN} - 128 \end{bmatrix}$$

dispc-005

If the active range for the luminance samples (Y) and chrominance samples (Cb and Cr) is [0:255], the range selection is done by setting to 0x1 FULLRANGE bit DISPC\_VIDp\_ATTRIBUTES[11]. The values of R, G, and B output components are clipped to the range [0:255].

**Figure 10-49. YCbCr to RGB Registers (FULLRANGE=1), 8 Bits Outputs**

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{Cr} & R_{Cb} \\ G_Y & G_{Cr} & G_{Cb} \\ B_Y & B_{Cr} & B_{Cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} \\ Cr_{IN} - 128 \\ Cb_{IN} - 128 \end{bmatrix}$$

dispc-006

- For configuration 2 with a RGB30 output  
If the active range for the luminance samples (Y) is [64:940] and [64:960] for the chrominance samples (Cb and Cr), the range selection is done by setting to 0x0 FULLRANGE bit DISPC\_VIDp\_ATTRIBUTES[11]. The values of R, G, and B output components are clipped to the range [0:1023].

**NOTE:** The scaling and Color Space Conversion clipping is set by the same bit, FULLRANGE bit DISPC\_VIDp\_ATTRIBUTES[11].

**Figure 10-50. YCbCr to RGB Registers (FULLRANGE=0), 10 Bits Outputs**

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{Cr} & R_{Cb} \\ G_Y & G_{Cr} & G_{Cb} \\ B_Y & B_{Cr} & B_{Cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} - 64 \\ Cr_{IN} - 512 \\ Cb_{IN} - 512 \end{bmatrix}$$

dispc-074

If the active range for the luminance samples (Y) and chrominance samples (Cb and Cr) is [0:1023], the range selection is done by setting to 0x1 FULLRANGE bit DISPC\_VIDp\_ATTRIBUTES[11]. The values of R, G, and B output components are clipped to the range [0:1023].

**Figure 10-51. YCbCr to RGB Registers (FULLRANGE=1), 10 Bits Outputs**

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{Cr} & R_{Cb} \\ G_Y & G_{Cr} & G_{Cb} \\ B_Y & B_{Cr} & B_{Cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} \\ Cr_{IN} - 512 \\ Cb_{IN} - 512 \end{bmatrix}$$

dispc-075

### 10.2.4.10.3.1 Chrominance Resampling

Two methods are supported to resample chrominance:

- Averaging of the chrominance: done by software followed by hardware conversion when the video pipeline in configuration 1.
- Filtering of the chrominance using the scaler Unit (chrominance resampling and rescaling can be combined together in order to support native rescaling of YUV format) when the video pipeline is in configuration 2.

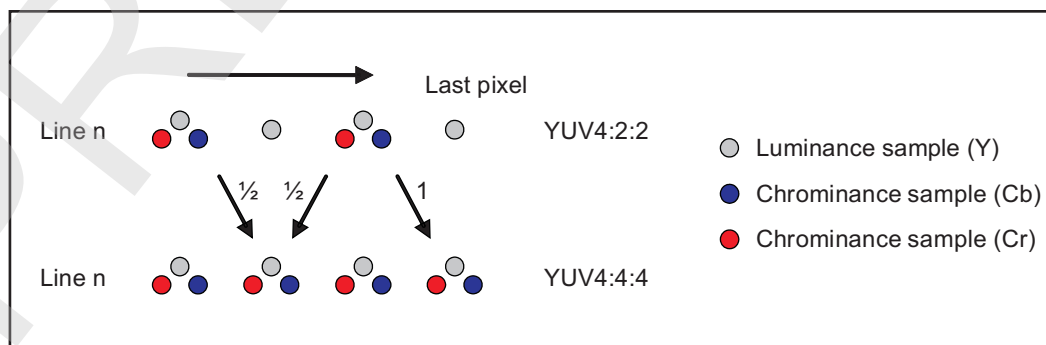
To convert the YUV4:2:2 encoded pixel values into YUV4:4:4 format, the averaging of the chrominance technique can be used as described in [Figure 10-52](#). The missing chrominance samples (Cb and Cr) are interpolated using the average values of the two closest values on the same line ( , ) or are repeated from the second pixel in the same 32-bit container, see [Figure 10-53](#). For the last pixel, the chrominance samples are duplicated using the values from the previous pixel otherwise the chrominance samples are averaged using the two adjacent values. [Figure 10-54](#) represents the flow of the pixel.

**Figure 10-52. Averaging of the Chrominance Formula**

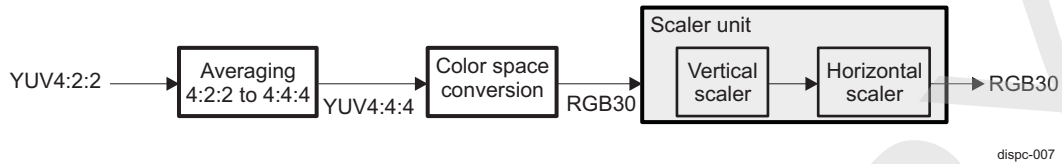
$$Cb_n(YCbCr\ 444) = \frac{Cb_{n-1}(YCbCr\ 422) + Cb_{n+1}(YCbCr\ 422)}{2} \text{ (n odd)}$$

$$Cr_n(YCbCr\ 444) = \frac{Cr_{n-1}(YCbCr\ 422) + Cr_{n+1}(YCbCr\ 422)}{2} \text{ (n odd)}$$

dispc-010

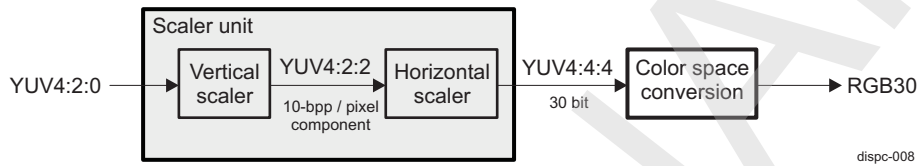
**Figure 10-53. Averaging of the Chrominance Representation**

**Figure 10-54. YUV4:2:2 to RGB30 Using Averaging of the Chrominance**

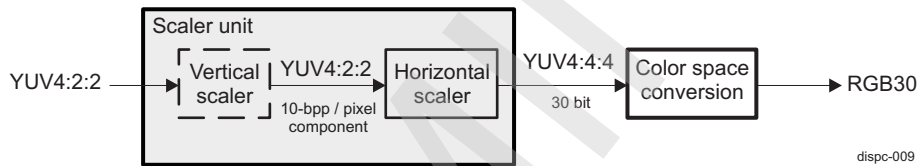


The scaler unit can be used to resample the chrominance of YUV4:2:2, YUV4:2:0 as shown in [Figure 10-55](#) and [Figure 10-56](#). The settings of the scaler unit to perform chrominance resampling is detailed in [Section 10.2.4.10.4](#).

**Figure 10-55. YUV4:2:0 to RGB30 Using Scaler Unit for Resampling Chrominance**



**Figure 10-56. YUV4:2:2 to RGB30 Using Scaler Unit for Resampling Chrominance**



**NOTE:** If rotation needs to be supported, YUV4:2:2 and YUV4:2:0 (0/180 degree rotation) chrominance resampling is done as shown on [Figure 10-56](#) and [Figure 10-55](#). For YUV4:2:2 (90/270 degree rotation) data are preprocessed to present YUV4:4:4 on the scaler input (duplication of the missing chroma), as shown on [Figure 10-54](#).

#### 10.2.4.10.4 Scaler Unit

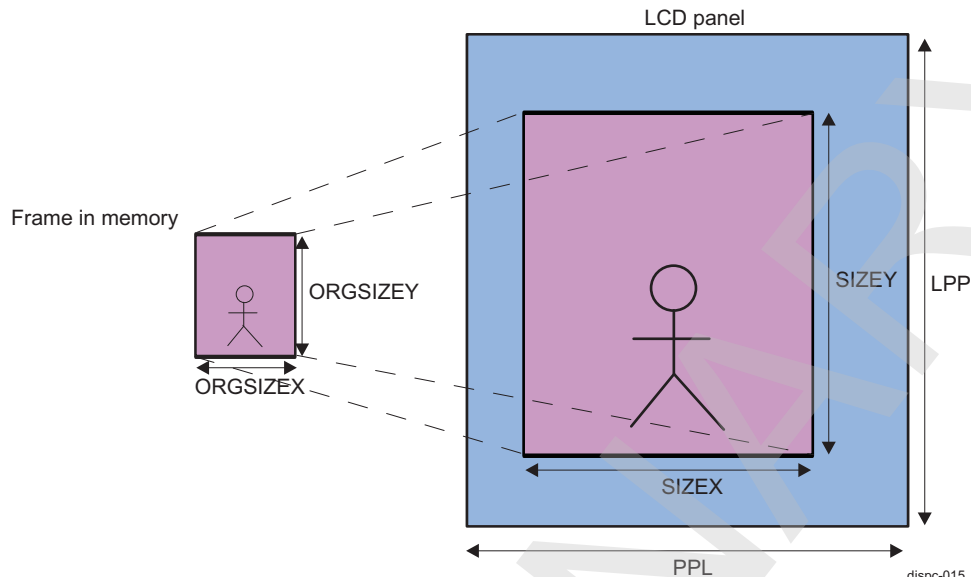
All the video formats are supported including the formats including alpha blending. The Alpha blending is scaled with the same parameters than RGB color components. For the YUV formats, the Y and Cb/Cr are processed independently. The filter is based on a Finite Impulse Response filter (FIR). The filtering can be used for different processing:

- Upsampling of the picture.
- Downsampling of the picture.
- Antiflicker reduction.
- De-interlacing using "bob" algorithm
- Chrominance resampling in case of YUV data formats

**NOTE:** User shall ensure the the resizing frame shall be displayed in the LCD/screen boundaries.

[Figure 10-57](#) shows an example of video upsampling.

Figure 10-57. Video Upsampling



The up/downsampling filter is a polyphase filter with 5 taps and 8 phases for the horizontal filter, and a programmable number of taps (3 or 5) and 8 phases for vertical filter. The input buffer has 5 input memory lines. The following limitations shall be considered:

- The upsampling ratio is up to x8.
- The downsampling ratio using 3-tap configuration is down to x0.5 for RGB format
- The downsampling ratio using 5-tap configuration is down to x0.25 for RGB format
- If the input format is changed from YUV4:2:2 to 4:2:0 (WB pipeline), the downsampling ratio is further reduced:
  - Using 5-tap configuration, the ratio is down to x0.5 for RGB format
  - Using 3-tap configuration, no down-scaling is available

For the vertical up/downsampling in 3-taps configuration, the equations are :

For RGB formats	For YUV formats
$A_{out}(n) = \left( \sum_{i=-1}^{i=1} C_{vi}(\phi) * A_{in}(n+i) \right) >> 5$	$Y_{out}(n) = \left( \sum_{i=-1}^{i=1} C_{vi}(\phi_y) * Y_{in}(n+i) \right) >> 5$
$R_{out}(n) = \left( \sum_{i=-1}^{i=1} C_{vi}(\phi) * R_{in}(n+i) \right) >> 5$	$Cr_{out}(n) = \left( \sum_{i=-1}^{i=1} C_{vci}(\phi_c) * Cb_{in}(n+i) \right) >> 5$
$G_{out}(n) = \left( \sum_{i=-1}^{i=1} C_{vi}(\phi) * G_{in}(n+i) \right) >> 5$	$Cb_{out}(n) = \left( \sum_{i=-1}^{i=1} C_{vci}(\phi_c) * Cr_{in}(n+i) \right) >> 5$
$B_{out}(n) = \left( \sum_{i=-1}^{i=1} C_{vi}(\phi) * B_{in}(n+i) \right) >> 5$	

dispc-013

(7)

For the vertical up/downsampling in 5-taps configuration, the equations are :



<p>For RGB formats</p> $A_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{vi}(\Phi) * A_{in}(n+i) \right) \gg 5$ $R_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{vi}(\Phi) * R_{in}(n+i) \right) \gg 5$ $G_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{vi}(\Phi) * G_{in}(n+i) \right) \gg 5$ $B_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{vi}(\Phi) * B_{in}(n+i) \right) \gg 5$	<p>For YUV formats</p> $Y_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{vi}(\Phi_y) * Y_{in}(n+i) \right) \gg 5$ $Cb_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{vci}(\Phi_c) * Cb_{in}(n+i) \right) \gg 5$ $Cr_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{vci}(\Phi_c) * Cr_{in}(n+i) \right) \gg 5$
--	---

dispc-012

(8)

For the horizontal up/downsampling in 5-taps configuration, the equations are :

<p>For RGB formats</p> $A_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{hi}(\Phi) * A_{in}(n+i) \right) \gg 7$ $R_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{hi}(\Phi) * R_{in}(n+i) \right) \gg 7$ $G_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{hi}(\Phi) * G_{in}(n+i) \right) \gg 7$ $B_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{hi}(\Phi) * B_{in}(n+i) \right) \gg 7$	<p>For YUV formats</p> $Y_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{hi}(\Phi_y) * Y_{in}(n+i) \right) \gg 7$ $Cb_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{hci}(\Phi_c) * Cb_{in}(n+i) \right) \gg 7$ $Cr_{out}(n) = \left( \sum_{i=-2}^{i=2} C_{hci}(\Phi_c) * Cr_{in}(n+i) \right) \gg 7$
--	---

dispc-014

(9)

**NOTE:** The pixel (n + 1) is the previous pixel in respect to pixel (n). The line (n + 1) is the previous line in respect to line (n).

The coefficients Ci() depend on the phase between input and output pixels.

**NOTE:** The coefficients are different for Y and Cr, Cb filtering since the calculations are independent due to the chrominance resampling for YUV4:2:2 and YUV4:2:0

The vertical filter is first applied to the encoded input pixel data and then the horizontal filter is applied on the resulting pixel values in order to generate the output pixel values. The vertical input of the filter consists of a 5 lines of 2048\*32-bits for both 3-taps and 5-taps configurations.

**Table 10-37. Line Buffer Width for Scaler Unit**

Vertical taps	Maximum Input Width (pixels)
3, 5	2048 x 32-bit

At the beginning of the frame scaling processing, the first line is duplicated to fill up the two first lines in 3-taps configuration and the three first lines in 5-taps configuration.

At the end of the frame processing, the last line is duplicated if the scaling logic requires loading more lines and the last line has been reached.

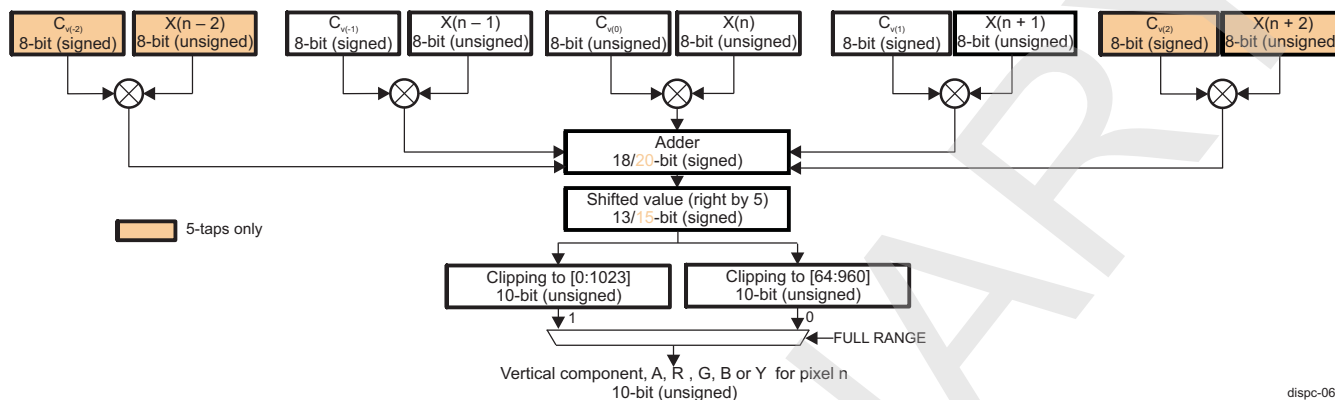
The programmable coefficients of the polyphase filters are signed 8-bit values (except for the central coefficient which is unsigned). The video scalars have an 8 bit input and a 10 bit output. The vertical scaling changes the 8 bit input into an 10 bit clipped outputs and the horizontal scaling takes the 10 bit inputs.

Figure 10-67 and Figure 10-68 details the scaler macro-architecture for the component A, R, G, B and Y. Figure 10-69 and Figure 10-70 details the scaler macro-architecture for component Cr and Cb.

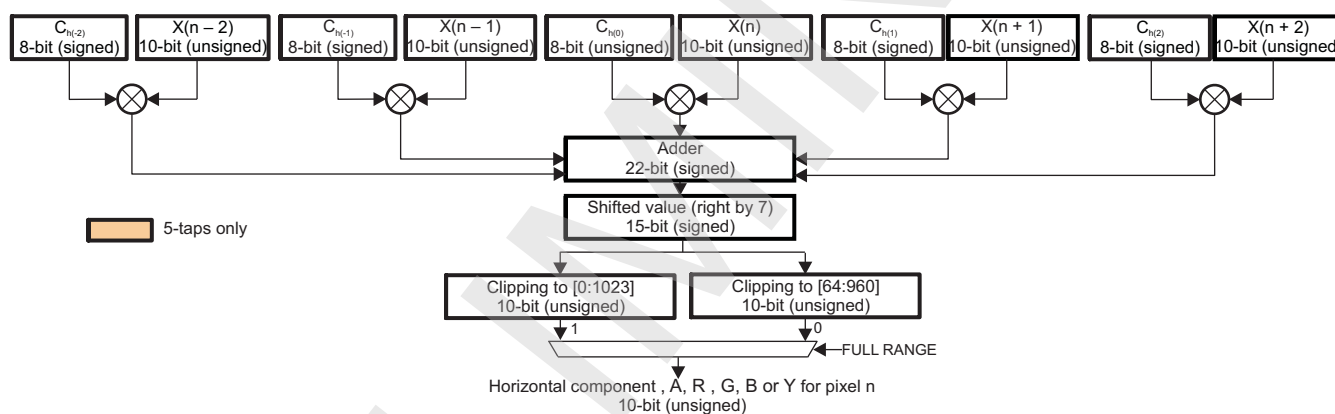
The scaling output can be clipped to an output range of [0:1023] or [64:960] by configuring FULLRANGE bit DISPC\_VIDp\_ATTRIBUTES[11].

**NOTE:** The scaling and Color Space Conversion clipping is set by the same bit, FULLRANGE bit DISPC\_VIDp\_ATTRIBUTES[11].

**Figure 10-58. Macro-architecture of the Vertical Scaling for A, R, G, B and Y Component**



**Figure 10-59. Macro-architecture of the Horizontal Scaling for A, R, G, B and Y Component**



**Figure 10-60. Macro-architecture of the Vertical Scaling for Cr and Cb Component**

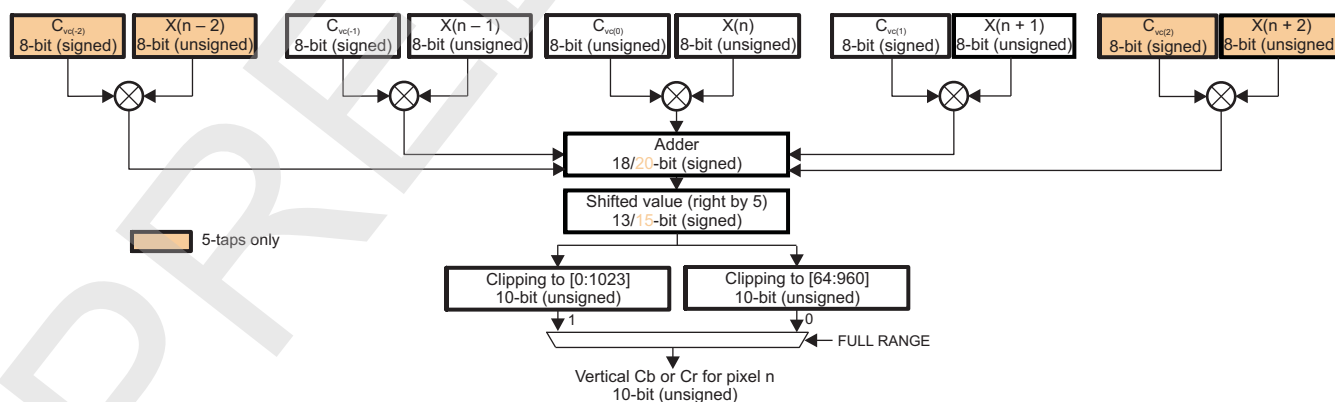
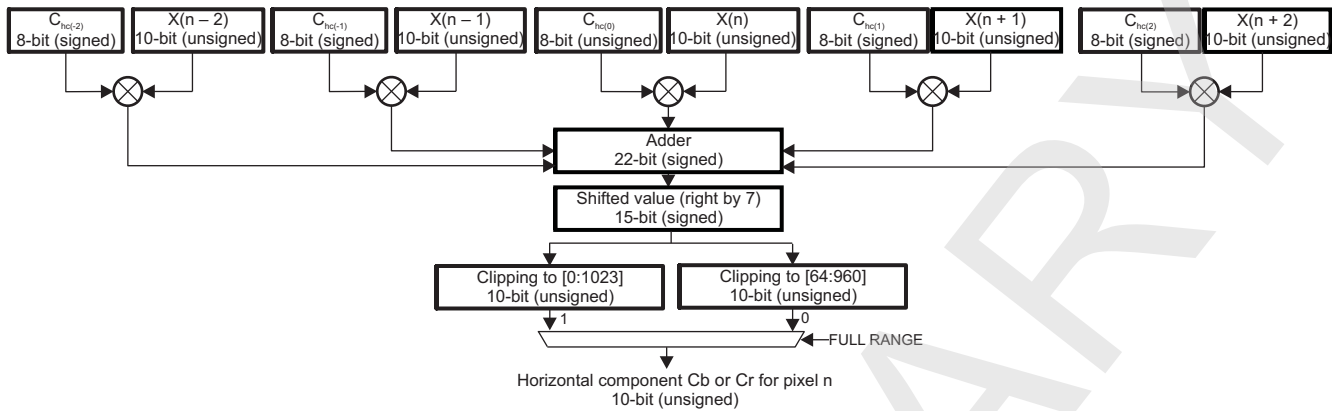


Figure 10-61. Macro-architecture of the Horizontal Scaling for Cr and Cb Component



dispc-072

Table 10-38 and Table 10-39 resume all the bit fields in function to set for each coefficients.

Table 10-38. Register Bifield Associated to Coefficient for ARGB and Y Configuration in VIDp Scaler

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Vertical	Cv(-2)	N/A	FIRVC22	DISPC_VIDp_FIR_COEF_V_i
	Cv(-1)		FIRVC2	DISPC_VIDp_FIR_COEF_HV_i
	Cv(0)		FIRVC1	DISPC_VIDp_FIR_COEF_HV_i
	Cv(1)		FIRVC0	DISPC_VIDp_FIR_COEF_HV_i
	Cv(2)		FIRVC00	DISPC_VIDp_FIR_COEF_V_i
Horizontal	Ch(-2)	N/A	FIRHC4	DISPC_VIDp_FIR_COEF_HV_i
	Ch(-1)		FIRHC3	DISPC_VIDp_FIR_COEF_H_i
	Ch(0)		FIRHC2	DISPC_VIDp_FIR_COEF_H_i
	Ch(1)		FIRHC1	DISPC_VIDp_FIR_COEF_H_i
	Ch(2)		FIRHC0	DISPC_VIDp_FIR_COEF_H_i

Table 10-39. Register Bifield Associated to Coefficient for Cb and Cr Configuration in VIDp Scaler

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Vertical	Cvc(-2)	N/A	FIRVC22	DISPC_VIDp_FIR_COEF_V2_i
	Cvc(-1)		FIRVC2	DISPC_VIDp_FIR_COEF_HV2_i
	Cvc(0)		FIRVC1	DISPC_VIDp_FIR_COEF_HV2_i
	Cvc(1)		FIRVC0	DISPC_VIDp_FIR_COEF_HV2_i
	Cvc(2)		FIRVC00	DISPC_VIDp_FIR_COEF_V2_i

**Table 10-39. Register Bitfield Associated to Coefficient for Cb and Cr Configuration in VIDp Scaler  
(continued)**

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Horizontal	Chc(-2)	N/A	FIRHC4	DISPC_VIDp_FIR_COE F_HV2_i
	Chc(-1)		FIRHC3	DISPC_VIDp_FIR_COE F_H2_i
	Chc(0)		FIRHC2	DISPC_VIDp_FIR_COE F_H2_i
	Chc(1)		FIRHC1	DISPC_VIDp_FIR_COE F_H2_i
	Chc(2)		FIRHC0	DISPC_VIDp_FIR_COE F_H2_i

The VID scaler unit vertical or/and horizontal sampling is defined by setting/resetting the DISPC\_VIDp\_ATTRIBUTES[5] RESIZEENABLE bit.

A set of configuration must be valid before enabling the video up/downsampling block.

The following fields define the configuration of the video up/downsampling block for VIDp:

- Vertical up/downsampling increment value DISPC\_VIDp\_FIR[28:16] FIRVINC bit field. The unsigned integer value range is [1:4096]. The software calculates the value using the following equation:

$$FIRVINC = 1024 * \left( \frac{ORG\text{SIZE}Y}{\text{SIZE}Y} \right)$$

dispc-066

(10)

**NOTE:**

- If the DISPC\_VIDp\_FIR[28:16] FIRVINC bit field value is greater than 4096, it is clipped to 4096.
- If DISPC\_VIDp\_SIZE[26:16] SIZEY equals 0x1, SIZEY is replaced by 0x2 in the previous equation.
- The DISPC\_VIDp\_PICTURE\_SIZE[26:16] ORGSIZEY and DISPC\_VIDp\_SIZE[26:16] SIZEY bit field values must be programmed with the value desired minus 1.

- Horizontal up/downsampling increment value DISPC\_VIDp\_FIR[11:0] FIRHINC bit field: The unsigned integer value range is [1:4096]. The software calculates the value using the following equation:

$$FIRHINC = 1024 * \left( \frac{ORGSIZEX}{SIZEX} \right)$$

dispc-067

(11)

**NOTE:**

- If the DISPC\_VIDp\_FIR[12:0] FIRHINCbit field value is greater than 4096, it is clipped to 4096.
  - If DISPC\_VIDp\_SIZE[ 10:0] SIZEX equals 1, DISPC\_VIDp\_SIZE[ 10:0] SIZEX is replaced by 2 in the previous equation.
  - The DISPC\_VIDp\_PICTURE\_SIZE[10:0] ORGSIZEX and DISPC\_VIDp\_SIZE[ 10:0] SIZEX bit field values must be programmed with the value desired minus 1.
- 
- Vertical up/downsampling accumulator value DISPC\_VIDp\_ACCU\_j[26:16] VERTICALACCU bit field: The signed integer value range is [-1024:1023]. The accumulator value indicates on which phase the vertical filtering starts. The register DISPC\_VIDp\_ACCU\_0 is used for progressive output and for interlace output, the two registers DISPC\_VIDp\_ACCU\_0 and DISPC\_VIDp\_ACCU\_1 are used. Similarly, DISPC\_VIDp\_ACCU2\_0 and DISPC\_VIDp\_ACCU2\_1 are used in progressive or interlace output to set accumulator value of Cb and Cr components when scaling YUV format.
  - Vertical up/downsampling line buffer configuration DISPC\_VIDp\_ATTRIBUTES[21] VERTICALTAPS bit: The default value at reset time is 0x0 (3-tap configuration is used). If the bit field is reset, the 3-tap configuration is used.
  - Horizontal up/downsampling accumulator value DISPC\_VIDp\_ACCU\_j[10:0] HORIZONTALACCU bit field: The signed integer value range is [-1024:1023]. The accumulator value indicates on which phase the horizontal filtering starts. The register DISPC\_VIDp\_ACCU\_0 is used for progressive output and for interlace output, the two registers DISPC\_VIDp\_ACCU\_0 and DISPC\_VIDp\_ACCU\_1 are used. Similarly, DISPC\_VIDp\_ACCU2\_0 and DISPC\_VIDp\_ACCU2\_1 are used in progressive or interlace output to set accumulator value of Cb and Cr components when scaling YUV format.

**Table 10-40. Vertical/Horizontal Accumulator Phase**

Accumulator Value	Phases f
0	0
128 or -896	1
256 or -768	2
384 or -640	3
512 or -512	4
640 or -384	5
768 or -256	6
896 or -128	7

- Vertical up/downsampling coefficients:
  - The 3-tap vertical up/downsampling coefficients are defined in DISPC\_VIDp\_FIR\_COEF\_HV\_i registers. There are eight registers for the eight phases with three coefficients for each, or a total of 24 programmable coefficients for the vertical up/downsampling block. Each register contains two 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one).
  - The 5-tap vertical up/downsampling coefficients: 2 extra-tap vertical up/downsampling coefficients are defined in DISPC\_VIDp\_FIR\_COEF\_V\_i registers. There are 8 registers for the 8 phases with 2 coefficients for each of them so a total of 16 programmable coefficients for the vertical up/downsampling block used in addition of the 3-tap registers defined above
 For YUV vertical up/downsampling coefficients are set in DISPC\_VIDp\_FIR\_COEF\_HV2\_i and DISPC\_VIDp\_FIR\_COEF\_V2\_i registers. [Table 10-38](#) and [Table 10-39](#) resumes all coefficients and there respective registers.
- Horizontal up/downsampling coefficients:
  - The DISPC\_VIDp\_FIR\_COEF\_H\_i registers and the DISPC\_VIDp\_FIR\_COEF\_HV\_i register define

the 5-tap horizontal up/downsampling coefficients. Each DISPC\_VIDp\_FIR\_COEF\_H\_i register contains three 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one). Each DISPC\_VIDp\_FIR\_COEF\_HV\_i register contains one 8-bit signed coefficient. A total of 40 programmable coefficients for the horizontal up/downsampling block are used.

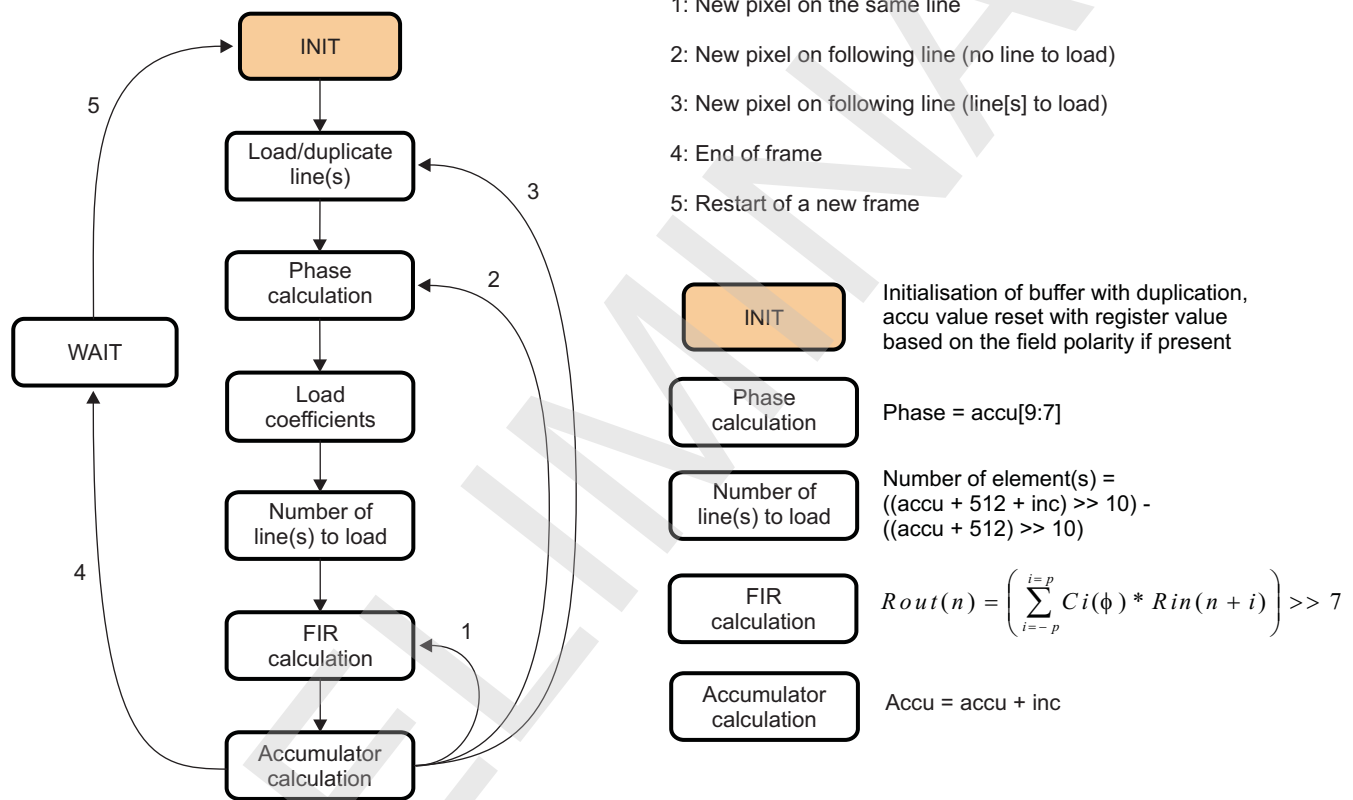
For YUV horizontal up/downsampling coefficients are set in DISPC\_VIDp\_FIR\_COEF\_HV2\_i and DISPC\_VIDp\_FIR\_COEF\_H2\_i registers. Table 10-38 and Table 10-39 resumes all coefficients and there respective registers.

### 10.2.4.10.4.1 Scaling Algorithms

The up/downsampling finite state machines (FSM) below are detailed in this section.

Figure 10-62 presents the vertical up/downsampling FSM.

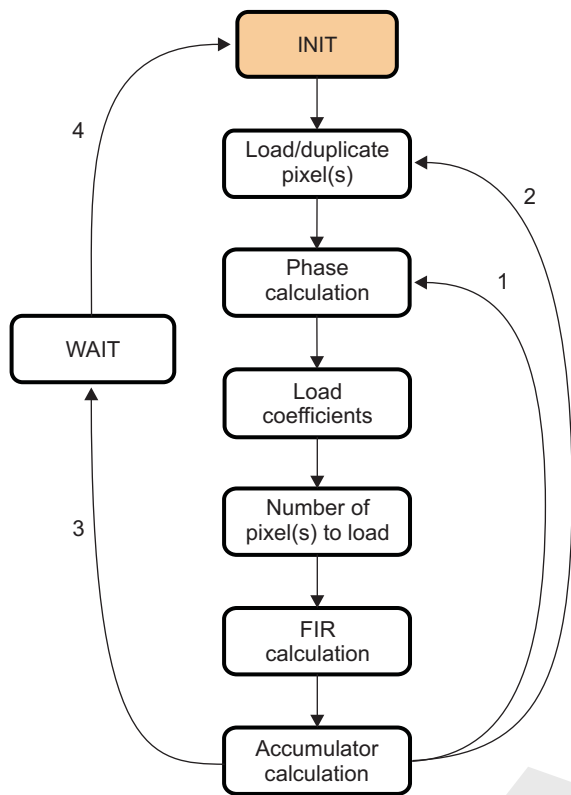
**Figure 10-62. Vertical Up/Downsampling Algorithm**



dispc-086

Figure 10-63 presents the horizontal up/downsampling FSM.

Figure 10-63. Horizontal Up/Downsampling Algorithm



- 1: New pixel (no pixel to load)
- 2: New pixel (pixel(s) to load)
- 3: End of line
- 4: Restart of a line

**INIT** Initialization of buffer with duplication, accu value reset with register value based on the field polarity if present

**Phase calculation** Phase = accu[9:7]

**Number of line(s) to load** Number of element(s) = ((accu + 512 + inc) >> 10) - ((accu + 512) >> 10)

**FIR calculation**  $R_{out}(n) = \left( \sum_{i=-p}^{i=p} C_i(\phi) * R_{in}(n + i) \right) >> 7$

**Accumulator calculation** Accu = accu + inc

dispc-087

10.2.4.10.4.2 Scaling limitations

Table 10-41 to Table 10-44 indicate the minimum ratio between the pixel clock frequency (PCLK) and the functional clock (DISPC\_CORE\_CLK) in function of input pixel format when using the scaler unit. PCLK and FCLK are asynchronous. For each LCD output, there is a dedicated LCD clock and programmable with LCD and PCD divisor values in DISPC\_DIVISORo[23:16][7:0].

**NOTE:** The DISPC\_CORE\_CLK is derived from DISPC\_FCLK via the LCD divisor

Table 10-41. Pixel Clock Frequency Limitations - Any pixel format - Active Matrix Display

DISPC_CORE_CLK/LCDn_PCLK Minimum ratio	Horizontal Resampling						
			Off	Up	1:1 - 1:2	1:2 - 1:3	1:3 - 1:4
Vertical Resampling	Off		2(1 <sup>(1)</sup> )	2(1 <sup>(1)</sup> )	2	3	4
	Up		2(1 <sup>(1)</sup> )	2(1 <sup>(1)</sup> )	2	3	4
	1:1 - 1:2	3-tap	2	2	4	6	8
		5-tap	2	2	4	6	8
	1:2 - 1:4		4	4	8	12	16

(1) The minimum ratio can be one if the data are output on the rising edge of the PCLK (DISPC\_POL\_FREQo.IPC = 0); otherwise minimum ratio shall be 2.

**Table 10-42. Pixel Clock Frequency Limitations - Any pixel format - Passive Matrix Display - Mono4**

F_CORE_CLK/LCDn_PCLK Minimum ratio	Horizontal Resampling						
Vertical Resampling		Off	Up	1:1 - 1:2	1:2 - 1:3	1:3 - 1:4	
	Off	4	4	8	12	16	
	Up	4	4	8	12	16	
	1:1 - 1:2	3-tap	8	8	16	24	32
		5-tap	8	8	16	24	32
1:2 - 1:4	16	16	32	48	64		

**Table 10-43. Pixel Clock Frequency Limitations - Any pixel format - Passive Matrix Display - Mono8**

F_CORE_CLK/LCDn_PCLK Minimum ratio	Horizontal Resampling						
Vertical Resampling		Off	Up	1:1 - 1:2	1:2 - 1:3	1:3 - 1:4	
	Off	8	8	16	24	32	
	Up	8	8	16	24	32	
	1:1 - 1:2	3-tap	16	16	32	48	64
		5-tap	16	16	32	48	64
1:2 - 1:4	32	32	64	96	128		

**Table 10-44. Pixel Clock Frequency Limitations - Any pixel format - Passive Matrix Display - Color**

F_CORE_CLK/LCDn_PCLK Minimum ratio	Horizontal Resampling						
Vertical Resampling		Off	Up	1:1 - 1:2	1:2 - 1:3	1:3 - 1:4	
	Off	3	3	6	9	12	
	Up	3	3	6	9	12	
	1:1 - 1:2	3-tap	6	6	12	18	24
		5-tap	6	6	12	18	24
1:2 - 1:4	12	12	24	36	48		

#### 10.2.4.11 Write-back Pipeline

The write-back pipeline is used to store in the system memory the capture of the overlay output or the output of one of the pipelines. The WB pipeline consists of a Color Space Conversion Unit, a Scaler unit and a RGB truncation logic. Since the overlay works on ARGB32-8888 format and the video accelerator works on YUV format, the color space conversion from RGB to YUV is used to directly output to memory the format that can be encoded with no extra processing.

The write-back pipeline is connected to all the pipeline outputs (GFX, VD1, VID2, and VID3 pipelines) and to the output of the three overlay managers (LCD1, LCD2 and TV). The input is selected by setting CHANNELIN field [DISPC\\_WB\\_ATTRIBUTES\[18:16\]](#) and the capture frame rate is set in CAPTUREMODE field [DISPC\\_WB\\_ATTRIBUTES\[26:24\]](#).

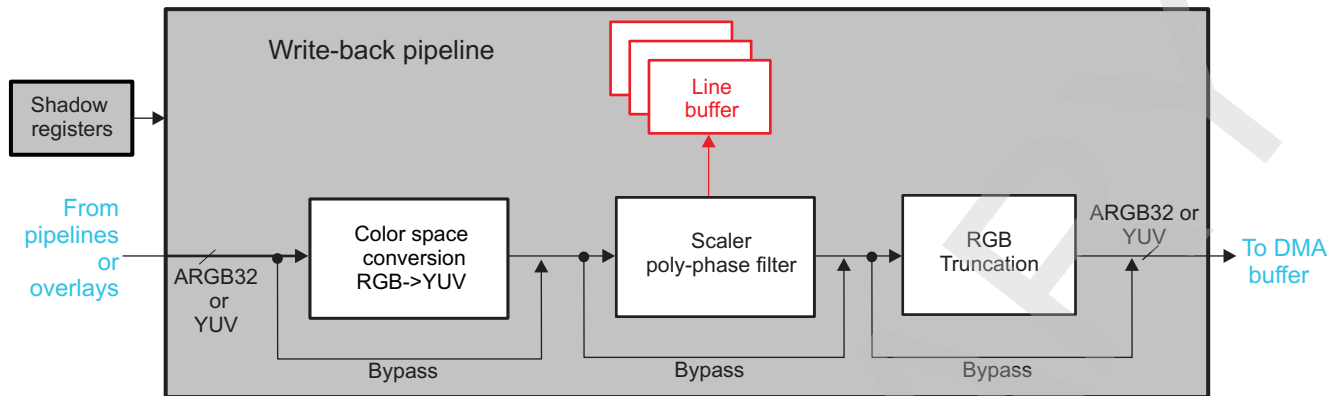
Since the output format of the TV overlay manager is ARGB40, the graphics pipeline output is ARGB40, the video pipeline output are YUV4:2:2, YUV4:2:0 or ARGB40 and the WB input is ARGB32 then the WB input does not take in consideration the 2 LSBs of each ARGB components.

The WB pipeline is enabled by setting to 0x1 ENABLE bit [DISPC\\_WB\\_ATTRIBUTES\[0\]](#).

[Figure 10-64](#) represents the graphics pipeline.



Figure 10-64. Write-back Pipeline



dispc-016

10.2.4.11.1 Color Space Conversion Unit RGB to YUV

The RGB to YUV color space conversion module converts the encoded pixel values from RGB24 into YUV4:4:4 format. For YUV4:2:0 or YUV4:2:2 formats, a chrominance sub-sampling is required after converting the RGB into YUV values. Figure 10-65 and Figure 10-66 show the 3 x 3 11-bit coefficients used to convert from RGB24 into YUV4:4:4. The user sets the coefficients according to the standard used to encode the pixel data in YUV color space. Table 10-45 resumes the coefficients with their respective bit fields.

Table 10-45. Color Space Conversion RGB to YUV Bit Field Setting

Coefficients	Bit Field Registers
Y <sub>R</sub>	YR field DISPC_WB_CONV_COEF0[10:0]
Y <sub>G</sub>	YG field DISPC_WB_CONV_COEF0[26:16]
Y <sub>B</sub>	YB field DISPC_WB_CONV_COEF1[10:0]
Cr <sub>R</sub>	CRR field DISPC_WB_CONV_COEF1[26:16]
Cr <sub>G</sub>	CRG field DISPC_WB_CONV_COEF2[10:0]
Cr <sub>B</sub>	CRB field DISPC_WB_CONV_COEF2[26:16]
Cb <sub>R</sub>	CBR field DISPC_WB_CONV_COEF3[10:0]
Cb <sub>G</sub>	CBG field DISPC_WB_CONV_COEF3[26:16]
Cb <sub>B</sub>	CBB field DISPC_WB_CONV_COEF4[10:0]

If the active range for the luminance samples (Y) is [16:235] and [16:240] for the chrominance samples (Cb and Cr), the values of Y, Cb, and Cr output components are clipped to the range [0:255]. The range selection is done by setting to 0x0 FULLRANGE bit DISPC\_WB\_ATTRIBUTES[11].

Figure 10-65. RGB to YCbCr (FULLRANGE=0)

$$\begin{bmatrix} Y_{OUT} \\ Cb_{OUT} \\ Cr_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} Y_R & Y_G & Y_B \\ Cb_R & Cb_G & Cb_B \\ Cr_R & Cr_G & Cr_B \end{bmatrix} * \begin{bmatrix} R_{IN} \\ G_{IN} \\ B_{IN} \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

dispc-017

If the active range for the luminance samples (Y) and or the chrominance samples (Cb and Cr) is [0:255], the values of Y, Cb, and Cr output components are clipped to the range [0:255]. The range selection is done by setting to 0x1 FULLRANGE bit DISPC\_WB\_ATTRIBUTES[11].

**Figure 10-66. RGB to YCbCr (FULLRANGE=1)**

$$\begin{bmatrix} Y_{OUT} \\ Cb_{OUT} \\ Cr_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} Y_R & Y_G & Y_B \\ Cb_R & Cb_G & Cb_B \\ Cr_R & Cr_G & Cr_B \end{bmatrix} * \begin{bmatrix} R_{IN} \\ G_{IN} \\ B_{IN} \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

dispc-018

**10.2.4.11.2 Scaler Unit**

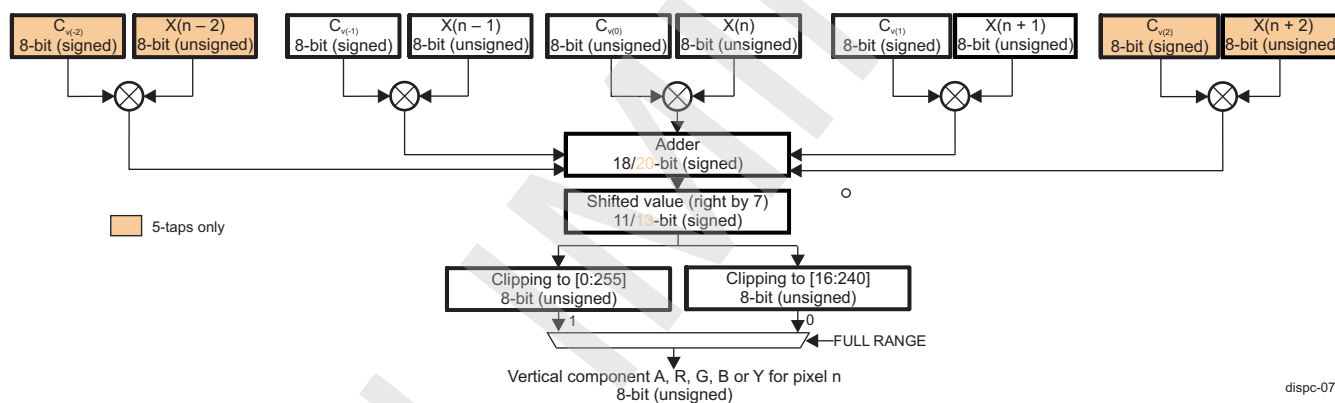
The WB pipeline scaler Unit functional aspect is identical to the video pipelines Scaler Unit one, refer to [Section 10.2.4.10.4](#), except in the output width when scaling ARGB components. The resulting output format being ARGB32 instead of ARGB40.

The programmable coefficients of the polyphase filters are signed 8-bit values (except for the central coefficient which is unsigned). The WB scaler component has an 8 bit input and an 8 bit output.

[Figure 10-67](#) and [Figure 10-68](#) details the scaler macro-architecture for the component A, R, G, B and Y. [Figure 10-69](#) and [Figure 10-70](#) details the scaler macro-architecture for component Cr and Cb.

The scaling output can be clipped to an output range of [0:255] or [16:240] by configuring FULLRANGE bit DISPC\_WB\_ATTRIBUTES[11].

**Figure 10-67. Macro-architecture of the Vertical Scaling for A, R, G, B and Y Component**



**Figure 10-68. Macro-architecture of the Horizontal Scaling for A, R, G, B and Y Component**

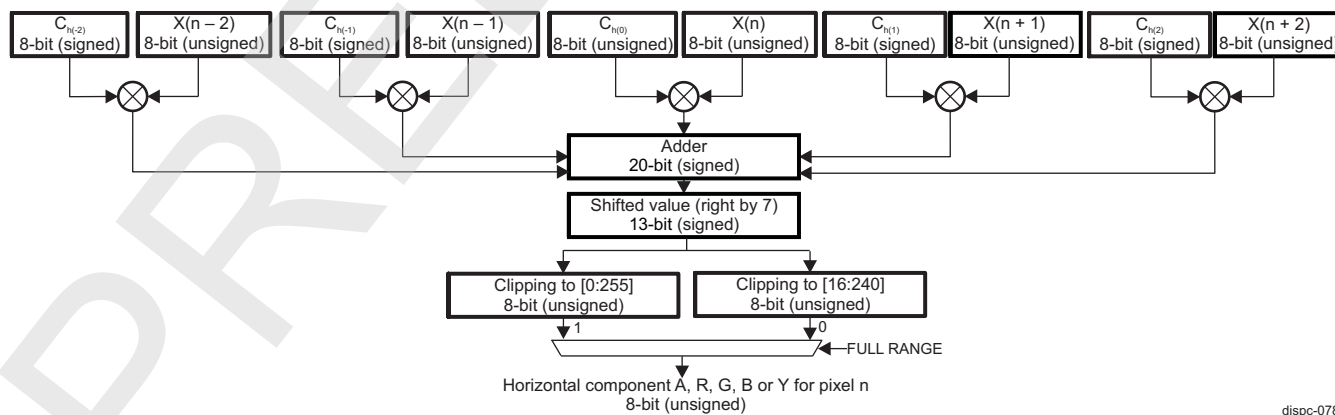
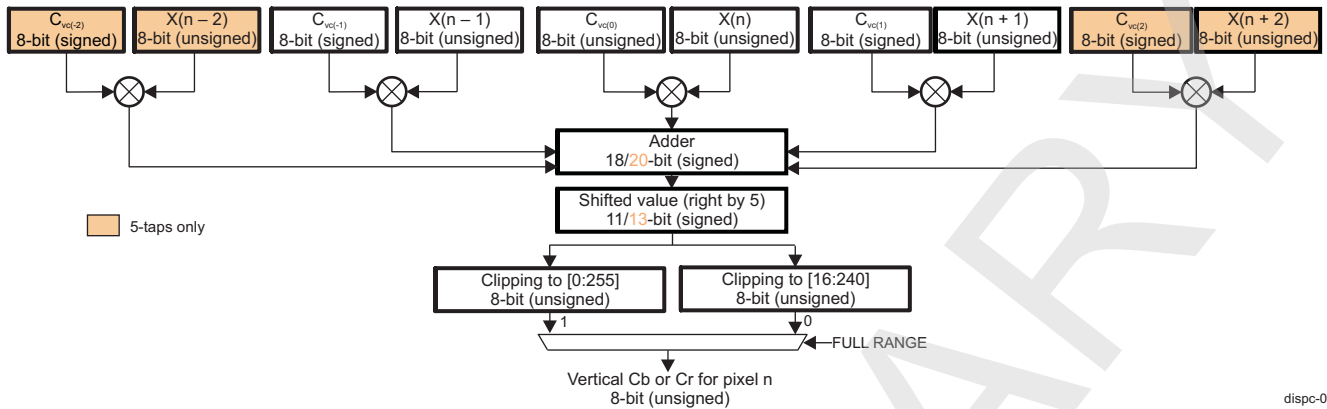
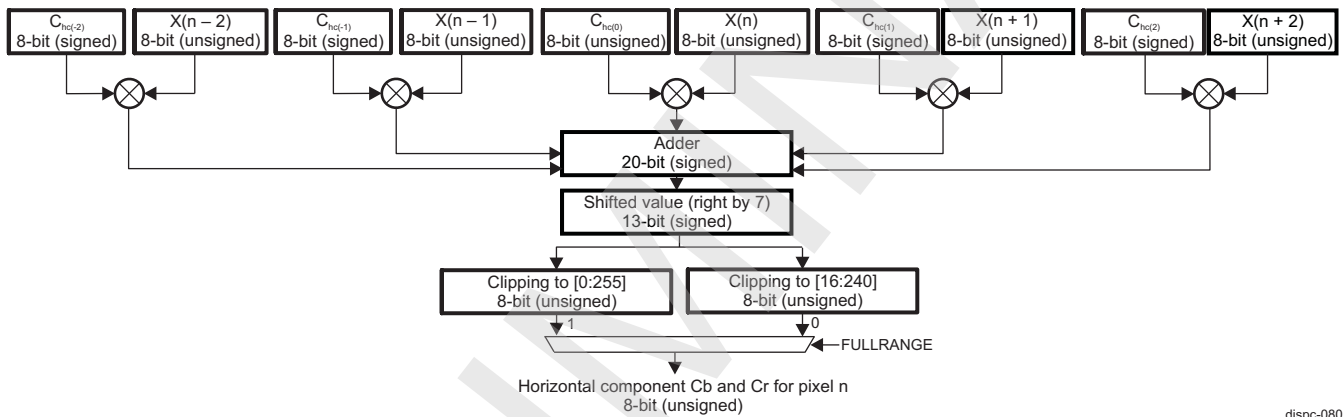


Figure 10-69. Macro-architecture of the Vertical Scaling for Cr and Cb Component



dispc-079

Figure 10-70. Macro-architecture of the Horizontal Scaling for Cr and Cb Component



dispc-080

Table 10-46 and Table 10-47 resume all the bit fields in function to set for each coefficients.

Table 10-46. Register Bifield Associated to Coefficient for ARGB and Y Configuration in WB Scaler

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Vertical	Cv(-2)		FIRVC22	DISPC_WB_FIR_COEF_V_i
	Cv(-1)	FIRVC2	FIRVC2	DISPC_WB_FIR_COEF_HV_i
	Cv(0)	FIRVC1	FIRVC1	DISPC_WB_FIR_COEF_HV_i
	Cv(1)	FIRVC0	FIRVC0	DISPC_WB_FIR_COEF_HV_i
	Cv(2)		FIRVC00	DISPC_WB_FIR_COEF_V_i
Horizontal	Ch(-2)		FIRHC4	DISPC_WB_FIR_COEF_HV_i
	Ch(-1)		FIRHC3	DISPC_WB_FIR_COEF_H_i
	Ch(0)	N/A	FIRHC2	DISPC_WB_FIR_COEF_H_i
	Ch(1)		FIRHC1	DISPC_WB_FIR_COEF_H_i
	Ch(2)		FIRHC0	DISPC_WB_FIR_COEF_H_i

**Table 10-47. Register Bifield Associated to Coefficient for Cb and Cr Configuration in WB Scaler**

Taps	Coefficient	3 Taps	5 Taps	Registers
		Bit Field	Bit Field	
Vertical	Cvc(-2)		FIRVC22	DISPC_WB_FIR_COEF_V2_i
	Cvc(-1)	FIRVC2	FIRVC2	DISPC_WB_FIR_COEF_HV2_i
	Cvc(0)	FIRVC1	FIRVC1	DISPC_WB_FIR_COEF_HV2_i
	Cvc(1)	FIRVC0	FIRVC0	DISPC_WB_FIR_COEF_HV2_i
	Cvc(2)		FIRVC00	DISPC_WB_FIR_COEF_V2_i
Horizontal	Chc(-2)		FIRHC4	DISPC_WB_FIR_COEF_HV2_i
	Chc(-1)		FIRHC3	DISPC_WB_FIR_COEF_H2_i
	Chc(0)	N/A	FIRHC2	DISPC_WB_FIR_COEF_H2_i
	Chc(1)		FIRHC1	DISPC_WB_FIR_COEF_H2_i
	Chc(2)		FIRHC0	DISPC_WB_FIR_COEF_H2_i

The WB scaler unit vertical or/and horizontal sampling is defined by setting/resetting the [DISPC\\_WB\\_ATTRIBUTES\[6:5\] RESIZEENABLE](#) bit.

A set of configuration must be valid before enabling the video up/downsampling block.

The following fields define the configuration of the video up/downsampling block for WB:

- Vertical up/downsampling increment value [DISPC\\_WB\\_FIR\[28:16\] FIRVINC](#) bit field. The unsigned integer value range is [1:4096]. The software calculates the value using the following equation:

$$FIRVINC = 1024 * \left( \frac{ORGSIZEY}{SIZEY} \right)$$

dispc-066

(12)

**NOTE:**

- If the [DISPC\\_WB\\_FIR\[28:16\] FIRVINC](#) bit field value is greater than 4096, it is clipped to 4096.
- If [DISPC\\_WB\\_SIZE\[26:16\] SIZEY](#) equals 0x1, SIZEY is replaced by 0x2 in the previous equation.
- The [DISPC\\_WB\\_PICTURE\\_SIZE\[26:16\] ORGSIZEY](#) and [DISPC\\_WB\\_SIZE\[26:16\] SIZEY](#) bit field values must be programmed with the value desired minus 1.

- Horizontal up/downsampling increment value **DISPC\_WB\_FIR**[11:0] FIRHINC bit field: The unsigned integer value range is [1:4096]. The software calculates the value using the following equation:

$$FIRHINC = 1024 * \left( \frac{ORGSIZEX}{SIZE_X} \right)$$

dispc-067

(13)

**NOTE:**

- If the **DISPC\_WB\_FIR**[12:0] FIRHINCbit field value is greater than 4096, it is clipped to 4096.
  - If **DISPC\_WB\_SIZE**[ 10:0] SIZE\_X equals 1, **DISPC\_WB\_SIZE**[ 10:0] SIZE\_X is replaced by 2 in the previous equation.
  - The **DISPC\_WB\_PICTURE\_SIZE**[10:0] ORGSIZE\_X and **DISPC\_WB\_SIZE**[ 10:0] SIZE\_X bit field values must be programmed with the value desired minus 1.
- 
- Vertical up/downsampling accumulator value **DISPC\_WB\_ACCU\_j**[26:16] VERTICALACCU bit field: The signed integer value range is [-1024:1023]. The accumulator value indicates on which phase the vertical filtering starts. The register **DISPC\_WB\_ACCU\_0** is used for progressive output and for interlace output, the two registers **DISPC\_WB\_ACCU\_0** and **DISPC\_WB\_ACCU\_1** are used. Similarly, **DISPC\_WB\_ACCU2\_0** and **DISPC\_WB\_ACCU2\_1** are used in progressive or interlace output to set accumulator value of Cb and Cr components when scaling YUV format.
  - Vertical up/downsampling line buffer configuration **DISPC\_WB\_ATTRIBUTES**[21] VERTICALTAPS bit: The default value at reset time is 0x0 (3-tap configuration is used). If the bit field is reset, the 3-tap configuration is used.
  - Horizontal up/downsampling accumulator value **DISPC\_WB\_ACCU\_j**[10:0] HORIZONTALACCU bit field: The signed integer value range is [-1024:1023]. The accumulator value indicates on which phase the horizontal filtering starts. The register **DISPC\_WB\_ACCU\_0** is used for progressive output and for interlace output, the two registers **DISPC\_WB\_ACCU\_0** and **DISPC\_WB\_ACCU\_1** are used. Similarly, **DISPC\_WB\_ACCU2\_0** and **DISPC\_WB\_ACCU2\_1** are used in progressive or interlace output to set accumulator value of Cb and Cr components when scaling YUV format.

**Table 10-48. Vertical/Horizontal Accumulator Phase**

Accumulator Value	Phases f
0	0
128 or -896	1
256 or -768	2
384 or -640	3
512 or -512	4
640 or -384	5
768 or -256	6
896 or -128	7

- Vertical up/downsampling coefficients:
  - The 3-tap vertical up/downsampling coefficients are defined in **DISPC\_WB\_FIR\_COEF\_HV\_i** registers. There are eight registers for the eight phases with three coefficients for each, or a total of 24 programmable coefficients for the vertical up/downsampling block. Each register contains two 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one).
  - The 5-tap vertical up/downsampling coefficients: 2 extra-tap vertical up/downsampling coefficients are defined in **DISPC\_WB\_FIR\_COEF\_V\_i** registers. There are 8 registers for the 8 phases with 2 coefficients for each of them so a total of 16 programmable coefficients for the vertical up/downsampling block used in addition of the 3-tap registers defined above
 For YUV vertical up/downsampling coefficients are set in **DISPC\_WB\_FIR\_COEF\_HV2\_i** and **DISPC\_WB\_FIR\_COEF\_V2\_i** registers. [Table 10-46](#) and [Table 10-47](#) resumes all coefficients and there respective registers.
- Horizontal up/downsampling coefficients:

- The `DISPC_WB_FIR_COEF_H_i` registers and the `DISPC_WB_FIR_COEF_HV_i` register define the 5-tap horizontal up/downsampling coefficients. Each `DISPC_WB_FIR_COEF_H_i` register contains three 8-bit signed coefficients and one 8-bit unsigned coefficient (the central one). Each `DISPC_WB_FIR_COEF_HV_i` register contains one 8-bit signed coefficient. A total of 40 programmable coefficients for the horizontal up/downsampling block are used.

For YUV horizontal up/downsampling coefficients are set in `DISPC_WB_FIR_COEF_HV2_i` and `DISPC_WB_FIR_COEF_H2_i` registers. Table 10-46 and Table 10-47 resumes all coefficients and there respective registers.

### 10.2.4.11.3 RGB Truncation Logic

The truncation logic is used to convert a pixels from ARGB-32bits format, into a lower color depth: 12-bit or 16-bit format based. Setting to 0x1 TRUNCATIONENABLE bit `DISPC_WB_ATTRIBUTES[10]` enables the truncation into the pixel format defined by FORMAT `DISPC_WB_ATTRIBUTES[4:1]`. The truncation is done by removing the necessary LSB of each component to match the setted output format. Table 10-49 details the truncation done on each component of the pixel.

**Table 10-49. Truncation Logic**

Output Formats	A[7:0]	R[7:0]	G[7:0]	GB[7:0]
	MSB LSB	MSB LSB	MSB LSB	MSB LSB
xRGB12-4444	Ignored	R[7:4]	G[7:4]	B[7:4]
RGBx12-4444	Ignored	R[7:4]	G[7:4]	B[7:4]
RGB16-565	Ignored	R[7:3]	G[7:2]	B[7:3]
xRGB16-1555	Ignored	R[7:3]	G[7:3]	B[7:3]
ARGB16-4444	A[7:4]	R[7:4]	G[7:4]	B[7:4]
RGBA16-4444	A[7:4]	R[7:4]	G[7:4]	B[7:4]
ARGB16-1555	A[7]	R[7:3]	G[7:3]	B[7:3]

**NOTE:** If there is no Alpha field in the pixel format description and 0s or 1's should fill the container. 0s shall be used for transparent and 1's for opaque. For example xRGB12 pixel format, the upper 4 bits are set to 0s since RGB value is only 12 bits inside a 16-bit container.

### 10.2.4.12 LCD Outputs

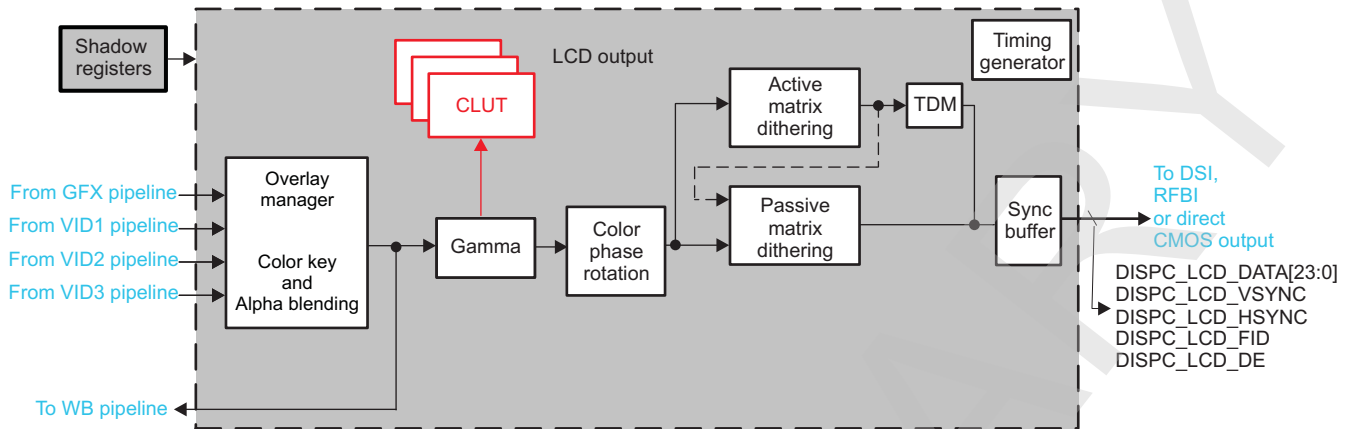
The LCD1 and LCD2 output paths consist of several processing blocks, see Figure 10-71:

- Overlay manager
- Gamma correction Unit
- Color Phase Rotation also used for RGB to YUV conversion.
- Active Matrix Dithering with TDM
- Passive Matrix Dithering
- Synchronization buffer
- Timing generator

The display subsystem supports two types of display technologies (both monochrome and color modes):

- Passive matrix displays: supporting up to 16, 256, or 3375 colors to be displayed in each frame, depending on the color depth. The monochrome LCD has 15 grayscale levels available
- Active matrix displays: the configuration of colors depends on the color depth:
  - 24 BPP supports 16,777,216 colors.
  - 18 BPP supports 262,144 colors.
  - 16 BPP supports 65,536 colors.
  - 12 BPP supports 4096 colors.

Figure 10-71. LCD Output Architecture



dispc-028

### 10.2.4.12.1 Overlay Manager

The overlay mechanism consists of displaying more than one layer (GFX, VID1 to VID3 layers) using

- A priority rule based on a Z-order: Application can set the ordering layer of the frames.
- Transparency color keys: Destination and Source transparency color keys can be set.
- Alpha blending values: Using the A component of pixel or a blending set by a user for a layer, a level of transparency can be determined.

Each pipeline, GFX, VID1, VID2 and VID3, is assigned to a single overlay and in consequence to a single display controller output, LCD1, LCD2, TV or WB pipeline. An overlay manager can be connected to all four pipelines outputs simultaneously. The pipeline output is directed using CHANNELOUT and CHANNELOUT2 bit field [DISPC\\_GFX\\_ATTRIBUTES/DISPC\\_VIDp\\_ATTRIBUTES\[8\]\[31:30\]](#). [Table 10-50](#) resumes the bitfield settings to direct a pipeline to a LCD/TV or WB output. The default value direct all pipelines to LCD1.

Table 10-50. Pipeline connection to LCD, TV or WB output

Overlay Manager/output	DISPC_GFX_ATTRIBUTES/DISPC_VIDp_ATTRIBUTES	
	CHANNELOUT bit	CHANNELOUT2 bit field
LCD1	0x0	0x0
LCD2	0x0	0x1
WB	0x0	0x3
TV	0x1	0x0

**NOTE:** The settings CHANNELOUT = 0x1 and CHANNELOUT2 = 0x1, 0x2 or 0x3 are reserved.

The output of each LCD overlay manager is connected to Color Phase Rotation block through the palette unit in the case of "Gamma" correction.

**NOTE:**

- When the pixel format is ARGB or RGBA, the color key match logic uses only the RGB value defined by ARGB or RGBA. The alpha blending factor is ignored.
- For LCD1 output, the same Color Look Up Table (CLUT) is used for BITMAP support by the graphics pipeline and for gamma correction. In case of BITMAP format for graphics, gamma correction is not available on the primary LCD output.

### 10.2.4.12.1.1 Priority Rule

The overlay manager is configured using Z-order parameter. The Z-order value defined for each pipeline indicates the visibility order to the window on the screen. If the Z-order value of window A is lower than the Z-order to the layer B, layer A is displayed below layer B. The transparency color keys and the alpha blending factors are then used to blend the layers together, refer to [Section 10.2.4.12.1.2](#) and [Section 10.2.4.12.1.3](#). The Z-order is enable by setting to 0x1 ZORDERENABLE bit `DISPC_GFX_ATTRIBUTES[25]` or `DISPC_VIDp_ATTRIBUTES[25]` and by defining the Z-order in ZORDER field `DISPC_GFX_ATTRIBUTES[27:26]` and `DISPC_VIDp_ATTRIBUTES[27:26]`. [Table 10-51](#) resumes the registers settings to enable and set the Z-order of a pipelines. [Table 10-51](#) shows the default Z-order values when `LCDALPHABLENDERENABLE` and `ZORDERENABLE` are disable.

**Table 10-51. Z-Order Register Settings and Default Configuration**

Pipeline	LCDALPHA BLENDERENABLE <sup>(1)</sup>	ZORDERENABLE	ZORDER	Resulting Z-order number
GFX	0	0	Don't care	0
	0	1	ZORDER	ZORDER
	1	Don't care	Don't care	3
VID1	0	0	Don't care	1
	0	1	ZORDER	ZORDER
	1	Don't care	Don't care	0
VID2	0	0	Don't care	2
	0	1	ZORDER	ZORDER
	1	Don't care	Don't care	1
VID3	0	0	Don't care	3
	0	1	ZORDER	ZORDER
	1	Don't care	Don't care	2

<sup>(1)</sup> Applies only to LCD1.

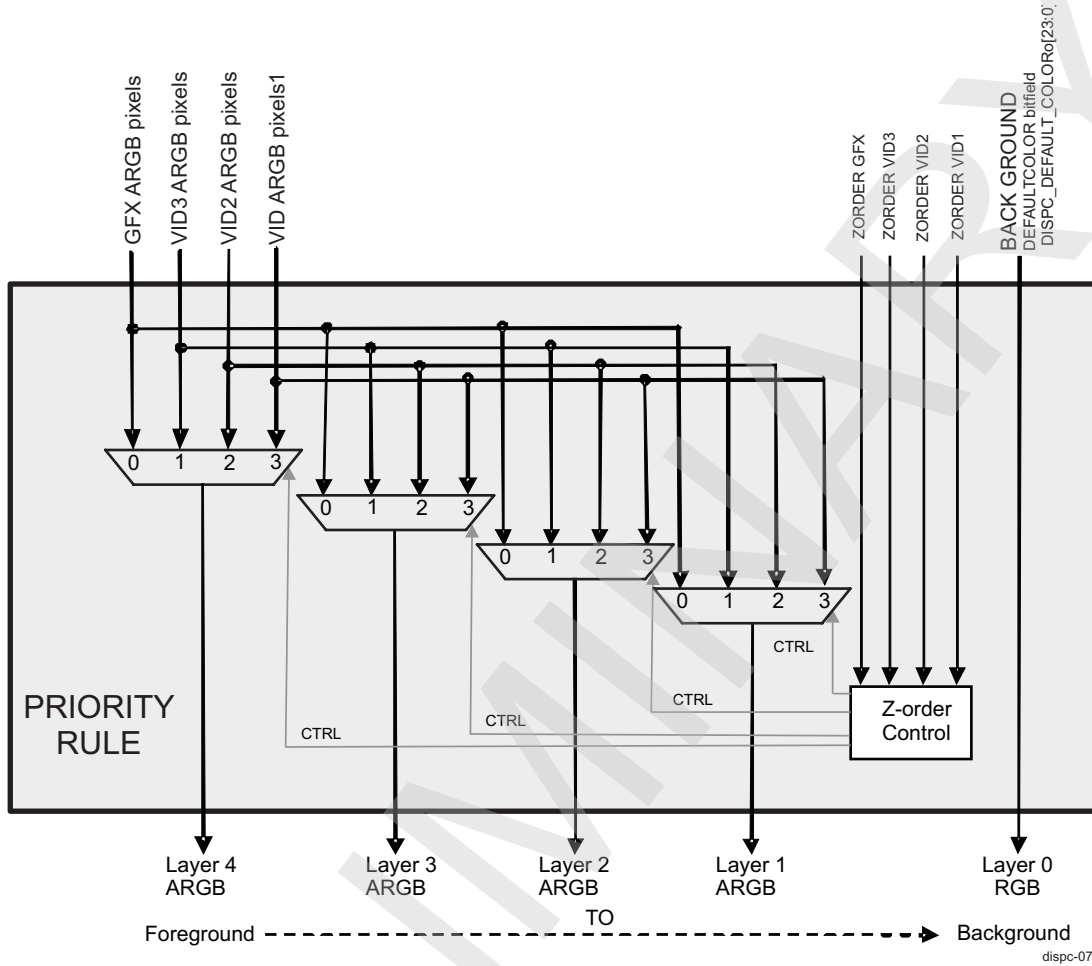
[Figure 10-72](#) represents the architecture of the priority rule.

**NOTE:**

- If `ZORDERENABLE = 1`, each Z-order must be different for each active pipeline. It is not possible to use the same value for more than one pipeline.
- Two modes are maintained for backward compatibility with OMAP3430:
  - `LCDALPHABLENDERENABLE = 0` and `ZORDERENABLE = 0` equivalent to the normal mode overlay settings
  - `LCDALPHABLENDERENABLE = 1` equivalent to the Alpha mode overlay settings.
- When OMAP3430 backward compatibility mode is to be used, the following must be considered:
  - If Z-order is disabled for the VID1 and VID2 pipelines, the pipeline data is output on the LCD1/LCD2/TV overlay.
  - If Z-order is disabled for the VID3 pipeline, the pipeline data is output only on the LCD2 overlay.



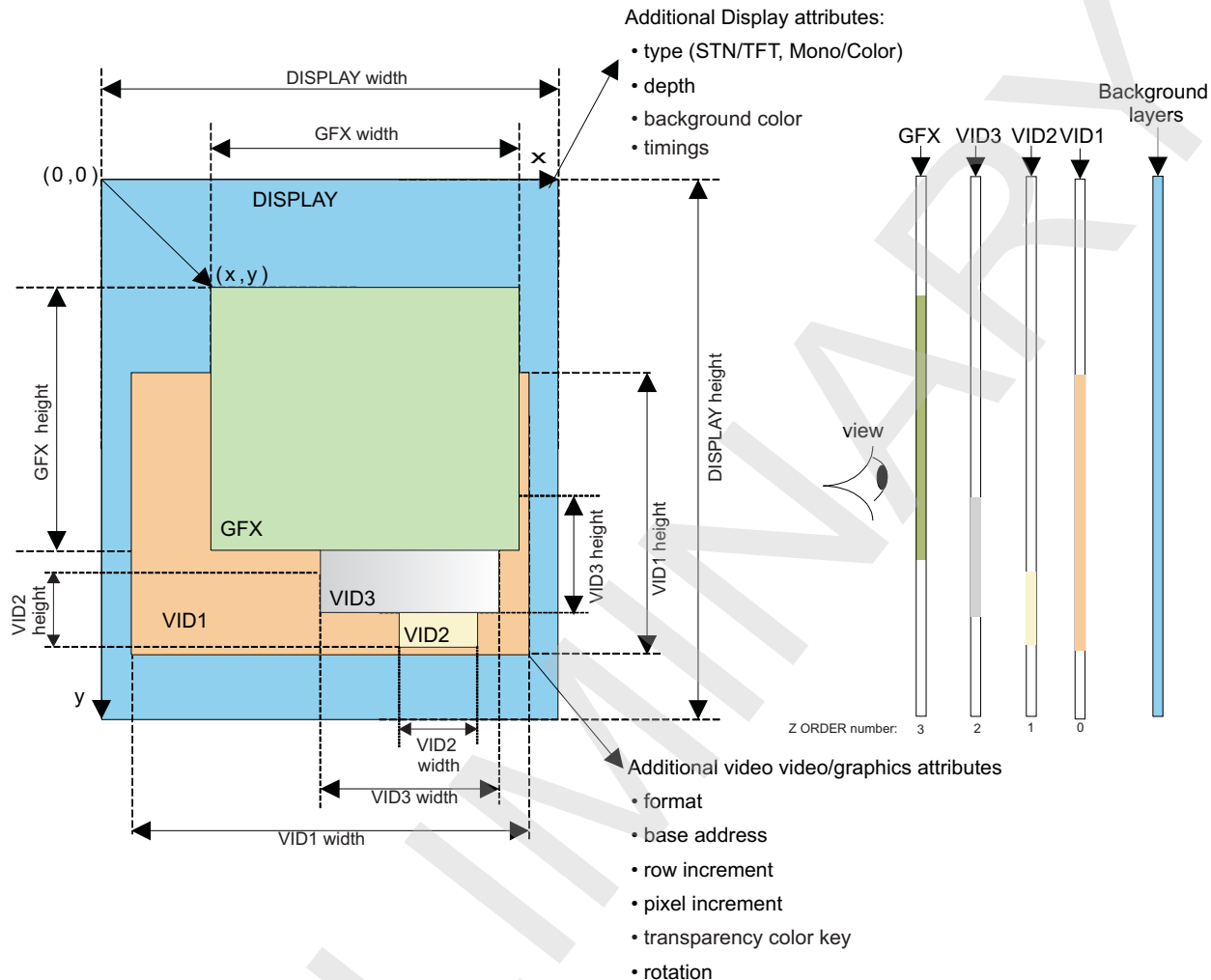
Figure 10-72. Priority Rule Architecture



Each enabled layer (pipeline) shall have its height and width defined in `SIZEX` and `SIZEY` fields `DISPC_GFX_SIZE[26:16][10:0]` `DISPC_VIDp_SIZE[26:16][10:0]` and its x and y positions defined in `POSX` and `POSY` fields `DISPC_GFX_POSITION[26:16][10:0]` `DISPC_VIDp_POSITION[26:16][10:0]`. If there are no graphics or video-encoded pixels at a specific position, the programmable, solid background color appears. The solid background color is set in `DEFAULTCOLOR` bit field `DISPC_DEFAULT_COLORo[23:0]`. Figure 10-73 represents an example of priority rule.

The Z-order re-ordering block is responsible for mapping to the blender logic the pipelines always in the same order: from background to foreground.

**Figure 10-73. Example of Priority Rule: From Lower to Higher VID1, VID2, VID3, GFX**

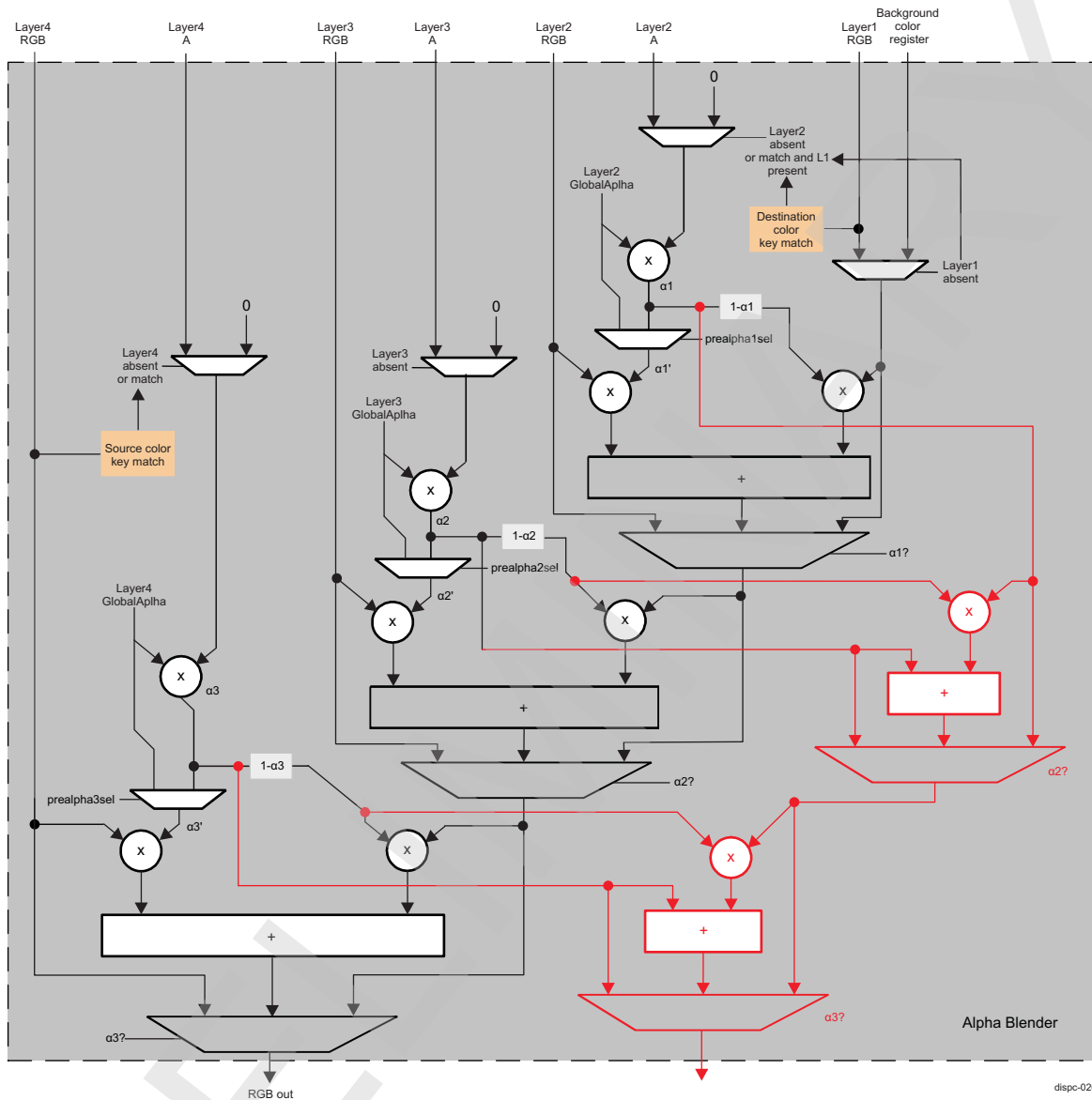


dispc-019

### 10.2.4.12.1.2 Alpha Blender

Figure 10-74 shows the alpha blending processing in detail.

Figure 10-74. Alpha Blending Architecture with Premultiplied Alpha Support



**NOTE:** "1-alpha" operator corresponds to the basic 1's complement operation.

The alpha blending value is defined by:

- The component value A when using an ARGB or a RGBA pixel formats.
  - For ARGB-1555, the alpha blending is defined using a 1-bit value. It is converted into an 8-bit value by duplicating the 1-bit value, refer to [Table 10-52](#).
  - For ARGB-4444, the alpha blending is defined using a 4-bit value. It is converted into an 8-bit value by duplicating the 4-bit value, refer to [Table 10-52](#).
  - If the pixel format contains no alpha blending value, the pixel alpha value is considered to be 0xFF and if alpha is equals to 0xFF then there is no multiplication.
  - For BITMAP or YUV formats, there is no alpha blending factor associated to each pixel value. Only the global alpha blending factor associated to the window displaying the BITMAP or YUV formats is used
- The global alpha blending value set in VID3GLOBALALPHA, VID2GLOBALALPHA, VID1GLOBALALPHA, GFXGLOBALALPHA field `DISPC_GLOBAL_ALPHA[31:0]` for LCD.

- A total alpha blending value can be used when a combination of the pixel alpha blending value A and a global alpha blending are present. The resulting alpha value is determined as :  $\text{Alpha} = (\text{Pixel Alpha} * \text{Global Alpha}) / 2$

Table 10-52 indicates the percentage of alpha blending in function of the Alpha blending value on 8 bits.

**Table 10-52. Alpha Blending - ARGB**

Alpha Blending 1-Bit Value	Alpha Blending 4-Bit Value	Alpha Blending 8-Bit Value (Converted value or resulting alpha)	% Blending
0x0	0x0	0x00	100% (transparent)
N/A	0x1	0x11	93.33%
N/A	0x2	0x22	86.6%
N/A	...	...	...
N/A	0xE	0xEE	6.6%
0x1	0xF	0xFF	0% (opaque)

### Premultiplied Alpha

The image ARGB may have its RGB component already premultiplied with the Alpha (ARGB) where:

- $R = A * R$
- $G = A * G$
- $B = A * B$

The processing in that case is as follows:

- Color component of premultiplied layer are multiplied with the GlobalAlpha, if GlobalAlpha is not equal to zero
- Color component of the composed underlying layer are multiplied with  $(1-A) * \text{GlobalAlpha}$

The additional premultiplied alpha option is associated with the pipelines (GFX, VID1, VID2 and VID3). The option is accessible through the [28]PREMULTIPLYALPHA bit for the respective pipeline register: [DISPC\\_GFX\\_ATTRIBUTES](#), [DISPC\\_VID1\\_ATTRIBUTES](#), [DISPC\\_VID2\\_ATTRIBUTES](#) and [DISPC\\_VID3\\_ATTRIBUTES](#). The following settings are available:

- PREMULTIPLYALPHA bit = 0 : Source is not premultiplied with alpha. Full blending is done in the DISPC.
- PREMULTIPLYALPHA bit = 1 : Source is premultiplied with alpha. Partial blending is done.

---

**NOTE:** The *prealpha* controls in [Figure 10-74](#), correspond to the PREMULTIPLYALPHA of the pipelines mapped on the respective layers.

---

The logic marked in red colour in [Figure 10-74](#) corresponds to the alpha value, computed when the Write Back channel copies back to memory the premultiplied color component:  $A(\text{destination}) = A(\text{source}) + (1-A(\text{source}) * A(\text{destination}))$

When [DISPC\\_WB\\_ATTRIBUTES\[7\]](#) ALPHAENABLE bit is cleared, or when the overlay channel is not selected for WB, the computation of the A(destination) is disabled. The default value for [DISPC\\_WB\\_ATTRIBUTES\[7\]](#) ALPHAENABLE bit is 0x0. When the WB is configured to copy back one of the output channels (output of overlay), the following configurations are available:

- ALPHAENABLE bit is set to 0x1: The WB pipe copies back to memory the premultiplied alpha calculated through the overlay.
- ALPHAENABLE bit is set to 0x0: The alpha value is not written back.

---

**NOTE:** The [DISPC\\_WB\\_ATTRIBUTES\[7\]](#) ALPHAENABLE bit is effective only when one of the output channels is written back, otherwise it is ignored.

---

#### 10.2.4.12.1.3 Transparency Color Keys

The two transparency color keys are the source transparency color key and the destination transparency

color key. The transparency color key can be used only with BITMAP formats (1-, 2-, 4-, and 8-bit per pixel) and RGB formats (ARGB, RGB, RGBA, and RGBx) in this case the A information is ignored for the comparison between the pixel value and the color key value. It is possible to use YUV formats with some care since the comparison is between the input pixel value of the overlay manager from pipeline (GFX or one of VID pipelines depending on the Z-order) and the color key value. The YUV data is converted to RGB format. The user shall consider the color space conversion processing in order to define the RGB color key value used for the comparison in case the original format is YUV.

The transparency color key is enabled by setting to 0x1 TCKLCDENABLE bit [DISPC\\_CONFIG1\[10\]](#) for LCD1, [DISPC\\_CONFIG2\[10\]](#) for LCD2 and TCKTVENABLE bit [DISPC\\_CONFIG1\[12\]](#) for TV. The transparent color key is determined in TRANSCOLORKEY field [DISPC\\_TRANS\\_COLOR0\[23:0\]](#) for LCD, [DISPC\\_TRANS\\_COLOR2\[23:0\]](#) for LCD2 and [DISPC\\_TRANS\\_COLOR1\[23:0\]](#) for TV

**NOTE:**

- The video source transparency color key and graphics destination transparency color key cannot be active at the same time.
- For CLUT bitmaps, the palette index is compared to the transparency color key and not to the palette value pointed out by the palette index.

• Video source transparency color key

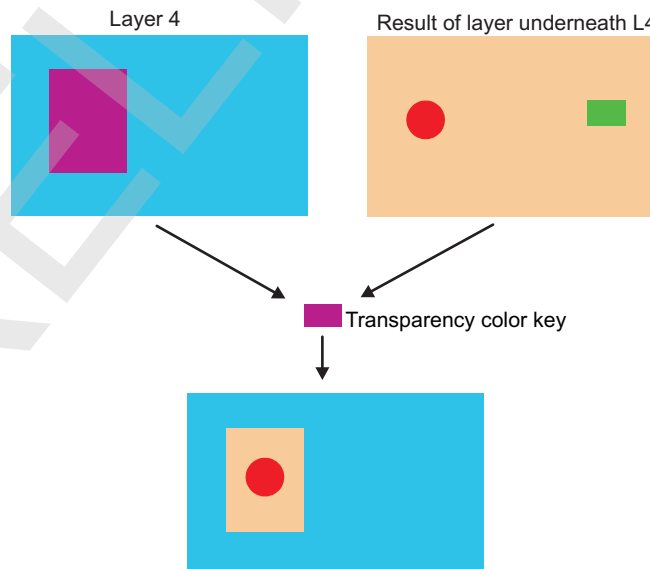
The video source transparency color key value defines the encoded pixel data considered as transparent pixel. The encoded pixel values with the source color key value are not visible and the under layers encoded pixel values or solid background color are visible (the pixel alpha blending value of the Layer 4 is forced to 0x00, fully transparent).

The scaler can be enabled as a preprocessing in the VID pipeline. But it is necessary to consider the pixel scaling preprocessing in order to define the color key value to be used after the rescaling for the comparison between the input pixel value to the overlay manager and the color key value.

The source transparency color key mode is selected by setting to 0x1 TCKLCDSELECTION bit [DISPC\\_CONFIG1\[11\]](#) for LCD1, [DISPC\\_CONFIG2\[11\]](#) for LCD2 and TCKTVENABLE bit [DISPC\\_CONFIG1\[13\]](#) for TV.

[Figure 10-75](#) shows an example of source color key. The pixels with the transparency color key are not displayed; instead, pixels of the resulting layer underneath are shown.

**Figure 10-75. Source Transparency Color Key Example**



dispc-021

• Graphics destination transparency color key value

The graphics destination transparency color key value defines the encoded pixels in the Layer 1 which will not be displayed, other Layer 1 pixel (non-equal to destination transparency color key) will be displayed over the Layer 2. The encoded pixel values with the destination color key value are pixels not

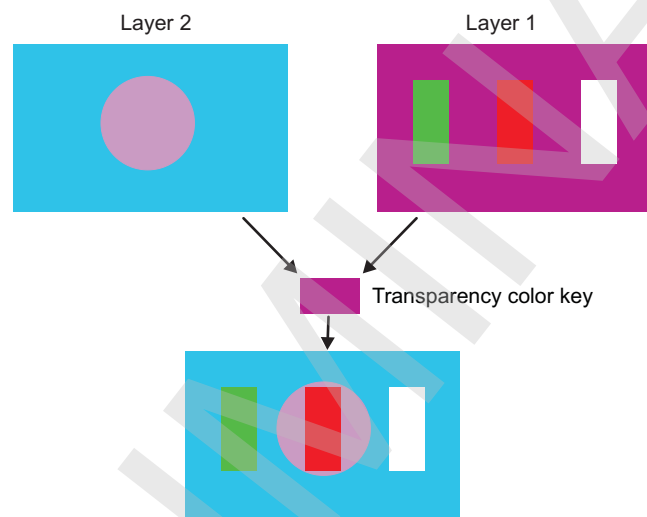
visible on the screen since pixels at the same position in the Layer 2 are visible otherwise encoded pixels are visible above the Layer 2. The destination transparency color key is applicable only if Layer 1 is overlapping Layer 2 (refer to Z-order section for details on layer position depending on the Z-order parameter, [Section 10.2.4.12.1.1](#)), otherwise the destination transparency color key is ignored.

The scaler can be enabled as a preprocessing in the VID pipeline. But it is necessary to consider the pixel scaling preprocessing in order to define the color key value to be used after the rescaling for the comparison between the input pixel value to the overlay manager and the color key value.

The destination transparency color key mode is selected by setting to 0x0 TCKLCDSELECTION bit [DISPC\\_CONFIG1\[11\]](#) for LCD, [DISPC\\_CONFIG2\[11\]](#) for LCD2 and TCKTVSELECTION bit [DISPC\\_CONFIG1\[13\]](#) for TV.

[Figure 10-75](#) shows an example of destination color key. The pixels, equals to the transparency color key, are not displayed and are replaced by Layer 2 pixel. but all other Layer 1 pixels, differs from transparency color key, are displayed over Layer 2.

**Figure 10-76. Destination Transparency Color Key Example**



dispc-022

#### 10.2.4.12.1.4 Overlay Optimization

The overlay optimization consists in fetching only the required pixels meaning pixels that contribute to the final picture to be displayed (LCD1, LCD2 or TV). The decision to fetch the pixel from memory is based on the information available in the registers and on the rules:

- The layer is enabled
- The global alpha blending factor for the layer is different than 0x00
- The current layer is behind a non-opaque layer (global alpha blending factor is different than 0xFF for the layer above)

The result of the overlay optimization is a reduction of the bandwidth by fetching only the mandatory pixels. The overlay mechanism is independent for each overlay: LCD1, LCD2 and TV. Since each layer (GFX, VID1, VID2, and VID3) can be associated at a time only to one overlay, it is possible to optimize the fetch of the pixels for each layer based on the overlay information. The overlay optimization needs to run on the DMA engine time window and not on the display time window. The pixels are fetched by the DMA engine in advance to the display processing.

The overlay optimization is enabled by setting to 0x1 OVERLAYOPTIMIZATION bit [DISPC\\_CONTROL1\[12\]](#) for LCD1, OVERLAYOPTIMIZATION bit [DISPC\\_CONTROL2\[12\]](#) for LCD2 and TVOVERLAYOPTIMIZATION bit [DISPC\\_CONTROL2\[13\]](#) for TV

**NOTE:**

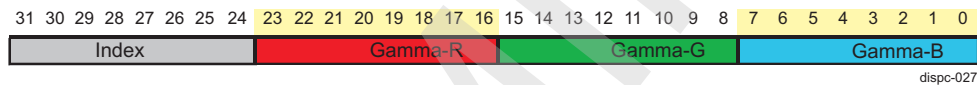
- The overlay optimization is not functional when a layer is using BITMAP1, BITMAP2 or BITMAP4 pixel formats
- The pixel alpha blending factor in case of ARGB and RGBA formats can not be used in order to take advantage of the pixel fully transparent (alpha blending factor equals to 0x00).

**10.2.4.12.2 Gamma Correction Unit**

In gamma curve mode, the selected encoded pixel values based on the color keys by the overlay manager from the video or graphics paths are sent to the Gamma curve table. Each component of encoded pixel value is used as a pointer to index one out of 256 24-bit gamma curve entries in the table. Each 8-bit component is replaced with the 8-bit table value corresponding to R, G or B component. The table is loaded by software. It is possible to load only part of the table. For each access to the table, the 24-bit value is associated with index in the table by concatenating the 24-bit value (LSB of 32-bit access) and the 8-bit index value (MSB of the 32-bit access). The [Figure 10-77](#) describes the format of one of the gamma curve value in the memory.

**NOTE:** If the CLUT is used in the GFX pipeline for palette, the gamma correction unit is not available on primary LCD.

**Figure 10-77. Data Memory Organization for Gamma Mode in LCD output**



**10.2.4.12.3 Color Phase Rotation Unit**

The Color Phase Rotation (CPR) can be used to correct the LCD output colorimetry in case of non pure white backlight.

The CPR is enabled by setting to 0x1 CPR bit DISPC\_CONFIGo[15]. The coefficients are programmed in the registers:

- Red 10-bit signed coefficients in DISPC\_CPRo\_COEF\_R
- Green 10-bit signed coefficients in DISPC\_CPRo\_COEF\_G
- Blue 10-bit signed coefficients in DISPC\_CPRo\_COEF\_B

The color phase rotation can be selected for passive and active matrix panel. The logic is integrated after the LCD overlay manager or the palette while using the gamma correction and before the spatial/temporal dithering. The color phase rotation can be selected to correct the nonpure white backlight of the LCD module by using a programmable matrix to convert the 24-bit RGB pixel value into a new 24-bit RGB pixel value. The matrix is programmed through a set of nine 10-bit signed coefficients. The output of the calculation is clipped to [0:255]. The color phase rotation is processed by the equation shown in [Figure 10-78](#). [Table 10-53](#) lists all coefficients with their respective bit field registers for settings.

**Figure 10-78. Color Phase Rotation Matrix**

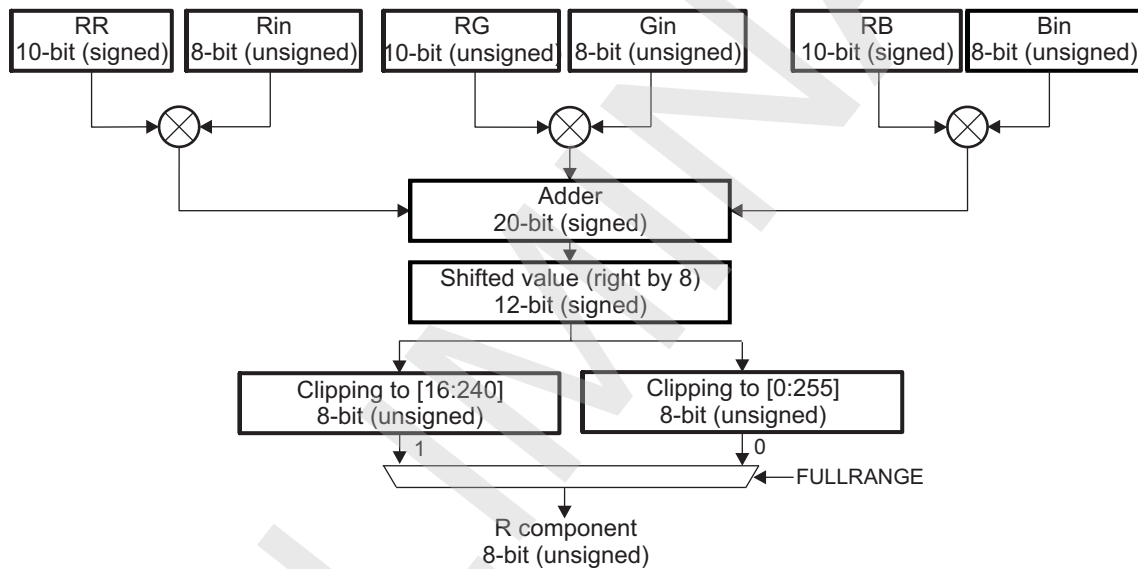
$$\begin{bmatrix} R_{out} \\ G_{out} \\ B_{out} \end{bmatrix} = \begin{bmatrix} R_r & R_g & R_b \\ G_r & G_g & G_b \\ B_r & B_g & B_b \end{bmatrix} * \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix}$$

dispc-023

**Table 10-53. CPR or RGB to YUV Coefficients with Bit Fields Associated**

Registers	Bit Field	Color Space Conversion	RGB to YUV
DISPC_CPRo_COEF_R	RR	Rr	Yr
	RG	Rg	Yg
	RB	Rb	Yb
DISPC_CPRo_COEF_G	GR	Gr	Cbr
	GG	Gg	Cbg
	GB	Gb	Cbb
DISPC_CPRo_COEF_B	BR	Br	Crr
	BG	Bg	Crg
	BB	Bb	Crb

Figure 10-79 shows the color phase rotation macro-architecture.

**Figure 10-79. Color Phase Rotation Macro Architecture**

dispc-024

#### 10.2.4.12.4 Passive Matrix

The Passive matrix displays supports 16, 256, or 3375 colors to be displayed in each frame, depending on the color depth. The monochrome LCD has 15 grayscale levels available. To enable a passive matrix output, STNTFT bit DISPC\_CONTROLo[3] must be set to 0x0.

Spatial/Temporal dithering is performed when using the passive matrix data path. The spatial/temporal dithering logic can be selected in order to enhance the quality of the Passive Matrix. When selected, the pixels are preprocessed by the Spatial/Temporal Dithering logic before the Passive Matrix Display dithering logic. The output format of the Spatial/Temporal Dithering logic is RGB 12-bit (not configurable). The dithering logic is integrated after the color phase rotation. The dithering algorithm is based on the Frame Rate Control. The dithering logic can process the pixels over a single frame, two frames, or four frames. The number of frame is selected by setting SPATIALTEMPORALDITHERINGFRAMES field DISPC\_CONTROLo[31:30]. The temporal dithering logic is enabled by setting to 0x1 STDITHERENABLE bit DISPC\_CONTROLo[7]. The output format of the dithering logic is RGB 12-bit (not configurable).

After the pixels, from GFX, VID pipelines, are merged depending on the transparency status and overlay, the result is sent to a color/grayscale temporal dither generator. The monochrome data or each RGB color component are encoded on 4 bits, which are the 4 MSBs of 8-bit pixel-encoded component defined by the merge of the graphics data and the video data.



These 4-bit values are used to select on the 16 intensity levels. The gray/color intensity is controlled by turning individual pixels on and off at varying period rates, making the average time the pixel is off longer than the average time the pixel is on, thus producing more intense grays/colors. The dithering generator also uses the intensity of adjacent pixels in the calculation to give the screen image a smooth appearance. The proprietary dither algorithm is optimized to provide a range of intensity values that matches the visual perception of color/gray graduations.

Single-panel monochrome screens use either four or eight data lines; single-panel color screens use eight data pins. The output width is configured by setting M8B bit DISPC\_CONTROLo[4] and the monochrome or color mode is selected by MONOCOLOR bit DISPC\_CONTROLo[2]

#### **10.2.4.12.5 Active Matrix**

Depending on the color depth, the Active matrix output:

- 24 BPP supports 16,777,216 colors.
- 18 BPP supports 262,144 colors.
- 16 BPP supports 65,536 colors.
- 12 BPP supports 4096 colors.

When in active matrix path configuration, after setting STNTFT bit DISPC\_CONTROLo[3] must be set to 0x1, 2 sub-modules need to be configured

- Spatial/Temporal Dithering
- Multiple Cycle Output Format (TDM)

##### **10.2.4.12.5.1 Spatial/Temporal Dithering**

When the active matrix path is used, the spatial/temporal dithering logic can be selected in order to enhance the quality of the Active Matrix outputs. The encoded pixel values are used by spatial/temporal dithering logic to display the data in a lower color depth on the LCD panel. The dithering logic is integrated after the color phase rotation and before the TDM. The spatial/temporal dithering algorithm is based on the (x,y) pixel position and Frame Rate Control. The dithering logic can process the pixels over a single frame, two frames, or four frames. The number of frame is selected by setting SPATIALTEMPORALDITHERINGFRAMES field DISPC\_CONTROLo[31:30]. In the case of a single frame, only spatial processing is applied and in multiple frames, spatial and temporal processing are applied to the pixels. The spatial/temporal dithering logic is enabled by setting to 0x1 STDITHERENABLE bit DISPC\_CONTROLo[7].

---

**NOTE:**

- If the interface data bus is smaller than the pixel format size and spatial/temporal dithering is not enabled, the MSBs of the pixel color components are output on the interface data bus
  - If the interface data bus is wider than the pixel format size, by programming the pixel components replication active/inactive, the MSB is replicated to the LSB of the interface data bus or the LSB is filled with 0s.
- 

##### **10.2.4.12.5.2 Multiple Cycle Output Format (TDM)**

The pixels, after the active matrix display processing, are formatted on one or multiple cycles (from one to three cycles). The number of bits for each cycle is set in the DISPC\_DATAo\_CYCLE1 for the 1<sup>st</sup> cycle, DISPC\_DATAo\_CYCLE2 for the 2<sup>nd</sup> cycle and DISPC\_DATAo\_CYCLE3 for the 3<sup>rd</sup> cycle. The interface data bus width can be 8-, 9-, 12-, or 16-bit. The configuration of the data bus is done in TFTDATALINES bit field DISPC\_CONTROLo[9:8].

When the TDM is disabled, the display controller outputs the pixels using the conventional formats: Passive matrix display/active matrix display monochrome/color.

---

**NOTE:** The passive matrix dithering logic is always bypassed in active matrix display.

---

Figure 10-80 to Figure 10-83 represent some various examples of TDM settings in function of pixel data formats and the interface data bus width.

**Figure 10-80. 8-bit Interface Settings**

	24-BPP		
	1st cycle	2nd cycle	3rd cycle
Data[7]	R0[7]	G0[7]	B0[7]
Data[6]	R0[6]	G0[6]	B0[6]
Data[5]	R0[5]	G0[5]	B0[5]
Data[4]	R0[4]	G0[4]	B0[4]
Data[3]	R0[3]	G0[3]	B0[3]
Data[2]	R0[2]	G0[2]	B0[2]
Data[1]	R0[1]	G0[1]	B0[1]
Data[0]	R0[0]	G0[0]	B0[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x2  
 DISPC\_DATA0\_CYCLE1 = 0x00000008  
 DISPC\_DATA0\_CYCLE2 = 0x00000008  
 DISPC\_DATA0\_CYCLE3 = 0x00000008

	18-BPP		
	1st cycle	2nd cycle	3rd cycle
Data[7]	R0[5]	G0[3]	x
Data[6]	R0[4]	G0[2]	x
Data[5]	R0[3]	G0[1]	x
Data[4]	R0[2]	G0[0]	x
Data[3]	R0[1]	B0[5]	x
Data[2]	R0[0]	B0[4]	x
Data[1]	G0[5]	B0[3]	B0[1]
Data[0]	G0[4]	B0[2]	B0[0]

DISPC\_CONTROL0.TDMCycleFormat = 0x2  
 DISPC\_DATA0\_CYCLE1 = 0x00000008  
 DISPC\_DATA0\_CYCLE2 = 0x00000008  
 DISPC\_DATA0\_CYCLE3 = 0x00000002

	16-BPP	
	1st cycle	2nd cycle
Data[7]	R0[4]	G0[2]
Data[6]	R0[3]	G0[1]
Data[5]	R0[2]	G0[0]
Data[4]	R0[1]	B0[4]
Data[3]	R0[0]	B0[3]
Data[2]	G0[5]	B0[2]
Data[1]	G0[4]	B0[1]
Data[0]	G0[3]	B0[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x1  
 DISPC\_DATA0\_CYCLE1 = 0x00000008  
 DISPC\_DATA0\_CYCLE2 = 0x00000008

	12-BPP	
	1st cycle	2nd cycle
Data[7]	R0[3]	x
Data[6]	R0[2]	x
Data[5]	R0[1]	x
Data[4]	R0[0]	x
Data[3]	G0[3]	B0[3]
Data[2]	G0[2]	B0[2]
Data[1]	G0[1]	B0[1]
Data[0]	G0[0]	B0[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x1  
 DISPC\_DATA0\_CYCLE1 = 0x00000008  
 DISPC\_DATA0\_CYCLE2 = 0x00000004

dispc-057

Figure 10-81. 9-bit Interface Settings

	24-BPP		
	1st cycle	2nd cycle	3rd cycle
Data[8]	R0[7]	G0[6]	x
Data[7]	R0[6]	G0[5]	x
Data[6]	R0[5]	G0[4]	x
Data[5]	R0[4]	G0[3]	B0[5]
Data[4]	R0[3]	G0[2]	B0[4]
Data[3]	R0[2]	G0[1]	B0[3]
Data[2]	R0[1]	G0[0]	B0[2]
Data[1]	R0[0]	B0[7]	B0[1]
Data[0]	G0[7]	B0[6]	B0[0]

DISPC\_CONTROLo.TDMCycleFormat = 0x2  
 DISPC\_DATAo\_CYCLE1 = 0x00000009  
 DISPC\_DATAo\_CYCLE2 = 0x00000009  
 DISPC\_DATAo\_CYCLE3 = 0x00000006

	18-BPP	
	1st cycle	2nd cycle
Data[8]	R0[5]	G0[2]
Data[7]	R0[4]	G0[1]
Data[6]	R0[3]	G0[0]
Data[5]	R0[2]	B0[5]
Data[4]	R0[1]	B0[4]
Data[3]	R0[0]	B0[3]
Data[2]	G0[5]	B0[2]
Data[1]	G0[4]	B0[1]
Data[0]	G0[3]	B0[0]

DISPC\_CONTROLo.TDMCYCLEFORMAT = 0x1  
 DISPC\_DATAo\_CYCLE1 = 0x00000009  
 DISPC\_DATAo\_CYCLE2 = 0x00000009

	16-BPP	
	1st cycle	2nd cycle
Data[8]	R0[4]	x
Data[7]	R0[3]	x
Data[6]	R0[2]	G0[1]
Data[5]	R0[1]	G0[0]
Data[4]	R0[0]	B0[4]
Data[3]	G0[5]	B0[3]
Data[2]	G0[4]	B0[2]
Data[1]	G0[3]	B0[1]
Data[0]	G0[2]	B0[0]

DISPC\_CONTROLo.TDMCYCLEFORMAT = 0x1  
 DISPC\_DATAo\_CYCLE1 = 0x00000009  
 DISPC\_DATAo\_CYCLE2 = 0x00000007

	12-BPP	
	1st cycle	2nd cycle
Data[8]	R0[3]	x
Data[7]	R0[2]	x
Data[6]	R0[1]	x
Data[5]	R0[0]	x
Data[4]	G0[3]	x
Data[3]	G0[2]	x
Data[2]	G0[1]	B0[2]
Data[1]	G0[0]	B0[1]
Data[0]	B0[3]	B0[0]

DISPC\_CONTROLo.TDMCYCLEFORMAT = 0x1  
 DISPC\_DATAo\_CYCLE1 = 0x00000009  
 DISPC\_DATAo\_CYCLE2 = 0x00000003

dispc-058

**Figure 10-82. 12-bit Interface Settings**

	24-BPP	
	1st cycle	2nd cycle
Data[11]	R0[7]	G0[3]
Data[10]	R0[6]	G0[2]
Data[9]	R0[5]	G0[1]
Data[8]	R0[4]	G0[0]
Data[7]	R0[3]	B0[7]
Data[6]	R0[2]	B0[6]
Data[5]	R0[1]	B0[5]
Data[4]	R0[0]	B0[4]
Data[3]	G0[7]	B0[3]
Data[2]	G0[6]	B0[2]
Data[1]	G0[5]	B0[1]
Data[0]	G0[4]	B0[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x1  
 DISPC\_DATA0\_CYCLE1 = 0x0000000C  
 DISPC\_DATA0\_CYCLE2 = 0x0000000C

	18-BPP		
	1st cycle	2nd cycle	3rd cycle
Data[11]	R0[5]	B0[5]	G1[5]
Data[10]	R0[4]	B0[4]	G1[4]
Data[9]	R0[3]	B0[3]	G1[3]
Data[8]	R0[2]	B0[2]	G1[2]
Data[7]	R0[1]	B0[1]	G1[1]
Data[6]	R0[0]	B0[0]	G1[0]
Data[5]	G0[5]	R1[5]	B1[5]
Data[4]	G0[4]	R1[4]	B1[4]
Data[3]	G0[3]	R1[3]	B1[3]
Data[2]	G0[2]	R1[2]	B1[2]
Data[1]	G0[1]	R1[1]	B1[1]
Data[0]	G0[0]	R1[0]	B1[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x3  
 DISPC\_DATA0\_CYCLE1 = 0x0000000C  
 DISPC\_DATA0\_CYCLE2 = 0x000060606  
 DISPC\_DATA0\_CYCLE3 = 0x000C0000

	16-BPP	
	1st cycle	2nd cycle
Data[11]	R0[4]	x
Data[10]	R0[3]	x
Data[9]	R0[2]	x
Data[8]	R0[1]	x
Data[7]	R0[0]	x
Data[6]	G0[5]	x
Data[5]	G0[4]	x
Data[4]	G0[3]	x
Data[3]	G0[2]	B0[3]
Data[2]	G0[1]	B0[2]
Data[1]	G0[0]	B0[1]
Data[0]	B0[4]	B0[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x1  
 DISPC\_DATA0\_CYCLE1 = 0x0000000C  
 DISPC\_DATA0\_CYCLE2 = 0x00000004

	12-BPP
	1st cycle
Data[11]	R0[3]
Data[10]	R0[2]
Data[9]	R0[1]
Data[8]	R0[0]
Data[7]	G0[3]
Data[6]	G0[2]
Data[5]	G0[1]
Data[4]	G0[0]
Data[3]	B0[3]
Data[2]	B0[2]
Data[1]	B0[1]
Data[0]	B0[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x0  
 DISPC\_DATA0\_CYCLE1 = 0x0000000C

dispc-059

Figure 10-83. 16-bit Interface Settings

	24-BPP		
	1st cycle	2nd cycle	3rd cycle
Data[15]	R0[7]	B0[7]	G1[7]
Data[14]	R0[6]	B0[6]	G1[6]
Data[13]	R0[5]	B0[5]	G1[5]
Data[12]	R0[4]	B0[4]	G1[4]
Data[11]	R0[3]	B0[3]	G1[3]
Data[10]	R0[2]	B0[2]	G1[2]
Data[9]	R0[1]	B0[1]	G1[1]
Data[8]	R0[0]	B0[0]	G1[0]
Data[7]	G0[7]	R1[7]	B1[7]
Data[6]	G0[6]	R1[6]	B1[6]
Data[5]	G0[5]	R1[5]	B1[5]
Data[4]	G0[4]	R1[4]	B1[4]
Data[3]	G0[3]	R1[3]	B1[3]
Data[2]	G0[2]	R1[2]	B1[2]
Data[1]	G0[1]	R1[1]	B1[1]
Data[0]	G0[0]	R1[0]	B1[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x3  
 DISPC\_DATA0\_CYCLE1 = 0x00000010  
 DISPC\_DATA0\_CYCLE2 = 0x00080808  
 DISPC\_DATA0\_CYCLE3 = 0x00100000

	18-BPP	
	1st cycle	2nd cycle
Data[15]	R0[5]	x
Data[14]	R0[4]	x
Data[13]	R0[3]	x
Data[12]	R0[2]	x
Data[11]	R0[1]	x
Data[10]	R0[0]	x
Data[9]	G0[5]	x
Data[8]	G0[4]	x
Data[7]	G0[3]	X
Data[6]	G0[2]	x
Data[5]	G0[1]	x
Data[4]	G0[0]	x
Data[3]	B0[5]	x
Data[2]	B0[4]	x
Data[1]	B0[3]	B0[1]
Data[0]	B0[2]	B0[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x1  
 DISPC\_DATA0\_CYCLE1 = 0x00000010  
 DISPC\_DATA0\_CYCLE2 = 0x00000002

	16-BPP
	1st cycle
Data[15]	R0[4]
Data[14]	R0[3]
Data[13]	R0[2]
Data[12]	R0[1]
Data[11]	R0[0]
Data[10]	G0[5]
Data[9]	G0[4]
Data[8]	G0[3]
Data[7]	G0[2]
Data[6]	G0[1]
Data[5]	G0[0]
Data[4]	B0[4]
Data[3]	B0[3]
Data[2]	B0[2]
Data[1]	B0[1]
Data[0]	B0[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x0  
 DISPC\_DATA0\_CYCLE1 = 0x00000010

	12-BPP
	1st cycle
Data[15]	x
Data[14]	x
Data[13]	x
Data[12]	x
Data[11]	R0[3]
Data[10]	R0[2]
Data[9]	R0[1]
Data[8]	R0[0]
Data[7]	G0[3]
Data[6]	G0[2]
Data[5]	G0[1]
Data[4]	G0[0]
Data[3]	B0[3]
Data[2]	B0[2]
Data[1]	B0[1]
Data[0]	B0[0]

DISPC\_CONTROL0.TDMCYCLEFORMAT = 0x0  
 DISPC\_DATA0\_CYCLE1 = 0x0000000C

dispc-060

### 10.2.4.12.6 Synchronized Buffer Update

A synchronization mismatch between the frame buffer and the display refreshes, named tearing effect, can lead to images that appear to be stretched on the screen. To avoid it, a synchronization mechanism is used between the DISPC and the process that updates the buffer. An interrupt is generated when the display reaches a predefined line number. The PROGRAMMEDLINENUMBER\_IRQ interrupt (DISPC\_IRQSTATUS and DISPC\_IRQENABLE) is a level signal and stays active during the programmed line of the display.

### 10.2.4.12.7 Timing Generator and Panel Settings

The size of a LCD panel is defined by:

- Number of lines, LPP field DISPC\_SIZE\_LCD0[26:16] with a value from 1 to 2048
- Number of pixels per line PPL field DISPC\_SIZE\_LCD0[10:0] with a value from 1 to 2048

Standard HSYNC/VSYNC timing generation are programmable for each LCD outputs independantly:

- Horizontal front porch is set in HFP field DISPC\_TIMING\_Ho[19:8]
- Horizontal back porch is set in HBP DISPC\_TIMING\_Ho[31:20]
- Horizontal synchronization pulse width is set in HSW field DISPC\_TIMING\_Ho[7:0]
- Vertical front porch is set in VFP field DISPC\_TIMING\_Vo[19:8]
- Vertical back porch is set in VBP field DISPC\_TIMING\_Vo[31:20]
- Vertical synchronization pulse width is set in VSW field DISPC\_TIMING\_Vo[7:0]

Horizontal, vertical synchronization and ACBIAS signals polarity are programmable by setting IVS, IHS and IEO bit DISPC\_POL\_FREQo[12][13][15]. These signals can be gated by setting VSYNCGATED and HSYNCGATED bit DISPC\_CONFIGo[7][6] respectively.

The latch of data can be driven on rising or falling edge of pixel clock by setting IPC bit DISPC\_POL\_FREQo[14]. The drive of HSYNC and VSYNC signal in function of the pixel clock is done by setting RF bit DISPC\_POL\_FREQo[16].

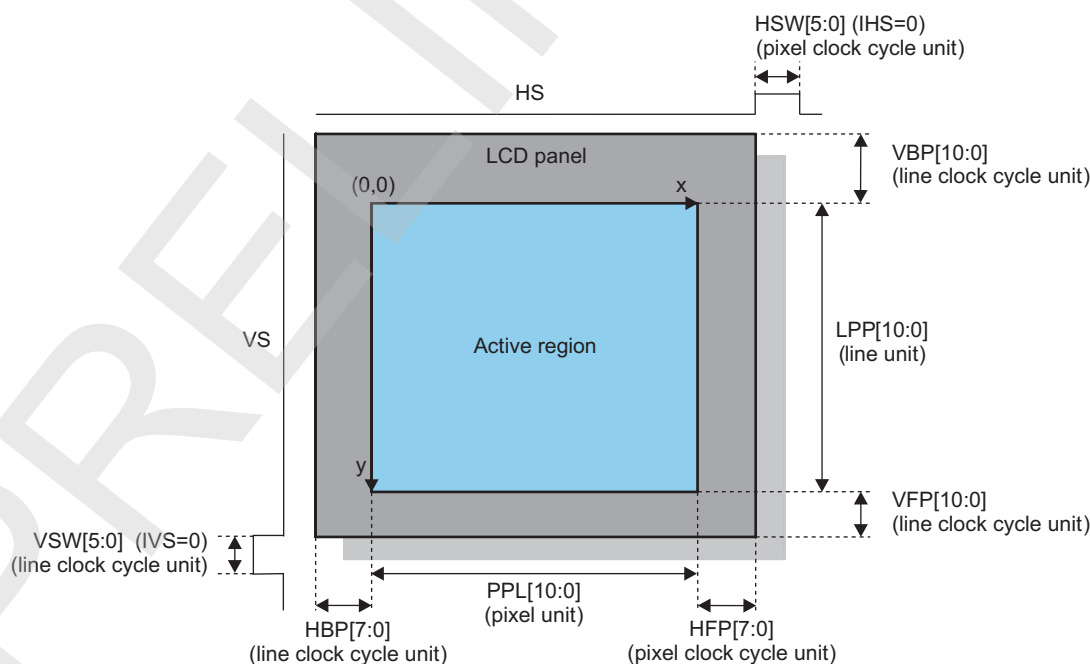
Table 10-54 describes the programming rules for LCD timing.

**Table 10-54. Programming Rules**

	No Downsampling	Downsampling H or V	Downsampling H + V
$(HBP + HSW + HFP) * PCD$	> 8	> 10	> 20

Figure 10-84 shows the timing values description in the case of an active matrix display.

**Figure 10-84. Timing Values (Active Matrix Display)**



dispc-031

The 8-bit pixel clock divider, PCD field DISPC\_DIVISOR[7:0], selects the pixel clock frequency. This bit field generates a range of pixel clock frequencies from LC/2 to LC/256, where LC is the logic clock from the divided functional clock of the display controller by the DISPC\_DIVISOR[23:16] LCD bit field.

The pixel clock is defined by the following equation:

$$\text{Pixel Clock} = (\text{FunctionalClock}/\text{LCD}[7:0])/\text{PCD}[7:0]$$

The pixel clock can be gated by setting to 0x1 PIXELCLOCKGATED bit DISPC\_CONFIGo[5].

The LCD output can be configured in progressive output or interlace output. The selection is done by writing into the bit field OUTPUTMODEENABLE bit DISPC\_CONFIGo[22]. The reset value is 0x0 which means progressive mode. When the progressive mode is selected, the FID signal associated to the LCD output is driven low (inactive state). The selection can be changed only if the corresponding LCD output is disabled. The configuration is independent for each LCD output.

When in interlaced mode, the FIDFIRST bit DISPC\_CONFIGo[23] indicates which field is outputted first:

- 0x0: even field first (FID=0)
- 0x1: odd field first (FID=1)

#### 10.2.4.12.8 Stall mode

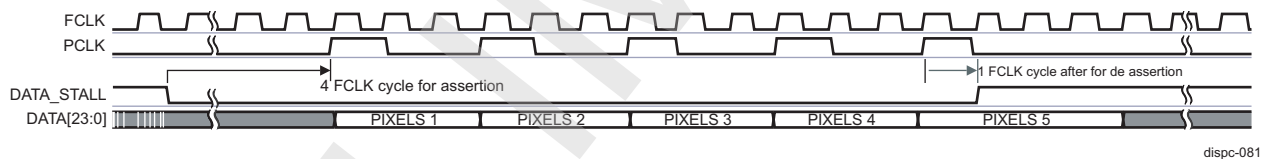
The stall mode is used to indicate when the DISPC has to stop sending data on the corresponding LCD output interface in order to avoid a overflow on the interface level. The stall mode is available when RFBI and DSI interface are used. The Interface asserts the stall signal to stop data output by the DISPC. It is deasserted by the interface to indicate when new data must be outputted by the display controller.

.Figure 10-85 details the RFBI data stall mode activated.

The STALL mode is selected by setting STALLMODE bit field DISPC\_CONTROLo[11].

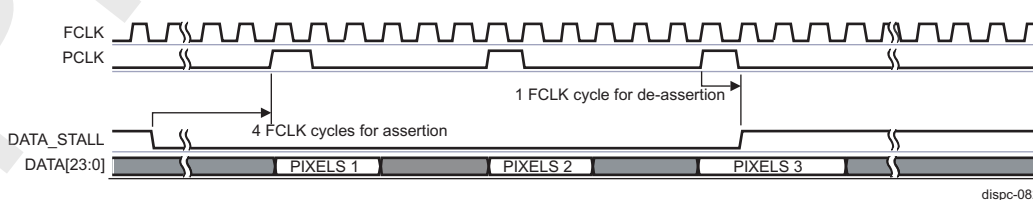
**NOTE:** The GOLCD bit DISPC\_CONTROLo[5] shall not be set but the DISPC configuration (DMA engine, pipelines associated to the LCD output) shall be set before enabling the LCD output by setting LCDENABLE bit field DISPC\_CONTROLo[1].

Figure 10-85. RFBI Data Stall Signal Diagram



To avoid an underflow of the DMA buffer, the DMA buffer handshake shall be enabled by setting BUFFERHANDCHECK bit DISPC\_CONFIGo[16]. The fullness of the buffers associated to the pipelines used for the LCD output is checked before providing data to the pipeline when the STALL signal is inactive. It prevents to empty the DMA buffer when the RFBI or DSI module requests data and there is not enough data in the DMA buffer of the DISPC. The feature shall be enabled only when the STALL mode is used (STALLMODE set to 0x1).When the DMA buffer handshake is activated, the pixel transfer to the RFBI or DSI module, outside a STALL period, can be stopped and restarted when there are enough data in the DMA buffer. The DMA buffer handshake ensures that the underflow does not occur for the pipelines associated with the LCD output when RFBI/DSI interfaces are used. The rate of the data transfer to the RFBI or DSI is, then, fully dependent on the state of the DMA buffer. Figure 10-86 details the RFBI data stall with FIFO handcheck mode activated.

Figure 10-86. RFBI Data Stall Signal Diagram With Handcheck



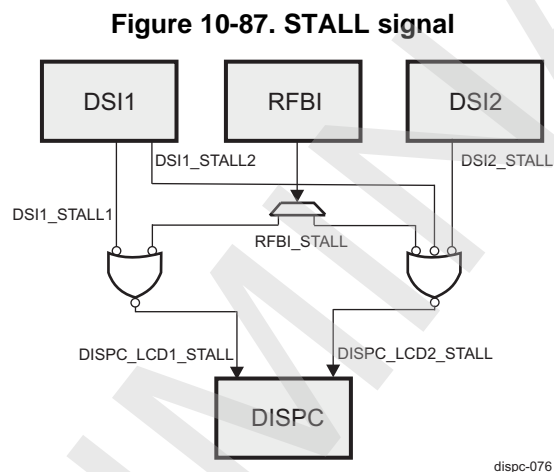
The stall mode is independently used when the RFBI or DSI protocol engines are connected to the LCD outputs and receive the pixels in order to perform re-formatting of the data.

The STALL signals from the RFBI and DSI interfaces are merged into a single signal in the Display Sub-System (DSS). Only one signal for each LCD output is connected the DISPC. [Figure 10-87](#) show the stall signal implementation.

One STALL signal is generated by the RFBI. The RFBI module asserts the stall signal when no more data has to be output by the DISPC. It is deasserted to indicate when new data have to be output by the display controller.

Two STALL signals are generated by DSI1 Protocol engine. It is used by the DSI1 protocol engine when data are provided from DISPC to DSI1 protocol engine using the two video ports in command mode only. If one of the two video port is in video mode, the STALL signal associated with the video port from the DSI1 protocol engine is not used (needs to be in inactive state from the DSI1 protocol engine IP).

One STALL signal is generated by DSI2 Protocol engine IP. It is used by the DSI2 protocol engine when data are provided from DISPC to DSI2 protocol engine using the single video port in command mode only. If the video port is in video mode, the STALL signal associated with the video port from the DSI2 protocol engine is not used (needs to be in inactive state from the DSI2 protocol engine IP).



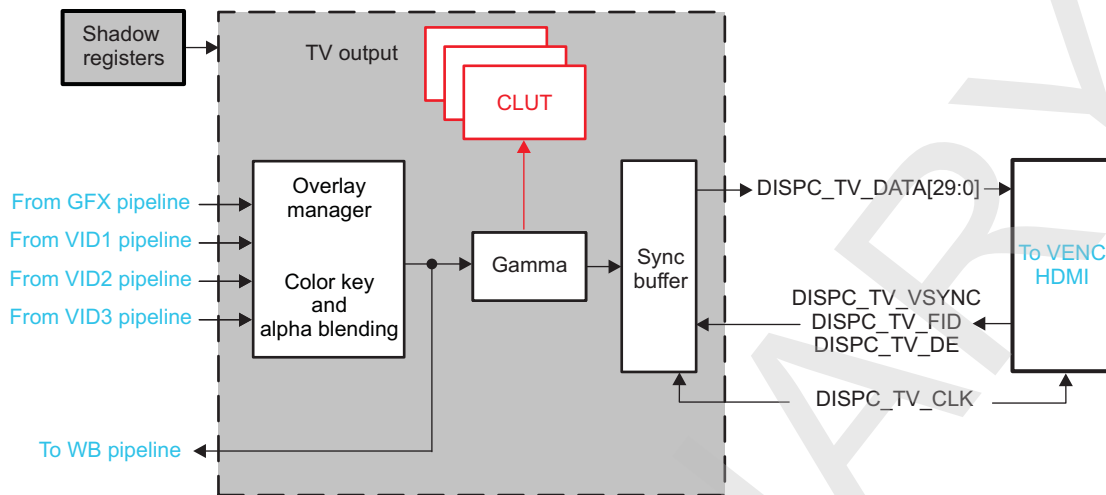
#### 10.2.4.13 TV Output

The TV output paths consist of several processing blocks, see [Figure 10-88](#):

- Overlay manager
- Gamma correction Unit
- Synchronization buffer



Figure 10-88. TV Output Architecture



dispc-029

The TV output is connected to Video Encoder or HDMI. In analog output or HDMI, the DISPC TV output receives an external clock, DISPC\_TV\_CLK, and based on the VSYNC generated by the VENC or HDMI, hold time, vertical offset and horizontal offset, outputs the pixels synchronously. The size of the field/frame to output defines the number of pixels to output on each line and the number of lines for each field/frame.

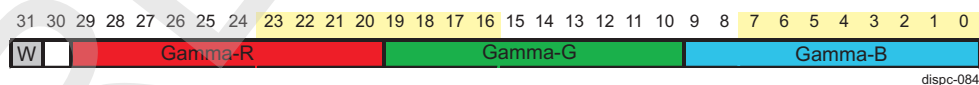
#### 10.2.4.13.1 Overlay Manager

The overlay mechanism is identical as in LCD1 and LCD2, refer to [Section 10.2.4.12.1](#)

#### 10.2.4.13.2 Gamma Correction Unit

The gamma correction unit works as described in [Section 10.2.4.12.2](#). The only difference resides in the input pixel format RGB30. Each component of encoded pixel value is used as a pointer to index one out of 1024 30-bit gamma curve entries in the table. Each 10-bit component is replaced with the 10-bit table value corresponding to R, G or B component. The table is loaded by software, through the DISPC\_GAMMA\_TABLE2 register. It is possible to load only part of the table. For each write access to the table, the 30-bit gamma value is associated with bit[31] INDEX to indicate that a new table is defined. Setting bit[31] INDEX to 0x1 for the first time, resets the internal index counter. All the following accesses are considered to be for incremented index addressing in the table (bit [31] INDEX set to 0 for each subsequent access). [Figure 10-89](#) describes the format of one of the gamma curve value in the memory.

Figure 10-89. Data Memory Organization for Gamma Mode in TV output

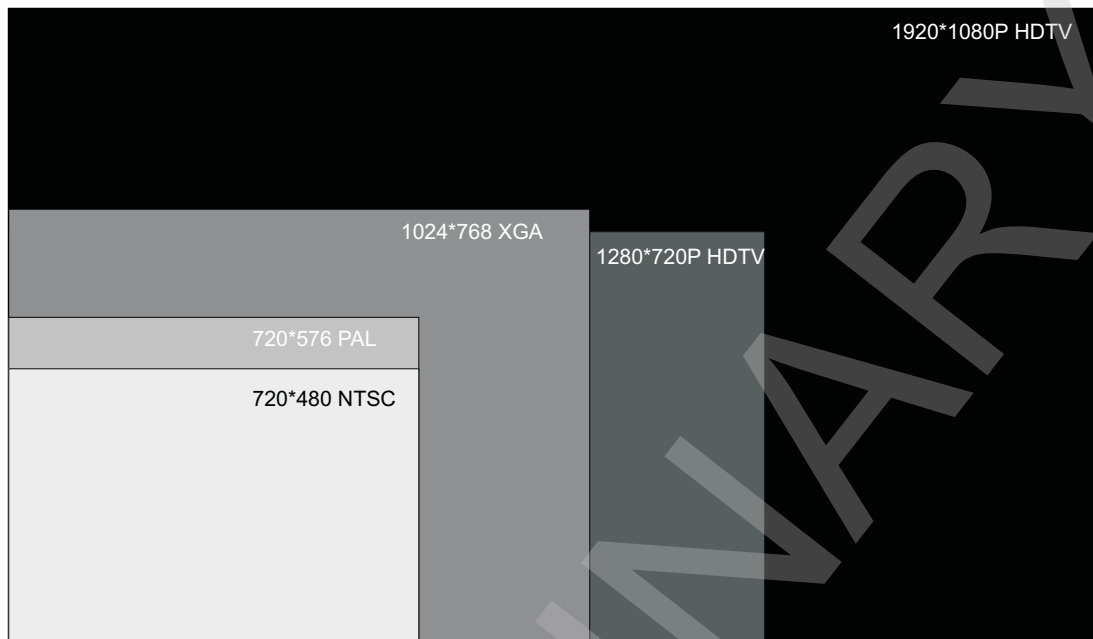


#### 10.2.4.13.3 Synchronized Buffer Update

The Synchronized Buffer Update is identical as in LCD1 and LCD2, refer to [Section 10.2.4.12.6](#).

#### 10.2.4.13.4 Timing and TV Format Settings

[Figure 10-90](#) represents the TV formats supported.

**Figure 10-90. TV Formats**

dispc-061

The size of a TV output format is defined by:

- Number of lines, LPP field [DISPC\\_SIZE\\_TV\[26:16\]](#) with a value from 1 to 2048
- Number of pixels per line PPL field [DISPC\\_SIZE\\_TV\[10:0\]](#) with a value from 1 to 2048

The hold time of the pixels on the data bus is determined in clock cycles by HT field [DISPC\\_CONTROL2\[19:17\]](#). The default value at reset time is 0x0 (1 cycle).

- When connected to the video NTSC/PAL encoder in field mode, [Table 10-56](#) indicates the [DISPC\\_SIZE\\_TV](#) (PPL and LPP) value for each standard PAL and NTSC.

**Table 10-55. PPL and LLP value for TV standard**

Standards		Active Pixels/Line	Active Lines	Digital Clock	<a href="#">DISPC_SIZE_TV</a>
ITU-R BT.601	NTSC(J, M, 4.43)	720	480	13.5MHz	0x00EF02CF
	PAL(B, D, G, H, I, N)	720	576	13.5MHz	0x011F02CF
	PAL(M)	720	480	13.5MHz	0x00EF02CF
Square Pixel	NTSC(J, M, 4.43)	640	480	12.2727MHz	0x00EF027F
	PAL(B, D, G, H, I, N)	768	576	14.75MHz	0x011F02FF
	PAL(M)	640	480	12.2727MHz	0x00EF027F

- When connected to the HDMI encoder, the [Table 10-56](#) indicates the [DISPC\\_SIZE\\_TV](#) (PPL and LPP) value for each HD standard.

**Table 10-56. PPL and LLP value for HD standard**

Standards		Active Pixels/Line	Active Lines	Digital Clock	<a href="#">DISPC_SIZE_TV</a>
HD standard	720p	1280	720	74.25 / 74.125 MHz (60/59.99..frames/s)	0x02CF04FF
	1080i	1920	540	74.25 / 74.125 MHz (60/59.99..frames/s)	0x021B07FF
	1080p	1920	1080	148.5 / 148.25 MHz (60/59.99..frames/s)	0x043707FF

### 10.2.4.14 Shadows Registers

Some display controller registers are termed *shadow registers*. A shadow register change has no direct effect on the configuration of the display controller. The registers are shadow registers in order to allow software to change the values of the registers at any time. When all the registers for a given configuration have been written into the registers, the software needs to set one bit only in order to validate the configuration. When the hardware reaches the end of the current frame and sees that the bit field has been set by software, the new configuration is now the configuration used by the hardware.

The bits enabling the hardware to use the new configuration are:

- [DISPC\\_CONTROL1](#)[5] GOLCD bit for all the registers associated to the LCD1 output, and for all registers of WB and DMA, if the LCD1 channel is captured. The update of the registers of the WB and DMA are further delayed by the [DISPC\\_WB\\_ATTRIBUTES2](#)[7:0] WBDELAYCOUNTER or done when the next frame is captured ([DISPC\\_WB\\_ATTRIBUTES](#)[26:24] CAPTUREMODE bit shall be different from 0). The update of the registers will occurs at the VFP start period.
- [DISPC\\_CONTROL1](#)[6] GOTV bit for all the registers associated to the TV output, and for all registers of WB and DMA, if the TV channel is captured. The update of the registers of the WB and DMA are further delayed by the [DISPC\\_WB\\_ATTRIBUTES2](#)[7:0] WBDELAYCOUNTER or done when the next frame is captured ([DISPC\\_WB\\_ATTRIBUTES](#)[26:24] CAPTUREMODE bit shall be different from 0). The update of the registers will occurs at the external EVSYNC.
- [DISPC\\_CONTROL2](#)[5] GOLCD bit for all the registers associated to the LCD2 output, and for all registers of WB and DMA, if the LCD2 channel is captured. The update of the registers of the WB and DMA are further delayed by the [DISPC\\_WB\\_ATTRIBUTES2](#)[7:0] WBDELAYCOUNTER or done when the next frame is captured ([DISPC\\_WB\\_ATTRIBUTES](#)[26:24] CAPTUREMODE bit shall be different from 0). The update of the registers will occurs at the VFP start period.
- [DISPC\\_WB\\_ATTRIBUTES](#)[0] ENABLE bit for all the registers associated to the WB, if the transfer memory-to-memory is not associated with a channel out.
- Both, [DISPC\\_CONTROL2](#)[6]GOWB bit, and [5]GOLCD and [6]GOTV bits in [DISPC\\_CONTROL1](#)/[DISPC\\_CONTROL2](#) registers, combined with the synchronization event of the channel output selected for write back. This applies to all registers associated with the selected channel out and further delayed by the WBDELAYCOUNTER, for all registers of the Write back and DMA.

**NOTE:** Before setting either the GOLCD, GOTV or GOWB bits, the user must ensure that the bits are cleared. The hardware resets the bit when the update is completed.

Table 10-57 lists the shadow registers.

**Table 10-57. Shadow Registers**

Shadow Register Name	Updated on VFP Start Period (LCD1 pipeline)	Updated on VFP Start Period (LCD2 pipeline)	Updated on External VSYNC (TV pipeline)	Updated on ENF of Frame (WB pipeline)
<a href="#">DISPC_REVISION</a>				
<a href="#">DISPC_SYSCONFIG</a>				
<a href="#">DISPC_SYSSTATUS</a>				
<a href="#">DISPC_IRQSTATUS</a>				
<a href="#">DISPC_IRQENABLE</a>				
<a href="#">DISPC_CONTROL1</a>	x		x	
<a href="#">DISPC_CONFIG1</a>	x	x	x	x
<a href="#">DISPC_DEFAULT_COLOR0</a>	x			
<a href="#">DISPC_DEFAULT_COLOR1</a>		x		
<a href="#">DISPC_TRANS_COLOR0</a>	x			
<a href="#">DISPC_TRANS_COLOR1</a>		x		

**Table 10-57. Shadow Registers (continued)**

Shadow Register Name	Updated on VFP Start Period (LCD1 pipeline)	Updated on VFP Start Period (LCD2 pipeline)	Updated on External VSYNC (TV pipeline)	Updated on ENF of Frame (WB pipeline)
DISPC_LINE_STATUS				
DISPC_LINE_NUMBER	x			
DISPC_TIMING_H1	x			
DISPC_TIMING_V1	x			
DISPC_POL_FREQ1	x			
DISPC_DIVISOR1	x			
DISPC_GLOBAL_ALPHA	x	x	x	x
DISPC_SIZE_TV			x	
DISPC_SIZE_LCD1	x			
DISPC_GFX_BA <sub>j</sub> <sup>(1)</sup>	x	x	x	x
DISPC_GFX_POSITION	x	x	x	x
DISPC_GFX_SIZE	x	x	x	x
DISPC_GFX_ATTRIBUTES	x	x	x	x
DISPC_GFX_BUF_THRESHOLD	x	x	x	x
DISPC_GFX_BUF_SIZE_STATUS	x	x	x	x
DISPC_GFX_ROW_INCREMENT	x	x	x	x
DISPC_GFX_PIXEL_INCREMENT	x	x	x	x
DISPC_GFX_WINDOW_SKIP	x	x	x	x
DISPC_GFX_TABLE_BA	x	x	x	x
DISPC_VID1_BA <sub>j</sub> <sup>(2)</sup>	x	x	x	x
DISPC_VID1_POSITION	x	x	x	x
DISPC_VID1_SIZE	x	x	x	x
DISPC_VID1_ATTRIBUTES	x	x	x	x
DISPC_VID1_BUF_THRESHOLD	x	x	x	x
DISPC_VID1_BUF_SIZE_STATUS	x	x	x	x
DISPC_VID1_ROW_INCREMENT	x	x	x	x
DISPC_VID1_PIXEL_INCREMENT	x	x	x	x
DISPC_VID1_FIR	x	x	x	x
DISPC_VID1_PICTURE_SIZE	x	x	x	x
DISPC_VID1_ACCU <sub>j</sub> <sup>(2)</sup>	x	x	x	x
DISPC_VID1_FIR_COEFF <sub>H_i</sub> <sup>(3)</sup>	x	x	x	x
DISPC_VID1_FIR_COEFF <sub>HV_i</sub> <sup>(3)</sup>	x	x	x	x

(1) j = 0 to 1

(2) j = 0 to 1

(3) i = 0 to 7

Table 10-57. Shadow Registers (continued)

Shadow Register Name	Updated on VFP Start Period (LCD1 pipeline)	Updated on VFP Start Period (LCD2 pipeline)	Updated on External VSYNC (TV pipeline)	Updated on ENF of Frame (WB pipeline)
DISPC_VID1_CONV_C OEF0	x	x	x	x
DISPC_VID1_CONV_C OEF1	x	x	x	x
DISPC_VID1_CONV_C OEF2	x	x	x	x
DISPC_VID1_CONV_C OEF3	x	x	x	x
DISPC_VID1_CONV_C OEF4	x	x	x	x
DISPC_VID2_BA_j <sup>(2)</sup>	x	x	x	x
DISPC_VID2_POSITION	x	x	x	x
DISPC_VID2_SIZE	x	x	x	x
DISPC_VID2_ATTRIBUTES	x	x	x	x
DISPC_VID2_BUF_THRESHOLD	x	x	x	x
DISPC_VID2_BUF_SIZE_STATUS	x	x	x	x
DISPC_VID2_ROW_IN_C	x	x	x	x
DISPC_VID2_PIXEL_IN_C	x	x	x	x
DISPC_VID2_FIR	x	x	x	x
DISPC_VID2_PICTURE_SIZE	x	x	x	x
DISPC_VID2_ACCU_j <sup>(4)</sup>	x	x	x	x
DISPC_VID2_FIR_COEF_H_i <sup>(5)</sup>	x	x	x	x
DISPC_VID2_FIR_COEF_HV_i <sup>(5)</sup>	x	x	x	x
DISPC_VID2_CONV_C OEF0	x	x	x	x
DISPC_VID2_CONV_C OEF1	x	x	x	x
DISPC_VID2_CONV_C OEF2	x	x	x	x
DISPC_VID2_CONV_C OEF3	x	x	x	x
DISPC_VID2_CONV_C OEF4	x	x	x	x
DISPC_DATA1_CYCLE 1	x	x	x	
DISPC_DATA1_CYCLE 2	x	x	x	
DISPC_DATA1_CYCLE 3	x	x	x	
DISPC_VID1_FIR_COEF_V_j <sup>(5)</sup>	x	x	x	x

<sup>(4)</sup> j = 0 to 1

<sup>(5)</sup> i = 0 to 7

**Table 10-57. Shadow Registers (continued)**

Shadow Register Name	Updated on VFP Start Period (LCD1 pipeline)	Updated on VFP Start Period (LCD2 pipeline)	Updated on External VSYNC (TV pipeline)	Updated on ENF of Frame (WB pipeline)
DISPC_VID2_FIR_COEF_V_i <sup>(5)</sup>	x	x	x	x
DISPC_CPR1_COEF_R	x			
DISPC_CPR1_COEF_G	x			
DISPC_CPR1_COEF_B	x			
DISPC_GFX_PRELOAD	x	x	x	x
DISPC_VID1_PRELOAD	x	x	x	x
DISPC_VID2_PRELOAD	x	x	x	x
DISPC_CONTROL2	x	x		
DISPC_VID3_ACCU_j <sup>(5)</sup>	x	x	x	x
DISPC_VID3_BA_j <sup>(5)</sup>	x	x	x	x
DISPC_VID3_FIR_COEF_H_i <sup>(5)</sup>	x	x	x	x
DISPC_VID3_FIR_COEF_HV_i <sup>(5)</sup>	x	x	x	x
DISPC_VID3_FIR_COEF_V_i <sup>(6)</sup>	x	x	x	x
DISPC_VID3_ATTRIBUTES	x	x	x	x
DISPC_VID3_CONV_COEF0	x	x	x	x
DISPC_VID3_CONV_COEF1	x	x	x	x
DISPC_VID3_CONV_COEF2	x	x	x	x
DISPC_VID3_CONV_COEF3	x	x	x	x
DISPC_VID3_CONV_COEF4	x	x	x	x
DISPC_VID3_BUF_SIZE_STATUS	x	x	x	x
DISPC_VID3_BUF_THRESHOLD	x	x	x	x
DISPC_VID3_FIR	x	x	x	x
DISPC_VID3_PICTURE_SIZE	x	x	x	x
DISPC_VID3_PIXEL_IN_C	x	x	x	x
DISPC_VID3_POSITION	x	x	x	x
DISPC_VID3_PRELOAD	x	x	x	x
DISPC_VID3_ROW_IN_C	x	x	x	x
DISPC_VID3_SIZE	x	x	x	x
DISPC_DEFAULT_COLOR2		x		
DISPC_TRANS_COLOR2		x		

<sup>(6)</sup> i = 0 to 7

Table 10-57. Shadow Registers (continued)

Shadow Register Name	Updated on VFP Start Period (LCD1 pipeline)	Updated on VFP Start Period (LCD2 pipeline)	Updated on External VSYNC (TV pipeline)	Updated on ENF of Frame (WB pipeline)
DISPC_CPR2_COEF_B		X		
DISPC_CPR2_COEF_G		X		
DISPC_CPR2_COEF_R		X		
DISPC_DATA2_CYCLE 1		X		
DISPC_DATA2_CYCLE 2		X		
DISPC_DATA2_CYCLE 3		X		
DISPC_SIZE_LCD2		X		
DISPC_TIMING_H2		X		
DISPC_TIMING_V2		X		
DISPC_POL_FREQ2		X		
DISPC_DIVISOR2		X		
DISPC_WB_ACCU_j <sup>(7)</sup>	X	X	X	X
DISPC_WB_BA_j <sup>(7)</sup>	X	X	X	X
DISPC_WB_FIR_COEF_H_i <sup>(8)</sup>	X	X	X	X
DISPC_WB_FIR_COEF_HV_i <sup>(8)</sup>	X	X	X	X
DISPC_WB_FIR_COEF_V_i <sup>(8)</sup>	X	X	X	X
DISPC_WB_ATTRIBUTES	X	X	X	X
DISPC_WB_CONV_COEF0	X	X	X	X
DISPC_WB_CONV_COEF1	X	X	X	X
DISPC_WB_CONV_COEF2	X	X	X	X
DISPC_WB_CONV_COEF3	X	X	X	X
DISPC_WB_CONV_COEF4	X	X	X	X
DISPC_WB_BUF_SIZE_STATUS	X	X	X	X
DISPC_WB_BUF_THRESHOLD	X	X	X	X
DISPC_WB_FIR	X	X	X	X
DISPC_WB_PICTURE_SIZE	X	X	X	X
DISPC_WB_PIXEL_INC	X	X	X	X
DISPC_WB_ROW_INC	X	X	X	X
DISPC_WB_SIZE	X	X	X	X
DISPC_VID1_BA_UV_j <sup>(9)</sup>	X	X	X	X
DISPC_VID2_BA_UV_j <sup>(9)</sup>	X	X	X	X

<sup>(7)</sup> j = 0 to 1

<sup>(8)</sup> i = 0 to 7

<sup>(9)</sup> j = 0 to 1

**Table 10-57. Shadow Registers (continued)**

Shadow Register Name	Updated on VFP Start Period (LCD1 pipeline)	Updated on VFP Start Period (LCD2 pipeline)	Updated on External VSYNC (TV pipeline)	Updated on ENF of Frame (WB pipeline)
DISPC_VID3_BA_UV_j <sup>(9)</sup>	X	X	X	X
DISPC_WB_BA_UV_j <sup>(9)</sup>	X	X	X	X
DISPC_CONFIG2		X		
DISPC_VID1_ATTRIBUTES2	X	X	X	X
DISPC_VID2_ATTRIBUTES2	X	X	X	X
DISPC_VID3_ATTRIBUTES2	X	X	X	X
DISPC_GAMMA_TABLE0	X	X	X	X
DISPC_GAMMA_TABLE1	X	X	X	X
DISPC_GAMMA_TABLE2	X	X	X	X
DISPC_VID1_FIR2	X	X	X	X
DISPC_VID1_ACCU2_j <sup>(9)</sup>	X	X	X	X
DISPC_VID1_FIR_COEFF_H2_i <sup>(8)</sup>	X	X	X	X
DISPC_VID1_FIR_COEFF_HV2_i <sup>(10)</sup>	X	X	X	X
DISPC_VID1_FIR_COEFF_V2_i <sup>(10)</sup>	X	X	X	X
DISPC_VID2_FIR2	X	X	X	X
DISPC_VID2_ACCU2_j <sup>(11)</sup>	X	X	X	X
DISPC_VID2_FIR_COEFF_H2_i <sup>(10)</sup>	X	X	X	X
DISPC_VID2_FIR_COEFF_HV2_i <sup>(10)</sup>	X	X	X	X
DISPC_VID2_FIR_COEFF_V2_i <sup>(10)</sup>	X	X	X	X
DISPC_VID3_FIR2	X	X	X	X
DISPC_VID3_ACCU2_j <sup>(11)</sup>	X	X	X	X
DISPC_VID3_FIR_COEFF_H2_i <sup>(10)</sup>	X	X	X	X
DISPC_VID3_FIR_COEFF_HV2_i <sup>(10)</sup>	X	X	X	X
DISPC_VID3_FIR_COEFF_V2_i <sup>(10)</sup>	X	X	X	X
DISPC_WB_FIR2	X	X	X	X
DISPC_WB_ACCU2_j <sup>(11)</sup>	X	X	X	X
DISPC_WB_FIR_COEFF_H2_i <sup>(10)</sup>	X	X	X	X
DISPC_WB_FIR_COEFF_HV2_i <sup>(10)</sup>	X	X	X	X
DISPC_WB_FIR_COEFF_V2_i <sup>(10)</sup>	X	X	X	X

<sup>(10)</sup> i = 0 to 7<sup>(11)</sup> j = 0 to 1



**Table 10-57. Shadow Registers (continued)**

Shadow Register Name	Updated on VFP Start Period (LCD1 pipeline)	Updated on VFP Start Period (LCD2 pipeline)	Updated on External VSYNC (TV pipeline)	Updated on ENF of Frame (WB pipeline)
DISPC_GLOBAL_BUFFER	x	x	x	x
DISPC_DIVISOR	x	x	x	x

## 10.2.5 Display Controller Programming Guide

### 10.2.5.1 Display Controller Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

#### 10.2.5.1.1 Global Initialization

##### 10.2.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the DISPC module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DISPC. Refer to the DISPC Module Integration and Environment Sections for further information.

**Table 10-58. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. See the "module clocks enable programming model" in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Control Module	Module specific pad muxing must be set in the Control Module. Set the pad mux control by configuring CONTROL.CONTROL_PADCONF_<Pad> [2:0] MUXMODE = 0x1.  Module specific pad pull-up must be set in the Control Module. Set the pad pull-up by configuring CONTROL.CONTROL_PADCONF_<Pad> [4] PULLTYPESELECT = 0x1
MPU INTC or DSP INTC	MPU and/or DSP interrupt controller configuration must be done to enable the interrupts from DISPC module. See the "module interrupt configuration programming model" in <a href="#">Chapter 17, Interrupt Controller</a> .
Interconnect	See <a href="#">Chapter 13, Interconnect</a> for more information about the interconnect configuration.

### 10.2.5.1.2 Operational Modes Configuration

**Table 10-59. DISPC Configuration**

Step	Register / Bit Field / Programming Model
<b>For a GFX pipeline configuration</b>	
Configure the GFX DMA channel	Refer to <a href="#">Table 10-60</a>
Configure the GFX pipeline	Refer to <a href="#">Table 10-63</a>
Configure the LCD or TV output	For LCD output, refer to <a href="#">Table 10-81</a> For TV output, refer to <a href="#">Table 10-87</a>
<b>For a Video pipeline configuration</b>	
Configure the video DMA channel	Refer to <a href="#">Table 10-61</a>
Configure the video pipeline	Refer to <a href="#">Table 10-68</a>
Configure the LCD or TV output	For LCD output, refer to <a href="#">Table 10-81</a> For TV output, refer to <a href="#">Table 10-87</a>
<b>For a WB pipeline configuration</b>	
Configure the WB DMA channel	Refer to <a href="#">Table 10-62</a>
Configure the WB pipeline	For video pipelines, refer to <a href="#">Table 10-76</a>

#### 10.2.5.1.2.1 DMA Configuration

##### 10.2.5.1.2.1.1 Main-sequence - DISPC DMA Channel Configuration

This subsequence describes the parameters of the GFX, video or WB DMA channel parameters.

**Table 10-60. Configure the GFX DMA Channel**

Step	Register / Bit Field / Programming Model	Value
Set the base address for RGB pixel format or BITMAP component format according to memory access type, rotation, mirroring (refer to section DMA)	DISPC_GFX_BA_j[31:0] BA	0x---
Set the rotation flag	DISPC_GFX_ATTRIBUTES[13:12] ROTATION	0x--
Set the number of bytes to increment at the end of the row	DISPC_GFX_ROW_INC[31:0] ROWINC	0x--
Set the number of bytes to increment between two pixels	DISPC_GFX_PIXEL_INC[7:0] PIXELINC	0x--
Determine the FIFO preload mode	DISPC_GFX_ATTRIBUTES[11] BUFPRELOAD	0x--
Set the preload value	DISPC_GFX_PRELOAD[11:0] PRELOAD	0x--
Determine the burst type	DISPC_GFX_ATTRIBUTES[29] BURSTTYPE	0x--
Set the burst size	DISPC_GFX_ATTRIBUTES[7:6] BURSTSIZE	0x--
Set the high level of DMA FIFO threshold	DISPC_GFX_BUF_THRESHOLD[31:16] BUFHIGHTRESHOLD	0x--
Set the low level of DMA FIFO threshold	DISPC_GFX_BUF_THRESHOLD[15:0] BUFLOWTRESHOLD	0x--
Enable Selfrefresh	DISPC_GFX_ATTRIBUTES[24] SELFREFRESH	0x--
Select priority over the other pipeline	DISPC_GFX_ATTRIBUTES[23] ARBITRATION	0x--

**Table 10-61. Configure the Video DMA Channel**

Step	Register / Bit Field / Programming Model	Value
Set the base address for RGB pixel format or Y component format according to memory access type, rotation, mirroring (refer to section DMA)	DISPC_VIDp_BA_j[31:0] BA	0x---

**Table 10-61. Configure the Video DMA Channel (continued)**

Step	Register / Bit Field / Programming Model	Value
Set the base address for Cb and Cr components according to memory access type, rotation, mirroring (refer to section DMA) <sup>(1)</sup>	DISPC_VIDp_BA_UV_j[31:0] BA	0x---
Set the rotation flag	DISPC_VIDp_ATTRIBUTES[13:12] ROTATION	0x--
Set the number of bytes to increment at the end of the row	DISPC_VIDp_ROW_INC[31:0] ROWINC	0x--
Set the number of bytes to increment between two pixels	DISPC_VIDp_PIXEL_INC[7:0] PIXELINC	0x--
Set the X original image size	DISPC_VIDp_PICTURE_SIZE[10:0] ORGSIZEX	0x--
Set the Y original image size	DISPC_VIDp_PICTURE_SIZE[26:16] ORGSIZEY	0x--
Determine the FIFO preload mode	DISPC_VIDp_ATTRIBUTES[19] BUFPRELOAD	0x--
Set the preload value	DISPC_VIDp_PRELOAD[11:0] PRELOAD	0x--
Determine the burst type	DISPC_VIDp_ATTRIBUTES[29] BURSTTYPE	0x--
Set the burst size	DISPC_VIDp_ATTRIBUTES[15:14] BURSTSIZE	0x--
Set the high level of DMA FIFO threshold	DISPC_VIDp_BUF_THRESHOLD[31:16] BUFHIGHTRESHOLD	0x--
Set the low level of DMA FIFO threshold	DISPC_VIDp_BUF_THRESHOLD[15:0] BUFLOWTRESHOLD	0x--
Enable Selfrefresh	DISPC_VIDp_ATTRIBUTES[24] SELFREFRESH	0x--
Select priority over the other pipeline	DISPC_VIDp_ATTRIBUTES[23] ARBITRATION	0x--

<sup>(1)</sup> Applicable only for YUV pixel format

**Table 10-62. Configure the WB DMA Channel**

Step	Register / Bit Field / Programming Model	Value
Set the base address for RGB pixel format or Y component format according to memory access type, rotation, mirroring (refer to section DMA)	DISPC_WB_BA_j[31:0] BA	0x---
Set the base address for Cb and Cr components according to memory access type, rotation, mirroring (refer to section DMA) <sup>(1)</sup>	DISPC_WB_BA_UV_j[31:0] BA	0x---
Set the stride of CbCr component <sup>(2)</sup>	DISPC_WB_ATTRIBUTES[22] DOUBLESTRIDE	0x--
Set the rotation flag	DISPC_WB_ATTRIBUTES[13:12] ROTATION	0x--
Set the number of bytes to increment at the end of the row	DISPC_WB_ROW_INC[31:0] ROWINC	0x--
Set the number of bytes to increment between two pixels	DISPC_WB_PIXEL_INC[7:0] PIXELINC	0x--
Set the X final image size in system memory	DISPC_WB_PICTURE_SIZE[10:0] ORGSIZEX	0x--
Set the Y final image size in system memory	DISPC_WB_PICTURE_SIZE[26:16] ORGSIZEY	0x--
Set the burst size	DISPC_WB_ATTRIBUTES[15:14] BURSTSIZE	0x--
Set the high level of DMA FIFO threshold	DISPC_WB_BUF_THRESHOLD[31:16] BUFHIGHTRESHOLD	0x--
Set the low level of DMA FIFO threshold	DISPC_WB_BUF_THRESHOLD[15:0] BUFLOWTRESHOLD	0x--
Select priority over the other pipeline	DISPC_WB_ATTRIBUTES[23] ARBITRATION	0x--

<sup>(1)</sup> Applicable only for YUV pixel format

<sup>(2)</sup> Applicable only for YUV pixel format

### 10.2.5.1.2.2 GFX Pipeline Configuration

#### 10.2.5.1.2.2.1 Main Sequence - Configure the GFX Pipeline

This procedure details the steps for a GFX pipeline configuration described in [Table 10-63](#)

**Table 10-63. Configure the GFX Pipeline**

Step	Register / Bit Field / Programming Model	Value
Configure the GFX Window	Refer to <a href="#">Table 10-64</a>	
Configure the GFX pipeline processing	Refer to <a href="#">Table 10-65</a>	
Configure the GFX pipeline layer output	Refer to <a href="#">Table 10-67</a>	
Validate the GFX configuration according to outputs associated to the pipeline	<a href="#">DISPC_CONTROL1</a> [5] GOLCD	
	<a href="#">DISPC_CONTROL2</a> [5] GOLCD	
	<a href="#">DISPC_CONTROL2</a> [6] GOWB	
	<a href="#">DISPC_CONTROL1</a> [6] GOTV	
Enable the GFX pipeline	<a href="#">DISPC_GFX_ATTRIBUTES</a> [0] ENABLE	0x1

#### 10.2.5.1.2.2.2 Subsequence - Configure the GFX Window

This subsequence describes the parameters of the image to be displayed on the LCD panel.

**Table 10-64. Configure the GFX Window**

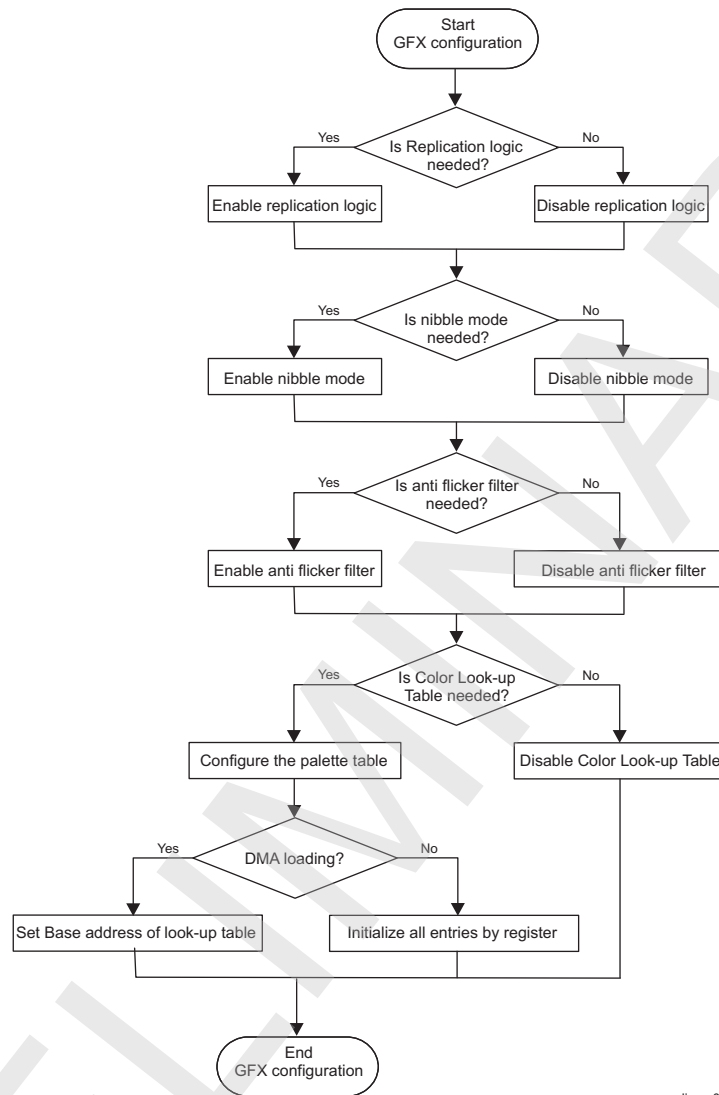
Step	Register / Bit Field / Programming Model	Value
Select the format of image	<a href="#">DISPC_GFX_ATTRIBUTES</a> [4:1] FORMAT	0x--
Set the X size of image to be displayed onto LCD panel	<a href="#">DISPC_GFX_SIZE</a> [10:0] SIZEX	0x--
Set the Y size of image to be displayed onto LCD panel	<a href="#">DISPC_GFX_SIZE</a> [10:0] SIZEY	0x--
Set the X position of image in respect to LCD panel	<a href="#">DISPC_GFX_POSITION</a> [10:0] POSX	0x--
Set the Y position of image in respect to LCD panel	<a href="#">DISPC_GFX_POSITION</a> [26:16] POSY	0x--

#### 10.2.5.1.2.2.3 Subsequence - Configure the GFX Pipeline Processing

**Table 10-65. Configure the GFX Pipeline Processing**

Step	Register / Bit Field / Programming Model	Value
Enable replication logic	<a href="#">DISPC_GFX_ATTRIBUTES</a> [5] REPLICATIONENABLE	0x1
Enable nibble mode	<a href="#">DISPC_GFX_ATTRIBUTES</a> [9] NIBBLEMODE	0x1
Enable anti flicker filter	<a href="#">DISPC_GFX_ATTRIBUTES</a> [24] ANTIFLICKER	0x1
Configure the palette table	Refer to <a href="#">Table 10-66</a>	0x--

Figure 10-91. Video Pipeline Processing Configuration



dispc-091

10.2.5.1.2.2.4 Subsequence - Configure the Palette Table

This subsequence resumes the settings for configuring the the Color Look-Up table as palette for GFX pipeline,

NOTE:

- It is software responsibility to ensure that there is no visible effect when modifying the table since it is not under hardware control and the synchronization done using the DMA engine inside the DISPC to load the table when it is not used for displaying the picture on the screen is not present for this mode.
- If GFX pipeline is in BITMAP pixel format, Gamma corrections is not available for LCD1.

Table 10-66. Configure the Palette Table

Step	Register / Bit Field / Programming Model	Value
Set look up table as palette table	DISPC_CONFIG1[3] PALETTEGAMMATALE	0x0
Select the load mode of the palette look-up table	DISPC_CONFIG1[2:1] LOADMODE	0x--

**Table 10-66. Configure the Palette Table (continued)**

Step	Register / Bit Field / Programming Model	Value
DMA loading		
Set base address of look-up table	DISPC_GFX_TABLE_BA[31:0] TABLEBA	0x--
Initialize all entries by register		
Initialize all entries for the palette table by setting the table index and the RGB values associated to this index	DISPC_GAMMA_TABLE0[31:24][23:16][15:8][7:0], INDEX, VALUE_R, VALUE_G, VALUE_B	0x--

#### 10.2.5.1.2.2.5 Subsequence - Configure the GFX Pipeline Layer Output

This subsequence resume the video layer settings available at the pipeline level necessary when using the overlay manager.

**Table 10-67. Configure the GFX Pipeline Layer Output**

Step	Register / Bit Field / Programming Model	Value
Set the LCD/TV output	DISPC_GFX_ATTRIBUTES[8] CHANNELOUT	0x--
If DISPC_GFX_ATTRIBUTES[8]=0, set the LCD output	DISPC_GFX_ATTRIBUTES[31:30] CHANNELOUT2	0x--
Set the Z-order priority of the layer for overlay manager	DISPC_GFX_ATTRIBUTES[27:26] ZORDER	0x--
Enable the video pipeline Z-order	DISPC_GFX_ATTRIBUTES[25] ZORDERENABLE	0x--
Set the Global Alpha value for the Alpha blender unit	DISPC_GLOBAL_ALPHA[7:0] GFXGLOBALALPHA	0x--

#### 10.2.5.1.2.3 Video Pipeline Configuration

##### 10.2.5.1.2.3.1 Main Sequence - Configure the Video Pipeline

This procedure details the steps for a Video pipeline configuration described in [Table 10-68](#).

**Table 10-68. Configure the Video Pipeline**

Step	Register / Bit Field / Programming Model	Value
Configure the video window	Refer to <a href="#">Table 10-69</a>	
Configure the video pipeline processing	Refer to <a href="#">Table 10-70</a>	
Configure the video pipeline layer output	Refer to <a href="#">Table 10-75</a>	
Validate the video configuration according to outputs associated to the pipeline	DISPC_CONTROL1[5] GOLCD	
	DISPC_CONTROL2[5] GOLCD	
	DISPC_CONTROL2[6] GOWB	
	DISPC_CONTROL1[6] GOTV	
Enable video pipeline	DISPC_VIDp_ATTRIBUTES[0] ENABLE	0x1

##### 10.2.5.1.2.3.2 Subsequence - Configure the Video Window

This subsequence describes the parameters of the image to be displayed on the LCD panel.

**Table 10-69. Configure the Video Window**

Step	Register / Bit Field / Programming Model	Value
Select the format of image	DISPC_VIDp_ATTRIBUTES[4:1] FORMAT	0x--
Set the X size of image to be displayed onto LCD panel	DISPC_VIDp_SIZE[10:0] SIZEX	0x--
Set the Y size of image to be displayed onto LCD panel	DISPC_VIDp_SIZE[10:0] SIZEY	0x--

**Table 10-69. Configure the Video Window (continued)**

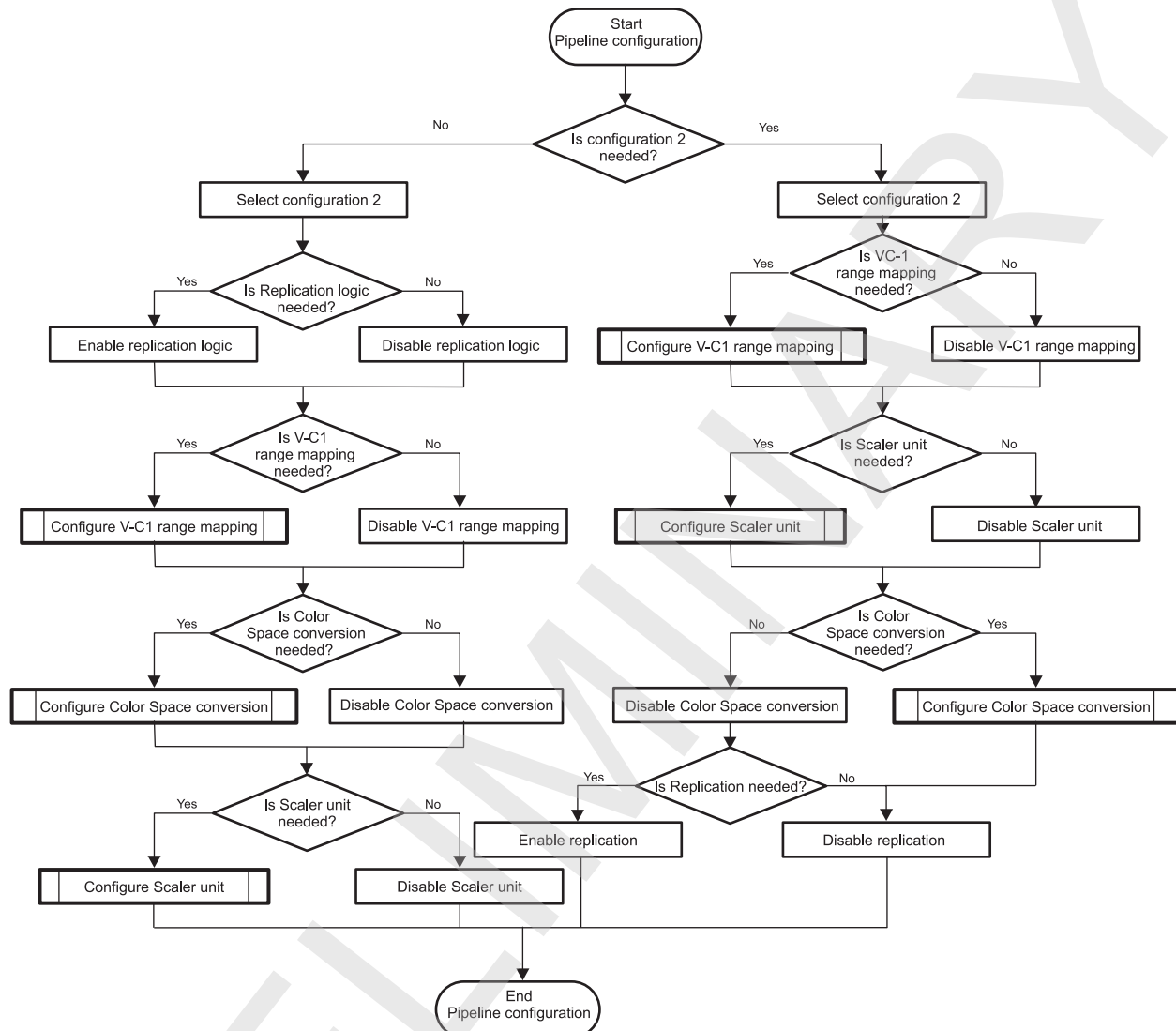
Step	Register / Bit Field / Programming Model	Value
Set the X position of image in respect to LCD panel	DISPC_VIDp_POSITION[10:0] POSX	0x--
Set the Y position of image in respect to LCD panel	DISPC_VIDp_POSITION[26:16] POSY	0x--

### 10.2.5.1.2.3.3 Subsequence -Configure the Video Pipeline Processing

**Table 10-70. Configure the Video Pipeline Processing**

Step	Register / Bit Field / Programming Model	Value
Select the video pipeline configuration 1 or 2	DISPC_VIDp_ATTRIBUTES2[8] YUVCHROMARESAMPLING	0x--
Enable/disable Replication Logic <sup>(1)</sup>	DISPC_VIDp_ATTRIBUTES[10] REPLICATIONENABLE	0x1
Configure the VC-1 range mapping <sup>(1)</sup>	Refer to <a href="#">Section 10.2.5.1.2.3.4</a>	
Configure the video Color Space Conversion <sup>(1)</sup>	Refer to <a href="#">Section 10.2.5.1.2.3.5</a>	
Configure the video Scaler Unit <sup>(1)</sup>	Refer to <a href="#">Section 10.2.5.1.2.3.6</a>	

- <sup>(1)</sup> This module configuration can be optional depending on:
- The video mode configuration selected, refer to [Figure 10-92](#)
  - The video format and application needs

**Figure 10-92. Video Pipeline Processing Configuration**

dispc-089

**10.2.5.1.2.3.4 Subsequence - Configure the VC-1 Range Mapping****Table 10-71. Configure the VC-1 Range Mapping**

Step	Register / Bit Field / Programming Model	Value
Set the Y component VC-1 range mapping	DISPC_VIDp_ATTRIBUTES2[3:1] VC1_RANGE_Y	0x--
Set the Cb and Cr component VC-1 range mapping	DISPC_VIDp_ATTRIBUTES2[6:4] VC1_RANGE_CBCR	0x--
Enable VC-1 range mapping	DISPC_VIDp_ATTRIBUTES2[0] VC1ENABLE	0x1

**10.2.5.1.2.3.5 Subsequence - Configure the Video Color Space Conversion****Table 10-72. Configure the Video Color Space Conversion**

Step	Register / Bit Field / Programming Model	Value
Select the range of the Color Sapce Conversion	DISPC_VIDp_ATTRIBUTES[11] FULLRANGE	0x--
Set the RCr and RY coefficients	DISPC_VIDp_CONV_COEF0[26:16][10:0] RCR, RY	0x--



**Table 10-72. Configure the Video Color Space Conversion (continued)**

Step	Register / Bit Field / Programming Model	Value
Set the GY and RCb coefficients	DISPC_VIDp_CONV_COEF1[26:16][10:0] GY, RCb	0x--
Set the GCb and GCr coefficients	DISPC_VIDp_CONV_COEF2[26:16][10:0] GCB, GCR	0x--
Set the BCr and BY coefficients	DISPC_VIDp_CONV_COEF3[26:16][10:0] BCR, BY	0x--
Set the BCb coefficient	DISPC_VIDp_CONV_COEF4[10:0] BCB	0x--
Enable Color Space Conversion	DISPC_VIDp_ATTRIBUTES[9] COLORCONVENABLE	0x1

### 10.2.5.1.2.3.6 Subsequence - Configure the Video Scaler Unit

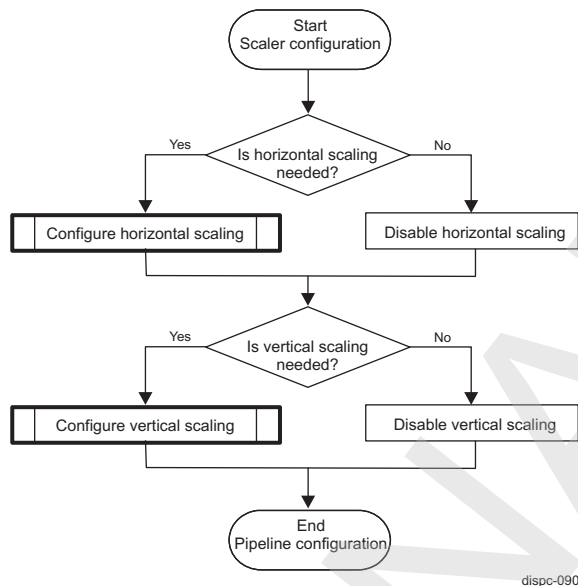
This subsequence configures the video scaler unit. [Table 10-73](#) is applicable for RGB pixel format. [Table 10-73](#) and [Table 10-74](#) are applicable for YUV pixel format.

**Table 10-73. Configure the Video Scaler Unit for RGB Pixel Formats or Y Component**

Step	Register / Bit Field / Programming Model	Value
Configure horizontal scaling		
Set the horizontal resizing ratio	DISPC_VIDp_FIR[12:0] FIRHINC	0x--
Set the horizontal FIR coefficients	DISPC_VIDp_FIR_COEF_H_i[31:24][23:16][15:8][7:0] FIRHC3, FIRHC2, FIRHC1, FIRHC0 DISPC_VIDp_FIR_COEF_HV_i[7:0] FIRHC4	0x-- 0x--
Set the horizontal accumulators value	DISPC_VIDp_ACCU_j[10:0] HORIZONTALACCU	0x--
Configure vertical scaling		
Select number of vertical taps	DISPC_VIDp_ATTRIBUTES[21] VERTICALTAPS	0x--
Set the vertical resizing ratio	DISPC_VIDp_FIR[28:16] FIRVINC	0x--
Set the vertical FIR coefficients for RGB pixel format or Y component	DISPC_VIDp_FIR_COEF_HV_i[31:24][23:16][15:8] FIRVC2, FIRVC1, FIRVC0 Only for 5-taps vertical: DISPC_VIDp_FIR_COEF_V_i[15:8][7:0] FIRVC22, FIRVC00	0x-- 0x--
Set the vertical accumulators value for RGB pixel format or Y component	DISPC_VIDp_ACCU_j[26:16] VERTICALACCU	0x--
Enable horizontal and vertical Scaler Unit	DISPC_VIDp_ATTRIBUTES[6:5] RESIZEENABLE	0x--

**Table 10-74. Configure the Video Scaler Unit for Cb and Cr Components**

Step	Register / Bit Field / Programming Model	Value
Configure horizontal scaling		
Set the horizontal resizing ratio for Cb and Cr components	DISPC_VIDp_FIR2[12:0] FIRHINC	0x--
Set the horizontal FIR coefficients for Cb and Cr components	DISPC_VIDp_FIR_COEF_H2_i[31:24][23:16][15:8][7:0] FIRHC3, FIRHC2, FIRHC1, FIRHC0 DISPC_VIDp_FIR_COEF_HV2_i[7:0] FIRHC4	0x-- 0x--
Set the horizontal accumulators value for Cb and Cr components	DISPC_VIDp_ACCU2_j[10:0] HORIZONTALACCU	0x--
Configure vertical scaling		
Set the vertical resizing ratio for Cb and Cr components	DISPC_VIDp_FIR2[28:16] FIRVINC	0x--
Set the vertical FIR coefficients for Cb and Cr components	DISPC_VIDp_FIR_COEF_HV2_i[31:24][23:16][15:8] FIRVC2, FIRVC1, FIRVC0 Only for 5-taps vertical: DISPC_VIDp_FIR_COEF_V2_i[15:8][7:0] FIRVC22, FIRVC00	0x-- 0x--
Set the vertical accumulators value for Cb and Cr components	DISPC_VIDp_ACCU2_j[26:16] VERTICALACCU	0x--

**Figure 10-93. Scaler Unit Programming Flow**

#### 10.2.5.1.2.3.7 Subsequence - Configure the Video Pipeline Layer Output

This subsequence resume the video layer settings available at the pipeline level necessary when using the overlay manager.

**Table 10-75. Configure the Video Pipeline Layer Output**

Step	Register / Bit Field / Programming Model	Value
Set the LCD/TV output	DISPC_VIDp_ATTRIBUTES[16] CHANNELOUT	0x--
If DISPC_VIDp_ATTRIBUTES[16]=0, set the LCD output	DISPC_VIDp_ATTRIBUTES[31:30] CHANNELOUT2	0x--
Set the Z-order priority of the layer for overlay manager	DISPC_VIDp_ATTRIBUTES[27:26] ZORDER	0x--
Enable the video pipeline Z-order	DISPC_VIDp_ATTRIBUTES[25] ZORDERENABLE	0x--
Set the Global Alpha value for the Alpha blender unit	DISPC_GLOBAL_ALPHA[31:24][23:16][15:8] VID3GLOBALALPHA, VID2GLOBALALPHA , VID1GLOBALALPHA	0x--

#### 10.2.5.1.2.4 WB Pipeline Configuration

##### 10.2.5.1.2.4.1 Main Sequence - Configure the WB Pipeline

This procedure details the steps for a WB pipeline configuration described in [Table 10-76](#).

**Table 10-76. Configure the WB Pipeline**

Step	Register / Bit Field / Programming Model	Value
Configure the Capture window	Refer to <a href="#">Table 10-77</a>	
Configure the WB Scaler unit	Refer to <a href="#">Table 10-78</a>	
Enable truncation logic to match pixel size defined in the format of image	DISPC_WB_ATTRIBUTES[10] TRUNCATIONENABLE	0x--
	DISPC_WB_ATTRIBUTES[4:1] FORMAT	
if DISPC_WB_ATTRIBUTES[10] != , Configure the WB Color Space Conversion unit	Refer to <a href="#">Table 10-80</a>	
Validate the configuration according to the registers modification.	DISPC_CONTROL2[6] GOWB	

**Table 10-76. Configure the WB Pipeline (continued)**

Step	Register / Bit Field / Programming Model	Value
Enable WB pipeline	DISPC_WB_ATTRIBUTES[0] ENABLE	0x1

#### 10.2.5.1.2.4.2 Subsequence - Configure the Capture Window

This subsequence describes the parameters of the image to be captured in the system memory.

**Table 10-77. Configure the Capture Window**

Step	Register / Bit Field / Programming Model	Value
Set the input source	DISPC_WB_ATTRIBUTES[18:16] CHANNELIN	0x--
Select the format of image	DISPC_WB_ATTRIBUTES[4:1] FORMAT	0x--
Set the X size of image to be captured	DISPC_WB_SIZE[10:0] SIZEX	0x--
Set the Y size of image to be captured	DISPC_WB_SIZE[10:0] SIZEY	0x--

#### 10.2.5.1.2.4.3 Subsequence - Configure the WB Scaler Unit

This subsequence configures the scaler unit. [Table 10-73](#) is applicable for RGB pixel format. [Table 10-73](#) and [Table 10-74](#) are applicable for YUV pixel format.

**Table 10-78. Configure the WB Scaler Unit for RGB Pixel Formats or Y Component**

Step	Register / Bit Field / Programming Model	Value
Configure horizontal scaling		
Set the horizontal resizing ratio	DISPC_WB_FIR[12:0] FIRHINC	0x--
Set the horizontal FIR coefficients	DISPC_WB_FIR_COEF_H_i[31:24][23:16][15:8][7:0] FIRHC3, FIRHC2, FIRHC1, FIRHC0	0x--
	DISPC_WB_FIR_COEF_HV_i[7:0] FIRHC4	0x--
Set the horizontal accumulators value	DISPC_WB_ACCU_j[10:0] HORIZONTALACCU	0x--
Configure vertical scaling		
Select number of vertical taps	DISPC_WB_ATTRIBUTES[21] VERTICALTAPS	0x--
Set the vertical resizing ratio	DISPC_WB_FIR[28:16] FIRVINC	0x--
Set the vertical FIR coefficients for RGB pixel format or Y component	DISPC_WB_FIR_COEF_HV_i[31:24][23:16][15:8] FIRVC2, FIRVC1, FIRVC0	0x--
	Only for 5-taps vertical: DISPC_WB_FIR_COEF_V_i[15:8][7:0] FIRVC22, FIRVC00	0x--
Set the vertical accumulators value for RGB pixel format or Y component	DISPC_WB_ACCU_j[26:16] VERTICALACCU	0x--
Enable horizontal and vertical Scaler Unit	DISPC_WB_ATTRIBUTES[6:5] RESIZEENABLE	0x--

**Table 10-79. Configure the WB Scaler Unit for Cb and Cr Components**

Step	Register / Bit Field / Programming Model	Value
Configure horizontal scaling		
Set the horizontal resizing ratio for Cb and Cr components	DISPC_WB_FIR2[12:0] FIRHINC	0x--
Set the horizontal FIR coefficients for Cb and Cr components	DISPC_WB_FIR_COEF_H2_i[31:24][23:16][15:8][7:0] FIRHC3, FIRHC2, FIRHC1, FIRHC0	0x--
	DISPC_WB_FIR_COEF_HV2_i[7:0] FIRHC4	0x--
Set the horizontal accumulators value for Cb and Cr components	DISPC_WB_ACCU2_j[10:0] HORIZONTALACCU	0x--
Configure vertical scaling		
Set the vertical resizing ratio for Cb and Cr components	DISPC_WB_FIR2[28:16] FIRVINC	0x--

**Table 10-79. Configure the WB Scaler Unit for Cb and Cr Components (continued)**

Step	Register / Bit Field / Programming Model	Value
Set the vertical FIR coefficients for Cb and Cr components	DISPC_WB_FIR_COEF_HV2_i[31:24][23:16][15:8] FIRVC2, FIRVC1, FIRVC0	0x--
	Only for 5-taps vertical: DISPC_WB_FIR_COEF_V2_i[15:8][7:0] FIRVC22, FIRVC00	0x--
Set the vertical accumulators value for Cb and Cr components	DISPC_WB_ACCU2_j[26:16] VERTICALACCU	0x--

#### 10.2.5.1.2.4.4 Subsequence - Configure the WB Color Space Conversion Unit

**Table 10-80. Configure the WB Color Space Conversion Unit**

Step	Register / Bit Field / Programming Model	Value
Select the range of the Color Space Conversion	DISPC_WB_ATTRIBUTES[11] FULLRANGE	0x--
Set the RCr and RY coefficients	DISPC_WB_CONV_COEF0[26:16][10:0] YG, YR	0x--
Set the GY and RCb coefficients	DISPC_WB_CONV_COEF1[26:16][10:0] CRR, YB	0x--
Set the GCb and GCr coefficients	DISPC_WB_CONV_COEF2[26:16][10:0] CRB, CRG	0x--
Set the BCr and BY coefficients	DISPC_WB_CONV_COEF3[26:16][10:0] CBG, CBR	0x--
Set the BCb coefficient	DISPC_WB_CONV_COEF4[10:0] CBB	0x--
Enable Color Space Conversion	DISPC_WB_ATTRIBUTES[9] COLORCONVENABLE	0x1

#### 10.2.5.1.2.5 LCD output configuration

##### 10.2.5.1.2.5.1 Main Sequence - Configure the LCD Output

This procedure details the LCD output configuration.

**Table 10-81. Configure the LCD Output**

Step	Register / Bit Field / Programming Model
Configure the LCD overlay manager	<a href="#">Table 10-82</a>
Configure the gamma table for gamma correction	<a href="#">Section 10.2.5.1.2.5.3</a>
Configure the Color Phase Rotation	<a href="#">Table 10-85</a>
Configure the LCD panel timings and parameters	<a href="#">Table 10-86</a>
Validate the LCD output configuration according	DISPC_CONTROL1[5] GOLCD
	DISPC_CONTROL2[5] GOLCD
Enable LCD output	DISPC_CONTROL1[0] LCDENABLE
	DISPC_CONTROL2[0] LCDENABLE

##### 10.2.5.1.2.5.2 Subsequence - Configure the Overlay Manager

This subsequence resumes the overlay manager settings and transparency color key configuration.

**Table 10-82. Configure the LCD Overlay Manager**

Step	Register / Bit Field / Programming Model	Value
Set the LCD panel background color	DISPC_DEFAULT_COLORo[23:0] DEFAULTCOLOR	0x--
Enable/Disable overlay optimization	DISPC_CONTROLo[12] OVERLAYOPTIMIZATION	0x--
Enable the Alpha blender <sup>(1)</sup>	DISPC_CONFIG1[18] LCDALPHABLENDERENABLE	0x1
Configure the transparency color key		
Set source or destination transparency color key mode	DISPC_CONFIGo[11] TCKLCDSELECTION	0x--

<sup>(1)</sup> Backward compatibility with OMAP34xx only available for LCD1.

**Table 10-82. Configure the LCD Overlay Manager (continued)**

Step	Register / Bit Field / Programming Model	Value
Set the transparency color value	DISPC_TRANS_COLORo[23:0] TRANSCOLORKEY	0x--
Enable transparency color key mode	DISPC_CONFIGo[10] TCKLCDENABLE	0x--

#### 10.2.5.1.2.5.3 Subsequence - Configure the Gamma Table for Gamma Correction

This subsequence resumes the settings for configuring the gamma correction for LCD1, see [Table 10-83](#) and LCD2, see [Table 10-84](#).

**NOTE:**

- It is software responsibility to ensure that there is no visible effect when modifying the table since it is not under hardware control and the synchronization done using the DMA engine inside the DISPC to load the table when it is not used for displaying the picture on the screen is not present for this mode.
- If GFX pipeline is in BITMAP pixel format, Gamma corrections is not available for LCD1.

**Table 10-83. Configure the Gamma Table for LCD1**

Step	Register / Bit Field / Programming Model	Value
Initialize all entries for the gamma table by setting the table index and the RGB values associated to this index	DISPC_GAMMA_TABLE0[31:24][23:16][15:8][7:0], INDEX, VALUE_R, VALUE_G, VALUE_B	0x--
Set look up table as gamma table	DISPC_CONFIG1[3] PALETTEGAMMABLE	0x1
Select the load mode of the Gamma look-up table	DISPC_CONFIG1[2:1] LOADMODE	0x--

**Table 10-84. Configure the Gamma Table for LCD2**

Step	Register / Bit Field / Programming Model	Value
Enable Gamma table for LCD2 and TV	DISPC_CONFIG1[9] GAMATABLEENABLE	0x1
Initialize all entries for the gamma table by setting the table index and the RGB values associated to this index.	DISPC_GAMMA_TABLE1[31:24][23:16][15:8][7:0], INDEX, VALUE_R, VALUE_G, VALUE_B	0x--

#### 10.2.5.1.2.5.4 Subsequence - Configure the Color Phase Rotation

[Table 10-85](#) resumes the settings for the Color Phase Rotation unit.

**Table 10-85. Configure the Color Phase Rotation**

Step	Register / Bit Field / Programming Model	Value
Set the Red coefficients	DISPC_CPRo_COEF_R [31:22][20:11][9:0] RR, RG, RB	0x--
Set the Green coefficients	DISPC_CPRo_COEF_G [31:22][20:11][9:0] GR, GG, GB	0x--
Set the Blue coefficients	DISPC_CPRo_COEF_B [31:22][20:11][9:0] BR, BG, BB	0x--
Disable Color Phase Rotation unit in RGB to YUV conversion	DISPC_CONFIGo[24] COLORCONVENABLE	0x0
Enable Color Phase Rotation unit	DISPC_CONFIGo[15] CPR	0x1

#### 10.2.5.1.2.5.5 Subsequence - Configure the LCD panel timings and parameters

This subsequence resumes the setting for horizontal and vertical synchronization and also the signals polarity.

**Table 10-86. Configure the LCD Panel Timings and Parameters**

Step	Register / Bit Field / Programming Model	Value
<b>Configure Spatial/Temporal Dithering</b>		
Select Spatial/Temporal number of frames	DISPC_CONTROLo[31:30] SPATIALTEMPORALDITHERINGFRAMES	0x--
Enable Spatial/Temporal dithering	DISPC_CONTROLo[7] STDITHERENABLE	0x--
<b>Configure AC-bias</b>		
Configure the VSYNC, HSYNC and AC bias polarity	DISPC_POL_FREQo	0x--
Configure the gating of AC bias polarity	DISPC_CONFIGo[8] ACBIASGATED	0x--
Configure the AC bias polarity	DISPC_POL_FREQo[15] IEO	0x--
Set the AC bias frequency	DISPC_POL_FREQo[7:0] ACB	0x--
Set the number of AC bias transistion per interrupt	DISPC_POL_FREQo[11:8] ACBI	0x--
<b>Configure the pixel clock</b>		
Set the DISPC logic clock divisor	DISPC_DIVISORo[23:16] LCD	0x--
Set the pixel clock divisor	DISPC_DIVISORo[7:0] PCD	0x--
Configure the gating of Pixel clock	DISPC_CONFIGo[5] PIXELCLOCKGATED	0x--
<b>Configure the data</b>		
Set the pixel clock edge to drive data output	DISPC_POL_FREQo[14] IPC	0x--
Configure the gating of datas output	DISPC_CONFIGo[4] PIXELDATAGATED	0x--
Set the data output mode	DISPC_CONFIGo[22] OUTPUTMODEENABLE	0x--
<b>Configure the panel parameters</b>		
Set the vertical TV size	DISPC_SIZE_LCDo[26:16] LPP	0x--
Set the horizontal TV size	DISPC_SIZE_LCDo[10:0] PPL	0x--
Set the panel type	DISPC_CONTROLo[3] STNTFT	0x--
<b>Configure the refresh rate and horizontal and vertical parameters</b>		
Set the vertical synchronization timing	DISPC_TIMING_Vo[31:20][19:8][7:0],VBP, VFP, VSW	0x--
Configure the VSYNC polarity	DISPC_POL_FREQo[12] IVS	0x--
Configure the gating of VSYNC	DISPC_CONFIGo[7] VSYNCGATED	0x--
Set the horizontal synchronization timing	DISPC_TIMING_Ho[31:20][19:8][7:0],HBP, HFP, HSW	0x--
Configure the HSYNC polarity	DISPC_POL_FREQo[13] IHS	0x--
Configure the gating of HSYNC	DISPC_CONFIGo[6] HSYNCGATED	0x--
Set the opposition of HSYNC and VSYNC driving	DISPC_POL_FREQo[17] ONOFF	0x--
If DISPC_POL_FREQo[17] = 1, set the pixel clock edge to drive HSYNC and VSYNC	DISPC_POL_FREQo[16] RF	0x--
Set the alignment of HSYNC and VSYNC	DISPC_POL_FREQo[18] ALIGN	0x--

### 10.2.5.1.2.6 TV Output Configuration

#### Main Sequence - Configure the TV Output

This procedure details the TV output configuration.

**Table 10-87. Configure the TV Output**

Step	Register / Bit Field / Programming Model
Configure the TV overlay manager	Refer to <a href="#">Table 10-88</a>
Configure the gamma table for gamma correction	Refer to <a href="#">Table 10-89</a>
Configure the TV panel timings and parameters	Refer to <a href="#">Table 10-90</a>
Validate the TV output configuration according	DISPC_CONTROL1[6] GOTV
Enable TV output	DISPC_CONTROL1[1] TVENABLE

### 10.2.5.1.2.6.1 Subsequence - Configure the TV Overlay Manager

This subsequence resumes the overlay manager settings and transparency color key configuration.

**Table 10-88. Configure the TV Overlay Manager**

Step	Register / Bit Field / Programming Model	Value
Set the TV panel background color	DISPC_DEFAULT_COLOR1[23:0] DEFAULTCOLOR	0x--
Enable/Disable overlay optimization	DISPC_CONTROL2[13] TVOVERLAYOPTIMIZATION	0x--
Enable the Alpha blender <sup>(1)</sup>	DISPC_CONFIG1[19] TVALPHABLENDERENABLE	0x1
Configure the transparency color key		
Set source or destination transparency color key mode	DISPC_CONFIG1[13] TCKTVSELECTION	0x--
Set the transparency color value	DISPC_TRANS_COLOR1[23:0] TRANSCOLORKEY	0x--
Enable transparency color key mode	DISPC_CONFIG1[10] TCKTVENABLE	0x--

<sup>(1)</sup> Backward compatibility with OMAP34xx.

### 10.2.5.1.2.6.2 Subsequence - Configure the Gamma Table for Gamma Correction

**NOTE:**

- It is software responsibility to ensure that there is no visible effect when modifying the table since it is not under hardware control and the synchronization done using the DMA engine inside the DISPC to load the table when it is not used for displaying the picture on the screen is not present for this mode.

**Table 10-89. Configure the Gamma Table for TV Output**

Step	Register / Bit Field / Programming Model	Value
Enable Gamma table for LCD2 and TV	DISPC_CONFIG1[9] GAMATABLEENABLE	0x1
Initialize all entries for the gamma table by setting the table index and the RGB values. For more information, see <a href="#">Section 10.2.4.13.2, Gamma Correction Unit</a> .	DISPC_GAMMA_TABLE2[31][29:20][19:10][9:0], INDEX, VALUE_R, VALUE_G, VALUE_B	0x--

### 10.2.5.1.2.6.3 Subsequence - Configure the TV Panel Timings and Parameters

This subsequence resumes the setting for horizontal and vertical synchronization and also the signals polarity.

**Table 10-90. Configure the TV Panel Timings and Parameters**

Step	Register / Bit Field / Programming Model	Value
Set the hold time for TV data outputs	DISPC_CONFIG1[19:17]	0x--
Set the vertical TV size	DISPC_SIZE_TV[26:16] LPP	See <a href="#">Table 10-91</a> for NTSC/PAL and HD standards
Set the horizontal TV size	DISPC_SIZE_TV[10:0] PPL	

**Table 10-91. DISPC\_SIZE\_TV values NTSC/PAL and HD Standard**

Standards		Active Pixels/line (PPL)	Active Lines (LPP)	Digital clock (DISPC_CONTROL1. HT=0)	DISPC_SIZE_TV
ITU-R BT.601	NTSC(J, M, 4.43)	720	480	13.5 MHz	0x00EF02CF
	PAL(B, D, G, H, I, N)	720	576	13.5 MHz	0x011F02CF
	PAL(M)	720	480	13.5 MHz	0x00EF02CF

**Table 10-91. DISPC\_SIZE\_TV values NTSC/PAL and HD Standard (continued)**

Standards		Active Pixels/line (PPL)	Active Lines (LPP)	Digital clock (DISPC_CONTROL1. HT=0)	DISPC_SIZE_TV
Square Pixel	NTSC(J, M, 4.43)	640	480	12.2727 MHz	0x00EF027F
	PAL(B, D, G, H, I, N)	768	576	14.75 MHz	0x011F02FF
	PAL(M)	640	480	12.2727 MHz	0x00EF027F
HDTV	720p	1280	720	74.25/74.125 MHz (60/59.99..frames/s)	0x02CF04FF
	1080i	1920	540	74.25/74.125 MHz (60/59.99..fields/s)	0x021B07FF
	1080p	1920	1080	148.5/148.25 MHz (60/59.99..frames/s)	0x043707FF

## 10.2.6 Display Controller Use Cases and Tips

### 10.2.6.1 Hardware Cursor

The video layer or graphics layer can be used to display the hardware cursor. The encoded pixel data for the cursor image are in xRGB12-4444, ARGB16-4444, RGBx12-4444, RGBA16-4444, ARGB16-1555, RGB16-565, RGB24-888, xRGB24-88888, RGBA32-8888 or ARGB32-8888 formats if the source transparency color key is used otherwise any pixel format can be used for the cursor image considering the limitation in term of pixel supported by the pipeline used to display the cursor image. In order to display none rectangle cursor, the transparency color key can be used or the alpha blending (pixel alpha blending), refer to [Section 10.2.4.12.1.3](#). The global alpha blending can be used in addition to the transparency source color key to make some fading effect when the cursor (with a non rectangle shape) appears and disappears on the screen. The image of the cursor can be stored totally inside the DMA buffer in order to use the selfrefresh mode, refer to [Section 10.2.4.6.5.2](#). In that case, the image is loaded once and then displayed without accessing the system memory for loading the image for each frame. It saves bandwidth on interconnect and memory.

## 10.2.7 Display Controller Register Manual

### CAUTION

The main access to all DISPC registers is through the L3 interconnect.  
The access through L4\_PER interconnect is provided for back software compatibility.

### 10.2.7.1 Display Controller Instance Summary

**Table 10-92. DISPC Instance Summary**

Module Name	L4_PER Base Address	L3 Base Address	Size
DISPC	0x4804 1000	0x5800 1000	4KB

### 10.2.7.2 Display Controller Logical Register Mapping

**Table 10-93. DISPC\_VIDp\_BA\_j Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_BA_j</a>	Base Address of Video Pipeline 1
<a href="#">DISPC_VID2_BA_j</a>	Base Address of Video Pipeline 2
<a href="#">DISPC_VID3_BA_j</a>	Base Address of Video Pipeline 3



**Table 10-94. DISPC\_VIDp\_BA\_UV\_j Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_BA_UV_j</a>	Base Address of UV components for Video Pipeline 1
<a href="#">DISPC_VID2_BA_UV_j</a>	Base Address of UV components for Video Pipeline 2
<a href="#">DISPC_VID3_BA_UV_j</a>	Base Address of UV components for Video Pipeline 3

**Table 10-95. DISPC\_VIDp\_POSITION Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_POSITION</a>	Position of the video window 1
<a href="#">DISPC_VID2_POSITION</a>	Position of the video window 2
<a href="#">DISPC_VID3_POSITION</a>	Position of the video window 3

**Table 10-96. DISPC\_VIDp\_SIZE Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_SIZE</a>	Size of the video window 1
<a href="#">DISPC_VID2_SIZE</a>	Size of the video window 2
<a href="#">DISPC_VID3_SIZE</a>	Size of the video window 3

**Table 10-97. DISPC\_VIDp\_ATTRIBUTES Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_ATTRIBUTES</a>	Configuration of the video pipeline 1
<a href="#">DISPC_VID2_ATTRIBUTES</a>	Configuration of the video pipeline 2
<a href="#">DISPC_VID3_ATTRIBUTES</a>	Configuration of the video pipeline 3

**Table 10-98. DISPC\_VIDp\_ATTRIBUTES2 Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_ATTRIBUTES2</a>	Configuration of the video pipeline 1
<a href="#">DISPC_VID2_ATTRIBUTES2</a>	Configuration of the video pipeline 2
<a href="#">DISPC_VID3_ATTRIBUTES2</a>	Configuration of the video pipeline 3

**Table 10-99. DISPC\_VIDp\_BUF\_THRESHOLD Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_BUF_THRESHOLD</a>	Configuration of the buffer for the video pipeline 1
<a href="#">DISPC_VID2_BUF_THRESHOLD</a>	Configuration of the buffer for the video pipeline 2
<a href="#">DISPC_VID3_BUF_THRESHOLD</a>	Configuration of the buffer for the video pipeline 3

**Table 10-100. DISPC\_VIDp\_ROW\_INC Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_ROW_INC</a>	Configuration of the row increment for the video pipeline 1
<a href="#">DISPC_VID2_ROW_INC</a>	Configuration of the row increment for the video pipeline 2
<a href="#">DISPC_VID3_ROW_INC</a>	Configuration of the row increment for the video pipeline 3

**Table 10-101. DISPC\_VIDp\_PIXEL\_INC Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_PIXEL_INC</a>	Configuration of the pixel increment for the video pipeline 1
<a href="#">DISPC_VID2_PIXEL_INC</a>	Configuration of the pixel increment for the video pipeline 2
<a href="#">DISPC_VID3_PIXEL_INC</a>	Configuration of the pixel increment for the video pipeline 3

**Table 10-102. DISPC\_VIDp\_FIR Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_FIR</a>	Configuration of the scaler for the video pipeline 1
<a href="#">DISPC_VID2_FIR</a>	Configuration of the scaler for the video pipeline 2
<a href="#">DISPC_VID3_FIR</a>	Configuration of the scaler for the video pipeline 3

**Table 10-103. DISPC\_VIDp\_PICTURE\_SIZE Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_PICTURE_SIZE</a>	Size of the video window 1 before processing
<a href="#">DISPC_VID2_PICTURE_SIZE</a>	Size of the video window 2 before processing
<a href="#">DISPC_VID3_PICTURE_SIZE</a>	Size of the video window 3 before processing

**Table 10-104. DISPC\_VIDp\_ACCU\_j Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_ACCU_j</a>	Configuration of the accumulator for the video pipeline 1
<a href="#">DISPC_VID2_ACCU_j</a>	Configuration of the accumulator for the video pipeline 2
<a href="#">DISPC_VID2_ACCU_j</a>	Configuration of the accumulator for the video pipeline 3

**Table 10-105. DISPC\_VIDp\_FIR\_COEF\_H\_i Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_FIR_COEF_H_i</a>	Configuration of the horizontal scaling coefficients for the video pipeline 1
<a href="#">DISPC_VID2_FIR_COEF_H_i</a>	Configuration of the horizontal scaling coefficients for the video pipeline 2
<a href="#">DISPC_VID3_FIR_COEF_H_i</a>	Configuration of the horizontal scaling coefficients or the video pipeline 3

**Table 10-106. DISPC\_VIDp\_FIR\_COEF\_HV\_i Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_FIR_COEF_HV_i</a>	Configuration of the horizontal scaling coefficients for the video pipeline 1
<a href="#">DISPC_VID2_FIR_COEF_HV_i</a>	Configuration of the horizontal scaling coefficients for the video pipeline 2
<a href="#">DISPC_VID3_FIR_COEF_HV_i</a>	Configuration of the horizontal scaling coefficients for the video pipeline 3

**Table 10-107. DISPC\_VIDp\_FIR\_COEF\_V\_i Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_FIR_COEF_V_i</a>	Configuration of the vertical scaling coefficients for the video pipeline 1
<a href="#">DISPC_VID2_FIR_COEF_V_i</a>	Configuration of the vertical scaling coefficients for the video pipeline 2
<a href="#">DISPC_VID3_FIR_COEF_V_i</a>	Configuration of the vertical scaling coefficients for the video pipeline 3

**Table 10-108. DISPC\_VIDp\_FIR\_COEF\_H2\_i Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_FIR_COEF_H2_i</a>	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 1
<a href="#">DISPC_VID2_FIR_COEF_H2_i</a>	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 2
<a href="#">DISPC_VID3_FIR_COEF_H2_i</a>	Configuration of the horizontal Cb and Cr scaling coefficients or the video pipeline 3

**Table 10-109. DISPC\_VIDp\_FIR\_COEF\_HV2\_i Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_FIR_COEF_HV2_i</a>	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 1
<a href="#">DISPC_VID2_FIR_COEF_HV2_i</a>	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 2
<a href="#">DISPC_VID3_FIR_COEF_HV2_i</a>	Configuration of the horizontal Cb and Cr scaling coefficients for the video pipeline 3

**Table 10-110. DISPC\_VIDp\_FIR\_COEF\_V2\_i Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_FIR_COEF_V2_i</a>	Configuration of the vertical Cb and Cr scaling coefficients for the video pipeline 1
<a href="#">DISPC_VID2_FIR_COEF_V2_i</a>	Configuration of the vertical Cb and Cr scaling coefficients for the video pipeline 2
<a href="#">DISPC_VID3_FIR_COEF_V2_i</a>	Configuration of the vertical Cb and Cr scaling coefficients for the video pipeline 3

**Table 10-111. DISPC\_VIDp\_CONV\_COEF0 Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_CONV_COEF0</a>	Configuration of the Color Space Conversion coefficients for the video pipeline 1
<a href="#">DISPC_VID2_CONV_COEF0</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 2
<a href="#">DISPC_VID3_CONV_COEF0</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 3

**Table 10-112. DISPC\_VIDp\_CONV\_COEF1 Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_CONV_COEF1</a>	Configuration of the Color Space Conversion coefficients for the video pipeline 1
<a href="#">DISPC_VID2_CONV_COEF1</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 2
<a href="#">DISPC_VID3_CONV_COEF1</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 3

**Table 10-113. DISPC\_VIDp\_CONV\_COEF2 Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_CONV_COEF2</a>	Configuration of the Color Space Conversion coefficients for the video pipeline 1
<a href="#">DISPC_VID2_CONV_COEF2</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 2
<a href="#">DISPC_VID3_CONV_COEF2</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 3

**Table 10-114. DISPC\_VIDp\_CONV\_COEF3 Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_CONV_COEF3</a>	Configuration of the Color Space Conversion coefficients for the video pipeline 1
<a href="#">DISPC_VID2_CONV_COEF3</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 2
<a href="#">DISPC_VID3_CONV_COEF3</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 3

**Table 10-115. DISPC\_VIDp\_CONV\_COEF4 Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_VID1_CONV_COEF4</a>	Configuration of the Color Space Conversion coefficients for the video pipeline 1
<a href="#">DISPC_VID2_CONV_COEF4</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 2
<a href="#">DISPC_VID3_CONV_COEF4</a>	Configuration of the Color Space Conversion coefficient for the video pipeline 3

**Table 10-116. DISPC\_CONTROLo Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_CONTROL1</a>	Configuration control of the LCD1 and TV.
<a href="#">DISPC_CONTROL2</a>	Configuration control of the LCD2.

**Table 10-117. DISPC\_CONFIGo Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_CONFIG1</a>	Configuration of the LCD1 and TV.
<a href="#">DISPC_CONFIG2</a>	Configuration of the LCD2.

**Table 10-118. DISPC\_DEFAULT\_COLORo Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_DEFAULT_COLOR0</a>	Configuration of the background color for LCD1.
<a href="#">DISPC_DEFAULT_COLOR1</a>	Configuration of the background color for LCD2.

**Table 10-119. DISPC\_TRANS\_COLORo Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_TRANS_COLOR0</a>	Configuration of the transparency color key for LCD1.
<a href="#">DISPC_TRANS_COLOR1</a>	Configuration of the transparency color key for LCD2.

**Table 10-120. DISPC\_GAMMA\_TABLEo Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_GAMMA_TABLE0</a>	Configuration of the palette table for GFX or the gamme table for LCD1.
<a href="#">DISPC_GAMMA_TABLE1</a>	Configuration of the gamma table for LCD2.
<a href="#">DISPC_GAMMA_TABLE2</a>	Configuration of the gamme table for TV output.

**Table 10-121. DISPC\_TIMING\_Ho Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_TIMING_H1</a>	Configuration of the horizontal timing for LCD1.
<a href="#">DISPC_TIMING_H2</a>	Configuration of the horizontal timing for LCD2.

**Table 10-122. DISPC\_TIMING\_Vo Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_TIMING_V1</a>	Configuration of the vertical timing for LCD1.
<a href="#">DISPC_TIMING_V2</a>	Configuration of the vertical timing for LCD2.

**Table 10-123. DISPC\_POL\_FREQo Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_POL_FREQ1</a>	Configuration of the output signals for LCD1.
<a href="#">DISPC_POL_FREQ2</a>	Configuration of the output signals for LCD2.

**Table 10-124. DISPC\_DIVISORo Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_DIVISOR1</a>	Configuration of the divisors for LCD1.
<a href="#">DISPC_DIVISOR2</a>	Configuration of the divisors for LCD2.

**Table 10-125. DISPC\_SIZE\_LCDo Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_SIZE_LCD1</a>	Configuration of the divisors for LCD1.
<a href="#">DISPC_SIZE_LCD2</a>	Configuration of the divisors for LCD2.

**Table 10-126. DISPC\_SIZE Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_SIZE_LCD1</a>	Configuration of the LCD size on LCD1.
<a href="#">DISPC_SIZE_LCD2</a>	Configuration of the LCD size on LCD2.

**Table 10-127. DISPC\_DATAo\_CYCLE1 Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_DATA1_CYCLE1</a>	Configuration of the output data format for 1st cycle on LCD1.
<a href="#">DISPC_DATA2_CYCLE1</a>	Configuration of the output data format for 1st cycle on LCD2.

**Table 10-128. DISPC\_DATAo\_CYCLE2 Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_DATA1_CYCLE2</a>	Configuration of the output data format for 2nd cycle on LCD1.
<a href="#">DISPC_DATA2_CYCLE2</a>	Configuration of the output data format for 2nd cycle on LCD2.

**Table 10-129. DISPC\_DATAo\_CYCLE3 Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_DATA1_CYCLE3</a>	Configuration of the output data format for 3rd cycle on LCD1.
<a href="#">DISPC_DATA2_CYCLE3</a>	Configuration of the output data format for 3rd cycle on LCD2.

**Table 10-130. DISPC\_CPRo\_COEF\_R Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_CPR1_COEF_R</a>	Configuration of the the CPR matrix coefficients for the Red component on LCD1.
<a href="#">DISPC_CPR2_COEF_R</a>	Configuration of the the CPR matrix coefficients for the Red component on LCD2.

**Table 10-131. DISPC\_CPRo\_COEF\_G Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_CPR1_COEF_G</a>	Configuration of the the CPR matrix coefficients for the Green component on LCD1.
<a href="#">DISPC_CPR2_COEF_G</a>	Configuration of the the CPR matrix coefficients for the Green component on LCD2.

**Table 10-132. DISPC\_CPRo\_COEF\_B Logical Register Mapping**

Hardware Register	Description
<a href="#">DISPC_CPR1_COEF_B</a>	Configuration of the the CPR matrix coefficients for the Blue component on LCD1.
<a href="#">DISPC_CPR2_COEF_B</a>	Configuration of the the CPR matrix coefficients for the Blue component on LCD2.

## 10.2.7.3 Display Controller Registers

### 10.2.7.3.1 Display Controller Register Summary

**Table 10-133. Display Controller Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DISPC L4_PER Physical Address	DISPC L3 Physical Address
DISPC_REVISION	R	32	0x0000 0000	0x4804 1000	0x5800 1000
DISPC_SYSCONFIG	RW	32	0x0000 0010	0x4804 1010	0x5800 1010
DISPC_SYSSTATUS	R	32	0x0000 0014	0x4804 1014	0x5800 1014
DISPC_IRQSTATUS	RW	32	0x0000 0018	0x4804 1018	0x5800 1018
DISPC_IRQENABLE	RW	32	0x0000 001C	0x4804 101C	0x5800 101C
DISPC_CONTROL1	RW	32	0x0000 0040	0x4804 1040	0x5800 1040
DISPC_CONFIG1	RW	32	0x0000 0044	0x4804 1044	0x5800 1044
RESERVED	R	32	0x0000 0048	0x4804 1048	0x5800 1048
DISPC_DEFAULT_COLOR0	RW	32	0x0000 004C	0x4804 104C	0x5800 104C
DISPC_DEFAULT_COLOR1	RW	32	0x0000 0050	0x4804 1050	0x5800 1050
DISPC_TRANS_COLOR0	RW	32	0x0000 0054	0x4804 1054	0x5800 1054
DISPC_TRANS_COLOR1	RW	32	0x0000 0058	0x4804 1058	0x5800 1058
DISPC_LINE_STATUS	R	32	0x0000 005C	0x4804 105C	0x5800 105C
DISPC_LINE_NUMBER	RW	32	0x0000 0060	0x4804 1060	0x5800 1060
DISPC_TIMING_H1	RW	32	0x0000 0064	0x4804 1064	0x5800 1064
DISPC_TIMING_V1	RW	32	0x0000 0068	0x4804 1068	0x5800 1068
DISPC_POL_FREQ1	RW	32	0x0000 006C	0x4804 106C	0x5800 106C
DISPC_DIVISOR1	RW	32	0x0000 0070	0x4804 1070	0x5800 1070
DISPC_GLOBAL_ALPHA	RW	32	0x0000 0074	0x4804 1074	0x5800 1074
DISPC_SIZE_TV	RW	32	0x0000 0078	0x4804 1078	0x5800 1078
DISPC_SIZE_LCD1	RW	32	0x0000 007C	0x4804 107C	0x5800 107C
DISPC_GFX_BA_j <sup>(1)</sup>	RW	32	0x0000 0080 + (0x4 * j)	0x4804 1080 + (0x4 * j)	0x5800 1080 + (0x4 * j)
DISPC_GFX_POSITION	RW	32	0x0000 0088	0x4804 1088	0x5800 1088
DISPC_GFX_SIZE	RW	32	0x0000 008C	0x4804 108C	0x5800 108C
DISPC_GFX_ATTRIBUTES	RW	32	0x0000 00A0	0x4804 10A0	0x5800 10A0
DISPC_GFX_BUF_THRESHOLD	RW	32	0x0000 00A4	0x4804 10A4	0x5800 10A4
DISPC_GFX_BUF_SIZE_STATUS	R	32	0x0000 00A8	0x4804 10A8	0x5800 10A8
DISPC_GFX_ROW_INC	RW	32	0x0000 00AC	0x4804 10AC	0x5800 10AC
DISPC_GFX_PIXEL_INC	RW	32	0x0000 00B0	0x4804 10B0	0x5800 10B0
DISPC_GFX_WINDOW_SKIP	R	32	0x0000 00B4	0x4804 10B4	0x5800 10B4
DISPC_GFX_TABLE_BA	RW	32	0x0000 00B8	0x4804 10B8	0x5800 10B8
DISPC_VID1_BA_j <sup>(1)</sup>	RW	32	0x0000 00BC + (0x4 * j)	0x4804 10BC + (0x4 * j)	0x5800 10BC + (0x4 * j)
DISPC_VID1_POSITION	RW	32	0x0000 00C4	0x4804 10C4	0x5800 10C4
DISPC_VID1_SIZE	RW	32	0x0000 00C8	0x4804 10C8	0x5800 10C8
DISPC_VID1_ATTRIBUTES	RW	32	0x0000 00CC	0x4804 10CC	0x5800 10CC
DISPC_VID1_BUF_THRESHOLD	RW	32	0x0000 00D0	0x4804 10D0	0x5800 10D0
DISPC_VID1_BUF_SIZE_STATUS	R	32	0x0000 00D4	0x4804 10D4	0x5800 10D4
DISPC_VID1_ROW_INC	RW	32	0x0000 00D8	0x4804 10D8	0x5800 10D8
DISPC_VID1_PIXEL_INC	RW	32	0x0000 00DC	0x4804 10DC	0x5800 10DC
DISPC_VID1_FIR	RW	32	0x0000 00E0	0x4804 10E0	0x5800 10E0

<sup>(1)</sup> j = 0 to 1

**Table 10-133. Display Controller Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DISPC L4_PER Physical Address	DISPC L3 Physical Address
DISPC_VID1_PICTURE_SIZE	RW	32	0x0000 00E4	0x4804 10E4	0x5800 10E4
DISPC_VID1_ACCU_j <sup>(1)</sup>	RW	32	0x0000 00E8 + (0x4 * j)	0x4804 10E8 + (0x4 * j)	0x5800 10E8 + (0x4 * j)
DISPC_VID1_FIR_COEF_H_j <sup>(2)</sup>	RW	32	0x0000 00F0 + (0x8 * i)	0x4804 10F0 + (0x8 * i)	0x5800 10F0 + (0x8 * i)
DISPC_VID1_FIR_COEF_HV_j <sup>(2)</sup>	RW	32	0x0000 00F4 + (0x8 * i)	0x4804 10F4 + (0x8 * i)	0x5800 10F4 + (0x8 * i)
DISPC_VID1_CONV_COEF0	RW	32	0x0000 0130	0x4804 1130	0x5800 1130
DISPC_VID1_CONV_COEF1	RW	32	0x0000 0134	0x4804 1134	0x5800 1134
DISPC_VID1_CONV_COEF2	RW	32	0x0000 0138	0x4804 1138	0x5800 1138
DISPC_VID1_CONV_COEF3	RW	32	0x0000 013C	0x4804 113C	0x5800 113C
DISPC_VID1_CONV_COEF4	RW	32	0x0000 0140	0x4804 1140	0x5800 1140
DISPC_VID2_BA_j <sup>(1)</sup>	RW	32	0x0000 014C + (0x4 * j)	0x4804 114C + (0x4 * j)	0x5800 114C + (0x4 * j)
DISPC_VID2_POSITION	RW	32	0x0000 0154	0x4804 1154	0x5800 1154
DISPC_VID2_SIZE	RW	32	0x0000 0158	0x4804 1158	0x5800 1158
DISPC_VID2_ATTRIBUTES	RW	32	0x0000 015C	0x4804 115C	0x5800 115C
DISPC_VID2_BUF_THRESHOLD	RW	32	0x0000 0160	0x4804 1160	0x5800 1160
DISPC_VID2_BUF_SIZE_STATUS	R	32	0x0000 0164	0x4804 1164	0x5800 1164
DISPC_VID2_ROW_INC	RW	32	0x0000 0168	0x4804 1168	0x5800 1168
DISPC_VID2_PIXEL_INC	RW	32	0x0000 016C	0x4804 116C	0x5800 116C
DISPC_VID2_FIR	RW	32	0x0000 0170	0x4804 1170	0x5800 1170
DISPC_VID2_PICTURE_SIZE	RW	32	0x0000 0174	0x4804 1174	0x5800 1174
DISPC_VID2_ACCU_j <sup>(3)</sup>	RW	32	0x0000 0178 + (0x4 * j)	0x4804 1178 + (0x4 * j)	0x5800 1178 + (0x4 * j)
DISPC_VID2_FIR_COEF_H_j <sup>(4)</sup>	RW	32	0x0000 0180 + (0x8 * i)	0x4804 1180 + (0x8 * i)	0x5800 1180 + (0x8 * i)
DISPC_VID2_FIR_COEF_HV_j <sup>(4)</sup>	RW	32	0x0000 0184 + (0x8 * i)	0x4804 1184 + (0x8 * i)	0x5800 1184 + (0x8 * i)
DISPC_VID2_CONV_COEF0	RW	32	0x0000 01C0	0x4804 11C0	0x5800 11C0
DISPC_VID2_CONV_COEF1	RW	32	0x0000 01C4	0x4804 11C4	0x5800 11C4
DISPC_VID2_CONV_COEF2	RW	32	0x0000 01C8	0x4804 11C8	0x5800 11C8
DISPC_VID2_CONV_COEF3	RW	32	0x0000 01CC	0x4804 11CC	0x5800 11CC
DISPC_VID2_CONV_COEF4	RW	32	0x0000 01D0	0x4804 11D0	0x5800 11D0
DISPC_DATA1_CYCLE1	RW	32	0x0000 01D4	0x4804 11D4	0x5800 11D4
DISPC_DATA1_CYCLE2	RW	32	0x0000 01D8	0x4804 11D8	0x5800 11D8
DISPC_DATA1_CYCLE3	RW	32	0x0000 01DC	0x4804 11DC	0x5800 11DC
DISPC_VID1_FIR_COEF_V_j <sup>(4)</sup>	RW	32	0x0000 01E0 + (0x4 * i)	0x4804 11E0 + (0x4 * i)	0x5800 11E0 + (0x4 * i)
DISPC_VID2_FIR_COEF_V_j <sup>(4)</sup>	RW	32	0x0000 0200 + (0x4 * i)	0x4804 1200 + (0x4 * i)	0x5800 1200 + (0x4 * i)
DISPC_CPR1_COEF_R	RW	32	0x0000 0220	0x4804 1220	0x5800 1220
DISPC_CPR1_COEF_G	RW	32	0x0000 0224	0x4804 1224	0x5800 1224
DISPC_CPR1_COEF_B	RW	32	0x0000 0228	0x4804 1228	0x5800 1228
DISPC_GFX_PRELOAD	RW	32	0x0000 022C	0x4804 122C	0x5800 122C
DISPC_VID1_PRELOAD	RW	32	0x0000 0230	0x4804 1230	0x5800 1230

<sup>(2)</sup> i = 0 to 7

<sup>(3)</sup> j = 0 to 1

<sup>(4)</sup> i = 0 to 7

**Table 10-133. Display Controller Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DISPC L4_PER Physical Address	DISPC L3 Physical Address
DISPC_VID2_PRELOAD	RW	32	0x0000 0234	0x4804 1234	0x5800 1234
DISPC_CONTROL2	RW	32	0x0000 0238	0x4804 1238	0x5800 1238
DISPC_VID3_ACCU_j <sup>(3)</sup>	RW	32	0x0000 0300 + (0x4 * j)	0x4804 1300 + (0x4 * j)	0x5800 1300 + (0x4 * j)
DISPC_VID3_BA_j <sup>(3)</sup>	RW	32	0x0000 0308 + (0x4 * j)	0x4804 1308 + (0x4 * j)	0x5800 1308 + (0x4 * j)
DISPC_VID3_FIR_COEF_H_i <sup>(4)</sup>	RW	32	0x0000 0310 + (0x8 * i)	0x4804 1310 + (0x8 * i)	0x5800 1310 + (0x8 * i)
DISPC_VID3_FIR_COEF_HV_i <sup>(4)</sup>	RW	32	0x0000 0314 + (0x8 * i)	0x4804 1314 + (0x8 * i)	0x5800 1314 + (0x8 * i)
DISPC_VID3_FIR_COEF_V_i <sup>(4)</sup>	RW	32	0x0000 0350 + (0x4 * i)	0x4804 1350 + (0x4 * i)	0x5800 1350 + (0x4 * i)
DISPC_VID3_ATTRIBUTES	RW	32	0x0000 0370	0x4804 1370	0x5800 1370
DISPC_VID3_CONV_COEF0	RW	32	0x0000 0374	0x4804 1374	0x5800 1374
DISPC_VID3_CONV_COEF1	RW	32	0x0000 0378	0x4804 1378	0x5800 1378
DISPC_VID3_CONV_COEF2	RW	32	0x0000 037C	0x4804 137C	0x5800 137C
DISPC_VID3_CONV_COEF3	RW	32	0x0000 0380	0x4804 1380	0x5800 1380
DISPC_VID3_CONV_COEF4	RW	32	0x0000 0384	0x4804 1384	0x5800 1384
DISPC_VID3_BUF_SIZE_STATUS	R	32	0x0000 0388	0x4804 1388	0x5800 1388
DISPC_VID3_BUF_THRESHOLD	RW	32	0x0000 038C	0x4804 138C	0x5800 138C
DISPC_VID3_FIR	RW	32	0x0000 0390	0x4804 1390	0x5800 1390
DISPC_VID3_PICTURE_SIZE	RW	32	0x0000 0394	0x4804 1394	0x5800 1394
DISPC_VID3_PIXEL_INC	RW	32	0x0000 0398	0x4804 1398	0x5800 1398
DISPC_VID3_POSITION	RW	32	0x0000 039C	0x4804 139C	0x5800 139C
DISPC_VID3_PRELOAD	RW	32	0x0000 03A0	0x4804 13A0	0x5800 13A0
DISPC_VID3_ROW_INC	RW	32	0x0000 03A4	0x4804 13A4	0x5800 13A4
DISPC_VID3_SIZE	RW	32	0x0000 03A8	0x4804 13A8	0x5800 13A8
DISPC_DEFAULT_COLOR2	RW	32	0x0000 03AC	0x4804 13AC	0x5800 13AC
DISPC_TRANS_COLOR2	RW	32	0x0000 03B0	0x4804 13B0	0x5800 13B0
DISPC_CPR2_COEF_B	RW	32	0x0000 03B4	0x4804 13B4	0x5800 13B4
DISPC_CPR2_COEF_G	RW	32	0x0000 03B8	0x4804 13B8	0x5800 13B8
DISPC_CPR2_COEF_R	RW	32	0x0000 03BC	0x4804 13BC	0x5800 13BC
DISPC_DATA2_CYCLE1	RW	32	0x0000 03C0	0x4804 13C0	0x5800 13C0
DISPC_DATA2_CYCLE2	RW	32	0x0000 03C4	0x4804 13C4	0x5800 13C4
DISPC_DATA2_CYCLE3	RW	32	0x0000 03C8	0x4804 13C8	0x5800 13C8
DISPC_SIZE_LCD2	RW	32	0x0000 03CC	0x4804 13CC	0x5800 13CC
DISPC_TIMING_H2	RW	32	0x0000 0400	0x4804 1400	0x5800 1400
DISPC_TIMING_V2	RW	32	0x0000 0404	0x4804 1404	0x5800 1404
DISPC_POL_FREQ2	RW	32	0x0000 0408	0x4804 1408	0x5800 1408
DISPC_DIVISOR2	RW	32	0x0000 040C	0x4804 140C	0x5800 140C
DISPC_WB_ACCU_j <sup>(5)</sup>	RW	32	0x0000 0500 + (0x4 * j)	0x4804 1500 + (0x4 * j)	0x5800 1500 + (0x4 * j)
DISPC_WB_BA_j <sup>(5)</sup>	RW	32	0x0000 0508 + (0x4 * j)	0x4804 1508 + (0x4 * j)	0x5800 1508 + (0x4 * j)
DISPC_WB_FIR_COEF_H_i <sup>(6)</sup>	RW	32	0x0000 0510 + (0x8 * i)	0x4804 1510 + (0x8 * i)	0x5800 1510 + (0x8 * i)

<sup>(5)</sup> j = 0 to 1<sup>(6)</sup> i = 0 to 7



**Table 10-133. Display Controller Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DISPC L4_PER Physical Address	DISPC L3 Physical Address
DISPC_WB_FIR_COEF_HV_j <sup>(6)</sup>	RW	32	0x0000 0514 + (0x8 * i)	0x4804 1514 + (0x8 * i)	0x5800 1514 + (0x8 * i)
DISPC_WB_FIR_COEF_V_j <sup>(6)</sup>	RW	32	0x0000 0550 + (0x4 * i)	0x4804 1550 + (0x4 * i)	0x5800 1550 + (0x4 * i)
DISPC_WB_ATTRIBUTES	RW	32	0x0000 0570	0x4804 1570	0x5800 1570
DISPC_WB_CONV_COEF0	RW	32	0x0000 0574	0x4804 1574	0x5800 1574
DISPC_WB_CONV_COEF1	RW	32	0x0000 0578	0x4804 1578	0x5800 1578
DISPC_WB_CONV_COEF2	RW	32	0x0000 057C	0x4804 157C	0x5800 157C
DISPC_WB_CONV_COEF3	RW	32	0x0000 0580	0x4804 1580	0x5800 1580
DISPC_WB_CONV_COEF4	RW	32	0x0000 0584	0x4804 1584	0x5800 1584
DISPC_WB_BUF_SIZE_STATUS	R	32	0x0000 0588	0x4804 1588	0x5800 1588
DISPC_WB_BUF_THRESHOLD	RW	32	0x0000 058C	0x4804 158C	0x5800 158C
DISPC_WB_FIR	RW	32	0x0000 0590	0x4804 1590	0x5800 1590
DISPC_WB_PICTURE_SIZE	RW	32	0x0000 0594	0x4804 1594	0x5800 1594
DISPC_WB_PIXEL_INC	RW	32	0x0000 0598	0x4804 1598	0x5800 1598
DISPC_WB_ROW_INC	RW	32	0x0000 05A4	0x4804 15A4	0x5800 15A4
DISPC_WB_SIZE	RW	32	0x0000 05A8	0x4804 15A8	0x5800 15A8
DISPC_VID1_BA_UV_j <sup>(7)</sup>	RW	32	0x0000 0600 + (0x4 * j)	0x4804 1600 + (0x4 * j)	0x5800 1600 + (0x4 * j)
DISPC_VID2_BA_UV_j <sup>(7)</sup>	RW	32	0x0000 0608 + (0x4 * j)	0x4804 1608 + (0x4 * j)	0x5800 1608 + (0x4 * j)
DISPC_VID3_BA_UV_j <sup>(7)</sup>	RW	32	0x0000 0610 + (0x4 * j)	0x4804 1610 + (0x4 * j)	0x5800 1610 + (0x4 * j)
DISPC_WB_BA_UV_j <sup>(7)</sup>	RW	32	0x0000 0618 + (0x4 * j)	0x4804 1618 + (0x4 * j)	0x5800 1618 + (0x4 * j)
DISPC_CONFIG2	RW	32	0x0000 0620	0x4804 1620	0x5800 1620
DISPC_VID1_ATTRIBUTES2	RW	32	0x0000 0624	0x4804 1624	0x5800 1624
DISPC_VID2_ATTRIBUTES2	RW	32	0x0000 0628	0x4804 1628	0x5800 1628
DISPC_VID3_ATTRIBUTES2	RW	32	0x0000 062C	0x4804 162C	0x5800 162C
DISPC_GAMMA_TABLE0	W	32	0x0000 0630	0x4804 1630	0x5800 1630
DISPC_GAMMA_TABLE1	W	32	0x0000 0634	0x4804 1634	0x5800 1634
DISPC_GAMMA_TABLE2	W	32	0x0000 0638	0x4804 1638	0x5800 1638
DISPC_VID1_FIR2	RW	32	0x0000 063C	0x4804 163C	0x5800 163C
DISPC_VID1_ACCU2_j <sup>(7)</sup>	RW	32	0x0000 0640 + (0x4 * j)	0x4804 1640 + (0x4 * j)	0x5800 1640 + (0x4 * j)
DISPC_VID1_FIR_COEF_H2_j <sup>(8)</sup>	RW	32	0x0000 0648 + (0x8 * i)	0x4804 1648 + (0x8 * i)	0x5800 1648 + (0x8 * i)
DISPC_VID1_FIR_COEF_HV2_j <sup>(8)</sup>	RW	32	0x0000 064C + (0x8 * i)	0x4804 164C + (0x8 * i)	0x5800 164C + (0x8 * i)
DISPC_VID1_FIR_COEF_V2_j <sup>(8)</sup>	RW	32	0x0000 0688 + (0x4 * i)	0x4804 1688 + (0x4 * i)	0x5800 1688 + (0x4 * i)
DISPC_VID2_FIR2	RW	32	0x0000 06A8	0x4804 16A8	0x5800 16A8
DISPC_VID2_ACCU2_j <sup>(7)</sup>	RW	32	0x0000 06AC + (0x4 * j)	0x4804 16AC + (0x4 * j)	0x5800 16AC + (0x4 * j)
DISPC_VID2_FIR_COEF_H2_j <sup>(8)</sup>	RW	32	0x0000 06B4 + (0x8 * i)	0x4804 16B4 + (0x8 * i)	0x5800 16B4 + (0x8 * i)
DISPC_VID2_FIR_COEF_HV2_j <sup>(8)</sup>	RW	32	0x0000 06B8 + (0x8 * i)	0x4804 16B8 + (0x8 * i)	0x5800 16B8 + (0x8 * i)

<sup>(7)</sup> j = 0 to 1

<sup>(8)</sup> i = 0 to 7

**Table 10-133. Display Controller Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DISPC L4_PER Physical Address	DISPC L3 Physical Address
DISPC_VID2_FIR_COEF_V2_i <sup>(8)</sup>	RW	32	0x0000 06F4 + (0x4 * i)	0x4804 16F4 + (0x4 * i)	0x5800 16F4 + (0x4 * i)
DISPC_VID3_FIR2	RW	32	0x0000 0724	0x4804 1724	0x5800 1724
DISPC_VID3_ACCU2_j <sup>(7)</sup>	RW	32	0x0000 0728 + (0x4 * j)	0x4804 1728 + (0x4 * j)	0x5800 1728 + (0x4 * j)
DISPC_VID3_FIR_COEF_H2_i <sup>(8)</sup>	RW	32	0x0000 0730 + (0x8 * i)	0x4804 1730 + (0x8 * i)	0x5800 1730 + (0x8 * i)
DISPC_VID3_FIR_COEF_HV2_i <sup>(8)</sup>	RW	32	0x0000 0734 + (0x8 * i)	0x4804 1734 + (0x8 * i)	0x5800 1734 + (0x8 * i)
DISPC_VID3_FIR_COEF_V2_i <sup>(8)</sup>	RW	32	0x0000 0770 + (0x4 * i)	0x4804 1770 + (0x4 * i)	0x5800 1770 + (0x4 * i)
DISPC_WB_FIR2	RW	32	0x0000 0790	0x4804 1790	0x5800 1790
DISPC_WB_ACCU2_j <sup>(7)</sup>	RW	32	0x0000 0794 + (0x4 * j)	0x4804 1794 + (0x4 * j)	0x5800 1794 + (0x4 * j)
DISPC_WB_FIR_COEF_H2_i <sup>(8)</sup>	RW	32	0x0000 07A0 + (0x8 * i)	0x4804 17A0 + (0x8 * i)	0x5800 17A0 + (0x8 * i)
DISPC_WB_FIR_COEF_HV2_i <sup>(8)</sup>	RW	32	0x0000 07A4 + (0x8 * i)	0x4804 17A4 + (0x8 * i)	0x5800 17A4 + (0x8 * i)
DISPC_WB_FIR_COEF_V2_i <sup>(8)</sup>	RW	32	0x0000 07E0 + (0x4 * i)	0x4804 17E0 + (0x4 * i)	0x5800 17E0 + (0x4 * i)
DISPC_GLOBAL_BUFFER	RW	32	0x0000 0800	0x4804 1800	0x5800 1800
DISPC_DIVISOR	RW	32	0x0000 0804	0x4804 1804	0x5800 1804
DISPC_WB_ATTRIBUTES2	RW	32	0x0000 0810	0x4804 1810	0x5800 1810

<sup>(9)</sup> i = 0 to 7

10.2.7.3.2 Display Controller Register Description

Table 10-134. DISPC\_REVISION

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	IP Revision		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

Table 10-135. Register Call Summary for Register DISPC\_REVISION

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

Table 10-136. DISPC\_SYSCONFIG

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	This register allows to control various parameters of the OCP interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MIDLEMODE	RESERVED	CLOCKACTIVITY	RESERVED	WARMRESET	SIDLEMODE	ENWAKEUP	SOFTRESET	AUTOIDLE							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Write 0s for future compatibility. Reads returns 0	R	0x00000
13:12	MIDLEMODE	Master interface power management, standby/wait control 0x0: Force-standby.MStandby is only asserted when the module is disabled. MStandby is only asserted when the module is disabled. 0x1: No-Standby:MStandby is never asserted 0x2: Smart-Standby.MStandby is asserted based on the internal activity of the module 0x3: Reserved	RW	0x0
11:10	RESERVED	Write 0s for future compatibility. Reads returns 0	R	0x0
9:8	CLOCKACTIVITY	Clocks activity during wake up mode period 0x0: OCP and Functional clocks can be switched off 0x1: Functional clocks can be switched off and OCP clocks are maintained during wake up period 0x2: OCP clocks can be switched off and Functional clocks are maintained during wake up period 0x3: OCP and Functional clocks are maintained during wake up period	RW	0x0

Bits	Field Name	Description	Type	Reset
7:6	RESERVED	Write 0s for future compatibility. Reads returns 0	R	0x0
5	WARMRESET	Warm reset. Set this bit to 1 triggers a module warm reset. The bit is automatically reset by the hardware. During reads, it always returns 0. The warm reset keep the configuration registers unchanged. 0x0: Normal mode 0x1: the warmreset is set	RW	0
4:3	SIDLEMODE	Slave interface power management, Idle req/ack control 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module. 0x3: Reserved	RW	0x0
2	ENWAKEUP	WakeUp feature control 0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0x0: Normal mode 0x1: The module is reset	RW	0
0	AUTOIDLE	Internal OCP clock gating strategy 0x0: OCP clock is free-running 0x1: Automatic OCP L3 and L4 clocks gating strategy is applied, based on the OCP interface activity. Automatic functional clock gating is also applied to the functional clock based on the module activity (for instance DISPC_<pipe>_ATTRIBUTES.ENABLE)	RW	1

**Table 10-137. Register Call Summary for Register DISPC\_SYSCONFIG**

Display Controller

- [Software Reset: \[0\] \[1\]](#)
- [Power Management: \[2\] \[3\] \[4\]](#)
- [DMA Engine: \[5\]](#)
- [Shadows Registers: \[6\]](#)
- [Display Controller Registers: \[7\]](#)

**Table 10-138. DISPC\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014																																																														
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC																																																												
<b>Description</b>	This register provides status information about the module, excluding the interrupt status information.																																																														
<b>Type</b>	R																																																														
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> </thead> <tbody> <tr> <td colspan="16">RESERVED</td> <td colspan="11">RESERVED</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">RESETDONE</td> </tr> </tbody> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																RESERVED											RESETDONE
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
RESERVED																RESERVED											RESETDONE																																				
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																											
31:1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000																																																											
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is on-going Read 0x1: Reset completed	R	1																																																											

**Table 10-139. Register Call Summary for Register DISPC\_SYSSTATUS**

Display Controller

- Software Reset: [0]
- Shadows Registers: [1]
- Display Controller Registers: [2]

**Table 10-140. DISPC\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	DISPC
<b>Physical Address</b>	See Table 10-133.		
<b>Description</b>	This register regroups all the status of the module internal events that generate an interrupt. Write 1 to a given bit resets this bit		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED							WBBUFFEROVERFLOW_IRQ	FRAMEDONETV_IRQ	FRAMEDONEWB_IRQ	FRAMEDONE2_IRQ	ACBIASCOUNTSTATUS2_IRQ	VID3BUFFERUNDERFLOW_IRQ	VID3ENDWINDOW_IRQ	VSYNC2_IRQ	SYNCLOST2_IRQ	WAKEUP_IRQ	SYNCLOSTTV_IRQ	SYNCLOST1_IRQ	VID2ENDWINDOW_IRQ	VID2BUFFERUNDERFLOW_IRQ	VID1ENDWINDOW_IRQ	VID1BUFFERUNDERFLOW_IRQ	OCERROR_IRQ	PALETTEGAMMALOADING_IRQ	GFXENDWINDOW_IRQ	GFXBUFFERUNDERFLOW_IRQ	PROGRAMMEDLINENUMBER_IRQ	ACBIASCOUNTSTATUS1_IRQ	EVSYNC_ODD_IRQ	EVSYNC_EVEN_IRQ	VSYNC1_IRQ	FRAMEDONE1_IRQ

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Write 0s for future compatibility. Reads returns 0	R	0x00
25	WBBUFFER OVERFLOW_IRQ	Write-back DMA Buffer Overflow. The DMA buffer is full. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
24	FRAME DONETV_IRQ	Frame Done for the TV. The TV output has been disabled by user. All the data have been sent. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
23	FRAME DONEWB_IRQ	Frame Done for the write-back channel. The write-back channel has output the frame. All the data of the frame have been sent to the memory. There is no pending data inside the DMA engine for the write-back channel to be transferred to memory. It is available only when the write-back pipeline transfers back to memory the output of one of the pipelines. In case of overlay capture, the interrupt is not generated and the user shall use the FrameDone for the corresponding captured output. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
22	FRAME DONE2_IRQ	Frame Done for the secondary LCD. The secondary LCD output has been disabled by user. All the data have been sent. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
21	ACBIASCOUNT STATUS2_IRQ	AC Bias Count Status for the secondary LCD 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
20	VID3BUFFER UNDERFLOW_IRQ	Video 3 DMA Buffer Underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses)  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
19	VID3END WINDOW_IRQ	The end of the video 3 Window has been reached. It is detected by the overlay manager when the full video 3 has been displayed.  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
18	VSYNC2_IRQ	Vertical Synchronization for the secondary LCD  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
17	SYNC LOST2_IRQ	Synchronization Lost on the secondary LCD output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the secondary LCD output.  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
16	WAKEUP_IRQ	Wake-up  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
15	SYNCLOST TV_IRQ	Synchronization Lost on the TV output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the TV output.  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
14	SYNC LOST1_IRQ	Synchronization Lost on the primary LCD output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the primary LCD output.  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
13	VID2END WINDOW_IRQ	The end of the video 2 Window has been reached. It is detected by the overlay manager when the full video 2 has been displayed.  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
12	VID2BUFFER UNDERFLOW_IRQ	Video 2 DMA Buffer Underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses)  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
11	VID1END WINDOW_IRQ	The end of the video 1 Window has been reached. It is detected by the overlay manager when the full video 1 has been displayed.  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
10	VID1BUFFER UNDERFLOW_IRQ	Video 1 DMA Buffer Underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses)  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0
9	OCPERROR_IRQ	OCP Error. L3 Inteconnect has sent SResp=ERR.  0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
8	PALETTEGAMMA LOADING_IRQ	<p>Palette Gamma Loading status. The palette used as Color Look Up Table (CLUT) for the graphics BITMAP formats (1-, 2-, 4-, or 4-bpp) or as gamma table for the overlay output for the primary LCD output has been loaded successfully.</p> <p>0x0: READS: Event is false. WRITES: Status bit unchanged.</p> <p>0x1: READS: Event is true (pending). WRITES: Status bit is reset.</p>	RW W1toClr	0
7	GFXEND WINDOW_IRQ	<p>The end of the graphics Window has been reached. It is detected by the overlay manager when the full graphics has been displayed.</p> <p>0x0: READS: Event is false. WRITES: Status bit unchanged.</p> <p>0x1: READS: Event is true (pending). WRITES: Status bit is reset.</p>	RW W1toClr	0
6	GFXBUFFER UNDERFLOW_IRQ	<p>Graphics DMA Buffer Underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses)</p> <p>0x0: READS: Event is false. WRITES: Status bit unchanged.</p> <p>0x1: READS: Event is true (pending). WRITES: Status bit is reset.</p>	RW W1toClr	0
5	PROGRAMMED LINENUMBER_IRQ	<p>Programmed Line Number. It indicates that the scan of the primary LCD has reached the programmed user line number.</p> <p>0x0: READS: Event is false. WRITES: Status bit unchanged.</p> <p>0x1: READS: Event is true (pending). WRITES: Status bit is reset.</p>	RW W1toClr	0
4	ACBIASCOUNT STATUS1_IRQ	<p>AC Bias Count Status for the primary LCD</p> <p>0x0: READS: Event is false. WRITES: Status bit unchanged.</p> <p>0x1: READS: Event is true (pending). WRITES: Status bit is reset.</p>	RW W1toClr	0
3	EVSYNC_ ODD_IRQ	<p>VSYNC for odd field from the TV encoder (VENC or HDMI)</p> <p>0x0: READS: Event is false. WRITES: Status bit unchanged.</p> <p>0x1: READS: Event is true (pending). WRITES: Status bit is reset.</p>	RW W1toClr	0
2	EVSYNC_ EVEN_IRQ	<p>VSYNC for even field from the TV encoder (VENC or HDMI)</p> <p>0x0: READS: Event is false. WRITES: Status bit unchanged.</p> <p>0x1: READS: Event is true (pending). WRITES: Status bit is reset.</p>	RW W1toClr	0
1	VSYNC1_IRQ	<p>Vertical Synchronization for the primary LCD.</p> <p>0x0: READS: Event is false. WRITES: Status bit unchanged.</p> <p>0x1: READS: Event is true (pending). WRITES: Status bit is reset.</p>	RW W1toClr	0
0	FRAME DONE1_IRQ	<p>Frame Done for the primary LCD. The primary LCD output has been disabled by user. All the data have been sent.</p> <p>0x0: READS: Event is false. WRITES: Status bit unchanged.</p> <p>0x1: READS: Event is true (pending). WRITES: Status bit is reset.</p>	RW W1toClr	0

**Table 10-141. Register Call Summary for Register DISPC\_IRQSTATUS**

Display Controller

- [Interrupt Request: \[0\]](#)
- [LCD Outputs: \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-142. DISPC\_IRQENABLE**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	This register allows to mask/unmask the module internal sources of interrupt, on an event-by-event basis		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED								WBBUFFEROVERFLOW_EN	FRAMEDONETV_EN	FRAMEDONEWB_EN	FRAMEDONE2_EN	ACBIASCOUNTSTATUS2_EN	VID3BUFFERUNDERFLOW_EN	VID3ENDWINDOW_EN	VSYNC2_EN	SYNCLOST2_EN	WAKEUP_EN	SYNCLOSTTV_EN	SYNCLOST1_EN	VID2ENDWINDOW_EN	VID2BUFFERUNDERFLOW_EN	ENDVID1WINDOW_EN	VID1BUFFERUNDERFLOW_EN	OCERROR_EN	PALETTEGAMMA_EN	GFXENDWINDOW_EN	GFXBUFFERUNDERFLOW_EN	PROGRAMMEDLINENUMBER_EN	ACBIASCOUNTSTATUS1_EN	EVSYNC_ODD_EN	EVSYNC_EVEN_EN	VSYNC1_EN	FRAMEDONE_EN

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Write 0s for future compatibility. Reads returns 0	R	0x00
25	WBBUFFER OVERFLOW_EN	Write-back DMA Buffer Overflow. The DMA buffer is full 0x0: WBufferOverflow is masked 0x1: WBufferOverflow generates an interrupt when it occurs	RW	0
24	FRAME DONETV_EN	Frame Done for the TV. The TV output has been disabled by user. All the data have been sent. 0x0: Frame Done for the TV output is masked 0x1: Frame Done for the TV output generates an interrupt when it occurs	RW	0
23	FRAME DONEWB_EN	Frame Done for the write-back channel. The write-back channel has output the frame. All the data have been sent for the frame have been sent to the memory. There is no pending data inside the DMA engine for the write-back channel to be transferred to memory. 0x0: Frame Done for the write-back is masked 0x1: Frame Done for the write-back generates an interrupt when it occurs	RW	0
22	FRAME DONE2_EN	Frame Done for the secondary LCD. The secondary LCD output has been disabled by user. All the data have been sent. 0x0: Frame Done for the secondary LCD is masked 0x1: Frame Done for the secondary LCD generates an interrupt when it occurs	RW	0
21	ACBIASCOUNT STATUS2_EN	AC Bias Count Status for the secondary LCD 0x0: ACBiasCountStatus for the secondary LCD output is masked 0x1: ACBiasCountStatus for the secondary LCD output generates an interrupt when it occurs	RW	0
20	VID3BUFFER UNDERFLOW_EN	Video 3 DMA Buffer Underflow. The DMA buffer is not necessary empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: Vid3BufferUnderflow is masked 0x1: Vid3BufferUnderflow generates an interrupt when it occurs	RW	0
19	VID3END WINDOW_EN	The end of the video 3 Window has been reached. It is detected by the overlay manager when the full video 3 has been displayed. 0x0: Vid3EndWindow is masked 0x1: Vid3EndWindow generates an interrupt when it occurs	RW	0



Bits	Field Name	Description	Type	Reset
18	VSYNC2_EN	Vertical Synchronization for the secondary LCD 0x0: VSYNC for the secondary LCD output is masked 0x1: VSYNC for the secondary LCD output generates an interrupt when it occurs	RW	0
17	SYNC LOST2_EN	Synchronization Lost on the secondary LCD output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the secondary LCD output. 0x0: Synchronization Lost on the secondary LCD output is masked 0x1: Synchronization Lost on the secondary LCD output generates an interrupt when it occurs	RW	0
16	WAKEUP_EN	Wake Up Mask 0x0: WakeUp is masked 0x1: WakeUp generates an interrupt when it occurs	RW	0
15	SYNC LOSTTV_EN	Synchronization Lost on the TV output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with the TV output. 0x0: Synchronization Lost on the TV output is masked 0x1: Synchronization Lost on the TV output generates an interrupt when it occurs	RW	0
14	SYNC LOST1_EN	Synchronization Lost for the primary LCD 0x0: SyncLost for the primary LCD output is masked 0x1: SyncLost for the primary LCD output generates an interrupt when it occurs	RW	0
13	VID2END WINDOW_EN	The end of the video 2 Window has been reached. It is detected by the overlay manager when the full video 2 has been displayed. 0x0: Vid2EndWindow is masked 0x1: Vid2EndWindow generates an interrupt when it occurs	RW	0
12	VID2BUFFER UNDERFLOW_EN	Video 2 DMA Buffer Underflow. The DMA buffer is not necessary empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: Vid2BufferUnderflow is masked 0x1: Vid2BufferUnderflow generates an interrupt when it occurs	RW	0
11	ENDVID1 WINDOW_EN	The end of the video 1 Window has been reached. It is detected by the overlay manager when the full video 1 has been displayed. 0x0: EndVid1Window is masked 0x1: EndVid1Window generates an interrupt when it occurs	RW	0
10	VID1BUFFER UNDERFLOW_EN	Video 1 DMA Buffer Underflow. The DMA buffer is not necessary empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: Vid1BufferUnderflow is masked 0x1: Vid1BufferUnderflow generates an interrupt when it occurs	RW	0
9	OCPERROR_EN	OCP Error. L3 Inteconnect has sent SResp=ERR. 0x0: OCPErrror is masked 0x1: OCPErrror generates an interrupt when it occurs	RW	0
8	PALETTE GAMMA_EN	Palette Gamma Loading mask. The palette used as Color Look Up Table (CLUT) for the graphics BITMAP formats (1-, 2-, 4-, or 4-bpp) or as gamme table for the overlay output for the primary LCD output has been loaded succesfully. 0x0: PaletteGamma is masked 0x1: PaletteGamma generates an interrupt when it occurs	RW	0
7	GFXEND WINDOW_EN	The end of the graphics Window has been reached. It is detected by the overlay manager when the full graphics has been displayed. 0x0: GfxEndWindow is masked 0x1: GfxEndWindow generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
6	GFXBUFFER UNDERFLOW_EN	Graphics DMA Buffer Underflow. The DMA buffer is not necessarily empty but required data are not present in the DMA buffer (due to out of order responses)  0x0: GfxBufferUnderflow is masked 0x1: GfxBufferUnderflow generates an interrupt when it occurs	RW	0
5	PROGRAMMED LINENUMBER_EN	Programmed Line Number. It indicates that the scan of the primary LCD has reached the programmed user line number.  0x0: ProgrammedLineNumber is masked 0x1: ProgrammedLineNumber generates an interrupt when it occurs	RW	0
4	ACBIASCOUNT STATUS1_EN	AC Bias Count Status for the primary LCD  0x0: ACBiasCountStatus for the primary LCD output is masked 0x1: ACBiasCountStatus for the primary LCD output generates an interrupt when it occurs	RW	0
3	EVSYNC_ODD_EN	VSYNC for odd field from the TV encoder (VENC or HDMI)  0x0: EVSYNC_ODD for the TV output is masked 0x1: EVSYNC_ODD for the TV output generates an interrupt when it occurs	RW	0
2	EVSYNC_EVEN_EN	VSYNC for even field from the TV encoder (VENC or HDMI)  0x0: EVSYNC_EVEN for the TV output is masked 0x1: EVSYNC_EVEN for the TV output generates an interrupt when it occurs	RW	0
1	VSYN1_EN	Vertical Synchronization for the primary LCD.  0x0: VSYNC for the primary LCD output is masked 0x1: VSYNC for the primary LCD output generates an interrupt when it occurs	RW	0
0	FRAMEDONE_EN	Frame Done for the primary LCD. The primary LCD output has been disabled by user. All the data have been sent.  0x0: FrameDone for the primary LCD output is masked 0x1: FrameDone for the primary LCD output generates an interrupt when it occurs	RW	0

**Table 10-143. Register Call Summary for Register DISPC\_IRQENABLE**

Display Controller

- [Interrupt Request: \[0\]](#)
- [LCD Outputs: \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-144. DISPC\_CONTROL1**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The control register configures the Display Controller module for the primary LCD and TV outputs.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPATIALTEMPORALDITHERINGFRAMES																															
LCDENABLEPOL																															
LCDENABLESIGNAL																															
PCKFREEENABLE																															
TDMUNUSEDBITS																															
TDMCYCLEFORMAT																															
TDMPARALLELMODE																															
TDMENABLE																															
HT																															
GPOUT1																															
GPOUT0																															
GPIN1																															
GPIN0																															
OVERLAYOPTIMIZATION																															
STALLMODE																															
RESERVED																															
TFTDATALINES																															
STDITHERENABLE																															
GOTV																															
GOLCD																															
M8B																															
STNTFT																															
MONOCOLOR																															
TVENABLE																															
LCDENABLE																															

Bits	Field Name	Description	Type	Reset
31:30	SPATIALTEMPORAL DITHERINGFRAMES	Spatial/Temporal dithering number of frames for the primary LCD output wr: VFP start period of primary LCD  0x0: Spatial only 0x1: Spatial and temporal over 2 frames 0x2: Spatial and temporal over 4 frames 0x3: Reserved	RW	0x0
29	LCDENABLEPOL	Write 0s for future compatibility. Reads return 0.	R	0
28	LCDENABLESIGNAL	Write 0s for future compatibility. Reads return 0.	R	0
27	PCKFREEENABLE	Write 0s for future compatibility. Reads return 0.	R	0
26:25	TDMUNUSEDBITS	State of unused bits (TDM mode only) for the primary LCD output. wr: VFP start period of primary LCD  0x0: low level (0) 0x1: high level (1) 0x2: unchanged from previous state 0x3: reserved	RW	0x0
24:23	TDMCYCLEFORMAT	Cycle format (TDM mode only) for the primary LCD output wr: VFP start period of primary LCD  0x0: 1 cycle for 1 pixel 0x1: 2 cycles for 1 pixel 0x2: 3 cycles for 1 pixel 0x3: 3 cycles for 2 pixels	RW	0x0
22:21	TDMPARALLELMODE	Output Interface width (TDM mode only) for the primary LCD output wr: VFP start period of primary LCD  0x0: 8-bit parallel output interface selected 0x1: 9-bit parallel output interface selected 0x2: 12-bit parallel output interface selected 0x3: 16-bit parallel output interface selected	RW	0x0
20	TDMENABLE	Enable the multiple cycle format (TDM mode only used for TFT mode with the RFBI enable bit off) for the primary LCD output. wr: VFP start period of primary LCD  0x0: TDM disabled 0x1: TDM enabled	RW	0
19:17	HT	Hold Time for TV output wr: EVSYNC Encoded value (from 1 to 8) to specify the number of external digital clock periods to hold the data (programmed value = value minus 1)	RW	0x0

## Display Controller

www.ti.com

Bits	Field Name	Description	Type	Reset
16	GPOUT1	General Purpose Output Signal wr:immediate 0x0: The GPout1 is reset 0x1: The GPout1 is set	RW	0
15	GPOUT0	General Purpose Output Signal wr:immediate 0x0: The GPout0 is reset 0x1: The GPout0 is set	RW	0
14	GPIN1	General Purpose Input Signal wr: immediately Read 0x0: The GPin1 has been reset Read 0x1: The GPin1 has been set	R	0
13	GPIN0	General Purpose Input Signal wr: immediately Read 0x0: The GPin0 has been reset Read 0x1: The GPin0 has been set	R	0
12	OVERLAYOPTI MIZATION	Overlay Optimization for the primary LCD output wr: VFP start period of the primary LCD  0x0: All the data for all the enabled pipelines are fetched from memory regardless of the overlay/alpha blending configuration. 0x1: The data not used by the overlay manager because of overlap between layers with no alpha blending between them shall not be fetched from memory in order to optimize the bandwidth.	RW	0
11	STALLMODE	STALL Mode for the primary LCD output wr: VFP start period of primary LCD 0x0: Normal mode selected 0x1: STALL mode selected. The Display Controller sends the data without considering the VSYNC/HSYNC. The LCD output is disabled at the end of the transfer of the frame. The S/W has to re-enable the LCD output in order to generate a new frame. The stall mode is used in RFBI and DSI command modes.	RW	0
10	RESERVED	Reserved for non-GP devices only	R	0
9:8	TFTDATALINES	Number of lines of the primary LCD interface wr: VFP start period of primary LCD 0x0: 12-bit output aligned on the LSB of the pixel data interface 0x1: 16-bit output aligned on the LSB of the pixel data interface 0x2: 18-bit output aligned on the LSB of the pixel data interface 0x3: 24-bit output aligned on the LSB of the pixel data interface	RW	0x0
7	STDITHERENABLE	Spatial Temporal dithering enable for the primary LCD output wr: VFP start period of primary LCD 0x0: Spatial/Temporal dithering logic disabled 0x1: Spatial/Temporal dithering logic enabled	RW	0
6	GOTV	GO Command for the TV output. It is used to synchronized the pipelines (graphics and/or video ones) associated with the TV output. wr: immediate 0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) associated with the TV output using the user values. The hardware resets the bit when the update is completed. 0x1: The user has finished to program the shadow registers of the pipeline(s) associated with the TV output and the hardware can update the internal registers at the external VSYNC.	RW	0
5	GOLCD	GO Command for the primary LCD output. It is used to synchronized the pipelines (graphics and/or video ones) associated with the primary LCD output. wr: immediate 0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) connected to the LCD output using the user values. The hardware resets the bit when the update is completed. 0x1: The user has finished to program the shadow registers of the pipeline(s) associated with the LCD output and the hardware can update the internal registers at the VFP start period	RW	0

Bits	Field Name	Description	Type	Reset
4	M8B	Mono 8-bit mode of the primary LCD wr: VFP start period of primary LCD output  0x0: Pixel data [3:0] is used to output four pixel values to the panel at each pixel clock transition. (only in Passive Mono 4-bit mode).  0x1: Pixel data [7:0] is used to output eight pixel values to the panel each pixel clock transition. (only in Passive Mono 8-bit mode).	RW	0
3	STNTFT	LCD Display type of the primary LCD wr: VFP start period of primary LCD output  0x0: Passive or STN display operation enabled. STN dither logic is enabled.  0x1: Active or TFT display operation enabled. STN Dither logic and output FIFO bypassed.	RW	0
2	MONOCOLOR	Monochrome/Color selection for the primary LCD wr: VFP start period of primary LCD output  0x0: Color operation enabled (STN mode only)  0x1: Monochrome operation enabled (STN mode only)	RW	0
1	TVENABLE	Enable the TV output wr: immediate effect only occurs at the end of the current frame.  0x0: TV output disabled (at the end of the current field if interlace output when the bit is reset)  0x1: TV output enabled	RW	0
0	LCDENABLE	Enable the primary LCD outputs wr: immediate Effect only occurs at the end of the current frame  0x0: LCD output disabled (at the end of the frame when the bit is reset)  0x1: LCD output enabled	RW	0

**Table 10-145. Register Call Summary for Register DISPC\_CONTROL1**

Display Controller

- [LCD Outputs: \[0\]](#)
- [Shadows Registers: \[1\] \[2\] \[3\] \[4\]](#)
- [Display Controller Low-level Programming Models: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [Display Controller Logical Register Mapping: \[14\]](#)
- [Display Controller Registers: \[15\]](#)

**Table 10-146. DISPC\_CONFIG1**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The control register configures the Display Controller module for the primary LCD output and TV output. Shadow register, updated on VFP start period of primary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FULLRANGE	COLORCONVENABLE	FIDFIRST	OUTPUTMODEENABLE	RESERVED	TVALPHABLENDERENABLE	LCDALPHABLENDERENABLE	BUFFERFILLING	BUFFERHANDCHECK	CPR	BUFFERMERGE	TCKTVSELECTION	TCKTVENABLE	TCKLCDSELECTION	TCKLCDENABLE	GAMATABLEENABLE	ACBIASGATED	VSYNCGATED	HSYNGGATED	PIXELCLOCKGATED	PIXELDATAGATED	PALETTEGAMMATABLE	LOADMODE	PIXELGATED

## Display Controller

www.ti.com

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
25	FULLRANGE	Color Space Conversion full range setting. wr: VFP start of primary LCD 0x0: Limited range selected. 0x1: Full range selected.	RW	0
24	COLORCONV ENABLE	Enable the color space conversion. It shall be reset when CPR bit field is set to 0x1. wr: VFP start of primary LCD 0x0: Disable Color Space Conversion RGB to YUV 0x1: Enable Color Space Conversion RGB to YUV	RW	0
23	FIDFIRST	Selects the first field to output in case of interlace mode. In case of progressive mode, the value is not used. wr: VFP start of primary LCD 0x0: First field is even. 0x1: Odd field is first.	RW	0
22	OUTPUTMODE ENABLE	Selects between progressive and interlace mode for the primary LCD output. wr: VFP start of primary LCD 0x0: Progressive mode selected. 0x1: Interlace mode selected.	RW	0
21:20	RESERVED	Write 0s for future compatibility. Reads return 0.	RW	0
19	TVALPHABLENDER ENABLE	Selects the alpha blender overlay manager for the TV output instead of the color key alpha blender (LCD output). The bit field is deprecated. It is present for software backward compatibility only. When it is enabled, the Z-order defined in each ATTRIBUTES registers for only the pipelines associated pipeline connected to the TV output are invalid and replaced by the following: graphics z-order = 3, video3 z-order = 2, video2 z-order = 1 and video1 z-order=0 If it disabled, the z-order and z-order enable bit fields defined in each ATTRIBUTS register are used. wr: EVSYNC start of primary LCD 0x0: Alpha blender is disabled. 0x1: The alpha blender is enabled.	RW	0
18	LCDALPHABLENDER ENABLE	Selects the alpha blender overlay manager for the primary LCD output instead of the color key alpha blender (LCD output). The bit field is deprecated. It is present for software backward compatibility only. When it is enabled, the Z-order defined in each ATTRIBUTES registers for only the pipelines associated with the primary LCD output are invalid and replaced by the following: graphics z-order = 3, video3 z-order = 2, video2 z-order = 1 and video1 z-order=0 If it disabled, the z-order and z-order enable bit fields defined in each ATTRIBUTES register are used. wr: VFP start of primary LCD 0x0: Alpha blender is disabled. The color key alpha blending is used. 0x1: The alpha blender is enabled.	RW	0
17	BUFFERFILLING	Controls if the DMA buffers are refilled only when the LOW threshold is reached or if all DMA buffers are refilled when at least one of them reaches the LOW threshold. wr: immediate 0x0: Each DMA buffer is refilled when it reaches LOW threshold. 0x1: All DMA buffers are refilled up to high threshold when at least one of them reaches the LOW threshold. (only active DMA buffers shall be considered and when reaching the end of the frame the DMA buffer goes to empty condition so no need to fill it again).	RW	0
16	BUFFERHAND CHECK	Controls the handshake between DMA buffer and STALL signal in order to prevent from underflow. The bit shall be set to 0 when the module is not in STALL mode. (primary LCD output) wr: VFP start of primary LCD 0x0: Only the STALL signal (generated by RFBI, DSI1 or DSI2 depending on which IP uses the LCD output) is used regardless of the DMA buffer fullness information in order to provide data to the RFBI, DSI1 or DS2 module. 0x1: The STALL signal (generated by RFBI, DSI1 or DSI2 depending on which IP uses the LCD output) is used in combination with the DMA buffer fullness information in order to provide data to the RFBI, DSI1 or DSI2 module only when it does not generated buffer underflow.	RW	0

Bits	Field Name	Description	Type	Reset
15	CPR	Color Phase Rotation Control (primary LCD output). It shall be reset when ColorConvEnable bit field is set to 1 wr: VFP start period of primary LCD output  0x0: Color Phase Rotation Disabled 0x1: Color Phase Rotation Enabled	RW	0
14	BUFFERMERGE	Buffer merge control wr: EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory or VFP When enabled, the <a href="#">DISPC_GLOBAL_BUFFER</a> register is ignored. This bit must be set to zero when the write back channel is used. When <a href="#">DISPC_CONTROL2.GOWB</a> is used BUFFERMERGE MUST be zero. When <a href="#">DISPC_CONTROL2.GOWB</a> is used BUFFERMERGE MUST be zero. wr: immediate  0x0: DMA buffer merge disabled Each DMA buffer is dedicated to one pipeline. 0x1: DMA buffer merge enabled All the DMA buffers are merged into a single one to be used by the single active pipeline.	RW	0
13	TCKTV SELECTION	Transparency Color Key Selection (TV output) wr: EVSYNC 0x0: Destination transparency color key selected 0x1: Source transparency color key selected	RW	0
12	TCKTVENABLE	Transparency Color Key Enabled (TV output) wr: EVSYNC 0x0: Disable the transparency color key for the TV output 0x1: Enable the transparency color key for the TV output	RW	0
11	TCKLCD SELECTION	Transparency Color Key Selection (primary LCD output) wr: VFP start period of primary LCD output 0x0: Destination transparency color key selected 0x1: Source transparency color key selected	RW	0
10	TCKLCDENABLE	Transparency Color Key Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: Disable the transparency color key for the LCD 0x1: Enable the transparency color key for the LCD	RW	0
9	GAMATABLE ENABLE	For backward compatibility, an enable bit has been added on the 2 additional gamma tables (secondary display & TV). Gama table of LCD1 is always enabled. 0x0: Gamma table LDC2 & TV are bypassed 0x1: Gamma table LCD2 & TV are enabled	RW	0
8	ACBIASGATED	ACBias Gated Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: AcBias Gated Disabled 0x1: AcBias Gated Enabled	RW	0
7	VSYNCGATED	VSYNC Gated Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: VSYNC Gated Disabled 0x1: VSYNC Gated Enabled	RW	0
6	HSYNCGATED	HSYNC Gated Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: HSYNC Gated Disabled 0x1: HSYNC Gated Enabled	RW	0
5	PIXELCLOCK GATED	Pixel Clock Gated Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: Pixel Clock Gated Disabled 0x1: Pixel Clock Gated Enabled	RW	0

Bits	Field Name	Description	Type	Reset
4	PIXELDATAGATED	Pixel Data Gated Enabled (primary LCD output) wr: VFP start period of primary LCD output 0x0: Pixel Data Gated Disabled 0x1: Pixel Data Gated Enabled	RW	0
3	PALETTEGAMMA TABLE	Palette/Gamma Table selection wr: VFP start period of primary LCD output or VFP start period of secondary LCD output or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the graphics pipeline: primary LCD, secondary LCD, TV output or write-back to the memory. In case of the table is used as gamma table, it is used for the primary LCD output only. 0x0: LUT used as palette (only if graphics format is BITMAP1, 2, 4, and 8) 0x1: LUT used as gamma table (only if graphics format is NOT BITMAP1, 2, 4, and 8 or no graphics window present)	RW	0
2:1	LOADMODE	Loading Mode for the Palette/Gamma Table wr: VFP start period of primary LCD output or VFP start period of secondary LCD output or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory 0x0: Palette/Gamma Table and data are loaded every frame 0x1: Palette/Gamma Table to be loaded. The user sets the bit when the palette/gamma table has to be loaded. Hardware resets the bit to 0x2 when table has been loaded. ( <a href="#">DISPC_GFX_ATTRIBUTES.ENABLE</a> has to be set to 1). 0x2: Frame data only loaded every frame 0x3: Palette/Gamma Table and frame data loaded on first frame then switch to 0x2 (Hardware).	RW	0x0
0	PIXELGATED	Pixel Gated Enable (only for TFT) (primary LCD output) wr: VFP start period of primary LCD output 0x0: Pixel clock always toggles (only in TFT mode) 0x1: Pixel clock only toggles when there is valid data to display. (only in TFT mode)	RW	0

**Table 10-147. Register Call Summary for Register DISPC\_CONFIG1**

## Display Controller

- [Power Management: \[0\]](#)
- [Graphics Pipeline: \[1\] \[2\]](#)
- [LCD Outputs: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [Shadows Registers: \[9\]](#)
- [Display Controller Low-level Programming Models: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\]](#)
- [Display Controller Logical Register Mapping: \[21\]](#)
- [Display Controller Registers: \[22\]](#)



**Table 10-148. DISPC\_DEFAULT\_COLOR0**

<b>Address Offset</b>	0x0000 004C	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The control register allows to configure the default solid background color for the primary LCD. Shadow register, updated on VFP start period of the primary LCD	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEFAULTCOLOR																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	DEFAULTCOLOR	24-bit RGB color value to specify the default solid color to display when there is no data from the overlays.	RW	0x000000

**Table 10-149. Register Call Summary for Register DISPC\_DEFAULT\_COLOR0**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Logical Register Mapping: \[1\]](#)
  - [Display Controller Registers: \[2\]](#)

**Table 10-150. DISPC\_DEFAULT\_COLOR1**

<b>Address Offset</b>	0x0000 0050	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The control register allows to configure the default solid background color for the TV output. Shadow register, updated on EVSYNC	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEFAULTCOLOR																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	DEFAULTCOLOR	24-bit RGB color value to specify the default solid color to display when there is no data from the overlays.	RW	0x000000

**Table 10-151. Register Call Summary for Register DISPC\_DEFAULT\_COLOR1**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Low-level Programming Models: \[1\]](#)
  - [Display Controller Logical Register Mapping: \[2\]](#)
  - [Display Controller Registers: \[3\]](#)

**Table 10-152. DISPC\_TRANS\_COLOR0**

<b>Address Offset</b>	0x0000 0054	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register sets the transparency color value for the video/graphics overlays for the primary LCD output. Shadow register, updated on VFP start period of the primary LCD	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TRANSCOLORKEY																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	TRANSCOLORKEY	Transparency Color Key Value in RGB format [0] BITMAP 1 (CLUT), [23,1] set to 0s [1:0] BITMAP 2 (CLUT), [23,2] set to 0s [3:0] BITMAP 4 (CLUT), [23,4] set to 0s [7:0] BITMAP 8 (CLUT), [23,8] set to 0s [11:0] RGB 12, [23,12] set to 0s [15:0] RGB 16, [23,16] set to 0s [23:0] RGB 24	RW	0x000000

**Table 10-153. Register Call Summary for Register DISPC\_TRANS\_COLOR0**

Display Controller

- [LCD Outputs: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Logical Register Mapping: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-154. DISPC\_TRANS\_COLOR1**

<b>Address Offset</b>	0x0000 0058	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register sets the transparency color value for the video/graphics overlays for the TV output. Shadow register, updated on EVSYNC	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TRANSCOLORKEY																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	TRANSCOLORKEY	Transparency Color Key Value in RGB format [0] BITMAP 1 (CLUT), [23,1] set to 0s [1:0] BITMAP 2 (CLUT), [23,2] set to 0s [3:0] BITMAP 4 (CLUT), [23,4] set to 0s [7:0] BITMAP 8 (CLUT), [23,8] set to 0s [11:0] RGB 12, [23,12] set to 0s [15:0] RGB 16, [23,16] set to 0s [23:0] RGB 24	RW	0x000000

**Table 10-155. Register Call Summary for Register DISPC\_TRANS\_COLOR1**

Display Controller

- [LCD Outputs: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Low-level Programming Models: \[2\]](#)
- [Display Controller Logical Register Mapping: \[3\]](#)
- [Display Controller Registers: \[4\]](#)

**Table 10-156. DISPC\_LINE\_STATUS**

<b>Address Offset</b>	0x0000 005C	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The control register indicates the current primary LCD panel display line number.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINENUMBER															

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
10:0	LINENUMBER	Current LCD panel line number Current display line number. The first active line has the value 0. During blanking lines the line number is not incremented.	R	0x000

**Table 10-157. Register Call Summary for Register DISPC\_LINE\_STATUS**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-158. DISPC\_LINE\_NUMBER**

<b>Address Offset</b>	0x0000 0060	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The control register indicates the primary LCD panel display line number for the interrupt and the DMA request. Shadow register, updated on VFP start period of primary LCD.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINENUMBER															

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
10:0	LINENUMBER	LCD panel line number programming LCD line number defines the line on which the programmable interrupt is generated and the DMA request occurs.	RW	0x000

**Table 10-159. Register Call Summary for Register DISPC\_LINE\_NUMBER**

Display Controller

- [System DMA Request: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-160. DISPC\_TIMING\_H1**

<b>Address Offset</b>	0x0000 0064	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the timing logic for the HSYNC signal. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBP								HFP								HSW															

Bits	Field Name	Description	Type	Reset
31:20	HBP	Horizontal Back Porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display (program to value minus 1). When in BT mode and interlaced, this field corresponds to the vertical field blanking No 2 for Even Field.	RW	0x000
19:8	HFP	Horizontal front porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the end of a line transmission before line clock is asserted (program to value minus 1). When in BT mode and interlaced, this field corresponds to the vertical field blanking No 1 for Even Field.	RW	0x000
7:0	HSW	Horizontal synchronization pulse width Encoded value (from 1 to 256) to specify the number of pixel clock periods to pulse the line clock at the end of each line (program to value minus 1). When in BT mode, this field corresponds to the horizontal blanking	RW	0x00

**Table 10-161. Register Call Summary for Register DISPC\_TIMING\_H1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-162. DISPC\_TIMING\_V1**

<b>Address Offset</b>	0x0000 0068	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the timing logic for the VSYNC signal. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBP								VFP								VSW															

Bits	Field Name	Description	Type	Reset
31:20	VBP	Vertical back porch Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the beginning of a frame.	RW	0x000
19:8	VFP	Vertical front porch Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the end of each frame.	RW	0x000
7:0	VSW	Vertical synchronization pulse width In active mode, encoded value (from 1 to 256) to specify the number of line clock periods (program to value minus 1) to pulse the frame clock (VSYNC) pin at the end of each frame after the end of frame wait (VFP) period elapses. Frame clock uses as VSYNC signal in active mode. In passive mode, encoded value (from 1 to 256) to specify the number of extra line clock periods (program to value minus 1) to insert after the vertical front porch (VFP) period has elapsed.	RW	0x00

**Table 10-163. Register Call Summary for Register DISPC\_TIMING\_V1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-164. DISPC\_POL\_FREQ1**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the signal configuration. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														ALIGN	ONOFF	RF	IEO	IPC	IHS	IVS	ACBI				ACB						

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
18	ALIGN	Defines the alignment between HSYNC and VSYNC assertion. 0x0: VSYNC and HSYNC are not aligned 0x1: VSYNC and HSYNC assertions are aligned.	RW	0
17	ONOFF	HSYNC/VSYNC Pixel clock Control On/Off 0x0: HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data 0x1: HSYNC and VSYNC are driven according to bit 16	RW	0
16	RF	Program HSYNC/VSYNC Rise or Fall 0x0: HSYNC and VSYNC are driven on falling edge of pixel clock (if bit 17 set to 1) 0x1: HSYNC and VSYNC are driven on rising edge of pixel clock (if bit 17 set to 1)	RW	0
15	IEO	Invert output enable 0x0: Ac-bias is active high (active display mode) 0x1: Ac-bias is active low (active display mode)	RW	0
14	IPC	Invert pixel clock 0x0: Data is driven on the LCD data lines on the rising-edge of the pixel clock 0x1: Data is driven on the LCD data lines on the falling-edge of the pixel clock	RW	0
13	IHS	Invert HSYNC 0x0: Line clock pin is active high and inactive low 0x1: Line clock pin is active low and inactive high	RW	0
12	IVS	Invert VSYNC 0x0: Frame clock pin is active high and inactive low 0x1: Frame clock pin is active low and inactive high	RW	0
11:8	ACBI	AC Bias Pin transitions per interrupt Value (from 0 to 15) used to specify the number of AC Bias pin transitions	RW	0x0
7:0	ACB	AC Bias Pin Frequency Value (from 0 to 255) used to specify the number of line clocks to count before transitioning the AC Bias pin. This pin is used to periodically invert the polarity of the power supply to prevent DC charge build-up within the display.	RW	0x00

**Table 10-165. Register Call Summary for Register DISPC\_POL\_FREQ1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-166. DISPC\_DIVISOR1**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the divisors. It is used for the primary LCD output Shadow register, updated on VFP start period of primary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LCD								RESERVED								PCD							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:16	LCD	Display Controller Logic Clock Divisor Value (from 1 to 255) to specify the intermediate pixel clock frequency based on the LCD1_CLK. The value 0 is invalid.	RW	0x04
15:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
7:0	PCD	Pixel Clock Divisor Value (from 1 to 255) to specify the frequency of the pixel clock based on the LCD1_CLK divided by <a href="#">DISPC_DIVISOR1.LCD</a> value. The values 0 is invalid.	RW	0x01

**Table 10-167. Register Call Summary for Register DISPC\_DIVISOR1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 10-168. DISPC\_GLOBAL\_ALPHA**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register defines the global alpha value for the graphics and three video pipelines. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory for each bit field depending on the association of the each pipeline with the primary LCD, secondary LCD or TV output.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VID3GLOBALALPHA								VID2GLOBALALPHA								VID1GLOBALALPHA								GFXGLOBALALPHA							

Bits	Field Name	Description	Type	Reset
31:24	VID3GLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xFF
23:16	VID2GLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xFF
15:8	VID1GLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xFF

Bits	Field Name	Description	Type	Reset
7:0	GFXGLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xFF

**Table 10-169. Register Call Summary for Register DISPC\_GLOBAL\_ALPHA**

Display Controller

- [LCD Outputs: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Low-level Programming Models: \[2\] \[3\]](#)
- [Display Controller Registers: \[4\]](#)

**Table 10-170. DISPC\_SIZE\_TV**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the size of the TV output field (interlace), frame (progressive) (horizontal and vertical). Shadow register, updated on EVSYNC. A delta value is used to indicate if the odd field has same vertical size as the even field or +/- one line.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LPP								DELTA_LPP	RESERVED	PPL													

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	LPP	Lines per panel Encoded value (from 1 to 2048) to specify the number of lines per panel.	RW	0x000
15:14	DELTA_LPP	Indicates the delta size value of the odd field compared to the even field 0x0: Same size 0x1: Odd size = Even size +1 0x2: Odd size = Even Size -1	RW	0x0
13:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
10:0	PPL	Pixels per line Encoded value (from 1 to 2048) to specify the number of pixels contains within each line on the display.	RW	0x000

**Table 10-171. Register Call Summary for Register DISPC\_SIZE\_TV**

Display Controller

- [TV Output: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Shadows Registers: \[6\]](#)
- [Display Controller Low-level Programming Models: \[7\] \[8\] \[9\]](#)
- [Display Controller Registers: \[10\] \[11\]](#)

**Table 10-172. DISPC\_SIZE\_LCD1**

<b>Address Offset</b>	0x0000 007C	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the panel size (horizontal and vertical). Shadow register, updated on VFP start period of primary LCD. A delta value is used to indicate if the odd field has same vertical size as the even field or +/- one line.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LPP								DELTA_LPP	RESERVED	PPL													

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	LPP	Lines per panel Encoded value (from 1 to 2048) to specify the number of lines per panel (program to value minus 1).	RW	0x000
15:14	DELTA_LPP	Indicates the delta size value of the odd field compared to the even field 0x0: same size 0x1: Odd size = Even size +1 0x2: Odd size = Even Size -1	RW	0x0
13:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
10:0	PPL	Pixels per line Encoded value (from 1 to 2048) to specify the number of pixels contains within each line on the display (program to value minus 1). In STALL mode, any value is valid. In non STALL mode, only values multiple of 8 pixels are valid.	RW	0x000

**Table 10-173. Register Call Summary for Register DISPC\_SIZE\_LCD1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\] \[2\]](#)
- [Display Controller Registers: \[3\] \[4\]](#)

**Table 10-174. DISPC\_GFX\_BA\_j**

<b>Address Offset</b>	0x0000 0080 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the base address of the graphics buffer displayed in the graphics window (0 & 1 :for ping-pong mechanism with external trigger, based on the field polarity, 0 only used when graphics pipeline on the LCD output and 0 & 1 when on the TV output). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															



Bits	Field Name	Description	Type	Reset
31:0	BA	Graphics base address Base address of the graphics buffer (aligned on pixel size boundary) (in case 1-, 2-, and 4-bpp, byte alignment is required, in case of RGB24 packed format, 4-pixel alignment is required) When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

**Table 10-175. Register Call Summary for Register DISPC\_GFX\_BA\_j**

Display Controller

- [DMA Engine: \[0\]](#)
- [Rotation and Mirroring: \[1\] \[2\] \[3\] \[4\]](#)
- [Shadows Registers: \[5\]](#)
- [Display Controller Low-level Programming Models: \[6\]](#)
- [Display Controller Registers: \[7\]](#)

**Table 10-176. DISPC\_GFX\_POSITION**

<b>Address Offset</b>	0x0000 0088																																																																	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC																																																																
<b>Description</b>	The register configures the position of the graphics window. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td style="background-color:yellow;">23</td><td style="background-color:yellow;">22</td><td style="background-color:yellow;">21</td><td style="background-color:yellow;">20</td><td style="background-color:yellow;">19</td><td style="background-color:yellow;">18</td><td style="background-color:yellow;">17</td><td style="background-color:yellow;">16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td style="background-color:yellow;">7</td><td style="background-color:yellow;">6</td><td style="background-color:yellow;">5</td><td style="background-color:yellow;">4</td><td style="background-color:yellow;">3</td><td style="background-color:yellow;">2</td><td style="background-color:yellow;">1</td><td style="background-color:yellow;">0</td> </tr> <tr> <td colspan="8">RESERVED</td> <td colspan="8">POSY</td> <td colspan="8">RESERVED</td> <td colspan="8">POSX</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED								POSY								RESERVED								POSX							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
RESERVED								POSY								RESERVED								POSX																																										
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00																																																														
26:16	POSY	Y position of the graphics window. Encoded value (from 0 to 2047) to specify the Y position of the graphics window on the screen. The line at the top has the Y-position 0.	RW	0x000																																																														
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00																																																														
10:0	POSX	X position of the graphics window. Encoded value (from 0 to 2047) to specify the X position of the graphics window on the screen. The first pixel on the left of the screen has the X-position 0.	RW	0x000																																																														

**Table 10-177. Register Call Summary for Register DISPC\_GFX\_POSITION**

Display Controller

- [LCD Outputs: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Low-level Programming Models: \[2\] \[3\]](#)
- [Display Controller Registers: \[4\]](#)

**Table 10-178. DISPC\_GFX\_SIZE**

<b>Address Offset</b>	0x0000 008C		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the size of the graphics window. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SIZEY								RESERVED								SIZEX							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	SIZEY	Number of lines of the graphics window. Encoded value (from 1 to 2048) to specify the number of lines of the graphics window (program to value minus 1).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	SIZESX	Number of pixels of the graphics window. Encoded value (from 1 to 2048) to specify the number of pixels per line of the graphics window (program to value minus 1).	RW	0x000

**Table 10-179. Register Call Summary for Register DISPC\_GFX\_SIZE**

Display Controller

- [DMA Engine: \[0\]](#)
- [LCD Outputs: \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Low-level Programming Models: \[3\] \[4\]](#)
- [Display Controller Registers: \[5\]](#)

**Table 10-180. DISPC\_GFX\_ATTRIBUTES**

<b>Address Offset</b>	0x0000 00A0		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the graphics attributes. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANNELOUT2	BURSTTYPE	PREMULTIPLYALPHA	ZORDER	ZORDERENABLE	ANTIFLICKER	RESERVED								SELFREFRESHAUTO	RESERVED	SELFREFRESH	ARBITRATION	ROTATION	BUFPRELOAD	RESERVED	NIBBLEMODE	CHANNELOUT	BURSTSIZE	REPLICATIONENABLE	FORMAT				ENABLE		

Bits	Field Name	Description	Type	Reset
31:30	CHANNELOUT2	It is not used if CHANNELOUT is set to TV. Reserved when CHANNELOUT = 1 (should set to zero) wr: immediate  0x0: primary LCD output selected. 0x1: Secondary LCD output selected. 0x3: Write-back output to the memory selected.	RW	0x0
29	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targeted by the DMA engine. (It does not apply to the palette loading OCP requests using INCR burst only)  0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0
28	PREMULTIPLYALPHA	The field configures the DISPC GFX to process incoming data as premultiplied alpha data or non premultiplied alpha data. Default setting is non premultiplied alpha data.  0x0: Non premultiplied alpha data color component 0x1: Premultiplied alpha data color component	RW	0
27:26	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is software responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. If bit 25 is set to 0, the ZORDER bit field is ignored and replaced by the value 0.  0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x3: Z-order 3: layer above all the other layers 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3	RW	0x0
25	ZORDERENABLE	Z-order Enable. The bit field ZORDER is only used when the Z-order is enabled.  0x0: Z-order disabled. The Z-order of the layer is 0. 0x1: Z-order enabled. The Z-order is defined by the bit field ZORDER (bits 26 and 27).	RW	0
24	ANTIFLICKER	Antiflicker filtering using a 3-tap filter with hardcoded coefficients (1/4, 1/2, 1/4)  0x0: Antiflicker disabled. 0x1: Antiflicker enabled.	RW	0
23:18	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
17	SELFREFRESHAUTO	Automatic self-refresh mode  0x0: The transition from Selfrefresh "disabled" to "enabled" is controlled by software 0x1: The transition from Selfrefresh "disabled" to "enabled" is controlled only by hardware	RW	0
16	RESERVED		R	0
15	SELFREFRESH	Enables the self refresh of the graphics window from its own DMA buffer. This bit should be set only after having set the GO bit of the channel and read back a zero in its field.  0x0: The graphics pipeline accesses the interconnect to fetch data from the system memory. 0x1: The graphics pipeline does not need anymore to fetch data from memory. Only the graphics DMA buffer is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0

Bits	Field Name	Description	Type	Reset
14	ARBITRATION	<p>Determines the priority of the graphics pipeline. When the graphics pipeline is one of the high priority pipelines. The arbitration wheel gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them.</p> <p>0x0: The graphics pipeline is one of the normal priority pipeline.</p> <p>0x1: The graphics pipeline is one of the high priority pipeline.</p>	RW	0
13:12	ROTATION	<p>Graphics Rotation Flag</p> <p>0x0: No rotation</p> <p>0x1: Rotation by 90 degrees</p> <p>0x3: Rotation by 270 degrees</p> <p>0x2: Rotation by 180 degrees</p>	RW	0x0
11	BUFPRELOAD	<p>Graphics Preload Value</p> <p>0x0: Hardware prefetches pixels up to the preload value defined in the preload register</p> <p>0x1: Hardware prefetches pixels up to high threshold value</p>	RW	0
10	RESERVED	<p>Write 0s for future compatibility. Reads return 0.</p>	R	0
9	NIBBLEMODE	<p>Graphics Nibble Mode (only for 1-, 2- and 4-bpp)</p> <p>0x0: Nibble mode is disabled</p> <p>0x1: Nibble mode is enabled</p>	RW	0
8	CHANNELOUT	<p>Graphics Channel Out configuration: LCD, WB or TV. wr: immediate</p> <p>0x0: LCD output or WB to the memory selected. bit fields 31 and 30 defines the output associated (primary, secondary or write-back).</p> <p>0x1: TV output selected</p>	RW	0
7:6	BURSTSIZE	<p>Graphics DMA Burst Size</p> <p>0x0: 2x128bit bursts</p> <p>0x1: 4x128bit bursts</p> <p>0x3: Reserved</p> <p>0x2: 8x128bit bursts</p>	RW	0x2
5	REPLICATIONENABLE	<p>Graphics Replication Enabled: RGB, ARGB, and RGBA formats are converted into ARGB32-8888 using replication of the MSBs or '0s</p> <p>0x0: Disable Graphics replication logic. The conversion to ARGB32-8888 is done by adding 0s for the LSBs</p> <p>0x1: Enable Graphics replication logic. The conversion to ARGB32-8888 is done by duplicating the MSBs for the LSBs</p>	RW	1

Bits	Field Name	Description	Type	Reset
4:1	FORMAT	<p>Graphics format. It defines the pixel format when fetching the graphics picture into memory.</p> <p>0x6: RGB16-565</p> <p>0x1: BITMAP2 (CLUT is required to be used)</p> <p>0xA: RGBx12-4444</p> <p>0x7: ARGB16-1555</p> <p>0xD: RGBA32-8888</p> <p>0x0: BITMAP1 (CLUT is required to be used)</p> <p>0x2: BITMAP4 (CLUT is required to be used)</p> <p>0x8: xRGB24-8888 (32-bit container)</p> <p>0x9: RGB24-888 (24-bit container)</p> <p>0xB: RGBA12-4444</p> <p>0x4: xRGB12-4444</p> <p>0x5: ARGB16-4444</p> <p>0xF: xRGB15-1555</p> <p>0xC: ARGB32-8888</p> <p>0x3: BITMAP8 (CLUT is required to be used)</p> <p>0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)</p>	RW	0x0
0	ENABLE	<p>Graphics Enable</p> <p>0x0: Graphics disabled (graphics pipeline inactive and graphics window not present)</p> <p>0x1: Graphics enabled (graphics pipeline active and graphics window present on the screen)</p>	RW	0

**Table 10-181. Register Call Summary for Register DISPC\_GFX\_ATTRIBUTES**

Display Controller

- [DMA Engine: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Memory Format: \[5\]](#)
- [Graphics Pipeline: \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [LCD Outputs: \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [Shadows Registers: \[16\]](#)
- [Display Controller Low-level Programming Models: \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\]](#)
- [Display Controller Registers: \[33\] \[34\]](#)

**Table 10-182. DISPC\_GFX\_BUF\_THRESHOLD**

<b>Address Offset</b>	0x0000 00A4																																																																
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .																																																																
<b>Description</b>	The register configures the graphics buffer. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory																																																																
<b>Type</b>	RW																																																																
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">31</td><td style="text-align: center;">30</td><td style="text-align: center;">29</td><td style="text-align: center;">28</td><td style="text-align: center;">27</td><td style="text-align: center;">26</td><td style="text-align: center;">25</td><td style="text-align: center;">24</td><td style="text-align: center;">23</td><td style="text-align: center;">22</td><td style="text-align: center;">21</td><td style="text-align: center;">20</td><td style="text-align: center;">19</td><td style="text-align: center;">18</td><td style="text-align: center;">17</td><td style="text-align: center;">16</td><td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td colspan="16" style="text-align: center;">BUFHIGHTHRESHOLD</td> <td colspan="16" style="text-align: center;">BUFLOWTHRESHOLD</td> </tr> </table>		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																		
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD																																																	

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x04FF
15:0	BUFLOWTHRESHOLD	DMA buffer Low Threshold Number of 128-bits defining the threshold value. The value put in this register should always be greater than zero.	RW	0x04F8

**Table 10-183. Register Call Summary for Register DISPC\_GFX\_BUF\_THRESHOLD**

Display Controller

- [DMA Engine: \[0\] \[1\] \[2\]](#)
- [Shadows Registers: \[3\]](#)
- [Display Controller Low-level Programming Models: \[4\] \[5\]](#)
- [Display Controller Registers: \[6\]](#)

**Table 10-184. DISPC\_GFX\_BUF\_SIZE\_STATUS**

<b>Address Offset</b>	0x0000 00A8		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register defines the Graphics buffer size		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	DMA buffer Size in number of 128-bits	R	0x0500

**Table 10-185. Register Call Summary for Register DISPC\_GFX\_BUF\_SIZE\_STATUS**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-186. DISPC\_GFX\_ROW\_INC**

<b>Address Offset</b>	0x0000 00AC		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the number of bytes to increment at the end of the row. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded unsigned value to specify the number of bytes to increment at the end of the row in the graphics buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*bpp means increment of n pixels. The value 1- (n+1)*bpp means decrement of n pixels.	RW	0x0000 0001

**Table 10-187. Register Call Summary for Register DISPC\_GFX\_ROW\_INC**

Display Controller

- [DMA Engine: \[0\] \[1\] \[2\]](#)
- [Rotation and Mirroring: \[3\] \[4\] \[5\] \[6\]](#)
- [Shadows Registers: \[7\]](#)
- [Display Controller Low-level Programming Models: \[8\]](#)
- [Display Controller Registers: \[9\]](#)

**Table 10-188. DISPC\_GFX\_PIXEL\_INC**

<b>Address Offset</b>	0x0000 00B0	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the number of bytes to increment between two pixels. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PIXELINC															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000000
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between two pixels in the graphics buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*bpp means increment of n pixels.	RW	0x01

**Table 10-189. Register Call Summary for Register DISPC\_GFX\_PIXEL\_INC**

Display Controller

- [DMA Engine: \[0\] \[1\]](#)
- [Rotation and Mirroring: \[2\] \[3\] \[4\] \[5\]](#)
- [Shadows Registers: \[6\]](#)
- [Display Controller Low-level Programming Models: \[7\]](#)
- [Display Controller Registers: \[8\]](#)

**Table 10-190. DISPC\_GFX\_WINDOW\_SKIP**

<b>Address Offset</b>	0x0000 00B4	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	Write 0s for future compatibility. Reads return 0.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WINDOWSKIP																															

Bits	Field Name	Description	Type	Reset
31:0	WINDOWSKIP	Number of bytes to skip during the video window 1. Encoded signed value (from -231-1 to 231) to specify the number of bytes to skip in the graphics buffer when the video window 1 is displayed on top of the graphics and no transparency color is enabled.	R	0x0000 0000

**Table 10-191. Register Call Summary for Register DISPC\_GFX\_WINDOW\_SKIP**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-192. DISPC\_GFX\_TABLE\_BA**

<b>Address Offset</b>	0x0000 00B8	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the base address of the palette buffer or the gamma table buffer. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABLEBA																															

Bits	Field Name	Description	Type	Reset
31:0	TABLEBA	Base address of the palette/gamma table buffer (24-bit entries in 32-bit containers, aligned on 32-bit boundary).	RW	0x0000 0000

**Table 10-193. Register Call Summary for Register DISPC\_GFX\_TABLE\_BA**

Display Controller

- [Graphics Pipeline: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Low-level Programming Models: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-194. DISPC\_VID1\_BA\_j**

<b>Address Offset</b>	0x0000 00BC + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the base address of the video buffer for the video window 1 (DISPC_VID1_BA_0 & DISPC_VID1_BA_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID1_BA_0 is used). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															



Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address Base address of the video buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV4:2:2, 2-pixel alignment is required, and YUV4:2:0, byte alignment is supported)). It case of YUV4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

**Table 10-195. Register Call Summary for Register DISPC\_VID1\_BA\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-196. DISPC\_VID1\_POSITION**

<b>Address Offset</b>	0x0000 00C4																																																																		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC																																																																
<b>Description</b>	The register configures the position of the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory																																																																		
<b>Type</b>	RW																																																																		
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="8">RESERVED</td> <td colspan="8">POSY</td> <td colspan="8">RESERVED</td> <td colspan="8">POSX</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED								POSY								RESERVED								POSX							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
RESERVED								POSY								RESERVED								POSX																																											
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																															
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00																																																															
26:16	POSY	Y position of the video window 1 Encoded value (from 0 to 2047) to specify the Y position of the video window 1 .The line at the top has the Y-position 0.	RW	0x000																																																															
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00																																																															
10:0	POSX	X position of the video window 1 Encoded value (from 0 to 2047) to specify the X position of the video window 1. The first pixel on the left of the display screen has the X-position 0.	RW	0x000																																																															

**Table 10-197. Register Call Summary for Register DISPC\_VID1\_POSITION**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-198. DISPC\_VID1\_SIZE**

<b>Address Offset</b>	0x0000 00C8		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the size of the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SIZEY								RESERVED								SIZEX							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	SIZEY	Number of lines of the video 1 Encoded value (from 1 to 2048) to specify the number of lines of the video window 1. Program to value minus 1.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	SIZEX	Number of pixels of the video window 1 Encoded value (from 1 to 2048) to specify the number of pixels of the video window 1. Program to value minus 1.	RW	0x000

**Table 10-199. Register Call Summary for Register DISPC\_VID1\_SIZE**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-200. DISPC\_VID1\_ATTRIBUTES**

<b>Address Offset</b>	0x0000 00CC		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the attributes of the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANNELOUT2	BURSTTYPE	PREMULTIPLYALPHA	ZORDER	ZORDERENABLE	SELFREFRESH	ARBITRATION	DOUBLESTRIDE	VERTICALTAPS	DMAOPTIMIZATION	BUFPRELOAD	RESERVED	SELFREFRESHAUTO	CHANNELOUT	BURSTSIZE	ROTATION	FULLRANGE	REPLICATIONENABLE	COLORCONVENABLE	VRESIZECONF	HRESIZECONF	RESIZEENABLE	FORMAT					ENABLE				

Bits	Field Name	Description	Type	Reset
31:30	CHANNELOUT2	It is not used if CHANNELOUT is set to TV. Reserved when CHANNELOUT = 1 (should be set to zero) wr: immediate  0x0: primary LCD output selected. 0x1: Secondary LCD output selected. 0x3: Write-back output to the memory selected.	RW	0x0
29	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targeted by the DMA engine.  0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0
28	PREMULTIPHYALPHA	The field configures the DISPC VID1 to process incoming data as pre-multiplied alpha data or non pre-multiplied alpha data. Default setting is non pre-multiplied alpha data.  0x0: Non pre-multiplied alpha data color component 0x1: Pre-multiplied alpha data color component	RW	0
27:26	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is software responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. If bit 25 is set to 0, the ZORDER bit field is ignored and replaced by the value 0.  0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x3: Z-order 3: layer above all the other layers 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3	RW	0x0
25	ZORDERENABLE	Z-order Enable. The bit field ZORDER is only used when the Z-order is enabled.  0x0: Z-order disabled. The Z-order of the layer is 0. 0x1: Z-order enabled. The Z-order is defined by the bit field ZORDER (bits 26 and 27).	RW	0
24	SELFREFRESH	Enables the self refresh of the video window from its own DMA buffer only.  0x0: The video pipeline accesses the interconnect to fetch data from the system memory. 0x1: The video pipeline does not need anymore to fetch data from memory. Only the DMA buffer associated with the video1 is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0
23	ARBITRATION	Determines the priority of the video pipeline. The video pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them.  0x0: The video pipeline is one of the normal priority pipeline. 0x1: The video pipeline is one of the high priority pipeline.	RW	0
22	DOUBLESTRIDE	Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride. It is only used in case of YUV4:2:0.  0x0: The CbCr stride value is equal to the Y stride. 0x1: The CbCr stride value is double to the Y stride.	RW	0

Bits	Field Name	Description	Type	Reset
21	VERTICALTAPS	Video Vertical Resize Tap Number. The vertical polyphase filter can be configured in 3-tap or 5-tap configuration. According to the number of taps, the maximum input picture width is double while using 3-tap compared to 5-tap.  0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. The associated bit fields for the 2 other taps coefficients do not need to be initialized.  0x1: 5 taps are used for the vertical filtering logic.	RW	0
20	DMAOPTIMIZATION	Write 0s for future compatibility. Reads return 0.	R	0
19	BUFPRELOAD	Video Preload Value  0x0: Hardware prefetches pixels up to the preload value defined in the preload register 0x1: Hardware prefetches pixels up to high threshold value	RW	0
18	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
17	SELFREFRESHAUTO	Automatic self-refresh mode  0x0: The transition from SELFREFRESH "disabled" to "enabled" is controlled by SW. 0x1: The transition from SELFREFRESH "disabled" to "enabled" is controlled only by hardware.	RW	0
16	CHANNELOUT	Video Channel Out configuration: LCD, WB or TV. wr: immediate  0x0: LCD output or WB to the memory selected. bit fields 31 and 30 defines the output associated (primary, secondary or write-back). 0x1: TV output selected	RW	0
15:14	BURSTSIZE	Video DMA Burst Size  0x0: 2x128bit bursts 0x1: 4x128bit bursts 0x3: Reserved 0x2: 8x128bit bursts	RW	0x2
13:12	ROTATION	Video Rotation Flag  0x0: No rotation 0x1: Rotation by 90 degrees 0x3: Rotation by 270 degrees 0x2: Rotation by 180 degrees	RW	0x0
11	FULLRANGE	Color Space Conversion full range setting.  0x0: Limited range selected: 16 subtracted from Y before color space conversion 0x1: Full range selected: Y is not modified before the color space conversion	RW	0
10	REPLICATIONENABLE	Replication Enable  0x0: Disable Video replication logic 0x1: Enable Video replication logic	RW	1
9	COLORCONVENABLE	Enable the color space conversion. The hardware does not enable/disable the conversion based on the pixel format. The bit field shall be reset when the format is not YUV. 0x0: Disable Color Space Conversion YUV to RGB 0x1: Enable Color Space Conversion YUV to RGB	RW	0
8	VRESIZECONF	Write 0s for future compatibility. Reads return 0.	R	0

Bits	Field Name	Description	Type	Reset
7	HRESIZECONF	Write 0s for future compatibility. Reads return 0.	R	0
6:5	RESIZEENABLE	Video Resize Enable 0x0: Disable both horizontal and vertical resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0
4:1	FORMAT	Video Format. It defines the pixel format when fetching the video 1 picture into memory. 0x6: RGB16-565 0x1: RGB12x-4444 0xA: YUV2 4:2:2 co-sited 0x7: ARGB16-1555 0xD: RGBA32-8888 0x0: NV12 4:2:0 2 buffers (Y + UV) 0x2: RGBA12-4444 0x8: xRGB24-8888 (32-bit container) 0x9: RGB24-888 (24-bit container) 0xB: UYVY 4:2:2 co-sited 0x5: ARGB16-4444 0xF: xRGB15-1555 0xC: ARGB32-8888 0x4: xRGB12-4444 0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)	RW	0x0
0	ENABLE	Video Enable 0x0: Video disabled (video pipeline inactive and window not present) 0x1: Video enabled (video pipeline active and window present on the screen)	RW	0

**Table 10-201. Register Call Summary for Register DISPC\_VID1\_ATTRIBUTES**

Display Controller

- [LCD Outputs: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Logical Register Mapping: \[2\]](#)
- [Display Controller Registers: \[3\] \[4\] \[5\] \[6\]](#)

**Table 10-202. DISPC\_VID1\_BUF\_THRESHOLD**

<b>Address Offset</b>	0x0000 00D0		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the video buffer associated with the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	Video DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x07FF
15:0	BUFLOWTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x07F8

**Table 10-203. Register Call Summary for Register DISPC\_VID1\_BUF\_THRESHOLD**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-204. DISPC\_VID1\_BUF\_SIZE\_STATUS**

<b>Address Offset</b>	0x0000 00D4		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register defines the Video buffer size for the video pipeline 1.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	Video 1 DMA buffer Size in number of 128-bits	R	0x0800

**Table 10-205. Register Call Summary for Register DISPC\_VID1\_BUF\_SIZE\_STATUS**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-206. DISPC\_VID1\_ROW\_INC**

<b>Address Offset</b>	0x0000 00D8	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the number of bytes to increment at the end of the row for the buffer associated with the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded signed value (from $2^{31}1$ to $2^{31}$ ) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value $1 + n * \text{bpp}$ means increment of n pixels. The value $1 (n + 1) * \text{bpp}$ means decrement of n pixels.	RW	0x0000 0001

**Table 10-207. Register Call Summary for Register DISPC\_VID1\_ROW\_INC**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-208. DISPC\_VID1\_PIXEL\_INC**

<b>Address Offset</b>	0x0000 00DC	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the number of bytes to increment between two pixels for the buffer associated with the video window 2. The register is used only when the TILER is not present in the system in order to perform low performance rotation. When the TILER IP is present it is highly recommended to use it for performing the rotation. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PIXELINC															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between two pixels in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value $1+n*\text{bpp}$ means increment of n pixels. For YUV4:2:0, Max supported value is 128.	RW	0x01

**Table 10-209. Register Call Summary for Register DISPC\_VID1\_PIXEL\_INC**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-210. DISPC\_VID1\_FIR**

<b>Address Offset</b>	0x0000 00E0	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 1. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVINC								RESERVED								FIRHINC							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

**Table 10-211. Register Call Summary for Register DISPC\_VID1\_FIR**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-212. DISPC\_VID1\_PICTURE\_SIZE**

<b>Address Offset</b>	0x0000 00E4	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the size of the video picture associated with the video layer 1 before up/down-scaling. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ORGSIZEY								RESERVED								ORGSIZEX							



Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	ORGSIZEY	Number of lines of the video picture. Encoded value (from 1 to 2048) to specify the number of lines of the video picture in memory (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded to 2 <sup>11</sup> .	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	ORGSIZEX	Number of pixels of the video picture Encoded value (from 1 to 2048) to specify the number of pixels of the video picture in memory (program to value minus 1). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit. (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded to 2 <sup>11</sup> .	RW	0x000

**Table 10-213. Register Call Summary for Register DISPC\_VID1\_PICTURE\_SIZE**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-214. DISPC\_VID1\_ACCU\_j**

<b>Address Offset</b>	0x0000 00E8 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 1 (DISPC_VID1_ACCU_0 & DISPC_VID1_ACCU_1 for ping-pong mechanism with external trigger, based on the field polarity) It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU								RESERVED								HORIZONTALACCU							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from -1024 to 1023).	RW	0x000

**Table 10-215. Register Call Summary for Register DISPC\_VID1\_ACCU\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-216. DISPC\_VID1\_FIR\_COEF\_H\_i**

<b>Address Offset</b>	0x0000 00F0 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-217. Register Call Summary for Register DISPC\_VID1\_FIR\_COEF\_H\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-218. DISPC\_VID1\_FIR\_COEF\_HV\_i**

<b>Address Offset</b>	0x0000 00F4 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-219. Register Call Summary for Register DISPC\_VID1\_FIR\_COEF\_HV\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-220. DISPC\_VID1\_CONV\_COEF0**

<b>Address Offset</b>	0x0000 0130	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RCR								RESERVED				RY											

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	RCR	RCr Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RY	RY Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-221. Register Call Summary for Register DISPC\_VID1\_CONV\_COEF0**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-222. DISPC\_VID1\_CONV\_COEF1**

<b>Address Offset</b>	0x0000 0134	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								GY								RESERVED				RCB											

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	GY	GY Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RCB	RCb Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-223. Register Call Summary for Register DISPC\_VID1\_CONV\_COEF1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-224. DISPC\_VID1\_CONV\_COEF2**

<b>Address Offset</b>	0x0000 0138	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	RESERVED	GCB RESERVED GCR
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.
26:16	GCB	GCb Coefficient Encoded signed value (from -1024 to 1023).
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.
10:0	GCR	GCr Coefficient Encoded signed value (from -1024 to 1023).
	<b>Type</b>	<b>Reset</b>
	R	0x00
	RW	0x000
	R	0x00
	RW	0x000

**Table 10-225. Register Call Summary for Register DISPC\_VID1\_CONV\_COEF2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-226. DISPC\_VID1\_CONV\_COEF3**

<b>Address Offset</b>	0x0000 013C	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	RESERVED	BCR RESERVED BY
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.
26:16	BCR	BCr coefficient Encoded signed value (from -1024 to 1023).
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.
10:0	BY	BY coefficient Encoded signed value (from -1024 to 1023).
	<b>Type</b>	<b>Reset</b>
	R	0x00
	RW	0x000
	R	0x00
	RW	0x000

**Table 10-227. Register Call Summary for Register DISPC\_VID1\_CONV\_COEF3**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Logical Register Mapping: \[1\]](#)
  - [Display Controller Registers: \[2\]](#)

**Table 10-228. DISPC\_VID1\_CONV\_COEF4**

<b>Address Offset</b>	0x0000 0140	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCB															

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
10:0	BCB	BCb Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-229. Register Call Summary for Register DISPC\_VID1\_CONV\_COEF4**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Logical Register Mapping: \[1\]](#)
  - [Display Controller Registers: \[2\]](#)

**Table 10-230. DISPC\_VID2\_BA\_j**

<b>Address Offset</b>	0x0000 014C + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the base address of the video buffer for the video window 2 (DISPC_VID2_BA_0 & DISPC_VID2_BA_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID2_BA_0 is used). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address Base address of the video buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV4:2:2, 2-pixel alignment is required, and YUV4:2:0, byte alignment is supported). It case of YUV4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

**Table 10-231. Register Call Summary for Register DISPC\_VID2\_BA\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-232. DISPC\_VID2\_POSITION**

<b>Address Offset</b>	0x0000 0154	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the position of the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	RESERVED	POSY RESERVED POSX
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.
26:16	POSY	Y position of the video window 2 Encoded value (from 0 to 2047) to specify the Y position of the video window 2 .The line at the top has the Y-position 0.
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.
10:0	POSX	X position of the video window 2 Encoded value (from 0 to 2047) to specify the X position of the video window 2. The first pixel on the left of the display screen has the X-position 0.

**Table 10-233. Register Call Summary for Register DISPC\_VID2\_POSITION**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-234. DISPC\_VID2\_SIZE**

<b>Address Offset</b>	0x0000 0158																														
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .								<b>Instance</b>	DISPC																					
<b>Description</b>	The register configures the size of the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SIZEY								RESERVED								SIZEX							
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>														<b>Type</b>	<b>Reset</b>														
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.														R	0x00														
26:16	SIZEY	Number of lines of the video 2 Encoded value (from 1 to 2048) to specify the number of lines of the video window 2. Program to value minus 1.														RW	0x000														
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.														R	0x00														
10:0	SIZEX	Number of pixels of the video window 2 Encoded value (from 1 to 2048) to specify the number of pixels of the video window 2. Program to value minus 1.														RW	0x000														

**Table 10-235. Register Call Summary for Register DISPC\_VID2\_SIZE**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Logical Register Mapping: \[1\]](#)
  - [Display Controller Registers: \[2\]](#)

**Table 10-236. DISPC\_VID2\_ATTRIBUTES**

<b>Address Offset</b>	0x0000 015C																														
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .								<b>Instance</b>	DISPC																					
<b>Description</b>	The register configures the attributes of the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANNELOUT2	BURSTTYPE	PREMULTIPLYALPHA	ZORDER	ZORDERENABLE	SELFREFRESH	ARBITRATION	DOUBLESTRIDE	VERTICALTAPS	DMAOPTIMIZATION	BUFPRELOAD	RESERVED	SELFREFRESHAUTO	CHANNELOUT	BURSTSIZE	ROTATION	FULLRANGE	REPLICATIONENABLE	COLORCONVENABLE	VRESIZECONF	HRESIZECONF	RESIZEENABLE	FORMAT				ENABLE					

Bits	Field Name	Description	Type	Reset
31:30	CHANNELOUT2	It is not used if CHANNELOUT is set to TV. Reserved when CHANNELOUT = 1 (should set to zero) wr: immediate  0x0: primary LCD output selected. 0x1: Secondary LCD output selected. 0x3: Write-back output to the memory selected.	RW	0x0
29	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targeted by the DMA engine. 0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0
28	PREMULTIPLYALPHA	The field configures the DISPC VID2 to process incoming data as pre-multiplied alpha data or non pre-multiplied alpha data. Default setting is non pre-multiplied alpha data. 0x0: Non pre-multiplied alpha data color component 0x1: Pre-multiplied alpha data color component	RW	0
27:26	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is software responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. If bit 25 is set to 0, the ZORDER bit field is ignored and replaced by the value 0.  0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x3: Z-order 3: layer above all the other layers 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3	RW	0x0
25	ZORDERENABLE	Z-order Enable. The bit field ZORDER is only used when the Z-order is enabled. 0x0: Z-order disabled. The Z-order of the layer is 0. 0x1: Z-order enabled. The Z-order is defined by the bit field ZORDER (bits 26 and 27).	RW	0
24	SELFREFRESH	Enables the self refresh of the video window from its own DMA buffer only. 0x0: The video pipeline accesses the interconnect to fetch data from the system memory. 0x1: The video pipeline does not need anymore to fetch data from memory. Only the DMA buffer associated with the video2 is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0
23	ARBITRATION	Determines the priority of the video pipeline. The video pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The video pipeline is one of the normal priority pipeline. 0x1: The video pipeline is one of the high priority pipeline.	RW	0
22	DOUBLESTRIDE	Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride. It is only used in case of YUV4:2:0. 0x0: The CbCr stride value is equal to the Y stride. 0x1: The CbCr stride value is double to the Y stride.	RW	0



Bits	Field Name	Description	Type	Reset
21	VERTICALTAPS	Video Vertical Resize Tap Number  0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. The associated bit fields for the 2 other taps coefficients do not need to be initialized.  0x1: 5 taps are used for the vertical filtering logic.	RW	0
20	DMAOPTIMIZATION	Write 0s for future compatibility. Reads return 0.	R	0
19	BUFPRELOAD	Video Preload Value  0x0: Hardware prefetches pixels up to the preload value defined in the preload register  0x1: Hardware prefetches pixels up to high threshold value	RW	0
18	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
17	SELFREFRESHAUTO	Automatic self-refresh mode  0x0: The transition from SELFREFRESH "disabled" to "enabled" is controlled by SW.  0x1: The transition from SELFREFRESH "disabled" to "enabled" is controlled only by hardware.	RW	0
16	CHANNELOUT	Video Channel Out configuration: LCD, WB or TV. wr: immediate  0x0: LCD output or WB to the memory selected. bit fields 31 and 30 defines the output associated (primary, secondary or write-back).  0x1: TV output selected	RW	0
15:14	BURSTSIZE	Video DMA Burst Size  0x0: 2x128bit bursts  0x1: 4x128bit bursts  0x3: Reserved  0x2: 8x128bit bursts	RW	0x2
13:12	ROTATION	Video Rotation Flag  0x0: No rotation  0x1: Rotation by 90 degrees  0x3: Rotation by 270 degrees  0x2: Rotation by 180 degrees	RW	0x0
11	FULLRANGE	Color Space Conversion full range setting.  0x0: Limited range selected: 16 subtracted from Y before color space conversion  0x1: Full range selected: Y is not modified before the color space conversion	RW	0
10	REPLICATIONENABLE	Replication Enable  0x0: Disable Video replication logic  0x1: Enable Video replication logic	RW	1
9	COLORCONVENABLE	Enable the color space conversion. The hardware does not enable/disable the conversion based on the pixel format. The bit field shall be reset when the format is not YUV.  0x0: Disable Color Space Conversion YUV to RGB  0x1: Enable Color Space Conversion YUV to RGB	RW	0
8	VRESIZECONF	Write 0s for future compatibility. Reads return 0.	R	0
7	HRESIZECONF	Write 0s for future compatibility. Reads return 0.	R	0

Bits	Field Name	Description	Type	Reset
6:5	RESIZEENABLE	Video Resize Enable 0x0: Disable both horizontal and vertical resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0
4:1	FORMAT	Video Format. It defines the pixel format when fetching the video 2 picture into memory. 0x6: RGB16-565 0x1: RGB12x-4444 0xA: YUV2 4:2:2 co-sited 0x7: ARGB16-1555 0xD: RGBA32-8888 0x0: NV12 4:2:0 2 buffers (Y + UV) 0x2: RGBA12-4444 0x8: xRGB24-8888 (32-bit container) 0x9: RGB24-888 (24-bit container) 0xB: UYVY 4:2:2 co-sited 0x5: ARGB16-4444 0xF: xRGB15-1555 0xC: ARGB32-8888 0x4: xRGB12-4444 0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)	RW	0x0
0	ENABLE	VidEnable 0x0: Video disabled (video pipeline inactive and window not present) 0x1: Video enabled (video pipeline active and window present on the screen)	RW	0

**Table 10-237. Register Call Summary for Register DISPC\_VID2\_ATTRIBUTES**

Display Controller

- [LCD Outputs: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Logical Register Mapping: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-238. DISPC\_VID2\_BUF\_THRESHOLD**

<b>Address Offset</b>	0x0000 0160		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the DMA buffer associated with the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x07FF
15:0	BUFLOWTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x07F8

**Table 10-239. Register Call Summary for Register DISPC\_VID2\_BUF\_THRESHOLD**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-240. DISPC\_VID2\_BUF\_SIZE\_STATUS**

<b>Address Offset</b>	0x0000 0164		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register defines the DMA buffer size for the video pipeline 2.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	DMA buffer size in number of 128 bits	R	0x0800

**Table 10-241. Register Call Summary for Register DISPC\_VID2\_BUF\_SIZE\_STATUS**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-242. DISPC\_VID2\_ROW\_INC**

<b>Address Offset</b>	0x0000 0168
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .
<b>Instance</b>	DISPC
<b>Description</b>	The register configures the number of bytes to increment at the end of the row for the buffer associated with the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded signed value (from $2^{31}1$ to $2^{31}$ ) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value $1 + n * \text{bpp}$ means increment of n pixels. The value $1 (n + 1) * \text{bpp}$ means decrement of n pixels.	RW	0x0000 0001

**Table 10-243. Register Call Summary for Register DISPC\_VID2\_ROW\_INC**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-244. DISPC\_VID2\_PIXEL\_INC**

<b>Address Offset</b>	0x0000 016C
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .
<b>Instance</b>	DISPC
<b>Description</b>	The register configures the number of bytes to increment between two pixels for the buffer associated with the video window 2. The register is used only when the TILER is not present in the system in order to perform low performance rotation. When the TILER IP is present it is highly recommended to use it for performing the rotation. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PIXELINC															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between two pixels in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value $1+n*\text{bpp}$ means increment of n pixels. For YUV4:2:0, Max supported value is 128.	RW	0x01

**Table 10-245. Register Call Summary for Register DISPC\_VID2\_PIXEL\_INC**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-246. DISPC\_VID2\_FIR**

<b>Address Offset</b>	0x0000 0170		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 2. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVINC								RESERVED								FIRHINC							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

**Table 10-247. Register Call Summary for Register DISPC\_VID2\_FIR**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-248. DISPC\_VID2\_PICTURE\_SIZE**

<b>Address Offset</b>	0x0000 0174		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the size of the video picture associated with the video layer 2 before up/down-scaling. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ORGSIZEY								RESERVED								ORGSIZEX							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	ORGSIZEY	Number of lines of the video picture Encoded value (from 1 to 2048) to specify the number of lines of the video picture in memory (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded $2^{11}$ .	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	ORGSIZEX	Number of pixels of the video picture Encoded value (from 1 to 2048) to specify the number of pixels of the video picture in memory (program to value minus 1). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit. (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded $2^{11}$ .	RW	0x000

**Table 10-249. Register Call Summary for Register DISPC\_VID2\_PICTURE\_SIZE**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-250. DISPC\_VID2\_ACCU\_j**

<b>Address Offset</b>	0x0000 0178 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 2 (DISPC_VID2_ACCU_0 & DISPC_VID2_ACCU_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2</a> .GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU								RESERVED								HORIZONTALACCU							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from -1024 to 1023).	RW	0x000

**Table 10-251. Register Call Summary for Register DISPC\_VID2\_ACCU\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\] \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-252. DISPC\_VID2\_FIR\_COEF\_H\_i**

<b>Address Offset</b>	0x0000 0180 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-253. Register Call Summary for Register DISPC\_VID2\_FIR\_COEF\_H\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-254. DISPC\_VID2\_FIR\_COEF\_HV\_i**

<b>Address Offset</b>	0x0000 0184 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-255. Register Call Summary for Register DISPC\_VID2\_FIR\_COEF\_HV\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-256. DISPC\_VID2\_CONV\_COEF0**

<b>Address Offset</b>	0x0000 01C0			
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC		
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory			
<b>Type</b>	RW			
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
	RESERVED	RCR RESERVED RY		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	RCR	RCr Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RY	RY Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-257. Register Call Summary for Register DISPC\_VID2\_CONV\_COEF0**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-258. DISPC\_VID2\_CONV\_COEF1**

<b>Address Offset</b>	0x0000 01C4			
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC		
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory			
<b>Type</b>	RW			
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
	RESERVED	GY RESERVED RCB		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	GY	GY Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RCB	RCb Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000



**Table 10-259. Register Call Summary for Register DISPC\_VID2\_CONV\_COEF1**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Logical Register Mapping: \[1\]](#)
  - [Display Controller Registers: \[2\]](#)

**Table 10-260. DISPC\_VID2\_CONV\_COEF2**

<b>Address Offset</b>	0x0000 01C8	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								GCB								RESERVED								GCR							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	GCB	GCB Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	GCR	GCR Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-261. Register Call Summary for Register DISPC\_VID2\_CONV\_COEF2**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Logical Register Mapping: \[1\]](#)
  - [Display Controller Registers: \[2\]](#)

**Table 10-262. DISPC\_VID2\_CONV\_COEF3**

<b>Address Offset</b>	0x0000 01CC	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BCR								RESERVED								BY							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	BCR	BCR coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	BY	BY coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-263. Register Call Summary for Register DISPC\_VID2\_CONV\_COEF3**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-264. DISPC\_VID2\_CONV\_COEF4**

<b>Address Offset</b>	0x0000 01D0																																																													
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC																																																												
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:5%; text-align:center;">31</td><td style="width:5%; text-align:center;">30</td><td style="width:5%; text-align:center;">29</td><td style="width:5%; text-align:center;">28</td><td style="width:5%; text-align:center;">27</td><td style="width:5%; text-align:center;">26</td><td style="width:5%; text-align:center;">25</td><td style="width:5%; text-align:center;">24</td><td style="width:5%; text-align:center;">23</td><td style="width:5%; text-align:center;">22</td><td style="width:5%; text-align:center;">21</td><td style="width:5%; text-align:center;">20</td><td style="width:5%; text-align:center;">19</td><td style="width:5%; text-align:center;">18</td><td style="width:5%; text-align:center;">17</td><td style="width:5%; text-align:center;">16</td><td style="width:5%; text-align:center;">15</td><td style="width:5%; text-align:center;">14</td><td style="width:5%; text-align:center;">13</td><td style="width:5%; text-align:center;">12</td><td style="width:5%; text-align:center;">11</td><td style="width:5%; text-align:center;">10</td><td style="width:5%; text-align:center;">9</td><td style="width:5%; text-align:center;">8</td><td style="width:5%; text-align:center;">7</td><td style="width:5%; text-align:center;">6</td><td style="width:5%; text-align:center;">5</td><td style="width:5%; text-align:center;">4</td><td style="width:5%; text-align:center;">3</td><td style="width:5%; text-align:center;">2</td><td style="width:5%; text-align:center;">1</td><td style="width:5%; text-align:center;">0</td> </tr> <tr> <td colspan="16" style="text-align:center;">RESERVED</td> <td colspan="12" style="text-align:center;">BCB</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																BCB											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
RESERVED																BCB																																														
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																										
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000																																																										
10:0	BCB	BCb Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000																																																										

**Table 10-265. Register Call Summary for Register DISPC\_VID2\_CONV\_COEF4**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-266. DISPC\_DATA1\_CYCLE1**

<b>Address Offset</b>	0x0000 01D4																																																					
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC																																																				
<b>Description</b>	The control register configures the output data format for 1st cycle. Shadow register, updated on VFP start period of primary LCD																																																					
<b>Type</b>	RW																																																					
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:5%; text-align:center;">31</td><td style="width:5%; text-align:center;">30</td><td style="width:5%; text-align:center;">29</td><td style="width:5%; text-align:center;">28</td><td style="width:5%; text-align:center;">27</td><td style="width:5%; text-align:center;">26</td><td style="width:5%; text-align:center;">25</td><td style="width:5%; text-align:center;">24</td><td style="width:5%; text-align:center;">23</td><td style="width:5%; text-align:center;">22</td><td style="width:5%; text-align:center;">21</td><td style="width:5%; text-align:center;">20</td><td style="width:5%; text-align:center;">19</td><td style="width:5%; text-align:center;">18</td><td style="width:5%; text-align:center;">17</td><td style="width:5%; text-align:center;">16</td><td style="width:5%; text-align:center;">15</td><td style="width:5%; text-align:center;">14</td><td style="width:5%; text-align:center;">13</td><td style="width:5%; text-align:center;">12</td><td style="width:5%; text-align:center;">11</td><td style="width:5%; text-align:center;">10</td><td style="width:5%; text-align:center;">9</td><td style="width:5%; text-align:center;">8</td><td style="width:5%; text-align:center;">7</td><td style="width:5%; text-align:center;">6</td><td style="width:5%; text-align:center;">5</td><td style="width:5%; text-align:center;">4</td><td style="width:5%; text-align:center;">3</td><td style="width:5%; text-align:center;">2</td><td style="width:5%; text-align:center;">1</td><td style="width:5%; text-align:center;">0</td> </tr> <tr> <td colspan="8" style="text-align:center;">RESERVED</td> <td style="text-align:center;">RESERVED</td> <td colspan="4" style="text-align:center;">NBBITSPIXEL2</td> <td style="text-align:center;">RESERVED</td> <td style="text-align:center;">RESERVED</td> <td style="text-align:center;">RESERVED</td> <td colspan="4" style="text-align:center;">NBBITSPIXEL1</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED								RESERVED	NBBITSPIXEL2				RESERVED	RESERVED	RESERVED	NBBITSPIXEL1			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																							
RESERVED								RESERVED	NBBITSPIXEL2				RESERVED	RESERVED	RESERVED	NBBITSPIXEL1																																						
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																		
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0																																																		
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel 2 on the output interface	RW	0x0																																																		
23:21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0																																																		
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00																																																		

Bits	Field Name	Description	Type	Reset
15:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

**Table 10-267. Register Call Summary for Register DISPC\_DATA1\_CYCLE1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-268. DISPC\_DATA1\_CYCLE2**

<b>Address Offset</b>	0x0000 01D8	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The control register configures the output data format for 2nd cycle. Shadow register, updated on VFP start period of primary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				NBBITSPIXEL2				RESERVED				BITALIGNMENTPIXEL1				RESERVED				NBBITSPIXEL1			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

**Table 10-269. Register Call Summary for Register DISPC\_DATA1\_CYCLE2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-270. DISPC\_DATA1\_CYCLE3**

<b>Address Offset</b>	0x0000 01DC		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The control register configures the output data format for 3rd cycle. Shadow register, updated on VFP start period of primary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED				RESERVED								RESERVED											
BITALIGNMENTPIXEL2								NBBITSPIXEL2				BITALIGNMENTPIXEL1								NBBITSPIXEL1											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

**Table 10-271. Register Call Summary for Register DISPC\_DATA1\_CYCLE3**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-272. DISPC\_VID1\_FIR\_COEF\_V\_i**

<b>Address Offset</b>	0x0000 01E0 + (0x4 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC22								FIRVC00															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

**Table 10-273. Register Call Summary for Register DISPC\_VID1\_FIR\_COEF\_V\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-274. DISPC\_VID2\_FIR\_COEF\_V\_i**

<b>Address Offset</b>	0x0000 0200 + (0x4 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC22								FIRVC00															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

**Table 10-275. Register Call Summary for Register DISPC\_VID2\_FIR\_COEF\_V\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-276. DISPC\_CPR1\_COEF\_R**

<b>Address Offset</b>	0x0000 0220	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the color phase rotation matrix coefficients for the Red component. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RR								RESERVED	RG								RESERVED	RB													

Bits	Field Name	Description	Type	Reset
31:22	RR	RR Coefficient Encoded signed value (from -512 to 511).	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	RG	RG Coefficient Encoded signed value (from -512 to 511).	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	RB	RB Coefficient Encoded signed value (from -512 to 511).	RW	0x000

**Table 10-277. Register Call Summary for Register DISPC\_CPR1\_COEF\_R**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-278. DISPC\_CPR1\_COEF\_G**

<b>Address Offset</b>	0x0000 0224	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the color phase rotation matrix coefficients for the Green component. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GR								RESERVED	GG				RESERVED	GB																	

Bits	Field Name	Description	Type	Reset
31:22	GR	GR Coefficient Encoded signed value (from -512 to 511).	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	GG	GG Coefficient Encoded signed value (from -512 to 511).	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	GB	GB Coefficient Encoded signed value (from -512 to 511).	RW	0x000

**Table 10-279. Register Call Summary for Register DISPC\_CPR1\_COEF\_G**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-280. DISPC\_CPR1\_COEF\_B**

<b>Address Offset</b>	0x0000 0228	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the color phase rotation matrix coefficients for the Blue component. It is used for the primary LCD output. Shadow register, updated on VFP start period of primary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR								RESERVED	BG								RESERVED	BB													

Bits	Field Name	Description	Type	Reset
31:22	BR	BR Coefficient Encoded signed value (from -512 to 511).	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	BG	BG Coefficient Encoded signed value (from -512 to 511).	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	BB	BB Coefficient Encoded signed value (from -512 to 511).	RW	0x000

**Table 10-281. Register Call Summary for Register DISPC\_CPR1\_COEF\_B**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-282. DISPC\_GFX\_PRELOAD**

<b>Address Offset</b>	0x0000 022C
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .
<b>Description</b>	The register configures the graphics DMA buffer Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PRELOAD																			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	PRELOAD	DMA buffer preload value Number of 128-bit words defining the preload value.	RW	0x100

**Table 10-283. Register Call Summary for Register DISPC\_GFX\_PRELOAD**

Display Controller

- [DMA Engine: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Low-level Programming Models: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-284. DISPC\_VID1\_PRELOAD**

<b>Address Offset</b>	0x0000 0230	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the DMA buffer of the video 1 pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PRELOAD															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	PRELOAD	DMA buffer preload value Number of 128-bit words defining the preload value.	RW	0x100

**Table 10-285. Register Call Summary for Register DISPC\_VID1\_PRELOAD**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-286. DISPC\_VID2\_PRELOAD**

<b>Address Offset</b>	0x0000 0234	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the DMA buffer of the video 2 pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PRELOAD															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	PRELOAD	DMA buffer preload value Number of 128-bit words defining the preload value.	RW	0x100

**Table 10-287. Register Call Summary for Register DISPC\_VID2\_PRELOAD**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)



**Table 10-288. DISPC\_CONTROL2**

<b>Address Offset</b>	0x0000 0238	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The control register configures the Display Controller module for the secondary LCD output.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
SPATIALTEMPORALDITHERINGFRAMES		RESERVED		TDMUNUSEDBITS		TDMCYCLEFORMAT		TDMPARALLELMODE		TDMENABLE		RESERVED				TVOVERLAYOPTIMIZATION		OVERLAYOPTIMIZATION		STALLMODE		RESERVED		TFTDATALINES		STDITHERENABLE		GOWB		GOLCD		M8B		STNIFT		MONOCOLOR		RESERVED		LCDENABLE	

Bits	Field Name	Description	Type	Reset
31:30	SPATIALTEMPORAL DITHERINGFRAMES	Spatial/Temporal dithering number of frames for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: Spatial only 0x1: Spatial and temporal over 2 frames 0x2: Spatial and temporal over 4 frames 0x3: Reserved	RW	0x0
29:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
26:25	TDMUNUSED BITS	State of unused bits (TDM mode only) for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: low level (0) 0x1: high level (1) 0x2: unchanged from previous state 0x3: reserved	RW	0x0
24:23	TDMCYCLE FORMAT	Cycle format (TDM mode only) for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: 1 cycle for 1 pixel 0x1: 2 cycles for 1 pixel 0x2: 3 cycles for 1 pixel 0x3: 3 cycles for 2 pixels	RW	0x0
22:21	TDMPARALLELMODE	Output Interface width (TDM mode only) for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: 8-bit parallel output interface selected 0x1: 9-bit parallel output interface selected 0x2: 12-bit parallel output interface selected 0x3: 16-bit parallel output interface selected	RW	0x0
20	TDMENABLE	Enable the multiple cycle format (TDM mode only used for Active Matrix mode with the RFBI enable bit off) for the secondary LCD output wr: VFP start period of secondary LCD output 0x0: TDM disabled 0x1: TDM enabled	RW	0

## Display Controller

www.ti.com

Bits	Field Name	Description	Type	Reset
19:14	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
13	TVOVERLAY OPTIMIZATION	<p>Overlay Optimization for the TV output wr: VFP or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory</p> <p>0x0: All the data for all the enabled pipelines are fetched from memory regardless of the overlay/alpha blending configuration.</p> <p>0x1: The data not used by the overlay manager because of overlap between layers with no alpha blending between them shall not be fetched from memory in order to optimize the bandwidth.</p>	RW	0
12	OVERLAY OPTIMIZATION	<p>Overlay Optimization for the secondary LCD output wr: VFP or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory</p> <p>0x0: All the data for all the enabled pipelines are fetched from memory regardless of the overlay/alpha blending configuration.</p> <p>0x1: The data not used by the overlay manager because of overlap between layers with no alpha blending between them shall not be fetched from memory in order to optimize the bandwidth.</p>	RW	0
11	STALLMODE	<p>STALL Mode for the secondary LCD output wr: VFP start period of secondary LCD output</p> <p>0x0: Normal mode selected</p> <p>0x1: STALL mode selected. The Display Controller sends the data without considering the VSYNC/HSYNC. The LCD output is disabled at the end of the transfer of the frame. The S/W has to re-enable the LCD output in order to generate a new frame.</p>	RW	0
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:8	TFTDATALINES	<p>Number of lines of the secondary LCD interface wr: VFP start period of secondary LCD output</p> <p>0x0: 12-bit output aligned on the LSB of the pixel data interface</p> <p>0x1: 16-bit output aligned on the LSB of the pixel data interface</p> <p>0x2: 18-bit output aligned on the LSB of the pixel data interface</p> <p>0x3: 24-bit output aligned on the LSB of the pixel data interface</p>	RW	0x0
7	STDITHER ENABLE	<p>Spatial Temporal dithering enable for the secondary LCD output wr: VFP start period of secondary LCD output</p> <p>0x0: Spatial/Temporal dithering logic disabled</p> <p>0x1: Spatial/Temporal dithering logic enabled</p>	RW	0
6	GOWB	<p>GO Command for the write-back output. It is used to synchronized the pipelines (graphics and/or video ones) associated with the write-back output to the memory. wr:immediate</p> <p>0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) connected to the write-back pipeline using the user values. The hardware resets the bit when the update is completed.</p> <p>0x1: The user has finished to program the shadow registers of the pipeline(s) associated with the write-back pipeline and the hardware can update the internal registers immediatly</p>	RW	0
5	GOLCD	<p>GO Command for the secondary LCD output. It is used to synchronized the pipelines (graphics and/or video ones) associated with the secondary LCD output. wr:immediate</p> <p>0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) connected to the LCD output using the user values. The hardware resets the bit when the update is completed.</p> <p>0x1: The user has finished to program the shadow registers of the pipeline(s) associated with the LCD output and the hardware can update the internal registers at the VFP start period</p>	RW	0

Bits	Field Name	Description	Type	Reset
4	M8B	Mono 8-bit mode of the secondary LCD wr: VFP start period of secondary LCD output  0x0: Pixel data [3:0] is used to output four pixel values to the panel at each pixel clock transition. (only in Passive Mono 4-bit mode).  0x1: Pixel data [7:0] is used to output eight pixel values to the panel each pixel clock transition. (only in Passive Mono 8-bit mode).	RW	0
3	STNTFT	LCD Display type of the secondary LCD wr: VFP start period of secondary LCD output  0x0: Passive Matrix display operation enabled. Passive Matrix dither logic is enabled.  0x1: Active or TFT display operation enabled. STN Dither logic and output FIFO bypassed.	RW	0
2	MONOCOLOR	Monochrome/Color selection for the secondary LCD wr: VFP start period of secondary LCD output  0x0: Color operation enabled (Passive Matrix mode only)  0x1: Monochrome operation enabled (Passive Matrix mode only)	RW	0
1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
0	LCDENABLE	Enable the secondary LCD output wr:immediate  0x0: LCD output disabled (at the end of the frame when the bit is reset)  0x1: LCD output enabled	RW	0

**Table 10-289. Register Call Summary for Register DISPC\_CONTROL2**

Display Controller

- [LCD Outputs: \[0\] \[1\]](#)
- [TV Output: \[2\]](#)
- [Shadows Registers: \[3\] \[4\] \[5\] \[6\]](#)
- [Display Controller Low-level Programming Models: \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [Display Controller Logical Register Mapping: \[15\]](#)
- [Display Controller Registers: \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\] \[44\] \[45\] \[46\] \[47\] \[48\] \[49\] \[50\] \[51\] \[52\] \[53\] \[54\] \[55\] \[56\] \[57\] \[58\] \[59\] \[60\] \[61\] \[62\] \[63\] \[64\] \[65\] \[66\] \[67\] \[68\] \[69\] \[70\] \[71\] \[72\] \[73\] \[74\] \[75\] \[76\] \[77\] \[78\] \[79\] \[80\] \[81\] \[82\] \[83\] \[84\] \[85\] \[86\] \[87\] \[88\] \[89\] \[90\] \[91\] \[92\] \[93\] \[94\] \[95\] \[96\] \[97\] \[98\] \[99\] \[100\] \[101\] \[102\] \[103\] \[104\] \[105\] \[106\] \[107\] \[108\] \[109\] \[110\] \[111\] \[112\] \[113\] \[114\] \[115\] \[116\] \[117\] \[118\] \[119\] \[120\] \[121\] \[122\] \[123\] \[124\] \[125\] \[126\] \[127\] \[128\] \[129\] \[130\] \[131\] \[132\] \[133\] \[134\] \[135\]](#)

**Table 10-290. DISPC\_VID3\_ACCU\_j**

<b>Address Offset</b>	0x0000 0300 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 3 (DISPC_VID3_ACCU_0 & DISPC_VID3_ACCU_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU								RESERVED								HORIZONTALACCU							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from -1024 to 1023).	RW	0x000

**Table 10-291. Register Call Summary for Register DISPC\_VID3\_ACCU\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-292. DISPC\_VID3\_BA\_j**

<b>Address Offset</b>	0x0000 0308 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the base address of the video buffer for the video window 3 (DISPC_VID3_BA_0 & DISPC_VID3_BA_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID3_BA_0 is used). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address Base address of the video buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV4:2:2, 2-pixel alignment is required, and YUV4:2:0, byte alignment is supported). In case of YUV4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

**Table 10-293. Register Call Summary for Register DISPC\_VID3\_BA\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-294. DISPC\_VID3\_FIR\_COEF\_H\_i**

<b>Address Offset</b>	0x0000 0310 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-295. Register Call Summary for Register DISPC\_VID3\_FIR\_COEF\_H\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-296. DISPC\_VID3\_FIR\_COEF\_HV\_i**

Address Offset	0x0000 0314 + (0x8 * i)	Index	i = 0 to 7
Physical Address	See <a href="#">Table 10-133</a> .	Instance	DISPC
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-297. Register Call Summary for Register DISPC\_VID3\_FIR\_COEF\_HV\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-298. DISPC\_VID3\_FIR\_COEF\_V\_i**

<b>Address Offset</b>	0x0000 0350 + (0x4 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FIRVC22						FIRVC00									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

**Table 10-299. Register Call Summary for Register DISPC\_VID3\_FIR\_COEF\_V\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-300. DISPC\_VID3\_ATTRIBUTES**

<b>Address Offset</b>	0x0000 0370	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the attributes of the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANNELOUT2	BURSTTYPE	PREMULTIPLYALPHA	ZORDER	ZORDERENABLE	SELFREFRESH	ARBITRATION	DOUBLESTRIDE	VERTICALTAPS	DMAOPTIMIZATION	BUFPRELOAD	RESERVED	SELFREFRESHAUTO	CHANNELOUT	BURSTSIZE	ROTATION	FULLRANGE	REPLICATIONENABLE	COLORCONVENABLE	VRESIZECONF	HRESIZECONF	RESIZEENABLE	FORMAT				ENABLE					

Bits	Field Name	Description	Type	Reset
31:30	CHANNELOUT2	It is not used if CHANNELOUT is set to TV. Reserved when CHANNELOUT = 1 (should be set to zero) wr: immediate  0x0: primary LCD output selected. 0x1: Secondary LCD output selected. 0x3: Write-back output to the memory selected.	RW	0x0

Bits	Field Name	Description	Type	Reset
29	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targeted by the DMA engine. 0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0
28	PREMULTIPLYALPHA	The field configures the DISPC VID3 to process incoming data as pre-multiplied alpha data or non pre-multiplied alpha data. Default setting is non pre-multiplied alpha data. 0x0: Non pre-multiplied alpha data color component 0x1: Pre-multiplied alpha data color component	RW	0
27:26	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is software responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. If bit 25 is set to 0, the ZORDER bit field is ignored and replaced by the value 0. 0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x3: Z-order 3: layer above all the other layers 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3	RW	0x0
25	ZORDERENABLE	Z-order Enable. The bit field ZORDER is only used when the Z-order is enabled. 0x0: Z-order disabled. The Z-order of the layer is 0. 0x1: Z-order enabled. The Z-order is defined by the bit field ZORDER (bits 26 and 27).	RW	0
24	SELFREFRESH	Enables the self refresh of the video window from its own DMA buffer only. 0x0: The video pipeline accesses the interconnect to fetch data from the system memory. 0x1: The video pipeline does not need anymore to fetch data from memory. Only the DMA buffer associated with the video3 is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0
23	ARBITRATION	Determines the priority of the video pipeline. The video pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The video pipeline is one of the normal priority pipeline. 0x1: The video pipeline is one of the high priority pipeline.	RW	0
22	DOUBLESTRIDE	Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride. It is only used in case of YUV4:2:0. 0x0: The CbCr stride value is equal to the Y stride. 0x1: The CbCr stride value is double to the Y stride.	RW	0
21	VERTICALTAPS	Video Vertical Resize Tap Number 0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. The associated bit fields for the 2 other taps coefficients do not need to be initialized. 0x1: 5 taps are used for the vertical filtering logic.	RW	0

## Display Controller

www.ti.com

Bits	Field Name	Description	Type	Reset
20	DMAOPTIMIZATION	Write 0s for future compatibility. Reads return 0.	R	0
19	BUFPRELOAD	Video Preload Value  0x0: Hardware prefetches pixels up to the preload value defined in the preload register 0x1: Hardware prefetches pixels up to high threshold value	RW	0
18	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
17	SELFREFRESHAUTO	Automatic self-refresh mode  0x0: The transition fromSELFREFRESH "disabled" to "enabled" is controlled by SW. 0x1: The transition from SELFREFRESH "disabled" to "enabled" is controlled only by hardware.	RW	0
16	CHANNELOUT	Video Channel Out configuration: LCD, WB or TV. wr: immediate  0x0: LCD output or WB to the memory selected. bit fields 31 and 30 defines the output associated (primary, secondary or write-back). 0x1: TV output selected	RW	0
15:14	BURSTSIZE	Video DMA Burst Size  0x0: 2x128bit bursts 0x1: 4x128bit bursts 0x3: Reserved 0x2: 8x128bit bursts	RW	0x2
13:12	ROTATION	Video Rotation Flag  0x0: No rotation 0x1: Rotation by 90 degrees 0x3: Rotation by 270 degrees 0x2: Rotation by 180 degrees	RW	0x0
11	FULLRANGE	Color Space Conversion full range setting.  0x0: Limited range selected: 16 subtracted from Y before color space conversion 0x1: Full range selected: Y is not modified before the color space conversion	RW	0
10	REPLICATIONENABLE	Replication Enable  0x0: Disable Video replication logic 0x1: Enable Video replication logic	RW	1
9	COLORCONVENABLE	Enable the color space conversion. The hardware does not enable/disable the conversion based on the pixel format. The bit field shall be reset when the format is not YUV. 0x0: Disable Color Space Conversion YUV to RGB 0x1: Enable Color Space Conversion YUV to RGB	RW	0
8	VRESIZECONF	Write 0s for future compatibility. Reads return 0.	R	0
7	HRESIZECONF	Write 0s for future compatibility. Reads return 0.	R	0
6:5	RESIZEENABLE	Video Resize Enable  0x0: Disable both horizontal and vertical resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0



Bits	Field Name	Description	Type	Reset
4:1	FORMAT	Video Format. It defines the pixel format when fetching the video 3 picture into memory.  0x6: RGB16-565 0x1: RGB12x-4444 0xA: YUV2 4:2:2 co-sited 0x7: ARGB16-1555 0xD: RGBA32-8888 0x0: NV12 4:2:0 2 buffers (Y + UV) 0x2: RGBA12-4444 0x8: RGB24-8888 (32-bit container) 0x9: RGB24-888 (24-bit container) 0xB: UYVY 4:2:2 co-sited 0x5: ARGB16-4444 0xF: xRGB15-1555 0xC: ARGB32-8888 0x4: xRGB12-4444 0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)	RW	0x0
0	ENABLE	Video Enable  0x0: Video disabled (video pipeline inactive and window not present) 0x1: Video enabled (video pipeline active and window present on the screen)	RW	0

**Table 10-301. Register Call Summary for Register DISPC\_VID3\_ATTRIBUTES**

Display Controller

- [LCD Outputs: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Logical Register Mapping: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-302. DISPC\_VID3\_CONV\_COEF0**

<b>Address Offset</b>	0x0000 0374
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RCR								RESERVED								RY							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	RCR	RCr Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	RY	RY Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-303. Register Call Summary for Register DISPC\_VID3\_CONV\_COEF0**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-304. DISPC\_VID3\_CONV\_COEF1**

<b>Address Offset</b>	0x0000 0378	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	RESERVED	GY RESERVED RCB
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.
26:16	GY	GY Coefficient Encoded signed value (from -1024 to 1023).
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.
10:0	RCB	RCb Coefficient Encoded signed value (from -1024 to 1023).
<b>Type</b>		<b>Reset</b>
		0x00
		0x000
		0x00
		0x000

**Table 10-305. Register Call Summary for Register DISPC\_VID3\_CONV\_COEF1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-306. DISPC\_VID3\_CONV\_COEF2**

<b>Address Offset</b>	0x0000 037C	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	RESERVED	GCB RESERVED GCR
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.
26:16	GCB	GCb Coefficient Encoded signed value (from -1024 to 1023).
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.
10:0	GCR	GCr Coefficient Encoded signed value (from -1024 to 1023).
<b>Type</b>		<b>Reset</b>
		0x00
		0x000
		0x00
		0x000

**Table 10-307. Register Call Summary for Register DISPC\_VID3\_CONV\_COEF2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-308. DISPC\_VID3\_CONV\_COEF3**

<b>Address Offset</b>	0x0000 0380		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BCR								RESERVED				BY											

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	BCR	BCr coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	BY	BY coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-309. Register Call Summary for Register DISPC\_VID3\_CONV\_COEF3**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-310. DISPC\_VID3\_CONV\_COEF4**

<b>Address Offset</b>	0x0000 0384		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCB															

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
10:0	BCB	BCb Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-311. Register Call Summary for Register DISPC\_VID3\_CONV\_COEF4**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-312. DISPC\_VID3\_BUF\_SIZE\_STATUS**

<b>Address Offset</b>	0x0000 0388	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register defines the DMA buffer size for the video pipeline 3.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	DMA buffer Size in number of 128-bits.	R	0x0800

**Table 10-313. Register Call Summary for Register DISPC\_VID3\_BUF\_SIZE\_STATUS**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-314. DISPC\_VID3\_BUF\_THRESHOLD**

<b>Address Offset</b>	0x0000 038C	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the DMA buffer associated with the video pipeline 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x07FF
15:0	BUFLOWTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x07F8

**Table 10-315. Register Call Summary for Register DISPC\_VID3\_BUF\_THRESHOLD**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-316. DISPC\_VID3\_FIR**

<b>Address Offset</b>	0x0000 0390		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 3. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVINC								RESERVED								FIRHINC							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

**Table 10-317. Register Call Summary for Register DISPC\_VID3\_FIR**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-318. DISPC\_VID3\_PICTURE\_SIZE**

<b>Address Offset</b>	0x0000 0394		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the size of the video picture associated with the video layer 3 before up/down-scaling. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ORGSIZEY								RESERVED								ORGSIZEX							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	ORGSIZEY	Number of lines of the video picture Encoded value (from 1 to 2048) to specify the number of lines of the video picture in memory (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded $2^{11}$ .	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00

Bits	Field Name	Description	Type	Reset
10:0	ORGSIZEX	Number of pixels of the video picture Encoded value (from 1 to 2048) to specify the number of pixels of the video picture in memory (program to value minus 1). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit. (program to value minus 1). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded $2^{11}$ .	RW	0x000

**Table 10-319. Register Call Summary for Register DISPC\_VID3\_PICTURE\_SIZE**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-320. DISPC\_VID3\_PIXEL\_INC**

<b>Address Offset</b>	0x0000 0398		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the number of bytes to increment between two pixels for the buffer associated with the video window 3. The register is used only when the TILER is not present in the system in order to perform low performance rotation. When the TILER IP is present it is highly recommended to use it for performing the rotation. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PIXELINC															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between two pixels in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value $1+n*bpp$ means increment of n pixels. For YUV4:2:0, Max supported value is 128.	RW	0x01

**Table 10-321. Register Call Summary for Register DISPC\_VID3\_PIXEL\_INC**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-322. DISPC\_VID3\_POSITION**

<b>Address Offset</b>	0x0000 039C		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the position of the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								POSY								RESERVED								POSX							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	POSY	Y position of the video window 2 Encoded value (from 0 to 2047) to specify the Y position of the video window 2 .The line at the top has the Y-position 0.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	POSX	X position of the video window 2 Encoded value (from 0 to 2047) to specify the X position of the video window 2. The first pixel on the left of the display screen has the X-position 0.	RW	0x000

**Table 10-323. Register Call Summary for Register DISPC\_VID3\_POSITION**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-324. DISPC\_VID3\_PRELOAD**

<b>Address Offset</b>	0x0000 03A0		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the DMA buffer of the video 3 pipeline. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PRELOAD															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000
11:0	PRELOAD	DMA buffer preload value Number of 128-bit words defining the preload value.	RW	0x100

**Table 10-325. Register Call Summary for Register DISPC\_VID3\_PRELOAD**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-326. DISPC\_VID3\_ROW\_INC**

<b>Address Offset</b>	0x0000 03A4	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the number of bytes to increment at the end of the row for the buffer associated with the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded signed value (from $2^{31}1$ to $2^{31}$ ) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value $1 + n * \text{bpp}$ means increment of n pixels. The value $1 (n + 1) * \text{bpp}$ means decrement of n pixels.	RW	0x0000 0001

**Table 10-327. Register Call Summary for Register DISPC\_VID3\_ROW\_INC**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-328. DISPC\_VID3\_SIZE**

<b>Address Offset</b>	0x0000 03A8	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the size of the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SIZEY								RESERVED								SIZEX							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	SIZY	Number of lines of the video 3 Encoded value (from 1 to 2048) to specify the number of lines of the video window 3. Program to value minus 1.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	SIZEX	Number of pixels of the video window 3 Encoded value (from 1 to 2048) to specify the number of pixels of the video window 3. Program to value minus 1.	RW	0x000

**Table 10-329. Register Call Summary for Register DISPC\_VID3\_SIZE**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)



**Table 10-330. DISPC\_DEFAULT\_COLOR2**

<b>Address Offset</b>	0x0000 03AC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .
<b>Description</b>	The control register allows to configure the default solid background color for the secondary LCD Shadow register, updated on VFP start period of secondary LCD
<b>Type</b>	RW

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED		DEFAULTCOLOR	

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	DEFAULTCOLOR	24-bit RGB color value to specify the default solid color to display when there is no data from the overlays.	RW	0x000000

**Table 10-331. Register Call Summary for Register DISPC\_DEFAULT\_COLOR2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-332. DISPC\_TRANS\_COLOR2**

<b>Address Offset</b>	0x0000 03B0
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .
<b>Description</b>	The register sets the transparency color value for the video/graphics overlays for the secondary LCD output. Shadow register, updated on VFP start period of the secondary LCD
<b>Type</b>	RW

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED		TRANSCOLORKEY	

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:0	TRANSCOLORKEY	Transparency Color Key Value in RGB format [0] BITMAP 1 (CLUT), [23,1] set to 0s [1:0] BITMAP 2 (CLUT), [23,2] set to 0s [3:0] BITMAP 4 (CLUT), [23,4] set to 0s [7:0] BITMAP 8 (CLUT), [23,8] set to 0s [11:0] RGB 12, [23,12] set to 0s [15:0] RGB 16, [23,16] set to 0s [23:0] RGB 24	RW	0x000000

**Table 10-333. Register Call Summary for Register DISPC\_TRANS\_COLOR2**

Display Controller

- [LCD Outputs: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-334. DISPC\_CPR2\_COEF\_B**

<b>Address Offset</b>	0x0000 03B4		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the color phase rotation matrix coefficients for the Blue component. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR								RESERVED	BG								RESERVED	BB													

Bits	Field Name	Description	Type	Reset
31:22	BR	BR Coefficient Encoded signed value (from -512 to 511).	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	BG	BG Coefficient Encoded signed value (from -512 to 511).	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	BB	BB Coefficient Encoded signed value (from -512 to 511).	RW	0x000

**Table 10-335. Register Call Summary for Register DISPC\_CPR2\_COEF\_B**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-336. DISPC\_CPR2\_COEF\_G**

<b>Address Offset</b>	0x0000 03B8		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the color phase rotation matrix coefficients for the Green component. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GR								RESERVED	GG								RESERVED	GB													

Bits	Field Name	Description	Type	Reset
31:22	GR	GR Coefficient Encoded signed value (from -512 to 511).	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	GG	GG Coefficient Encoded signed value (from -512 to 511).	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	GB	GB Coefficient Encoded signed value (from -512 to 511).	RW	0x000

**Table 10-337. Register Call Summary for Register DISPC\_CPR2\_COEF\_G**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-338. DISPC\_CPR2\_COEF\_R**

<b>Address Offset</b>	0x0000 03BC	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the color phase rotation matrix coefficients for the Red component. Shadow register, updated on VFP start period of secondary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RR								RESERVED	RG								RESERVED	RB													

Bits	Field Name	Description	Type	Reset
31:22	RR	RR Coefficient Encoded signed value (from -512 to 511).	RW	0x000
21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
20:11	RG	RG Coefficient Encoded signed value (from -512 to 511).	RW	0x000
10	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
9:0	RB	RB Coefficient Encoded signed value (from -512 to 511).	RW	0x000

**Table 10-339. Register Call Summary for Register DISPC\_CPR2\_COEF\_R**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-340. DISPC\_DATA2\_CYCLE1**

<b>Address Offset</b>	0x0000 03C0	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The control register configures the output data format for 1st cycle. Shadow register, updated on VFP start period of secondary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				BITALIGNMENTPIXEL2				RESERVED	NBBITSPIXEL2				RESERVED	BITALIGNMENTPIXEL1				RESERVED	NBBITSPIXEL1												

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

**Table 10-341. Register Call Summary for Register DISPC\_DATA2\_CYCLE1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-342. DISPC\_DATA2\_CYCLE2**

<b>Address Offset</b>	0x0000 03C4	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The control register configures the output data format for 2nd cycle. Shadow register, updated on VFP start period of secondary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				BITALIGNMENTPIXEL2				RESERVED				NBBITSPIXEL2				RESERVED				BITALIGNMENTPIXEL1				RESERVED				NBBITSPIXEL1			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

**Table 10-343. Register Call Summary for Register DISPC\_DATA2\_CYCLE2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-344. DISPC\_DATA2\_CYCLE3**

<b>Address Offset</b>	0x0000 03C8	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The control register configures the output data format for 3rd cycle. Shadow register, updated on VFP start period of secondary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
RESERVED								BITALIGNMENTPIXEL2								RESERVED								NBBITSPIXEL2								RESERVED								BITALIGNMENTPIXEL1								RESERVED								NBBITSPIXEL1							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0s for future compatibility Reads return 0.	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

**Table 10-345. Register Call Summary for Register DISPC\_DATA2\_CYCLE3**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-346. DISPC\_SIZE\_LCD2**

<b>Address Offset</b>	0x0000 03CC		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the panel size (horizontal and vertical). It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD. A delta value is used to indicate if the odd field has same vertical size as the even field or +/- one line.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LPP								DELTA_LPP		RESERVED		PPL											

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	LPP	Lines per panel Encoded value (from 1 to 2048) to specify the number of lines per panel (program to value minus 1).	RW	0x000
15:14	DELTA_LPP	Indicates the delta size value of the odd field compared to the even field 0x0: same size 0x1: odd size = even size +1 0x2: Odd size = even size -1	RW	0x0
13:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
10:0	PPL	Pixels per line Encoded value (from 1 to 2048) to specify the number of pixels contains within each line on the display (program to value minus 1). In STALL mode, any value is valid. In non STALL mode, only values multiple of 8 pixels are valid.	RW	0x000

**Table 10-347. Register Call Summary for Register DISPC\_SIZE\_LCD2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\] \[2\]](#)
- [Display Controller Registers: \[3\] \[4\]](#)

**Table 10-348. DISPC\_TIMING\_H2**

<b>Address Offset</b>	0x0000 0400		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the timing logic for the HSYNC signal. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBP								HFP								HSW															

Bits	Field Name	Description	Type	Reset
31:20	HBP	Horizontal Back Porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display (program to value minus 1).	RW	0x000
19:8	HFP	Horizontal front porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the end of a line transmission before line clock is asserted (program to value minus 1).	RW	0x000

Bits	Field Name	Description	Type	Reset
7:0	HSW	Horizontal synchronization pulse width Encoded value (from 1 to 256) to specify the number of pixel clock periods to pulse the line clock at the end of each line (program to value minus 1).	RW	0x00

**Table 10-349. Register Call Summary for Register DISPC\_TIMING\_H2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-350. DISPC\_TIMING\_V2**

<b>Address Offset</b>	0x0000 0404	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the timing logic for the VSYNC signal. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBP								VFP								VSW															

Bits	Field Name	Description	Type	Reset
31:20	VBP	Vertical back porch Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the beginning of a frame before the first set of pixels is output to the display.	RW	0x000
19:8	VFP	Vertical front porch Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the end of each frame.	RW	0x000
7:0	VSW	Vertical synchronization pulse width In active mode, encoded value (from 1 to 256) to specify the number of line clock periods (program to value minus 1) to pulse the frame clock (VSYNC) pin at the end of each frame after the end of frame wait (VFP) period elapses. Frame clock uses as VSYNC signal in active mode. In passive mode, encoded value (from 1 to 256) to specify the number of extra line clock periods (program to value minus 1) to insert after the vertical front porch (VFP) period has elapsed.	RW	0x00

**Table 10-351. Register Call Summary for Register DISPC\_TIMING\_V2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-352. DISPC\_POL\_FREQ2**

<b>Address Offset</b>	0x0000 0408	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the signal configuration. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												ALIGN	ONOFF	RF	IEO	IPC	IHS	IVS	ACBI				ACB								

## Display Controller

www.ti.com

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
18	ALIGN	Defines the alignment between HSYNC and VSYNC assertion. 0x0: VSYNC and HSYNC are not aligned 0x1: VSYNC and HSYNC assertions are aligned.	RW	0
17	ONOFF	HSYNC/VSYNC Pixel clock Control On/Off 0x0: HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data 0x1: HSYNC and VSYNC are driven according to bit 16	RW	0
16	RF	Program HSYNC/VSYNC Rise or Fall 0x0: HSYNC and VSYNC are driven on falling edge of pixel clock (if bit 17 set to 1) 0x1: HSYNC and VSYNC are driven on rising edge of pixel clock (if bit 17 set to 1)	RW	0
15	IEO	Invert output enable 0x0: Ac-bias is active high (active display mode) 0x1: Ac-bias is active low (active display mode)	RW	0
14	IPC	Invert pixel clock 0x0: Data is driven on the LCD data lines on the rising-edge of the pixel clock 0x1: Data is driven on the LCD data lines on the falling-edge of the pixel clock	RW	0
13	IHS	Invert HSYNC 0x0: Line clock pin is active high and inactive low 0x1: Line clock pin is active low and inactive high	RW	0
12	IVS	Invert VSYNC 0x0: Frame clock pin is active high and inactive low 0x1: Frame clock pin is active low and inactive high	RW	0
11:8	ACBI	AC Bias Pin transitions per interrupt Value (from 0 to 15) used to specify the number of AC Bias pin transitions	RW	0x0
7:0	ACB	AC Bias Pin Frequency Value (from 0 to 255) used to specify the number of line clocks to count before transitioning the AC Bias pin. This pin is used to periodically invert the polarity of the power supply to prevent DC charge build-up within the display.	RW	0x00

Table 10-353. Register Call Summary for Register DISPC\_POL\_FREQ2

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

Table 10-354. DISPC\_DIVISOR2

<b>Address Offset</b>	0x0000 040C																														
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .								<b>Instance</b>	DISPC																					
<b>Description</b>	The register configures the divisors. It is used for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LCD								RESERVED				PCD											



Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:16	LCD	Display Controller Logic Clock Divisor Value (from 1 to 255) to specify the intermediate pixel clock frequency based on the LCD2_CLK. The value 0 is invalid.	RW	0x04
15:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
7:0	PCD	Pixel Clock Divisor Value (from 1 to 255) to specify the frequency of the pixel clock based on the LCD2_CLK divided by DISPC_DIVISOR2.LCD value. The values 0 is invalid.	RW	0x01

**Table 10-355. Register Call Summary for Register DISPC\_DIVISOR2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\] \[3\]](#)

**Table 10-356. DISPC\_WB\_ACCU\_j**

Address Offset	0x0000 0500 + (0x4 * j)	Index	j = 0 to 1
Physical Address	See <a href="#">Table 10-133</a> .	Instance	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the write back pipeline (DISPC_WB_ACCU_0 & DISPC_WB_ACCU_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for ARGB and Y setting. Shadow register, updated when DISPC_CONTROL2.GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU								RESERVED				HORIZONTALACCU											

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from -1024 to 1023).	RW	0x000

**Table 10-357. Register Call Summary for Register DISPC\_WB\_ACCU\_j**

Display Controller

- [Write-back Pipeline: \[0\] \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Low-level Programming Models: \[3\] \[4\]](#)
- [Display Controller Registers: \[5\]](#)

**Table 10-358. DISPC\_WB\_BA\_j**

<b>Address Offset</b>	0x0000 0508 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the base address of the WB buffer (DISPC_WB_BA_0 & DISPC_WB_BA_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_WB_BA_0 is used). Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Write-back base address Base address of the WB buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV4:2:2, 2-pixel alignment is required, and YUV4:2:0, byte alignment is supported)). It case of YUV4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

**Table 10-359. Register Call Summary for Register DISPC\_WB\_BA\_j**

Display Controller

- [DMA Engine: \[0\]](#)
- [Rotation and Mirroring: \[1\] \[2\] \[3\] \[4\]](#)
- [Shadows Registers: \[5\]](#)
- [Display Controller Low-level Programming Models: \[6\]](#)
- [Display Controller Registers: \[7\]](#)

**Table 10-360. DISPC\_WB\_FIR\_COEF\_H\_i**

<b>Address Offset</b>	0x0000 0510 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00

Bits	Field Name	Description	Type	Reset
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-361. Register Call Summary for Register DISPC\_WB\_FIR\_COEF\_H\_i**

Display Controller

- [Write-back Pipeline: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Shadows Registers: \[6\]](#)
- [Display Controller Low-level Programming Models: \[7\]](#)
- [Display Controller Registers: \[8\]](#)

**Table 10-362. DISPC\_WB\_FIR\_COEF\_HV\_i**

<b>Address Offset</b>	0x0000 0514 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-363. Register Call Summary for Register DISPC\_WB\_FIR\_COEF\_HV\_i**

Display Controller

- [Write-back Pipeline: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Shadows Registers: \[7\]](#)
- [Display Controller Low-level Programming Models: \[8\] \[9\]](#)
- [Display Controller Registers: \[10\]](#)

**Table 10-364. DISPC\_WB\_FIR\_COEF\_V\_i**

<b>Address Offset</b>	0x0000 0550 + (0x4 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for ARGB and Y setting. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FIRVC22						FIRVC00									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

**Table 10-365. Register Call Summary for Register DISPC\_WB\_FIR\_COEF\_V\_i**

Display Controller

- [Write-back Pipeline: \[0\] \[1\] \[2\]](#)
- [Shadows Registers: \[3\]](#)
- [Display Controller Low-level Programming Models: \[4\]](#)
- [Display Controller Registers: \[5\]](#)

**Table 10-366. DISPC\_WB\_ATTRIBUTES**

<b>Address Offset</b>	0x0000 0570	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the attributes of the viwrite back pipeline. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
IDLENUMBER		IDLESIZE		CAPTUREMODE		ARBITRATION		DOUBLESTRIDE		VERTICALTAPS		RESERVED		WRITEBACKMODE		CHANNELIN		BURSTSIZE		RESERVED		FULLRANGE		TRUNCATIONENABLE		COLORCONVENABLE		BURSTTYPE		ALPHAENABLE		RESIZEENABLE		FORMAT		ENABLE	

Bits	Field Name	Description	Type	Reset
31:28	IDLENUMBER	<p>Determines the number of idles between requests on the L3 interconnect.</p> <p>It is only used when the write-back pipeline does data transfer from memory to memory.</p> <p>When the output of an overlay is stored in memory through the write-back pipeline in capture mode, the bit field IDLENUMBER is ignored since a timing generator is used to time the transfer.</p> <p>The number of IDLE cycles is IDLENUMBER (from 0 to 15) if IDLESIZE=0.</p> <p>The number of IDLE cycles is IDLENUMBERx8 (from 0 to 120) if IDLESIZE=1 and BURSTSIZE=2.</p> <p>The number of IDLE cycles is IDLENUMBERx4 (from 0 to 60) if IDLESIZE=1 and BURSTSIZE=1.</p> <p>The number of IDLE cycles is IDLENUMBERx2 (from 0 to 30) if IDLESIZE=1 and BURSTSIZE=0.</p>	RW	0x0
27	IDLESIZE	<p>Determines if the IDLENUMBER corresponds to a number of bursts or singles.</p> <p>0x0: The number of idles between requests is defined by IDLENUMBER as number of cycles.</p> <p>0x1: The number of idles between requests is defined by IDLENUMBER multiplied by burst size as number of cycles.</p>	RW	0
26:24	CAPTUREMODE	<p>Defines the frame rate capture.</p> <p>0x6: Only one out of six frames is captured. The first one is captured then the second one is skipped and so on.</p> <p>0x1: Only one frame is captured.</p> <p>0x7: Only one out of seven frames is captured. The first one is captured then the second one is skipped and so on.</p> <p>0x0: All frames are captures until the write-back channel is disabled or there is no more data generated by the overlay or the pipeline attached to the write-back channel.</p> <p>0x2: Only one out of two frames is captured. The first one is captured then the second one is skipped and so on.</p> <p>0x4: Only one out of four frames is captured. The first one is captured then the second one is skipped and so on.</p> <p>0x5: Only one out of five frames is captured. The first one is captured then the second one is skipped and so on.</p> <p>0x3: Only one out of three frames is captured. The first one is captured then the second one is skipped and so on.</p>	RW	0x0
23	ARBITRATION	<p>Determines the priority of the write-back pipeline.</p> <p>The write-back pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them.</p> <p>0x0: The write-back pipeline is one of the normal priority pipeline.</p> <p>0x1: The write-back pipeline is one of the high priority pipeline.</p>	RW	0
22	DOUBLESTRIDE	<p>Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride.</p> <p>It is only used in case of YUV4:2:0.</p> <p>0x0: The CbCr stride value is equal to the Y stride.</p> <p>0x1: The CbCr stride value is double to the Y stride.</p>	RW	0

## Display Controller

www.ti.com

Bits	Field Name	Description	Type	Reset
21	VERTICALTAPS	Video Vertical Resize Tap Number 0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. 0x1: 5 taps are used for the vertical filtering logic.	RW	0
20	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
19	WRITEBACKMODE	When connected to the overlay output of a channel the write back can operate as a simple transfer from memory to memory (composition engine) or as a capture channel. 0x0: Capture mode (default mode) 0x1: Memory-to-memory mode	RW	0x0
18:16	CHANNELIN	Video Channel In configuration wr: immediate 0x6: Video3 pipeline output 0x1: Secondary LCD output 0x0: Primary LCD overlay output 0x2: TV overlay output 0x4: Video1 pipeline output 0x5: Video2 pipeline output 0x3: Graphics pipeline output	RW	0x0
15:14	BURSTSIZE	Write-back DMA Burst Size 0x0: 2x128bit bursts 0x1: 4x128bit bursts 0x3: Reserved 0x2: 8x128bit bursts	RW	0x2
13:12	RESERVED	RESERVED	RW	0x0
11	FULLRANGE	Color Space Conversion full range setting. 0x0: Limited range selected: 16 subtracted from Y before color space conversion 0x1: Full range selected: Y is not modified before the color space conversion	RW	0
10	TRUNCATIONENABLE	It applies only when the input format to the write-back pipeline from the overlay or directly from one of the pipelines is ARGB32. If the format is one of the YUV supported formats, the bit field is ignored. 0x0: Disable truncation logic 0x1: Enable truncation logic from ARGB32 to the pixel format defined in the field FORMAT.	RW	0
9	COLORCONVENABLE	Enable the color space conversion. The hardware does not enable/disable the conversion based on the pixel format. The bit field shall be reset when the format is not YUV. 0x0: Disable Color Space Conversion RGB to YUV 0x1: Enable Color Space Conversion RGB to YUV	RW	0
8	BURSTTYPE	The type of burst can be INCR (incremental) or BLCK (2D block). The 2D block is required when the TILER is targetted by the DMA engine. 0x0: INC burst type is used. 0x1: 2D block burst type is used.	RW	0

Bits	Field Name	Description	Type	Reset
7	ALPHAENABLE	Premultiplied alpha enable Read 0x1: Enabled Read 0x0: Disabled. This bit also disable the logic present in the associated channel out that compute the alpha component sent to the WB pipe. When the WB is configured to copy back one of the output channels (output of overlay), the following configurations are available: 0x1: The WB pipe copies back to memory the premultiplied alpha calculated through the overlay. 0x0: The alpha value is not written back.	RW	0
6:5	RESIZEENABLE	Resize Enable 0x0: Disable the resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0
4:1	FORMAT	Write-back Format. It defines the pixel format when storing the write-back picture into memory. 0x6: RGB16-565 0x1: RGB12x-4444 0xA: YUV2 4:2:2 co-sited 0x7: ARGB16-1555 0xD: RGBA32-8888 0x0: NV12 4:2:0 2 buffers (Y + UV) 0x2: RGBA12-4444 0x8: xRGB24-8888 (32-bit container) 0x9: RGB24-888 (24-bit container) 0xB: UYVY 4:2:2 co-sited 0x5: ARGB16-4444 0xF: xRGB15-1555 0xC: ARGB32-8888 0x4: xRGB12-4444 0xE: RGBx24-8888 (24-bit RGB aligned on MSB of the 32-bit container)	RW	0x0
0	ENABLE	Write-back Enable. wr: immediate 0x0: Write-back disabled 0x1: Write-back enabled	RW	0

**Table 10-367. Register Call Summary for Register DISPC\_WB\_ATTRIBUTES**

Display Controller

- [DMA Engine: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Write-back Pipeline: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [LCD Outputs: \[16\] \[17\] \[18\]](#)
- [Shadows Registers: \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [Display Controller Low-level Programming Models: \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\]](#)
- [Display Controller Registers: \[38\]](#)

**Table 10-368. DISPC\_WB\_CONV\_COEF0**

<b>Address Offset</b>	0x0000 0574		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the write back pipeline (YUV4:4:4 to RGB24) Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								YG								RESERVED								YR							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	YG	YG Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	YR	YR Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-369. Register Call Summary for Register DISPC\_WB\_CONV\_COEF0**

Display Controller

- [Write-back Pipeline: \[0\] \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Low-level Programming Models: \[3\]](#)
- [Display Controller Registers: \[4\]](#)

**Table 10-370. DISPC\_WB\_CONV\_COEF1**

<b>Address Offset</b>	0x0000 0578		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRR								RESERVED								YB							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	CRR	CrR Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	YB	YB Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-371. Register Call Summary for Register DISPC\_WB\_CONV\_COEF1**

Display Controller

- [Write-back Pipeline: \[0\] \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Low-level Programming Models: \[3\]](#)
- [Display Controller Registers: \[4\]](#)



**Table 10-372. DISPC\_WB\_CONV\_COEF2**

<b>Address Offset</b>	0x0000 057C		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRB								RESERVED								CRG							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	CRB	CrB Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	CRG	CrG Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-373. Register Call Summary for Register DISPC\_WB\_CONV\_COEF2**

Display Controller

- [Write-back Pipeline: \[0\] \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Low-level Programming Models: \[3\]](#)
- [Display Controller Registers: \[4\]](#)

**Table 10-374. DISPC\_WB\_CONV\_COEF3**

<b>Address Offset</b>	0x0000 0580		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CBG								RESERVED								CBR							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	CBG	CbG coefficient Encoded signed value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	CBR	CbR coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-375. Register Call Summary for Register DISPC\_WB\_CONV\_COEF3**

Display Controller

- [Write-back Pipeline: \[0\] \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Low-level Programming Models: \[3\]](#)
- [Display Controller Registers: \[4\]](#)

**Table 10-376. DISPC\_WB\_CONV\_COEF4**

<b>Address Offset</b>	0x0000 0584		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CBB															

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
10:0	CBB	CbB Coefficient Encoded signed value (from -1024 to 1023).	RW	0x000

**Table 10-377. Register Call Summary for Register DISPC\_WB\_CONV\_COEF4**

Display Controller

- [Write-back Pipeline: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Low-level Programming Models: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-378. DISPC\_WB\_BUF\_SIZE\_STATUS**

<b>Address Offset</b>	0x0000 0588		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register defines the DMA buffer size for the write back pipeline.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:0	BUFSIZE	DMA buffer Size in number of 128-bits.	R	0x0800

**Table 10-379. Register Call Summary for Register DISPC\_WB\_BUF\_SIZE\_STATUS**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-380. DISPC\_WB\_BUF\_THRESHOLD**

<b>Address Offset</b>	0x0000 058C		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the DMA buffer associated with the write-back pipeline. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x07FF
15:0	BUFLOWTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x07F8

**Table 10-381. Register Call Summary for Register DISPC\_WB\_BUF\_THRESHOLD**

Display Controller

- [DMA Engine: \[0\] \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Low-level Programming Models: \[3\] \[4\]](#)
- [Display Controller Registers: \[5\]](#)

**Table 10-382. DISPC\_WB\_FIR**

<b>Address Offset</b>	0x0000 0590		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the resize factors for horizontal and vertical up/downsampling of the write back pipeline. It is used for ARGB and Y setting. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				FIRVINC												RESERVED				FIRHINC											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

**Table 10-383. Register Call Summary for Register DISPC\_WB\_FIR**

Display Controller

- [Write-back Pipeline: \[0\] \[1\] \[2\] \[3\]](#)
- [Shadows Registers: \[4\]](#)
- [Display Controller Low-level Programming Models: \[5\] \[6\]](#)
- [Display Controller Registers: \[7\]](#)

**Table 10-384. DISPC\_WB\_PICTURE\_SIZE**

<b>Address Offset</b>	0x0000 0594	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the size of the write-back picture associated with the write back pipeline after up/down-scaling. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ORGSIZEY								RESERVED								ORGSIZEX							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	ORGSIZEY	Number of lines of the video picture Encoded value (from 1 to 2048) to specify the number of lines of the video picture in memory (program to value minus 1).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	ORGSIZEX	Number of pixels of the video picture Encoded value (from 1 to 2048) to specify the number of pixels of the video picture in memory (program to value minus 1). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit.	RW	0x000

**Table 10-385. Register Call Summary for Register DISPC\_WB\_PICTURE\_SIZE**

Display Controller

- [Write-back Pipeline: \[0\] \[1\]](#)
- [Shadows Registers: \[2\]](#)
- [Display Controller Low-level Programming Models: \[3\] \[4\]](#)
- [Display Controller Registers: \[5\]](#)

**Table 10-386. DISPC\_WB\_PIXEL\_INC**

<b>Address Offset</b>	0x0000 0598		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the number of bytes to increment between two pixels for the buffer associated with the write back pipeline. The register is used only when the TILER is not present in the system in order to perform low performance rotation. When the TILER IP is present it is highly recommended to use it for performing the rotation. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PIXELINC															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
7:0	PIXELINC	Values other than 1 are invalid	RW	0x01

**Table 10-387. Register Call Summary for Register DISPC\_WB\_PIXEL\_INC**

Display Controller

- [DMA Engine: \[0\]](#)
- [Rotation and Mirroring: \[1\] \[2\] \[3\] \[4\]](#)
- [Shadows Registers: \[5\]](#)
- [Display Controller Low-level Programming Models: \[6\]](#)
- [Display Controller Registers: \[7\]](#)

**Table 10-388. DISPC\_WB\_ROW\_INC**

<b>Address Offset</b>	0x0000 05A4		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the number of bytes to increment at the end of the row for the buffer associated with the write back pipeline. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded signed value (from $2^{31}1$ to $2^{31}$ ) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value $1 + n * \text{bpp}$ means increment of n pixels. The value $1 (n + 1) * \text{bpp}$ means decrement of n pixels.	RW	0x0000 0001

**Table 10-389. Register Call Summary for Register DISPC\_WB\_ROW\_INC**

Display Controller

- [DMA Engine: \[0\]](#)
- [Rotation and Mirroring: \[1\] \[2\] \[3\] \[4\]](#)
- [Shadows Registers: \[5\]](#)
- [Display Controller Low-level Programming Models: \[6\]](#)
- [Display Controller Registers: \[7\]](#)

**Table 10-390. DISPC\_WB\_SIZE**

<b>Address Offset</b>	0x0000 05A8			
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC		
<b>Description</b>	The register configures the size of the output of overlay connected to the write-back pipeline when the overlay output is only used by the write-back pipeline. When the overlay is output on the primary LCD or secondary LCD or TV outputs, the size of the frame is defined in the <a href="#">DISPC_SIZE_LCD1</a> , <a href="#">DISPC_SIZE_LCD2</a> , and <a href="#">DISPC_SIZE_TV</a> respectively. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline			
<b>Type</b>	RW			
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
	RESERVED	SIZEY RESERVED SIZEX		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	SIZEY	Number of lines of the Write-back picture Encoded value (from 1 to 2048) to specify the number of lines of the write-back picture. Program to value minus 1.	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	SIZEX	Number of pixels of the Write-back picture Encoded value (from 1 to 2048) to specify the number of pixels of the write-back picture. Program to value minus 1.	RW	0x000

**Table 10-391. Register Call Summary for Register DISPC\_WB\_SIZE**

Display Controller

- [Write-back Pipeline: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Shadows Registers: \[5\]](#)
- [Display Controller Low-level Programming Models: \[6\] \[7\]](#)
- [Display Controller Registers: \[8\]](#)

**Table 10-392. DISPC\_VID1\_BA\_UV\_j**

<b>Address Offset</b>	0x0000 0600 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the base address of the UV buffer for the video window 1. (DISPC_VID1_BA_UV_0 & DISPC_VID1_BA_UV_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID1_BA_UV_0 is used). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2</a> .GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address aligned on 16-bit boundary Base address of the UV video buffer used only in case of YUV4:2:0-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

**Table 10-393. Register Call Summary for Register DISPC\_VID1\_BA\_UV\_j**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Logical Register Mapping: \[1\]](#)
  - [Display Controller Registers: \[2\]](#)

**Table 10-394. DISPC\_VID2\_BA\_UV\_j**

<b>Address Offset</b>	0x0000 0608 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the base address of the UV buffer for the video window 2. (DISPC_VID2_BA_UV_0 & DISPC_VID2_BA_UV_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID2_BA_UV_0 is used). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2</a> .GOWB is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address aligned on 16-bit boundary Base address of the UV video buffer used only in case of YUV4:2:0-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

**Table 10-395. Register Call Summary for Register DISPC\_VID2\_BA\_UV\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-396. DISPC\_VID3\_BA\_UV\_j**

<b>Address Offset</b>	0x0000 0610 + (0x4 * j)	<b>Index</b>	j = 0 to 1																																																																
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC																																																																
<b>Description</b>	The register configures the base address of the UV buffer for the video window 3. (DISPC_VID3_BA_UV_0 & DISPC_VID3_BA_UV_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID3_BA_UV_0 is used). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory																																																																		
<b>Type</b>	RW																																																																		
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td style="background-color: #ffff00;">23</td><td style="background-color: #ffff00;">22</td><td style="background-color: #ffff00;">21</td><td style="background-color: #ffff00;">20</td><td style="background-color: #ffff00;">19</td><td style="background-color: #ffff00;">18</td><td style="background-color: #ffff00;">17</td><td style="background-color: #ffff00;">16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td style="background-color: #ffff00;">7</td><td style="background-color: #ffff00;">6</td><td style="background-color: #ffff00;">5</td><td style="background-color: #ffff00;">4</td><td style="background-color: #ffff00;">3</td><td style="background-color: #ffff00;">2</td><td style="background-color: #ffff00;">1</td><td style="background-color: #ffff00;">0</td> </tr> <tr> <td colspan="32">BA</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BA																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
BA																																																																			

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address aligned on 16-bit boundary Base address of the UV video buffer used only in case of YUV4:2:0-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

**Table 10-397. Register Call Summary for Register DISPC\_VID3\_BA\_UV\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)



**Table 10-398. DISPC\_WB\_BA\_UV\_j**

<b>Address Offset</b>	0x0000 0618 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the base address of the UV buffer for the write-back pipeline. (DISPC_WB_BA_UV_0 & DISPC_WB_BA_UV_1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_WB_BA_UV_0 is used). Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address aligned on 16-bit boundary Base address of the UV video buffer used only in case of YUV4:2:0-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0000 0000

**Table 10-399. Register Call Summary for Register DISPC\_WB\_BA\_UV\_j**

Display Controller

- [DMA Engine: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Low-level Programming Models: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-400. DISPC\_CONFIG2**

<b>Address Offset</b>	0x0000 0620	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The control register configures the Display Controller module for the secondary LCD output. Shadow register, updated on VFP start period of secondary LCD or EVSYNC		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RESERVED								FULLRANGE	COLORCONVENABLE	FIDFIRST	OUTPUTMODEENABLE	RESERVED				BUFFERHANDCHECK	CPR	RESERVED				TKKLDSELECTION	TKKLDENABLE	RESERVED	ACBIASGATED	VSYNCGATED	HSYNCGATED	PIXELCLOCKGATED	PIXELDATAGATED	RESERVED				PIXELGATED

## Display Controller

www.ti.com

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
25	FULLRANGE	Color Space Conversion full range setting. 0x0: Limited range selected. 0x1: Full range selected.	RW	0
24	COLORCONV ENABLE	Enable the color space conversion. It shall be reset when CPR bit field is set to 0x1. 0x0: Disable Color Space Conversion RGB to YUV 0x1: Enable Color Space Conversion RGB to YUV	RW	0
23	FIDFIRST	Selects the first field to output in case of interlace mode. In case of progressive mode, the value is not used. 0x0: First field is even. 0x1: Odd field is first.	RW	0
22	OUTPUTMODE ENABLE	Selects between progressive and interlace mode for the secondary LCD output. 0x0: Progressive mode selected. 0x1: Interlace mode selected.	RW	0
21:17	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
16	BUFFERHAND CHECK	Controls the handshack between DMA buffer and STALL signal in order to prevent from underflow. The bit shall be set to 0 when the module is not in STALL mode. (secondary LCD output) 0x0: Only the STALL signal (generated by RFBI or DS12 depending on which IP uses the LCD output) is used regardless of the DMA buffer fullness information in order to provide data to the RFBI or DS2 module. 0x1: The STALL signal (generated by RFBI or DS12 depending on which IP uses the LCD output) is used in combination with the DMA buffer fullness information in order to provide data to the RFBI or DS12 module only when it does not generated buffer underflow.	RW	0
15	CPR	Color Phase Rotation Control secondary LCD output). It shall be reset when ColorConvEnable bit field is set to 1. wr: VFP start period of secondary LCD output 0x0: Color Phase Rotation Disabled 0x1: Color Phase Rotation Enabled	RW	0
14:12	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
11	TCKLCD SELECTION	Transparency Color Key Selection (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Destination transparency color key selected 0x1: Source transparency color key selected	RW	0
10	TCKLCDENABLE	Transparency Color Key Enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Disable the transparency color key for the LCD 0x1: Enable the transparency color key for the LCD	RW	0
9	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
8	ACBIASGATED	ACBias Gated Enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: ACBias Gated Disabled 0x1: ACBias Gated Enabled	RW	0
7	VSYNCGATED	VSYNC Gated Enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: VSYNC Gated Disabled 0x1: VSYNC Gated Enabled	RW	0
6	HSYNCGATED	HSYNC Gated Enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: HSYNC Gated Disabled 0x1: HSYNC Gated Enabled	RW	0

Bits	Field Name	Description	Type	Reset
5	PIXELCLOCK GATED	Pixel Clock Gated Enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Pixel Clock Gated Disabled 0x1: Pixel Clock Gated Enabled	RW	0
4	PIXELDATA GATED	Pixel Data Gated Enabled (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Pixel Data Gated Disabled 0x1: Pixel Data Gated Enabled	RW	0
3:1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
0	PIXELGATED	Pixel Gated Enable (only for Active Matrix) (secondary LCD output) wr: VFP start period of secondary LCD output 0x0: Pixel clock always toggles (only in Active Matrix mode) 0x1: Pixel clock only toggles when there is valid data to display. (only in Active Matrix mode)	RW	0

**Table 10-401. Register Call Summary for Register DISPC\_CONFIG2**

Display Controller

- [LCD Outputs: \[0\] \[1\] \[2\]](#)
- [Shadows Registers: \[3\]](#)
- [Display Controller Logical Register Mapping: \[4\]](#)
- [Display Controller Registers: \[5\]](#)

**Table 10-402. DISPC\_VID1\_ATTRIBUTES2**

<b>Address Offset</b>	0x0000 0624	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the attributes of the video window 1. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																YUVCHROMASAMPLING	RESERVED	VC1_RANGE_CBCCR	VC1_RANGE_Y	VC1ENABLE											

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
8	YUVCHROMARE SAMPLING	The YUV chrominance can be resampled using averaging of the adjacent chrominance samples, without using the polyphase filter for 4:2:2 input or can be calculated using the polyphase filter for 4:2:2/4:2:0. The polyphase filter is mandatory for the 4:2:0 format. This bit controls the order in which the processing is done on the video pipe.  0x0: When input is 4:2:2, the missing chrominance samples are calculated by averaging the adjacent samples if <a href="#">DISPC_VID1_ATTRIBUTES. ROTATION=0</a> only. Other rotation configurations are not supported.  0x1: For 4:2:2 (or 4:2:0), the missing chrominance samples are calculated by filtering the adjacent samples (5-tap polyphase filter). See <a href="#">Figure 10-47, Configuration 2: Video Pipeline</a> . All rotation configurations are supported.	RW	0
7	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
6:4	VC1_RANGE_CBCR	Defines the VC-1 range value for the CbCr component from 0 to 7.	RW	0x0
3:1	VC1_RANGE_Y	Defines the VC-1 range value for the Y component from 0 to 7.	RW	0x0
0	VC1ENABLE	Enable/disable the VC-1 range mapping processing. The bit field is ignored if the format is not one of the supported YUV formats.  0x0: VC-1 range mapping disabled 0x1: VC-1 range mapping enabled	RW	0

**Table 10-403. Register Call Summary for Register DISPC\_VID1\_ATTRIBUTES2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-404. DISPC\_VID2\_ATTRIBUTES2**

<b>Address Offset</b>	0x0000 0628	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the attributes of the video window 2. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																YUVCHROMARE SAMPLING	RESERVED	VC1_RANGE_CBCR	VC1_RANGE_Y	VC1ENABLE											

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
8	YUVCHROMARE SAMPLING	The YUV chrominance can be resampled using averaging of the adjacent chrominance samples, without using the polyphase filter for 4:2:2 input or can be calculated using the polyphase filter for 4:2:2/4:2:0. The polyphase filter is mandatory for the 4:2:0 format. This bit controls the order in which the processing is done on the video pipe.  0x0: When input is in 4:2:2, the missing chrominance samples are calculated by averaging the adjacent samples if <a href="#">DISPC_VID1_ATTRIBUTES</a> . ROTATION=0 only. Other rotation configurations are not supported.  0x1: For 4:2:2 (or 4:2:0), the missing chrominance samples are calculated by filtering the adjacent samples (5-tap polyphase filter). See <a href="#">Figure 10-47, Configuration 2: Video Pipeline</a> . All rotation configurations are supported.	RW	0
7	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
6:4	VC1_RANGE_CBCR	Defines the VC-1 range value for the CbCr component from 0 to 7.	RW	0x0
3:1	VC1_RANGE_Y	Defines the VC-1 range value for the Y component from 0 to 7.	RW	0x0
0	VC1ENABLE	Enable/disable the VC-1 range mapping processing. The bit field is ignored if the format is not one of the supported YUV formats.  0x0: VC-1 range mapping disabled 0x1: VC-1 range mapping enabled	RW	0

**Table 10-405. Register Call Summary for Register DISPC\_VID2\_ATTRIBUTES2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-406. DISPC\_VID3\_ATTRIBUTES2**

<b>Address Offset</b>	0x0000 062C	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the attributes of the video window 3. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																YUVCHROMARE SAMPLING	RESERVED	VC1_RANGE_CBCR			VC1_RANGE_Y			VC1ENABLE							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x000000
8	YUVCHROMARE SAMPLING	The YUV chrominance can be resampled using averaging of the adjacent chrominance samples, without using the polyphase filter for 4:2:2 input or can be calculated using the polyphase filter for 4:2:2/4:2:0. The polyphase filter is mandatory for the 4:2:0 format. This bit controls the order in which the processing is done on the video pipe.  0x0: When input is in 4:2:2, the missing chrominance samples are calculated by averaging the adjacent samples if <a href="#">DISPC_VID1_ATTRIBUTES</a> . ROTATION=0 only. Other rotation configurations are not supported.  0x1: For 4:2:2 (or 4:2:0), the missing chrominance samples are calculated by filtering the adjacent samples (5-tap polyphase filter). See <a href="#">Figure 10-47, Configuration 2: Video Pipeline</a> . All rotation configurations are supported.	RW	0
7	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
6:4	VC1_RANGE_CBCR	Defines the VC-1 range value for the CbCr component from 0 to 7.	RW	0x0
3:1	VC1_RANGE_Y	Defines the VC-1 range value for the Y component from 0 to 7.	RW	0x0
0	VC1ENABLE	Enable/disable the VC-1 range mapping processing. The bit field is ignored if the format is not one of the supported YUV formats.  0x0: VC-1 range mapping disabled 0x1: VC-1 range mapping enabled	RW	0

**Table 10-407. Register Call Summary for Register DISPC\_VID3\_ATTRIBUTES2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-408. DISPC\_GAMMA\_TABLE0**

<b>Address Offset</b>	0x0000 0630																															
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .																															
<b>Description</b>	The register configures the look up table used as color look up table for BITMAP formats (1-, 2-, 4, and 8-bpp) on the graphics pipeline or as gamma table on the primary LCD output.																															
<b>Type</b>	W																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INDEX								VALUE_R								VALUE_G								VALUE_B							
<b>Bits</b>	31:24								23:16								15:8								7:0							
<b>Field Name</b>	INDEX								VALUE_R								VALUE_G								VALUE_B							
<b>Description</b>	Defines the location in the table where the bit field VALUE is stored.								8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX.								8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX.								8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX.							
<b>Type</b>	W								W								W								W							
<b>Reset</b>	0x00								0x00								0x00								0x00							

**Table 10-409. Register Call Summary for Register DISPC\_GAMMA\_TABLE0**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Low-level Programming Models: \[1\] \[2\]](#)
- [Display Controller Logical Register Mapping: \[3\]](#)
- [Display Controller Registers: \[4\]](#)

**Table 10-410. DISPC\_GAMMA\_TABLE1**

<b>Address Offset</b>	0x0000 0634		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the gamma table on the secondary LCD output.		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDEX								VALUE_R								VALUE_G								VALUE_B							

Bits	Field Name	Description	Type	Reset
31:24	INDEX	Defines the location in the table where the bit field VALUE is stored.	W	0x00
23:16	VALUE_R	8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX.	W	0x00
15:8	VALUE_G	8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX.	W	0x00
7:0	VALUE_B	8-bit value used to defined the value to store at the location in the table defined by the bit field INDEX.	W	0x00

**Table 10-411. Register Call Summary for Register DISPC\_GAMMA\_TABLE1**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Low-level Programming Models: \[1\]](#)
- [Display Controller Logical Register Mapping: \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-412. DISPC\_GAMMA\_TABLE2**

<b>Address Offset</b>	0x0000 0638		
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the gamma table on the TV output.		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDEX	RESERVED	VALUE_R										VALUE_G										VALUE_B									

Bits	Field Name	Description	Type	Reset
31	INDEX	Setting this bit to 1 resets the internal index counter to zero. Each subsequent access to the register (with the INDEX bit kept at 0) increments the address for the next storage location into the table memory.	W	0
30	RESERVED		W	0
29:20	VALUE_R	10-bit color component value to store in the table.	W	0x000
19:10	VALUE_G	10-bit color component value to store in the table.	W	0x000
9:0	VALUE_B	10-bit color component value to store in the table.	W	0x000

**Table 10-413. Register Call Summary for Register DISPC\_GAMMA\_TABLE2**

Display Controller

- [TV Output: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Low-level Programming Models: \[2\]](#)
- [Display Controller Logical Register Mapping: \[3\]](#)
- [Display Controller Registers: \[4\]](#)

**Table 10-414. DISPC\_VID1\_FIR2**

<b>Address Offset</b>	0x0000 063C	
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b> DISPC
<b>Description</b>	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 1. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				FIRVINC												RESERVED				FIRHINC											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

**Table 10-415. Register Call Summary for Register DISPC\_VID1\_FIR2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)



**Table 10-416. DISPC\_VID1\_ACCU2\_j**

<b>Address Offset</b>	0x0000 0640 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	<p>The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 1 (DISPC_VID1_ACCU2_0 &amp; DISPC_VID1_ACCU2_1 for ping-pong mechanism with external trigger, based on the field polarity) It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU								RESERVED								HORIZONTALACCU							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from -1024 to 1023).	RW	0x000

**Table 10-417. Register Call Summary for Register DISPC\_VID1\_ACCU2\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-418. DISPC\_VID1\_FIR\_COEF\_H2\_i**

<b>Address Offset</b>	0x0000 0648 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	<p>The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-419. Register Call Summary for Register DISPC\_VID1\_FIR\_COEF\_H2\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-420. DISPC\_VID1\_FIR\_COEF\_HV2\_i**

<b>Address Offset</b>	0x0000 064C + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-421. Register Call Summary for Register DISPC\_VID1\_FIR\_COEF\_HV2\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-422. DISPC\_VID1\_FIR\_COEF\_V2\_i**

<b>Address Offset</b>	0x0000 0688 + (0x4 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 1 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC22								FIRVC00															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

**Table 10-423. Register Call Summary for Register DISPC\_VID1\_FIR\_COEF\_V2\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-424. DISPC\_VID2\_FIR2**

<b>Address Offset</b>	0x0000 06A8	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 2. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				FIRVINC												RESERVED				FIRHINC											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

Bits	Field Name	Description	Type	Reset
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

**Table 10-425. Register Call Summary for Register DISPC\_VID2\_FIR2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-426. DISPC\_VID2\_ACCU2\_j**

Address Offset	0x0000 06AC + (0x4 * j)	Index	j = 0 to 1
Physical Address	See <a href="#">Table 10-133</a> .	Instance	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 2 (DISPC_VID2_ACCU2_0 & DISPC_VID2_ACCU2_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU								RESERVED								HORIZONTALACCU							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from -1024 to 1023).	RW	0x000

**Table 10-427. Register Call Summary for Register DISPC\_VID2\_ACCU2\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-428. DISPC\_VID2\_FIR\_COEF\_H2\_i**

<b>Address Offset</b>	0x0000 06B4 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
FIRHC3	FIRHC2	FIRHC1	FIRHC0

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-429. Register Call Summary for Register DISPC\_VID2\_FIR\_COEF\_H2\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-430. DISPC\_VID2\_FIR\_COEF\_HV2\_i**

<b>Address Offset</b>	0x0000 06B8 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
FIRVC2	FIRVC1	FIRVC0	FIRHC4

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-431. Register Call Summary for Register DISPC\_VID2\_FIR\_COEF\_HV2\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-432. DISPC\_VID2\_FIR\_COEF\_V2\_i**

<b>Address Offset</b>	0x0000 06F4 + (0x4 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 2 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC22								FIRVC00															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

**Table 10-433. Register Call Summary for Register DISPC\_VID2\_FIR\_COEF\_V2\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-434. DISPC\_VID3\_FIR2**

<b>Address Offset</b>	0x0000 0724	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the resize factors for horizontal and vertical up/downsampling of the video window 3. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				FIRVINC												RESERVED				FIRHINC											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

**Table 10-435. Register Call Summary for Register DISPC\_VID3\_FIR2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-436. DISPC\_VID3\_ACCU2\_j**

Address Offset	0x0000 0728 + (0x4 * j)	Index	j = 0 to 1
Physical Address	See <a href="#">Table 10-133</a> .	Instance	DISPC
Description	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the video window 3 (DISPC_VID3_ACCU2_0 & DISPC_VID3_ACCU2_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU								RESERVED								HORIZONTALACCU							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from -1024 to 1023).	RW	0x000

**Table 10-437. Register Call Summary for Register DISPC\_VID3\_ACCU2\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Registers: \[1\]](#)

**Table 10-438. DISPC\_VID3\_FIR\_COEF\_H2\_i**

<b>Address Offset</b>	0x0000 0730 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-439. Register Call Summary for Register DISPC\_VID3\_FIR\_COEF\_H2\_i**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Logical Register Mapping: \[1\]](#)
- [Display Controller Registers: \[2\]](#)

**Table 10-440. DISPC\_VID3\_FIR\_COEF\_HV2\_i**

<b>Address Offset</b>	0x0000 0734 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00



**Table 10-441. Register Call Summary for Register DISPC\_VID3\_FIR\_COEF\_HV2\_i**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Logical Register Mapping: \[1\]](#)
  - [Display Controller Registers: \[2\]](#)

**Table 10-442. DISPC\_VID3\_FIR\_COEF\_V2\_i**

<b>Address Offset</b>	0x0000 0770 + (0x4 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the video window 3 for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated on VFP start period of primary LCD or VFP start period of the secondary LCD or EVSYNC or when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline). The synchronization event is defined based on the output using the pipeline: primary LCD, secondary LCD, TV output or write-back to the memory		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC22								FIRVC00															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

**Table 10-443. Register Call Summary for Register DISPC\_VID3\_FIR\_COEF\_V2\_i**

- Display Controller
- [Shadows Registers: \[0\]](#)
  - [Display Controller Logical Register Mapping: \[1\]](#)
  - [Display Controller Registers: \[2\]](#)

**Table 10-444. DISPC\_WB\_FIR2**

<b>Address Offset</b>	0x0000 0790	<b>Instance</b>	DISPC
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .		
<b>Description</b>	The register configures the resize factors for horizontal and vertical up/downsampling of the write-back pipeline. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				FIRVINC												RESERVED				FIRHINC											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
28:16	FIRVINC	Vertical increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400
15:13	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
12:0	FIRHINC	Horizontal increment of the up/downsampling filter for Cb and Cr. Encoded value (from 1 to 4096). The value 0 is invalid. The values greater than 4096 are invalid.	RW	0x0400

**Table 10-445. Register Call Summary for Register DISPC\_WB\_FIR2**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Low-level Programming Models: \[1\] \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-446. DISPC\_WB\_ACCU2\_j**

<b>Address Offset</b>	0x0000 0794 + (0x4 * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-133</a> .	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the resize accumulator init values for horizontal and vertical up/downsampling of the write back pipeline (DISPC_WB_ACCU2_0 & DISPC_WB_ACCU2_1 for ping-pong mechanism with external trigger, based on the field polarity). It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU								RESERVED								HORIZONTALACCU							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
26:16	VERTICALACCU	Vertical initialization accu value Encoded value (from -1024 to 1023).	RW	0x000
15:11	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
10:0	HORIZONTALACCU	Horizontal initialization accu value Encoded value (from -1024 to 1023).	RW	0x000

**Table 10-447. Register Call Summary for Register DISPC\_WB\_ACCU2\_j**

Display Controller

- [Shadows Registers: \[0\]](#)
- [Display Controller Low-level Programming Models: \[1\] \[2\]](#)
- [Display Controller Registers: \[3\]](#)

**Table 10-448. DISPC\_WB\_FIR\_COEF\_H2\_i**

<b>Address Offset</b>	0x0000 07A0 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a>	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRHC3								FIRHC2								FIRHC1								FIRHC0							

Bits	Field Name	Description	Type	Reset
31:24	FIRHC3	Signed coefficient C3 for the horizontal up/down-scaling with the phase n	RW	0x00
23:16	FIRHC2	Unsigned coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x00
15:8	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x00
7:0	FIRHC0	Signed coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-449. Register Call Summary for Register DISPC\_WB\_FIR\_COEF\_H2\_i**

Display Controller

- [Write-back Pipeline: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Shadows Registers: \[5\]](#)
- [Display Controller Low-level Programming Models: \[6\]](#)
- [Display Controller Registers: \[7\]](#)

**Table 10-450. DISPC\_WB\_FIR\_COEF\_HV2\_i**

<b>Address Offset</b>	0x0000 07A4 + (0x8 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a>	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical and horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIRVC2								FIRVC1								FIRVC0								FIRHC4							

Bits	Field Name	Description	Type	Reset
31:24	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x00
23:16	FIRVC1	Unsigned coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x00
15:8	FIRVC0	Signed coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x00

Bits	Field Name	Description	Type	Reset
7:0	FIRHC4	Signed coefficient C4 for the horizontal up/down-scaling with the phase n	RW	0x00

**Table 10-451. Register Call Summary for Register DISPC\_WB\_FIR\_COEF\_HV2\_i**

Display Controller

- [Write-back Pipeline: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Shadows Registers: \[6\]](#)
- [Display Controller Low-level Programming Models: \[7\] \[8\]](#)
- [Display Controller Registers: \[9\]](#)

**Table 10-452. DISPC\_WB\_FIR\_COEF\_V2\_i**

<b>Address Offset</b>	0x0000 07E0 + (0x4 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 10-133</a>	<b>Instance</b>	DISPC
<b>Description</b>	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the write back pipeline for the phases from 0 to 7. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the hardware, any value can be used for the bit fields. Shadow register, updated when <a href="#">DISPC_CONTROL2.GOWB</a> is set to 1 by software and current WB frame is finished (no more data in the write-back pipeline) when the WB pipeline is directly connected to one of the pipelines (graphics or video), otherwise updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC22								FIRVC00															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
15:8	FIRVC22	Signed coefficient C22 for the vertical up/down-scaling with the phase n	RW	0x00
7:0	FIRVC00	Signed coefficient C00 for the vertical up/down-scaling with the phase n	RW	0x00

**Table 10-453. Register Call Summary for Register DISPC\_WB\_FIR\_COEF\_V2\_i**

Display Controller

- [Write-back Pipeline: \[0\] \[1\] \[2\]](#)
- [Shadows Registers: \[3\]](#)
- [Display Controller Low-level Programming Models: \[4\]](#)
- [Display Controller Registers: \[5\]](#)

**Table 10-454. DISPC\_GLOBAL\_BUFFER**

<b>Address Offset</b>	0x0000 0800		
<b>Physical Address</b>	See <a href="#">Table 10-133</a>	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the DMA buffers allocations to the pipeline (graphics, video1, video2, video3 and write-back). Both TOP and BOTTOM must be allocated to the same pipeline.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		WB_BOTTOM_BUFFER		WB_TOP_BUFFER		VID3_BOTTOM_BUFFER		VID3_TOP_BUFFER		VID2_BOTTOM_BUFFER		VID2_TOP_BUFFER		VID1_BOTTOM_BUFFER		VID1_TOP_BUFFER		GFX_BOTTOM_BUFFER		GFX_TOP_BUFFER											

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
29:27	WB_BOTTOM_BUFFER	Write-back DMA BOTTOM buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to write-back pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x4
26:24	WB_TOP_BUFFER	Write-back DMA TOP buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to write-back pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x4
23:21	VID3_BOTTOM_BUFFER	Video3 DMA BOTTOM buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to video3 pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x3
20:18	VID3_TOP_BUFFER	Video3 DMA TOP buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to video3 pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x3
17:15	VID2_BOTTOM_BUFFER	Video2 DMA BOTTOM buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to video2 pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x2

Bits	Field Name	Description	Type	Reset
14:12	VID2_TOP_BUFFER	Video2 DMA TOP buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to video2 pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x2
11:9	VID1_BOTTOM_BUFFER	Video1 DMA BOTTOM buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to video1 pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x1
8:6	VID1_TOP_BUFFER	Video1 DMA TOP buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to video 1 pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x1
5:3	GFX_BOTTOM_BUFFER	Graphics DMA BOTTOM buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to graphics pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x0
2:0	GFX_TOP_BUFFER	Graphics DMA TOP buffer (half of the full DMA buffer) allocation to one of the pipelines. By default to graphics pipeline. 0x0: DMA buffer allocated to the graphics pipeline. 0x1: DMA buffer allocated to the video1 pipeline. 0x2: DMA buffer allocated to the video2 pipeline. 0x3: DMA buffer allocated to the video3 pipeline. 0x4: DMA buffer allocated to the write-back pipeline.	RW	0x0

**Table 10-455. Register Call Summary for Register DISPC\_GLOBAL\_BUFFER**

Display Controller

- [DMA Engine: \[0\]](#)
- [Shadows Registers: \[1\]](#)
- [Display Controller Registers: \[2\] \[3\]](#)

**Table 10-456. DISPC\_DIVISOR**

<b>Address Offset</b>	0x0000 0804		
<b>Physical Address</b>	See <a href="#">Table 10-133</a>	<b>Instance</b>	DISPC
<b>Description</b>	The register configures the divisor value for generating the core functional clock. There is a backward compatibility mode enabled by default in order to use <a href="#">DISPC_DIVISOR1.LCD</a> value instead of <a href="#">DISPC_DIVISOR.LCD</a> bit field for generating the core functional clock.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LCD								RESERVED								ENABLE							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00
23:16	LCD	Display Controller Logic Clock Divisor Value (from 1 to 255) to specify the frequency of the Display Controller logic clock based on the function clock. The value 0 is invalid.	RW	0x01
15:1	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000
0	ENABLE	When the bit field is set to 1, the bit field LCD is used to generated the core functional clock from the input clock. When the bit field is set to 0, the value <a href="#">DISPC_DIVISOR1.LCD</a> is used instead. 0x0: <a href="#">DISPC_DIVISOR1.LCD</a> bit field is used 0x1: <a href="#">DISPC_DIVISOR.LCD</a> bit field is used	RW	0

**Table 10-457. Register Call Summary for Register DISPC\_DIVISOR**

Display Controller

- [Clock configuration: \[0\] \[1\] \[2\] \[3\]](#)
- [LCD Outputs: \[4\] \[5\]](#)
- [Shadows Registers: \[6\]](#)
- [Display Controller Registers: \[7\] \[8\] \[9\]](#)

**Table 10-458. DISPC\_WB\_ATTRIBUTES2**

<b>Address Offset</b>	0x0000 0810		
<b>Physical Address</b>	See <a href="#">Table 10-133</a>		
<b>Description</b>	The register set the counter to control the delay to flush the WB pipe after the end of the frame in capture mode. Shadow register, updated on VFP start period of primary LCD or VFP start period of secondary LCD or EVSYNC depending on which overlay output is selected as an input to the WB pipeline		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WBDELAYCOUNT															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	WBDELAYCOUNT	Delays the WB pipe flush after the end of the frame.delay = n x (1/F_clk) n = 0:255	RW	0x00

---

**Table 10-459. Register Call Summary for Register DISPC\_WB\_ATTRIBUTES2**

---

Display Controller

- DMA Engine: [0] [1] [2]
  - Shadows Registers: [3] [4] [5]
  - Display Controller Registers: [6]
-



## 10.3 MIPI Display Serial Interface

This section describes the MIPI® display serial interface (DSI) module for the device.

---

**NOTE:** This section contains information that is Copyright © 2005-2008 MIPI Alliance, Inc. All rights reserved.

MIPI Alliance Member Confidential.

All rights reserved. This material is reprinted with the permission of the MIPI Alliance, Inc. No part(s) of this document may be disclosed, reproduced or used for any purpose other than as needed to support the use of the products of TI.

See [OMAP4430 MIPI Disclaimer](#) for details.

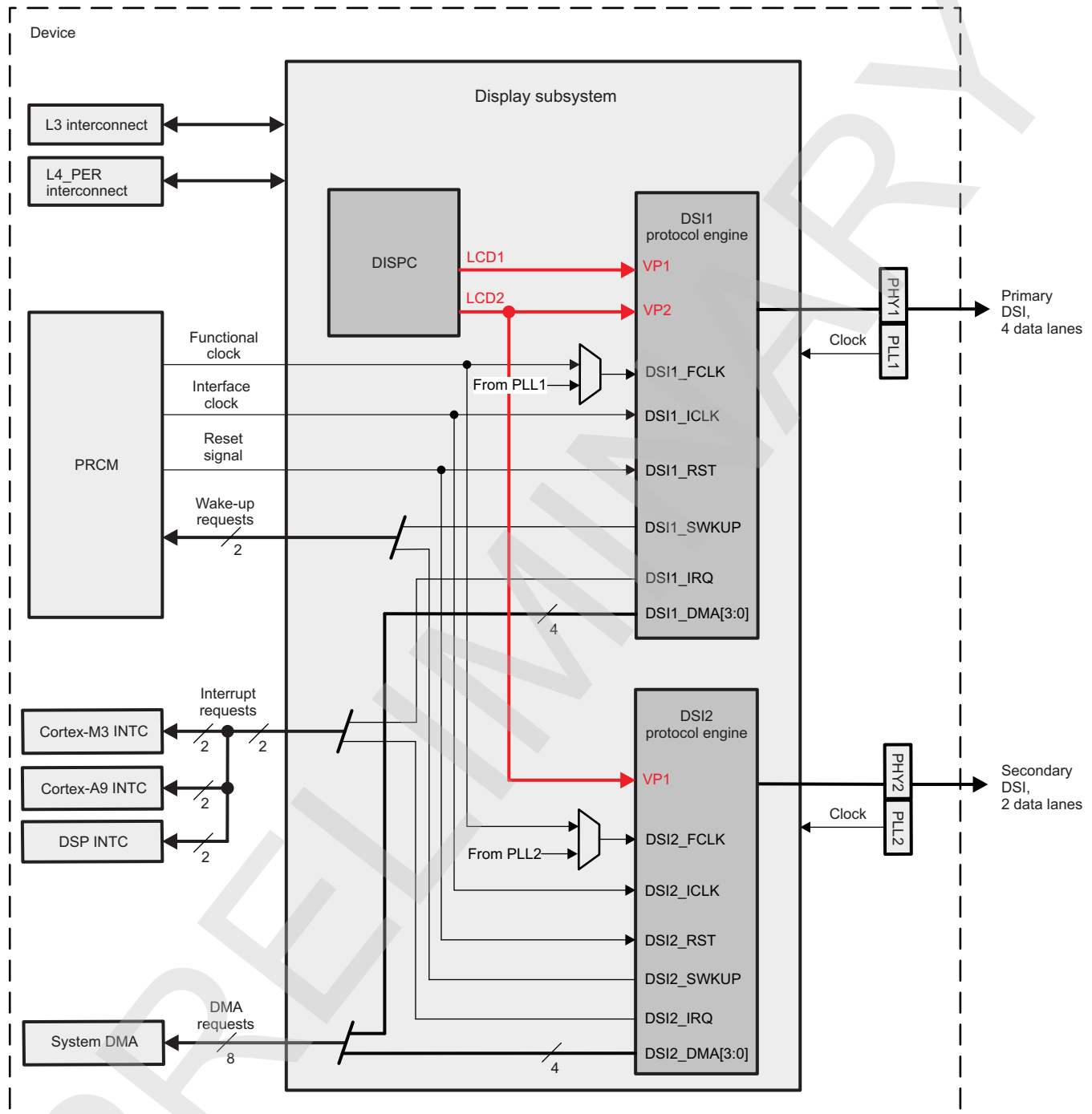
---

### 10.3.1 MIPI DSI Overview

The display serial interface (DSI) module connects to a DSI display module directly or through an external DSI bridge. The DSI module is connected to DISPC and L3/L4\_PER interconnects

[Figure 10-94](#) is a block diagram of the DSI module.

Figure 10-94. DSI Overview



ds1-001

There are two DSI modules in the display subsystem (DSS) that provide the following main features:

- DSI 1 (primary)
  - Supports MIPI-DSI (up to four data lanes + one clock lane)
  - Supports video mode and command mode
  - Support multiple displays (one video mode and one command mode). In command mode, pixel data are provided from the level 3 (L3) interface or from one of the overlay manager outputs.
  - Bidirectional data link support for command mode (only one data lane is used in reverse direction)
  - RGB16, RGB18 nonpacked, and RGB24 formats supported for command mode

- RGB16, RGB18 packed and nonpacked, and RGB24 formats supported for video mode
- Burst support for the video mode
- Supports up to four data-configurable lanes, in addition to the clock signaling
- Maximum data rate of 824 Mbps per data pair for four-data lane configuration, and 900 Mbps for up to three-data lane configuration
- Data splitter for 2-data, 3-data and 4-data lane configurations
- Data interleaving support for one synchronous stream (video mode) from the DISPC and up to three asynchronous streams (command mode) from the interconnect concurrently, or up to four asynchronous streams (command mode) from the interconnect
- Transfer of pixels and data received on the video port or L3 interconnect to the display through DSI DSI\_PHY
- Serial configuration port (SCP) for the DSI\_PHY complex input/output (I/O) and DSI phase-locked loop (PLL)
- Connection to the DSI\_PHY complex I/O through PPI
- Error-correction code (ECC) and checksum generation
- MIPI DCS support
- DSI 2 (secondary)
  - Supports MIPI-DSI (up to two data lanes + one clock lane)
  - Supports video mode and command mode
  - Support multiple displays (one video mode and one command mode). In command mode, pixel data are provided from the L3 interface or from one of the overlay manager outputs.
  - Bidirectional data link support for command mode (only one data lane is used in reverse direction)
  - RGB16, RGB18 nonpacked, and RGB24 formats supported for command mode
  - RGB16, RGB18 packed and nonpacked, and RGB24 formats supported for video mode
  - Burst support for the video mode
  - Supports one data-configurable lane, in addition to the clock signaling
  - Maximum data rate of 900 Mbps per data pair
  - Data splitter for 2-data lane configuration
  - Data interleaving support for one synchronous stream (video mode) from the DISPC and up to three asynchronous streams (command mode) from the interconnect concurrently, or up to four asynchronous streams (command mode) from the interconnect
  - Transfer pixels and data received on the video port or Interface interconnect to the display through DSI DSI\_PHY
  - Serial configuration port (SCP) for the DSI\_PHY complex I/O and DSI PLL
  - Connection to the DSI\_PHY complex I/O through the physical layer (PHY) protocol interface (PPI)
  - ECC and checksum generation
  - MIPI DCS support

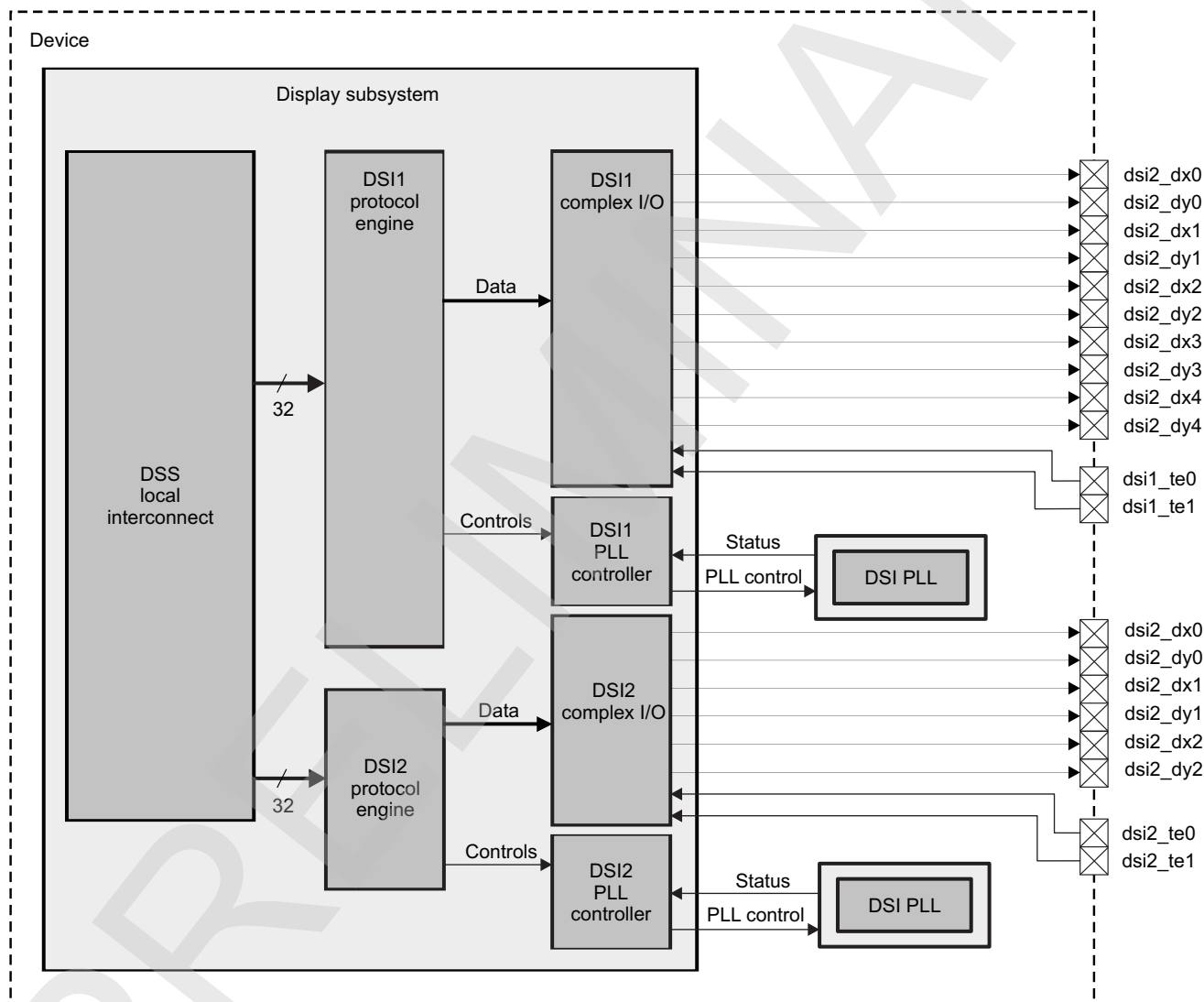
### 10.3.2 DSI Environment

**NOTE:** Copyright 2005-2008 MIPI Alliance, Inc. All rights reserved. MIPI Alliance Member Confidential.

This section describes the DSI application fields from an environment point of view (external connections). It describes the DSI connectivity options, lists all possible interfaces, and describes the protocol and data format used in each case.

Figure 10-95 is a block diagram of the DSI environment.

**Figure 10-95. DSI Environment**

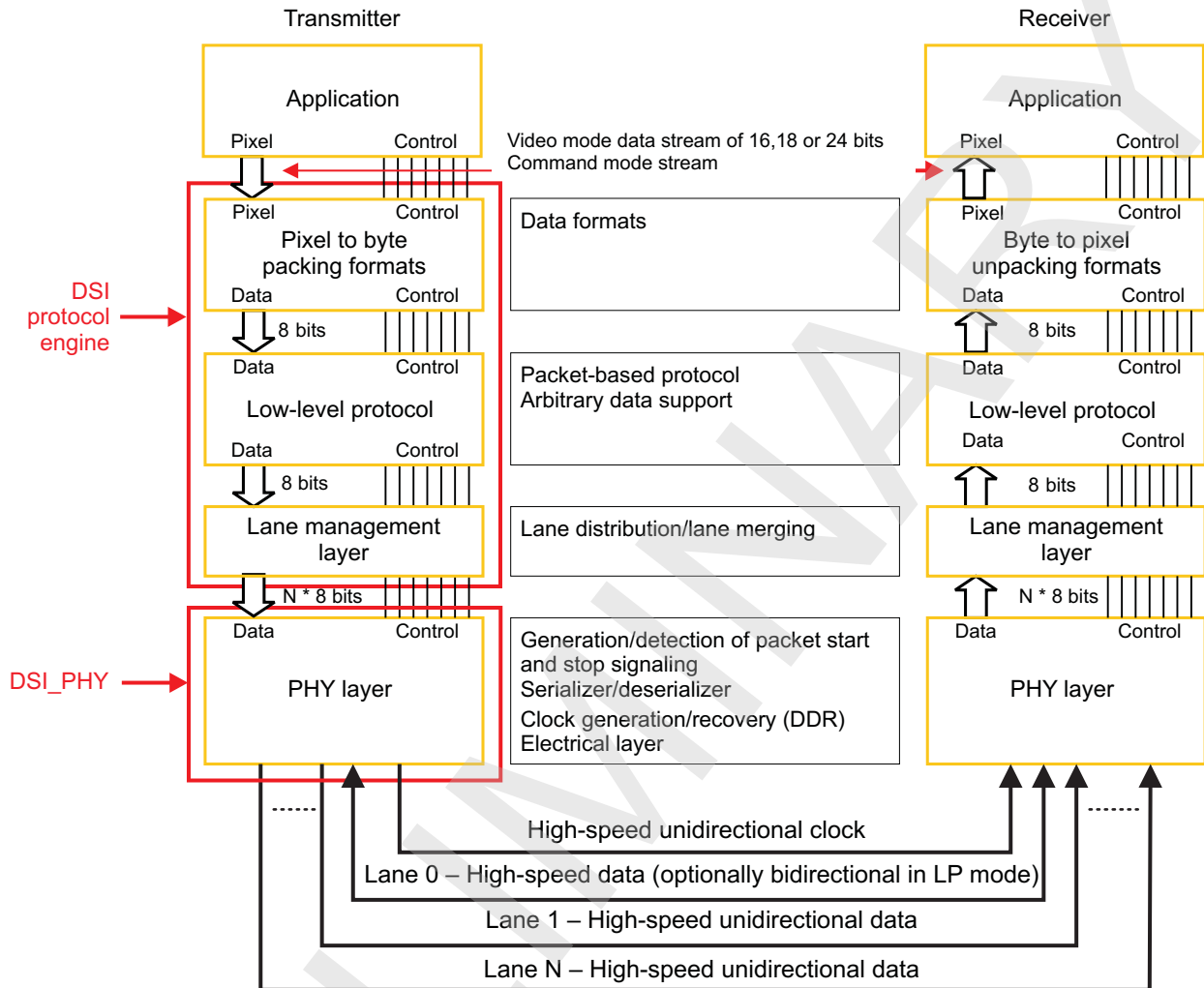


dsi-002

The DSI is a bidirectional differential serial interface with data/clock for the PHY (configured in unidirectional link in case the display module is only unidirectional). The maximum capacity of a DSI data transfer is 824 Mbps for four data lanes. The speed of the link can be configured by software only when the DSI\_PHY is in STOP state or ULTRALOW-POWER state (ULPS).

Figure 10-96 shows the high-level data flow of the DSI transmitter/receiver.

Figure 10-96. DSI Transmitter/Receiver Data Flow



dsi-032

10.3.2.1 DSI PHY

Table 10-460 lists the DSI1 I/Os.

Table 10-460. I/O Description of DSI1

Signal Name		I/O <sup>(1)</sup>	Description	Value at Reset
dsi_dx0	line 1	O	Serial data/clock output	N/A
dsi_dy0				
dsi_dx1	line 2	O	Serial data/clock output	N/A
dsi_dy1				
dsi_dx2	line 3	O	Serial data/clock output	N/A
dsi_dy2				
dsi_dx3	line 4	O	Serial data/clock output	N/A
dsi_dy3				
dsi_dx4	line 5	O	Serial data/clock output	N/A
dsi_dy4				
dsi_te0	te0	I	DSI1 tearing effect (TE) input 0	N/A

<sup>(1)</sup> I = Input; O = Output

**Table 10-460. I/O Description of DSI1 (continued)**

Signal Name		I/O <sup>(1)</sup>	Description	Value at Reset
dsi_te1	te1	I	DSI1 TE input 1	

**NOTE:** Each serial line (line 1 through line 5) can be used as a clock lane or a data lane. All polarities are supported. The MIPI DSI 1.01 protocol requires at least one clock line and one data lane.

Table 10-461 lists the DS2 I/Os.

**Table 10-461. I/O Description of DSI2**

Signal Name		I/O <sup>(1)</sup>	Description	Value at Reset
dsi_dx0	line 1	O	Serial data/clock output	N/A
dsi_dy0				
dsi_dx1	line 2	O	Serial data/clock output	N/A
dsi_dy1				
dsi_dx2	line 3	O	Serial data/clock output	N/A
dsi_dy2				
dsi_te0	te0	I	DSI2 TE input 0	N/A
dsi_te1	te1	I	DSI2 TE input 1	

<sup>(1)</sup> I = Input; O = Output

**NOTE:** Each serial line (line 1 through line 3) can be used as a clock lane or a data lane. All polarities are supported. The MIPI DSI 1.01 protocol requires at least one clock line and one data lane.

Lanes support four operating modes:

- HS mode: High-speed transmit mode
- LP mode: Low-power transmit mode (also called low-power state [LPS])
- ULPS: ULTRALOW-POWER state used between two transmissions
- Off mode: Lane is off.

### 10.3.2.1.1 Data/Clock Configuration

Data-clock signaling consists of one to four data pairs and one clock pair. The minimum configuration is one data pair and one clock pair.

- The data signal carries the bit serial data. The DSI transmitter in the host sends the data in-quadrature with the dual data rate (DDR) clock in HS mode; otherwise, the clock is extracted from the received data in LS mode. The data is transmitted byte-wise LSB first.
- The clock signal carries the DDR clock signal in HS transmission.
- Software users must configure the order of the data lanes to indicate the byte order while splitting the byte stream for each DSI\_PHY into bytes.

Table 10-462 lists some of the DSI lane configurations.

**NOTE:** All combinations for the order of the clock and data lanes are supported but not explicitly described in Table 10-462.

**Table 10-462. DSI Lane Configuration**

DSI_PHY Lane Configuration	Data/Clock Lane Position					Description
	1	2	3	4	5	
Mode CLK + DATA1						Single data lane
	CLK	DATA1				
	DATA1	CLK				
Mode CLK + DATA1 + DATA2						Two data lanes
	CLK	DATA1	DATA2			
	CLK	DATA2	DATA1			
	DATA1	CLK	DATA2			
	DATA2	CLK	DATA1			
	DATA1	DATA2	CLK			
	DATA2	DATA1	CLK			
Mode CLK + DATA1 + DATA2 + DATA3						Three data lanes
	CLK	DATA1	DATA2	DATA3		
	CLK	DATA3	DATA2	DATA1		
	DATA1	CLK	DATA2	DATA3		
	DATA1	CLK	DATA3	DATA2		
	DATA2	CLK	DATA1	DATA3		
	DATA2	CLK	DATA3	DATA1		
	DATA3	CLK	DATA1	DATA2		
	DATA3	CLK	DATA2	DATA1		
	DATA3	DATA1	CLK	DATA2		
	DATA3	DATA2	CLK	DATA1		
	DATA3	DATA2	CLK	DATA1		
	DATA1	DATA2	CLK	DATA3		
	DATA2	DATA3	CLK	DATA1		
	DATA1	DATA3	CLK	DATA2		
	DATA1	DATA2	DATA3	CLK		
	DATA3	DATA2	DATA1	CLK		
Mode CLK + DATA1 + DATA2 + DATA3 + DATA4						Four data lanes
	CLK	DATA1	DATA2	DATA3	DATA4	
	DATA4	DATA3	DATA2	DATA1	CLK	
	DATA3	DATA1	CLK	DATA2	DATA4	
	DATA4	DATA2	CLK	DATA1	DATA3	
	DATA2	DATA1	CLK	DATA3	DATA4	
	DATA4	DATA3	CLK	DATA1	DATA2	
	DATA3	DATA4	CLK	DATA2	DATA1	
	DATA1	DATA2	CLK	DATA4	DATA3	
	DATA1	DATA4	CLK	DATA3	DATA2	
	DATA1	DATA3	CLK	DATA4	DATA2	

**NOTE:**

- The byte on Dn is sent before the byte on Dn + 1; all the combinations of data and clock are supported through programming of the **DSI\_COMPLEXIO\_CFG1** register. The **CLOCK\_POSITION** and **CLOCK\_POL** bit fields configure which lane transmits the clock and define its polarity. Eight bit fields (**DATA1\_POSITION** and **DATA1\_POL**, through **DATA4\_POSITION** and **DATA4\_POL**) configure the data lanes and their polarity. The **DATA2\_POSITION** through **DATA4\_POSITION** bit fields can be set to 0; in this case, only the data lane defined in the **DATA1\_POSITION** bit field is used, and data is transmitted on only one clock lane and one data lane.
- Only **DATA1** is bidirectional in command mode. The low-power received information is always sent by the display panel using **DATA1**. Because any lane of the DSI complex I/O can be configured as data lane **DATA1**, all lanes of the complex I/O are bidirectional.

**10.3.2.1.2 ULPS**

Each lane can be put in ULPS by software configuration. The ULPS mode requires the following conditions:

- The lane must be in ULP state.
- For data lanes, no data must be pending in the DSI module.
- For data lane 1, no bus turnaround (BTA) should have been sent. The DSI module must have control of the bus.

For more information on ULPS configuration, see [Section 10.3.4.4.16](#), *Ultralow-Power State Configuration*.

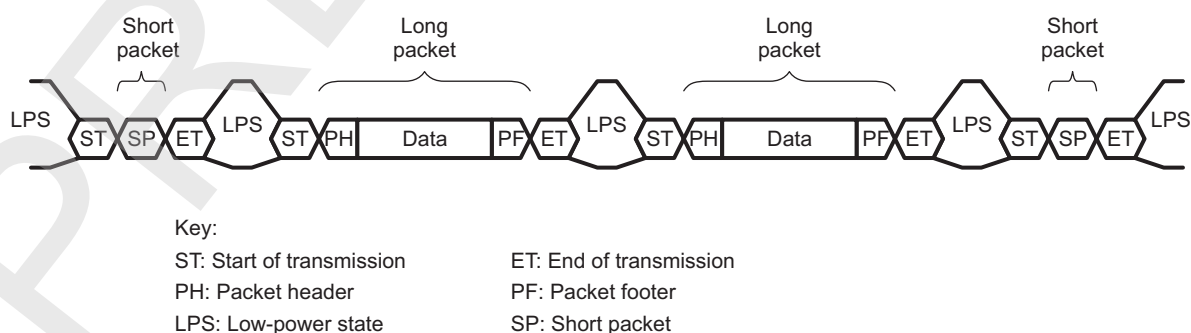
**10.3.2.2 DSI Protocol Layer**

Low-level protocol (LLP) is a byte-oriented protocol. It supports short and long packet formats. The DSI protocol layer defines how the display data is transported onto the PHY. Packets can be sent using HS mode or low-speed (LS) mode. LLP is selected through DSI registers. The features of the DSI protocol layer are:

- Transport of arbitrary data (payload independent)
- 8-bit word size
- Support for up to four interleaved VCs on the same link
- Special packets for frame start, frame end, line start, and line end information
- Descriptor for the type, pixel depth, and format of application-specific payload data
- ECC for 1-bit or 2-bit error detection in the header
- 16-bit checksum code for error detection

[Figure 10-97](#) shows the protocol layer with short and long packets.

**Figure 10-97. Protocol Layer With Short and Long Packets**



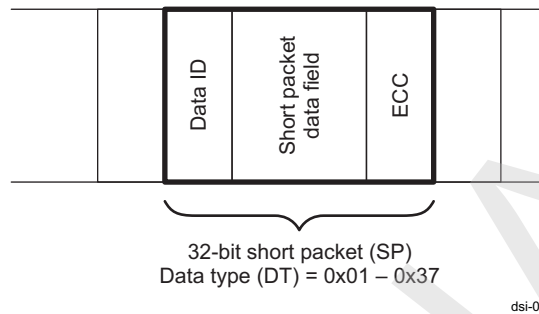
dsi-004



### 10.3.2.2.1 Short Packet

Figure 10-98 shows the structure of the short packet. A short packet must contain an 8-bit data ID followed by two command or data bytes and an 8-bit ECC. No packet footer (PF) should be present. Short packets must be 4 bytes long. The ECC byte allows correction of single-bit errors and detection of 2-bit errors in the short packet.

Figure 10-98. Short Packet Structure



**NOTE:** The short packets can be sent in LP mode or HS mode.

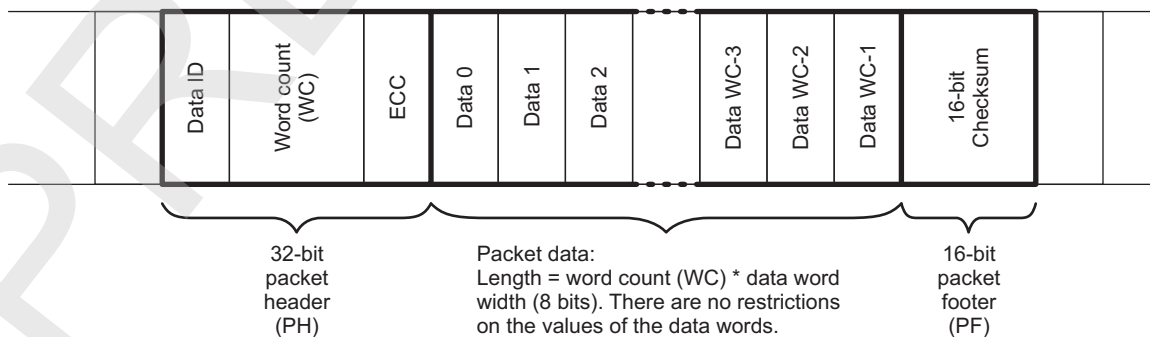
The `DSI_VC_SHORT_PACKET_HEADER_i` register is used to send only short packets (ECC can be calculated by hardware or by software users for debug purposes). This register is not used for video mode data because the short packets are generated by hardware based on the following events:

- Synchronization events received on the video port (assertion/deassertion of the HSYNC and VSYNC input signals)
- `DSI_CTRL[18]` VP\_HSYNC\_END
- `DSI_CTRL[17]` VP\_HSYNC\_START
- `DSI_CTRL[16]` VP\_VSYNC\_END
- `DSI_CTRL[15]` VP\_VSYNC\_START
- `DSI_CTRL[11]` VP\_VSYNC\_POL
- `DSI_CTRL[10]` VP\_HSYNC\_POL
- `DSI_VC_CTRL_i[1]` SOURCE

### 10.3.2.2.2 Long Packet

Figure 10-99 shows the structure of the long packet. A long packet must consist of three elements: 32-bit packet header (PH), application-specific data payload with a variable number of bytes, and 16-bit PF.

Figure 10-99. Long Packet Structure



The PH is further composed as follows:

- 8-bit data identifier: The data identifier defines the VC for the data and the DT for the application-specific payload data.

- 16-bit word count: The word count defines the number of bytes in the data payload between the end of the PH and the start of the PF. The PH and the PF must not be included in the word count.
- 8-bit ECC: The ECC byte allows single-bit errors to be corrected and 2-bit errors to be detected in the PH. This includes the data identifier and the word count fields.

After the end of the PH, the receiver reads the next word count  $\times$  bytes of the data payload. There are no limitations on the value of a data word within the data payload block (that is, no embedded codes are used). Once the receiver has read the data payload, it reads the checksum in the PF. The host processor must always calculate and transmit a checksum in the PF. Peripherals are not required to calculate a checksum. In the special case of a 0-byte data payload, if the payload length is 0, the checksum calculation results in (0xFFFF). If the checksum is not calculated, the PF must consist of 2 bytes of 0s (0x0000). In the generic case, the length of the data payload must be a multiple of bytes. In addition, each data format can impose additional restrictions on the length of the payload data (that is, a multiple of 4 bytes). Each byte is transmitted LSB first. Payload data can be transmitted in any byte order, restricted only by data format requirements. Multibyte elements, such as word count and checksum, must be transmitted least-significant byte (LSByte) first.

---

**NOTE:** Long packets can be sent in LP or HS mode.

---

The [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register provides headers for long packets (ECC is always calculated by hardware). The register is used for video and command modes.

If video mode is enabled for a virtual channel, it is not possible to transfer data concurrently (interleaved in a frame) using long packets received on the video and L3 interconnect ports because the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register is used by the video mode. The register can be unprogrammed by the user to send long packets received on the L3 interconnect port only when there is no expected data on the video port. Software must program the register correctly to send sequentially long packets in video and command modes.

The word count (WC) defined in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register for the VC associated with the video port indicates the number of bytes to receive (one line or two lines can be used, depending on the WC and the size of the line buffer). The total size defined in the WC of the header register cannot exceed the size of the line buffer multiplied by the number of buffer lines.

The [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register is used to provide payload data for long packets (checksum is calculated by hardware when [DSI\\_VC\\_CTRL\\_i\[7\]](#) CS\_TX\_EN is set to 1; otherwise, the value is 0x00). The register is not used in video mode because payload data are provided by the video port.

Software must ensure that the following sequence for write accesses to the header and payload registers ([DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) and [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#), respectively) is followed:

- A long PH value with WC = 0 written in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register can be followed by any access.
- A long PH value with WC > 0 written in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register must be followed by one or more writes to the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register defined by the WC value before writing again to the same [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register.

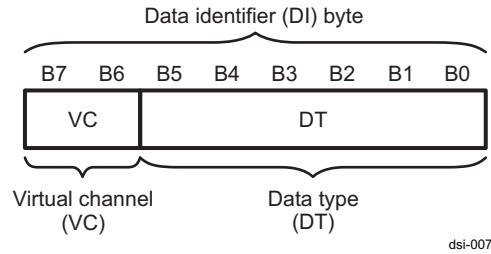
#### **CAUTION**

If this sequence is not followed, no error is generated. Access to other DSI registers during this sequence is allowed.

### 10.3.2.2.3 Data Identifier

The data identifier byte contains the values for the virtual channel ID and data type, as shown in [Figure 10-100](#). The virtual channel ID is contained in the 2 MSBs of the data identifier byte and identify the data as directed to one of four virtual channels. The value of the data type is contained in the 6 LSBs of the data identifier byte.

**Figure 10-100. Data Identifier Structure**

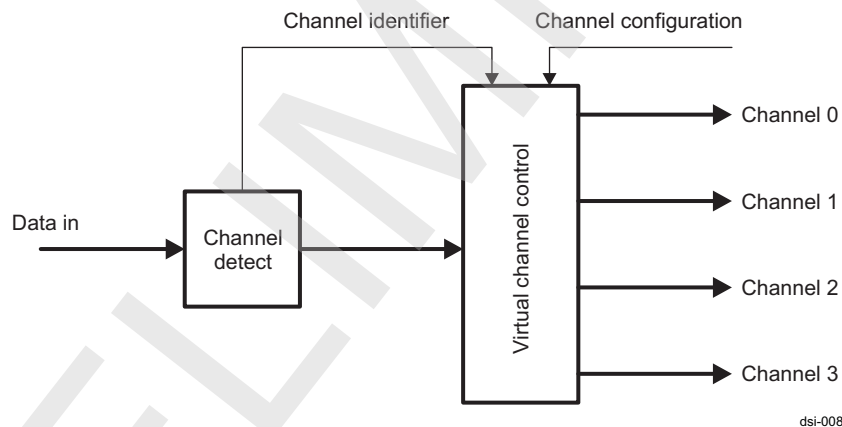


**10.3.2.2.3.1 VC ID – VC Field, DI[7:6]**

The host can service up to four peripherals with tagged commands or blocks of data using the VC ID field of the header for packets targeted at different peripherals. The VC ID enables one serial stream to service two or more virtual peripherals by multiplexing packets onto a common transmission channel. Each packet sent in a single transmission has its own VC assignment and can be directed to different peripherals. The VC ID is defined in the [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) and [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) registers for short and long packets, respectively. It will not be modified by hardware. There is one set of registers for each VC. Each set of registers defines the characteristics of the traffic between the host and the display associated with the VC.

Figure 10-101 shows the VC controller.

**Figure 10-101. VC Controller**



**10.3.2.2.3.2 Data Type Field DT[5:0]**

The data type field specifies whether the packet is long or short and the packet format. The data type field, and the word count field for long packets, informs the receiver about how many bytes to expect in the rest of the packet. This is necessary because the beginning and end of a packet are not indicated by special packet start/end sync codes. This permits packets to convey arbitrary data, but it also requires the PH to specify the size of the packet.

**10.3.2.2.4 Synchronization Codes**

Each frame can be identified by two synchronization codes:

- For the start of the vertical synchronization (VSSC) pulse
- For the end of the vertical synchronization (VSEC) pulse

Each line can be identified by two synchronization codes:

- For the start of the horizontal synchronization (HSSC) pulse
- For the end of the horizontal synchronization (HSEC) pulse

The synchronization events are not be required by the display (peripheral): they are optional. Users can

program which synchronization events are generated to the display from the timings received from the DISPC in video mode. When data are received on the L3 interconnect port, the synchronization codes are not automatically generated by the protocol engine. They can be provided on the L3 interconnect port by writing to the registers with limited timing control. It is highly recommended to use the video port from the DISPC to receive the synchronization events to automatically generate short synchronization packets to the peripheral.

When the DSI protocol engine detects that the VSYNC signal from the DISPC transitions from inactive to active state, the VSSC short packet replaces the following HSSC corresponding to the following HSYNC synchronization short packet (if the generation is enabled).

When the transition from active to inactive state is detected, the VSEC short packet is generated (if the generation is enabled) replacing the HSEC synchronization packet corresponding to the following HSYNC.

When the DSI protocol engine detects that the HSYNC signal from the DISPC transitions from inactive to active state, the HSSC short packet is generated (if the generation is enabled).

When the transition from active to inactive state is detected, the HSEC short packet is generated (if generation is enabled).

For the first frame, any HSYNC and data received on the video port before the first VSYNC must be ignored. Because the first VSYNC sent to the display is also recognized as an HSYNC for the first line, there is no HSYNC sent for the first line. To send the synchronization codes, the DSI protocol engine uses the short packets. [Table 10-463](#) lists the 6-bit DT synchronization code values.

**Table 10-463. Synchronization Codes**

Synchronization Code	Value	Comments
Vertical sync start code (VSSC)	0x1	Optional
Vertical sync end code (VSEC)	0x11	Optional
Horizontal sync start code (HSSC)	0x21	Optional
Horizontal sync end code (HSEC)	0x31	Optional

### 10.3.2.2.5 Blanking

To keep the DSI link in HS state while using video mode, during blanking periods, long blanking packets are sent to the display. The DSI\_VM\_TIMINGx (where x = 1 to 7) registers define the size of the long blanking packets after:

- Horizontal sync start code (short packet)
- Horizontal sync end code (short packet)
- Vertical sync start code (short packet)
- Vertical sync end code (short packet)
- Pixels (long packet)

[Table 10-464](#) defines the short packet values for the synchronization packets:

**Table 10-464. Sync Short Packet Values**

VC ID	Sync Code	Header (First Byte)	Header (Second Byte): Data Field LSB	Header (Third Byte): Data Field MSB	Header (ECC)
0x0	0x1	0x1	0x0	0x0	See Notes following this table.
	0x11	0x11			
	0x21	0x21			
	0x31	0x31			
0x1	0x1	0x41			
	0x11	0x51			

**Table 10-464. Sync Short Packet Values (continued)**

VC ID	Sync Code	Header (First Byte)	Header (Second Byte): Data Field LSB	Header (Third Byte): Data Field MSB	Header (ECC)
0x2	0x21	0x61			
	0x31	0x91			
	0x1	0x81			
	0x11	0x81			
	0x21	0xA1			
0x3	0x31	0xB1			
	0x1	0xC1			
	0x11	0xD1			
	0x21	0xC1			
	0x31	0xF1			

**NOTE:**

- If the ECC is enabled by setting the [DSI\\_VC\\_CTRL\\_i\[8\] ECC\\_TX\\_EN](#) bit to 1 for the VC in video mode, the ECC value is calculated; otherwise, 0x00 is used for the blanking long packets and sync short packets. If the CRC is enabled by setting the [DSI\\_VC\\_CTRL\\_i\[7\] CS\\_TX\\_EN](#) bit to 1 for the VC in video mode, the checksum value is calculated; otherwise, 0x00 is used for the blanking long packets.
- In other cases, when the [DSI\\_VC\\_CTRL\\_i\[7\] CS\\_TX\\_EN](#) bit is set to 0, the value 0x00 is always used for the CRC (long packets). When the [DSI\\_VC\\_CTRL\\_i\[8\] ECC\\_TX\\_EN](#) bit is set to 0, the value 0x00 is used for the ECC for short and long packets, except when the header is provided by the register, because the ECC field is available in the register. It can be used to generate invalid ECC values when the header is provided by the register.

The DSI link (lanes and clock separately) can be put in ULPS. While using the blanking values formerly defined, the packets (short and long) are considered in HS mode.

Timing parameters VSA, VBP, VFP, HSA, HBP, HFP, VACT, and  $t_l$  are defined in the [DSI\\_VM\\_TIMING1](#) through [DSI\\_VM\\_TIMING8](#) registers. HSA, HBP, HFP, and  $t_l$  are defined using the byte clock unit (TXBYTECLKHS - HS transmit byte clock generated by DSI\_PHY) and also in low-power clock cycles (TxClkEsc). VSA, VBP, VFP, and VACT are defined by the number of lines. When the HS blanking packets are sent during the blanking periods, the parameters are used to determine the blanking packet payload size (taking into account the 4-byte header and the 2-byte checksum).

The configuration of the DISPC timing generator must be used when the DISPC timings are used to generate DSI HS video mode transfer.

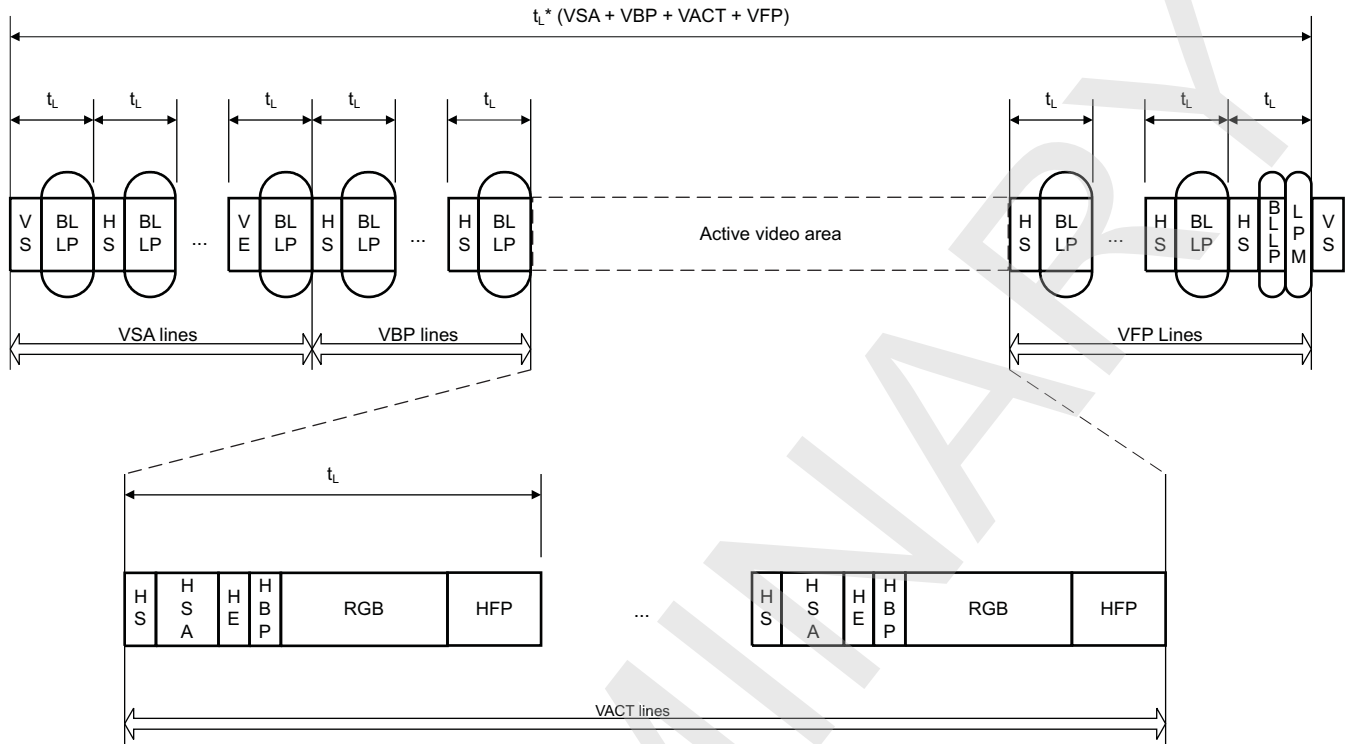
Special care must be taken in the case of the last line of the frame. The LPS transition is required when the link is in HS mode for the whole frame.

When BTA is sent for the data packets, the following blanking period cannot be used for sending any data from the TX FIFO. When the blanking period starts with one HS packet from one VC, it can be followed only by another HS packet from the same VC, or by a trigger (for example, BTA). When there is no more HS data to send for this VC, the lane is in LPS.

When the blanking period starts with one LP packet from one VC, it can be followed only by another LP packet from the same VC, by another VC, by trigger (BTA, for example), or by extra LP NULL packets. If the trigger is sent, it is not possible to send any more data. When there is no more data from the TX FIFO to send in LP mode or the trigger has been sent, the lane is put into LPS. If the lanes must be kept in HS mode during blanking periods (except for the last blanking period of the frame), the HS blanking packets must be used. If one trigger is sent at the beginning of the blanking period, the rest of the blanking period is in LPS.

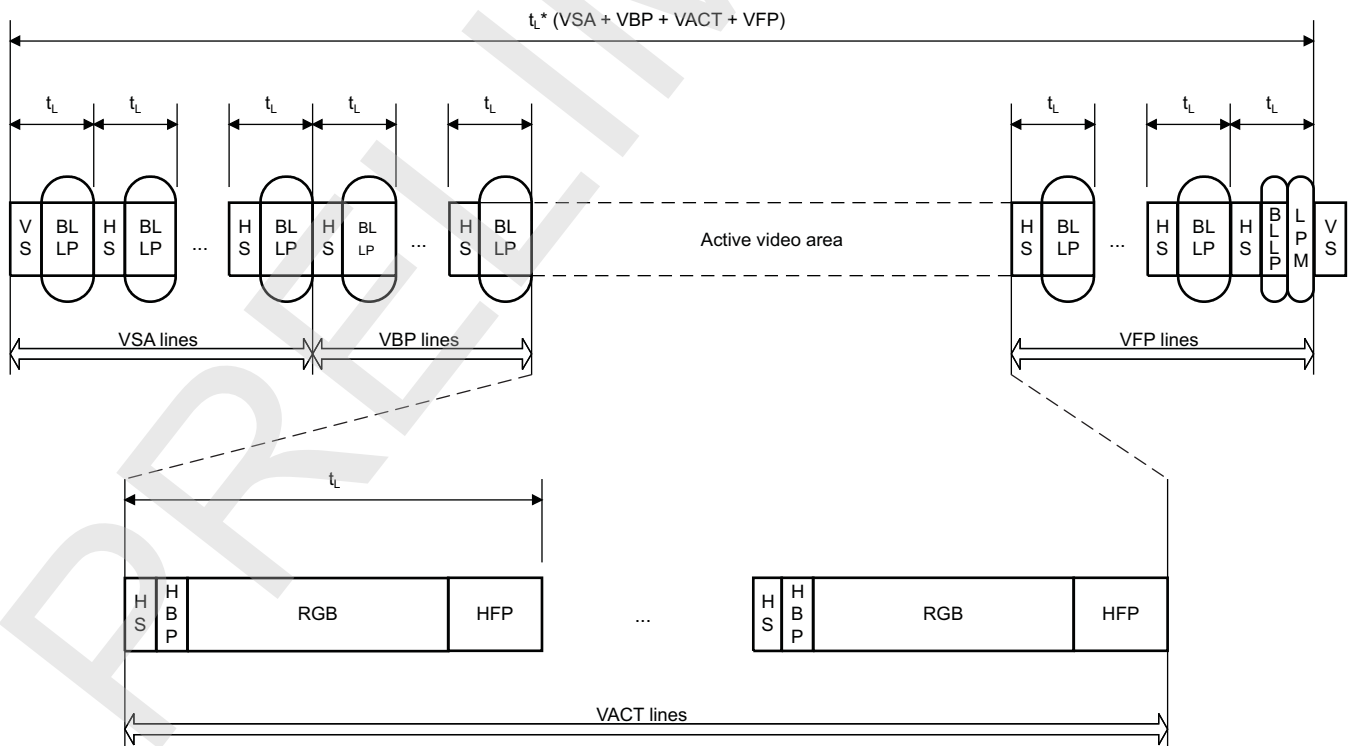
[Figure 10-102](#) and [Figure 10-103](#) show a nonburst transfer in DSI video mode with and without VE and HE, respectively. [Figure 10-104](#) shows a burst transfer in DSI video mode without VE and HE.

**Figure 10-102. DSI Video Mode: Nonburst Transfer With VE and HE**



dsi-009

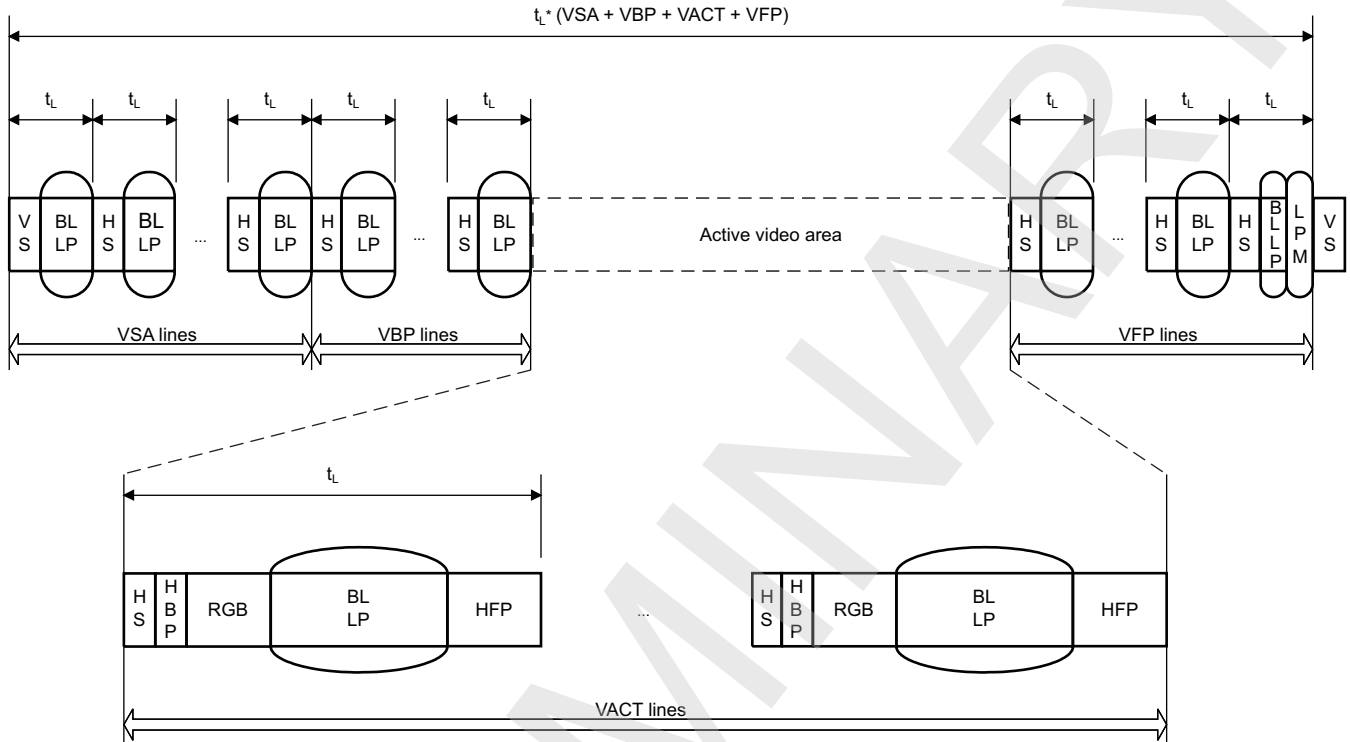
**Figure 10-103. DSI Video Mode: Nonburst Transfer Without VE and HE**



dss-010

**NOTE:** HSA timing is not used and does not have to be programmed when HE short packet is not generated.

**Figure 10-104. DSI Video Mode: Burst Transfer Without VE and HE**



dsi-011

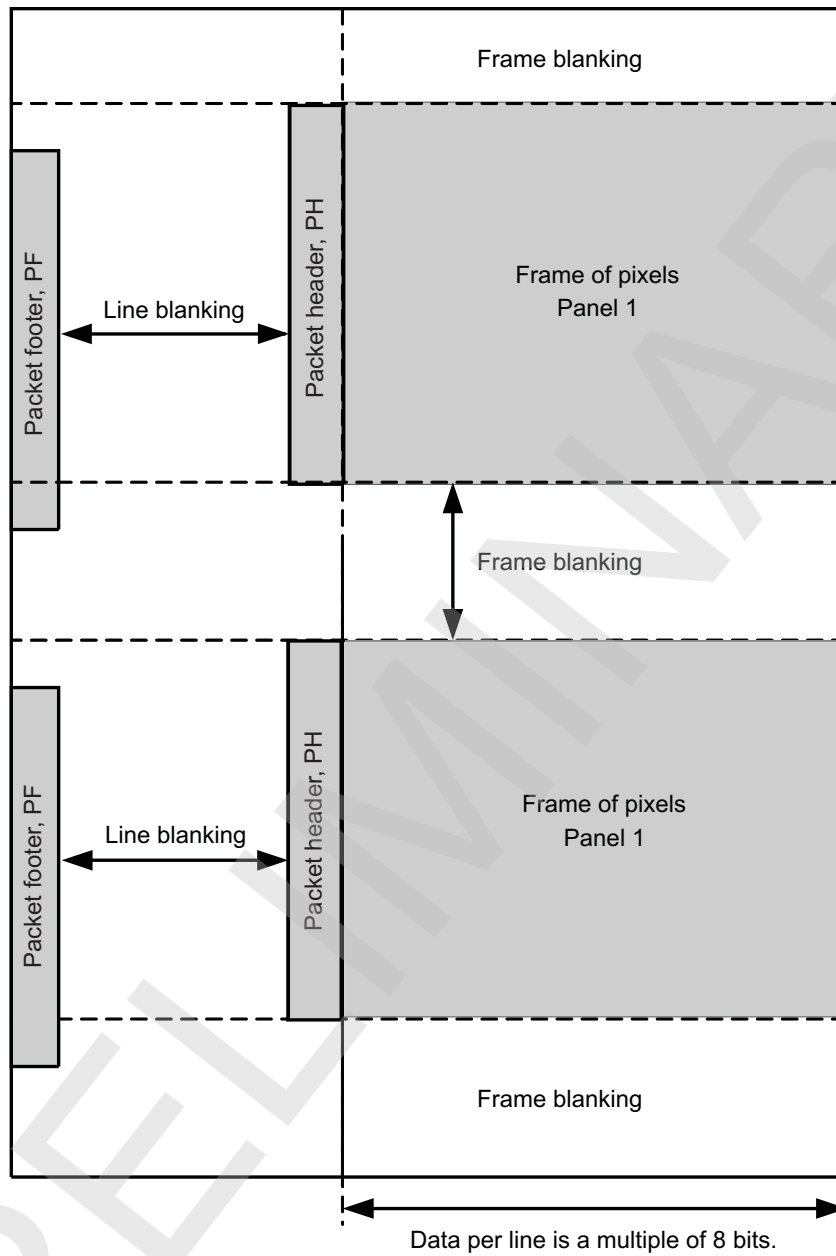
**NOTE:** HSA timing is not used and does not have to be programmed when HE short packet is not generated.

In [Figure 10-103](#) and [Figure 10-104](#), if HSYNC end short packet is not generated (HSA does not exist), HBP must not be 0.

### 10.3.2.2.6 Frame Structures

[Figure 10-105](#) shows the DSI general frame structure.

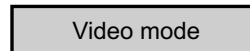
Figure 10-105. DSI General Frame Structure



KEY:

PH: Packet header

PF: Packet footer

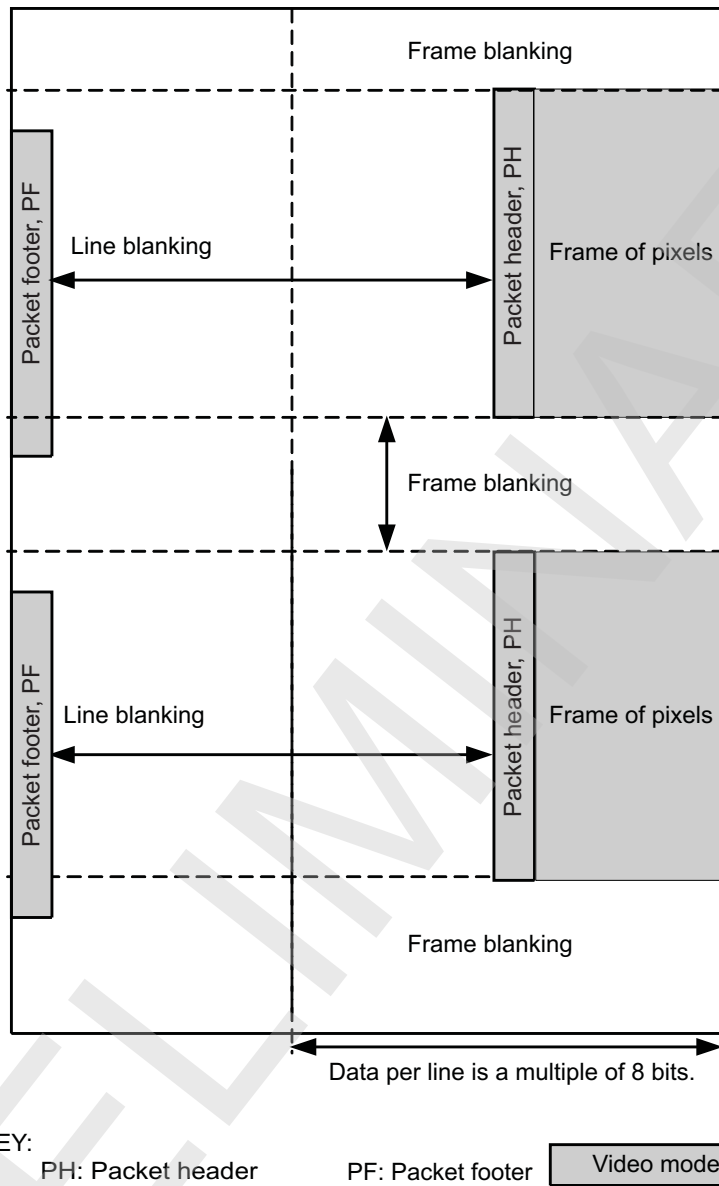


dsi-012

Figure 10-106 shows the general frame structure using burst mode.



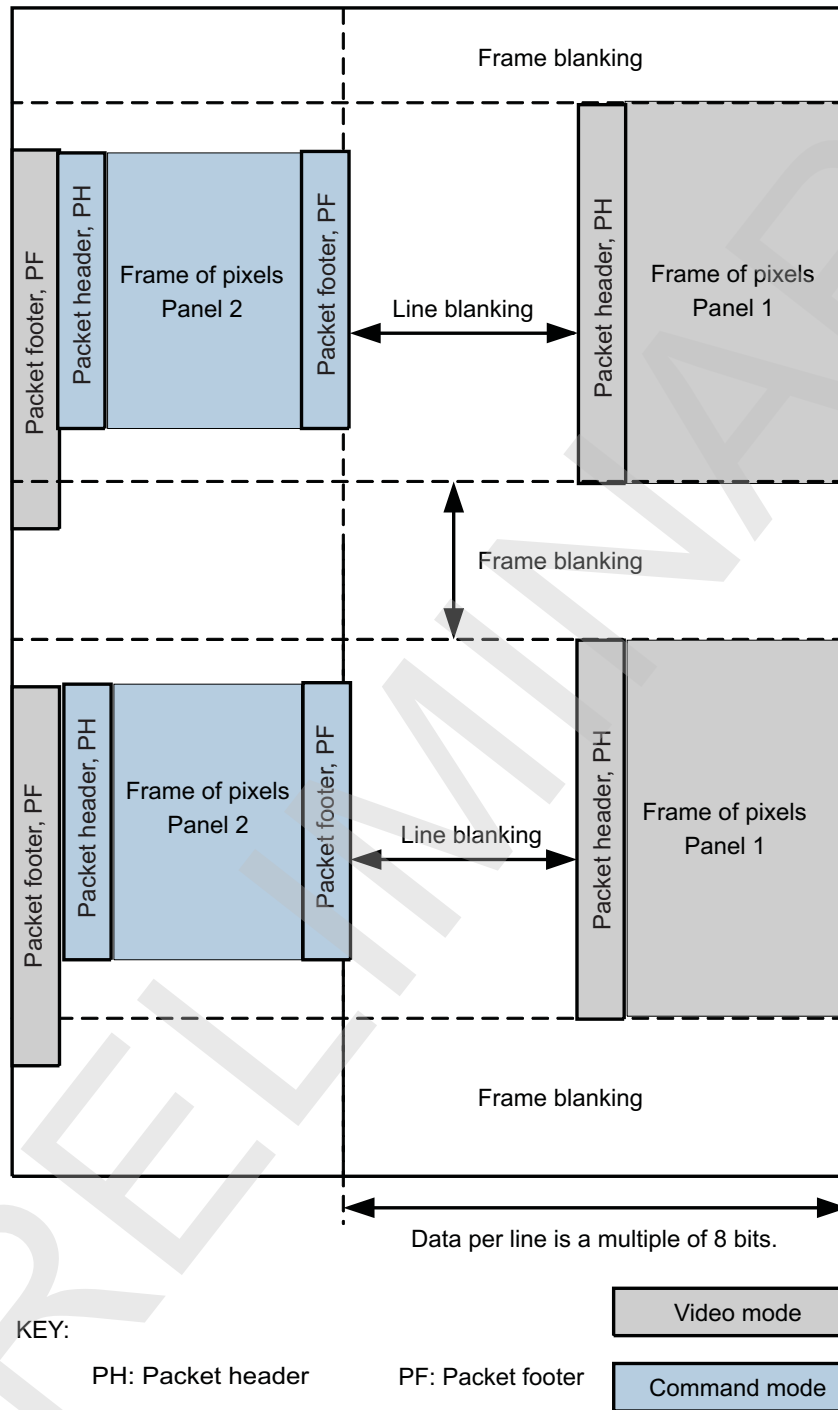
Figure 10-106. DSI General Frame Structure Using Burst Mode



dsi-013

Figure 10-107 shows the general frame structure using burst mode and interleaving.

**Figure 10-107. DSI General Frame Structure Using Burst Mode and Interleaving**



dsi-014

### 10.3.2.2.7 VCs

The DSI protocol layer transports VCs. VCs separate different data flows, which are interleaved in the same data stream. Each VC is identified by a unique channel identification number in the PH. The channel identification number is encoded in 2 bits. The DSI protocol engine determines the channel identifier number to be used for generating the PH and multiplexes the interleaved data streams. The DSI protocol engine supports multiple (up to 4) concurrent VCs. [Table 10-465](#) summarizes the VC values used for each channel.

**Table 10-465. VC Values**

VC Number	Value
VC 0	0x0
VC 1	0x1
VC 2	0x2
VC 3	0x3

If multiple displays are connected to a single DSI port on the host, a hub can be used to route the data stream to the appropriate display based on the VC ID. Typically, VC ID 0x0 is used for the primary display and 0x1 for the secondary. The hub may have its own VC ID to provide communication capability between the host and the hub.

There is one set of registers for each VC. The attributes of the VC define the following characteristics:

- Transfer mode ([DSI\\_VC\\_CTRL\\_i\[4\]](#) MODE bit):
  - Video mode
  - Command mode
- Data type
- Source ([DSI\\_VC\\_CTRL\\_i\[1\]](#) SOURCE bit)
  - Video port
  - Slave interconnect port
- HS or LP forward transmission
- Automatic BTA generation
  - Short packets ([DSI\\_VC\\_CTRL\\_i\[2\]](#) BTA\_SHORT\_EN bit)
  - Long packets ([DSI\\_VC\\_CTRL\\_i\[3\]](#) BTA\_LONG\_EN bit)
- DMA request configurations for RX and TX
  - DMA request number ([DSI\\_VC\\_CTRL\\_i\[29:27\]](#) DMA\_RX\_REQ\_NB bit field for RX FIFO and [DSI\\_VC\\_CTRL\\_i\[23:21\]](#) DMA\_TX\_REQ\_NB bit field for TX FIFO)
  - DMA threshold ([DSI\\_VC\\_CTRL\\_i\[26:24\]](#) DMA\_RX\_THRESHOLD bit field for RX FIFO and [DSI\\_VC\\_CTRL\\_i\[19:17\]](#) DMA\_TX\_THRESHOLD bit field for TX FIFO)
- Mode speed ([DSI\\_VC\\_CTRL\\_i\[9\]](#) MODE\_SPEED bit)
- ECC transmission ([DSI\\_VC\\_CTRL\\_i\[8\]](#) ECC\_TX\_EN bit)
- CS transmission ([DSI\\_VC\\_CTRL\\_i\[7\]](#) CS\_TX\_EN bit)

---

**NOTE:** The VC ID is not calculated by the DSI module but is provided while writing into the [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) and [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) registers.

---

### 10.3.2.3 DSI Multilane Layer

A layer consists of lane splitter logic to split the incoming byte stream into a serial stream. The bits are sent with the LSB first. The number of active lanes is configurable through a register. The order of the lanes is configurable. The number of lanes can be changed only in ULPS or when all data lanes are in STOP state.

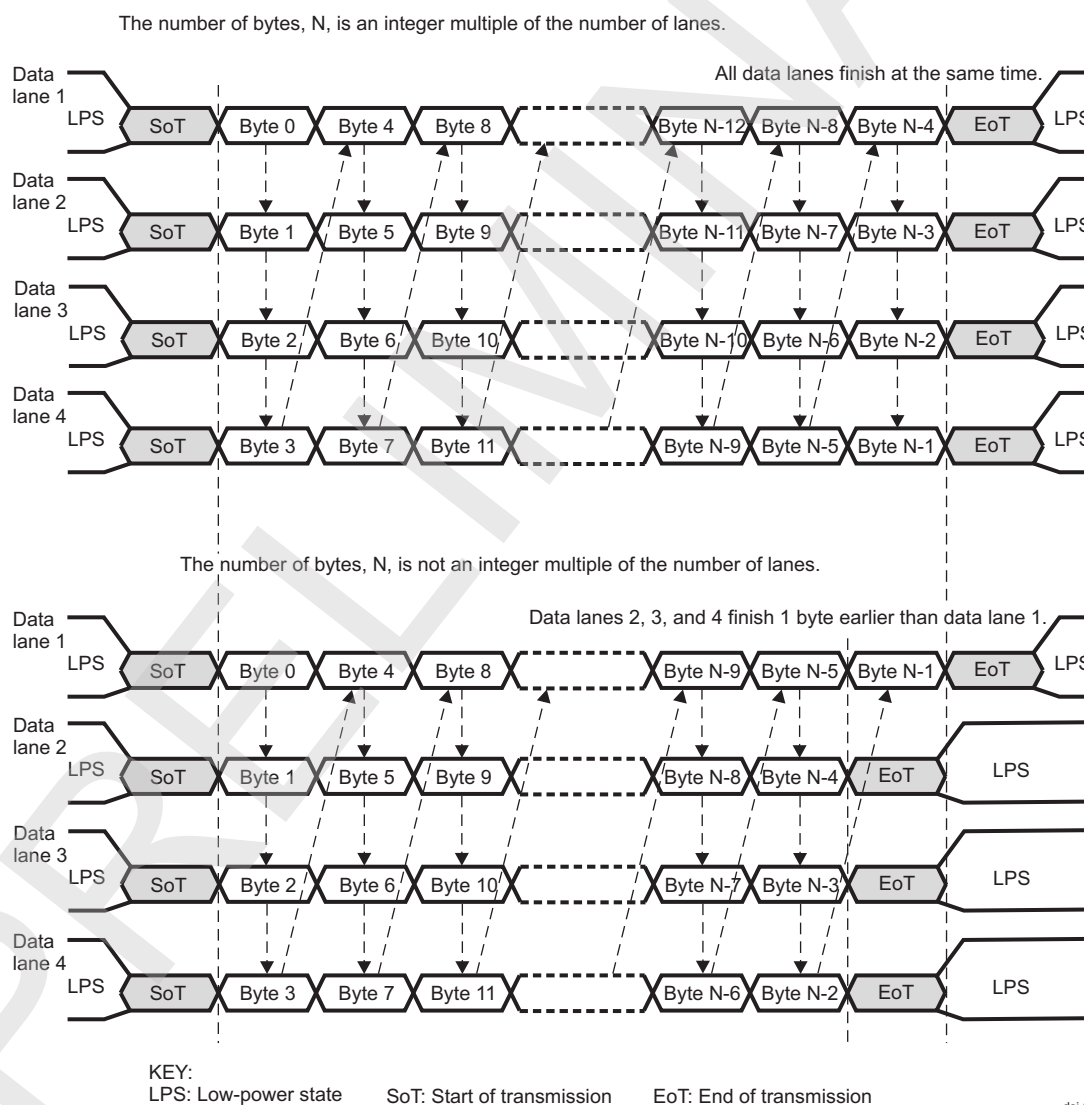
### 10.3.2.3.1 SoT and EoT in Multilane Configurations

Because a HS transmission is composed of an arbitrary number of bytes that may not be an integer multiple of the number of lanes, some lanes may run out of data before others. Therefore, the lane management layer, as it buffers up the final set of less-than-N bytes, deasserts its valid data signal into lanes for which there is no further data. Although all lanes start simultaneously with parallel start-of-transmissions (SoTs), each lane operates independently and may complete the HS transmission before the other lanes, sending an end-of-transmission (EoT) one cycle (byte) earlier.

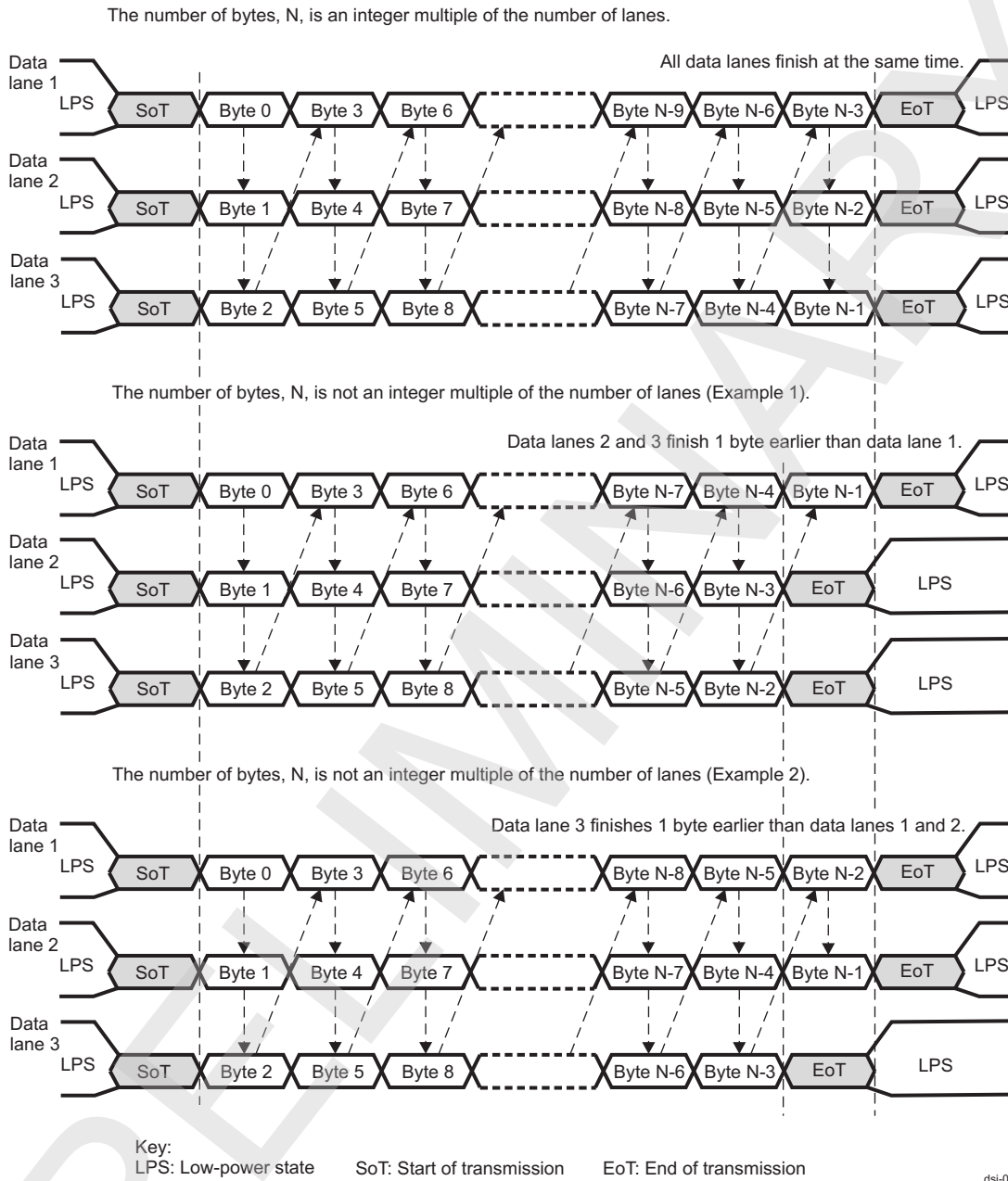
### 10.3.2.3.2 Lane Splitter

The lane splitter can split the byte stream into two, three, or four lanes (for one lane, the splitter is bypassed). Figure 10-108 through Figure 10-111 show the byte position in each serial link for one-, two-, three-, and four-data lane configurations. The byte stream always starts from lane 1. It finishes on one of the lanes, depending on the number of bytes to send and the number of lanes. The splitter module is used only when packets are sent using HS mode. In LS mode, only data lane 1 is used.

**Figure 10-108. Four-Data Lane Configuration**

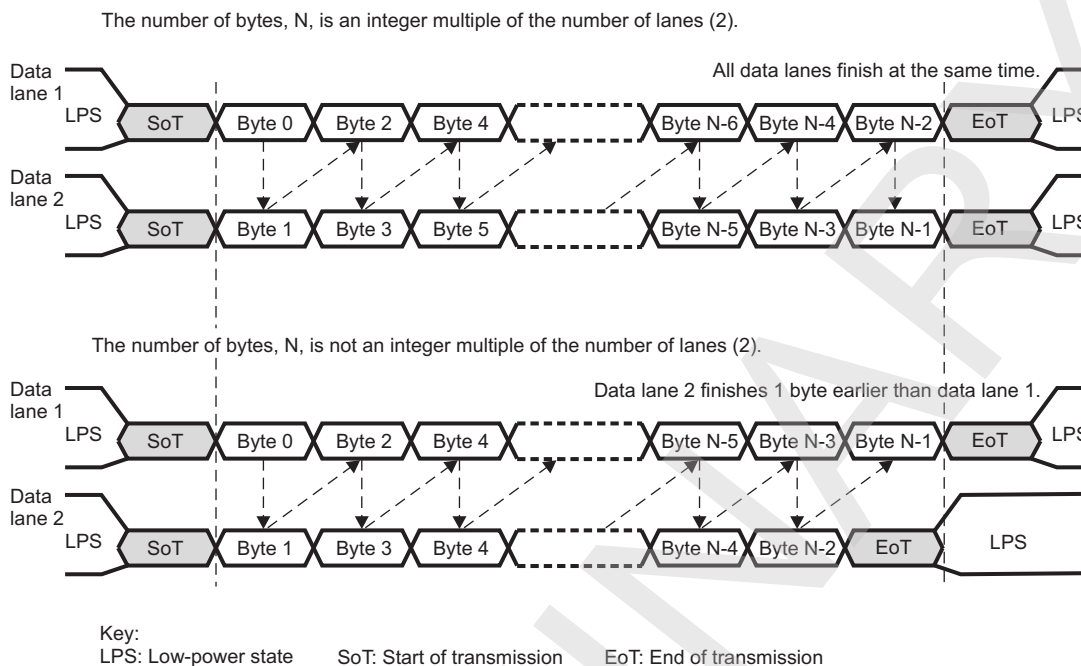


**Figure 10-109. Three-Data Lane Configuration**



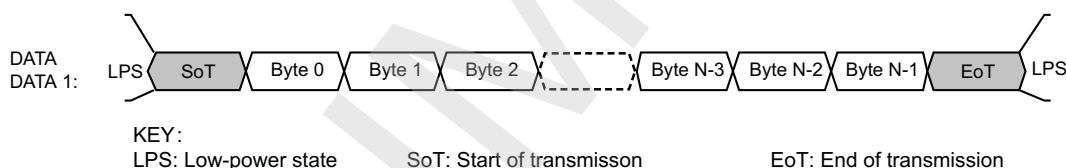
dsi-015

**Figure 10-110. Two-Data Lane Configuration**



dsi-016

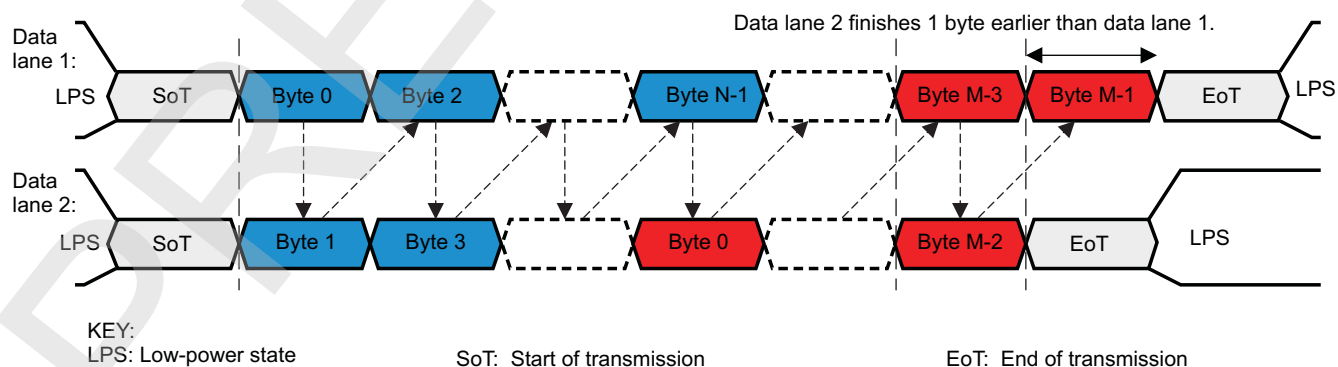
**Figure 10-111. One-Data Lane Configuration**



dsi-017

For back-to-back packets, the byte stream is considered as a single packet by the splitter module. [Figure 10-112](#) shows an example of two packets sent back-to-back. N bytes are used for the first packet and M bytes for the second packet.

**Figure 10-112. Two Packets Using Two-Data Lane Configuration (Example)**



dsi-019

### 10.3.2.4 DSI Pixel Data Formats

This section describes how the DSI-supported pixel data formats in video mode are transmitted over the serial interface. The DSI protocol engine can cope with all data formats if the data line length sent through the DSI physical protocol is a multiple of a pixel. This condition is required for the DSI protocol engine to work properly.

#### 10.3.2.4.1 Pixel Data Formats in Video Mode

The host can send different pixel formats in video mode. [Table 10-466](#) lists the pixel formats supported by the DSI in video mode.

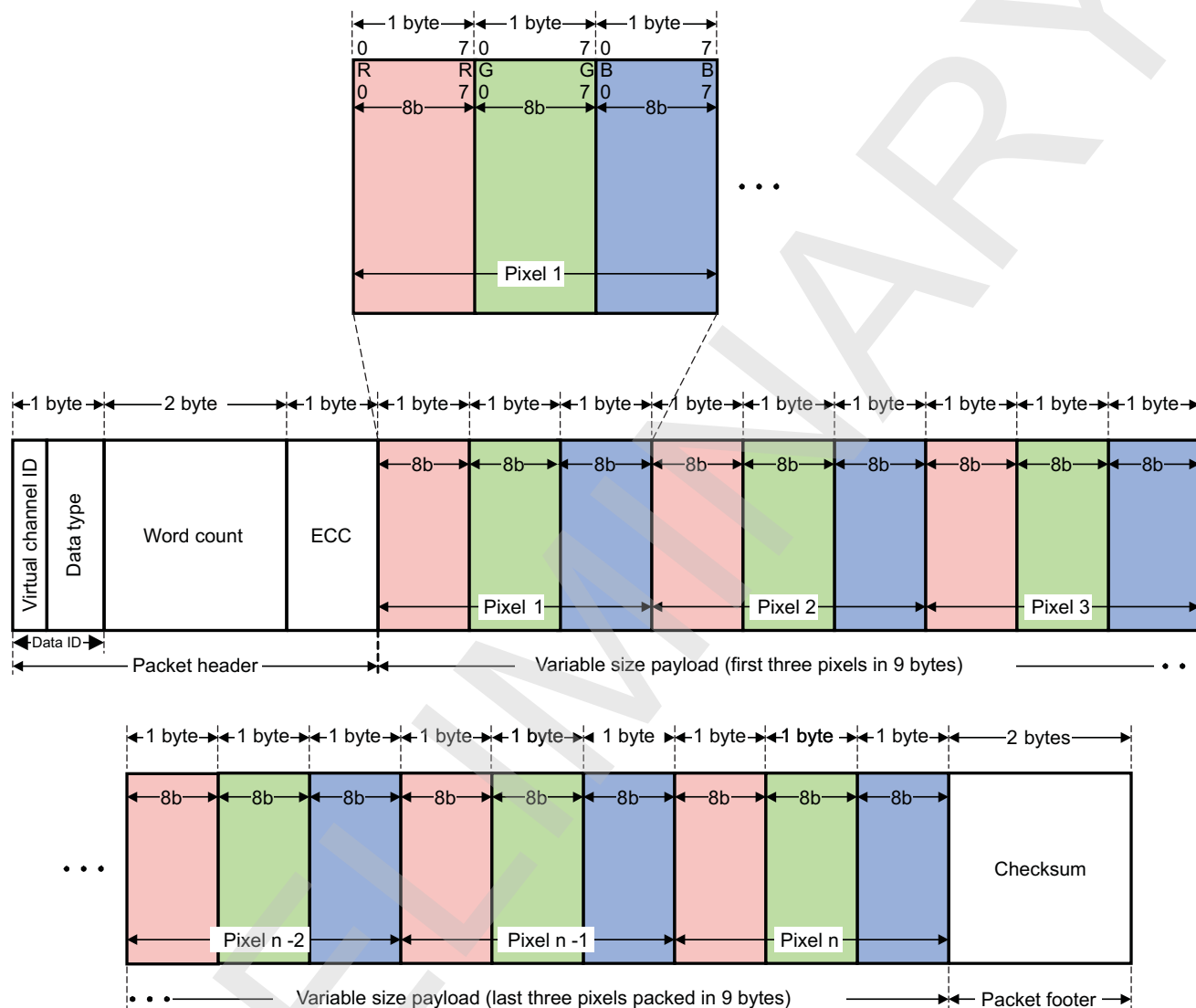
**Table 10-466. Pixel Data Format in Video Mode**

Mode	Description
RGB888 (using 24-bit container)	RGB888
RGB666 (using 24-bit container)	RGB666
RGB666 (18-bit packet using 18-bit container)	RGB666_PACKET
RGB565 (using 16-bit container)	RGB565

### 10.3.2.4.2 24 bpp – RGB Color Format, Long Packet (Video Mode)

Figure 10-113 shows the RGB888 format.

Figure 10-113. 24-bpp RGB Color Format, Long Packet (Video Mode)



ds1-020

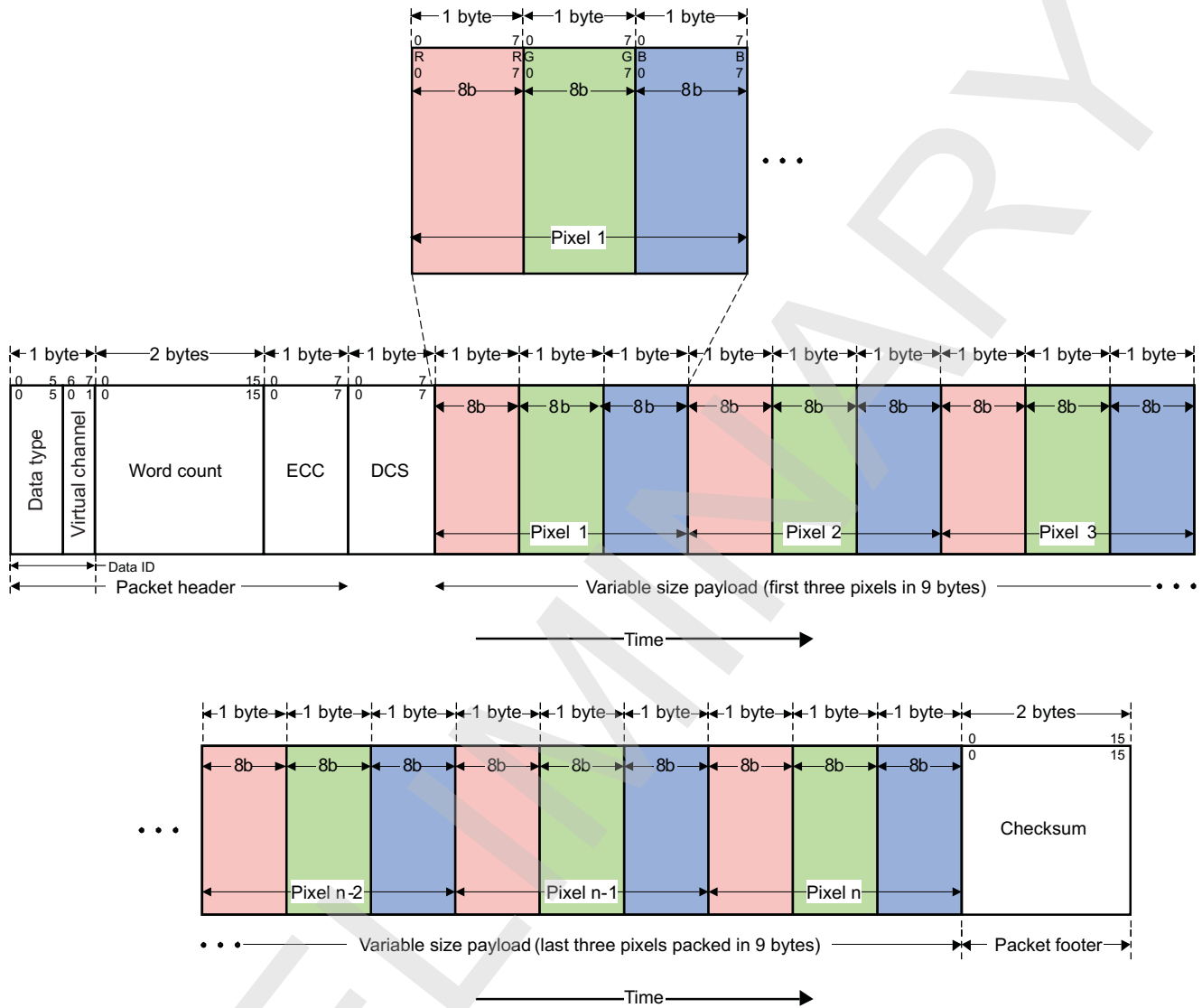
Packed-pixel stream, 24-bit format is a long packet used to transmit image data formatted as 24-bit pixels to a video-mode display module. The packet consists of the data ID byte, a 2-byte WC, an ECC byte, a payload of length WC bytes, and a 2-byte checksum. The pixel format is R (8 bits), G (8 bits), and B (8 bits), in that order. Each color component occupies 1 byte in the pixel stream; no components are split across byte boundaries. Within a color component, the LSB is sent first and the MSB last.

### 10.3.2.4.3 24 Bits per Pixel - RGB Color Format, Long Packet (Command Mode)

Figure 10-114 shows the RGB888 format.



Figure 10-114. 24-bpp RGB Color Format, Long Packet (Command Mode)



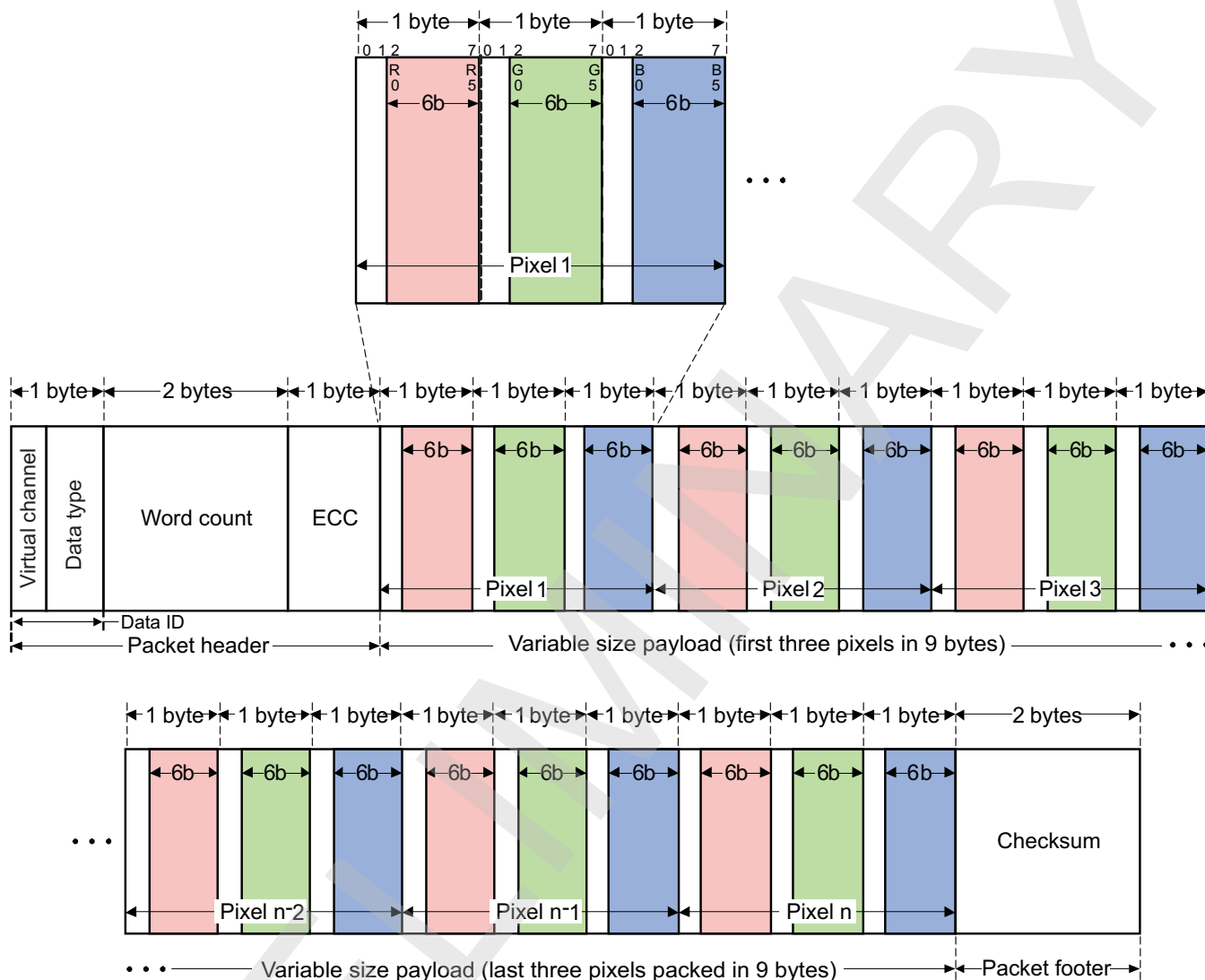
dsi-021

This format is the same as the packed pixel stream format.

**10.3.2.4.4 18 bpp (Loosely Packed) – RGB Color Format, Long Packet (Transparent for DSI Protocol Engine) (Video Mode)**

Figure 10-115 shows the RGB666 format.

**Figure 10-115. 18-bpp (Loosely Packed) RGB Color Format, Long Packet (Transparent for DSI Protocol Engine) (Video Mode)**



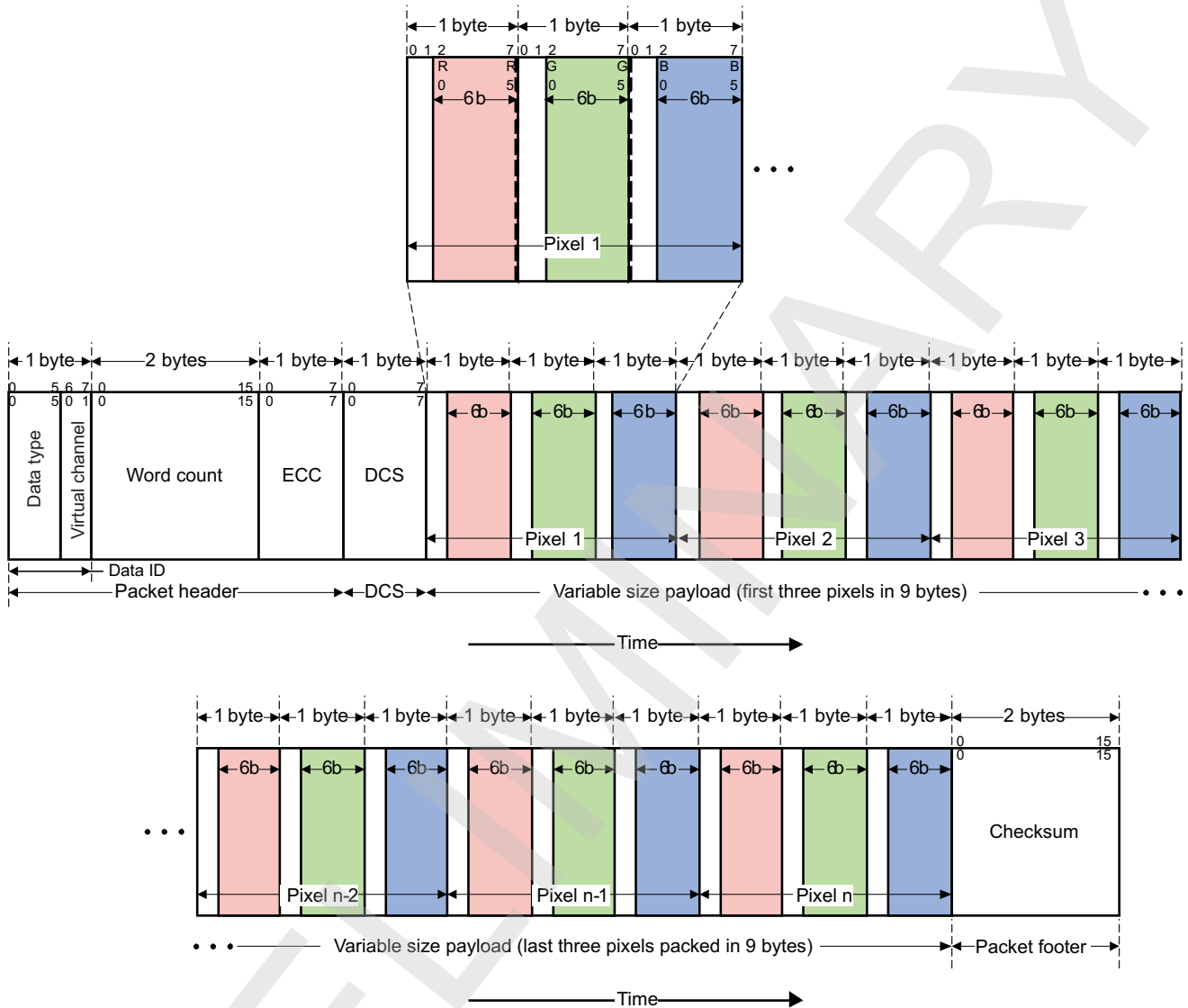
ds1-022

In 18-bit loosely packed pixel format, each R, G, and B color component is 6 bits but is shifted to the upper bits of the byte, such that the valid pixel bits occupy bits [7:2] of each byte. Bits [1:0] of each payload byte representing active pixels are ignored. As a result, each pixel requires 3 bytes as it is transmitted across the link. This requires more bandwidth than packed format but requires less shifting and multiplexing logic in the packing and unpacking functions on each end of the link. This format is used to transmit RGB image data formatted as pixels to a video mode display module that displays 18-bit pixels. The packet consists of the data ID byte, a 2-byte WC, an ECC byte, a payload of length WC bytes, and a 2-byte checksum. The pixel format is R (6 bits), G (6 bits), and B (6 bits), in that order. Within a color component, the LSB is sent first and the MSB last.

#### 10.3.2.4.5 18-bpp (Loosely Packed) – RGB Color Format, Long Packet (Command Mode)

Figure 10-116 shows the RGB666 format.

Figure 10-116. 18-bpp (Loosely Packed) RGB Color Format, Long Packet (Command Mode)



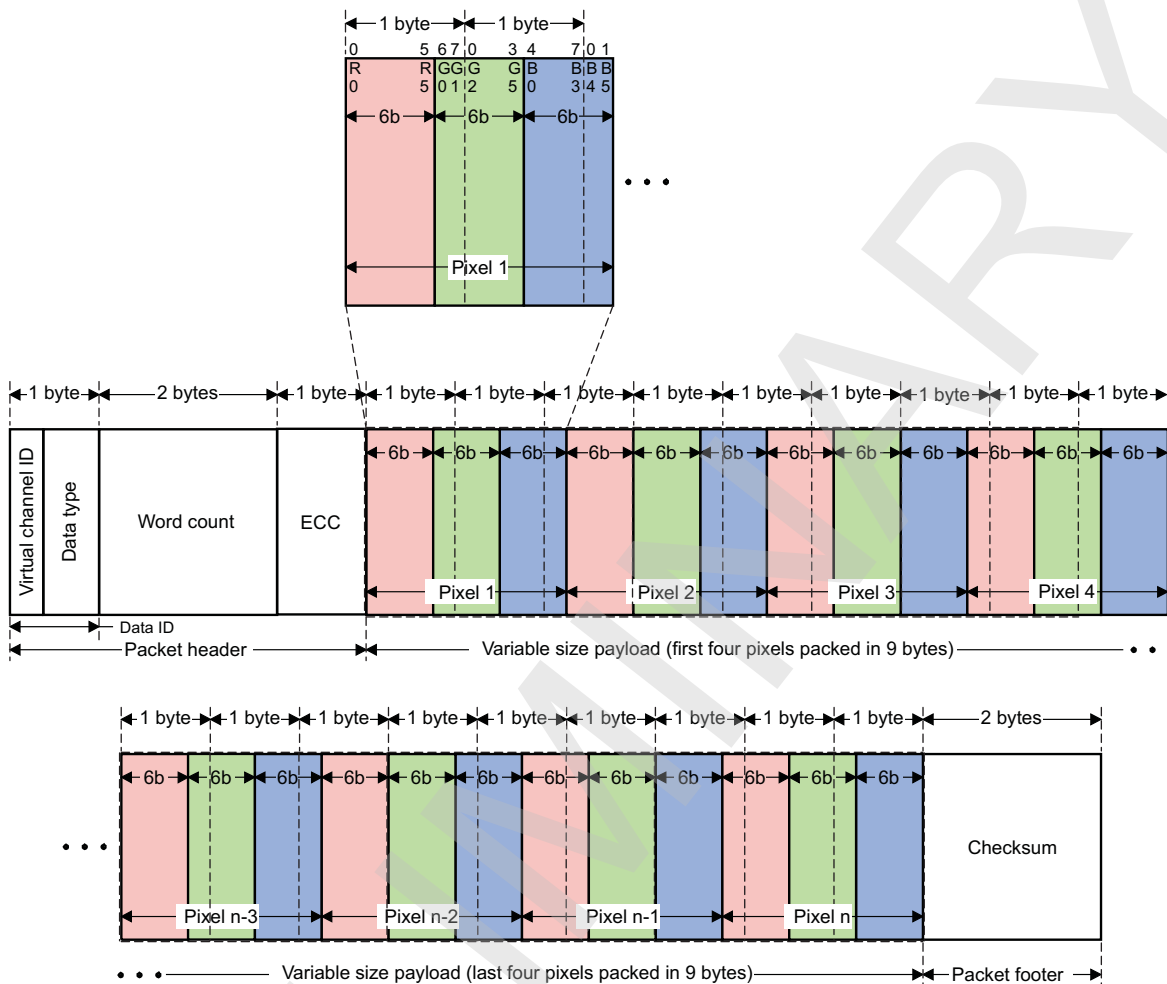
dsi-023

This format is the same as the packed-pixel stream format.

### 10.3.2.4.6 18-bpp (Packed) – RGB Color Format, Long Packet (Video Mode)

Figure 10-117 shows the RGB666\_PACKED format.

**Figure 10-117. 18-bpp (Packed) RGB Color Format, Long Packet (Video Mode)**



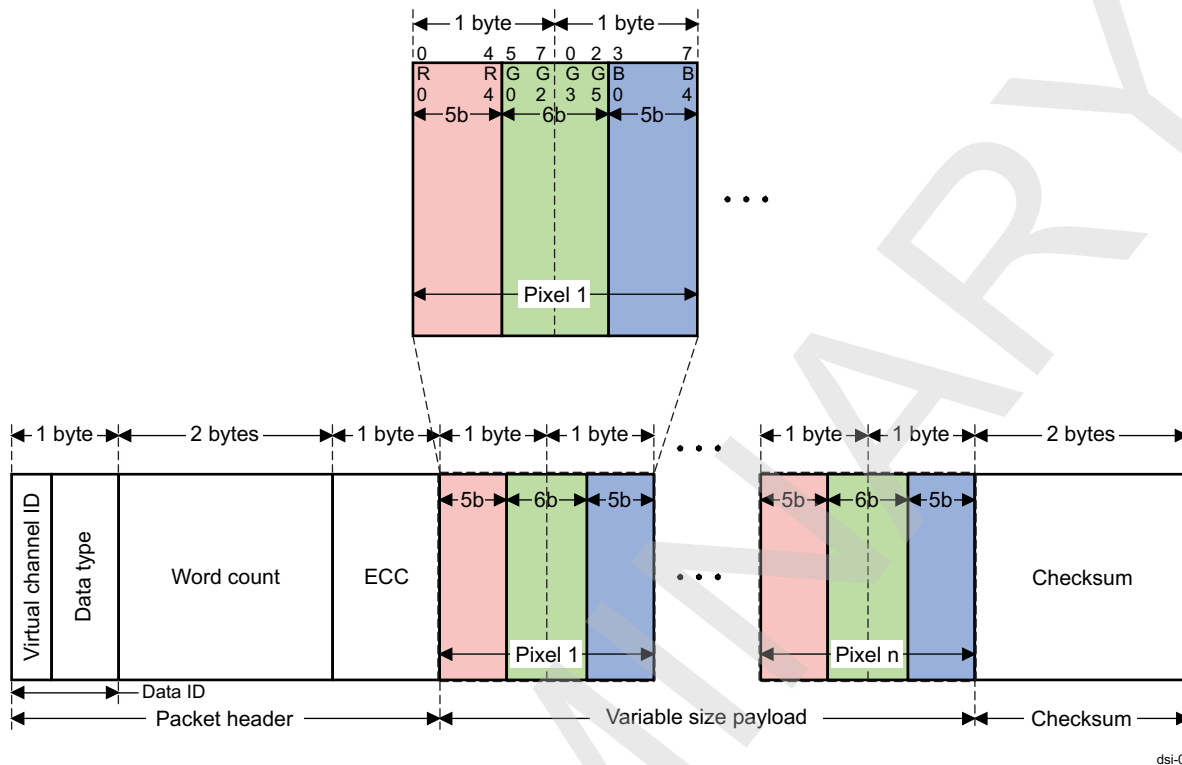
dsi-024

Packed pixel stream 18-bit format is a long packet. It is used to transmit RGB image data formatted as pixels to a video mode display module that displays 18-bit pixels. The packet consists of the data ID byte, a 2-byte WC, an ECC byte, a payload of length WC bytes, and a 2-byte checksum. The pixel format is R (6 bits), G (6 bits), and B (6 bits), in that order. Within a color component, the LSB is sent first and the MSB last. With this format, it is strongly recommended that the total line width be a multiple of 4 pixels (9 bytes). This format is not supported in command mode.

### 10.3.2.4.7 16 bpp – RGB Color Format, Long Packet (Video Mode)

Figure 10-118 shows the RGB565 format.

Figure 10-118. 16-bpp RGB Color Format, Long Packet (Video Mode)

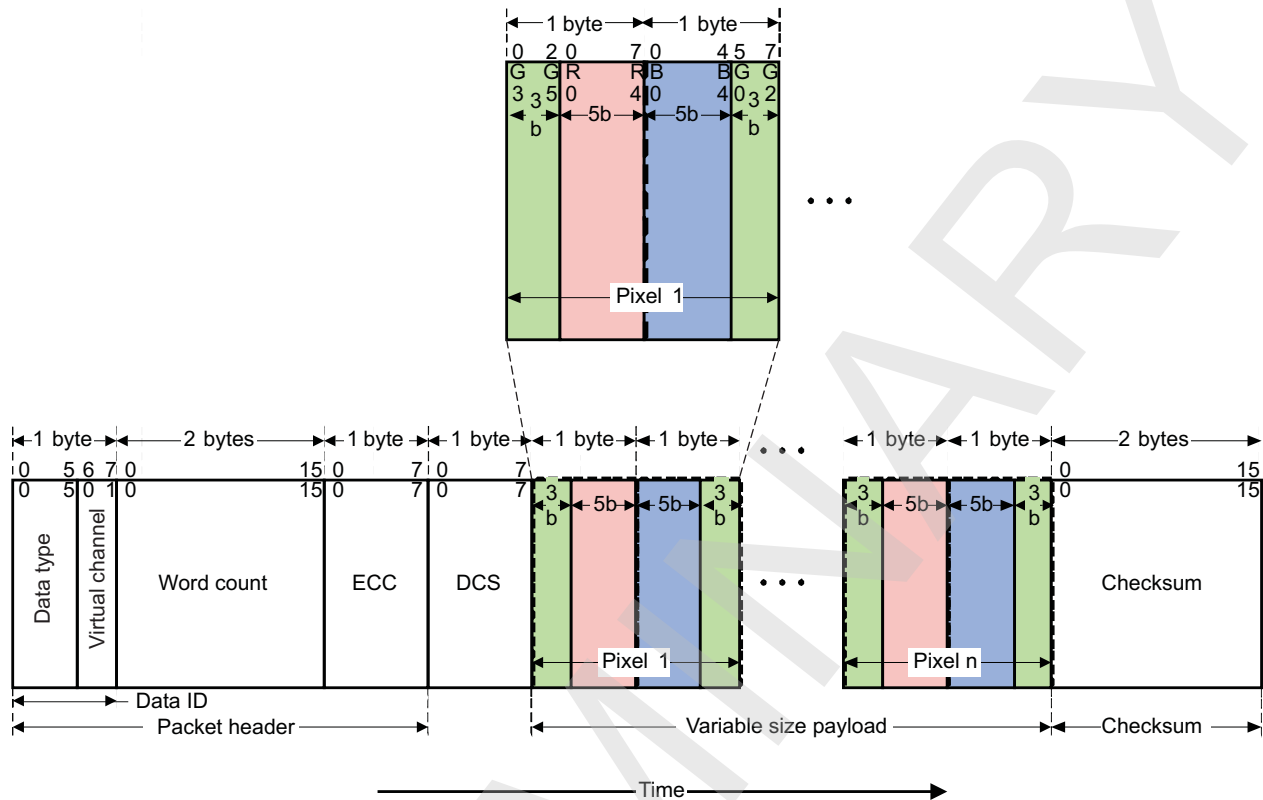


Packed-pixel stream, 16-bit format is a long packet used to transmit image data formatted as 16-bit pixels to a video mode display module. The packet consists of the data ID byte, a 2-byte WC, an ECC byte, a payload of length WC bytes, and a 2-byte checksum. The pixel format is R (5 bits), G (6 bits; split across 2 bytes), and B (5 bits), in that order. Within a color component, the LSB is sent first and the MSB last.

10.3.2.4.8 16 bpp – RGB Color Format, Long Packet (Command Mode)

Figure 10-119 shows the RGB565 format.

**Figure 10-119. 16-bpp RGB Color Format, Long Packet (Command Mode)**



dsi-026

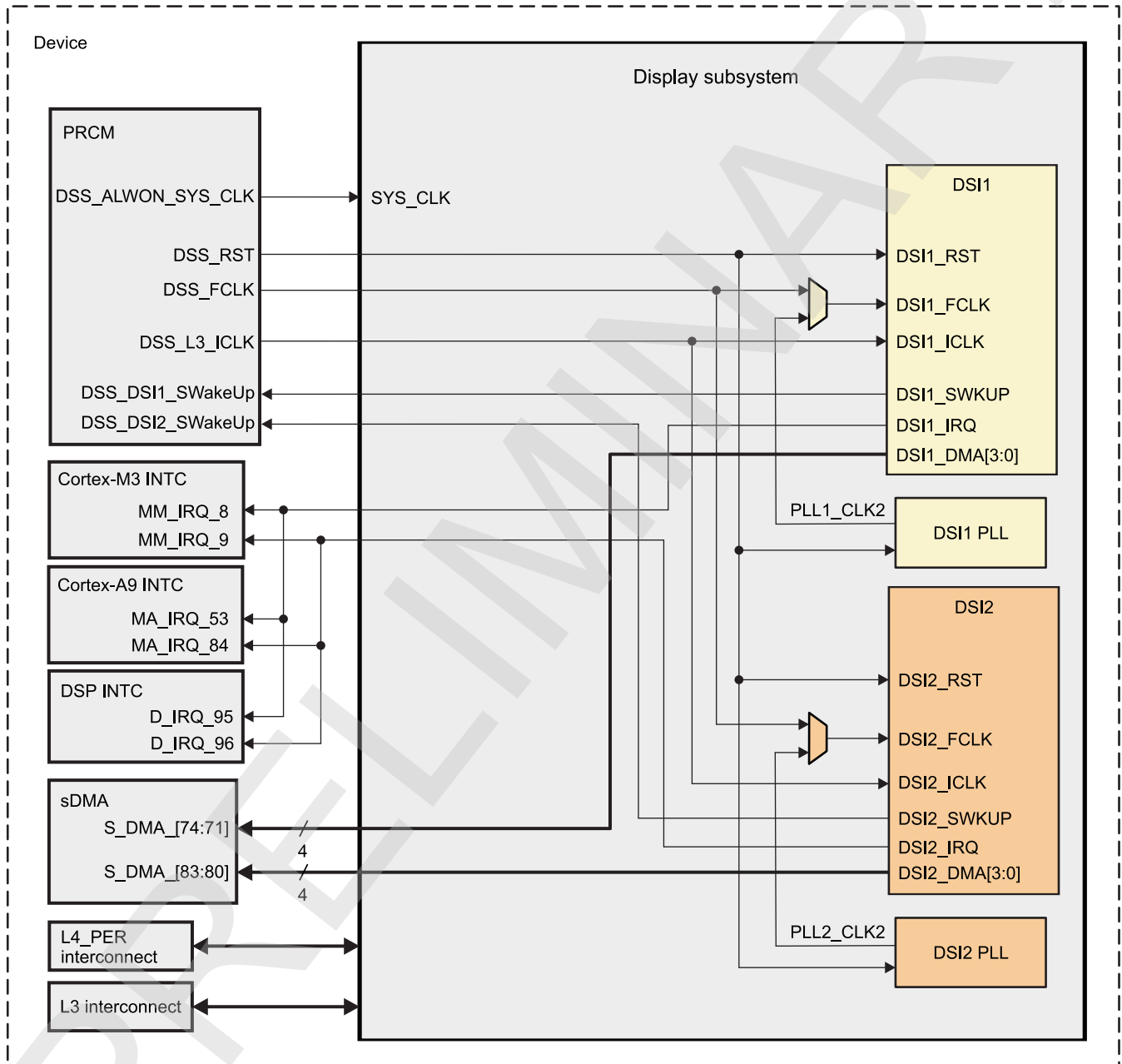
This is the opposite byte order from the packed pixel stream format. To retain flexibility, the byte order is made programmable using `DSI_VC_CTRL_i[12] RGB565_ORDER` bit, with this order the default for command mode through the video port.

### 10.3.3 DSI Integration

This section describes the integration of the DSI in the device, including information about clocks, resets, and hardware requests.

Figure 10-120 shows the integration of the DSI in the device.

Figure 10-120. DSI Integration



ds1-027

**NOTE:** For more information about the wake-up request, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

[Table 10-467](#) through [Table 10-470](#) summarize the integration of the DSI module in the device.

**Table 10-467. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
MIPI DSI1	PD_DSS	L3 for data L4_PER for configuration
MIPI DSI2	PD_DSS	L3 for data L4_PER for configuration

**NOTE:** L3 interconnect is used primarily for data.

L4\_PER interconnect is used primarily for configuration but is also used for backward software compatibility.

**Table 10-468. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
	SYS_CLK	DSS_ALWON_SYS_CLK	PRCM module	Functional clock for the DSI PLLs
MIPI DSI1	DSI1_FCLK	DSS_FCLK PLL1_CLK2	PRCM module DSI1 PLL	Functional clock
	DSI1_ICLK	DSS_L3_ICLK	PRCM module	Interface clock
MIPI DSI2	DSI2_FCLK	DSS_FCLK PLL2_CLK2	PRCM module DSI2 PLL	Functional clock
	DSI2_ICLK	DSS_L3_ICLK	PRCM module	Interface clock
Resets				
MIPI DSI1	DSI1_RST	DISPSS_RST	PRCM module	Nonretention reset
MIPI DSI2	DSI2_RST	DISPSS_RST	PRCM module	Nonretention reset

**NOTE:** For information about clock distribution and configuration inside the module, see [Section 10.3.4.2, DSI Clock Configuration](#).

**Table 10-469. Interrupt Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
MIPI DSI1	DSS_DSI1_IRQ	D_IRQ_95	DSP INTC	Display DSI1 interrupt request
	DSS_DSI1_IRQ	MA_IRQ_53	Cortex-A9 INTC	Display DSI1 interrupt request
	DSS_DSI1_IRQ	MM_IRQ_8	Cortex-M3 INTC	Display DSI1 interrupt request
MIPI DSI2	DSS_DSI2_IRQ	D_IRQ_96	DSP INTC	Display DSI2 interrupt request
	DSS_DSI2_IRQ	MA_IRQ_84	Cortex-A9 INTC	Display DSI2 interrupt request
	DSS_DSI2_IRQ	MM_IRQ_9	Cortex-M3 INTC	Display DSI2 interrupt request

**NOTE:** For a description of the interrupt source, see [Section 10.3.4.3, DSI Interrupt Requests](#).

**Table 10-470. DMA Requests**

DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
MIPI DSI1	DSS_DSI1_DMA0	S_DMA_71	sDMA	DSI1 DMA request 0
	DSS_DSI1_DMA1	S_DMA_72	sDMA	DSI1 DMA request 1



**Table 10-470. DMA Requests (continued)**

	DSS_DSI1_DMA2	S_DMA_73	sDMA	DSI1 DMA request 2
	DSS_DSI1_DMA3	S_DMA_74	sDMA	DSI1 DMA request 3
MIPI DSI2	DSS_DSI2_DMA0	S_DMA_80	sDMA	DSI2 DMA request 0
	DSS_DSI2_DMA1	S_DMA_81	sDMA	DSI2 DMA request 1
	DSS_DSI2_DMA2	S_DMA_82	sDMA	DSI2 DMA request 2
	DSS_DSI2_DMA3	S_DMA_83	sDMA	DSI2 DMA request 3

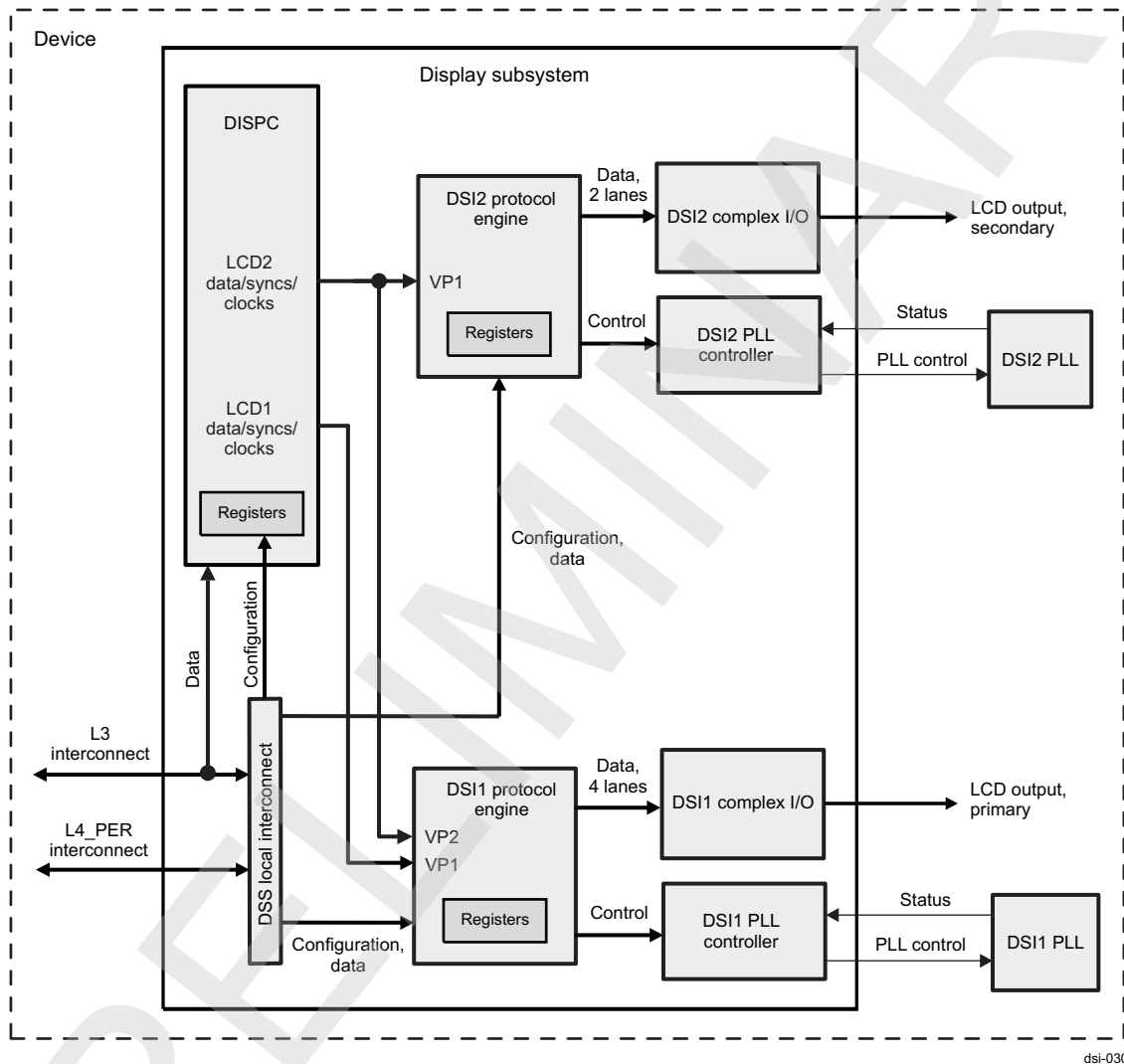
### 10.3.4 DSI Functional Description

This section describes the functions of the DSI1 and DSI2 modules.

#### 10.3.4.1 DSI Block Diagram

Figure 10-121 is a schematic of DSI1 and DSI2.

Figure 10-121. DSI Schematic



#### 10.3.4.2 DSI Clock Configuration

Figure 10-122 shows the clock tree for DSI1 and DSI2.

Figure 10-122. DSI Clock Tree

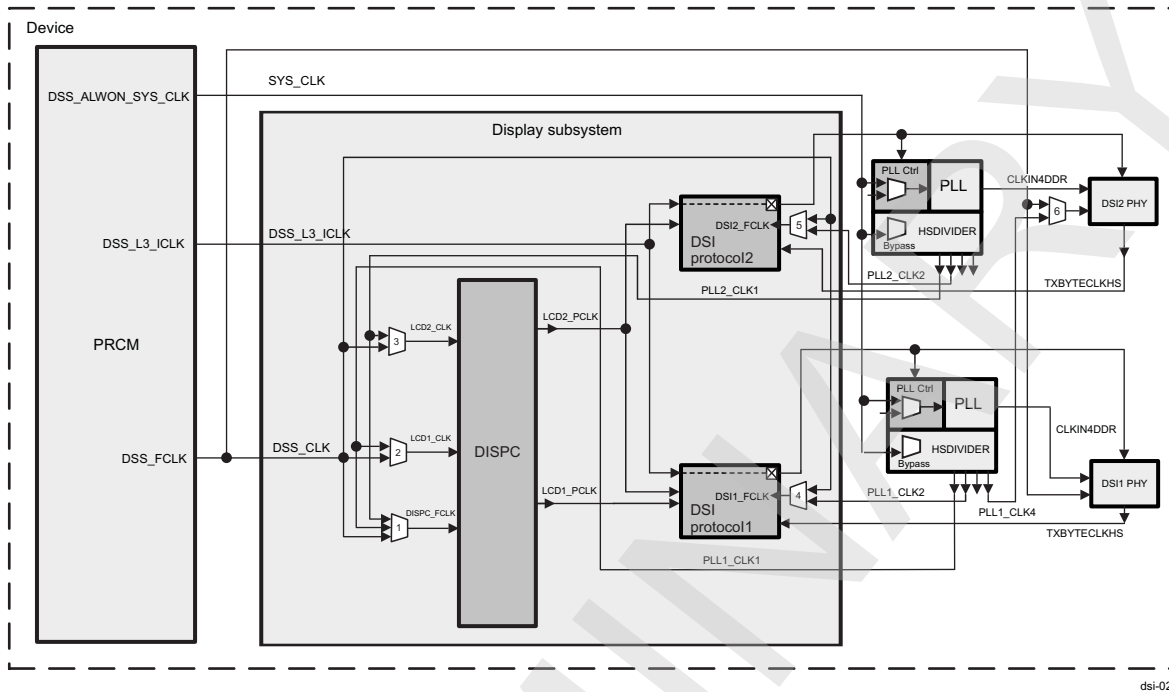


Table 10-471 shows the controls for clock multiplexers.

Table 10-471. DSI Multiplexer Controls

Destination	Source Signal Name	Source	Multiplexer Number	DSS Register Bit Field
DISPC functional clock (DISPC_FCLK)	DSS_CLK	PER_PLL	1	For more information, see <a href="#">Section 10.1.2</a> , <i>DSS Integration</i> .
	PLL1_CLK1	PLL1	1	
	PLL2_CLK1	PLL2	1	
	PLL3_CLK1	PLL3	1	
DSI1 functional clock (LCD1_CLK)	DSS_CLK	PER_PLL	2	For more information, see <a href="#">Section 10.1.2</a> , <i>DSS Integration</i> .
	PLL1_CLK1	PLL1	2	
DSI2 functional clock (LCD2_CLK)	DSS_CLK	PER_PLL	3	For more information, see <a href="#">Section 10.1.2</a> , <i>DSS Integration</i> .
	PLL2_CLK1	PLL2	3	
DSI1 protocol engine functional clock (DSI1_FCLK)	DSS_CLK	PER_PLL	4	For more information, see <a href="#">Section 10.1.2</a> , <i>DSS Integration</i> .
	PLL1_CLK2	PLL1	4	
DSI2 protocol engine functional clock (DSI2_FCLK)	DSS_CLK	PER_PLL	5	For more information, see <a href="#">Section 10.1.2</a> , <i>DSS Integration</i> .
	PLL2_CLK2	PLL2	5	

**NOTE:** Multiplexer 6, shown in [Figure 10-122](#), allows the DSS\_FCLK or PLL1\_CLK4 clock to be sent on DSI2 PHY bypass input. The use of the bypassed clock allows both DSI modules to run on DSS\_FCLK or on a single PLL (DSI1 PLL).

- To select the clock source for the DSI2 PHY bypass input, configure the CONTROL.CONTROL\_DSS\_CONTROL[0] DSS\_MUX6\_SELECT bit in the device control module. See [Chapter 18, Control Module](#).
- To enable the bypass input for the DSI1 PHY or DSI2 PHY module, set the DSI\_PLL\_CONFIGURATION2[15] BYPASSEN bit to 1.

### 10.3.4.3 DSI Interrupt Requests

The DSI protocol engine requires a single interrupt line. The [DSI\\_IRQSTATUS](#) register indicates the general interrupt events. Each VC and complex I/O has a dedicated interrupt register: [DSI\\_VC\\_IRQSTATUS\\_i](#) and [DSI\\_COMPLEXIO\\_IRQSTATUS](#), respectively.

[Table 10-472](#) lists the DSI global interrupt events.

**Table 10-472. DSI Global Interrupts**

IRQ Name	Description
Resynchronization_IRQ	Resynchronization in video mode
TA_TO_IRQ	Turnaround timer expired
LDO_POWER_GOOD_IRQ	Signal LDOPWRGOOD from the DSI_PHY changes its state for the supply VDDALDODSIPLL from up to down or down to up. VDALDODSIPLL is a 1.2-V power supply for the PLL. The voltage is supplied by the internal or external LDO.
SYNC_LOST_IRQ	Synchronization with video port is lost (video mode only).
ACK_TRIGGER_IRQ	Acknowledge trigger is received.
TE_TRIGGER_IRQ	TE PHY trigger is received.
WAKEUP_IRQ	Occurs when the SWakeUp signal is asserted.
HS_TX_TO_IRQ	HS TX time-out Interrupt
LP_RX_TO_IRQ	LS RX time-out Interrupt
COMPLEXIO_ERR_IRQ	Error signaling from complex I/O: The interrupt is triggered when an error is received from the complex I/O (events are defined in <a href="#">DSI_COMPLEXIO_IRQSTATUS</a> ).
PLL_RECAL_IRQ	PLL recalibration event (assertion of DSIRecal signal from the DSI PLL control module)
PLL_UNLOCK_IRQ	PLL un-lock event (deassertion of DSILock signal from the DSI PLL control module)
PLL_LOCK_IRQ	PLL lock event (assertion of DSILock signal from the DSI PLL control module)
VIRTUAL_CHANNEL3_IRQ	VC 3 error signaling from DSI VC 3: The interrupt is triggered when an error is received from DSI VC 3 (events are defined in <a href="#">DSI_VC3_IRQENABLE</a> ).
VIRTUAL_CHANNEL2_IRQ	VC 2 error signaling from DSI VC 2: The interrupt is triggered when an error is received from DSI VC 2 (events are defined in <a href="#">DSI_VC2_IRQENABLE</a> ).
VIRTUAL_CHANNEL1_IRQ	VC 1 error signaling from DSI VC 1: The interrupt is triggered when an error is received from DSI VC 1 (events are defined in <a href="#">DSI_VC1_IRQENABLE</a> ).
VIRTUAL_CHANNEL0_IRQ	VC 0 error signaling from DSI VC 0: The interrupt is triggered when an error is received from DSI VC 0 (events are defined in <a href="#">DSI_VC0_IRQENABLE</a> ).
TE0_LINE_IRQ	TE detected on TE0 CMOS input signal
TE1_LINE_IRQ	TE detected on TE1 CMOS input signal

**NOTE:** For information about other internal DSI interrupt events, see [Table 10-488](#) and [Table 10-494](#).

### 10.3.4.4 DSI Protocol Engine

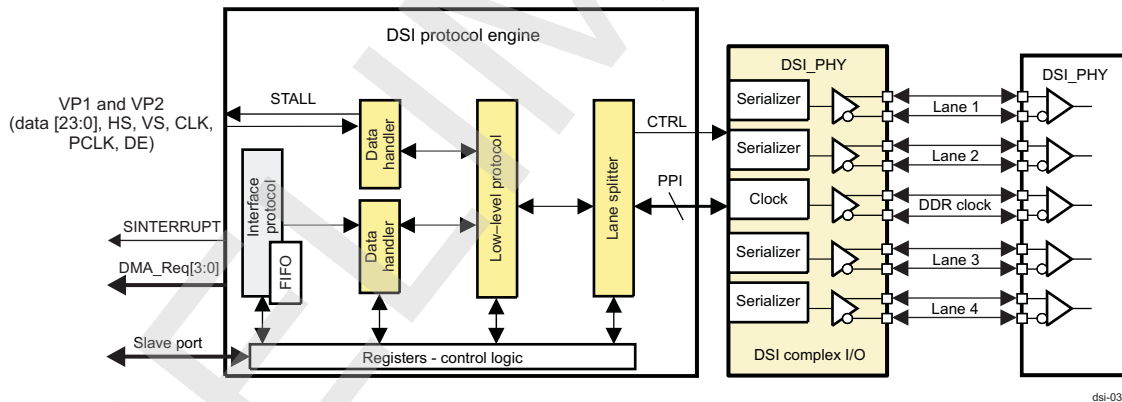
**NOTE:** Copyright 2005-2008 MIPI Alliance, Inc. All rights reserved. MIPI Alliance Member Confidential.

The DSI protocol engine integrates the DSI with the display through the DSI\_PHY module, local interconnect interface, and video interface from the DISPC. The DSI\_PHY or complex I/O module is described in [Section 10.3.4.6, DSI Complex I/O](#). The DSI transmitter (protocol engine + PHY) port can be connected to multiple displays using a single DSI host port. The DSI protocol engine controls the DSI PLL control module described in [Section 10.3.4.5, DSI PLL Controllers](#). The DSI transmitter port can be used in video mode or/and command mode.

#### 10.3.4.4.1 DSI Protocol Overview

The DSI protocol engine receives data from the DISPC video port and/or the L3 interconnect slave port, encapsulates them with the VC ID, generates the ECC and checksum, and splits the data into byte streams to the DSI\_PHY to be sent using LS or HS protocol. The DSI protocol engine receives data and acknowledges from the panel using the same DSI link in case of bidirectional panel. Multiple data streams can be interleaved to support multiple panels connected to the same host DSI port. [Figure 10-123](#) shows the DSI protocol engine, with an example of data lane and clock lane ordering. It is possible to configure other orders through registers as described in [Section 10.3.2.1.1, Data/Clock Configuration](#).

**Figure 10-123. DSI Protocol Engine**



**NOTE:** [Figure 10-123](#) gives information for DSI1 with its four data lanes. DSI2 has only two data lanes.

**NOTE:** The order of the PHY pairs (clock and data lanes) is informative. Each PHY pair can be clock or data. DSI complex I/O receives the configuration for the pin order and the differential  $\pm$  in a pair from the settings in the [DSI\\_COMPLEXIO\\_CFG1](#) register.

#### 10.3.4.4.2 DSI Global Register Controls

Before receiving data from the DSI complex I/O, the DSI\_PHY SCP registers in the DSI complex I/O must be configured. [Table 10-473](#) lists the register access width limitations for all the DSI modules.

**Table 10-473. Register Access Width Limitations**

Register Name	Register Access Width
All DSI complex I/O registers (DSI_PHY SCP)	32-bit only
All DSI PLL control module registers	32-bit only
<a href="#">DSI_VC_LONG_PACKET_HEADER_i</a>	32-bit only
<a href="#">DSI_VC_SHORT_PACKET_HEADER_i</a>	32-bit only
<a href="#">DSI_VC_LONG_PACKET_PAYLOAD_i</a>	16- and 32-bit
All other DSI protocol engine registers	8-, 16-, and 32-bit

**CAUTION**

If a different access width than listed in [Table 10-473](#) is used, an interface error is generated in response to the write using SResp = ERR.

The DSI protocol engine is globally controlled by the [DSI\\_CTRL](#) register. The interface to the complex I/O is enabled by setting the [DSI\\_CTRL\[0\] IF\\_EN](#) bit. When the interface is disabled, it is possible to provide data to the TX FIFO and read pending data in the RX FIFO. When the [DSI\\_CTRL\[0\] IF\\_EN](#) bit is set to 1, the pending packets are sent to the DSI complex I/O, and the data transfer from the video port is ignored until the first vertical sync event is received.

When the [DSI\\_CTRL\[0\] IF\\_EN](#) bit is reset by software, the DSI protocol engine must finish the transfer of the pending data in the TX FIFO and wait for a response if BTA has been sent (the protocol engine is in receive mode); hardware then resets the [DSI\\_CTRL\[0\] IF\\_EN](#) bit. When using video mode, the VC associated with the video port must be enabled before enabling the interface according to the following sequence:

1. The [DSI\\_CTRL\[0\] IF\\_EN](#) bit is equal to 0.
2. Enable the VC associated with video mode by setting the [DSI\\_VC\\_CTRL\\_i\[0\] VC\\_EN](#) bit.
3. Set the [DSI\\_CTRL\[0\] IF\\_EN](#) bit to 1.

**10.3.4.4.3 DSI Transfer Modes**

Two main transfer modes are supported by the DSI modules:

- Video mode: Pixels are received from the video port (VP1 or VP2.) There are some real-time constraints (pixels must be sent at the pixel frequency required by the panel module).
- Command mode: Pixels can be received from the video port or from the L3 interconnect slave port. There are no real-time constraints except that the TE must be avoided by starting the transfer at the correct time during the scan of the display and that it must be fast enough.

**10.3.4.4.3.1 Video Mode**

Video mode refers to the MIPI display pixel interface (DPI) standard. The sync events and pixels must be sent according to the display mode timings. Data is received from the video ports. The DISPC fetches the data from the system memory and provides it to the DSI protocol engine through VP1 or VP2. The short packets used for the sync event use precalculated 32-bit values. The long packets are constructed using the header defined in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) registers. In case of synchronization short packets for video mode, because hardware generates short packets without using the [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) register, if the [DSI\\_VC\\_CTRL\\_i\[8\] ECC\\_TX\\_EN](#) bit is set, the ECC is calculated; otherwise, 0 is used. The feature is used to generate incorrect ECC for debug purpose and to ease the check of the link and peripheral error detection and correction.

The [DSI\\_VM\\_TIMING1](#) through [DSI\\_VM\\_TIMING8](#) registers define the timings of the video mode.

The [DSI\\_VM\\_TIMING2\[27:24\] WINDOW\\_SYNC](#) bit field defines the synchronization period. The recommended value is 0x4 based on the implementation of the resynchronization scheme.

The [DSI\\_CTRL\[20\]](#) [BLANKING\\_MODE](#) bit defines whether long blanking packets or LPS are used during the blanking periods (except HFP, HBP, and HSA defined by other bits) when there is no pending data in TX FIFO ready to be sent. Software must ensure that there is no data in the TX FIFO, no BTA, no RESET trigger sent, and the [DSI\\_VC\\_CTRL\\_i\[9\]](#) [MODE\\_SPEED](#) bit is set to 1 (HS mode) to keep the video mode transfer in HS mode during blanking periods (except for the last blanking period, because it is required to go to LPS at least once per frame).

The [DSI\\_CTRL\[21\]](#) [HFP\\_BLANKING\\_MODE](#), [DSI\\_CTRL\[22\]](#) [HBP\\_BLANKING\\_MODE](#), and [DSI\\_CTRL\[23\]](#) [HSA\\_BLANKING\\_MODE](#) bits define whether during these blanking packets data from TX FIFO or long blanking packets are sent.

To ensure that the writes to the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register are handled correctly as header information for video mode long packets, the following registers must be programmed:

- [DSI\\_VC\\_CTRL\\_i\[0\]](#) [VC\\_EN](#) bit set to 0
- [DSI\\_VC\\_CTRL\\_i\[4\]](#) [MODE](#) bit set to 1
- [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register access
- [DSI\\_VC\\_CTRL\\_i\[0\]](#) [VC\\_EN](#) bit set to 1

---

**NOTE:** The hardware ignores [DSI\\_VC\\_CTRL\\_i\[1\]](#) [SOURCE](#) and [DSI\\_VC\\_CTRL\\_i\[9\]](#) [MODE\\_SPEED](#) bits when video mode is selected (the [DSI\\_VC\\_CTRL\\_i\[4\]](#) [MODE](#) bit set to 1).

---

The [SYNC\\_LOST\\_IRQ](#) and [RESYNCHRONIZATION\\_IRQ](#) interrupt events indicate whether the DSI protocol engine has not been able to resynchronize the video port timing to its own timing base or if it has been done. The [RESYNCHRONIZATION\\_IRQ](#) indicates to the software user that the video port works but the configuration of the timings for the DISPC and for DSI protocol engine may need to be modified to avoid the resynchronization. The [SYNC\\_LOST\\_IRQ](#) and [RESYNCHRONIZATION\\_IRQ](#) events can be respectively monitored in [DSI\\_IRQSTATUS\[18\]](#) [SYNC\\_LOST\\_IRQ](#) and [DSI\\_IRQSTATUS\[5\]](#) [RESYNCHRONIZATION\\_IRQ](#) status bits.

#### 10.3.4.4.3.2 Command Mode

The command mode refers to the MIPI display command set (DCS) standard. The commands, parameters, and pixels are sent to the display module with limited real-time constraints (as defined in [Section 10.3.4.4.3.1, Video Mode](#)). The pixels can be provided on one of the two video ports by the DISPC or on the L3 interconnect slave port.

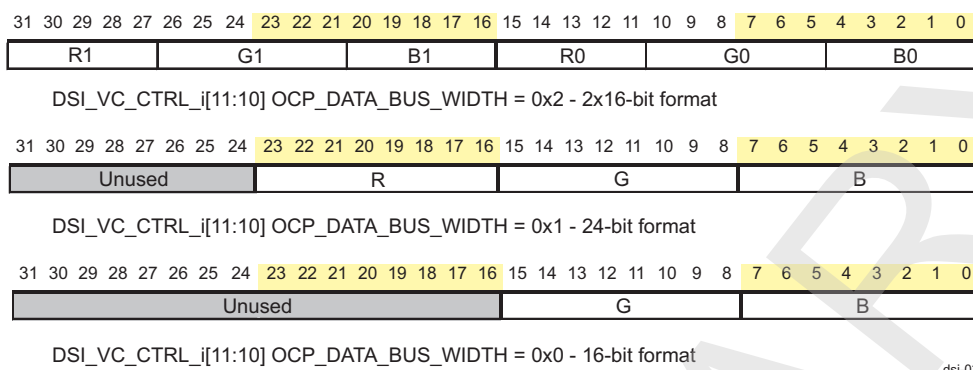
The [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) registers are used for the header of long packets; the [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) registers are used for the short packets.

For the payload, the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register is used. Each 32-bit payload data is written to the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register from the MPU subsystem or sDMA. It is buffered to send packets with higher rate than the L3 interconnect frequency can provide. If [DSI\\_VC\\_CTRL\\_i\[30\]](#) [DCS\\_CMD\\_ENABLE](#) = 1, the word count defined in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register is used to determine the number of bytes to be sent using the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register minus the size of the DCS command header added by the protocol engine at the beginning of the payload (also sent using the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register). If [DSI\\_VC\\_CTRL\\_i\[30\]](#) [DCS\\_CMD\\_ENABLE](#) = 0, the word count defined in the register [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register is used to determine the number of bytes to be sent using the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register.

The [DSI\\_VC\\_CTRL\\_i\[11:10\]](#) [OCP\\_DATA\\_BUS\\_WIDTH](#) bit field is used to define the size of the data on the L3 interconnect slave port for the write access (read access to the RX FIFO is not affected by the bit field value):

- 0x0: 16-bit data width (LSB of the 32-bit interface port data bus)
- 0x1: 24-bit data width (LSB of the 32-bit interface port data bus)
- 0x2: 2x16-bit data width (first pixel on the LSB of the 32-bit Interface port data bus and second pixel on the MSB of the 32-bit interface port data bus for the same interface access)
- 0x3: 32-bit data width (no bit, no byte, no pixel manipulation)

[Figure 10-124](#) shows the different data width configurations of the L3 interconnect slave port.

**Figure 10-124. Interface Data Configuration**

**NOTE:** To provide 18-bit pixel loosely packed format to the L3 interconnect slave port, the [DSI\\_VC\\_CTRL\\_i\[11:10\] OCP\\_DATA\\_BUS\\_WIDTH](#) bit field must be set to 0x1. The 18-bit pixel packed format is not supported in case of command mode on the L3 interconnect slave port.

The write into the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register is required before accessing the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register. The DSI protocol engine must extract the length of the payload and discard the extra data sent using the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register. Writes into the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register are valid only if the VC is enabled; otherwise, the write is ignored. When receiving pixels on the video port, only the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register is used. The video port pixels are used for the payload.

#### 10.3.4.4.3.2.1 Command Mode TX FIFO

The single TX FIFO is used to store data from the L3 interconnect before sending it to the panel. The configuration of the FIFO for a specific VC must be done only when the VC is disabled.

The VC must not be enabled when data is still pending in the TX FIFO for the corresponding space allocated for the VC from the previous active period. When the VC space in the TX FIFO is empty, the VC can be enabled.

For each VC, two dedicated registers ([DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) and [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#)) are used to provide data for long packets. The [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) register is used to provide data for short packets (32 bits long).

For each long packet, the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register must be written first, and then the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register. The only exception is when the word count defined in the header equals 0. In that case, it is not required to write into the payload register. For consecutive long packets, the header must be written into the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register even if the value remains the same.

The TX FIFO stores all the pending bytes to be sent to the peripheral(s). Multiple receivers can be addressed using the VC capability.

The 32-bit write requests for each VC to the TX FIFO are kept in order while sending the data to the DSI\_PHY inside the VC requests. The only exception is the last 32-bit write for the last bytes of the payload data, because it can be 1, 2, 3, or 4 bytes.

When the last transfer is a 32-bit write, but the number of valid bytes is only 1, 2, or 3 (calculated using the header word count and the number of bytes are received for the payload), hardware stores the 32-bit value into the TX FIFO but the invalid bytes are not sent and are discarded.

When the word count defined in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register is not a multiple of the request threshold value defined in the [DSI\\_VC\\_CTRL\\_i\[19:17\] DMA\\_TX\\_THRESHOLD](#) bit field, 32-bit requests and/or bytes are discarded by hardware to store in FIFO only the exact number of valid bytes.

The triggering event to send data to the DSI link is one of the following events:



- All bytes have been received in the FIFO (header + payload).
- The space of the FIFO allocated for the VC is full.
- The space of the FIFO allocated for the VC is not enough to request more data using DMA request (the threshold value is bigger than the space left in the TX FIFO for the VC).
- When the VC is disabled, the remaining bytes in the FIFO are sent to the DSI link.

**NOTE:** In case video mode is active, the blanking period must be large enough to allow the transfer of the packet(s).

Sequential arbitration must be set ([DSI\\_CTRL\[3\]](#) bit `TX_FIFO_ARBITRATION = 0x1`) if only one VC is used to send multiple packets during the same blanking period.

When consecutive packets should be sent in HS mode, to ensure that there is no LP transition between them, at least one of the following conditions must be valid:

- Packets from the same VC
- Short packets or long packets with a payload size multiple of 4 bytes

To stop a transfer immediately when packets are being transferred from the TX FIFO to DSI\_PHY, a flush of the FIFO is necessary. It can be required in a dead-lock situation to flush the FIFO even if bytes are still in the FIFO. In that case, it is necessary to ensure having a known state of the whole DSI protocol engine module (software reset may be required) to start of new transfer through the TX FIFO.

To flush the FIFO (discard the data) for some pending bytes:

1. Change the size of the space of FIFO to 0 by writing [DSI\\_TX\\_FIFO\\_VC\\_SIZE](#) `VCi_FIFO_SIZE` bit fields (where *i* is the VC number from 0 to 3).
2. Disable the VC by resetting the [DSI\\_VC\\_CTRL\\_i\[0\]](#) `VC_EN` bit to 0.
3. Wait for [DSI\\_VC\\_CTRL\\_i\[15\]](#) `VC_BUSY = 0` (`VC_BUSY` is set to 0 after TX FIFO flush is done).

It is necessary to check that there is no pending request before changing the size of the allocated FIFO for the VC by reading the relevant [DSI\\_VC\\_CTRL\\_i\[15\]](#) `VC_BUSY` bit or by using the interrupt `PACKET_SENT_IRQ`. This interrupt is monitored by reading the [DSI\\_VC\\_IRQSTATUS\\_i\[2\]](#) `PACKET_SENT_IRQ` status bit.

The [DSI\\_CTRL\[3\]](#) `TX_FIFO_ARBITRATION` bit defines the arbitration scheme:

- Round-robin between enabled VCs with pending ready requests (that is, all bytes for the packets are in the FIFO or the space of the FIFO for the VC is full) starting from the VC that has the lowest VC ID number.
- Sequential: All the pending ready requests for one VC are sent before moving to another VC. The condition of "space of the FIFO is full" is evaluated after the end of each packet.

To use in-order for all requests for all channels, a single VC must be used. (The VC ID defined in the header provided to hardware using the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) or [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) register is not used and not modified by the DSI protocol engine.)

The [DSI\\_TX\\_FIFO\\_VC\\_SIZE](#) register defines the allocated number of 33-bit values for each VC in the TX FIFO and the start address for each VC. The size of the space allocated in the TX FIFO is defined by the [DSI\\_TX\\_FIFO\\_VC\\_SIZE](#) register - `VCi_FIFO_SIZE` bit fields. It must be a multiple of the threshold defined in the [DSI\\_VC\\_CTRL\\_i\[19:17\]](#) `DMA_TX_THRESHOLD` bit field. To change the size of the space of the memory allocated for a specific VC, the VC must be disabled by setting the [DSI\\_VC\\_CTRL\\_i\[0\]](#) `VC_EN` bit to 0. The whole FIFO cannot be used by all the VCs at the same time because a VC can be disabled to change one or multiple parameters. Software users must configure the start address and size for each VC correctly.

[Table 10-474](#) lists the corresponding values for the size of the space allocated in the FIFO.

**Table 10-474. VC TX FIFO Size Values**

<a href="#">DSI_TX_FIFO_VC_SIZE.VCi_FIFO_SIZE</a> (where <i>i</i> = 0, 3)	Space Size (up to the Size of the FIFO)
0	0 x 33 bits
1	32 x 33 bits

**Table 10-474. VC TX FIFO Size Values (continued)**

<a href="#">DSI_TX_FIFO_VC_SIZE</a> .VC <sub>i</sub> _FIFO_SIZE (where i = 0, 3)	Space Size (up to the Size of the FIFO)
2	64 x 33 bits
3	96 x 33 bits
4	128 x 33 bits

**NOTE:** TX FIFO is 128 x 33 bits. Therefore, the sum of all VC FIFO allocations cannot exceed 128 x 33 bits.

**CAUTION**

Bit 32 of each location in the FIFO is used by the DSI protocol engine to tag the data in the TX FIFO. It is for internal use only.

Table 10-475 lists the start address of the space in the FIFO.

**Table 10-475. VC TX FIFO Start Address**

<a href="#">DSI_TX_FIFO_VC_SIZE</a> .VC <sub>x</sub> _FIFO_ADD (where x = 0, 2)	Start Address
0	0
1	32
2	64
3	96
4	128

**CAUTION**

The different VC spaces must not overlap.

When the TX FIFO is full:

- The overflow interrupt (FIFO\_TX\_OVF\_IRQ) is generated in the [DSI\\_VC\\_IRQSTATUS\\_i\[3\]](#) FIFO\_TX\_OVF\_IRQ status bit.
- There is no L3 interconnect error generated.
- The interface commands are accepted but the data are not written into the FIFO.

To ensure that all writes are stored correctly in the TX FIFO, the FIFO must not be full. The room in the space allocated for the VC in the TX FIFO is given in the [DSI\\_TX\\_FIFO\\_VC\\_EMPTYNESS](#) register. When there is no space allocated in the TX FIFO for the VC, the [DSI\\_TX\\_FIFO\\_VC\\_EMPTYNESS](#) register indicates a value of 0 for the VC space emptiness.

While waiting to receive the first VSYNC event on the video port to start video mode on the DSI link, no command data from TX FIFO is sent on the interface. This is required to ensure that, when receiving the VSYNC event, there is no ongoing command mode transfer that could delay the start of video mode on the the DSI link.

**10.3.4.4.3.2.2 Command Mode RX FIFO**

The RX FIFO stores the data received from the DSI complex I/O. The data are always packed in the RX FIFO (single or multiple packets received during single or multiple BTA periods).

The read requests (single or burst) can be less than, equal to, or greater than the packet size. If the packet size is smaller than the read request, the following packet(s) is also transferred. If the packet size is longer than the read request, only part of the packet is transferred. In that case, the logic keeps the VC information to provide the rest of the data during the next read request(s).

The [DSI\\_RX\\_FIFO\\_VC\\_SIZE](#) register defines the allocated number of 33-bit values for each VC in the RX FIFO and the start address for each VC. Only the enabled VCs are considered.

To change the size of the memory space allocated for a specific VC, the VC must be disabled by resetting the [DSI\\_VC\\_CTRL\\_i\[0\]](#) VC\_EN bit to 0. The entire FIFO cannot be used by the entire VC at the same time because a VC can be disabled to change one or multiple parameters. Software users must configure the start address and size for each VC correctly.

[Table 10-476](#) lists the corresponding values for the size of the space allocated in the FIFO:

**Table 10-476. VC RX FIFO Size Values**

<a href="#">DSI_RX_FIFO_VC_SIZE.VCi_FIFO_SIZE</a> (where i = 0, 3)	Space Size (up to the size of the FIFO)
0	0 x 33 bits
1	32 x 33 bits
2	64 x 33 bits
3	96 x 33 bits
4	128 x 33 bits

**NOTE:** RX FIFO is 128 x 33 bits. Therefore, the sum of all VC FIFO allocations cannot exceed 128 x 33 bits.

**CAUTION**

Bit 32 of each location in the FIFO is used by the DSI protocol engine to tag the data in the RX FIFO. It is for internal use only.

[Table 10-477](#) lists the start address of the space in the FIFO.

**Table 10-477. VC RX FIFO Start Address**

<a href="#">DSI_RX_FIFO_VC_SIZE.VCx_FIFO_ADD</a> (where x = 0, 2)	Start Address
0	0
1	32
2	64
3	96
4	128

**CAUTION**

The different VC spaces must not overlap.

While reading the received bytes in the RX FIFO, only the [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) register is used, because hardware does not keep track of the header position for long packets and the start/end of each packet. Software must extract the information from the bytes read from the RX FIFO. There is no specific hardware to track the received bytes in the RX FIFO. The [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) and [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) registers are not used.

The ECC is used only by the first header when receiving multiple packets during the same LP RX transfer from the peripheral because the DSI protocol engine does not parse the header to identify the length of the packets. In case of multiple packets, the checksum is not enabled because hardware checks the checksum considering a single packet. The ECC in the first header is used to correct and check the header. For the following headers in the same LP RX transfer, hardware does not detect a header and cannot check and/or detect errors in the headers of the packets except for the first packet.

When the RX FIFO is empty:

- No interface error is generated.
- Commands are accepted and the data for the responses are 0s.

#### 10.3.4.4.3.2.3 Command Mode DMA Requests

The DMA requests (DSI\_DMA\_REQ) are used to allow automatic transfer by the sDMA or MPU (with less efficiency and throughput capability) from the DSI RX FIFO to the system memory and from the system memory to the DSI TX FIFO. Two independent DMA requests for RX FIFO and TX FIFO for the same VC are supported. The read and write accesses can use burst structure.

The thresholds used for requests for the TX FIFO and RX FIFO are programmable through the [DSI\\_VC\\_CTRL\\_i\[19:17\] DMA\\_TX\\_THRESHOLD](#) and [DSI\\_VC\\_CTRL\\_i\[26:24\] DMA\\_RX\\_THRESHOLD](#) bit fields for TX FIFO and RX FIFO, respectively. DMA requests are asserted on the threshold value. The size of the space allocated in TX FIFO for each VC must be a multiple of the value of the [DSI\\_VC\\_CTRL\\_i\[19:17\] DMA\\_TX\\_THRESHOLD](#) bit field.

In the case of TX FIFO, if all the bytes defined by the word count field in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register are received, the DMA request is no longer asserted. Even during the last transfer, less than the [DMA\\_TX\\_THRESHOLD](#) number of bytes are received because the word count is not a multiple of the [DMA\\_TX\\_THRESHOLD](#) value.

In the case of RX FIFO, while the DMA request is used to transfer the data from the RX FIFO to the system memory, the sDMA must be programmed to read the exact number of received bytes in the FIFO. If users do not know the size of the received bytes, the direct access of the RX FIFO through the [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) register is performed until the [DSI\\_VC\\_CTRL\\_i\[20\] RX\\_FIFO\\_NOT\\_EMPTY](#) bit goes to 0.

The only exception is when the LP data transfer finishes and the threshold value is not reached. In this case, the DMA request is asserted. The drain of the FIFO is supported in that configuration to empty the FIFO even if the number of data received is not a multiple of the threshold value.

The use of each DMA request is programmable by software. The [DSI\\_VC\\_CTRL\\_i\[23:21\] DMA\\_TX\\_REQ\\_NB](#) bit field is dedicated to DMA request numbering for the TX FIFO. The [DSI\\_VC\\_CTRL\\_i\[29:27\] DMA\\_RX\\_REQ\\_NB](#) bit field is dedicated to DMA request numbering for the RX FIFO.

When the DMA request is used to indicate the number of 32-bit values ready in the RX FIFO that BTA has received from the peripheral, which indicates an end of the transfer from the peripheral to the host for a transfer to the system memory, the DMA request that corresponds to the VC ID is generated.

The sDMA transfers the number of 32-bit values defined in the threshold register or the exact number of bytes received from the peripheral (the user must know the number of expected received bytes to program the sDMA correctly). When the sDMA transfers a multiple number of threshold values, the DSI protocol engine sends 0s for the data when there is no more received data in the RX FIFO for the VC. Software users must parse the data and determine the valid bytes.

The size of the DMA transfer can be determined by reading the [DSI\\_RX\\_FIFO\\_VC\\_FULLNESS](#) register. The BTA interrupt (BTA\_IRQ) must be used to know when to read the number of received bytes. The BTA interrupt must be monitored by reading the [DSI\\_VC\\_IRQSTATUS\\_i\[5\] BTA\\_IRQ](#) status bit. The DMA request must not be enabled until the sDMA is programmed with the correct number of data to read from RX FIFO.

This method must also be used when the RX FIFO space for the VC is expected to overflow because the expected number of data to be received is greater than the space allocated for the VC. Instead, the DMA request is asserted as soon as the threshold is reached or when BTA is received.

When the DMA request is used to indicate the number of 33-bit entries empty in the TX FIFO for a transfer from the system memory, the DMA request corresponding to the VC ID is generated.

---

**NOTE:** For the most efficient transfer, the size of the request (read or write, single or burst) must be aligned with the threshold value.

---

Concurrent access using interlaced requests (read/write) to the TX and RX FIFO is supported for the same VC ID or different VC IDs.

### 10.3.4.4.3.3 Video + Command Mode

The two modes can be interlaced to send two DSI streams to two types of panels: video or command. The number of concurrent video streams is limited to one. The number of concurrent command mode streams is limited to four when there is no video stream, and three when there is a video stream. If one DSI stream is using video mode, the command mode pixels must be provided only on the L3 interconnect slave port.

### 10.3.4.4.3.4 Burst Mode

Frequency and transparent burst modes are supported:

- Frequency burst mode is used to reduce the HS period by increasing the clock frequency on the DSI link. In some cases, it reduces power consumption on the link. The non-HS period used typically to drive the main panel can be used to send data to the secondary panel or to allow feedback (acknowledge) from the primary and secondary panels. The DSI protocol engine must buffer a full line before sending the HS packets for the line. A double-buffering mechanism is required to send a line while the following one is being received on the video port. For more information about double-buffering, see [Section 10.3.4.4.5.4, Ping-Pong Buffer](#).
- The transparent burst mode is used by increasing the pixel clock frequency generated by the DISPC with an increase of the horizontal blanking period.

### 10.3.4.4.3.5 Interleaving Mode

Video mode can output command mode packets, which are provided to DSI through the L3 interconnect, during the blanking periods of the video stream sequence on the PPI link. These command mode packets can be programmed as HS packets or low-power packets.

During a video stream sequence on the PPI link, four types of gap exist:

- BLLP gap: Blanking period during VSA, VBP, and VFP lines
- HSA gap: Blanking period during VACT lines; always between HS and HE short packet
- HBP gap: Blanking period during VACT lines; always between HS/HE short packet and data pixel long packet
- HFP gap: Blanking period during VACT lines; always between data pixel long packet and the end of the current VACT line

To perform interleaving in a particular gap, video mode must be set to go into LPS during the blanking gap. Each type of gap has separate configurable register bits that determine whether a blanking long packet will be sent or the link will go into LPS during the gap on the PPI link. If LPS is set during a gap, the DSI module performs interleaving during that period.

Two set of registers are available for:

- HS interleaving (when HS command mode packets must be sent during a video stream on the PPI link)
- Low-power interleaving (when low-power command mode packets must be sent during a video stream on the PPI link)

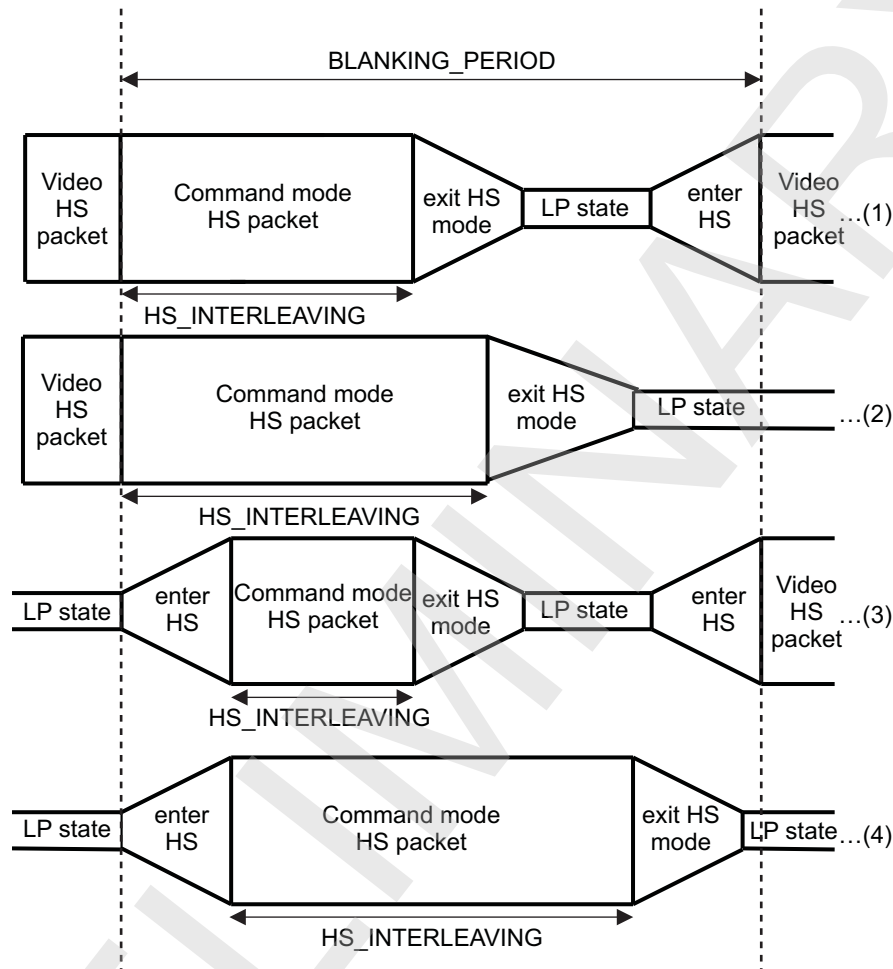
#### 10.3.4.4.3.5.1 HS Command Mode Interleaving

[Figure 10-125](#) shows the various HS mode scenarios in interleaving mode during a blanking gap. For each type of blanking gap, a dedicated bit field determines the number of TXBYTECLKHS clock cycles used for interleaving in HS command mode packets.

- The [DSI\\_VM\\_TIMING6\[31:16\]](#) BL\_HS\_INTERLEAVING bit field defines the number of TXBYTECLKHS clock cycles used to interleave HS command mode packets during a BLLP gap.
- The [DSI\\_VM\\_TIMING4\[7:0\]](#) HBP\_HS\_INTERLEAVING bit field defines the number of TXBYTECLKHS clock cycles used to interleave HS command mode packets during an HBP gap.
- The [DSI\\_VM\\_TIMING4\[15:8\]](#) HFP\_HS\_INTERLEAVING bit field defines the number of TXBYTECLKHS clock cycles used to interleave HS command mode packets during an HFP gap.
- The [DSI\\_VM\\_TIMING4\[23:16\]](#) HSA\_HS\_INTERLEAVING bit field defines the number of TXBYTECLKHS clock cycles used to interleave HS command mode packets during an HSA gap.

These programmable values must be programmed to satisfy the timings for the clock and data lane to enter and exit HS mode latency. According to the scenario, different equations must be considered when calculating the register values.

**Figure 10-125. HS Command Mode Interleaving**



ds-i-327

**NOTE:** For calculations and equations, the following abbreviations are used: EXIT\_CLK\_HS\_MODE represents the exit HS mode latency for the clock lane. There is no dedicated register for this value but the programmer must know this value for further calculations:

$$\text{EXIT\_CLK\_HS\_MODE} = T_{\text{CLK-TRAIL}} + T_{\text{HS-EXIT}}$$

**NOTE:** For more information about other timing parameters used in the following equations, see [Section 10.3.4.4.4.1, Timing Parameters for an LP to HS Transaction](#), and [Section 10.3.4.4.2, Timing Parameters for an HS to LP Transaction](#).

For the following equations, BLANKING\_PERIOD represents the BLLP, HSA, HBP, or HFP blanking periods. The HS\_INTERLEAVING period represents the maximal period HS command mode packets. Its value is set in the BL\_HS\_INTERLEAVING, HSA\_HS\_INTERLEAVING, HBP\_HS\_INTERLEAVING, or HFP\_HS\_INTERLEAVING bit fields, depending on the blanking type.

In each scenario, two calculations are present, depending on the value of the [DSI\\_CLK\\_CTRL\[13\] DDR\\_CLK\\_ALWAYS\\_ON](#) bit:

- DDR\_CLK\_ALWAYS\_ON = 1: Clock lane is always active.

- **DDR\_CLK\_ALWAYS\_ON = 0:** Clock lane is activated only when there are HS packets to be sent on the PPI link.

### CAUTION

When the **DDR\_CLK\_ALWAYS\_ON** bit is programmed to 1, the following must be considered:

- LP state can be achieved on the PPI link only if the data lane has enough time to go out from HS mode and enter a new HS mode during a period inside which the PPI link is meant to be in LP state.
- This can be expressed with the following equation:  

$$\text{EXIT\_HS\_MODE\_LATENCY} + \max\{\text{ENTER\_HS\_MODE\_LATENCY}, 2\} + 1 \leq \text{BLANKING\_PERIOD}$$
- If the equation cannot be satisfied, because the LP state period is too short, software must program the DSI module in a way that during this blanking period a HS blanking packet is sent.

When the **DDR\_CLK\_ALWAYS\_ON** bit is programmed to 0, the following must be considered:

- LP state can be achieved on the PPI link only if the data and clock lanes have enough time to go out from HS mode and enter a new HS mode during a period inside which the PPI link is meant to be in LP state.
- This can be expressed with the following equations:  

$$\text{EXIT\_HS\_MODE\_LATENCY} + \max\{\text{ENTER\_HS\_MODE\_LATENCY}, 2\} + 1 \leq \text{BLANKING\_PERIOD}$$

$$\text{DDR\_CLK\_POST} + \text{EXIT\_CLK\_HS\_MODE} + \text{DDR\_CLK\_PRE} + \text{ENTER\_HS\_MODE\_LATENCY}(\text{data lane}) + 1 \leq \text{BLANKING\_PERIOD}$$
- If both equations cannot be satisfied, because the LP state period is too short, software must program the DSI module in a way that during this blanking period a HS blanking packet is sent.

- Scenario 1: The gap for interleaving starts and ends with a regular video stream HS packet.
  - **DDR\_CLK\_ALWAYS\_ON = 1**  

$$\text{HS\_INTERLEAVING} = \text{BLANKING\_PERIOD} - (\text{EXIT\_HS\_MODE\_LATENCY} + \max\{\text{ENTER\_HS\_MODE\_LATENCY}, 2\} + 1)$$
  - **DDR\_CLK\_ALWAYS\_ON = 0**  

$$\text{HS\_INTER1} = \text{BLANKING\_PERIOD} - (\text{EXIT\_HS\_MODE\_LATENCY} + \max\{\text{ENTER\_HS\_MODE\_LATENCY}, 2\} + 1)$$

$$\text{HS\_INTER2} = \text{BLANKING\_PERIOD} - (\text{DDR\_CLK\_POST} + \text{EXIT\_CLK\_HS\_MODE} + \text{DDR\_CLK\_PRE} + \text{ENTER\_HS\_MODE\_LATENCY} + 1)$$

$$\text{HS\_INTERLEAVING} = \min\{\text{HS\_INTER1}, \text{HS\_INTER2}\}$$
- Scenario 2: The gap for interleaving starts with a regular video stream HS packet and ends in LPS.
  - **DDR\_CLK\_ALWAYS\_ON = 1**  

$$\text{HS\_INTERLEAVING} = \text{BLANKING\_PERIOD} - (\text{EXIT\_HS\_MODE\_LATENCY} + 3)$$
  - **DDR\_CLK\_ALWAYS\_ON = 0**  

$$\text{HS\_INTER1} = \text{BLANKING\_PERIOD} - (\text{EXIT\_HS\_MODE\_LATENCY} + 3)$$

$$\text{HS\_INTER2} = \text{BLANKING\_PERIOD} - (\text{DDR\_CLK\_POST} + \text{EXIT\_CLK\_HS\_MODE} + 3)$$

$$\text{HS\_INTERLEAVING} = \min\{\text{HS\_INTER1}, \text{HS\_INTER2}\}$$
- Scenario 3: The gap for interleaving starts with the LPS and ends with a regular video stream HS packet.
  - **DDR\_CLK\_ALWAYS\_ON = 1**  

$$\text{HS\_INTERLEAVING} = \text{BLANKING\_PERIOD} - (\text{ENTER\_HS\_MODE\_LATENCY} + \text{EXIT\_HS\_MODE\_LATENCY} + \max\{\text{ENTER\_HS\_MODE\_LATENCY}, 2\} + 1)$$
  - **DDR\_CLK\_ALWAYS\_ON = 0**

$$\begin{aligned} \text{HS\_INTER1} &= \text{BLANKING\_PERIOD} - (\text{DDR\_CLK\_PRE} + \text{ENTER\_HS\_MODE\_LATENCY} + \\ &\text{EXIT\_HS\_MODE\_LATENCY} + \max\{\text{ENTER\_HS\_MODE\_LATENCY}, 2\} + 1) \\ \text{HS\_INTER2} &= \text{BLANKING\_PERIOD} - (\text{DDR\_CLK\_PRE} + \text{ENTER\_HS\_MODE\_LATENCY} + \\ &\text{DDR\_CLK\_POST} + \text{EXIT\_CLK\_HS\_MODE} + \text{DDR\_CLK\_PRE} + \text{ENTER\_HS\_MODE\_LATENCY} + \\ &1) \\ \text{HS\_INTERLEAVING} &= \min\{\text{HS\_INTER1}, \text{HS\_INTER2}\} \end{aligned}$$

- Scenario 4: The gap for interleaving starts with the LPS and ends with a regular video stream HS packet.
  - $\text{DDR\_CLK\_ALWAYS\_ON} = 1$   
 $\text{HS\_INTERLEAVING} = \text{BLANKING\_PERIOD} - (\text{ENTER\_HS\_MODE\_LATENCY} + \text{EXIT\_HS\_MODE\_LATENCY} + 3)$
  - $\text{DDR\_CLK\_ALWAYS\_ON} = 0$   
 $\text{HS\_INTER1} = \text{BLANKING\_PERIOD} - (\text{DDR\_CLK\_PRE} + \text{ENTER\_HS\_MODE\_LATENCY} + \text{EXIT\_HS\_MODE\_LATENCY} + 3)$   
 $\text{HS\_INTER2} = \text{BLANKING\_PERIOD} - (\text{DDR\_CLK\_PRE} + \text{ENTER\_HS\_MODE\_LATENCY} + \text{DDR\_CLK\_POST} + \text{EXIT\_CLK\_HS\_MODE} + 1)$   
 $\text{HS\_INTERLEAVING} = \min\{\text{HS\_INTER1}, \text{HS\_INTER2}\}$

#### 10.3.4.4.3.5.2 LP Command Mode Interleaving

Figure 10-126 shows the various LP mode scenarios in interleaving mode during a blanking gap. For each type of blanking gap, a dedicated bit field determines the number of TXBYTECLKHS clock cycles used for interleaving in LP command mode packets.

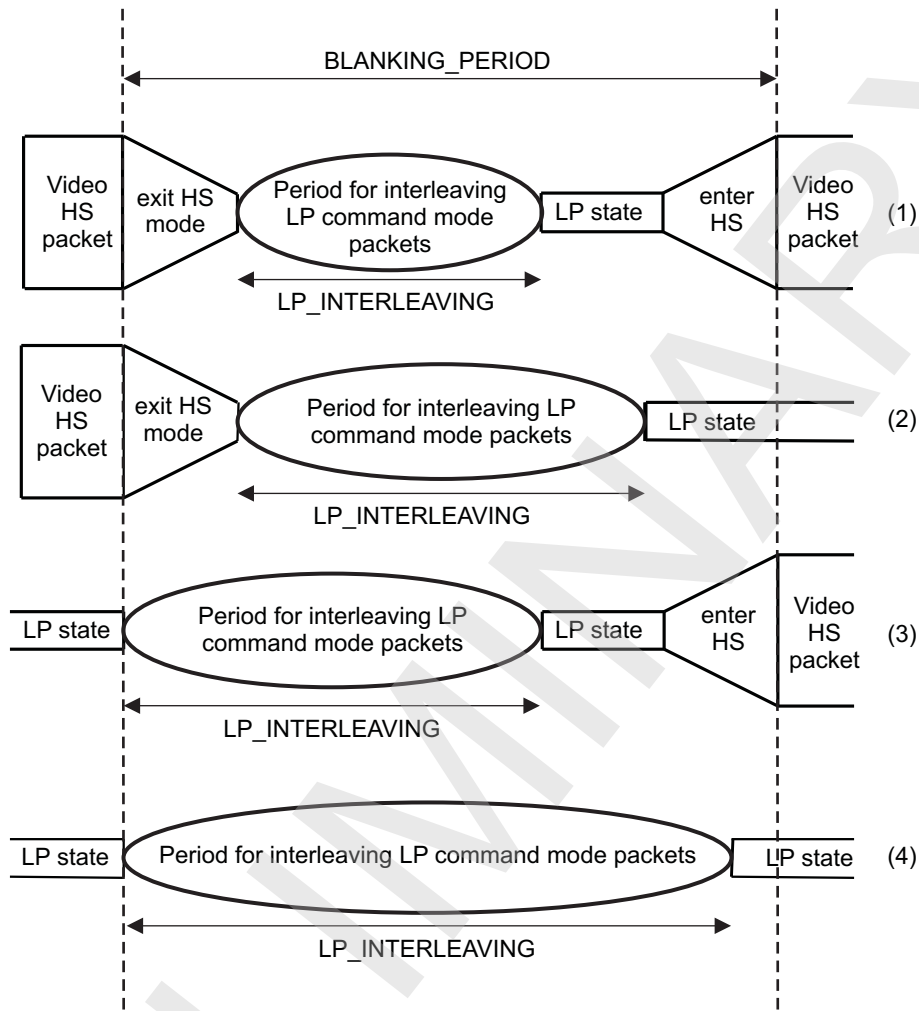
- The `DSI_VM_TIMING6[15:0]` `BL_LP_INTERLEAVING` bit field defines the number of TXBYTECLKHS clock cycles used to interleave the HS command mode packets during a BLLP gap.
- The `DSI_VM_TIMING5[7:0]` `HBP_LP_INTERLEAVING` bit field defines the number of TXBYTECLKHS clock cycles used to interleave the HS command mode packets during an HBP gap.
- The `DSI_VM_TIMING5[15:8]` `HFP_LP_INTERLEAVING` bit field defines the number of TXBYTECLKHS clock cycles used to interleave HS command mode packets during an HFP gap.
- The `DSI_VM_TIMING5[23:16]` `HSA_LP_INTERLEAVING` bit field defines the number of TXBYTECLKHS clock cycles used to interleave HS command mode packets during an HSA gap.

These values must be programmed to satisfy the timings for clock and data lane enter and exit LP mode latency. Clock lane timings do not affect LP command mode interleaving, because the clock lane can be controlled separately, compared with the data lane HS and low-power mutually exclusive control. Clock lanes can be in HS mode while the data lanes are in HS data transfer mode, low-power data transfer mode, or in LPS.

According to this scenario, different equations must be considered for calculating register values.



Figure 10-126. LP Command Mode Interleaving



dsi-328

For the following equations, BLANKING\_PERIOD represents the BLLP, HSA, HBP, or HFP blanking periods. The LP\_INTERLEAVING period represents the maximal period in LP command mode packets. Its value is set in the BL\_LP\_INTERLEAVING, HSA\_LP\_INTERLEAVING, HBP\_LP\_INTERLEAVING, or HFP\_LP\_INTERLEAVING bit fields, depending on the blanking type.

ALLOWED\_HSBYTE\_CLOCKS\_FOR\_LP represents the number of TXBYTECLKHS clock cycles during which LP interleaving can appear.

To calculate the LP\_INTERLEAVING value:

1. Calculate how many TXBYTECLKHS clock cycles can be reserved for LP interleaving during the appropriate blanking video mode gap.
2. Calculate the value of LP\_INTERLEAVING according to the results of Step 1.

Step 1:

- Scenario 1: The gap for interleaving starts and ends with a regular video stream HS packet.  

$$\text{ALLOWED\_HSBYTE\_CLOCKS\_FOR\_LP} = \text{BLANKING\_PERIOD} - (\text{EXIT\_HS\_MODE\_LATENCY} + \max\{\text{ENTER\_HS\_MODE\_LATENCY}, 2\} + 1)$$
- Scenario 2: The gap for interleaving starts with a regular video stream HS packet and ends in LPS.  

$$\text{ALLOWED\_HSBYTE\_CLOCKS\_FOR\_LP} = \text{BLANKING\_PERIOD} - (\text{EXIT\_HS\_MODE\_LATENCY} + 1)$$
- Scenario 3: The gap for interleaving starts with the LPS and ends with a regular video stream HS packet.

$$\text{ALLOWED\_HSBYTE\_CLOCKS\_FOR\_LP} = \text{BLANKING\_PERIOD} - (\max\{\text{ENTER\_HS\_MODE\_LATENCY}, 2\} + 1)$$

- Scenario 4: The gap for interleaving starts with the LPS and ends with a regular video stream HS packet.

$$\text{ALLOWED\_HSBYTE\_CLOCKS\_FOR\_LP} = \text{BLANKING\_PERIOD} - 1$$

After finishing Step 1, the time period available for LP interleaving is known:

$$T_{lp\_available} = \text{ALLOWED\_HSBYTE\_CLOCKS\_FOR\_LP} * T_{TXBYTECLKHS}$$

Step 2:

The resulting value must be programmed in the appropriate video mode register for LP interleaving.

$$\text{LP\_INTERLEAVING} < \left[ \frac{T_{lp\_available} - 8 * T_{hsbyte\_clk} - 5 * T_{dsif\_clk} - 26}{T_{txclkesc}} \right]_{16}$$

dsi-E124

$T_{hsbyte\_clk}$ : Period of TXBYTECLKHS clock of DSI\_PHY module

$T_{dsif\_clk}$ : Period of DSI functional clock

$T_{txclkesc}$ : Period of LP transmit escape clock

#### 10.3.4.4.4 Clock Requirements

The serial clock generated by the DSI host and sent to the display can be a continuous clock. The clock lane supports clock transmission even if there is no data to send for displays that require continuous clock. It is software-programmed through the [DSI\\_CLK\\_CTRL\[13\] DDR\\_CLK\\_ALWAYS\\_ON](#) bit: This bit can be programmed only when the interface is disabled (the [DSI\\_CTRL\[0\] IF\\_EN](#) bit set to 0).

The peripheral can use two different kinds of clocks. The first one is the DDR clock provided on the clock lane. The second clock is some transitions on the data lane 1 even if there is no valid data to send using LP mode.

The LP clock (TxClkEsc) frequency provided to the DSI complex I/O is in the range of 67 percent to 150 percent of the peripheral low-power (LP) clock frequency. It is generated internally by the DSI protocol engine module using the DSI functional clock. The DSI functional clock is divided by 1, 2, 3, up to 8191 using the value programmed in the [DSI\\_CLK\\_CTRL\[12:0\] LP\\_CLK\\_DIVISOR](#) bit field. The LP clock generated from the DSI functional clock must be in the range of 20 MHz to 32 kHz. The duty cycle must be 50/50 (minimum high time = 24ns). LP clock frequency visible on the pads (DP x or DN) is half the frequency of TxClkEsc.

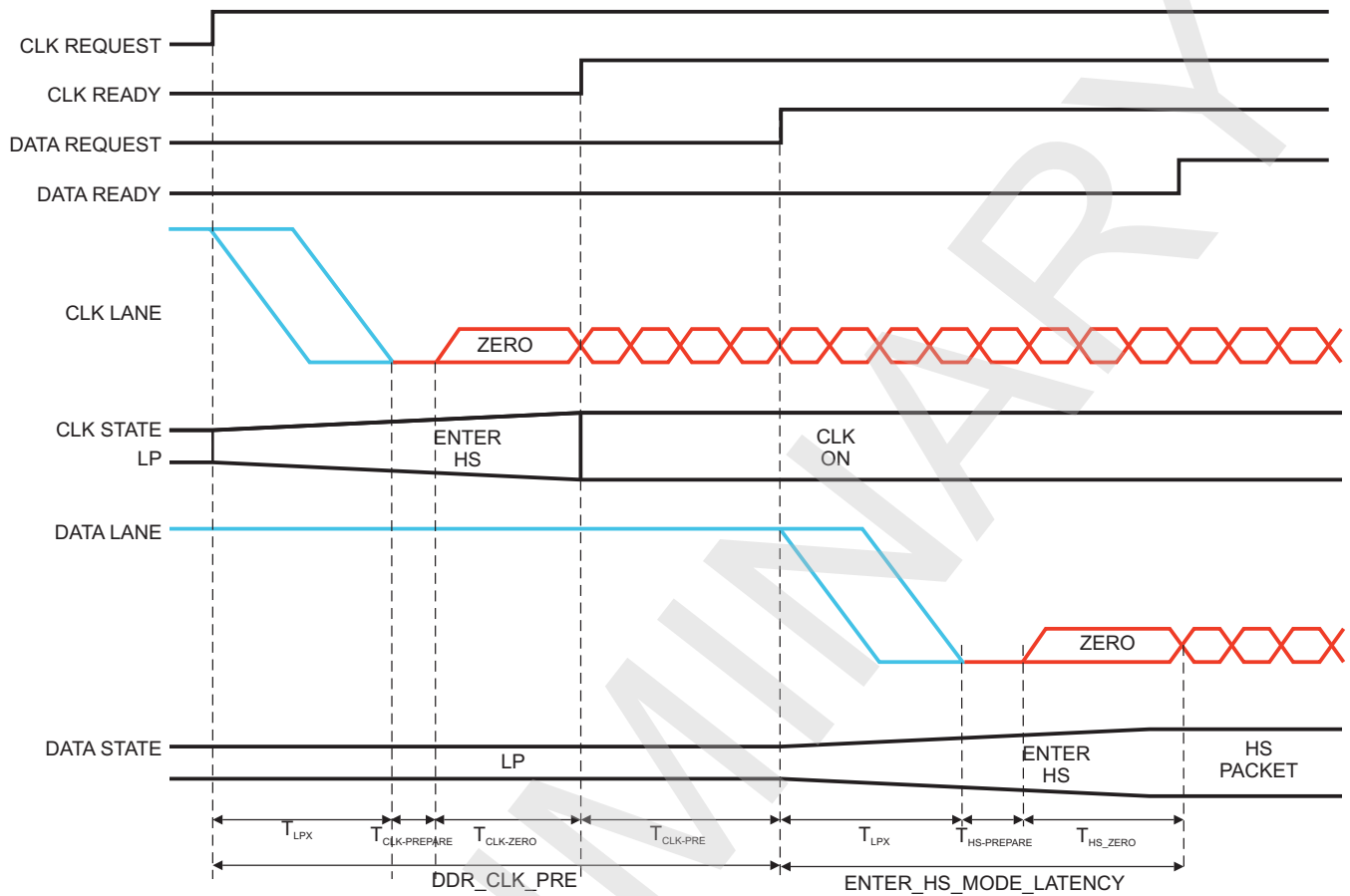
The [DSI\\_CLK\\_CTRL\[20\] LP\\_CLK\\_ENABLE](#) bit is used to enable or disable the clock. When disabled, the value of the [DSI\\_CLK\\_CTRL\[12:0\] LP\\_CLK\\_DIVISOR](#) bit field is ignored and does not have to be programmed by software users.

##### 10.3.4.4.4.1 Timing Parameters for an LP to HS Transaction

[Figure 10-127](#) shows the timing requirement when switching the data and clock lane state from LP to HS.

[Table 10-478](#) lists the LP to HS timing parameters.

Figure 10-127. LP to HS Timing



dsi-325

Table 10-478. LP to HS Timing Parameters

Timing	Description	Register
$T_{LPX}$	Length of any LPS period. The value set in the <a href="#">DSI_PHY_REGISTER1[20:16]</a> REG_TLPXBY2 bit field is half of the $T_{LPX}$ .	<a href="#">DSI_PHY_REGISTER1[20:16]</a> REG_TLPXBY2
$T_{CLK-PREPARE}$	Time to drive the CLK lane to LP-00 state, to prepare for HS clock transmission	<a href="#">DSI_PHY_REGISTER2[7:0]</a> REG_TCLKPREPARE
$T_{CLK-ZERO}$	Time to drive the CLK lane to HS-0 state before starting the clock	<a href="#">DSI_PHY_REGISTER1[7:0]</a> REG_TCLKZERO
$T_{CLK-PRE}$	Time that the HS clock must be driven before any associated data lane begins the transition from LP to HS mode $T_{CLK-PRE} = DDR\_CLK\_PRE - T_{LPX} - T_{CLK-PREPARE} - T_{CLK-ZERO}$	
$T_{HS-PREPARE}$	Time to drive the data lane to LP-00 state, to prepare for HS packet transmission	<a href="#">DSI_PHY_REGISTER0[31:24]</a> REG_THSPREPARE
$T_{HS-ZERO}$	Time to drive the data lane to HS-0 state before the synchronous sequence. $T_{HS-ZERO} = DSI\_PHY\_REGISTER0[23:16]$ THSPRPR_THSZERO - $T_{HS-PREPARE}$	<a href="#">DSI_PHY_REGISTER0[23:16]</a> REG_THSPRPR_THSZERO
$DDR\_CLK\_PRE$	Time between the CLK lane request assertion and the data request assertion to switch the data lanes to HS	<a href="#">DSI_CLK_TIMING[15:8]</a> DDR_CLK_PRE

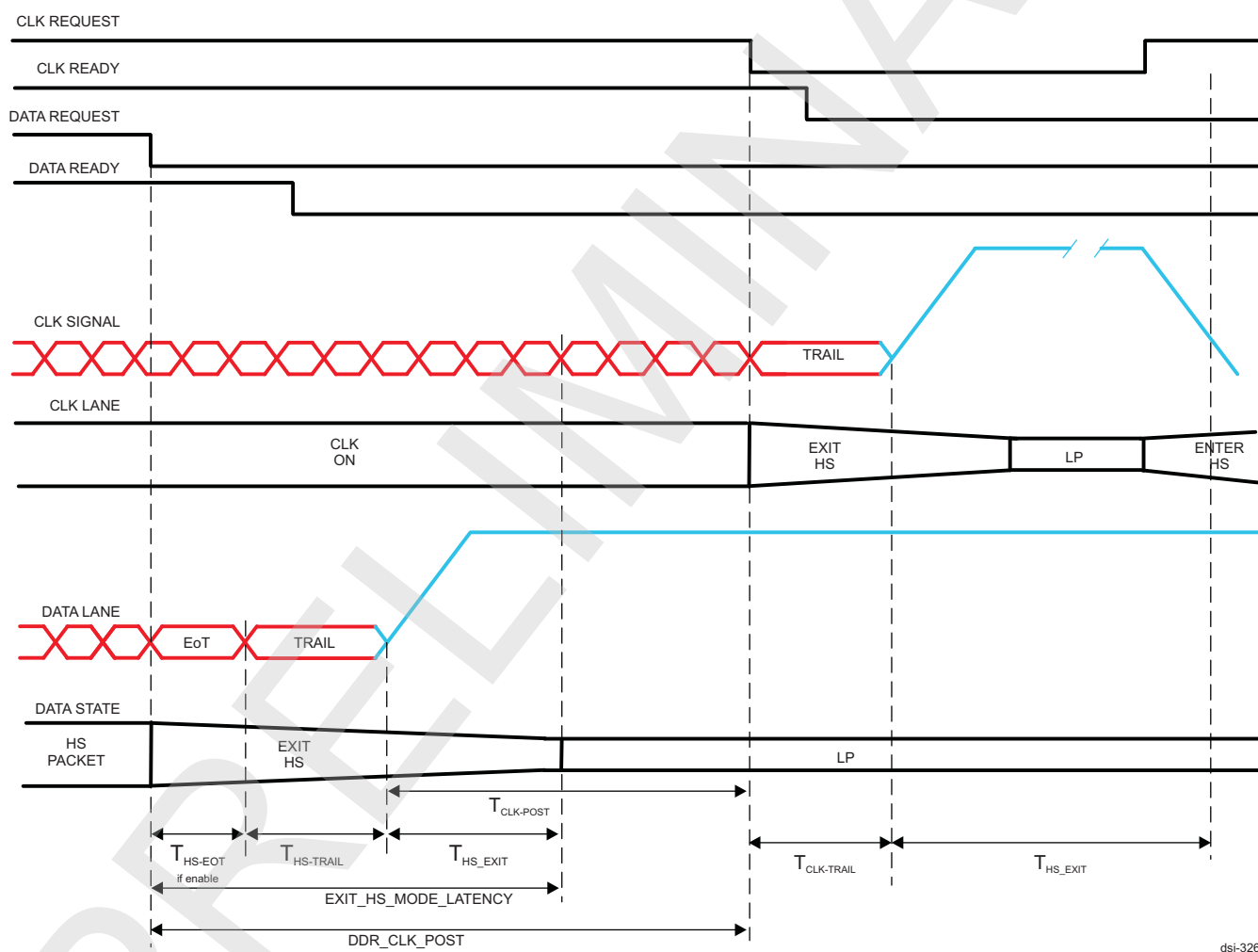
**Table 10-478. LP to HS Timing Parameters (continued)**

Timing	Description	Register
ENTER_HS_MODE_LATENCY	Time to enter into HS mode. It is critical that $\text{ENTER\_HS\_MODE\_LATENCY} = 1 + \text{DIVROUNDUP}(2 * \text{TLXBY2}, 4) + \text{DIVROUNDUP}(\text{THSPREPRE}, 4) + \text{DIVROUNDUP}(\text{THSPRPR\_THSZERO} - \text{THSPREPRE} + 3, 4)^{(1)}$	DSI_VM_TIMING7[31:16] ENTER_HS_MODE_LATENCY

<sup>(1)</sup> The formula  $\text{DIVROUNDUP}(\text{value}, \text{div})$  is equivalent to  $\text{ROUNDUP}(\text{value}/\text{div})$ .

#### 10.3.4.4.2 Timing Parameters for an HS to LP Transaction

Figure 10-128 shows the timing requirement when switching the state of the data and clock lanes from HS to LP. Table 10-479 lists the HS to LP timing parameters.

**Figure 10-128. HS to LP Timing****Table 10-479. HS to LP Timing Parameters**

Timing	Description	Register
$T_{\text{HS-EOT}}$	If EoT is enabled, a delay is added to EXIT_HS_MODE_LATENCY to send the EoT packet. The EoT period depends on the number of data lanes: 1 data lane = 4 DDR clocks 2 data lanes = 2 DDR clocks	

**Table 10-479. HS to LP Timing Parameters (continued)**

Timing	Description	Register
T <sub>HS-TRAIL</sub>	Time to drive flipped differential state after last payload data bit of a HS transmission burst	DSI_PHY_REGISTER0[15:8] REG_THSTRAIL
T <sub>HS-EXIT</sub>	Time to drive data lane to LP-11 state, after HS burst	DSI_PHY_REGISTER0[7:0] REG_THSEXIT
T <sub>CLK-POST</sub>	Time that the transmitter must continue sending HS clock after the last associated data lane transitions to LP mode	
T <sub>CLK-TRAIL</sub>	Time to drive HS differential state after last payload clock bit of a HS transmission burst	DSI_PHY_REGISTER1[15:8] EG_TCLKTRAIL
DDR_CLK_POST	Time between the data lane request deassertion and the CLK request deassertion to switch the data lanes into LP mode. The DDR_CLK_POST value must follow the rule: DDR_CLK_POST ≥ T <sub>HS-TRAIL</sub> + T <sub>HS-EXIT</sub> + T <sub>CLK-POST</sub>	DSI_CLK_TIMING[7:0] DDR_CLK_POST
EXIT_HS_MODE_LATENCY	Time to exit in HS mode. It is critical that EXIT_HS_MODE_LATENCY = DIVROUNDUP((T <sub>HS-TRAIL</sub> + T <sub>HS-EXIT</sub> ), 4) + 1 + T <sub>HS-EOT</sub> <sup>(1)</sup>	DSI_VM_TIMING7[15:0] EXIT_HS_MODE_LATENCY

<sup>(1)</sup> The formula DIVROUNDUP(value, div) is equivalent to ROUNDUP(value/div).

**10.3.4.4.3 Extra LP Transitions**

Some DSI receivers require extra clock cycles in LP mode to process the data. The DSI protocol engine can be programmed to send automatically one NULL long packet. It applies only when no more data are ready to be sent from the internal FIFO to the peripheral on the last LS transfer. The same value is used for all the VCs sending packets in LS mode.

The size of the payload is defined by the DSI\_CLK\_CTRL[17:16] LP\_CLK\_NULL\_PACKET\_SIZE bit field. The header value depends on the size of the payload as described in Table 10-480 and Table 10-481.

**NOTE:** In Table 10-480 and Table 10-481, ECC and checksum are enabled.

**Table 10-480. Extra NULL PH**

VC ID	Payload size (DSI_CLK_CTRL[17:16] LP_CLK_NULL_PACKET_SIZE)	Header (1st Byte)	Header (2nd Byte): WC LSB	Header (3rd Byte): WC MSB	Header (ECC)
0x0	0	0x9	0x0	0x0	0x9
	1		0x1		0x13
	2		0x2		0x2F
	3		0x3		0x35
0x1	0	0x49	0x0		0x1F
	1		0x1		0x05
	2		0x2		0x39
	3		0x3		0x23
0x2	0	0x89	0x0		0x10
	1		0x1		0x0A
	2		0x2		0x36
	3		0x3		0x2C
0x3	0	0xC9	0x0		0x06
	1		0x1		0x1C
	2		0x2		0x20
	3		0x3		0x3A

**Table 10-481. Extra NULL Packet Payload**

Payload size (DSI_CLK_CTRL[17:16] LP_CLK_NULL_PACKET_ SIZE)	Payload (1st byte)	Payload (2nd byte)	Payload (3rd byte)	Payload (CRC) LSB	Payload (CRC) MSB
0	NA <sup>(1)</sup>	NA	NA	0xFF	0xFF
1	0	NA	NA	0x87	0x0F
2	0	0	NA	0xB8	0xF0
3	0	0	0	0x33	0x39

<sup>(1)</sup> N/A = Not available

#### 10.3.4.4.5 Video Port Interface

**NOTE:** The signals described in this section are internal and not bounded outside the device. This section describes the internal connections between the DISPC and the DSI protocol engine.

Table 10-482 and Table 10-483 summarize the video interface signals. DSI1 has two video ports (VP1 and VP2) and DSI2 has one video port (VP1). These interfaces are used to connect the DISPC outputs to the DSI protocol engine to send real-time data streams. For video mode, only the active matrix timings are supported by DSI protocol engine.

When the two video ports of DSI1 are used for two video mode transfers, only the synchronization signals on video port 1 are used by the DSI protocol engine. The synchronization signals on video port 2 are ignored. When only one video mode is enabled, it must be connected to video port 1. It is not possible to enable video mode on video port 2 without enabling video mode on video 1 (dual-video mode). The HSYNC/VSYNC/DE/DATA signals are driven on the rising or falling edge of the pixel clock (VP\_PCLK).

**Table 10-482. Video Port Signals for DSI1 and DSI2 Protocol Engines**

Signal Name	Type <sup>(1)</sup>	Description
VP1_HSYNC	I	Horizontal synchronization signal
VP1_VSYNC	I	Vertical synchronization signal
VP1_DATA[23:0]	I	Parallel output data: Bits 0 to 23
VP1_PCLK	I	Pixel clock. In case of STALL configuration, it is used to indicate when new data is on the data bus during the clock period of VP_CLK.
VP1_DE	I	Data enable
VP1_STALL	O	The stall signal must be deasserted to receive pixel and asserted to stop receiving pixel. (It can be used only while the DISPC is configured in STALL mode; in that mode, HSYNC and VSYNC are not generated).
VP1_CLK	I	DISPC internal clock. It is a free-running clock. The VP_CLK is a divided clock from VP_PCLK.

<sup>(1)</sup> I = Input; O = Output

**Table 10-483. Video Port Signals for DSI1 Protocol Engine**

Signal Name	Type <sup>(1)</sup>	Description
VP2_HSYNC	I	Horizontal synchronization signal
VP2_VSYNC	I	Vertical synchronization signal
VP2_DATA[23:0]	I	Parallel output data: Bits 0 to 23
VP2_PCLK	I	Pixel clock. In case of STALL configuration, it is used to indicate when new data is on the data bus during the clock period of VP_CLK.
VP2_DE	I	Data enable
VP2_STALL	O	The stall signal must be deasserted to receive pixel and asserted to stop receiving pixel. (It can be used only while the DISPC is configured in STALL mode; in that mode, HSYNC and VSYNC are not generated).
VP2_CLK	I	DISPC internal clock. It is a free-running clock. The VP_CLK is a divided clock from VP_PCLK.

<sup>(1)</sup> I = Input; O = Output

**NOTE:**

- The polarities of VP\_HSYNC and VP\_VSYNC are programmable by setting the [DSI\\_CTRL](#) and [DSI\\_CTRL2](#) registers.
- Clocks VP\_CLK and VP\_PCLK can have the same frequency.
- The number of bits to be captured on the video port (the width of the data bus) is defined in the [DSI\\_CTRL\[7:6\]](#) VP\_DATA\_BUS\_WIDTH bit field for DSI1 and the [DSI\\_CTRL2\[7:6\]](#) VP\_DATA\_BUS\_WIDTH for DSI2. The supported formats are 16, 18, and 24 bits.
- For the 18-bit pixel format using 24 bits on the DSI link, the format to be used by the DSI protocol engine is 24-bit because the DISPC must convert the 18-bit pixels into 24-bit pixels. There is no logic inside the DSI protocol engine to convert from 18-bit format to 24-bit format.

The data received on the video port can be stored in the line buffer memories or sent directly on the DSI interface in two cases:

- The line buffer size is too small compared to the line from the DISPC.
- There is no line buffer instantiated. If there is no line buffer, the burst mode, defined as frequency burst mode, cannot be used. Only transparency burst mode is supported.

**NOTE:** The [DSI\\_CTRL\[13:12\]](#) LINE\_BUFFER and [DSI\\_CTRL2\[13:12\]](#) LINE\_BUFFER bit fields define the number of lines to be used for transferring data from the video port to the DSI1 and DSI2 link.

**10.3.4.4.5.1 Video Port Used for Video Mode**

If the video port is used for video mode, VP\_STALL is not used. [Table 10-484](#) and [Table 10-485](#) list the active signals on the video port.

**Table 10-484. Video Port 1 for DSI1 and DSI2 in the Context of Video Mode**

Signal Name	Type <sup>(1)</sup>	Description
VP1_HSYNC	I	Horizontal synchronization signal
VP1_VSYNC	I	Vertical synchronization signal
VP1_DATA[23:0]	I	Parallel output data: Bits 0 to 23
VP1_PCLK	I	Pixel clock
VP1_DE	I	Data enable
VP1_CLK	I	It is a free-running clock used as the DISPC functional clock.

<sup>(1)</sup> I = Input; O = Output

**Table 10-485. Video Port 2 for DSI1 in the Context of Video Mode**

Signal Name	Type <sup>(1)</sup>	Description
VP2_HSYNC	I	Horizontal synchronization signal
VP2_VSYNC	I	Vertical synchronization signal
VP2_DATA[23:0]	I	Parallel output data: Bits 0 to 23
VP2_PCLK	I	Pixel clock
VP2_DE	I	Data enable
VP2_CLK	I	It is a free-running clock used as the DISPC functional clock.

<sup>(1)</sup> I = Input; O = Output

Three video modes are available:

- No-line buffer: The data received on the video port are directly outputted to the DSI port without buffering. The ratio of VP\_PCLK and the DSI HS clock period must ensure the same throughput on the two ports (the two clocks must be generated using the same PLL; the subsystem must provide this configuration).

- One-line buffer:
  - The data can be transferred as described in the no-line buffer configuration.
  - The data are first stored in the line buffer; and then when all the data for one line are received, the DSI protocol engine sends the entire line. Software must adjust timings to let enough time for storing all line data into the line buffer before sending to DSI outputs. The synchronization packets are never stored in the line buffer.
- Two-line buffers:
  - The data can be transferred as described in the one-line buffer configuration.
  - One line is stored when the second line is output on the DSI port. This allows burst capability. While receiving the first line of the frame, no RGB packets are sent, because the line buffers are empty. To send the last line of the frame, a dummy line must be provided by the display controller to flush the line buffers. The synchronization packets are never stored in the line buffer.

---

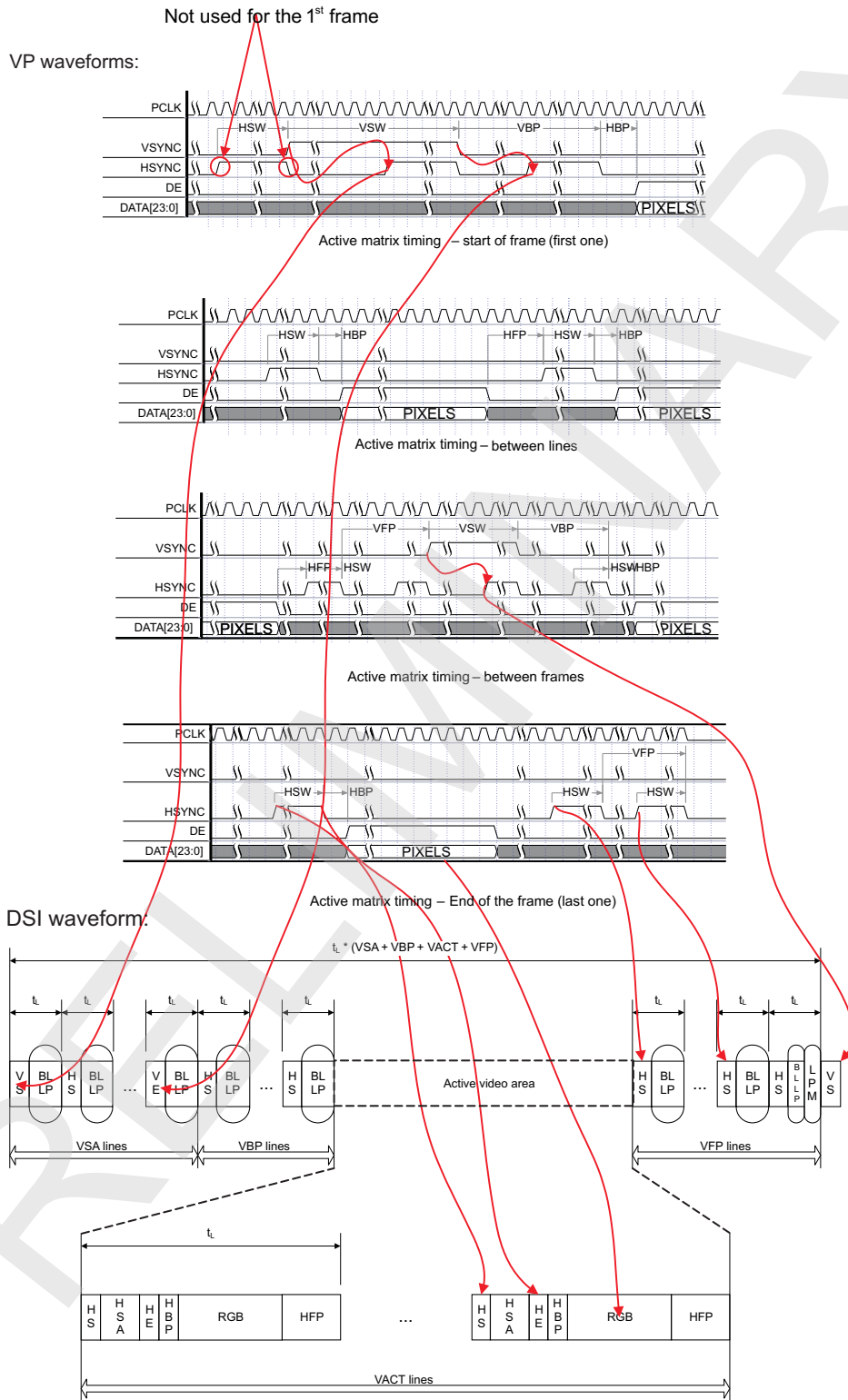
**NOTE:** If more active lines are received on the video port than the number of lines defined in the [DSI\\_VM\\_TIMING3\[15:0\]](#) VACT bit field, the extra lines are discarded by the DSI protocol engine. These lines are treated as blanking lines.

---

[Figure 10-129](#) through [Figure 10-131](#) show these three video modes.



Figure 10-129. DSI Video Mode Without Burst (No-Line Buffer)



**Figure 10-130. DSI Video Mode Without Burst (One-Line Buffer)**

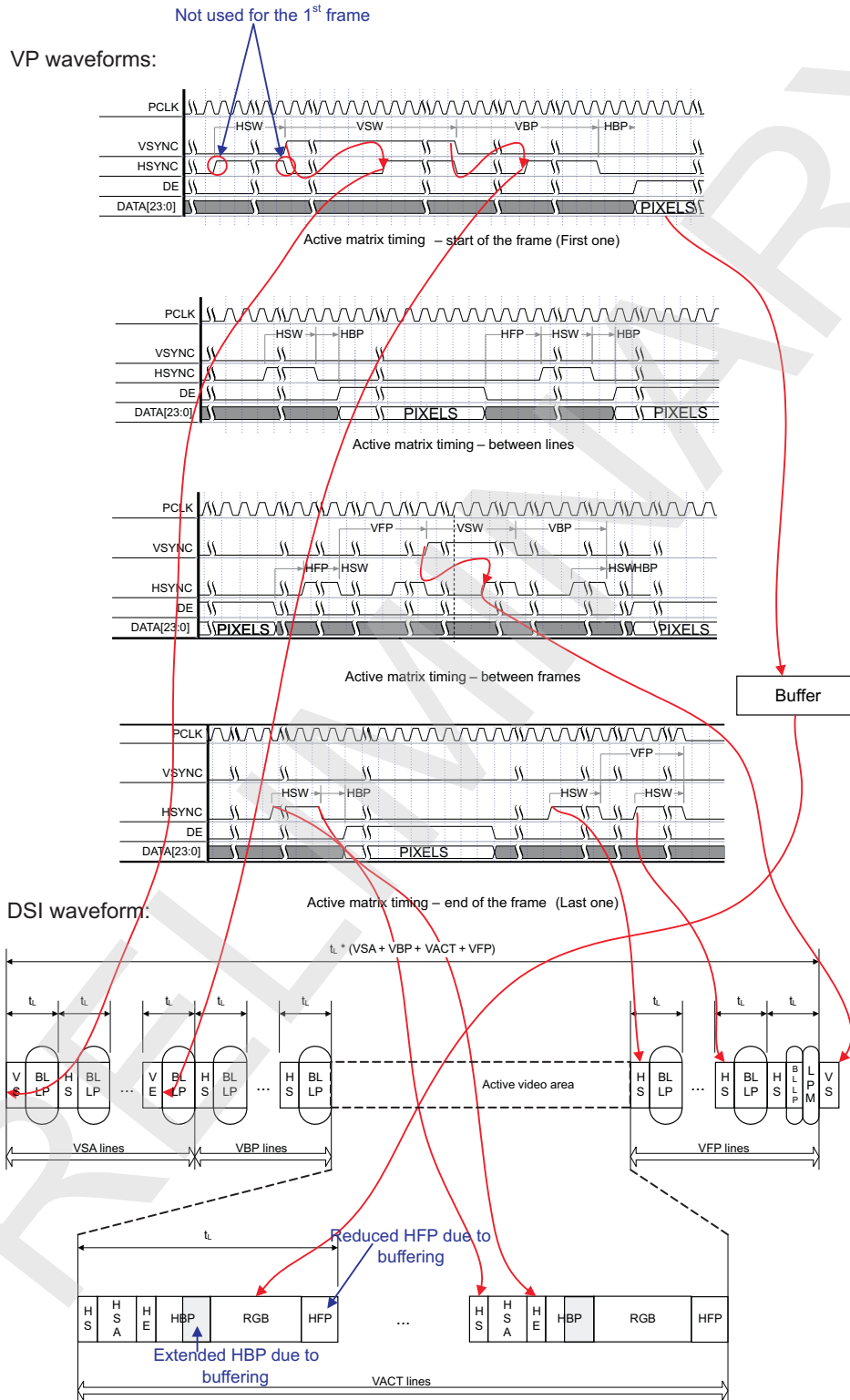
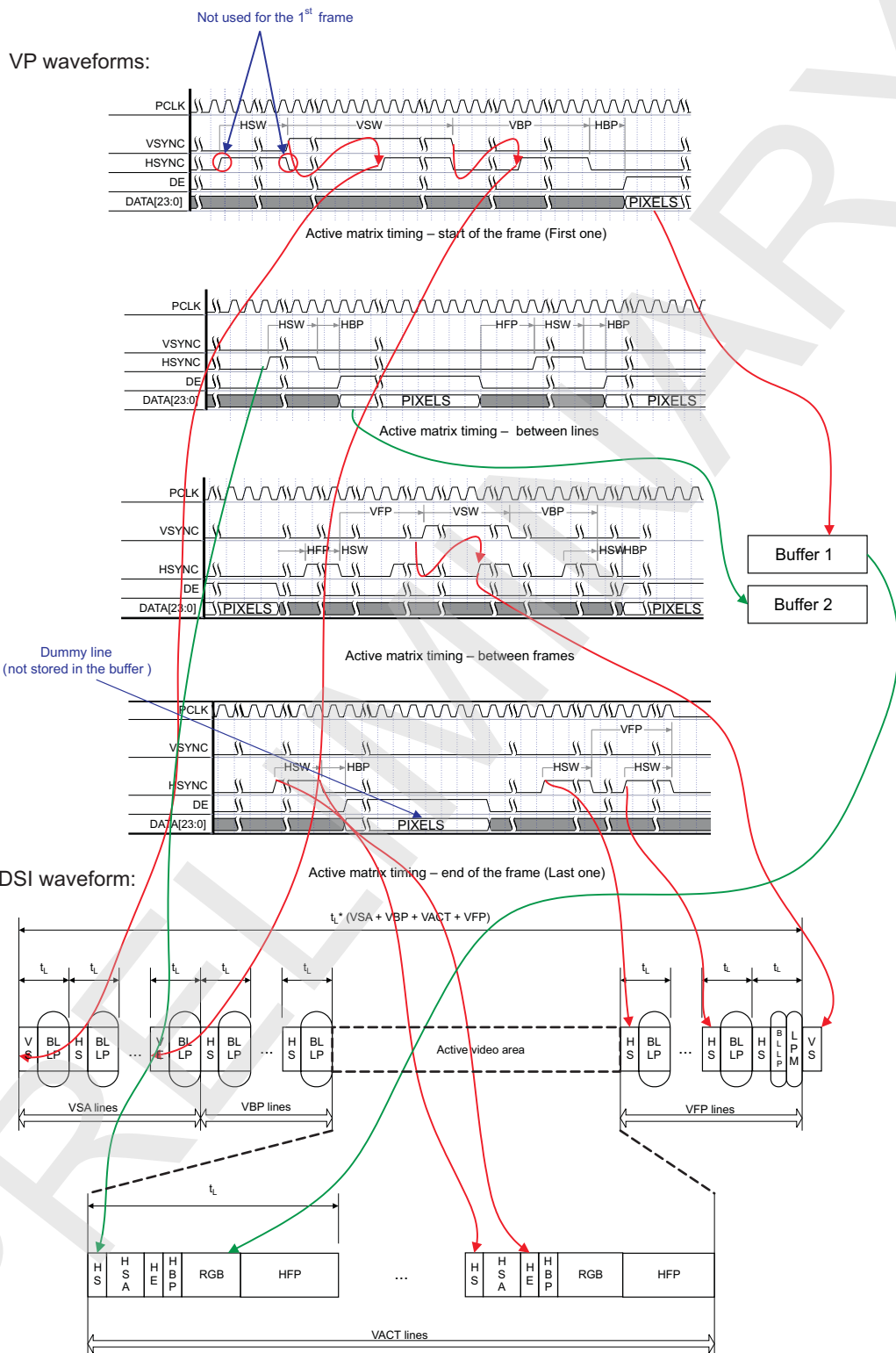


Figure 10-131. DSI Video Mode With Burst (Two-Line Buffers)



dsi-037

**NOTE:** In [Figure 10-129](#) through [Figure 10-131](#):

- When HSYNC start and HSYNC end short packets are not generated (HSA does not exist), the HBP signal must not be 0.
- Software must ensure that VBP is always defined so that there is at least one HSYNC during VBP.
- In blanking low-power mode (BL-LP), two options are possible:
  - The lane remains in ULPS, and the [DSI\\_CTRL\[20\]](#) BLANKING\_MODE bit is set to 0x0.
  - Dummy bytes are sent during LP with the [DSI\\_CTRL\[20\]](#) BLANKING\_MODE bit set to 0x1; the number of sent bytes is determined by the [DSI\\_VM\\_TIMING6\[15:0\]](#) BL\_LP\_INTERLEAVING bit field.

If the signal VP\_DE is not asserted during enough VP\_PCLK cycles to be able to capture the number of bytes defined in the word count of the header, the module must send the valid data received on the video port followed by bytes of 0s to match the required number of bytes to transmit. The VP\_PCLK must be present during all extra cycles where the DSI protocol engine is expecting pixels.

If the VP\_DE signal is asserted for too many VP\_PCLK cycles, the module stops capturing the data on the video port while the number of bytes to capture, as defined in the word count field of the header, is reached.

The HS checks that the received synchronization events on the video port (VSYNC and HSYNC) are within the synchronization window based on expected timings. If the timings (internal and received) are out of sync, the interrupt for out-of-sync is generated and the interface is disabled (the [DSI\\_CTRL\[0\]](#) IF\_EN bit is automatically reset by hardware). The unsynchronization window is defined by the [DSI\\_VM\\_TIMING2\[27:24\]](#) WINDOW\_SYNC bit field.

#### 10.3.4.4.5.2 Video Port Used for Command Mode

If the video port is used for command mode, the VP\_HSYNC, VP\_VSYNC, and VP\_DE signals are not used. [Table 10-486](#) and [Table 10-487](#) describe the active signals on video port 1 and video port 2, respectively.

**Table 10-486. Video Port 1 for DSI1 and DSI2 in the Context of Command Mode**

Signal Name	Type <sup>(1)</sup>	Description
VP1_DATA[23:0]	I	Parallel output data: Bits 0 to 23
VP1_PCLK	I	One pulse is generated every time new data is output on the data bus.
VP1_STALL	O	The stall signal must be deasserted to receive pixel and asserted to stop receiving pixel. (It can be used only while the DISPC is configured in stall mode; in that mode, HSYNC and VSYNC are not generated).
VP1_CLK	I	DISPC internal clock: It is a free-running clock used as the DISPC functional clock.

<sup>(1)</sup> I = Input; O = Output

**Table 10-487. Video Port 2 for DSI1 in the Context of Command Mode**

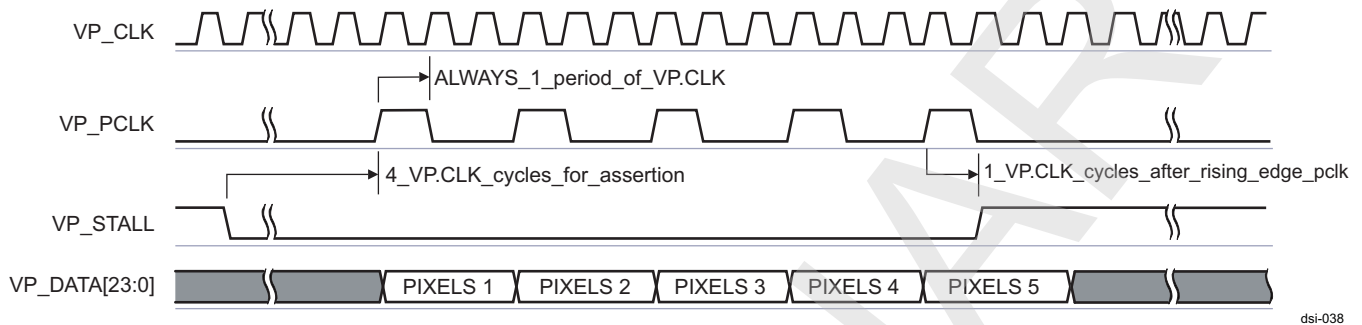
Signal Name	Type <sup>(1)</sup>	Description
VP2_DATA[23:0]	I	Parallel output data: Bits 0 to 23
VP2_PCLK	I	One pulse is generated every time new data is output on the data bus.
VP2_STALL	O	The stall signal must be deasserted to receive pixel and asserted to stop receiving pixel. (It can be used only while the DISPC is configured in stall mode; in that mode, HSYNC and VSYNC are not generated).
VP2_CLK	I	DISPC internal clock: It is a free running-clock used as the DISPC functional clock.

<sup>(1)</sup> I = Input; O = Output

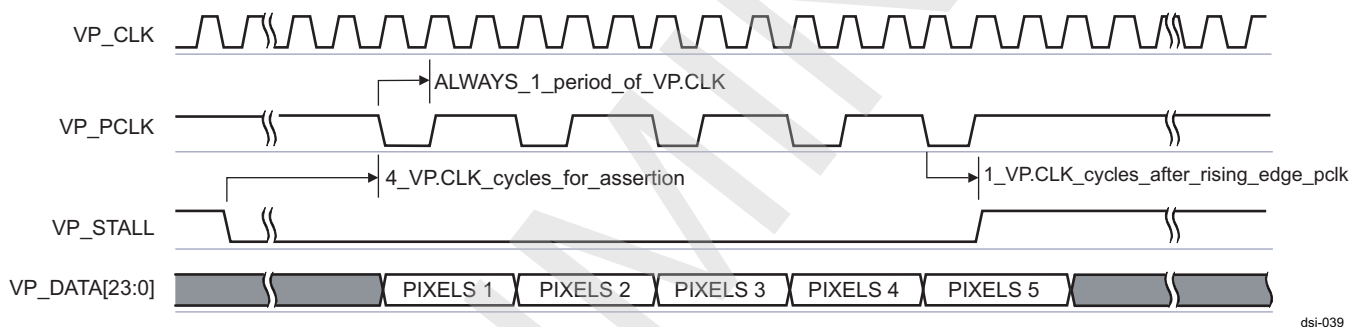
**NOTE:** The stall signal must be deasserted to receive pixels and asserted to stop receiving pixels.

Figure 10-132 and Figure 10-133 show the VP\_STALL signal assertion and deassertion on rising and falling edges, respectively.

**Figure 10-132. Stall Timing With Pixel on Rising Edge**



**Figure 10-133. Stall Timing With Pixel on Falling Edge**



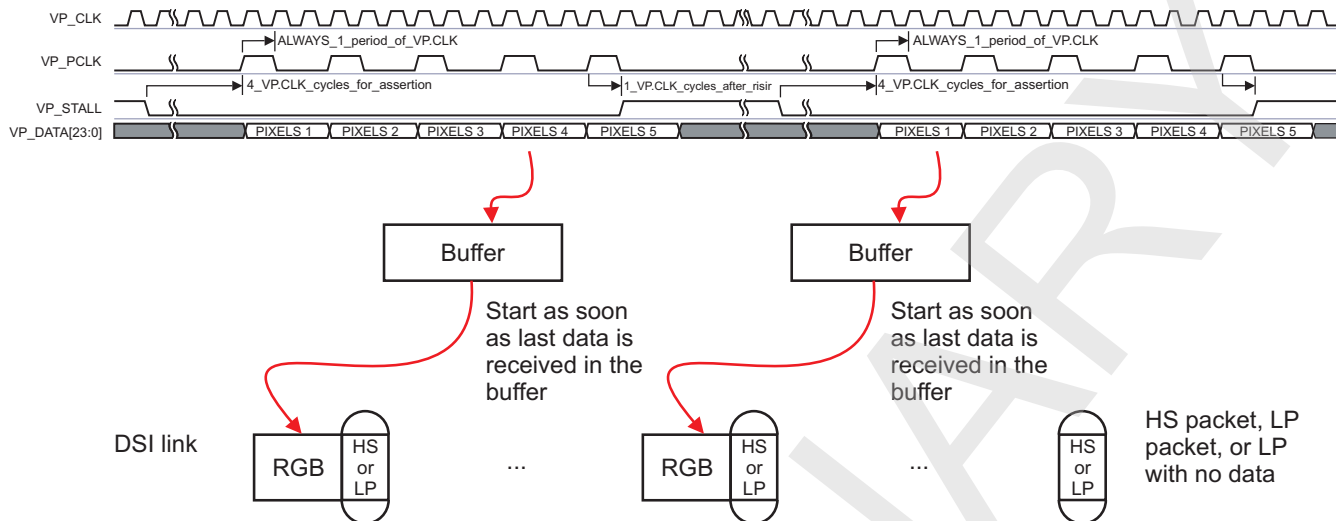
The VP\_STALL signal is asserted to stop the transfer when the last data is output. The data can be output on the rising or falling edge of the VP\_PCLK through registers in the DISPC module. Data output on the falling edge of VP\_PCLK is supported by the DSI protocol engine.

The VP\_PCLK clock is generated from VP\_CLK; these two clocks are balanced. Assertion and deassertion of VP\_PCLK is done on the rising edge of VP\_CLK. The width of the VP\_PCLK pulse depends on the configuration of the clock divisor in the DISPC. For more information, see Section 10.2, *Display Controller*. In the DSI protocol engine, the information is defined in the DSI\_CTRL[4] VP\_CLK\_RATIO bit and must be aligned with the DISPC configuration.

Deassertion of the VP\_STALL signal occurs at least four VP\_CLK cycles before assertion of VP\_PCLK. Assertion of VP\_STALL occurs one cycle VP\_CLK after deassertion of VP\_PCLK for the last pixel to be transferred. The VP\_CLK clock is generated by the DISPC under software control. It can be kept running between assertion and deassertion of VP\_STALL.

The stall assertion/deassertion depends on the number of bytes to be received considering the size of the video port bus defined in the DSI\_CTRL[7:6] VP\_DATA\_BUS\_WIDTH bit field for video port 1 and the DSI\_CTRL2[7:6] VP\_DATA\_BUS\_WIDTH bit field for video port 2.

Figure 10-134 shows the data flow in command mode using the video port.

**Figure 10-134. Data Flow in Command Mode Using the Video Port**

dsii-040

Two command modes are available:

- One-line buffer: The data are stored in the line buffer before being sent.
- Two-line buffers: The two lines are used if the word count defined in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register is bigger than the line size; otherwise, one-line buffer must be used.

---

**NOTE:** In command mode, the video port can be used only in one- or two-line buffer configuration. No-line buffer configuration is not allowed.

---

The packets can be sent using high speed or low speed.

The DCS command in the payload can be inserted automatically using the [DSI\\_VC\\_CTRL\\_i\[30\]](#) [DCS\\_CMD\\_ENABLE](#) bit. If TE is used, hardware automatically inserts the DCS Write Start command for the first packet of the frame transfer and DCS Write Continue command for all subsequent packets. The [DSI\\_VC\\_CTRL\\_i\[31\]](#) [DCS\\_CMD\\_CODE](#) bit is ignored.

#### 10.3.4.4.5.3 Burst Mode

When burst mode is enabled, the video port receives data from the DISPC at the pixel clock. The DSI protocol engine buffers the data in its own line FIFO (double-line buffer size). The read speed of the line can be twice the pixel clock to increase the blanking time of the video mode and to allow command mode traffic to be interleaved during the blanking period. Burst mode uses a dual-line buffer.

The DSI port can output data from one line buffer while the second line buffer is accessed by the video port. The two processes are concurrent but they do not access the same line at the same time. The DSI transfer can start only when the entire video port line is transferred into a line buffer. The switch is controlled by the VP\_HS signal on the video port side and by an internal signal on the DSI port indicating that the last data for the current line has been written into the line buffer.

---

**NOTE:** The line buffers are used to store only the pixels. The synchronization codes are not stored in the line buffers; they must be sent according to the video port timings.

---



---

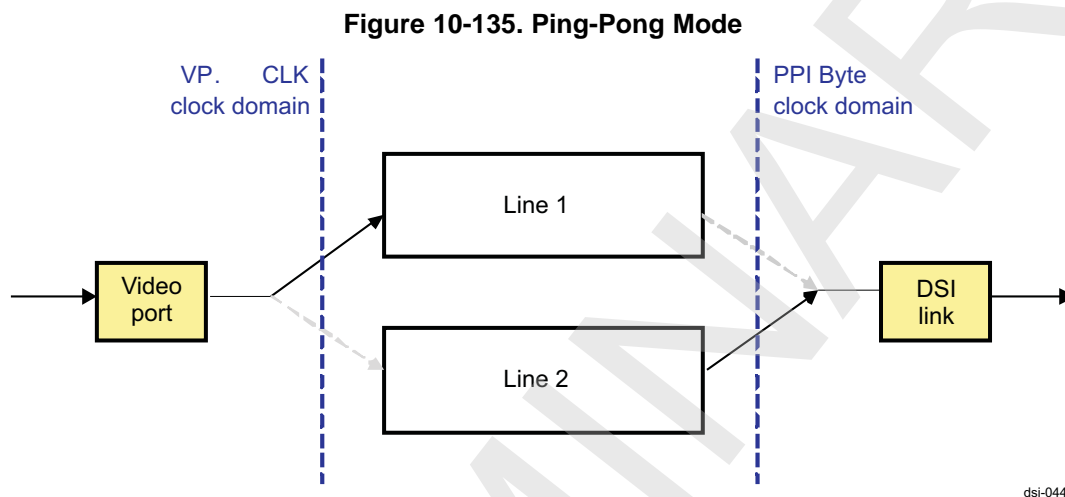
**NOTE:** The number of line buffers to be used while receiving data on the video port can be selected by setting the [DSI\\_CTRL\[13:12\]](#) [LINE\\_BUFFER](#) bit field.

---

#### 10.3.4.4.5.4 Ping-Pong Buffer

The ping-pong buffer stores data incoming from the DISPC in one of the two line buffers. When one line buffer is used to store the pixels received on the video port, the second line buffer is used to provide data on the DSI link. The ping-pong buffer is supported in command mode, provided the size of the packet defined in the header register is less than the size of each line buffer (768 x 32 bits for VP1 of DSI2 and VP2 of DSI1, and 960 x 32 bits for VP1 of DSI1). If the size of the packet is greater than the size of the line buffer, the ping-pong mechanism cannot be used (both lines are used as a single line).

Figure 10-135 shows the video port used in ping-pong mode .



The ping-pong buffer status can be checked by the [DSI\\_VC\\_CTRL\\_i\[14\] PP\\_BUSY](#) bit.

- When PP\_BUSY equals 1, the ping-pong buffer is active and the line buffers are not ready to receive data; therefore, a new header cannot be updated.
- When PP\_BUSY equals 0, at least one line buffer is empty; therefore, a new header can be updated. PP\_BUSY is then set to 0x1. If both line buffers are empty, two headers can be written, one following the other. PP\_BUSY remains at 0x0 after the first header is written, and is set to 0x1 after second header is written.

An IRQ is available to update header on events. The IRQ is enabled by setting the [DSI\\_VC\\_IRQENABLE\\_i\[8\] PP\\_BUSY\\_CHANGE\\_IRQ](#) bit to 0x1, and its status is accessible on the [DSI\\_VC\\_IRQSTATUS\\_i\[8\] PP\\_BUSY\\_CHANGE\\_IRQ](#) bit.

#### 10.3.4.4.6 SCP Interface

The SCP interface is used to transfer register values from the DSI protocol engine to the DSI PLL control module and to the DSI complex I/O. It spends several cycles to serialize the data to be sent. Software users must consider the delay in processing the transfer of the data from/to the L3 Interconnect slave port to/from the module.

##### 10.3.4.4.6.1 Shadowing Register

The first three SCP registers for the DSI complex I/O address map must be implemented as shadow registers. The shadowing mechanism is enabled and disabled using the [DSI\\_COMPLEXIO\\_CFG1\[31\] SHADOWING](#) bit:

- When setting the [DSI\\_COMPLEXIO\\_CFG1\[31\] SHADOWING](#) bit to 1, the transfer of the values from the three first local interconnect port registers into the three first registers of the DSI complex I/O ([DSI\\_PHY\\_REGISTER0](#) through [DSI\\_PHY\\_REGISTER2](#)) is done only when the DISPC\_UPDATE\_SYNC signal from the DISPC is active and the [DSI\\_COMPLEXIO\\_CFG1\[30\] GOBIT](#) bit is set to 1. If there is no pending update for the three registers when the DISPC\_UPDATE\_SYNC signal is asserted, the [DSI\\_COMPLEXIO\\_CFG1\[30\] GOBIT](#) bit is reset by hardware and there is no SCP transfer.
  - If there is only one register to update, only the corresponding new value is transferred. The second

and third registers in the DSI complex I/O are not updated. When the transfer completes, the [DSI\\_COMPLEXIO\\_CFG1\[30\]](#) GOBIT bit is reset by hardware.

- If the two registers need to be updated, the register with the lower address is transferred first, followed by the second register. When the transfers complete, the [DSI\\_COMPLEXIO\\_CFG1\[30\]](#) GOBIT bit is reset by hardware.
- If the three registers need to be updated, the register with the lowest address is transferred first, and then the other registers are transferred according to incremental address. When the transfers complete, the GOBIT bit is reset by hardware.

When there is an on-going transfer (read or write) to any SCP register, the transfer must complete before updating the shadowing registers.

- When unsetting the [DSI\\_COMPLEXIO\\_CFG1\[31\]](#) SHADOWING bit to 0, if the transfer into the first two DSI complex I/O registers has already started, it should be finished.

---

**NOTE:** When reading the shadow registers, the local value stored in the DSI protocol engine is returned if the update is pending; otherwise, the values stored in the DSI complex I/O are returned.

---

#### 10.3.4.4.6.2 Busy Signal

The SCPBusy signal indicates there is still activity using the SCPClk clock provided by the PRCM module. The SCPClk clock is the devided DSS\_L3\_ICLK/4 clock.

#### 10.3.4.4.7 Timers

---

**NOTE:** Among the timers described in this section, only the HS TX, LP RX, and TurnRequests timers generate interrupts immediately when the timer value is null. The ForceTxStopMode timer ends counting instantly and ForceTxStopMode is not asserted.

---

#### 10.3.4.4.7.1 $T_{wakeUp}$ Timer

The  $T_{wakeUp}$  timer is not implemented in the DSI protocol engine. The general-purpose (GP) timer must be used to handle wakeup. This timer is used for exiting ULP status mode for the active lanes (clock and/or data lanes). The sequence to exit ULP state is:

1. Change the state of TxULPSExit for each lane to ACTIVE.
2. Wait for the interrupt that indicates all lanes with TxULPSExit ACTIVE have acknowledged by asserting ULPSActiveNot. This is done by reading the [DSI\\_COMPLEXIO\\_IRQSTATUS](#) register - ULPSACTIVENOT\_ALLi\_IRQ bit fields (where i = 0, 1).
3. Start the application wake-up timer (GP timer).
4. Wait for the time-out.
5. Change the TxUlpsClk signals to INACTIVE state for the clock lane and/or TxRequestEsc INACTIVE state for the data lane(s).

---

**NOTE:** The minimum time for the wake-up period is 1 ms.

---

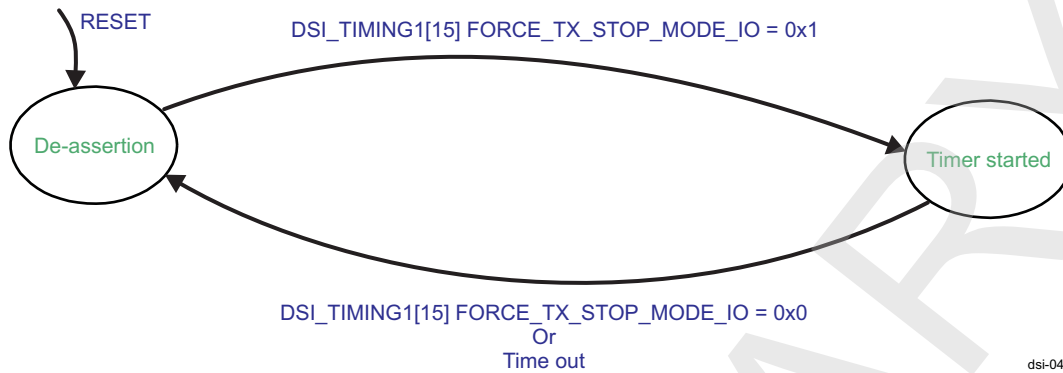
To enter ULPS mode for clock lane, TxUlpsClk state must be changed to ACTIVE state. To enter ULPS mode for data lane, TxRequestEsc state must be changed to ACTIVE state (also TxUlpsEsc, if it is not in ACTIVE state already).

#### 10.3.4.4.7.2 ForceTxStopMode FSM

The ForceTxStopMode signal is used at initialization time (DSI complex I/O). [Figure 10-136](#) shows the ForceTxStopMode finite state-machine (FSM) to assert and deassert the ForceTxStopMode signal.



**Figure 10-136. ForceTxStopMode FSM**



The DSI protocol engine asserts the ForceTxStopMode signal by setting the [DSI\\_TIMING1\[15\] FORCE\\_TX\\_STOP\\_MODE\\_IO](#) bit to 1. Asserting the [FORCE\\_TX\\_STOP\\_MODE\\_IO](#) bit allows the lanes to be initialized. The lanes are in STOP state when the ForceTxStopMode signal is high.

No data can be sent before the ForceTxStopMode signal is deasserted. The deassertion time is defined by the [STOP\\_STATE\\_COUNTER\\_IO](#), [STOP\\_STATE\\_X4\\_IO](#) and [STOP\\_STATE\\_X16\\_IO](#) bit fields in [DSI\\_TIMING1\[14:0\]](#) register. The [DSI\\_TIMING1\[15:0\] FORCE\\_TX\\_STOP\\_MODE\\_IO](#) bit is reset by hardware when the time is reached.

This bit can be reset by software.

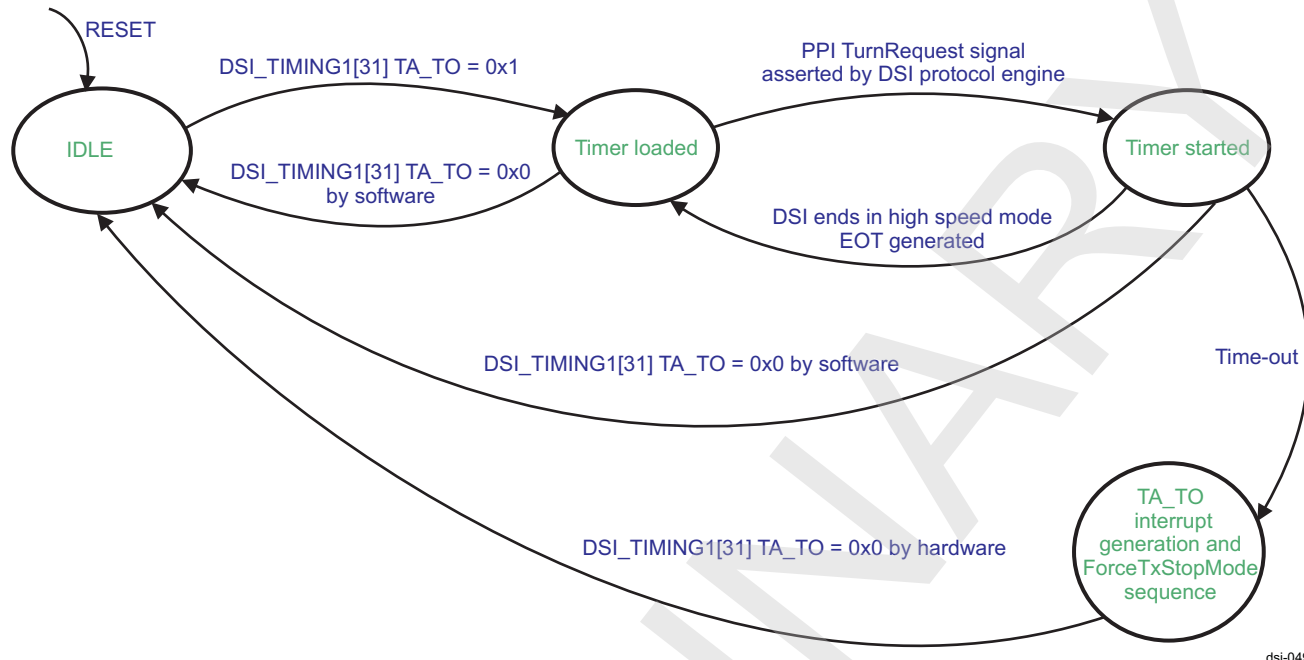
The calculation of the number of DSI\_CLK functional clock cycles assertion period is defined by:

$$\text{Total period in DSI_FCLK cycles} = \text{STOP\_STATE\_COUNTER\_IO} \times ((\text{STOP\_STATE\_X16\_IO} \times 15) + 1) \times ((\text{STOP\_STATE\_X4\_IO} \times 3)).$$

#### 10.3.4.4.7.3 TurnRequest FSM

The TurnRequest signal is used to request a turnaround. It is valid only for the data lane 1 because the other data lanes cannot be used in reverse direction to receive data from the DSI receiver. When the TurnRequest signal is asserted, the TA\_TO timer is started. [Figure 10-137](#) shows the TurnRequest FSM to assert and deassert the TurnRequest signal.

Figure 10-137. TurnRequest FSM



dsii-049

The DSI protocol engine asserts the TurnRequest signal during one TxClkEsc cycle when turnaround is enabled through the `DSI_VC_CTRL_i[6]` BTA\_EN bit (for more information, see Section 10.3.4.4.8, *Bus Turnaround*). The `DSI_TIMING1[31]` TA\_TO bit is set/reset by software to enable or disable the timer for turnaround procedure failure. It can be reset by software or automatically by hardware when the time-out occurs.

The timer is loaded with the value in the number of DSI\_FCLK cycles:

$$\text{DSI\_TIMING1}[28:16] \text{ TA\_TO\_COUNTER} \times ((\text{DSI\_TIMING1}[30] \text{ TA\_TO\_X16} \times 15) + 1) \times ((\text{DSI\_TIMING1}[29] \text{ TA\_TO\_X8} \times 7) + 1).$$

When the TA\_TO\_IRQ interrupt is generated (the turnaround timer expired and the procedure failed), hardware automatically resets the `DSI_TIMING1[31]` TA\_TO bit and asserts the ForceTXStopMode signal (see Section 10.3.4.4.7.2, *ForceTxStopMode FSM*) for the DSI\_PHY to drive the LP-11 STOP state. The ForceTXStopMode timer is used to define the minimum duration of the LP-11 state. The STOP state can be longer if there is no activity.

Hardware resets the `DSI_TIMING1[15]` FORCE\_TX\_STOP\_MODE\_IO bit, followed by an internal logic reset except for all register values and TX FIFO content, and then resets the `DSI_CTRL[0]` IF\_EN bit. Software must take action to recover (for example, by resetting the peripheral if it is not responding). It must wait for the `DSI_CTRL[0]` IF\_EN bit to be reset to 0 before starting the recovery sequence.

#### 10.3.4.4.7.4 Peripheral Reset Timer

The peripheral reset timer is not implemented in the DSI protocol engine module. Instead, a GP timer must be used in case of reset of the peripheral to determine when the peripheral is ready to operate again.

#### 10.3.4.4.7.5 HS TX Timer

The HS TX timer is used to detect when the host has been in TX mode for too long. When time-out occurs, the EOT is forced. The timer is reloaded when a start of HS transmission occurs. It is enabled and disabled by software through the `DSI_TIMING2[31]` HS\_TX\_TO bit. The interrupt HS\_TX\_TO\_IRQ is generated when the timer expires. The `DSI_IRQSTATUS[14]` HS\_TX\_TO\_IRQ bit is set to 1 when the HS TX time-out occurs.

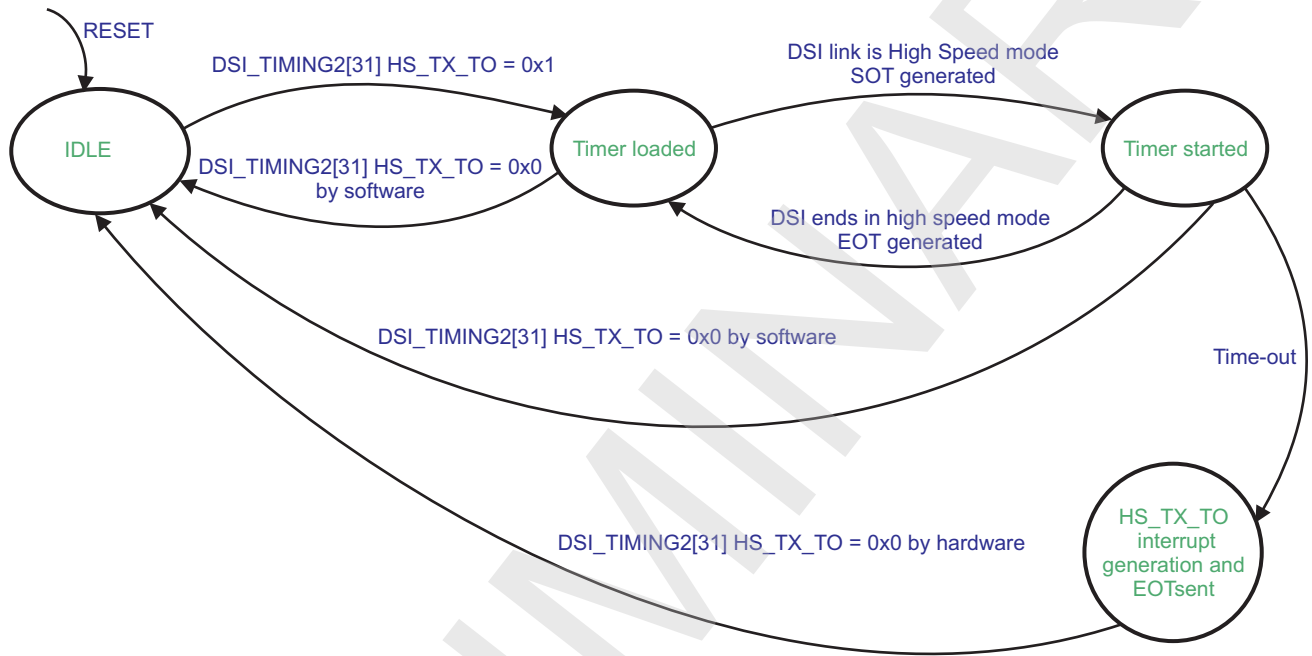
The maximum time supported is 20 ms. It can be used to determine that at least once per frame in video mode, the HS mode is stopped to enter ULPS.

The timer is loaded with the value in the number of TXBYTECLKHS:

$$DSI\_TIMING2[28:16] HS\_TX\_TO\_COUNTER \times (1 + (DSI\_TIMING2[30] HS\_TX\_TO\_X64 \times 63) \times (1 + (DSI\_TIMING2[29] HS\_TX\_TO\_X16 \times 15)))$$

Figure 10-138 shows the HS TX timer FSM.

Figure 10-138. HS TX Timer FSM



dsii-050

When the time-out occurs, hardware sends an EOT request for the DSI complex I/O to drive LP-11 STOP state. This is followed by the generation of the interrupt. Hardware performs an internal logic reset including the TX FIFO content but excluding the register values, and then resets the `DSI_CTRL[0] IF_EN` bit.

Software must take action to recover (for example, by resetting the peripheral if it is not responding). It must wait for the `DSI_CTRL[0] IF_EN` bit to be reset to 0 before taking any recovery action.

10.3.4.4.7.6 LP RX Timer

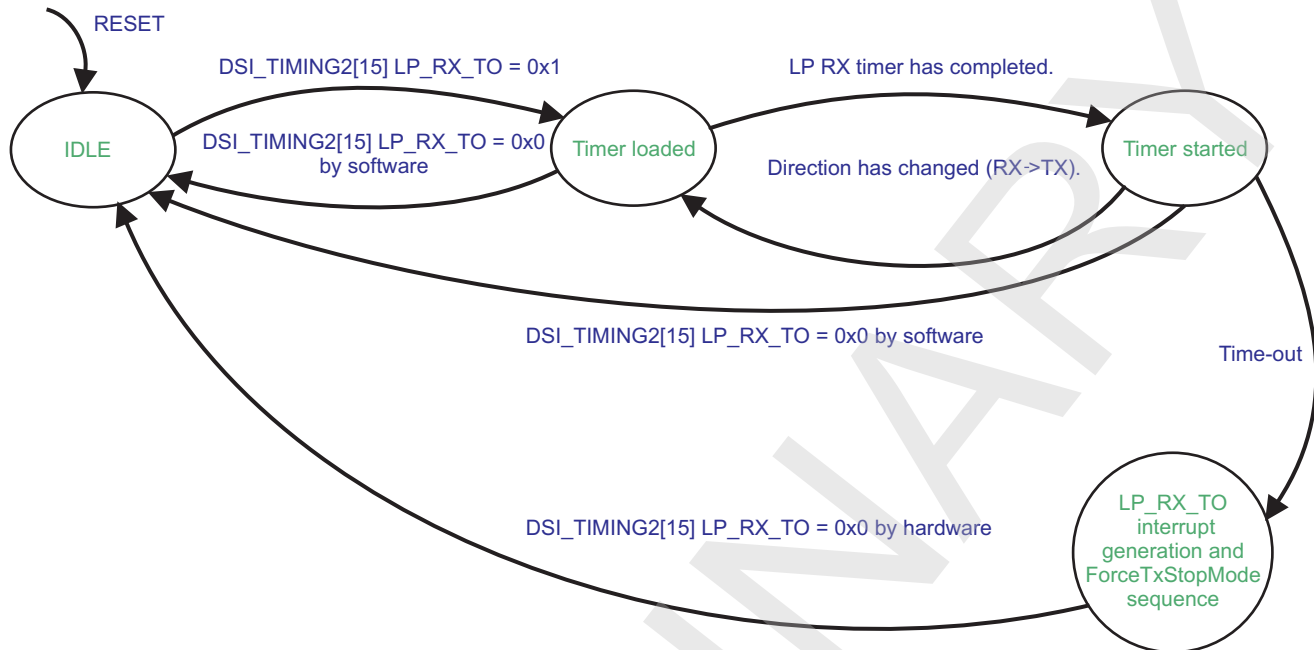
When the host is in low-power receive mode after a BTA, the LP RX timer is loaded. When the timer expires, the host requests the DSI complex I/O to drive LP-11. The interrupt `LP_RX_TO_IRQ` is generated when the timer expires. The `DSI_IRQSTATUS[15] LP_RX_TO_IRQ` bit is set to 1 when the LP RX time-out occurs.

The `DSI_TIMING2[15] LP_RX_TO` bit is set/reset by software to enable or disable the timer.

The timer is loaded with the value in the number of DSI\_CLK functional clock cycles:

$$DSI\_TIMING2[12:0] LP\_RX\_TO\_COUNTER \times ((DSI\_TIMING2[14] LP\_RX\_TO\_X16 \times 15) + 1) \times ((DSI\_TIMING2[13] LP\_RX\_TO\_X4 \times 3) + 1)$$

Figure 10-139 shows the LP RX timer FSM.

**Figure 10-139. LP RX Timer FSM**

ds1-051

When the interrupt is generated, hardware automatically resets the `DSI_TIMING2[15] LP_RX_TO` bit and then asserts `ForceTXStopMode` for the DSI complex I/O to drive LP-11 STOP state. The `ForceTXStopMode` timer is used to define the minimum duration of the LP-11 state. The STOP state can be longer if there is no activity.

Hardware resets the `ForceTXStopMode` bit, followed by an internal logic reset except for all register values and TX FIFO content, and then resets the `DSI_CTRL[0] IF_EN` bit. Software must take action to recover (for example, by resetting the peripheral if it is not responding). It must wait for the `DSI_TIMING1[15] FORCE_TX_STOP_MODE_IO` and `DSI_CTRL[0] IF_EN` bits to be reset before starting the recovery sequence. The TX FIFO is not flushed (the FIFO is flushed only when `DSI_VC_CTRL_i[0] VC_EN` is set to 1).

#### 10.3.4.4.8 Bus Turnaround

The bus turnaround (BTA) is not automatically sent by default after each packet is sent to the panel(s). It is programmable independently for each VC ID. BTA generation can be enabled when the `DSI_VC_CTRL_i[6] BTA_EN` bit is set to 1 by software. Software must ensure that when the BTA is sent to the peripheral, enough time is allocated for the response and for the BTA from the peripheral to the host. For more information about possible DSI PHY timing adjustments during the turnaround procedure, see [Section 10.3.4.6.5.3, Turnaround Request in Transmit Mode](#), and [Section 10.3.4.6.5.4, Turnaround Request in Receive Mode](#).

When setting the `DSI_VC_CTRL_i[6] BTA_EN` bit to 1, one BTA is sent manually to the peripheral. This manual mode can be used for packets in command or video mode.

Acknowledgment from the peripheral for successful BTA is indicated by asserting the `BTA_IRQ` interrupt, if it is enabled in the `DSI_VC_IRQENABLE_i[5] BTA_IRQ_EN` bit. To monitor the BTA interrupt, the user must read the `DSI_VC_IRQSTATUS_i[5] BTA_IRQ` status bit.

#### CAUTION

The BTA should be sent when the RX FIFO is empty. Users must empty the RX FIFO before sending BTA to the peripheral to ensure that when receiving new data from the peripheral, the allocated spaces for all the VCs are empty.

In automatic mode, the BTA is sent automatically at the end of short or long packet transmission, if the [DSI\\_VC\\_CTRL\\_j\[2\] BTA\\_SHORT\\_EN](#) or [DSI\\_VC\\_CTRL\\_j\[3\] BTA\\_LONG\\_EN](#) bit is set to 1.

Two modes can be used for each VC ID:

- Automatic: After each packet, a BTA is sent. To determine the size of the long packet, the protocol engine on the host side reads the word count defined in the header (in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_j](#) register) and uses it to determine the last data to be sent on the DSI link. For short packets, the size is always 4 bytes. The BTA is then sent to the peripheral. The word count is also used to determine how many bytes are to be transferred from the 32-bit write access to the payload register ([DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_j](#)).
- Manual:
  - Data transfer through L3 interconnect: When transferring data using the L3 interconnect port, while all data have been provided to the DSI protocol engine, users can select the BTA for the last packet provided to the L3 interconnect port only by setting the [DSI\\_VC\\_CTRL\\_j\[6\] BTA\\_EN](#) bit, or for the last packets and the following ones by setting automatic mode.
  - Data transfer through video port: In the case of data transfer using the video port, the [DSI\\_VC\\_CTRL\\_j\[6\] BTA\\_EN](#) bit can be selected at any time during the transfer of the packet.
- For video mode packets (data and synchronization events), it is not possible to determine when the BTA is sent; therefore, it is highly recommended to use manual BTA mode only in command mode. It is possible, however, to use the BTA for a VC in video mode.

For data provided on the video port, an interrupt for end of packet transfer [PACKET\\_SENT\\_IRQ](#) is provided to indicate when the packet is completely sent by the DSI complex I/O. [PACKET\\_SENT\\_IRQ](#) can be monitored in the [DSI\\_VC\\_IRQSTATUS\\_j\[2\] PACKET\\_SENT\\_IRQ](#) status bit. The BTA is available even if the space allocated in the TX FIFO for the corresponding VC is empty. It must be sent later even if no packet was sent before the BTA request. The [DSI\\_VC\\_CTRL\\_j\[6\] BTA\\_EN](#) bit must be reset if the BTA request was sent, even if automatic mode for this specific type of packets is enabled.

The BTA is supported for video and command mode packets. It is not possible to send the BTA during the blanking periods of video mode when HS blanking packets should be sent; that is, when one of the following bits is set to 1:

- [DSI\\_CTRL\[20\] BLANKING\\_MODE](#)
- [DSI\\_CTRL\[21\] HFP\\_BLANKING\\_MODE](#)
- [DSI\\_CTRL\[22\] HBP\\_BLANKING\\_MODE](#)
- [DSI\\_CTRL\[23\] HSA\\_BLANKING\\_MODE](#)

Therefore, in video mode, the BTA request is delayed until there is a blanking period without HS blanking packets.

When the TurnRequest signal is asserted (always only for data lane 1), the [TA\\_TO](#) timer is started. If the direction signal is not changed according to the turnaround request, the [TA\\_TO](#) interrupt is generated. When the direction signal is in output mode, any data on the input data bus is ignored because the DSI is in transmission mode (data and triggers must be ignored). For more information about the [TA\\_TO](#) timer, see [Section 10.3.4.4.7.3, TurnRequest FSM](#).

#### **10.3.4.4.9 PHY Triggers**

The DSI protocol engine uses three triggers that are supported only for data lane 1:

- Reset trigger from host to display
- Tearing effect (TE) trigger from display to host
- Acknowledge trigger from display to host

**CAUTION**

Each trigger is associated with a dedicated user-configurable receive or transmit pattern, which is loaded in the [DSI\\_PHY\\_REGISTER3](#) or [DSI\\_PHY\\_REGISTER4](#) bit fields. The default (reset) values of the bit fields are aligned with the MIPI D-PHY specification v0.92. If any of these values need to be changed, the following must be considered:

- If any of the bit fields is written with a nondefault value, the other bit fields in the same register must also be configured with different values. This is to ensure that two different trigger bit fields are not programmed with the same pattern.
- If two or more bit fields are written with equal values, unpredictable behavior of the DSI PHY module may occur.

**10.3.4.4.9.1 Reset**

The DSI protocol engine can use one of the triggers of the DSI\_PHY to send a reset to the display. The reset trigger pattern is configurable through the [DSI\\_PHY\\_REGISTER3](#)[31:24] REG\_TXTRIGGERESC3 bit field. To send the reset pattern to the peripheral, the [DSI\\_CTRL](#)[5] TRIGGER\_RESET bit must be set to 1. When software requires the trigger reset pattern to be sent, the DSI protocol engine resets its own logic, but not the registers. Software can select between two reset modes:

- Immediate reset: All pending requests in TX FIFO not already considered for transfer scheduling, the RX FIFO requests, and the data from video port are ignored. Only the current transfer on the DSI link and those already scheduled are transmitted. All the other transfers are discarded.
- Synchronized reset: The mode is valid only if there is VC using the video mode and if it is active. The principle is to wait for the current video frame to be transferred on the link. Any data on the video port after the current frame are ignored.

To select reset mode, the [DSI\\_CTRL](#)[14] TRIGGER\_RESET\_MODE bit must be programmed.

**CAUTION**

For both reset modes, the DSI protocol engine flushes the FIFOs, synchronization buffers, and line buffers before resetting the [DSI\\_CTRL](#)[0] IF\_EN bit.

**10.3.4.4.9.2 Tearing Effect**

TE on the display is avoided by having synchronization information from the display. It is used only in command mode and is not functional in video mode.

There are two types of TE triggers:

- DSI PHY TE trigger
- CMOS TE line (one of the two input signals can be selected for the TE trigger of a VC; the signal is asynchronous).

Software must set and send the appropriate sequence to receive the TE trigger pattern from the peripheral on the DSI link, or use the CMOS TE line. The value of the expected TE trigger pattern can be configured through the [DSI\\_PHY\\_REGISTER4](#)[23:16] REG\_RXTRIGGERESC2 bit field.

When the TE PHY trigger pattern is received, the DSI protocol engine generates the TE\_TRIGGER\_IRQ interrupt with a TE event, if the interrupt is enabled. To enable the interrupt, the [DSI\\_IRQENABLE](#)[16] TE\_TRIGGER\_IRQ\_EN bit must be set to 1. The [DSI\\_IRQSTATUS](#)[16] TE\_TRIGGER\_IRQ status bit indicates whether the interrupt event has been generated.

When using the TE CMOS signal, the DSI protocol engine generates the TE0\_LINE or TE1\_LINE interrupt, if it is enabled, depending on which line is used. One or multiple VCs can be synchronized using the same TE line. Both lines can be active at the same time for different VCs.

The [DSI\\_VC\\_TE\\_i\[30\]](#) TE\_EN bit must be set and the [DSI\\_VC\\_TE\\_i\[29\]](#) TE\_LINE bit must be reset to indicate that hardware must use the TE PHY trigger to start the transfer of the data from the related VC. TE\_EN is reset when all the data are sent to the peripheral.

The [DSI\\_VC\\_TE\\_i\[30\]](#) TE\_EN, [DSI\\_VC\\_TE\\_i\[29\]](#) TE\_LINE, and [DSI\\_VC\\_TE\\_i\[28\]](#) TE\_LINE\_NB bits must be set to select between the TE0 and TE1 line to indicate that hardware must use the TE0 line or TE1 line to start the transfer of the data from the related VC. The bits are reset when all the data are sent to the peripheral.

The [DSI\\_TE\\_VSYNC\\_WIDTH\\_j](#), [DSI\\_TE\\_HSYNC\\_WIDTH\\_j](#), and [DSI\\_TE\\_HSYNC\\_NUMBER\\_j](#) registers indicate which mode is used to detect the start of the transfer:

- If [DSI\\_TE\\_HSYNC\\_NUMBER\\_j\[10:0\]](#) LINE\_NUMBER = 0, [DSI\\_TE\\_HSYNC\\_WIDTH](#) is ignored. When a pulse on the TE line is detected to be wider than the [DSI\\_TE\\_VSYNC\\_WIDTH\\_j\[19:8\]](#) MIN\_VSYNC\_PULSE\_WIDTH bit field, data transfer starts from the protocol engine to the peripheral.
- If the [DSI\\_TE\\_HSYNC\\_NUMBER\\_j\[10:0\]](#) LINE\_NUMBER bit field is greater than 0, [DSI\\_TE\\_HSYNC\\_WIDTH\\_j](#) is used. The DSI protocol engine waits until it detects a pulse wider than the [DSI\\_TE\\_VSYNC\\_WIDTH\\_j\[19:8\]](#) MIN\_VSYNC\_PULSE\_WIDTH bit field, and then counts each pulse wider than [DSI\\_TE\\_HSYNC\\_WIDTH\\_j\[19:8\]](#) MIN\_HSYNC\_PULSE\_WIDTH but smaller than [DSI\\_TE\\_VSYNC\\_WIDTH\\_j\[19:8\]](#) MIN\_VSYNC\_PULSE\_WIDTH. When the number of HSYNC pulses is reached, data transfer from the DSI protocol engine to the peripheral starts. In case the number of HSYNC pulses is not reached due to a new VSYNC pulse being detected, the counter of HSYNC pulses restarts from 0.

---

**NOTE:** [DSI\\_TE\\_HSYNC\\_WIDTH\\_j\[19:8\]](#) MIN\_HSYNC\_PULSE\_WIDTH is always programmed with a value smaller than [DSI\\_TE\\_VSYNC\\_WIDTH\\_j\[19:8\]](#) MIN\_VSYNC\_PULSE\_WIDTH.

---

**NOTE:** The following must always be set before [DSI\\_VC\\_TE\\_i\[30\]](#) TE\_EN is enabled:

- [DSI\\_TE\\_HSYNC\\_WIDTH\\_j\[19:8\]](#) MIN\_HSYNC\_PULSE\_WIDTH
  - [DSI\\_TE\\_VSYNC\\_WIDTH\\_j\[19:8\]](#) MIN\_VSYNC\_PULSE\_WIDTH
  - [DSI\\_VC\\_TE\\_i\[29\]](#) TE\_LINE
  - [DSI\\_VC\\_TE\\_i\[28\]](#) TE\_LINE\_NB
- 

The [DSI\\_VC\\_TE\\_i\[31\]](#) TE\_START bit must be used when automatic mode is disabled ([DSI\\_VC\\_TE\\_i\[30\]](#) TE\_EN bit = 0). It allows users to start the transfer manually based on application events or based on the TE trigger interrupt ([TE\\_TRIGGER\\_IRQ](#)).

The number of bytes to be transferred is defined by using the [DSI\\_VC\\_TE\\_i\[23:0\]](#) TE\_SIZE bit field. The TE\_SIZE bit field is decremented for each payload byte (it does not include checksum) sent on the DSI link. The register content must not be modified by software during a transfer. The [DSI\\_VC\\_TE\\_i\[23:0\]](#) TE\_SIZE bit field must be set first to indicate that the following accesses to the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register must be used for TE transfer.

The data can be provided from two sources (select by setting the [DSI\\_VC\\_CTRL\\_i\[1\]](#) SOURCE bit):

- L3 interconnect slave port using DMA request: The DMA request [DSI\\_DMA\\_REQi](#) must be asserted only when the TE trigger is received or when the [DSI\\_VC\\_TE\\_i\[31\]](#) TE\_START bit is set by the user and must no longer be asserted when all the bytes defined in the [DSI\\_VC\\_TE\\_i\[23:0\]](#) TE\_SIZE bit field are sent on the DSI link. The VC is associated with a DMA request by programming the number in the [DSI\\_VC\\_CTRL\\_i\[23:21\]](#) DMA\_TX\_REQ\_NB bit field. The [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register is used to provide the number of bytes defined by the [DSI\\_VC\\_TE\\_i\[23:0\]](#) TE\_SIZE bit field (the checksum value is not provided in the [DSI\\_VC\\_LONG\\_PACKET\\_PAYLOAD\\_i](#) register). The size of the header is not considered in the number of bytes to transfer. The [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) register is not used.
- Video port: The DMA request is not asserted. The data are captured in the line buffer using the STALL mechanism. If no line buffer is instantiated (that is, the [DSI\\_CTRL\[13:12\]](#) LINE\_BUFFER bit field is set to 0), it is not possible to use the video port to provide data. The line buffer is filled according to the word count defined in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register header. The value must be written before the TE trigger event is received or before the [DSI\\_VC\\_TE\\_i\[31\]](#) TE\_START bit is set to 1 by software. If the total number of bytes defined by the [DSI\\_VC\\_TE\\_i\[23:0\]](#) TE\_SIZE bit field is not a

multiple of the word count defined in the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) register, all the packets are the same size defined by the WC of the header, except the last transfer. The size of the last transfer is defined by the remaining bytes to send. Because the [DSI\\_VC\\_TE\\_i\[23:0\]](#) TE\_SIZE bit field is modified after each packet transfer, the size of the last packet is equal to the value of the [DSI\\_VC\\_TE\\_i\[23:0\]](#) TE\_SIZE bit field before the last transfer (the header and the payload checksum sizes are not included in the [DSI\\_VC\\_TE\\_i\[23:0\]](#) TE\_SIZE bit field).

When the transfer completes, the value of the [DSI\\_VC\\_TE\\_i\[23:0\]](#) TE\_SIZE bit field is equal to 0. Software must ensure that the pending data in the TX FIFO for the corresponding VC using TE are related to the TE transfer. Any data in the TX FIFO that must be sent before the reception of the TE trigger must be sent before the TE. This is done by not enabling the TE trigger until all data for the corresponding VC are sent to the peripheral. Software can check that the space allocated for the VC in the TX FIFO is empty by reading the [DSI\\_VC\\_CTRL\\_i\[5\]](#) TX\_FIFO\_NOT\_EMPTY status bit.

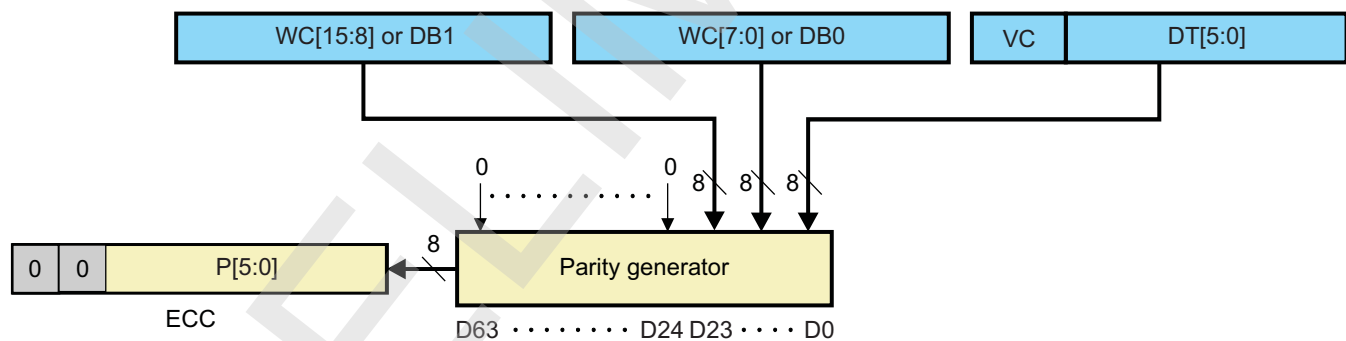
#### 10.3.4.4.9.3 Acknowledge

The corresponding acknowledge interrupt (ACK\_TRIGGER\_IRQ) is generated when the acknowledge trigger is received. The value of the expected acknowledge trigger pattern can be configured through the [DSI\\_PHY\\_REGISTER4\[15:8\]](#) REG\_RXTRIGGERESC1 bit field. To enable the acknowledge interrupt, set the [DSI\\_IRQENABLE\[17\]](#) ACK\_TRIGGER\_IRQ\_EN bit to 1. When the interrupt is generated, the [DSI\\_IRQSTATUS\[17\]](#) ACK\_TRIGGER\_IRQ status bit is set to 1.

#### 10.3.4.4.10 ECC Generation

The DSI protocol uses a 4-byte PH. Because ECC generation requires a fixed word length of 64-bits, the packet headers must be padded with additional bits to form a full 8-byte value for ECC generation and checking. The PH minus the ECC byte should occupy bits D[23:0], and the pad bits should occupy bits D[63:24]. All padding bits must be 0 to generate the ECC byte. ECC can be generated using a parallel approach, as shown in [Figure 10-140](#).

**Figure 10-140. 64-Bit ECC Generation on TX Side**



dsi-052

The ECC generation/check can be enabled and disabled by software. It is defined by a common bit for all the VCs:

- The [DSI\\_CTRL\[2\]](#) ECC\_RX\_EN bit enables and disables ECC generation in the receive direction.
- The [DSI\\_VC\\_CTRL\\_i\[8\]](#) ECC\_TX\_EN bit enables and disables ECC generation in the transmit direction

The ECC can be provided while writing the ECC value directly into the [DSI\\_VC\\_LONG\\_PACKET\\_HEADER\\_i](#) and [DSI\\_VC\\_SHORT\\_PACKET\\_HEADER\\_i](#) registers. The [DSI\\_VC\\_CTRL\\_i\[8\]](#) ECC\_TX\_EN bit indicates whether the ECC value will be calculated or whether the value written in the register will be used instead for command and video modes.



### 10.3.4.4.11 Checksum Generation for Long Packet Payloads

Long packets are comprised of a PH protected by an ECC byte and a payload of 0 to  $2^{16} - 1$  bytes. To detect the errors during the transmission of long packets, a checksum is calculated over the payload portion of the data packet. For the special case of a 0-length payload, the 2-byte checksum is set to 0xFFFF. The checksum can only indicate the presence of one or more errors in the payload. Unlike ECC, the checksum does not enable error correction. For this reason, checksum calculation is not useful for some unidirectional DSI implementations, because the peripheral has no way to report errors to the host processor. Checksum generation and transmission is mandatory for host processors sending long packets to peripherals; it is optional for peripherals transmitting long packets to the host processor. However, the format of long packets is fixed; the peripherals that do not support checksum generation must transmit 2 bytes with a value of 0x0000 in place of the checksum bytes when sending long packets to the host processor. The host processor must disable checksum checking for received long packets from peripherals that do not support checksum generation.

Checksum must be realized as a 16-bit CRC with a generator polynomial of  $x^{16}+x^{12}+x^5+x^0$ . The LSByte is sent first, followed by the MSByte. Within the byte, the LSB is sent first.

Figure 10-141 shows the checksum transmission.

Figure 10-141. Checksum Transmission

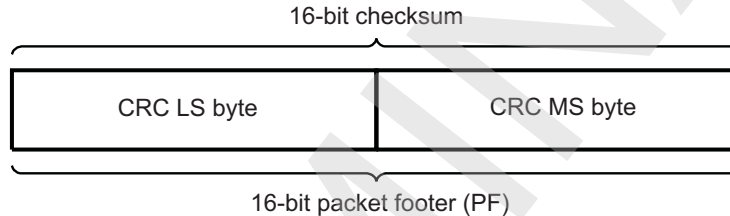
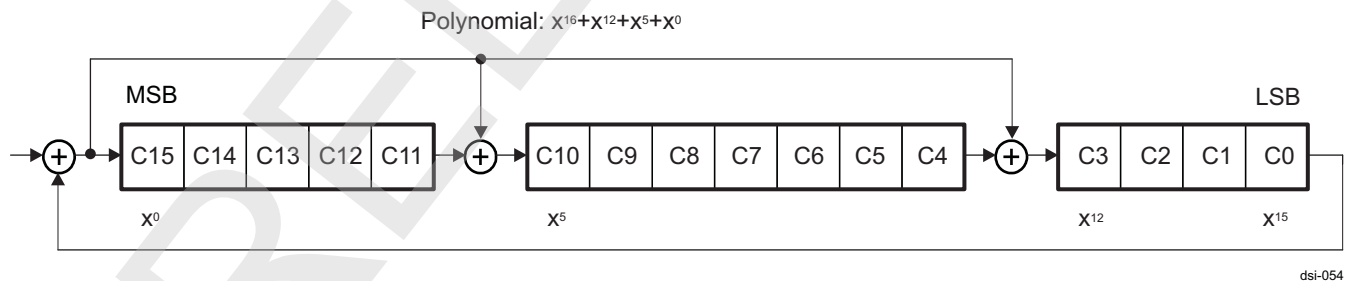


Figure 10-142 shows the CRC implementation. The CRC shift register is initialized to 0xFFFF before packet data enters. Packet data not including the PH then enters as a bitwise data stream from the left, LSB first. Each bit is fed through the CRC shift register before it is passed to the output for transmission to the peripheral. After all bytes in the packet payload have passed through the CRC shift register, the shift register contains the checksum. C15 contains the MSB of the checksum, and C0 contains the LSB of the 16-bit checksum. The checksum is then appended to the data stream and sent to the receiver.

Figure 10-142. 16 Bit CRC Generation Using a Shift Register



Checksum generation/check can be enabled and disabled by software. It is defined by a common bit for all the VCs:

- The **DSI\_CTRL[1] CS\_RX\_EN** bit enables and disables checksum generation in the receive direction.
- The **DSI\_VC\_CTRL\_i[7] CS\_TX\_EN** bit enables and disables checksum generation in the transmit direction

### 10.3.4.4.12 EOT Packet

A new packet type allows the DSI protocol (rather than the DSI\_PHY) at the display to detect the HS EOT. This is a fixed short packet (4 bytes) that is added at every HS to LP transition. This function is enabled by the **DSI\_CTRL[19] EOT\_ENABLE** bit.

The EOT packet has a fixed format:

- Data Type = DI [5:0] = 0b001000
- Virtual Channel = DI [7:6] = 0b00
- Payload Data [15:0] = 0x0F0F
- ECC [7:0] = 0x01

When more than one data lane is used, the EOT packet bytes are distributed across the multiple lanes. EOT packet generation is supported only for the end of HS transmissions. No EOT packet is added at the end of LP transmissions. For LP reception, any EOT packet received is simply passed through as any other packet, but no internal decode or use is made of the EOT information.

#### 10.3.4.4.13 Software Reset

The DSI protocol engine can be reset by software. This reset can be done for debug purposes or after a protocol error and has the same effect as the hardware reset. The DSI protocol engine can be reset by setting the [DSI\\_SYSCONFIG\[1\] SOFT\\_RESET](#) bit to 1. Software can monitor the [DSI\\_SYSSTATUS\[0\] RESET\\_DONE](#) status bit to wait for the reset to complete. If after five reads the [DSI\\_SYSSTATUS\[0\] RESET\\_DONE](#) status bit still returns 0, it can be assumed that an error occurred during the reset stage.

---

**NOTE:** This software reset is optional because a hardware reset is always performed on the DSI protocol engine at device reset.

---

#### 10.3.4.4.14 Power Management

The power-management behavior of the DSI protocol engine is controlled by the [DSI\\_SYSCONFIG](#) register. This register controls the way the module interacts with the PRCM module. The [DSI\\_SYSCONFIG\[0\] AUTO\\_IDLE](#) bit must be set to 1 (default value) to enable automatic clock gating in the module.

The DSI protocol engine implements a handshake protocol on the L3 interconnect slave port with the PRCM module. The protocol engine provides control signal [CIO\\_CLK\\_ICG](#) to gate the SCPClk ([DSS\\_L3\\_ICLK/4](#)). It allows reduction of the power consumption of the complex I/O while the DSI link is not in use ([DSI\\_CLK\\_CTRL\[14\] CIO\\_CLK\\_ICG](#)).

#### 10.3.4.4.15 Power Control of DSI PHY Complex I/O and DSI PLL

The DSI protocol engine can control and send power commands for the DSI complex I/O and DSI PLL controller modules.

##### 10.3.4.4.15.1 Complex I/O Power Control Commands

###### 10.3.4.4.15.1.1 Complex I/O Power Control Commands

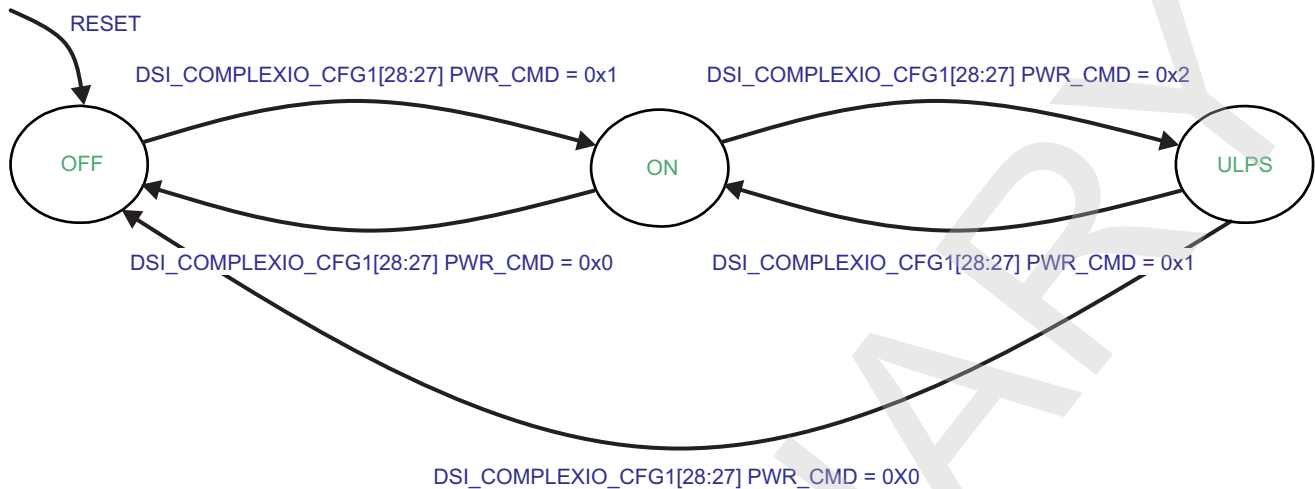
The DSI complex I/O can be set in three modes:

- OFF: In this power state, the complete DSI\_PHY circuit is powered down. The internal LDO is OFF.
- ON: In this power state, the complete DSI\_PHY circuit is powered on and functional.
- ULPS: In this power state, the ULPS exit detection circuit power switch is ON for the lanes that are in receive ULPS. For the lanes that are in transmit ULPS, the circuitry for weak pulldown is ON. The ULTRALOW POWER state must be used only when all the three lanes are in ULPS (transmit or receive).

###### 10.3.4.4.15.1.2 Complex I/O Power FSM

[Figure 10-143](#) shows the FSM to control the power state of the complex I/O.

Figure 10-143. Complex I/O Power FSM



dsi-045

The PwrCmdOff, PwrCmdUlp, and PwrCmdOn commands control the state transition of the DSI complex I/O. The `DSI_COMPLEXIO_CFG1[28:27] PWR_CMD` bit field allows the state to be modified. The allowed transitions are: OFF - ON, and ON - ULPS, and ULPS - OFF. The `DSI_COMPLEXIO_CFG1[26:25] PWR_STATUS` bit field gives a status on the current state of the DSI complex I/O.

**CAUTION**

- In automatic mode, software must ensure that the DSI complex I/O is in ON state (that is, ON command already sent) before sending requests to the complex I/O.
- In a command request to change to a state that is the current one (acknowledge has been received), the command is ignored (nothing is sent to the DSI complex I/O).
- To change the state to ULPS, all ULPSActiveNot signals must be low. The ULPSActiveNot\_ALL0\_IRQ interrupt indicates the state of the ULPSActiveNot signals. The change from ULPS to ON state is required before starting the ULP exit sequence (for information, see [Section 10.3.4.4.7.1, T<sub>wakeup</sub> Timer](#)).

**10.3.4.4.15.2 DSI PLL Power Control Commands**

The DSI PLL controller module can be set in four modes:

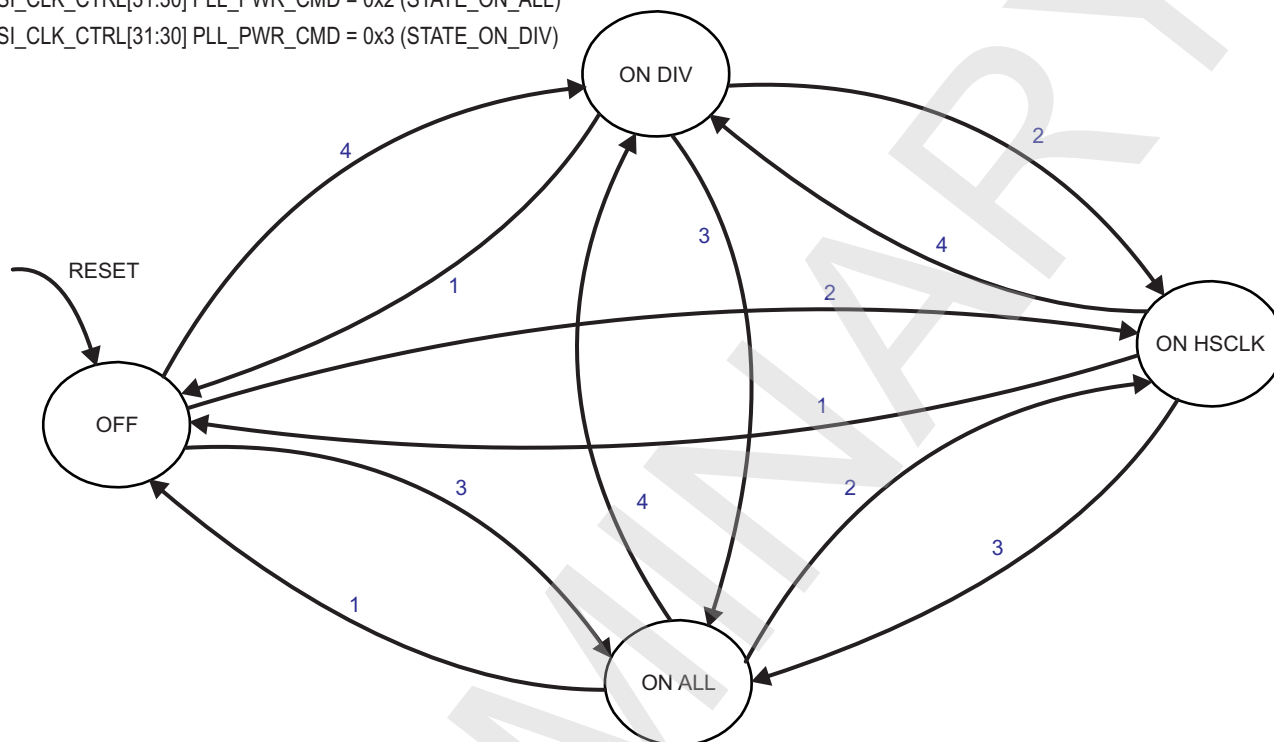
- OFF: DSI PLL and HSDIVIDER are OFF.
- ON ALL: Both DSI PLL and HSDIVIDER are ON. The CLKIN4DDR clock is provided to the DSI complex I/O and the second clock output is provided to the HSDIVIDER.
- ON HSCLK: The DSI PLL is ON. The HSDIVIDER is OFF. The CLKIN4DDR clock is provided to the DSI complex I/O, but the second clock output is not provided to the HSDIVIDER.
- ON DIV: DSI PLL and HSDIVIDER are ON. The CLKIN4DDR clock is not provided to the DSI complex I/O, but the second clock output is provided to the HSDIVIDER.

**10.3.4.4.15.2.1 DSI-PLL Power FSM**

Figure 10-144 shows the DSI PLL power FSM.

**Figure 10-144. DSI PLL Power FSM**

- 1 = DSI\_CLK\_CTRL[31:30] PLL\_PWR\_CMD = 0x0 (STATE\_OFF)  
 2 = DSI\_CLK\_CTRL[31:30] PLL\_PWR\_CMD = 0x1 (STATE\_ON\_HSCLK)  
 3 = DSI\_CLK\_CTRL[31:30] PLL\_PWR\_CMD = 0x2 (STATE\_ON\_ALL)  
 4 = DSI\_CLK\_CTRL[31:30] PLL\_PWR\_CMD = 0x3 (STATE\_ON\_DIV)



dsi-046

The following commands control the state transition of the DSI PLL control module:

- PLLPwrCmdOff
- PLLPwrCmdOnAll
- PLLPwrCmdOnDIV
- PLLPwrCmdOnHSCLK

Setting the [DSI\\_CLK\\_CTRL\[31:30\] PLL\\_PWR\\_CMD](#) bit field modifies the state. The [DSI\\_CLK\\_CTRL\[29:28\] PLL\\_PWR\\_STATUS](#) bit field gives a status on the current state of the DSI PLL controller.

---

**NOTE:** If a command requests to change to a state that is the current one (acknowledge has been received), the command is ignored (nothing is sent to the DSI PLL control module).

---

#### 10.3.4.4.15.2.1.1 DSI PLL HS Clock Signals

The DSIStopClk signal is provided to the DSI PLL control module. It indicates when the DSI protocol engine does not need to use the HS mode and the PLL HS output (CLKIN4DDR clock) can be stopped. The following conditions must also be met when DISPC\_UPDATE\_SYNC may be generated by the DISPC, because that must result in the PLL HS output being stopped.

When the interface is disabled ([DSI\\_CTRL\[0\] IF\\_EN](#) bit set to 0) and the clock lane is stopped ([DSI\\_CLK\\_CTRL\[13\] DDR\\_CLK\\_ALWAYS\\_ON](#) = 0), the signal DSIStopClk is asserted.

Assertion of the DSIStopClk depends on the following conditions:

- Clock lane TxRequestHS is deasserted (the DDR clock on the clock lane is no longer required). For TxRequestHS deassertion, the following conditions are required:
  - The [DSI\\_CLK\\_CTRL\[13\] DDR\\_CLK\\_ALWAYS\\_ON](#) bit must be reset to 0 and no HS data transfer

must be on-going or already scheduled.

- No VC ACTIVE in video mode. No VC using the video mode is enabled. If the VC is enabled, the mode is command mode only (the `DSI_VC_CTRL_i[0]` VC\_EN bit set to 1 and `DSI_VC_CTRL_i[4]` MODE bit set to 0).
- No command mode requiring HS transfer (one or more VCs using command mode can be active)
- Or `DSI_CTRL[0]` IF\_EN bit is reset to 0 (the previous conditions are not all required, except the `DSI_CLK_CTRL[13]` DDR\_CLK\_ALWAYS\_ON bit must be set to 0 after resetting the `DSI_CTRL[0]` IF\_EN bit)

The deassertion of the DSIStopClk depends on one of the following conditions (the DSI interface is enabled by setting the `DSI_CTRL[0]` IF\_EN bit to 1):

- Clock lane TxRequestHS must be asserted (the DDR clock on the clock lane is required).
- One video mode VC ACTIVE
- At least one VC in command mode requiring HS transfer
- The `DSI_CLK_CTRL[13]` DDR\_CLK\_ALWAYS\_ON bit is set to 1 by software users (the `DSI_CTRL[0]` IF\_EN bit must be reset to 0 for updating the value of the DDR\_CLK\_ALWAYS\_ON bit).

Automatic assertion/deassertion is enabled by using the `DSI_CLK_CTRL[18]` HS\_AUTO\_STOP\_ENABLE bit.

Manual mode can be used by setting/resetting the `DSI_CLK_CTRL[19]` HS\_MANUAL\_STOP\_CTRL bit to assert/deassert the DSIStopClk signal.

---

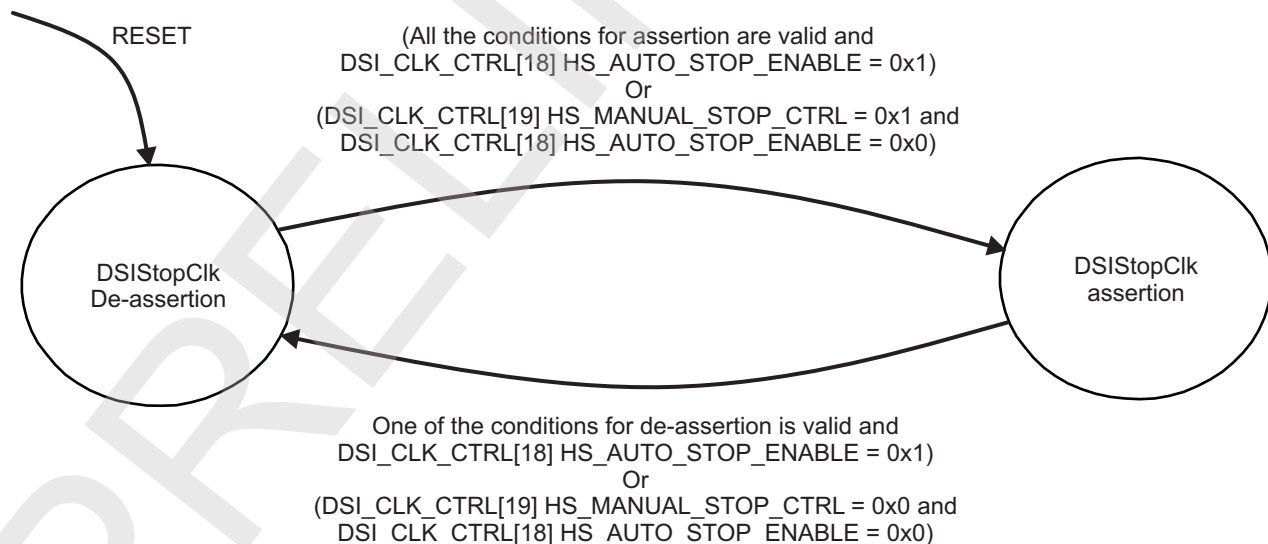
**NOTE:** Setting the `DSI_CTRL[24]` DISPC\_UPDATE\_SYNC bit to 0x1 enables the use of the `Dispc_Update_Sync` signal to synchronize.

---

#### 10.3.4.4.15.2.1.2 DSI-PLL HS Clock FSM

Figure 10-145 shows the DSI PLL HS clock FSM.

**Figure 10-145. DSI PLL HS Clock FSM**



dsii-047

When DSIStopClk is used there is a latency through other modules (DSI PLL controller and DSI\_PHY) before TXBYTECLKHS is stopped. This latency must be considered to prevent any issue when DSIStopClk is deasserted soon after being asserted. This is done using a hardware timer programmed using the `DSI_STOPCLK_TIMING[7:0]` DSI\_STOPCLK\_LATENCY bit field. This timer is programmed in number of periods of the DSI protocol functional clock (DSI\_FCLK). At reset value, the timer is programmed with value 0x80 (128).

**CAUTION**

The programmed value in the [DSI\\_STOPCLK\\_TIMING\[7:0\]](#) DSI\_STOPCLK\_LATENCY bit field must be greater than ((3 x L3\_ICLK period) + (5 x CLKIN4DDR period))/(DSI\_FCLK period).

**10.3.4.4.16 Ultralow-Power State Configuration**

This section describes how to enter/exit to/from ULTRALOW-POWER state (ULPS).

**NOTE:** The [DSI\\_COMPLEXIO\\_CFG2\[9:0\]](#) LANEx\_ULPS\_SIGy bits (*x* range is 1 to 5, corresponding to lane 1 to lane 5 for DSI1, and 1 to 3, corresponding to lane 1 to lane 3 for DSI2; *y* range is 1 to 2) must be read back after writing to verify that the write operations are effective before proceeding to the next step. This is to take latency at low TxClkEsc frequencies into account.

**10.3.4.4.16.1 Entering ULPS**

To enter into ULPS for a clock lane, the following sequence is required:

1. Wait for [DSI\\_COMPLEXIO\\_CFG2\[16\]](#) HS\_BUSY and [DSI\\_COMPLEXIO\\_CFG2\[17\]](#) LP\_BUSY bits to be reset to 0 and ensure that the [DSI\\_CLK\\_CTRL\[13\]](#) DDR\_CLK\_ALWAYS\_ON bit is 0.
2. TxUlpsClk state should change from inactive to active by setting the [DSI\\_COMPLEXIO\\_CFG2\[9:5\]](#) LANEx\_ULPS\_SIG2 bit to 1.

To enter into ULPS for a data lane, the following sequence is required:

1. Wait for all TX\_FIFOs for all VCs working in HS are empty, for video mode is not active, and for [DSI\\_COMPLEXIO\\_CFG2\[16\]](#) HS\_BUSY bit is reset to 0 (in addition for data lane 1, [DSI\\_COMPLEXIO\\_CFG2\[17\]](#) LP\_BUSY bit is reset to 0)
2. TxRequestEsc state should change from inactive to active by setting the [DSI\\_COMPLEXIO\\_CFG2\[9:5\]](#) LANEx\_ULPS\_SIG2 bit to 1.

**10.3.4.4.16.2 Exiting ULPS**

To exit from ULPS for a clock lane, the following sequence is required:

1. Change the state of TxUlpsExit for each lane to ACTIVE by setting the [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bit to 1.
2. Wait for the ULPSACTIVENOT\_ALL1\_IRQ interrupt indicating that all lanes with TxUlpsExit active have acknowledged by asserting UlpsActiveNot. This is performed by monitoring the [DSI\\_COMPLEXIO\\_IRQSTATUS\[31\]](#) ULPSACTIVENOT\_ALL1\_IRQ status bit.
3. Start the wake-up timer (GPTimer).
4. Wait for the time-out.
5. Change TxUlpsClk signals to INACTIVE state for the clock lane by resetting the [DSI\\_COMPLEXIO\\_CFG2\[9:5\]](#) LANEx\_ULPS\_SIG2 bit to 0.
6. Reset the [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bit to 0.

**NOTE:** When the [DSI\\_COMPLEXIO\\_CFG2\[9:5\]](#) LANEx\_ULPS\_SIG2 and [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bits are both being written to 0, they can be combined into one write. Both bits must be read back to confirm they are effective before proceeding.

To exit from ULPS for a clock lane, in case the complex I/O is in OFF state (the DSI protocol engine sends the complex I/O to OFF state (PWROFF command) by setting [DSI\\_COMPLEXIO\\_CFG1\[28:27\]](#) PWR\_CMD = 0x0), the sequence is:

1. Change TxUlpsClk signals to INACTIVE state for the clock lane by resetting the [DSI\\_COMPLEXIO\\_CFG2\[9:5\]](#) LANEx\_ULPS\_SIG2 bit to 0.

2. Change the state of TxUlpsExit for clock lane to INACTIVE state by resetting the [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bit to 0. This step is necessary only in case a PWROFF command (the command for power control of the complex I/O) is issued while the sequence for exiting is in progress (TxUlpsExit signal is already in ACTIVE state).

---

**NOTE:** When the [DSI\\_COMPLEXIO\\_CFG2](#) LANEx\_ULPS\_SIG2[9:5] and [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bits are both being written to 0, they can be combined into one write. Both bits must be read back to confirm they are effective before proceeding.

---

To exit from ULPS for a data lane, the following sequence is required:

1. Change the state of TxUlpsExit for each lane to ACTIVE by setting the [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bit to 1.
2. Wait for the ULPSACTIVENOT\_ALL1\_IRQ interrupt indicating that all lanes with TxUlpsExit active have acknowledged by asserting UlpsActiveNot. This is performed by monitoring the [DSI\\_COMPLEXIO\\_IRQSTATUS\[31\]](#) ULPSACTIVENOT\_ALL1\_IRQ status bit.
3. Start the application wake-up timer (GPTimer).
4. Wait for the time-out.
5. Change TxRequestEsc signals to INACTIVE state for the data lane by resetting the [DSI\\_COMPLEXIO\\_CFG2\[9:5\]](#) LANEx\_ULPS\_SIG2 bit to 0.
6. Reset the [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bit to 0.

---

**NOTE:** When the [DSI\\_COMPLEXIO\\_CFG2\[9:5\]](#) LANEx\_ULPS\_SIG2 and [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bits are both being written to 0, they can be combined into one write. Both bits must be read back to confirm they are effective before proceeding.

---

To exit from ULPS for a data lane, in case the complex I/O is in OFF state (the DSI protocol engine sends the complex I/O into OFF state by setting [DSI\\_COMPLEXIO\\_CFG1\[28:27\]](#) PWR\_CMD = 0x0), the sequence is:

1. Change TxRequestEsc signals to INACTIVE state by resetting the [DSI\\_COMPLEXIO\\_CFG2\[9:5\]](#) LANEx\_ULPS\_SIG2 bit to 0.
2. Change the state of TxUlpsExit to INACTIVE state by resetting the [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bit to 0. This step is necessary only in case a PWROFF command is issued while the sequence for exiting is in progress (TxUlpsExit signal is already in ACTIVE state).

---

**NOTE:** When the [DSI\\_COMPLEXIO\\_CFG2\[9:5\]](#) LANEx\_ULPS\_SIG2 and [DSI\\_COMPLEXIO\\_CFG2\[4:0\]](#) LANEx\_ULPS\_SIG1 bits are both being written to 0, they can be combined into one write. Both bits must be read back to confirm they are effective before proceeding.

---

When the sequence for entering/exiting into/from ULP state is started for specific lanes, users should wait for the completion of the sequence before changing the state of the same or other lanes.

#### 10.3.4.4.17 Interrupts

There is a single interrupt request generated by each DSI: DSI\_IRQ. This interrupt line is merged with another interrupt line from the DISPC\_IRQ into a single interrupt request IRQ. The DSI\_IRQ events are generated only for the enabled VC(s). Two registers are used to enable and monitor the DSI interrupt events:

- **DSI\_IRQENABLE:** This register enables and disables interrupt event reporting for the VCs. Each event for the VC is configured in the [DSI\\_VC\\_IRQENABLE\\_i](#) register dedicated to the VC number. In addition, it includes 1 bit to enable error reporting for the complex I/O. The interrupt is triggered when any error is received from the complex I/O.

- **DSI\_IRQSTATUS:** The register flags which VC(s) have generated an interrupt. Based on the VC number, the **DSI\_VC\_IRQSTATUS\_i** register indicates the event generating the interrupt. In addition, it includes 1 bit for the status of error reporting for the complex I/O.

Table 10-488 lists the DSI VC interrupt events

**Table 10-488. DSI VC Interrupts**

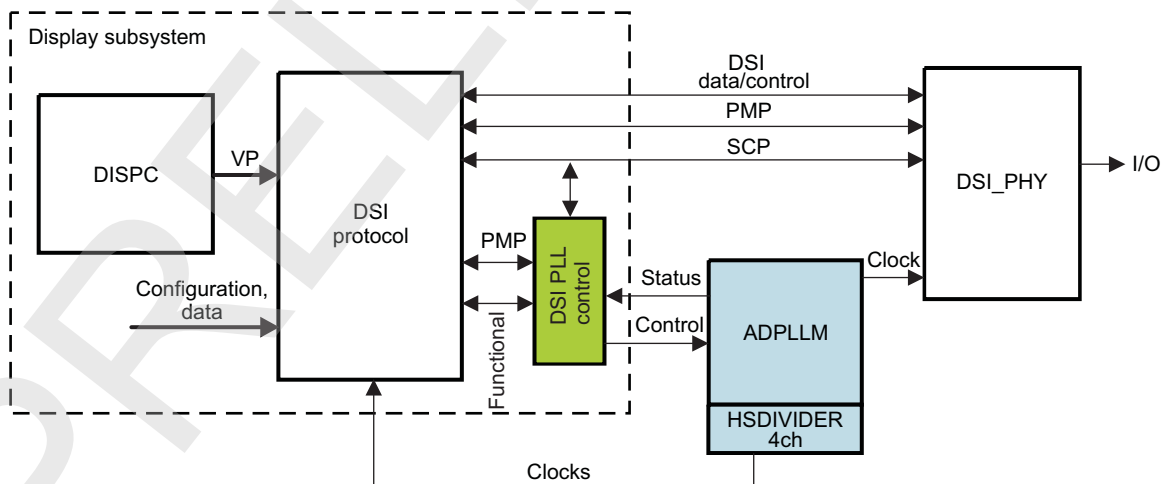
IRQ Name	Description
ECC_CORRECTION_IRQ	Indicates whether a 1-bit error correction has occurred using ECC (short and long packets).
PACKET_SENT_IRQ	Indicates that a packet has been sent. It is used when BTA manual mode is used.
CS_IRQ	Virtual channel. Checksum of the payload mismatch detection.
FIFO_RX_OVF_IRQ	RX FIFO overflow. The FIFO used on the L3 interconnect slave port for buffering the data received on the DSI link has overflowed.
FIFO_TX_OVF_IRQ	TX FIFO overflow. The FIFO used on the L3 interconnect slave port for buffering the data received on the interface slave port has overflowed.
BTA_IRQ	BTA is received from the peripheral (the VC ID used for the last BTA request transfer to the peripheral will be used to determine which VC will be used to flag the interrupt).
ECC_NO_CORRECTION_IRQ	ECC error (short and long packets). No correction of the header because of more than 1-bit error.
FIFO_TX_UDF_IRQ	TX FIFO underflow. The FIFO used on the slave port for buffering the data received on the L3 interconnect slave port has underflowed in the middle of a packet transfer.

### 10.3.4.5 DSI PLL Controllers

#### 10.3.4.5.1 DSI PLL Controllers Overview

Two DSI PLL controller modules are part of the DSS. They use the SCP (see Section 10.3.4.4.6, *Serial Configuration Port Interface*) and power-management port (PMP) as the primary interfaces to the DSI protocol engines. The SCP interface is used to set the configuration of the digital phase-locked loop (DPLL) and HSDIVIDER modules, primarily the various counter values. The PMP is used to control the power state of the DPLL and HSDIVIDER modules. Figure 10-146 is an overview of a single DSI PLL controller module in the display subsystem.

**Figure 10-146. DSI PLL Controller Overview**



dsi-056

**NOTE:** The DSI PLL controller modules do not have an interface to L3/L4\_PER interconnects. The programmable features are managed by registers mapped into the DSI protocol engines.

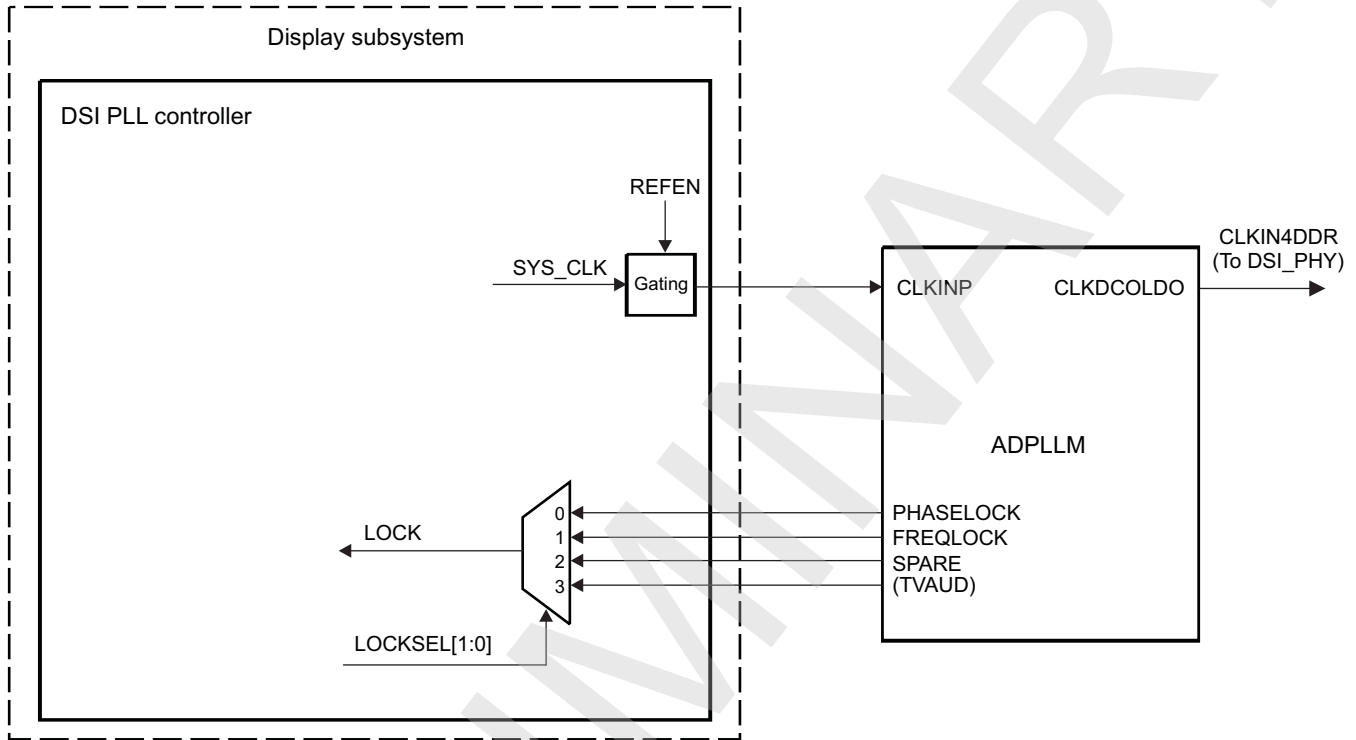


### 10.3.4.5.2 DSI PLL Controllers Architecture

The DSI PLLs use instances of the ADPLL modules. The pixel clock (PCLK) frequency range is up to 103 MHz.

Figure 10-147 shows the internal reference diagram of a single DSI PLL.

Figure 10-147. DSI PLL Reference Diagram



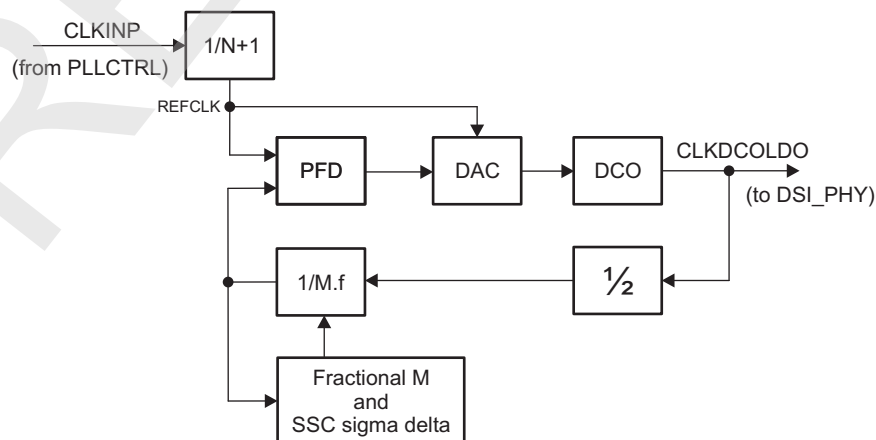
dsi-057

The DSI PLL clock output (CLKDCOLDO) corresponds to the CLKIN4DDR clock of a DSI complex I/O (DSI\_PHY) module.

Only the SYS\_CLK clock is used as an input reference clock for the ADPLL. The DSI PLL controller can gate this clock by setting the [DSI\\_PLL\\_CONFIGURATION2\[13\]](#) PLL\_REFEN bit.

Figure 10-148 is a simplified block diagram of the ADPLL instance used for the DSI modules.

Figure 10-148. ADPLL Functional Block Diagram



dsi-029

The input clock CLKINP goes to a predivider  $N + 1$ . The entire loop runs on the REFCLK clock after this predivider. The value of  $N + 1$  is controlled through the [DSI\\_PLL\\_CONFIGURATION1\[8:1\] PLL\\_REGN](#) bit field.

The output clock CLKDCOLDO is synthesized by a digitally controlled oscillator (the DCO block), that automatically detects the frequency range. The CLKDCOLDO frequency can be given with  $CLKDCOLDO = CLKINP * 2 * M / (N + 1)$ . For that purpose the feedback multiplier  $M$  must be configured through the [DSI\\_PLL\\_CONFIGURATION1\[20:9\] PLL\\_REGM](#) bit field.

The ADPLL module supports fractional synthesis. The frequency multiplication factor  $M$  can be programmed as fractional. This is done through the use of a sigma delta feedback divider ( $M$ ). A fractional value (Fractional  $M$ ) of 18 bits is supported, thus enabling control for better accuracy. Programming the fractional value is done by setting the [DSI\\_PLL\\_CONFIGURATION4\[17:0\] PLL\\_REGM\\_F](#) register bit field.

The module also supports spread spectrum clocking (SSC) on its output clocks. SSC is used to spread the spectral peaking of the clock to reduce electromagnetic interference (EMI). When SSC is enabled, the clock spectrum is spread by the amount of frequency spread, and the attenuation is given by the ratio of the frequency spread ( $f$ ) and the modulation frequency ( $f_m$ ), that is,  $\{10 * \log_{10}(df/f_m)\} - 4$  dB.

The SSC is performed by changing the feedback divider ( $M$ ) in a triangular pattern, which means the frequency of the output clock varies in a triangular pattern. The frequency of the triangular pattern is modulation frequency ( $f_m$ ). The peak ( $dM$ ) or the amplitude of the triangular pattern as a percent of  $M$  is equal to the percent of the frequency spread ( $f$ ) (that is,  $dM/M = df/F_{out}$ ).

The SSC can be enabled and disabled by asserting the [DSI\\_PLL\\_SSC\\_CONFIGURATION1\[0\] EN\\_SSC](#) bit. The acknowledge signal SSCACK, observed by the [DSI\\_PLL\\_STATUS\[12\] SSC\\_EN\\_ACK](#) bit, notifies the exact start and end of SSC. When  $EN\_SSC$  is deasserted, SSC is disabled only after completion of one full cycle of the triangular pattern given by the modulation frequency. This is done to maintain the average frequency.

The modulation frequency ( $f_m$ ) can be programmed as a ratio of  $REFCLK/4$ ; that is, the value programmed in the [DSI\\_PLL\\_SSC\\_CONFIGURATION2\[29:20\] MODFREQDIVIDER](#) bit field must be  $= REFCLK/(4 * f_m)$ .

To define the frequency spread ( $df$ ),  $M$  must be controlled as explained above. To define  $M$ , the step size of  $M$  for each REFCLK during the triangular pattern must be programmed (that is,  $M = ModFreqDivider * DeltaMStep$ ).  $DeltaMStep$  is controlled by the [DSI\\_PLL\\_SSC\\_CONFIGURATION2\[19:0\] DELTAM](#) bit field. The frequency spread achieved, however, has an overshoot of 20 percent or an inaccuracy of +20 percent.

If the [DSI\\_PLL\\_SSC\\_CONFIGURATION1\[2\] DOWNSPREAD](#) bit is set to 1, the frequency spread on the lower side is twice the programmed value. The frequency spread on the higher side is 0 (except for the overshoot, as described above).

### 10.3.4.5.3 DSI PLL Operations

The DSI PLL configuration signals operate according to [Table 10-489](#), which indicates the operation when the PLLs are not locked.

**Table 10-489. DSI PLL Operation Modes When Not Locked**

DSI PLL Operation Mode	Stop Mode Low Power <sup>(1)</sup>	Stop Mode Fast Relock <sup>(1)</sup>	Idle Bypass
Mode Description	Output clocks stopped Lowest power standby	Output clocks stopped Fastest start-up time	Selects when PLLs and HSDIVERS bypass clocks are used
<a href="#">DSI_PLL_CONFIGURATION2[0]</a> PLL_IDLE	0	0	1
<a href="#">DSI_PLL_CONFIGURATION2[6]</a> PLL_LOWCURRSTBY	1	0	1
<a href="#">DSI_PLL_CONFIGURATION1[0]</a> PLL_STOPMODE	1	1	x

<sup>(1)</sup> This mode must be used for better performance.

When locked, the PLL output frequency is: Input frequency \* 2 \* M/(N + 1), where:

- M multiplier is programmed in the [DSI\\_PLL\\_CONFIGURATION1](#)[20:9] PLL\_REGM bit field.
- N divider is programmed in the [DSI\\_PLL\\_CONFIGURATION1](#)[8:1] PLL\_REGN bit field.

#### 10.3.4.5.4 DSI PLL Controllers Shadowing Mechanism

The configuration registers are accessed through the DSI protocol engine register spaces using the SCP interface. This includes all the configuration signals and returning status signals.

#### CAUTION

All writes must be 32-bit operations, because the SCP interface always transfers 32 bits; 16-or 8-bit operations may lead to unpredictable errors.

A shadow mechanism is implemented for appropriate register values so that configurations can optionally be updated in synchronization with the DISPC and DSI protocol engines. The front porch time from the DISPC indicates the time when making the update of the value. All the required updated values must be written before this signal is asserted. For more information, see [Section 10.3.4.4.6.1, Shadowing Register](#).

#### 10.3.4.5.5 Error Handling

The PLL lock and recalibration signals can be monitored to detect the loss of lock or the requirement to recalibrate (caused by a large temperature change since the last lock request):

- The [DSI\\_PLL\\_STATUS](#)[1] PLL\_LOCK status bit gives the DSI PLL lock state.
- The [DSI\\_PLL\\_STATUS](#)[2] PLL\_RECAL status bit informs whether the PLL must be uncalibrated.

These signals can also generate interrupts at the DSI protocol engine level:

- The PLL\_LOCK\_IRQ interrupt indicates that the DSI PLL control module has sent a lock request to the DSI PLL. To monitor this event, read the [DSI\\_IRQSTATUS](#)[7] PLL\_LOCK\_IRQ bit. Set this bit to 1 to clear the status bit.
- The PLL\_UNLOCK\_IRQ interrupt indicates that the DSI PLL control module has sent an unlock request to the DSI PLL. To monitor this event, read the [DSI\\_IRQSTATUS](#)[8] PLL\_UNLOCK\_IRQ bit. Set this bit to 1 to clear the status bit.
- The PLL\_RECAL\_IRQ interrupt indicates that the DSI PLL control module has sent a recalibration request to the DSI PLL. To monitor this event, read the [DSI\\_IRQSTATUS](#)[9] PLL\_RECAL\_IRQ bit. Set this bit to 1 to clear the status bit.

The PLL reference loss and limp status signals can also be monitored:

- The [DSI\\_PLL\\_STATUS](#)[3] PLL\_LOSSREF status bit informs whether the DSI PLL has lost the reference.
- The [DSI\\_PLL\\_STATUS](#)[4] PLL\_LIMP bit informs about the DSI PLL limp status.

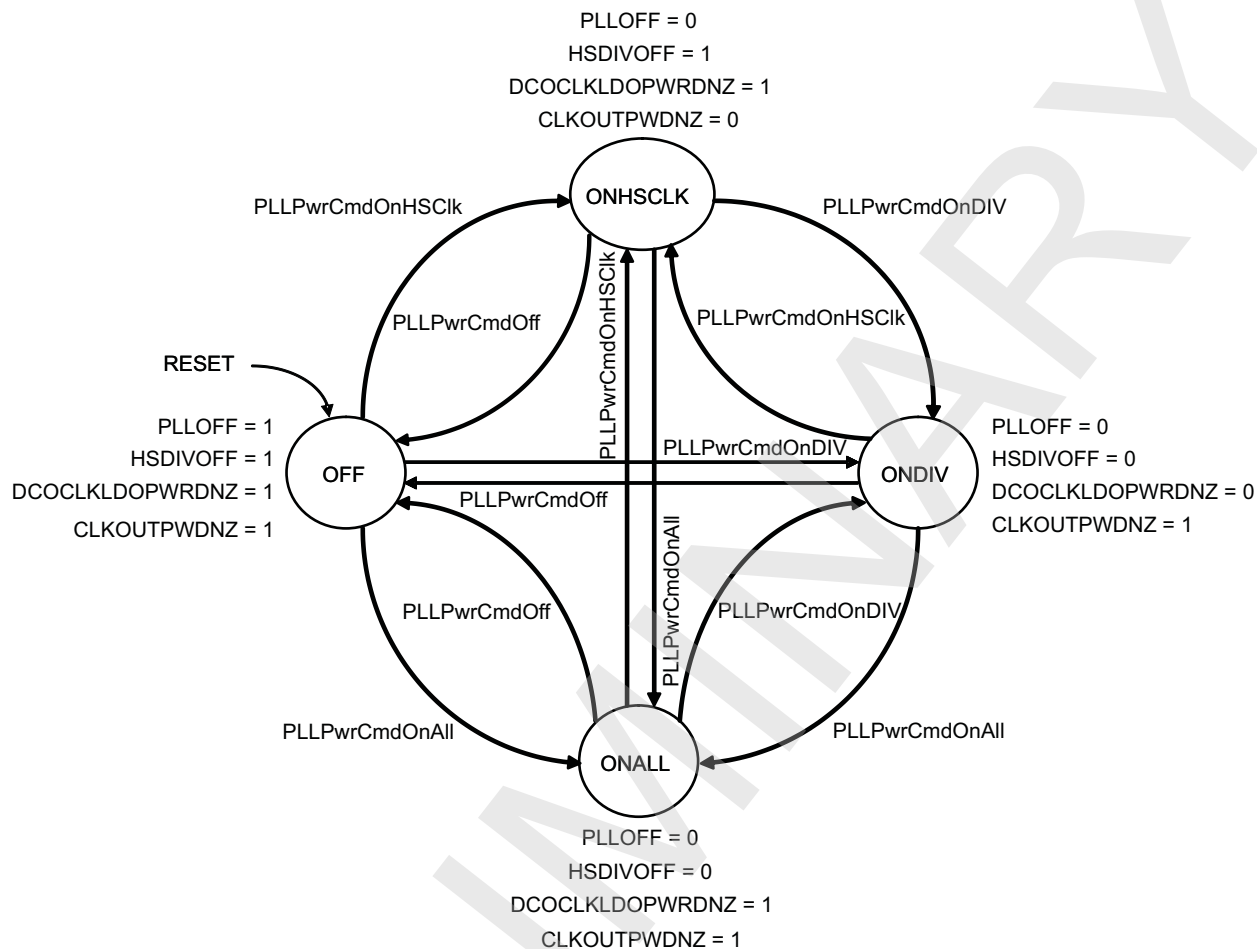
#### 10.3.4.5.6 Software Reset

The DSI PLL control modules do not have their own software reset. They are reset by the global DSS\_RST signal at PRCM module level. See [Section 10.1.2.2, DSS Resets](#). Nevertheless, software users can monitor the reset statuses of the DSI PLL control modules by reading the [DSI\\_PLL\\_STATUS](#)[0] PLLCTRL\_RESET\_DONE status bits.

#### 10.3.4.5.7 Power Management

PLLCTRL manages only the LDO power of the ADPLL and HSDIVIDER; this is done by overriding the SYSRESET signals. All other power-management signals are integrated with the display subsystem power management.

[Figure 10-149](#) shows the power states that are controlled through the PMP.

**Figure 10-149. Power State Diagram**

dsi-058

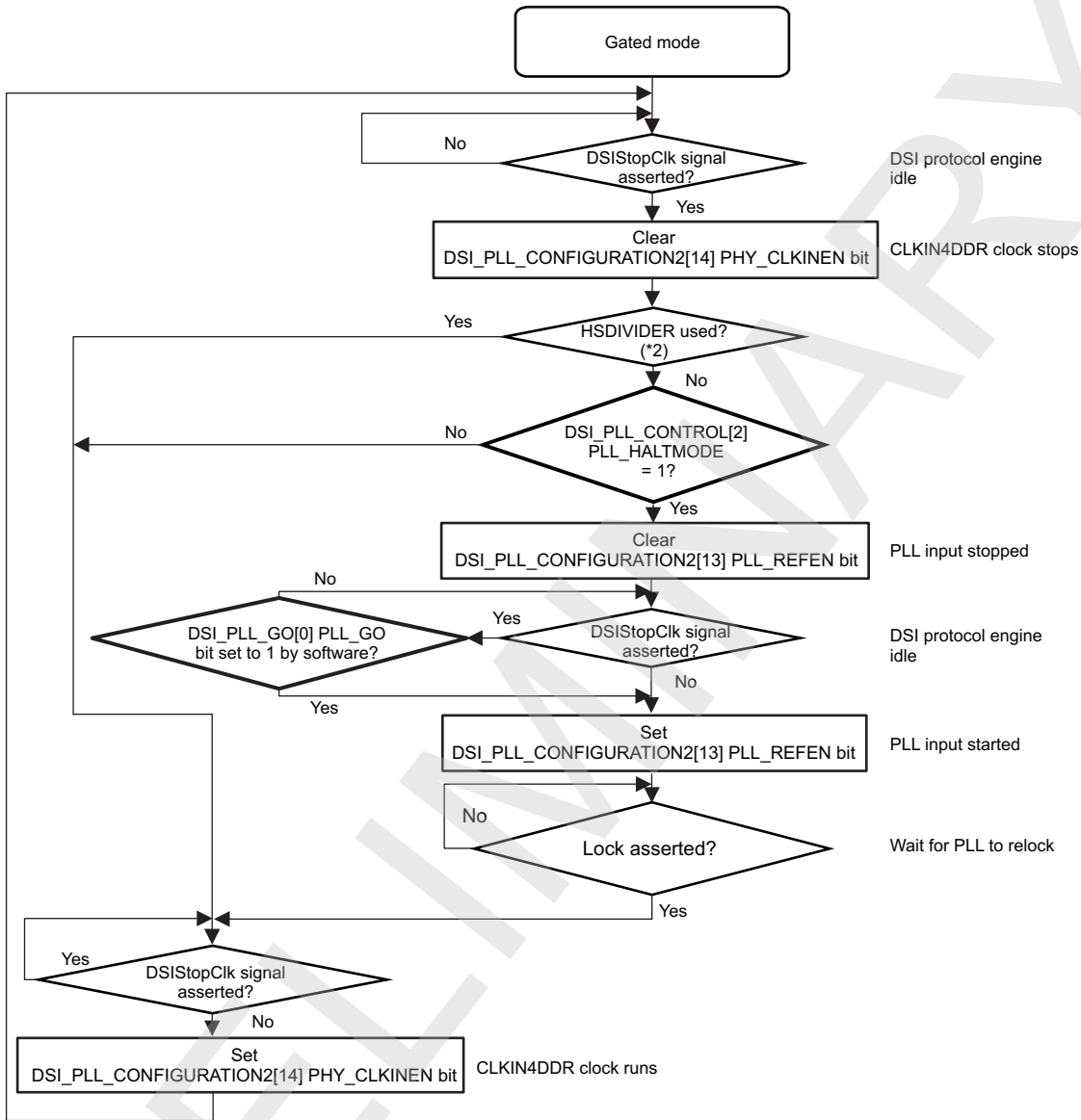
#### 10.3.4.5.8 DSI PLL Clock-Gating Sequence

Clock gating can be used to reduce system power consumption when the DSI protocol engine indicates that it does not need the clock. If the HSDIVIDER is not used, the PLL can also be stopped (at the cost of additional unstarting latency).

The DSI protocol engines must verify when the PLLs have unstarted by inspecting the LOCK signal (the [DSI\\_PLL\\_STATUS\[1\]](#) PLL\_LOCK status bit), because the TXBYTECLKHS clocks are stopped when the CLKIN4DDR clocks generated by the DSI PLL are stopped. This eliminates the need for any explicit feedback that the clock has been unstarted. This flow chart should run even if the [DSI\\_PLL\\_GO\[0\]](#) PLL\_GO bit is not set.

[Figure 10-150](#) shows the DSI PLL gated mode sequence.

Figure 10-150. Gated Mode Sequence



dsi-064

(1) All thick-outlined blocks show operations performed by software. Other blocks show operations performed by hardware.

(2) HSDIVIDER is used if any of the following bits is set to 1:

- DSS\_CTRL[12] LCD2\_CLK\_SWITCH
- DSS\_CTRL[0] LCD1\_CLK\_SWITCH
- DSS\_CTRL[1] DSI1\_CLK\_SWITCH
- DSS\_CTRL[10] DSI2\_CLK\_SWITCH

(3) For more information, see [Section 10.1 Display Subsystem Overview](#).

Table 10-490. Register Call Summary for Gated Mode Sequence

Register Name	Register Name	Register Name
<a href="#">DSI_PLL_CONFIGURATION2</a>	<a href="#">DSI_PLL_CONTROL</a>	<a href="#">DSI_PLL_GO</a>

#### 10.3.4.5.9 DSI PLL Clock Sequence

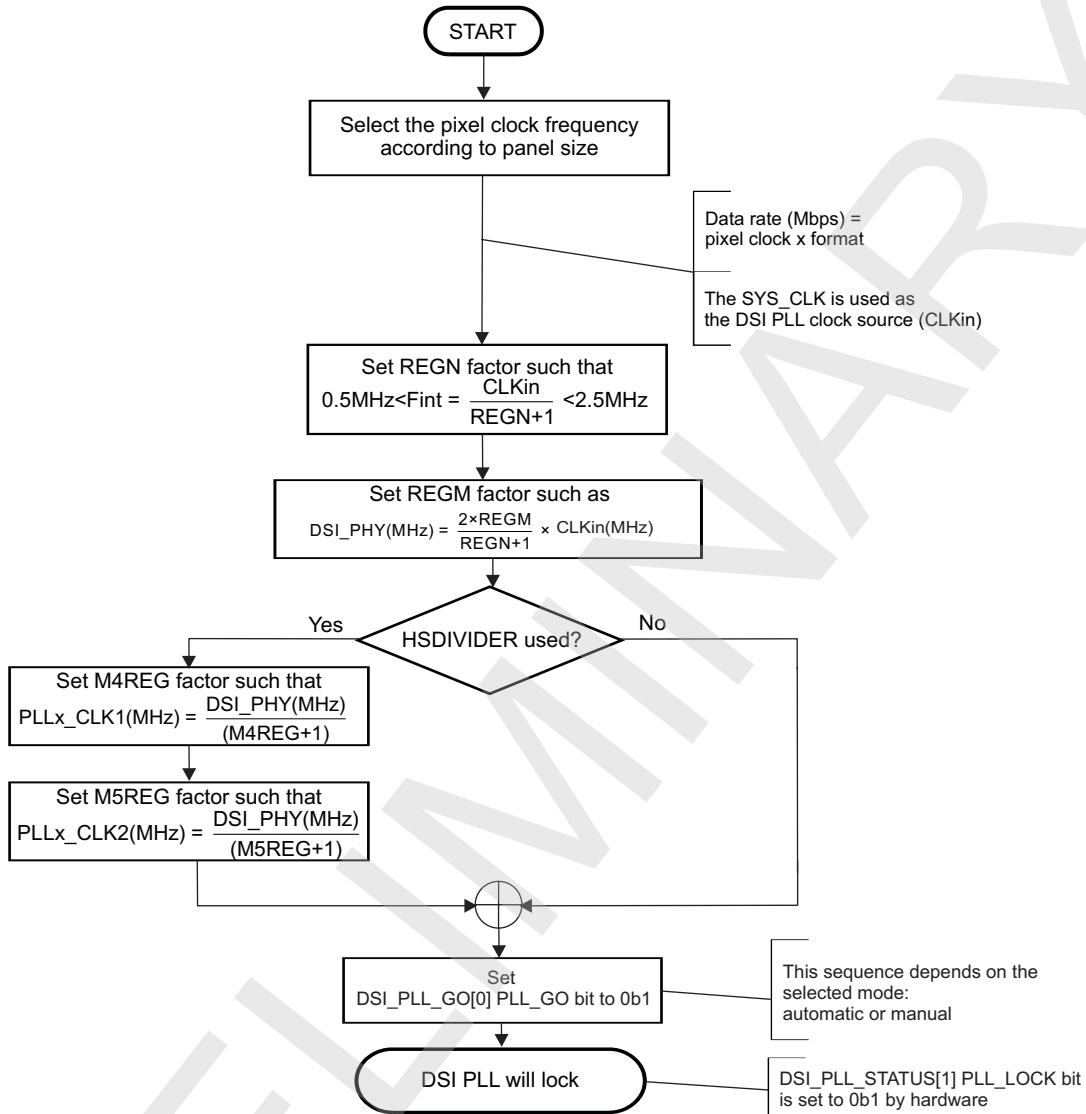
The two DSI PLLs generate the CLKIN4DDR clocks. Each HSDIVIDER outputs two clocks: PLLx\_CLK1, connected to the DISPC; and PLLx\_CLK2, connected to the DSI protocol engine (where x is the number of the dedicated DSI PLLs [1 or 2]). If these two clocks are not used, the HSDIVIDER functions are not required. In addition, the HSDIVIDER of the DSI1 PLL generates one more clock, PLL1\_CLK4, which is used as bypass input for DSI2\_PHY.

The CLKIN4DDR clocks are twice the data rate and four times the DSI output clock frequency. The DSI PLL factors must be calculated based on the required input and output frequencies, keeping the PLL internal reference frequency in the appropriate range:

- REGM factor is programmed by the [DSI\\_PLL\\_CONFIGURATION1](#)[20:9] PLL\_REGM bit field.
- REGN factor is programmed by the [DSI\\_PLL\\_CONFIGURATION1](#)[8:1] PLL\_REGN bit field.
- M4REG factor is programmed by the [DSI\\_PLL\\_CONFIGURATION1](#)[25:21] M4\_CLOCK\_DIV bit field, and applies to PLLx\_CLK1.
- M5REG factor is programmed by the [DSI\\_PLL\\_CONFIGURATION1](#)[30:26] M5\_CLOCK\_DIV bit field, and applies to PLLx\_CLK2.
- M7 factor is programmed by the [DSI\\_PLL\\_CONFIGURATION3](#)[9:5] M7\_CLOCK\_DIV field, and applies to PLL1\_CLK4.

[Figure 10-151](#) shows the programming sequence.

Figure 10-151. DSI PLL Programming Sequence



dsi-067

**NOTE:**

- The M4REG and M5REG factors must be set respecting the following conditions:
  - The PLLx\_CLK1 frequency must be a multiple of PCLK frequency (for proper settings of the PCD and LCD factors in the DISPC).
  - The PLLx\_CLK1 and PLLx\_CLK2 frequencies must be lower than 173 MHz.
- Most of the other DSI PLL programming values are available for software flexibility, but it is not recommended to update the values in normal use. For the recommended DSI PLL values, see Section 10.3.4.5.11, *DSI PLL Recommended Values*.

Table 10-491. Register Call Summary for DSI PLL Programming Sequence

Register Name	Register Name	Register Name
DSI_PLL_CONFIGURATION2	DSI_PLL_GO	DSI_PLL_STATUS

### 10.3.4.5.10 DSI PLL Go Sequence

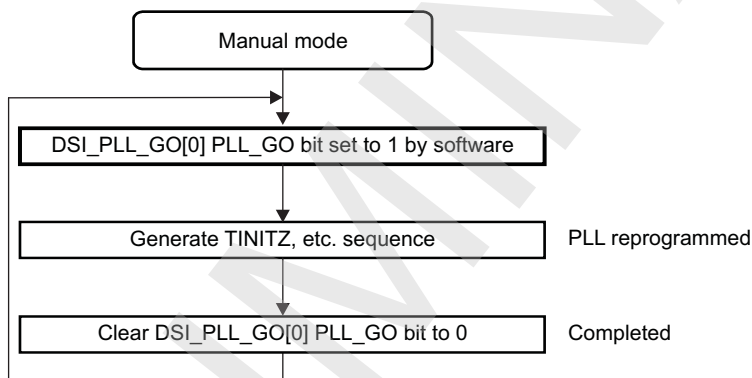
In manual mode (the [DSI\\_PLL\\_CONTROL\[0\]](#) PLL\_AUTOMODE bit is set to 0), the DPLL requires a sequence on TINITZ, TENABLE, and TENABLEDIV to update the configuration values and start the locking sequence.

When all the configuration values are programmed into the registers, the GO bit must be set. The appropriate sequence is then sent on the TINITZ, TENABLE, and TENABLEDIV pins, respecting the timing requirements of the ADPLL. The [DSI\\_PLL\\_GO\[0\]](#) PLL\_GO bit is cleared to 0 at the end of the sequence.

The TENABLEDIV signal is shared with the HSDIVIDER module so that it is programmed at the same time. In this mode, software must deassert CLKINEN by unsetting the [DSI\\_PLL\\_CONFIGURATION2\[14\]](#) PHY\_CLKINEN to 0 and assert HSDIVBYPASS correctly by setting the [DSI\\_PLL\\_CONFIGURATION2\[20\]](#) HSDIVBYPASS bit to 1 to prevent uncontrolled frequencies affecting DSI\_PHY and the display subsystem during PLL locking. In manual mode, the shadow register is updated anyway so that valid values are present when later selecting automatic mode.

[Figure 10-152](#) shows the DSI PLL Go flow chart in manual mode (the [DSI\\_PLL\\_CONTROL\[0\]](#) PLL\_AUTOMODE bit is set to 0).

**Figure 10-152. DSI PLL Go Sequence (Manual Mode)**



dsi-062

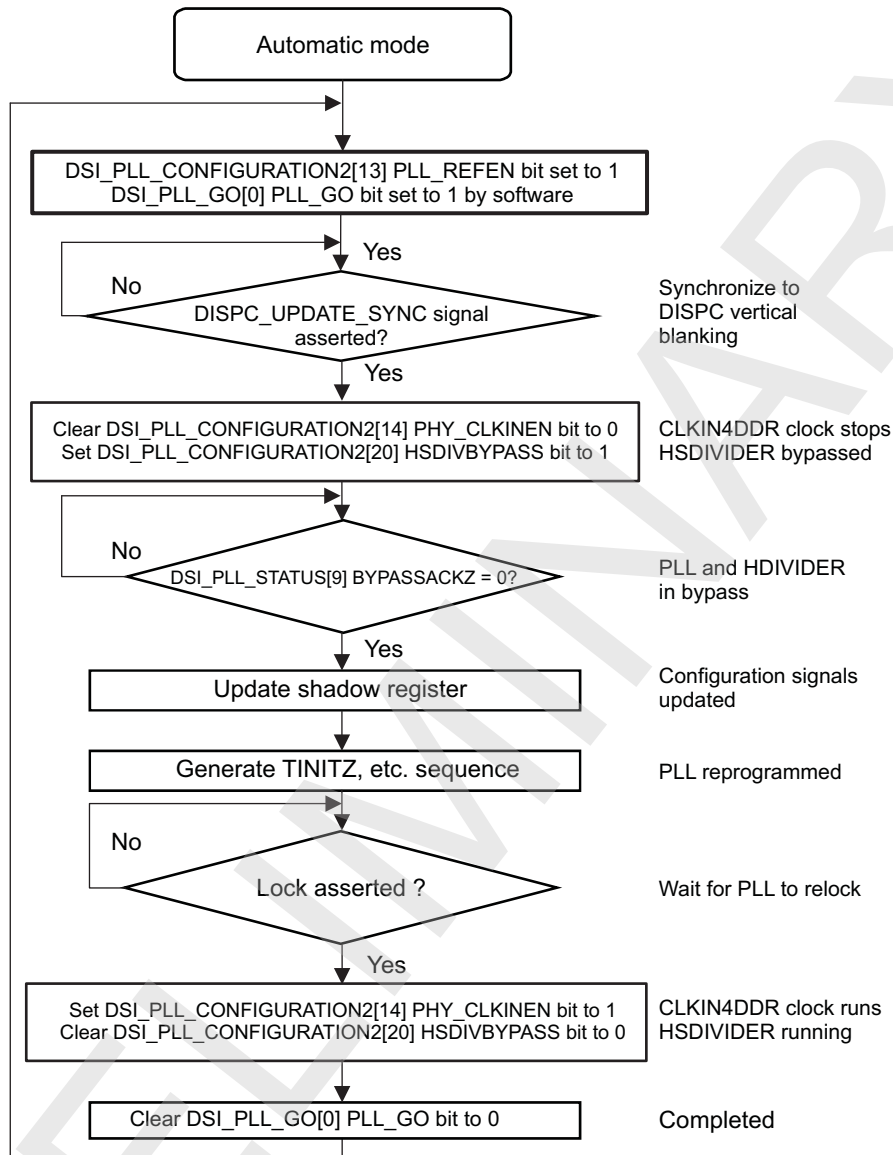
- (1) All thick-outlined blocks show operations performed by software. Other blocks show operations performed by hardware.

In automatic mode (the [DSI\\_PLL\\_CONTROL\[0\]](#) PLL\_AUTOMODE bit is set to 1), the TINITZ, TENABLE, and TENABLEDIV sequence and the update of the PLL configuration from the [DSI\\_PLL\\_CONFIGURATION2](#) register are deferred until the time of the front porch time signal sent by the DISPC module. This is intended to simplify the software to implement a configuration change (such as a frequency change to support a different link bandwidth). In this mode, CLKINEN, HSDIVBYPASS, and REFEN are automatically controlled and the register value is overridden.

[Figure 10-153](#) shows the DSI PLL Go flow chart in automatic mode (the [DSI\\_PLL\\_CONTROL\[0\]](#) PLL\_AUTOMODE bit is set to 1).



Figure 10-153. DSI PLL Go Sequence (Automatic Mode)



dsi-063

(1) All thick-outlined blocks show operations performed by software. Other blocks show operations performed by hardware.

Table 10-492. Register Call Summary for DSI PLL Go Sequence (Automatic Mode)

Register Name	Register Name	Register Name
DSI_PLL_CONFIGURATION2	DSI_PLL_STATUS	DSI_PLL_GO

10.3.4.5.11 DSI PLL Recommended Values

Table 10-493 shows the DSI PLL recommended values.

Table 10-493. Recommended Programming Values

Field Name	Value	Description
DSI_PLL_CONTROL[4] HSDIV_SYSRESET	0	Allow power FSM to control
DSI_PLL_CONTROL[3] PLL_SYSRESET	0	Allow power FSM to control

**Table 10-493. Recommended Programming Values (continued)**

Field Name	Value	Description
DSI_PLL_CONTROL[2] PLL_HALTMODE	-	For information, see <a href="#">Section 10.3.4.5.8, DSI PLL Clock-Gating Sequence</a> .
DSI_PLL_CONTROL[1] PLL_GATEMODE	-	For information, see <a href="#">Section 10.3.4.5.8, DSI PLL Clock-Gating Sequence</a> .
DSI_PLL_CONTROL[0] PLL_AUTOMODE	-	For information, see <a href="#">Section 10.3.4.5.8, DSI PLL Clock-Gating Sequence</a> .
DSI_PLL_GO[0] PLL_GO	1 -> 0	Write 1 when PLL is to be (re)locked with new parameters. This bit is cleared by hardware when the PLL request completes.
DSI_PLL_CONFIGURATION1[30:36] M5_CLOCK_DIV	See <sup>(1)</sup> .	DSI protocol engine clock divider
DSI_PLL_CONFIGURATION1[25:21] M4_CLOCK_DIV	See <sup>(1)</sup> .	DSS clock divider
DSI_PLL_CONFIGURATION1[20:9] PLL_REGM	See <sup>(1)</sup> .	Feedback clock divider
DSI_PLL_CONFIGURATION1[8:1] PLL_REGN	See <sup>(2)</sup> .	Reference clock divider
DSI_PLL_CONFIGURATION1[0] PLL_STOPMODE	1	Required to use the GATEMODE bit
DSI_PLL_CONFIGURATION2[20] HSDIVBYPASS	0	PLL controls HSDIVIDER bypass
DSI_PLL_CONFIGURATION2[19] M5_CLOCK_PWDN	0	If PLL/HSDIVIDER is used as the DSI protocol clock source
DSI_PLL_CONFIGURATION2[18] M5_CLK_EN	1	If PLL/HSDIVIDER is used as the DSI protocol clock source
DSI_PLL_CONFIGURATION2[17] M4_CLOCK_PWDN	0	If PLL/HSDIVIDER is used as the clock source
DSI_PLL_CONFIGURATION2[16] M4_CLOCK_EN	1	If PLL/HSDIVIDER is used as the clock source
DSI_PLL_CONFIGURATION2[15] BYPASSEN	0	To use PLL as the clock source. For small displays, it may be possible to use the display subsystem functional clock, DSS_CLK, in which case this bit must be set to 1. See <a href="#">Section 10.3.4.2, DSI Clock Configuration</a> .
DSI_PLL_CONFIGURATION2[14] PHY_CLKINEN	1	Enable CLKIN4DDR clock
DSI_PLL_CONFIGURATION2[13] PLL_REFEN	1	Enable PLL reference
DSI_PLL_CONFIGURATION2[10:9] PLL_LOCKSEL	0x0	Phase lock criteria to lock the PLL
DSI_PLL_CONFIGURATION2[8] PLL_DRIFTGUARDEN	0x0	The RECAL status/interrupt must be used to decide when to perform a PLL uncalibration. No automatic uncalibration is performed
DSI_PLL_CONFIGURATION2[6] PLL_LOWCURRSTDBY	0/1	Set to 0 for fast PLL unlock, but higher standby current. Set to 1 for leakage level standby current, but longer unlock time.
DSI_PLL_CONFIGURATION2[5] PLLLPMODE	0	Normal operation. For smaller display sizes, it may be possible to set to 1.
DSI_PLL_CONFIGURATION2[0] PLL_IDLE	0	PLL active

<sup>(1)</sup> The value of the bit field must be set according to the desired clock frequency.

<sup>(2)</sup> The value of the bit field must be set according to the desired clock frequency.

### 10.3.4.6 DSI Complex I/O

#### 10.3.4.6.1 DSI Complex I/O Overview

DSI\_PHY is a complex I/O with five unidirectional (HS) lane modules. This includes four data lane modules and one clock lane module. Each lane module has two data pads (DX, DY). These data pads are connected with a complementary lane module on the DSI receiver device using a point-to-point interconnect.

Lane modules support HS burst mode. Forward direction and reverse direction escape modes are also supported. Escape modes can be used for low-power data transmission, among other things.

The maximum data rate supported in HS mode is 824 Mbps for four data lanes, and 900 Mbps for up to three data lanes. The lane module function and position are configurable; that is, any lane module can be chosen as a clock lane module, and the DX/DY data pad for each lane module can be configured as a DP or DN pin defined by the D-PHY specification.

DSI\_PHY interacts with the higher layers of the DSI link through the PPI. DSI\_PHY does not include a PLL; a high-frequency clock input is expected in HS mode (CLKIN4DDR). DSI\_PHY also supports the SCP interface to set various configuration and control registers.

#### 10.3.4.6.2 Software Reset

The clock domain using the TXBYTECLKHS byte clock from the DSI complex I/O has dedicated reset-done information in the [DSI\\_COMPLEXIO\\_CFG1\[29\]](#) RESET\_DONE bit. The [DSI\\_SYSCONFIG\[1\]](#) SOFT\_RESET bit is used to reset the byte clock power domain. A dummy read using the SCP interface to any DSI\_PHY register is required after DSI\_PHY reset to complete the reset of the DSI complex I/O.

#### 10.3.4.6.3 Reset-Done Bits

The DSI complex I/O has several clock domains. The reset status for each clock domain is provided in the [DSI\\_PHY\\_REGISTERS5](#) register:

- [DSI\\_PHY\\_REGISTERS5\[31\]](#) RESETDONETXBYTECLK bit: Reset done for the TXBYTECLK domain.
- [DSI\\_PHY\\_REGISTERS5\[28:24\]](#) RESETDONETXCLKESC<sub>i</sub> bit field: Reset done for the TXCLKESC domain for lane *i* (where *i* = 0 to 4).
- [DSI\\_PHY\\_REGISTERS5\[30\]](#) RESETDONESCPCCLK bit: Reset done for the SCP clock domain. Software users must perform a dummy read on this bit to initiate the reset sequence of the SCP FSM. When the reset sequence completes, the RESETDONESCPCCLK signal goes high, and software users can read again the [DSI\\_PHY\\_REGISTERS5\[30\]](#) RESETDONESCPCCLK bit to ensure that the value is now 1.

---

**NOTE:** Software must not write in the DSI\_PHY SCP registers before the [DSI\\_PHY\\_REGISTERS5\[30\]](#) RESETDONESCPCCLK bit is set to 1.

---

- [DSI\\_PHY\\_REGISTERS5\[29\]](#) RESETDONEPWRCLK bit: Reset done for the PWR clock domain. The reset sequence of the PWR FSM is complete when the RESETDONEPWRCLK signal goes high.

#### 10.3.4.6.4 Pad Configuration

The number of lanes is configurable through the [DSI\\_COMPLEXIO\\_CFG1](#) register.

It is not allowed to change on the fly the position (by modifying the DATA<sub>x</sub>\_POSITION, where *x* = 1 through 4, and CLOCK\_POSITION fields), the P/N order (positive/negative order of the differential pair by modifying the DATA<sub>y</sub>\_POL, where *y* = 1 through 4, and CLOCK\_POL), or the number of active data lanes (by modifying the [DSI\\_COMPLEXIO\\_CFG1\[10:8\]](#) DATA2\_POSITION, [DSI\\_COMPLEXIO\\_CFG1\[14:12\]](#) DATA3\_POSITION, and [DSI\\_COMPLEXIO\\_CFG1\[18:16\]](#) DATA4\_POSITION bit fields). To add or remove lanes 2, 3, or 4, it is required to be in OFF mode for the DSI complex I/O.

The configuration of the DSI complex I/O (number of data lanes, position, differential order) must not be changed while the [DSI\\_CLK\\_CTRL\[20\]](#) LP\_CLK\_ENABLE bit is set to 1. For hardware to recognize a new configuration of the complex I/O (done in the [DSI\\_COMPLEXIO\\_CFG1](#) register), the following sequence shall be followed:

1. Set the [DSI\\_CTRL\[0\]](#) IF\_EN bit to 1.
  2. Reset the [DSI\\_CTRL\[0\]](#) IF\_EN bit to 0.
  3. Set [DSI\\_CLK\\_CTRL\[20\]](#) LP\_CLK\_ENABLE bit to 1.
  4. Set again the [DSI\\_CTRL\[0\]](#) IF\_EN bit to 1.
- If the sequence is not followed, the DSI complex I/O configuration is undetermined.

The minimum requirement for the number of lanes is one clock lane and one data lane. By default, data lanes 2, 3, and 4 are not connected (the reset value of the DATAx\_POSITION bit fields is 0).

### 10.3.4.6.5 Display Timing Configuration

---

**NOTE:** Copyright 2005-2008 MIPI Alliance, Inc. All rights reserved. MIPI Alliance Member Confidential.

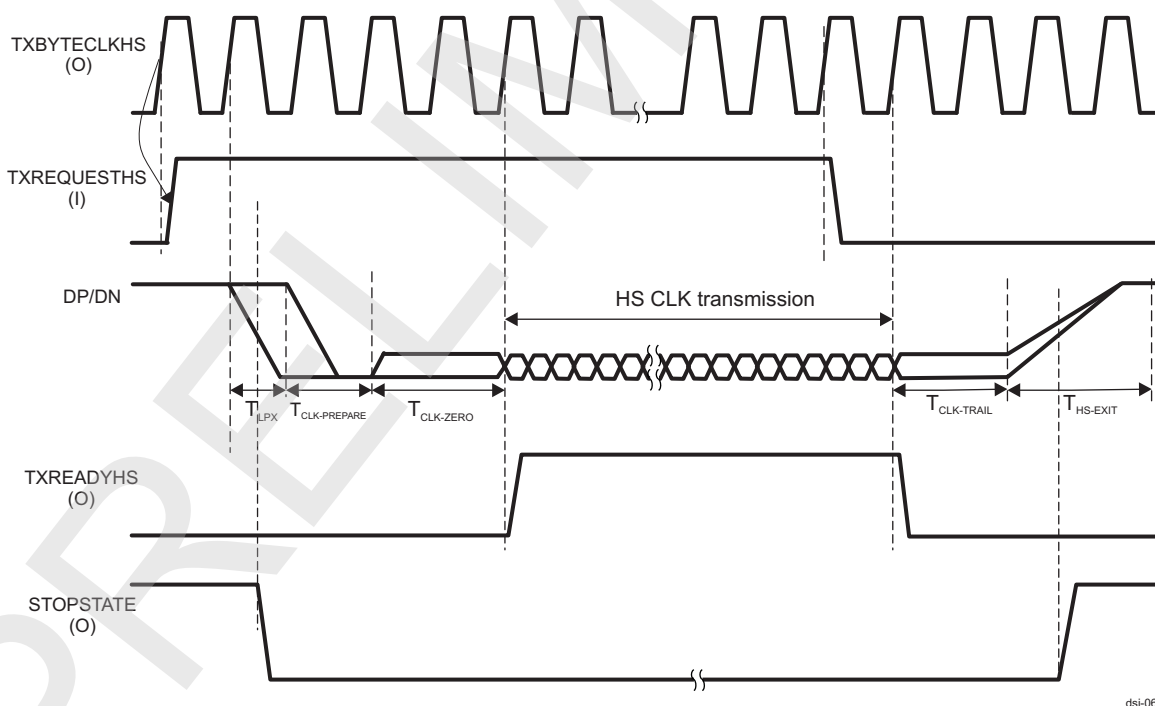
---

Depending on the CLKIN4DDR clock frequency settings programmed with the DSI PLL control module, software users must program accordingly the timing parameters in the DSI complex I/O registers.

#### 10.3.4.6.5.1 High-Speed Clock Transmission

Figure 10-154 shows an example of HS clock transmission.

**Figure 10-154. HS Clock Transmission**



TXBYTECLKHS is an output clock that is derived by dividing CLKIN4DDR by 16. The TXBYTECLKHS clock is driven by DSI\_PHY when:

- The CLKIN4DDR is running and [DSI\\_PLL\\_CONFIGURATION2\[14\]](#) PHY\_CLKINEN bit is set to '1'.
- DSI\_PHY is in ON power state ([DSI\\_COMPLEXIO\\_CFG1\[28:27\]](#) PWR\_CMD bit = 1) and at least one lane is enabled.

To begin transmission, the protocol drives TXREQUESTHS high on a rising edge of TXBYTECLKHS. The PHY detects this signal on the next rising edge, after which it initiates the LP SoT procedure.

During an HS clock transmission, these parameters are defined in multiples of CLKIN4DDR and are programmed by the following bit fields:

- TLPX timing is programmed by [DSI\\_PHY\\_REGISTER1\[20:16\]](#) TLPXBY2.
- THS-PREPARE timing is programmed by [DSI\\_PHY\\_REGISTER0\[31:24\]](#) THSPREPARE.
- TCLK-ZERO timing is programmed by [DSI\\_PHY\\_REGISTER1\[7:0\]](#) TCLKZERO.

TCLK-ZERO is extended, if required, so that the entire LP SoT procedure lasts an integer number of TXBYTECLKHS cycles.

At the end of the SoT procedure, HS clock transmission begins. At the same time, TXREADYHS is made high.

To stop clock transmission, the protocol drives TXREQUESTHS low on a rising edge of TXBYTECLKHS. The DSI\_PHY detects this change in TXREQUESTHS on the next edge and stops clock transmission. TXREADYHS is made low.

The DSI\_PHY then goes through the LP EoT procedure. The TCLK-TRAIL and THS-EXIT parameters are also multiples of CLKIN4DDR and are programmed by the following bit fields:

- TCLK-TRAIL timing is programmed by [DSI\\_PHY\\_REGISTER1\[15:8\]](#) TCLKTRAIL.
- THS-EXIT timing is programmed by [DSI\\_PHY\\_REGISTER0\[7:0\]](#) THSEXIT.

The DSI\_PHY completes the SoT and EoT procedures, once begun, regardless of any change in the PPI signals. If TXREQUESTHS goes low during the SoT procedure, the PHY starts the EoT procedure immediately after finishing the SoT procedure and no clock is transmitted.

STOPSTATE is high whenever the line is in LP-11 state, as determined by the outputs of the low-power receivers. This signal is not synchronized with TXBYTECLKHS.

The HS clock must be present for a period of time before (TCLK-PRE) and after (TCLK-POST) HS data transmission. The protocol must ensure that these timings are met by asserting and deasserting TXREQUESTHS appropriately.

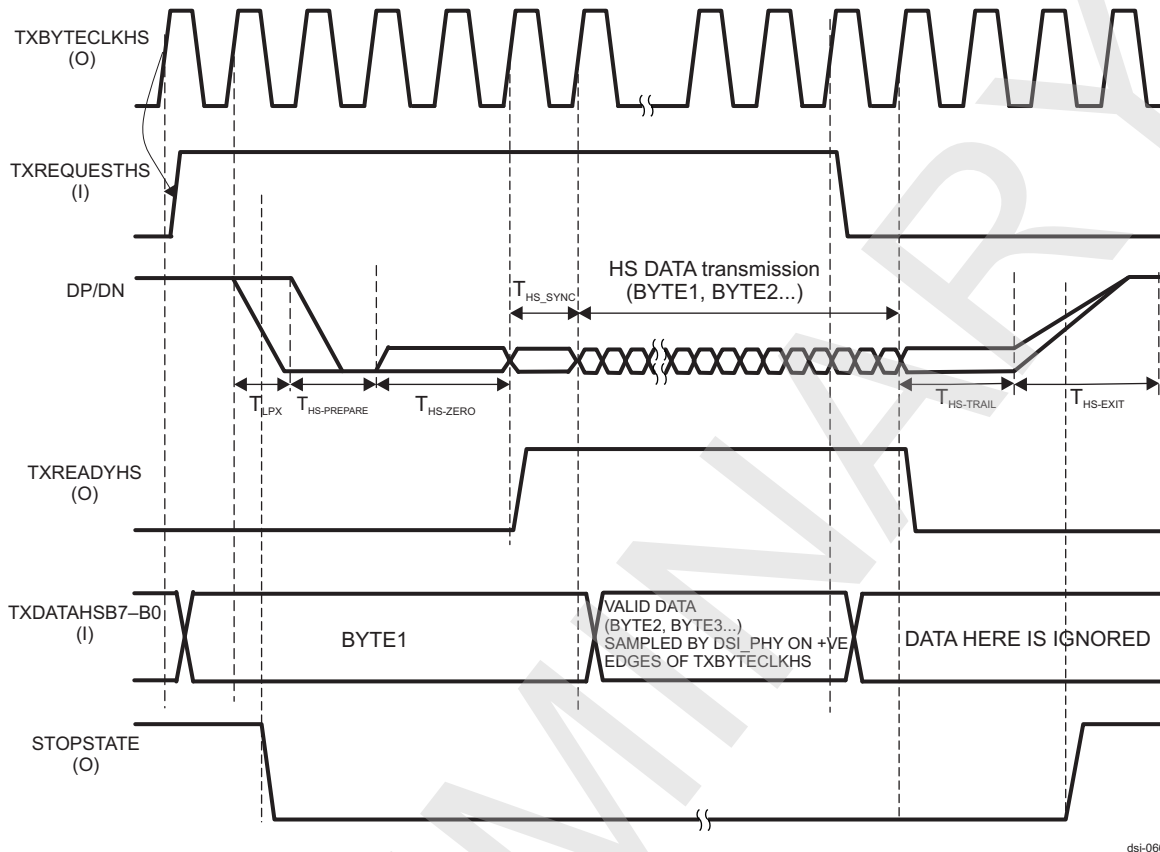
The PHY ensures that the clock signal has a quadrature phase with respect to a toggling bit sequence on any data lane, and a rising edge in the center of the first transmitted bit of each data byte. These relations are not described in the timing diagram.

CLKIN4DDR can be shut off 300 ns after the clock lane goes to STOPSTATE. Alternatively, CLKIN4DDR can be shut down after TCLK-Trail + THS-Exit + 2 Txbyteclk periods after the TxRequestHS falling edge is received by DSI\_PHY.

The DSI protocol engine must ensure that TXREQUESTESC, TXULPSCLK, and TURNREQUEST are low whenever TXREQUESTHS is asserted.

#### **10.3.4.6.5.2 High-Speed Data Transmission**

[Figure 10-155](#) shows an example of HS data transmission.

**Figure 10-155. HS Data Transmission**

To begin transmission, the protocol drives TXDATAHS with the first byte of data on a rising edge of TXBYTECLKHS. It also makes TXREQUESTHS high on the same rising edge. The PHY detects TXREQUESTHS going high on the next rising edge of TXBYTECLKHS, following which it initiates the LP SoT procedure.

During an HS data transmission, these timings are multiples of CLKIN4DDR and are programmed by the following bit fields:

- TLPX timing is programmed by [DSI\\_PHY\\_REGISTER1\[20:16\]](#) TLPXBY2.
- THS-PREPARE + THS-ZERO timing is programmed by [DSI\\_PHY\\_REGISTER0\[23:16\]](#) THSPRPR\_THSZERO.

THS-ZERO is extended, if required, so that the entire LP SoT procedure lasts an integer number of TXBYTECLKHS cycles. THS-SYNC corresponds to the length of the sync pattern (8 HS bits), and can be configured through the [DSI\\_PHY\\_REGISTER2\[31:24\]](#) HSSYNCPATTERN bit field.

Toward the end of the SoT procedure, the PHY makes TXREADYHS high on a positive edge of TXBYTECLKHS and then starts accepting data from TXDATAHS from the next positive edge onward. The protocol is expected to provide (new) valid data on TXDATAHS on every positive edge of TXBYTECLKHS if TXREADYHS is high.

At the end of the SoT procedure, HS data transmission begins. HS data transmission happens LSB first.

To stop data transmission, the protocol drives TXREQUESTHS low on a rising edge of TXBYTECLKHS. The PHY detects this change in TXREQUESTHS on the next edge and stops data transmission. TXREADYHS is made low and data on TXDATAHS, from that point, is ignored.

The PHY then goes through the LP EoT procedure. THS-TRAIL and THS-EXIT are also multiples of CLKIN4DDR and are programmed by the following bit fields:

- THS-TRAIL timing is programmed by [DSI\\_PHY\\_REGISTER0\[15:8\]](#) THSTRAIL.
- THS-EXIT timing is programmed by [DSI\\_PHY\\_REGISTER0\[7:0\]](#) THSEXIT.

The PHY completes the SoT and EoT procedures, once begun, regardless of any change in PPI signals. If TXREQUESTHS goes low during the SoT procedure, the PHY starts the EoT procedure immediately after finishing the SoT procedure and no data is transmitted.

STOPSTATE is high whenever the line is in LP-11 state, as determined by the outputs of the low-power receivers. This signal is not synchronized with TXBYTECLKHS.

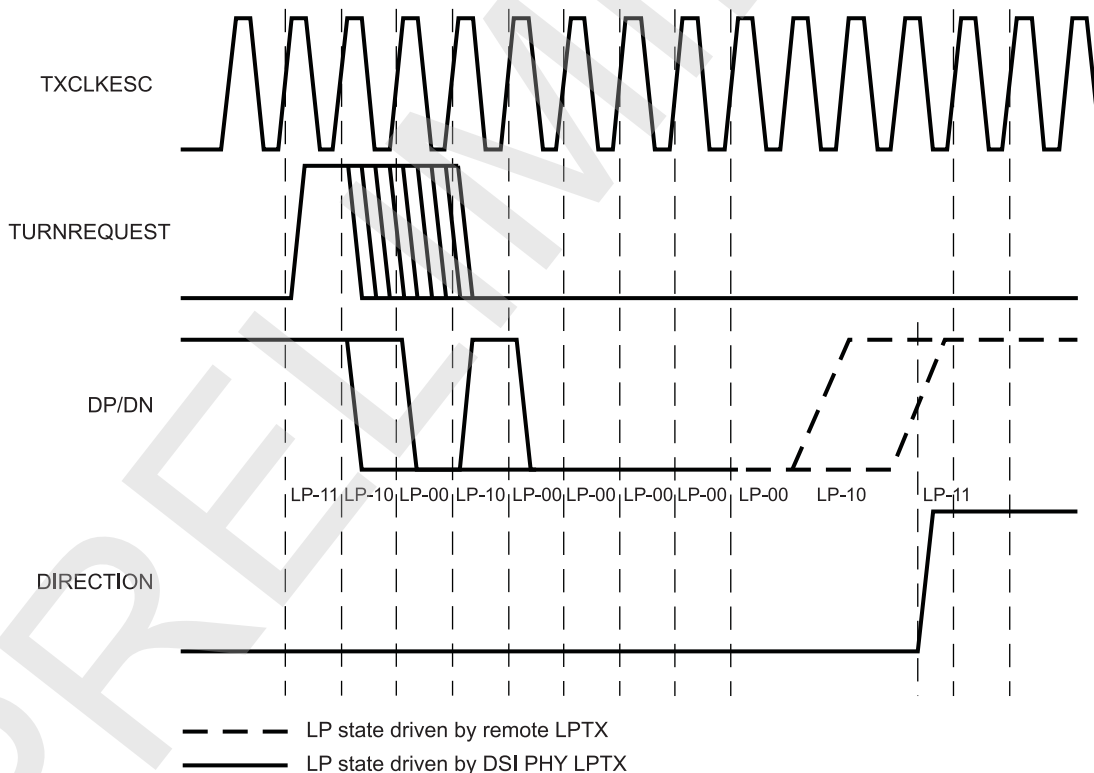
The protocol must ensure that TXREQUESTESC, TXULPSCLK, and TURNREQUEST are low whenever TXREQUESTHS is asserted.

### 10.3.4.6.5.3 Turnaround Request in Transmit Mode

When the DSI PHY is in transmit mode, the DSI protocol engine can request a turnaround by making the TurnRequest signal high for at least one clock cycle of TxClkEsc (see Section 10.3.4.4.7.3, TurnRequest FSM).

The DSI PHY transmits the turnaround request pattern (LP 11-10-00-10-00-00-00) (see Figure 10-156). The number of 00 states at the end of the pattern is defined by the  $T_{TA-GO}$  timing parameter and is programmable through the DSI\_PHY\_REGISTER1[31:29] REG\_TTAGO bit field, in number of TxClkEsc clocks. Following the transmission of the pattern, the DSI PHY disables its LP transmitters and waits for an acknowledgment from the remote device. The remote device detects the turnaround request and acknowledges it by driving LP 10, followed by the STOP state. When this acknowledgment is received, the DSI PHY switches to receive mode and indicates the completion of the turnaround procedure by changing the direction (BTA\_IRQ is asserted, as described in Section 10.3.4.4.8, Bus Turnaround).

Figure 10-156. Turnaround Request in Transmit Mode



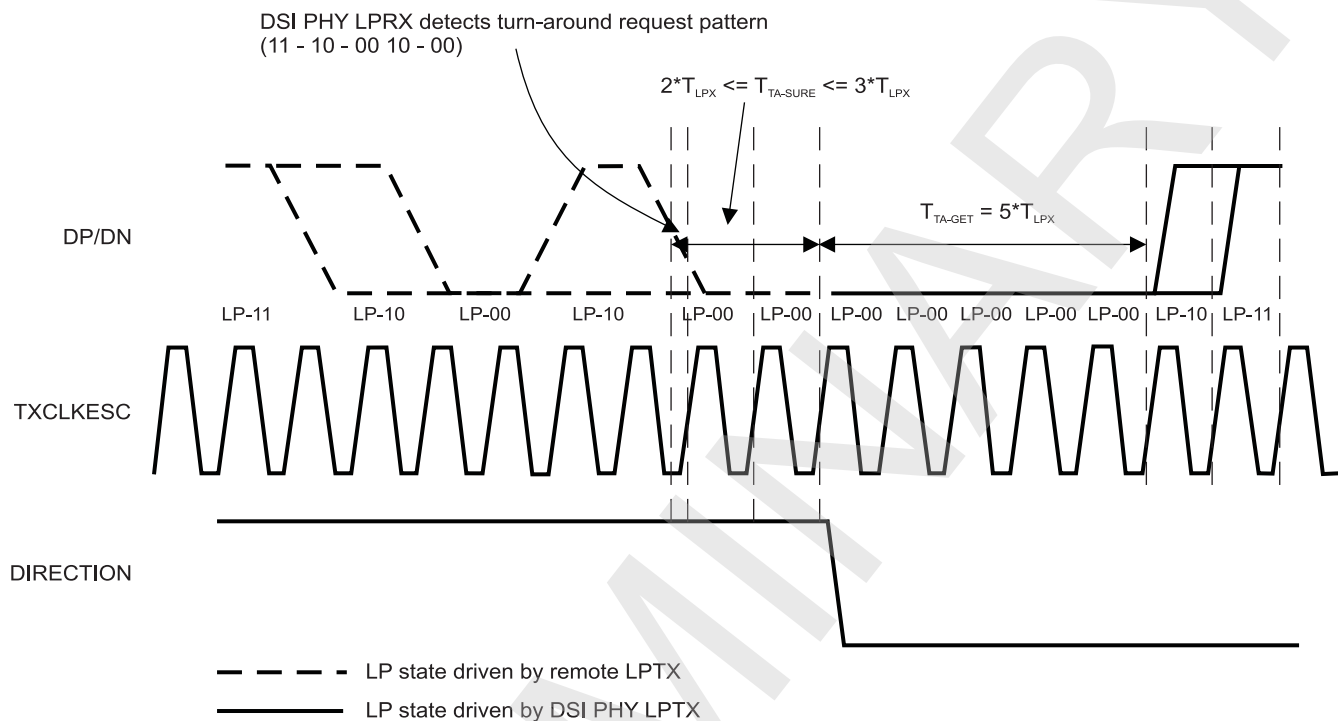
dsii-028

The DSI protocol engine must not stop TxClkEsc after the turnaround process completes (the DSI\_CLK\_CTRL[20] LP\_CLK\_ENABLE bit must be kept at 1), because TxClkEsc is also used in handling a turnaround request transmitted by a remote slave device (see Section 10.3.4.6.5.4, Turnaround Request in Receive Mode).

#### 10.3.4.6.5.4 Turnaround Request in Receive Mode

When DSI PHY is in receive mode, an LP pattern of 11-10-00-10-00 on DP/DN lines indicates a turnaround request from the remote device (see [Figure 10-157](#)).

**Figure 10-157. Turnaround Request in Receive Mode**



dsi-029

If the line stays in LP-00 for a time  $T_{TA-SURE}$ , the DSI PHY accepts the turnaround request, changes the direction, transmits LP-00 for a time  $T_{TA-GET}$ , and then transmits the acknowledgment pattern LP-10, followed by the STOP state.

This completes the turnaround procedure. The  $T_{TA-SURE}$  and  $T_{TA-GET}$  timing parameters are programmable through [DSI\\_PHY\\_REGISTER1](#) in number of TxClkEsc clocks:

- $T_{TA-SURE}$  can be configured in the [28:27] REG\_TTASURE bit field.
- $T_{TA-GET}$  can be configured in the [26:24] REG\_TTAGET bit field.

#### 10.3.4.6.5.5 Other DSI PHY Transmission and Reception

The timings of the following sequences defined in the DSI\_PHY protocol cannot be programmed by the user:

- Low-power data transmission
- Escape mode trigger command transmission
- ULPS command transmission on data lanes
- ULPS transmission on clock lane
- Low-power data in receive mode
- Low-power trigger in receive mode
- ULPS command on clock lane in receive mode
- ULPS command on data lane in receive mode

#### 10.3.4.6.6 Power Management

Power management of the DSI\_PHY analog circuitry is done through the internal power FSM, which is controlled by the subsystem. The control interface is a basic command and acknowledge protocol. The



subsystem is the master of the interface. There is a dedicated command pin for each power state of DSI\_PHY and a common acknowledge pin. The subsystem sends the power command and clock. The internal FSM does the sequencing and control generation of internal power domains. It then asserts the acknowledge, which signals the completion of the power state transition. The internal power control FSM is powered by a constant power supply.

The main features of the power control interface are:

- One dedicated command line per power state
- Only one command line can be asserted at a time.
- One acknowledge signal common to all command lines
- The command line cannot be cleared unless the acknowledge has been returned.
- All the command lines and the acknowledge signal must be cleared before triggering a new command.

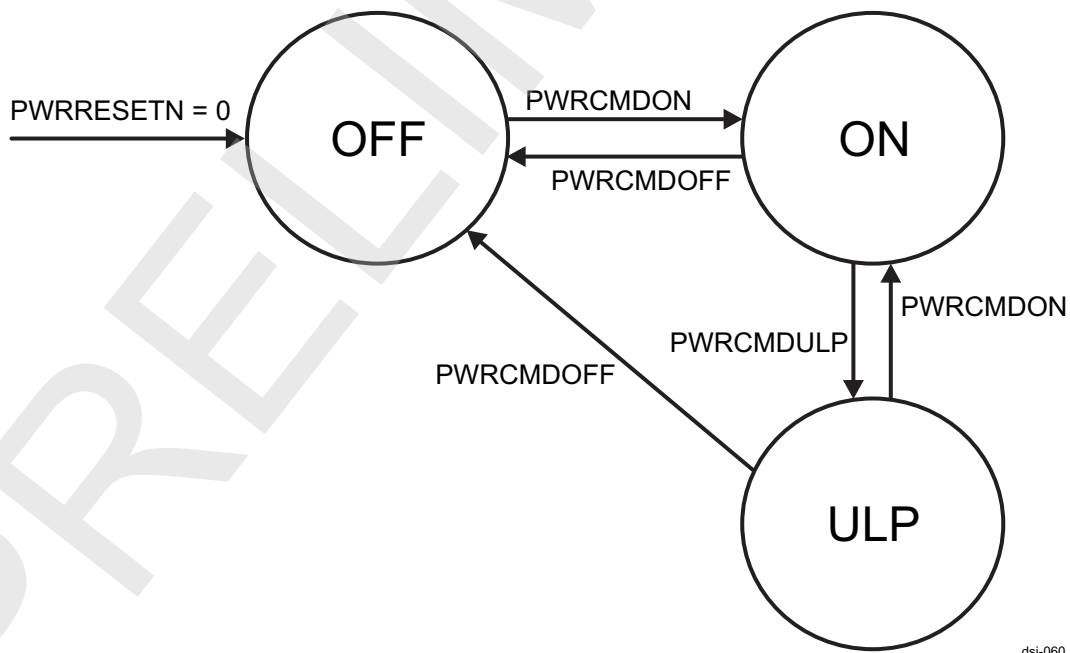
DSI\_PHY supports three power states:

- OFF/GPIO mode (OFF): The entire analog circuit is powered down. Internal LDO is OFF. All analog circuitry on 1.8-V supply is powered down. The ULPS exit detection circuit power switch is OFF. DSI\_PHY takes leakage level current.
- ON: Complete analog circuit is powered on and functional. LDO is ON. (DSI\_PHY calibrates every time the module enters power on state.)
- ULPS: Only the DSI\_PHY mode ULPS exit detection circuit power switch is ON. All the other analog circuitry is switched OFF. LDO is OFF.

**NOTE:** ULPS must be entered only when all used lane modules go into ULPS state. PHY power state must be returned to ON power state when any one of the used modules goes out of ULPS. No PPI signal except TXULPSEXIT must be toggled when in this power state.

Figure 10-158 shows the power states of the DSI complex I/O

Figure 10-158. Power State Diagram



#### 10.3.4.6.7 Error Handling

A dedicated register for the DSI complex I/O ( [DSI\\_COMPLEXIO\\_IRQSTATUS](#)) indicates the state of each error provided by the DSI complex I/O error signals. The DSI\_PHY reports the following errors:

- [DSI\\_COMPLEXIO\\_IRQSTATUS](#)[9:5] ERRESC<sub>i</sub>\_IRQ: ERRESC is asserted if an unrecognized Escape entry command is received. This remains high until the next change in the line state.
- [DSI\\_COMPLEXIO\\_IRQSTATUS](#)[4:0] ERRSYNCESC<sub>i</sub>\_IRQ: If the number of bits received during a low-power data transmission is not a multiple of 8 when the transmission ends, ERRSYNCESC is made high and remains high until the next change in line state.
- [DSI\\_COMPLEXIO\\_IRQSTATUS](#)[14:10] ERRCONTROL<sub>i</sub>\_IRQ: ERRCONTROL is asserted if an incorrect line state sequence is detected. For example, if a turnaround request or escape mode request is immediately followed by a STOP state instead of the required BRIDGE state, this signal is asserted and remains asserted until the next change in the line state.
- [DSI\\_COMPLEXIO\\_IRQSTATUS](#) ERRCONTENTIONLP0<sub>i</sub>\_IRQ: ERRCONTENTION0LPDX and ERRCONTENTION0LPDY are asserted when the lane module detects a contention situation on lines DX and DY, respectively, while trying to drive the lines low. Contention is detected only if it lasts at least 50 ns.
- [DSI\\_COMPLEXIO\\_IRQSTATUS](#) ERRCONTENTIONLP1<sub>i</sub>\_IRQ: ERRCONTENTION1LPDX and ERRCONTENTION1LPDY are asserted when the lane module detects a contention situation on lines DX and DY, respectively, while trying to drive the lines high. Contention is detected only if it lasts at least 50 ns.

The ULPSACTIVENOT signal goes low, which indicates to the protocol that the PHY has entered ULPS. When all the ULPSActiveNot signals are low, the [DSI\\_COMPLEXIO\\_IRQSTATUS](#)[30] ULPSACTIVENOT\_ALL0\_IRQ event is generated. When all the ULPSActiveNot signals are high, the [DSI\\_COMPLEXIO\\_IRQSTATUS](#)[31] ULPSACTIVENOT\_ALL1\_IRQ event is generated.

When any of the events defined in the [DSI\\_COMPLEXIO\\_IRQSTATUS](#) register happen, the [DSI\\_IRQSTATUS](#)[10] COMPLEXIO\_ERR\_IRQ bit is set to 1 at DSI protocol engine level.

[Table 10-494](#) lists the DSI complex I/O interrupt events.

**Table 10-494. DSI Complex I/O Interrupts**

IRQ Name	Description
ULPSActiveNot_ALL0_IRQ	All the ULPSActiveNOT signals are 0.
ULPSActiveNot_ALL1_IRQ	All the ULPSActiveNOT signals corresponding to the lanes with TXULPSExit being high are high.
STATEULPM5_IRQ	Lane 5 in ULPS
STATEULPM4_IRQ	Lane 4 in ULPS
STATEULPM3_IRQ	Lane 3 in ULPS
STATEULPM2_IRQ	Lane 2 in ULPS
STATEULPM1_IRQ	Lane 1 in ULPS
ERRCONTROL5_IRQ	Control error for lane 5
ERRCONTROL4_IRQ	Control error for lane 4
ERRCONTROL3_IRQ	Control error for lane 3
ERRCONTROL2_IRQ	Control error for lane 2
ERRCONTROL1_IRQ	Control error for lane 1
ERRESC5_IRQ	Escape entry error for lane 5 (edge trigger interrupt)
ERRESC4_IRQ	Escape entry error for lane 4 (edge trigger interrupt)
ERRESC3_IRQ	Escape entry error for lane 3 (edge trigger interrupt)
ERRESC2_IRQ	Escape entry error for lane 2 (edge trigger interrupt)
ERRESC1_IRQ	Escape entry error for lane 1 (edge trigger interrupt)
ERRCONTENTIONLP1_1_IRQ	Contention LP1 error for lane 1
ERRCONTENTIONLP0_1_IRQ	Contention LP0 error for lane 1
ERRCONTENTIONLP1_2_IRQ	Contention LP1 error for lane 2
ERRCONTENTIONLP0_2_IRQ	Contention LP0 error for lane 2
ERRCONTENTIONLP1_3_IRQ	Contention LP1 error for lane 3
ERRCONTENTIONLP0_3_IRQ	Contention LP0 error for lane 3
ERRCONTENTIONLP1_4_IRQ	Contention LP1 error for lane 4

**Table 10-494. DSI Complex I/O Interrupts (continued)**

<b>IRQ Name</b>	<b>Description</b>
ERRCONTENTIONLP0_4_IRQ	Contention LP0 error for lane 4
ERRCONTENTIONLP1_5_IRQ	Contention LP1 error for lane 5
ERRCONTENTIONLP0_5_IRQ	Contention LP0 error for lane 5
ERRSYNCEESC5_IRQ	Low-power data transmission synchronization error for lane 5
ERRSYNCEESC4_IRQ	Low-power data transmission synchronization error for lane 4
ERRSYNCEESC3_IRQ	Low-power data transmission synchronization error for lane 3
ERRSYNCEESC2_IRQ	Low-power data transmission synchronization error for lane 2
ERRSYNCEESC1_IRQ	Low-power data transmission synchronization error for lane 1

Software must take appropriate action when receiving the interrupt indicating the error from the complex I/O. The action can be:

- Reset the DSI protocol engine module
- Reset the peripheral through the reset trigger or directly driving the hardware reset pin of the display module
- Ignore the error

### 10.3.5 DSI Programming Model

This section describes the low-level hardware programming sequences for the configuration and use of the DSI module.

#### 10.3.5.1 DSI Global Initialization

##### 10.3.5.1.1 Surrounding Modules Global Initialization

This section describes the requirements to initialize the surrounding modules when the DSI module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DSI. For more information, see [Section 10.3.3, DSI Integration](#), and [Section 10.3.2, DSI Environment](#).

[Table 10-495](#) describes the global initialization of the surrounding modules.

**Table 10-495. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	DSI module interface and functional clocks must be enabled. The module power management (idle and sleep modes) must be configured. For more information, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
MPU INTC	MPU interrupt controller (INTC) configuration must be done to enable interrupts from the DSI module. For more information about the module configuration, see <a href="#">Chapter 17, Interrupt Controllers</a> .
DSP INTC	DSP INTC configuration must be done to enable interrupts from the DSI module. For more information about the module configuration, see <a href="#">Chapter 5, DSP Subsystem</a> .
sDMA	DMA configuration must be done to enable DSI module DMA channel requests. For more information about the configuration, see <a href="#">Chapter 16, sDMA</a> .
L3 / L4_PER Interconnect	For information about the interconnect configuration, see <a href="#">Chapter 13, Interconnect</a> .

##### 10.3.5.1.2 DSI Global Initialization

###### 10.3.5.1.2.1 Main Sequence - DSI Global Initialization

This procedure initializes the DSI after a power on or software reset.

[Table 10-496](#) summarizes the DSI global initialization.

**Table 10-496. DSI Global Initialization**

Steps	Registers	Value
Reset IRQ status.	<a href="#">DSI_IRQSTATUS</a> [31:0]	0x0
Set clock activity.	<a href="#">DSI_SYSCONFIG</a> [9:8] CLOCKACTIVITY	x
Configure power management.	<a href="#">DSI_SYSCONFIG</a> [4:3] SIDLEMODE	x
Reset DSI protocol engine.	<a href="#">DSI_SYSCONFIG</a> [1] SOFT_RESET	0x1
Wait until five reads of reset status?	Software test condition	
Read reset status.	<a href="#">DSI_SYSSTATUS</a> [0] RESET_DONE	
<b>IF:</b> Is reset ongoing?	<a href="#">DSI_SYSSTATUS</a> [0] RESET_DONE	= 0x0
Error occurred during reset stage <sup>(1)</sup>		
<b>ENDIF</b>		

### 10.3.5.2 DSI Operational Modes Configuration

#### 10.3.5.2.1 DSI Video Mode Using the DISPC Video Port

##### 10.3.5.2.1.1 Main Sequence - DSI PLL Setup

Table 10-497 lists the steps required to configure DSI PLL.

**Table 10-497. DSI PLL Configuration Registers**

Steps	Registers	Value
Turn ON/OFF the PLL and HSDIVIDER.	DSI_CLK_CTRL[31:30] PLL_PWR_CMD	x
Wait until the status of the PLL is set (value according to the previous step)?	DSI_CLK_CTRL[29:28] PLL_PWR_STATUS	= x
Set the divider value for DSI protocol engine clock source M5REG.	DSI_PLL_CONFIGURATION1[30:26] M5_CLOCK_DIV	x
Set the divider value for clock source M4REG.	DSI_PLL_CONFIGURATION1[25:21] M4_CLOCK_DIV	x
Set the value for M divider for PLL.	DSI_PLL_CONFIGURATION1[20:9] DSI_PLL_REGM	x
Set the value for N divider for PLL.	DSI_PLL_CONFIGURATION1[8:1] DSI_PLL_REGN	x
Enable/disable PLL STOPMODE.	DSI_PLL_CONFIGURATION1[0] PLL_STOPMODE	x
Enable/disable DSI protocol engine clock divider.	DSI_PLL_CONFIGURATION2[18] M5_CLOCK_EN	x
Enable/disable M4 clock source divider.	DSI_PLL_CONFIGURATION2[16] M4_CLOCK_EN	x
Enable/disable DSI PHY clock.	DSI_PLL_CONFIGURATION2[14] PHY_CLKINEN	x
Enable/disable PLL reference clock.	DSI_PLL_CONFIGURATION2[13] PLL_REFEN	x
Set PLL automatic update mode.	DSI_PLL_CONTROL[0] PLL_AUTOMODE	x
Request PLL locking sequence.	DSI_PLL_GO[0] PLL_GO	0x1
Read until PLL_GO bit = 0?	DSI_PLL_GO[0] PLL_GO	= 0x0
Wait until PLL is locked?	DSI_PLL_STATUS[1] PLL_LOCK	= 0x1
Set the ratio for the LP clock.	DSI_CLK_CTRL[12:0] LP_CLK_DIVISOR	x
Enable/disable DSS_L3_ICLK clock gating to DSI_PHY and PLL module.	DSI_CLK_CTRL[14] CIO_CLK_ICG	x
Enable/disable the automatic assertion/deassertion of DSIStopClk.	DSI_CLK_CTRL[18] HS_AUTO_STOP_ENABLE	x
Configure the DSI functional clock synchronization and clock speed.	DSI_CLK_CTRL[21] LP_RX_SYNCHRO_ENABLE	x
Turn ON/OFF PLL and HSDIVIDER.	DSI_CLK_CTRL[31:30] PLL_PWR_CMD	x

##### 10.3.5.2.1.2 Main Sequence - Set Up DSI Protocol Engine

###### 10.3.5.2.1.2.1 Set Up DSI Control Registers

Table 10-498 lists the steps to set up the DSI control registers. Table 10-499 lists the steps to set up the DSI complex I/O registers.

**Table 10-498. DSI Control Registers**

Steps	Registers	Value
Enable the interrupt event for lost synchronization with the video port.	DSI_IRQENABLE[18] SYNC_LOST_IRQ_EN	0x1
Enable interrupt event for sent packet on desired VC.	DSI_VC_IRQENABLE_i[2] PACKET_SENT_IRQ_EN	x
Enable HSYNC END pulse detection (short packet generation).	DSI_CTRL[17] VP_HSYNC_START	0x1
Enable VSYNC START pulse detection (short packet generation).	DSI_CTRL[15] VP_VSYNC_START	0x1
Select trigger reset mode.	DSI_CTRL[14] TRIGGER_RESET_MODE	x

**Table 10-498. DSI Control Registers (continued)**

Steps	Registers	Value
Set the number of line buffers.	DSI_CTRL[13:12] LINE_BUFFER	x
Set video port HSYNC signal polarity.	DSI_CTRL[10] VP_HSYNC_POL	x
Set video port VSYNC signal polarity.	DSI_CTRL[11] VP_VSYNC_POL	x
Set video port data enable signal polarity.	DSI_CTRL[9] VP_DE_POL	x
Set the size of the video port data bus.	DSI_CTRL[7:6] VP_DATA_BUS_WIDTH	x
Set the arbitration scheme for granting the VC pending ready requests in the TX FIFO.	DSI_CTRL[3] TX_FIFO_ARBITRATION	x
Enable/disable the ECC check for the received header.	DSI_CTRL[2] ECC_RX_EN	0x1

**Table 10-499. DSI Complex I/O Registers**

Steps	Registers	Value
Configure synchronized update on the shadow register.	DSI_COMPLEXIO_CFG1[30] GOBIT	x
Set the complex I/O power state.	DSI_COMPLEXIO_CFG1[28:27] PWR_CMD	x
Set the position of data lane 1.	DSI_COMPLEXIO_CFG1[6:4] DATA1_POSITION	x
Set the position of the clock lane.	DSI_COMPLEXIO_CFG1[2:0] CLOCK_POSITION	x
Clear the Interrupt status register.	DSI_COMPLEXIO_IRQSTATUS	0xFFFF FFFF
Disable all interrupt events.	DSI_COMPLEXIO_IRQENABLE	0x0
Enable the DSI protocol engine.	DSI_CTRL[0] IF_EN	0x1
Disable the DSI protocol engine.	DSI_CTRL[0] IF_EN	0x0
Wait until the interface is disabled?	DSI_CTRL[0] IF_EN	= 0x0
Enable the low-power clock (TXCLKESC).	DSI_CLK_CTRL[20] LP_CLK_ENABLE	0x1
Wait until reset is done?	DSI_COMPLEXIO_CFG1[29] RESET_DONE	= 0x1
Check if power control is ON?	DSI_COMPLEXIO_CFG1[26:25] PWR_STATUS	= 0x1
Wait until internal module reset is complete?	DSI_SYSSTATUS[0] RESETDONE	= 0x1

### 10.3.5.2.1.2.2 Configure DSI Timing and VCs

Table 10-500 lists the steps to configure DSI timing and the VCs.

**Table 10-500. DSI Timing Registers**

Steps	Registers	Value
Determine the number of DSI_FCLK clock cycles for the STOP-STATE counter.	DSI_TIMING1[12:0] STOP_STATE_COUNTER_IO	0x-
Determine the number of BYTE_CLK clock cycles for the HS RX timer.	DSI_TIMING2[28:16] HS_TX_TO_COUNTER	0x-
Enable/disable the multiplication factor of 16 for the number of BYTE_CLK clock cycles for the HS TX timer.	DSI_TIMING2[29] HS_TX_TO_X16	0x-
Determine the number of DSI_FCLK clock cycles for the LP RX timer.	DSI_TIMING2[12:0] LP_RX_TO_COUNTER	0x-
Enable/disable the multiplication factor of 16 for the number of DSI_FCLK clock cycles for the LP RX timer.	DSI_TIMING2[14] LP_RX_TO_X16	0x-
Set the horizontal sync active period.	DSI_VM_TIMING1[31:24] HSA	0x-
Set the horizontal front porch.	DSI_VM_TIMING1[23:12] HFP	0x-
Set the horizontal back porch.	DSI_VM_TIMING1[11:0] HBP	0x-
Set the number of BYTE clock cycles for the sync window.	DSI_VM_TIMING2[27:24] WINDOW_SYNC	0x-
Set the vertical sync active period.	DSI_VM_TIMING2[23:16] VSA	0x-

**Table 10-500. DSI Timing Registers (continued)**

Steps	Registers	Value
Set the vertical front porch.	DSI_VM_TIMING2[15:8] VFP	0x-
Set the vertical back porch.	DSI_VM_TIMING2[7:0] VBP	0x-
Set the line length.	DSI_VM_TIMING3[31:16] TL	0x-
Set the number of active lines.	DSI_VM_TIMING3[15:0] VACT	0x-
Set the number of TXBYTECLKHS clock cycles for entering HS mode.	DSI_VM_TIMING7[31:16] ENTER_HS_MODE_LATENCY	0x-
Set the number of TXBYTECLKHS clock cycles for exiting HS mode.	DSI_VM_TIMING7[15:0] EXIT_HS_MODE_LATENCY	0x-
Set the number of TXBYTECLKHS clock cycles between the start of the DDR clock and the assertion of the data request signal.	DSI_CLK_TIMING[15:8] DDR_CLK_PRE	0x-
Set the number of TXBYTECLKHS clock cycles after the deassertion of the data request signal and the stop of the DDR clock.	DSI_CLK_TIMING[7:0] DDR_CLK_POST	0x-
Configure the RX FIFO DMA request.	DSI_VC_CTRL_i[29:27] DMA_RX_REQ_NB	0x-
Configure the TX FIFO DMA request.	DSI_VC_CTRL_i[23:21] DMA_TX_REQ_NB	0x-
Enable/disable the ECC generation for the transmit header.	DSI_VC_CTRL_i[8] ECC_TX_EN	0x-
Enable/disable the checksum generation for the transmit payload.	DSI_VC_CTRL_i[7] CS_TX_EN	0x-
Select video mode.	DSI_VC_CTRL_i[4] MODE	0x-

**10.3.5.2.1.3 Main Sequence - Configure DSI\_PHY**

Table 10-501 summarizes the timing in the functions of DDR\_CLK\_P. For more information about the timing calculation, see Section 10.3.4.4.4, *Clock Requirements*.

**Table 10-501. Calculate DSI\_PHY Timing**

Steps	Registers	Value
Settings of the DSI protocol timing. For a complete description of the timing specifications, see Section 10.3.4.4.4, <i>Clock Requirements</i> .	DSI_PHY_REGISTER0[31:24] REG_THSPREPARE	ceil(70ns/DDR clock period) + 2
	DSI_PHY_REGISTER0[23:16] REG_PRPR_THSZERO	ceil(175ns/DDR clock period) + 2
	DSI_PHY_REGISTER0[7:0] REG_THSEXIT	ceil(145ns/DDR clock period)
	DSI_PHY_REGISTER0[15:8] REG_THSTRAIL	ceil(60ns/DDR clock period) + 5
	DSI_PHY_REGISTER2[7:0] REG_TCLKPREPARE	ceil(65ns/DDR clock period)
	DSI_PHY_REGISTER1[7:0] REG_TCLKZERO	ceil(265ns/DDR clock period)
	DSI_PHY_REGISTER1[15:8] REG_TCLKTRAIL	ceil(60ns/DDR clock period) + 2
	DSI_PHY_REGISTER1[20:16] REG_TLPXBY2	ceil(25ns/DDR clock period)

**10.3.5.2.1.4 Main Sequence - Drive STOP State**

Table 10-502 lists the steps to drive STOP state.

**Table 10-502. Drive STOP State**

Steps	Registers	Value
Force TX stop mode.	DSI_TIMING1[15] FORCE_TX_STOP_MODE_IO	0x1
Wait until FORCE_TX_STOP_MODE_IO = 0?	DSI_TIMING1[15] FORCE_TX_STOP_MODE_IO	= 0x0

### 10.3.5.2.1.5 Main Sequence - Enable Video Mode

Table 10-502 lists the steps to enable video mode.

**Table 10-503. Enable Video Mode**

Steps	Registers	Value
Set up long packet header.	<a href="#">DSI_VC_LONG_PACKET_HEADER_i</a>	0x-
Enable desired VC.	<a href="#">DSI_VC_CTRL_i[1]</a> VC_EN	0x1
Enable DSI module.	<a href="#">DSI_CTRL[0]</a> IF_EN	0x1
Wait until IF_EN = 1?	<a href="#">DSI_CTRL[0]</a> IF_EN	= 0x1

### 10.3.5.2.2 DSI Command Mode Using the DISPC Video Port

This section explains the basic programming model of command mode using the DISPC video port.

#### 10.3.5.2.2.1 Main Sequence - Configure DSI PLL

Table 10-504 lists the steps required to configure the DSI PLL.

**Table 10-504. Configure DSI PLL**

Steps	Register/Bit Field/Programming	Value
Set PLL and HSDIVIDER to ON state.	<a href="#">DSI_CLK_CTRL[31:30]</a> PLL_PWR_CMD	0x2
Wait until PLL and HSDIVIDER are enabled?	<a href="#">DSI_CLK_CTRL[29:28]</a> PLL_PWR_STATUS	= 0x2
Set the M5REG value.	<a href="#">DSI_PLL_CONFIGURATION1[30:26]</a> M5_CLK_DIV	x
Set the M4REG value.	<a href="#">DSI_PLL_CONFIGURATION1[25:21]</a> M4_CLOCK_DIV	x
Set the REGN value.	<a href="#">DSI_PLL_CONFIGURATION1[8:1]</a> PLL_REGN	x
Set the REGM value.	<a href="#">DSI_PLL_CONFIGURATION1[20:9]</a> PLL_REGM	x
Select PLL STOPMODE.	<a href="#">DSI_PLL_CONFIGURATION1[0]</a> PLL_STOPMODE	0x1
Enable/disable PLL reference clock control.	<a href="#">DSI_PLL_CONFIGURATION2[13]</a> PLL_REFEN	x
Enable/disable CLKIN4DDR control.	<a href="#">DSI_PLL_CONFIGURATION2[14]</a> PHY_CLKINEN	x
Enable/disable DSS clock divider.	<a href="#">DSI_PLL_CONFIGURATION2[16]</a> M4_CLOCK_EN	x
Enable/disable DSI protocol engine clock divider.	<a href="#">DSI_PLL_CONFIGURATION2[18]</a> M5_CLOCK_EN	x
Configure synchronization of configuration update with DISPC_UPDATE_SYNC.	<a href="#">DSI_PLL_CONTROL[0]</a> PLL_AUTOMODE	x
Start PLL locking sequence.	<a href="#">DSI_PLL_GO[0]</a> PLL_GO	0x1
Wait until there is no pending action?	<a href="#">DSI_PLL_GO[0]</a> PLL_GO	= 0x0
Check whether PLL is locked?	<a href="#">DSI_PLL_STATUS[1]</a> PLL_LOCK	= 0x1
Set the LP mode clock ratio.	<a href="#">DSI_CLK_CTRL[12:0]</a> LP_CLK_DIVISOR	x
Configure DSS_L3_ICLK clock to the DSI complex I/O.	<a href="#">DSI_CLK_CTRL[14]</a> CIO_CLK_ICG	x
Enable/disable the automatic assertion/deassertion of the DSISopClk signal.	<a href="#">DSI_CLK_CTRL[18]</a> HS_AUTO_STOP_ENABLE	x
Define the DSI functional clock value and synchronization.	<a href="#">DSI_CLK_CTRL[21]</a> LP_RX_SYNCHRO_ENABLE	x
Turn on PLL and HSDIVIDER.	<a href="#">DSI_CLK_CTRL[31:30]</a> PLL_PWR_CMD	0x2

#### 10.3.5.2.2.2 Main Sequence - Switch to DSI PLL Clock Source

Table 10-505 lists the sequence to switch the DSI and DISPC module clocks to DSI PLL clock source.



**Table 10-505. Switch to DSI PLL Clock Source**

Steps	Reference	Value
Switch DISPC clock to PLL1_CLK1.	For more information, see <a href="#">Section 10.1.2, DSS Integration</a>	
Switch DSI clock to PLL1_CLK2.	For more information, see <a href="#">Section 10.1.2, DSS Integration</a> .	

### 10.3.5.2.2.3 Main Sequence - Configure DSI Protocol Engine

#### 10.3.5.2.2.3.1 Set Up DSI Control Registers

[Table 10-506](#) lists the steps required to set up the DSI control registers. [Table 10-507](#) lists the steps to set up the DSI complex I/O registers.

**Table 10-506. DSI Control Registers**

Steps	Register/Bit Field/Programming	Value
Enable IRQ for lost synchronization with video port.	<a href="#">DSI_IRQENABLE</a> [18] SYNC_LOST_IRQ_EN	0x1
Enable IRQ to indicate that packet has been sent on desired VC.	<a href="#">DSI_VC_IRQENABLE_i</a> [2] PACKET_SENT_IRQ_EN	0x1
Set the trigger reset mode.	<a href="#">DSI_CTRL</a> [14] TRIGGER_RESET_MODE	x
Configure the number of active line buffers.	<a href="#">DSI_CTRL</a> [13:12] LINE_BUFFER	x
Set the size of the video port data bus.	<a href="#">DSI_CTRL</a> [7:6] VP_DATA_BUS_WIDTH	x
Define the ratio between VP_CLK and VP_PCLK.	<a href="#">DSI_CTRL</a> [4] VP_CLK_RATIO	x
Set the arbitration scheme for granting the VC pending ready requests in the TX FIFO.	<a href="#">DSI_CTRL</a> [3] TX_FIFO_ARBITRATION	x
Enable/disable the ECC check for the received header.	<a href="#">DSI_CTRL</a> [2] ECC_RX_EN	x

**Table 10-507. DSI Complex I/O Registers**

Steps	Register/Bit Field/Programming	Value
Determine the position of the clock lane.	<a href="#">DSI_COMPLEXIO_CFG1</a> [2:0] CLOCK_POSITION	x
Set the position of data lane 1.	<a href="#">DSI_COMPLEXIO_CFG1</a> [6:4] DATA1_POSITION	x
Configure the power state of the complex I/O.	<a href="#">DSI_COMPLEXIO_CFG1</a> [28:27] PWR_CMD	x
Enable the synchronization of the shadow registers with DISPC_UPDATE_SYNC.	<a href="#">DSI_COMPLEXIO_CFG1</a> [30] GOBIT	0x1
Clear the complex I/O IRQ status.	<a href="#">DSI_COMPLEXIO_IRQSTATUS</a>	0xFFFF FFFF
Disable all complex I/O IRQ events.	<a href="#">DSI_COMPLEXIO_IRQENABLE</a>	0x0
Enable the DSI protocol engine interface.	<a href="#">DSI_CTRL</a> [0] IF_EN	0x1
Disable the DSI protocol engine interface.	<a href="#">DSI_CTRL</a> [0] IF_EN	0x0
Wait until interface is disabled?	<a href="#">DSI_CTRL</a> [0] IF_EN	= 0x0
Enable the LP clock.	<a href="#">DSI_CLK_CTRL</a> [20] LP_CLK_ENABLE	0x1
Check whether internal reset is complete?	<a href="#">DSI_COMPLEXIO_CFG1</a> [29] RESET_DONE	= 0x1
Check whether power state of complex I/O is ON?	<a href="#">DSI_COMPLEXIO_CFG1</a> [26:25] PWR_STATUS	= 0x1
Check whether reset is complete?	<a href="#">DSI_SYSSTATUS</a> [0] RESETDONE	= 0x1

#### 10.3.5.2.2.3.2 Configure DSI Timing and VCs

[Table 10-508](#) lists the steps to configure DSI timing and the VCs.

**Table 10-508. DSI Timing Registers**

Steps	Register/Bit Field/Programming	Value
Determine the number of DSI_FCLK clock cycles for the STOP-STATE counter.	DSI_TIMING1[12:0] STOP_STATE_COUNTER_IO	0x-
Enable/disable the multiplication factor of 4 for the number of DSI_FCLK clock cycles for the STOP-STATE counter.	DSI_TIMING1[13] STOP_STATE_X4_IO	0x-
Enable/disable the multiplication factor of 16 for the number of DSI_FCLK clock cycles for the STOP-STATE counter.	DSI_TIMING1[14] STOP_STATE_X16_IO	0x-
Clear turnaround timer settings.	DSI_TIMING1[30:16]	0x0
Determine the number of DSI_FCLK clock cycles for the LP RX timer.	DSI_TIMING2[12:0] LP_RX_TO_COUNTER	0x-
Enable/disable the multiplication factor of 4 for the number of DSI_FCLK clock cycles for the LP RX timer.	DSI_TIMING2[13] LP_RX_TO_X4	0x-
Enable/disable the multiplication factor of 16 for the number of DSI_FCLK clock cycles for the LP RX timer.	DSI_TIMING2[14] LP_RX_TO_X16	0x-
Determine the number of BYTE_CLK clock cycles for the HS RX timer.	DSI_TIMING2[28:16] HS_TX_TO_COUNTER	0x-
Enable/disable the multiplication factor of 16 for the number of BYTE_CLK clock cycles for the HS TX timer.	DSI_TIMING2[29] HS_TX_TO_X16	0x-
Enable/disable the multiplication factor of 64 for the number of BYTE_CLK clock cycles for the HS TX timer.	DSI_TIMING2[30] HS_TX_TO_X64	0x-
Set the number of TXBYTECLKHS clock cycles	DSI_CLK_TIMING[15:8] DDR_CLK_PRE and DSI_CLK_TIMING[7:0] DDR_CLK_POST	0x-
<b>Configuration of VCi</b>		
Select source of data and enable VP_STALL.	DSI_VC_CTRL_i[1] SOURCE	0x1
Enable/disable the checksum generation for the transmit payload.	DSI_VC_CTRL_i[7] CS_TX_EN	0x-
Enable/disable the ECC generation for the transmit header.	DSI_VC_CTRL_i[8] ECC_TX_EN	0x-
Enable/disable HS mode to send short and long packets to the peripheral.	DSI_VC_CTRL_i[9] MODE_SPEED	0x-
Configure DMA request for TX FIFO.	DSI_VC_CTRL_i[23:21] DMA_TX_REQ_NB	0x-
Configure DMA request for RX FIFO.	DSI_VC_CTRL_i[29:27] DMA_RX_REQ_NB	0x-
<b>Configuration TX and RX FIFO</b>		
Set size of the RX FIFO allocated for used VC.	DSI_RX_FIFO_VC_SIZE	0x-
Set size of the TX FIFO allocated for used VC.	DSI_TX_FIFO_VC_SIZE	0x-

#### 10.3.5.2.2.4 Main Sequence - Configure DSI\_PHY Timing

Table 10-509 summarizes DSI\_PHY timing settings. For more information about timing calculation, see Section 10.3.4.4.4, *Clock Requirements*.

**Table 10-509. Configure DSI\_PHY Timing**

Steps	Register/Bit Field/Programming	Value
Settings of the DSI protocol timing. For a complete description of timing specifications, see <a href="#">Section 10.3.4.4.4, Clock Requirements</a> .	DSI_PHY_REGISTER0[31:24] REG_THSPREPARE	ceil(70ns/DDR clock period) + 2
	DSI_PHY_REGISTER0[23:16] REG_THSPRPR_THSZERO	ceil(175ns/DDR clock period) + 2
	DSI_PHY_REGISTER0[7:0] REG_THSEXIT	ceil(145ns/DDR clock period)
	DSI_PHY_REGISTER0[15:8] REG_THSTRAIL	ceil(60ns/DDR clock period) + 5
	DSI_PHY_REGISTER2[7:0] REG_TCLKPREPARE	ceil(65ns/DDR clock period)
	DSI_PHY_REGISTER1[7:0] REG_TCLKZERO	ceil(265ns/DDR clock period)
	DSI_PHY_REGISTER1[15:8] REG_TCLKTRAIL	ceil(60ns/DDR clock period) + 2
	DSI_PHY_REGISTER1[20:16] REG_TLPXBY2	ceil(25ns/DDR clock period)

**10.3.5.2.2.5 Main Sequence - Drive STOP State**

Table 10-510 lists the steps to drive the STOP state.

**Table 10-510. Drive STOP State**

Steps	Register/Bit Field/Programming	Value
Force TX stop mode.	DSI_TIMING1[15] FORCE_TX_STOP_MODE_IO	0x1
Wait until the end of TX stop mode assertion?	DSI_TIMING1[15] FORCE_TX_STOP_MODE_IO	= 0x1

**10.3.5.2.2.6 Main Sequence - Enable Command Mode Using DISPC Video Port**

Table 10-511 lists the steps to enable DISPC to send frames continuously. Two BTAs must be generated:

- The first BTA gives bus possession to the display module.
- The second BTA obtains the TE trigger.

**Table 10-511. Enable Command Mode and Automatic TE**

Steps	Register/Bit Field/Programming	Value
Insert DCS write memory continue/start code.	DSI_VC_CTRL_i[31] DCS_CMD_CODE	0x-
Enable/disable automatic insertion of DCS command codes when data is sourced by the video port.	DSI_VC_CTRL_i[30] DCS_CMD_ENABLE	0x-
Enable used VC.	DSI_VC_CTRL_i[0] VC_EN	0x1
Enable DSI protocol engine interface.	DSI_CTRL[0] IF_EN	0x1
Wait until interface is enabled?	DSI_CTRL[0] IF_EN	= 0x0
Send the sequence to receive the TE trigger from the peripheral.	DSI_VC_SHORT_PACKET_HEADER_i[31:0] HEADER	0x-
Wait until packet is sent?	DSI_VC_IRQSTATUS_i[2] PACKET_SENT_IRQ	= 0x1
Write 1 to reset the status bit.	DSI_VC_IRQSTATUS_i[2] PACKET_SENT_IRQ	0x1

**NOTE:** Keep Reserved bits at reset value in [DSI\\_PHY\\_REGISTER1](#) and [DSI\\_PHY\\_REGISTER2](#).

**10.3.5.2.2.7 Main Sequence - Send Frame Data to LCD Panel Using Automatic TE**

Table 10-512 summarizes the steps to send a frame data to the LCD panel using automatic TE.

**Table 10-512. Send Frame Data to LCD Panel Using Automatic TE**

Steps	Register/Bit Field/Programming	Value
Enable the transfer between DISPC and DSI. Reset after the transfer is done.	For more information, see <a href="#">Section 10.2</a> , <i>Display Controller</i> chapter.	
Specify the number of bytes to send. When DCS insertion is used, word count (WC) must include this one DCS byte.	<a href="#">DSI_VC_TE_i[23:0]</a> TE_SIZE	(WC+1)*LPP
Set up long packet header.	<a href="#">DSI_VC_LONG_PACKET_HEADER_i[31:0]</a> HEADER	0x-
Enable/disable TE control.	<a href="#">DSI_VC_TE_i[30]</a> TE_EN	0x1
Wait until RX FIFO is empty?	<a href="#">DSI_VC_CTRL_i[20]</a> RX_FIFO_NOT_EMPTY	= 0x0
Wait until TX FIFO is not full?	<a href="#">DSI_VC_CTRL_i[16]</a> TX_FIFO_FULL	= 0x0
Enable first BTA to give bus possession to the display module.	<a href="#">DSI_VC_CTRL_i[6]</a> BTA_EN	0x1
Wait until BTA IRQ event occurs?	<a href="#">DSI_VC_IRQSTATUS_i[5]</a> BTA_IRQ	= 0x1
Write 1 to clear BTA IRQ status bit.	<a href="#">DSI_VC_IRQSTATUS_i[5]</a> BTA_IRQ	0x1
Enable the second BTA to get the TE trigger.	<a href="#">DSI_VC_CTRL_i[6]</a> BTA_EN	0x1
Wait until BTA IRQ event occurs?	<a href="#">DSI_VC_IRQSTATUS_i[5]</a> BTA_IRQ	= 0x1
Write 1 to clear BTA IRQ status bit.	<a href="#">DSI_VC_IRQSTATUS_i[5]</a> BTA_IRQ	0x1
Wait until transfer is complete?	<a href="#">DSI_VC_TE_i[30]</a> TE_EN	= 0x0

### 10.3.6 DSI Register Manual

This section describes the DSI module registers.

**CAUTION**

The main access to all DSI registers is through the L3 interconnect.  
The access through L4\_PER interconnect is provided for back software compatibility.

#### 10.3.6.1 DSI Instance Summary

Table 10-513 summarizes the DSI module instances.

**Table 10-513. DSI Instance Summary**

Module Name	L3 Base Address	L4_PER Base Address	Size
DSI1_PROTOCOL_ENGINE	0x5800 4000	0x4804 4000	512 bytes
DSI1_PHY	0x5800 4200	0x4804 4200	64 bytes
DSI1_PLLCTRL	0x5800 4300	0x4804 4300	32 bytes
DSI2_PROTOCOL_ENGINE	0x5800 5000	0x4804 5000	512 bytes
DSI2_PHY	0x5800 5200	0x4804 5200	64 bytes
DSI2_PLLCTRL	0x5800 5300	0x4804 5300	32 bytes

#### 10.3.6.2 DSI\_PROTOCOL\_ENGINE Registers

##### 10.3.6.2.1 DSI\_PROTOCOL\_ENGINE Register Summary

**Table 10-514. DSI1\_PROTOCOL\_ENGINE Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSI1_PROTOCOL_ENGINE L3 Physical Address	DSI1_PROTOCOL_ENGINE L4_PER Physical Address
DSI_REVISION	R	32	0x0000 0000	0x5800 4000	0x4804 4000
DSI_SYSCONFIG	RW	32	0x0000 0010	0x5800 4010	0x4804 4010
DSI_SYSSTATUS	R	32	0x0000 0014	0x5800 4014	0x4804 4014
DSI_IRQSTATUS	RW	32	0x0000 0018	0x5800 4018	0x4804 4018
DSI_IRQENABLE	RW	32	0x0000 001C	0x5800 401C	0x4804 401C
DSI_CTRL	RW	32	0x0000 0040	0x5800 4040	0x4804 4040
DSI_GNQ	R	32	0x0000 0044	0x5800 4044	0x4804 4044
DSI_COMPLEXIO_CFG1	RW	32	0x0000 0048	0x5800 4048	0x4804 4048
DSI_COMPLEXIO_IRQSTATUS	RW	32	0x0000 004C	0x5800 404C	0x4804 404C
DSI_COMPLEXIO_IRQENABLE	RW	32	0x0000 0050	0x5800 4050	0x4804 4050
DSI_CLK_CTRL	RW	32	0x0000 0054	0x5800 4054	0x4804 4054
DSI_TIMING1	RW	32	0x0000 0058	0x5800 4058	0x4804 4058
DSI_TIMING2	RW	32	0x0000 005C	0x5800 405C	0x4804 405C
DSI_VM_TIMING1	RW	32	0x0000 0060	0x5800 4060	0x4804 4060
DSI_VM_TIMING2	RW	32	0x0000 0064	0x5800 4064	0x4804 4064
DSI_VM_TIMING3	RW	32	0x0000 0068	0x5800 4068	0x4804 4068
DSI_CLK_TIMING	RW	32	0x0000 006C	0x5800 406C	0x4804 406C

**Table 10-514. DSI1\_PROTOCOL\_ENGINE Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DSI1_PROTOCOL_ENGINE L3 Physical Address	DSI1_PROTOCOL_ENGINE L4_PER Physical Address
DSI_TX_FIFO_VC_SIZE	RW	32	0x0000 0070	0x5800 4070	0x4804 4070
DSI_RX_FIFO_VC_SIZE	RW	32	0x0000 0074	0x5800 4074	0x4804 4074
DSI_COMPLEXIO_CFG2	RW	32	0x0000 0078	0x5800 4078	0x4804 4078
DSI_RX_FIFO_VC_FULLNESS	R	32	0x0000 007C	0x5800 407C	0x4804 407C
DSI_VM_TIMING4	RW	32	0x0000 0080	0x5800 4080	0x4804 4080
DSI_TX_FIFO_VC_EMPTYNESS	R	32	0x0000 0084	0x5800 4084	0x4804 4084
DSI_VM_TIMING5	RW	32	0x0000 0088	0x5800 4088	0x4804 4088
DSI_VM_TIMING6	RW	32	0x0000 008C	0x5800 408C	0x4804 408C
DSI_VM_TIMING7	RW	32	0x0000 0090	0x5800 4090	0x4804 4090
DSI_STOPCLK_TIMING	RW	32	0x0000 0094	0x5800 4094	0x4804 4094
DSI_CTRL2	RW	32	0x0000 0098	0x5800 4098	0x4804 4098
DSI_VM_TIMING8	RW	32	0x0000 009C	0x5800 409C	0x4804 409C
DSI_TE_HSYNC_WIDTH_j <sup>(1)</sup>	RW	32	0x0000 00A0 + (0xc * j)	0x5800 40A0 + (0xc * j)	0x4804 40A0 + (0xc * j)
DSI_TE_VSYNC_WIDTH_j <sup>(1)</sup>	RW	32	0x0000 00A4 + (0xc * j)	0x5800 40A4 + (0xc * j)	0x4804 40A4 + (0xc * j)
DSI_TE_HSYNC_NUMBER_j <sup>(1)</sup>	RW	32	0x0000 00A8 + (0xc * j)	0x5800 40A8 + (0xc * j)	0x4804 40A8 + (0xc * j)
DSI_VC_CTRL_j <sup>(2)</sup>	RW	32	0x0000 0100 + (0x20 * i)	0x5800 4100 + (0x20 * i)	0x4804 4100 + (0x20 * i)
DSI_VC_TE_j <sup>(2)</sup>	RW	32	0x0000 0104 + (0x20 * i)	0x5800 4104 + (0x20 * i)	0x4804 4104 + (0x20 * i)
DSI_VC_LONG_PACKET_HEADER_j <sup>(2)</sup>	W	32	0x0000 0108 + (0x20 * i)	0x5800 4108 + (0x20 * i)	0x4804 4108 + (0x20 * i)
DSI_VC_LONG_PACKET_PAYLOAD_j <sup>(2)</sup>	W	32	0x0000 010C + (0x20 * i)	0x5800 410C + (0x20 * i)	0x4804 410C + (0x20 * i)
DSI_VC_SHORT_PACKET_HEADER_j <sup>(2)</sup>	RW	32	0x0000 0110 + (0x20 * i)	0x5800 4110 + (0x20 * i)	0x4804 4110 + (0x20 * i)
DSI_VC_IRQSTATUS_j <sup>(2)</sup>	RW	32	0x0000 0118 + (0x20 * i)	0x5800 4118 + (0x20 * i)	0x4804 4118 + (0x20 * i)
DSI_VC_IRQENABLE_j <sup>(2)</sup>	RW	32	0x0000 011C + (0x20 * i)	0x5800 411C + (0x20 * i)	0x4804 411C + (0x20 * i)

<sup>(1)</sup> j = 0 to 1<sup>(2)</sup> i = 0 to 3**Table 10-515. DSI2\_PROTOCOL\_ENGINE Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSI2_PROTOCOL_ENGINE L3 Physical Address	DSI2_PROTOCOL_ENGINE L4_PER Physical Address
DSI_REVISION	R	32	0x0000 0000	0x5800 5000	0x4804 5000
DSI_SYSCONFIG	RW	32	0x0000 0010	0x5800 5010	0x4804 5010
DSI_SYSSTATUS	R	32	0x0000 0014	0x5800 5014	0x4804 5014

**Table 10-515. DSI2\_PROTOCOL\_ENGINE Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DSI2_PROTOCOL_ENGINE L3 Physical Address	DSI2_PROTOCOL_ENGINE L4_PER Physical Address
DSI_IRQSTATUS	RW	32	0x0000 0018	0x5800 5018	0x4804 5018
DSI_IRQENABLE	RW	32	0x0000 001C	0x5800 501C	0x4804 501C
DSI_CTRL	RW	32	0x0000 0040	0x5800 5040	0x4804 5040
DSI_GNQ	R	32	0x0000 0044	0x5800 5044	0x4804 5044
DSI_COMPLEXIO_CFG1	RW	32	0x0000 0048	0x5800 5048	0x4804 5048
DSI_COMPLEXIO_IRQSTATUS	RW	32	0x0000 004C	0x5800 504C	0x4804 504C
DSI_COMPLEXIO_IRQENABLE	RW	32	0x0000 0050	0x5800 5050	0x4804 5050
DSI_CLK_CTRL	RW	32	0x0000 0054	0x5800 5054	0x4804 5054
DSI_TIMING1	RW	32	0x0000 0058	0x5800 5058	0x4804 5058
DSI_TIMING2	RW	32	0x0000 005C	0x5800 505C	0x4804 505C
DSI_VM_TIMING1	RW	32	0x0000 0060	0x5800 5060	0x4804 5060
DSI_VM_TIMING2	RW	32	0x0000 0064	0x5800 5064	0x4804 5064
DSI_VM_TIMING3	RW	32	0x0000 0068	0x5800 5068	0x4804 5068
DSI_CLK_TIMING	RW	32	0x0000 006C	0x5800 506C	0x4804 506C
DSI_TX_FIFO_VC_SIZE	RW	32	0x0000 0070	0x5800 5070	0x4804 5070
DSI_RX_FIFO_VC_SIZE	RW	32	0x0000 0074	0x5800 5074	0x4804 5074
DSI_COMPLEXIO_CFG2	RW	32	0x0000 0078	0x5800 5078	0x4804 5078
DSI_RX_FIFO_VC_FULLNESS	R	32	0x0000 007C	0x5800 507C	0x4804 507C
DSI_VM_TIMING4	RW	32	0x0000 0080	0x5800 5080	0x4804 5080
DSI_TX_FIFO_VC_EMPTYNESS	R	32	0x0000 0084	0x5800 5084	0x4804 5084
DSI_VM_TIMING5	RW	32	0x0000 0088	0x5800 5088	0x4804 5088
DSI_VM_TIMING6	RW	32	0x0000 008C	0x5800 508C	0x4804 508C
DSI_VM_TIMING7	RW	32	0x0000 0090	0x5800 5090	0x4804 5090
DSI_STOPCLK_TIMING	RW	32	0x0000 0094	0x5800 5094	0x4804 5094
DSI_CTRL2	RW	32	0x0000 0098	0x5800 5098	0x4804 5098
DSI_VM_TIMING8	RW	32	0x0000 009C	0x5800 509C	0x4804 509C
DSI_TE_HSYNC_WIDTH <sub>j</sub> <sup>(1)</sup>	RW	32	0x0000 00A0 + (0xc * j)	0x5800 50A0 + (0xc * j)	0x4804 50A0 + (0xc * j)
DSI_TE_VSYNC_WIDTH <sub>j</sub> <sup>(1)</sup>	RW	32	0x0000 00A4 + (0xc * j)	0x5800 50A4 + (0xc * j)	0x4804 50A4 + (0xc * j)
DSI_TE_HSYNC_NUMBER <sub>j</sub> <sup>(1)</sup>	RW	32	0x0000 00A8 + (0xc * j)	0x5800 50A8 + (0xc * j)	0x4804 50A8 + (0xc * j)
DSI_VC_CTRL <sub>i</sub> <sup>(2)</sup>	RW	32	0x0000 0100 + (0x20 * i)	0x5800 5100 + (0x20 * i)	0x4804 5100 + (0x20 * i)
DSI_VC_TE <sub>i</sub> <sup>(2)</sup>	RW	32	0x0000 0104 + (0x20 * i)	0x5800 5104 + (0x20 * i)	0x4804 5104 + (0x20 * i)
DSI_VC_LONG_PACKET_HEADER <sub>i</sub> <sup>(2)</sup>	W	32	0x0000 0108 + (0x20 * i)	0x5800 5108 + (0x20 * i)	0x4804 5108 + (0x20 * i)

<sup>(1)</sup> j = 0 to 1

<sup>(2)</sup> i = 0 to 3

**Table 10-515. DSI2\_PROTOCOL\_ENGINE Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DSI2_PROTOCOL_ENGINE L3 Physical Address	DSI2_PROTOCOL_ENGINE L4_PER Physical Address
DSI_VC_LONG_PACKET_PAYLOAD_i <sup>(2)</sup>	W	32	0x0000 010C + (0x20 * i)	0x5800 510C + (0x20 * i)	0x4804 510C + (0x20 * i)
DSI_VC_SHORT_PACKET_HEADER_i <sup>(2)</sup>	RW	32	0x0000 0110 + (0x20 * i)	0x5800 5110 + (0x20 * i)	0x4804 5110 + (0x20 * i)
DSI_VC_IRQSTAT_US_i <sup>(2)</sup>	RW	32	0x0000 0118 + (0x20 * i)	0x5800 5118 + (0x20 * i)	0x4804 5118 + (0x20 * i)
DSI_VC_IRQENAB_LE_i <sup>(2)</sup>	RW	32	0x0000 011C + (0x20 * i)	0x5800 511C + (0x20 * i)	0x4804 511C + (0x20 * i)

**10.3.6.2.2 DSI\_PROTOCOL\_ENGINE Register Description****Table 10-516. DSI\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	IP Revision		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI Internal data**Table 10-517. Register Call Summary for Register DSI\_REVISION**

MIPI Display Serial Interface

- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[0\] \[1\]](#)

**Table 10-518. DSI\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	SYSTEM CONFIGURATION REGISTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLOCKACTIVITY	RESERVED	SIDLEMODE	ENWAKEUP	SOFT_RESET	AUTO_IDLE										



Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9:8	CLOCKACTIVITY	Clocks activity during wake up mode period 0x0: Interface and Functional clocks can be switched off 0x1: Functional clocks can be switched off and Interface clocks are maintained during wake up period 0x2: Interface clocks can be switched off and Functional clocks are maintained during wake up period 0x3: Interface and Functional clocks are maintained during wake up period	RW	0x0
7:5	RESERVED		R	0x0
4:3	SIDLEMODE	Slave interface power management, Idle req/ack control 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module. 0x3: Reserved	RW	0x2
2	ENWAKEUP	Wake-up mode enable bit 0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0
1	SOFT_RESET	Software reset. Set the bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads return 0. 0x0: Normal mode. 0x1: The module is reset	RW	0
0	AUTO_IDLE	Internal interface gating strategy 0x0: Interface clock is free-running. 0x1: Automatic Interface clock gating strategy is applied based on the interface activity.	RW	1

**Table 10-519. Register Call Summary for Register DSI\_SYSCONFIG**

- MIPI Display Serial Interface
- [Software Reset: \[0\]](#)
  - [Power Management: \[1\] \[2\]](#)
  - [Software Reset: \[3\]](#)
  - [DSI Global Initialization: \[4\] \[5\] \[6\]](#)
  - [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[7\] \[8\]](#)

**Table 10-520. DSI\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014																																																													
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b> DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE																																																												
<b>Description</b>	SYSTEM STATUS REGISTER This register provides status information about the module, excluding the interrupt status register.																																																													
<b>Type</b>	R																																																													
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="11"></td> <td>RESET_DONE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																											RESET_DONE
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
RESERVED																											RESET_DONE																																			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	RESET_DONE	Internal reset monitoring Read 0x0: Internal module reset is on going. Read 0x1: Reset completed.	R	1

**Table 10-521. Register Call Summary for Register DSI\_SYSSTATUS**

MIPI Display Serial Interface

- [Software Reset: \[0\] \[1\]](#)
- [DSI Global Initialization: \[2\] \[3\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[4\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[5\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[6\] \[7\]](#)

**Table 10-522. DSI\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>		
<b>Description</b>	<p>INTERRUPT STATUS REGISTER - All virtual channels + Complex I/O + PLL            This register associates one bit for each virtual channel in order to determine which virtual channel has generated the interrupt. The virtual channel shall be enabled for events to be generated on that virtual channel.            If the virtual channel is disabled, the interrupt is not generated.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED								TE1_LINE_IRQ	TE0_LINE_IRQ	TA_TO_IRQ	LDO_POWER_GOOD_IRQ	SYNC_LOST_IRQ	ACK_TRIGGER_IRQ	TE_TRIGGER_IRQ	LP_RX_TO_IRQ	HS_TX_TO_IRQ	RESERVED	COMPLEXIO_ERR_IRQ	PLL_RECAL_IRQ	PLL_UNLOCK_IRQ	PLL_LOCK_IRQ	RESERVED	RESYNCHRONIZATION_IRQ	WAKEUP_IRQ	VIRTUAL_CHANNEL3_IRQ	VIRTUAL_CHANNEL2_IRQ	VIRTUAL_CHANNEL1_IRQ	VIRTUAL_CHANNEL0_IRQ					

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x000
22	TE1_LINE_IRQ	The VSYNC and corresponding HSYNC pulses defined in DSI_TE_HSYNC_NUMBER for the line TE1 have been received by the DSI protocol engine and have trigger the start of the data transfer to the peripheral.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
21	TE0_LINE_IRQ	The VSYNC and corresponding HSYNC pulses defined in DSI_TE_HSYNC_NUMBER for the line TE0 have been received by the DSI protocol engine and have trigger the start of the data transfer to the peripheral.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0

Bits	Field Name	Description	Type	Reset
20	TA_TO_IRQ	Turn-around Time out. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
19	LDO_POWER_GOOD_IRQ	Transition of the status signal LDOPWRGOOD from the DSI_PHY indicating a state change for the supply VDDALDODSIPLL from up to down or down to up. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
18	SYNC_LOST_IRQ	Synchronization with Video port is lost (Video mode only) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
17	ACK_TRIGGER_IRQ	Acknowledge Trigger 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
16	TE_TRIGGER_IRQ	Tearing Effect Trigger 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
15	LP_RX_TO_IRQ	Interrupt for Low Power Rx Time out 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
14	HS_TX_TO_IRQ	Interrupt for High Speed Tx Time out. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
13:11	RESERVED		R	0x0
10	COMPLEXIO_ERR_IRQ	Error signaling from Complex I/O: status of the complex I/O errors received from the complex I/O (events are defined in <a href="#">DSI_COMPLEXIO_IRQSTATUS</a> ). Read 0x0: READS: Event is false. Read 0x1: READS: Event is true (pending).	R	0
9	PLL_RECAL_IRQ	PLL recal event (assertion of DSIRecal signal from the DSI PLL Control module) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
8	PLL_UNLOCK_IRQ	PLL un-clock event (de-assertion of DSILock signal from the DSI PLL Control module) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0

Bits	Field Name	Description	Type	Reset
7	PLL_LOCK_IRQ	PLL clock event (assertion of DSILock signal from the DSI PLL Control module) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
6	RESERVED		R	0
5	RESYNCHRONIZATION_IRQ	Video mode resynchronization indicates to the software users that the video port works but the configuration of the timings for the display controller (DISPC) and for DSI Protocol engine may need to be modified to avoid the resynchronization to occur. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
4	WAKEUP_IRQ	Wakeup 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
3	VIRTUAL_CHANNEL3_IRQ	Virtual channel 3 Read 0x0: READS: Event is false. Read 0x1: READS: Event is true (pending).	R	0
2	VIRTUAL_CHANNEL2_IRQ	Virtual channel 2 Read 0x0: READS: Event is false. Read 0x1: READS: Event is true (pending).	R	0
1	VIRTUAL_CHANNEL1_IRQ	Virtual channel 1 Read 0x0: READS: Event is false. Read 0x1: READS: Event is true (pending).	R	0
0	VIRTUAL_CHANNEL0_IRQ	Virtual channel 0 Read 0x0: READS: Event is false. Read 0x1: READS: Event is true (pending).	R	0

**Table 10-523. Register Call Summary for Register DSI\_IRQSTATUS**

## MIPI Display Serial Interface

- [DSI Interrupt Requests: \[0\]](#)
- [DSI Transfer Modes: \[1\] \[2\]](#)
- [Timers: \[3\] \[4\]](#)
- [PHY Triggers: \[5\] \[6\]](#)
- [Interrupts: \[7\]](#)
- [Error Handling: \[8\] \[9\] \[10\]](#)
- [Error Handling: \[11\]](#)
- [DSI Global Initialization: \[12\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[13\] \[14\]](#)

**Table 10-524. DSI\_IRQENABLE**

<b>Address Offset</b>	0x0000 001C		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	INTERRUPT ENABLE REGISTER - This register associates one bit for each virtual channel in order to enable/disable each virtual channel individually.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TE1_LINE_IRQ_EN	TE0_LINE_IRQ_EN	TA_TO_IRQ_EN	LDO_POWER_GOOD_IRQ_EN	SYNC_LOST_IRQ_EN	ACK_TRIGGER_IRQ_EN	TE_TRIGGER_IRQ_EN	LP_RX_TO_IRQ_EN	HS_TX_TO_IRQ_EN	RESERVED						PLL_RECAL_IRQ_EN	PLL_UNLOCK_IRQ_EN	PLL_LOCK_IRQ_EN	RESERVED	RESYNCHRONIZATION_IRQ_EN	WAKEUP_IRQ_EN	RESERVED		

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x000
22	TE1_LINE_IRQ_EN	The VSYNC and corresponding HSYNC pulses defined in DSI_TE_HSYNC_NUMBER for the line TE1 have been received by the DSI protocol engine and have trigger the start of the data transfer to the peripheral. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
21	TE0_LINE_IRQ_EN	The VSYNC and corresponding HSYNC pulses defined in DSI_TE_HSYNC_NUMBER for the line TE0 have been received by the DSI protocol engine and have trigger the start of the data transfer to the peripheral. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
20	TA_TO_IRQ_EN	Turn-around Time out. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
19	LDO_POWER_GOOD_IRQ_EN	Transition of the status signal LDOPWRGOOD from the DSI_PHY indicating a state change for the supply VDDALDODSIPLL from up to down or down to up. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
18	SYNC_LOST_IRQ_EN	Synchronization with Video port is lost (Video mode only) 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
17	ACK_TRIGGER_IRQ_EN	Acknowledge trigger 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
16	TE_TRIGGER_IRQ_EN	Tearing Effect trigger 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
15	LP_RX_TO_IRQ_EN	Interrupt for Low Power Rx Time out. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
14	HS_TX_TO_IRQ_EN	Interrupt for High Speed Tx Time out. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
13:10	RESERVED		R	0x0
9	PLL_RECAL_IRQ_EN	PLL recal event (assertion of DSIRecal signal from the DSI PLL Control module) 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
8	PLL_UNLOCK_IRQ_EN	PLL un-clock event (de-assertion of DSILock signal from the DSI PLL Control module) 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	PLL_LOCK_IRQ_EN	PLL clock event (assertion of DSILock signal from the DSI PLL Control module) 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
6	RESERVED		R	0
5	RESYNCHRONIZATION_IRQ_EN	Video mode resynchronization 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	WAKEUP_IRQ_EN	Wakeup 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
3:0	RESERVED		R	0x0

**Table 10-525. Register Call Summary for Register DSI\_IRQENABLE**

MIPI Display Serial Interface

- [PHY Triggers: \[0\] \[1\]](#)
- [Interrupts: \[2\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[3\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[4\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[5\] \[6\]](#)

**Table 10-526. DSI\_CTRL**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	GLOBAL CONTROL REGISTER This register controls the DSI Protocol Engine module. This register shall not be modified dynamically (except IF_EN bit fields).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DISPC_UPDATE_SYNC	HSA_BLANKING_MODE	HBP_BLANKING_MODE	HFP_BLANKING_MODE	BLANKING_MODE	EOT_ENABLE	VP_HSYNC_END	VP_HSYNC_START	VP_VSYNC_END	VP_VSYNC_START	TRIGGER_RESET_MODE	LINE_BUFFER	VP_VSYNC_POL	VP_HSYNC_POL	VP_DE_POL	VP_CLK_POL	VP_DATA_BUS_WIDTH	TRIGGER_RESET	VP_CLK_RATIO	TX_FIFO_ARBITRATION	ECC_RX_EN	CS_RX_EN	IF_EN	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	DISPC_UPDATE_SYNC	Determines if the Dispc_Update_Sync signal from the display controller is used.  0x0: Dispc_Update_Sync signal is not used. 0x1: Dispc_Update_Sync signal is used.	RW	0
23	HSA_BLANKING_MODE	Blanking mode  0x0: Packets in TX FIFO are sent during HSA blanking period of video mode or LPS is used. 0x1: LONG BLANKING PACKETS only are used during HSA blanking period of video mode.	RW	0
22	HBP_BLANKING_MODE	Blanking mode  0x0: Packets in TX FIFO are sent during HBP blanking period of video mode or LPS is used. 0x1: LONG BLANKING PACKETS only are used during HBP blanking period of video mode.	RW	0
21	HFP_BLANKING_MODE	Blanking mode  0x0: Packets in TX FIFO are sent during HFP blanking period of video mode or LPS is used. 0x1: LONG BLANKING PACKETS only are used during HFP blanking period of video mode.	RW	0
20	BLANKING_MODE	Blanking mode  0x0: LPS is used during blanking periods of video mode (except HSA, HBP, HFP defined in HSA_BLANKING_MODE, HBP_BLANKING_MODE AND HFP_BLANKING_MODE respectively) when there is no command mode data in TX FIFO ready to be sent. So blanking periods can be different during the frame depending on the TX FIFO. 0x1: LONG BLANKING PACKETS are used during blanking periods of video mode (except HSA, HBP, HFP defined in HSA_BLANKING_MODE, HBP_BLANKING_MODE AND HFP_BLANKING_MODE respectively) regardless of the packets present in the TX FIFO ready to be sent	RW	0
19	EOT_ENABLE	Enable EOT packets at the end of HS transmission.  0x0: No EOT packets 0x1: EOT packet is sent at all HS to LP transitions.	RW	0
18	VP_HSYNC_END	HSYNC end pulse.  0x0: Disabled. No HSYNC END short packet is generated. 0x1: Enabled. While the HSYNC END pulse is detected, the associated short packet HSYNC END is generated.	RW	0

Bits	Field Name	Description	Type	Reset
17	VP_HSYNC_START	HSYNC start pulse.  0x0: Disabled. No HSYNC START short packet is generated.  0x1: Enabled. While the HSYNC start pulse is detected, the associated short packet HSYNC START is generated.	RW	0
16	VP_VSYNC_END	VSYNC end pulse.  0x0: Disabled. No VSYNC END short packet is generated.  0x1: Enabled. While the VSYNC END pulse is detected, the associated short packet VSYNC END is generated.	RW	0
15	VP_VSYNC_START	VSYNC start pulse.  0x0: Disabled. No VSYNC START short packet is generated.  0x1: Enabled. While the VSYNC START pulse is detected, the associated short packet VSYNC START is generated.	RW	0
14	TRIGGER_RESET_MODE	Selection of the trigger reset mode  0x0: Synchronized: the mode is only valid if there is virtual channel using the video mode and it is active. The principal is to wait for the current video frame to be transferred on the link. Any data received after the VSYNC are ignored.  0x1: Immediate: all pending requests in TX FIFO are taken into account for transfer scheduling, the RX FIFO is ignored, and the data from video port are ignored as soon as possible. Only the current transfer on DSI link and already scheduled ones are transmitted. All the other transfers are discarded.	RW	0
13:12	LINE_BUFFER	Number of line buffers to be used while receiving data on the video port. The valid values are from 0 to <a href="#">DSI_GNQ[17:16]</a> VP1_NB_LINE_BUFFER.  0x0: No line buffer 0x1: 1 line buffer 0x2: 2 line buffers	RW	0x0
11	VP_VSYNC_POL	VP vertical synchronization signal polarity  0x0: VSYNC signal on the video port is active low. 0x1: VSYNC signal on the video port is active high.	RW	0
10	VP_HSYNC_POL	VP horizontal synchronization signal polarity  0x0: HSYNC signal on the video port is active low. 0x1: HSYNC signal on the video port is active high.	RW	0
9	VP_DE_POL	VP data enable signal polarity  0x0: DE signal on the video port is active low. 0x1: DE signal on the video port is active high.	RW	0
8	VP_CLK_POL	VP pixel clock polarity  0x0: The DSI Protocol Engine module captures the data on the VP on the pixel clock falling edge. The module connected to the VP shall drive the data on the pixel clock rising edge.  0x1: The DSI Protocol Engine module captures the data on the VP on the pixel clock raising edge. The module connected to the VP shall drive the data on the pixel clock falling edge.	RW	1



Bits	Field Name	Description	Type	Reset
7:6	VP_DATA_BUS_WIDTH	<p>Defines the size of the video port data bus</p> <p>0x0: 16-bits data width (LSB of the 24-bit video port data bus)</p> <p>0x1: 18-bits data width (LSB of the 24-bit video port data bus)</p> <p>0x2: 24-bits data width (LSB of the 24-bit video port data bus)</p>	RW	0x0
5	TRIGGER_RESET	<p>Send the reset trigger to the peripheral.</p> <p>0x0: READS: Reset trigger generation is completed. It is reset by hardware when it is completed. WRITES: Cancellation of the request for Reset trigger generation (maybe too late since it is already on going)</p> <p>0x1: READS: Generation of the reset trigger has been requested by user (could be on going but not completed yet). WRITES: Request for Reset trigger to be sent to the peripheral.</p>	RW	0
4	VP_CLK_RATIO	<p>The field indicates the clock ratio between VP_CLK and VP_PCLK. The clock VP_PCLK is generated from VP_CLK. It is divided down. The information is only used when the video port is used to provide data in command mode. In the case of video mode, it is not used.</p> <p>0x0: The clock VP_PCLK is the clock VP_CLK divided by 2. The duty cycle of VP_PCLK is 50/50.</p> <p>0x1: The clock VP_PCLK is the clock VP_CLK divided by 3 or more. The duty cycle of VP_PCLK is not 50/50 for odd ratio numbers (3,5,7,...).</p>	RW	0
3	TX_FIFO_ARBITRATION	<p>Defines the arbitration scheme for granting the virtual channel pending ready requests in the TX FIFO</p> <p>0x0: Round-Robin Scheme is used</p> <p>0x1: Sequential Scheme is used</p>	RW	0
2	ECC_RX_EN	<p>Enables the Error Correction Code check for the received header (short and long packets for all virtual channel ids).</p> <p>0x0: Disabled</p> <p>0x1: Enabled</p>	RW	0
1	CS_RX_EN	<p>Enables the checksum check for the received payload (long packet only for all virtual channel ids).</p> <p>0x0: Disabled</p> <p>0x1: Enabled</p>	RW	0
0	IF_EN	<p>Enables the module. When the module is disabled the signals from the complex I/O are gated (no updates of the interrupt status register). It is not possible to change the bit fields in the register <a href="#">DSI_CTRL</a> except IF_EN when it is enabled. All the other registers can be changed except the ones that require <a href="#">DSI_VC_CTRL_i[0]</a> VC_EN to be equal to 0 to be modified.</p> <p>0x0: The interface is disabled. If one of the virtual channel uses the video mode with the video port to receive the data, the DSI protocol engines is disabled when the next VSYNC is received and all the data in the FIFO for the other virtual channels in command mode are sent to the peripherals (if BTA_EN is enabled, the DSI protocol needs to wait for the response and BTA from the peripheral before disabling all the internal logic since an acknowledge is requested).</p> <p>0x1: The interface is enabled immediately, the data acquisition on the video port starts on the next VSYNC (video mode) or first data received in the Slave port FIFO (command mode).</p>	RW	0

**Table 10-527. Register Call Summary for Register DSI\_CTRL**

## MIPI Display Serial Interface

- [Short Packet: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [DSI Global Register Controls: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [DSI Transfer Modes: \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [Clock Requirements: \[19\]](#)
- [Video Port Interface: \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\]](#)
- [Timers: \[29\] \[30\] \[31\] \[32\] \[33\] \[34\]](#)
- [Bus Turnaround: \[35\] \[36\] \[37\] \[38\]](#)
- [PHY Triggers: \[39\] \[40\] \[41\] \[42\]](#)
- [ECC Generation: \[43\]](#)
- [Checksum Generation for Long Packet Payloads: \[44\]](#)
- [EOT Packet: \[45\]](#)
- [Power Control of DSI PHY Complex I/O and DSI PLL: \[46\] \[47\] \[48\] \[49\] \[50\] \[51\]](#)
- [Pad Configuration: \[52\] \[53\] \[54\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[55\] \[56\] \[57\] \[58\] \[59\] \[60\] \[61\] \[62\] \[63\] \[64\] \[65\] \[66\] \[67\] \[68\] \[69\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[70\] \[71\] \[72\] \[73\] \[74\] \[75\] \[76\] \[77\] \[78\] \[79\] \[80\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[81\] \[82\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Description: \[83\] \[84\] \[85\] \[86\] \[87\] \[88\] \[89\] \[90\]](#)

**Table 10-528. DSI\_GNQ**

<b>Address Offset</b>	0x0000 0044	
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b> DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	GENERIC PARAMETER REGISTER This register provide a way to read the generic parameters used in the design.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								NB_VIDEO_PORTS	VP2_NB_LINE_BUFFER	RESERVED	VP2_LINE_BUFFER_SIZE	VP1_NB_LINE_BUFFER	RESERVED	VP1_LINE_BUFFER_SIZE	NB_DATA_LANES	NB_DMA_REQUEST	RX_FIFODEPTH	TX_FIFODEPTH													

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	NB_VIDEO_PORTS	Number of video ports Read 0x0: Video port 1 only is present Read 0x1: Video port 1 and video port 2 are present	R	0
23:22	VP2_NB_LINE_BUFFER	Determines the number of video buffer lines associated to video port 2. Read 0x0: No line buffer Read 0x1: 1 line buffer of the size defined in LINE_BUFFER_SIZE Read 0x2: 2 line buffers of the size defined in LINE_BUFFER_SIZE	R	0x2 for DSI1 0x0 for DSI2
21	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
20:18	VP2_LINE_BUFFER_SIZE	<p>Determines the video line buffer size associated to video port 2 .</p> <p>Read 0x1: 512x24-bits, 682x18-bits, 768x16bits (memory of 384x32-bits)</p> <p>Read 0x2: 682x24-bits, 910x18-bits, 1024x16bits (memory of 512x32-bits)</p> <p>Read 0x3: 853x24-bits, 1137x18-bits, 1280x16bits (memory of 640x32-bits)</p> <p>Read 0x4: 1024x24-bits, 1365x18-bits, 1536x16bits (memory of 768x32-bits)</p> <p>Read 0x5: 1194x24-bits, 1592x18-bits, 1792x16bits (memory of 896x32-bits)</p> <p>Read 0x6: 1365x24-bits, 1820x18-bits, 2048x16bits (memory of 1024x32-bits)</p>	R	0x4 for DSI1 0x1 for DSI2
17:16	VP1_NB_LINE_BUFFER	<p>Determines the number of video buffer lines associated to video port 1.</p> <p>Read 0x0: No line buffer</p> <p>Read 0x1: 1 line buffer of the size defined in LINE_BUFFER_SIZE</p> <p>Read 0x2: 2 line buffers of the size defined in LINE_BUFFER_SIZE</p>	R	0x2
15	RESERVED		R	0
14:12	VP1_LINE_BUFFER_SIZE	<p>Determines the video line buffer size associated to video port 1 .</p> <p>Read 0x1: 512x24-bits, 682x18-bits, 768x16bits (memory of 384x32-bits)</p> <p>Read 0x2: 682x24-bits, 910x18-bits, 1024x16bits (memory of 512x32-bits)</p> <p>Read 0x3: 853x24-bits, 1137x18-bits, 1280x16bits (memory of 640x32-bits)</p> <p>Read 0x4: 1024x24-bits, 1365x18-bits, 1536x16bits (memory of 768x32-bits)</p> <p>Read 0x5: 1194x24-bits, 1592x18-bits, 1792x16bits (memory of 896x32-bits)</p> <p>Read 0x6: 1365x24-bits, 1820x18-bits, 2048x16bits (memory of 1024x32-bits)</p>	R	0x6 for DSI1 0x4 for DSI2
11:9	NB_DATA_LANES	<p>Determines the number of data lanes supported by the DSI protocol engine .</p> <p>Read 0x1: 1 Data lane</p> <p>Read 0x2: 2 Data lanes</p> <p>Read 0x3: 3 Data lanes</p> <p>Read 0x4: 4 Data lanes</p>	R	0x4 for DSI1 0x2 for DSI2
8:6	NB_DMA_REQUEST	<p>Determines the number of DMA_REQ signals.</p> <p>Read 0x0: No DMA request</p> <p>Read 0x1: 1 DMA request</p> <p>Read 0x2: 2 DMA requests</p> <p>Read 0x3: 3 DMA requests</p> <p>Read 0x4: 4 DMA requests</p>	R	0x4
5:3	RX_FIFODEPTH	<p>Determines the data RX FIFO depth (32-bit words) on the slave port.</p> <p>Read 0x4: 32x 33 bits</p> <p>Read 0x5: 64x 33 bits</p> <p>Read 0x6: 128 x 33 bits</p> <p>Read 0x7: 256 x 33 bits</p>	R	0x6

Bits	Field Name	Description	Type	Reset
2:0	TX_FIFODEPTH	Determines the data TX FIFO depth (33-bit words) on the slave port. Read 0x4: 32x 33 bits Read 0x5: 64x 33 bits Read 0x6: 128 x 33 bits Read 0x7: 256 x 33 bits	R	0x6

**Table 10-529. Register Call Summary for Register DSI\_GNQ**

MIPI Display Serial Interface

- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[0\] \[1\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Description: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\]](#)

**Table 10-530. DSI\_COMPLEXIO\_CFG1**

<b>Address Offset</b>	0x0000 0048		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	COMPLEXIO CONFIGURATION REGISTER for the complex I/O This register contains the lane configuration for the order and position of the lanes (clock and data) and the polarity order for the control of the PHY differential signals in addition to the control bit for the power FSM.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SHADOWING	GOBIT	RESET_DONE	PWR_CMD	PWR_STATUS	RESERVED	RESERVED	RESERVED	LDO_POWER_GOOD_STATE	USE_LDO_EXTERNAL	DATA4_POL	DATA4_POSITION	DATA3_POL	DATA3_POSITION	DATA2_POL	DATA2_POSITION	DATA1_POL	DATA1_POSITION	CLOCK_POL	CLOCK_POSITION												

Bits	Field Name	Description	Type	Reset
31	SHADOWING	Shadowing configuration. 0x0: Disabled. The writes to the <a href="#">DSI_PHY_REGISTER0</a> through <a href="#">DSI_PHY_REGISTER2</a> registers is done like the other SCP registers. 0x1: Enabled. The writes to the <a href="#">DSI_PHY_REGISTER0</a> through <a href="#">DSI_PHY_REGISTER2</a> registers is done only when the GOBIT is set and when the signal DISPCUpdataSync from the display controller module is active.	RW	0

Bits	Field Name	Description	Type	Reset
30	GOBIT	<p>Allows the synchronized update of the shadow registers when the signal DISPCUpdateSync is active.</p> <p>0x0: Resets the GOBIT. The hardware has finished the update of the shadow SCP registers. The bit is reset by Hardware. The software can reset the bit in case the user decides to abort it. There is no guarantee that the software reset is done before the transfer of the values to the complex IO.</p> <p>0x1: Set the GOBIT. Only when the transfer of the new values for the three first registers is completed (3, 2, 1, or 0 transfers are performed based on the number of registers to update), the GOBIT is reset. The DISPCUpdateSync signal is used to synchronize the update. The bit shall be set only when it is in reset state.</p>	RW	0
29	RESET_DONE	<p>Internal reset monitoring of the power domain using the TXBYTECLKHS clock from the complex I/O</p> <p>Read 0x0: Internal module reset is on going.</p> <p>Read 0x1: Reset completed.</p>	R	1
28:27	PWR_CMD	<p>Command for power control of the complex I/O</p> <p>0x0: Command to change to OFF state</p> <p>0x1: Command to change to ON state</p> <p>0x2: Command to change to ULTRALOW-POWER state</p>	RW	0x0
26:25	PWR_STATUS	<p>Status of the power control of the complex I/O</p> <p>Read 0x0: Complex I/O in OFF state</p> <p>Read 0x1: Complex I/O in ON state</p> <p>Read 0x2: Complex I/O in ULTRALOW-POWER state</p>	R	0x0
24:22	RESERVED		R	0x0
21	LDO_POWER_GOOD_STATE	<p>Indicates the state of the signal LDOPWRGOOD. VDDALDODSIPLL: 1.2V power supply for the PLL. The voltage is supplied by the internal or external LDO. The interrupt LDO_POWER_GOOD_IRQ is generated when a transition is detected on the signal LDOPWRGOOD from the DSI_PHY.</p> <p>Read 0x0: VDDALDODSIPLL power supply is down</p> <p>Read 0x1: VDDALDODSIPLL power supply is up</p>	R	0
20	USE_LDO_EXTERNAL	<p>Select the external LDO for the DSI_PHY.</p> <p>0x0: DSI_PHY internal LDO is used.</p> <p>0x1: External LDO is used. DSI_PHY LDO is tri-stated.</p>	RW	0
19	DATA4_POL	<p>+/- differential pin order of DATA lane 4. Applicable only to DS1. For DS12 this bit field is RESERVED.</p> <p>0x0: +/- pin order (DSi.DX=+ and DSi.DY=-)</p> <p>0x1: -/+ pin order (DSi.DX=- and DSi.DY=+)</p>	RW	0
18:16	DATA4_POSITION	<p>Position and order of the DATA lane 4. Applicable only to DS1. For DS12 this bit field is RESERVED.</p> <p>0x0: Not used/connected</p> <p>0x1: Data lane 4 is at the position 1 (line 1).</p> <p>0x2: Data lane 4 is at the position 2 (line 2).</p> <p>0x3: Data lane 4 is at the position 3 (line 3).</p> <p>0x4: Data lane 4 is at the position 4 (line 4).</p> <p>0x5: Data lane 4 is at the position 5 (line 5).</p>	RW	0x0

Bits	Field Name	Description	Type	Reset
15	DATA3_POL	+/- differential pin order of DATA lane 3. Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: +/- pin order (DS1.DX=+ and DS1.DY=-) 0x1: -/+ pin order (DS1.DX=- and DS1.DY=+)	RW	0
14:12	DATA3_POSITION	Position and order of the DATA lane 3. Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: Not used/connected 0x1: Data lane 3 is at the position 1 (line 1). 0x2: Data lane 3 is at the position 2 (line 2). 0x3: Data lane 3 is at the position 3 (line 3). 0x4: Data lane 3 is at the position 4 (line 4). 0x5: Data lane 3 is at the position 5 (line 5).	RW	0x0
11	DATA2_POL	+/- differential pin order of DATA lane 2.  0x0: +/- pin order (DS1.DX=+ and DS1.DY=-) 0x1: -/+ pin order (DS1.DX=- and DS1.DY=+)	RW	0
10:8	DATA2_POSITION	Position and order of the DATA lane 2.  0x0: Not used/connected 0x1: Data lane 2 is at the position 1 (line 1). 0x2: Data lane 2 is at the position 2 (line 2). 0x3: Data lane 2 is at the position 3 (line 3). 0x4: Data lane 2 is at the position 4 (line 4). 0x5: Data lane 2 is at the position 5 (line 5).	RW	0x0
7	DATA1_POL	+/- pin differential pin order of DATA lane 1  0x0: +/- pin order (DS1.DX=+ and DS1.DY=-) 0x1: -/+ pin order (DS1.DX=- and DS1.DY=+)	RW	0
6:4	DATA1_POSITION	Position and order of the DATA lane 1. The data lane 1 is always present.  0x1: Data lane 1 is at the position 1 (line 1). 0x2: Data lane 1 is at the position 2 (line 2). 0x3: Data lane 1 is at the position 3 (line 3). 0x4: Data lane 1 is at the position 4 (line 4). 0x5: Data lane 1 is at the position 5 (line 5).	RW	0x0
3	CLOCK_POL	+/- differential pin order of CLOCK lane.  0x0: +/- pin order (DS1.DX=+ and DS1.DY=-) 0x1: -/+ pin order (DS1.DX=- and DS1.DY=+)	RW	0
2:0	CLOCK_POSITION	Position and order of the CLOCK lane. 0, 5, 6 and 7 are reserved. The clock lane is always present but cannot be at the position 5 even if the COMPLEX I/O consists of 5 lanes.  0x1: Clock lane is at the position 1 (line 1). 0x2: Clock lane is at the position 2 (line 2). 0x3: Clock lane is at the position 3 (line 3). 0x4: Clock lane is at the position 4 (line 4).	RW	0x0

**Table 10-531. Register Call Summary for Register DSI\_COMPLEXIO\_CFG1**

MIPI Display Serial Interface

- [Data/Clock Configuration: \[0\]](#)
- [DSI Protocol Overview: \[1\]](#)
- [SCP Interface: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [Power Control of DSI PHY Complex I/O and DSI PLL: \[9\] \[10\]](#)
- [Ultralow-Power State Configuration: \[11\] \[12\]](#)
- [Software Reset: \[13\]](#)
- [Pad Configuration: \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [Display Timing Configuration: \[19\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[20\] \[21\] \[22\] \[23\] \[24\] \[25\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[32\] \[33\]](#)

**Table 10-532. DSI\_COMPLEXIO\_IRQSTATUS**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>		
<b>Description</b>	INTERRUPT STATUS REGISTER - All errors from complex I/O		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ULPSACTIVENOT_ALL1_IRQ	ULPSACTIVENOT_ALL0_IRQ	ERRCONTENTIONLP1_5_IRQ	ERRCONTENTIONLP0_5_IRQ	ERRCONTENTIONLP1_4_IRQ	ERRCONTENTIONLP0_4_IRQ	ERRCONTENTIONLP1_3_IRQ	ERRCONTENTIONLP0_3_IRQ	ERRCONTENTIONLP1_2_IRQ	ERRCONTENTIONLP0_2_IRQ	ERRCONTENTIONLP1_1_IRQ	ERRCONTENTIONLP0_1_IRQ	STATEULPS5_IRQ	STATEULPS4_IRQ	STATEULPS3_IRQ	STATEULPS2_IRQ	STATEULPS1_IRQ	ERRCONTROL5_IRQ	ERRCONTROL4_IRQ	ERRCONTROL3_IRQ	ERRCONTROL2_IRQ	ERRCONTROL1_IRQ	ERRSC5_IRQ	ERRSC4_IRQ	ERRSC3_IRQ	ERRSC2_IRQ	ERRSC1_IRQ	ERRSYNCESC5_IRQ	ERRSYNCESC4_IRQ	ERRSYNCESC3_IRQ	ERRSYNCESC2_IRQ	ERRSYNCESC1_IRQ

Bits	Field Name	Description	Type	Reset
31	ULPSACTIVENOT_ALL1_IRQ	All the ULPSActiveNOT signals corresponding to the lanes with TXULPSExit being high are high.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
30	ULPSACTIVENOT_ALL0_IRQ	All signals ULPSActiveNOT are 0  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
29	ERRCONTENTIONLP1_5_IRQ	Contention LP1 error for lane 5 Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0

## MIPI Display Serial Interface

www.ti.com

Bits	Field Name	Description	Type	Reset
28	ERRCONTENTIONLP0_5_IRQ	Contention LP0 error for lane 5 Applicable only to DS1. For DSI2 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
27	ERRCONTENTIONLP1_4_IRQ	Contention LP1 error for lane 4 Applicable only to DS1. For DSI2 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
26	ERRCONTENTIONLP0_4_IRQ	Contention LP0 error for lane 4 Applicable only to DS1. For DSI2 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
25	ERRCONTENTIONLP1_3_IRQ	Contention LP1 error for lane 3  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
24	ERRCONTENTIONLP0_3_IRQ	Contention LP0 error for lane 3  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
23	ERRCONTENTIONLP1_2_IRQ	Contention LP1 error for lane 2  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
22	ERRCONTENTIONLP0_2_IRQ	Contention LP0 error for lane 2  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
21	ERRCONTENTIONLP1_1_IRQ	Contention LP1 error for lane 1  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
20	ERRCONTENTIONLP0_1_IRQ	Contention LP0 error for lane 1  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
19	STATEULPS5_IRQ	lane 5 in ULTRALOW-POWER State Applicable only to DS1. For DSI2 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0



Bits	Field Name	Description	Type	Reset
18	STATEULPS4_IRQ	lane 4 in ultralow-power mode Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
17	STATEULPS3_IRQ	lane 3 in ULTRALOW-POWER state  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
16	STATEULPS2_IRQ	lane 2 in ULTRALOW-POWER state  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
15	STATEULPS1_IRQ	lane 1 in ULTRALOW-POWER state  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
14	ERRCONTROL5_IRQ	Control error for lane 5 Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
13	ERRCONTROL4_IRQ	Control error for lane 4 Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
12	ERRCONTROL3_IRQ	Control error for lane 3  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
11	ERRCONTROL2_IRQ	Control error for lane 2  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
10	ERRCONTROL1_IRQ	Control error for lane 1  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
9	ERRESC5_IRQ	Escape entry error for lane 5 Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0

Bits	Field Name	Description	Type	Reset
8	ERRESC4_IRQ	Escape entry error for lane 4 Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
7	ERRESC3_IRQ	Escape entry error for lane 3  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
6	ERRESC2_IRQ	Escape entry error for lane 2  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
5	ERRESC1_IRQ	Escape entry error for lane 1  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
4	ERRSYNCESC5_IRQ	Low power Data transmission synchronization error for lane 5 Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
3	ERRSYNCESC4_IRQ	Low power Data transmission synchronization error for lane 4 Applicable only to DS1. For DS12 this bit field is RESERVED.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
2	ERRSYNCESC3_IRQ	Low power Data transmission synchronization error for lane 3  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
1	ERRSYNCESC2_IRQ	Low power Data transmission synchronization error for lane 2  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
0	ERRSYNCESC1_IRQ	Low power Data transmission synchronization error for lane 1  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0

**Table 10-533. Register Call Summary for Register DSI\_COMPLEXIO\_IRQSTATUS**

MIPI Display Serial Interface

- DSI Interrupt Requests: [0] [1]
- Timers: [2]
- Ultralow-Power State Configuration: [3] [4]
- Error Handling: [5] [6] [7] [8] [9] [10] [11] [12] [13]
- DSI Video Mode Using the DISPC Video Port: [14]
- DSI Command Mode Using the DISPC Video Port: [15]
- DSI\_PROTOCOL\_ENGINE Register Summary: [16] [17]
- DSI\_PROTOCOL\_ENGINE Register Description: [18]

**Table 10-534. DSI\_COMPLEXIO\_IRQENABLE**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Physical Address</b>	See Table 10-514 and Table 10-515		
<b>Description</b>	INTERRUPT ENABLE REGISTER - All errors from complex I/O		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ULPSACTIVENOT_ALL1_IRQ_EN	ULPSACTIVENOT_ALL0_IRQ_EN	ERRCONTENTIONLP1_5_IRQ_EN	ERRCONTENTIONLP0_5_IRQ_EN	ERRCONTENTIONLP1_4_IRQ_EN	ERRCONTENTIONLP0_4_IRQ_EN	ERRCONTENTIONLP1_3_IRQ_EN	ERRCONTENTIONLP0_3_IRQ_EN	ERRCONTENTIONLP1_2_IRQ_EN	ERRCONTENTIONLP0_2_IRQ_EN	ERRCONTENTIONLP1_1_IRQ_EN	ERRCONTENTIONLP0_1_IRQ_EN	STATEULPS5_IRQ_EN	STATEULPS4_IRQ_EN	STATEULPS3_IRQ_EN	STATEULPS2_IRQ_EN	STATEULPS1_IRQ_EN	ERRCONTROL5_IRQ_EN	ERRCONTROL4_IRQ_EN	ERRCONTROL3_IRQ_EN	ERRCONTROL2_IRQ_EN	ERRCONTROL1_IRQ_EN	ERRRSC5_IRQ_EN	ERRRSC4_IRQ_EN	ERRRSC3_IRQ_EN	ERRRSC2_IRQ_EN	ERRRSC1_IRQ_EN	ERRSYNCSESC5_IRQ_EN	ERRSYNCSESC4_IRQ_EN	ERRSYNCSESC3_IRQ_EN	ERRSYNCSESC2_IRQ_EN	ERRSYNCSESC1_IRQ_EN

Bits	Field Name	Description	Type	Reset
31	ULPSACTIVENOT_ALL1_IRQ_EN	All the ULPSActiveNOT signals corresponding to the lanes with TXULPSExit being high are high. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
30	ULPSACTIVENOT_ALL0_IRQ_EN	All signals ULPSActiveNOT are 0 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
29	ERRCONTENTIONLP1_5_IRQ_EN	Contention LP1 error for lane 5 Applicable only to DS1. For DS12 this bit field is RESERVED. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
28	ERRCONTENTIONLP0_5_IRQ_EN	Contention LP0 error for lane 5 Applicable only to DS1. For DS12 this bit field is RESERVED. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

## MIPI Display Serial Interface

www.ti.com

Bits	Field Name	Description	Type	Reset
27	ERRCONTENTIONLP1_4_IRQ_EN	Contention LP1 error for lane 4 Applicable only to DS1. For DSI2 this bit field is RESERVED.  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
26	ERRCONTENTIONLP0_4_IRQ_EN	Contention LP0 error for lane 4 Applicable only to DS1. For DSI2 this bit field is RESERVED.  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
25	ERRCONTENTIONLP1_3_IRQ_EN	Contention LP1 error for lane 3  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
24	ERRCONTENTIONLP0_3_IRQ_EN	Contention LP0 error for lane 3  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
23	ERRCONTENTIONLP1_2_IRQ_EN	Contention LP1 error for lane 2  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
22	ERRCONTENTIONLP0_2_IRQ_EN	Contention LP0 error for lane 2  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
21	ERRCONTENTIONLP1_1_IRQ_EN	Contention LP1 error for lane 1  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
20	ERRCONTENTIONLP0_1_IRQ_EN	Contention LP0 error for lane 1  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
19	STATEULPS5_IRQ_EN	lane 5 in ULTRALOW-POWER state Applicable only to DS1. For DSI2 this bit field is RESERVED.  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
18	STATEULPS4_IRQ_EN	lane 4 in ULTRALOW-POWER state Applicable only to DS1. For DSI2 this bit field is RESERVED.  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
17	STATEULPS3_IRQ_EN	lane 3 in ULTRALOW-POWER state  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
16	STATEULPS2_IRQ_EN	lane 2 in ULTRALOW-POWER state  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
15	STATEULPS1_IRQ_EN	lane 1 in ULTRALOW-POWER state  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
14	ERRCONTROL5_IRQ_EN	Control error for lane 5 Applicable only to DS1. For DSI2 this bit field is RESERVED.  0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
13	ERRCONTROL4_IRQ_EN	Control error for lane 4 Applicable only to DS1. For DS12 this bit field is RESERVED. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
12	ERRCONTROL3_IRQ_EN	Control error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
11	ERRCONTROL2_IRQ_EN	Control error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
10	ERRCONTROL1_IRQ_EN	Control error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
9	ERRESC5_IRQ_EN	Escape entry error for lane 5 Applicable only to DS1. For DS12 this bit field is RESERVED. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
8	ERRESC4_IRQ_EN	Escape entry error for lane 4 Applicable only to DS1. For DS12 this bit field is RESERVED. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	ERRESC3_IRQ_EN	Escape entry error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
6	ERRESC2_IRQ_EN	Escape entry error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
5	ERRESC1_IRQ_EN	Escape entry error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	ERRSYNCESEC5_IRQ_EN	Low power Data transmission synchronization error for lane 5 Applicable only to DS1. For DS12 this bit field is RESERVED. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
3	ERRSYNCESEC4_IRQ_EN	Low power Data transmission synchronization error for lane 4 Applicable only to DS1. For DS12 this bit field is RESERVED. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
2	ERRSYNCESEC3_IRQ_EN	Low power Data transmission synchronization error for lane 3 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
1	ERRSYNCESEC2_IRQ_EN	Low power Data transmission synchronization error for lane 2 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
0	ERRSYNCSESC1_IRQ_EN	Low power Data transmission synchronization error for lane 1 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

**Table 10-535. Register Call Summary for Register DSI\_COMPLEXIO\_IRQENABLE**

MIPI Display Serial Interface

- [DSI Video Mode Using the DISPC Video Port: \[0\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[1\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[2\] \[3\]](#)

**Table 10-536. DSI\_CLK\_CTRL**

<b>Address Offset</b>	0x0000 0054		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	CLOCK CONTROL This register controls the CLOCK GENERATION. The register can be modified only when IF_EN is reset.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL_PWR_CMD	PLL_PWR_STATUS	RESERVED						LP_RX_SYNCHRO_ENABLE	LP_CLK_ENABLE	HS_MANUAL_STOP_CTRL	HS_AUTO_STOP_ENABLE	LP_CLK_NULL_PACKET_SIZE	LP_CLK_NULL_PACKET_ENABLE	CIO_CLK_ICG	DDR_CLK_ALWAYS_ON	LP_CLK_DIVISOR															

Bits	Field Name	Description	Type	Reset
31:30	PLL_PWR_CMD	Command for power control of the DSI PLL Control module 0x0: Command to change to OFF state 0x1: Command to change to ON state for PLL only (HSDIVISER is OFF) 0x2: Command to change to ON state for both PLL and HSDIVISER 0x3: Command to change to ON state for both PLL and HSDIVISER (no clock output to the DSI complex I/O)	RW	0x0
29:28	PLL_PWR_STATUS	Status of the power control of the DSI PLL Control module Read 0x0: DSI PLL Control module in OFF state Read 0x1: DSI PLL Control module in ON state for PLL only (HSDIVISER is OFF) Read 0x2: DSI PLL Control module in ON state for both PLL and HSDIVISER Read 0x3: DSI PLL Control module in ON state for both PLL and HSDIVISER (no clock output to the DSI complex I/O)	R	0x0

Bits	Field Name	Description	Type	Reset
27:22	RESERVED		R	0x00
21	LP_RX_SYNCHRO_ENABLE	<p>Defines if the functional is higher or lower than 30 MHz. The information is used to define synchronization to be used for RxValidEsc.</p> <p>0x0: The DSI functional clock is equal or slower than 30 MHz. The synchronization is falling/rising.</p> <p>0x1: The DSI functional clock is higher than 30 MHz. The synchronization is rising/rising.</p>	RW	0
20	LP_CLK_ENABLE	<p>Controls the gating of the TXCLKESC clock.</p> <p>0x0: Disabled. The clock is not generated. The value of LP_CLK_DIVISOR is not used and does not need to be programmed.</p> <p>0x1: Enabled. The clock is generated. The value of LP_CLK_DIVISOR is used and needs to be programmed.</p>	RW	0
19	HS_MANUAL_STOP_CTRL	<p>In case HS_AUTO_STOP_ENABLE=0, the bit field allows manual control of the assertion/de-assertion of the signal DSISStopClk by the user.</p> <p>0x0: DSISStopClk de-assertion unconditionally.</p> <p>0x1: DSISStopClk assertion unconditionally.</p>	RW	0
18	HS_AUTO_STOP_ENABLE	<p>Enables the automatic assertion/de-assertion of DSISStopClk signal.</p> <p>0x0: Auto mode disabled.</p> <p>0x1: Auto mode enabled.</p>	RW	0
17:16	LP_CLK_NULL_PACKET_SIZE	<p>Indicates the size of LP NULL Packets to be sent automatically when after the last LP packet transfer. It is used by the receiver to drain its internal pipeline. The valid values are from 0 to 3 bytes for the payload size.</p>	RW	0x0
15	LP_CLK_NULL_PACKET_ENABLE	<p>Enables the generation of NULL packet in low speed.</p> <p>0x0: Disabled. The NULL packet is not sent in LP mode after the last LP packet.</p> <p>0x1: Enabled. The NULL packet is sent in LP mode after the last LP packet.</p>	RW	0
14	CIO_CLK_ICG	<p>Gates SCPClk clock provided to DSI_PHY and PLL-CTRL module.</p> <p>0x0: Disabled. SCPClk is not generated. It remains at 0.</p> <p>0x1: Enabled. SCPClk is generated (OCP_CLK/4)</p>	RW	0
13	DDR_CLK_ALWAYS_ON	<p>Defines if the DDR clock is also sent when there is no HS packets sent to the peripheral (low-power mode). So TXRequest for the clock lane is not de-asserted.</p> <p>0x0: Disabled. The DDR clock is only provided when HS packets are sent.</p> <p>0x1: Enabled. The DDR clock is always sent to the peripheral regardless of the state of the data lanes (HS or LS mode).</p>	RW	0
12:0	LP_CLK_DIVISOR	<p>Defines the ratio to be used for the generation of the low-power mode clock from DSI functional clock. The supported values are from 1 to 8191(the value 0 is invalid). The output frequency shall be in the range between 20 MHz and 32 kHz.</p>	RW	0x0001

**Table 10-537. Register Call Summary for Register DSI\_CLK\_CTRL**

## MIPI Display Serial Interface

- [DSI Transfer Modes](#): [0]
- [Clock Requirements](#): [1] [2] [3] [4] [5] [6] [7]
- [Power Management](#): [8]
- [Power Control of DSI PHY Complex I/O and DSI PLL](#): [9] [10] [11] [12] [13] [14] [15] [16]
- [Ultralow-Power State Configuration](#): [17]
- [Pad Configuration](#): [18] [19]
- [Display Timing Configuration](#): [20]
- [DSI Video Mode Using the DISPC Video Port](#): [21] [22] [23] [24] [25] [26] [27] [28]
- [DSI Command Mode Using the DISPC Video Port](#): [29] [30] [31] [32] [33] [34] [35] [36]
- [DSI\\_PROTOCOL\\_ENGINE Register Summary](#): [37] [38]
- [DSI\\_PROTOCOL\\_ENGINE Register Description](#): [39] [40]

**Table 10-538. DSI\_TIMING1**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	<b>TIMING1 REGISTER</b> This register controls the DSI Protocol Engine module timers. Any bit field can be modified while <a href="#">DSI_CTRL[0] IF_EN</a> is set to '1'. It is used to indicate the number of DSI1_CLK and DSI2_CLK functional clocks cycles for the timers FORCE_TX_STOP_TIMER and TA_TO_TIMER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TA_TO	TA_TO_X16	TA_TO_X8	TA_TO_COUNTER													FORCE_TX_STOP_MODE_IO	STOP_STATE_X16_IO	STOP_STATE_X4_IO	STOP_STATE_COUNTER_IO												

Bits	Field Name	Description	Type	Reset
31	TA_TO	Enables the turn-around timer 0x0: Turn-around counter is disabled. 0x1: Turn-around counter is enabled (required to receive TA interrupt in case the turn-around procedure is not successful).	RW	0
30	TA_TO_X16	Multiplication factor for the number of DSI_CLK functional clocks cycles defined in TA_TO_COUNTER bit field 0x0: The number of DSI_CLK functional clock cycles defined in TA_TO_COUNTER is multiplied by 1x 0x1: The number of DSI_CLK functional clock cycles defined in TA_TO_COUNTER is multiplied by 16x	RW	1
29	TA_TO_X8	Multiplication factor for the number of DSI_CLK functional clock cycles defined in TA_TO_COUNTER bit field 0x0: The number of DSI_CLK functional clock cycles defined in TA_TO_COUNTER is multiplied by 1x 0x1: The number of DSI_CLK functional clock cycles defined in TA_TO_COUNTER is multiplied by 8x	RW	1



Bits	Field Name	Description	Type	Reset
28:16	TA_TO_COUNTER	Turn around counter. It indicates the number of DSI_CLK functional clock to wait for the change of the Direction PPI signal according to the TurnRequest signal The value is from 0 to 8191.	RW	0x1FFF
15	FORCE_TX_STOP_MODE_IO	Control of ForceTxStopMode signal  0x0: De-assertion of ForceTxStopMode. The hardware reset the bit at the end of the ForceTXStopMode assertion. The software can reset the bit in order to stop the assertion of the ForceTXStopMode signal prior to the completion of the period.  0x1: Assertion of ForceTxStopMode	RW	0
14	STOP_STATE_X16_IO	Multiplication factor for the number of DSI_CLK functional clock cycles defined in STOP_STATE_COUNTER_IO bit field  0x0: The number of DSI_CLK functional clock cycles defined in STOP_STATE_COUNTER_IO is multiplied by 1x  0x1: The number of DSI_CLK functional clock cycles defined in STOP_STATE_COUNTER_IO is multiplied by 16x	RW	1
13	STOP_STATE_X4_IO	Multiplication factor for the number of DSI_CLK functional clock cycles defined in STOP_STATE_COUNTER_IO bit field  0x0: The number of DSI_CLK functional clock cycles defined in STOP_STATE_COUNTER is multiplied by 1x  0x1: The number of DSI_CLK functional clock cycles defined in STOP_STATE_COUNTER_IO is multiplied by 4x	RW	1
12:0	STOP_STATE_COUNTER_IO	Stop state counter. It indicates the number of DSI_CLK functional clock to assert ForceTXStopMode signal. The value is from 0 to 8191.	RW	0x1FFF

**Table 10-539. Register Call Summary for Register DSI\_TIMING1**

MIPI Display Serial Interface

- [Timers: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[10\] \[11\] \[12\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[19\] \[20\]](#)

**Table 10-540. DSI\_TIMING2**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>		
<b>Description</b>	TIMING2 REGISTER This register controls the DSI Protocol Engine module timers. Any bit field can be modified while <a href="#">DSI_CTRL[0] IF_EN</a> is set to '1'. It is used to indicate the number of DSI_FCLK clock cycles for the timer LP_RX_TIMER and the number of BYTE_CLK functional clock cycles for the timer HS_TX_TIMER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS_TX_TO	HS_TX_TO_X64	HS_TX_TO_X16	HS_TX_TO_COUNTER											LP_RX_TO	LP_RX_TO_X16	LP_RX_TO_X4	LP_RX_TO_COUNTER														

Bits	Field Name	Description	Type	Reset
31	HS_TX_TO	Enables the HS TX timer.  0x0: Time-out counter is disabled.  0x1: Time-out counter is enabled (required to receive TA interrupt in case the turn-around procedure is not successful).	RW	0
30	HS_TX_TO_X64	Multiplication factor for the number of BYTE_CLK functional clock cycles defined in HS_TX_COUNTER bit field. BYTE_CLK is a high speed transmit byte clock signal generated by the DSI_PHY.  0x0: The number of BYTE_CLK functional clock cycles defined in HS_TX_TO_COUNTER is multiplied by 1x 0x1: The number of BYTE_CLK functional clock cycles defined in HS_TX_TO_COUNTER is multiplied by 64x	RW	1
29	HS_TX_TO_X16	Multiplication factor for the number of BYTE_CLK functional clock cycles defined in HS_TX_COUNTER bit field.  0x0: The number of BYTE_CLK functional clock cycles defined in HS_TX_TO_COUNTER is multiplied by 1x 0x1: The number of BYTE_CLK functional clock cycles defined in HS_TX_TO_COUNTER is multiplied by 16x	RW	1
28:16	HS_TX_TO_COUNTER	HS_TX_TIMER counter. It indicates the number of BYTE_CLK functional clock for the HS TX timer. The value is from 0 to 8191.	RW	0x1FFF
15	LP_RX_TO	Enables the LP RX timer.  0x0: Turn-around counter is disabled.  0x1: Turn-around counter is enabled (required to receive TA interrupt in case the turn-around procedure is not successful).	RW	0
14	LP_RX_TO_X16	Multiplication factor for the number of DSI_FCLK clock cycles defined in LP_RX_COUNTER bit field  0x0: The number of DSI_FCLK clock cycles defined in LP_RX_TO_COUNTER is multiplied by 1x 0x1: The number of DSI_FCLK clock cycles defined in LP_RX_TO_COUNTER is multiplied by 16x	RW	1
13	LP_RX_TO_X4	Multiplication factor for the number of DSI_FCLK clock cycles defined in LP_RX_COUNTER bit field  0x0: The number of DSI_FCLK clock cycles defined in LP_RX_TO_COUNTER is multiplied by 1x 0x1: The number of DSI_FCLK clock cycles defined in LP_RX_TO_COUNTER is multiplied by 4x	RW	1
12:0	LP_RX_TO_COUNTER	LP_RX_TIMER counter. It indicates the number of DSI_FCLK clock for the LP RX timer. The value is from 0 to 8191.	RW	0x1FFF

**Table 10-541. Register Call Summary for Register DSI\_TIMING2**

MIPI Display Serial Interface

- [Timers: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[9\] \[10\] \[11\] \[12\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[19\] \[20\]](#)

**Table 10-542. DSI\_VM\_TIMING1**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	VIDEO MODE TIMING REGISTER This register defines the video mode timing.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSA								HFP								HBP															

Bits	Field Name	Description	Type	Reset
31:24	HSA	Defines the horizontal Sync active period used in video mode in number of byte clock cycles (TXBYTECLKHS clock) The supported values are from 0 to 255.	RW	0x00
23:12	HFP	Defines the horizontal front porch used in video mode in number of byte clock cycles (TXBYTECLKHS clock) The supported values are from 0 to 4095	RW	0x000
11:0	HBP	Defines the horizontal back porch used in video mode in number of byte clock cycles (TXBYTECLKHS clock) The supported values are from 0 to 4095	RW	0x000

**Table 10-543. Register Call Summary for Register DSI\_VM\_TIMING1**

MIPI Display Serial Interface

- [Blanking: \[0\]](#)
- [DSI Transfer Modes: \[1\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[2\] \[3\] \[4\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[5\] \[6\]](#)

**Table 10-544. DSI\_VM\_TIMING2**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	VIDEO MODE TIMING REGISTER This register defines the video mode timing.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOW_SYNC				VSA				VFP				VBP															

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOW_SYNC	Number of BYTE clock cycles for the synchronization window. An interrupt for synchronization lost is generated when the received synchronzation on video port is not inside the window. DSI does not change its own timings if the synch is inside the window. The valid values are from 4 to 15.	RW	0x0

Bits	Field Name	Description	Type	Reset
23:16	VSA	Defines the vertical Sync active period used in video mode in number of lines. The supported values are from 0 to 255 It is used to generate the short packet for End of Vertical synchronization.	RW	0x00
15:8	VFP	Defines the vertical front porch used in video mode in number of lines. The supported values are from 0 to 255	RW	0x00
7:0	VBP	Defines the vertical back porch used in video mode in number of lines. The supported values are from 0 to 255	RW	0x00

**Table 10-545. Register Call Summary for Register DSI\_VM\_TIMING2**

MIPI Display Serial Interface

- [DSI Transfer Modes: \[0\]](#)
- [Video Port Interface: \[1\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[2\] \[3\] \[4\] \[5\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[6\] \[7\]](#)

**Table 10-546. DSI\_VM\_TIMING3**

<b>Address Offset</b>	0x0000 0068																																																													
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b> DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE																																																												
<b>Description</b>	VIDEO MODE TIMING REGISTER This register defines the video mode timing.																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">31</td><td style="text-align: center;">30</td><td style="text-align: center;">29</td><td style="text-align: center;">28</td><td style="text-align: center;">27</td><td style="text-align: center;">26</td><td style="text-align: center;">25</td><td style="text-align: center;">24</td><td style="text-align: center;">23</td><td style="text-align: center;">22</td><td style="text-align: center;">21</td><td style="text-align: center;">20</td><td style="text-align: center;">19</td><td style="text-align: center;">18</td><td style="text-align: center;">17</td><td style="text-align: center;">16</td><td style="text-align: center;">15</td><td style="text-align: center;">14</td><td style="text-align: center;">13</td><td style="text-align: center;">12</td><td style="text-align: center;">11</td><td style="text-align: center;">10</td><td style="text-align: center;">9</td><td style="text-align: center;">8</td><td style="text-align: center;">7</td><td style="text-align: center;">6</td><td style="text-align: center;">5</td><td style="text-align: center;">4</td><td style="text-align: center;">3</td><td style="text-align: center;">2</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td> </tr> <tr> <td colspan="16" style="text-align: center;">TL</td> <td colspan="12" style="text-align: center;">VACT</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	TL																VACT											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
TL																VACT																																														
Bits	Field Name	Description	Type	Reset																																																										
31:16	TL	Defines the number of length of the line in video mode in number of byte clock cycles (TXBYTECLKHS clock) The supported values are from 0 to 8192. The values from 8193 to 65535 are not supported.	RW	0x0000																																																										
15:0	VACT	Defines the number of active lines used in video mode. The supported values are from 0 to 65535	RW	0x0000																																																										

**Table 10-547. Register Call Summary for Register DSI\_VM\_TIMING3**

MIPI Display Serial Interface

- [Video Port Interface: \[0\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[1\] \[2\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[3\] \[4\]](#)

**Table 10-548. DSI\_CLK\_TIMING**

<b>Address Offset</b>	0x0000 006C		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	CLOCK TIMING REGISTER This register controls the DSI Protocol Engine module timers. This register shall not be modified while <a href="#">DSI_CTRL.IF_EN</a> is set to '1'.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DDR_CLK_PRE								DDR_CLK_POST															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:8	DDR_CLK_PRE	Indicates the number of TXBYTECLKHS clock cycles between the start of the DDR clock and the assertion of the data request signal. The values from 1 to 255 are valid. The value 0 is reserved. The value is not used if <a href="#">DSI_CLK_CTRL[13] DDR_CLK_ALWAYS_ON</a> is set to '1' since the DDR clock is always present.	RW	0x01
7:0	DDR_CLK_POST	Indicates the number of TXBYTECLKHS clock cycles after the de-assertion of the data request signal and the stop of the DDR clock. The values from 1 to 255 are valid. The value 0 is reserved. The value is not used if <a href="#">DSI_CLK_CTRL[13] DDR_CLK_ALWAYS_ON</a> is set to '1' since the DDR clock is always present.	RW	0x01

**Table 10-549. Register Call Summary for Register DSI\_CLK\_TIMING**

MIPI Display Serial Interface

- [Clock Requirements: \[0\] \[1\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[2\] \[3\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[4\] \[5\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[6\] \[7\]](#)

**Table 10-550. DSI\_TX\_FIFO\_VC\_SIZE**

<b>Address Offset</b>	0x0000 0070		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	Defines the corresponding memory entries allocated for each virtual channel. The virtual channel shall be disabled in order to allocate/un-allocate some entries in the TX FIFO.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VC3_FIFO_SIZE			RESERVED	VC3_FIFO_ADD			VC2_FIFO_SIZE			RESERVED	VC2_FIFO_ADD			VC1_FIFO_SIZE			RESERVED	VC1_FIFO_ADD			VC0_FIFO_SIZE			RESERVED	VC0_FIFO_ADD						

Bits	Field Name	Description	Type	Reset
31:28	VC3_FIFO_SIZE	Size of the FIFO allocated for virtual channel 3. For a complete description, refer to <a href="#">Table 10-474, Virtual Channel TX FIFO Size Values</a> .	RW	0x0
27	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
26:24	VC3_FIFO_ADD	Address of the space allocated in the FIFO for virtual channel 3. For a complete description, refer to <a href="#">Table 10-475</a> , <i>Virtual Channel TX FIFO Start Address</i> .	RW	0x0
23:20	VC2_FIFO_SIZE	Size of the FIFO allocated for virtual channel 2. For a complete description, refer to <a href="#">Table 10-474</a> , <i>Virtual Channel TX FIFO Size Values</i> .	RW	0x0
19	RESERVED		R	0
18:16	VC2_FIFO_ADD	Address of the space allocated in the FIFO for virtual channel 2. For a complete description, refer to <a href="#">Table 10-475</a> , <i>Virtual Channel TX FIFO Start Address</i> .	RW	0x0
15:12	VC1_FIFO_SIZE	Size of the FIFO allocated for virtual channel 1. For a complete description, refer to <a href="#">Table 10-474</a> , <i>Virtual Channel TX FIFO Size Values</i> .	RW	0x0
11	RESERVED		R	0
10:8	VC1_FIFO_ADD	Address of the space allocated in the FIFO for virtual channel 1. For a complete description, refer to <a href="#">Table 10-475</a> , <i>Virtual Channel TX FIFO Start Address</i> .	RW	0x0
7:4	VC0_FIFO_SIZE	Size of the FIFO allocated for virtual channel 0. For a complete description, refer to <a href="#">Table 10-474</a> , <i>Virtual Channel TX FIFO Size Values</i> .	RW	0x0
3	RESERVED		R	0
2:0	VC0_FIFO_ADD	Address of the space allocated in the FIFO for virtual channel 0. For a complete description, refer to <a href="#">Table 10-475</a> , <i>Virtual Channel TX FIFO Start Address</i> .	RW	0x0

**Table 10-551. Register Call Summary for Register DSI\_TX\_FIFO\_VC\_SIZE**

MIPI Display Serial Interface

- [DSI Transfer Modes: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[5\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[6\] \[7\]](#)

**Table 10-552. DSI\_RX\_FIFO\_VC\_SIZE**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>		
<b>Description</b>	Defines the corresponding memory entries allocated for each virtual channel and the addresses. The virtual channel shall be disabled in order to allocate/un-allocate some entries in the RX FIFO.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
VC3_FIFO_SIZE				RESERVED	VC3_FIFO_ADD				VC2_FIFO_SIZE				RESERVED	VC2_FIFO_ADD				VC1_FIFO_SIZE				RESERVED	VC1_FIFO_ADD				VC0_FIFO_SIZE				RESERVED	VC0_FIFO_ADD			

Bits	Field Name	Description	Type	Reset
31:28	VC3_FIFO_SIZE	Size of the FIFO allocated for virtual channel 3. For a complete description, refer to <a href="#">Table 10-476</a> , <i>Virtual Channel RX FIFO Size Values</i> .	RW	0x0
27	RESERVED		R	0
26:24	VC3_FIFO_ADD	Address of the space allocated in the FIFO for virtual channel 3. For a complete description, refer to <a href="#">Table 10-477</a> , <i>Virtual Channel RX FIFO Start Address</i> .	RW	0x0

Bits	Field Name	Description	Type	Reset
23:20	VC2_FIFO_SIZE	Size of the FIFO allocated for virtual channel 2. For a complete description, refer to <a href="#">Table 10-476</a> , <i>Virtual Channel RX FIFO Size Values</i> .	RW	0x0
19	RESERVED		R	0
18:16	VC2_FIFO_ADD	Address of the space allocated in the FIFO for virtual channel 2. For a complete description, refer to <a href="#">Table 10-477</a> , <i>Virtual Channel RX FIFO Start Address</i> .	RW	0x0
15:12	VC1_FIFO_SIZE	Size of the FIFO allocated for virtual channel 1. For a complete description, refer to <a href="#">Table 10-476</a> , <i>Virtual Channel RX FIFO Size Values</i> .	RW	0x0
11	RESERVED		R	0
10:8	VC1_FIFO_ADD	Address of the space allocated in the FIFO for virtual channel 1. For a complete description, refer to <a href="#">Table 10-477</a> , <i>Virtual Channel RX FIFO Start Address</i> .	RW	0x0
7:4	VC0_FIFO_SIZE	Size of the FIFO allocated for virtual channel 0. For a complete description, refer to <a href="#">Table 10-476</a> , <i>Virtual Channel RX FIFO Size Values</i> .	RW	0x0
3	RESERVED		R	0
2:0	VC0_FIFO_ADD	Address of the space allocated in the FIFO for virtual channel 0. For a complete description, refer to <a href="#">Table 10-477</a> , <i>Virtual Channel RX FIFO Start Address</i> .	RW	0x0

**Table 10-553. Register Call Summary for Register DSI\_RX\_FIFO\_VC\_SIZE**

MIPI Display Serial Interface

- [DSI Transfer Modes: \[0\] \[1\] \[2\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[3\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[4\] \[5\]](#)

**Table 10-554. DSI\_COMPLEXIO\_CFG2**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>		
<b>Description</b>	COMPLEXIO CONFIGURATION REGISTER for the complex I/O This register contains the lane configuration for the ULPS for each lane.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED														LP_BUSY	HS_BUSY	RESERVED										LANE5_ULPS_SIG2	LANE4_ULPS_SIG2	LANE3_ULPS_SIG2	LANE2_ULPS_SIG2	LANE1_ULPS_SIG2	LANE5_ULPS_SIG1	LANE4_ULPS_SIG1	LANE3_ULPS_SIG1	LANE2_ULPS_SIG1	LANE1_ULPS_SIG1

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17	LP_BUSY	Indicates when there are still pending operations for VCs configured for LP mode. Forced to 1 when at least one VC is enabled and configured for LP mode.  Read 0x0: LP logic is idle Read 0x1: LP logic is active	R	0

Bits	Field Name	Description	Type	Reset
16	HS_BUSY	Indicates when there are still pending operations for VCs configured for HS mode. Forced to 1 when at least one VC is enabled and configured for HS mode. Read 0x0: HS logic is idle Read 0x1: HS logic is active	R	0
15:10	RESERVED		R	0x00
9	LANE5_ULPS_SIG2	Applicable only to DS1. For DS12 this bit field is RESERVED. Enables the ULPS for the lane 5. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane 5 is a data lane. The state of the signal TxUlpsClk is changed if lane 5 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0x0: READ:Inactive state effective. WRITE: Request to change to inactive state 0x1: READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkExc.	RW	0
8	LANE4_ULPS_SIG2	Applicable only to DS1. For DS12 this bit field is RESERVED. Enables the ULPS for the lane 4. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane 4 is a data lane. The state of the signal TxUlpsClk is changed if lane 4 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0x0: READ: Inactive state effective. WRITE: Request to change to inactive state 0x1: READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkExc.	RW	0
7	LANE3_ULPS_SIG2	Enables the ULPS for the lane 3. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane 3 is a data lane. The state of the signal TxUlpsClk is changed if lane 3 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective. 0x0: READ: Inactive state effective. WRITE: Request to change to inactive state 0x1: READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkExc.	RW	0



Bits	Field Name	Description	Type	Reset
6	LANE2_ULPS_SIG2	<p>Enables the ULPS for the lane 2. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane 2 is a data lane. The state of the signal TxUlpsClk is changed if lane 2 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective.</p> <p>0x0: READ: Inactive state effective. WRITE: Request to change to inactive state</p> <p>0x1: READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkExc.</p>	RW	0
5	LANE1_ULPS_SIG2	<p>Enables the ULPS for the lane 1. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxRequestEsc is changed if lane 1 is a data lane. The state of the signal TxUlpsClk is changed if lane 1 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective.</p> <p>0x0: READ: Inactive state effective. WRITE: Request to change to inactive state</p> <p>0x1: READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkExc.</p>	RW	0
4	LANE5_ULPS_SIG1	<p>Applicable only to DS1. For DSI2 this bit field is RESERVED.</p> <p>Enables the ULPS for the lane 5. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane 5 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective.</p> <p>0x0: READ:Inactive state effective. WRITE: Request to change to inactive state</p> <p>0x1: READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpsEsc is asserted for one period of TxClkExc</p>	RW	0

Bits	Field Name	Description	Type	Reset
3	LANE4_ULPS_SIG1	<p>Applicable only to DS1. For DS12 this bit field is RESERVED.</p> <p>Enables the ULPS for the lane 4. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane 4 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective.</p> <p>0x0: READ: Inactive state effective. WRITE: Request to change to inactive state</p> <p>0x1: READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpEsc is asserted for one period of TxClkExc</p>	RW	0
2	LANE3_ULPS_SIG1	<p>Enables the ULPS for the lane 3. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane 3 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective.</p> <p>0x0: READ: Inactive state effective. WRITE: Request to change to inactive state</p> <p>0x1: READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpEsc is asserted for one period of TxClkExc</p>	RW	0
1	LANE2_ULPS_SIG1	<p>Enables the ULPS for the lane 2. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane 2 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective.</p> <p>0x0: READ: Inactive state effective. WRITE: Request to change to inactive state</p> <p>0x1: READ:Active state effective. WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpEsc is asserted for one period of TxClkExc</p>	RW	0
0	LANE1_ULPS_SIG1	<p>Enables the ULPS for the lane 1. The hardware shall change the state of the lane to ULPS only when it is in stop state and there is no data pending inside the DSI protocol engine and the DSI protocol engine has control of the bus (BTA has not been sent). The state of the signal TxULPSExit is changed if the lane 1 is a clock lane. There will be a latency depending on the frequency of TxClkExc. This bit should be read back to confirm a write has been effective.</p> <p>0x0: READ:Inactive state effective. WRITE: Request to change to inactive state</p> <p>0x1: READ:Active state effective WRITE: Change request to active. If the lane is a data lane, TxRequestEsc is asserted and synchronously TxUlpEsc is asserted for one period of TxClkExc</p>	RW	0

**Table 10-555. Register Call Summary for Register DSI\_COMPLEXIO\_CFG2**

MIPI Display Serial Interface

- [Ultralow-Power State Configuration: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[25\] \[26\]](#)

**Table 10-556. DSI\_RX\_FIFO\_VC\_FULLNESS**

<b>Address Offset</b>	0x0000 007C		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	Defines the fullness of each space allocated for each virtual channel.		
<b>Type</b>	R		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
VC3_FIFO_FULLNESS	VC2_FIFO_FULLNESS	VC1_FIFO_FULLNESS	VC0_FIFO_FULLNESS

Bits	Field Name	Description	Type	Reset
31:24	VC3_FIFO_FULLNESS	Fullness of the FIFO allocated for virtual channel 3. The valid values are from 0 to <a href="#">DSI_GNQ[5:3]</a> RX_FIFODEPTH-1 corresponding to 1x33-bit,...up to <a href="#">DSI_GNQ[5:3]</a> RX_FIFODEPTH x33-bit.	R	0x00
23:16	VC2_FIFO_FULLNESS	Fullness of the FIFO allocated for virtual channel 2. The valid values are from 0 to <a href="#">DSI_GNQ[5:3]</a> RX_FIFODEPTH-1 corresponding to 1x33-bit,...up to <a href="#">DSI_GNQ[5:3]</a> RX_FIFODEPTH x33-bit.	R	0x00
15:8	VC1_FIFO_FULLNESS	Fullness of the FIFO allocated for virtual channel 1. The valid values are from 0 to <a href="#">DSI_GNQ[5:3]</a> RX_FIFODEPTH-1 corresponding to 1x33-bit,...up to <a href="#">DSI_GNQ[5:3]</a> RX_FIFODEPTH x33-bit.	R	0x00
7:0	VC0_FIFO_FULLNESS	Fullness of the FIFO allocated for virtual channel 0. The valid values are from 0 to <a href="#">DSI_GNQ[5:3]</a> RX_FIFODEPTH-1 corresponding to 1x33-bit,...up to <a href="#">DSI_GNQ[5:3]</a> RX_FIFODEPTH x33-bit.	R	0x00

**Table 10-557. Register Call Summary for Register DSI\_RX\_FIFO\_VC\_FULLNESS**

MIPI Display Serial Interface

- [DSI Transfer Modes: \[0\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[1\] \[2\]](#)

**Table 10-558. DSI\_VM\_TIMING4**

<b>Address Offset</b>	0x0000 0080		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	VIDEO MODE TIMING REGISTER This register defines the video mode timing.		
<b>Type</b>	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	HSA_HS_INTERLEAVING	HFP_HS_INTERLEAVING	HBP_HS_INTERLEAVING

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	HSA_HS_INTERLEAVING	Defines the number of TXBYTECLKHS clock cycles that can be used for interleaving High Speed command mode packet into Video Mode stream during HSA blanking period. The supported values are from 0 to 255.	RW	0x00
15:8	HFP_HS_INTERLEAVING	Defines the number of TXBYTECLKHS clock cycles that can be used for interleaving High Speed command mode packet into Video Mode stream during HFP blanking period. The supported values are from 0 to 255	RW	0x00
7:0	HBP_HS_INTERLEAVING	Defines the number of TXBYTECLKHS clock cycles that can be used for interleaving High Speed command mode packet into Video Mode stream during HBP blanking period. The supported values are from 0 to 255	RW	0x00

**Table 10-559. Register Call Summary for Register DSI\_VM\_TIMING4**

MIPI Display Serial Interface

- [DSI Transfer Modes: \[0\] \[1\] \[2\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[3\] \[4\]](#)

**Table 10-560. DSI\_TX\_FIFO\_VC\_EMPTYNESS**

<b>Address Offset</b>	0x0000 0084	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>		
<b>Description</b>	Defines the emptiness of each space allocated for each virtual channel.		
<b>Type</b>	R		
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
VC3_FIFO_EMPTYNESS	VC2_FIFO_EMPTYNESS	VC1_FIFO_EMPTYNESS	VC0_FIFO_EMPTYNESS

Bits	Field Name	Description	Type	Reset
31:24	VC3_FIFO_EMPTYNESS	Emptiness of the FIFO allocated for virtual channel 3.The valid values are from 0 to <a href="#">DSI_GNQ[2:0]</a> TX_FIFODEPTH-1 corresponding to 1x33-bit,...up to <a href="#">DSI_GNQ[2:0]</a> TX_FIFODEPTH x33-bit.	R	0x00
23:16	VC2_FIFO_EMPTYNESS	Emptiness of the FIFO allocated for virtual channel 2.The valid values are from 0 to <a href="#">DSI_GNQ[2:0]</a> TX_FIFODEPTH-1 corresponding to 1x33-bit,...up to <a href="#">DSI_GNQ[2:0]</a> TX_FIFODEPTH x33-bit.	R	0x00
15:8	VC1_FIFO_EMPTYNESS	Emptiness of the FIFO allocated for virtual channel 1.The valid values are from 0 to <a href="#">DSI_GNQ[2:0]</a> TX_FIFODEPTH-1 corresponding to 1x33-bit,...up to <a href="#">DSI_GNQ[2:0]</a> TX_FIFODEPTH x33-bit.	R	0x00
7:0	VC0_FIFO_EMPTYNESS	Emptiness of the FIFO allocated for virtual channel 0.The valid values are from 0 to <a href="#">DSI_GNQ[2:0]</a> TX_FIFODEPTH-1 corresponding to 1x33-bit,...up to <a href="#">DSI_GNQ[2:0]</a> TX_FIFODEPTH x33-bit.	R	0x00

**Table 10-561. Register Call Summary for Register DSI\_TX\_FIFO\_VC\_EMPTYNESS**

MIPI Display Serial Interface

- [DSI Transfer Modes: \[0\] \[1\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[2\] \[3\]](#)

**Table 10-562. DSI\_VM\_TIMING5**

<b>Address Offset</b>	0x0000 0088		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	VIDEO MODE TIMING REGISTER This register defines the video mode timing.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HSA_LP_INTERLEAVING				HFP_LP_INTERLEAVING				HBP_LP_INTERLEAVING															

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	HSA_LP_INTERLEAVING	Defines the number of bytes of Low Power command mode packets that can be sent on PPI link during HSA blanking period. The supported values are from 0 to 255.	RW	0x00
15:8	HFP_LP_INTERLEAVING	Defines the number of bytes of Low Power command mode packets that can be sent on PPI link during HFP blanking period. The supported values are from 0 to 255	RW	0x00
7:0	HBP_LP_INTERLEAVING	Defines the number of bytes of Low Power command mode packets that can be sent on PPI link during HBP blanking period. The supported values are from 0 to 255	RW	0x00

**Table 10-563. Register Call Summary for Register DSI\_VM\_TIMING5**

MIPI Display Serial Interface

- [DSI Transfer Modes: \[0\] \[1\] \[2\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[3\] \[4\]](#)

**Table 10-564. DSI\_VM\_TIMING6**

<b>Address Offset</b>	0x0000 008C		
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	VIDEO MODE TIMING REGISTER This register defines the video mode timing.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BL_HS_INTERLEAVING												BL_LP_INTERLEAVING																			

Bits	Field Name	Description	Type	Reset
31:16	BL_HS_INTERLEAVING	Defines the number of TXBYTECLKHS clock cycles that can be used for interleaving High Speed command mode packet into Video Mode stream during blanking periods during VSA, VBP, VFP periods inside one video frame on PPI link. The supported values are from 0 to 65535 .	RW	0x0000
15:0	BL_LP_INTERLEAVING	Defines the maximum number of bytes of Low Power command mode packets that can be sent on PPI link during blanking periods during VSA, VBP or VFP periods inside one video frame on PPI link. The supported values are from 0 to 65535	RW	0x0000

**Table 10-565. Register Call Summary for Register DSI\_VM\_TIMING6**

MIPI Display Serial Interface

- [DSI Transfer Modes: \[0\] \[1\]](#)
- [Video Port Interface: \[2\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[3\] \[4\]](#)

**Table 10-566. DSI\_VM\_TIMING7**

<b>Address Offset</b>	0x0000 0090	
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b> DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	Defines the minimum number of HS bytes clock cycles that are required to allow for the delays in entering and exiting HS mode. The supported values are from 0 to 65535	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENTER_HS_MODE_LATENCY																EXIT_HS_MODE_LATENCY															

Bits	Field Name	Description	Type	Reset
31:16	ENTER_HS_MODE_LATENCY	Defines the number of TXBYTECLKHS clock cycles necessary for entering to HS mode. It corresponds to the delay in number of HS clock cycles from assertion of TxRequestHS signal to 1 until assertion of TxReadyHS signal to 1. The supported values are from 0 to 65535 .	RW	0x0000
15:0	EXIT_HS_MODE_LATENCY	Defines the number of TXBYTECLKHS clock cycles necessary for exiting from HS mode. It corresponds to the maximum delay in number of TXBYTECLKHS clock from de-assertion of TxRequestHS signal until PPI link is in LP-11 state from which a new entrance to HS mode can be initiated which does not take more than ENTER_HS_MODE_LATENCY clock cycles. The supported values are from 0 to 65535	RW	0x0000

**Table 10-567. Register Call Summary for Register DSI\_VM\_TIMING7**

MIPI Display Serial Interface

- [Clock Requirements: \[0\] \[1\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[2\] \[3\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[4\] \[5\]](#)

**Table 10-568. DSI\_STOPCLK\_TIMING**

<b>Address Offset</b>	0x0000 0094	
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b> DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	Number of functional clock cycles to wait for TXBYTECLKHS to stop/start after change in DSIStopClk signal	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DSI_STOPCLK_LATENCY															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	DSI_STOPCLK_LATENCY	Clock gating latency from DSI Protocol to TXBYTECLKHS	RW	0x80

**Table 10-569. Register Call Summary for Register DSI\_STOPCLK\_TIMING**

MIPI Display Serial Interface

- [Power Control of DSI PHY Complex I/O and DSI PLL: \[0\] \[1\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[2\] \[3\]](#)

**Table 10-570. DSI\_CTRL2**

<b>Address Offset</b>	0x0000 0098	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>		
<b>Description</b>	Additional control bits for use with Video Port 2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE_BUFFER	VP_VSYNC_POL	VP_HSYNC_POL	VP_DE_POL	VP_CLK_POL	VP_DATA_BUS_WIDTH	RESERVED	VP_CLK_RATIO	RESERVED							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x000000
13:12	LINE_BUFFER	Number of line buffers to be used while receiving data on the video port. The valid values are from 0 to <a href="#">DSI_GNQ[23:22]</a> VP2_NB_LINE_BUFFER. 0x0: No line buffer 0x1: 1 line buffer 0x2: 2 line buffers	RW	0x0
11	VP_VSYNC_POL	VP vertical synchronization signal polarity 0x0: VSYNC signal on the video port is active low. 0x1: VSYNC signal on the video port is active high.	RW	0
10	VP_HSYNC_POL	VP horizontal synchronization signal polarity 0x0: HSYNC signal on the video port is active low. 0x1: HSYNC signal on the video port is active high.	RW	0
9	VP_DE_POL	VP data enable signal polarity 0x0: DE signal on the video port is active low. 0x1: DE signal on the video port is active high.	RW	0
8	VP_CLK_POL	VP pixel clock polarity 0x0: The DSI Protocol Engine module captures the data on the VP on the pixel clock falling edge. The module connected to the VP shall drive the data on the pixel clock rising edge. 0x1: The DSI Protocol Engine module captures the data on the VP on the pixel clock raising edge. The module connected to the VP shall drive the data on the pixel clock falling edge.	RW	1

Bits	Field Name	Description	Type	Reset
7:6	VP_DATA_BUS_WIDTH	Defines the size of the video port data bus 0x0: 16-bits data width (LSB of the 24-bit video port data bus) 0x1: 18-bits data width (LSB of the 24-bit video port data bus) 0x2: 24-bits data width (LSB of the 24-bit video port data bus)	RW	0x0
5	RESERVED		R	0
4	VP_CLK_RATIO	The field indicates the clock ratio between VP_CLK and VP_PCLK. The clock VP_PCLK is generated from VP_CLK. It is divided down. The information is only used when the video port is used to provide data in command mode. In the case of video mode, it is not used. 0x0: The clock VP_PCLK is the clock VP_CLK divided by 2. The duty cycle of VP_PCLK is 50/50. 0x1: The clock VP_PCLK is the clock VP_CLK divided by 3 or more. The duty cycle of VP_PCLK is not 50/50 for odd ratio numbers (3,5,7,...).	RW	0
3:0	RESERVED		R	0x0

**Table 10-571. Register Call Summary for Register DSI\_CTRL2**

MIPI Display Serial Interface

- [Video Port Interface: \[0\] \[1\] \[2\] \[3\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[4\] \[5\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Description: \[6\] \[7\] \[8\] \[9\]](#)

**Table 10-572. DSI\_VM\_TIMING8**

<b>Address Offset</b>	0x0000 009C																																																			
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b> DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE																																																		
<b>Description</b>	VIDEO MODE TIMING REGISTER This register defines the video mode timing.																																																			
<b>Type</b>	RW																																																			
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="2">HFPX</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																HFPX	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																					
RESERVED																HFPX																																				
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																
31:2	RESERVED		R	0x0000 0000																																																
1:0	HFPX	Extension to the HFP register. Additional bits added to MSB.	RW	0x0																																																

**Table 10-573. Register Call Summary for Register DSI\_VM\_TIMING8**

MIPI Display Serial Interface

- [Blanking: \[0\]](#)
- [DSI Transfer Modes: \[1\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[2\] \[3\]](#)



**Table 10-574. DSI\_TE\_HSYNC\_WIDTH\_j**

<b>Address Offset</b>	0x0000 00A0 + (0xc * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	The register configures the TE HSYNC minimum pulse width for TE0 and TE1 CMOS signals. The input TE signal is asynchronous and needs to be resynchronized to DSI_CLK clock domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								MIN_HSYNC_PULSE_WIDTH								RESERVED															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:8	MIN_HSYNC_PULSE_WIDTH	Programmable min HSYNC pulse width. Minimum HSYNC pulse width. Number of DSI_CLK clock cycles times 256 to determine when HSYNC pulse occurs. The value 0 is invalid.	RW	0x001
7:0	RESERVED		R	0x00

**Table 10-575. Register Call Summary for Register DSI\_TE\_HSYNC\_WIDTH\_j**

MIPI Display Serial Interface

- [PHY Triggers: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[5\] \[6\]](#)

**Table 10-576. DSI\_TE\_VSYNC\_WIDTH\_j**

<b>Address Offset</b>	0x0000 00A4 + (0xc * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	The register configures the TE VSYNC minimum pulse width for TE0 and TE1 CMOS signals. The input TE signal is asynchronous and needs to be resynchronized to DSI_CLK clock domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								MIN_VSYNC_PULSE_WIDTH								RESERVED															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:8	MIN_VSYNC_PULSE_WIDTH	Programmable min VSYNC pulse width. Minimum VSYNC pulse width. Number of DSI_CLK clock cycles times 256 to determine when VSYNC pulse occurs. The value 0 is invalid. The value shall be greater than MIN_HSYNC_PULSE_WIDTH when DSI_TE_HSYNC_NUMBER is greater than 0.	RW	0x001
7:0	RESERVED		R	0x00

**Table 10-577. Register Call Summary for Register DSI\_TE\_VSYNC\_WIDTH\_j**

MIPI Display Serial Interface

- [PHY Triggers: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[6\] \[7\]](#)

**Table 10-578. DSI\_TE\_HSYNC\_NUMBER\_j**

<b>Address Offset</b>	0x0000 00A8 + (0xc * j)	<b>Index</b>	j = 0 to 1
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	The register configures the number of HSYNC to synchronize the beginning of the transfer on DSI link based on the number of HSYNC pulse received on the TE line. The input TE signal is asynchronous and needs to be resynchronized to DSI_CLK clock domain.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LINE_NUMBER															

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10:0	LINE_NUMBER	Programmable line number Line number from 0 to 2047. Number of HSYNC after the VSYNC occurs before the beginning of the transfer. Any HSYNC before VSYNC is ignored.	RW	0x000

**Table 10-579. Register Call Summary for Register DSI\_TE\_HSYNC\_NUMBER\_j**

MIPI Display Serial Interface

- [PHY Triggers: \[0\] \[1\] \[2\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[3\] \[4\]](#)

**Table 10-580. DSI\_VC\_CTRL\_i**

<b>Address Offset</b>	0x0000 0100 + (0x20 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	CONTROL REGISTER - Virtual channel This register controls the virtual channel.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DCS_CMD_CODE	DCS_CMD_ENABLE	DMA_RX_REQ_NB	DMA_RX_THRESHOLD	DMA_TX_REQ_NB	RX_FIFO_NOT_EMPTY	DMA_TX_THRESHOLD	TX_FIFO_FULL	VC_BUSY	PP_BUSY	VP_SOURCE	RGB565_ORDER	OCP_DATA_BUS_WIDTH	MODE_SPEED	ECC_TX_EN	CS_TX_EN	BTA_EN	TX_FIFO_NOT_EMPTY	MODE	BTA_LONG_EN	BTA_SHORT_EN	SOURCE	VC_EN										

Bits	Field Name	Description	Type	Reset
31	DCS_CMD_CODE	DCS command code value to insert between header and video port or Interface slave data when enabled by DCS_CMD_ENABLE (only when TE mechanism is not used otherwise the bit field DCS_CMD_CODE is ignored by DSI protocol engine).  0x0: DCS write memory continue code is inserted. 0x1: DCS write memory start code is inserted.	RW	0

Bits	Field Name	Description	Type	Reset
30	DCS_CMD_ENABLE	Enables automatic insertion of DCS command codes when data is sourced by the video port.  0x0: DCS command code is NOT inserted when command mode traffic is coming from the Video Port or Interface slave port.  0x1: DCS command code is inserted automatically when command mode traffic is coming from the Video Port or Interface slave port.	RW	0
29:27	DMA_RX_REQ_NB	Selection of the use of the DMA request (associated to the RX FIFO)  0x0: DMA_req0 is selected (valid only if DSI_GNQ[8:6] NB_DMA_REQUEST is greater than 0)  0x1: DMA_req1 is selected (valid only if DSI_GNQ[8:6] NB_DMA_REQUEST is greater than 1)  0x2: DMA_req2 is selected (valid only if DSI_GNQ[8:6] NB_DMA_REQUEST is greater than 2)  0x3: DMA_req3 is selected (valid only if DSI_GNQ[8:6] NB_DMA_REQUEST is equal to 3)  0x4: No DMA req selected	RW	0x0
26:24	DMA_RX_THRESHOLD	Defines the threshold value for the DMA request (associated to the RX FIFO)  0x0: 1 x 32 bits  0x1: 2 x 32 bits  0x2: 4 x 32 bits  0x3: 8 x 32 bits  0x4: 16 x 32 bits  0x5: 32 x 32 bits	RW	0x0
23:21	DMA_TX_REQ_NB	Selection of the use of the DMA request (associated to the TX FIFO)  0x0: DMA_req0 is selected (valid only if DSI_GNQ[8:6] NB_DMA_REQUEST is greater than 0)  0x1: DMA_req1 is selected (valid only if DSI_GNQ[8:6] NB_DMA_REQUEST is greater than 1)  0x2: DMA_req2 is selected (valid only if DSI_GNQ[8:6] NB_DMA_REQUEST is greater than 2)  0x3: DMA_req3 is selected (valid only if DSI_GNQ[8:6] NB_DMA_REQUEST is equal to 3)  0x4: No DMA req selected	RW	0x0
20	RX_FIFO_NOT_EMPTY	FIFO status in command mode. Otherwise, this bit can be ignored.  Read 0x0: The RX FIFO is empty (the FIFO does not contain any data for the virtual channel)  Read 0x1: The RX FIFO is not empty (the FIFO contains at least one byte for the virtual channel)	R	0
19:17	DMA_TX_THRESHOLD	Defines the threshold value for the DMA request (associated to the TX FIFO)  0x0: 1 x 32 bits  0x1: 2 x 32 bits  0x2: 4 x 32 bits  0x3: 8 x 32 bits  0x4: 16 x 32 bits  0x5: 32 x 32 bits	RW	0x0

Bits	Field Name	Description	Type	Reset
16	TX_FIFO_FULL	FIFO status in command mode. Otherwise, this bit can be ignored. Read 0x0: The TX FIFO is not full (the FIFO can accept at least one more 32-bit value) Read 0x1: The TX FIFO is full	R	0
15	VC_BUSY	Indicates if previously scheduled activities (packets, BTA) are still being processed. Forced to 1 if VC is enabled. Software should check this bit is 0 before changing channel configuration. Read 0x0: No pending operations for this VC Read 0x1: Pending operations for this VC	R	0
14	PP_BUSY	Ping-pong buffer busy status. Read 0x0: Software is permitted to write a new header for VP command mode traffic. Read 0x1: Software is NOT permitted to write a new header for VP command mode traffic.	R	0
13	VP_SOURCE	Selection between video port 1 and video port 2. If <code>DSI_GNQ[24] NB_VIDEO_PORTS = 0</code> , the bit field is ignored since only video port 1 is present. 0x0: The video port 1 is selected. 0x1: The video port 2 is selected. Note: This selection applies only to DSI1. For DSI2 to work properly, the VP_SOURCE bit must always be set to 0x0.	RW	0
12	RGB565_ORDER	Byte order for RGB565 0x0: Byte order as for DBI compliance 0x1: Byte order as for video mode	RW	0
11:10	OCP_DATA_BUS_WIDTH	Defines the size of the Interface data bus 0x0: 16-bits data width (LSB of the 32-bit Interface port data bus) 0x1: 24-bits data width (LSB of the 32-bit Interface port data bus) 0x2: 2x16-bits data width (first pixel on the LSB of the 32-bit Interface port data bus and second pixel on the MSB of the 32-bit Interface port data bus for the same Interface access) 0x3: 32-bits data width	RW	0x0
9	MODE_SPEED	Selection of the mode. The information is used by hardware only if <code>MODE=COMMAND_MODE</code> otherwise it is ignored. 0x0: Low-power mode (CMOS) is used to send short and long packets to the peripheral. 0x1: High Speed mode (SLVS) is used to send short and long packets to the peripheral.	RW	0
8	ECC_TX_EN	Enables the Error Correction Code generation for the transmit header (short and long packets). 0x0: Disabled 0x1: Enabled	RW	0
7	CS_TX_EN	Enables the checksum generation for the transmit payload (long packet only). 0x0: Disabled. The value 0x00 is used. 0x1: Enabled. The checksum value is calculated by HW.	RW	0

Bits	Field Name	Description	Type	Reset
6	BTA_EN	<p>Send the bus turn around to the peripheral. It can be used when the automatic mode is enabled (BTA_SHORT_EN=1 or/and BTA_LONG_EN=1). In that case only one BTA is sent to the peripheral. The manual mode is used to allow the user to define for which packets, the turn around is required for example getting acknowledge from the peripheral.</p> <p>0x0: READS: BTA generation is completed. It is reset by hardware when it is completed. WRITES: Cancellation of the BTA generation (not guarantee since it could already on going, shall not be used).</p> <p>0x1: READS: BTA generation has been requested by user (it could be on going but not completed). WRITES: Request for BTA generation.</p>	RW	0
5	TX_FIFO_NOT_EMPTY	<p>FIFO status</p> <p>Read 0x0: The TX FIFO is empty (the FIFO does not contain any data for the virtual channel)</p> <p>Read 0x1: The TX FIFO is not empty (the FIFO contains at least one byte for the virtual channel)</p>	R	0
4	MODE	<p>Selection of the mode</p> <p>0x0: Command mode.</p> <p>0x1: Video mode. The bit fields MODE_SPEED and SOURCE are not used by hardware. The bit field VP_SOURCE is used to select between video port 1 and video port 2 when two video ports are present.</p>	RW	0
3	BTA_LONG_EN	<p>Enables the automatic bus turn-around after completion of each long packet transmission.</p> <p>0x0: Disabled</p> <p>0x1: Enabled</p>	RW	0
2	BTA_SHORT_EN	<p>Enables the automatic bus turn-around after completion of each short packet transmission.</p> <p>0x0: Disabled</p> <p>0x1: Enabled</p>	RW	0
1	SOURCE	<p>Selection of the source between Interface and the Video port(s) (used by the hardware only if MODE=COMMAND MODE (0x0) otherwise it is ignored). The number of available video ports is defined in <a href="#">DSI_GNQ[24]</a> NB_VIDEO_PORTS bit.</p> <p>0x0: All the data are provided by the slave port. Any transfer on the video port is ignored for this virtual channel.</p> <p>0x1: If MODE=VIDEO_MODE. the data received on the video port 1 or video port 2, selected using VP_SOURCE (pixels and enabled synchronization events using <a href="#">DSI_CTRL[17]</a> VP_HSYNC_START, <a href="#">DSI_CTRL[18]</a> VP_HSYNC_END, <a href="#">DSI_CTRL[15]</a> VP_VSYNC_START, <a href="#">DSI_CTRL[16]</a> VP_VSYNC_END for video port 1 and <a href="#">DSI_CTRL2[]</a> VP_HSYNC_START, <a href="#">DSI_CTRL2[]</a> VP_HSYNC_END, <a href="#">DSI_CTRL2[]</a> VP_VSYNC_START, <a href="#">DSI_CTRL2[]</a> VP_VSYNC_END for video port 2) are sent on the virtual channel (only one virtual channel can be associated with the video port, it is the software responsibility to ensure that no more than one virtual channel is enabled with the video port as the main source for data).</p> <p>If MODE=COMMAND_MODE, the VP.STALL signal is used by the protocol engine to indicate when new data are required. The synchronization signals are not generated by the display controller. Regardless of the MODE, no data can be provided on the Interface slave port.</p>	RW	0

Bits	Field Name	Description	Type	Reset
0	VC_EN	<p>Enables the virtual channel.</p> <p>0x0: Disabled. The virtual channel shall be disabled for any register change in the DSI_VC_... registers to the corresponding VC ID (except for setting the <a href="#">DSI_VC_CTRL_i</a>[6] BTA_EN, <a href="#">DSI_VC_TE_i</a>[23:0] TE_SIZE and <a href="#">DSI_VC_TE_i</a>[31] TE_START bit fields, and <a href="#">DSI_VC_LONG_...</a>, <a href="#">DSI_VC_SHORT_...</a>, <a href="#">DSI_VC_IRQ...</a> registers).</p> <p>0x1: Enabled. No change is allowed to the virtual channel registers expect resetting the VC_EN.</p>	RW	0

**Table 10-581. Register Call Summary for Register DSI\_VC\_CTRL\_i**

## MIPI Display Serial Interface

- [Short Packet: \[0\]](#)
- [Long Packet: \[1\]](#)
- [Blanking: \[2\] \[3\] \[4\] \[5\]](#)
- [VCs: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\]](#)
- [16 bpp – RGB Color Format, Long Packet \(Command Mode\): \[17\]](#)
- [DSI Global Register Controls: \[18\]](#)
- [DSI Transfer Modes: \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\]](#)
- [Video Port Interface: \[44\] \[45\] \[46\]](#)
- [Timers: \[47\] \[48\]](#)
- [Bus Turnaround: \[49\] \[50\] \[51\] \[52\] \[53\] \[54\] \[55\]](#)
- [PHY Triggers: \[56\] \[57\] \[58\]](#)
- [ECC Generation: \[59\] \[60\]](#)
- [Checksum Generation for Long Packet Payloads: \[61\]](#)
- [Power Control of DSI PHY Complex I/O and DSI PLL: \[62\] \[63\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[64\] \[65\] \[66\] \[67\] \[68\] \[69\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[70\] \[71\] \[72\] \[73\] \[74\] \[75\] \[76\] \[77\] \[78\] \[79\] \[80\] \[81\] \[82\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[83\] \[84\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Description: \[85\] \[86\]](#)

**Table 10-582. DSI\_VC\_TE\_i**

<b>Address Offset</b>	0x0000 0104 + (0x20 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	CONTROL REGISTER - Virtual channel This register controls the tearing effect logic. It defines the size of the transfer when TE occurs and enables the automatic TE mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE_START	TE_EN	TE_LINE	TE_LINE_NB	RESERVED				TE_SIZE																							

Bits	Field Name	Description	Type	Reset
31	TE_START	<p>Manual control of the start of the transfer. The user can use the TE interrupt in order to know that the TE trigger has been received prior to set the TE_START bit field. It is not mandatory to use the TE interrupt.</p> <p>0x0: Indicates the end of the transfer. The bit can be used by user to cancel the transfer if not already started. The FIFO shall be flushed by software to ensure there is no data remaining in it.</p> <p>0x1: Starts the transfer of the data. The size is defined in TE_SIZE. The bit field is set until the transfer is completed. It is reset by hardware when the transfer is completed.</p>	RW	0
30	TE_EN	<p>Tearing Effect Control</p> <p>0x0: Disables the automatic transfer. The user shall use the interruption in order to know when TE PHY trigger is received or when the TE is detected on the input CMOS signals. The hardware reset the bit field when the transfer is completed (TE_SIZE=0).</p> <p>0x1: Enables the automatic transfer of the data using the TE PHY trigger or one of the TE input signals as a synchronization event. The bit field TE_LINE defines if the CMOS signal is used or if the PHY trigger is used.</p>	RW	0
29	TE_LINE	<p>0x0: Disabled the TE CMOS signalling for the automatic data transfer. The DSI PHY trigger is used for the automatic data transfer.</p> <p>0x1: Enables the TE CMOS signalling for the automatic data transfer. The DSI PHY trigger is not used for the automatic data transfer.</p>	RW	0
28	TE_LINE_NB	<p>Selection between TE0 and TE1 CMOS signals.</p> <p>0x0: TE0 CMOS input line is selected</p> <p>0x1: TE1 CMOS input line is selected</p>	RW	0
27:24	RESERVED		R	0x0
23:0	TE_SIZE	<p>Defines the number of byte (payload data excluding the check -sum) to be sent. The write into the register <a href="#">DSI_VC_LONG_PACKET_HEADER_i</a> shall be performed by the user before sending data from the register <a href="#">DSI_VC_LONG_PACKET_PAYLOAD_i</a>. The register value is decremented for every byte sent of the DSI link. At the end of the transfer (TE_SIZE=0), the bit field TE_EN is reset by hardware. The DMA request will be asserted when the trigger is received in order to receive data in the TX FIFO. It should not be deasserted until all data (TE_SIZE) have been received in the FIFO.</p>	RW	0x000000

**Table 10-583. Register Call Summary for Register DSI\_VC\_TE\_i**

## MIPI Display Serial Interface

- [PHY Triggers: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[21\] \[22\] \[23\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[24\] \[25\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Description: \[26\] \[27\]](#)

**Table 10-584. DSI\_VC\_LONG\_PACKET\_HEADER\_i**

<b>Address Offset</b>	0x0000 0108 + (0x20 * i)	<b>Index</b>	i = 0 to 3																																																																
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE																																																																
<b>Description</b>	<p>LONG PACKET HEADER INFORMATION -Virtual channel  This register sets the 32-bit DATA_ID + Word count + ECC (the virtual channel id can be different than VC).  The ECC will be computed if ECC_TX_EN is set to 1.  DATA_ID is located at bit[7:0]  WC is located at bit[23:8]  ECC is located at bit[31:24]  (Least significant byte first and least significant bit first)</p>																																																																		
<b>Type</b>	W																																																																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 8%;">31</td><td style="width: 3%;">30</td><td style="width: 3%;">29</td><td style="width: 3%;">28</td><td style="width: 3%;">27</td><td style="width: 3%;">26</td><td style="width: 3%;">25</td><td style="width: 3%;">24</td><td style="width: 3%;">23</td><td style="width: 3%;">22</td><td style="width: 3%;">21</td><td style="width: 3%;">20</td><td style="width: 3%;">19</td><td style="width: 3%;">18</td><td style="width: 3%;">17</td><td style="width: 3%;">16</td><td style="width: 3%;">15</td><td style="width: 3%;">14</td><td style="width: 3%;">13</td><td style="width: 3%;">12</td><td style="width: 3%;">11</td><td style="width: 3%;">10</td><td style="width: 3%;">9</td><td style="width: 3%;">8</td><td style="width: 3%;">7</td><td style="width: 3%;">6</td><td style="width: 3%;">5</td><td style="width: 3%;">4</td><td style="width: 3%;">3</td><td style="width: 3%;">2</td><td style="width: 3%;">1</td><td style="width: 3%;">0</td> </tr> <tr> <td colspan="32" style="text-align: center;">HEADER</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	HEADER																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
HEADER																																																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																															
31:0	HEADER	Packet header information: DATA ID + WORD COUNT +ECC	W	0x0000 0000																																																															

**Table 10-585. Register Call Summary for Register DSI\_VC\_LONG\_PACKET\_HEADER\_i**

## MIPI Display Serial Interface

- [Long Packet: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Data Identifier: \[7\]](#)
- [VCs: \[8\]](#)
- [DSI Global Register Controls: \[9\]](#)
- [DSI Transfer Modes: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\]](#)
- [Video Port Interface: \[26\]](#)
- [Bus Turnaround: \[27\]](#)
- [PHY Triggers: \[28\] \[29\] \[30\]](#)
- [ECC Generation: \[31\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[32\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[33\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[34\] \[35\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Description: \[36\]](#)



**Table 10-586. DSI\_VC\_LONG\_PACKET\_PAYLOAD\_i**

<b>Address Offset</b>	0x0000 010C + (0x20 * i)	<b>Index</b>	i = 0 to 3																																																																
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE																																																																
<b>Description</b>	<p>LONG PACKET PAYLOAD INFORMATION -Virtual channel                      This register sets the payload information (excluding checksum). The hardware shall capture the word count in the packet header (in DSI_VC_LONG_PACKET_HEADER) in order to determine the last valid data (the virtual channel id can be different than VC).                      Byte1 is bit[7:0]                      Byte2 is bit[15:8]                      Byte3 is bit[23:16]                      Byte4 is bit[31:24]                      Byten is sent before Byten+1 (Least significant byte first and least significant bit first)</p>																																																																		
<b>Type</b>	W																																																																		
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">PAYLOAD</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	PAYLOAD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
PAYLOAD																																																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																															
31:0	PAYLOAD	Packet payload information (excluding checksum)	W	0x0000 0000																																																															

**Table 10-587. Register Call Summary for Register DSI\_VC\_LONG\_PACKET\_PAYLOAD\_i**

- MIPI Display Serial Interface
- [Long Packet](#): [0] [1] [2]
  - [DSI Global Register Controls](#): [3]
  - [DSI Transfer Modes](#): [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]
  - [Bus Turnaround](#): [14]
  - [PHY Triggers](#): [15] [16]
  - [DSI\\_PROTOCOL\\_ENGINE Register Summary](#): [17] [18]
  - [DSI\\_PROTOCOL\\_ENGINE Register Description](#): [19]

**Table 10-588. DSI\_VC\_SHORT\_PACKET\_HEADER\_i**

<b>Address Offset</b>	0x0000 0110 + (0x20 * i)	<b>Index</b>	i = 0 to 3																																																																
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE																																																																
<b>Description</b>	<p>SHORT PACKET HEADER INFORMATION -Virtual channel                      This register sets the 24-bit DATA_ID + Short Packet Data Field + ECC (the virtual channel id can be different than VC).                      The ECC will be computed if ECC_TX_EN is set to 1.                      DATA_ID is located at bit[7:0]                      Short Packet Data field is located at bit[23:8]                      ECC is located at bit[31:24]                      (Least significant byte first and least significant bit first)</p>																																																																		
<b>Type</b>	RW																																																																		
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">HEADER</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	HEADER																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
HEADER																																																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																															
31:0	HEADER	WRITES: Packet header information: DATA ID + DATA FIELD +ECC written into the TX FIFO READS: 32-bit values read from the RX FIFO	RW	0x0000 0000																																																															

**Table 10-589. Register Call Summary for Register DSI\_VC\_SHORT\_PACKET\_HEADER\_i**

MIPI Display Serial Interface

- [Short Packet: \[0\]](#)
- [Data Identifier: \[1\]](#)
- [VCs: \[2\]](#)
- [DSI Global Register Controls: \[3\]](#)
- [DSI Transfer Modes: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [PHY Triggers: \[10\]](#)
- [ECC Generation: \[11\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[12\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[13\] \[14\]](#)

**Table 10-590. DSI\_VC\_IRQSTATUS\_i**

<b>Address Offset</b>	0x0000 0118 + (0x20 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	INTERRUPT STATUS REGISTER - Virtual channel This register regroups all the events related to the virtual channel.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																								PP_BUSY_CHANGE_IRQ	FIFO_TX_UDF_IRQ	ECC_NO_CORRECTION_IRQ	BTA_IRQ	FIFO_RX_OVF_IRQ	FIFO_TX_OVF_IRQ	PACKET_SENT_IRQ	ECC_CORRECTION_IRQ	CS_IRQ

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	PP_BUSY_CHANGE_IRQ	Video Port ping-pong buffer busy status. PP_BUSY has changed from 1 to 0.  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
7	FIFO_TX_UDF_IRQ	FIFO underflow status. The FIFO used on the L3 interconnect slave port for buffering the data for the virtual channel has underflowed which means that the data for the current packet have not been received in time since the transfer of the packet are already started (transfer started since the packet size is bigger than space allocated in the FIFO).  0x0: READS: Event is false. WRITES: Status bit unchanged.  0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0

Bits	Field Name	Description	Type	Reset
6	ECC_NO_CORRECTION_IRQ	ECC error status (short and long packets). No correction of the header because of more than 1-bit error. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
5	BTA_IRQ	Virtual channel - BTA status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
4	FIFO_RX_OVF_IRQ	FIFO overflow error status. The FIFO used on the L3 interconnect slave port for buffering the data received on the DSI link for the virtual channel has overflowed. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
3	FIFO_TX_OVF_IRQ	FIFO overflow error status. The FIFO used on the L3 interconnect slave port for buffering the data for the virtual channel has overflowed. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
2	PACKET_SENT_IRQ	Indicates that a packet has been sent. It is used when BTA manual mode is used. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
1	ECC_CORRECTION_IRQ	Virtual channel - ECC has been used to do the correction of the only 1-bit error status (short and long packet). 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0
0	CS_IRQ	Virtual channel - checksum mismatch status. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0

**Table 10-591. Register Call Summary for Register DSI\_VC\_IRQSTATUS\_i**

## MIPI Display Serial Interface

- [DSI Interrupt Requests: \[0\]](#)
- [DSI Transfer Modes: \[1\] \[2\] \[3\]](#)
- [Video Port Interface: \[4\]](#)
- [Bus Turnaround: \[5\] \[6\]](#)
- [Interrupts: \[7\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[14\] \[15\]](#)

**Table 10-592. DSI\_VC\_IRQENABLE\_i**

<b>Address Offset</b>	0x0000 011C + (0x20 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 10-514</a> and <a href="#">Table 10-515</a>	<b>Instance</b>	DSI1_PROTOCOL_ENGINE DSI2_PROTOCOL_ENGINE
<b>Description</b>	INTERRUPT ENABLE REGISTER - Virtual channel This register regroups all the events related to virtual channel.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PP_BUSY_CHANGE_IRQ_EN	FIFO_TX_UDF_IRQ_EN	ECC_NO_CORRECTION_IRQ_EN	BTA_IRQ_EN	FIFO_RX_OVF_IRQ_EN	FIFO_TX_OVF_IRQ_EN	PACKET_SENT_IRQ_EN	ECC_CORRECTION_IRQ_EN	CS_IRQ_EN							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0000000
8	PP_BUSY_CHANGE_IRQ_EN	Video Port ping-pong buffer busy status. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
7	FIFO_TX_UDF_IRQ_EN	FIFO underflow enable. The FIFO used for buffering the data received on the L3 interface slave port for the virtual channel has underflowed which means that the data for the current packet have not been received in time since the transfer of the packet are already started (transfer started since the packet size is bigger than space allocated in the FIFO). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
6	ECC_NO_CORRECTION_IRQ_EN	ECC error (short and long packets). No correction of the header because of more than 1-bit error. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
5	BTA_IRQ_EN	Virtual channel -Bus turn around reception 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
4	FIFO_RX_OVF_IRQ_EN	FIFO overflow enable. The FIFO used on the L3 interconnect slave port for buffering the data received on the DSI link for the virtual channel has overflowed. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
3	FIFO_TX_OVF_IRQ_EN	FIFO overflow enable. The FIFO used on the L3 interconnect slave port for buffering the data received on the Interface slave port for the virtual channel has overflowed. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

Bits	Field Name	Description	Type	Reset
2	PACKET_SENT_IRQ_EN	Indicates that a packet has been sent. It is used when BTA manual mode is used. 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
1	ECC_CORRECTION_IRQ_EN	Virtual channel - ECC has been used to correct the only 1-bit error (short and long packet). 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0
0	CS_IRQ_EN	Virtual channel - checksum of the payload mismatch detection 0x0: Event is masked 0x1: Event generates an interrupt when it occurs	RW	0

**Table 10-593. Register Call Summary for Register DSI\_VC\_IRQENABLE\_i**

MIPI Display Serial Interface

- [Video Port Interface: \[0\]](#)
- [Bus Turnaround: \[1\]](#)
- [Interrupts: \[2\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[3\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[4\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Summary: \[5\] \[6\]](#)

### 10.3.6.3 DSI\_PHY Registers

#### 10.3.6.3.1 DSI\_PHY Register Summary

**Table 10-594. DSI1\_PHY Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSI1_PHY L3 Physical Address	DSI1_PHY L4_PER Physical Address
<a href="#">DSI_PHY_REGIST ER0</a>	RW	32	0x0000 0000	0x5800 4200	0x4804 4200
<a href="#">DSI_PHY_REGIST ER1</a>	RW	32	0x0000 0004	0x5800 4204	0x4804 4204
<a href="#">DSI_PHY_REGIST ER2</a>	RW	32	0x0000 0008	0x5800 4208	0x4804 4208
<a href="#">DSI_PHY_REGIST ER3</a>	RW	32	0x0000 000C	0x5800 420C	0x4804 420C
<a href="#">DSI_PHY_REGIST ER4</a>	RW	32	0x0000 0010	0x5800 4210	0x4804 4210
<a href="#">DSI_PHY_REGIST ER5</a>	W	32	0x0000 0014	0x5800 4214	0x4804 4214

**Table 10-595. DSI2\_PHY Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSI2_PHY L3 Physical Address	DSI2_PHY L4_PER Physical Address
<a href="#">DSI_PHY_REGIST ER0</a>	RW	32	0x0000 0000	0x5800 5200	0x4804 5200
<a href="#">DSI_PHY_REGIST ER1</a>	RW	32	0x0000 0004	0x5800 5204	0x4804 5204
<a href="#">DSI_PHY_REGIST ER2</a>	RW	32	0x0000 0008	0x5800 5208	0x4804 5208

**Table 10-595. DSI2\_PHY Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DSI2_PHY L3 Physical Address	DSI2_PHY L4_PER Physical Address
DSI_PHY_REGIST ER3	RW	32	0x0000 000C	0x5800 520C	0x4804 520C
DSI_PHY_REGIST ER4	RW	32	0x0000 0010	0x5800 5210	0x4804 5210
DSI_PHY_REGIST ER5	W	32	0x0000 0014	0x5800 5214	0x4804 5214

**10.3.6.3.2 DSI\_PHY Register Description**

**NOTE:** Copyright 2005-2008 MIPI Alliance, Inc. All rights reserved. MIPI Alliance Member Confidential.

**Table 10-596. DSI\_PHY\_REGISTER0**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 10-594</a> and <a href="#">Table 10-595</a>	<b>Instance</b>	DSI1_PHY DSI2_PHY
<b>Description</b>	Configuration register for HS mode timings		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_THSPREPARE								REG_THSPRPR_THSZERO								REG_THSTRAIL								REG_THSEXIT							

Bits	Field Name	Description	Type	Reset
31:24	REG_THSPREPARE	REG_THSPREPARE timing parameter in multiples of DDR clock period. DDR clock = CLKIN4DDR/4.  D-PHY specification: 40 ns + 4 * UI ÷ 85 ns + 6 * UI. UI = Unit Interval, equal to the duration of any HS state on the clock lane.  Actual value seen on line: = REG_THSPREPARE timer + analog delay and slew on signals = REG_THSPREPARE * DDR_Clock_Period + (-26.5 ns --- +4 ns)  PROGRAMMED VALUE = ceil( 70 ns / DDR Clock Period ) + 2. Default value is programmed for 400 MHz.	RW	0x1E
23:16	REG_THSPRPR_THSZERO	REG_THSPREPARE_THSZERO timing parameter in multiples of DDR clock period. DDR clock = CLKIN4DDR/4.  D-PHY specification: > 145 ns + 10 * UI.  Actual value seen on line: N = REG_THSPREPARE_THSZERO -REG_THSPREPARE  M = REG_THSPREPARE = {ceil[(N + 3)/4] * 4 + ceil(M/4) * 4 + 3} * DDR_Clock_Period + (~ - 29 ns --- 0 ns)  PROGRAMMED VALUE = ceil( 175 ns/ DDR Clock Period ) + 2. Default value is programmed for 400 MHz.	RW	0x48
15:8	REG_THSTRAIL	REG_THSTRAIL timing parameter in multiples of DDR clock period. DDR clock = CLKIN4DDR/4.  D-PHY specification: > 60 ns + 4 * UI.	RW	0x1D

Bits	Field Name	Description	Type	Reset
		Actual value seen on line: $N = \text{REG\_THSTRAIL} = \{\text{ceil}[(N+3)/4] * 4 - 2.75\} * \text{DDR\_Clock\_Period} + (\sim 0 \text{ ns} \text{ --- } 5 \text{ ns}).$ PROGRAMMED VALUE = $\text{ceil}(60 \text{ ns} / \text{DDR Clock Period}) + 5.$ Default value is programmed for 400 MHz.		
7:0	REG_THSEXIT	REG_THSEXIT timing parameter in multiples of DDR clock frequency. DDR clock = CLKIN4DDR/4. D-PHY specification: > 100 ns. Actual value seen on line: $N = \text{REG\_THSEXIT} = \text{THSEXIT timer} + \text{analog delay and slew on LP signals} = \{\text{ceil}(N/4)*4\} * \text{DDR\_Clock\_Period} - (\sim 3 \text{ ns} \text{ --- } 45 \text{ ns}).$ PROGRAMMED VALUE = $\text{ceil}(145 \text{ ns} / \text{DDR Clock Period}).$ Default value is programmed for 400 MHz.	RW	0x3A

**Table 10-597. Register Call Summary for Register DSI\_PHY\_REGISTER0**

MIPI Display Serial Interface

- [Clock Requirements: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [SCP Interface: \[5\]](#)
- [Display Timing Configuration: \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[11\] \[12\] \[13\] \[14\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[15\] \[16\] \[17\] \[18\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Description: \[19\] \[20\]](#)
- [DSI\\_PHY Register Summary: \[21\] \[22\]](#)

**Table 10-598. DSI\_PHY\_REGISTER1**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	DSI1_PHY DSI2_PHY
<b>Physical Address</b>	See <a href="#">Table 10-594</a> and <a href="#">Table 10-595</a>		
<b>Description</b>	Configuration register for LP mode and HS mode timings		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_TTAGO			REG_TTASURE			REG_TTAGET			RESERVED			REG_TLPXBY2				REG_TCLKTRAIL						REG_TCLKZERO									

Bits	Field Name	Description	Type	Reset
31:29	REG_TTAGO	TTA-GO timing in terms of number of TXCLKESC clocks. 0x0: 2 cycles 0x1: 3 cycles 0x2: 4 cycles 0x3: 5 cycles 0x4: 6 cycles 0x5: 7 cycles 0x6: 8 cycles 0x7: 9 cycles Default value: 4 cycles	RW	0x2
28:27	REG_TTASURE	TTA-SURE timing in terms of number of TXCLKESC clocks.	RW	0x0

Bits	Field Name	Description	Type	Reset
		0x0: 2 cycles 0x1: 1 cycles 0x2: 3 cycles 0x3: 4 cycles Default value: 2 cycles		
26:24	REG_TTAGET	TTA-GET timing in terms of number of TXCLKESC clocks. 0x0: 3 cycles 0x1: 4 cycles 0x2: 5 cycles 0x3: 6 cycles 0x4: 7 cycles 0x5: 8 cycles 0x6: 9 cycles 0x7: 10 cycles Default value: 5 cycles	RW	0x2
23:21	RESERVED	Reserved	R	0x0
20:16	REG_TLPXBY2	(TLPX)/2 timing parameter in multiples of DDR clock frequency. DDR clock = CLKIN4DDR/4. PROGRAMMED VALUE = ceil (25 ns / DDR Clock Period ) Actual value seen on line: $N = \text{REG\_TLPXBY2} = \text{ceil} (2 * N/4) * 4 * \text{DDR\_Clock\_period}.$ Default value is programmed for 400 MHz. This is the internal timer value. The value seen on line will have variance due to rise/fall mismatch effects. Note: TLPX is used to define the length of LP-01 state in HS start of Transmission sequences on clock and data lanes. For all other purposes TLPX is defined by the period of TXLPESC	RW	0x0A
15:8	REG_TCLKTRAIL	REG_TCLKTRAIL timing parameter in multiples of DDR clock frequency. DDR clock = CLKIN4DDR/4. D-PHY specification: > 60 ns. Actual value seen on line: $N = \text{REG\_TCLKTRAIL} = \{ \text{ceil} [(N + 3)/4] * 4 - 1.5 \} * \text{DDR\_Clock\_Period} + (\sim 0 \text{ ns} \text{ --- } 5 \text{ ns}).$ PROGRAMMED VALUE = ceil (60 ns / DDR Clock Period ) + 2. Default value is programmed for 400 MHz	RW	0x1A
7:0	REG_TCLKZERO	REG_TCLKZERO timing parameter in multiples of DDR clock period. DDR clock = CLKIN4DDR/4. D-PHY specification: (REG_TCLKPREPARE + REG_TCLKZERO) > 300 ns. Derived specification for REG_TCLKZERO (Min REG_TCLKPREPARE = 38 ns): REG_TCLKZERO > 262 ns. Actual value seen on line: $N = \text{REG\_TCLKZERO}$ $M = \text{REG\_TCLKPREPARE} = \{ \text{ceil} [(N + 3)/4] * 4 + \text{ceil}(M/4) * 4 - M + 2 \} * \text{DDR\_Clock\_Period} + (\sim 0 \text{ ns} \text{ --- } +5 \text{ ns}).$ PROGRAMMED VALUE = ceil (265 ns / DDR Clock Period) Default value is programmed for 400 MHz	RW	0x6A



**Table 10-599. Register Call Summary for Register DSI\_PHY\_REGISTER1**

MIPI Display Serial Interface

- [Clock Requirements: \[0\] \[1\] \[2\] \[3\]](#)
- [Display Timing Configuration: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[10\] \[11\] \[12\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[13\] \[14\] \[15\] \[16\]](#)
- [DSI\\_PHY Register Summary: \[17\] \[18\]](#)

**Table 10-600. DSI\_PHY\_REGISTER2**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	DSI1_PHY DSI2_PHY
<b>Physical Address</b>	See <a href="#">Table 10-594</a> and <a href="#">Table 10-595</a>		
<b>Description</b>	Sync pattern		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSSYNCPATTERN								RESERVED								REG_TCLKPREPARE															

Bits	Field Name	Description	Type	Reset
31:24	HSSYNCPATTERN	Default : 184 (10111000). MSB (last received bit of sync pattern), LSB (first received bit of sync pattern).	RW	0xB8
23:8	RESERVED	Reserved. Read returns zero. Write only zero for future compatibility.	R	0x0
7:0	REG_TCLKPREPARE	TCLK-PREPARE timing parameter in multiples of DDR clock period.  D-PHY specification: 38 ns ÷ 95 ns.  Actual value seen on line: = REG_TCLKPREPARE timer + analog delay and slew on LP signals = REG_TCLKPREPARE * DDR_Clock_Period + (~ -25 ns --- + 5 ns)  PROGRAMMED VALUE = ceil (65 ns / DDR Clock Period).  Default value is programmed for 400 MHz.	RW	0x1A

**Table 10-601. Register Call Summary for Register DSI\_PHY\_REGISTER2**

MIPI Display Serial Interface

- [Clock Requirements: \[0\]](#)
- [SCP Interface: \[1\]](#)
- [Display Timing Configuration: \[2\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[3\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[4\] \[5\]](#)
- [DSI\\_PROTOCOL\\_ENGINE Register Description: \[6\] \[7\]](#)
- [DSI\\_PHY Register Summary: \[8\] \[9\]](#)

**Table 10-602. DSI\_PHY\_REGISTER3**

<b>Address Offset</b>	0x0000 000C	
<b>Physical Address</b>	See <a href="#">Table 10-594</a> and <a href="#">Table 10-595</a>	<b>Instance</b> DSI1_PHY DSI2_PHY
<b>Description</b>	Transmitted pattern	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_TXTRIGGERESC3								REG_TXTRIGGERESC2								REG_TXTRIGGERESC1								REG_TXTRIGGERESC0							

Bits	Field Name	Description	Type	Reset
31:24	REG_TXTRIGGERESC3	Transmitted pattern when REG_TXTRIGGERESC3 is asserted (first bit transmitted to last bit transmitted). Default: 01100010	RW	0x62
23:16	REG_TXTRIGGERESC2	Default: 01011101	RW	0x5D
15:8	REG_TXTRIGGERESC1	Default: 00100001	RW	0x21
7:0	REG_TXTRIGGERESC0	Default: 10100000	RW	0xA0

**Table 10-603. Register Call Summary for Register DSI\_PHY\_REGISTER3**

MIPI Display Serial Interface

- [PHY Triggers: \[0\] \[1\]](#)
- [DSI\\_PHY Register Summary: \[2\] \[3\]](#)

**Table 10-604. DSI\_PHY\_REGISTER4**

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Table 10-594</a> and <a href="#">Table 10-595</a>	<b>Instance</b> DSI1_PHY DSI2_PHY
<b>Description</b>	Received pattern	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_RXTRIGGERESC3								REG_RXTRIGGERESC2								REG_RXTRIGGERESC1								REG_RXTRIGGERESC0							

Bits	Field Name	Description	Type	Reset
31:24	REG_RXTRIGGERESC3	Received pattern for which REG_RXTRIGGERESC3 is asserted (first bit transmitted to last bit transmitted). Default: 01100010	RW	0x62
23:16	REG_RXTRIGGERESC2	Default: 01011101	RW	0x5D
15:8	REG_RXTRIGGERESC1	Default: 00100001	RW	0x21
7:0	REG_RXTRIGGERESC0	Default: 10100000	RW	0xA0

**Table 10-605. Register Call Summary for Register DSI\_PHY\_REGISTER4**

MIPI Display Serial Interface

- [PHY Triggers: \[0\] \[1\] \[2\]](#)
- [DSI\\_PHY Register Summary: \[3\] \[4\]](#)

**Table 10-606. DSI\_PHY\_REGISTER5**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 10-594</a> and <a href="#">Table 10-595</a>	<b>Instance</b>	DSI1_PHY DSI2_PHY
<b>Description</b>	Reset done bits		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESETDONETXBYTECLK	RESETDONESCPCCLK	RESETDONEPWRCLK	RESETDONETXCLKESC4	RESETDONETXCLKESC3	RESETDONETXCLKESC2	RESETDONETXCLKESC1	RESETDONETXCLKESC0	RESERVED																							

Bits	Field Name	Description	Type	Reset
31	RESETDONETXBYTECLK	RESETDONETXBYTECLK 0x0: No reset 0x1: Reset done for the TXBYTECLK domain	R	0
30	RESETDONESCPCCLK	RESETDONESCPCCLK 0x0: No reset 0x1: Reset done for the SCP clock domain	R	0
29	RESETDONEPWRCLK	RESETDONEPWRCLK 0x0: No reset 0x1: Reset done for the PWR clock domain	R	0
28	RESETDONETXCLKESC4	RESETDONETXCLKESC4 0x0: No reset 0x1: Reset done for the TXCLKESC domain for lane 4	R	0
27	RESETDONETXCLKESC3	RESETDONETXCLKESC3 0x0: No reset 0x1: Reset done for the TXCLKESC domain for lane 3	R	0
26	RESETDONETXCLKESC2	RESETDONETXCLKESC2 0x0: No reset 0x1: Reset done for the TXCLKESC domain for lane 2	R	0
25	RESETDONETXCLKESC1	RESETDONETXCLKESC1 0x0: No reset 0x1: Reset done for the TXCLKESC domain for lane 1	R	0
24	RESETDONETXCLKESC0	RESETDONETXCLKESC0 0x0: No reset 0x1: Reset done for the TXCLKESC domain for lane 0	R	0
23:0	RESERVED	Read-Only register. Read returns zero.	R	0x000000

**Table 10-607. Register Call Summary for Register DSI\_PHY\_REGISTER5**

MIPI Display Serial Interface

- [Reset-Done Bits: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [DSI\\_PHY Register Summary: \[7\] \[8\]](#)

### 10.3.6.4 DSI\_PLLCTRL Registers

#### 10.3.6.4.1 DSI\_PLLCTRL Register Summary

**Table 10-608. DSI1\_PLLCTRL Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSI1_PLLCTRL L3 Physical Address	DSI1_PLLCTRL L4_PER Physical Address
DSI_PLL_CONTROL	RW	32	0x0000 0000	0x5800 4300	0x4804 4300
DSI_PLL_STATUS	R	32	0x0000 0004	0x5800 4304	0x4804 4304
DSI_PLL_GO	RW	32	0x0000 0008	0x5800 4308	0x4804 4308
DSI_PLL_CONFIGURATION1	RW	32	0x0000 000C	0x5800 430C	0x4804 430C
DSI_PLL_CONFIGURATION2	RW	32	0x0000 0010	0x5800 4310	0x4804 4310
DSI_PLL_CONFIGURATION3	RW	32	0x0000 0014	0x5800 4314	0x4804 4314
DSI_PLL_SSC_CONFIGURATION1	RW	32	0x0000 0018	0x5800 4318	0x4804 4318
DSI_PLL_SSC_CONFIGURATION2	RW	32	0x0000 001C	0x5800 431C	0x4804 431C
DSI_PLL_CONFIGURATION4	RW	32	0x0000 0020	0x5800 4320	0x4804 4320

**Table 10-609. DSI2\_PLLCTRL Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSI2_PLLCTRL L3 Physical Address	DSI2_PLLCTRL L4_PER Physical Address
DSI_PLL_CONTROL	RW	32	0x0000 0000	0x5800 5300	0x4804 5300
DSI_PLL_STATUS	R	32	0x0000 0004	0x5800 5304	0x4804 5304
DSI_PLL_GO	RW	32	0x0000 0008	0x5800 5308	0x4804 5308
DSI_PLL_CONFIGURATION1	RW	32	0x0000 000C	0x5800 530C	0x4804 530C
DSI_PLL_CONFIGURATION2	RW	32	0x0000 0010	0x5800 5310	0x4804 5310
DSI_PLL_CONFIGURATION3	RW	32	0x0000 0014	0x5800 5314	0x4804 5314
DSI_PLL_SSC_CONFIGURATION1	RW	32	0x0000 0018	0x5800 5318	0x4804 5318
DSI_PLL_SSC_CONFIGURATION2	RW	32	0x0000 001C	0x5800 531C	0x4804 531C
DSI_PLL_CONFIGURATION4	RW	32	0x0000 0020	0x5800 5320	0x4804 5320

**10.3.6.4.2 DSI\_PLLCTRL Register Description**

**Table 10-610. DSI\_PLL\_CONTROL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 10-608</a> and <a href="#">Table 10-609</a>	<b>Instance</b>	DSI1_PLLCTRL DSI2_PLLCTRL
<b>Description</b>	This register controls the PLL reset/power and modes		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							HSDIV_SYSRESET	PLL_SYSRESET	PLL_HALTMODE	PLL_GATEMODE	PLL_AUTOMODE				

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reads as zero.	R	0x00000000
4	HSDIV_SYSRESET	Force HSDIVIDER SYSRESET 0x0: HSDIVIDER SYSRESET controlled by power FSM 0x1: HSDIVIDER SYSRESET forced active	RW	0
3	PLL_SYSRESET	Force ADPLL SYSRESET 0x0: PLL SYSRESET controlled by power FSM 0x1: PLL SYSRESET forced active	RW	0
2	PLL_HALTMODE	Allow PLL to be halted if no activity. Reserved when DSI1_PLLCTRL_AUTO is 0. 0x0: PLL will not be halted 0x1: PLL will be halted based on activity	RW	0
1	PLL_GATEMODE	Allow PLL clock gating for power saving Reserved when DSI1_PLLCTRL_AUTO is 0. 0x0: CLKIN4DDR clock on 0x1: CLKIN4DDR clock gated by DSI Protocol Engine activity	RW	0
0	PLL_AUTOMODE	Automatic update mode. If this bit is set then the configuration updates will be synchronised to DISPCUpdateSync. If this bit is clear configuration updates will be done immediately. Reserved when DSI1_PLLCTRL_AUTO is 0. 0x0: Manual mode 0x1: Automatic mode	RW	0

**Table 10-611. Register Call Summary for Register DSI\_PLL\_CONTROL**

MIPI Display Serial Interface

- [DSI PLL Clock-Gating Sequence: \[0\]](#)
- [DSI PLL Go Sequence: \[1\] \[2\] \[3\] \[4\]](#)
- [DSI PLL Recommended Values: \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[10\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[11\]](#)
- [DSI\\_PLLCTRL Register Summary: \[12\] \[13\]](#)

**Table 10-612. DSI\_PLL\_STATUS**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 10-608</a> and <a href="#">Table 10-609</a>	<b>Instance</b>	DSI1_PLLCTRL DSI2_PLLCTRL
<b>Description</b>	This register contains the status information		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								SSC_EN_ACK	M7_CLOCK_ACK	RESERVED	BYPASSACKZ	M5_CLOCK_ACK	M4_CLOCK_ACK	PLL_BYPASS	PLL_HIGHJITTER	PLL_LIMP	PLL_LOSSREF	PLL_RECAL	PLL_LOCK	DSI1_PLLCTRL_RESET_DONE			

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reads as zero.	R	0x00000
12	SSC_EN_ACK	Spread Spectrum Clocking acknowledge Read 0x0: Spread Spectrum Clocking inactive Read 0x1: Spread Spectrum Clocking active	R	0
11	M7_CLOCK_ACK	Acknowledge for enable of clock Verify the status before selecting this source in the clock mux Read 0x0: M7 clock inactive Read 0x1: M7 clock active	R	0
10	RESERVED	Reserved	R	0
9	BYPASSACKZ	State of bypass mode on PHY and HSDIVIDER Read 0x0: DSI_PHY and HSDIVIDER have switched to using the bypass clocks. Read 0x1: PLL outputs are still being used by DSI_PHY or HSDIVIDER	R	0
8	M5_CLOCK_ACK	Acknowledge for enable of DSI Protcol Engine clock Verify the status before selecting this source in the DSI Protcol Engine clock mux Read 0x0: M5 Protocol Engine clock inactive Read 0x1: M5 Protocol Engine clock active	R	0
7	M4_CLOCK_ACK	Acknowledge for enable of clock Verify the status before selecting this source in the clock mux Read 0x0: M4 clock inactive Read 0x1: M4 clock active	R	0
6	PLL_BYPASS	DSI PLL Bypass status Read 0x0: PLL not bypassing Read 0x1: PLL bypass	R	0
5	PLL_HIGHJITTER	PLL High Jitter status Read 0x0: PLL in normal jitter condition Read 0x1: PLL in high jitter condition: Phase error > 24%	R	0

Bits	Field Name	Description	Type	Reset
4	PLL_LIMP	PLL Limp status Read 0x0: LIMP mode inactive Read 0x1: LIMP mode active	R	0
3	PLL_LOSSREF	PLL Reference Loss status Read 0x0: Reference input active Read 0x1: Reference input inactive	R	0
2	PLL_RECAL	PLL recalibration status If this bit is active, the PLL needs to be recalibrated Read 0x0: Recalibration is not required Read 0x1: Recalibration is required	R	0
1	PLL_LOCK	PLL Lock status See the programming guide for the use of this bit Read 0x0: PLL is not locked Read 0x1: PLL is locked	R	0
0	DSI1_PLLCTRL_RESET_DONE	DSI1_PLLCTRL reset done status Read 0x0: Reset is in progress Read 0x1: Reset has completed	R	0

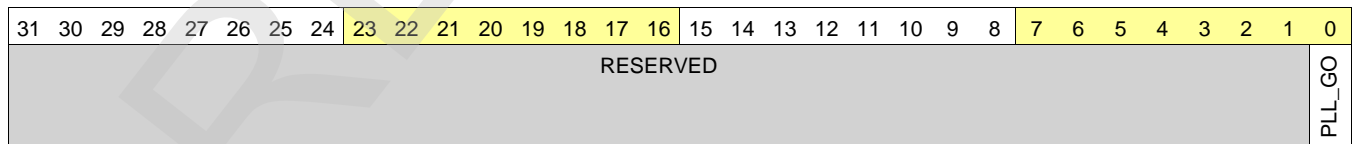
**Table 10-613. Register Call Summary for Register DSI\_PLL\_STATUS**

MIPI Display Serial Interface

- [DSI PLL Controllers Architecture: \[0\]](#)
- [Error Handling: \[1\] \[2\] \[3\] \[4\]](#)
- [Software Reset: \[5\]](#)
- [DSI PLL Clock-Gating Sequence: \[6\]](#)
- [DSI PLL Clock Sequence: \[7\]](#)
- [DSI PLL Go Sequence: \[8\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[9\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[10\]](#)
- [DSI\\_PLLCTRL Register Summary: \[11\] \[12\]](#)

**Table 10-614. DSI\_PLL\_GO**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	DSI1_PLLCTRL DSI2_PLLCTRL
<b>Physical Address</b>	See <a href="#">Table 10-608</a> and <a href="#">Table 10-609</a>		
<b>Description</b>	This register contains the GO bit		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved. Write only zero for future compatibility. Reads return zero.	R	0x0000 0000
0	PLL_GO	Request (re-)locking sequence of the PLL. If the AutoMode bit is set, then this will be deferred until DISPCUpdate Sync goes active 0x0: No pending action 0x1: Request PLL (re-)locking/locking pending	RW	0

**Table 10-615. Register Call Summary for Register DSI\_PLL\_GO**

MIPI Display Serial Interface

- [DSI PLL Clock-Gating Sequence: \[0\] \[1\]](#)
- [DSI PLL Clock Sequence: \[2\]](#)
- [DSI PLL Go Sequence: \[3\] \[4\]](#)
- [DSI PLL Recommended Values: \[5\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[6\] \[7\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[8\] \[9\]](#)
- [DSI\\_PLLCTRL Register Summary: \[10\] \[11\]](#)

**Table 10-616. DSI\_PLL\_CONFIGURATION1**

<b>Address Offset</b>	0x0000 000C		
<b>Physical Address</b>	See <a href="#">Table 10-608</a> and <a href="#">Table 10-609</a>	<b>Instance</b>	DSI1_PLLCTRL DSI2_PLLCTRL
<b>Description</b>	This register contains the latched PLL and HSDIVDER configuration bits		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	M5_CLOCK_DIV			M4_CLOCK_DIV			PLL_REGM									PLL_REGN					PLL_STOPMODE										

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30:26	M5_CLOCK_DIV	Divider value for Protocol Engine clock source M5REG	RW	0x00
25:21	M4_CLOCK_DIV	Divider value for clock source M4REG	RW	0x00
20:9	PLL_REGM	M Divider for PLL	RW	0x000
8:1	PLL_REGN	N Divider for PLL (Reference)	RW	0x00
0	PLL_STOPMODE	PLL STOPMODE 0x0: STOPMODE is not selected 0x1: STOPMODE is selected	RW	0

**Table 10-617. Register Call Summary for Register DSI\_PLL\_CONFIGURATION1**

MIPI Display Serial Interface

- [DSI PLL Controllers Architecture: \[0\] \[1\]](#)
- [DSI PLL Operations: \[2\] \[3\] \[4\]](#)
- [DSI PLL Clock Sequence: \[5\] \[6\] \[7\] \[8\]](#)
- [DSI PLL Recommended Values: \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [DSI\\_PLLCTRL Register Summary: \[24\] \[25\]](#)



**Table 10-618. DSI\_PLL\_CONFIGURATION2**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 10-608</a> and <a href="#">Table 10-609</a>	<b>Instance</b>	DSI1_PLLCTRL DSI2_PLLCTRL
<b>Description</b>	This register contains the unlatched PLL and HSDIVDER configuration bits These bits are "shadowed" when automatic mode is selected		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RESERVED				M7_CLOCK_PWDN	M7_CLOCK_EN	RESERVED	RESERVED	RESERVED	HSDIVBYPASS	M5_CLOCK_PWDN	M5_CLOCK_EN	M4_CLOCK_PWDN	M4_CLOCK_EN	BYPASSEN	PHY_CLKINEN	PLL_REFEN	RESERVED	RESERVED	PLL_LOCKSEL	PLL_DRIFTGUARDEN	RESERVED	PLL_LOWCURRSTBY	PLL_PLLLPMODE	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	PLL_IDLE

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Read as zero.	R	0x00
26	M7_CLOCK_PWDN	Power down for M7 clock source 0x0: M7 clock divider is active 0x1: M7 clock divider is powered-down	RW	0
25	M7_CLOCK_EN	Enable for M7 clock source 0x0: M7 clock divider is disabled 0x1: M7 clock divider is enabled	RW	0
24	RESERVED	Reserved	R	0
23	RESERVED	Reserved	R	0
22:21	RESERVED	Reserved	R	0x0
20	HSDIVBYPASS	Forces HSDIVIDER to bypass mode 0x0: HSDIVIDER in normal operation. Bypass controlled by PLL. 0x1: HSDIVIDER forced to bypass mode.	RW	0
19	M5_CLOCK_PWDN	Power down for Protocol Engine clock source 0x0: DSI Protocol Engine clock divider is active 0x1: DSI Protocol Engine clock divider is powered-down	RW	0
18	M5_CLOCK_EN	Enable for Protocol Engine clock source 0x0: DSI Protocol Engine clock divider is disabled 0x1: DSI Protocol Engine clock divider is enabled	RW	0
17	M4_CLOCK_PWDN	Power down for M4 clock source 0x0: clock divider is active 0x1: clock divider is powered-down	RW	0
16	M4_CLOCK_EN	Enable for M4 clock source 0x0: clock divider is disabled 0x1: clock divider is enabled	RW	0
15	BYPASSEN	Selects functional clock asCLKIN4DDR clock source 0x0: PLL controls CLKIN4DDR clock source: PLL DCO if PLL is locked functional clock if not locked 0x1: Force functional clock to be used as CLKIN4DDR clock source	RW	0

Bits	Field Name	Description	Type	Reset
14	PHY_CLKINEN	CLKIN4DDR clock control 0x0: CLKIN4DDR clock is disabled 0x1: CLKIN4DDR clock is enabled	RW	0
13	PLL_REFEN	PLL reference clock control 0x0: PLL reference clock disabled 0x1: PLL reference clock enabled	RW	0
12	RESERVED	Reserved. Keep at reset value.	R	0
11	RESERVED	Reserved. Keep at reset value.	R	0
10:9	PLL_LOCKSEL	Selects the lock criteria for the PLL 0x0: Phase Lock 0x1: Frequency Lock 0x2: Spare	RW	0x0
8	PLL_DRIFTGUARDEN	PLL DRIFTGUARDEN 0x0: Only RECAL flag is asserted in case of temperature drift. The programmer should take appropriate action. 0x1: Temperature drift will initiate automatic recalibration. RECAL flag will be asserted while this is taking place.	RW	0
7	RESERVED		R	0
6	PLL_LOWCURRSTBY	PLL LOW CURRENT STANDBY 0x0: LOWCURRSTBY is not selected 0x1: LOWCURRSTBY is selected	RW	0
5	PLL_PLLLPMode	Select the power / performance of the PLL 0x0: Full performance, minimised jitter 0x1: Reduced power, increased jitter	RW	0
4	RESERVED	Reads return zero.	R	0
3:1	RESERVED	Reserved	R	0
0	PLL_IDLE	PLL IDLE: 0x0: IDLE is not selected 0x1: IDLE is selected	RW	0

**Table 10-619. Register Call Summary for Register DSI\_PLL\_CONFIGURATION2**

MIPI Display Serial Interface

- [DSI Clock Configuration: \[0\]](#)
- [DSI PLL Controllers Architecture: \[1\]](#)
- [DSI PLL Operations: \[2\] \[3\]](#)
- [DSI PLL Clock-Gating Sequence: \[4\]](#)
- [DSI PLL Clock Sequence: \[5\]](#)
- [DSI PLL Go Sequence: \[6\] \[7\] \[8\] \[9\]](#)
- [DSI PLL Recommended Values: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\]](#)
- [Display Timing Configuration: \[23\]](#)
- [DSI Video Mode Using the DISPC Video Port: \[24\] \[25\] \[26\] \[27\]](#)
- [DSI Command Mode Using the DISPC Video Port: \[28\] \[29\] \[30\] \[31\]](#)
- [DSI\\_PLLCTRL Register Summary: \[32\] \[33\]](#)

**Table 10-620. DSI\_PLL\_CONFIGURATION3**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 10-608</a> and <a href="#">Table 10-609</a>	<b>Instance</b>	DSI1_PLLCTRL DSI2_PLLCTRL
<b>Description</b>	HSDIVIDER configuration bits for the M5 and M6 dividers		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								M7_CLOCK_DIV				RESERVED											

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:10	RESERVED	Reserved	R	0x0000
9:5	M7_CLOCK_DIV	Divider value for M7 divider.	RW	0x00
4:0	RESERVED	Reserved	R	0x00

**Table 10-621. Register Call Summary for Register DSI\_PLL\_CONFIGURATION3**

MIPI Display Serial Interface

- [DSI PLL Clock Sequence: \[0\]](#)
- [DSI\\_PLLCTRL Register Summary: \[1\] \[2\]](#)

**Table 10-622. DSI\_PLL\_SSC\_CONFIGURATION1**

<b>Address Offset</b>	0x0000 0018		
<b>Physical Address</b>	See <a href="#">Table 10-608</a> and <a href="#">Table 10-609</a>	<b>Instance</b>	DSI1_PLLCTRL DSI2_PLLCTRL
<b>Description</b>	Configuration for PLL Spread Spectrum Clocking modulation		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DOWNSPREAD		RESERVED		EN_SSC											

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	RESERVED	R	0x0000 0000
2	DOWNSPREAD	Forces the clock spreading only in the down spectrum. 0x0: Clock spreading not forced. 0x1: Spectrum spreading only in down direction.	RW	0
1	RESERVED	Reserved. Reads return 0.	R	0
0	EN_SSC	Spread Spectrum Clocking enable 0x0: Spread Spectrum Clocking disabled 0x1: Spread Spectrum Clocking enabled	RW	0

**Table 10-623. Register Call Summary for Register DSI\_PLL\_SSC\_CONFIGURATION1**

MIPI Display Serial Interface

- [DSI PLL Controllers Architecture: \[0\] \[1\]](#)
- [DSI\\_PLLCTRL Register Summary: \[2\] \[3\]](#)

**Table 10-624. DSI\_PLL\_SSC\_CONFIGURATION2**

<b>Address Offset</b>	0x0000 001C		
<b>Physical Address</b>	See <a href="#">Table 10-608</a> and <a href="#">Table 10-609</a>	<b>Instance</b>	DSI1_PLLCTRL DSI2_PLLCTRL
<b>Description</b>	Configuration for PLL Spread Spectrum Clocking modulation		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								MODFREQDIVIDER									DELTAM														

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reads as zero	R	0x0
29:20	MODFREQDIVIDER	Modulation Frequency Divider control for SSC	RW	0x000
19:0	DELTAM	DeltaM control for SSC	RW	0x00000

**Table 10-625. Register Call Summary for Register DSI\_PLL\_SSC\_CONFIGURATION2**

MIPI Display Serial Interface

- [DSI PLL Controllers Architecture: \[0\] \[1\]](#)
- [DSI\\_PLLCTRL Register Summary: \[2\] \[3\]](#)

**Table 10-626. DSI\_PLL\_CONFIGURATION4**

<b>Address Offset</b>	0x0000 0020		
<b>Physical Address</b>	See <a href="#">Table 10-608</a> and <a href="#">Table 10-609</a>	<b>Instance</b>	DSI1_PLLCTRL DSI2_PLLCTRL
<b>Description</b>	Allows setting the fractional M divider for PLL.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														PLL_REGM_F																	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reads as zero.	R	0x0000
17:0	PLL_REGM_F	Fractional part of M divider.	RW	0x00000

**Table 10-627. Register Call Summary for Register DSI\_PLL\_CONFIGURATION4**

MIPI Display Serial Interface

- [DSI PLL Controllers Architecture: \[0\]](#)
- [DSI\\_PLLCTRL Register Summary: \[1\] \[2\]](#)

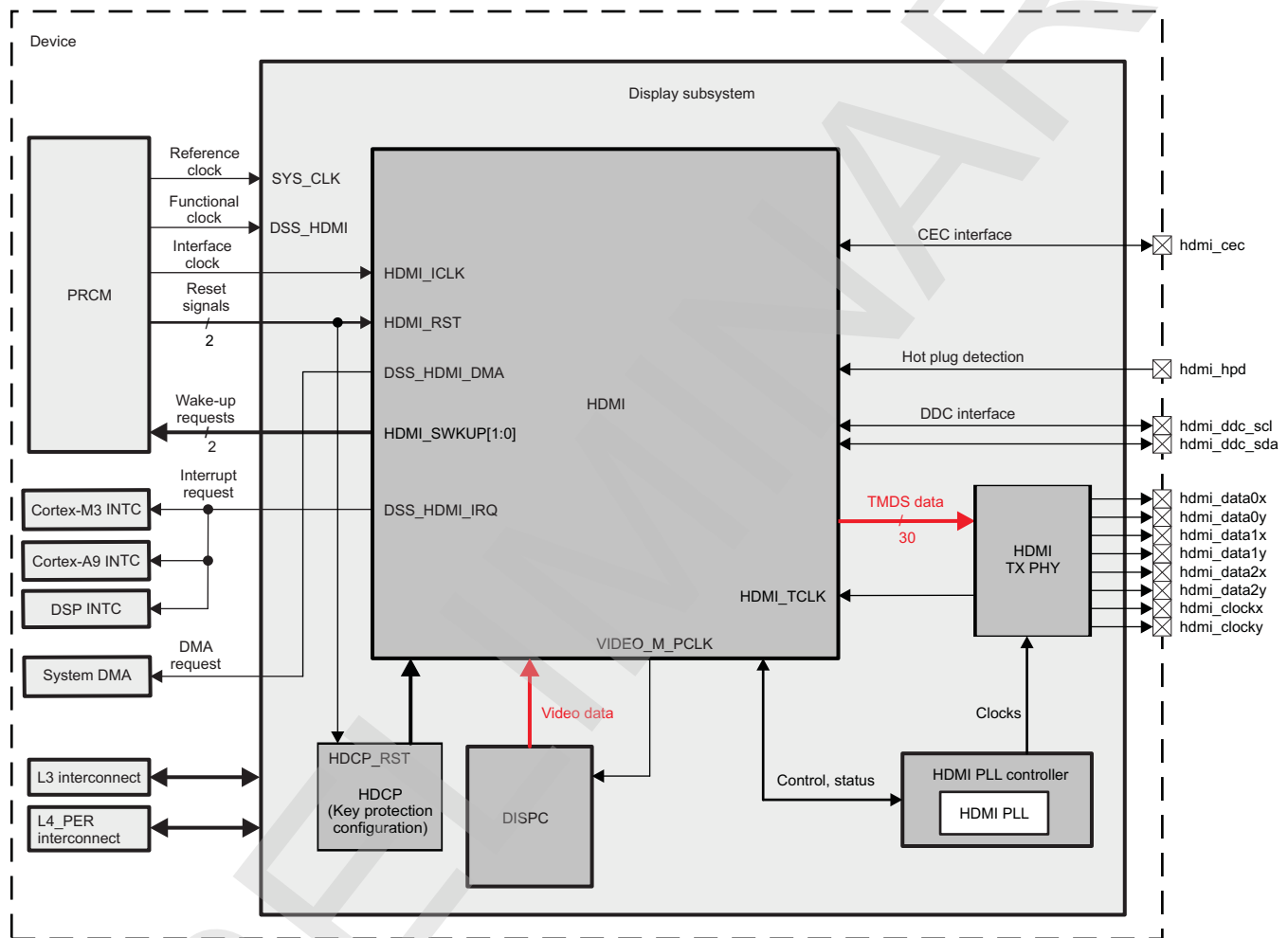
## 10.4 High-Definition Multimedia Interface

This section introduces the high-definition multimedia interface (HDMI) module. It also describes its main functions and connections in the device.

### 10.4.1 High-Definition Multimedia Interface (HDMI) Overview

Figure 10-159 shows an overview of the HDMI module.

Figure 10-159. HDMI Overview



hdmi-001

#### 10.4.1.1 HDMI Main Features

The HDMI module provides the following main features:

- HDMI 1.3, HDCP 1.2, and DVI 1.0 compliant
- EIA/CEA-861-D video format support (refer to [Table 10-628](#) for more details)
- VESA DMT video format support (refer to [Table 10-629](#) for more details)
- Support for deep-color mode:
  - 10-bit/component color depth up to 1080p @60Hz
  - 12-bit/component color depth up to 720p/1080i @60Hz
- Supports up to 148.5 MHz pixel clock (1920 x 1080p @60Hz)
- Video formats: 24-bit RGB
- Uncompressed multi-channel (up to 8 channels) audio (L-PCM) support

- Master I2C interface for Display Data Channel (DDC) connection
- Consumer Electronic Control (CEC) interface
- Integrated High-bandwidth Digital Content Protection (HDCP) encryption engine for transmitting protected audio and video content (authentication performed by software)
- Integrated Transition Minimized Differential Signaling (TMDS) and TER4 encoders for data island support
- Integrated TMDS PHY (3 TMDS differential data lanes + TMDS differential clock lane)
  - Up to 1,85625 Gbps per lane at (1080p @60Hz at 10 bits/component, lower resolutions at 12 bits/component)
  - 928,125Mbps per lane at (720p/1080i @60Hz 10 bits/component, lower resolutions at 12 bits/component)
  - 337,5 Mbps at (576p @50Hz/480p @60Hz 10 bits/component)

#### 10.4.1.2 HDMI Video Formats and Timings

##### 10.4.1.2.1 CEA-861-D video formats and timings

Table 10-628 presents the video timings supported by the HDMI module, according to the CEA-861-D standart:

**Table 10-628. HDMI Video Timings (CEA-861-D)**

Refresh Rate	Resolution
24 Hz (Low field rate)	1920 x 1080p
50 Hz	1920 x 1080p
	1920 x 1080i
	1280 x 720p
	2880 x 576p
	1440 x 576p
	1440 x 576i
59.94 Hz	720 x 576p
	1920 x 1080p
	2880 x 480p
	1440 x 480p
	1440 x 480i
	720 x 480p
60 Hz	640 x 480p
	1920 x 1080p
	1920 x 1080i
	2880 x 480p
	1280 x 720p
	1440 x 480p
	1440 x 480i
	720 x 480p
640 x 480p	

##### 10.4.1.2.2 VESA DMT video formats and timings

Table 10-629 presents the video timings supported by the HDMI module, according to the VESA DMT standart:

**Table 10-629. HDMI Video timings (VESA DMT)**

Refresh Rate	Resolution
60 Hz	640 x 480p
	800 x 600p
	848 x 480p
	1024 x 768p
	1280 x 768p
	1280 x 800p
	1280 x 960p
	1280 x 1024p
	1360 x 768p
	1366 x 768p
	1400 x 1050p
	1440 x 900p
	1680 x 1050p
60 Hz (Reduced Blanking)	1280 x 768p
	1280 x 800p
	1400 x 1050p
	1440 x 900p
1680 x 1050p	

## 10.5 Remote Frame Buffer Interface

This section describes the remote frame buffer interface (RFBI).

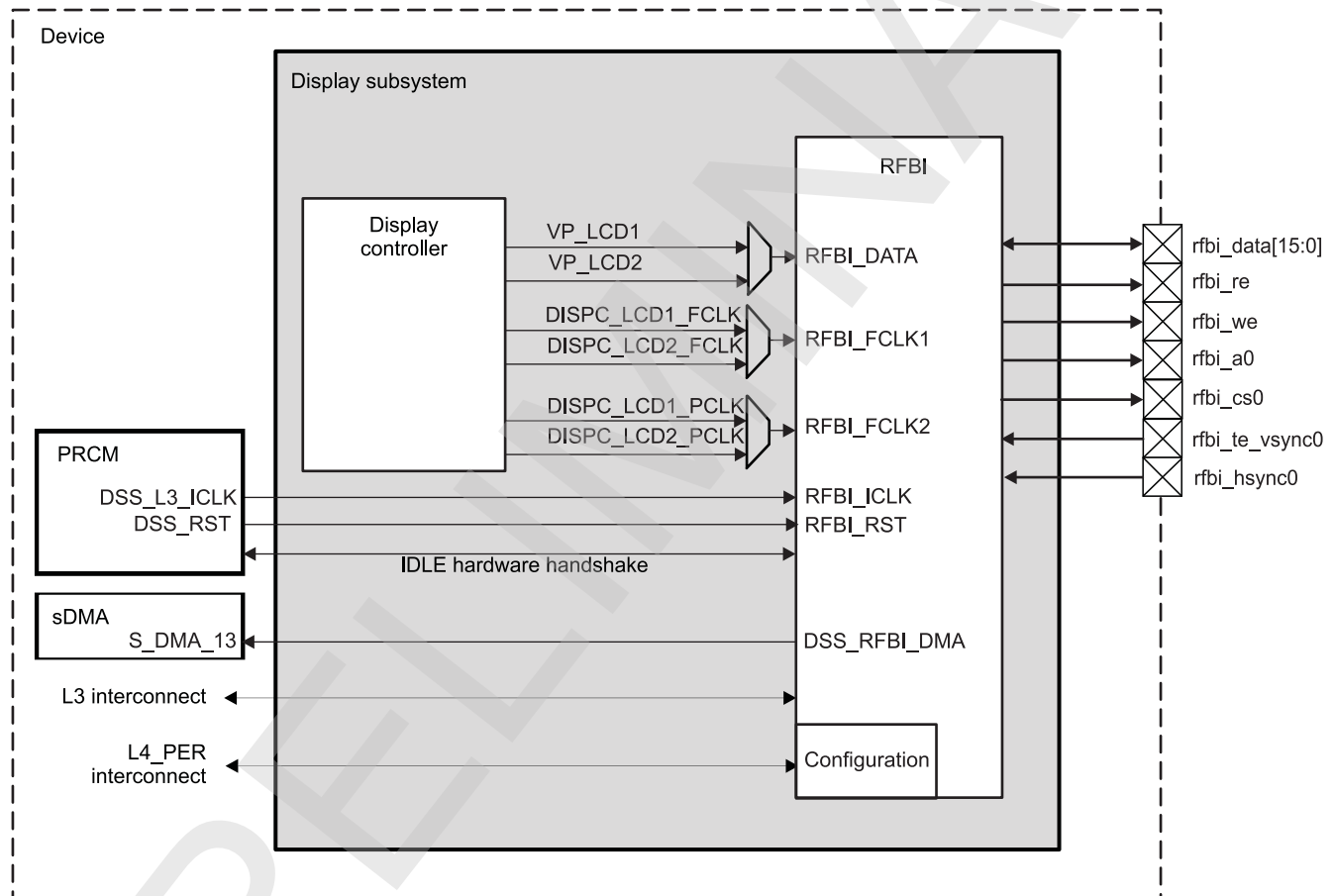
### 10.5.1 Remote Frame Buffer Interface Overview

The RFBI module is part of the display subsystem that provides the logic to display a picture from the memory frame buffer (SDRAM or SRAM) on a LCD panel.

The RFBI is connected to the L3 and L4\_PER interconnect through the display subsystem local interconnect. It is preferred from a performance point of view to use the L4\_PER interconnect for configuration and the L3 interconnect for data transfer.

Figure 10-160 shows an overview of the RFBI.

**Figure 10-160. RFBI Highlight**



rfbi-001

The RFBI supports the MIPI DBI protocol with the following features:

- Access to RFB direct MPU interface
  - Sends commands to the RFB panel through the slave port interconnect
  - Sends data, received from the DISPC or from the MPU through the DISPC pixel data bus, to the RFB panel
  - Reads data/status from the RFB to the slave port interconnect
- RFB interface
  - 8-/9-/16-bit parallel interface
  - One programmable configuration for one LCD connected to the RFBI
- Data formats



- Programmable pixel modes (12-/16-/18-/24-BPP modes in RGB format)
- Programmable output formats on one/multiple cycles per pixel (data from the DISPC and the L3 interconnect)
- FIFO
  - One slave port with DMA request and interconnect FIFO of 24 × 32-bit depth (for write access to the [RFBI\\_DATA](#) register only)
  - One video port FIFO of 8 × 24-bit depth receiving data from the DISPC through [RFBI\\_DATA](#)

## 10.5.2 RFBI Environment

In the display subsystem, data are read by the DISPC from the memory frame buffer (SDRAM, SRAM, IVA-HD SL2, etc.), and sent through the RFBI to the external LCD/RFB.

The DISPC associated with the RFBI implements the MIPI DBI protocol.

The DISPC provides the required control signals to interface the memory frame buffer (SDRAM, SRAM or IVAHD-SL2) directly to the external displays. The RFBI is connected to the memory through the L3 interconnect and has its own DMA (with embedded FIFOs) to read data from the system memory. There is one master port (L3 interconnect) and two slave ports (one for L4\_PER interconnect and the other for L3 interconnect).

The DISPC\_CONTROL1[16] GPOUT[1] and DISPC\_CONTROL1[15] GPOUT[0] bits control selection of the display subsystem modules (see [Table 10-630](#)).

**Table 10-630. I/O Pad Mode Selection**

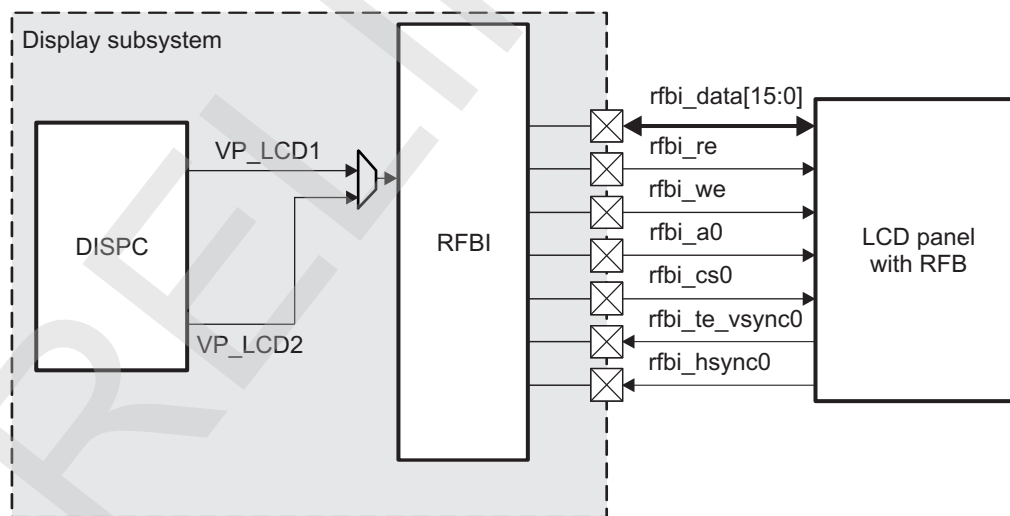
DISPC_CONTROL1[16] GPOUT1 Bit	DISPC_CONTROL1[15] GPOUT0 Bit	
	0	1
0	00 (reset)	01 (RFBI mode)
1	10 (invalid)	11 (invalid)

The RFB of the LCD panel is directly connected to the RFBI of the device. The RFBI controls the reads/writes from/to the RFB. The RFBI receives data from its slave port or output from the DISPC and generates the signals to control the LCD panel. Through the RFBI, the MPU can send commands or parameter/display data to the LCD panel and directly set the DISPC registers to read/write data from/to the memory in the LCD panel. The RFBI can manage only one LCD panel.

### 10.5.2.1 RFBI Parallel Interface (MIPI DBI Protocol)

[Figure 10-161](#) shows the LCD support parallel interface in RFBI mode.

**Figure 10-161. RFBI LCD Support Interface**



rfbi-002

[Table 10-631](#) describes the interface signals to/from the LCD panel.

**Table 10-631. LCD Interface Signals**

Signal Name	Type <sup>(1)</sup>	Description
rfbi_data[15:0]	I/O	RFBI I/O data
rfbi_re	O	Read access signal

<sup>(1)</sup> I = Input; O = Output; I/O = Input/Output

**Table 10-631. LCD Interface Signals (continued)**

Signal Name	Type <sup>(1)</sup>	Description
rfb_we	O	Write access signal
rfb_a0	O	Command/data selection signal
rfb_cs0	O	Chip-select (CS) signal for LCD 1
rfb_te_vsync0	I	Tearing effect (TE) synchronization signal (TE or VSYNC for LCD panel 1)
rfb_hsync0	I	HSYNC from LCD panel 1

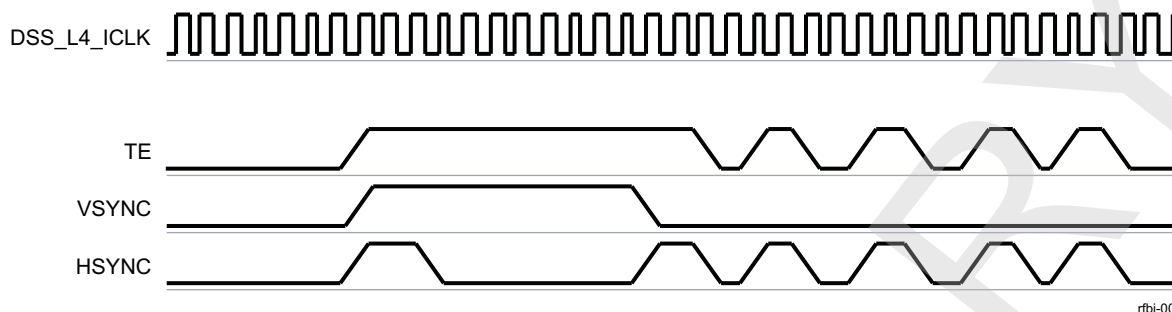
- **rfb\_data[15:0]:** The pixel data comprises the RFBI pixel data (bits 15:0). A write/read command must be sent to the LCD panel to send/read the data.  
Before any data access, the application must send commands and parameters when it is necessary to configure an LCD panel. The data is used as input in read operations during production test and also to read the status of the registers in the LCD panel and pixels from the embedded frame buffer in the LCD panel.
- **rfb\_re:** This is the read-enabled signal used to indicate when a read from the embedded memory in the LCD panel is ongoing. The RFBI registers describe the behavior of the read signal (off/on/cycle time). The polarity of the read-enabled signal is programmable. The read is used to get status/data information from the LCD panel.
- **rfb\_we:** The write-enabled signal is used to indicate when a write is ongoing. The RFBI registers allow flexible behavior of the write signal (programmable off/on/cycle times and signal polarity).
- **rfb\_a0:** The signal is asserted to indicate its status: Command or data. The polarity is programmable and the status of the signal depends on the RFBI registers written by the application (CMD/READ/STATUS/PARAM/PIXEL). The register in use by the hardware defines the status of rfb\_a0. The order of the writes/reads to the RFBI registers CMD/READ/STATUS/PARAM/PIXEL defines the transitions of A0.
- **rfb\_cs0:** The signal is the chip-select (CS0) asserted to indicate which LCD panel is selected and must be ready to receive/transmit commands and data. When RE or WE is on, CS0 must not be changed. To select the trigger mode, configure the [RFBI\\_CONFIG\[3:2\]](#) TRIGGERMODE bit field (0x0: Internal trigger mode with the [RFBI\\_CONTROL\[4\]](#) ITE bit, 0x1: External trigger mode with the TE signal rfb\_te\_vsync0, 0x2: External trigger mode with the rfb\_te\_vsync0, and rfb\_hsync0 signals with the programmable line counter).
- **rfb\_te\_vsync0** (only used by RFBI if [RFBI\\_CONFIG\[3:2\]](#) TRIGGERMODE = 0x1 or 0x2, otherwise ignored): Based on the trigger mode selected, the signal is the TE pulse signal or the LCD panel vertical synchronization (VSYNC) pulse signal. rfb\_te\_vsync0 is used by the TE logic as the synchronization signal to send the pixel to the LCD panel.
- **rfb\_hsync0** (only used by RFBI if [RFBI\\_CONFIG\[3:2\]](#) TRIGGERMODE = 0x2, otherwise ignored): The HSYNC pulse signals indicate to the RFBI when horizontal synchronization occurs. The polarity of the HSYNC signals is programmable. The minimum pulse width of the signal is two RFBI\_ICLK clock cycles. rfb\_hsync0 is used by the TE logic as a synchronization signal to send the pixel to the LCD panel.

### 10.5.2.1.1 Description of the Tearing Effect Pulse Signal

The TE signal can be generated externally by the RFBI depending on the value of the [RFBI\\_CONFIG\[3:2\]](#) TRIGGERMODE bit field:

- TRIGGERMODE = 0x1: external trigger mode with TE signal. TE signal corresponds to rfb\_te\_vsync0 pad.
- TRIGGERMODE = 0x2: external trigger mode with HSYNC/VSYNC signals. HSYNC signal corresponds to rfb\_hsync0 pad and VSYNC signal to rfb\_te\_vsync0 pad.

The externally generated TE synchronization signal is a logical OR or AND operation between the HSYNC and VSYNC signals (see [Figure 10-162](#)). The logical operation (OR or AND) depends on the HSYNC and VSYNC signals polarity. The VSYNC signal indicates to the RFBI when vertical synchronization occurs; the HSYNC signal indicates to the RFBI when horizontal synchronization occurs.

**Figure 10-162. External Generation of TE Signal Based on Logical OR Operation Between HSYNC and VSYNC (Active-High)**


- TE signal is connected to rfb<sub>i</sub>\_te\_vsync0 external pad when the [RFBI\\_CONFIG\[3:2\]](#) TRIGGERMODE bit field is set to 0x1.
- HSYNC and VSYNC signals are connected to rfb<sub>i</sub>\_hsync0 and rfb<sub>i</sub>\_te\_vsync0, respectively, when the [RFBI\\_CONFIG\[3:2\]](#) TRIGGERMODE bit field is set to 0x2.

The RFBI module detects the VSYNC and HSYNC pulses embedded in the received signal. VSYNC is detected based on the minimum pulse width defined by the [RFBI\\_VSYNC\\_WIDTH](#) register. VSYNC is not triggered by an inactive-to-active edge.

HSYNC is detected based on the minimum pulse width defined by the [RFBI\\_HSYNC\\_WIDTH](#) register. HSYNC is not triggered by an inactive-to-active edge.

The signal is generated from external logic based on VSYNC and HSYNC of the LCD panel. The automatic trigger can be programmed based on the external RFBI\_TE signal or use the [RFBI\\_CONTROL\[4\]](#) ITE bit to start data capture (internal trigger mode).

The polarity of the TE signal is programmable in the [RFBI\\_CONFIG](#) register. HSYNC and VSYNC pulses embedded in the TE signal have the same polarity, which is active high for an ORed signal and active low for an ANDed signal. The minimum pulse width of the signal is two RFBI\_ICLK clock cycles. Hardware resets the line counter when VSYNC occurs and increments it at every HSYNC. Transfer to the LCD panel begins when the line counter reaches the programmable line number.

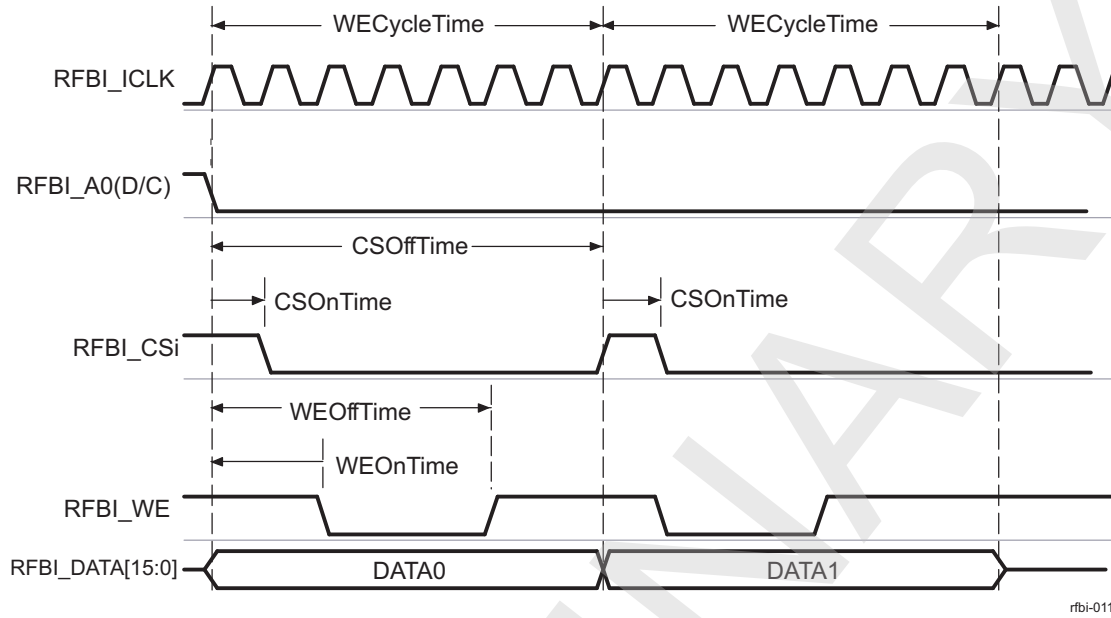
### 10.5.2.2 Transaction Timing Diagrams

[Table 10-632](#) lists the programmable timing fields. [Figure 10-163](#) through [Figure 10-165](#) show timing diagrams of read/write transactions to the LCD panel for the RFBI. In these figures, the polarity 0 (active low) is used to describe the RFBI\_A0, RFBI\_CS*i*, RFBI\_RE, and RFBI\_WE signals.

**Table 10-632. Programmable Timing Fields in RFBI Mode**

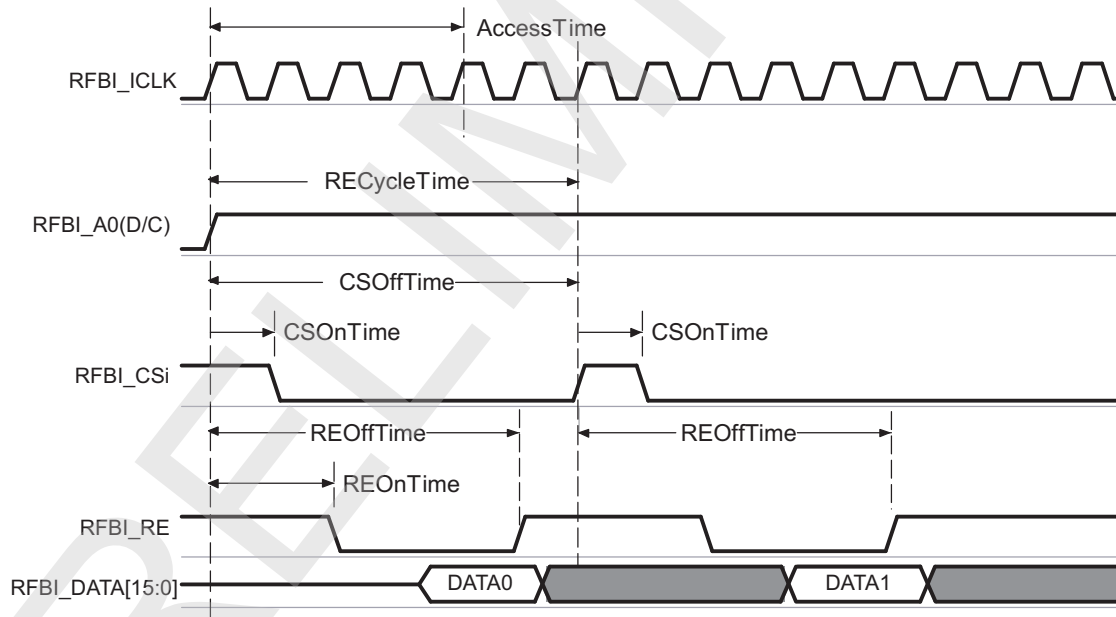
Timing Name	Register Field	Description
CSONTime	<a href="#">RFBI_ONOFF_TIME[3:0]</a> CSONTIME bit field	CS assertion time from start access time
CSoFFTime	<a href="#">RFBI_ONOFF_TIME[9:4]</a> CSOFFTIME bit field	CS deassertion time from start access time
WECycleTime	<a href="#">RFBI_CYCLE_TIME[5:0]</a> WECYCLETIME bit field	The time when A0 becomes valid until write cycle completion
WEOnTime	<a href="#">RFBI_ONOFF_TIME[13:10]</a> WEONTIME bit field	WE assertion delay time from start access time
WEOffTime	<a href="#">RFBI_ONOFF_TIME[19:14]</a> WEOFFTIME bit field	WE deassertion delay time from start access time
RECycleTime	<a href="#">RFBI_CYCLE_TIME[11:6]</a> RECYCLETIME bit field	The time when A0 becomes valid until read cycle completion
REOnTime	<a href="#">RFBI_ONOFF_TIME[23:20]</a> REONTIME bit field	RE assertion delay time from start access time
REOffTime	<a href="#">RFBI_ONOFF_TIME[29:24]</a> REOFFTIME bit field	RE deassertion delay time from start access time
CSPulseWidth	<a href="#">RFBI_CYCLE_TIME[17:12]</a> CSPULSEWIDTH bit field	The time when write cycle time or read cycle time completes

Figure 10-163. Command Data Write

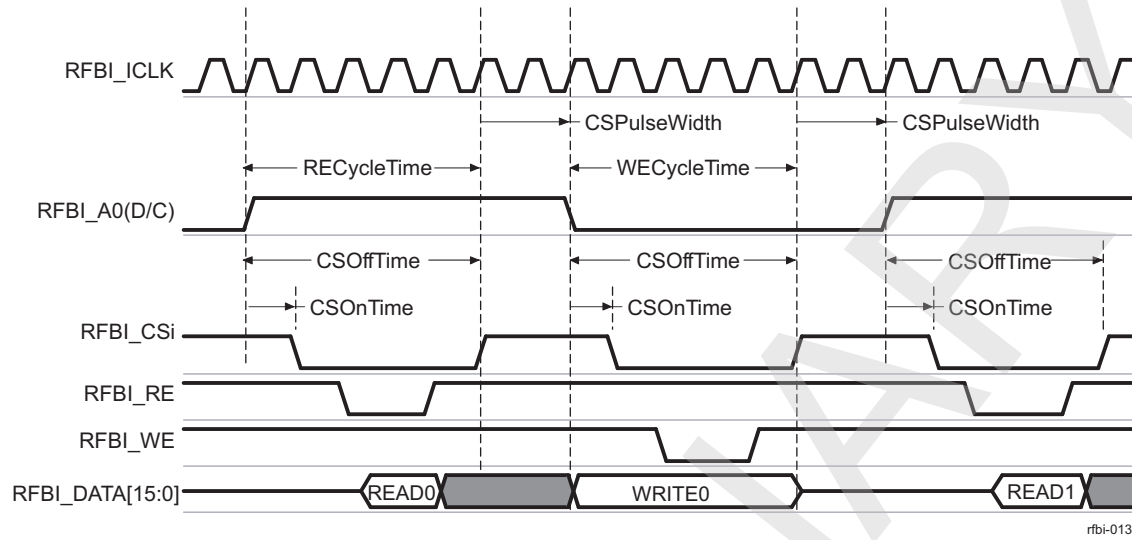


rbi-011

Figure 10-164. Display Data Read



rbi-012

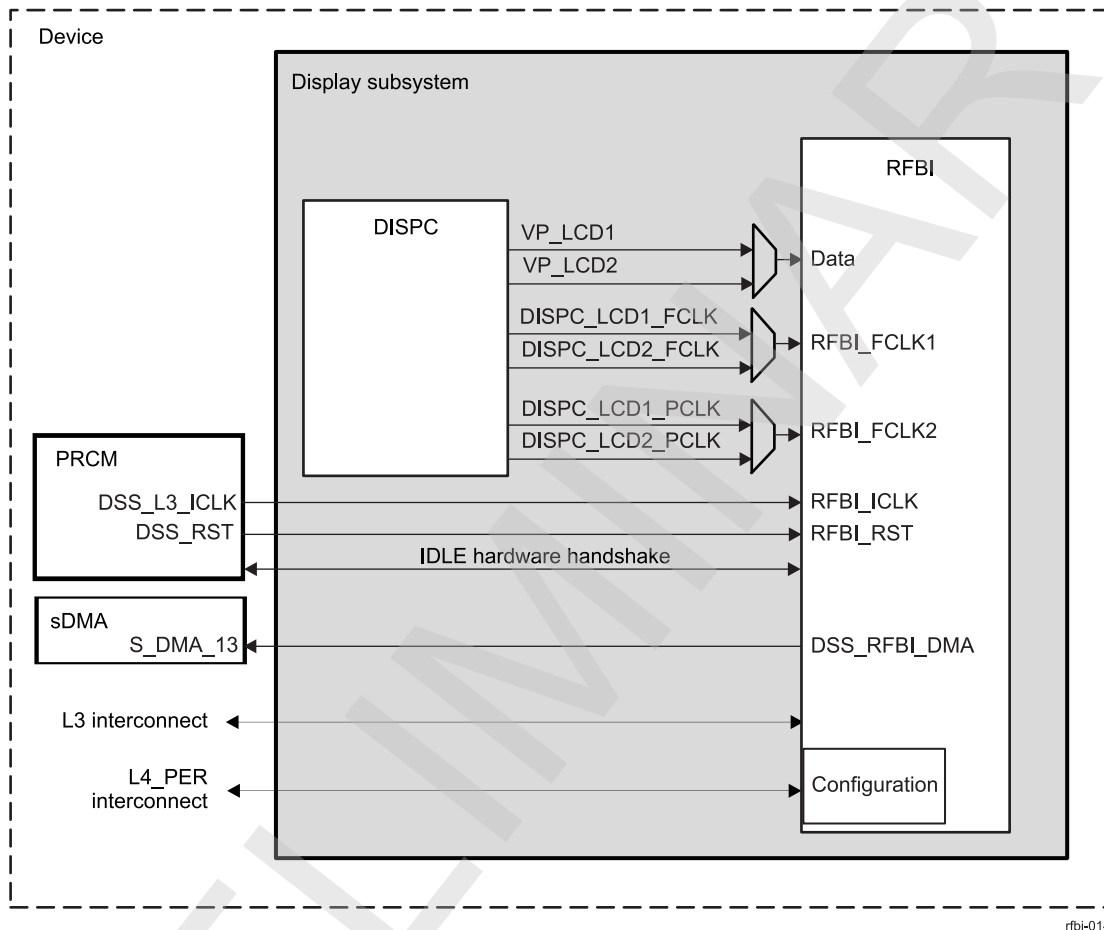
**Figure 10-165. Read to Write and Write to Read**

### 10.5.3 RFBI Integration

This section describes the RFBI integration in the device, including information about clocks, resets, and hardware requests.

Figure 10-166 shows the RFBI module integration.

Figure 10-166. RFBI Integration



**NOTE:** For more information about the IDLE hardware handshake, see [Section 3.1.1.1.2, Module-Level Clock Management, Chapter 3, Power, Reset, and Clock Management](#).

Table 10-633 through Table 10-635 summarize the integration of the module in the device.

Table 10-633. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
RFBI	DSS	No	L3 L4_PER

**Table 10-634. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
RFBI	RFBI_FCLK1	DISPC_LCDx_FCLK	DISPC	Logic clock used by the RFBI when capturing data on the video port and for the generation of the RFBI_STALL signal
	RFBI_FCLK2	DISPC_LCDx_PCLK	DISPC	Pixel clock used by the RFBI to capture the pixels
	RFBI_ICLK	DSS_L3_ICLK	PRCM	L3 interface clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
RFBI	RFBI_RST	DSS_RST	PRCM	Display subsystem global hardware reset

**Table 10-635. Hardware Requests**

DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
RFBI	DSS_RFBI_DMA	S_DMA_13	sDMA	RFBI DMA request

**NOTE:** For DMA source description, see [Section 10.5.3.1, DMA Requests](#).

### 10.5.3.1 DMA Requests

All peripherals internal to the device use the transition-sensitive scheme for DMA requests. For more information about the transition-sensitive scheme, see [Chapter 16, sDMA](#).

[Table 10-636](#) describes the RFBI DMA request.

**Table 10-636. RFBI DMA Request Description**

DMA	Source	Description	Sensitivity <sup>(1)</sup>
RFBI_DMA	DSS_RFBI_DMA	The DMA request can be used to transfer data using the sDMA T from memory to RFBI interconnect slave FIFO.	T

<sup>(1)</sup> T: Transition-sensitive (fixed in hardware) active low request



### 10.5.4 RFBI Functional Description

This section describes the functions of the RFBI.

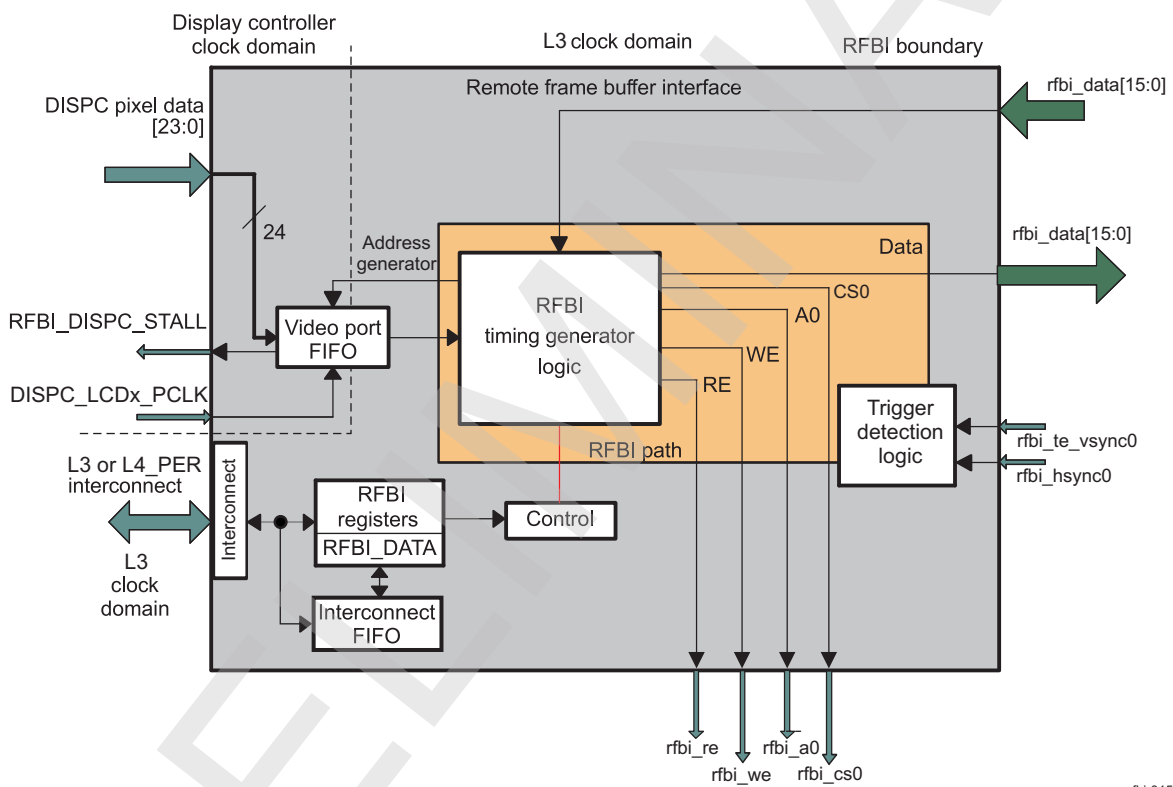
#### 10.5.4.1 Block Diagram

The RFBI can capture the output pixel from the DISPC and send the data to the RFB in the LCD panel. The application configures the RFBI, sends commands, reads data, and configures the DISPC to send data fetched from the system memory by the DISPC DMA engine. The commands/data are sent using an 8-, 9-, or 16-bit parallel interface.

The DISPC is configured to send the data in 12-, 16-, 18-, or 24-BPP format. In the video port FIFO, the encoded pixel values are in an LSB alignment independent of the endianness in system memory.

Figure 10-167 is an overview of the RFBI architecture.

Figure 10-167. RFBI Architecture Overview



rfbi-015

#### 10.5.4.2 Clock Configuration

The RFBI functional clocks (RFBI\_FCLK1 and RFBI\_FCLK2) are provided internally to the display subsystem by the DISPC. For more details about the DISPC clocks, see, Section 10.2 Display Controller.

#### 10.5.4.3 Software Reset

To perform a software reset, set the RFB\_SYSCONFIG[1] SOFTRESET bit to 1. The RFB\_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the RFB\_SYSCONFIG[1] SOFTRESET bit is automatically reset. Software must ensure that the software reset completes before performing any RFBI operation.

#### 10.5.4.4 Power Management

Table 10-637 describes power-management features available for the RFBI.

**NOTE:** For information about source clock gating description, see [Section 3.1.1.1.2, Module-Level Clock Management](#) in [Chapter 3, Power, Reset and Clock Management](#).

**Table 10-637. Local Power Management Features**

Feature	Registers	Description
Clock autogating	<a href="#">RFBI_SYSCONFIG</a> [0] AUTOIDLE bit	This bit allows a local power optimization inside the module by gating functional clocks (RFBI_FCLK1, RFBI_FCLK2) and interface clock (RFBI_ICLK) upon the internal activity.
Slave idle modes	<a href="#">RFBI_SYSCONFIG</a> [4:3] SIDLEMODE bit field	Force-idle, no-idle, and smart-idle. For more details, see below.
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	N/A	Feature not available
Wake-up sources enable	N/A	Feature not available

As shown in [Table 10-637](#), the RFBI supports the No-Idle, Force-idle, and Smart-Idle modes.

The conditions of assertion of the SIdleAck signal are:

- No-idle mode: SIdleAck is never asserted.
- Force-idle mode: SIdleAck is asserted unconditionally after one RFBI\_ICLK clock cycle delay with respect to assertion of IdleReq.
- Smart-idle mode: SIdleAck is asserted when the following conditions are satisfied:
  - The RFBI stops using the RFBI\_ICLK clock and all the data have been sent to the RFB (LCD panel).

Once SIdleAck signal is asserted:

- The RFBI\_ICLK (DSS\_L3\_ICLK) clock can shutdown at any time.
- Any transaction on the RFBI configuration port is ignored

The conditions of deassertion of the SIdleAck signal are:

- Force-idle mode: In this mode SIdleAck is deasserted after one RFBI\_ICLK clock cycle delay with respect to deassertion of IdleReq.
- Smart-idle mode: In this mode SIdleAck is deasserted after one RFBI\_ICLK clock cycle delay with respect to deassertion of IdleReq.

Once SIdleAck is released, the RFBI is fully operational (guaranteed only in the case of smart-idle mode).

#### 10.5.4.5 Interrupt Requests

The RFBI does not generate interrupts but the following DISPC interrupts can be enabled depending on which LCD output of DISPC is currently used for connection to the RFBI:

- FRAMEDONE1\_IRQ
- PROGRAMMEDLINENUMBER\_IRQ
- GFXFIFOUNDERFLOW\_IRQ
- GFXENDWINDOW\_IRQ
- PALETTEGAMMALOADING\_IRQ
- OCPERROR\_IRQ
- VID1FIFOUNDERFLOW\_IRQ
- VID1ENDWINDOW\_IRQ
- VID2FIFOUNDERFLOW\_IRQ
- VID2ENDWINDOW\_IRQ
- SYNCLOST1\_IRQ
- WAKEUP\_IRQ

- SYNCLOST2\_IRQ
- VID3FIFOUNDERFLOW\_IRQ
- VID3ENDWINDOW\_IRQ
- FRAMEDONE2\_IRQ

**NOTE:** For more details about DISPC interrupts, see, [Section 10.2 Display Controller](#).

The other DISPC interrupts are not used in the RFBI.

### 10.5.4.6 DMA Requests

[Table 10-638](#) lists event flags and their mask, that can cause RFBI DMA requests.

**Table 10-638. DMA Requests Events**

DMA Request Name	DMA Request Mask	Map to	Description
DSS_RFBI_DMA	RFBI_CONTROL[7] DISABLE_DMA_REQ	S_DMA_XX	DMA request used to transfer data using the sDMA from memory to the RFBI interconnect FIFO

**NOTE:** For more details about DMA requests, see [Figure 10-173](#).

### 10.5.4.7 Video Port FIFO

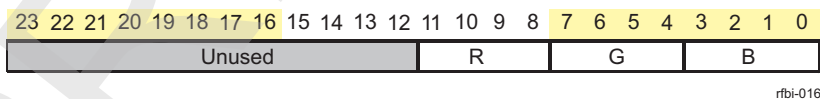
#### 10.5.4.7.1 Description

The input video port FIFO receives data from the DISPC at the pixel clock. The data in the video port FIFO are read by the RFBI and sent to the LCD panel. The video port FIFO is 24 bits wide and each pixel in 12-, 16-, 18-, and 24-BPP format is stored in the video port FIFO using one 24-bit value aligned on the 24-bit LSB. [Section 10.5.4.9, RFBI Output Parallel Modes](#), provides examples of various output configurations based on the interface width (up to 16 bits) and pixel format output (up to 24 bits). Setting the [RFBI\\_CONTROL\[1\]](#) RFBIMODE bit to 0 directs the MPU to send commands, parameters, and data from the input video port FIFO.

#### 10.5.4.7.2 RFBI Input Formats

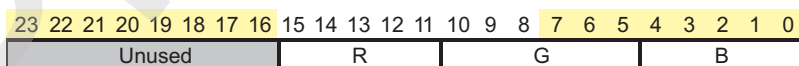
The encoded pixel formats supported at the RFBI input interface connected to the DISPC are: RGB12-444, RGB16-565, RGB18-666, and RGB24-888. This input data width is controlled by the [RFBI\\_CONFIG\[6:5\]](#) DATATYPE bit field. The following graphics describe the bit representation of each format in the video port FIFO. The output of the DISPC is aligned on the LSB of the interface. The pixels are then formatted in accordance with the configuration of the output intercaces (multiple cycles); for more details, see [Section 10.5.4.9.1, Cycle Mode Selection](#).

- RGB12-444 format:



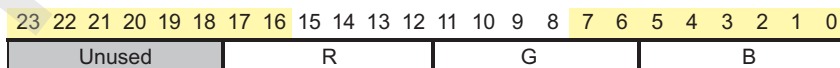
rfbi-016

- RGB16-565 format:



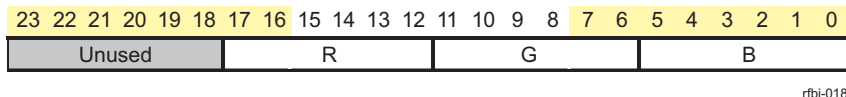
rfbi-017

- RGB18-666 format:



rfbi-018

- RGB24-888 format:



**NOTE:** These pixel formats are also supported when data are provided from the slave port interconnect (for writing parameters).

### 10.5.4.7.3 Stall Mechanism

The stall signal (RFBI\_STALL) is generated by the video port FIFO and used to indicate when the DISPC must stop sending data on the corresponding LCD output interface. This stall signal is used for both LCD outputs (DISPC\_LCD1\_DATA\_STALL and DISPC\_LCD2\_DATA\_STALL) only when the DISPC is configured in stall mode. The stall mode is activated by setting the DISPC\_CONTROLo[11] STALLMODE bit to 1 for the primary LCD output and DISPC\_CONTROLo[11] STALLMODE bit to 1 for the secondary LCD output.

**NOTE:** When the DISPC is configured in stall mode, the minimum transfer size is 1 byte.

The RFBI\_STALL signal allows the RFBI to reformat the data.

RFBI\_STALL is asserted when at least one of the following cases occurs:

- Default status when there is no data to capture from the DISPC
- High FIFO threshold reached
- End of transfer (number of data to output)
- RFBI reset
- [RFBI\\_CONTROL\[0\]](#) ENABLE bit reset to 0x0

RFBI\_STALL is deasserted when the [RFBI\\_CONTROL\[0\]](#) ENABLE bit is set to 0x1 and at least one of the following cases occurs:

- Low FIFO threshold reached
- External TE occurs and the [RFBI\\_CONFIG\[3:2\]](#) TRIGGERMODE bit field is set to 0x1 for automatic external trigger (start of the transfer, the FIFO pointers are reset, the FIFO is empty).
- [RFBI\\_CONTROL\[4\]](#) ITE bit set to 0x1 by users (start of transfer, the FIFO pointers are reset, the FIFO is empty)

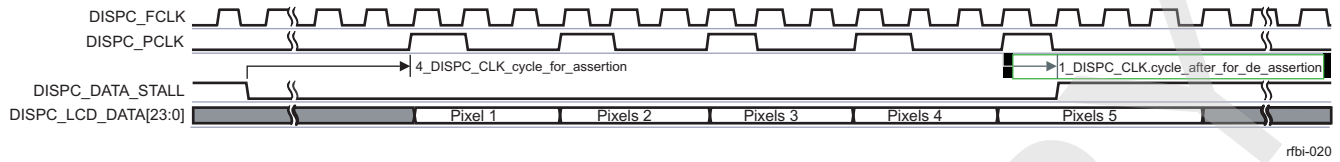
**NOTE:** This stall mechanism is also used by the DSI protocol engines. When multiple modules are connected to the same LCD output (RFBI and DSI protocol engines), the STALL signals from the modules are merged into a single DISPC\_LCDx\_DATA\_STALL signal inside the display subsystem. Only one signal for each LCD output is connected the DISPC (two signals total for the two LCD outputs). In that case, only one module can control the STALL assertion/deassertion. The module which is not using the LCD output interface at that time is not disturbed because the STALL signal is asserted/deasserted by another module. For more details, see chapter *Display Subsystem*.

The RFBI asserts the STALL signal to stop data output by the DISPC. It is deasserted to indicate when new data must be output by the DISPC.

#### 10.5.4.7.3.1 Data Stall Diagram Without DMA Buffer Handshake

Figure 10-168 shows the RFBI data stall diagram when DISPC\_DIVISOR[7:0] PCD bit field is set to 3.

**Figure 10-168. RFBI Data Stall Signal Diagram With PCD=3**



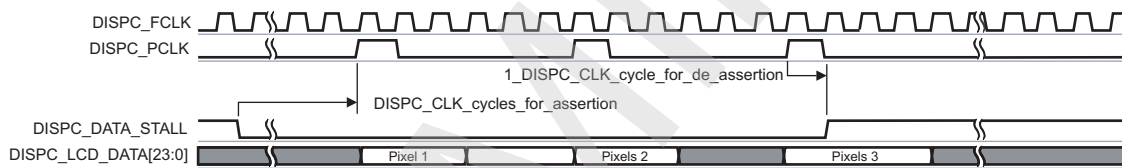
rfti-020

#### 10.5.4.7.3.2 Data Stall Diagram with DMA Buffer Handshake

To avoid underflow of the DMA buffer, the DMA buffer handshake feature can be enabled by setting the DISPC\_CONFIG0[16] BUFFERHANDCHECK bit to 1. The fullness of the FIFOs associated with the pipelines used for the LCD output is checked when the STALL signal is inactive before providing data to the pipeline. This prevents emptying the FIFO when the RFBI requests data and there is not enough data in the DISPC DMA buffer. This feature must be enabled only when stall mode is used (DISPC\_CONTROL0[11] STALLMODE bit set to 1).

When the DMA buffer handshake feature is activated, the pixel transfer to the RFBI during STALL inactivity period can be stopped (no DISPC\_LCD1\_PCLK or DISPC\_LCD2\_PCLK pulse) and restarted when there is enough data in the DMA buffer. The DMA buffer handshake ensures that underflow cannot occur for the pipelines associated with the LCD output in stall mode. Figure 10-169 shows the RFBI data stall with DMA buffer handshake mode activated (PCD=3).

**Figure 10-169. RFBI Data Stall Signal Diagram With Handshake With PCD=3**



rfti-021

### 10.5.4.8 Interconnect FIFO

#### 10.5.4.8.1 Description

The interconnect FIFO receives the data from RFBI\_DATA write requests to the slave port interconnect slave port. The data in the interconnect FIFO are read by the RFBI and sent to the LCD panel. The interconnect FIFO is 32 bits wide. The size of the interconnect FIFO is 24 words of 32 bits (that is, 24 words of RFBI\_DATA). The address of the RFBI\_DATA register is used to access the interconnect FIFO. The DSS\_RFBI\_DMA DMA request can be used to transfer data using the system DMA from memory to interconnect FIFO.

#### 10.5.4.8.2 Using DMA Request With Interconnect FIFO

##### 10.5.4.8.2.1 Threshold for DMA Request Generation

The RFBI\_CONTROL[6:5] HIGHTHRESHOLD bit field is used to define the threshold to be used for the generation of the DMA request to receive data into the interconnect FIFO (24 x 32 FIFO depth) through the address of the register RFBI\_DATA. The sDMA configuration regarding the size of the burst depends on the value of the RFBI\_CONTROL[6:5] HIGHTHRESHOLD bit field. The supported values are 4 x 32, 8 x 32 and 16 x 32. When for example, 4 words of 32 bits are set for the threshold, the sDMA sends a burst of 4 x 32 bits only. The sDMA receives the DMA request and is in charge of providing the correct number of bytes.

#### 10.5.4.8.2.2 Disabling DMA Request

If the [RFBI\\_CONTROL\[7\] DISABLE\\_DMA\\_REQ](#) bit is reset, the DMA request is generated when there is enough room in the interconnect FIFO to accept the full burst. In case the RFBI receives writes slave port requests to the [RFBI\\_DATA](#) location when the interconnect FIFO is full, the request is not accepted. The RFBI waits for a free entry in the interconnect FIFO to accept the slave port request.

If the [RFBI\\_CONTROL\[7\] DISABLE\\_DMA\\_REQ](#) bit is set, the DMA request is not generated. The threshold value is ignored.

---

**NOTE:** Software users can access the [RFBI\\_DATA](#) location without using the DMA request and without programming the high threshold value (backward compatibility mode).

---

#### 10.5.4.8.2.3 Smart DMA Request Mode

If the [RFBI\\_CONTROL\[8\] SMART\\_DMA\\_REQ](#) bit is reset (smart DMA mode disabled), the DMA request is asserted and deasserted depending on the interconnect FIFO space even if `Midlereq` is high in smart-idle/no-idle mode and the entire burst gets error responses from the RFBI. The RFBI waits for a free entry in the interconnect FIFO to accept the burst request.

If the [RFBI\\_CONTROL\[8\] SMART\\_DMA\\_REQ](#) bit is set (smart DMA mode enabled), the DMA request is deasserted after two [RFBI\\_ICLK](#) clock cycles if it has been asserted for more than or equal to two [RFBI\\_ICLK](#) clock cycles and `Midlereq` is high in smart-idle or no-idle mode. If asserting time is less than two [RFBI\\_ICLK](#) clock cycles, no more burst requests are accepted even if the space is available in the interconnect FIFO.

### 10.5.4.9 RFBI Output Parallel Modes

#### 10.5.4.9.1 Cycle Mode Selection

The `RFBI_CONFIG[10:9]` `CYCLEFORMAT` bit field defines the number of cycles to output a pixel. The number of cycles determines the number of registers used to format the data in the interconnect FIFO with the appropriate number of bits (starting from the LSB) and with the alignment on the interface. The data are formatted based on the configuration of the `RFBI_DATA_CYCLEi` registers.

- One cycle: `RFBI_CONFIG[10:9]` `CYCLEFORMAT` bit field = 0x00 and the `RFBI_DATA_CYCLE1` register
- Two cycles: `RFBI_CONFIG [10:9]` `CYCLEFORMAT` bit field = 0x01 and the `RFBI_DATA_CYCLE1` and `RFBI_DATA_CYCLE2` registers
- Three cycles: `RFBI_CONFIG[10:9]` `CYCLEFORMAT` bit field = 0x10 and the `RFBI_DATA_CYCLE1`, `RFBI_DATA_CYCLE2`, and `RFBI_DATA_CYCLE3` registers

Figure 10-170 through Figure 10-172 list the bits position of the pixel during the cycles for an 8-, 9- and 16-bit parallel output, respectively.

Figure 10-170. 8-Bit Interface Settings

	24-BPP		
	1st cycle	2nd cycle	3rd cycle
Data[7]	R0[7]	G0[7]	B0[7]
Data[6]	R0[6]	G0[6]	B0[6]
Data[5]	R0[5]	G0[5]	B0[5]
Data[4]	R0[4]	G0[4]	B0[4]
Data[3]	R0[3]	G0[3]	B0[3]
Data[2]	R0[2]	G0[2]	B0[2]
Data[1]	R0[1]	G0[1]	B0[1]
Data[0]	R0[0]	G0[0]	B0[0]

`RFBI_CONFIG.CYCLEFORMAT` = 0x2  
`RFBI_DATA_CYCLE1` = 0x00000008  
`RFBI_DATA_CYCLE2` = 0x00000008  
`RFBI_DATA_CYCLE3` = 0x00000008

	18-BPP		
	1st cycle	2nd cycle	3rd cycle
Data[7]	R0[5]	G0[3]	x
Data[6]	R0[4]	G0[2]	x
Data[5]	R0[3]	G0[1]	x
Data[4]	R0[2]	G0[0]	x
Data[3]	R0[1]	B0[5]	x
Data[2]	R0[0]	B0[4]	x
Data[1]	G0[5]	B0[3]	B0[1]
Data[0]	G0[4]	B0[2]	B0[0]

`RFBI_CONFIG.CYCLEFORMAT` = 0x2  
`RFBI_DATA_CYCLE1` = 0x00000008  
`RFBI_DATA_CYCLE2` = 0x00000008  
`RFBI_DATA_CYCLE3` = 0x00000002

	16-BPP	
	1st cycle	2nd cycle
Data[7]	R0[4]	G0[2]
Data[6]	R0[3]	G0[1]
Data[5]	R0[2]	G0[0]
Data[4]	R0[1]	B0[4]
Data[3]	R0[0]	B0[3]
Data[2]	G0[5]	B0[2]
Data[1]	G0[4]	B0[1]
Data[0]	G0[3]	B0[0]

`RFBI_CONFIG.CYCLEFORMAT` = 0x1  
`RFBI_DATA_CYCLE1` = 0x00000008  
`RFBI_DATA_CYCLE2` = 0x00000008

	12-BPP	
	1st cycle	2nd cycle
Data[7]	R0[3]	x
Data[6]	R0[2]	x
Data[5]	R0[1]	x
Data[4]	R0[0]	x
Data[3]	G0[3]	B0[3]
Data[2]	G0[2]	B0[2]
Data[1]	G0[1]	B0[1]
Data[0]	G0[0]	B0[0]

`RFBI_CONFIG.CYCLEFORMAT` = 0x1  
`RFBI_DATA_CYCLE1` = 0x00000008  
`RFBI_DATA_CYCLE2` = 0x00000004

rbi-026

**Figure 10-171. 9-Bit Interface Settings**

24-BPP				18-BPP		
	1st cycle	2nd cycle	3rd cycle	1st cycle	2nd cycle	
Data[8]	R0[7]	G0[6]	x	R0[5]	G0[2]	
Data[7]	R0[6]	G0[5]	x	R0[4]	G0[1]	
Data[6]	R0[5]	G0[4]	x	R0[3]	G0[0]	
Data[5]	R0[4]	G0[3]	B0[5]	R0[2]	B0[5]	
Data[4]	R0[3]	G0[2]	B0[4]	R0[1]	B0[4]	
Data[3]	R0[2]	G0[1]	B0[3]	R0[0]	B0[3]	
Data[2]	R0[1]	G0[0]	B0[2]	G0[5]	B0[2]	
Data[1]	R0[0]	B0[7]	B0[1]	G0[4]	B0[1]	
Data[0]	G0[7]	B0[6]	B0[0]	G0[3]	B0[0]	

RFBI\_CONFIG.CYCLEFORMAT= 0x2  
 RFBI\_DATA\_CYCLE1 = 0x00000009  
 RFBI\_DATA\_CYCLE2 = 0x00000009  
 RFBI\_DATA\_CYCLE3 = 0x00000006

RFBI\_CONFIG.CYCLEFORMAT = 0x1  
 RFBI\_DATA\_CYCLE1 = 0x0000000912  
 RFBI\_DATA\_CYCLE2 = 0x00000009

16-BPP			12-BPP		
	1st cycle	2nd cycle	1st cycle	2nd cycle	
Data[8]	R0[4]	x	R0[3]	x	
Data[7]	R0[3]	x	R0[2]	x	
Data[6]	R0[2]	G0[1]	R0[1]	x	
Data[5]	R0[1]	G0[0]	R0[0]	x	
Data[4]	R0[0]	B0[4]	G0[3]	x	
Data[3]	G0[5]	B0[3]	G0[2]	x	
Data[2]	G0[4]	B0[2]	G0[1]	B0[2]	
Data[1]	G0[3]	B0[1]	G0[0]	B0[1]	
Data[0]	G0[2]	B0[0]	B0[3]	B0[0]	

RFBI\_CONFIG.CYCLEFORMAT = 0x1  
 RFBI\_DATA\_CYCLE1 = 0x00000009  
 RFBI\_DATA\_CYCLE2 = 0x00000007

RFBI\_CONFIG.CYCLEFORMAT = 0x1  
 RFBI\_DATA\_CYCLE1 = 0x00000009  
 RFBI\_DATA\_CYCLE2 = 0x00000003

rfbi-027



Figure 10-172. 16-Bit Interface Settings

	24-BPP		
	1st cycle	2nd cycle	3rd cycle
Data[15]	R0[7]	B0[7]	G1[7]
Data[14]	R0[6]	B0[6]	G1[6]
Data[13]	R0[5]	B0[5]	G1[5]
Data[12]	R0[4]	B0[4]	G1[4]
Data[11]	R0[3]	B0[3]	G1[3]
Data[10]	R0[2]	B0[2]	G1[2]
Data[9]	R0[1]	B0[1]	G1[1]
Data[8]	R0[0]	B0[0]	G1[0]
Data[7]	G0[7]	R1[7]	B1[7]
Data[6]	G0[6]	R1[6]	B1[6]
Data[5]	G0[5]	R1[5]	B1[5]
Data[4]	G0[4]	R1[4]	B1[4]
Data[3]	G0[3]	R1[3]	B1[3]
Data[2]	G0[2]	R1[2]	B1[2]
Data[1]	G0[1]	R1[1]	B1[1]
Data[0]	G0[0]	R1[0]	B1[0]

RFBI\_CONFIG.CYCLEFORMAT = 0x3  
 RFBI\_DATA\_CYCLE1 = 0x00000010  
 RFBI\_DATA\_CYCLE2 = 0x00080808  
 RFBI\_DATA\_CYCLE3 = 0x00100000

	18-BPP	
	1st cycle	2nd cycle
Data[15]	R0[5]	x
Data[14]	R0[4]	x
Data[13]	R0[3]	x
Data[12]	R0[2]	x
Data[11]	R0[1]	x
Data[10]	R0[0]	x
Data[9]	G0[5]	x
Data[8]	G0[4]	x
Data[7]	G0[3]	X
Data[6]	G0[2]	x
Data[5]	G0[1]	x
Data[4]	G0[0]	x
Data[3]	B0[5]	x
Data[2]	B0[4]	x
Data[1]	B0[3]	B0[1]
Data[0]	B0[2]	B0[0]

RFBI\_CONFIG.CYCLEFORMAT = 0x1  
 RFBI\_DATA\_CYCLE1 = 0x00000010  
 RFBI\_DATA\_CYCLE2 = 0x00000002

	16-BPP
	1st cycle
Data[15]	R0[4]
Data[14]	R0[3]
Data[13]	R0[2]
Data[12]	R0[1]
Data[11]	R0[0]
Data[10]	G0[5]
Data[9]	G0[4]
Data[8]	G0[3]
Data[7]	G0[2]
Data[6]	G0[1]
Data[5]	G0[0]
Data[4]	B0[4]
Data[3]	B0[3]
Data[2]	B0[2]
Data[1]	B0[1]
Data[0]	B0[0]

RFBI\_CONFIG.CYCLEFORMAT = 0x0  
 RFBI\_DATA\_CYCLE1 = 0x00000010

	12-BPP
	1st cycle
Data[15]	x
Data[14]	x
Data[13]	x
Data[12]	x
Data[11]	R0[3]
Data[10]	R0[2]
Data[9]	R0[1]
Data[8]	R0[0]
Data[7]	G0[3]
Data[6]	G0[2]
Data[5]	G0[1]
Data[4]	G0[0]
Data[3]	B0[3]
Data[2]	B0[2]
Data[1]	B0[1]
Data[0]	B0[0]

RFBI\_CONFIG.CYCLEFORMAT = 0x0  
 RFBI\_DATA\_CYCLE1 = 0x0000000C

rfbi-028

### 10.5.4.9.2 Unmodified Bits

In a cycle, if every bit in the interface does not have a pixel value, the status of the unused bits can be programmed to be 0, 1, or the previous value (I/O power consumption optimization). Based on the configuration, the undefined bits for each cycle are defined with the previous values of the bits at the same position in the previous cycle, 0s, or 1s (the unused bits can be at any position). The [RFBI\\_CONFIG\[12:11\] UNUSEDBITS](#) bit field is used.

### 10.5.4.9.3 Number of Pixels to Transfer

The **RFBI\_PIXEL\_CNT**[31:0] PIXELCNT bit field indicates the number of pixels to be transferred to the LCD panel. The value can be changed only when the **RFBI\_CONTROL**[0] ENABLE bit is reset.

During the transfer, the hardware decrements the register when a pixel is sent to the RFB. When the **RFBI\_CONTROL**[0] ENABLE bit is set and a new value is written in the **RFBI\_PIXEL\_CNT** register, and the current value in the register is a not 0 (the remaining number of pixels to transfer), the ongoing transfer is aborted.

### 10.5.4.10 RFBI Timing Generator

#### 10.5.4.10.1 Configuration Selection

The **RFBI\_CONTROL**[3:2] CONFIGSELECT bit field selects the chip select.

#### 10.5.4.10.2 Read/Write

Depending on the status of A0, WE, and RE, the commands and display/parameter data are written to the panel (handled by the state-machine for the commands/parameter data and stored in memory for the display data), or the display data/status values are read from the LCD panel (status and display data in the LCD panel memory). The polarity of A0 (RFBI\_A0 signal), WE (RFBI\_WE signal), RE (RFBI\_RE signal), and CS0 (RFBI\_CS0 signal) is programmable.

Table 10-639 describes the read/write function.

**Table 10-639. Read/Write Function Description**

A0 (RFBI_A0)	WE (RFBI_WE)	RE (RFBI_RE)	Function Description
1	0	1	Display data write, parameter data write
1	1	0	Display data read
0	1	0	Status read
0	0	1	Command data write

A minimum of RFBI\_Cs cycle time, as defined in Table 10-640, is required to keep the RFBI\_CS0 signal asserted between write transfers of multiple pixels.

Table 10-640 lists the minimum cycle time for RFBI\_CS0, depending on the source of pixels (DISPC or slave port) and the cycle format (1 pixel/cycle, 1 pixel/2 cycles, or 1 pixel/3 cycles).

**Table 10-640. Minimum Cycle Time for CS0/WE Always Asserted**

RFBI Performance	<b>RFBI_CONFIG</b> [10:9] CYCLEFORMAT	<b>RFBI_CONFIG</b> [8:7] PORTFORMAT	Minimum Cycle Time (in Number of RFBI_ICLK Clock Cycles)
Slave port interconnect	1 pixel/cycle	1 pixel	5
	1 pixel/2 cycles	1 pixel	4
	1 pixel/3 cycles	1 pixel	4
	1 pixel/cycle	2 pixels	4
	1 pixel/2 cycles	2 pixels	4
	1 pixel/3 cycles	2 pixels	4
DISPC	1 pixel/cycle	N/A	4
	1 pixel/2 cycles	N/A	3
	1 pixel/3 cycles	N/A	3
	2 pixel/3 cycles	N/A	6

### 10.5.4.10.3 RFBI State-Machine

The RFBI\_A0, RFBI\_RE, and RFBI\_WE signals are asserted and deasserted based on the register accessed (RFBI\_CMD, RFBI\_PARAM, RFBI\_DATA, RFBI\_READ, or RFBI\_STATUS). (See Table 10-639.) When the RFBI\_SYSSTATUS[8] BUSY bit is set by hardware, any access to the registers is stalled, except for the RFBI\_DATA register.

The RFBI\_SYSSTATUS[9] BUSYRFBIDATA bit indicates whether there are still pending data in the interconnect FIFO associated with the RFBI\_DATA register only.

- **Command register**  
Write one command at a time by writing through the slave port interconnect into the RFBI\_CMD register. If the previous command is not processed, the RFBI\_SYSSTATUS[8] BUSY bit is set by hardware and the access to writing a new command is stalled.
- **Parameter register**  
Write one parameter at a time by writing in the RFBI\_PARAM register.  
If the previous parameter is not processed, the RFBI\_SYSSTATUS[8] BUSY bit is set by hardware and the access to writing a new parameter is stalled.
- **Data register**  
Write one or two pixels at a time by writing in the RFBI\_DATA register.  
The pixels are formatted based on the specified cycle format. If two pixels are written into the 32-data register, the RFBI\_CONFIG[8:7] PORTFORMAT bit field indicates the number of pixels for each slave port access to the register and the order of the pixels.  
If the previous data are not processed, the RFBI\_SYSSTATUS[8] BUSY bit is set by hardware and any access for writing new data is stalled. When the RFBI\_SYSSTATUS[8] BUSY bit is reset by hardware, the access is not stalled.
- **Read/status register**  
Send through the command and parameter registers the correct information to receive data in the data or status register. The read data from the LCD panel is initiated by writing into the RFBI\_READ or RFBI\_STATUS registers. In this case, the RFBI\_SYSSTATUS[8] BUSY bit is set until the data are available in the register.  
When the RFBI\_SYSSTATUS[8] BUSY bit is set by hardware, the read or write access is stalled until the register is updated with a new value from the LCD panel. To avoid the stall, the software can poll the RFBI\_SYSSTATUS[8] BUSY bit until it is reset by hardware. To receive the data, send the appropriate command/parameters.

### 10.5.4.10.4 RFBI Timings

The timing registers can be accessed only when there is no transaction in progress (based on the value of the RFBI\_CONTROL[3:2] CONFIGSELECT bit field). Granularity is defined using the RFBI\_CONFIG[4] TIMEGRANULARITY bit. This feature allows the extension of programmable ranges of timing parameters for the RFBI interface. See Table 10-641 for the configuration values of the timing bits.

- **Chip-select assertion/deassertion time (CSOnTime/CSOffTime)**  
RFBI\_A0 setup time to chip-select assertion is assured by the programmable chip-select assertion time from the start access time:  
RFBI\_ONOFF\_TIME[3:0] CSONTIME bit field.  
The chip-select deassertion time from the start access time is programmable:  
RFBI\_ONOFF\_TIME[9:4] CSOFTIME bit field

#### **CAUTION**

Configuring RFBI\_ONOFF\_TIME[3:0] CSONTIME = RFBI\_ONOFF\_TIME[9:4] CSOFTIME = 0 is not supported and must be avoided. This configuration creates contention on the bus and progressively damages the LCD panel.

- **Chip-select pulse width (CSPulseWidth)**

The total chip-select pulse width is the time when write cycle time or read cycle time has completed and is programmable:

[RFBI\\_CYCLE\\_TIME\[17:12\]](#) CSPULSEWIDTH bit field

It applies on the read-to-write, write-to-read, read-to-read, and write-to-write access based on:

- The [RFBI\\_CYCLE\\_TIME\[19\]](#) RRENABLE bit: Read-to-read access
- The [RFBI\\_CYCLE\\_TIME\[20\]](#) WWENABLE bit: Write-to-write access
- The [RFBI\\_CYCLE\\_TIME\[18\]](#) RWENABLE bit: Read-to-write access
- The [RFBI\\_CYCLE\\_TIME\[21\]](#) WRENABLE bit: Write-to-read access

By default, it applies to any access (read-to-read, read-to-write, write-to-read, write-to-write) when the chip-select CS0 is activated by setting the [RFBI\\_CONTROL\[3:2\]](#) CONFIGSELECT bit field to 0x1.

- Access time

Access time is the time delay between A0 assertion to data sampling before RE signal deassertion; access time is programmable:

[RFBI\\_CYCLE\\_TIME\[27:22\]](#) ACESSTIME bit field

When reading the data on the bus, the data are sampled at the end of the access time, which occurs before the end of the read off time ([RFBI\\_ONOFF\\_TIME\[29:24\]](#) REOFFTIME).

- Write-enable cycle time (WECycleTime)

The total write-enable cycle time is the time when A0 becomes valid until write cycle completion; the write-enable cycle time is programmable:

The [RFBI\\_CYCLE\\_TIME\[5:0\]](#) WECYCLETIME bit field

- Write-enable assertion/deassertion time (WEOnTime/WEOffTime)

The WE assertion delay time from start access time is programmable:

[RFBI\\_ONOFF\\_TIME\[13:10\]](#) WEONTIME bit field

The WE deassertion delay time from the start access time is programmable:

[RFBI\\_ONOFF\\_TIME\[19:14\]](#) WEOFFTIME bit field

- Read-enable cycle time (RECycleTime)

The total read-enable cycle time is the time when A0 becomes valid until read cycle completion; the read-enable cycle time is programmable:

The [RFBI\\_CYCLE\\_TIME\[11:6\]](#) RECYCLETIME bit field

- Read-enable assertion/deassertion time (REOnTime/REOffTime)

The RE assertion delay time from the start access time is programmable:

[RFBI\\_ONOFF\\_TIME\[23:20\]](#) REONTIME bit field

The RE deassertion delay time from the start access time is programmable:

[RFBI\\_ONOFF\\_TIME\[29:24\]](#) REOFFTIME bit field

At cycle time completion (read access or write access), all control signals (RFBI\_CS0, RFBI\_WE, and RFBI\_RE) are deasserted regardless of their deassertion time parameter values, if they are not deasserted already.

However, an exception to this forced deassertion exists when a pipelined request to CS0 is pending. Also, a control signal with deassertion time parameters equal to the cycle time parameter is not necessarily deasserted when a pipelined request to the same chip-select or different chip-select is pending. This prevents any unnecessary glitch transitions.

If no inactive cycles are required between successive accesses to the same chip-select (the [RFBI\\_CYCLE\\_TIME\[17:12\]](#) CSPULSEWIDTH bit field = 0), and if assertion time parameters associated with the following access equal 0, the asserted control signals (RFBI\_CS0, RFBI\_WE, and RFBI\_RE) stay asserted. This applies only to write-to-write access combination. In case of read-to-write, read-to-read, or write-to-read sequences, the RFBI\_CS0, RFBI\_WE, and RFBI\_RE signals are always deasserted.

[Table 10-641](#) lists the configuration values for each timing bit.

**Table 10-641. RFBI Timings Configuration**

Timing Configuration Bits	Granularity <sup>(1)</sup>	
	One	Two
<a href="#">RFBI_ONOFF_TIME</a> [3:0] CSONTIME	0 to 15	0 to 30
<a href="#">RFBI_ONOFF_TIME</a> [9:4] CSOFFTIME	0 to 63	0 to 126
<a href="#">RFBI_CYCLE_TIME</a> [17:12] CSPULSEWIDTH	0 to 63	0 to 126
<a href="#">RFBI_CYCLE_TIME</a> [27:22] ACCESSTIME	0 to 63	0 to 126
<a href="#">RFBI_CYCLE_TIME</a> [5:0] WECYCLETIME	0 to 63	0 to 126
<a href="#">RFBI_ONOFF_TIME</a> [13:10] WEONTIME	0 to 15	0 to 30
<a href="#">RFBI_ONOFF_TIME</a> [19:14] WEOFFTIME	0 to 63	0 to 126
<a href="#">RFBI_CYCLE_TIME</a> [11:6] RECYCLETIME	0 to 63	0 to 126
<a href="#">RFBI_ONOFF_TIME</a> [23:20] REONTIME	0 to 15	0 to 30
<a href="#">RFBI_ONOFF_TIME</a> [29:24] REOFFTIME	0 to 63	0 to 126

<sup>(1)</sup> Number of RFBI\_ICLK clock cycles. The granularity can be configured using the [RFBI\\_CONFIG](#)[4] TIMEGRANULARITY bit.

### 10.5.4.11 Trigger Detection Logic

#### 10.5.4.11.1 Trigger Mode

Setting the [RFBI\\_CONFIG](#)[3:2] TRIGGERMODE bit field configures the different trigger modes:

- Internal trigger mode with the internal programmable [RFBI\\_CONTROL](#)[4] ITE bit
- External trigger mode with external TE signal ([RFBI\\_TE\\_VSYNC](#))
- External trigger mode with external VSYNC/HSYNC signals

#### 10.5.4.11.2 Internal Trigger Mode

##### 10.5.4.11.2.1 ITE Bit

Set the [RFBI\\_CONTROL](#)[4] ITE bit to start capturing the data from the DISPC. The DISPC must be configured in the RFBI mode to account for the [RFBI\\_DISPC\\_STALL](#) signal. Setting the trigger mode to external ([RFBI\\_CONFIG](#)[3:2] TRIGGERMODE bit field set to 0x1 or 0x2) causes the [RFBI\\_CONTROL](#)[4] ITE bit to be ignored. The chip-select CS0 must be selected ([RFBI\\_CONTROL](#)[3:2] CONFIGSELECT set to 0x1) when this bit is set by users.

#### 10.5.4.11.3 External Trigger Mode

There are two external trigger modes:

- TE only: VSYNC and HSYNC are merged by logical OR operation, and are detected through VSYNC and HSYNC pulse widths.
- HSYNC/VSYNC: This mode uses the two external signals which are detected through VSYNC and HSYNC pulse widths.

##### 10.5.4.11.3.1 Programmable Line Number

When the trigger mode is set to external trigger mode with HSYNC and VSYNC or the TE, hardware resets the line counter when the VSYNC occurs and, after a programmable number of lines programmed by the user in the [RFBI\\_LINE\\_NUMBER](#)[10:0] LINENUMBER bit field (the HSYNC pulse occurs for every line), the transfer to the LCD panel begins. When the programmable line number is 0, only the VSYNC pulse indicates the beginning of the transfer in both modes: HSYNC/VSYNC and TE (logical OR operation between HSYNC and VSYNC).

### 10.5.4.11.3.2 VSYNC Pulse Width (Minimum Value)

The [RFBI\\_VSYNC\\_WIDTH](#)[15:0] MINVSYNCPULSEWIDTH bit field defines the minimum number of RFBI\_ICLK clock cycles of the VSYNC pulse for detection on VSYNC. It allows differentiation between VSYNC and HSYNC, which are ORed on the same signal, and is also used in HSYNC/VSYNC mode on the two separate input lines.

- The VSYNC pulse width must be equal to at least two RFBI\_ICLK clock cycles when HSYNC is not present (TE trigger mode only).
- The VSYNC pulse width must be equal to at least four RFBI\_ICLK clock cycles when HSYNC is present (TE trigger mode only).

### 10.5.4.11.3.3 HSYNC Pulse Width (Minimum Value)

The [RFBI\\_HSYNC\\_WIDTH](#)[15:0] MINHSYNCPULSEWIDTH bit field defines the minimum number of RFBI\_ICLK clock cycles of the HSYNC pulse for detection on HSYNC. It allows differentiation between VSYNC and HSYNC, which are ORed on the same signal, and is also used in VSYNC/HSYNC mode on the separate two input lines. To be detected, the HSYNC/VSYNC pulse width must always be equal to at least two RFBI\_ICLK clock cycles. See [Table 10-642](#).

**Table 10-642. Minimum Pulse Width (HSYNC/VSYNC)**

Configuration Bits	TE Mode	HSYNC/VSYNC Mode
<a href="#">RFBI_HSYNC_WIDTH</a> [15:0] MINHSYNCPULSEWIDTH field value	2	2
<a href="#">RFBI_VSYNC_WIDTH</a> [15:0] MINVSYNCPULSEWIDTH field value	4	2

The pulse received by the RFBI must be at least two RFBI\_ICLK clock cycles to be detected. In case of TE mode, since the minimum value for differentiate VSYNC and HSYNC is two RFBI\_ICLK clock cycles, the VSYNC pulse width must be at least four RFBI\_ICLK clock cycles and the HSYNC pulse width must be at least two RFBI\_ICLK clock cycles

## 10.5.5 RFBI Programming Guide

### 10.5.5.1 RFBI Low-Level Programming Models

This section covers the low-level hardware programming sequences for configuration and use of the RFBI.

#### 10.5.5.1.1 Global Initialization

##### 10.5.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the RFBI is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the RFBI. For more information, see [Section 10.5.2, RFBI Environment](#), and [Section 10.5.3, RFBI Integrator](#).

[Table 10-643](#) describes the global initialization of surrounding modules.

**Table 10-643. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	RFBI_ICLK interface clock must be enabled. See <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
DISPC	RFBI_FCLK1 and RFBI_FCLK2 functional clocks must be enabled. Refer to the section TBD in the DISPC chapter.
Control module	<p>rfbi_re pad muxing must be set in the SYSCTRL_PADCONF_CORE module.</p> <p>rfbi_we pad muxing must be set in the SYSCTRL_PADCONF_CORE module.</p> <p>rfbi_cs0 pad muxing must be set in the SYSCTRL_PADCONF_CORE module.</p> <p>rfbi_te_vsync0 pad muxing must be set in the SYSCTRL_PADCONF_CORE module.</p> <p>rfbi_data[15:0] pads muxing must be set in the SYSCTRL_PADCONF_CORE module.</p> <p>rfbi_hsync0 pad muxing must be set in the SYSCTRL_PADCONF_CORE module.</p> <p>rfbi_a0 pad muxing must be set in the SYSCTRL_PADCONF_CORE module.</p> <p>See <a href="#">Chapter 18, Control Module</a>.</p>
MPU INTC	MPU interrupt controller configuration must be done to enable the interrupts from the DISPC module. See <a href="#">Chapter 17, Interrupt Controllers</a> .
sDMA	DMA configuration must be done to enable the RFBI module DMA channel requests. See <a href="#">Chapter 16, Direct Memory Access</a> .
Interconnect	See <a href="#">Chapter 13, Interconnect</a> , for more information about the interconnect configuration.

**NOTE:** The MPU INTC and the sDMA configurations are necessary if the interrupt- and DMA-based communication modes are used.

### 10.5.5.1.1.2 RFBI Global Initialization

#### 10.5.5.1.1.2.1 Main Sequence RFBI Global Initialization

This procedure initializes the RFB and DISPC after a power on or software reset. See [Table 10-644](#).

**Table 10-644. RFBI Global Initialization**

Step	Register/Bit Field Programming Model	Value
DISPC configuration in RFBI mode	See <a href="#">Section 10.5.5.1.1.2.2</a> , <i>Subsequence: DISPC Configuration</i> .	
RFBI configuration	See <a href="#">Section 10.5.5.1.1.2.3</a> , <i>Subsequence: RFBI Configuration</i> .	
LCD panel configuration	See <a href="#">Section 10.5.5.1.1.2.4</a> , <i>Subsequence: LCD Panel Configuration</i> .	

#### 10.5.5.1.1.2.2 Subsequence: DISPC Configuration

This procedure configures the DISPC registers in RFBI mode. See [Table 10-645](#).

**Table 10-645. DISPC Configuration in RFBI Mode**

Step	Register/Bit Field Programming Model	Value
DISPC pipeline configuration	Configuration of the pipeline associated to the LCD output (DMA engine, pipelines associated to the LCD output, etc.). See <a href="#">Section 10.2.4</a> , <i>Display Controller Functional Description</i>	
Enable TFT mode.	DISPC_CONTROLo[3] STNTFT	0x1
Program the DISPC in RFBI mode.	DISPC_CONTROLo[11] STALLMODE	0x1
Reset the DISPC signal configuration to default value.	DISPC_POL_FREQo	0x00
Select the RFBI data path.	DISPC_CONTROL1[16:15] GPOUT	0x1
Enable the hardware handshake to avoid the DISPC FIFO underflow (applies to the pipeline connected to the LCD output).	DISPC_CONFIGo[16] BUFFERHANDCHECK	0x1

**NOTE:** In RFBI mode, the DISPC\_CONTROLo[5] GOLCD bit must not be set to 1. It must be disabled before resetting the DISPC\_CONTROLo[11] STALLMODE bit to 0.

The hardware handshake applies to the pipelines connected to the LCD output. It must be disabled before resetting the DISPC\_CONTROLo[11] STALLMODE bit to 0.

#### 10.5.5.1.1.2.3 Subsequence: RFBI Configuration

This procedure configures the RFBI registers. See [Table 10-646](#).

#### CAUTION

The RFBI configuration registers for CS0 configuration can be accessed only when the configuration is not in use (that is, when the [RFBI\\_CONTROL\[3:2\] CONFIGSELECT](#) is set to 0x0).

**Table 10-646. RFBI Configuration – Main Sequence**

Step	Register/Bit Field Programming Model	Value
Enable the RFBI data path.	<a href="#">RFBI_CONTROL[1] RFBIMODE</a>	0x0
DMA mode configuration	See <a href="#">Section 10.5.5.1.1.2.3.1</a> , <i>Subsequence: DMA Mode Configuration</i> .	
Disable the chip-select and configuration (this step is necessary to configure the timing registers).	<a href="#">RFBI_CONTROL[3:2] CONFIGSELECT</a>	0x0



**Table 10-646. RFBI Configuration – Main Sequence (continued)**

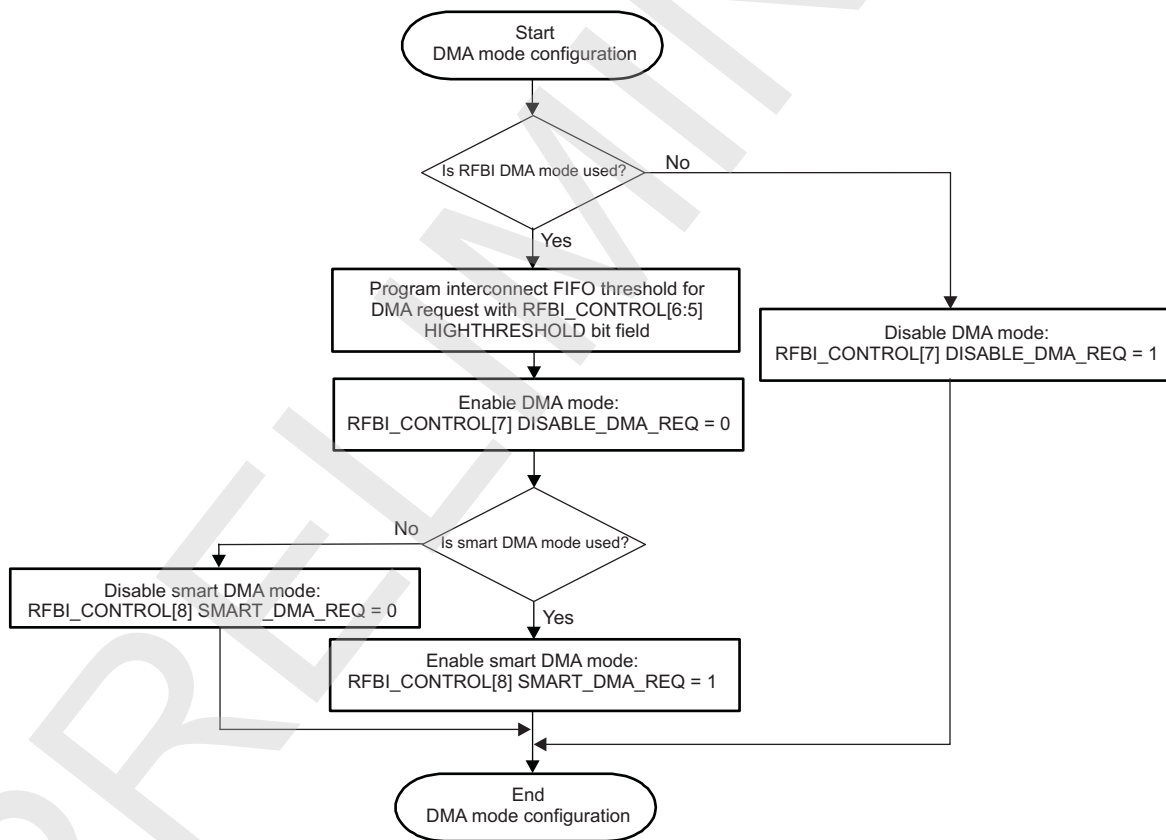
Step	Register/Bit Field Programming Model	Value
Timing and polarity signals configuration	See <a href="#">Section 10.5.5.1.1.2.3.2, Subsequence: Timing and Polarity Signals Configuration</a> .	
Configure the parallel input data width.	RFBI_CONFIG[6:5] DATATYPE	x <sup>(1)</sup>
Configure the number of cycles.	RFBI_CONFIG[10:9] CYCLEFORMAT	x <sup>(1)</sup>
Trigger mode settings	See <a href="#">Section 10.5.5.1.1.2.3.3, Subsequence: Trigger Mode Settings</a> .	
Program the data format per cycle.	RFBI_DATA_CYCLE_1	xxx <sup>(1)</sup>
	RFBI_DATA_CYCLE_2	xxx <sup>(1)</sup>
	RFBI_DATA_CYCLE_3	xxx <sup>(1)</sup>
Select the configuration of CS0	RFBI_CONTROL[3:2] CONFIGSELECT	0x1

<sup>(1)</sup> Values depend on the application

**10.5.5.1.1.2.3.1 Subsequence: DMA Mode Configuration**

Figure 10-173 shows DMA mode configuration.

**Figure 10-173. DMA Mode Configuration**



rbi-024

Table 10-647 describes DMA mode configuration.

**Table 10-647. Register Call Summary for Subsequence: DMA Mode Configuration**

Register Name
RFBI_CONTROL

### 10.5.5.1.1.2.3.2 Subsequence: Timing and Polarity Signals Configuration

Table 10-648 describes timing and polarity signals configuration.

**Table 10-648. Timing and Polarity Signals Configuration**

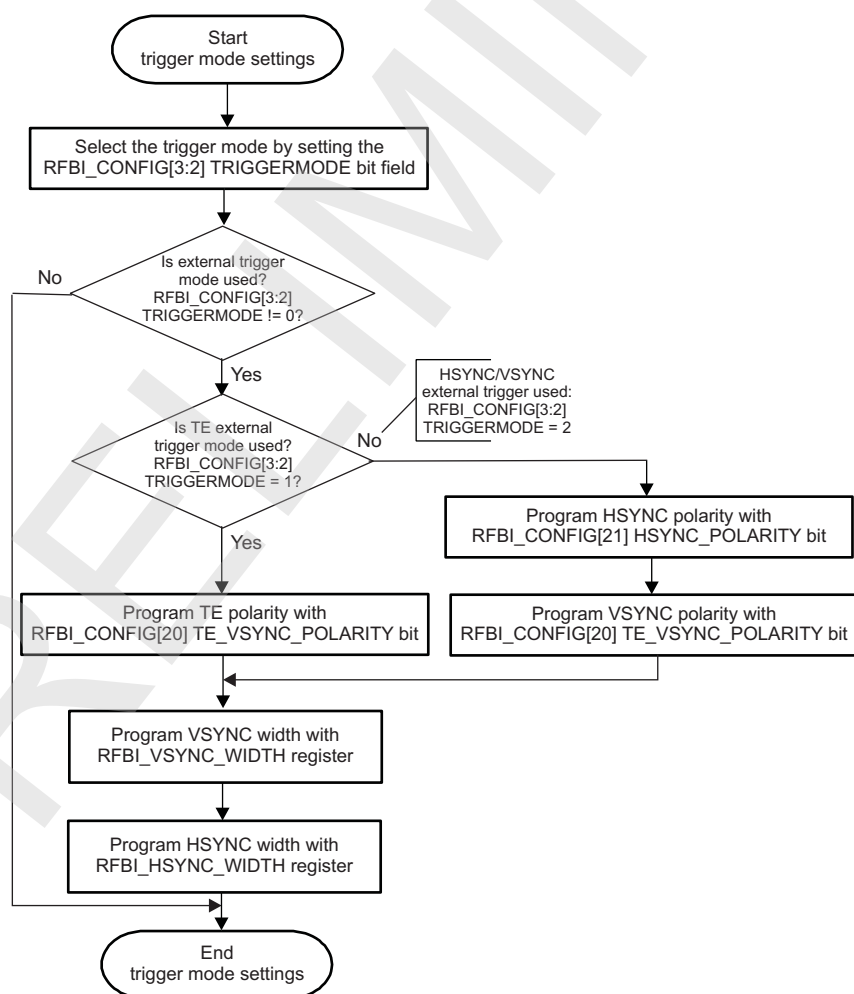
Step	Register/Bit Field Programming Model	Value
Program ON/OFF timings.	RFBI_ONOFF_TIME	xxx <sup>(1)</sup>
Program cycle timings.	RFBI_CYCLE_TIME	xxx <sup>(1)</sup>
Program CS polarity.	RFBI_CONFIG[19] CSPOLARITY	x <sup>(1)</sup>
Program WE polarity.	RFBI_CONFIG[18] WEPOLARITY	x <sup>(1)</sup>
Program RE polarity.	RFBI_CONFIG[17] REPOLARITY	x <sup>(1)</sup>
Program A0 polarity.	RFBI_CONFIG[16] A0POLARITY	x <sup>(1)</sup>
Program unused bits.	RFBI_CONFIG[12:11] UNUSEDBITS	x <sup>(1)</sup>
Program the factor x2 latency.	RFBI_CONFIG[4] TIMEGRANULARITY	x <sup>(1)</sup>

<sup>(1)</sup> Values depend on the application.

### 10.5.5.1.1.2.3.3 Subsequence: Trigger Mode Settings

Figure 10-174 shows trigger mode settings.

**Figure 10-174. Trigger Mode Settings**



rbi-025

Table 10-649 describes register call summary for subsequence: trigger mode settings

**Table 10-649. Register Call Summary for Subsequence: Trigger Mode Settings**

Register Name
RFBI_CONFIG
RFBI_VSYNC_WIDTH
RFBI_HSYNC_WIDTH

**10.5.5.1.1.2.4 Subsequence: LCD Panel Configuration**

This procedure consists of sending and receiving commands. See [Table 10-650](#) and [Table 10-651](#).

**Table 10-650. RFBI Command Interface: Writing Command to the LCD Panel**

Step	Register/Bit Field Programming Model	Value
Write command to the LCD panel.	RFBI_CMD[8:0] CMD	0x <sup>(1)</sup>
Wait until BUSY != 1.	RFBI_SYSSTATUS[8] BUSY	

<sup>(1)</sup> Value depends on the command sent to the LCD panel.

**Table 10-651. RFBI Command Interface: Reading Command From the LCD Panel**

Step	Register/Bit Field Programming Model	Value
Initialize reading command.	RFBI_STATUS[8:0] STATUS	0x
Initialize reading command.	RFBI_READ [8:0] READ	0x
Wait until BUSY != 1.	RFBI_SYSSTATUS[8] BUSY	
Read command from the LCD panel.	RFBI_STATUS[8:0] STATUS	0x <sup>(1)</sup>
	RFBI_READ [8:0] READ	0x <sup>(1)</sup>

<sup>(1)</sup> Value depends on the command read from the LCD panel.

**10.5.5.1.2 Operational Modes Configuration**

**10.5.5.1.2.1 RFBI Start Transfer**

**10.5.5.1.2.1.1 Main Sequence: RFBI Start Transfer With Polling Method**

[Table 10-652](#) describes the RFBI start transfer polling method.

**Table 10-652. RFBI Start Transfer With Polling Method**

Step	Register/Bit Field Programming Model	Value
Enable DISPC module output.	See <a href="#">Section 10.5.5.1.2.1.4, Subsequence: DISPC Output Enable</a> .	
Enable RFBI module output.	See <a href="#">Section 10.5.5.1.2.1.5, Subsequence: RFBI Interrupts Enable</a> .	
Detect event.	See <a href="#">Table 10-658</a> .	

**10.5.5.1.2.1.2 Main Sequence: RFBI Start Transfer With Interrupts and DMA Method**

[Table 10-653](#) describes the RFBI start transfer interrupts method.

**Table 10-653. RFBI Start Transfer With Interrupts Method**

Step	Register/Bit Field Programming Model	Value
Enable DISPC interrupts.	See <a href="#">Table 10-654</a> .	
Enable DISPC module output.	See <a href="#">Section 10.5.5.1.2.1.4, Subsequence: DISPC Output Enable</a> .	
Enable RFBI module output.	See <a href="#">Section 10.5.5.1.2.1.5, RFBI Interrupts Enable</a> .	
Detect Event.	See <a href="#">Table 10-658</a> .	

**10.5.5.1.2.1.3 Subsequence: DISPC Interrupts Enable**

This procedure enables the DISPC interrupts used in RFBI mode. See [Table 10-654](#).

**Table 10-654. DISPC Interrupts Enable**

Step	Register/Bit Field Programming Model	Value
Clears all interrupts	DISPC_IRQSTATUS	0xFFFF
Enable the FRAMEDONEi IRQ.	DISPC_IRQENABLE FRAMEDONEi_EN	0x1
Enable the BUFFERUNDERFLOW IRQ.	DISPC_IRQENABLE BUFFERUNDERFLOW_EN	0x1
Enable the SYNCLOSTi IRQ.	DISPC_IRQENABLE SYNCLOSTi_EN	0x1

**NOTE:** The other interrupts requests listed in (see TBD) can also be enabled depending on the application.

#### 10.5.5.1.2.1.4 Subsequence: DISPC Output Enable

This procedure enables the DISPC output. See [Table 10-655](#).

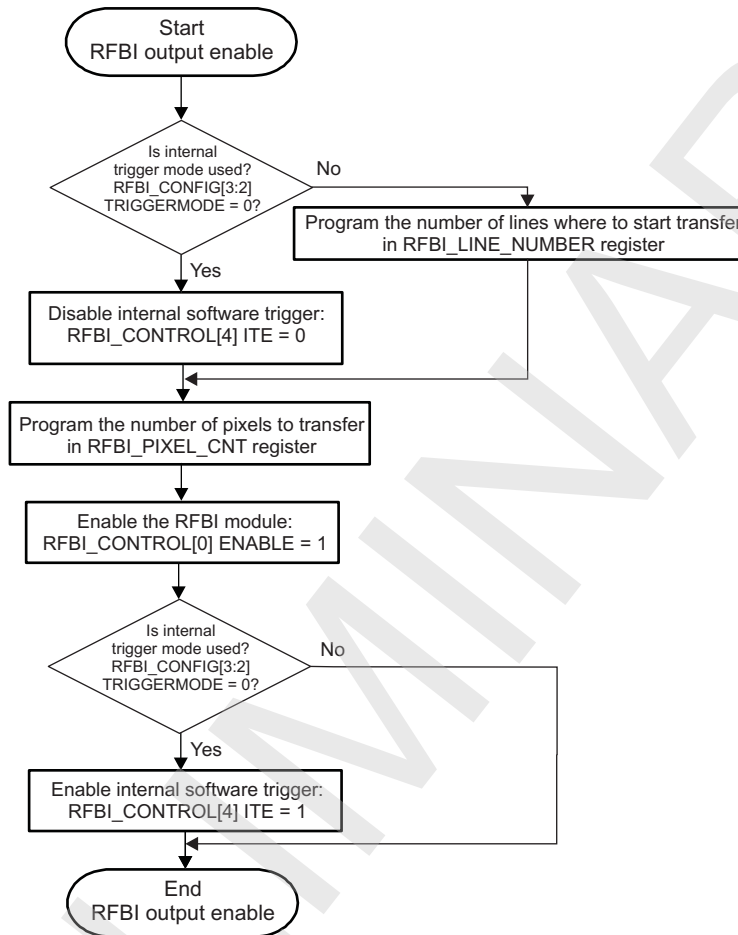
**Table 10-655. DISPC Output Enable**

Step	Register/Bit Field Programming Model	Value
Enable the video or GFX pipeline.	DISPC_VIDp_ATTRIBUTES[0] ENABLE	0x1
	DISPC_GFX_ATTRIBUTES[0] ENABLE	0x1
Enable the DISPC output.	DISPC_CONTROLo[0] LCDENABLE	0x1

10.5.5.1.2.1.5 Subsequence: RFBI Output Enable

This procedure enables the RFBI output. Figure 10-175 details the settings for this subsequence.

Figure 10-175. RFBI Output Enable Flow Chart



rfbi-022

Table 10-656 describes RFBI output enable.

Table 10-656. Register Call Summary for Subsequence: RFBI Output Enable

Register Name		
RFBI_CONFIG	RFBI_CONTROL	RFBI_LINE_NUMBER
RFBI_PIXEL_CNT		

Table 10-657 describes the RFBI behavior depending on the RFBI output enable.

Table 10-657. RFBI Behavior

RFBI_CONTROL[0] ENABLE Bit Value	RFBI Behavior
0	Slave port interconnect can write command/parameter/data and read data/status from the RFB. Slave port interconnect access can only be done to the active CS0.
1	The DISPC sends pixels to the RFB.

**NOTE:** The LCD output is disabled at the end of the transfer of the frame. The software must reenables the LCD output to generate a new frame by setting the DISPC\_CONTROLo[0] LCDENABLE bit to 1.

### CAUTION

The following registers must not be modified when the module is enabled (the RFBI\_CONTROL[0] ENABLE bit set to 1):

- RFBI\_CONTROL
- RFBI\_PIXEL\_CNT
- RFBI\_LINE\_NUMBER
- RFBI\_STATUS
- RFBI\_CONFIG
- RFBI\_ONOFF\_TIME
- RFBI\_CYCLE\_TIME
- RFBI\_DATA\_CYCLE<sub>i</sub> (i = 1 to 3)
- RFBI\_VSYNC\_WIDTH
- RFBI\_HSYNC\_WIDTH

**NOTE:** The RFBI\_STALL STALL signal is asserted when the RFBI is disabled. Through the slave port, pixels can be sent to the LCD panel only when the pixel count has reached 0x0.

#### 10.5.5.1.2.1.6 Subsequence: Detect Event

Table 10-658 describes the detect event.

**Table 10-658. Detect Event**

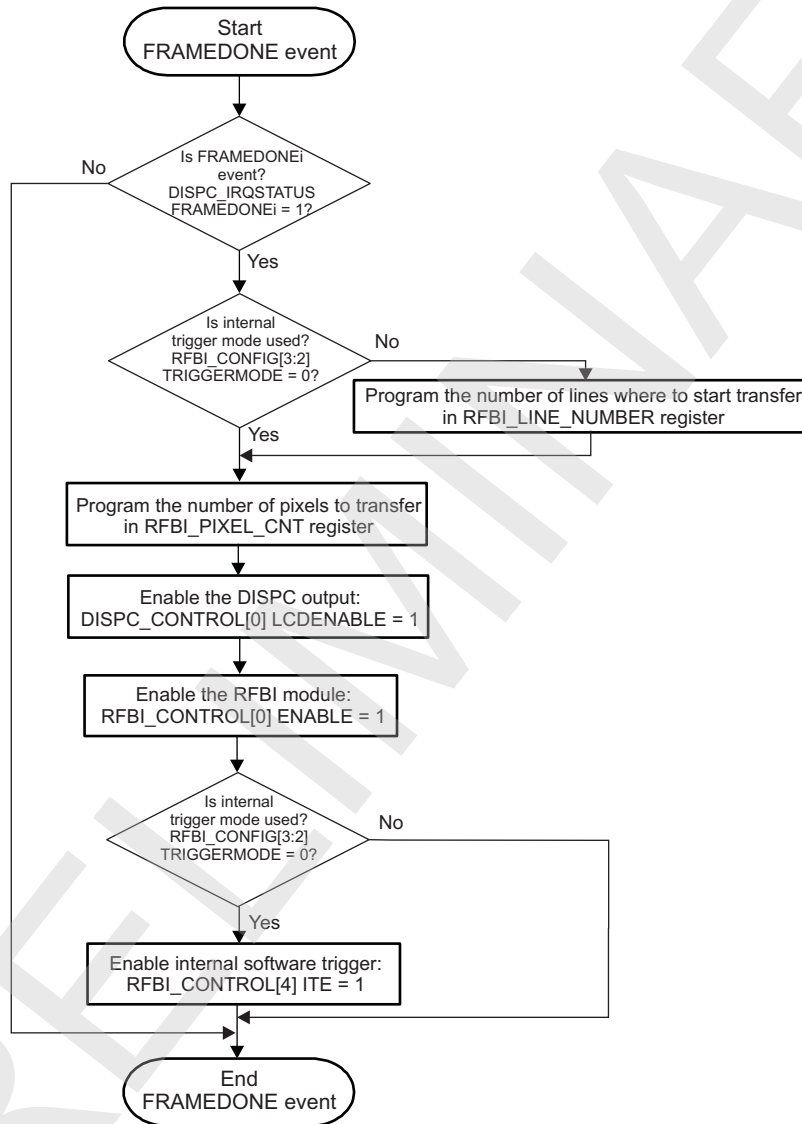
Step	Register/Bit Field Programming Model	Value
Wait until DISPC_IRQSTATUS != 0.	DISPC_IRQSTATUS	
Service event.	See Section 10.5.5.1.3, RFBI Events Servicing.	

10.5.5.1.3 RFBI Events Servicing

10.5.5.1.3.1 FRAMEDONE Interrupt Servicing

The FRAMEDONE event occurs each time a frame has been transferred to the LCD panel. [Figure 10-176](#) details how to serve this event.

Figure 10-176. FRAMEDONE Event Servicing Flow Chart



rfbi-023

Table 10-659 describes FRAMEDONE interrupt servicing.

Table 10-659. Register Call Summary for RFBI Events Servicing: FRAMEDONE Interrupt Servicing

Register Name		
DISPC_IRQSTATUS	RFBI_CONFIG	RFBI_LINE_NUMBER
RFBI_PIXEL_CNT	DISPC_CONTROL	RFBI_CONTROL

### 10.5.6 RFBI Hardware Status Features

Table 10-660 lists the RFBI hardware status features.

**Table 10-660. RFBI Hardware Status Features**

Feature	Type	Register/Bit Field/Observability Control	Description
Data pending	Status	RFBI_SYSTATUS[9] BUSYRFBIDATA	It is set to 1 when some data are pending to be processed from interconnect FIFO.
Slave port bus busy	Status	RFBI_SYSTATUS[9] BUSY	It is set to 1 when the slave port bus is busy and access of some RFBI registers ( <a href="#">RFBI_CMD</a> , <a href="#">RFBI_STATUS</a> , <a href="#">RFBI_PARAM</a> , and <a href="#">RFBI_READ</a> ) is stalled.



### 10.5.7 RFBI Register Manual

**CAUTION**

The main access to all RFBI registers is through the L3 interconnect.  
 The access through L4\_PER interconnect is provided for back software compatibility.

#### 10.5.7.1 RFBI Instance Summary

**Table 10-661. RFBI Instance Summary**

Module Name	L3 Base Address	L4_PER Base Address	Size
RFBI	0x5800 2000	0x4804 2000	4KB

#### 10.5.7.2 RFBI Registers

##### 10.5.7.2.1 RFBI Register Summary

**Table 10-662. RFBI Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	RFBI L3 Physical Address	RFBI L4_PER Physical Address
RFBI_REVISION	R	32	0x0000 0000	0x5800 2000	0x4804 2000
RFBI_SYSCONFIG	RW	32	0x0000 0010	0x5800 2010	0x4804 2010
RFBI_SYSSTATUS	R	32	0x0000 0014	0x5800 2014	0x4804 2014
RFBI_CONTROL	RW	32	0x0000 0040	0x5800 2040	0x4804 2040
RFBI_PIXEL_CNT	RW	32	0x0000 0044	0x5800 2044	0x4804 2044
RFBI_LINE_NUMBER	RW	32	0x0000 0048	0x5800 2048	0x4804 2048
RFBI_CMD	W	32	0x0000 004C	0x5800 204C	0x4804 204C
RFBI_PARAM	W	32	0x0000 0050	0x5800 2050	0x4804 2050
RFBI_DATA	W	32	0x0000 0054	0x5800 2054	0x4804 2054
RFBI_READ	RW	32	0x0000 0058	0x5800 2058	0x4804 2058
RFBI_STATUS	RW	32	0x0000 005C	0x5800 205C	0x4804 205C
RFBI_CONFIG	RW	32	0x0000 0060	0x5800 2060	0x4804 2060
RFBI_ONOFF_TIME	RW	32	0x0000 0064	0x5800 2064	0x4804 2064
RFBI_CYCLE_TIME	RW	32	0x0000 0068	0x5800 2068	0x4804 2068
RFBI_DATA_CYCLE <sub>Ei</sub> <sup>(1)</sup>	RW	32	0x0000 006C + (0x04 * i)	0x5800 206C + (0x04 * i)	0x4804 206C + (0x04 * i)
RFBI_VSYNC_WIDTH	RW	32	0x0000 0090	0x5800 2090	0x4804 2090
RFBI_HSYNC_WIDTH	RW	32	0x0000 0094	0x5800 2094	0x4804 2094

<sup>(1)</sup> i = 0 to 2

### 10.5.7.2.2 RFBI Register Description

**Table 10-663. RFBI\_REVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	RFBI
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .		
<b>Description</b>	This register contains the IP revision.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
REVISION																																

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See <sup>(1)</sup>

<sup>(1)</sup> Ti internal data

**Table 10-664. Register Call Summary for Register RFBI\_REVISION**

Remote Frame Buffer Interface

- [RFBI registers: \[0\]](#)

**Table 10-665. RFBI\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	RFBI
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .		
<b>Description</b>	This register controls various parameters of the slave port interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	RESERVED	SIDLEMODE	RESERVED	SOFTRESET	AUTOIDLE										

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Write 0's for future compatibility. Reads returns 0.	RO	0x00000000
6	RESERVED	Write 0's for future compatibility. Reads returns 0.	RO	0
5	RESERVED	Write 0's for future compatibility. Reads returns 0.	RO	0
4:3	SIDLEMODE	Slave interface power management, Idle req/ack control. 0x0: Force-idle. An idle request is acknowledged unconditionally. 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module.	RW	0x0
2	RESERVED	Write 0's for future compatibility. Reads returns 0.	RO	0
1	SOFTRESET	Software reset Sets this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0x0: Normal mode 0x1: The module is reset	RW	0

Bits	Field Name	Description	Type	Reset
0	AUTOIDLE	Internal clock gating strategy (RFBI_ICLK clock and DISPC clock) 0x0: RFBI_ICLK clock and DISPC clock are free-running 0x1: Automatic clock gating strategy is applied for the RFBI_ICLK clock and DISPC clock, based on the slave port interface and internal activity.	RW	1

**Table 10-666. Register Call Summary for Register RFBI\_SYSCONFIG**

Remote Frame Buffer Interface

- [Software Reset: \[0\] \[1\]](#)
- [Power Management: \[2\] \[3\]](#)
- [RFBI registers: \[4\] \[5\]](#)

**Table 10-667. RFBI\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	RFBI
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .		
<b>Description</b>	This register provides status information about the module, excluding the interrupt status information.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUSYRFBIDATA		BUSY		RESERVED								RESETDONE			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved for module-specific status information. Read returns 0.	RO	0x000000
9	BUSYRFBIDATA	Data are pending to be processed from interconnect FIFO. Read 0x0: No data pending Read 0x1: Some data are pending	R	0
8	BUSY	Slave port busy status bit Read 0x0: The access to the following register is not stalled: <a href="#">RFBI_CMD</a> , <a href="#">RFBI_STATUS</a> , <a href="#">RFBI_PARAM</a> , <a href="#">RFBI_READ</a> . Read 0x1: The access to any of the following registers is stalled: <a href="#">RFBI_CMD</a> , <a href="#">RFBI_STATUS</a> , <a href="#">RFBI_PARAM</a> , <a href="#">RFBI_READ</a> .	R	0
7:1	RESERVED	Reserved. Read returns 0.	RO	0x00
0	RESETDONE	Internal reset monitoring. It can be used to determine when a hardware reset is completed or when a software reset is completed (software has set <a href="#">RFBI_SYSCONFIG[0] SOFTRESET</a> to 1). Read 0x0: Internal module reset is ongoing Read 0x1: Reset completed	R	1

**Table 10-668. Register Call Summary for Register RFBI\_SYSSTATUS**

Remote Frame Buffer Interface

- [Software Reset: \[0\]](#)
- [RFBI Timing Generator: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [RFBI Low-Level Programming Models: \[10\] \[11\]](#)
- [RFBI registers: \[12\]](#)

**Table 10-669. RFBI\_CONTROL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	RFBI
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .		
<b>Description</b>	The register configures the RFBI module.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SMART_DMA_REQ	DISABLE_DMA_REQ	HIGHTHRESHOLD	ITE	CONFIGSELECT	RFBIMODE	ENABLE									

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0000000
8	SMART_DMA_REQ	Smart DMA request  0x0: The dmareq is asserted and deasserted depending on FIFO space even if Midlreq is high in smart-idle/no-idle mode and the entire burst gets error responses from the module.  0x1: The dmareq is deasserted after 2 clock cycles if it has been asserted for more than or equal to 2 clock cycles and Midlreq is high in smart-idle or no-idle mode. No more burst requests will be given even if the space is available in the FIFO.	RW	0
7	DISABLE_DMA_REQ	Disable DMA request  0x0: The dmareq is enabled and the signal is generated based on the space available and the request coming into the data register.  0x1: The dmareq is disabled and the signal is not generated at all based on space in FIFO. It stays high until the DISABLE DM AREQ is high even if there is space in FIFO to take requests.	RW	0
6:5	HIGHTHRESHOLD	Defines the FIFO high threshold used by hardware to assert DMA request. Used only if data written to <a href="#">RFBI_DATA</a> are sent using system DMA.  0x0: Size of the transfer of 4 words of 32-bit wide 0x1: Size of the transfer of 8 words of 32-bit wide 0x2: Size of the transfer of 16 words of 32-bit wide	RW	0x0
4	ITE	Internal Trigger  0x0: Hardware waits for ITE bit to be set if in internal trigger mode for the configuration in use.  0x1: User sets the ITE bit to start the transfer, when hardware takes into account the bit, the hardware resets it.	RW	0

Bits	Field Name	Description	Type	Reset
3:2	CONFIGSELECT	Select the CS and configuration 0x0: No CS selected 0x1: CS0 selected 0x2: Reserved 0x3: Reserved	RW	0x0
1	RFBIMODE	RFBI Mode 0x0: The RFBI mode is selected. 0x1: The RFBI mode is not selected.	RW	1
0	ENABLE	Enable/Disable flag 0x0: Disable the RFBI module. 0x1: Enable the RFBI module.	RW	0

**Table 10-670. Register Call Summary for Register RFBI\_CONTROL**

Remote Frame Buffer Interface

- [RFBI Parallel Interface \(MIPI DBI Protocol\): \[0\] \[1\]](#)
- [DMA Requests: \[2\]](#)
- [Video Port FIFO: \[3\] \[4\] \[5\] \[6\]](#)
- [Interconnect FIFO: \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [RFBI Output Parallel Modes: \[13\] \[14\]](#)
- [RFBI Timing Generator: \[15\] \[16\] \[17\]](#)
- [Trigger Detection Logic: \[18\] \[19\] \[20\] \[21\]](#)
- [RFBI Low-Level Programming Models: \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [RFBI registers: \[32\]](#)

**Table 10-671. RFBI\_PIXEL\_CNT**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	RFBI
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .		
<b>Description</b>	The register configures the RFBI pixel count value.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIXELCNT																															

Bits	Field Name	Description	Type	Reset
31:0	PIXELCNT	Pixel counter value The software indicates the number of pixels to transfer to the LCD panel frame buffer. The value is set when the module is disabled. During the transfer the hardware decrements the register when a pixel has been sent to the RFB.	RW	0x0000 0000

**Table 10-672. Register Call Summary for Register RFBI\_PIXEL\_CNT**

Remote Frame Buffer Interface

- [RFBI Output Parallel Modes: \[0\] \[1\]](#)
- [RFBI Low-Level Programming Models: \[2\] \[3\] \[4\]](#)
- [RFBI registers: \[5\]](#)

**Table 10-673. RFBI\_LINE\_NUMBER**

<b>Address Offset</b>	0x0000 0048		
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .	<b>Instance</b>	RFBI
<b>Description</b>	The register configures the number of lines to synchronize the beginning of the transfer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														LINENUMBER																	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x000000
10:0	LINENUMBER	Programmable line number Line number from 0 to 2047. Number of HSYNC after the VSYNC occurs before the beginning of the transfer.	RW	0x000

**Table 10-674. Register Call Summary for Register RFBI\_LINE\_NUMBER**

Remote Frame Buffer Interface

- [Trigger Detection Logic: \[0\]](#)
- [RFBI Low-Level Programming Models: \[1\] \[2\] \[3\]](#)
- [RFBI registers: \[4\] \[5\] \[6\]](#)

**Table 10-675. RFBI\_CMD**

<b>Address Offset</b>	0x0000 004C		
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .	<b>Instance</b>	RFBI
<b>Description</b>	The register configures the RFBI command.		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														CMD																	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x000000
7:0	CMD	Command Value	W	0x00

**Table 10-676. Register Call Summary for Register RFBI\_CMD**

Remote Frame Buffer Interface

- [RFBI Timing Generator: \[0\] \[1\]](#)
- [RFBI Low-Level Programming Models: \[2\]](#)
- [RFBI Hardware Status Features: \[3\]](#)
- [RFBI registers: \[4\] \[5\] \[6\]](#)

**Table 10-677. RFBI\_PARAM**

<b>Address Offset</b>	0x0000 0050	
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .	<b>Instance</b> RFBI
<b>Description</b>	The register configures the RFBI parameter.	
<b>Type</b>	W	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PARAM															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x000000
7:0	PARAM	Parameter value	W	0x00

**Table 10-678. Register Call Summary for Register RFBI\_PARAM**

Remote Frame Buffer Interface

- [RFBI Timing Generator: \[0\] \[1\]](#)
- [RFBI Hardware Status Features: \[2\]](#)
- [RFBI registers: \[3\] \[4\] \[5\]](#)

**Table 10-679. RFBI\_DATA**

<b>Address Offset</b>	0x0000 0054	
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .	<b>Instance</b> RFBI
<b>Description</b>	The register configures the RFBI data.	
<b>Type</b>	W	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data value 12/16/18/24/2x16 bit value depending on: DataType [11:0] 12-bit DataType [15:0] 16-bit DataType [17:0] 18-bit DataType [23:0] 24-bit DataType [31:0] 2x16-bit	W	0x0000 0000

**Table 10-680. Register Call Summary for Register RFBI\_DATA**

Remote Frame Buffer Interface

- [Remote Frame Buffer Interface Overview: \[0\] \[1\]](#)
- [Interconnect FIFO: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [RFBI Timing Generator: \[8\] \[9\] \[10\] \[11\]](#)
- [RFBI registers: \[12\] \[13\]](#)

**Table 10-681. RFBI\_READ**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .	<b>Instance</b>	RFBI
<b>Description</b>	The register configures the RFBI read.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																READ															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x000000
7:0	READ	Read value	RW	0x00

**Table 10-682. Register Call Summary for Register RFBI\_READ**

Remote Frame Buffer Interface

- [RFBI Timing Generator: \[0\] \[1\]](#)
- [RFBI Low-Level Programming Models: \[2\] \[3\]](#)
- [RFBI Hardware Status Features: \[4\]](#)
- [RFBI registers: \[5\] \[6\] \[7\]](#)

**Table 10-683. RFBI\_STATUS**

<b>Address Offset</b>	0x0000 005C		
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .	<b>Instance</b>	RFBI
<b>Description</b>	The register configures the RFBI status.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STATUS															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x000000
7:0	STATUS	Status value	RW	0x00

**Table 10-684. Register Call Summary for Register RFBI\_STATUS**

Remote Frame Buffer Interface

- [RFBI Timing Generator: \[0\] \[1\]](#)
- [RFBI Low-Level Programming Models: \[2\] \[3\] \[4\]](#)
- [RFBI Hardware Status Features: \[5\]](#)
- [RFBI registers: \[6\] \[7\] \[8\]](#)



**Table 10-685. RFBI\_CONFIG**

<b>Address Offset</b>	0x0000 0060
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .
<b>Description</b>	The control register sets the configuration for the LCD 0 and LCD 1.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HSYNCPOLARITY	TE_VSYNC_POLARITY	CSPOLARITY	WEPOLARITY	REPOLARITY	A0POLARITY	RESERVED	UNUSEDBITS	CYCLEFORMAT	PORTFORMAT	DATATYPE	TIMEGRANULARITY	TRIGGERMODE	PARALLELMODE										

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x000
21	HSYNCPOLARITY	HSYNC polarity 0x0: HSYNC active low 0x1: HSYNC active high	RW	1
20	TE_VSYNC_POLARITY	TE or VSYNC polarity 0x0: active low 0x1: active high	RW	1
19	CSPOLARITY	CS polarity 0x0: CS active low 0x1: CS active high	RW	0
18	WEPOLARITY	WE polarity 0x0: Active low 0x1: Active high	RW	0
17	REPOLARITY	RE polarity 0x0: Active low 0x1: Active high	RW	0
16	A0POLARITY	A0 polarity 0x0: A0 active low 0x1: A0 active high	RW	1
15:13	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0
12:11	UNUSEDBITS	State of unused bits 0x0: Low level (0) 0x1: High level (1) 0x2: Unchanged from previous state	RW	0x0
10:9	CYCLEFORMAT	Cycle format 0x0: 1 cycle for 1 pixel 0x1: 2 cycles for 1 pixel 0x2: 3 cycles for 1 pixel 0x3: 3 cycles for 2 pixels	RW	0x0

Bits	Field Name	Description	Type	Reset
8:7	PORTFORMAT	Slave port write access format 0x0: 1 pixel per slave port access to the register data 0x2: 2 pixels per slave port access to the register data with first pixel at the position [15:0] 0x3: 2 pixels per slave port access to the register data with first pixel at the position [31:16]	RW	0x0
6:5	DATATYPE	Data type from the DISPC and slave port 0x0: 12-bit 0x1: 16-bit 0x2: 18-bit 0x3: 24-bit	RW	0x0
4	TIMEGRANULARITY	Multiplies signal timing latencies by 2 0x0: x2 latencies disabled 0x1: x2 latencies enabled	RW	0
3:2	TRIGGERMODE	Trigger mode 0x0: 00 Internal trigger mode (ITE bit mode) 0x1: External trigger mode (Tearing Effect Signal, rfb_i_tevsync0 with programmable line counter defined in <a href="#">RFBI_LINE_NUMBER</a> register) 0x2: External trigger mode (rfbi_tevsync0/rfbi_hsync0 with programmable line counter defined in <a href="#">RFBI_LINE_NUMBER</a> register)	RW	0x0
1:0	PARALLELMODE	Parallel mode 0x0: 8-bit parallel output interface selected 0x1: 9-bit parallel output interface selected 0x2: Reserved 0x3: 16-bit parallel output interface selected	RW	0x0

**Table 10-686. Register Call Summary for Register RFBI\_CONFIG**

Remote Frame Buffer Interface

- [RFBI Parallel Interface \(MIPI DBI Protocol\): \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Video Port FIFO: \[7\] \[8\]](#)
- [RFBI Output Parallel Modes: \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [RFBI Timing Generator: \[14\] \[15\] \[16\] \[17\]](#)
- [Trigger Detection Logic: \[18\] \[19\]](#)
- [RFBI Low-Level Programming Models: \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [RFBI registers: \[32\]](#)

**Table 10-687. RFBI\_ONOFF\_TIME**

<b>Address Offset</b>	0x0000 0064																															
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .										<b>Instance</b>						RFBI															
<b>Description</b>	The control register configures the RFBI timings for the LCD 0 and LCD 1.																															
<b>Type</b>	RW																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>RESERVED</b>																																

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0
29:24	REOFFTIME	Read enable deassertion time from start access time. Number of RFBI_ICLK cycles	RW	0x00
23:20	REONTIME	Read enable assertion time from start access time. Number of RFBI_ICLK cycles	RW	0x0
19:14	WEOFFTIME	Write enable deassertion time from start access time. Number of RFBI_ICLK cycles	RW	0x00
13:10	WEONTIME	Write enable assertion time from start access time. Number of RFBI_ICLK cycles	RW	0x0
9:4	CSOFFTIME	CS deassertion time from start access time. Number of RFBI_ICLK cycles	RW	0x00
3:0	CSONTIME	CS assertion time from start access time. Number of RFBI_ICLK cycles	RW	0x0

**Table 10-688. Register Call Summary for Register RFBI\_ONOFF\_TIME**

Remote Frame Buffer Interface

- [Transaction Timing Diagrams: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [RFBI Timing Generator: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\]](#)
- [RFBI Low-Level Programming Models: \[21\] \[22\]](#)
- [RFBI registers: \[23\]](#)

**Table 10-689. RFBI\_CYCLE\_TIME**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	RFBI
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .		
<b>Description</b>	The control register configures the RFBI timings for the LCD 0 and LCD 1.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				ACCESSTIME				WRENABLE	WWENABLE	RRENABLE	RWENABLE	CSPULSEWIDTH					RECYCLETIME				WECYCLETIME										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0
27:22	ACCESSTIME	Access time number of RFBI_ICLK cycles	RW	0x00
21	WRENABLE	Write-to-read pulse width enable (same CS) 0x0: CSPULSEWIDTH does not apply on Write-to-Read access 0x1: CSPULSEWIDTH applies on Write-to-Read access	RW	0
20	WWENABLE	Write-to-write pulse width enable (same CS) 0x0: CSPULSEWIDTH does not apply on Write-to-Write access 0x1: CSPULSEWIDTH applies on Write-to-Write access	RW	0
19	RRENABLE	Read-to-read pulse width enable (same CS) 0x0: CSPULSEWIDTH does not apply on Read-to-Read access 0x1: CSPULSEWIDTH applies on Read-to-Read access	RW	0
18	RWENABLE	Read-to-write pulse width enable (same CS): 0x0: CSPULSEWIDTH does not apply on Read-to-Write access 0x1: CSPULSEWIDTH applies on Read-to-Write access	RW	0
17:12	CSPULSEWIDTH	CS pulse width number of RFBI_ICLK cycles	RW	0x00

Bits	Field Name	Description	Type	Reset
11:6	RECYCLETIME	RE cycle time number of RFBI_ICLK cycles	RW	0x00
5:0	WECYCLETIME	WE cycle time number of RFBI_ICLK cycles	RW	0x00

**Table 10-690. Register Call Summary for Register RFBI\_CYCLE\_TIME**

Remote Frame Buffer Interface

- [Transaction Timing Diagrams: \[0\] \[1\] \[2\]](#)
- [RFBI Timing Generator: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [RFBI Low-Level Programming Models: \[16\] \[17\]](#)
- [RFBI registers: \[18\]](#)

**Table 10-691. RFBI\_DATA\_CYCLEI**

<b>Address Offset</b>	0x0000 006C + (0x04 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .	<b>Instance</b>	RFBI
<b>Description</b>	The control register configures the RFBI data format for ith cycle (i = 1 for first cycle, i = 2 for second cycle, i = 3 for third cycle).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				BITALIGNMENTPIXEL2				RESERVED				NBBITSPIXEL2				RESERVED				BITALIGNMENTPIXEL1				RESERVED				NBBITSPIXEL1			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0
27:24	BITALIGNMENTPIXEL2	Alignment of the bits from pixel 2 on the output interface	RW	0x0
23:21	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0
20:16	NBBITSPIXEL2	Number of bits from the pixel 2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00
15:12	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0
11:8	BITALIGNMENTPIXEL1	Alignment of the bits from pixel 1 on the output interface	RW	0x0
7:5	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0
4:0	NBBITSPIXEL1	Number of bits from the pixel 1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x00

**Table 10-692. Register Call Summary for Register RFBI\_DATA\_CYCLEI**

Remote Frame Buffer Interface

- [RFBI Output Parallel Modes: \[0\]](#)
- [RFBI Low-Level Programming Models: \[1\]](#)
- [RFBI registers: \[2\]](#)

**Table 10-693. RFBI\_VSYNC\_WIDTH**

<b>Address Offset</b>	0x0000 0090		
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .	<b>Instance</b>	RFBI
<b>Description</b>	The register configures the RFBI VSYNC minimum pulse width.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MINVSYNCPULSEWIDTH															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0000
15:0	MINVSYNCPULSEWIDTH	Programmable minimum VSYNC pulse width Minimum VSYNC pulse width from 0 to 65535. Number of RFBI_ICLK clock cycles to determine when VSYNC pulse occurs. The values 0 and 1 are invalid.	RW	0x0000

**Table 10-694. Register Call Summary for Register RFBI\_VSYNC\_WIDTH**

- Remote Frame Buffer Interface
- [RFBI Parallel Interface \(MIPI DBI Protocol\): \[0\]](#)
  - [Trigger Detection Logic: \[1\] \[2\]](#)
  - [RFBI Low-Level Programming Models: \[3\] \[4\]](#)
  - [RFBI registers: \[5\]](#)

**Table 10-695. RFBI\_HSYNC\_WIDTH**

<b>Address Offset</b>	0x0000 0094		
<b>Physical Address</b>	See <a href="#">Table 10-662</a> .	<b>Instance</b>	RFBI
<b>Description</b>	The register configures the RFBI HSYNC minimum pulse width.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MINHSYNCPULSEWIDTH															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Reads return 0.	RO	0x0000
15:0	MINHSYNCPULSEWIDTH	Programmable minimum HSYNC pulse width minimum HSYNC pulse width from 0 to 65535. Number of RFBI_ICLK clock cycles to determine when HSYNC pulse occurs. The values 0 and 1 are invalid.	RW	0x0000

**Table 10-696. Register Call Summary for Register RFBI\_HSYNC\_WIDTH**

- Remote Frame Buffer Interface
- [RFBI Parallel Interface \(MIPI DBI Protocol\): \[0\]](#)
  - [Trigger Detection Logic: \[1\] \[2\]](#)
  - [RFBI Low-Level Programming Models: \[3\] \[4\]](#)
  - [RFBI registers: \[5\]](#)

## 10.6 Video Encoder

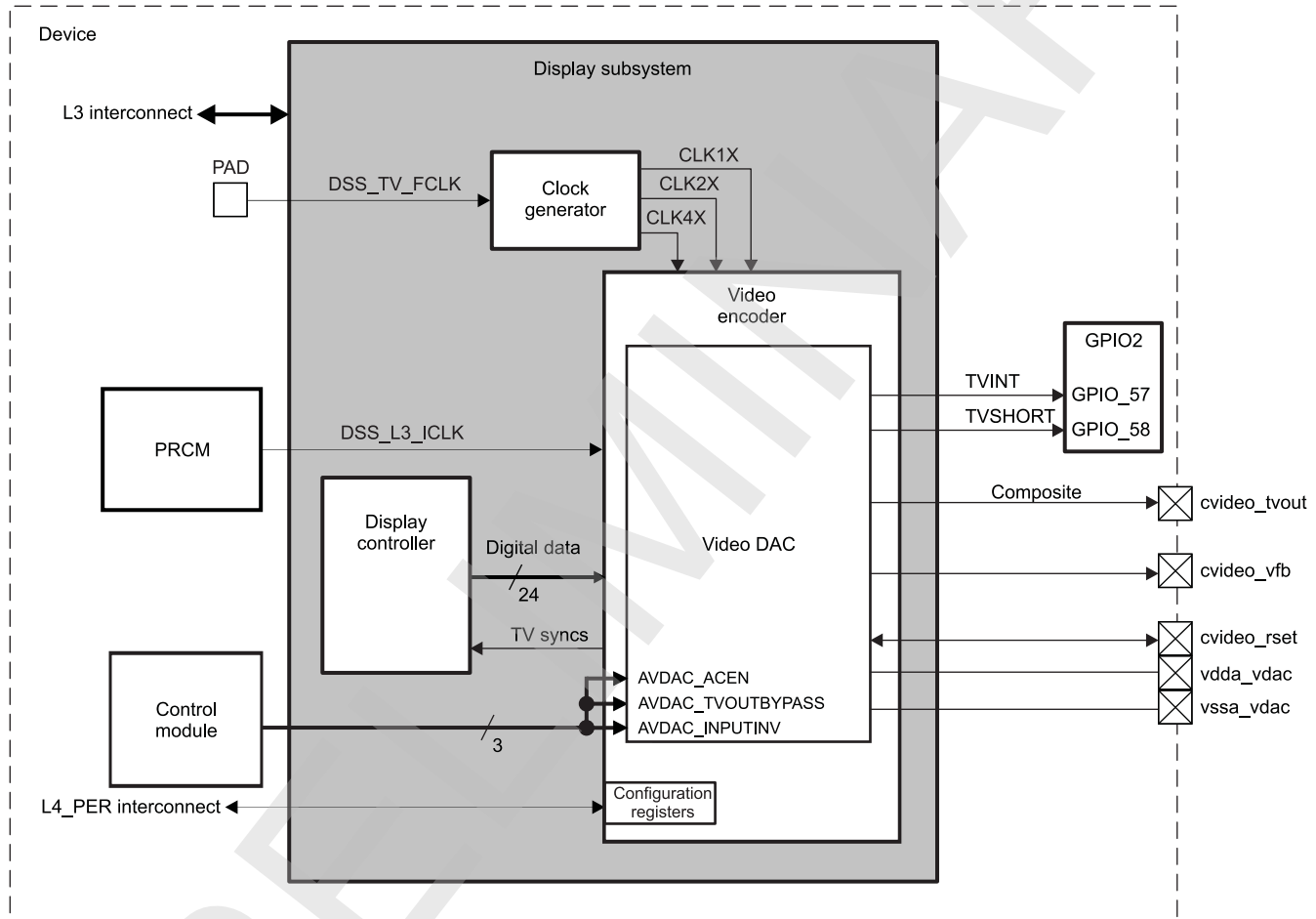
This section describes the video encoder for the device.

### 10.6.1 Video Encoder Overview

The display subsystem provides the logic to display a video frame from the memory frame buffer (either SDRAM or SRAM) on a TV set.

Figure 10-177 shows a block diagram of the display subsystem.

**Figure 10-177. Video Encoder Overview**



Venc001

The video encoder includes the following features:

- Supports the following standards for composite video (CVBS) :
  - NTSC-J, M
  - PAL-B, D, G, H, I
  - PAL-M
- Input data interface compatible with the following protocols:
  - 24-bit input bus compatible with external sync
  - RGB 4:4:4
- 10-bit interface for internal digital-to-analog converter (DAC) that supports:
  - 1.1V digital power supply, 1.8V analog power supply
  - 10-bit resolution
  - DNL within 1 LSB and INL within 1 LSB

- Sample rate of up to 60 megasamples per second (MSPS)
- Supports composite video standart output
- Supports TVOUT buffer bypass mode (DAC only mode)
- Full-scale voltage output: 1.2 Vpp with a 75-Ω load
- Internal TV detect feature
- Signal-to-noise ratio (SNR) is 54 dB (taking into account the ac coupling)
- Suitable for low-power wireless applications
- Power-down mode with less than 12-μA standby current
- Differential gain error and differential phase error: within 3% and 1°, respectively
- TV output data supports ITU-R BT 470-7 recommendation standard for consumer market
- Master clock input 13.5 MHz, 27 MHz (supports ITU-R 601 sampling for NTSC/PAL), and 54 MHz
- Programmable horizontal sync, vertical timing, and waveforms
- Programmable subcarrier frequency and SCH
- Internal test pattern generation (color bar, flat field, color burst)
- 2x/4x oversampling
- TV detection gating pulse generation
- Supports square pixel sampling (NTSC: 12.27 MHz, 24.54 MHz, 49.09 MHz PAL: 14.75 MHz, 29.5 MHz, 59 MHz)

**CAUTION**

In square pixel mode, an external clock generator is required to provide sampling frequencies.

### 10.6.2 Video Encoder Environment

To display on a TV, the path used is as follow:

- Display controller
- Video encoder
- 10-bit DAC with video amplifiers

The display controller module receives synchronization signals from the video encoder and synchronously sends pixel data to the video encoder with these signals. The digital output of the display controller is always a 24-bit RGB value based on a pixel request from the video encoder. For more information on the display controller settings and features, see [Section 10.2, Display Controller](#).

The video encoder converts RGB video signals to conform to the NTSC/PAL standard analog video. The video encoder includes an integrated synchronization signal generator and a video digital-to-analog converter (DAC) with video amplifiers, data manager, luma stage, chroma stage, modulator, and a control interface.

The video encoder also provides the synchronization signals to the display controller: VSYNC, active VIDEO (AVID), and field ID (FID).

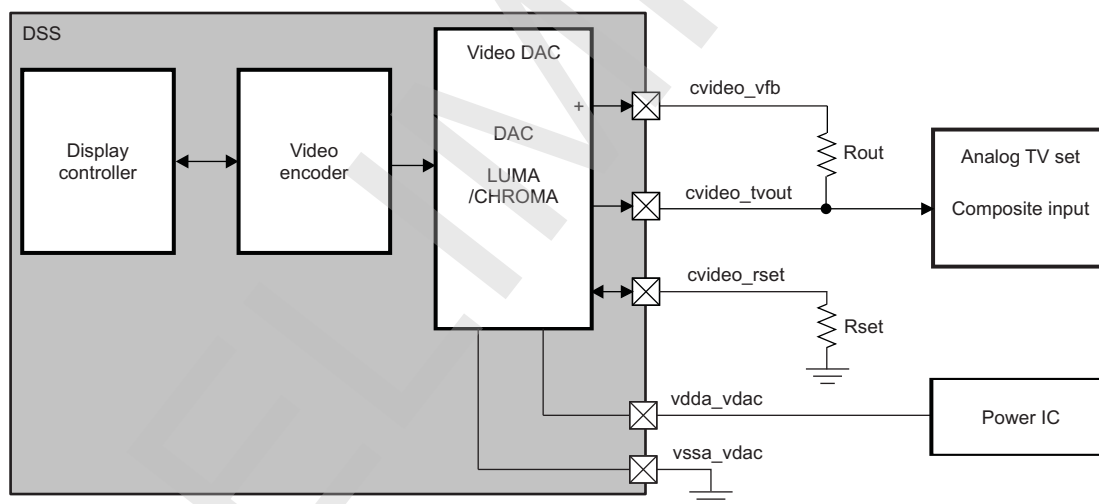
---

**NOTE:** To enhance the TV color display, it is highly recommended to set the DSS\_CONTROL[4] DAC\_DEMEN bit.

---

Figure 10-178 through Figure 10-180 show the TV display interface in composite mode.

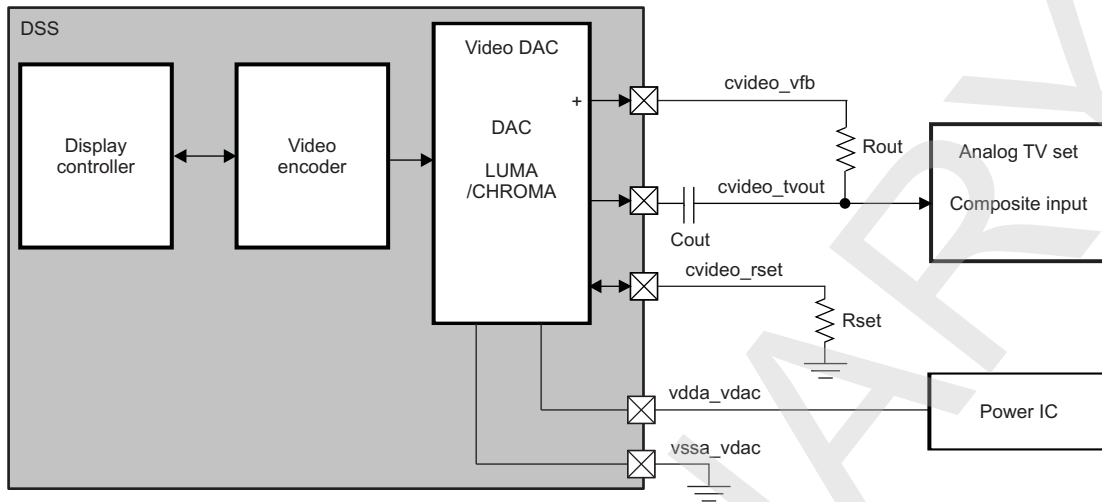
**Figure 10-178. Video Encoder Environment, Normal Mode DC-Coupling**



Venc-003

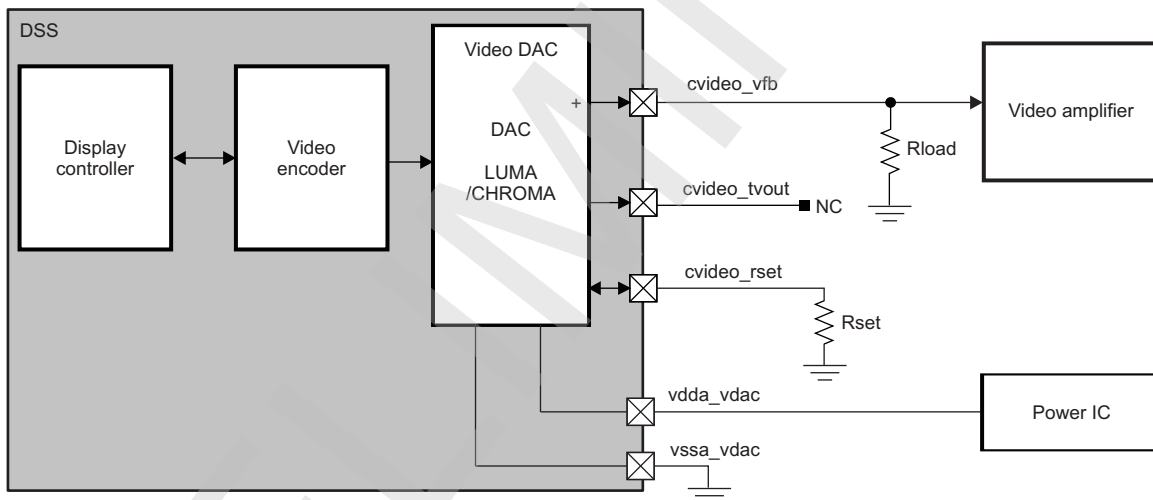


Figure 10-179. Video Encoder Environment, Normal Mode AC-Coupling



venc-015

Figure 10-180. Video Encoder Environment, Bypass Mode



venc-016

Table 10-697 describes the interface signals to/from the TV set for TV display support.

Table 10-697. TV Display Interface Pins

Pin Name	Type <sup>(1)</sup>	Description
cvideo_tvout	O	Analog luma or composite video output. An external resistor (Rout) is connected between this node and the <i>cvideo_vfb</i> pin. The nominal value of Rout is 2700 Ω. This is the output node that drives the load (75 Ω). In AC-coupling mode, a series capacitor (Cout) must be connected. The nominal value for Cout is ≥ 200 μF.
cvideo_vfb	O	In Normal mode (video DAC+Buffer), it acts as the buffer feedback node. An external resistor (Rout) shall be connected between this node and <i>cvideo_tvout</i> pin. In TVOUT Bypass mode, it acts as the video DAC output. A load resistor (Rload) shall be connected to the pin to ground. The nominal value of Rload is 1500 Ω.

<sup>(1)</sup> I = input; O = output; power = power pin

**Table 10-697. TV Display Interface Pins (continued)**

Pin Name	Type <sup>(1)</sup>	Description
cvideo_rset	I/O	External resistor (Rset) shall be connected to this pin to set the reference current of the video DAC. The value of the resistor depends on the mode of operation (Normal or TVOUT Bypass). In Normal mode, the nominal value for Rset is 4700 $\Omega$ . In TVOUT Bypass mode, the nominal value for Rset is 10000 $\Omega$ .
vdda_vdac	Power	Analog supply voltage for the video DAC
vssa_vdac	Power	Analog ground for the video DAC

**CAUTION**

- *cvideo\_tvout* is a very high-frequency analog signal and must be routed with extreme care. As a result, the path of this signal must be as short as possible, and as isolated as possible from other interfering signals.
- During board design, the onboard traces and parasites must be matched for the channel. *cvideo\_vfb* pin is the most sensitive pins in the TV out system. Low onboard resistance is required for the traces that connect the Rout to the *cvideo\_vfb* and *cvideo\_tvout* pins. The resistance on those trace affects output impedance matching. Therefore, Rout resistor is suggested to be placed as close as possible to the device pins. The onboard traces lead to the TV OUT pin must have a characteristic impedance of 75 Ohm starting from the closest possible place to the device pin output.
- If the TV output is not used, the analog pins *cvideo\_rset* and *vssa\_dac* must be grounded. To avoid current leakage, the following bits must be set to 0:
  - DSS.DSS\_CONTROL[5] DAC\_POWERDN\_BGZ
  - VENC\_OUTPUT\_CONTROL[1] COMPOSITE\_ENABLE
  - PRM.CM\_DSS\_DSS\_CLKCTRL[11] OPTFCLKEN\_TV\_FCLK
  - CONTROL.CONTROL\_AVDAC[30] AVDAC\_TVOUTBYPASS

Table 10-698 lists the typical values of the external components for different modes of the TV display interface.

**Table 10-698. Typical Values for Rout, Rset, Rload, and Cout**

	Composite, DC-Coupled	Composite, AC-Coupled	Bypass Mode	Unit
Rout	2700	2700	N/A	$\Omega$
Rset	4700	6800	10,000	$\Omega$
Rload	N/A	N/A	1500	$\Omega$
Cout	N/A	$\geq 220$	N/A	$\mu\text{F}$

### 10.6.3 Video Encoder Integration

This section describes the integration of the display subsystem and details clocks, resets, hardware requests, and power modes.

Figure 10-181 shows the integration of the display subsystem in the device.

Figure 10-181. Video Encoder Integration

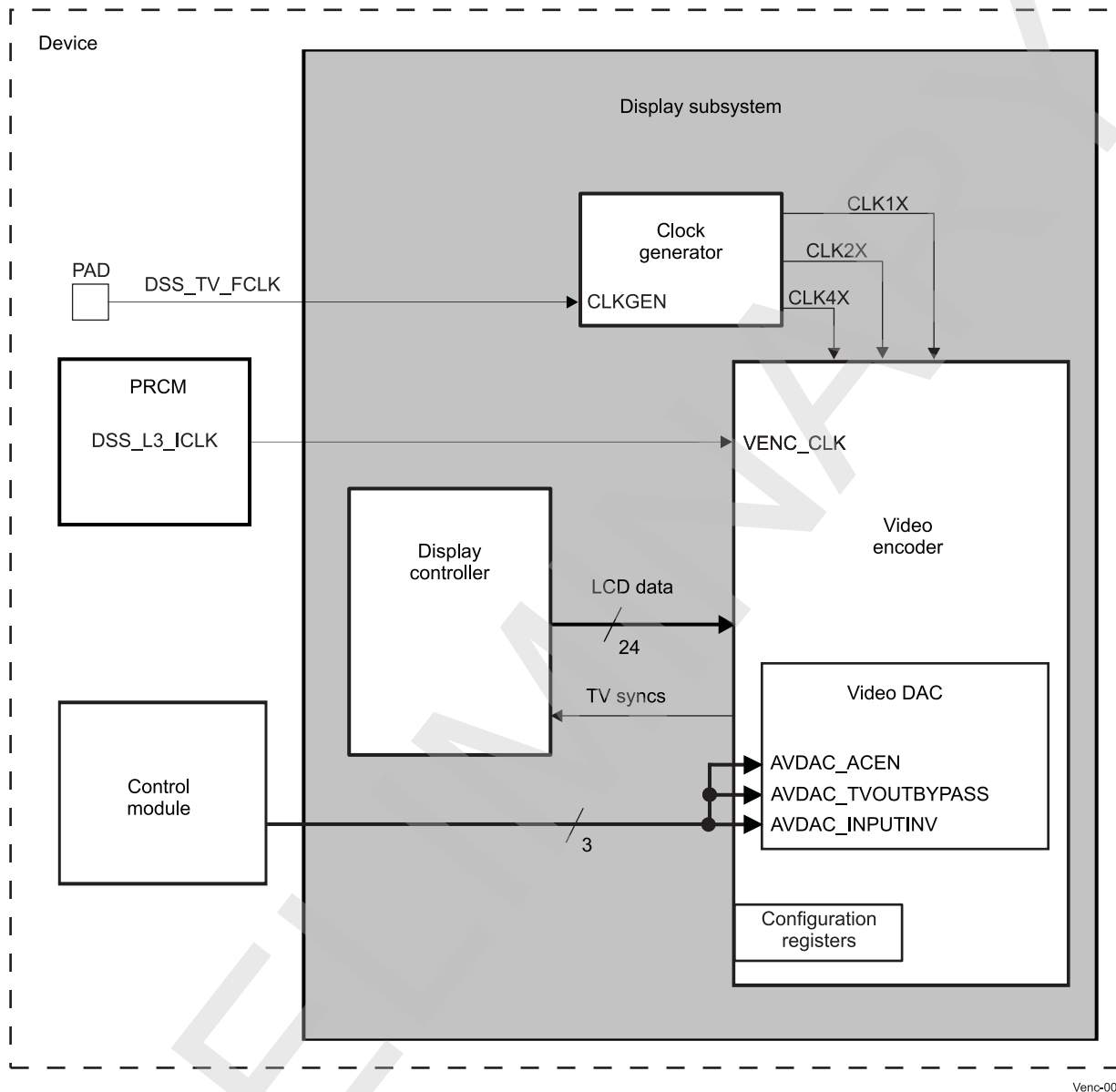


Table 10-699 describes the different clocks with their possible frequency values.

Table 10-699. Clocks

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Video encoder	VENC_CLK	DSS_L3_ICLK	PRCM	For information about PRCM clock gating and management, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Clock generator	CLKGEN	DSS_TV_FCLK	PAD	Functional clock provided on pad. Controlled by the PRCM module.
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description

**Table 10-699. Clocks (continued)**

<b>Video encoder DAC</b>	DSS_RST	DSS_RST	PRCM	Reset signal of the display subsystem power domain, from the PRCM module.
--------------------------	---------	---------	------	---

- Video encoder clock:  
The CLKGEN is divided into three balanced clocks, depending on the clock mode selected (see [Table 10-700](#)).

**Table 10-700. Digital Clock Division for the Video Encoder**

Clock Output	Clock Mode	
	Clock mode 0	Clock mode 1
CLK4X	DSS_TV_FCLK	DSS_TV_FCLK or 0 (gated)
CLK2X	DSS_TV_FCLK/2	DSS_TV_FCLK
CLK1X	DSS_TV_FCLK/4	DSS_TV_FCLK/2

**NOTE:** Clock mode 1 can be used for power-saving purposes, or if a 27-MHz external clock is provided to the video encoder.

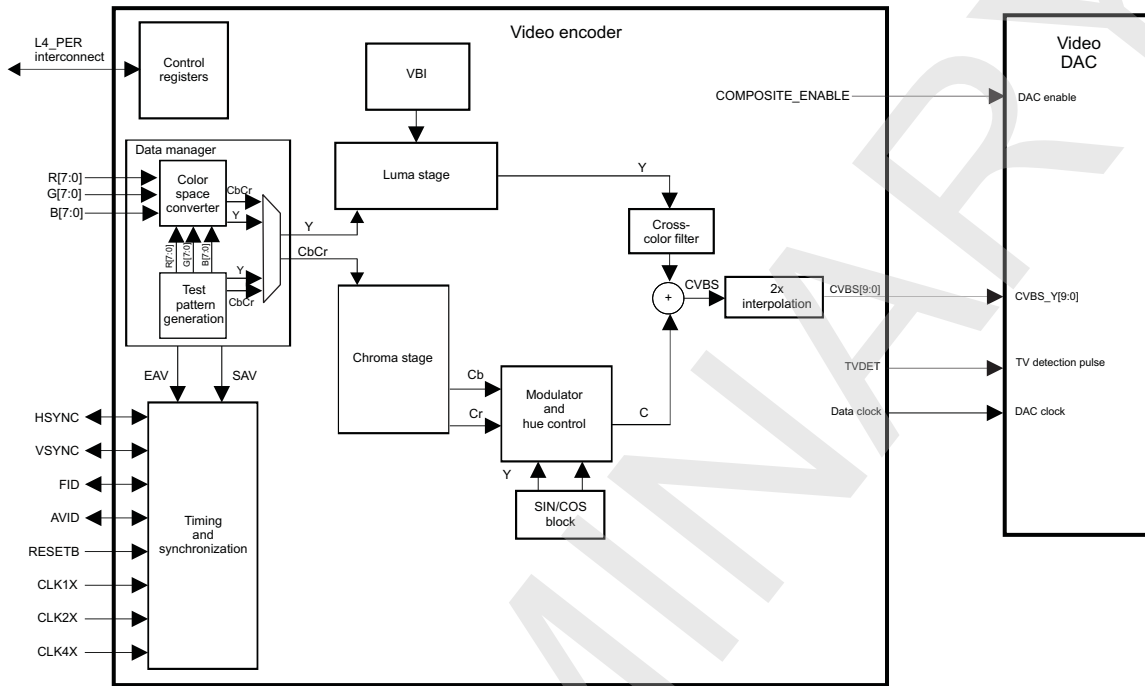
**NOTE:** The DSS\_TV\_FCLK clock is sourced from the `fref_slicer_in` input pin. To enable DSS\_TV\_FCLK, it must be first ungated through the `PRCM.ALTCLKSRC[3] ENABLE_EXT` bit and then enabled in the `PRCM.CM_DSS_DSS_CLKCTRL[11] OPTFCLKEN_TV_FCLK` bit. An external clock generator must be connected to the `fref_slicer_in` pin. In this case, a 54-MHz clock is needed for PAL or NTSC 601, a 49.04-MHz clock is needed for NTSC square pixel, and a 59-MHz clock is needed for PAL square pixel.

- Video DAC clocks:  
The video data are latched on the positive edge of the DSS\_TV\_FCLK clock.

### 10.6.4 Video Encoder Functional Description

Figure 10-182 shows an overview of the video encoder architecture.

Figure 10-182. Video Encoder Architecture Overview



#### 10.6.4.1 Data Manager

The data manager allows user to select two sources of data inputs:

- Data coming from the Display Controller.
- Data coming from the test pattern generation.

The source is selected by programming SVDS field [VENC\\_F\\_CONTROL\[7:6\]](#).

##### 10.6.4.1.1 Color Space Converter

In the display subsystem, the input format from the display controller is always 24-bit RGB coming from the display controller or the test pattern generation. The RGB-to-YCbCr color space converter converts the 24-bit RGB pixel data into 16-bit YCbCr 4:2:2 data.

##### 10.6.4.1.2 Test Pattern Generation

For diagnostic purposes, the data manager can be forced to output 100/100 color bar RGB/YCbCr data. To use the test pattern generation, SVDS [VENC\\_F\\_CONTROL\[7:6\]](#) field shall be set to 0x1. If FMT field [VENC\\_F\\_CONTROL\[1:0\]](#) is set to 0x0 then the 24-bits RGB test pattern data will be converted into 16-bit YCbCr 4:2:2 data by the Color Space Converter. Otherwise if FMT field [VENC\\_F\\_CONTROL\[1:0\]](#) is set to 0x2 then 16-bit YCbCr 4:2:2 test pattern data are outputted directly, the color space converted is bypassed. [Table 10-701](#) resume the test pattern generator datas.

Table 10-701. 100/100 Color Bar Table

COLOR	R	G	B	Y	Cb	Cr
White	255	255	255	235	128	128
Yellow	255	255	0	210	16	146
Cyan	0	255	255	170	166	16
Green	0	255	0	145	54	34

**Table 10-701. 100/100 Color Bar Table (continued)**

Magenta	255	0	255	106	202	222
Red	255	0	0	81	90	240
Blue	0	0	255	41	240	110
Black	0	0	0	16	128	128

#### 10.6.4.2 Luma Stage

The luma stage includes a luma pipeline delay, luma shaping, 2x interpolation filter, and luma variable delay. The luma pipeline delay block is used to match luma path length to chroma path length. In the luma gain shaper, a programmable gain is first applied to the luminance data output. The luminance gain is defined by the [VENC\\_GAIN\\_Y](#) register. Horizontal sync, vertical sync, and setup insertion are then performed.

Black level and blank level are programmable through the [VENC\\_BLACK\\_LEVEL](#) and [VENC\\_BLANK\\_LEVEL](#) registers. All the transition edges of the luminance signal, such as sync edges and active video edges, are properly shaped and filtered to keep the bandwidth within the standards.

After all required components of the luminance signal are added, the resulting signal is low-passed and interpolated to 2x-pixel rate. This 2x interpolation simplifies the external analog reconstruction filter design and improves the signal-to-noise ratio.

#### 10.6.4.3 Chroma Stage

The chroma stage includes a low-pass filter, first-stage 2x interpolation, chroma gain shaper, and second-stage 2x interpolation. A pair of programmable gains adjusts the time-multiplexed U/V signal. The gains for U and V are independently controlled by the [VENC\\_GAIN\\_U](#) and [VENC\\_GAIN\\_V](#) register bits.

#### 10.6.4.4 Subcarrier and Burst Generation

The encoder uses a 32-bit subcarrier increment to synthesize the subcarrier. The value of the subcarrier increments required to generate the desired subcarrier frequency for NTSC and PAL format is found by:

$$S\_CARR = \text{ROUND} ([F_{sc}/F_{CLK2X}] \times 2^{32})$$

where:

$F_{sc}$  = Frequency of the subcarrier

$F_{CLK2X}$  = Frequency of the internal video encoder

The [VENC\\_S\\_CARR](#) register controls the subcarrier frequency. The [VENC\\_C\\_PHASE](#) register controls the phase of the subcarrier. The phase of the color subcarrier is reset to [VENC\\_C\\_PHASE](#). [Table 10-702](#) presents the [VENC\\_S\\_CARR](#) register values depending the standard and pixel type used.

**Table 10-702. VENC\_S\_CARR Register Recommended Values**

Standard	Pixel Type	Subcarrier Frequency (Fsc) (MHz)	Fclkenc (MHz)	VENC_S_CARR Register Value (hexa)
NTSC-M, J	ITU-R601	3.579545	27	0x21F07C1F
PAL-M	ITU-R601	3.5756083125	27	0x21E6EFE3
PAL-B, D, G, H, I	ITU-R601	4.43361875	27	0x2A098ACB
NTSC-M, J	Square pixel	3.579545	24.5454 <sup>(1)</sup>	0x25555555
PAL-M	Square pixel	3.579561149	24.5454 <sup>(1)</sup>	0x1F15C01E
PAL-B, D, G, H, I	Square pixel	4.43361875	29.50 <sup>(1)</sup>	0x26798C0C

<sup>(1)</sup> In square pixel mode, an external clock generator is needed to provide sampling frequencies (49.09 MHz for NTSC square pixel or 59 MHz for PAL square pixel).

The color subcarrier reset has four modes:

- No reset
- Reset every two lines
- Reset every two fields
- Reset every eight fields

The [VENC\\_C\\_PHASE](#) register can be used to adjust the SCH (subcarrier to horizontal sync phase). The [VENC\\_BSTAMP\\_WSS\\_DATA\[6:0\]](#) BSTAP bit field sets the amplitude of the color burst.

Phase alternation line refers to the encoding scheme in which the subcarrier alternates between two phases every scan line. Setting to 0x1, the [VENC\\_M\\_CONTROL\[1\]](#) PAL bit enables phase alternation line encoding. Otherwise, a normal subcarrier is generated. Two possible alternation sequences are possible by setting the [VENC\\_M\\_CONTROL\[5\]](#) PALPHS bit.

### 10.6.4.5 Vertical Blanking Interval

#### 10.6.4.5.1 Closed Caption Encoding

The encoder can be programmed to encode closed-caption data and extended data in the selected line. The closed-caption data are sent to the encoder through the L4\_PER interconnect. The data stream consists of 7-bit US-ASCII code and 1 odd-parity bit (see [Table 10-703](#)).

**Table 10-703. Closed-Caption Data Format**

MSB						LSB	
Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Odd parity

The standard service encodes closed caption in both fields; the extended service encodes closed caption in even fields. When set to 1, the [VENC\\_L21\\_WC\\_CTL L21EN\[0\]](#) bit enables closed-caption encoding in odd fields. When set to 1, the [VENC\\_L21\\_WC\\_CTL L21EN\[1\]](#) bit enables closed-caption encoding in even fields.

To select the scan line where the CC data are encoded, program the [VENC\\_LN\\_SEL\[4:0\]](#) SLINE bit field.

**CAUTION**

The setting of the [SLINE\[4:0\]](#) bit field value depends on the video standard:

- PAL mode: Because there is a one-line offset, program the desired line number – 1. To activate the closed caption on line 21 (0x15), program the value  $0x15 - 1 = 0x14$ . The default value is  $0x15 + 1 = 0x16$  (line 22).
- NTSC mode: Because there is a four-line offset, program the desired line number – 4. To activate the closed caption on line 21 (0x15), program the value  $0x15 - 4 = 0x11$ . The default value is  $0x15 + 4 = 0x19$  (line 25).

The [VENC\\_LN\\_SEL\[25:16\]](#) LN21\_RUNIN bit field should be kept at reset value (0x10B). Four closed-caption data registers contain the data to be encoded. The [VENC\\_LINE21\[15:8\]](#) L21O and [VENC\\_LINE21\[7:0\]](#) L21O bit fields contain the first and the second bytes, respectively, of closed-caption data to be encoded in the odd field. The [VENC\\_LINE21\[31:24\]](#) L21E and [VENC\\_LINE21\[23:16\]](#) L21E bit fields contain the first and the second bytes, respectively, of data to be encoded in the even field.

Immediately after the closed-caption data is written to the registers, in either the odd field or even field, the corresponding closed-caption status bit ([VENC\\_STATUS\[4\]](#) CCE or [VENC\\_STATUS\[3\]](#) CCO) is reset to 0 to indicate that the closed-caption data is available in the closed-caption data registers and yet to be encoded.

Immediately after the closed-caption data is encoded, the [VENC\\_STATUS\[4\]](#) CCE bit or the [VENC\\_STATUS\[3\]](#) CCO bit is set to 1 to indicate that the closed-caption data has been encoded and is ready to accept new data. As seen in [Figure 10-183](#), a null character is automatically inserted if the closed-caption data is not written to the closed-caption data registers in time for encoding.

The running clock frequency is controlled by the [VENC\\_CC\\_CARR\\_WSS\\_CARR\[15:0\]](#) FCC bit field which should be kept at reset value (0x2631) to get 5034960.5Hz (32xline) for NTSC-601. The closed-caption running clock common frequencies are detailed in [Table 10-704](#).

**Table 10-704. Closed-Caption RunClock Frequency Settings**

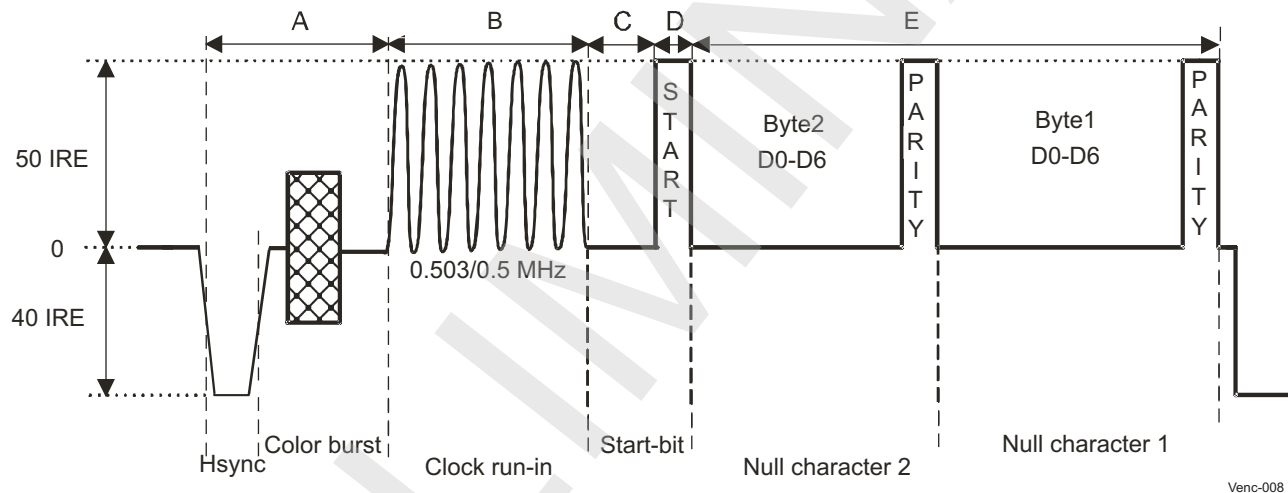
	NTSC-601	PAL-601	NTSC Square Pixel	PAL Square Pixel
<a href="#">VENC_CC_CARR_WSS_CARR[15:0]</a> FCC bit field value	0x2631	0x25ED	0x2A03	0x22B6

The closed-caption data is encoded in nonreturn-to-zero (NRZ) format. Additionally, the data translates to the IRE scale as follows:

- 0 = 0 IRE
- Sync pulse = -40 IRE
- 1 = 50 IRE

[Figure 10-183](#) shows the parameters of closed-caption line data implemented in different standards.

**Figure 10-183. Closed Captioning Timing**



**Table 10-705. Closed-Caption Standard Timing Values**

Intervals	Description	Timing Values for Encoding			Timing Values for Decoding		
		Minimal	Nominal	Maximal	Lower Bound	Nominal	Upper Bound
A	HSYNC to clock running	10.250 $\mu$ s	10.500 $\mu$ s	10.750 $\mu$ s	10.000 $\mu$ s	10.500 $\mu$ s	11.000 $\mu$ s
B	Clock running		12.910 $\mu$ s			12.910 $\mu$ s	
C	Clock running to third start bit		3.972 $\mu$ s			3.972 $\mu$ s	
D	Start bit		1.986 $\mu$ s			1.986 $\mu$ s	
E	Data characters		31.778 $\mu$ s			31.778 $\mu$ s	

**NOTE:**

- The interval A is controlled by the [VENC\\_LN\\_SEL\[25:16\]](#) LN21\_RUNIN bit field.
- The interval B is controlled by [VENC\\_CC\\_CARR\\_WSS\\_CARR\[15:0\]](#) FCC bit field.

All timing values listed in [Table 10-705](#) are measured from the mid-point (half amplitude) on all edges.

For a complete description of closed-caption standard including CGMS-A copy protection, see the CEA-608-x standard.



**10.6.4.5.2 Wide-Screen Signaling (WSS) Encoding**

The encoder can embed data, encoded in accordance with the IEC61880 and ITU-R 1119 data insertion standard, within the vertical blanking interval.

The encoder supports WSS data insertion on line 20 of every frame in the NTSC format. WSS data insertion is enabled by activating the [VENC\\_L21\\_WC\\_CTL\[14:13\]](#) EVEN\_ODD\_EN bit and by programming the [VENC\\_BSTAMP\\_WSS\\_DATA\[27:8\]](#) WSS\_DATA bit field.

The running clock frequency is controlled by the [VENC\\_CC\\_CARR\\_WSS\\_CARR\[31:16\]](#) FWSS bit field. The wide-screen signaling running clock common frequencies are detailed in [Table 10-706](#)

**Table 10-706. Wide-Screen Signaling RunClock Frequency Settings**

	NTSC-601	PAL-601	NTSC Square Pixel	PAL Square Pixel
<a href="#">VENC_CC_CARR_WSS_CARR[31:16]</a> FWSS bit field value	0x043F	0x2F72	0x04AC	0x2B6D

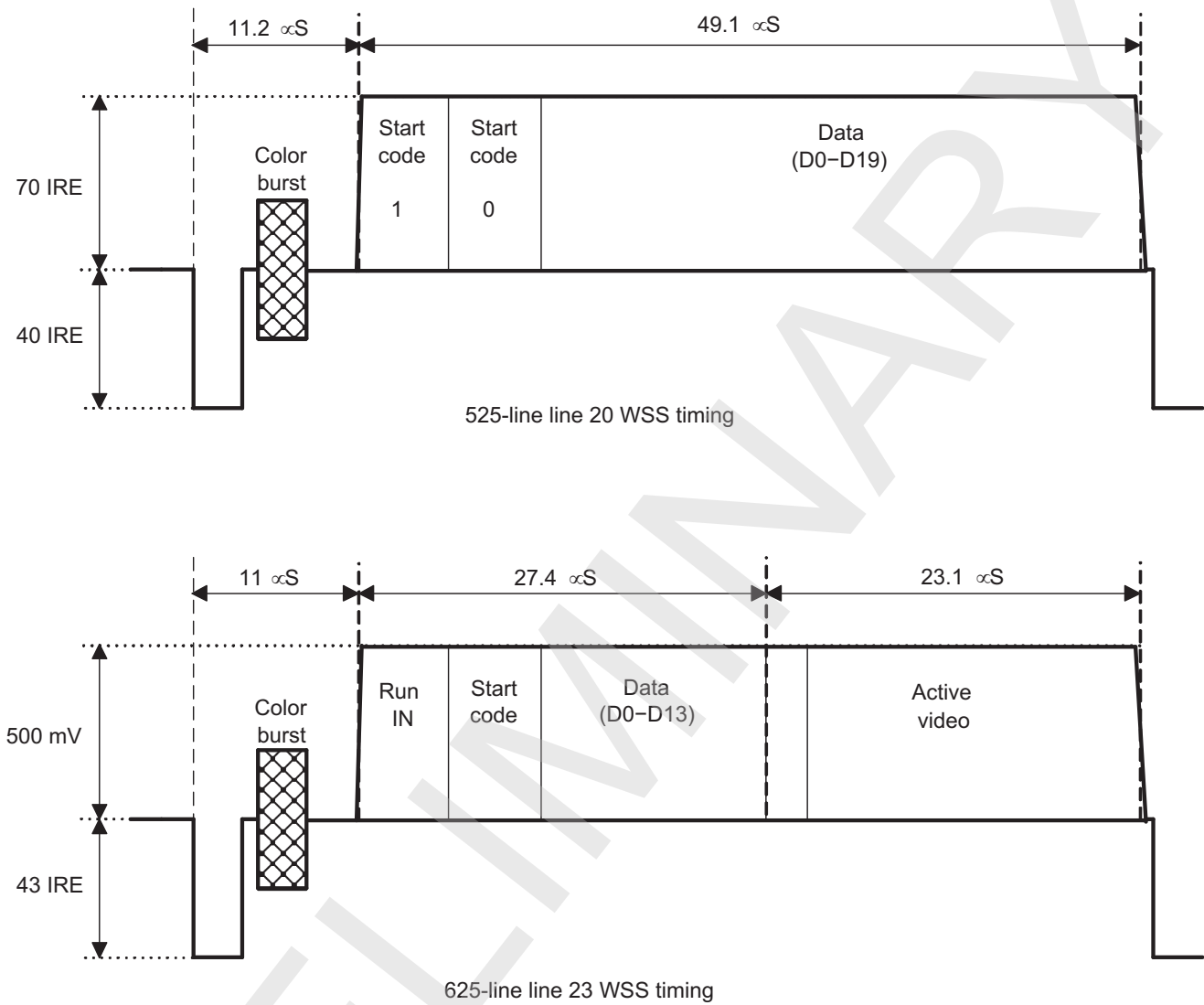
To select the line where the WSS data are encoded, program the [VENC\\_L21\\_WC\\_CTL\[12:8\]](#) LINE bit field.

**CAUTION**

The setting of the LINE[12:8] bit field value depends on the video standard:

- PAL mode: Because there is a one-line offset, program the desired line number – 1. The recommended value on line 23 is  $0x16 + 1 = 0x17$ . The default value is  $0x14 + 1 = 0x15$  (line 21).
- NTSC mode: Because there is a four-line offset, program the desired line number – 4. The recommended value on line 20 is  $0x10 + 4 = 0x14$ . The default value is  $0x14 + 4 = 0x18$  (line 24).

The WSS encoding block assumes that a full 10-bit video range is used to determine the 70 percent of peak-white amplitude of a logic-1 bit. The encoder also supports WSS data insertion on line 23 in the PAL format. Both waveforms are shown in [Figure 10-184](#).

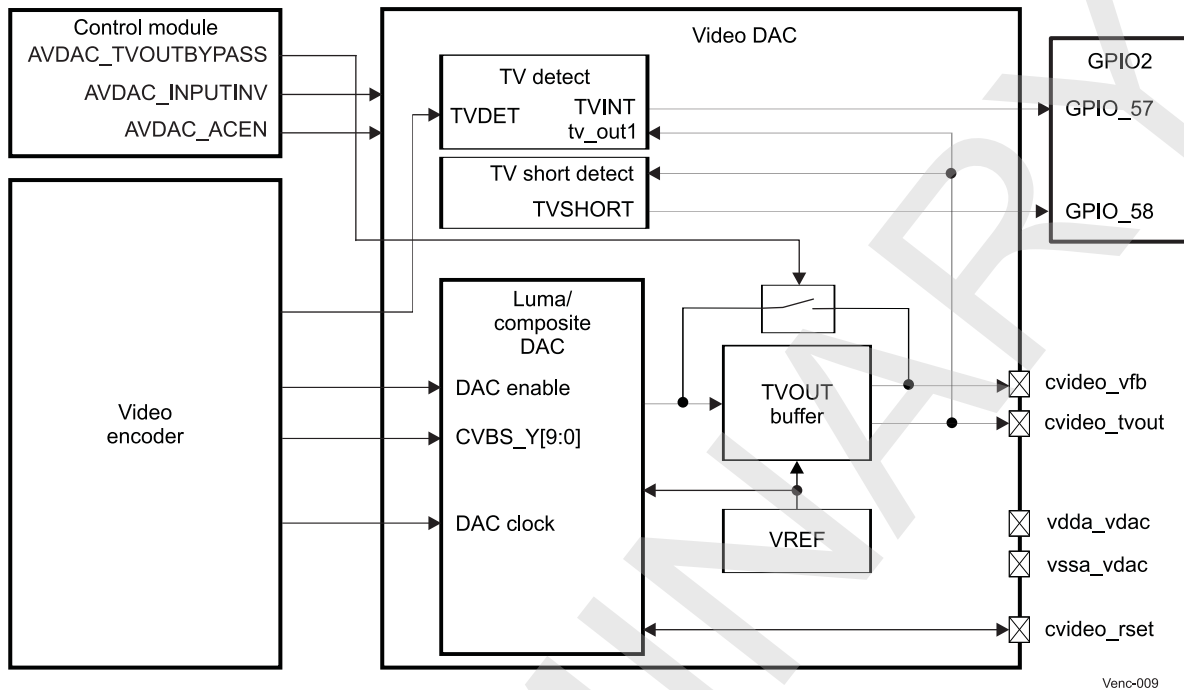
**Figure 10-184. WSS Timing**

Venc-007

**10.6.4.6 Video DAC**

Figure 10-185 shows the architecture of the 10-bit video DAC.

Figure 10-185. 10-Bit Video DAC Architecture



The display subsystem provides the necessary control signals to interface the memory frame buffer directly to external displays (TV sets). A 10-bit current steering DAC is used to generate the video analog signal. The video mode DAC also includes the TV detection/disconnection and power-down mode features.

#### 10.6.4.6.1 Video DC/AC Coupled TV Load

The 10-bit video DAC supports both dc-coupled and ac-coupled TV loads. The CONTROL.CONTROL\_AVDAC[31] AVDAC\_ACEN bit is used to define which output coupling is used. This bit is the first one to be programmed according to the TV load on the PCB board.

**NOTE:** When dc coupling is used, note that there is a 385-mV dc offset at the TVOUT output.

#### 10.6.4.6.2 TV Detection/Disconnection Pulse Generation and Use

##### 10.6.4.6.2.1 TV Detection/Disconnection Pulse Generation

The TV detect block is an integral part of the video DAC module.

**NOTE:**

- The TV detection/disconnection feature is supported on the video DAC.
- The TV disconnection feature is recommended for power saving purpose. The TV detection/disconnection is only operational when video out is active. Therefore to detect cable connection automatically, it is necessary to periodically activate the video out to test for cable presence.

This block compares the output ( *cvideo\_tvout* ) to reference voltage, to sense the condition of the load. To operate, the TV detect requires two digital signals, TVACEN and TVDET. The TVACEN signal

indicates to both TVOUT buffer and the TV detect circuit if the load is ac or dc coupling to adjust accordingly. TVDET is a digital pulse at the frequency of the TV sync pulses. The operation of the circuit is based on the difference in voltage levels in the output of the buffer depending of the load status. The TV detect block compares the output against a couple of references and the result is latched at the start of every sync pulse. The status is read later in with the TVDET pulse rising edge.

The following registers are used to set the TV detection/disconnection pulse:

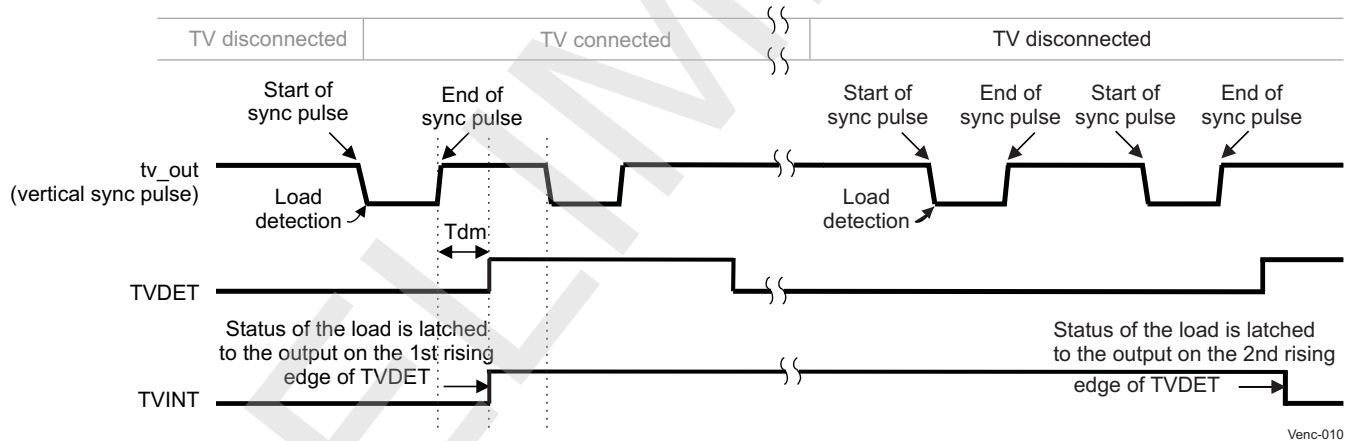
- The `VENC_TVDEETGP_INT_START_STOP_X` register defines which pixels are used to start and stop the pulse inside their respective line.
- The `VENC_TVDEETGP_INT_START_STOP_Y` register defines which lines are used to start and stop the pulse.
- The `VENC_GEN_CTRL[0]` EN bit enables or disables the TVDET pulse (0: Disable; 1: Enable).
- The `VENC_GEN_CTRL[16]` TVDP bit sets the TVDET pulse polarity (0: Active low; 1: Active high).

#### 10.6.4.6.2.2 Recommended TV Detection/Disconnection Pulse Waveform

To enable the detection/disconnection of the load, the circuit requires that the TVDET pulse resembles the following waveform. By using the video encoder registers, the TVDET pulse polarity, start and stop is programmable. The only critical parameter is  $T_{dm}$ , which should be longer than the delay through the DAC and TVOUT buffer, which is at least 750 ns.

If dc-coupling is selected, the TVINT output signal for TV detection is latched at the rising edge of the first TVDET signal but the TVINT output signal for TV disconnection is latched after the next rising edge of the TVDET signal that happens during the next VSYNC timing. [Figure 10-186](#) shows the waveforms for the dc-coupling TV detect pulse (TVDET) when load is connected and disconnected.

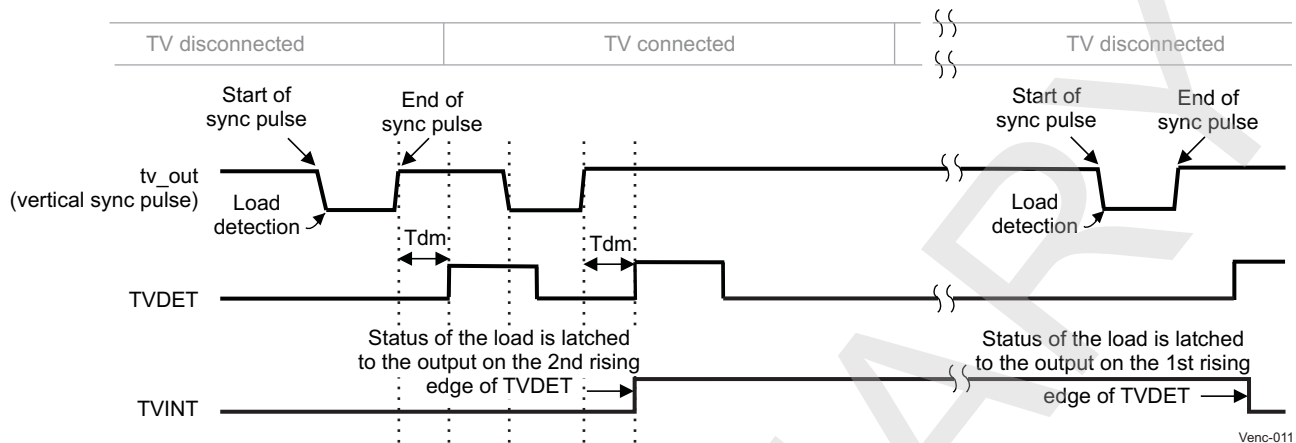
**Figure 10-186. DC-Coupling TV Detect Waveforms for TV Connected and Disconnected**



Venc-010

If ac-coupling is selected, the TVINT output signal for TV detection is latched after the next rising edge of the TVDET signal that happens during the next VSYNC timing but the TVINT output signal for TV disconnection is latched at the rising edge of the first TVDET signal. [Figure 10-187](#) shows the waveforms for the ac-coupling TV detect pulse (TVDET) when load is connected and disconnected.

**Figure 10-187. AC-Coupling TV Detect Waveforms for TV Connected and Disconnected**



**NOTE:**

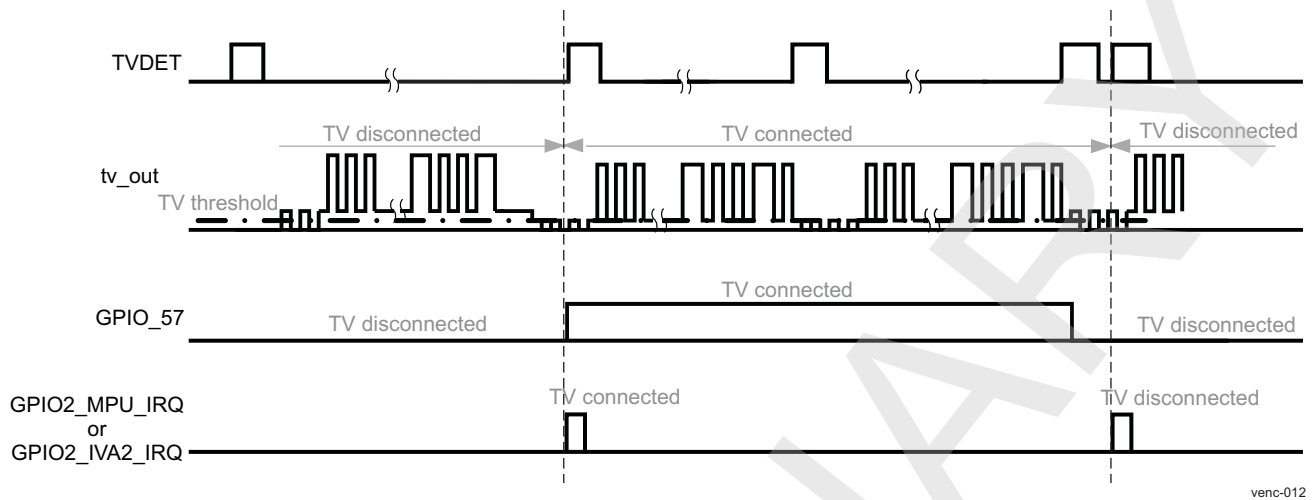
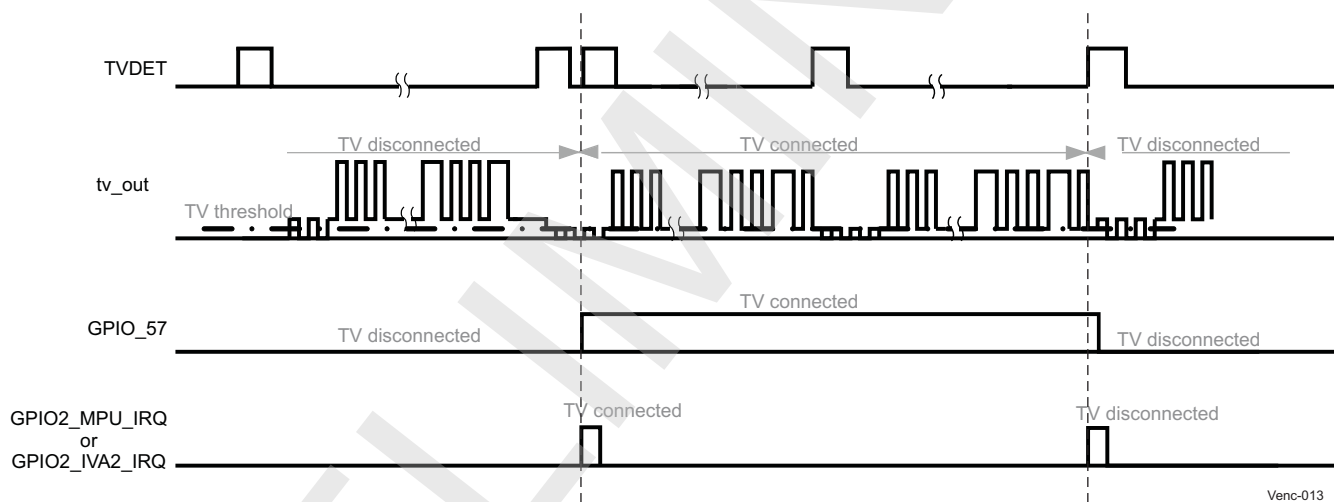
- When setting the [VENC\\_TVDETPGP\\_INT\\_START\\_STOP\\_X](#) register, software users must ensure that the TVDET signal is in the active area. To avoid any problem, the TVDET signal must not be longer than one line.
- The activation of the TVDET signal will not have a visual impact on the *cvideo\_tvout* output signal.

**10.6.4.6.2.3 TV Detection/Disconnection Use**

The TV-detection/TV-disconnection is based on the difference in voltage levels in the output of the TV buffer depending on the load status. The operation is slightly different for ac and for dc operation. For dc operation, the *cvideo\_tvout* voltage is compared against a voltage reference that makes the comparator trigger in each sync pulse while the load is connected. For ac operation, the *cvideo\_tvout* voltage is compared against a voltage reference that makes the comparator trigger in each sync pulse while the load is disconnected. In both cases, the TVINT output signal produces a logic 1 when a load is connected and a logic 0 when a load is disconnected.

For dc-coupling mode, see [Figure 10-188](#) and for ac-coupling mode, see [Figure 10-189](#)

- NOTE:** Because the video DAC and the video encoder must be awake for connection detection, consider that the video DAC can take up to 10  $\mu$ s to wake up.

**Figure 10-188. GPIO Signal Waveform Proposal for TV Detection/Disconnection in DC-Coupling Mode****Figure 10-189. GPIO Signal Waveform Proposal for TV Detection/Disconnection in AC-Coupling Mode**

### 10.6.4.6.3 TV Short Detection

A TV short-detection circuit, which can detect a short circuit condition at `cvideo_tvout` output during the vertical sync pulse, is integrated in the video DAC module (see [Figure 10-185](#)).

When the output is shorted to ground, the TVSHORT signal turns to logic high. The TVSHORT signal is internally mapped in the device to the GPIO\_58 signal (channel 26 of the GPIO2 module), which allows interrupt generation on this channel (see [Chapter 25, General-Purpose Interface](#)). TV short detection is valid after two video frames pass with the short/no-short condition.

TV short detection has priority over TV load detection (see [Section 10.6.4.6.2.2, TV Detection/Disconnection Pulse Generation and Use](#)). Whenever a short is detected (with load or no-load), TVINT output is masked as long as the short remains. When released, TV load detection resumes (see [Section 10.6.4.6.2.2, TV Detection/Disconnection Pulse Generation and Use](#)).

#### CAUTION

If TV short is detected, the video DAC must be powered down by setting the `DSS.DSS_CTRL[5] DAC_POWERDN_BGZ` bit to 0.

#### 10.6.4.6.4 Video DAC Normal Mode

The normal mode of operation (DAC + TVOUT buffer) is the default mode of the video DAC, with the CONTROL.CONTROL\_AVDAC[30] AVDAC\_TVOUTBYPASS bit set to 0 at device reset. The output of the DAC is internally connected to the TVOUT buffer, which can drive the composite video signal to 75 Ω load directly. For information about external connectivity, see Section 10.6.2, *Video Encoder Environment*.

In normal mode, higher input values of the DAC code result in lower output voltage levels by default, because of the inverting configuration of the TVOUT buffer. The CONTROL.CONTROL\_AVDAC[29] AVDAC\_INPUTINV bit provides inversion control over the data input of the DAC, to allow alignment of the output voltage with VENC data. If the AVDAC\_INPUTINV bit is set to 1, the DAC input code is inverted. Thus, higher DAC code values provided by the VENC result in higher output voltage.

#### 10.6.4.6.5 Video DAC Bypass Mode

The 10-bit video DAC has a TVOUT buffer bypass mode that turns off the TVOUT buffer and redirects directly the DAC output to the *cvideo\_vfb* pin.

This bypass mode is activated by setting the CONTROL.CONTROL\_AVDAC[30] AVDAC\_TVOUTBYPASS bit to 1. The reset value of the CONTROL.CONTROL\_AVDAC[30] AVDAC\_TVOUTBYPASS bit is 0 (that is, the TVOUT buffer is not bypassed).

**NOTE:** In bypass mode:

- *cvideo\_rset* pin requires a Rset resistor connected to the ground. The typical value of the Rset resistor is 10000 Ω.
- *cvideo\_vfb* pin require a Rload resistor connected to the ground. The typical value of the Rload resistor is 1500 Ω.

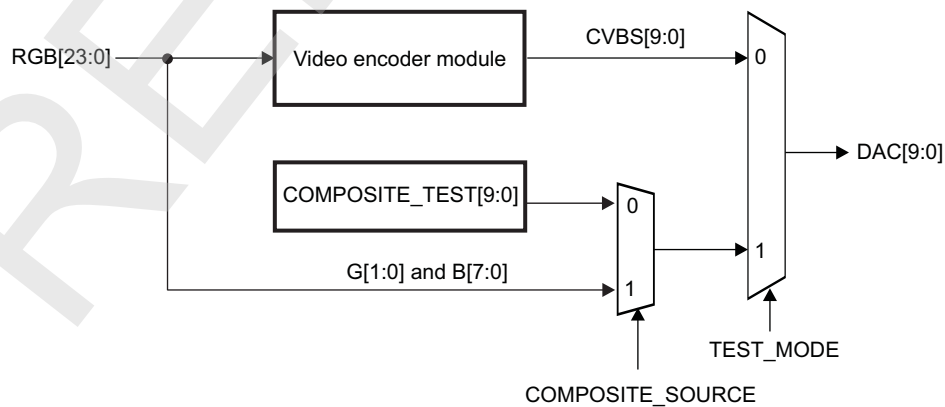
#### CAUTION

- The TV detect feature is not available in bypass mode.
- In bypass mode, an external amplifier is needed on the *cvideo\_vfb* pin.

#### 10.6.4.6.6 Video DAC Test Mode

10-bit external or internal datas values can be used to stimulate the DAC in debug, see Figure 10-190.

Figure 10-190. DAC Test Mode in Composite Video Mode



- Setting to 0x1 the VENC\_OUTPUT\_CONTROL[4] TEST\_MODE bit will place the DAC under test.
- By setting the value of the VENC\_OUTPUT\_CONTROL[6] COMPOSITE\_SOURCE bit to:
  - 0x0: the DAC input is coming from VENC\_OUTPUT\_TEST[9:0] COMPOSITE\_TEST bit field
  - 0x1: the DAC input is coming from the display controller G[1:0], B[7:0]. To perform this

configuration, the video encoder shall be configured to generate correct timing signals with the display controller even if the video encoder is bypassed from the data path perspective.

#### 10.6.4.6.7 Video DAC Power Management

After device reset, the DSS.DSS\_CTRL[5] DAC\_POWERDN\_BGZ bit is set to 0, and the video DAC is in power-down state.

Table 10-707 lists the possible power-management configurations and the corresponding register settings.

**Table 10-707. Video DAC Power-Management Configurations**

Power Management Controls				
DSS_CTRL[5] DAC_POWERDN_BGZ	VENC_OUTPUT_CONT ROL[1] COMPOSITE_ENABLE	CONTROL_AVDAC[30] AVDAC_TVOUTBYPAS S	CONTROL_AVDAC[31] AVDAC_ACEN	Power State
0	x	x	x	Total power down. Bandgap powered down.
1	0	x	x	Standby (analog in power down except bandgap/LDO)
1	1	0	0	Full power up in DC normal mode
1	1	0	1	Full power up in AC normal mode
1	1	1	x	Full power up in TVOUT bypass mode (DAC-only mode)



## 10.6.5 Video Encoder Programming Guide

### 10.6.5.1 Video Encoder Low-level Programming Models

#### 10.6.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the video emodule is to be used for the first time after a device reset. This initialization of the surrounding modules is based on the integration and environment of the video encoder (for more information, see [Section 10.6.2](#) and [Section 10.6.3](#)).

**Table 10-708. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. See <a href="#">Chapter 3, Power, Reset, and Clock Management</a>
Control module	Module-specific pad muxing and configuration must be set in the control module. See <a href="#">Chapter 18, Control Module</a> .
Display controller	See <a href="#">Section 10.2, Display Controller</a> , for more information about the configuration.

#### 10.6.5.1.2 Video Encoder Global Initialization

##### 10.6.5.1.2.1 Main Sequence—Start Video Encoder

[Table 10-709](#) details the procedure to initialize the video encoder after a power-on or software reset.

**Table 10-709. Start Video Encoder**

Step	Register/Bit Field/Programming Model	Value
Perform a video encoder reset.	See <a href="#">Section 10.6.5.1.2.3</a> .	
Save the DSS IRQ enable register value (DSS interrupts context).	DISPC_IRQENABLE	
Reset all DSS IRQ.	DISPC_IRQENABLE	0x00
Configure the video encoder registers for PAL 601 or NTSC 601.	See <a href="#">Table 10-716</a> .	0x--
Set the input video source and format.	VENC_F_CONTROL	See <a href="#">Table 10-716</a> .
Set the synchronization modes.	VENC_SYNC_CTRL	See <a href="#">Table 10-716</a> .
Configure the TV connection detection (optional).	See <a href="#">Section 10.6.5.1.2.4</a> .	
Enable the composite video signal output.	See <a href="#">Section 10.6.5.1.2.6</a> .	
Enable the display controller pipeline	See the display controller programming guide.	
Wait until the first VSYNC pulse signal.		
Clear the SYNCLOSTDIGITAL interrupt.	DISPC_IRQSTATUS[15] SYNCLOSTDIGITAL	0x1
Restore the DSS IRQ enable register.	DISPC_IRQENABLE	0x--

##### 10.6.5.1.2.2 Main Sequence—Stop Video Encoder

[Table 10-710](#) details the procedure to stop the output of the composite signal.

**Table 10-710. Start Video Encoder**

Step	Register/Bit Field/Programming Model	Value
Perform TV disconnection detection (optional).	See <a href="#">Section 10.6.5.1.2.5</a> .	
Disable the composite video signal output.	See <a href="#">Section 10.6.5.1.2.7</a> .	

### 10.6.5.1.2.3 Subsequence—Video Encoder Software Reset

Table 10-711 details the reset procedure.

**Table 10-711. Reset Video Encoder**

Step	Register/Bit Field/Programming Model	Value
Activate reset.	VENC_F_CONTROL[8] RESET	0x1
Wait until reset is complete.	VENC_F_CONTROL[8] RESET	0x0

**NOTE:** Before changing the standard (NTSC or PAL) and related registers, a software reset is required to properly initialize the VENC module.

### 10.6.5.1.2.4 Subsequence—TV Connection Detection

Table 10-712 details the TV connection detection procedure.

**Table 10-712. TV Connection Detection Subsequence**

Step	Register/Bit Field/Programming Model	Value
Set the pixel TV detection start and stop value.	VENC_TVDETPGP_INT_START_STOP_X	0x--
Set the line TV detection start and stop value.	VENC_TVDETPGP_INT_START_STOP_Y	0x--
Set the polarity of TV detection.	VENC_GEN_CTRL[16] TVDP	0x--
Enable TV detection.	VENC_GEN_CTRL[0] EN	0x1
Enable the composite video signal output.	See Section 10.6.5.1.2.6.	
Wait until TVINT = 1.	See the GPIO programming guide.	

**CAUTION**

- If ac coupling is selected, two TVDET pulses are required to set high the TVINT signal. Due to the internal logic of the video DAC, the TVINT signal is generated after the next positive edge of the TVDET signal that happens during the next VSYNC timing.
- If dc coupling is selected, only one TVDET pulse is required to set high the TVINT signal.

### 10.6.5.1.2.5 Subsequence—TV Disconnection Detection

Table 10-713 details the TV disconnection detection procedure.

**Table 10-713. TV Disconnection Detection Subsequence**

Step	Register/Bit Field/Programming Model	Value
Set the pixel TV detection start and stop value.	VENC_TVDETPGP_INT_START_STOP_X	0x--
Set the line TV detection start and stop value.	VENC_TVDETPGP_INT_START_STOP_Y	0x--
Set the polarity of TV detection.	VENC_GEN_CTRL[16] TVDP	0x--
Enable TV disconnection.	VENC_GEN_CTRL[0] EN	0x1
Wait until TVINT = 0.	See the GPIO programming guide.	

**CAUTION**

- If dc coupling is selected, two TVDET pulses are required to set low the TVINT signal. Because of the internal logic of the video DAC, the TVINT signal is generated after the next positive edge of the TVDET signal that happens during the next VSYNC timing.
- If ac coupling is selected, only one TVDET pulse is required to set low the TVINT signal.

**10.6.5.1.2.6 Subsequence—Enable Video Composite Signal Output**

Table 10-714 details the procedure to enable the video composite signal output.

**Table 10-714. Enable Video Composite Signal Output**

Step	Register/Bit Field/Programming Model	Value
Enable the composite video signal output.	VENC_OUTPUT_CONTROL	0x0000000A
Power up the vdda_vdac voltage for VDAC and the TVOUT buffer.		

**10.6.5.1.2.7 Subsequence—Disable Video Composite Signal Output**

Table 10-715 details the procedure to enable the video composite signal output.

**Table 10-715. Disable Video Composite Signal Output**

Step	Register/Bit Field/Programming Model	Value
Disable the composite video signal output.	VENC_OUTPUT_CONTROL	0x00000000
Power down the vdda_vdac voltage for VDAC and the TVOUT buffer.		

**10.6.6 Video Encoder Use Case and Tips**

**10.6.6.1 Video Encoder Register Settings**

Table 10-716 lists the register values to use in typical video encoder applications.

These values are validated programming values only for NTSC 601 and PAL 601 standards with a TV display resolution of 720\*488 and 720\*576, respectively.

**Table 10-716. Video Encoder Register Programming Values**

Register Name	NTSC 601	PAL 601
VENC_F_CONTROL	0x00000000	0x00000000
VENC_VIDOUT_CTRL	0x00000001	0x00000001
VENC_SYNC_CTRL	0x00008040	0x00000040
VENC_LLEN	0x00000359	0x0000035F
VENC_FLENS	0x0000020C	0x00000270
VENC_HFLTR_CTRL	0x00000000	0x00000000
VENC_CC_CARR_WSS_CARR	0x043F2631	0x2F7225ED
VENC_C_PHASE	0x00000000	0x00000000
VENC_GAIN_U	0x00000102	0x00000111
VENC_GAIN_V	0x0000016C	0x00000181
VENC_GAIN_Y	0x0000012F	0x00000140
VENC_BLACK_LEVEL	0x00000043	0x0000003B
VENC_BLANK_LEVEL	0x00000038	0x0000003B

**Table 10-716. Video Encoder Register Programming Values (continued)**

Register Name	NTSC 601	PAL 601
VENC_X_COLOR	0x00000007	0x00000007
VENC_M_CONTROL	0x00000001	0x00000002
VENC_BSTAMP_WSS_DATA	0x00000038	0x0000003F
VENC_S_CARR	0x21F07C1F	0x2A098ACB
VENC_LINE21	0x00000000	0x00000000
VENC_LN_SEL	0x01310011	0x01290015
VENC_L21_WC_CTL	0x0000F003	0x0000F603
VENC_HTRIGGER_VTRIGGER	0x00000000	0x00000000
VENC_SAVID_EAVID	0x069300F4	0x06A70108
VENC_FLEN_FAL	0x0016020C	0x00180270
VENC_LAL_PHASE_RESET	0x00060107	0x00040135
VENC_HS_INT_START_STOP_X	0x008E0350	0x00880358
VENC_HS_EXT_START_STOP_X	0x000F0359	0x000F035F
VENC_VS_INT_START_X	0x01A00000	0x01A70000
VENC_VS_INT_STOP_X_VS_INT_START_Y	0x020701A0	0x000001A7
VENC_VS_INT_STOP_Y_VS_EXT_START_X	0x01AC0024	0x01AF0000
VENC_VS_EXT_STOP_X_VS_EXT_START_Y	0x020D01AC	0x000101AF
VENC_VS_EXT_STOP_Y	0x00000006	0x00000025
VENC_AVID_START_STOP_X	0x03480078	0x03530083
VENC_AVID_START_STOP_Y	0x02060024	0x026C002E
VENC_FID_INT_START_X_FID_INT_START_Y	0x0001008A	0x0001008A
VENC_FID_INT_OFFSET_Y_FID_EXT_START_X	0x01AC0106	0x002E0138
VENC_FID_EXT_START_Y_FID_EXT_OFFSET_Y	0x01060006	0x01380001
VENC_TVDETPG_INT_START_STOP_X	0x00140001	0x00140001
VENC_TVDETPG_INT_START_STOP_Y	0x00010001	0x00010001
VENC_GEN_CTRL	0x00F90000	0x00FF0000
VENC_OUTPUT_CONTROL	0x0000000A	0x0000000A
VENC_OUTPUT_TEST	0x00000000	0x00000000

**NOTE:** The following display controller fields must be programmed to the NTSC 601 video standard:

- DISPC\_SIZE\_DIG[10:0] PPL =  $720 - 1 = 719 = 0x2CF$
- DISPC\_SIZE\_DIG[26:16] LPP =  $(488/2) - 1 = 243 = 0xF3$

The following display controller fields must be programmed to the PAL 601 video standard:

- DISPC\_SIZE\_DIG[10:0] PPL =  $720 - 1 = 719 = 0x2CF$
- DISPC\_SIZE\_DIG[26:16] LPP =  $(576/2) - 1 = 287 = 0x11F$

For both standard, the base addresses should be handle as follow:

- DISPC\_GFX\_BA0 or DISPC\_VIDn\_BA0 = Base address of even bit field data
- DISPC\_GFX\_BA1 or DISPC\_VIDn\_BA1 = Base address of odd bit field data

## 10.6.7 Video Encoder Register Manual

### CAUTION

The main access to all Video Encoder registers is through the L3 interconnect.  
The access through L4\_PER interconnect is provided for back software compatibility.

### 10.6.7.1 Video Encoder Instance Summary

**Table 10-717. Video Encoder Instance Summary**

Module Name	L3 Base Address	L4_PER Base Address	Size
VENC	0x5800 3000	0x4804 3000	4 KB

### CAUTION

The VENC registers have no register data width access restriction and can be accessed in 8-bit, 16-bit and 32-bit access.

### 10.6.7.2 Video Encoder Registers

#### 10.6.7.2.1 Video Encoder Register Summary

**Table 10-718. Video Encoder Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	VENC L3 Physical Address	VENC L4_PER Physical Address
VENC_REV_ID	R	32	0x0000 0000	0x5800 3000	0x4804 3000
VENC_STATUS	R	32	0x0000 0004	0x5800 3004	0x4804 3004
VENC_F_CONTRO L	RW	32	0x0000 0008	0x5800 3008	0x4804 3008
VENC_VIDOUT_CT RL	RW	32	0x0000 0010	0x5800 3010	0x4804 3010
VENC_SYNC_CTR L	RW	32	0x0000 0014	0x5800 3014	0x4804 3014
VENC_LLEN	RW	32	0x0000 001C	0x5800 301C	0x4804 301C
VENC_FLENS	RW	32	0x0000 0020	0x5800 3020	0x4804 3020
VENC_HFLTR_CT RL	RW	32	0x0000 0024	0x5800 3024	0x4804 3024
VENC_CC_CARR_ WSS_CARR	RW	32	0x0000 0028	0x5800 3028	0x4804 3028
VENC_C_PHASE	RW	32	0x0000 002C	0x5800 302C	0x4804 302C
VENC_GAIN_U	RW	32	0x0000 0030	0x5800 3030	0x4804 3030
VENC_GAIN_V	RW	32	0x0000 0034	0x5800 3034	0x4804 3034
VENC_GAIN_Y	RW	32	0x0000 0038	0x5800 3038	0x4804 3038
VENC_BLACK_LEV EL	RW	32	0x0000 003C	0x5800 303C	0x4804 303C
VENC_BLANK_LEV EL	RW	32	0x0000 0040	0x5800 3040	0x4804 3040
VENC_X_COLOR	RW	32	0x0000 0044	0x5800 3044	0x4804 3044
VENC_M_CONTRO L	RW	32	0x0000 0048	0x5800 3048	0x4804 3048

**Table 10-718. Video Encoder Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	VENC L3 Physical Address	VENC L4_PER Physical Address
VENC_BSTAMP_WSS_DATA	RW	32	0x0000 004C	0x5800 304C	0x4804 304C
VENC_S_CARR	RW	32	0x0000 0050	0x5800 3050	0x4804 3050
VENC_LINE21	RW	32	0x0000 0054	0x5800 3054	0x4804 3054
VENC_LN_SEL	RW	32	0x0000 0058	0x5800 3058	0x4804 3058
VENC_L21_WC_CTL	RW	32	0x0000 005C	0x5800 305C	0x4804 305C
VENC_HTRIGGER_VTRIGGER	RW	32	0x0000 0060	0x5800 3060	0x4804 3060
VENC_SAVID_EAV_ID	RW	32	0x0000 0064	0x5800 3064	0x4804 3064
VENC_FLEN_FAL	RW	32	0x0000 0068	0x5800 3068	0x4804 3068
VENC_LAL_PHASE_RESET	RW	32	0x0000 006C	0x5800 306C	0x4804 306C
VENC_HS_INT_START_STOP_X	RW	32	0x0000 0070	0x5800 3070	0x4804 3070
VENC_HS_EXT_START_STOP_X	RW	32	0x0000 0074	0x5800 3074	0x4804 3074
VENC_VS_INT_START_X	RW	32	0x0000 0078	0x5800 3078	0x4804 3078
VENC_VS_INT_STOP_X_VS_INT_START_Y	RW	32	0x0000 007C	0x5800 307C	0x4804 307C
VENC_VS_INT_STOP_Y_VS_EXT_START_X	RW	32	0x0000 0080	0x5800 3080	0x4804 3080
VENC_VS_EXT_STOP_X_VS_EXT_START_Y	RW	32	0x0000 0084	0x5800 3084	0x4804 3084
VENC_VS_EXT_STOP_Y	RW	32	0x0000 0088	0x5800 3088	0x4804 3088
VENC_AVID_START_STOP_X	RW	32	0x0000 0090	0x5800 3090	0x4804 3090
VENC_AVID_START_STOP_Y	RW	32	0x0000 0094	0x5800 3094	0x4804 3094
VENC_FID_INT_START_X_FID_INT_START_Y	RW	32	0x0000 00A0	0x5800 30A0	0x4804 30A0
VENC_FID_INT_OFFSET_Y_FID_EXT_START_X	RW	32	0x0000 00A4	0x5800 30A4	0x4804 30A4
VENC_FID_EXT_START_Y_FID_EXT_OFFSET_Y	RW	32	0x0000 00A8	0x5800 30A8	0x4804 30A8
VENC_TVDETGPI_NT_START_STOP_X	RW	32	0x0000 00B0	0x5800 30B0	0x4804 30B0
VENC_TVDETGPI_NT_START_STOP_Y	RW	32	0x0000 00B4	0x5800 30B4	0x4804 30B4
VENC_GEN_CTRL	RW	32	0x0000 00B8	0x5800 30B8	0x4804 30B8
VENC_OUTPUT_CONTROL	RW	32	0x0000 00C4	0x5800 30C4	0x4804 30C4
VENC_OUTPUT_TEST	RW	32	0x0000 00C8	0x5800 30C8	0x4804 30C8

10.6.7.2.2 Video Encoder Register Description

Table 10-719. VENC\_REV\_ID

Address Offset	0x0000 0000	Instance	VENC
Physical Address	See <a href="#">Table 10-718</a>		
Description	Revision ID for the encoder		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

Table 10-720. Register Call Summary for Register VENC\_REV\_ID

- Video Encoder
- [Video Encoder Register Summary: \[0\]](#)

Table 10-721. VENC\_STATUS

Address Offset	0x0000 0004	Instance	VENC
Physical Address	See <a href="#">Table 10-718</a>		
Description	STATUS		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CCE	CCO	FSQ													

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x00000000
4	CCE	Closed Caption Status for Even Field. This bit is set immediately after the data in registers LINE21_E0 and LINE21_E1 have been encoded to closed caption. This bit is reset when both of these registers are written.	R	0
3	CCO	Closed Caption Status for Odd Field. This bit is set immediately after the data in registers LINE21_O0 and LINE21_O1 have been encoded to closed caption. This bit is reset when both of these registers are written.	R	0
2:0	FSQ	Field Sequence ID. For PAL, all three FSQ[2:0] are used whereas for NTSC only FSQ[1:0] is meaningful. Furthermore, FSQ[0] represents ODD field when it is '0' and EVEN field when it is '1'. Read 0x0: ODD field Read 0x1: EVEN field	R	0x0

Table 10-722. Register Call Summary for Register VENC\_STATUS

- Video Encoder
- [Closed Caption Encoding: \[0\] \[1\] \[2\] \[3\]](#)
  - [Video Encoder Register Summary: \[4\]](#)

**Table 10-723. VENC\_F\_CONTROL**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	This register specifies the input video source and format		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RESET	SVDS	RGBF	BCOLOR	FMT				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	RESET	RESET the encoder 0x0: No effect 0x1: Reset the encoder, after reset, this bit is automatically set to zero.	RW	0
7:6	SVDS	Select Video Data Source. 0x0: Use external video source 0x1: Use internal Color BAR 0x2: Use background color 0x3: Reserved	RW	0x2
5	RGBF	RGB /YCrCb input coding range 0x0: The input RGB data are in binary format with coding range 0-255 The input YCrCb data are in binary format with coding range 0-255 0x1: The input RGB data are in binary format with coding range 16-235 The input YCrCb data are in binary format conforming to ITU-601 standard	RW	0
4:2	BCOLOR	Background color select 0x0: 0x1: 0x2: 0x3: 0x4: 0x5: 0x6: 0x7:	RW	0x1
1:0	FMT	These two bits specify the video input data stream format and timing 0x0: 24-bit 4:4:4 RGB 0x1: 24-bit 4:4:4 0x2: 16-bit 4:2:2 0x3: 8-bit ITU-R 656 4:2:2	RW	0x3

**Table 10-724. Register Call Summary for Register VENC\_F\_CONTROL**

## Video Encoder

- [Data Manager: \[0\]](#)
- [Test Pattern Generation: \[1\] \[2\] \[3\]](#)
- [Video Encoder Global Initialization: \[4\] \[5\] \[6\]](#)
- [Video Encoder Register Settings: \[7\]](#)
- [Video Encoder Register Summary: \[8\]](#)



**Table 10-725. VENC\_VIDOUT\_CTRL**

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b> VENC
<b>Description</b>	Encoder output clock	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MHZ_27_54			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MHZ_27_54	Encoder output clock 0x0: 54 MHz, 4x oversampling 0x1: 27MHz, 2x oversampling, the last 2x oversampling filter bypassed	RW	0

**Table 10-726. Register Call Summary for Register VENC\_VIDOUT\_CTRL**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-727. VENC\_SYNC\_CTRL**

<b>Address Offset</b>	0x0000 0014	
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b> VENC
<b>Description</b>	SYNC Control register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FREE	ESAV	IGNP	NBLNKS	VBLKM	HBLKM	M_S	FID_POL	RESERVED	VS_POL	HS_POL	RESERVED	FHVMOD			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	reserved_5	R	0x0000
15	FREE	Free running 0x0: Free running disabled 0x1: Free running enabled. HSYNC and VSYNC are ignored	RW	1
14	ESAV	Enable to detect F and V bits only on EAV in ITU-R 656 input mode 0x0: Detection of F and V bits on both EAV and SAV 0x1: Detection of F and V bits only on EAV	RW	0
13	IGNP	Ignore protection bits in ITU-R 656 input mode 0x0: Protection bits are not ignored 0x1: Protection bits are ignored	RW	0

Bits	Field Name	Description	Type	Reset
12	NBLNKS	Blank shaping 0x0: Blank shaping enabled 0x1: Blank shaping disabled	RW	0
11:10	VBLKM	Vertical blanking mode 0x0: Internal default blanking 0x1: Internal default blanking AND internal programmable blanking defined by FAL and LAL 0x2: Reserved 0x3: Reserved	RW	0x0
9:8	HBLKM	Horizontal blanking mode 0x0: Internal default blanking 0x1: Internal programmable blanking defined by SAVID and EAVID 0x2: External blanking defined by AVID 0x3: Reserved	RW	0x0
7	M_S	Encoder is master or slave of external sync 0x0: Sync master. Encoder outputs Vsync, Hsync, FID and AVID. 0x1: Sync slave. Encoder is synchronized to external Vsync, Hsync, FID and AVID.	RW	0
6	FID_POL	FID output polarity 0x0: ODD field = '0' EVEN field = '1' 0x1: ODD field = '1' EVEN field = '0'	RW	0
5:4	RESERVED		R	0x0
3	VS_POL	VS input polarity 0x0: VS is active high 0x1: VS is active low	RW	0
2	HS_POL	HS input polarity 0x0: HS is active high 0x1: HS is active low	RW	0
1	RESERVED		R	0
0	FHVMOD	FID extracted from external FID or HSYNC and VSYNC 0x0: External FID 0x1: Extracted from HSYNC and VSYNC	RW	0

**Table 10-728. Register Call Summary for Register VENC\_SYNC\_CTRL**

Video Encoder

- [Video Encoder Global Initialization: \[0\]](#)
- [Video Encoder Register Settings: \[1\]](#)
- [Video Encoder Register Summary: \[2\]](#)

**Table 10-729. VENC\_LLEN**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	LLEN		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LLEN_EN	RESERVED						LLEN								

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	LLEN_EN	LLEN_EN 0x0: disable 0x1: enable	RW	0
14:11	RESERVED		R	0x0
10:0	LLEN	LLEN[10:0] Line length or total number of pixels in a scan line including active video and blanking. Total number of pixels in a scan line = LLEN NOTE: A write to bit 11 of this bit field is illegal.	RW	0x359

**Table 10-730. Register Call Summary for Register VENC\_LLEN**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-731. VENC\_FLENS**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	FLENS		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FLENS															

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10:0	FLENS	The frame length or total number of lines in a frame including active video and blanking from the source image. Total number of lines in a frame from the source image = FLENS + 1	RW	0x20C

**Table 10-732. Register Call Summary for Register VENC\_FLENS**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-733. VENC\_HFLTR\_CTRL**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	HFLTR_CTRL		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CINTP		YINTP						

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:1	CINTP	Chrominance interpolation filter control 0x0: The chrominance interpolation filter is enabled 0x1: The first section of the chrominance interpolation filter is bypassed 0x2: The second section of the chrominance interpolation filter is bypassed 0x3: Both sections of the filter are bypassed	RW	0x0
0	YINTP	Luminance interpolation filter control 0x0: The luminance interpolation filter is enabled 0x1: The luminance interpolation filter is bypassed	RW	0

**Table 10-734. Register Call Summary for Register VENC\_HFLTR\_CTRL**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-735. VENC\_CC\_CARR\_WSS\_CARR**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	Frequency code control		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWSS												FCC																			

Bits	Field Name	Description	Type	Reset
31:16	FWSS	Wide screen signaling run-in code frequency control. For 50-Hz systems, FWSS = $2^{18} * 5 * 10^6 / (LLEN * Fh)$ , where LLEN = total number of pixels in a scan line Fh = line frequency.	RW	0x043F
15:0	FCC	Close caption run-in code frequency control. For 60-Hz system, FCC = $2^{18} * 0.5035 * 10^6 / (LLEN * Fh)$ For 50-Hz systems, FCC = $2^{18} * 0.500 * 10^6 / (LLEN * Fh)$ , where LLEN = total number of pixels in a scan line Fh = line frequency.	RW	0x2631

**Table 10-736. Register Call Summary for Register VENC\_CC\_CARR\_WSS\_CARR**

- Video Encoder
- [Closed Caption Encoding: \[0\] \[1\] \[2\]](#)
  - [Wide-Screen Signaling \(WSS\) Encoding: \[3\] \[4\]](#)
  - [Video Encoder Register Settings: \[5\]](#)
  - [Video Encoder Register Summary: \[6\]](#)

**Table 10-737. VENC\_C\_PHASE**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	C_PHASE		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CPHS															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	CPHS	Phase of the encoded video color subcarrier (including the color burst) relative to H-sync. The adjustable step is 360/256 degrees.	RW	0x00

**Table 10-738. Register Call Summary for Register VENC\_C\_PHASE**

- Video Encoder
- [Subcarrier and Burst Generation: \[0\] \[1\] \[2\]](#)
  - [Video Encoder Register Settings: \[3\]](#)
  - [Video Encoder Register Summary: \[4\]](#)

**Table 10-739. VENC\_GAIN\_U**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	Gain control for Cb signal		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																GU															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:0	GU	Gain control for Cb signal. Following are typical programming examples for NTSC and PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE GU = 0x102 NTSC with no pedestal: WHITE - BLACK = 100 IRE GU = 0x117 PAL with no pedestal: WHITE - BLACK = 100 IRE GU = 0x111	RW	0x102

**Table 10-740. Register Call Summary for Register VENC\_GAIN\_U**

- Video Encoder
- [Chroma Stage: \[0\]](#)
  - [Video Encoder Register Settings: \[1\]](#)
  - [Video Encoder Register Summary: \[2\]](#)

**Table 10-741. VENC\_GAIN\_V**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	Gain control of Cr signal		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																GV															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:0	GV	Gain control of Cr signal. Following are typical programming examples for NTSC and PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE GV = 0x16C NTSC with no pedestal: WHITE - BLACK = 100 IRE GV = 0x189 PAL with no pedestal: WHITE - BLACK = 100 IRE GV = 0x181	RW	0x16C

**Table 10-742. Register Call Summary for Register VENC\_GAIN\_V**

Video Encoder

- [Chroma Stage: \[0\]](#)
- [Video Encoder Register Settings: \[1\]](#)
- [Video Encoder Register Summary: \[2\]](#)

**Table 10-743. VENC\_GAIN\_Y**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	Gain control of Y signal		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																GY															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:0	GY	Gain control of Y signal. Following are typical programming examples for NTSC/PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE GY = 0x12F NTSC with no pedestal: WHITE - BLACK = 100 IRE GY = 0x147 PAL with no pedestal: WHITE - BLACK = 100 IRE GY = 0x140	RW	0x12F

**Table 10-744. Register Call Summary for Register VENC\_GAIN\_Y**

Video Encoder

- [Luma Stage: \[0\]](#)
- [Video Encoder Register Settings: \[1\]](#)
- [Video Encoder Register Summary: \[2\]](#)

**Table 10-745. VENC\_BLACK\_LEVEL**

<b>Address Offset</b>	0x0000 003C	
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b> VENC
<b>Description</b>	BLACK LEVEL	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BLACK															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6:0	BLACK	Black level setting. Following are typical programming examples for NTSC/PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE BLACK_LEVEL = 0x43 NTSC with no pedestal: WHITE - BLACK = 100 IRE BLACK_LEVEL = 0x38 PAL with no pedestal: WHITE - BLACK = 100 IRE BLACK_LEVEL = 0x3B	RW	0x43

**Table 10-746. Register Call Summary for Register VENC\_BLACK\_LEVEL**

Video Encoder

- [Luma Stage: \[0\]](#)
- [Video Encoder Register Settings: \[1\]](#)
- [Video Encoder Register Summary: \[2\]](#)

**Table 10-747. VENC\_BLANK\_LEVEL**

<b>Address Offset</b>	0x0000 0040	
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b> VENC
<b>Description</b>	BLANK LEVEL	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BLANK															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6:0	BLANK	Blank level setting. Following are typical programming examples for NTSC/PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE BLANK_LEVEL = 0x38 NTSC with no pedestal: WHITE - BLACK = 100 IRE BLANK_LEVEL = 0x38 PAL with no pedestal: WHITE - BLACK = 100 IRE BLANK_LEVEL = 0x3B	RW	0x38

**Table 10-748. Register Call Summary for Register VENC\_BLANK\_LEVEL**

Video Encoder

- [Luma Stage: \[0\]](#)
- [Video Encoder Register Settings: \[1\]](#)
- [Video Encoder Register Summary: \[2\]](#)

**Table 10-749. VENC\_X\_COLOR**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	Cross-Colour Control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XCE	RESERVED	XCBW	LCD												

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6	XCE	Cross color reduction enable for composite video output. Cross color does not affect S-video output 0x0: Cross color reduction is disabled 0x1: Cross color is enabled	RW	0
5	RESERVED		R	0
4:3	XCBW	Cross color reduction filter selection 0x0: The notch is at 32.8 % of the frequency of the encoding pixel clock 0x1: The notch is at 26.5 % of the frequency of the encoding pixel clock 0x2: The notch is at 30.0 % of the frequency of the encoding pixel clock 0x3: The notch is at 29.2 % of the frequency of the encoding pixel clock	RW	0x0
2:0	LCD	These three bits can be used for chroma channel delay compensation. Delay on Luma channel. 0x0: 0 0x1: 0.5 pixel clock period 0x2: 1.0 pixel clock period 0x3: 1.5 pixel clock period 0x4: -2.0 pixel clock period 0x5: -1.5 pixel clock period 0x6: -1.0 pixel clock period 0x7: -0.5 pixel clock period	RW	0x0

**Table 10-750. Register Call Summary for Register VENC\_X\_COLOR**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)



**Table 10-751. VENC\_M\_CONTROL**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	M_CONTROL		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PALI	PALN	PALPHS	CBW			PAL	FFRQ	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7	PALI	PAL I enable 0x0: Normal operation 0x1: PAL I Enable	RW	0
6	PALN	PAL N Enable 0x0: Normal operation 0x1: PAL N enable	RW	0
5	PALPHS	PAL switch phase setting 0x0: PAL switch phase is nominal 0x1: PAL switch phase is inverted compared to nominal	RW	0
4:2	CBW	Chrominance lowpass filter bandwidth control 0x0: -6db at 21.8 % of encoding pixel clock frequency 0x1: -6db at 19.8 % of encoding pixel clock frequency 0x2: -6db at 18.0 % of encoding pixel clock frequency 0x3: Reserved 0x4: Reserved 0x5: -6db at 23.7 % of encoding pixel clock frequency 0x6: -6db at 26.8 % of encoding pixel clock frequency 0x7: Chrominance lowpass filter bypass	RW	0x0
1	PAL	Phase alternation line encoding selection 0x0: Phase alternation line encoding disabled 0x1: Phase alternation line encoding enabled	RW	0
0	FFRQ	The value of this field and the SQP bit in the BSTAMP_WSS_DATA register control the number of horizontal pixels displayed per scan line # OF MODE SQP FFRQ PIXEL PER LINE ITU-R 601 NTSC 0 1 858 Square pixel NTSC 1 1 780 ITU-R 601 PAL 0 0 864 Square pixel PAL 1 0 944	RW	1

**Table 10-752. Register Call Summary for Register VENC\_M\_CONTROL**

Video Encoder

- [Subcarrier and Burst Generation: \[0\] \[1\]](#)
- [Video Encoder Register Settings: \[2\]](#)
- [Video Encoder Register Summary: \[3\]](#)

**Table 10-753. VENC\_BSTAMP\_WSS\_DATA**

<b>Address Offset</b>	0x0000 004C		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	BSTAMP and WSS_DATA		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WSS_D																0	BSTAP						

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:8	WSS_D	Wide Screen Signaling data NTSC: WORD 0 D0, D1 WORD 1 D2, D3, D4, D5 WORD 2 D6, D7, D8, D9, D10, D11, D12, D13 CRC D14, D15, D16, D17, D18, D19 PAL: GROUP A D0, D1, D2, D3 GROUP B D4, D5, D6, D7 GROUP C D8, D9, D10 GROUP D D11, D12, D13	RW	0x00000
7	SQP	Square-pixel sampling rate. See FFRQ in M_CONTROL register for programming information.  0x0: ITU-R 601 sampling rate 0x1: Square-pixel sampling rate	RW	0
6:0	BSTAP	Setting of amplitude of color burst.	RW	0x38

**Table 10-754. Register Call Summary for Register VENC\_BSTAMP\_WSS\_DATA**

## Video Encoder

- [Subcarrier and Burst Generation: \[0\]](#)
- [Wide-Screen Signaling \(WSS\) Encoding: \[1\]](#)
- [Video Encoder Register Settings: \[2\]](#)
- [Video Encoder Register Summary: \[3\]](#)

**Table 10-755. VENC\_S\_CARR**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	Color Subcarrier Frequency Registers.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC																															

Bits	Field Name	Description	Type	Reset
31:0	FSC	These four bytes' data are used to program color subcarrier frequency. These four bytes are determined by the following formula. $S\_CARR = \text{ROUND}((Fsc/Fclkenc) * 232)$ Where $Fsc$ = Frequency of the subcarrier $Fclkenc$ = Frequency of the internal video encoding clock = $2 * LLEN * Fh$ $LLEN$ = Number of pixels in a scan line $Fh$ = Line frequency See the description of $LLEN$ registers (subaddresses 0x42 and 0x43).	RW	0x21F0 7C1F

**Table 10-756. Register Call Summary for Register VENC\_S\_CARR**

- Video Encoder
- [Subcarrier and Burst Generation: \[0\] \[1\] \[2\]](#)
  - [Video Encoder Register Settings: \[3\]](#)
  - [Video Encoder Register Summary: \[4\]](#)

**Table 10-757. VENC\_LINE21**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	LINE 21		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L21E								L21O																							

Bits	Field Name	Description	Type	Reset
31:16	L21E	The two bytes of the closed caption data in the even field. For the data stream content, see <a href="#">Table 10-703</a> . [31:24] First byte of data [23:16] Second byte of data	RW	0x0000
15:0	L21O	The two bytes of the closed caption data in the odd field For the data stream content, see <a href="#">Table 10-703</a> . [15:8] First byte of data [7:0] Second byte of data	RW	0x0000

**Table 10-758. Register Call Summary for Register VENC\_LINE21**

- Video Encoder
- [Closed Caption Encoding: \[0\] \[1\] \[2\] \[3\]](#)
  - [Video Encoder Register Settings: \[4\]](#)
  - [Video Encoder Register Summary: \[5\]](#)

**Table 10-759. VENC\_LN\_SEL**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	LN_SEL		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LN21_RUNIN								RESERVED								SLINE							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	LN21_RUNIN	The two Bytes of the closed caption runin code position from the HSYNC	RW	0x10B
15:5	RESERVED		R	0x000
4:0	SLINE	Selects the line where closed caption or extended service data are encoded.  PAL mode: Because there is a one-line offset, program the desired line number – 1.	RW	0x15

Bits	Field Name	Description	Type	Reset
		NTSC mode: Because there is a four-line offset, program the desired line number – 4.		

**Table 10-760. Register Call Summary for Register VENC\_LN\_SEL**

Video Encoder

- [Closed Caption Encoding: \[0\] \[1\] \[2\]](#)
- [Video Encoder Register Settings: \[3\]](#)
- [Video Encoder Register Summary: \[4\]](#)

**Table 10-761. VENC\_L21\_WC\_CTL**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	L21 & WC_CTL registers		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																INV	EVEN_ODD_EN		LINE								RESERVED				L21EN	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	INV	WSS inverter 0x0: no effect 0x1: invert WSS data	RW	0
14:13	EVEN_ODD_EN	This bit controls the WSS encoding. 0x0: WSS encoding OFF 0x1: Enables encoding in 2nd field (odd field) 0x2: Enables encoding in 1st field (even field) 0x3: Enables encoding in both fields	RW	0x0
12:8	LINE	Selects the line where WSS data are encoded. PAL mode: Because there is a one-line offset, program the desired line number – 1. NTSC mode: Because there is a four-line offset, program the desired line number – 4.	RW	0x14
7:2	RESERVED		R	0x00
1:0	L21EN	Those bits controls the Line21 closed caption encoding according to the mode. 0x0: Line21 encoding OFF 0x1: Enables encoding in 1st field (ODD field) 0x2: Enables encoding in 2d field (EVEN field) 0x3: Enables encoding in both fields	RW	0x0

**Table 10-762. Register Call Summary for Register VENC\_L21\_WC\_CTL**

- Video Encoder
- [Closed Caption Encoding: \[0\] \[1\]](#)
  - [Wide-Screen Signaling \(WSS\) Encoding: \[2\] \[3\]](#)
  - [Video Encoder Register Settings: \[4\]](#)
  - [Video Encoder Register Summary: \[5\]](#)

**Table 10-763. VENC\_HTRIGGER\_VTRIGGER**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	HTRIGGER and VTRIGGER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VTRIG								RESERVED								HTRIG							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VTRIG	Vertical trigger reference for VSYNC. These bits specify the phase between VSYNC input and the lines in a field. The VTRIG field is expressed in units of half-line.	RW	0x000
15:11	RESERVED		R	0x00
10:0	HTRIG	Horizontal trigger phase, which sets HSYNC. HTRIG is expressed in half-pixels or clk2x (27 MHz) periods	RW	0x000

**Table 10-764. Register Call Summary for Register VENC\_HTRIGGER\_VTRIGGER**

- Video Encoder
- [Video Encoder Register Settings: \[0\]](#)
  - [Video Encoder Register Summary: \[1\]](#)

**Table 10-765. VENC\_SAVID\_EAVID**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	SAVID and EAVID		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EAVID								RESERVED								SAVID							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26:16	EAVID	End of active video. These bits define the ending pixel position on a horizontal display line where active video will be displayed.	RW	0x693
15:11	RESERVED		R	0x00
10:0	SAVID	Start of active video. These bits define the starting pixel position on a horizontal line where active video will be displayed.	RW	0x0F4

**Table 10-766. Register Call Summary for Register VENC\_SAVID\_EAVID**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-767. VENC\_FLEN\_FAL**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	FLEN and FAL		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FAL								RESERVED								FLEN							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:16	FAL	First Active Line of Field. These bits define the first active line of a field	RW	0x016
15:10	RESERVED		R	0x00
9:0	FLEN	Field length. These bits define the number of half_lines in each field. Length of field = (FLEN + 1) half_lines	RW	0x20C

**Table 10-768. Register Call Summary for Register VENC\_FLEN\_FAL**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-769. VENC\_LAL\_PHASE\_RESET**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	LAL and PHASE_RESET		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRES		SBLANK	RESERVED								LAL												

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18:17	PRES	Phase reset mode. 0x0: No reset 0x1: Reset every two lines 0x2: Reset every eight fields. Color subcarrier phase is reset to C_Phase (subaddress 5A) upon reset 0x3: Reset every four fields. Color subcarrier phase is reset to C_Phase (subaddress 5A) upon reset	RW	0x3

Bits	Field Name	Description	Type	Reset
16	SBLANK	Vertical blanking setting 0x0: Vertical blanking is defined by the setting of FAL and LAL registers. 0x1: Vertical blanking is forced automatically during field synchronization and equalization.	RW	0
15:9	RESERVED		R	0x00
8:0	LAL	Last Active Line of Field. These bits define the last active line of a field	RW	0x107

**Table 10-770. Register Call Summary for Register VENC\_LAL\_PHASE\_RESET**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-771. VENC\_HS\_INT\_START\_STOP\_X**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	HS_INT_START_STOP_X		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HS_INT_STOP_X								RESERVED								HS_INT_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	HS_INT_STOP_X	HSYNC internal stop. These bits define HSYNC internal stop pixel value	RW	0x07E
15:10	RESERVED		R	0x00
9:0	HS_INT_START_X	HSYNC internal start. These bits define HSYNC internal start pixel value	RW	0x34E

**Table 10-772. Register Call Summary for Register VENC\_HS\_INT\_START\_STOP\_X**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-773. VENC\_HS\_EXT\_START\_STOP\_X**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	HS_EXT_START_STOP_X		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HS_EXT_STOP_X								RESERVED								HS_EXT_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	HS_EXT_STOP_X	HSYNC external stop. These bits define HSYNC external stop pixel value	RW	0x00F
15:10	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
9:0	HS_EXT_START_X	HSYNC external start. These bits define HSYNC external start pixel value	RW	0x359

**Table 10-774. Register Call Summary for Register VENC\_HS\_EXT\_START\_STOP\_X**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-775. VENC\_VS\_INT\_START\_X**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	VS_INT_START_X		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VS_INT_START_X								RESERVED															

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VS_INT_START_X	VSYNC internal start. These bits define VSYNC internal start pixel value.	RW	0x1A0
15:0	RESERVED		R	0x0000

**Table 10-776. Register Call Summary for Register VENC\_VS\_INT\_START\_X**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-777. VENC\_VS\_INT\_STOP\_X\_VS\_INT\_START\_Y**

<b>Address Offset</b>	0x0000 007C	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	VS_INT_STOP_X and VS_INT_START_Y		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VS_INT_START_Y								RESERVED				VS_INT_STOP_X											

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VS_INT_START_Y	VSYNC internal start. These bits define VSYNC internal start line value	RW	0x209
15:10	RESERVED		R	0x00
9:0	VS_INT_STOP_X	VSYNC internal stop. These bits define VSYNC internal stop pixel value	RW	0x1A0

**Table 10-778. Register Call Summary for Register VENC\_VS\_INT\_STOP\_X\_VS\_INT\_START\_Y**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)



**Table 10-779. VENC\_VS\_INT\_STOP\_Y\_VS\_EXT\_START\_X**

<b>Address Offset</b>	0x0000 0080		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	VS_INT_STOP_Y and VS_EXT_START_X		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VS_EXT_START_X								RESERVED								VS_INT_STOP_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VS_EXT_START_X	VSYNC external start. These bits define VSYNC external start pixel value.	RW	0x1AC
15:10	RESERVED		R	0x00
9:0	VS_INT_STOP_Y	VSYNC internal stop. These bits define VSYNC internal stop line value.	RW	0x022

**Table 10-780. Register Call Summary for Register VENC\_VS\_INT\_STOP\_Y\_VS\_EXT\_START\_X**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-781. VENC\_VS\_EXT\_STOP\_X\_VS\_EXT\_START\_Y**

<b>Address Offset</b>	0x0000 0084		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	VS_EXT_STOP_X and VS_EXT_START_Y		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VS_EXT_START_Y								RESERVED								VS_EXT_STOP_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VS_EXT_START_Y	VSYNC external start. These bits define VSYNC external start line value.	RW	0x20D
15:10	RESERVED		R	0x00
9:0	VS_EXT_STOP_X	VSYNC external stop. These bits define VSYNC external stop pixel value.	RW	0x1AC

**Table 10-782. Register Call Summary for Register VENC\_VS\_EXT\_STOP\_X\_VS\_EXT\_START\_Y**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-783. VENC\_VS\_EXT\_STOP\_Y**

<b>Address Offset</b>	0x0000 0088	
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b> VENC
<b>Description</b>	VS_EXT_STOP_Y	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VS_EXT_STOP_Y															

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9:0	VS_EXT_STOP_Y	VSYNC external stop. These bits define VSYNC external stop line value.	RW	0x006

**Table 10-784. Register Call Summary for Register VENC\_VS\_EXT\_STOP\_Y**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-785. VENC\_AVID\_START\_STOP\_X**

<b>Address Offset</b>	0x0000 0090	
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b> VENC
<b>Description</b>	AVID_START_STOP_X	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								AVID_STOP_X								RESERVED								AVID_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	AVID_STOP_X	AVID stop. These bits define AVID stop pixel value	RW	0x348
15:10	RESERVED		R	0x00
9:0	AVID_START_X	AVID start. These bits define AVID start pixel value	RW	0x078

**Table 10-786. Register Call Summary for Register VENC\_AVID\_START\_STOP\_X**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-787. VENC\_AVID\_START\_STOP\_Y**

<b>Address Offset</b>	0x0000 0094		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	AVID_START_STOP_Y		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								AVID_STOP_Y								RESERVED								AVID_START_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	AVID_STOP_Y	AVID stop. These bits define AVID stop line value.	RW	0x206
15:10	RESERVED		R	0x00
9:0	AVID_START_Y	AVID start. These bits define AVID start line value	RW	0x026

**Table 10-788. Register Call Summary for Register VENC\_AVID\_START\_STOP\_Y**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-789. VENC\_FID\_INT\_START\_X\_FID\_INT\_START\_Y**

<b>Address Offset</b>	0x0000 00A0		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	FID_INT_START_X and FID_INT_START_Y		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FID_INT_START_Y								RESERVED								FID_INT_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	FID_INT_START_Y	FID internal stop. These bits define FID internal start line value	RW	0x001
15:10	RESERVED		R	0x00
9:0	FID_INT_START_X	FID internal start. These bits define FID internal start pixel value	RW	0x08A

**Table 10-790. Register Call Summary for Register VENC\_FID\_INT\_START\_X\_FID\_INT\_START\_Y**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-791. VENC\_FID\_INT\_OFFSET\_Y\_FID\_EXT\_START\_X**

<b>Address Offset</b>	0x0000 00A4		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	FID_INT_OFFSET_Y and FID_EXT_START_X		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FID_EXT_START_X								RESERVED								FID_INT_OFFSET_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	FID_EXT_START_X	FID external start. These bits define FID external start pixel value	RW	0x1AC
15:10	RESERVED		R	0x00
9:0	FID_INT_OFFSET_Y	FID internal offset. These bits define FID internal offset line value	RW	0x106

**Table 10-792. Register Call Summary for Register VENC\_FID\_INT\_OFFSET\_Y\_FID\_EXT\_START\_X**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-793. VENC\_FID\_EXT\_START\_Y\_FID\_EXT\_OFFSET\_Y**

<b>Address Offset</b>	0x0000 00A8		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	FID_EXT_START_Y and FID_EXT_OFFSET_Y		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FID_EXT_OFFSET_Y								RESERVED								FID_EXT_START_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	FID_EXT_OFFSET_Y	FID external offset. These bits define FID external offset line value	RW	0x106
15:10	RESERVED		R	0x00
9:0	FID_EXT_START_Y	FID external start. These bits define FID external start line value.	RW	0x006

**Table 10-794. Register Call Summary for Register VENC\_FID\_EXT\_START\_Y\_FID\_EXT\_OFFSET\_Y**

Video Encoder

- [Video Encoder Register Settings: \[0\]](#)
- [Video Encoder Register Summary: \[1\]](#)

**Table 10-795. VENC\_TVDETGP\_INT\_START\_STOP\_X**

<b>Address Offset</b>	0x0000 00B0		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	TVDETGP_INT_START_STOP_X		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TVDETGP_INT_STOP_X								RESERVED								TVDETGP_INT_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	TVDETGP_INT_STOP_X	TVDETGP internal stop. These bits define TVDETGP internal stop pixel value.	RW	0x014
15:10	RESERVED		R	0x00
9:0	TVDETGP_INT_START_X	TVDETGP internal start. These bits define TVDETGP internal start pixel value	RW	0x001

**Table 10-796. Register Call Summary for Register VENC\_TVDETGP\_INT\_START\_STOP\_X**

Video Encoder

- [TV Detection/Disconnection Pulse Generation and Use: \[0\] \[1\]](#)
- [Video Encoder Global Initialization: \[2\] \[3\]](#)
- [Video Encoder Register Settings: \[4\]](#)
- [Video Encoder Register Summary: \[5\]](#)

**Table 10-797. VENC\_TVDETGP\_INT\_START\_STOP\_Y**

<b>Address Offset</b>	0x0000 00B4		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	TVDETGP_INT_START_STOP_Y		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TVDETGP_INT_STOP_Y								RESERVED								TVDETGP_INT_START_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	TVDETGP_INT_STOP_Y	TVDETGP internal stop. These bits define TVDETGP internal stop line value.	RW	0x001
15:10	RESERVED		R	0x00
9:0	TVDETGP_INT_START_Y	TVDETGP internal start. These bits define TVDETGP internal start line value	RW	0x001

**Table 10-798. Register Call Summary for Register VENC\_TVDETGP\_INT\_START\_STOP\_Y**

Video Encoder

- [TV Detection/Disconnection Pulse Generation and Use: \[0\]](#)
- [Video Encoder Global Initialization: \[1\] \[2\]](#)
- [Video Encoder Register Settings: \[3\]](#)
- [Video Encoder Register Summary: \[4\]](#)

**Table 10-799. VENC\_GEN\_CTRL**

<b>Address Offset</b>	0x0000 00B8		
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b>	VENC
<b>Description</b>	TVDETGP enable and SYNC_POLARITY and UVPHASE_POL		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						MS	UVPHASE_POL_656	CBAR	HIP	VIP	HEP	VEP	AVIDP	FIP	FEP	TVDP	RESERVED											EN			

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	MS	UVPHASE_POL MS mode UV phase 0x0: CbCr 0x1: CrCb	RW	0
25	UVPHASE_POL_656	UVPHASE_POL 656 input mode UV phase 0x0: CbCr 0x1: CrCb	RW	0
24	CBAR	UVPHASE_POL CBAR mode UV phase 0x0: CbCr 0x1: CrCb	RW	0
23	HIP	HSYNC internal polarity 0x0: Active Low 0x1: Active High	RW	1
22	VIP	VSYNC internal polarity 0x0: Active Low 0x1: Active High	RW	1
21	HEP	HSYNC external polarity 0x0: Active Low 0x1: Active High	RW	1
20	VEP	VSYNC external polarity 0x0: Active Low 0x1: Active High	RW	1
19	AVIDP	AVID polarity 0x0: Active Low 0x1: Active High	RW	1
18	FIP	FID internal polarity 0x0: Active Low 0x1: Active High	RW	1
17	FEP	FID external polarity 0x0: Active Low 0x1: Active High	RW	1
16	TVDP	TVDETGP polarity 0x0: Active Low 0x1: Active High	RW	1

Bits	Field Name	Description	Type	Reset
15:1	RESERVED		R	0x0000
0	EN	TVDETGP generation enable 0x0: Disabled 0x1: Enabled	RW	0

**Table 10-800. Register Call Summary for Register VENC\_GEN\_CTRL**

Video Encoder

- [TV Detection/Disconnection Pulse Generation and Use: \[0\] \[1\]](#)
- [Video Encoder Global Initialization: \[2\] \[3\] \[4\] \[5\]](#)
- [Video Encoder Register Settings: \[6\]](#)
- [Video Encoder Register Summary: \[7\]](#)

**Table 10-801. VENC\_OUTPUT\_CONTROL**

<b>Address Offset</b>	0x0000 00C4	<b>Instance</b>	VENC
<b>Physical Address</b>	See <a href="#">Table 10-718</a>		
<b>Description</b>	Output channel control register Also contains some test control features		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																COMPOSITE_SOURCE	RESERVED	TEST_MODE	VIDEO_INVERT	RESERVED	COMPOSITE_ENABLE	RESERVED									

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00
6	COMPOSITE_SOURCE	Source of composite video data in test mode 0x0: Composite test data comes from internal register OUTPUT_TEST[9:0] 0x1: Composite test data comes from display controller video port G[1:0], B[7:0]	RW	0
5	RESERVED		RW	0x00
4	TEST_MODE	This enables the video DACs to be tested. The values sent to the DACs comes from a register for each output channel (Luma, Composite or Chroma) or from the display controller video port bits G[1:0], B[7:0], depending on the setting of the Source bits 0x0: Video outputs are in normal operation 0x1: Test mode. Video outputs are directly connected to either internal registers or the display controller video port.	RW	0
3	VIDEO_INVERT	Controls the video output polarity. This may be used to correct for inversion in an external video amplifier. 0x0: Video outputs are inverted 0x1: Video outputs are normal polarity	RW	1
2	RESERVED		RW	0x00
1	COMPOSITE_ENABLE	Enable the Composite output channel 0x0: Composite output is disabled 0x1: Composite output is enabled	RW	0

Bits	Field Name	Description	Type	Reset
0	RESERVED		RW	0x00

**Table 10-802. Register Call Summary for Register VENC\_OUTPUT\_CONTROL**

Video Encoder

- [Video Encoder Environment: \[0\]](#)
- [Video DAC Test Mode: \[1\] \[2\]](#)
- [Video DAC Power Management: \[3\]](#)
- [Video Encoder Global Initialization: \[4\] \[5\]](#)
- [Video Encoder Register Settings: \[6\]](#)
- [Video Encoder Register Summary: \[7\]](#)

**Table 10-803. VENC\_OUTPUT\_TEST**

<b>Address Offset</b>	0x0000 00C8	
<b>Physical Address</b>	See <a href="#">Table 10-718</a>	<b>Instance</b> VENC
<b>Description</b>	Test values for the Luma/Composite Video DAC	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																COMPOSITE_TEST															

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x00
9:0	COMPOSITE_TEST	In test mode, DAC input value (if composite video is selected)	RW	0x000

**Table 10-804. Register Call Summary for Register VENC\_OUTPUT\_TEST**

Video Encoder

- [Video DAC Test Mode: \[0\]](#)
- [Video Encoder Register Settings: \[1\]](#)
- [Video Encoder Register Summary: \[2\]](#)



## 2D/3D Graphics Accelerator

---

---

---

This chapter describes the 2D/3D graphics accelerator (SGX) for the device.

---

**NOTE:** The SGX subsystem is an instantiation by Texas Instruments of the POWERVR™ SGX540 core from Imagination Technologies Ltd.

This document contains materials that are © Imagination Technologies Ltd.

POWERVR and USSE are trademarks or registered trademarks of Imagination Technologies Ltd.

---

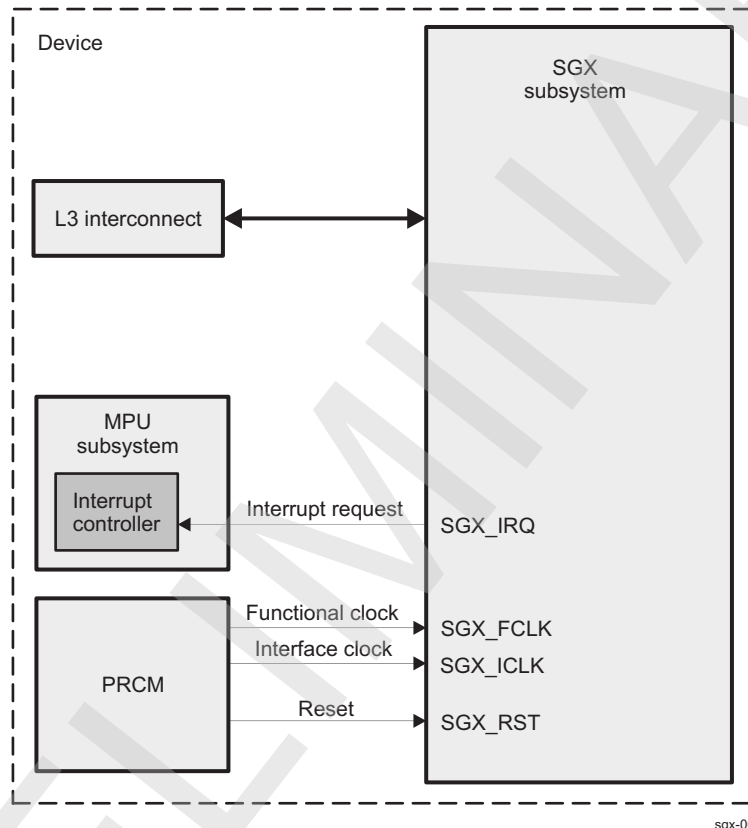
Topic	Page
11.1 SGX Overview .....	2610
11.2 SGX Integration .....	2613
11.3 SGX Functional Description .....	2615
11.4 SGX Register Manual .....	2618

## 11.1 SGX Overview

The 2D/3D graphics accelerator (SGX) subsystem accelerates 2-dimensional (2D) and 3-dimensional (3D) graphics applications. The SGX subsystem is based on the POWERVR® SGX core from Imagination Technologies. SGX is a new generation of programmable POWERVR graphics IP cores. The POWERVR SGX540 v1.2.0 architecture is scalable and can target all market segments from mainstream mobile devices to high-end desktop graphics. Targeted applications include feature phones, PDAs, and handheld gaming applications.

Figure 11-1 shows the SGX subsystem.

**Figure 11-1. SGX Overview**



The SGX graphics accelerator efficiently processes a number of various multimedia data types concurrently:

- Pixel data
- Vertex data
- General-purpose processing

The multithreaded architecture uses two levels of scheduling and data partitioning to enable zero overhead task switching.

### 11.1.1 POWERVR SGX Main Features

- 3D graphics, vector graphics, and video supported on common hardware
- Tile-based architecture
- Universal scalable shader engine (USSE™): a multithreaded engine that incorporates pixel and vertex shader functionality and reduces die area
- Advanced shader feature set in excess of Microsoft VS3.0, PS3.0, and OpenGL™2.0
- Industry standard application programming interface (API) support: OpenGL ES 1.1 and 2.0, OpenVG v1.1
- Fine-grained task switching, load balancing, and power management
- Advanced geometry, direct memory access (DMA)-driven operation for minimum CPU interaction
- Programmable high-quality image antialiasing
- Fully virtualized memory addressing for operating system (OS) operation in a unified memory architecture

### 11.1.2 SGX 3D Features

- Deferred pixel shading
- On-chip tile floating-point depth buffer
- 8-bit stencil with on-chip tile stencil buffer
- Scissor test
- Texture support:
  - Cube map
  - Projected textures
  - Nonsquare textures
- Texture formats:
  - RGBA 8888, 565, 1555
  - Monochromatic 8, 16, 16f, 32f, 32int
  - Dual channel, 8:8, 16:16, 16f:16f
  - Compressed textures PVR-TC1, PVR-TC2, ETC1
  - Programmable support for all YUV formats
- Resolution support:
  - Frame buffer maximum size = 2048 x 2048
  - Texture maximum size = 2048 x 2048
- Texture filtering:
  - Bilinear, trilinear, anisotropic
  - Independent minimum and maximum control
- Antialiasing:
  - 4x multisampling
  - Programmable sample positions
- Indexed primitive list support
  - Bus mastered
- Programmable vertex DMA
- Render to texture:
  - Including twiddled formats
  - Auto MipMap generation
- Multiple on-chip render targets (MRTs)

---

**NOTE:** Performance is limited when on-chip memory is not available for the SGX core.

---

### 11.1.3 SGX Universal Scalable Shader Engine – Key Features

The USSE is the engine core of the POWERVR SGX architecture and supports a broad range of instructions.

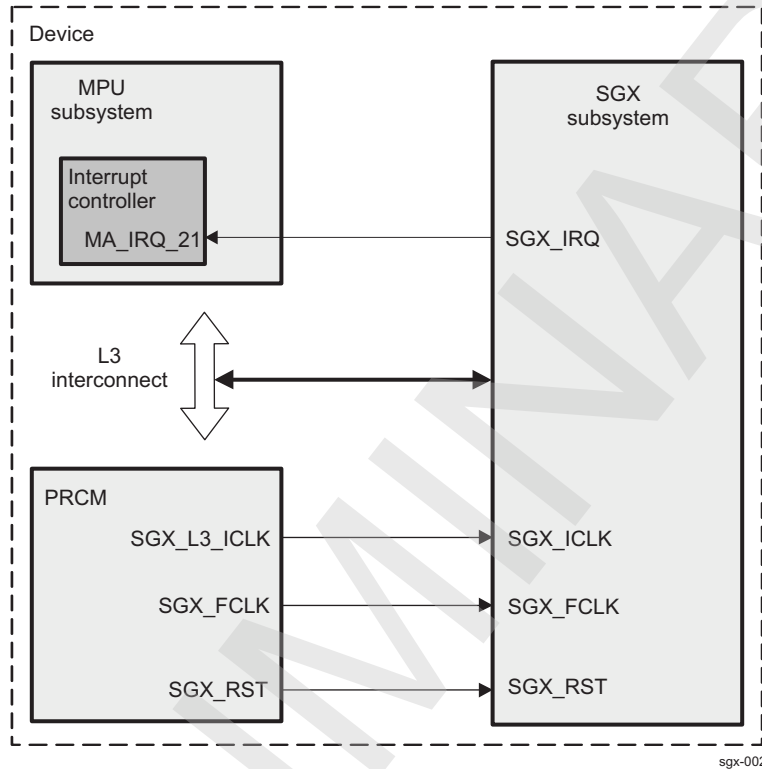
- Single programming model:
  - Multithreaded with 16 simultaneous execution threads and up to 64 simultaneous data instances
  - Zero-cost swapping in and out of threads
  - Cached program execution model
  - Dedicated pixel processing instructions
  - Dedicated vertex processing instructions
- SIMD execution unit supporting operations in:
  - 32-bit IEEE float
  - 2-way 16-bit fixed point
  - 4-way 8-bit integer
  - 32-bit bit-wise (logical only)
- Static and dynamic flow control:
  - Subroutine calls
  - Loops
  - Conditional branches
  - Zero-cost instruction predication
- Procedural geometry:
  - Allows generation of primitives
  - Effective geometry compression
  - High-order surface support
- External data access:
  - Permits reads from main memory using cache
  - Permits writes to main memory
  - Data fence facility
  - Dependent texture reads

## 11.2 SGX Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 11-2 shows SGX integration.

Figure 11-2. SGX Integration



The SGX subsystem is connected to the level 3 (L3) interconnect by a 128-bit master and a 64-bit slave interface.

Table 11-1 through Table 11-3 summarize the integration of the module in the device.

Table 11-1. Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
SGX	PD_GFX	L3

Table 11-2. Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SGX	SGX_ICLK	SGX_L3_ICLK	PRCM	SGX interface clock
SGX	SGX_FCLK	SGX_FCLK	PRCM	SGX functional clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SGX	SGX_RST	SGX_RST	PRCM	SGX reset signal

**Table 11-3. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
SGX	SGX_IRQ	MA_IRQ_21	Cortex-A9 INTC	SGX interrupt request to the MPU interrupt controller (INTC)

**NOTE:**

- For a description of the interrupt source, see [Section 11.3.5, Interrupt Requests](#).

### 11.3 SGX Functional Description

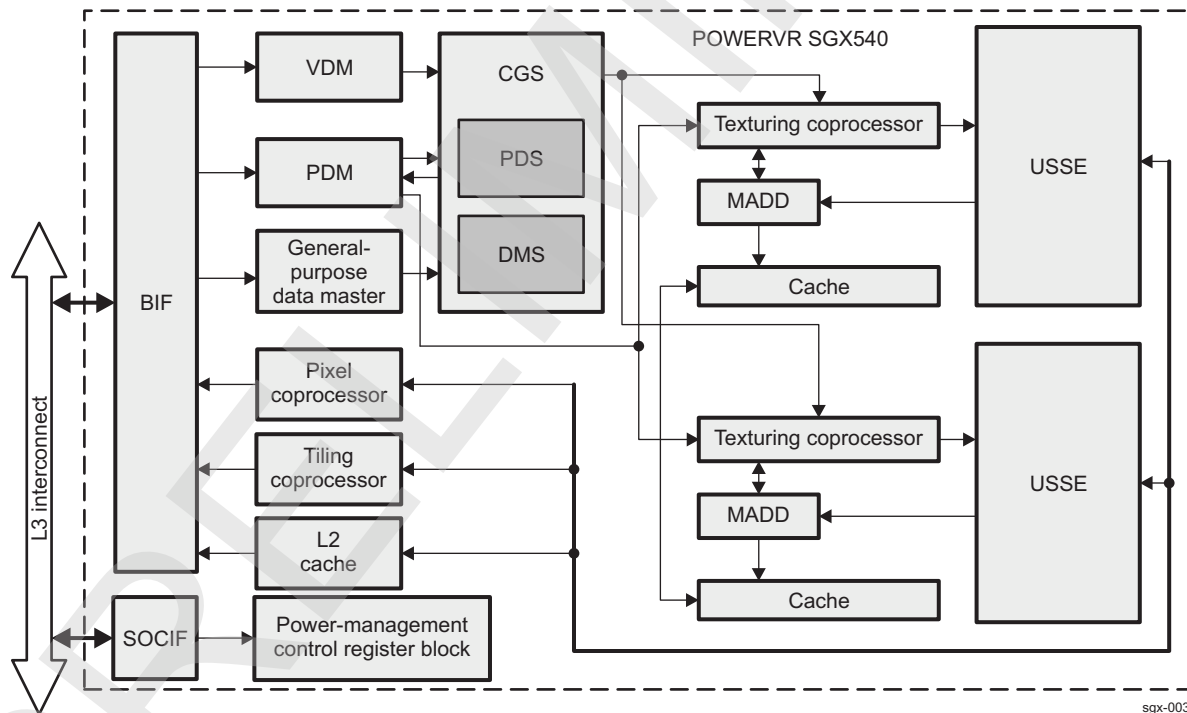
#### 11.3.1 SGX Block Diagram

The SGX subsystem is based on the POWERVR SGX540 core from Imagination Technologies. The architecture uses both programmable and hard-coded pipelines to perform various processing tasks required in 2D, 3D, and video processing. The SGX architecture comprises the following elements:

- Coarse grain scheduler (CGS)
  - Programmable data sequencer (PDS)
  - Data master selector (DMS)
- Vertex data master (VDM)
- Pixel data master (PDM)
- General-purpose data master
- USSE
- Tiling coprocessor
- Pixel coprocessor
- Texturing coprocessor
- Multilevel cache

Figure 11-3 shows the SGX block diagram.

Figure 11-3. SGX Block Diagram



##### 11.3.1.1 CGS

The CGS is the main system controller for the POWERVR SGX architecture and consists of two stages: the DMS and the PDS. The DMS processes requests from the data masters and determines which tasks can be executed given the resource requirements. The PDS then controls the loading and processing of data on the USSE.

##### 11.3.1.2 Data Masters

The SGX core includes three data masters:

- VDM: the initiator of transform and lighting processing within the system. The VDM reads an input control stream, which contains triangle index data and state data. The state data indicates the PDS program, the size of the vertices, and the amount of USSE output buffer resource available to the VDM. The triangle data is parsed to determine unique indices that must be processed by the USSE. These are grouped together based on the configuration provided by the driver and presented to the DMS.
- PDM: the initiator of the rasterization processing within the system. Each pixel pipeline processes pixels for a different half of a given tile, which allows for optimum efficiency within each pipe due to the locality of the data. For each task it determines the amount of resource required within the USSE. It merges this with the state address and issues a request for execution on the USSE to the DMS.
- General-purpose data master: responds to events within the system (such as the end of a pass of triangles from the ISP, end of a tile from the ISP, end of a render, or parameter stream breakpoint event). Each event causes either an interrupt to the host or the synchronized execution of a program on the PDS. The program might or might not cause a subsequent task to be executed on the USSE.

### 11.3.1.3 USSE

The USSE is a user-programmable processing unit. Although general in nature, its instructions and features are optimized for three types of task: processing vertices (vertex shading), processing pixels (pixel shading), and video/imaging processing.

### 11.3.1.4 Multilevel Cache

The multilevel cache is a 2-level cache consisting of two modules: the main cache and the mux/arbiter/demux/decompression unit (MADD). The MADD is a wrapper around the main cache module designed to manage and format requests to and from the cache as well as provide level 0 caching for texture and USSE requests.

The MADD can accept requests from the PDS, USSE, and texture address generator modules. Arbitration is performed between the three data streams; any required texture decompression is also performed.

### 11.3.1.5 Texturing Coprocessor

The texturing coprocessor performs texture address generation and formatting of texture data. It receives requests from the iterators or the USSE modules and translates these into requests to the multilevel cache. Data returned from the cache are then formatted based on the texture format selected and sent to the USSE for pixel-shading operations.

### 11.3.1.6 Tiling Coprocessor

To process pixels in a tiled manner, the screen is divided into tiles and arranged as groups of tiles by the tiling coprocessor. An inherent advantage of tiling architecture is that a large amount of vertex data can be rejected at this stage, thus reducing both the memory storage requirements and the amount of pixel processing performed.

### 11.3.1.7 Pixel Coprocessor

The pixel coprocessor is the final stage of the pixel-processing pipeline and controls the format of the final pixel data sent to the memory. It supplies the USSE with an address to the output buffer, and the USSE returns the relevant pixel data. The address order is determined by the mode of the frame buffer. The pixel coprocessor contains a dithering and packing function.

## 11.3.2 SGX Clock Configuration

The SGX subsystem operates from two clocks: an interface clock (SGX\_ICLK) and a functional clock (SGX\_FCLK). The power, reset, and clock management (PRCM) module generates and distributes both clocks inside the device.

- The SGX\_ICLK interface clock manages the data transfer on the L3 master and slave ports. The SGX\_ICLK frequency is selected based on the whole device L3 interconnect clock frequency. For more information on the interface clock, see [Section 3.6.11](#), *CD\_SGX Clock Domains*.



When no longer required by the SGX subsystem, SGX\_ICLK can be disabled by software at the PRCM level. For more information, see [Section 3.7.8, PD\\_SGX Description](#).

---

**NOTE:** SGX\_ICLK is cut only if the SGX is ready to go into IDLE state.

- SGX\_FCLK is the functional clock and is used inside the SGX subsystem to generate clock signals to multiple SGX clock domains.  
Depending on the clock source selection and the DPLL settings, SGX\_FCLK can support a frequency of up to 307 MHz.  
The SGX\_FCLK clock is provided by either the PRCM module or the peripheral DPLL, as described in [Section 3.6.11, CD\\_SGX Clock Domains](#). Selection is made at the PRCM level.  
When no longer needed by the SGX subsystem, SGX\_FCLK can be cut by software at the PRCM level, if the module is ready to enter the IDLE state. For more information, see [Section 3.7.8, PD\\_SGX Description](#).

### 11.3.3 SGX Software Reset

The SGX subsystem has its own reset domain. Global reset of the SGX is performed by activating the SGX\_RST signal in the SGX\_RST domain.

---

**NOTE:** The APIs delivered with the SGX provide a software reset functionally equivalent to a hardware reset.

### 11.3.4 SGX Power Management

The SGX subsystem has its own power domain (SGX power domain). For additional information about the SGX power domain, see [Section 3.7.8, PD\\_SGX Description](#), in [Chapter 3, Power, Reset, and Clock Management](#).

---

**NOTE:** The SGX handles automatic clock gating performed on the multiple internal module clock domains.

In addition, software-controlled clock gating is managed inside the SGX and handled by the related API delivered with the module.

### 11.3.5 SGX Interrupt Requests

The SGX subsystem can generate one interrupt signal to the MPU INTC mapped to SGX\_IRQ.

## 11.4 SGX Register Manual

### CAUTION

All SGX registers are limited to 32-bit data accesses; 8- and 16-bit accesses are not allowed because they can corrupt register content.

### 11.4.1 SGX Instance Summary

Table 11-4. SGX Instance Summary

Module Name	Base Address	Size
SGX	0x5600 0000	32MB

### 11.4.2 SGX OCP Registers

#### 11.4.2.1 SGX OCP Register Summary

Table 11-5. SGX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SGX Physical Address
<a href="#">OCP_REVISION</a>	R	32	0x0000 FE00	0x5600 FE00
<a href="#">OCP_HWINFO</a>	R	32	0x0000 FE04	0x5600 FE04
<a href="#">OCP_SYSCONFIG</a>	RW	32	0x0000 FE10	0x5600 FE10
<a href="#">OCP_IRQSTATUS_RAW_0</a>	RW	32	0x0000 FE24	0x5600 FE24
<a href="#">OCP_IRQSTATUS_RAW_1</a>	RW	32	0x0000 FE28	0x5600 FE28
<a href="#">OCP_IRQSTATUS_RAW_2</a>	RW	32	0x0000 FE2C	0x5600 FE2C
<a href="#">OCP_IRQSTATUS_0</a>	RW	32	0x0000 FE30	0x5600 FE30
<a href="#">OCP_IRQSTATUS_1</a>	RW	32	0x0000 FE34	0x5600 FE34
<a href="#">OCP_IRQSTATUS_2</a>	RW	32	0x0000 FE38	0x5600 FE38
<a href="#">OCP_IRQENABLE_SET_0</a>	RW	32	0x0000 FE3C	0x5600 FE3C
<a href="#">OCP_IRQENABLE_SET_1</a>	RW	32	0x0000 FE40	0x5600 FE40
<a href="#">OCP_IRQENABLE_SET_2</a>	RW	32	0x0000 FE44	0x5600 FE44
<a href="#">OCP_IRQENABLE_CLR_0</a>	RW	32	0x0000 FE48	0x5600 FE48
<a href="#">OCP_IRQENABLE_CLR_1</a>	RW	32	0x0000 FE4C	0x5600 FE4C
<a href="#">OCP_IRQENABLE_CLR_2</a>	RW	32	0x0000 FE50	0x5600 FE50
<a href="#">OCP_PAGE_CONFIG</a>	RW	32	0x0000 FF00	0x5600 FF00
<a href="#">OCP_INTERRUPT_EVENT</a>	RW	32	0x0000 FF04	0x5600 FF04
<a href="#">OCP_DEBUG_CONFIG</a>	RW	32	0x0000 FF08	0x5600 FF08
<a href="#">OCP_DEBUG_STATUS</a>	RW	32	0x0000 FF0C	0x5600 FF0C

#### 11.4.2.2 SGX OCP Register Description

**Table 11-6. OCP\_REVISION**

<b>Address Offset</b>	0x0000 FE00	
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .	<b>Instance</b> SGX
<b>Description</b>	OCP Revision Register.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISIONID																															

Bits	Field Name	Description	Type	Reset
31:0	REVISIONID	Revision value.	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 11-7. Register Call Summary for Register OCP\_REVISION**

- SGX Register Manual
- [SGX OCP Register Summary: \[0\]](#)

**Table 11-8. OCP\_HWINFO**

<b>Address Offset</b>	0x0000 FE04	
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .	<b>Instance</b> SGX
<b>Description</b>	Hardware implementation information	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
																	MEM_BUS_WIDTH	SYS_BUS_WIDTH													

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	MEM_BUS_WIDTH	Memory bus width: Read 0x0: Memory bus width is 64 bits. Read 0x1: Memory bus width is 128 bits.	R	-
1:0	SYS_BUS_WIDTH	System bus width: Read 0x0: System bus width is 32 bits. Read 0x1: System bus width is 64 bits. Read 0x2: System bus width is 128 bits. Read 0x3: Reserved	R	0x-

**Table 11-9. Register Call Summary for Register OCP\_HWINFO**

- SGX Register Manual
- [SGX OCP Register Summary: \[0\]](#)

**Table 11-10. OCP\_SYSCONFIG**

<b>Address Offset</b>	0x0000 FE10	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	System Configuration register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STANDBY_MODE		IDLE_MODE		RESERVED											

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5:4	STANDBY_MODE	Clock standby mode: 0x0: Force-standby mode 0x1: No-standby mode 0x2, 0x3: Smart-standby mode	RW	0x2
3:2	IDLE_MODE	Clock Idle mode: 0x0: Force-idle mode 0x1: No-idle mode 0x2, 0x3: Smart-idle mode	RW	0x2
1:0	RESERVED		R	0x0

**Table 11-11. Register Call Summary for Register OCP\_SYSCONFIG**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-12. OCP\_IRQSTATUS\_RAW\_0**

<b>Address Offset</b>	0x0000 FE24	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Raw IRQ 0 Status		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																INIT_MIN_INTERRUPT_RAW															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	INIT_MINTERRUPT_RAW	Interrupt 0 - Master port raw event: Write 0x0: No action. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event (used for debug).	RW	0

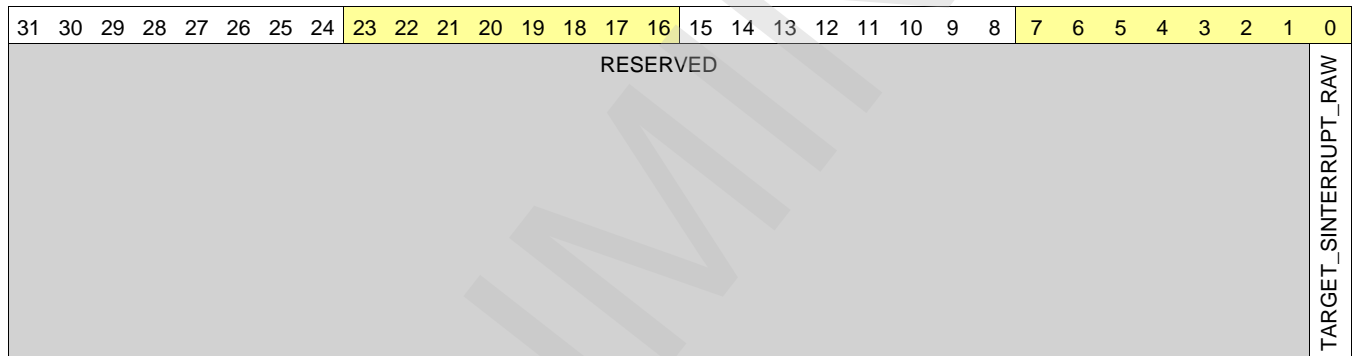
**Table 11-13. Register Call Summary for Register OCP\_IRQSTATUS\_RAW\_0**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-14. OCP\_IRQSTATUS\_RAW\_1**

<b>Address Offset</b>	0x0000 FE28	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Raw IRQ 1 Status. Slave port interrupt.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	TARGET_SINTERRUPT_RAW	Interrupt 1- Slave port raw event Write 0x0: No action. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event (used for debug).	RW	0

**Table 11-15. Register Call Summary for Register OCP\_IRQSTATUS\_RAW\_1**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-16. OCP\_IRQSTATUS\_RAW\_2**

<b>Address Offset</b>	0x0000 FE2C		
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .	<b>Instance</b>	SGX
<b>Description</b>	Raw IRQ 2 Status. Core interrupt.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												CORE_IRQ_RAW			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CORE_IRQ_RAW	Interrupt 2 - Core raw event Write 0x0: No action. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event (used for debug).	RW	0

**Table 11-17. Register Call Summary for Register OCP\_IRQSTATUS\_RAW\_2**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-18. OCP\_IRQSTATUS\_0**

<b>Address Offset</b>	0x0000 FE30		
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .	<b>Instance</b>	SGX
<b>Description</b>	Interrupt 0 Status event. Master port interrupt.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												INIT_MINTERRUPT_STATUS			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	INIT_MINTERRUPT_STATUS	Interrupt 0 - Master port status event Write 0x0: No action. Read 0x0: No event pending. Read 0x1: Event pending and interrupt enabled. Write 0x1: Clear event.	RW	0

**Table 11-19. Register Call Summary for Register OCP\_IRQSTATUS\_0**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-20. OCP\_IRQSTATUS\_1**

<b>Address Offset</b>	0x0000 FE34	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Interrupt 1 - slave port status event		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TARGET_SINTERRUPT_STATUS															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	TARGET_SINTERRUPT_STATU S	Interrupt 1 - Slave port status event Write 0x0: No action. Read 0x0: No event pending. Read 0x1: Event pending and interrupt enabled. Write 0x1: Clear event.	RW	0

**Table 11-21. Register Call Summary for Register OCP\_IRQSTATUS\_1**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-22. OCP\_IRQSTATUS\_2**

<b>Address Offset</b>	0x0000 FE38	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Interrupt 2 - Core status event		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CORE_IRQ_STATUS															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CORE_IRQ_STATUS	Interrupt 2 - Core status event Write 0x0: No action. Read 0x0: No event pending. Read 0x1: Event pending and interrupt enabled. Write 0x1: Clear event.	RW	0

**Table 11-23. Register Call Summary for Register OCP\_IRQSTATUS\_2**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-24. OCP\_IRQENABLE\_SET\_0**

<b>Address Offset</b>	0x0000 FE3C	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Enable Interrupt 0 - Master port		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																INIT_MINTERRUPT_ENABLE																

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	INIT_MINTERRUPT_ENABLE	Enable interrupt 0 - Master port Write 0x0: No action. Read 0x0: Interrupt is enabled. Read 0x1: Interrupt is disabled. Write 0x1: Enable interrupt.	RW	0

**Table 11-25. Register Call Summary for Register OCP\_IRQENABLE\_SET\_0**

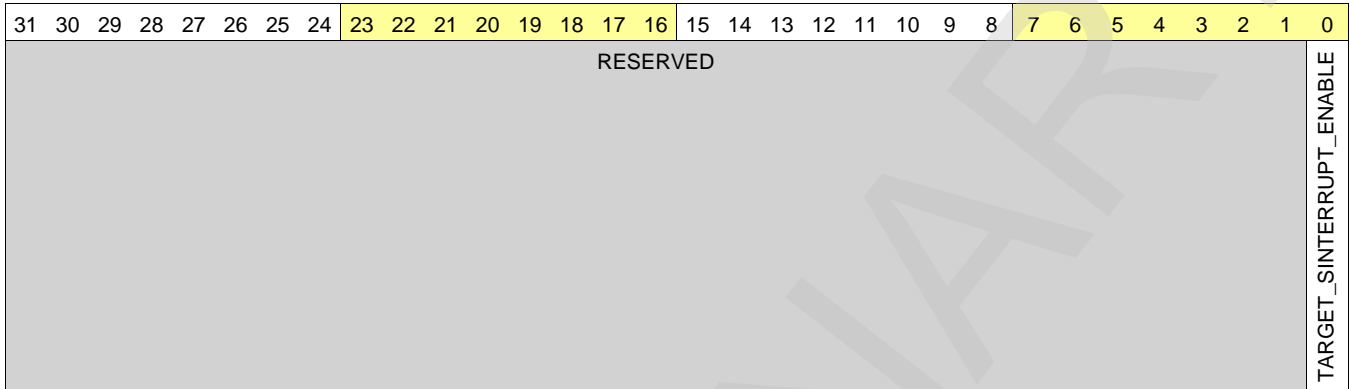
SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)



**Table 11-26. OCP\_IRQENABLE\_SET\_1**

<b>Address Offset</b>	0x0000 FE40	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Enable Interrupt 1. Target port interrupt.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	TARGET_SINTERRUPT_ENAB LE	Enable interrupt 1 - Slave port interrupt Write 0x0: No action. Read 0x0: Interrupt is enabled. Read 0x1: Interrupt is disabled. Write 0x1: Enable interrupt.	RW	0

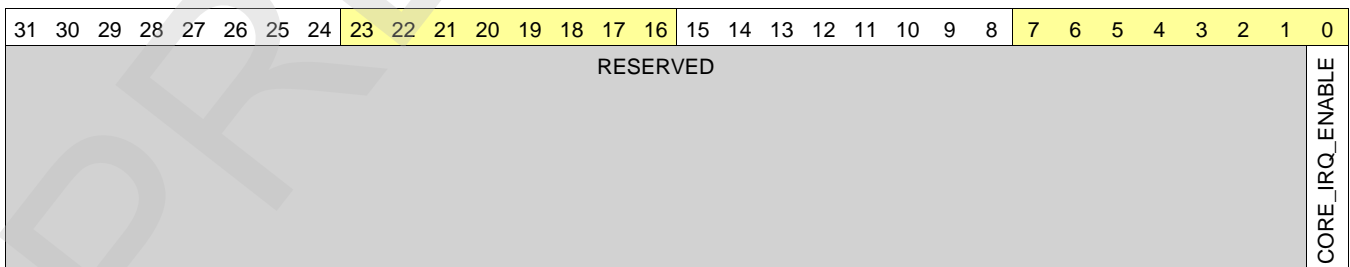
**Table 11-27. Register Call Summary for Register OCP\_IRQENABLE\_SET\_1**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-28. OCP\_IRQENABLE\_SET\_2**

<b>Address Offset</b>	0x0000 FE44	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Enable Interrupt 2. Core interrupt.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CORE_IRQ_ENABLE	Enable interrupt 2 - Core interrupt Write 0x0: No action. Read 0x0: Interrupt is enabled. Read 0x1: Interrupt is disabled. Write 0x1: Enable interrupt.	RW	0

**Table 11-29. Register Call Summary for Register OCP\_IRQENABLE\_SET\_2**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-30. OCP\_IRQENABLE\_CLR\_0**

<b>Address Offset</b>	0x0000 FE48	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Disable Interrupt 0 - Master port		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
																INIT_MINTERRUPT_DISABLE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	INIT_MINTERRUPT_DISABLE	Disable interrupt 0 - Master port Write 0x0: No action. Read 0x0: Interrupt is enabled. Read 0x1: Interrupt is disabled. Write 0x1: Disable interrupt.	RW	0

**Table 11-31. Register Call Summary for Register OCP\_IRQENABLE\_CLR\_0**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-32. OCP\_IRQENABLE\_CLR\_1**

<b>Address Offset</b>	0x0000 FE4C	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Disable Interrupt 1 - slave port		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												TARGET_SINTERRUPT_DISABLE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	TARGET_SINTERRUPT_DISABLE	Disable interrupt 1 - Slave port Write 0x0: No action. Read 0x0: Interrupt is enabled. Read 0x1: Interrupt is disabled. Write 0x1: Disable interrupt.	RW	0

**Table 11-33. Register Call Summary for Register OCP\_IRQENABLE\_CLR\_1**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-34. OCP\_IRQENABLE\_CLR\_2**

<b>Address Offset</b>	0x0000 FE50	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Disable Interrupt 2 - Core interrupt		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												CORE_IRQ_DISABLE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	CORE_IRQ_DISABLE	Disable interrupt 2 - Core interrupt Write 0x0: No action. Read 0x0: Interrupt is enabled. Read 0x1: Interrupt is disabled. Write 0x1: Disable interrupt.	RW	0

**Table 11-35. Register Call Summary for Register OCP\_IRQENABLE\_CLR\_2**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-36. OCP\_PAGE\_CONFIG**

<b>Address Offset</b>	0x0000 FF00	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Configure memory pages..		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											OCP_PAGE_SIZE	MEM_PAGE_CHECK_EN	MEM_PAGE_SIZE		

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000000
4:3	OCP_PAGE_SIZE	Defines the page size on OCP memory interface 0x0: Page size is 4KB. 0x1: Page size is 2KB 0x2: Page size is 1KB. 0x3: Page size is 512B.	RW	0x2
2	MEM_PAGE_CHECK_EN	Enable page boundary checking. 0x0: Page boundary checking disabled. 0x1: Page boundary checking enabled.	RW	1
1:0	MEM_PAGE_SIZE	Defines the page size on internal memory interface 0x0: Page size is 4KB. 0x1: Page size is 2KB 0x2: Page size is 1KB. 0x3: Page size is 512B.	RW	0x0

**Table 11-37. Register Call Summary for Register OCP\_PAGE\_CONFIG**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-38. OCP\_INTERRUPT\_EVENT**

<b>Address Offset</b>	0x0000 FF04	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Interrupt events		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED																TARGET_INVALID_OCP_CMD			TARGET_CMD_FIFO_FULL			TARGET_RESP_FIFO_FULL			RESERVED			INIT_MEM_REQ_FIFO_OVERRUN		INIT_READ_TAG_FIFO_OVERRUN		INIT_PAGE_CROSS_ERROR		INIT_RESP_ERROR		INIT_RESP_UNUSED_TAG		INIT_RESP_UNEXPECTED	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0000000
10	TARGET_INVALID_OCP_CMD	Invalid command from OCP Write 0x0: Clear the event. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event and interrupt if enabled (debug only).	RW	0
9	TARGET_CMD_FIFO_FULL	Command FIFO full Write 0x0: Write 0 to clear the event. Read 0x0: Read 0 implies no event pending. Read 0x1: Read 1 indicates event pending. Write 0x1: Write 1 to set event and interrupt if enabled (debug only).	RW	0
8	TARGET_RESP_FIFO_FULL	Response FIFO full. Write 0x0: Clear the event. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event and interrupt if enabled (debug only).	RW	0
7:6	RESERVED		R	0x0
5	INIT_MEM_REQ_FIFO_OVERRUN	Memory request FIFO overrun. Write 0x0: Clear the event. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event and interrupt if enabled (debug only).	RW	0
4	INIT_READ_TAG_FIFO_OVERRUN	Read tag FIFO overrun. Write 0x0: Clear the event. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event and interrupt if enabled (debug only).	RW	0

Bits	Field Name	Description	Type	Reset
3	INIT_PAGE_CROSS_ERROR	Memory page had been crossed during a burst. Write 0x0: Clear the event. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event and interrupt if enabled (debug only).	RW	0
2	INIT_RESP_ERROR	Receiving error response Write 0x0: Clear the event. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event and interrupt if enabled (debug only).	RW	0
1	INIT_RESP_UNUSED_TAG	Receiving response on an unused tag Write 0x0: Clear the event. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event and interrupt if enabled (debug only).	RW	0
0	INIT_RESP_UNEXPECTED	Receiving response when not expected Write 0x0: Clear the event. Read 0x0: No event pending. Read 0x1: Event pending. Write 0x1: Set event and interrupt if enabled (debug only).	RW	0

**Table 11-39. Register Call Summary for Register OCP\_INTERRUPT\_EVENT**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-40. OCP\_DEBUG\_CONFIG**

<b>Address Offset</b>	0x0000 FF08	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Configuration of debug modes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_INT_BYPASS								RESERVED																SELECT_INIT_IDLE	FORCEPASSDATA	FORCEINITIDLE	FORCETARGETIDLE				

Bits	Field Name	Description	Type	Reset
31	CORE_INT_BYPASS	Bypass OCP IPG interrupt logic. 0x0: Don't Bypass. 0x1: Bypass core interrupt to IO pin, ie disregard the interrupt enable setting in IPG register.	RW	0
30:6	RESERVED		R	0x0000000

Bits	Field Name	Description	Type	Reset
5	SELECT_INIT_IDLE	To select which idle the disconnect protocol should act on 0 0x0: Whole SGX Idle. 0x1: OCP initiator idle only.	RW	0
4	FORCE_PASS_DATA	Forces the initiator to pass data independent of disconnect protocol 0x0: Normal mode. Don't force. 0x1: Never fence request to OCP.	RW	0
3:2	FORCE_INIT_IDLE	Forces the OCP master port to idle. 0x0: Normal mode - no force. 0x1: Force port to be always idle. 0x2: Forces target port to never be in idle mode. 0x3: Normal mode. No force.	RW	0x0
1:0	FORCE_TARGET_IDLE	Forces the OCP target port to idle. 0x0: Normal mode - no force. 0x1: Force port to be always idle. 0x2: Forces target port to never be in idle mode. 0x3: Normal mode. No force.	RW	0x0

**Table 11-41. Register Call Summary for Register OCP\_DEBUG\_CONFIG**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

**Table 11-42. OCP\_DEBUG\_STATUS**

<b>Address Offset</b>	0x0000 FF0C	<b>Instance</b>	SGX
<b>Physical Address</b>	See <a href="#">Table 11-5</a> .		
<b>Description</b>	Status of debug.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD_DEBUG_STATE	CMD_RESP_DEBUG_STATE	TARGET_IDLE	RESP_FIFO_FULL	CMD_FIFO_FULL	RESP_ERROR			WHICH_TARGET_REGISTER			TARGET_CMD_OUT			INIT_MSTANDBY	INIT_MWAIT	INIT_MDISCREQ			INIT_MDISCACK	INIT_SCONNECT2	INIT_SCONNECT1	INIT_SCONNECT0		INIT_MCONNECT	TARGET_SIDLEACK	TARGET_SDISCACK	TARGET_SIDLEREQ	TARGET_SCONNECT		TARGET_MCONNECT	

Bits	Field Name	Description	Type	Reset
31	CMD_DEBUG_STATE	Target command state-machine 0x0: Idle 0x1: Accept command.	RW	-
30	CMD_RESP_DEBUG_STATE	Target response state-machine 0x0: Send accept. 0x1: Wait accept.	RW	-
29	TARGET_IDLE	Target idle	R	-
28	RESP_FIFO_FULL	Target response FIFO full	R	-

Bits	Field Name	Description	Type	Reset
27	CMD_FIFO_FULL	Target command FIFO full	R	–
26	RESP_ERROR	Respond to OCP with error, which could be caused by either address misalignment or invalid byte enable.	R	–
25:21	WHICH_TARGET_REGISTER	Indicates which OCP target registers to read	RW	0bxxxxx
20:18	TARGET_CMD_OUT	Command received from OCP Read 0x0: Command WRSYS received Read 0x1: Command RDSYS received Read 0x2: Command WR_ERROR received Read 0x3: Command RD_ERROR received Read 0x4: Command CHK_WRADDR_PAGE received. Not used. Read 0x5: Command CHK_RDADDR_PAGE received. Not used. Read 0x6: Command TARGET_REG_WRITE received. Read 0x7: Command TARGET_REG_READ received	R	0bxxx
17	INIT_MSTANDBY	Status of init_MStandby signal	R	–
16	INIT_MWAIT	Status of init_MWait signal	R	–
15:14	INIT_MDISCREQ	Disconnect status of the OCP interface Read 0x0: State is FUNCT Read 0x1: State is SLEEP TRANS Read 0x2: Reserved Read 0x3: State is IDLE.	R	0bxx
13	INIT_MDISCACK	Memory request FIFO full Write 0x0: Clear the event. Read 0x0: No event pending Read 0x1: Event pending Write 0x1: Set the event and interrupt if enabled (debug only)	RW	–
12	INIT_SCONNECT2	Defines whether to wait in M_WAIT state for MConnect FSM Read 0x0: Skip M_WAIT state. Read 0x1: Wait in M_WAIT state.	R	–
11	INIT_SCONNECT1	Defines the busy-ness state of the slave Read 0x0: Slave is drained. Read 0x1: Slave is loaded.	R	–
10	INIT_SCONNECT0	Disconnect from slave Read 0x0: Disconnect request from slave. Read 0x1: Connect request from slave.	R	–
9:8	INIT_MCONNECT	Initiator MConnect state Read 0x0: State is M_OFF. Read 0x1: State is M_WAIT. Read 0x2: State is M_DISC. Read 0x3: State is M_CON.	R	0bxx
7:6	TARGET_SIDLEACK	Acknowledge the SIdleAck state machine Read 0x0: State is FUNCT. Read 0x1: State is SLEEP TRANS. Read 0x2: Reserved Read 0x3: State is IDLE.	R	0bxx



Bits	Field Name	Description	Type	Reset
5:4	TARGET_SDISCACK	Acknowledge the SDiscAck state-machine Read 0x0: State is FUNCT. Read 0x1: State is TRANS. Read 0x2: Reserved Read 0x3: State is IDLE.	R	0bxx
3	TARGET_SIDLREQ	Request the target to go idle. Read 0x0: Don't go idle, or go active. Read 0x1: Go idle.	R	–
2	TARGET_SCONNECT	Target SConnect state Read 0x0: Disconnect interface. Read 0x1: Connect OCP interface.	R	–
1:0	TARGET_MCONNECT	Target MConnect state Read 0x0: Target is in M_OFF state. Read 0x1: Target is in M_WAIT disconnect state. Read 0x2: Target is in M_DISC state. Read 0x3: Target is in M_CON state.	R	0bxx

**Table 11-43. Register Call Summary for Register OCP\_DEBUG\_STATUS**

SGX Register Manual

- [SGX OCP Register Summary: \[0\]](#)

PRELIMINARY

## Audio Back End

---

---

---

This chapter describes the audio back end (ABE) module.

Topic	Page
12.1 ABE Overview .....	2636

## 12.1 ABE Overview

The audio back end (ABE) module is a subsystem that manages various audio and voice uplink and downlink streams between the initiator (the Cortex™-A9 microprocessor unit [MPU], digital signal processor [DSP], or direct memory access [DMA] controller) and the peripheral physical interfaces (multichannel buffered serial port [McBSP], SLIMbus®, digital microcontroller [DMIC], multichannel pulse density modulation [McPDM], and multichannel audio serial port [McASP]).

The ABE module handles the audio processing for the applications. It receives voice or audio samples from the initiator or the external audio chip (TWL6040 or other) and sends them to the peripheral interfaces or memories after processing. The ABE can perform buffering of audio samples, mix audio with a voice down stream and/or a microphone up stream (sidetone), and can apply some post-processing.

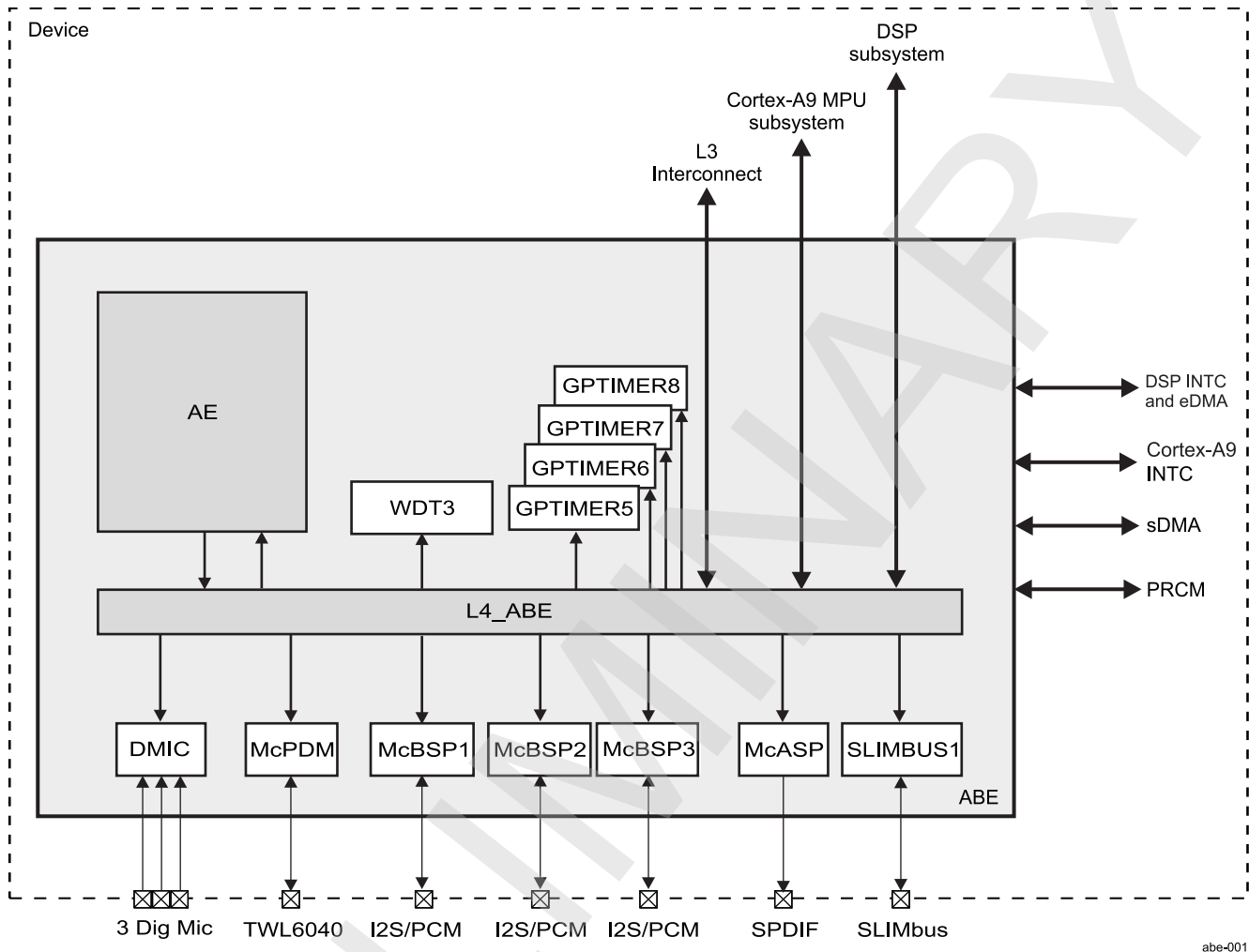
The ABE integrates the following modules:

- Peripheral physical interfaces:
  - Three McBSP modules
  - One McASP module
  - One mobile industry processor interface (MIPI) SLIMbus interface
  - One DMIC module to support up to six mono and up to three stereo digital microphones
  - One MCPDM module to support interconnect with an external audio chip.
- Audio engine (AE):
  - Performs real-time signal processing like sample rate conversion, filtering, equalizing, and side-tone
  - Audio traffic controller (ATC):
    - Performs data move in the ABE
    - Generates interrupt requests to the DSP and Cortex-A9 interrupt controllers
  - On-chip memory:
    - 64KB of RAM; 32-bit data memory (DMEM)
    - 6KB for coefficient memory RAM (CMEM)
    - 18KB for sample memory RAM (SMEM)
- Local L4 interconnect (L4\_ABE) to:
  - L3 interconnect
  - Cortex-A9 MPU
  - DSP

The Cortex-A9 MPU and the DSP have private access to the L4\_ABE interconnect. All other initiators (sDMA) can access the L4\_ABE only through the L3 interconnect.
- Timers:
  - Four general-purpose (GP) timers
  - One watchdog timer
- Clock and reset management: Receives clock and reset signals from the device PRCM module. The audio engine operates from the DPLL\_ABE, which is in the PRCM module. The ABE modules can be supplied with clock signals from the DPLL\_ABE or the DPLL\_PER. The ABE has its own hardware reset domain (AUDIO\_RST).
- Power management: The ABE module is power-independent. It has its own dedicated power domain (PD\_ABE) and can execute audio processing with the rest of the device in retention or in off mode.
- Each peripheral in the ABE can be accessed directly by the initiator when the AE is bypassed (legacy mode, for old software compatibility).

Figure 12-1 is a high-level overview of the ABE.

Figure 12-1. Audio Back End



abe-001

### 12.1.1 AE Subsystem

The AE is the core of the ABE subsystem. It performs the real-time audio processing within the ABE: mixing, muxing, volume control, smooth muting, sampling rate conversion, and side-tone equalization. The AE subsystem also processes and executes all data transfers. To optimize the global processing, the ATC moves the data. An IRQ/DMA controller controls system IRQ and DMA requests. Input DMA requests come from the ABE peripherals, and output DMA requests and IRQ are generated and sent to the system (Cortex-A9 MPU interrupt controller, DSP interrupt controller, system DMA [sDMA], and the DSP enhanced direct memory access controller [eDMA]).

The AE subsystem contains:

- AE: processes the sample in SMEM with help of CMEM.
  - SMEM
  - CMEM
- DMEM contains the audio buffer and communication circular buffer.
- ATC: Transfers data between all modules in the ABE
- IRQ DMA controller

### 12.1.2 Local Audio Interconnect

The ABE has its own audio local interconnect (L4\_ABE), which interconnects and manages the data flow among all ABE modules.

L4\_ABE main features are:

- Initiator ports: Cortex-A9 MPU, DSP, L3 interconnect, and AE subsystem. The sDMA operates with the L4\_ABE interconnect through the L3 interconnect initiator port. The eDMA operates with the L4\_ABE interconnect through the DSP initiator port.
- Target ports: AE subsystem, McBSP1/2/3, DMIC, McPDM, McASP, GPTIMER5/6/7/8, WDT3, and SLIMBUS1
- 8-, 16-, or 32-bit data, single access
- 4-bit x 32-bit initiator ports for each interconnect instance
- Auto clock-gating feature
- Little-endian transaction assumed for all packing/unpacking operations
- Nonblocking architecture with fair arbitration between threads
- Target interconnect interfaces support fully synchronous and divided synchronous accesses.

For more information, see [Chapter 13, Interconnect](#).

### 12.1.3 GP Timers

The ABE embeds four GP timers: GPTIMER5 through GPTIMER8. All GP timers have interrupts connected to the Cortex-A9 MPU, DSP, and AE subsystems.

GP timer controllers have the following functions:

- Interconnect slave interface (L4) supports:
  - 32-bit data bus width
  - 16-/32-bit access supported
  - 10-bit address bus width
  - Burst not supported
  - WNP (Write-Non-Posted) supported
- Interrupts generated on overflow, compare, and capture
- Free-running 32-bit upward counter
- Supported modes:
  - Compare and capture modes
  - Auto-reload mode
  - Start-stop mode
- Programmable divider clock source
- Dedicated input trigger for capture mode and dedicated OUT trigger/PWM signal
- On-the-fly read/write register
- Supports disconnect protocol

For more information, see [Section 22.2, General-Purpose Timers](#).

### 12.1.4 Watchdog Timer

The ABE embeds one watchdog timer, WDT3, that generates an interrupt condition to the Cortex-A9 MPU on overflow, which can be used by the application software through the PRCM module to indirectly trigger a global warm reset.

WDT3 main features are:

- Interconnect slave interface (L4) supports:
  - 32-bit data bus width
  - 16- and 32-bit access supported
  - 11-bit address bus width
  - Burst not supported

- WNP supported
- Free-running 32-bit upward counter
- Programmable divider clock source
- Watchdog is reset on power up or after a warm reset and then starts counting.
- Reset or interrupt actions on a timer overflow condition and according to the watchdog timer source
- Support of disconnect protocol

For more information, see [Section 22.3](#), *Watchdog Timers*.

### 12.1.5 *Digital Microphone Controller*

The DMIC allows support of up to three digital stereo microphones that send it a pulse-density modulated stream of bits, transferred on one period or one half-period of the clock (over-sampling clock) provided to the DMIC. Each microphone is directly connected to the TX filter decimator to extract the audio samples with a maximum OF 96 db SNR with a frequency sampling set to 96 kHz.

The DMIC has the following features:

- Six external pin connections (three data lines and three clock lines)
- Delivers one common clock (on the three clock lines) for all digital microphones
- Supports idle request/acknowledge protocol
- Supports stereo and mono digital microphones (up to three)
- Rising or falling edge configuration for the clock signal sampling
- DMIC-clock-programmable
- Interconnect slave interface (internal interconnect) supports 32-bit data bus width.
- One DMA request capability on a programmable FIFO threshold
- One RX FIFO (16-bit x 24-bit word depth) per microphone
- Complies with PRCM interrupts to the Cortex-A9 MPU and DSP subsystems
- Interconnect sample format: 32 bits (only 24 are significant)
- Supports idle request/acknowledge PRCM protocol

For more information, see [Section 23.7](#), *DMIC*.

### 12.1.6 *SLIMbus*

There is one SLIMbus module instantiation inside the ABE, SLIMbus1. The SLIMbus controller provides a bidirectional, multidrop, multichannel 2-line serial interface between the ABE and up to seven off-chip components in a system such as an audio codec, a Bluetooth® module, or an FM radio receiver/transmitter. Because of its versatility, SLIMbus can accommodate a wide range of peripherals and clocked frame-oriented protocols (I2S, pulse code modulation [PCM], time division multiplexing [TDM]).

SLIMbus has the following features:

- Supports idle request/acknowledge PRCM protocol
- Complies with MIPI SLIMbus standard version 1.0
- Interconnect slave interface (L4) supports:
  - 32-bit data bus width
  - 8-, 16-, and 32-bit access supported
  - 11-bit address bus width
  - Burst not supported
  - WNP supported
- Support of seven SLIMbus devices:
  - One manager
  - One framer
  - One interface
  - Four generic
- Message protocol partially implemented in SLIMbus module hardware:

- Information elements are not stored in the module.
- Message interpretation is not processed in the module.
- FIFO size: 32 bytes for TX, 32 bytes for RX
- Support for eight TX and eight RX independent ports (data channels):
  - Supported transport protocols: Isochronous, pulled, pushed, asynchronous
  - Programmable segment data length (sample size) from 8 to 32 bits (8, 12, 16, 20, 24, 28, 32), including DATA and AUX
  - Programmable FIFO sizes
  - Programmable port assignment to any of the four generic devices
  - One DMA request line per port
- Operating frequencies of 0 to 24.576 MHz
- Supports disconnect protocol

For more information, see [Section 23.9](#), *Serial Low-Power Inter-Chip Media Bus Controller*.

### 12.1.7 Multichannel Audio Serial Port

The multichannel audio serial port (McASP) is a GP audio serial port. It is useful for intercomponent (DIT) transmission. The McASP is intended to be flexible enough so that it may connect gluelessly to audio A/D, D/A, CODEC and SPDIF transmit physical layer components.

McASP has the following features:

- 2-interconnect slave interface supports:
  - One interconnect configuration slave interface
  - One interconnect DMA slave interface
  - 32-bit data bus width
  - 16- and 32-bit access
  - 10-bit address bus width
  - Burst not supported
  - WNP
- Supports idle request/acknowledge protocol
- Buffers for transmit/receive operations
- DMA requests (one per direction) link with the 32-bit register
- Up to four transmit channels
- Four serializers implemented
- Support of disconnect protocol

For more information, see [Section 23.8](#), *McASP*.

### 12.1.8 Multichannel Buffered Serial Port

There are three McBSP instantiations: McBSP1 through McBSP3. They provide a full-duplex direct serial interface between the ABE and external devices, such as other modems, BT chips, or codecs. Because of its versatility, a McBSP can accommodate a wide range of peripherals and clocked frame-oriented protocols (I2S, PCM, TDM).

List of recommended usage per McBSP:

- MCBSP1: BT voice/audio data
- MCBSP2: Digital baseband (DBB) voice data
- MCBSP3: MIDI FM data

The McBSP has the following features:

- Interconnect slave interface (internal interconnect) supports:
  - 32-bit data bus width
  - 8-, 16-, and 32- bit access
  - 10-bit address bus width



- Burst not supported
- WNP
- Buffers for transmit/receive operations: 128/128 32-bit words (McBSP1, 2, 3)
- Interrupts configurable in legacy mode (two requests) or PRCM-compliant (one request)
- DMA requests (one per direction) triggered with programmable FIFO thresholds
- Multidrop support
- Serial interface description
- Four-pin configuration (McBSP1, 2, 3)
- Full-duplex communication
- Multichannel selection modes
- Support to enable or block transfers in each channel
- Up to 128 channels for transmission and for reception
- Supported protocols
- Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected A/D and D/A devices:
  - I2S-compliant devices
  - TDM bus devices
  - PCM devices
- A wide selection of data sizes: 8, 12, 16, 20, 24, and 32 bits
- Supports idle request/acknowledge protocol
- Clock generation support:
  - Independent clocking and framing for reception and for transmission up to 48 MHz
  - Support for external generation of clock signals and frame-synchronization (frame-sync) signals
  - A programmable sample rate generator for internal generation and control of clock signals and frame-sync signals
- Support of disconnect protocol

For a complete description, see [Section 23.5](#), *Multichannel Buffered Serial Port*.

### **12.1.9 Multichannel Pulse Density Modulation Module**

The McPDM is a proprietary interface based on multichannel pulse density modulation. The module consists of five downlinks and three uplinks. Two other uplink channels are reserved for status communication.

The McPDM has the following features:

- Interconnect slave interface (L4\_ABE internal interconnect) supports:
  - 32-bit data bus width
  - 10-bit address bus width
  - Burst not supported
  - WNP
- Full-duplex communication:
  - Five audio downstream channels
  - three audio upstream channels
- RX/TX FIFO operations: 32-bit words per channels
- Complies with PRCM interrupts:
  - One to the MPU subsystem
  - One to the DSP subsystem
- DMA requests (one per direction) triggered with programmable FIFO thresholds; depending on the FIFO implementation, one per channel or one per direction
- Decimation filter for embedded uplink paths (five paths if two status paths)
- Oversampling for embedded uplinks (five paths)
- Sigma delta for embedded downlinks (three paths)

- Deserializers for the two status upstream channels
- Support of disconnect protocol

For more information, see [Section 23.6](#), *Multichannel PDM Controller*.

PRELIMINARY

## Interconnect

This chapter describes the device interconnect.

---

**NOTE:** The L3 interconnect is instantiation of the NoC interconnect from Arteris, Inc. Arteris is a registered trademark of Arteris, Inc.

This document contains materials that are ©Arteris, Inc., and that constitute proprietary information of Arteris, Inc.

All materials and trademarks are used under license from Arteris, Inc. For additional information, see the Arteris Reference manuals, or contact Arteris, Inc.

NoC is an abbreviation for Network On Chip.

---

**NOTE:** The L4 interconnects are instantiations of the Sonics3220™ interconnect from Sonics, Inc.

This document contains materials that are ©2003-2009 Sonics, Inc., and that constitute proprietary information of Sonics, Inc.

SonicsMX and Sonics3220, are trademarks or registered trademarks of Sonics, Inc. All such materials and trademarks are used under license from Sonics, Inc. For additional information, see the SonicsMX or Sonics3220 Reference manuals, or contact Sonics, Inc.

SMX is an abbreviation for SonicsMX

---

Topic	Page
<b>13.1 Interconnect Overview</b> .....	<b>2644</b>
<b>13.2 L3 Interconnect</b> .....	<b>2652</b>
<b>13.3 L4 Interconnects</b> .....	<b>2750</b>

## 13.1 Interconnect Overview

### 13.1.1 Terminology

The following terminology is critical to understanding the interconnect:

- **Initiator:** Module able to initiate read and write requests to the chip interconnect (typically: processors, DMA, etc.).
- **Target:** Unlike an initiator, a target module cannot generate read/write requests to the chip interconnect, but it can respond to these requests. However, it may generate interrupts or a DMA request to the system (typically: peripherals, memory controllers).

---

**NOTE:** A module can have several separate ports; therefore, a module can be an initiator and a target.

---

- **Agent:** Each connection of one module to one interconnect is done using an agent, which is an adaptation (sometimes configurable) between the module and the interconnect. A target module is connected by a target agent (TA), and an initiator module is connected by an initiator agent (IA).
- **Interconnect:** The decoding, routing, and arbitration logic that enables the connection between multiple initiator modules and multiple target modules connected on it.
- **Register target (RT):** Special TA used to access the interconnect internal configuration registers
- **Data-flow signal:** Any signal that is part of a clearly identified transfer or data flow (typically: command, address, byte enables, etc.). Signal behavior is defined by the protocol semantics.
- **Sideband signal:** Any signal whose behavior is not associated to a precise transaction or data flow.
- **Out-of-band error:** Any signal whose behavior is associated to a device error-reporting scheme, as opposed to in-band errors.

---

**NOTE:** Interrupt requests and DMA requests are not routed by the interconnect in the device.

---

- **Firewall:** A programmable feature integrated in a TA or L4 interconnect to prevent unauthorized access to or from a module. A firewall can be configured using three criteria:
  - Initiator requesting access
  - Address space access
  - Type of access
- **ConnID:** Any transaction in the system interconnect is tagged by an in-band qualifier ConnID, which uniquely identifies the initiator at a given interconnect point. A ConnID is transmitted in band with the request and is used for firewall and error-logging mechanism.
- **Firewall comparison mechanism:** A comparison made in the firewall between access in-band qualifiers and access permissions that are programmed in the firewall configuration registers. If the comparison is successful, access is allowed; otherwise, access is denied.
- **MCcmd qualifier:** Command bus that indicates the type of transfer requested. [Table 13-1](#) lists the commands encoded.

**Table 13-1. MCcmd Qualifier Description**

MCcmd[2:0]	Transaction Type
0 0 0	Idle
0 0 1	Write
0 1 0	Read
0 1 1	ReadEx
1 0 0	Not used
1 0 1	Write nonposted
1 1 0	Not used
1 1 1	Not used

- MReqInfo qualifier: Four MReqInfo qualifiers describe the access during the use of the firewall comparison mechanism, as described in [Table 13-2](#).

**Table 13-2. MReqInfo Qualifier Description**

Qualifiers	Description
MReqType	0: Data access
	1: Opcode fetch
MReqSupervisor	0: User mode
	1: Supervisor mode
MReqDebug	0: Functional access
	1: Debug access

- Register that configures the combination of the MReqInfo, allowing access permission to the TM based on the MReqInfo in-band qualifier values.
- SError: Target that indicates an error condition to the initiator.
- SResp qualifier: Response from the target to the initiator concerning the transaction, as described in [Table 13-3](#).

**Table 13-3. SResp Qualifier Description**

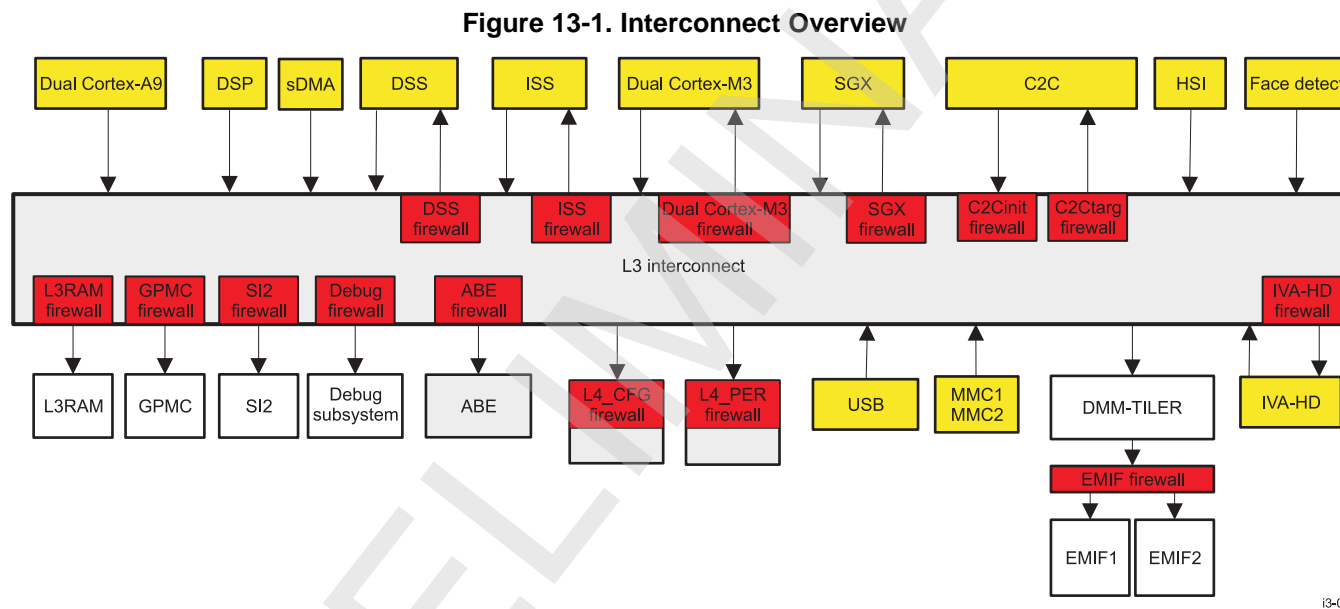
SResp[1:0]	Description
0 0	No response
0 1	Data valid/accept
1 0	Not used
1 1	Error

### 13.1.2 Architecture Overview

The device memory hierarchy includes four levels:

- Level 1 (L1) is internal to the central processing units (CPUs). It concerns data exchange with the internal Level1 cache memory subsystem, and it is the closest memory to the microprocessor unit (MPU) core and the IVAHD core.
- Level 2 (L2) is included in the IVAHD subsystem and the MPU subsystem.
- The chip-level interconnect consists of one level 3 (L3) interconnect and four level 4 (L4) interconnects. It enables communication among the modules and subsystems in the device.

Figure 13-1 shows an overview of the L3 and L4 interconnect architecture.



- L3 handles many types of data transfers, especially exchanges with system-on-chip (SoC)/external memories. L3 transfers data with a maximum width of 64 bits from the initiator to the target. The L3 interconnect is a little-endian platform.
- The L4 is composed of the following:
  - L4\_CFG: Includes the majority of the configuration interface for L3 system modules and peripheral interconnect; L4\_PER: Includes the main peripherals that require system direct memory access (sDMA) access.
  - L4\_WKUP: Includes peripherals attached to the WKUP power domain.

Modules are connected to the interconnect through an IA for the initiator module and a TA for target modules. Each module/subsystem connection is statically configured to tune the access, depending on the characteristics of the module.

To unauthorize a module or L4 interconnect access, some TAs include configurable firewalls (FWs). A firewall restricts or filters the accesses allowed to an initiator according to different access criteria. The firewalls can usually be configured by software.

The L3 and L4 interconnect default settings are fully functional; they enable all possible functional data paths and a minimal default protection setting. However, the interconnect parameters can be modified to fit user expectations.

### 13.1.3 Module Distribution

IAs and TAs provide the interface to connect the different modules and the interconnect.

Table 13-4 through Table 13-11 list the device modules, subsystems, and associated agents. The agents are listed for each interconnect domain:

- L3 initiator (master network interface units [NIUs] – MN) and target (slave NIUs – SN) agents
- L4\_PER initiator and target agents
- L4\_CFG initiator and target agents
- L4\_WKUP initiator and target agents

#### 13.1.3.1 L3 Interconnect Module Distribution

Table 13-4 and Table 13-5 list the IAs and TAs, respectively, of the L3 interconnect.

**Table 13-4. Master NIUs**

Master NIU Name	Description
L3_MN_CORTEXA9	Cortex™-A9 microprocessor unit subsystem
L3_MN_DSP	DSP subsystem
L3_MN_DSS	Display subsystem
L3_MN_MMC1/2	Multimedia controller
L3_MN_IVAHD	Video accelerator subsystem
L3_MN_SGX	Graphic accelerator
L3_MN_ISS	Imaging accelerator subsystem
L3_MN_CORTEXM3	Cortex™-M3 microprocessor subsystem
L3_MN_FACEDETECT	Face detection module
L3_MN_SDMA_rd	System DMA
L3_MN_SDMA_wr	System DMA
L3_MN_HSI	High-speed synchronous serial interface controller
L3_MN_FSUSB	Full-speed USB host controller
L3_MN_HSUSBOTG	High-speed (HS) on-the-go (OTG) USB controller
L3_MN_HSUSBHOST	Multiport USB host controller
L3_MN_C2C	Chip-to-chip (C2C)
L3_MN_DAP	JTAG®/emulation access to system resources

**Table 13-5. Slave NIUs**

Slave NIU Name	Description
L3_SN_OCM_RAM	On-chip memory controller
L3_SN_GPMC	General-purpose memory controller (GPMC)
L3_SN_C2C	Chip-to-chip
L3_SN_DMM1	Dynamic memory management
L3_SN_DMM2	Dynamic memory management

**Table 13-5. Slave NIUs (continued)**

Slave NIU Name	Description
L3_SN_SGX	Graphic accelerator
L3_SN_ISS	Imaging accelerator subsystem
L3_SN_IVAHD_CFG	Video accelerator subsystem configuration
L3_SN_IVAHD_SL2	Video accelerator subsystem shared memory
L3_SN_DSP	DSP subsystem
L3_SN_CORTEXM3	Cortex-M3 MPU subsystem
L3_SN_DSS	Display subsystem
L3 interconnect	L3 internal registers
L3_SN_ABE	Audio back end
L3_SN_L4_CFG	L4 interconnect configuration
L3_SN_L4_PER0/1/2/3	L4 interconnect peripherals
L3_SN_L4_EMU	L4 interconnect EMU subsystem

### 13.1.3.2 L4 Interconnect Module Distribution

IAs and TAs provide the interface to connect the different modules to their associated interconnect.

[Table 13-6](#) through [Table 13-11](#) list all modules and subsystems with their associated agents. The agents are listed for each L4 interconnect domain.

#### 13.1.3.2.1 L4\_PER Interconnect Agents

The L4\_PER interconnect handles transfers only to peripherals in the PER power domain. [Table 13-6](#) lists the PER TAs.

**Table 13-6. L4\_PER TAs**

Module Target Name	Description
PER_TA_UART1	Universal asynchronous receiver transmitter (UART) port 1 module
PER_TA_UART2	UART port 2 module
PER_TA_UART3	UART port 3 module
PER_TA_UART4	UART port 4 module
PER_TA_GPTIMER2	General-purpose timer 2 (GPTIMER2) module
PER_TA_GPTIMER3	GPTIMER3 module
PER_TA_GPTIMER4	GPTIMER4 module
PER_TA_GPTIMER9	GPTIMER9 module
PER_TA_GPTIMER10	GPTIMER10 module
PER_TA_GPTIMER11	GPTIMER11 module
PER_TA_DSS	Display subsystem configuration module
PER_TA_GPIO2	General-purpose input/output 2 (GPIO2) module
PER_TA_GPIO3	GPIO3 module
PER_TA_GPIO4	GPIO4 module
PER_TA_GPIO5	GPIO5 module
PER_TA_GPIO6	GPIO6 module
PER_TA_I2C1	Multimaster inter-integrated circuit 1 (I2C1) module
PER_TA_I2C2	Multimaster I2C2 module
PER_TA_I2C3	Multimaster I2C3 module
PER_TA_I2C4	Multimaster I2C4 module
PER_TA_SLIMBUS2	Silmbus 2 module
PER_TA_ELM	Error location module



**Table 13-6. L4\_PER TAs (continued)**

Module Target Name	Description
PER_TA_MCBSP4	Multichannel buffered serial port 4 (McBSP4) module
PER_TA_MCSPI1	Multichannel serial peripheral interface 1 (McBSP1) module
PER_TA_MCSPI2	McSPI2 module
PER_TA_MCSPI3	McSPI3 module
PER_TA_MCSPI4	McSPI4 module
PER_TA_HSMMC1	Multimedia card (MMC) controller 1 module
PER_TA_HSMMC2	MMC controller 2 module
PER_TA_HSMMC3	MMC controller 3 module
PER_TA_HSMMC4	MMC controller 4 module
PER_TA_HSMMC5	MMC controller 5 module
PER_TA_HDQ	Single-wire serial link low-rate module

Four ports communicate between the L3 interconnect and the L4\_PER interconnect to allow the L3 initiators to access the L4\_PER targets. [Table 13-7](#) lists the L4\_PER initiator TAs.

For the list of initiators authorized to access the L4 peripheral peripherals, see [Section 13.2.3.2.2, Connectivity Matrix](#).

**Table 13-7. L4\_PER IAs**

Module Initiator Name	Description
PER_IA_0	L3 sDMA RD interconnect port
PER_IA_1	L3 sDMA WR interconnect port
PER_IA_2	L3 MPU subsystem interconnect port
PER_IA_3	L3 others interconnect port

### 13.1.3.2.2 L4\_CFG Interconnect Agents

The L4\_CFG interconnect handles only transfers to peripherals in the CORE power domain. [Table 13-8](#) lists the TAs.

**Table 13-8. L4\_CFG TAs**

Module Target Name	Description
CFG_TA_SYSCTRL_GENERAL_CORE	System control general core module
CFG_TA_SYSCTRL_PADCONF_CORE	System control padconf core module
CFG_TA_CM1	Clock manager 1 module
CFG_TA_CM2	Clock manager 1 module
CFG_TA_SDMA	sDMA module
CFG_TA_HSI	HS synchronous serial interface (SSI) module
CFG_TA_SAR_ROM	Save-and-restore ROM
CFG_TA_HSUSBOTG	HS universal serial bus (USB) OTG module
CFG_TA_FSUSBHOST	FS USB OTG module
CFG_TA_HSUSBHOST	HS USB host module
CFG_TA_HSUSBPHY	HS USB PHY module
CFG_TA_HSUSBTLL	USB TTL module
CFG_TA_DSP	Digital signal processor (DSP) subsystem
CFG_TA_SR1	SmartReflex1 module
CFG_TA_SR2	SmartReflex2 module
CFG_TA_SR3	SmartReflex3 module

**Table 13-8. L4\_CFG TAs (continued)**

Module Target Name	Description
CFG_TA_MAILBOX	Mailbox module
CFG_TA_SPINLOCK	Spinlock module
CFG_TA_FACEDETECT	Face detect module
CFG_TA_EMIFFW	EMIF firewall
CFG_TA_GPMCFW	GPMC firewall
CFG_TA_OCMRAMFW	RAM firewall
CFG_TA_SGXFW	SGX firewall
CFG_TA_ISSF	ISS firewall
CFG_TA_CORTEXM3FW	Cortex-M3 firewall
CFG_TA_DSSF	Display subsystem (DSS) firewall
CFG_TA_SL2FW	SL2 firewall
CFG_TA_IVAHDFW	Image and video accelerator high-definition (IVA-HD) firewall
CFG_TA_EMUSSFW	EMU subsystem firewall
CFG_TA_ABEFW	Audio back-end (ABE) firewall
CFG_TA_L4WKUP	L4_WKUP connection

A unique port, CFG\_IA\_0, communicates between the L3 interconnect and the L4\_CFG interconnect to allow the L3 initiators to access the L4\_CFG targets (see [Table 13-9](#)).

For the list of initiators authorized to access the L4\_CFG peripherals, see [Section 13.2.3.2.2, Connectivity Matrix](#).

**Table 13-9. L4\_CFG IAs**

Module Initiator Name	Description
CFG_IA_0	L3 interconnect port

### 13.1.3.2.3 L4\_WKUP Interconnect Agents

The L4\_WKUP interconnect handles transfers only to peripherals in the WKUP power domain. [Table 13-10](#) lists the TAs. [Table 13-11](#) lists the L4\_WKUP IAs.

**Table 13-10. L4\_WKUP TAs**

Module Target Name	Description
WKUP_TA_GPIO1	GPIO1 module
WKUP_TA_32KTIMER	32-kHz timer module
WKUP_TA_SAR_RAM	Save-and-restore RAM
WKUP_TA_KEYBOARD	Keyboard module
WKUP_TA_SYSCTRL_GENERAL_WKUP	System control general WKUP module
WKUP_TA_SYSCTRL_PADCONF_WKUP	System control padconf WKUP module
WKUP_TA_PRM	Power reset management module
WKUP_TA_SCRM	System clock and reset management module
WKUP_TA_WDTIMER1	Watchdog timer 1 (WDTIMER1) module
WKUP_TA_WDTIMER2	WDTIMER2 module
WKUP_TA_GPTIMER1	GPTIMER1 module
WKUP_TA_GPTIMER2	GPTIMER12 module

**Table 13-11. L4\_WKUP IAs**

Module Initiator Name	Description
WKUP_IA_0	L4_CFG interconnect port

### 13.1.4 Connectivity Matrix

Figure 13-2 shows the functional paths between the L3 interconnect initiator modules and the L3 and L4 TAs. The functional paths are indicated by the use of the following:

- A cell contains a + sign when a functional path exists.
- A cell is blank when a functional path does not exist.

Figure 13-2. L3 Connectivity Matrix

Master NIU (MN)	Slave NIU (SN)																			
	L3_SN_DMM1	L3_SN_DMM2	L3_SN_C2C	L3_SN_GPMC	L3_SN_OCM_RAM	L3_SN_L4_PER0	L3_SN_L4_PER1	L3_SN_L4_PER2	L3_SN_L4_PER3	L3_SN_L4_CFG	L3_SN_EMU	L3_SN_CORTEXM3	L3_SN_ABE	L3_SN_SGX	L3_SN_ISS	L3_SN_DSS	L3_SN_SL2	L3_SN_IVAHD	L3 internal regs	
L3_MN_CORTEXA9	+		+	+	+			+		+	+	+	+	+	+	+	+	+	+	+
L3_MN_CORTEXM3	+		+	+	+				+	+	+		+	+		+	+	+	+	+
L3_MN_DSP	+			+					+	+	+		+		+					
L3_MN_IVAHD	+																			
L3_MN_SGX	+																	+		
L3_MN_ISS		+																+		
L3_MN_DSS		+																		
L3_MN_SDMA_RD		+	+	+	+	+				+	+	+	+		+	+	+	+	+	+
L3_MN_SDMA_WR		+	+	+	+	+	+			+	+	+	+		+	+	+	+	+	+
L3_MN_C2C									+	+			+							
L3_MN_HSUSBOTG		+			+								+							
L3_MN_DAP		+	+	+	+				+	+	+	+	+	+	+	+	+	+	+	+
L3_MN_FSUSB		+																		
L3_MN_HSUSBHOST		+											+							
L3_MN_HSI		+											+							
L3_MN_MMC1	+																			
L3_MN_MMC2	+																			

13-001

## 13.2 L3 Interconnect

This section describes the L3 interconnect and its components. With the exception of register points, each component includes functions for the request and response networks.

### 13.2.1 L3 Interconnect Overview

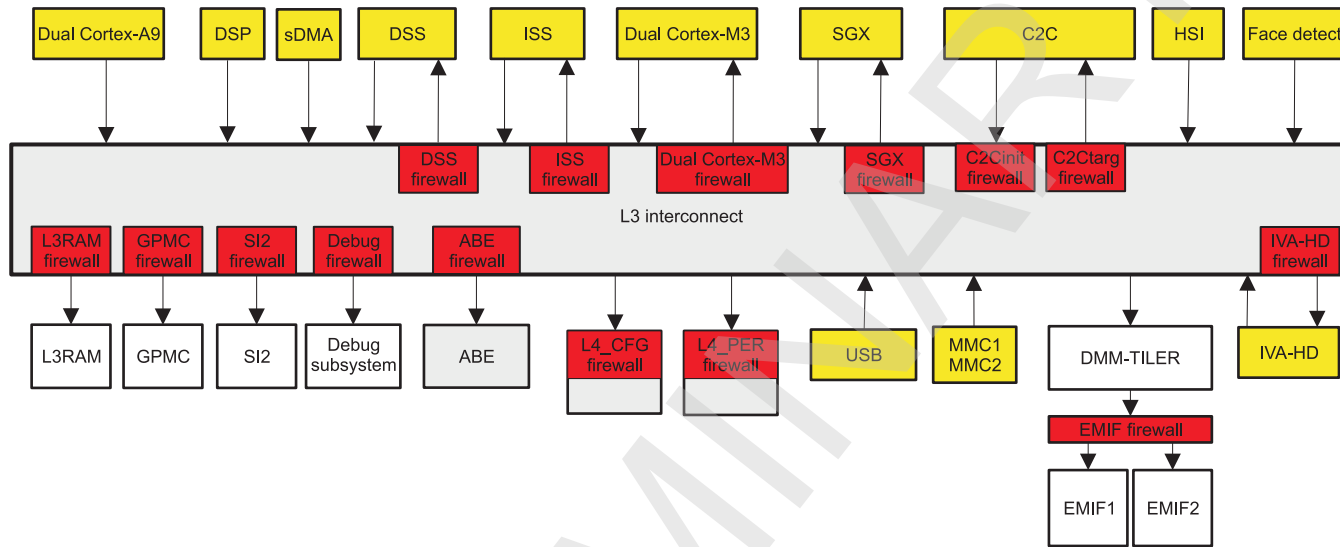
The L3 interconnect links cores in a flexible topology that couples low power with high performance. Innovative physical structures and advanced protocols ensure bandwidth and latency to individual IP cores, providing dedicated connections between IP cores and logical connections over a shared interconnect.

The main features of the L3 interconnects are:

- NIUs: Master NIUs for the IAs and slave NIUs for the TAs
- A partially depleted crossbar exchange network
- A special internal slave NIU for accessing L3 interconnect configuration registers
- True little-endian platform
- Transaction errors tracking and logging registers
- All signaling support for chip-level power-management infrastructure

[Figure 13-3](#) is an overview of the L3 interconnect and the peripherals attached to it.

Figure 13-3. L3 Interconnect Overview



i3-012

### 13.2.2 L3 Interconnect Integration

Table 13-12 through Table 13-14 summarize the integration of the module in the device.

**Table 13-12. Integration Attributes**

Module Instance	Attributes
	Power Domain
L3	PD_CORE

**Table 13-13. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
L3	L3_CLK2	L3_ICLK2	PRCM	Functional and interface clock
	L3_CLK1	L3_ICLK1	PRCM	Functional and interface clock
	L3_CLK3	INSTR_L3_ICLK	PRCM	Functional and interface clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
L3	L3_CORE_RET_RST	CORE_RET_RST	PRCM	Reset of L3_PER interconnect retention registers
	L3_CORE_RST	CORE_RST	PRCM	Reset of L3_PER interconnect

**Table 13-14. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
L3	L3_APP_IRQ	C2C_IRQ_18	C2C INTC	Interrupt to the SSCM
	L3_DBG_IRQ	MA_IRQ_9	Cortex-A9 INTC	Interrupt to the Cortex-A9
	L3_APP_IRQ	D_IRQ_84	DSP INTC	Interrupt to the DSP
	L3_APP_IRQ	MA_IRQ_10	Cortex-A9 INTC	Interrupt to the Cortex-A9

### 13.2.3 L3 Interconnect Functional Description

#### 13.2.3.1 Module Use in L3 Interconnect

The L3 interconnect network components have ConnID values for each master NIU or slave NIU. The ID uniquely identifies the master NIU or the slave NIU for an interconnect transfer. The interconnect uses ConnIDs for a number of purposes, including the following:

- Slave NIUs for error logging
- Power disconnect slave NIU for error logging
- FLAGMUX to mask interrupts: Three, one for each clock domain
- STATCOLL for configuring and monitoring: The STATCOLL components compute the traffic statistics within a user-defined window and periodically report to the user through the debug interface.
- Bandwidth regulator for configuration: The bandwidth critical masters are IVAHD, ISS, DSS, and SGX.
- Some rate adapters for configuration of throughput ratio

#### 13.2.3.2 Module Distribution

Master NIUs and slave NIUs provide the interface to connect the different modules to their associated interconnect.

Table 13-15 and Table 13-16 list all the modules and subsystems with their associated agents. The agents are listed for each L3 interconnect domain.

##### 13.2.3.2.1 L3 Interconnect Agents

Any initiator or target core is connected to the L3 interconnect through an NIU. NIUs act as entry points to the L3 interconnect, and also include various programming registers. Table 13-15 lists the supported master NIU ports.

Table 13-15. Master NIUs

Master NIU	Description
L3_MN_CORTEXA9	Cortex-A9 MPU subsystem
L3_MN_DSP	DSP subsystem
L3_MN_DSS	Display subsystem
L3_MN_MMC1/2	Multimedia controller
L3_MN_IVAHD	Video accelerator subsystem
L3_MN_SGX	Graphic accelerator
L3_MN_ISS	Imaging accelerator subsystem
L3_MN_CORTEXM3	Cortex-M3 MPU subsystem
L3_MN_FACEDETECT	Face detection module
L3_MN_SDMA_rd	sDMA
L3_MN_SDMA_wr	sDMA
L3_MN_HSI	HS synchronous serial interrupt controller (INTC)
L3_MN_FSUSB	Full-speed USB host controller
L3_MN_HSUSBOTG	HS OTG USB controller
L3_MN_HSUSBHOST	Multiport USB host controller
L3_MN_C2C	C2C
L3_MN_DAP	JTAG/emulation access to system resources

Table 13-16 lists the supported slave NIU ports.

**Table 13-16. Slave NIUs**

Slave NIU	Description
L3_SN_OCM_RAM	On-chip memory controller
L3_SN_GPMC	GPMC
L3_SN_C2C	C2C
L3_SN_DMM1	Dynamic memory management
L3_SN_DMM2	Dynamic memory management
L3_SN_SGX	Graphic accelerator
L3_SN_ISS	Imaging accelerator subsystem
L3_SN_IVAHD_CFG	Video accelerator subsystem configuration
L3_SN_IVAHD_SL2	Video accelerator subsystem shared memory
L3_SN_DSP	DSP subsystem
L3_SN_CORTEXM3	Cortex-M3 MPU subsystem
L3_SN_DSS	Display subsystem
L3 interconnect	L3 internal registers
L3_SN_ABE	Audio back end
L3_SN_L4_CFG	L4 interconnect configuration
L3_SN_L4_PER0/1/2/3	L4 interconnect peripherals
L3_SN_L4_EMU	L4 interconnect EMU subsystem

### 13.2.3.2.2 L3 Connectivity Matrix

The L3 interconnect is divided into three subclock domains:

- L3\_CLK1: Low-power domain
- L3\_CLK2: Peripherals and multimedia
- L3\_CLK3: Instrumentation (debug)

Each is implemented in a different clock domain.

#### 13.2.3.2.2.1 Clock Domain Mapping of the L3 Interconnect Modules

Each clock domain (CLK1, CLK2, and CLK3) has its own host, flag mux, slave NIUs, and bandwidth regulators.

**Table 13-17. L3 Clock Domains and Elements**

Domain	Elements
CLK1	HOST_CLK1
	DMM1/2
	ABE
	L4_CFG
	FLAGMUX
	DSS_BW
	PWR_DISK



**Table 13-17. L3 Clock Domains and Elements (continued)**

Domain	Elements
CLK2	HOST_CLK2
	GPMC
	OCM_RAM
	DSS
	ISS
	CORTEXM3
	SGX
	IVAHD
	SL2
	L4_PER0/1/2/3
	FLAGMUX
	ISS_BW
	ISS_BW
	IVAHD_BW
	SGX_BW
CLK3	C2C
	HOST_CLK3
	L4_EMU
	FLAGMUX
	STATCOLL

Figure 13-4 lists the functional paths between the L3 interconnect master NIUs and the L3 and L4 slave NIU agents. The functional paths in the figure are indicated by the following:

- A cell contains a plus sign when a functional path exists.
- A cell is blank when a functional path does not exist.

**Figure 13-4. L3 Connectivity Matrix**

Master NIU (MN)	Slave NIU (SN)																			
	L3_SN_DMM1	L3_SN_DMM2	L3_SN_C2C	L3_SN_GPMC	L3_SN_OCM_RAM	L3_SN_L4_PER0	L3_SN_L4_PER1	L3_SN_L4_PER2	L3_SN_L4_PER3	L3_SN_L4_CFG	L3_SN_EMU	L3_SN_CORTEXM3	L3_SN_ABE	L3_SN_SGX	L3_SN_ISS	L3_SN_DSS	L3_SN_SL2	L3_SN_IVAHD	L3 internal regs	
L3_MN_CORTEXA9	+		+	+	+			+		+	+	+	+	+	+	+	+	+	+	+
L3_MN_CORTEXM3	+		+	+	+				+	+	+		+	+		+	+	+	+	+
L3_MN_DSP	+			+					+	+	+		+		+					
L3_MN_IVAHD	+																			
L3_MN_SGX	+																		+	
L3_MN_ISS		+																	+	
L3_MN_DSS		+																		
L3_MN_SDMA_RD		+	+	+	+	+				+	+	+	+		+	+	+	+	+	+
L3_MN_SDMA_WR		+	+	+	+	+				+	+	+	+		+	+	+	+	+	+
L3_MN_C2C									+	+			+							
L3_MN_HSUSBOTG		+			+								+							
L3_MN_DAP		+	+	+	+				+	+	+	+	+	+	+	+	+	+	+	+
L3_MN_FSUSB		+																		
L3_MN_HSUSBHOST		+											+							
L3_MN_HSI		+											+							
L3_MN_MMC1	+																			
L3_MN_MMC2	+																			

I3-001

### 13.2.3.2.3 Master NIU Identification

A master NIU ID (ConnID) is assigned to every module in the device. The ID uniquely identifies the master NIU for an interconnect transfer (see Table 13-18). The interconnect uses ConnID values for a number of purposes, including:

- Master source identification for the protection mechanism
- Response route generation
- Firewall error logging
- L3 interconnect error logging

Table 13-18. ConnID Values

ConnID	Master NIU
0x0	Cortex-A9
0x1	DAP
0x2	DSP
0x3	IVAHD
0x4	ISS
0x4	Cortex-M3
0x4	FaceDetect
0x5	sDMA_rd
0x5	sDMA_wr
0x6	SGX
0x7	DSS
0x8	C2C
0x9	HSI
0xA	MMC1
0xA	MMC2
0xC	HSUSBHOST
0xC	FSUSB
0xC	HSUSBOTG
0xD	Reserved
0xE	Reserved
0xF	Reserved

### 13.2.3.3 Bandwidth Regulators

The bandwidth regulators prevent master NIUs from consuming too much bandwidth of a link, or a slave NIU that is shared between several data flows: packets are then transported at a slower rate. The value of a bandwidth can be programmed in the bandwidth regulator. When the bandwidth is below the programmed value, the pressure bit is set to 1, giving priority to this master. When the bandwidth is above the programmed value, the pressure bit is set to 0 and the concerned master has the same weight as others.

The bandwidth registers regulate the packet flow by applying flow control on the RX port, thus ensuring that the traffic does not exceed the allocated bandwidth. The next packet is sent only when an internal timer expires. The registers in this group are:

- **L3\_BW\_R\_WATERMARK**: Gives the amount of data allowed to exceed the average bandwidth during a short time period
- **L3\_BW\_R\_PRESS**: Describes the pressure applied to outgoing packets
- **L3\_BW\_R\_CLEARHISTORY**: Resets the traffic counter when set to 1. It is used after an update in the **L3\_BW\_R\_BANDWIDTH** and **L3\_BW\_R\_WATERMARK** registers (see [Section 13.2.5.1.6, L3 BW Regulator Register Summary and Description](#)).

### 13.2.3.4 Flag Mux Generators

The flag mux generator collects information such as errors and interrupts from slave NIUs and the interconnect firewall. The result signals are then sent to the MPU INTC without interfering with the interconnect traffic. The **L3\_FLAGMUX\_MASK** registers can be used to prevent the flag mux from seeing certain events.

The unit has a standard **COREREG** register for identification of the attached core type. The **L3\_FLAGMUX\_STDHOSTHDR\_VERSIONREG** register identifies the characteristics of the attached core.

The unit-specific registers (MASK bit 0 or bit 1 of the flag inputs, and L3\_FLAGMUX\_REGERR bit 0 or bit 1) are used to read the input errors. Each register is dedicated to reporting the bit corresponding to the register number; for example, [L3\\_FLAGMUX\\_REGERR0](#) reports on bit 0, and [L3\\_FLAGMUX\\_REGERR1](#) reports on bit 1. Any given L3\_FLAGMUX\_REGERR register reports the same bit for all flag source inputs (see [Table 13-162](#)).

### 13.2.3.5 Rate Adapter Generators

The rate adapter generators give the option to configure the L3 interconnect to work in auto-adaptive mode or pipeline mode for sending the packets by using the CNF register. The rate adapter can be configured in store-and-forward mode when the throughput ratio is not well defined, or in auto-adaptive mode when the throughput ratio is predictable. Because the ratios are predictable in the L3 interconnect, the rate adapters are always used in auto-adaptive mode, except for the rate adapter on the L4\_CFG and L4\_ABE responses path. L4\_CFG and L4\_ABE are configured in store-and-forward mode. The main rate adapt registers are:

- CLK1\_RATE\_ADAPT\_RESP\_32TO128\_CLK1: Removes WAIT cycles on responses coming from ABE, L4CFG, and power disconnect to host
- CLK2\_RATE\_ADAPT\_RESP\_32TO128\_CLK2: Removes WAIT cycles on responses coming from CLK2 32-bit slaves (see [Table 13-192](#))

### 13.2.3.6 Statistic Collectors Group

Statistic collectors are internal masters that share the same master address as the master NIUs. These components compute the traffic statistics within a defined window and periodically report through the DEBUG interface. The key features of the statistic collector are:

- Nonintrusive monitoring
- Programmable filters and counters
- Collects results at a programmable time interval

Event detectors are programmed through the [L3\\_STCOL\\_REQEVT](#) and [L3\\_STCOL\\_RSPEVT](#) configuration registers for request and response ports, respectively. The following events can be identified:

- Word transfer
- WAIT cycles
- Flow control
- Payload transfers
- Latency measurements

Performance monitoring is enabled through the [L3\\_STCOL\\_EN](#) register. The [L3\\_STCOL\\_SOFTEN](#) register enables software to monitor the performance. Event muxes are programmed through the [L3\\_STCOL\\_EVTMUX\\_SEL0](#) configuration register, which determines which port will be monitored by a filter configured by the filter registers (see [Table 13-240](#)).

Filters are programmed through the [Table 13-240](#) configuration register, along with additional selection criteria programmed through the mask/match registers (see [Table 13-201](#)). A filter can be configured to accept or reject:

- Read operations
- Write operations
- Errors
- Addresses

Filter operation is programmed through the L3\_STCOL\_OP registers (see [Table 13-201](#)).

### 13.2.3.7 L3 Protection and Firewalls

Device protection relies on L3 firewalls and their configuration.

### 13.2.3.7.1 L3 Firewall Reset

The values of L3 firewall registers on reset are tied in hardware or exported from the control module registers.

Values exported from the control module are intended to give defined rights to the firewalls at reset and thus ensure the content after going out of reset. Because the control module consists of retention flip-flops (RFFs), which immunize it against register content loss during retention power state, the registers of the control module storing the exported values are not subject to a soft reset.

The L3 firewall registers are also retention-capable and are immune against content loss during CORE RETENTION power state. The control module registers are reset by a cold reset only, whereas the L3 firewall registers are reset by clearing the [REGUPDATE\\_CONTROL\[1\] FW\\_LOAD\\_REQ](#) bit. When the [REGUPDATE\\_CONTROL\[1\] FW\\_LOAD\\_REQ](#) bit comes back automatically to 1, the exported values are loaded.

**CAUTION**

Before reprogramming the firewall registers and/or before using the FW\_LOAD\_REQ mechanism, the request must be asserted by configuring the [REGUPDATE\\_CONTROL\[0\] BUSY\\_REQ](#) bit.

To load the exported values at run time:

1. Set the [REGUPDATE\\_CONTROL\[0\] BUSY\\_REQ](#) bit to 0x1 to ensure that no transaction can reach the slave NIU (suspend).
2. Clear the [REGUPDATE\\_CONTROL\[1\] FW\\_LOAD\\_REQ](#) bit by writing 0x1 to it.
3. Wait until the [REGUPDATE\\_CONTROL\[1\] FW\\_LOAD\\_REQ](#) bit is reset to 0x1 by hardware.
4. Set the [REGUPDATE\\_CONTROL\[0\] BUSY\\_REQ](#) bit to 0x0 to allow transactions to reach the slave NIU (resume).

To reprogram the firewall registers at run time:

1. Set the [REGUPDATE\\_CONTROL\[0\] BUSY\\_REQ](#) bit to 0x1 to ensure that no transaction can reach the slave NIU (suspend).
2. Upload the new values to all firewall registers.
3. Set the [REGUPDATE\\_CONTROL\[0\] BUSY\\_REQ](#) bit to 0x0 to allow transactions to reach the slave NIU (resume).

---

**NOTE:** Hardware reset values can be modified by exported values from the control module at reset.

---

#### 13.2.3.7.1.1 L3 Firewall – Exported Reset Values

[Table 13-19](#) and [Table 13-20](#) list the exported reset values and mapping, respectively.

**Table 13-19. L3 Firewall Exported Reset Values**

<a href="#">MRM_PERMISSION_REGION_LOW_j</a> [15:12]	<a href="#">MRM_PERMISSION_REGION_LOW_j</a> [11:0]
0x0	0xFFF
0xF	0xFFF
0x0	0x038
0x2	0x038

**Table 13-20. L3 Firewall Exported Values Mapping**

<a href="#">CONTROL_CORE_L3_HW_FW_EXPORTED_VALUES_CONF</a> Bits	Slave NIU Firewall
[0] and [16]	GPMC
[1] and [17]	C2CINIT

**Table 13-20. L3 Firewall Exported Values Mapping (continued)**

CONTROL_CORE_L3_HW_FW_EXPORTED_VALUES_CONF Bits	Slave NIU Firewall
[2] and [18]	C2CTARG
[3] and [19]	L3 RAM
[4] and [20]	DSS
[5] and [21]	ISS and dual Cortex-M3 subsystem
[6] and [22]	SGX
[7] and [23]	SL2
[8] and [24]	IVA-HD
[11] and [27]	EMIF
[12] and [28]	DEBUG
[13] and [29]	L4 AUDIO

For more information, see [Chapter 18, Control Module](#).

### 13.2.3.7.2 Power Management

As part of the system-wide power-management scheme, the L3 interconnect goes into IDLE state after receiving a request from the power, reset, and clock management (PRCM) module after all commands are serviced. This function is handled by hardware.

To reduce power consumption, the L3 interconnect automatically performs internal clock autogating. This is managed by hardware; no software configurations or settings are required.

Retention is performed on all software-accessible registers:

- Statistic collectors
- Internal rate adapters
- Bandwidth regulators
- TAs
- Firewalls

This process prevents reconfiguration after a clock domain switches off.

### 13.2.3.7.3 L3 Firewall Functionality

The access to the slave NIUs is granted only to master NIUs according to in-band attributes sent in each transaction crossing the L3 interconnect, such as:

- MCMD: Specifies the type of access (read or write) required by the master NIU
- ConnID: Used to determine the permission of the master NIU
- MReqInfo: Transaction attribute adding information about the access type

[Table 13-21](#) lists the MReqInfo Values.

**Table 13-21. MReqInfo Values**

Qualifier	Access Definition	Access Description
MReqType	0: Data 1: Opcode	Indicates whether the request is for instruction fetch or data access
MReqDebug	0: Functional 1: Debug	When set, indicates that the request has been issued by a master NIU in DEBUG state
MReqSupervisor	0: User 1: Privilege	When set, indicates that the request is qualified with the supervisor attribute. It can be provided by a processor running in supervisor mode or by a module that inherited this attribute from the processor (DMA channel with a supervisor attribute).

The firewall comparison mechanism enables access to a protected slave NIU only when a correct combination of three MReqInfo in-band parameters is transmitted.

MReqInfo is a combination of a fixed 3-bit pattern that corresponds to a combination of the parameters MReqDebug, MReqType, and MReqSupervisor. See [Table 13-22](#).

**Table 13-22. L3 ReqInfo Mapping**

ReqInfo Name	MReqDebug	MReqType	MReqSupervisor	
Master NIUs	L3_MN_CORTEXA9	x	x	x
	L3_MN_CORTEXM3	x	x	x
	L3_MN_DSP	x	x	
	L3_MN_IVAHD			
	L3_MN_SGX			
	L3_MN_DSS			
	L3_MN_ISS			
	L3_MN_C2C	x	x	x
	L3_MN_DAP	x		x
	L3_MN_SDMA_RD		x	x
	L3_MN_SDMA_WR		x	x
	L3_MN_HSUSBOTG			
	L3_MN_HSUSBHOST			
	L3_MN_FSUSB			
	L3_MN_HSI			x
	L3_MN_FACEDETECT			
	L3_MN_MMC1/2			
	Slave NIUs	L3_SN_DMM1	x	x
L3_SN_DMM2		x	x	x
L3_SN_GPMC				
L3_SN_OCM_RAM				
L3_SN_L4_PER0/1/2/3		x		x
L3_SN_L4_CFG		x		x
L3_SN_L4_ABE		x		
L3_SN_CORTEXM3				
L3_SN_SGX				
L3_SN_DAP		x		
L3_SN_ISS				
L3_SN_DSS				x
L3_SN_IVAHD		x		
L3_SN_SL2				
L3_SN_C2C		x	x	x

**13.2.3.7.3.1 Protection Regions**

Each slave NIU address space is subdivided into protection regions (maximum of 10). The regions are configurable with a size of 4-KB granularity. The firewalls can also be multiport while using the description of the same regions for dual access memories or to support interleaving mechanisms on several memories.

[Table 13-23](#) lists the number of protected regions and ports for each slave NIU.

**Table 13-23. Slave NIU Firewall and Region Configuration**

Slave NIU	Firewall	Number of Regions	Number of Ports
L3 RAM	L3 RAM	10	1
GPMC	PMC	8	1
SL2	SL2	1	1
SGX	SGX	1	1
C2C	C2CTARG	1	1
L3 interconnect	C2CINIT	4	2
ISS	ISS	1	1
Dual Cortex-M3 subsystem	Dual Cortex-M3	1	1
DSS	DSS	1	1
IVA-HD	IVA-HD	1	1
DEBUG subsystem	DEBUG	2	1
ABE	ABE	1	1
EMIF	EMIF	8	3

Two types of regions are distinguished in a slave NIU firewall:

- Default region: Available in all slave NIUs. It covers the entire slave NIU address range and must be reset or overlaid by other firewall configured regions, because it always has the lowest priority.
- Normal region: The number of normal regions varies in a slave NIU; they have identical capabilities (see [Table 13-23](#)).

Each region has the following characteristics:

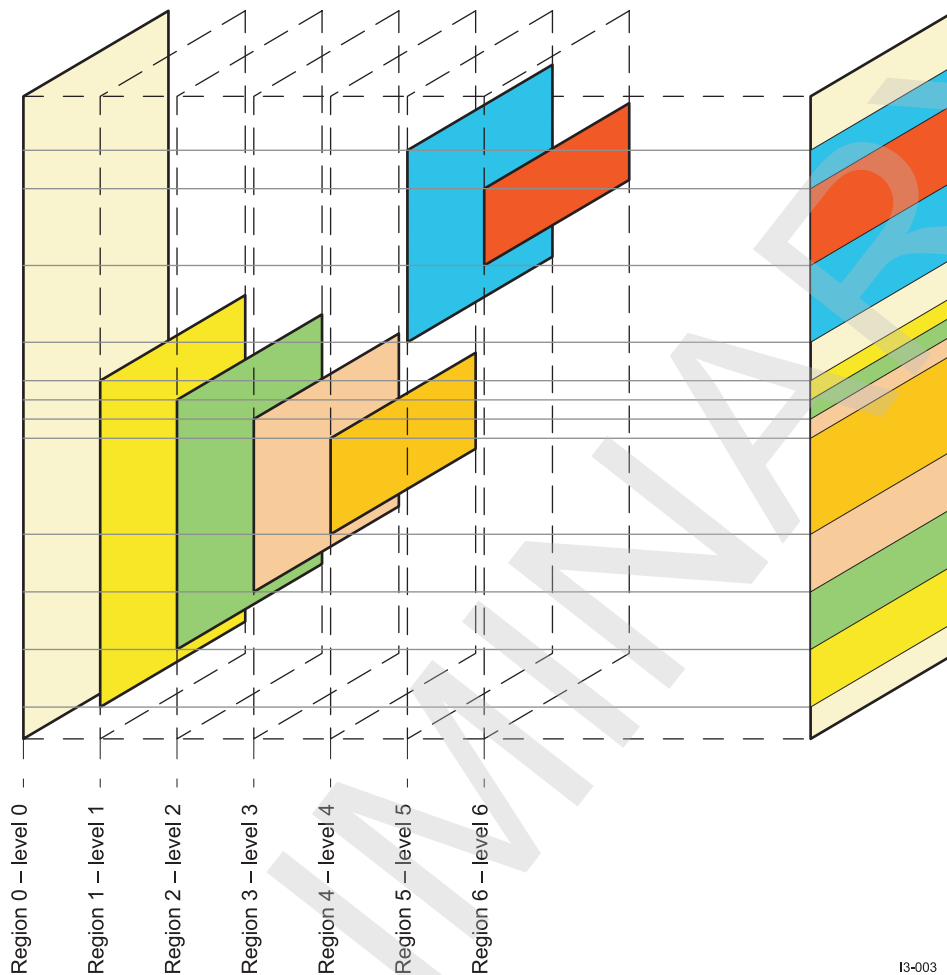
- Start address: Physical slave NIU start address
- End address: Physical slave NIU end address
- Specific access rights (see [Section 13.2.3.7.3.3](#), *Protection Mechanism per Region Examples*)
- Priority level from 0 (lowest) to 10 (highest)

Depending on its priority level, a region can override the settings of another region; the access rights of the region with the highest priority apply. All regions have a fixed (not configurable) priority level that corresponds to their number: Region 1 has priority level 1, region 2 has priority level 2, and so on.

[Figure 13-5](#) shows the priority level with associated regions. This priority level scheme allows multiplying the flexibility and capability of the firewall. [Figure 13-5](#) shows a 7-region firewall setting that creates 13 regions (twice the number of regions created than originally available).



Figure 13-5. L3 Interconnect Region Overlay and Priority Level Overview



13-003

The address range covered by the regions is defined in the `START_REGION_i` and `END_REGION_i` registers. The boundary checks are done on a minimum size of 4-KB pages; thus, bits [11:0] of those 32-bit registers are not checked.

The address space size of the slave NIUs ([bits [31:12]]) depends on the size of the slave NIU to protect (that is, if a memory is only 48KB, then the size is defined through bits [16:12] of the slave NIU start/end address registers of the firewall region (`START_REGION_i[16:12]` and `END_REGION_i[16:12]`)).

**NOTE:** The EMIF firewall sees only the 4GB that require protection.

Bit [31] is replaced by bit [32] of the EMIF address space seen by firewalls to have 2GB of LPDDR2-SDRAM and 2GB of LPDDR2-NVM contiguous.

For more information, see [Section 15.3, EMIF Controller](#).

On multiport firewalls (EMIF), the checking of each REGION {0, n} can be activated on one or several ports at the same time. However, if interleaving is not desired on some parts of the protected memory space, it is possible to apply or not apply region checking on selected ports using the region enable/disable on port instance capability of the `END_REGION_i` register.

Most slave NIUs support only one input port (port 0) except:

- The EMIF module protected by the EMIF firewall has three ports:
  - Port 0 for Cortex-A9 accesses
  - Port 1 for the first L3 interconnect accesses

- Port 2 for the second L3 interconnect accesses

A region can be applied or not to each port independently. To enable and disable the regions:

- For port 0: Set/clear the [END\\_REGION\\_i\[0\]](#) REGION\_ENABLE\_PORT0 bit (for all L3 firewalls).
- For port 1: Set/clear the [END\\_REGION\\_i\[1\]](#) REGION\_ENABLE\_PORT1 bit (only for Crypto and EMIF firewalls).
- For port 2: Set/clear the [END\\_REGION\\_i\[2\]](#) REGION\_ENABLE\_PORT2 bit (only for EMIF firewalls).

### 13.2.3.7.3.2 L3 Firewall Registers Overview

[Table 13-24](#) and [Table 13-25](#) list the L3 firewall permission-setting registers.

**Table 13-24. L3 Firewall Read/Write Permission-Setting Register**

Register Name	Bits	Field Name	Description (see <a href="#">Table 13-18</a> )	Field Modifiability
<a href="#">MRM_PERMISSION_REGION_HIGH_j</a>	31	RESERVED	Reserved	–
	30	RESERVED	Reserved	–
	29	RESERVED	Reserved	–
	28	RESERVED	Reserved	–
	27	RESERVED	Reserved	–
	26	RESERVED	Reserved	–
	25	W12	ConnID = 12 write permission	RW
	24	R12	ConnID = 12 read permission	RW
	23	RESERVED	Reserved	–
	22	RESERVED	Reserved	–
	21	W10	ConnID = 10 write permission	RW
	20	R10	ConnID = 10 read permission	RW
	19	W9	ConnID = 9 write permission	RW
	18	R9	ConnID = 9 read permission	RW
	17	W8	ConnID = 8 write permission	RW
	16	R8	ConnID = 8 read permission	RW
	15	W7	ConnID = 7 write permission	RW
	14	R7	ConnID = 7 read permission	RW
	13	W6	ConnID = 6 write permission	RW
	12	R6	ConnID = 6 read permission	RW
	11	W5	ConnID = 5 write permission	RW
	10	R5	ConnID = 5 read permission	RW
	9	W4	ConnID = 4 write permission	RW
	8	R4	ConnID = 4 read permission	RW
	7	W3	ConnID = 3 write permission	RW
	6	R3	ConnID = 3 read permission	RW
	5	W2	ConnID = 2 write permission	RW
	4	R2	ConnID = 2 read permission	RW
	3	W1	ConnID = 1 write permission	RW
	2	R1	ConnID = 1 read permission	RW
	1	W0	ConnID = 0 write permission	RW
	0	R0	ConnID = 0 read permission	RW

**Table 13-25. L3 Firewall Permission-Setting Register**

Register name	Type of Permission	Bits	Field Name	Description	Field Modifiability
MRM_PERMISSION_REGION_LOW_j	DEBUG	15	PUB_PRV_DEBUG	Privilege debug allowed	RW
		14	PUB_USR_DEBUG	User debug allowed	RW
	Access	13:12	Reserved	Reserved	RW
		11	PUB_PRV_WRITE	Privilege write allowed	RW
		10	PUB_PRV_READ	Privilege read allowed	RW
		9	PUB_PRV_EXE	Privilege exe allowed	RW
		8	PUB_USR_WRITE	User write access allowed	RW
		7	PUB_USR_READ	User read access allowed	RW
6	PUB_USR_EXE	User exe access allowed	RW		

### 13.2.3.7.3.3 Protection Mechanism per Region Examples

The access permission of each region is configurable and defined through the [MRM\\_PERMISSION\\_REGION\\_HIGH\\_j](#) and [MRM\\_PERMISSION\\_REGION\\_LOW\\_j](#) registers (see [Section 13.2.3.7.3.2, L3 Firewall Registers Overview](#)).

#### Master NIU permissions:

- To give read access to the master NIU with ConnID = n, set the [MRM\\_PERMISSION\\_REGION\\_HIGH\\_j\[n \\* 2\]](#) Rn bit.
- To give write access to the master NIU with ConnID = n, set the [MRM\\_PERMISSION\\_REGION\\_HIGH\\_j\[n \\* 2 + 1\]](#) Wn bit.

#### Debug permissions:

- To give privilege debug access, set the [MRM\\_PERMISSION\\_REGION\\_LOW\\_j\[15\]](#) PUB\_PRV\_DEBUG bit.
- To give user debug access, set the [MRM\\_PERMISSION\\_REGION\\_LOW\\_j\[14\]](#) PUB\_USR\_DEBUG bit.

#### User, read/write/executable permissions:

- To give privilege write access, set the [MRM\\_PERMISSION\\_REGION\\_LOW\\_j\[11\]](#) PUB\_PRV\_WRITE bit.
- To give privilege read access, set the [MRM\\_PERMISSION\\_REGION\\_LOW\\_j\[10\]](#) PUB\_PRV\_READ bit.
- To give privilege exe access, set the [MRM\\_PERMISSION\\_REGION\\_LOW\\_j\[9\]](#) PUB\_PRV\_EXE bit.
- To give user write access, set the [MRM\\_PERMISSION\\_REGION\\_LOW\\_j\[8\]](#) PUB\_USR\_WRITE bit.
- To give user read access, set the [MRM\\_PERMISSION\\_REGION\\_LOW\\_j\[7\]](#) PUB\_USR\_READ bit.
- To give user exe access, set the [MRM\\_PERMISSION\\_REGION\\_LOW\\_j\[6\]](#) PUB\_USR\_EXE bit.

Example: To provide debug write privilege access to the master NIU ConnID = 7, the following bits must be set:

- [MRM\\_PERMISSION\\_REGION\\_HIGH\\_j\[15\]](#) W7
- [MRM\\_PERMISSION\\_REGION\\_LOW\\_j\[15\]](#) PUB\_PRV\_DEBUG

### 13.2.3.7.3.4 L3 Firewall Error Logging

If a protection violation error is detected, the following signals are generated:

- An in-band error is generated to the master NIU of the access.
- An interrupt is generated to the Cortex-A9 and DSP INTCs.

The L3 interconnect does not differentiate errors generated by firewalls from all other supported types of errors.

An in-band error is generated by modules each time an access is not allowed. When an in-band error is sent back into the transaction it is seen as an external prefetch or data abort by the initiator, depending on whether the transaction was an instruction fetch or a data access.

Information about in-band errors is logged into two registers:

- [ERROR\\_LOG\\_k](#): Logs the information about the start/end address of the hit region and the qualifiers of the transaction
- [LOGICAL\\_ADDR\\_ERRLOG\\_k\[31:12\]](#): Logs the address of the failed access

**NOTE:** When a multiport firewall is implemented, these registers are duplicated for each port.

Table 13-26 lists the L3 firewall error-logging registers.

**Table 13-26. L3 Firewall Error-Logging Registers**

Register Name	Register Field Name	Field Modifiability	Parameter Comments
<a href="#">ERROR_LOG_k</a>	RESERVED[31:24]	Read only	Reads return 0s.
	BLK_BURST_VIOLATION[23]	Read/write	Read 0x1: 2D burst not allowed or exceeds allowed size. Write to clear the <a href="#">ERROR_LOG_k</a> and <a href="#">LOGICAL_ADDR_ERRLOG_k</a> registers.
	RESERVED[22]	Read only	Reads return 0s.
	REGION_START_ERRLOG[21:17]	Read/write	Read: Wrong access hit this region number. Write to clear the <a href="#">ERROR_LOG_k</a> and <a href="#">LOGICAL_ADDR_ERRLOG_k</a> registers.
	REGION_END_ERRLOG[16:12]	Read/write	Read: Wrong access hit this region number. Write to clear the <a href="#">ERROR_LOG_k</a> and <a href="#">LOGICAL_ADDR_ERRLOG_k</a> registers.
	REQINFO_ERRLOG[11:0]	Read/write	Mapping of the error according to the reqinfo vector: [11:8] ConnID [3:0] [7] MCMD [0] [3] MReqDebug [1] MReqSupervisor [0] MReqType

L3 firewall errors can be cleared by writing to the [ERROR\\_LOG\\_k](#) register in the firewall that recorded the error. Clearing the [ERROR\\_LOG\\_k](#) register deasserts the corresponding error if it exists.

The L3 firewall register [ERROR\\_LOG\\_k](#) must be cleared before clearing the [CONTROL\\_CORE\\_SEC\\_ERR\\_STATUS\\_FUNC](#) and [CONTROL\\_CORE\\_SEC\\_ERR\\_STATUS\\_DEBUG](#) registers in the control module.

When a protection violation occurs, an interrupt is sent to the MPU and IVAHD INTC (if enabled). An in-band error is sent back, and an error is logged in the [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#) and [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#) registers, depending on the functional mode:

- In application mode:
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC\[01\]](#) – L3 RAM protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC\[02\]](#) – GPMC protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC\[03\]](#) – EMIF protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC\[04\]](#) – IVAHD protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC\[05\]](#) – Cortex-M3 protection violation

- CONTROL\_SEC\_ERR\_STATUS\_FUNC[06] – SL2 protection violation
- CONTROL\_SEC\_ERR\_STATUS\_FUNC[12] – C2C slave NIU protection violation
- CONTROL\_SEC\_ERR\_STATUS\_FUNC[13] – SGX protection violation
- CONTROL\_SEC\_ERR\_STATUS\_FUNC[14] – DSS protection violation
- CONTROL\_SEC\_ERR\_STATUS\_FUNC[15] – ISS protection violation
- CONTROL\_SEC\_ERR\_STATUS\_FUNC[18] – DEBUG protection violation
- CONTROL\_SEC\_ERR\_STATUS\_FUNC[19] – ABE protection violation
- CONTROL\_SEC\_ERR\_STATUS\_FUNC[20] – C2C init protection violation
- In debug mode:
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[01] – L3 RAM protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[02] – GPMC protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[03] – EMIF protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[04] – IVAHD protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[05] – Cortex-M3 protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[06] – SL2 protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[12] – C2C slave NIU protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[13] – SGX protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[14] – DSS protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[15] – ISS protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[18] – DEBUG protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[19] – ABE protection violation
  - CONTROL\_SEC\_ERR\_STATUS\_DEBUG[20] – C2C init protection violation

For more information, see [Chapter 18, Control Module](#).

### 13.2.3.7.3.5 L3 Firewall Default Configuration

[Table 13-27](#) summarizes the configuration of the L3 firewalls.

**Table 13-27. L3 Firewalls Default Configurations**

Device/Region: 0						
Permission Type	Reset Value	Reset Value	Reset Type	Run Time	Firewall Register (where j = 0)	Control Module Register
ACCESS_PERMISSION	All	0xFFFF	Exported	Configurable	MRM_PERMISSION_REGION_LOW_j[11:0]	CONTROL_CORE_L3_HW_FW_EXPORTED_VALUES_CONF[k]
DEBUG_PERMISSION	All	0xF	Exported	Configurable	MRM_PERMISSION_REGION_LOW_j[15:12]	CONTROL_CORE_L3_HW_FW_EXPORTED_VALUES_CONF[l] CONTROL_CORE_L3_HW_FW_EXPORTED_VALUES_CONF[k]
INITIATOR_PERMISSION	All	0xFFFF FFFFF	Tied	Configurable	MRM_PERMISSION_REGION_HIGH_j[31:0]	N/A

**NOTE:** For the values of k and l, see [Table 13-28](#).

**Table 13-28. Control Module Register – Factorization**

Variable Value		Module Name	Regions
k	l		
3	19	L3 RAM firewall	Regions 1 to 9
0	16	GPMC firewall	Regions 1 to 7
7	23	SL2 firewall	Regions 1 to 7
6	22	SGX firewall	Regions 1 to 7
5	21	ISS firewall	Regions 1 to 7
5	21	Cortex-M3 firewall	Regions 1 to 7

**Table 13-28. Control Module Register – Factorization (continued)**

Variable Value		Module Name	Regions
4	20	DSS firewall	Regions 1 to 7
8	24	IVAHD firewall	Regions 1 to 7
12	28	DEBUG firewall	Region 1
13	29	ABE firewall	Region 1
1	17	C2C slave NIU firewall	Region 1
9	25	C2C init firewall	Regions 1 to 3
11	27	EMIF firewall	Regions 1 to 7

### 13.2.3.8 L3 Interconnect Error Handling

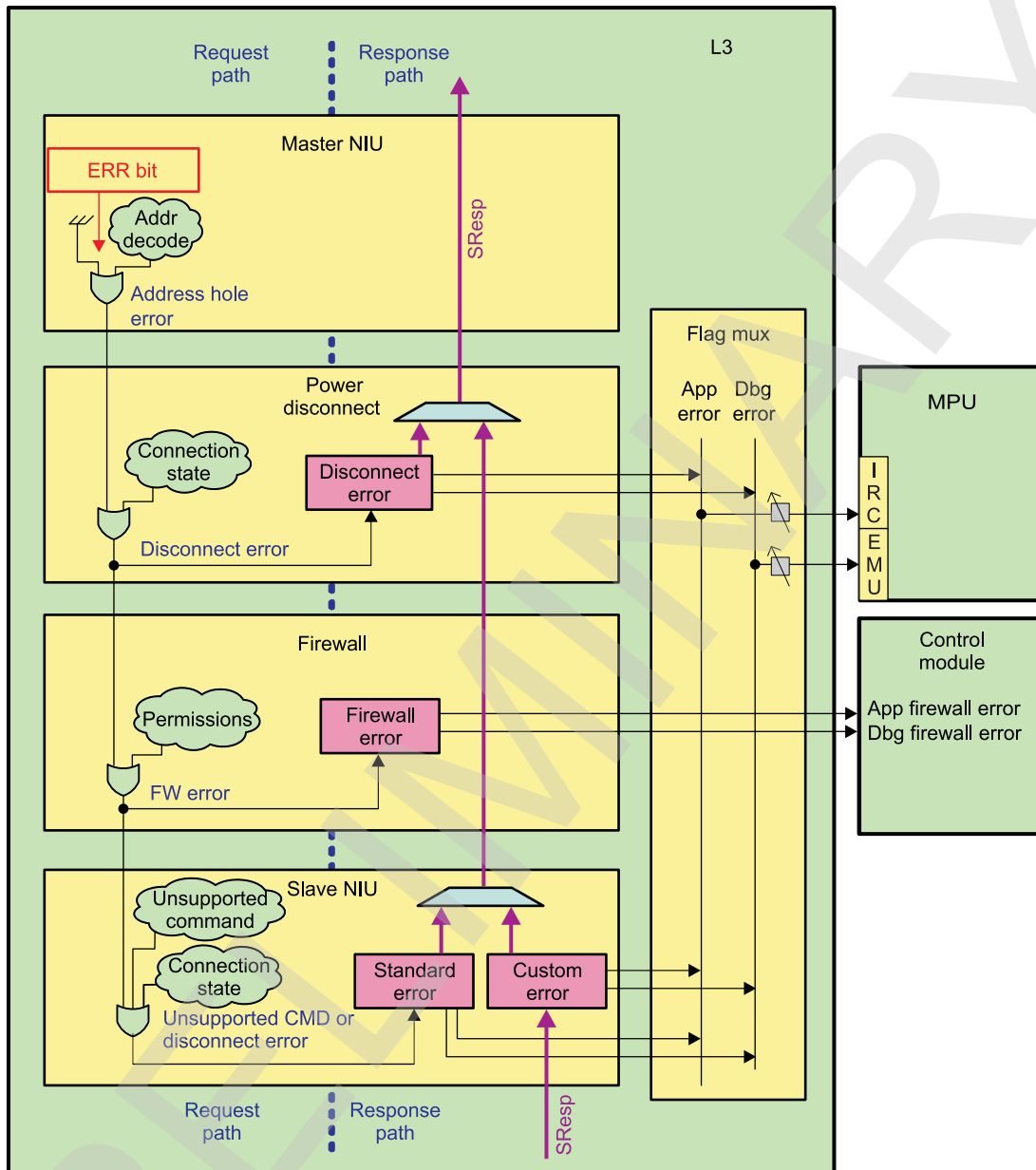
Error logging is enabled in the L3 interconnect. The three major types of errors are:

- Slave NIU errors
- Firewall errors (see [Section 13.2.3.7.3.4, L3 Firewall Error Logging](#))
- Flag mux errors

#### 13.2.3.8.1 Global Error-Routing Scheme

[Figure 13-6](#) shows the L3 global error-routing scheme.

Figure 13-6. L3 Global Error-Routing Scheme



13-005

### 13.2.3.8.2 Slave NIU Error Logging

Error logging is implemented only at slave NIUs. Because the interconnect does not support master NIU error logging, an erroneous packet must be created and sent to one of the slave NIUs. The slave NIU that receives an erroneous packet is predictable but can change per master (see [Table 13-29](#)).

**Table 13-29. L3 Connectivity and Holes Error Routing**

Master	Connectivity and Holes Errors Logged Into Slave NIUs
L3_MN_DAP	L3_SN_GPMC
L3_MN_CORTEXM3	
L3_MN_CORTEXA9	
L3_MN_SDMA_RD	
L3_MN_SDMA_WR	
L3_MN_DSP	L3_SN_DMM1
L3_MN_IVANHD	
L3_MN_MMC1	
L3_MN_MMC2	
L3_MN_SGX	L3_SN_DMM2
L3_MN_DSS	
L3_MN_ISS	
L3_MN_FACEDETECT	
L3_MN_FSUSB	L3_SN_OCM_RAM
L3_MN_HSUSBOTG	
L3_MN_HSI	L3_SN_ABE
L3_MN_HSUSBHOST	
L3_MN_C2C	L3_SN_L4_PER3

The slave NIU can be configured to report standard errors (errors generated within the interconnect):

- Firewall error: Protection violation; this error indicates that a request was rejected by a firewall and is reported to the control module. For more information, see [Section 13.2.3.7.3.4, L3 Firewall Error Logging](#).
- Address hole: This error reports an unknown address for a request. The address map is local to each master NIU; therefore, an address hole error is reported each time a master NIU requests an access to a slave NIU to which it is not logically connected, even if this address exists in the global L3 address map. This error is detected only once per burst.
- Unsupported commands: This error reports that the master NIU sent a command that cannot be processed, because the slave NIU cannot accept it and no conversion to another command is possible. This error is detected only once per burst.
- Report custom errors: Basically, when the slave answer is SResp = ERR
- Report severity level, for standard error and custom errors:
  - None: Error logging for this type of error is disabled.
  - Error: Error is logged for this type of error.
  - Fault: Error is logged and interrupt is generated for this type of error.
- Generate interrupt on 2 bits depending on the MReqDebug qualifier:
  - Application error -> Fault[0]
  - Debug error -> Fault[1]

By default, all slave NIUs are configured with standard and custom error levels set to FAULT. The errors are reported on the two flag muxes (see [Figure 13-6](#)), depending on the access type, application or debug. For more information, see [Section 13.2.3.8.3, Flag Mux Error Logging](#).

The slave NIU power-disconnect component also has error logging enabled. This is because the slave NIU in this case is in a clock domain that is switched off and therefore cannot catch the error. By nature, this component can generate only standard errors. By default, it is configured with the error level set to FAULT.

Wake up on demand: If an error packet reaches a slave NIU that is set with MDiscBehave = 1 (wake up on demand), then the active signal is asserted and L3 processes the error generation when the slave is awake. This is inefficient but it simplifies NIU implementation and should not be a problem, because errors are supposed to occur only during software debug.



**13.2.3.8.3 Flag Mux Error Logging**

All fault signals are sent to a flag mux component. There are four important FLAGMUX registers:

- [L3\\_FLAGMUX\\_MASK0](#): Masks application error sources
- [L3\\_FLAGMUX\\_MASK1](#): Masks debug error sources
- [L3\\_FLAGMUX\\_REGERR0](#): Checks which application error sources are active
- [L3\\_FLAGMUX\\_REGERR1](#): Checks which debug error sources are active

The two L3\_FLAGMUXMASK registers mask bit 0 or bit 1 of the flag inputs, and the L3\_FLAGMUX\_REGERR registers read input errors. Each register is dedicated to reporting the bit corresponding to the register number.

[Table 13-30](#) describes the mapping of the flags to the corresponding sources.

**Table 13-30. Interconnect Flag Mapping**

	Flag Mux Input	Source
CLK1 flag mux	0	L3_SN_DMM1
	1	L3_SN_DMM2
	2	L3_SN_ABE
	3	L3_SN_L4CFG
	4	CLK2 PWR DISC
CLK2 flag mux	0	L3_SN_Cortex-M3
	1	L3_SN_DSS
	2	L3_SN_GPMC
	3	L3_SN_ISS
	4	L3_SN_IVAHD
	6	L3_SN_L4PER0
	7	L3_SN_OCMCRAM
	8	L3_SN_GPMC
	9	L3_SN_SGX
	10	L3_SN_SL2
	11	L3_SN_C2C
	12	CLK1 PWR DISC
	15	L3_SN_L4PER3
16	L3_SN_L4PER1	
17	L3_SN_L4PER2	
CLK3 flag mux	0	DEBUGSS

**13.2.3.8.4 Severity Level of Standard and Custom Errors**

The slave NIU registers are important for error logging.

- The [L3\\_TARG\\_STDERRLOG\\_SVRTSTDLVL](#) register shows the severity level for standard errors. According to the severity level, error logging is disabled, enabled with level ERROR, or enabled with level ERROR and flag FAULT.
- The [L3\\_TARG\\_STDERRLOG\\_SVRTCUSTOMLVL](#) register shows the severity level for custom errors (see [Table 13-114](#)).
- The [L3\\_TARG\\_STDERRLOG\\_MAIN](#) register is the main register for error-logging management. It shows the validity of the logged information, standard or custom.
- The [L3\\_TARG\\_STDERRLOG\\_HDR](#) register stores packets in case of a standard error.
- The [L3\\_TARG\\_STDERRLOG\\_MSTADDR](#) register returns the MSTADDR field of the logged packet.
- The [L3\\_TARG\\_STDERRLOG\\_SLVADDR](#) register returns the SLVADDR field of the stored packet.
- The [L3\\_TARG\\_STDERRLOG\\_INFO](#) register saves the information field of the logged packet.

## 13.2.4 L3 Interconnect Programming Guide

### 13.2.4.1 L3 Interconnect Low-Level Programming Models

This section describes the low-level hardware programming sequences for configuring and using the L3 interconnect module.

#### 13.2.4.1.1 Global Initialization

##### 13.2.4.1.1.1 Global Initialization of Surrounding Modules

This section identifies the requirements for initializing the surrounding modules when the L3 interconnect module is to be used for the first time after a device reset. The initialization of surrounding modules is based on the integration and environment of the L3 interconnect. For more information, see [Section 13.2.2, L3 Interconnect Integration](#).

**Table 13-31. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	For information about the configuration of the PRCM module, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Control module	For information about the configuration of the control module, see <a href="#">Chapter 18, Control Module</a> .
MPU INTC	The MPU INTC must be configured to enable interrupts from the L3 interconnect module. See <a href="#">Chapter 17, Interrupt Controllers</a> .
sDMA	For information about the configuration of the sDMA, see <a href="#">Chapter 16, sDMA</a> .
L3 interconnect	For information about the configuration of the interconnect, see <a href="#">Chapter 13, Interconnect</a> .

### 13.2.4.2 Operational Modes Configuration

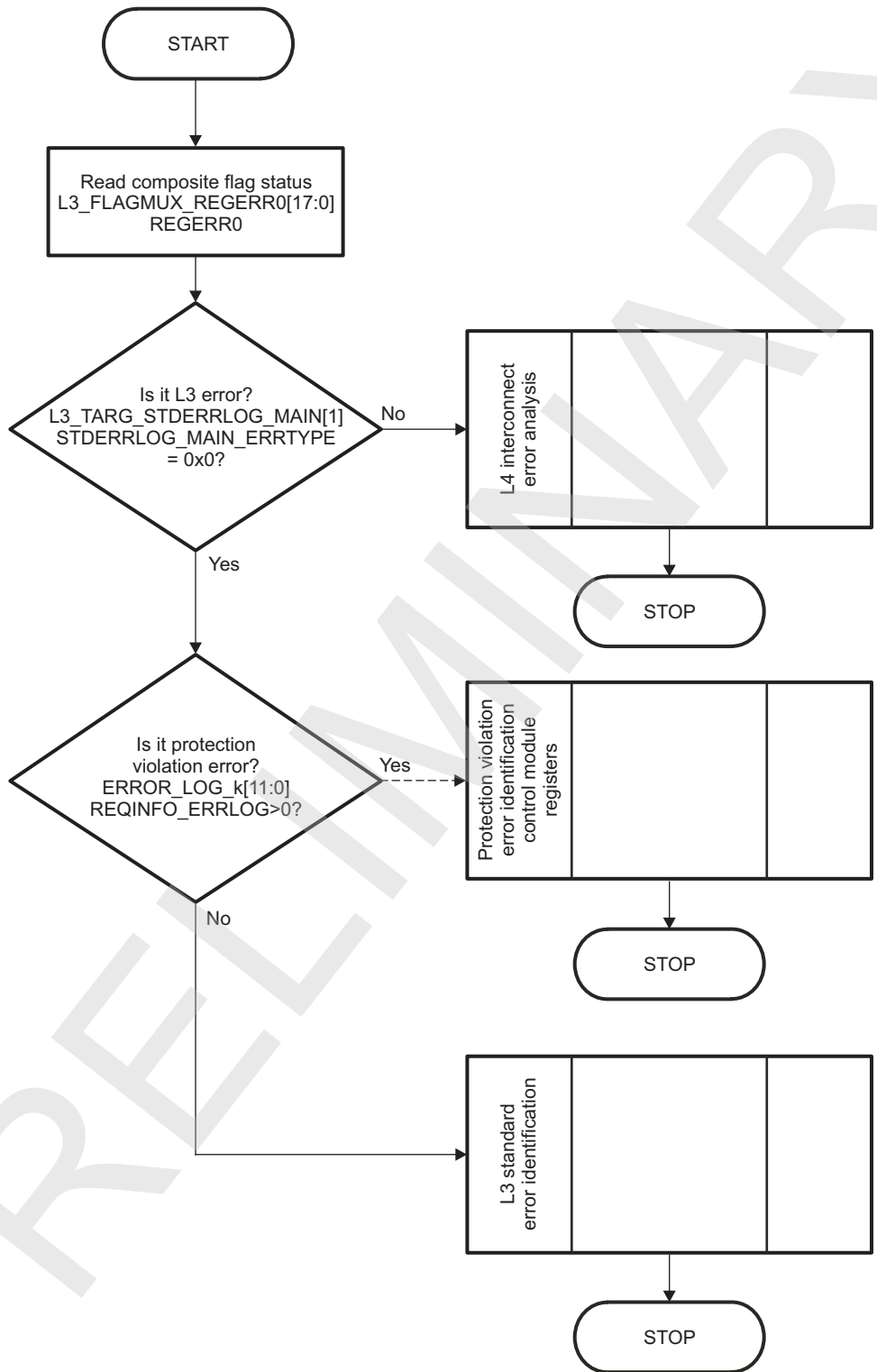
#### 13.2.4.2.1 L3 Interconnect Error Analysis Mode

##### 13.2.4.2.1.1 Main Sequence: L3 Interconnect Error Analysis Mode

The information required to analyze an error source is logged in several registers. The number of registers to access depends on the error source.

[Figure 13-7](#) shows the software sequence required in most cases.

Figure 13-7. Typical Error Analysis Sequence



I3-013

Table 13-32 lists the register call summary for error analysis mode in the main sequence.

**Table 13-32. Register Call Summary: Main Sequence – Error Analysis Mode**

Register Name	Register Name	Register Name
ERROR_LOG_k	L3_FLAGMUX_REGERR0	L3_FLAGMUX_REGERR1
L3_TARG_STDERRLOG_MAIN	CONTROL.CONTROL_SEC_ERR_STAT US_FUNC	CONTROL.CONTROL_SEC_ERR_STAT US_DEBUG

Table 13-33 lists the subprocess call summary for error analysis mode in the main sequence.

**Table 13-33. Subprocess Call Summary for Main Sequence – Error Analysis Mode**

Subprocess	Cross-Reference
L4 interconnect error analysis	See Section 13.2.4.2.1, <i>L4 Interconnect Error Analysis Mode</i> .
L3 interconnect protection violation error identification	See Section 13.2.4.2.1.1.1, <i>Subsequence: L3 Interconnect Protection Violation Error Identification</i> .
L3 interconnect unsupported command/address hole error identification	See Section 13.2.4.2.1.1.2, <i>Subsequence: L3 Interconnect Unsupported Command/Address Hole Error Identification</i> .
L3 interconnect reset FLAGMUX and module	See Section 13.2.4.2.1.1.3, <i>Subsequence: L3 Interconnect Reset FLAGMUX and Module</i> .

#### 13.2.4.2.1.1.1 Subsequence: L3 Interconnect Protection Violation Error Identification

The procedure listed in Table 13-34 describes protection violation error identification and where it is logged in the control module registers. Two types of errors are logged: application errors and debug errors.

**Table 13-34. Protection Violation Error Identification**

Step	Register/Bit Field/Programming Model	Value
Read the burst violation.	ERROR_LOG_k[23] BLK_BURST_VIOLATION	xxx
Read the initiator ID.	ERROR_LOG_k[11:8] CONNID	xxx
Read the command that caused the error.	ERROR_LOG_k[7] MCMD	xxx
Read the address of the request that caused the error.	LOGICAL_ADDR_ERRLOG_k[31:12] SLVOFS_LOGICAL	xxx
<b>IF:</b> Is it an application error?	L3_FLAGMUX_REGERR0[17:0] REGERROR0	= 0x0
Read the status bits to see which module firewall has worked.	CONTROL.CONTROL_SEC_ERR_STATUS_FUNC[20:0]	xxx
Clear the status bits.	CONTROL.CONTROL_SEC_ERR_STATUS_FUNC[20:0]	xxx
Clear the status bit.	L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_MAIN_CLRLOG	0x0
<b>ELSE IF</b>	L3_FLAGMUX_REGERR1[17:0] REGERROR1	= 0x1
Read the status bits to see the module.	CONTROL.CONTROL_SEC_ERR_STATUS_DEBUG[20:0]	xxx
Clear the status bits.	CONTROL.CONTROL_SEC_ERR_STATUS_DEBUG[20:0]	xxx
Clear the status bit.	L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_MAIN_CLRLOG	0x0
<b>ENDIF</b>		
Clear the burst violation.	ERROR_LOG_k[23] BLK_BURST_VIOLATION	0x0
Clear the error status.	ERROR_LOG_k[21:17] REGION_START_ERRLOG	0x00

#### 13.2.4.2.1.1.2 Subsequence: L3 Interconnect Standard Error Identification

The procedure listed in Table 13-35 describes the identification of standard errors inside the L3 interconnect. The standard errors are: unsupported command, address hole, and disconnect.

**Table 13-35. L3 Standard Error Identification**

Step	Register/Bit Field/Programming Model	Value
<b>IF:</b> Is an error detected?	L3_TARG_STDERRLOG_MAIN[18] STDERRLOG_MAIN_ERRCNT	= 0x1

**Table 13-35. L3 Standard Error Identification (continued)**

Step	Register/Bit Field/Programming Model	Value
Read the corresponding flag.	<a href="#">L3_FLAGMUX_REGERR0</a> [17:0] REGERROR0	xxx
Read the corresponding flag.	<a href="#">L3_FLAGMUX_REGERR1</a> [17:0] REGERROR1	xxx
Localize the slave NIU that generated the error.	See <a href="#">Table 13-30</a> , <i>Interconnect Flag Mapping</i> .	
<b>ELSE</b>		
Clear the error log.	<a href="#">L3_TARG_STDERRLOG_MAIN</a> [31] STDERRLOG_MAIN_CLRLOG	0x0
Clear the severity error status.	<a href="#">L3_TARG_STDERRLOG_SVRTSTDLVL</a> [1:0] STDERRLOG_SVRTSTDLVL_0	0x2
<b>ENDIF</b>		

#### 13.2.4.2.1.1.3 Subsequence: L3 Interconnect FLAGMUX Configuration

The procedure listed in [Table 13-36](#) gives information about the configuration of FLAGMUX masks.

**Table 13-36. FLAGMUX Configuration**

Step	Register/Bit Field/Programming Model	Value
Set the FLAGMUX masks to mask an event.	<a href="#">L3_FLAGMUX_MASK0</a> [17:0] MASK0 <a href="#">L3_FLAGMUX_MASK1</a> [17:0] MASK1	xxx
Read the REGERR bits to see if an error is recorded.	<a href="#">L3_FLAGMUX_REGERR0</a> [17:0] REGERR0 <a href="#">L3_FLAGMUX_REGERR1</a> [17:0] REGERR1	xxx
Clear the slave NIU error log and the FLAGMUX error.	<a href="#">L3_TARG_STDERRLOG_MAIN</a> [31] STDERRLOG_SVRTSTDLVL_0	0x2

## 13.2.5 L3 Interconnect Register Manual

### 13.2.5.1 L3 Register Group Summary

The registers in the L3 interconnect are divided into eight groups:

- Firewall registers [Table 13-38](#)
- HOST registers [Table 13-61](#)
- TARG registers [Table 13-97](#)
- PWR\_DISK registers [Table 13-139](#)
- FLAGMUX registers [Table 13-163](#)
- BW registers [Table 13-178](#)
- RA registers [Table 13-193](#)
- STATCOLL registers [Table 13-201](#)

#### 13.2.5.1.1 L3 Firewall Registers Summary and Description

**Table 13-37. L3 Firewall Instance Summary**

Module Name	Base Address	Size
C2C-master NIU firewall	0x4A20 4000	4KB
C2C-slave NIU firewall	0x4A20 6000	4KB
EMIF firewall	0x4A20 C000	4KB
GPMC firewall	0x4A21 0000	4KB
L3 RAM firewall	0x4A21 2000	4KB
SGX firewall	0x4A21 4000	4KB
ISS firewall	0x4A21 6000	4KB
Dual Cortex-M3 firewall	0x4A21 8000	4KB
DSS firewall	0x4A21 C000	4KB
SL2 firewall	0x4A21 E000	4KB
IVA-HD firewall	0x4A22 0000	4KB
Debug firewall	0x4A22 6000	4KB
L4-ABE firewall	0x4A22 8000	4KB

#### 13.2.5.1.1.1 L3 Firewall Registers Summary

**Table 13-38. L3 Firewall Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	EMIF Firewall L3 Physical Address	GPMC Firewall L3 Physical Address	L3 RAM Firewall L3 Physical Address
<a href="#">ERROR_LOG_k<sup>(1)</sup></a>	RW	32	0x000+(0x10*k)	0x4A20 C000+(0x10*k)	0x4A21 0000+(0x10*k)	0x4A21 2000+(0x10*k)
<a href="#">LOGICAL_ADDR_ERRLOG_k<sup>(1)</sup></a>	RO	32	0x004+(0x10*k)	0x4A20 C004+(0x10*k)	0x4A21 0004+(0x10*k)	0x4A21 2004+(0x10*k)
<a href="#">REGUPDATE_CONTROL</a>	RW	32	0x040	0x4A20 C040	0x4A21 0040	0x4A21 2040
<a href="#">START_REGION_i<sup>(2)</sup></a>	RW	32	0x080+(0x10*i)	0x4A20 C080+(0x10*i)	0x4A21 0080+(0x10*i)	0x4A21 2080+(0x10*i)
<a href="#">END_REGION_i<sup>(2)</sup></a>	RW	32	0x084+(0x10*i)	0x4A20 C084+(0x10*i)	0x4A21 0084+(0x10*i)	0x4A21 2084+(0x10*i)

<sup>(1)</sup> k = 0 to 2 for EMIF Firewall  
k = 0 for GPMC Firewall  
k = 0 for L3 RAM Firewall

<sup>(2)</sup> i = 1 to 7 for EMIF Firewall  
i = 1 to 7 for GPMC Firewall  
i = 1 to 9 for L3 RAM Firewall

**Table 13-38. L3 Firewall Registers Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	EMIF Firewall L3 Physical Address	GPMC Firewall L3 Physical Address	L3 RAM Firewall L3 Physical Address
<a href="#">MRM_PERMISSION_REGION_HIG_H<sub>j</sub></a> <sup>(3)</sup>	RW	32	0x08C+(0x10*j)	0x4A20 C08C+(0x10*j)	0x4A21 008C+(0x10*j)	0x4A21 208C+(0x10*j)
<a href="#">MRM_PERMISSION_REGION_LOW_j</a> <sup>(3)</sup>	RW	32	0x088+(0x10*j)	0x4A20 C088+(0x10*j)	0x4A21 0088+(0x10*j)	0x4A21 2088+(0x10*j)

- <sup>(3)</sup> j = 0 to 7 for EMIF Firewall  
 j = 0 to 7 for GPMC Firewall  
 j = 0 to 9 for L3 RAM Firewall

**Table 13-39. L3 Firewall Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SGX Firewall L3 Physical Address	ISS Firewall L3 Physical Address	Dual Cortex-M3 Firewall L3 Physical Address
<a href="#">ERROR_LOG_k</a> <sup>(1)</sup>	RW	32	0x000+(0x10*k)	0x4A21 4000+(0x10*k)	0x4A21 6000+(0x10*k)	0x4A21 8000+(0x10*k)
<a href="#">LOGICAL_ADDR_ERRLOG_k</a> <sup>(1)</sup>	RO	32	0x004+(0x10*k)	0x4A21 4004+(0x10*k)	0x4A21 6004+(0x10*k)	0x4A21 8004+(0x10*k)
<a href="#">REGUPDATE_CONTROL</a>	RW	32	0x040	0x4A21 4040	0x4A21 6040	0x4A21 8040
<a href="#">START_REGION_i</a> <sup>(2)</sup>	RW	32	0x080+(0x10*i)	N/A	N/A	N/A
<a href="#">END_REGION_i</a> <sup>(2)</sup>	RW	32	0x084+(0x10*i)	N/A	N/A	N/A
<a href="#">MRM_PERMISSION_REGION_HIGH_j</a> <sup>(3)</sup>	RW	32	0x08C+(0x10*j)	0x4A21 408C+(0x10*j)	0x4A21 608C+(0x10*j)	0x4A21 808C+(0x10*j)
<a href="#">MRM_PERMISSION_REGION_LOW_j</a> <sup>(3)</sup>	RW	32	0x088+(0x10*j)	0x4A21 4088+(0x10*j)	0x4A21 6088+(0x10*j)	0x4A21 8088+(0x10*j)

- <sup>(1)</sup> k = 0 for SGX Firewall  
 k = 0 for ISS Firewall  
 k = 0 for Dual Cortex-M3 Firewall
- <sup>(2)</sup> i = 0 for SGX Firewall  
 i = 0 for ISS Firewall  
 i = 0 for Dual Cortex-M3 Firewall
- <sup>(3)</sup> j = 0 for SGX Firewall  
 j = 0 for ISS Firewall  
 j = 0 for Dual Cortex-M3 Firewall

**Table 13-40. L3 Firewall Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DSS Firewall L3 Physical Address	SL2 Firewall L3 Physical Address	IVA-HD Firewall L3 Physical Address
<a href="#">ERROR_LOG_k</a> <sup>(1)</sup>	RW	32	0x000+(0x10*k)	0x4A21 C000+(0x10*k)	0x4A21 E000+(0x10*k)	0x4A22 0000+(0x10*k)
<a href="#">LOGICAL_ADDR_ERRLOG_k</a> <sup>(1)</sup>	RO	32	0x004+(0x10*k)	0x4A21 C004+(0x10*k)	0x4A21 E004+(0x10*k)	0x4A22 0004+(0x10*k)
<a href="#">REGUPDATE_CONTROL</a>	RW	32	0x040	0x4A21 C040	0x4A21 E040	0x4A22 0040
<a href="#">START_REGION_j</a>	RW	32	0x080+(0x10*i)	N/A	N/A	N/A
<a href="#">END_REGION_j</a>	RW	32	0x084+(0x10*i)	N/A	N/A	N/A
<a href="#">MRM_PERMISSION_REGION_HIGH_j</a> <sup>(2)</sup>	RW	32	0x08C+(0x10*j)	0x4A21 C08C+(0x10*j)	0x4A21 E08C+(0x10*j)	0x4A22 008C+(0x10*j)
<a href="#">MRM_PERMISSION_REGION_LOW_j</a> <sup>(2)</sup>	RW	32	0x088+(0x10*j)	0x4A21 C088+(0x10*j)	0x4A21 E088+(0x10*j)	0x4A22 0088+(0x10*j)

- <sup>(1)</sup> k = 0 for DSS Firewall  
 k = 0 for SL2 Firewall  
 k = 0 for IVA-HD Firewall
- <sup>(2)</sup> j = 0 for DSS Firewall  
 j = 0 for SL2 Firewall  
 j = 0 for IVA-HD Firewall

**Table 13-41. L3 Firewall Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Debug Firewall L3 Physical Address	L4-ABE Firewall L3 Physical Address
<a href="#">ERROR_LOG_k</a> <sup>(1)</sup>	RW	32	0x000+(0x10*k)	0x4A22 6000+(0x10*k)	0x4A22 8000+(0x10*k)
<a href="#">LOGICAL_ADDR_ERRLOG_k</a> <sup>(1)</sup>	RO	32	0x004+(0x10*k)	0x4A22 6004+(0x10*k)	0x4A22 8004+(0x10*k)
<a href="#">REGUPDATE_CONTROL</a>	RW	32	0x040	0x4A22 6040	0x4A22 8040
<a href="#">START_REGION_i</a> <sup>(2)</sup>	RW	32	0x080+(0x10*i)	0x4A22 6080+(0x10*i)	N/A
<a href="#">END_REGION_i</a> <sup>(2)</sup>	RW	32	0x084+(0x10*i)	0x4A22 6084+(0x10*i)	N/A
<a href="#">MRM_PERMISSION_REGION_HIGH_j</a> <sup>(3)</sup>	RW	32	0x08C+(0x10*j)	0x4A22 608C+(0x10*j)	0x4A22 808C+(0x10*j)
<a href="#">MRM_PERMISSION_REGION_LOW_j</a> <sup>(3)</sup>	RW	32	0x088+(0x10*j)	0x4A22 6088+(0x10*j)	0x4A22 8088+(0x10*j)

<sup>(1)</sup> k = 0 for Debug Firewall  
k = 0 for L4-ABE Firewall

<sup>(2)</sup> i = 1 for Debug Firewall  
i = 0 for L4-ABE Firewall

<sup>(3)</sup> j = 0 to 1 for Debug Firewall  
j = 0 for L4-ABE Firewall

**Table 13-42. L3 Firewall Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	C2C-Master NIU Firewall L3 Physical Address	C2C-Slave NIU Firewall L3 Physical Address
<a href="#">ERROR_LOG_k</a> <sup>(1)</sup>	RW	32	0x000+(0x10*k)	0x4A20 4000+(0x10*k)	0x4A20 6000+(0x10*k)
<a href="#">LOGICAL_ADDR_ERRLOG_k</a> <sup>(1)</sup>	RO	32	0x004+(0x10*k)	0x4A20 4004+(0x10*k)	0x4A22 6004+(0x10*k)
<a href="#">REGUPDATE_CONTROL</a>	RW	32	0x040	0x4A20 4040	0x4A22 6040
<a href="#">START_REGION_i</a> <sup>(2)</sup>	RW	32	0x080+(0x10*i)	0x4A20 4080+(0x10*i)	N/A
<a href="#">END_REGION_i</a> <sup>(2)</sup>	RW	32	0x084+(0x10*i)	0x4A20 4084+(0x10*i)	N/A
<a href="#">MRM_PERMISSION_REGION_HIGH_j</a> <sup>(3)</sup>	RW	32	0x08C+(0x10*j)	0x4A20 408C+(0x10*j)	0x4A22 608C+(0x10*j)
<a href="#">MRM_PERMISSION_REGION_LOW_j</a> <sup>(3)</sup>	RW	32	0x088+(0x10*j)	0x4A20 4088+(0x10*j)	0x4A22 6088+(0x10*j)

<sup>(1)</sup> k = 0 for C2C-Master NIU Firewall  
k = 0 for C2C-Slave NIU Firewall

<sup>(2)</sup> i = 1 to 3 for C2C-Master NIU Firewall

<sup>(3)</sup> j = 0 for C2C-Master NIU Firewall  
j = 0 to 3 for C2C-Slave NIU Firewall



13.2.5.1.1.2 L3 Firewall Registers Description

**NOTE:** Hardware reset values can be modified by exported values from the control module at reset.

**Table 13-43. ERROR\_LOG\_k**

<b>Address Offset</b>	0x0000 0000+(0x10*k)	<b>Index</b>	See Table 13-38 to Table 13-42.
<b>Physical Address</b>	See Table 13-38 to Table 13-42.		
<b>Description</b>	Error log register for port k		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BLK_BURST_VIOLATION	RESERVED	REGION_START_ERRLOG				REGION_END_ERRLOG				REQINFO_ERRLOG													

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads return 0s.	R	0x00
23	BLK_BURST_VIOLATION	Read 0x1: 2D burst not allowed or exceeding allowed size Write to clear <a href="#">ERROR_LOG_k</a> and <a href="#">LOGICAL_ADDR_ERRLOG_k</a> registers	RW	0
22	RESERVED	Reads return 0s.	R	0
21:17	REGION_START_ERRLOG	Read: Wrong access hit this region number Write to clear <a href="#">ERROR_LOG_k</a> and <a href="#">LOGICAL_ADDR_ERRLOG_k</a> registers	RW	0x00
16:12	REGION_END_ERRLOG	Read: Wrong access hit this region number Write to clear <a href="#">ERROR_LOG_k</a> and <a href="#">LOGICAL_ADDR_ERRLOG_k</a> registers	RW	0x00
11:0	REQINFO_ERRLOG	Read: Error in reqinfo vector mapped as follows: [11: 8] ConnID [7] MCMD [6:4] Reserved [3] MReqDebug [2] Reserved [1] MReqSupervisor [0] MReqType Write to clear <a href="#">ERROR_LOG_k</a> and <a href="#">LOGICAL_ADDR_ERRLOG_k</a> registers	RW	0x000

**Table 13-44. Register Call Summary for Register ERROR\_LOG\_k**

L3 Interconnect

- [L3 Protection and Firewalls](#): [0] [1] [2] [3] [4] [5] [6] [7]
- [Operational Modes Configuration](#): [8] [9] [10] [11] [12] [13]
- [L3 Register Group Summary](#): [14] [15] [16] [17] [18] [19] [20] [21] [22]

**Table 13-45. LOGICAL\_ADDR\_ERRLOG\_k**

<b>Address Offset</b>	0x0000 0004+(0x10*k)	<b>Index</b>	See Table 13-38 to Table 13-42.
<b>Physical Address</b>	See Table 13-38 to Table 13-42.		
<b>Description</b>	Logical Physical Address Error log register for port k		
<b>Type</b>	RO		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLVOFS_LOGICAL																															

Bits	Field Name	Description	Type	Reset
31 <sup>(1)</sup> :0	SLVOFS_LOGICAL	Address generated by the Initiator before being translated	R	0x00000

<sup>(1)</sup> \* = Size of the target

**Table 13-46. Register Call Summary for Register LOGICAL\_ADDR\_ERRLOG\_k**

L3 Interconnect

- [L3 Protection and Firewalls: \[0\] \[1\] \[2\] \[3\]](#)
- [Operational Modes Configuration: \[4\]](#)
- [L3 Register Group Summary: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)

**Table 13-47. REGUPDATE\_CONTROL**

<b>Address Offset</b>	0x0000 0040
<b>Physical Address</b>	See Table 13-38 to Table 13-42.
<b>Description</b>	Register update control register
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														FW_LOAD_REQ	BUSY_REQ

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads return 0s.	R	0x0000 0000
1	FW_LOAD_REQ	Hardware set/Software clear	R	0x1
0	BUSY_REQ	Busy request	RW	0x0

**Table 13-48. Register Call Summary for Register REGUPDATE\_CONTROL**

L3 Interconnect

- [L3 Protection and Firewalls: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [L3 Register Group Summary: \[9\] \[10\] \[11\] \[12\] \[13\]](#)

**Table 13-49. START\_REGION\_i**

<b>Address Offset</b>	0x0000 0080+(0x10*i)	<b>Index</b>	See <a href="#">Table 13-38</a> to <a href="#">Table 13-42</a> .
<b>Physical Address</b>	See <a href="#">Table 13-38</a> to <a href="#">Table 13-42</a> .		
<b>Description</b>	Start physical address of region i		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_REGION																RESERVED															

Bits	Field Name	Description	Type	Reset
31:12	START_REGION	Physical target start address of firewall region i. The size of this bit field depends on target addressable space, the maximum is [31:12]. See <a href="#">Table 13-51</a> .	RW	0x00000
11:0	RESERVED	Reads return 0s.	R	0x0000

**Table 13-50. Register Call Summary for Register START\_REGION\_i**

L3 Interconnect

- [L3 Protection and Firewalls: \[0\] \[1\]](#)
- [L3 Register Group Summary: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)

**Table 13-51. Size of START\_REGION\_i[] START\_REGION Bit Field**

Firewall	Bit Field
EMIF firewall	<a href="#">START_REGION_i[31:12]</a> START_REGION
GPMC firewall	<a href="#">START_REGION_i[30:12]</a> START_REGION
L3 RAM firewall	<a href="#">START_REGION_i[15:12]</a> START_REGION
SGX firewall	N/A
ISS firewall	N/A
Dual Cortex-M3 firewall	N/A
DSS firewall	N/A
SL2 firewall	N/A
IVA-HD firewall	N/A
Debug firewall	<a href="#">START_REGION_i[21:12]</a> START_REGION
L4-AUDIO firewall	N/A
C2C-Target firewall	N/A
C2C-Init firewall	<a href="#">START_REGION_i[28:12]</a> START_REGION

**Table 13-52. END\_REGION\_i**

<b>Address Offset</b>	0x0000 0084+(0x10*i)	<b>Index</b>	See <a href="#">Table 13-38</a> to <a href="#">Table 13-42</a> .
<b>Physical Address</b>	See <a href="#">Table 13-38</a> to <a href="#">Table 13-42</a> .		
<b>Description</b>	End physical address of region i		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
END_REGION																RESERVED											REGION_ENABLE_PORT2	REGION_ENABLE_PORT1	REGION_ENABLE_PORT0		

Bits	Field Name	Description	Type	Reset
31:12	END_REGION	Physical target end address of firewall region i. The size of this bit field depends on target addressable space, the maximum is [31:12]. See <a href="#">Table 13-54</a> .	RW	0x00000
11:3	RESERVED	Reads return 0s.	R	0x0000
2	REGION_ENABLE_PORT2	Enable this region for port 2 <sup>(1)</sup> .	RW	0x0
1	REGION_ENABLE_PORT1	Enable this region for port 1 <sup>(2)</sup> .	RW	0x0
0	REGION_ENABLE_PORT0	Enable this region for port 0.	RW	0x0

<sup>(1)</sup> Only for multiport firewalls with at least 2 ports: EMIF Firewall

<sup>(2)</sup> Only for multiport firewalls with at least 1 port: EMIF and Crypto Firewall

**Table 13-53. Register Call Summary for Register END\_REGION\_i**

L3 Interconnect

- [L3 Protection and Firewalls: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [L3 Register Group Summary: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

**Table 13-54. Size of END\_REGION\_i[] END\_REGION Bit Field**

Firewall	Bit Field
EMIF firewall	<a href="#">END_REGION_i[31:12]</a> END_REGION
GPMC firewall	<a href="#">END_REGION_i[30:12]</a> END_REGION
L3 RAM firewall	<a href="#">END_REGION_i[15:12]</a> END_REGION
SGX firewall	N/A
ISS firewall	N/A
Dual Cortex-M3 firewall	N/A
DSS firewall	N/A
SL2 firewall	N/A
IVA-HD firewall	N/A
Debug firewall	<a href="#">END_REGION_i[21:12]</a> END_REGION
L4-AUDIO firewall	N/A
C2C-Target firewall	N/A
C2C-Init firewall	<a href="#">END_REGION_i[28:12]</a> END_REGION

**Table 13-55. MRM\_PERMISSION\_REGION\_HIGH\_j**

<b>Address Offset</b>	0x0000 008C+(0x10*i)	<b>Index</b>	See <a href="#">Table 13-38</a> to <a href="#">Table 13-42</a> .
<b>Physical Address</b>	See <a href="#">Table 13-38</a> to <a href="#">Table 13-42</a> .		
<b>Description</b>	Region j Permission High		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED								W12	R12	RESERVED	RESERVED	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	RW	0x3F
25	W12	Master NIU ConnID = 12 write permission	RW	0x1
24	R12	Master NIU ConnID = 12 read permission	RW	0x1
23	RESERVED	Reserved	RW	0x1
22	RESERVED	Reserved	RW	0x1
21	W10	Master NIU ConnID = 10 write permission	RW	0x1
20	R10	Master NIU ConnID = 10 read permission	RW	0x1
19	W9	Master NIU ConnID = 9 write permission	RW	0x1
18	R9	Master NIU ConnID = 9 read permission	RW	0x1
17	W8	Master NIU ConnID = 8 write permission	RW	0x1
16	R8	Master NIU ConnID = 8 read permission	RW	0x1
15	W7	Master NIU ConnID = 7 write permission	RW	0x1
14	R7	Master NIU ConnID = 7 read permission	RW	0x1
13	W6	Master NIU ConnID = 6 write permission	RW	0x1
12	R6	Master NIU ConnID = 6 read permission	RW	0x1
11	W5	Master NIU ConnID = 5 write permission	RW	0x1
10	R5	Master NIU ConnID = 5 read permission	RW	0x1
9	W4	Master NIU ConnID = 4 write permission	RW	0x1
8	R4	Master NIU ConnID = 4 read permission	RW	0x1
7	W3	Master NIU ConnID = 3 write permission	RW	0x1
6	R3	Master NIU ConnID = 3 read permission	RW	0x1
5	W2	Master NIU ConnID = 2 write permission	RW	0x1
4	R2	Master NIU ConnID = 2 read permission	RW	0x1
3	W1	Master NIU ConnID = 1 write permission	RW	0x1
2	R1	Master NIU ConnID = 1 read permission	RW	0x1
1	W0	Master NIU ConnID = 0 write permission	RW	0x1
0	R0	Master NIU ConnID = 0 read permission	RW	0x1

**Table 13-56. Register Call Summary for Register MRM\_PERMISSION\_REGION\_HIGH\_j**

L3 Interconnect

- [L3 Protection and Firewalls: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [L3 Register Group Summary: \[6\] \[7\] \[8\] \[9\] \[10\]](#)

**Table 13-57. MRM\_PERMISSION\_REGION\_LOW\_j**

<b>Address Offset</b>	0x0000 0088+(0x10*i)	<b>Index</b>	See <a href="#">Table 13-38</a> to <a href="#">Table 13-42</a> .
<b>Physical Address</b>	See <a href="#">Table 13-38</a> to <a href="#">Table 13-42</a> .		
<b>Description</b>	Region j Permission Low		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	PUB_PRIV_DEBUG	PUB_USR_DEBUG	RESERVED	RESERVED	PUB_PRIV_WRITE	PUB_PRIV_READ	PUB_PRIV_EXE	PUB_USR_WRITE	PUB_USR_READ	PUB_USR_EXE	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
30	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
29	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
28	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
27	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
26	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
25	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
24	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
23	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
22	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
21	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
20	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
19	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
18	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
17	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
16	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
15	PUB_PRIV_DEBUG	Public Privilege Debug Allowed	RW	See <a href="#">Table 13-59</a> .
14	PUB_USR_DEBUG	Public User Debug Allowed	RW	See <a href="#">Table 13-59</a> .
13	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
12	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
11	PUB_PRIV_WRITE	Public Privilege Write Allowed	RW	See <a href="#">Table 13-59</a> .
10	PUB_PRIV_READ	Public Privilege Read Allowed	RW	See <a href="#">Table 13-59</a> .
9	PUB_PRIV_EXE	Public Privilege Exe Allowed	RW	See <a href="#">Table 13-59</a> .
8	PUB_USR_WRITE	Public User Write Access Allowed	RW	See <a href="#">Table 13-59</a> .
7	PUB_USR_READ	Public User Read Access Allowed	RW	See <a href="#">Table 13-59</a> .
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	See <a href="#">Table 13-59</a> .
5	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
4	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
3	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
2	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
1	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .
0	RESERVED	Reserved	RW	See <a href="#">Table 13-59</a> .

**Table 13-58. Register Call Summary for Register MRM\_PERMISSION\_REGION\_LOW\_j**

L3 Interconnect

- L3 Protection and Firewalls: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14]
- L3 Register Group Summary: [15] [16] [17] [18] [19]

**Table 13-59. Reset Value for MRM\_PERMISSION\_REGION\_LOW\_j**

Region	Reset Value
Region j = 0 (except EMIF firewall)	0xFFFF0000
Region j = 0 (for EMIF firewall)	0xFFFFFFFF
Region j > 0 (for all firewalls)	0xFFFFFFFF

**13.2.5.1.2 L3 Host Register Summary and Description**

**Table 13-60. HOST Instance Summary**

Module Name	Base Address	Size
CLK1_HOST_CLK1	0x4400 0000	8MB
CLK2_HOST_CLK2	0x4480 0000	8MB
CLK3_HOST_CLK3	0x4500 0000	8MB

**13.2.5.1.2.1 L3 HOST Register Summary**

**Table 13-61. HOST Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_HOST_C LK1 L3 Physical Address	CLK2_HOST_C LK2 L3 Physical Address	CLK3_HOST_C LK3 L3 Physical Address
L3_HOST_STDHOSHTDR_COREREG	R	32	0x0000 0000	0x4400 0000	0x4480 0000	0x4500 0000
L3_HOST_STDHOSHTDR_VERSIONREG	R	32	0x0000 0004	0x4400 0004	0x4480 0004	0x4500 0004
L3_HOST_STDHOSHTDR_MAINCTLREG	R	32	0x0000 0008	0x4400 0008	0x4480 0008	0x4500 0008
L3_HOST_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0040	0x4480 0040	0x4500 0040
L3_HOST_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0044	0x4480 0044	0x4500 0044
L3_HOST_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0048	0x4480 0048	0x4500 0048
L3_HOST_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 004C	0x4480 004C	0x4500 004C
L3_HOST_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0050	0x4480 0050	0x4500 0050
L3_HOST_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0054	0x4480 0054	0x4500 0054
L3_HOST_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0058	0x4480 0058	0x4500 0058
L3_HOST_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 005C	0x4480 005C	0x4500 005C
L3_HOST_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0060	0x4480 0060	0x4500 0060
L3_HOST_STDERRLOG_CUSTOMINFO_MST ADDR	R	32	0x0000 0064	0x4400 0064	0x4480 0064	0x4500 0064
L3_HOST_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0068	0x4400 0068	0x4480 0068	0x4500 0068
L3_HOST_STDERRLOG_CUSTOMINFO_WR	R	32	0x0000 006C	0x4400 006C	0x4480 006C	0x4500 006C
L3_HOST_STDERRLOG_CUSTOMINFO_ADD R	R	32	0x0000 0070	0x4400 0070	0x4480 0070	0x4500 0070
L3_HOST_STDERRLOG_CUSTOMINFO_DEC ERR	R	32	0x0000 0074	0x4400 0074	0x4480 0074	0x4500 0074

**13.2.5.1.2.2 L3 HOST Register Description**

**Table 13-62. L3\_HOST\_STDHOSTHDR\_COREREG**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b>	See <a href="#">Table 13-60</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x1A.	R	0x1A
15:1	RESERVED		R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

**Table 13-63. Register Call Summary for Register L3\_HOST\_STDHOSTHDR\_COREREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-64. L3\_HOST\_STDHOSTHDR\_VERSIONREG**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b>	See <a href="#">Table 13-60</a> .
<b>Description</b>			
<b>Type</b>	R		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

**Table 13-65. Register Call Summary for Register L3\_HOST\_STDHOSTHDR\_VERSIONREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-66. L3\_HOST\_STDHOSTHDR\_MAINCTLREG**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .		See <a href="#">Table 13-60</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STDHOSTHDR_MAINCTLREG_FLT	RESERVED		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	STDHOSTHDR_MAINCTLREG_FLT	Asserted when a Fault condition is detected: if the unit includes Error Logging, Flt is asserted when the FltCnt register field indicates a Fault, and deasserted when FltCnt is reset. If no Error Logging is implemented, this bit becomes unit-dependent. In all cases, Flt bit and Flt pin (service network) have the same logical level. Type: Status. Reset value: X.	R	0
1:0	RESERVED		R	0x0

**Table 13-67. Register Call Summary for Register L3\_HOST\_STDHOSTHDR\_MAINCTLREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-68. L3\_HOST\_STDERRLOG\_SVRTSTDLVL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	See Table 13-60.
<b>Physical Address</b>	See Table 13-61.		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							STDERRLOG_SVRTSTDLVL_0								

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	STDERRLOG_SVRTSTDLVL_0	Severity level parameters Type: Control. Reset value: 0x0. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x0

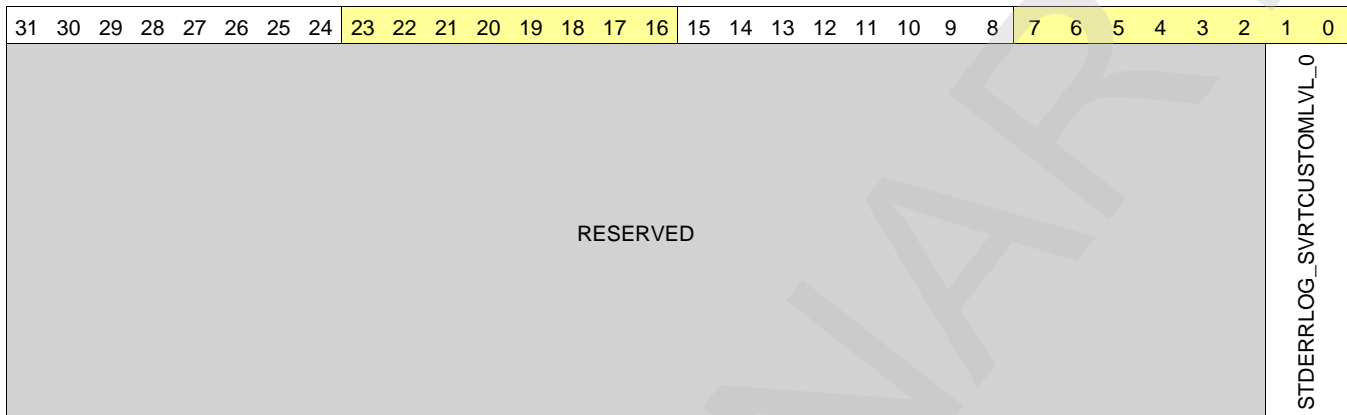
**Table 13-69. Register Call Summary for Register L3\_HOST\_STDERRLOG\_SVRTSTDLVL**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-70. L3\_HOST\_STDERRLOG\_SVRTCUSTOMLVL**

<b>Address Offset</b>	0x0000 0044	
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b> See <a href="#">Table 13-60</a> .
<b>Description</b>		
<b>Type</b>	RW	



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	STDERRLOG_SVRTCUSTOMLVL_0	Severity level parameters Type: Control. Reset value: 0x0. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x0

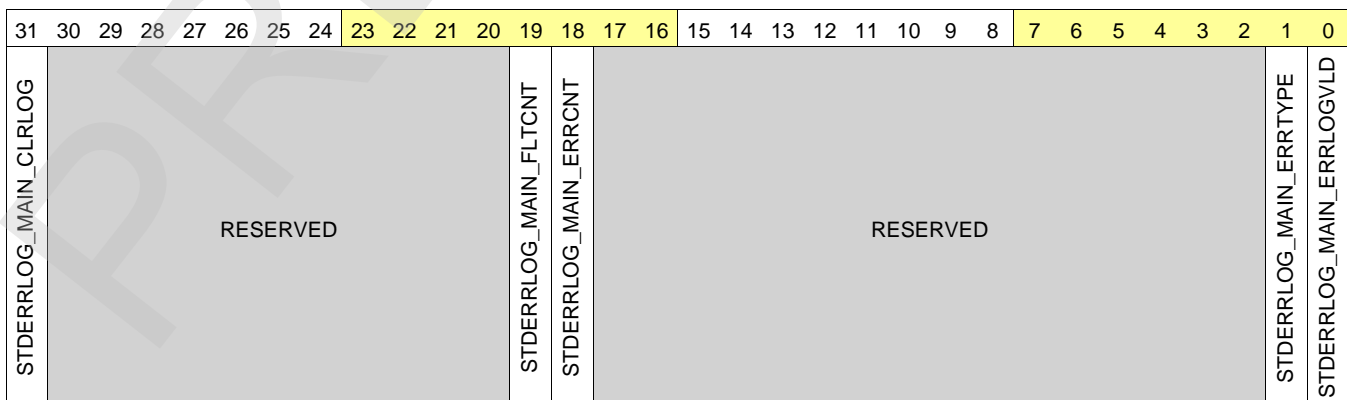
**Table 13-71. Register Call Summary for Register L3\_HOST\_STDERRLOG\_SVRTCUSTOMLVL**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-72. L3\_HOST\_STDERRLOG\_MAIN**

<b>Address Offset</b>	0x0000 0048	
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b> See <a href="#">Table 13-60</a> .
<b>Description</b>		
<b>Type</b>	RW	



Bits	Field Name	Description	Type	Reset
31	STDERRLOG_MAIN_CLRLOG	Clears "Error Logging Valid" bit when written to 1. Type: Give_AutoCleared. Reset value: 0x0.	RW	0
30:20	RESERVED		R	0x000
19	STDERRLOG_MAIN_FLTCNT	Asserted when at least one error with severity level FAULT is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
18	STDERRLOG_MAIN_ERRCNT	Asserted when at least one error with severity level ERROR is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
17:2	RESERVED		R	0x0000
1	STDERRLOG_MAIN_ERRTYPE	Indicates logging type. Type: Status. Reset value: X. Read 0x0: Logged Error format is standard (header and necker recorded). Read 0x1: Logged Error format is module dependent. CustomInfo register(s) may be implemented to store additional information.	R	0
0	STDERRLOG_MAIN_ERRLOGV LD	Error Logging Valid. Asserted when logging information is valid(indicates that an error has been logged). Type: Status. Reset value: X.	R	0

**Table 13-73. Register Call Summary for Register L3\_HOST\_STDERRLOG\_MAIN**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-74. L3\_HOST\_STDERRLOG\_HDR**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	See <a href="#">Table 13-60</a> .
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .		See <a href="#">Table 13-60</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDERRLOG_HDR_LEN1				RESERVED		STDERRLOG_HDR_STOPOFSWRPSZ				STDERRLOG_HDR_ERR		RESERVED				STDERRLOG_HDR_PRESSURE		RESERVED		STDERRLOG_HDR_OPCODE			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:18	STDERRLOG_HDR_LEN1	This field contains the number of payload cell(s) minus one of the logged packet. Type: Status. Reset value: X.	R	0x00
17:16	RESERVED		R	0x0
15:12	STDERRLOG_HDR_STOPOFS WRPSZ	StopOfs or WrapSize field of the logged packet (meaning depends on Wrp bit of logged opcode). Type: Status. Reset value: X.	R	0x0
11	STDERRLOG_HDR_ERR	Err bit of the logged packet. Type: Status. Reset value: X.	R	0

Bits	Field Name	Description	Type	Reset
10:7	RESERVED		R	0x0
6	STDERRLOG_HDR_PRESSURE	Pressure field of the logged packet. Type: Status. Reset value: X.	R	0
5:4	RESERVED		R	0x0
3:0	STDERRLOG_HDR_OPCODE	Opcode of the logged packet. Type: Status. Reset value: X.	R	0x0

**Table 13-75. Register Call Summary for Register L3\_HOST\_STDERRLOG\_HDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-76. L3\_HOST\_STDERRLOG\_MSTADDR**

<b>Address Offset</b>	0x0000 0050	
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b> See <a href="#">Table 13-60</a> .
<b>Description</b>		
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_MSTADDR															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0000000
7:0	STDERRLOG_MSTADDR	Master Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

**Table 13-77. Register Call Summary for Register L3\_HOST\_STDERRLOG\_MSTADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-78. L3\_HOST\_STDERRLOG\_SLVADDR**

<b>Address Offset</b>	0x0000 0054	
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b> See <a href="#">Table 13-60</a> .
<b>Description</b>		
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_SLVADDR															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x00000000
4:0	STDERRLOG_SLVADDR	Slave Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

**Table 13-79. Register Call Summary for Register L3\_HOST\_STDERRLOG\_SLVADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-80. L3\_HOST\_STDERRLOG\_INFO**

<b>Address Offset</b>	0x0000 0058	
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b> See <a href="#">Table 13-60</a> .
<b>Description</b>		
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_INFO															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_INFO	Info field of the logged packet. Type: Status. Reset value: X.	R	0x00

**Table 13-81. Register Call Summary for Register L3\_HOST\_STDERRLOG\_INFO**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-82. L3\_HOST\_STDERRLOG\_SLVOFSLSB**

<b>Address Offset</b>	0x0000 005C	
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b> See <a href="#">Table 13-60</a> .
<b>Description</b>		
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDERRLOG_SLVOFSLSB																															

Bits	Field Name	Description	Type	Reset
31:0	STDERRLOG_SLVOFSLSB	LSB of the "slave offset" field, concatenated with "start offset" field of the logged packet. Type: Status. Reset value: X.	R	0x0000 0000

**Table 13-83. Register Call Summary for Register L3\_HOST\_STDERRLOG\_SLVOFSLSB**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-84. L3\_HOST\_STDERRLOG\_SLVOFSMSB**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b>	See <a href="#">Table 13-60</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STDERRLOG_SLVOFSMSB			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STDERRLOG_SLVOFSMSB	MSB of the "slave offset" field of the logged packet (according to NTP packet format, this register field may exceed the actual "slave offset" size. Unused bits are stuck at 0, if any). Type: Status. Reset value: X.	R	0

**Table 13-85. Register Call Summary for Register L3\_HOST\_STDERRLOG\_SLVOFSMSB**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-86. L3\_HOST\_STDERRLOG\_CUSTOMINFO\_MSTADDR**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .	<b>Instance</b>	See <a href="#">Table 13-60</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STDERRLOG_CUSTOMINFO_MSTADDR			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_CUSTOMINFO_MSTADDR	Type: Status. Reset value: X.	R	0x00

**Table 13-87. Register Call Summary for Register L3\_HOST\_STDERRLOG\_CUSTOMINFO\_MSTADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-88. L3\_HOST\_STDERRLOG\_CUSTOMINFO\_INFO**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	See <a href="#">Table 13-60</a> .
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_CUSTOMINFO_INFO															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_CUSTOMINFO_INFO	Type: Status. Reset value: X.	R	0x00

**Table 13-89. Register Call Summary for Register L3\_HOST\_STDERRLOG\_CUSTOMINFO\_INFO**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-90. L3\_HOST\_STDERRLOG\_CUSTOMINFO\_WR**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	See <a href="#">Table 13-60</a> .
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .		
<b>Description</b>			
<b>Type</b>	R		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_CUSTOMINFO_WR															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STDERRLOG_CUSTOMINFO_WR	Type: Status. Reset value: X.	R	0

**Table 13-91. Register Call Summary for Register L3\_HOST\_STDERRLOG\_CUSTOMINFO\_WR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-92. L3\_HOST\_STDERRLOG\_CUSTOMINFO\_ADDR**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	See <a href="#">Table 13-60</a> .
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											STDERRLOG_CUSTOMINFO_ADDR																				

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20:0	STDERRLOG_CUSTOMINFO_A_DDR	Type: Status. Reset value: X.	R	0x000000

**Table 13-93. Register Call Summary for Register L3\_HOST\_STDERRLOG\_CUSTOMINFO\_ADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-94. L3\_HOST\_STDERRLOG\_CUSTOMINFO\_DECERR**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	See <a href="#">Table 13-60</a> .
<b>Physical Address</b>	See <a href="#">Table 13-61</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STDERRLOG_CUSTOMINFO_DECERR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STDERRLOG_CUSTOMINFO_D ECERR	Type: Status. Reset value: X.	R	0

**Table 13-95. Register Call Summary for Register L3\_HOST\_STDERRLOG\_CUSTOMINFO\_DECERR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

### 13.2.5.1.3 L3 TARG Register Summary and Description

**Table 13-96. TARG Instance Summary**

Module Name	Base Address	Size
CLK1_DMM1_TARG	0x4400 0100	4KB
CLK1_DMM2_TARG	0x4400 0200	4KB
CLK1_ABE_TARG	0x4400 0300	4KB
CLK1_L4CFG_TARG	0x4400 0400	4KB
CLK2_GPMC_TARG	0x4480 0100	4KB
CLK2_OCMRAM_TARG	0x4480 0200	4KB
CLK2_DSS_TARG	0x4480 0300	4KB
CLK2_ISS_TARG	0x4480 0400	4KB
CLK2_CORTEXM3_TARG	0x4480 0500	4KB
CLK2_SGX_TARG	0x4480 0600	4KB
CLK2_IVAHD_TARG	0x4480 0700	4KB
CLK2_SL2_TARG	0x4480 0800	4KB
CLK2_L4PER0_TARG	0x4480 0900	4KB
CLK2_L4PER1_TARG	0x4480 0A00	4KB
CLK2_L4PER2_TARG	0x4480 0B00	4KB
CLK2_L4PER3_TARG	0x4480 0C00	4KB
CLK2_C2C_TARG	0x4480 1600	4KB
CLK3_L4EMU_TARG	0x4500 0100	4KB

#### 13.2.5.1.3.1 L3 TARG Registers Summary

**Table 13-97. TARG Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_DMM1_TARG L3 Physical Address	CLK1_DMM2_TARG L3 Physical Address	CLK1_ABE_TARG L3 Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0100	0x4400 0100	0x4400 0200	0x4400 0300
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0104	0x4400 0104	0x4400 0204	0x4400 0304
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0108	0x4400 0108	0x4400 0208	0x4400 0308
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0110	0x4400 0110	0x4400 0210	0x4400 0310
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0140	0x4400 0140	0x4400 0240	0x4400 0340
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0144	0x4400 0144	0x4400 0244	0x4400 0344
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0148	0x4400 0148	0x4400 0248	0x4400 0348
L3_TARG_STDERRLOG_HDR	R	32	0x0000 014C	0x4400 014C	0x4400 024C	0x4400 034C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0150	0x4400 0150	0x4400 0250	0x4400 0350
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0154	0x4400 0154	0x4400 0254	0x4400 0354
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0158	0x4400 0158	0x4400 0258	0x4400 0358
L3_TARG_STDERRLOG_SLVOFSLB	R	32	0x0000 015C	0x4400 015C	0x4400 025C	0x4400 035C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0160	0x4400 0160	0x4400 0260	0x4400 0360
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0164	0x4400 0164	0x4400 0264	0x4400 0364
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0168	0x4400 0168	0x4400 0268	0x4400 0368
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 016C	0x4400 016C	0x4400 026C	0x4400 036C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0180	0x4400 0180	0x4400 0280	0x4400 0380

**Table 13-98. TARG Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_L4CFG_TARG L3 Physical Address	CLK2_GPMC_TARG L3 Physical Address	CLK2_OCMRAM_TARG L3 Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0100	0x4400 0400	0x4480 0100	0x4480 0200
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0104	0x4400 0404	0x4480 0104	0x4480 0204
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0108	0x4400 0408	0x4480 0108	0x4480 0208
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0110	0x4400 0410	0x4480 0110	0x4480 0210
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0140	0x4400 0440	0x4480 0140	0x4480 0240
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0144	0x4400 0444	0x4480 0144	0x4480 0244
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0148	0x4400 0448	0x4480 0148	0x4480 0248
L3_TARG_STDERRLOG_HDR	R	32	0x0000 014C	0x4400 044C	0x4480 014C	0x4480 024C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0150	0x4400 0450	0x4480 0150	0x4480 0250
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0154	0x4400 0454	0x4480 0154	0x4480 0254
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0158	0x4400 0458	0x4480 0158	0x4480 0258
L3_TARG_STDERRLOG_SLVOFSLB	R	32	0x0000 015C	0x4400 045C	0x4480 015C	0x4480 025C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0160	0x4400 0460	0x4480 0160	0x4480 0260
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0164	0x4400 0464	0x4480 0164	0x4480 0264
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0168	0x4400 0468	0x4480 0168	0x4480 0268
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 016C	0x4400 046C	0x4480 016C	0x4480 026C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0180	0x4400 0480	0x4480 0180	0x4480 0280

**Table 13-99. TARG Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK2_DSS_T ARG L3 Physical Address	CLK2_ISS_T ARG L3 Physical Address	CLK2_CORTEX M3_TARG L3 Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0100	0x4480 0300	0x4480 0400	0x4480 0500
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0104	0x4480 0304	0x4480 0404	0x4480 0504
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0108	0x4480 0308	0x4480 0408	0x4480 0508
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0110	0x4480 0310	0x4480 0410	0x4480 0510
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0140	0x4480 0340	0x4480 0440	0x4480 0540
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0144	0x4480 0344	0x4480 0444	0x4480 0544
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0148	0x4480 0348	0x4480 0448	0x4480 0548
L3_TARG_STDERRLOG_HDR	R	32	0x0000 014C	0x4480 034C	0x4480 044C	0x4480 054C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0150	0x4480 0350	0x4480 0450	0x4480 0550
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0154	0x4480 0354	0x4480 0454	0x4480 0554
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0158	0x4480 0358	0x4480 0458	0x4480 0558
L3_TARG_STDERRLOG_SLVOFSLB	R	32	0x0000 015C	0x4480 035C	0x4480 045C	0x4480 055C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0160	0x4480 0360	0x4480 0460	0x4480 0560
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0164	0x4480 0364	0x4480 0464	0x4480 0564
L3_TARG_STDERRLOG_CUSTOMINFO_MST ADDR	R	32	0x0000 0168	0x4480 0368	0x4480 0468	0x4480 0568
L3_TARG_STDERRLOG_CUSTOMINFO_OPC ODE	R	32	0x0000 016C	0x4480 036C	0x4480 046C	0x4480 056C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0180	0x4480 0380	0x4480 0480	0x4480 0580

**Table 13-100. TARG Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK2_SGX_T ARG L3 Physical Address	CLK2_IVAHD_TARG L3 Physical Address	CLK2_SL2_T ARG L3 Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0100	0x4480 0600	0x4480 0700	0x4480 0800
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0104	0x4480 0604	0x4480 0704	0x4480 0804
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0108	0x4480 0608	0x4480 0708	0x4480 0808
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0110	0x4480 0610	0x4480 0710	0x4480 0810
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0140	0x4480 0640	0x4480 0740	0x4480 0840
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0144	0x4480 0644	0x4480 0744	0x4480 0844
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0148	0x4480 0648	0x4480 0748	0x4480 0848
L3_TARG_STDERRLOG_HDR	R	32	0x0000 014C	0x4480 064C	0x4480 074C	0x4480 084C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0150	0x4480 0650	0x4480 0750	0x4480 0850
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0154	0x4480 0654	0x4480 0754	0x4480 0854
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0158	0x4480 0658	0x4480 0758	0x4480 0858
L3_TARG_STDERRLOG_SLVOFSLB	R	32	0x0000 015C	0x4480 065C	0x4480 075C	0x4480 085C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0160	0x4480 0660	0x4480 0760	0x4480 0860
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0164	0x4480 0664	0x4480 0764	0x4480 0864
L3_TARG_STDERRLOG_CUSTOMINFO_MST ADDR	R	32	0x0000 0168	0x4480 0668	0x4480 0768	0x4480 0868
L3_TARG_STDERRLOG_CUSTOMINFO_OPC ODE	R	32	0x0000 016C	0x4480 066C	0x4480 076C	0x4480 086C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0180	0x4480 0680	0x4480 0780	0x4480 0880

**Table 13-101. TARG Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK2_L4PER 0_TARG L3 Physical Address	CLK2_L4PER 1_TARG L3 Physical Address	CLK2_L4PER 2_TARG L3 Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0100	0x4480 0900	0x4480 0A00	0x4480 0B00
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0104	0x4480 0904	0x4480 0A04	0x4480 0B04
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0108	0x4480 0908	0x4480 0A08	0x4480 0B08
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0110	0x4480 0910	0x4480 0A10	0x4480 0B10
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0140	0x4480 0940	0x4480 0A40	0x4480 0B40
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0144	0x4480 0944	0x4480 0A44	0x4480 0B44
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0148	0x4480 0948	0x4480 0A48	0x4480 0B48
L3_TARG_STDERRLOG_HDR	R	32	0x0000 014C	0x4480 094C	0x4480 0A4C	0x4480 0B4C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0150	0x4480 0950	0x4480 0A50	0x4480 0B50
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0154	0x4480 0954	0x4480 0A54	0x4480 0B54
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0158	0x4480 0958	0x4480 0A58	0x4480 0B58
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 015C	0x4480 095C	0x4480 0A5C	0x4480 0B5C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0160	0x4480 0960	0x4480 0A60	0x4480 0B60
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0164	0x4480 0964	0x4480 0A64	0x4480 0B64
L3_TARG_STDERRLOG_CUSTOMINFO_MST ADDR	R	32	0x0000 0168	0x4480 0968	0x4480 0A68	0x4480 0B68
L3_TARG_STDERRLOG_CUSTOMINFO_OPC ODE	R	32	0x0000 016C	0x4480 096C	0x4480 0A6C	0x4480 0B6C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0180	0x4480 0980	0x4480 0A80	0x4480 0B80

**Table 13-102. TARG Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK2_L4PER3_TARG L3 Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0100	0x4480 0C00
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0104	0x4480 0C04
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0108	0x4480 0C08
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0110	0x4480 0C10
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0140	0x4480 0C40
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0144	0x4480 0C44
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0148	0x4480 0C48
L3_TARG_STDERRLOG_HDR	R	32	0x0000 014C	0x4480 0C4C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0150	0x4480 0C50
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0154	0x4480 0C54
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0158	0x4480 0C58
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 015C	0x4480 0C5C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0160	0x4480 0C60
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0164	0x4480 0C64
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADD R	R	32	0x0000 0168	0x4480 0C68
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 016C	0x4480 0C6C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0180	0x4480 0C80

**Table 13-103. TARG Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK2_C2C_TARG L3 Physical Address	CLK3_L4EMU_TARG L3 Physical Address
L3_TARG_STDHOSHTDR_COREREG	R	32	0x0000 0100	0x4480 1600	0x4500 0100
L3_TARG_STDHOSHTDR_VERSIONREG	R	32	0x0000 0104	0x4480 1604	0x4500 0104
L3_TARG_STDHOSHTDR_MAINCTLREG	RW	32	0x0000 0108	0x4480 1608	0x4500 0108
L3_TARG_STDHOSHTDR_NTTPADDR_0	R	32	0x0000 0110	0x4480 1610	0x4500 0110
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0140	0x4480 1640	0x4500 0140
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0144	0x4480 1644	0x4500 0144
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0148	0x4480 1648	0x4500 0148
L3_TARG_STDERRLOG_HDR	R	32	0x0000 014C	0x4480 164C	0x4500 014C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0150	0x4480 1650	0x4500 0150
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0154	0x4480 1654	0x4500 0154
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0158	0x4480 1658	0x4500 0158
L3_TARG_STDERRLOG_SLVOFSLB	R	32	0x0000 015C	0x4480 165C	0x4500 015C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0160	0x4480 1660	0x4500 0160
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0164	0x4480 1664	0x4500 0164
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0168	0x4480 1668	0x4500 0168
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 016C	0x4480 166C	0x4500 016C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0180	0x4480 1680	0x4500 0180

**13.2.5.1.3.2 L3 TARG Registers Description**

**Table 13-104. L3\_TARG\_STDHOSHTDR\_COREREG**

<b>Address Offset</b>	0x0080 0F00	<b>Instance</b>	See Table 13-96.
<b>Physical Address</b>	See Table 13-97.		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSHTDR_COREREG_CORECODE								RESERVED								STDHOSHTDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x13.	R	0x13
15:1	RESERVED		R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

**Table 13-105. Register Call Summary for Register L3\_TARG\_STDHOSTHDR\_COREREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 13-106. L3\_TARG\_STDHOSTHDR\_VERSIONREG**

Address Offset	0x0080 0F04	Instance	See <a href="#">Table 13-96</a> .
Physical Address	See <a href="#">Table 13-97</a> .		
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

**Table 13-107. Register Call Summary for Register L3\_TARG\_STDHOSTHDR\_VERSIONREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 13-108. L3\_TARG\_STDHOSTHDR\_MAINCTLREG**

<b>Address Offset</b>	0x0100 0108		
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	STDHOSTHDR_MAINCTLREG_CM	STDHOSTHDR_MAINCTLREG_FLT	RESERVED	STDHOSTHDR_MAINCTLREG_EN											

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4	RESERVED	Reserved	R	0
3	STDHOSTHDR_MAINCTLREG_CM	Reserved for internal testing. Must be set to 0. Type: Control. Reset value: 0x0.	R	0
2	STDHOSTHDR_MAINCTLREG_FLT	Asserted when a Fault condition is detected: if the unit includes Error Logging, Flt is asserted when the FltCnt register field indicates a Fault, and deasserted when FltCnt is reset. If no Error Logging is implemented, this bit becomes unit-dependent. In all cases, Flt bit and Flt pin (service network) have the same logical level. Type: Status. Reset value: X.	R	0
1	RESERVED	Reserved	R	0
0	STDHOSTHDR_MAINCTLREG_EN	Sets the global core enable. Note: A disabled master does not generate any NTTP requests, and a disabled slave replies with an error packet to any request it receives. Type: Control. Reset value: 0x1.	RW	1

**Table 13-109. Register Call Summary for Register L3\_TARG\_STDHOSTHDR\_MAINCTLREG**

L3 Interconnect

- [L3 Register Group Summary](#): [\[0\]](#) [\[1\]](#) [\[2\]](#) [\[3\]](#) [\[4\]](#) [\[5\]](#) [\[6\]](#)

**Table 13-110. L3\_TARG\_STDHOSTHDR\_NTTPADDR\_0**

<b>Address Offset</b>	0x0080 0F10		
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Description</b>			
<b>Type</b>	R		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDHOSTHDR_NTPADDR_0															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4:0	STDHOSTHDR_NTPADDR_0	Sets the RX port address. Type: Control. Reset value: 0x15.	R	0x15

**Table 13-111. Register Call Summary for Register L3\_TARG\_STDHOSTHDR\_NTPADDR\_0**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 13-112. L3\_TARG\_STDERRLOG\_SVRTSTDLVL**

<b>Address Offset</b>	0x0080 0F40	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .	<b>Description</b>	See <a href="#">Table 13-96</a> .
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_SVRTSTDLVL_0															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_SVRTSTDLVL_0	Severity level parameters Type: Control. Reset value: 0x2. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x2

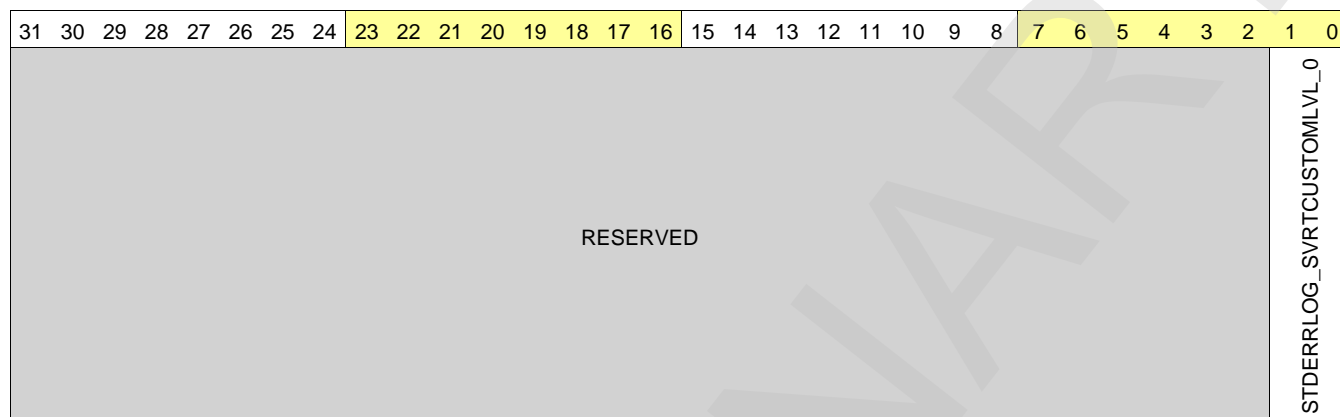
**Table 13-113. Register Call Summary for Register L3\_TARG\_STDERRLOG\_SVRTSTDLVL**

L3 Interconnect

- [L3 Interconnect Error Handling: \[0\]](#)
- [Operational Modes Configuration: \[1\]](#)
- [L3 Register Group Summary: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

**Table 13-114. L3\_TARG\_STDERRLOG\_SVRTCUSTOMLVL**

<b>Address Offset</b>	0x0080 0F44	
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .	<b>Instance</b> See <a href="#">Table 13-96</a> .
<b>Description</b>		
<b>Type</b>	RW	



Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_SVRTCUSTOMLVL_0	Severity level parameters Type: Control. Reset value: 0x2. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x2

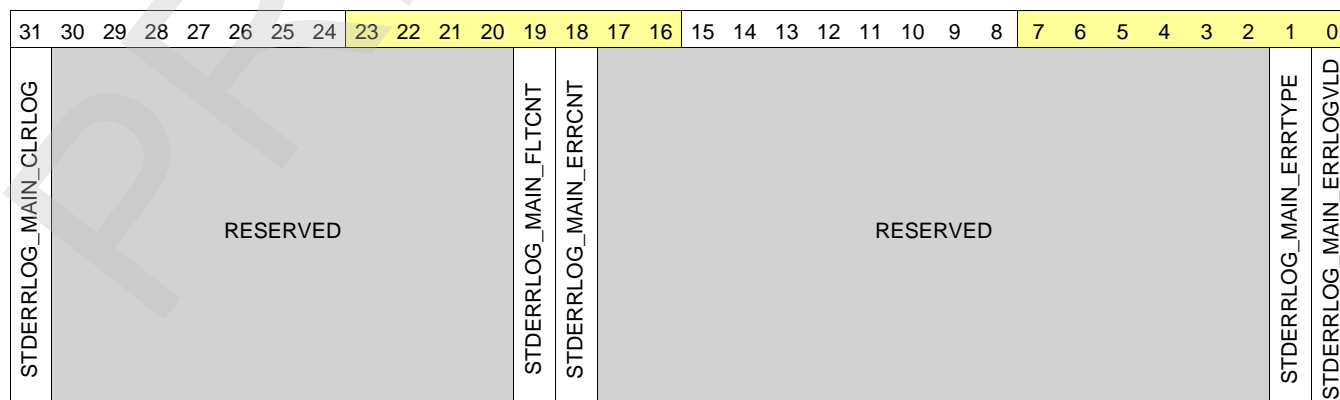
**Table 13-115. Register Call Summary for Register L3\_TARG\_STDERRLOG\_SVRTCUSTOMLVL**

L3 Interconnect

- [L3 Interconnect Error Handling: \[0\]](#)
- [L3 Register Group Summary: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

**Table 13-116. L3\_TARG\_STDERRLOG\_MAIN**

<b>Address Offset</b>	0x0080 0F48	
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .	<b>Instance</b> See <a href="#">Table 13-96</a> .
<b>Description</b>		
<b>Type</b>	RW	



Bits	Field Name	Description	Type	Reset
31	STDERRLOG_MAIN_CLRLOG	Clears "Error Logging Valid" bit when written to 1. Type: Give_AutoCleared. Reset value: 0x0.	RW	0
30:20	RESERVED	Reserved	R	0x000
19	STDERRLOG_MAIN_FLTCNT	Asserted when at least one error with severity level FAULT is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
18	STDERRLOG_MAIN_ERRCNT	Asserted when at least one error with severity level ERROR is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
17:2	RESERVED	Reserved	R	0x0000
1	STDERRLOG_MAIN_ERRTYPE	Indicates logging type. Type: Status. Reset value: X. Read 0x0: Logged Error format is standard (header and necker recorded). Read 0x1: Logged Error format is module dependent. CustomInfo register(s) may be implemented to store additional information.	R	0
0	STDERRLOG_MAIN_ERRLOGV LD	Error Logging Valid. Asserted when logging information is valid(indicates that an error has been logged). Type: Status. Reset value: X.	R	0

**Table 13-117. Register Call Summary for Register L3\_TARG\_STDERRLOG\_MAIN**

L3 Interconnect

- [L3 Interconnect Error Handling: \[0\]](#)
- [Operational Modes Configuration: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [L3 Register Group Summary: \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)

**Table 13-118. L3\_TARG\_STDERRLOG\_HDR**

<b>Address Offset</b>	0x0080 0F4C	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED								STDERRLOG_HDR_LEN1				RESERVED				STDERRLOG_HDR_STOPOFSWRPSZ				STDERRLOG_HDR_ERR		RESERVED				STDERRLOG_HDR_PRESSURE		RESERVED				STDERRLOG_HDR_OPCODE			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:18	STDERRLOG_HDR_LEN1	This field contains the number of payload cell(s) minus one of the logged packet. Type: Status. Reset value: X.	R	0x00
17:16	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
15:12	STDERRLOG_HDR_STOPOFS WRPSZ	StopOfs or WrapSize field of the logged packet (meaning depends on Wrp bit of logged opcode). Type: Status. Reset value: X.	R	0x0
11	STDERRLOG_HDR_ERR	Err bit of the logged packet. Type: Status. Reset value: X.	R	0
10:7	RESERVED	Reserved	R	0x0
6	STDERRLOG_HDR_PRESSUR E	Pressure field of the logged packet. Type: Status. Reset value: X.	R	0
5:4	RESERVED	Reserved	R	0x0
3:0	STDERRLOG_HDR_OPCODE	Opcode of the logged packet. Type: Status. Reset value: X.	R	0x0

**Table 13-119. Register Call Summary for Register L3\_TARG\_STDERRLOG\_HDR**

L3 Interconnect

- [L3 Interconnect Error Handling: \[0\]](#)
- [L3 Register Group Summary: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

**Table 13-120. L3\_TARG\_STDERRLOG\_MSTADDR**

<b>Address Offset</b>	0x0080 0F50	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_MSTADDR															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_MSTADDR	Master Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

**Table 13-121. Register Call Summary for Register L3\_TARG\_STDERRLOG\_MSTADDR**

L3 Interconnect

- [L3 Interconnect Error Handling: \[0\]](#)
- [L3 Register Group Summary: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

**Table 13-122. L3\_TARG\_STDERRLOG\_SLVADDR**

<b>Address Offset</b>	0x0080 0F54	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_SLVADDR															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4:0	STDERRLOG_SLVADDR	Slave Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

**Table 13-123. Register Call Summary for Register L3\_TARG\_STDERRLOG\_SLVADDR**

L3 Interconnect

- [L3 Interconnect Error Handling: \[0\]](#)
- [L3 Register Group Summary: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

**Table 13-124. L3\_TARG\_STDERRLOG\_INFO**

<b>Address Offset</b>	0x0080 0F58	
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .	<b>Instance</b> See <a href="#">Table 13-96</a> .
<b>Description</b>		
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_INFO															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000000
7:0	STDERRLOG_INFO	Info field of the logged packet. Type: Status. Reset value: X.	R	0x00

**Table 13-125. Register Call Summary for Register L3\_TARG\_STDERRLOG\_INFO**

L3 Interconnect

- [L3 Interconnect Error Handling: \[0\]](#)
- [L3 Register Group Summary: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

**Table 13-126. L3\_TARG\_STDERRLOG\_SLVOFSLSB**

<b>Address Offset</b>	0x0080 0F5C	
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .	<b>Instance</b> See <a href="#">Table 13-96</a> .
<b>Description</b>		
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDERRLOG_SLVOFSLSB																															

Bits	Field Name	Description	Type	Reset
31:0	STDERRLOG_SLVOFSLSB	LSB of the "slave offset" field, concatenated with "start offset" field of the logged packet. Type: Status. Reset value: X.	R	0x0000 0000

**Table 13-127. Register Call Summary for Register L3\_TARG\_STDERRLOG\_SLVOFSLSB**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 13-128. L3\_TARG\_STDERRLOG\_SLVOFSMSB**

<b>Address Offset</b>	0x0080 0F60		
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STDERRLOG_SLVOFSMSB			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	STDERRLOG_SLVOFSMSB	MSB of the "slave offset" field of the logged packet (according to NTP packet format, this register field may exceed the actual "slave offset" size. Unused bits are stuck at 0, if any). Type: Status. Reset value: X.	R	0

**Table 13-129. Register Call Summary for Register L3\_TARG\_STDERRLOG\_SLVOFSMSB**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 13-130. L3\_TARG\_STDERRLOG\_CUSTOMINFO\_INFO**

<b>Address Offset</b>	0x0080 0F64		
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STDERRLOG_CUSTOMINFO_INFO			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_CUSTOMINFO_I NFO	Info field of the response packet. Type: Status. Reset value: X.	R	0x00

**Table 13-131. Register Call Summary for Register L3\_TARG\_STDERRLOG\_CUSTOMINFO\_INFO**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 13-132. L3\_TARG\_STDERRLOG\_CUSTOMINFO\_MSTADDR**

<b>Address Offset</b>	0x0080 0F68	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_CUSTOMINFO_MSTADDR															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_CUSTOMINFO_MSTADDR	MstAddr field of the response packet. Type: Status. Reset value: X.	R	0x00

**Table 13-133. Register Call Summary for Register L3\_TARG\_STDERRLOG\_CUSTOMINFO\_MSTADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 13-134. L3\_TARG\_STDERRLOG\_CUSTOMINFO\_OPCODE**

<b>Address Offset</b>	0x0080 0F6C	<b>Instance</b>	See <a href="#">Table 13-96</a> .
<b>Physical Address</b>	See <a href="#">Table 13-97</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												STDERRLOG_CUSTOMINFO_OPCODE			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_CUSTOMINFO_OPCODE	Opcode of the response packet. Type: Status. Reset value: X.	R	0x0

**Table 13-135. Register Call Summary for Register L3\_TARG\_STDERRLOG\_CUSTOMINFO\_OPCODE**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

**Table 13-136. L3\_TARG\_ADDRSPACESIZELOG**

<b>Address Offset</b>	0x0080 0F80	<b>Instance</b>	See <a href="#">Table 13-96.</a>
<b>Physical Address</b>	See <a href="#">Table 13-97.</a>		See <a href="#">Table 13-96.</a>
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED												ADDRSPACESIZELOG			

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4:0	ADDRSPACESIZELOG	The address space size is equal to 2**AddrSpaceSizeLog * 4K in bytes. Type: Control. Reset value: 0x1F.	RW	0x1F

**Table 13-137. Register Call Summary for Register L3\_TARG\_ADDRSPACESIZELOG**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)



13.2.5.1.4 L3 PWR\_DISC Register Summary and Description

Table 13-138. PWR\_DISC Instance Summary

Module Name	Base Address	Size
CLK1_CLK1_TARG_PWR_DISC_CLK2	0x4400 0600	4KB
CLK2_CLK2_TARG_PWR_DISC_CLK1	0x4480 1100	4KB

13.2.5.1.4.1 L3 PWR\_DISC Register Summary

Table 13-139. PWR\_DISC Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_CLK1_TARG_PWR_DISC_CLK2 L3 Physical Address	CLK2_CLK2_TARG_PWR_DISC_CLK1 L3 Physical Address
L3_PWR_DISC_STDHSTHDR_COREREG	R	32	0x0000 0100	0x4400 0600	0x4480 1100
L3_PWR_DISC_STDHSTHDR_VERSIONREG	R	32	0x0000 0104	0x4400 0604	0x4480 1104
L3_PWR_DISC_STDHSTHDR_MAINCTLREG	R	32	0x0000 0108	0x4400 0608	0x4480 1108
L3_PWR_DISC_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0140	0x4400 0640	0x4480 1140
L3_PWR_DISC_STDERRLOG_MAIN	RW	32	0x0000 0148	0x4400 0648	0x4480 1140
L3_PWR_DISC_STDERRLOG_HDR	R	32	0x0000 014C	0x4400 064C	0x4480 114C
L3_PWR_DISC_STDERRLOG_MSTADDR	R	32	0x0000 0150	0x4400 0650	0x4480 1150
L3_PWR_DISC_STDERRLOG_SLVADDR	R	32	0x0000 0154	0x4400 0654	0x4480 1154
L3_PWR_DISC_STDERRLOG_INFO	R	32	0x0000 0158	0x4400 0658	0x4480 1158
L3_PWR_DISC_STDERRLOG_SLVOFSLSB	R	32	0x0000 015C	0x4400 065C	0x4480 115C
L3_PWR_DISC_STDERRLOG_SLVOFSMSB	R	32	0x0000 0160	0x4400 0660	0x4480 1160

13.2.5.1.4.2 L3 PWR\_DISC Register Description

Table 13-140. L3\_PWR\_DISC\_STDHSTHDR\_COREREG

Address Offset	0x0080 1100	Instance	
Physical Address	See Table 13-139.		See Table 13-138.
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	RESERVED	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x38.	R	0x38
15:1	RESERVED	RESERVED	R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

**Table 13-141. Register Call Summary for Register L3\_PWR\_DISC\_STDHOSTHDR\_COREREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-142. L3\_PWR\_DISC\_STDHOSTHDR\_VERSIONREG**

<b>Address Offset</b>	0x0080 1104	<b>Instance</b>	See <a href="#">Table 13-138</a> .
<b>Physical Address</b>	See <a href="#">Table 13-139</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

**Table 13-143. Register Call Summary for Register L3\_PWR\_DISC\_STDHOSTHDR\_VERSIONREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-144. L3\_PWR\_DISC\_STDHOSTHDR\_MAINCTLREG**

<b>Address Offset</b>	0x0080 1108	<b>Instance</b>	See <a href="#">Table 13-138</a> .
<b>Physical Address</b>	See <a href="#">Table 13-139</a> .		See <a href="#">Table 13-138</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		STDHOSTHDR_MAINCTLREG_CM	STDHOSTHDR_MAINCTLREG_FLT	RESERVED											

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4	RESERVED	Reserved	R	0
3	STDHOSTHDR_MAINCTLREG_CM	Reserved for internal testing. Must be set to 0. Type: Control. Reset value: 0x0.	R	0
2	STDHOSTHDR_MAINCTLREG_FLT	Asserted when a Fault condition is detected: if the unit includes Error Logging, Flt is asserted when the FltCnt register field indicates a Fault, and deasserted when FltCnt is reset. If no Error Logging is implemented, this bit becomes unit-dependent. In all cases, Flt bit and Flt pin (service network) have the same logical level. Type: Status. Reset value: X.	R	0
1:0	RESERVED	Reserved	R	0x0

**Table 13-145. Register Call Summary for Register L3\_PWR\_DISC\_STDHOSTHDR\_MAINCTLREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-146. L3\_PWR\_DISC\_STDERRLOG\_SVRTSTDLVL**

<b>Address Offset</b>	0x0080 1140		
<b>Physical Address</b>	See <a href="#">Table 13-139</a> .	<b>Instance</b>	See <a href="#">Table 13-138</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_SVRTSTDLVL_0															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_SVRTSTDLVL_0	Severity level parameters Type: Control. Reset value: 0x2. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x2

**Table 13-147. Register Call Summary for Register L3\_PWR\_DISC\_STDERRLOG\_SVRTSTDLVL**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-148. L3\_PWR\_DISC\_STDERRLOG\_MAIN**

<b>Address Offset</b>	0x0080 1148		
<b>Physical Address</b>	See <a href="#">Table 13-139</a> .	<b>Instance</b>	See <a href="#">Table 13-138</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDERRLOG_MAIN_CLRLOG	RESERVED										STDERRLOG_MAIN_FLTcnt	STDERRLOG_MAIN_ERRcnt	RESERVED										STDERRLOG_MAIN_ERRTYPE	STDERRLOG_MAIN_ERRLOGVLD							

Bits	Field Name	Description	Type	Reset
31	STDERRLOG_MAIN_CLRLOG	Clears "Error Logging Valid" bit when written to 1. Type: Give_AutoCleared. Reset value: 0x0.	RW	0
30:20	RESERVED	Reserved	R	0x000
19	STDERRLOG_MAIN_FLTCNT	Asserted when at least one error with severity level FAULT is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
18	STDERRLOG_MAIN_ERRCNT	Asserted when at least one error with severity level ERROR is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
17:2	RESERVED	Reserved	R	0x0000
1	STDERRLOG_MAIN_ERRTYPE	Indicates logging type. Type: Status. Reset value: X. Read 0x0: Logged Error format is standard (header and necker recorded). Read 0x1: Logged Error format is module dependent. CustomInfo register(s) may be implemented to store additional information.	R	0
0	STDERRLOG_MAIN_ERRLOGV LD	Error Logging Valid. Asserted when logging information is valid(indicates that an error has been logged). Type: Status. Reset value: X.	R	0

**Table 13-149. Register Call Summary for Register L3\_PWR\_DISC\_STDERRLOG\_MAIN**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-150. L3\_PWR\_DISC\_STDERRLOG\_HDR**

<b>Address Offset</b>	0x0080 114C	<b>Instance</b>	See Table 13-138.
<b>Physical Address</b>	See Table 13-139.		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED								STDERRLOG_HDR_LEN1				RESERVED				STDERRLOG_HDR_STOPOFSWRPSZ				STDERRLOG_HDR_ERR		RESERVED				STDERRLOG_HDR_PRESSURE		RESERVED				STDERRLOG_HDR_OPCODE			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:18	STDERRLOG_HDR_LEN1	This field contains the number of payload cell(s) minus one of the logged packet. Type: Status. Reset value: X.	R	0x00
17:16	RESERVED	Reserved	R	0x0
15:12	STDERRLOG_HDR_STOPOFS WRPSZ	StopOfs or WrapSize field of the logged packet (meaning depends on Wrp bit of logged opcode). Type: Status. Reset value: X.	R	0x0
11	STDERRLOG_HDR_ERR	Err bit of the logged packet. Type: Status. Reset value: X.	R	0

Bits	Field Name	Description	Type	Reset
10:7	RESERVED	Reserved	R	0x0
6	STDERRLOG_HDR_PRESSUR E	Pressure field of the logged packet. Type: Status. Reset value: X.	R	0
5:4	RESERVED	Reserved	R	0x0
3:0	STDERRLOG_HDR_OPCODE	Opcode of the logged packet. Type: Status. Reset value: X.	R	0x0

**Table 13-151. Register Call Summary for Register L3\_PWR\_DISC\_STDERRLOG\_HDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-152. L3\_PWR\_DISC\_STDERRLOG\_MSTADDR**

<b>Address Offset</b>	0x0080 1150		
<b>Physical Address</b>	See <a href="#">Table 13-139</a>	<b>Instance</b>	See <a href="#">Table 13-138</a> .
<b>Description</b>	.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_MSTADDR															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000000
7:0	STDERRLOG_MSTADDR	Master Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

**Table 13-153. Register Call Summary for Register L3\_PWR\_DISC\_STDERRLOG\_MSTADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-154. L3\_PWR\_DISC\_STDERRLOG\_SLVADDR**

<b>Address Offset</b>	0x0080 1154		
<b>Physical Address</b>	See <a href="#">Table 13-139</a> .	<b>Instance</b>	See <a href="#">Table 13-138</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_SLVADDR															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4:0	STDERRLOG_SLVADDR	Slave Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

**Table 13-155. Register Call Summary for Register L3\_PWR\_DISC\_STDERRLOG\_SLVADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-156. L3\_PWR\_DISC\_STDERRLOG\_INFO**

<b>Address Offset</b>	0x0080 1158	<b>Instance</b>	See <a href="#">Table 13-138</a> .
<b>Physical Address</b>	See <a href="#">Table 13-139</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STDERRLOG_INFO															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_INFO	Info field of the logged packet. Type: Status. Reset value: X.	R	0x00

**Table 13-157. Register Call Summary for Register L3\_PWR\_DISC\_STDERRLOG\_INFO**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-158. L3\_PWR\_DISC\_STDERRLOG\_SLVOFSLSB**

<b>Address Offset</b>	0x0080 115C	<b>Instance</b>	See <a href="#">Table 13-138</a> .
<b>Physical Address</b>	See <a href="#">Table 13-139</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDERRLOG_SLVOFSLSB																															

Bits	Field Name	Description	Type	Reset
31:0	STDERRLOG_SLVOFSLSB	LSB of the "slave offset" field, concatenated with "start offset" field of the logged packet. Type: Status. Reset value: X.	R	0x0000 0000

**Table 13-159. Register Call Summary for Register L3\_PWR\_DISC\_STDERRLOG\_SLVOFSLSB**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-160. L3\_PWR\_DISC\_STDERRLOG\_SLVOFSMSB**

<b>Address Offset</b>	0x0080 1160		
<b>Physical Address</b>	See <a href="#">Table 13-139</a> .	<b>Instance</b>	See <a href="#">Table 13-138</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STDERRLOG_SLVOFSMSB			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	STDERRLOG_SLVOFSMSB	MSB of the "slave offset" field of the logged packet (according to NTP packet format, this register field may exceed the actual "slave offset" size. Unused bits are stuck at 0, if any). Type: Status. Reset value: X.	R	0

**Table 13-161. Register Call Summary for Register L3\_PWR\_DISC\_STDERRLOG\_SLVOFSMSB**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

### 13.2.5.1.5 L3 FLAGMUX Register Summary and Description

**Table 13-162. FLAGMUX Instance Summary**

Module Name	Base Address	Size
CLK1_FLAGMUX_CLK1	0x4400 0500	4KB
CLK2_FLAGMUX_CLK2	0x4480 1000	4KB
CLK3_FLAGMUX_CLK3	0x4500 0200	4KB

#### 13.2.5.1.5.1 L3 FLAGMUX Register Summary

**Table 13-163. FLAGMUX Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset for CLK1	CLK1_FLAGMUX_CLK1 L3 Physical Address
<a href="#">L3_FLAGMUX_STDHOSTHDR_COREREG</a>	R	32	0x0000 0500	0x4400 0500
<a href="#">L3_FLAGMUX_STDHOSTHDR_VERSIONREG</a>	R	32	0x0000 0504	0x4400 0504
<a href="#">L3_FLAGMUX_MASK0</a>	RW	32	0x0000 0508	0x4400 0508
<a href="#">L3_FLAGMUX_REGERR0</a>	R	32	0x0000 050C	0x4400 050C
<a href="#">L3_FLAGMUX_MASK1</a>	RW	32	0x0000 0510	0x4400 0510
<a href="#">L3_FLAGMUX_REGERR1</a>	RW	32	0x0000 0514	0x4400 0514



**Table 13-164. FLAGMUX Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset for CLK2	CLK2_FLAGMUX_CLK2 L3 Physical Address	Address Offset for CLK3	CLK3_FLAGMUX_CLK3 L3 Physical Address
L3_FLAGMUX_STDHOSTHDR_COREREREG	R	32	0x0080 0100	0x4480 1000	0x0100 0200	0x4500 0200
L3_FLAGMUX_STDHOSTHDR_VE RSIONREG	R	32	0x0080 0104	0x4480 1004	0x0100 0204	0x4500 0204
L3_FLAGMUX_MASK0	RW	32	0x0080 0108	0x4480 1008	0x0100 0208	0x4500 0208
L3_FLAGMUX_REGERR0	R	32	0x0080 010C	0x4480 100C	0x0100 0210	0x4500 020C
L3_FLAGMUX_MASK1	RW	32	0x0080 0110	0x4480 1010	0x0100 0240	0x4500 0210
L3_FLAGMUX_REGERR1	RW	32	0x0080 0114	0x4480 1014	0x0100 0244	0x4500 0214

**13.2.5.1.5.2 L3 FLAGMUX Register Description**

**Table 13-165. L3\_FLAGMUX\_STDHOSTHDR\_COREREREG**

Address Offset	See Table 13-163.		
Physical Address	See Table 13-163.	Instance	See Table 13-162.
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED	Reserved	R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

**Table 13-166. Register Call Summary for Register L3\_FLAGMUX\_STDHOSHTDR\_COREREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\]](#)

**Table 13-167. L3\_FLAGMUX\_STDHOSHTDR\_VERSIONREG**

<b>Address Offset</b>	See <a href="#">Table 13-163</a> .		
<b>Physical Address</b>	See <a href="#">Table 13-163</a> .	<b>Instance</b>	See <a href="#">Table 13-162</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSHTDR_VERSIONREG_REVISIONID								STDHOSHTDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSHTDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSHTDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

**Table 13-168. Register Call Summary for Register L3\_FLAGMUX\_STDHOSHTDR\_VERSIONREG**

L3 Interconnect

- [Flag Mux Generators: \[0\]](#)
- [L3 Register Group Summary: \[1\] \[2\]](#)

**Table 13-169. L3\_FLAGMUX\_MASK0**

<b>Address Offset</b>	See <a href="#">Table 13-163</a> .		
<b>Physical Address</b>	See <a href="#">Table 13-163</a> .	<b>Instance</b>	See <a href="#">Table 13-162</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												MASK0																			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:0	MASK0	Mask flag inputs 0 Type: Control. Reset value: 0x3FFFF.	RW	0x3FFFF

**Table 13-170. Register Call Summary for Register L3\_FLAGMUX\_MASK0**

L3 Interconnect

- [L3 Interconnect Error Handling: \[0\]](#)
- [Operational Modes Configuration: \[1\]](#)
- [L3 Register Group Summary: \[2\] \[3\]](#)

**Table 13-171. L3\_FLAGMUX\_REGERR0**

<b>Address Offset</b>	See <a href="#">Table 13-163</a> .		
<b>Physical Address</b>	See <a href="#">Table 13-163</a> .	<b>Instance</b>	See <a href="#">Table 13-162</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														REGERR0																	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:0	REGERR0	Flag inputs 0 Type: Status. Reset value: X.	R	0x00000

**Table 13-172. Register Call Summary for Register L3\_FLAGMUX\_REGERR0**

L3 Interconnect

- [Flag Mux Generators: \[0\]](#)
- [L3 Interconnect Error Handling: \[1\]](#)
- [Operational Modes Configuration: \[2\] \[3\] \[4\] \[5\]](#)
- [L3 Register Group Summary: \[6\] \[7\]](#)

**Table 13-173. L3\_FLAGMUX\_MASK1**

<b>Address Offset</b>	See <a href="#">Table 13-163</a> .		
<b>Physical Address</b>	See <a href="#">Table 13-163</a> .	<b>Instance</b>	See <a href="#">Table 13-162</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														MASK1																	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:0	MASK1	Mask flag inputs 1 Type: Control. Reset value: 0x3FFFF.	RW	0x3FFFF

**Table 13-174. Register Call Summary for Register L3\_FLAGMUX\_MASK1**

L3 Interconnect

- [L3 Interconnect Error Handling: \[0\]](#)
- [Operational Modes Configuration: \[1\]](#)
- [L3 Register Group Summary: \[2\] \[3\]](#)

**Table 13-175. L3\_FLAGMUX\_REGERR1**

<b>Address Offset</b>	See <a href="#">Table 13-163</a> .	
<b>Physical Address</b>	See <a href="#">Table 13-163</a> .	<b>Instance</b> See <a href="#">Table 13-162</a> .
<b>Description</b>		
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														REGERR1																	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:0	REGERR1	Flag inputs 1 Type: Status. Reset value: X.	R	0x00000

**Table 13-176. Register Call Summary for Register L3\_FLAGMUX\_REGERR1**

L3 Interconnect

- [Flag Mux Generators: \[0\]](#)
- [L3 Interconnect Error Handling: \[1\]](#)
- [Operational Modes Configuration: \[2\] \[3\] \[4\] \[5\]](#)
- [L3 Register Group Summary: \[6\] \[7\]](#)

**13.2.5.1.6 L3 BW Regulator Register Summary and Description****Table 13-177. BW\_REGULATOR Instance Summary**

Module Name	Base Address	Size
CLK1_DSS_BW_REGULATOR	0x4400 0700	256 bytes
CLK2_ISS_BW_REGULATOR	0x4480 1300	256 bytes
CLK2_IVAHD_BW_REGULATOR	0x4480 1400	256 bytes
CLK2_SGX_BW_REGULATOR	0x4480 1500	256 bytes

**13.2.5.1.6.1 L3 BW Regulator Register Summary****Table 13-178. BW\_REGULATOR Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_DSS_BW_REGULATOR L3 Physical Address	CLK2_ISS_BW_REGULATOR L3 Physical Address	CLK2_IVAHD_BW_REGULATOR L3 Physical Address
<a href="#">L3_BW_R_STDHOSTHDR_CORE REG</a>	R	32	0x0000 0100	0x4400 0700	0x4480 1300	0x4480 1400
<a href="#">L3_BW_R_STDHOSTHDR_VERSIONREG</a>	R	32	0x0000 0104	0x4400 0704	0x4480 1304	0x4480 1404
<a href="#">L3_BW_R_BANDWIDTH</a>	RW	32	0x0000 0108	0x4400 0708	0x4480 1308	0x4480 1408
<a href="#">L3_BW_R_WATERMARK</a>	RW	32	0x0000 0110	0x4400 070C	0x4480 130C	0x4480 140C
<a href="#">L3_BW_R_PRESS</a>	R	32	0x0000 0140	0x4400 0710	0x4480 1310	0x4480 1410
<a href="#">L3_BW_R_CLEARHISTORY</a>	RW	32	0x0000 0144	0x4400 0714	0x4480 1314	0x4480 1414

**Table 13-179. BW\_REGULATOR Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK2_SGX_BW_REGULATOR
<a href="#">L3_BW_R_STDHOSTHDR_COREREG</a>	R	32	0x0000 0100	0x4480 1500

**Table 13-179. BW\_REGULATOR Registers Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	CLK2_SGX_BW_REGULATOR
<a href="#">L3_BW_R_STDHOSTHDR_VERSI ONREG</a>	R	32	0x0000 0104	0x4480 1504
<a href="#">L3_BW_R_BANDWIDTH</a>	RW	32	0x0000 0108	0x4480 1508
<a href="#">L3_BW_R_WATERMARK</a>	RW	32	0x0000 0110	0x4480 150C
<a href="#">L3_BW_R_PRESS</a>	R	32	0x0000 0140	0x4480 1510
<a href="#">L3_BW_R_CLEARHISTORY</a>	RW	32	0x0000 0144	0x4480 1514

**13.2.5.1.6.2 L3 BW Regulator Register Description**

**Table 13-180. L3\_BW\_R\_STDHOSTHDR\_COREREG**

<b>Address Offset</b>	0x0080 1300		
<b>Physical Address</b>	See <a href="#">Table 13-178.</a>	<b>Instance</b>	See <a href="#">Table 13-177.</a>
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CO RECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x31.	R	0x31
15:1	RESERVED	Reserved	R	0x0000
0	STDHOSTHDR_COREREG_VE NDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1.  Read 0x0: Third-party vendor. Read 0x1:	R	1

**Table 13-181. Register Call Summary for Register L3\_BW\_R\_STDHOSTHDR\_COREREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\]](#)

**Table 13-182. L3\_BW\_R\_STDHOSTHDR\_VERSIONREG**

<b>Address Offset</b>	0x0080 1304		
<b>Physical Address</b>	See <a href="#">Table 13-178.</a>	<b>Instance</b>	See <a href="#">Table 13-177.</a>
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

**Table 13-183. Register Call Summary for Register L3\_BW\_R\_STDHOSTHDR\_VERSIONREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\] \[1\]](#)

**Table 13-184. L3\_BW\_R\_BANDWIDTH**

<b>Address Offset</b>	0x0080 1308		
<b>Physical Address</b>	See <a href="#">Table 13-178.</a>	<b>Instance</b>	See <a href="#">Table 13-177.</a>
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BANDWIDTH																							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	BANDWIDTH	Bandwidth, in bytes per second. Type: Control. Reset value: 0x0.	RW	0x0000

**Table 13-185. Register Call Summary for Register L3\_BW\_R\_BANDWIDTH**

L3 Interconnect

- [Bandwidth Regulators: \[0\]](#)
- [L3 Register Group Summary: \[1\] \[2\]](#)

**Table 13-186. L3\_BW\_R\_WATERMARK**

<b>Address Offset</b>	0x0080 130C	
<b>Physical Address</b>	See <a href="#">Table 13-178.</a>	<b>Instance</b> See <a href="#">Table 13-177.</a>
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WATERMARK															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11:0	WATERMARK	Peak permissible bandwidth, in bytes. Type: Control. Reset value: 0x1.	RW	0x001

**Table 13-187. Register Call Summary for Register L3\_BW\_R\_WATERMARK**

L3 Interconnect

- [Bandwidth Regulators: \[0\] \[1\]](#)
- [L3 Register Group Summary: \[2\] \[3\]](#)

**Table 13-188. L3\_BW\_R\_PRESS**

<b>Address Offset</b>	0x0080 1310	
<b>Physical Address</b>	See <a href="#">Table 13-178.</a>	<b>Instance</b> See <a href="#">Table 13-177.</a>
<b>Description</b>		
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PRESS_LOW	PRESS_HIGH														

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1	PRESS_LOW	Pressure value inserted if the measured bandwidth is over the watermark. The pressure is bar graph encoded. Type: Control. Reset value: 0x0.	R	0
0	PRESS_HIGH	Pressure value inserted if the measured bandwidth is under the watermark. The pressure is bar graph encoded. Type: Control. Reset value: 0x1.	R	1

**Table 13-189. Register Call Summary for Register L3\_BW\_R\_PRESS**

L3 Interconnect

- [Bandwidth Regulators: \[0\]](#)
- [L3 Register Group Summary: \[1\] \[2\]](#)

**Table 13-190. L3\_BW\_R\_CLEARHISTORY**

<b>Address Offset</b>	0x0080 1314		
<b>Physical Address</b>	See <a href="#">Table 13-178</a> .	<b>Instance</b>	See <a href="#">Table 13-177</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
CLEARHISTORY																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	CLEARHISTORY	Write a 1 clear the traffic counter Type: Give_AutoCleared. Reset value: 0x0.	RW	0

**Table 13-191. Register Call Summary for Register L3\_BW\_R\_CLEARHISTORY**

L3 Interconnect

- [Bandwidth Regulators: \[0\]](#)
- [L3 Register Group Summary: \[1\] \[2\]](#)

**13.2.5.1.7 L3 Rate Adapt Register Summary and Description****Table 13-192. Rate Adapt Instance Summary**

Module Name	Base Address	Size
CLK1_RATE_ADAPT_RESP_32TO128_C LK1	0x4400 0800	4KB
CLK2_RATE_ADAPT_RESP_32TO128_C LK2	0x4480 1200	4KB

**13.2.5.1.7.1 L3 Rate Adapt Register Summary****Table 13-193. Rate Adapt Register Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_RATE_ADAPT_RESP_32 TO128_CLK1 L3 Physical Address	CLK2_RATE_ADAPT_RESP_32 O128_CLK2 L3 Physical Address
<a href="#">L3_RA_STDHOSTHDR_CORE REG</a>	R	32	0x0000 0100	0x4400 0800	0x4480 1200
<a href="#">L3_RA_STDHOSTHDR_VERSI ONREG</a>	R	32	0x0000 0104	0x4400 0804	0x4480 1204
<a href="#">L3_RA_CNF</a>	RW	32	0x0000 0108	0x4400 0808	0x4480 1208

**13.2.5.1.7.2 L3 Rate Adapt Register Description**



**Table 13-194. L3\_RA\_STDHOSTHDR\_COREREG**

<b>Address Offset</b>	0x0080 1200	<b>Instance</b>	See Table 13-192.
<b>Physical Address</b>	See Table 13-193.		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x2D.	R	0x2D
15:1	RESERVED	Reserved	R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

**Table 13-195. Register Call Summary for Register L3\_RA\_STDHOSTHDR\_COREREG**

- L3 Interconnect
- [L3 Register Group Summary: \[0\]](#)

**Table 13-196. L3\_RA\_STDHOSTHDR\_VERSIONREG**

<b>Address Offset</b>	0x0080 1204	<b>Instance</b>	I
<b>Physical Address</b>	See Table 13-193.		See Table 13-192.
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

**Table 13-197. Register Call Summary for Register L3\_RA\_STDHOSTHDR\_VERSIONREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-198. L3\_RA\_CNF**

<b>Address Offset</b>	0x0080 1208	<b>Instance</b>	See <a href="#">Table 13-192</a> .
<b>Physical Address</b>	See <a href="#">Table 13-193</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CNF_RATE		CNF_STANDFWD													

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4:1	CNF_RATE	Indicates the throughput ratio between input and output (Rate = [16 x (Incoming_Throughput/Outgoing_Throuput)] - 1), when bit StAndFwd bit is reset. Ignored when StAndFwd bit is set. Type: Control. Reset value: 0x3.	RW	0x3
0	CNF_STANDFWD	When this bit is set, the Packet Transport Unit stores the entire NTP packet, then forwards it on TX port. Type: Control. Reset value: 0x0.	RW	0

**Table 13-199. Register Call Summary for Register L3\_RA\_CNF**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

### 13.2.5.1.8 L3 STATCOLL Register Summary and Description

**Table 13-200. STATCOLL Instance Summary**

Module Name	Base Address	Size
CLK3_STATCOLL_SDRAM	0x4500 0400	512 bytes
CLK3_STATCOLL_LAT0	0x4500 0600	512 bytes
CLK3_STATCOLL_LAT1	0x4500 0804	512 bytes

### 13.2.5.1.8.1 L3 STATCOLL Register Summary

**Table 13-201. STATCOLL Register Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CLK3_STATCOLL_SDRAM L3 Physical Address	CLK3_STATCOLL_LAT0 L3 Physical Address	CLK3_STATCOLL_LAT1 L3 Physical Address
<a href="#">L3_STCOL_STDHOSTHDR_CORE REG</a>	R	32	0x0100 0400	0x4500 0400	0x4500 0600	0x4500 0800
<a href="#">L3_STCOL_STDHOSTHDR_VERSI ONREG</a>	R	32	0x0100 0404	0x4500 0404	0x4500 0604	0x4500 0804
<a href="#">L3_STCOL_EN</a>	RW	32	0x0100 0408	0x4500 0408	0x4500 0608	0x4500 0808
<a href="#">L3_STCOL_SOFTEN</a>	RW	32	0x0100 040C	0x4500 040C	0x4500 060C	0x4500 080C
<a href="#">L3_STCOL_TRIGEN</a>	RW	32	0x0100 0410	0x4500 0410	0x4500 0610	0x4500 0810
<a href="#">L3_STCOL_REQEVT</a>	RW	32	0x0100 0414	0x4500 0414	0x4500 0614	0x4500 0814
<a href="#">L3_STCOL_RSPEVT</a>	RW	32	0x0100 0418	0x4500 0418	0x4500 0618	0x4500 0818
<a href="#">L3_STCOL_EVTMUX_SELO</a>	RW	32	0x0100 041C	0x4500 041C	0x4500 061C	0x4500 081C
<a href="#">L3_STCOL_EVTMUX_SEL1</a>	RW	32	0x0100 0420	0x4500 0420	0x4500 0620	0x4500 0820
<a href="#">L3_STCOL_EVTMUX_SEL2</a>	RW	32	0x0100 0424	0x4500 0424	0x4500 0624	0x4500 0824
<a href="#">L3_STCOL_EVTMUX_SEL3</a>	RW	32	0x0100 0428	0x4500 0428	0x4500 0628	0x4500 0828
<a href="#">L3_STCOL_EVTMUX_SEL4</a>	RW	32	0x0100 042C	0x4500 042C	N/A	N/A
<a href="#">L3_STCOL_DUMP_IDENTIFIER</a>	R	32	0x0100 0430	0x4500 0430	0x4500 062C	0x4500 082C
<a href="#">L3_STCOL_DUMP_COLLECTTIME</a>	RW	32	0x0100 0434	0x4500 0434	0x4500 0630	0x4500 0830
<a href="#">L3_STCOL_DUMP_SLVADDR</a>	R	32	0x0100 0438	0x4500 0438	0x4500 0634	0x4500 834
<a href="#">L3_STCOL_DUMP_MSTADDR</a>	R	32	0x0100 043C	0x4500 043C	0x4500 0638	0x4500 0838
<a href="#">L3_STCOL_DUMP_SLVOFS</a>	RW	32	0x0100 0440	0x4500 0440	0x4500 063C	0x4500 083C
<a href="#">L3_STCOL_DUMP_MANUAL</a>	RW	32	0x0100 0444	0x4500 0444	0x4500 0640	0x4500 0840
<a href="#">L3_STCOL_DUMP_SEND</a>	RW	32	0x0100 0448	0x4500 0448	0x4500 0644	0x4500 0844
<a href="#">L3_STCOL_FILTER_i_GLOBALEN</a>	RW	32	0x0100 044C	0x4500 044C	0x4500 0648	0x4500 0848
<a href="#">L3_STCOL_FILTER_i_EN_k</a>	RW	32	0x0100 0450	0x4500 0450	0x4500 064C	0x4500 084C
<a href="#">L3_STCOL_FILTER_i_MASK_m_M STADDR</a>	RW	32	0x0100 0454	0x4500 0454	0x4500 0650	0x4500 0850
<a href="#">L3_STCOL_FILTER_i_MASK_m_R D</a>	RW	32	0x0100 0458	0x4500 0458	0x4500 065C	0x4500 085C
<a href="#">L3_STCOL_FILTER_i_MASK_m_W R</a>	RW	32	0x0100 045C	0x4500 045C	0x4500 0660	0x4500 0860
<a href="#">L3_STCOL_FILTER_i_MASK_m_E RR</a>	RW	32	0x0100 0460	0x4500 0460	0x4500 0664	0x4500 0864
<a href="#">L3_STCOL_FILTER_i_MASK_m_U SERINFO</a>	RW	32	0x0100 0468	0x4500 0468	N/A	N/A

**Table 13-201. STATCOLL Register Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	CLK3_STATCOLL_SDRAM L3 Physical Address	CLK3_STATCOLL_LAT0 L3 Physical Address	CLK3_STATCOLL_LAT1 L3 Physical Address
L3_STCOL_FILTER_i_MATCH_m_MSTADDR	RW	32	0x0100 046C	0x4500 046C	0x0100 066C	0x4500 086C
L3_STCOL_FILTER_i_MATCH_m_RD	RW	32	0x0100 0470	0x4500 0470	0x0100 0678	0x4500 0878
L3_STCOL_FILTER_i_MATCH_m_WR	RW	32	0x0100 0474	0x4500 0474	0x0100 067C	0x4500 087C
L3_STCOL_FILTER_i_MATCH_m_ERR	RW	32	0x0100 0478	0x4500 0478	0x0100 0680	0x4500 880
L3_STCOL_FILTER_i_MATCH_m_USERINFO	RW	32	0x0100 0480	0x4500 0480	N/A	N/A
L3_STCOL_OP_i_THRESHOLD_M_INVAL	RW	32	0x0100 04B8	0x4500 04B8	0x0100 0688	0x4500 0880
L3_STCOL_OP_i_THRESHOLD_M_AXVAL	RW	32	0x0100 04BC	0x4500 04BC	0x0100 068C	0x4500 088C
L3_STCOL_OP_i_EVTINFOSEL	RW	32	0x0100 04C0	0x4500 04C0	0x0100 0690	0x4500 0890
L3_STCOL_OP_i_SEL	RW	32	0x0100 04C4	0x4500 04C4	0x0100 0694	0x4500 0894

**13.2.5.1.8.2 L3 STATCOLL Register Description****Table 13-202. L3\_STCOL\_STDHOSTHDR\_COREREG**

<b>Address Offset</b>	0x0100 0400		
<b>Physical Address</b>	See Table 13-201.	<b>Instance</b>	See Table 13-200.
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x3A.	R	0x3A
15:1	RESERVED	Reserved	R	0x0000

Bits	Field Name	Description	Type	Reset
0	STDHOSTHDR_COREREG_VE NDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1.  Read 0x0: Third-party vendor. Read 0x1:	R	1

**Table 13-203. Register Call Summary for Register L3\_STCOL\_STDHOSTHDR\_COREREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-204. L3\_STCOL\_STDHOSTHDR\_VERSIONREG**

<b>Address Offset</b>	0x0100 0404	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .		
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_ REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_ COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

**Table 13-205. Register Call Summary for Register L3\_STCOL\_STDHOSTHDR\_VERSIONREG**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-206. L3\_STCOL\_EN**

<b>Address Offset</b>	0x0100 0408																																																																																														
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .															<b>Instance</b>								See <a href="#">Table 13-200</a> .																																																																							
<b>Description</b>																																																																																															
<b>Type</b>	RW																																																																																														
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> <td rowspan="2" style="writing-mode: vertical-rl; text-orientation: mixed;">EN</td> </tr> <tr> <td colspan="31">RESERVED</td> </tr> </table>																																31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	EN	RESERVED																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	EN																																																															
RESERVED																																																																																															
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>		<b>Type</b>	<b>Reset</b>																																																																																										
31:1	RESERVED	Reserved		R	0x0000 0000																																																																																										
0	EN	Enable performance monitoring, this will also shut down the clock if En = 0 Type: Control. Reset value: 0x0.		RW	0																																																																																										

**Table 13-207. Register Call Summary for Register L3\_STCOL\_EN**

L3 Interconnect

- [Statistic Collectors Group: \[0\]](#)
- [L3 Register Group Summary: \[1\]](#)

**Table 13-208. L3\_STCOL\_SOFTEN**

<b>Address Offset</b>	0x0100 040C																																																																																														
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .															<b>Instance</b>								See <a href="#">Table 13-200</a> .																																																																							
<b>Description</b>																																																																																															
<b>Type</b>	RW																																																																																														
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> <td rowspan="2" style="writing-mode: vertical-rl; text-orientation: mixed;">SOFTEN</td> </tr> <tr> <td colspan="31">RESERVED</td> </tr> </table>																																31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SOFTEN	RESERVED																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SOFTEN																																																															
RESERVED																																																																																															
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>		<b>Type</b>	<b>Reset</b>																																																																																										
31:1	RESERVED	Reserved		R	0x0000 0000																																																																																										
0	SOFTEN	Software enable for performance monitoring Type: Control. Reset value: 0x0.		RW	0																																																																																										

**Table 13-209. Register Call Summary for Register L3\_STCOL\_SOFTEN**

L3 Interconnect

- [Statistic Collectors Group: \[0\]](#)
- [L3 Register Group Summary: \[1\]](#)

**Table 13-210. L3\_STCOL\_TRIGEN**

<b>Address Offset</b>	0x0100 0410		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												TRIGEN			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	TRIGEN	TrigEn when set, it enable the external trigger start and stop Type: Control. Reset value: 0x0.	RW	0

**Table 13-211. Register Call Summary for Register L3\_STCOL\_TRIGEN**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-212. L3\_STCOL\_REQEVT**

<b>Address Offset</b>	0x0100 0414		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										REQEVT					

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:0	REQEVT	Req event select Type: Control. Reset value: 0x0. 0x0: Collect is disabled default value 0x1: Collect all event: hit always (cycle) 0x2: Collect transfers: actually used cycle for transferring aN NTTP word 0x3: Collect wait cycle: transfer has been delayed by source 0x4: Collect busy: transfer has been delayed by destination 0x5: Collect packet: new packet start 0x6: Collect data: data cycle transfer, write for requests, read for responses 0x7: Collect idles: transfer is not initiated by source 0x8: Collect latency: hit when actually detecting debug bit on response links	RW	0x0

**Table 13-213. Register Call Summary for Register L3\_STCOL\_REQEVT**

L3 Interconnect

- [Statistic Collectors Group: \[0\]](#)
- [L3 Register Group Summary: \[1\]](#)

**Table 13-214. L3\_STCOL\_RSPEVT**

<b>Address Offset</b>	0x0100 0418		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RSPEVT								

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3:0	RSPEVT	Rsp event select Type: Control. Reset value: 0x0. 0x0: Collect is disabled default value 0x1: Collect all event: hit always (cycle) 0x2: Collect transfers: actually used cycle for transferring a NTPP word 0x3: Collect wait cycle: transfer has been delayed by source 0x4: Collect busy: transfer has been delayed by destination 0x5: Collect packet: new packet start 0x6: Collect data: data cycle transfer, write for requests, read for responses 0x7: Collect idles: transfer is not initiated by source 0x8: Collect latency: hit when actually detecting debug bit on response links	RW	0x0

**Table 13-215. Register Call Summary for Register L3\_STCOL\_RSPEVT**

L3 Interconnect

- [Statistic Collectors Group: \[0\]](#)
- [L3 Register Group Summary: \[1\]](#)

**Table 13-216. L3\_STCOL\_EVTMUX\_SELO**

<b>Address Offset</b>	0x0100 041C		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							EVTMUX_SELO								

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SELO	The select of the mux 0 Type: Control. Reset value: 0x0.	RW	0x0



**Table 13-217. Register Call Summary for Register L3\_STCOL\_EVTMUX\_SEL0**

L3 Interconnect

- [Statistic Collectors Group: \[0\]](#)
- [L3 Register Group Summary: \[1\]](#)

**Table 13-218. L3\_STCOL\_EVTMUX\_SEL1**

<b>Address Offset</b>	0x0100 0420	<b>Instance</b>	
<b>Physical Address</b>	See <a href="#">Table 13-201.</a>		See <a href="#">Table 13-200.</a>
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVTMUX_SEL1															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL1	The select of the mux 1 Type: Control. Reset value: 0x0.	RW	0x0

**Table 13-219. Register Call Summary for Register L3\_STCOL\_EVTMUX\_SEL1**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-220. L3\_STCOL\_EVTMUX\_SEL2**

<b>Address Offset</b>	0x0100 0424	<b>Instance</b>	
<b>Physical Address</b>	See <a href="#">Table 13-201.</a>		See <a href="#">Table 13-200.</a>
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVTMUX_SEL2															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL2	The select of the mux 2 Type: Control. Reset value: 0x0.	RW	0x0

**Table 13-221. Register Call Summary for Register L3\_STCOL\_EVTMUX\_SEL2**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-222. L3\_STCOL\_EVTMUX\_SEL3**

<b>Address Offset</b>	0x0100 0428		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVTMUX_SEL3															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL3	The select of the mux 3 Type: Control. Reset value: 0x0.	RW	0x0

**Table 13-223. Register Call Summary for Register L3\_STCOL\_EVTMUX\_SEL3**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-224. L3\_STCOL\_EVTMUX\_SEL4**

<b>Address Offset</b>	0x0100 042C		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVTMUX_SEL4															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL4	The select of the mux 4 Type: Control. Reset value: 0x0.	RW	0x0

**Table 13-225. Register Call Summary for Register L3\_STCOL\_EVTMUX\_SEL4**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-226. L3\_STCOL\_DUMP\_IDENTIFIER**

<b>Address Offset</b>	0x0100 0430		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DUMP_IDENTIFIER															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	DUMP_IDENTIFIER	Probe identifier Type: Control. Reset value: 0x0.	R	0x0

**Table 13-227. Register Call Summary for Register L3\_STCOL\_DUMP\_IDENTIFIER**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-228. L3\_STCOL\_DUMP\_COLLECTTIME**

<b>Address Offset</b>	0x0100 0434		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DUMP_COLLECTTIME															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	DUMP_COLLECTTIME	Number of cycle to wait between two statistics frame Type: Control. Reset value: 0x0.	RW	0x0000

**Table 13-229. Register Call Summary for Register L3\_STCOL\_DUMP\_COLLECTTIME**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-230. L3\_STCOL\_DUMP\_SLVADDR**

<b>Address Offset</b>	0x0100 0438		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DUMP_SLVADDR															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4:0	DUMP_SLVADDR	Dump slave address Type: Control. Reset value: 0x19.	R	0x19

**Table 13-231. Register Call Summary for Register L3\_STCOL\_DUMP\_SLVADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-232. L3\_STCOL\_DUMP\_MSTADDR**

<b>Address Offset</b>	0x0100 043C		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DUMP_MSTADDR															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000000
7:0	DUMP_MSTADDR	Dump master address Type: Control. Reset value: 0xE0.	R	0xE0

**Table 13-233. Register Call Summary for Register L3\_STCOL\_DUMP\_MSTADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-234. L3\_STCOL\_DUMP\_SLVOFS**

<b>Address Offset</b>	0x0100 0440		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_SLVOFS																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_SLVOFS	Dump slave offset Type: Control. Reset value: 0x800.	RW	0x0000 0800

**Table 13-235. Register Call Summary for Register L3\_STCOL\_DUMP\_SLVOFS**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-236. L3\_STCOL\_DUMP\_MANUAL**

<b>Address Offset</b>	0x0100 0444	<b>Instance</b>	
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																																DUMP_MANUAL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	DUMP_MANUAL	Define the dump mode: if != 0 the dump is controlled by the Send register. Type: Control. Reset value: 0x0.	RW	0

**Table 13-237. Register Call Summary for Register L3\_STCOL\_DUMP\_MANUAL**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-238. L3\_STCOL\_DUMP\_SEND**

<b>Address Offset</b>	0x0100 0448	<b>Instance</b>	
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																																DUMP_SEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	DUMP_SEND	In manual mode, is used to send the dump content and initialize the counters. Type: Give_AutoCleared. Reset value: 0x0.	RW	0

**Table 13-239. Register Call Summary for Register L3\_STCOL\_DUMP\_SEND**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-240. L3\_STCOL\_FILTER\_i\_GLOBALEN**

<b>Address Offset</b>	0x0100 044C		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																																FILTER_i_GLOBALEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_GLOBALEN	Filter global enable Type: Control. Reset value: 0x0.	RW	0

**Table 13-241. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_GLOBALEN**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-242. L3\_STCOL\_FILTER\_i\_EN\_k**

<b>Address Offset</b>	0x0100 0450		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																																FILTER_i_EN0

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_EN0	Enable filter stage 0 Type: Control. Reset value: 0x0.	RW	0

**Table 13-243. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_EN\_k**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-244. L3\_STCOL\_FILTER\_i\_MASK\_m\_MSTADDR**

<b>Address Offset</b>	0x0100 0454			
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b> See <a href="#">Table 13-200</a> .		
<b>Description</b>	RW			
<b>Type</b>	RW			
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
RESERVED		FILTER_i_MASK_m_MSTADDR		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
31:8	RESERVED	Reserved	R	0x000000
7:0	FILTER_i_MASK_m_MSTADDR	Mask/Match of MstAddr Type: Control. Reset value: 0x0.	RW	0x00

**Table 13-245. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MASK\_m\_MSTADDR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-246. L3\_STCOL\_FILTER\_i\_MASK\_m\_RD**

<b>Address Offset</b>	0x0100 0458			
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b> See <a href="#">Table 13-200</a> .		
<b>Description</b>	RW			
<b>Type</b>	RW			
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0		
RESERVED		FILTER_i_MASK_m_RD		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MASK_m_RD	Mask/Match of Rd Type: Control. Reset value: 0x0.	RW	0

**Table 13-247. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MASK\_m\_RD**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-248. L3\_STCOL\_FILTER\_i\_MASK\_m\_WR**

<b>Address Offset</b>	0x0100 045C		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
FILTER_i_MASK_m_WR																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MASK_m_WR	Mask/Match of Wr Type: Control. Reset value: 0x0.	RW	0

**Table 13-249. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MASK\_m\_WR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-250. L3\_STCOL\_FILTER\_i\_MASK\_m\_ERR**

<b>Address Offset</b>	0x0100 0460		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
FILTER_i_MASK_m_ERR																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MASK_m_ERR	Mask/Match of Err Type: Control. Reset value: 0x0.	RW	0

**Table 13-251. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MASK\_m\_ERR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)



**Table 13-252. L3\_STCOL\_FILTER\_i\_MASK\_m\_USERINFO**

<b>Address Offset</b>	0x0100 0468	
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b> See <a href="#">Table 13-200</a> .
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														FILTER_i_MASK_m_USERINFO																	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:0	FILTER_i_MASK_m_USERINFO	Mask/Match of UserInfo Type: Control. Reset value: 0x0.	RW	0x00000

**Table 13-253. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MASK\_m\_USERINFO**

- L3 Interconnect
- [L3 Register Group Summary: \[0\]](#)

**Table 13-254. L3\_STCOL\_FILTER\_i\_MATCH\_m\_MSTADDR**

<b>Address Offset</b>	0x0100 046C	
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b> See <a href="#">Table 13-200</a> .
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							FILTER_i_MATCH_m_MSTADDR								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	FILTER_i_MATCH_m_MSTADD R	Mask/Match of MstAddr Type: Control. Reset value: 0x0.	RW	0x00

**Table 13-255. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MATCH\_m\_MSTADDR**

- L3 Interconnect
- [L3 Register Group Summary: \[0\]](#)

**Table 13-256. L3\_STCOL\_FILTER\_i\_MATCH\_m\_RD**

<b>Address Offset</b>	0x0100 0470		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												FILTER_i_MATCH_m_RD			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MATCH_m_RD	Mask/Match of Rd Type: Control. Reset value: 0x0.	RW	0

**Table 13-257. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MATCH\_m\_RD**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-258. L3\_STCOL\_FILTER\_i\_MATCH\_m\_WR**

<b>Address Offset</b>	0x0100 0474		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												FILTER_i_MATCH_m_WR			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MATCH_m_WR	Mask/Match of Wr Type: Control. Reset value: 0x0.	RW	0

**Table 13-259. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MATCH\_m\_WR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-260. L3\_STCOL\_FILTER\_i\_MATCH\_m\_ERR**

<b>Address Offset</b>	0x0100 0478		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FILTER_i_MATCH_m_ERR															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MATCH_m_ERR	Mask/Match of Err Type: Control. Reset value: 0x0.	RW	0

**Table 13-261. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MATCH\_m\_ERR**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-262. L3\_STCOL\_FILTER\_i\_MATCH\_m\_USERINFO**

<b>Address Offset</b>	0x0100 0480		
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												FILTER_i_MATCH_m_USERINFO																			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:0	FILTER_i_MATCH_m_USERINFO	Mask/Match of UserInfo Type: Control. Reset value: 0x0.	RW	0x00000

**Table 13-263. Register Call Summary for Register L3\_STCOL\_FILTER\_i\_MATCH\_m\_USERINFO**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-264. L3\_STCOL\_OP\_i\_THRESHOLD\_MINVAL**

<b>Address Offset</b>	0x0100 04B8	
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b> See <a href="#">Table 13-200</a> .
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OP_i_THRESHOLD_MINVAL															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11:0	OP_i_THRESHOLD_MINVAL	Min value Type: Control. Reset value: 0x0.	RW	0x000

**Table 13-265. Register Call Summary for Register L3\_STCOL\_OP\_i\_THRESHOLD\_MINVAL**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-266. L3\_STCOL\_OP\_i\_THRESHOLD\_MAXVAL**

<b>Address Offset</b>	0x0100 04BC	
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b> See <a href="#">Table 13-200</a> .
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OP_i_THRESHOLD_MAXVAL															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11:0	OP_i_THRESHOLD_MAXVAL	Max value Type: Control. Reset value: 0x0.	RW	0x000

**Table 13-267. Register Call Summary for Register L3\_STCOL\_OP\_i\_THRESHOLD\_MAXVAL**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-268. L3\_STCOL\_OP\_i\_EVTINFOSEL**

<b>Address Offset</b>	0x0100 04C0	
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .	<b>Instance</b> See <a href="#">Table 13-200</a> .
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OP_i_EVTINFOSEL															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	OP_i_EVTINFOSEL	Select event info data to add to counter (len/press or latency) Type: Control. Reset value: 0x0.  0x0: Select len from event info list 0x1: Select pressure if available from event info list 0x2: Select latency if available from event info list	RW	0x0

**Table 13-269. Register Call Summary for Register L3\_STCOL\_OP\_i\_EVTINFOSEL**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

**Table 13-270. L3\_STCOL\_OP\_i\_SEL**

<b>Address Offset</b>	0x0100 04C4	<b>Instance</b>	See <a href="#">Table 13-200</a> .
<b>Physical Address</b>	See <a href="#">Table 13-201</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OP_i_SEL															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3:0	OP_i_SEL	Select logical operation Type: Control. Reset value: 0x0.  0x0: Increment counter on each mask/match filter hit 0x1: Increment counter on each min/max level hit 0x2: Add to counter the selected event info value (len/press or latency) 0x3: increment counter when all filter event hits (And(Fi)) 0x4: Increment counter if any of filter event hits (Or(Fi)) 0x5: Add to counter the number of current request event that hit 0x6: Add to counter the number of current response event that hit 0x7: Add to counter the number of all event that hit 0x8: Increment counter on each selected external event hit	RW	0x0

**Table 13-271. Register Call Summary for Register L3\_STCOL\_OP\_i\_SEL**

L3 Interconnect

- [L3 Register Group Summary: \[0\]](#)

## 13.3 L4 Interconnects

This section details the device L4 interconnects.

### 13.3.1 L4 Interconnect Overview

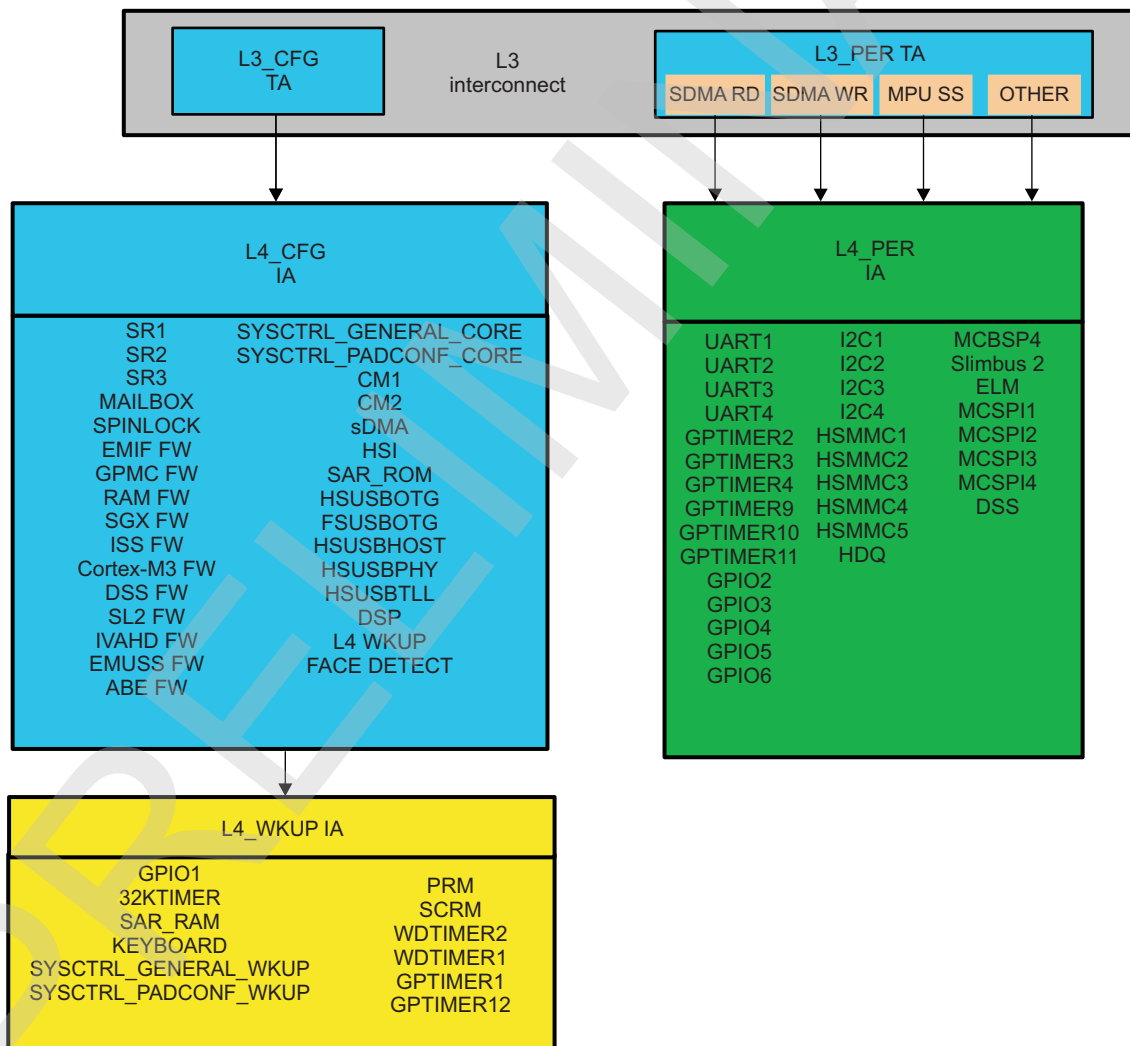
The device uses three separate L4 interconnect structures to connect peripheral modules. All L4s handle transfers with peripherals but are located in distinct power domains.

Figure 13-8 is an overview of the L4 interconnects and the peripherals attached to them.

The L4 interconnect is composed of the following interconnects:

- L4\_CFG: Includes the majority of the configuration interface for L3 system modules and peripheral interconnect
- L4\_PER: Includes the main peripherals that require sDMA access
- L4\_WKUP: Includes peripherals attached to the WKUP power domain

**Figure 13-8. L4 Interconnect Overview**



icnt-013

The main features of the L4 interconnects are:

- From one to four 32-bit initiator ports for each L4 interconnect instance.
- 8-, 16-, or 32-bit data, single, or burst transactions
- Little endian

- Nonblocking architecture with fair arbitration between threads
- Target interfaces: Fully synchronous or divided synchronous
- L4\_CFG and L4\_PER frequency equal half of L3 frequency
- Protection logic that provides user-configurable access control to targets by each initiator
- Implementation of RFFs to support low-voltage dynamic power switching

### 13.3.2 L4 Interconnect Integration

Table 13-272 and Table 13-273 summarize the integration of the module in the device.

**Table 13-272. Integration Attributes**

Module Instance	Attributes
	Power Domain
L4_PER	PD_L4_PER
L4_CFG	PD_CORE
L4_WKUP	PD_WKUP

**Table 13-273. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
L4_PER	L4_PER_CLK	PER_L4_ICLK	PRCM module	Functional and interface clock
L4_CFG	L4_CFG_CLK	CFG_L4_ICLK	PRCM module	Functional and interface clock
L4_WKUP	L4_WKUP_CLK	WKUP_L4_ICLK 2	PRCM module	Functional and interface clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
L4_PER	L4_PER_RST	L4_PER_RST	PRCM module	Reset of L4_PER interconnect
	L4_PER_RET_RST	L4_PER_RET_ RST	PRCM module	Reset of L4_PER interconnect retention registers. For information about retention reset, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
L4_CFG	L4_CFG_RST	CORE_RST	PRCM module	Reset of L4_CFG interconnect
	L4_CFG_RET_RST	CORE_RET_RS T	PRCM module	Reset of L4_CFG interconnect retention registers. For information about retention reset, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
L4_WKUP	L4_WKUP_RST	WKUP_RST	PRCM module	Reset of L4_WKUP interconnect

### 13.3.3 L4 Interconnect Functional Description

#### 13.3.3.1 Module Distribution

IAs and TAs provide the interface to connect the different modules to their associated interconnect.

Table 13-274 through Table 13-279 list all the modules and subsystems with their associated agents. The agents are listed for each L4 interconnect domain.

##### 13.3.3.1.1 L4\_PER Interconnect Agents

The L4\_PER interconnect handles transfers only to peripherals in the PER power domain. Table 13-274 lists the PER TAs.

**Table 13-274. L4 PER TAs**

Module Target Name	Description
PER_TA_UART1	UART port 1 module
PER_TA_UART2	UART port 2 module
PER_TA_UART3	UARTport 3 module
PER_TA_UART4	UART port 4 module
PER_TA_GPTIMER2	GPTIMER2 module
PER_TA_GPTIMER3	GPTIMER3 module
PER_TA_GPTIMER4	GPTIMER4 module
PER_TA_GPTIMER9	GPTIMER9 module
PER_TA_GPTIMER10	GPTIMER10 module
PER_TA_GPTIMER11	GPTIMER11 module
PER_TA_DSS	Display subsystem configuration module
PER_TA_GPIO2	General-purpose input/output 2 (GPIO2) module
PER_TA_GPIO3	GPIO3 module
PER_TA_GPIO4	GPIO4 module
PER_TA_GPIO5	GPIO5 module
PER_TA_GPIO6	GPIO6 module
PER_TA_I2C1	I2C1 module
PER_TA_I2C2	Multimaster I2C2 module
PER_TA_I2C3	Multimaster I2C3 module
PER_TA_I2C4	Multimaster I2C4 module
PER_TA_SLIMBUS2	Silmbus 2 module
PER_TA_ELM	Error location module
PER_TA_MCBSP4	McBSP4 module
PER_TA_MCSP11	McSPI1 module
PER_TA_MCSP12	MCSP12 module
PER_TA_MCSP13	MCSP13 module
PER_TA_MCSP14	MCSP14 module
PER_TA_HSMMC1	MMC controller 1 module
PER_TA_HSMMC2	MMC controller 2 module
PER_TA_HSMMC3	MMC controller 3 module
PER_TA_HSMMC4	MMC controller 4 module
PER_TA_HSMMC5	MMC controller 5 module
PER_TA_HDQ	Single-wire serial link low-rate module



Four ports communicate between the L3 interconnect and the L4\_PER interconnect to allow the L3 initiators to access the L4\_PER targets. [Table 13-275](#) lists the L4\_PER initiator TAs.

For the list of initiators authorized to access the L4 peripheral peripherals, see [Section 13.2.3.2.2, Connectivity Matrix](#).

**Table 13-275. L4\_PER IAs**

Module Initiator Name	Description
PER_IA_0	L3 system direct memory access (sDMA) RD interconnect port
PER_IA_1	L3 sDMA WR interconnect port
PER_IA_2	L3 MPU subsystem interconnect port
PER_IA_3	L3 others interconnect port

### 13.3.3.1.2 L4\_CFG Interconnect Agents

The L4\_CFG interconnect handles only transfers to peripherals in the core power domain. [Table 13-276](#) lists the TAs.

**Table 13-276. L4\_CFG TAs**

Module Target Name	Description
CFG_TA_SYSCTRL_GENERAL_CORE	System control general core module
CFG_TA_SYSCTRL_PADCONF_CORE	System control padconf core module
CFG_TA_CM1	Clock manager 1 module
CFG_TA_CM2	Clock manager 1 module
CFG_TA_SDMA	sDMA module
CFG_TA_HSI	HS SSI module
CFG_TA_SAR_ROM	Save-and-restore ROM
CFG_TA_HSUSBOTG	HS USB OTG module
CFG_TA_FSUSBHOST	FS USB OTG module
CFG_TA_HSUSBHOST	HS USB host module
CFG_TA_HSUSBPHY	HS USB PHY module
CFG_TA_HSUSBTLL	USB TTL module
CFG_TA_DSP	DSP subsystem
CFG_TA_SR1	SmartReflex1 module
CFG_TA_SR2	SmartReflex2 module
CFG_TA_SR3	SmartReflex3 module
CFG_TA_MAILBOX	Mailbox module
CFG_TA_SPINLOCK	Spinlock module
CFG_TA_FACEDETECT	Face detect module
CFG_TA_EMIFFW	EMIF firewall
CFG_TA_GPMCFW	GPMC firewall
CFG_TA_OCMRAMFW	RAM firewall
CFG_TA_SGXFW	SGX firewall
CFG_TA_ISSF	ISS firewall
CFG_TA_CORTEXM3FW	Cortex-M3 firewall
CFG_TA_DSSFW	Display subsystem (DSS) firewall
CFG_TA_SL2FW	SL2 firewall
CFG_TA_IVAHDFW	IVA-HD firewall
CFG_TA_EMUSSF	EMU subsystem firewall
CFG_TA_ABEFW	ABE firewall
CFG_TA_L4WKUP	L4_WKUP connection

A unique port, CFG\_IA\_0, communicates between the L3 interconnect and the L4\_CFG interconnect to allow the L3 initiators to access the L4\_CFG targets (see [Table 13-277](#)).

For the list of initiators authorized to access the L4\_CFG peripherals, see [Section 13.2.3.2.2, Connectivity Matrix](#).

**Table 13-277. L4\_CFG IAs**

Module Initiator Name	Description
CFG_IA_0	L3 interconnect port

### 13.3.3.1.3 L4\_WKUP Interconnect Agents

The L4-WKUP interconnect handles transfers only to peripherals in the WKUP power domain.

[Table 13-278](#) lists the TAs. [Table 13-279](#) lists the L4 WKUP initiator agents.

**Table 13-278. L4\_WKUP TAs**

Module Target Name	Description
WKUP_TA_GPIO1	GPIO1 module
WKUP_TA_32KTIMER	32-kHz timer module
WKUP_TA_SAR_RAM	Save-and-restore RAM
WKUP_TA_KEYBOARD	Keyboard module
WKUP_TA_SYSCTRL_GENERAL_WKUP	System control general WKUP module
WKUP_TA_SYSCTRL_PADCONF_WKUP	System control padconf WKUP module
WKUP_TA_PRM	Power reset management module
WKUP_TA_SCRM	System clock and reset management module
WKUP_TA_WDTIMER1	WDTIMER1 module
WKUP_TA_WDTIMER2	WDTIMER2 module
WKUP_TA_GPTIMER1	GPTIMER1 module
WKUP_TA_GPTIMER2	GPTIMER12 module

**Table 13-279. L4\_WKUP IAs**

Module Initiator Name	Description
WKUP_IA_0	L4_CFG interconnect port

### 13.3.3.2 Power Management

As part of the system-wide power-management scheme, the L4 interconnects go into IDLE state after receiving a request from the PRCM module after all commands are serviced. This function is handled by hardware. For more information, see [Chapter 3, Power, Reset, and Clock Management](#).

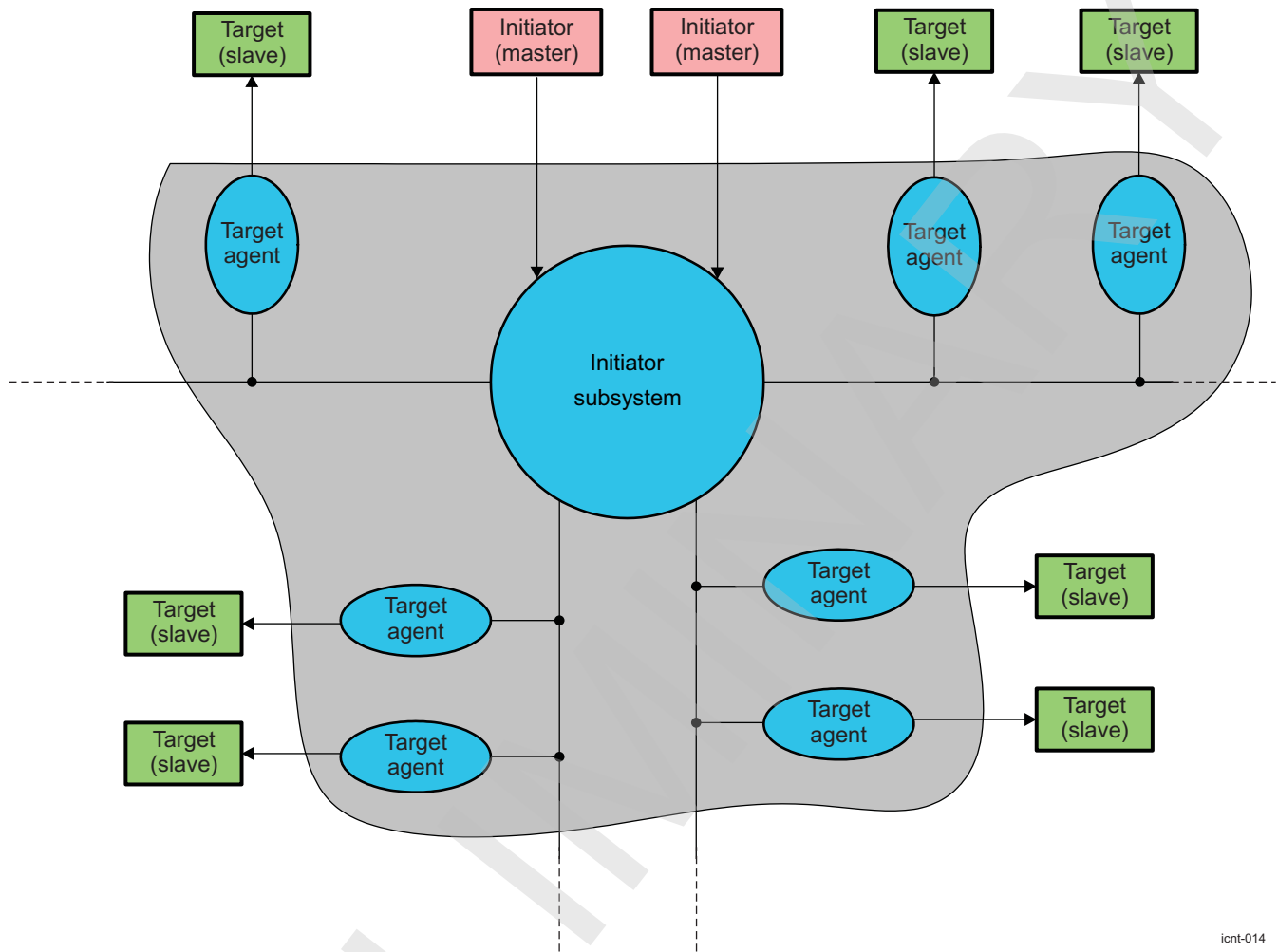
To reduce power consumption, each L4 interconnect automatically performs internal clock autogating. This is managed by hardware; no software configurations or settings are required.

Retention is performed on all software-accessible registers at IAs, TAs, and firewalls. This process prevents reconfiguration after a clock domain switches off. For more information, see [Chapter 3, Power, Reset, and Clock Management](#).

### 13.3.3.3 L4 Firewalls

[Figure 13-9](#) is an internal view of the L4 interconnects in the overall interconnect. This architecture, with one initiator subsystem centralizing all initiator master requests and distributing them to all target modules (peripherals), enables the L4 interconnect firewall functions to be centralized at the L4 initiator subsystem level. The L4 firewall filters the accesses based on the configurable protection groups defined in the L4 address protection (AP) registers. Each module or TA is assigned to a protection group. The configuration is also defined in the L4 AP and is programmable on a module-per-module basis.

Figure 13-9. L4 Initiator-Target Connectivity



icnt-014

**NOTE:** As Figure 13-9 shows, targets are attached to branches. Branches do not impact the function of the L4 interconnect but are present to simplify timing closure and reduce active power consumption.

Because of the large address spaces and the number of peripherals connected to an L4 interconnect, two parameters are used to set up access permission:

- Programmable groups for initiators
  - Eight protection groups for L4\_PER and L4\_CFG interconnect
- Segments divided into regions of 4KB
  - 100 regions for the L4\_PER interconnect
  - 104 regions for the L4\_CFG interconnect

**NOTE:**

- Regions and segments are present for the L4\_WKUP interconnect but cannot be programmed.
- L4\_WKUP protection is done through the L4\_CFG interconnect.

Protection group members are TAs with the same protection settings. A region is programmed to allow access to a unique selectable protection group. For better protection, different regions are grouped into protection group regions and associated with a protection group member.

All L4 interconnect information, such as numbers of protection groups, regions, and segments, is available in the [L4\\_LA\\_INITIATOR\\_INFO\\_L](#) register.

### 13.3.3.3.1 Protection Group

A protection group is defined by its initiators and REQ\_INFO is allowed. Two registers define these two settings:

- The CONNID\_BIT\_VECTOR field [L4\\_AP\\_PROT\\_GROUP\\_MEMBERS\\_k\\_L](#) is a 1-bit vector that defines which initiator belongs to a group. A protection group is accessible by an initiator when the bit position corresponding to its ConnID is set to 1 in the CONNID\_BIT\_VECTOR field. [Table 13-280](#) lists all the ConnID available at the L4 levels.
- The ENABLE field [L4\\_AP\\_PROT\\_GROUP\\_ROLES\\_k\\_L](#) lists all possible REQ\_INFO combinations associated with the [L4\\_AP\\_PROT\\_GROUP\\_MEMBERS\\_k\\_L](#) register. Setting a Req bit in this register determines the initiators type of access. For more information, see [Section 13.2.3.7.3, L3 Firewall Functionality](#). Two REQ\_INFOS are used in L4 interconnects: MReqDebug and MReqSupervisor.

---

**NOTE:** Permissions are identical for read and write accesses in L4 interconnect.

k indicates the protection group number.

L indicates the region number.

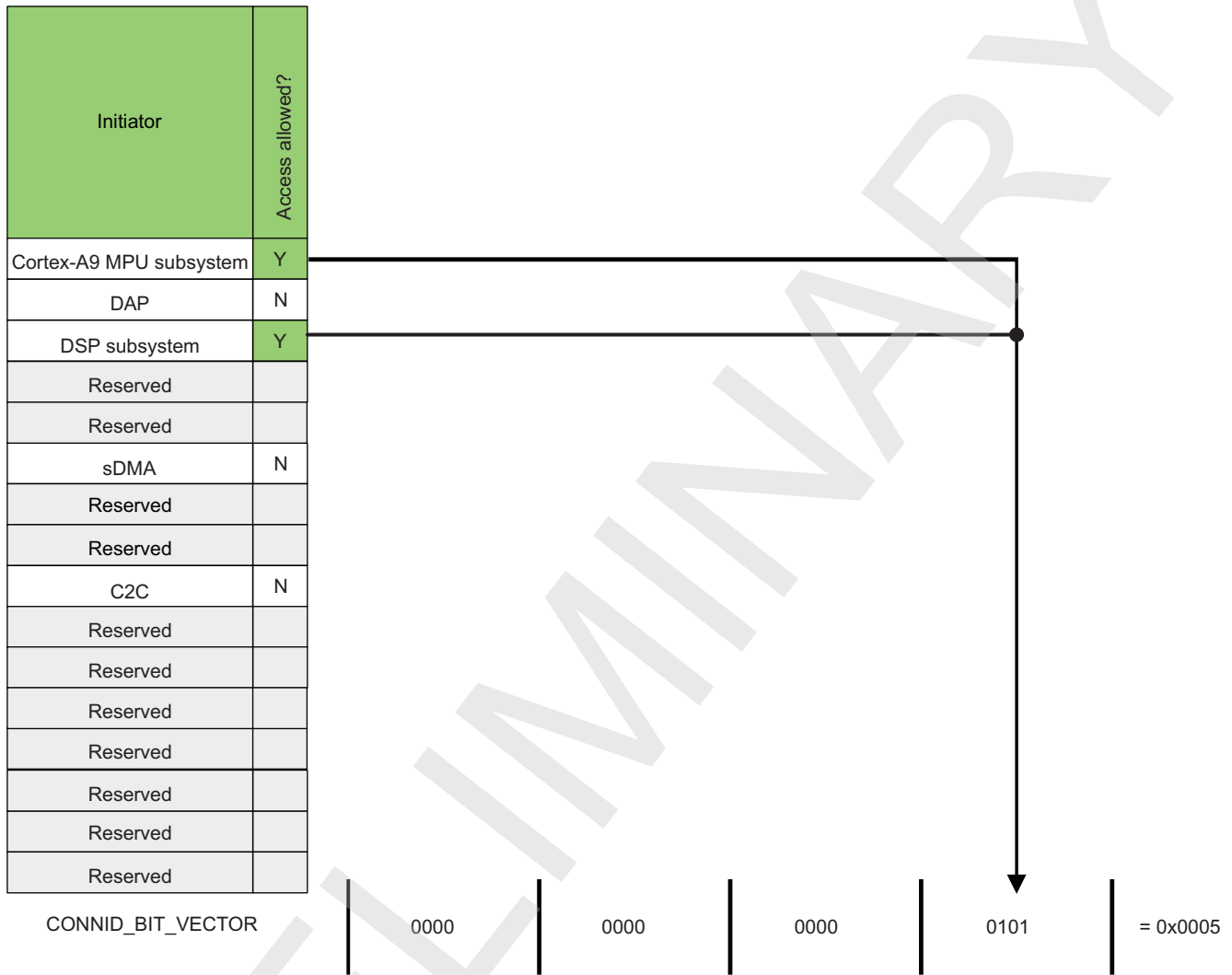
---

**Table 13-280. L4 ConnID Definition**

ConnID	Initiator
0	Cortex™-A9 MPU subsystem
1	DAP
2	DSP subsystem
3	Reserved
4	Reserved
5	sDMA
6	Reserved
7	Reserved
8	C2C
9	Reserved
A	Reserved
B	Reserved
C	Reserved
D	Reserved
E	Reserved
F	Reserved

Figure 13-10 is an example of CONNID\_BIT\_VECTOR.

**Figure 13-10. Example of CONNID\_BIT\_VECTOR L4\_AP\_PROT\_GROUP\_MEMBERS\_k**



Setting bits 0 and 2 in the PROT\_GROUP\_ID\_1 register defines a group initiator that can access targets in protection group 1, and includes the following:

- Cortex-A9 MPU subsystem
- DSP subsystem

Protection group 1 can be applied to multiple protection regions with no limitation. Each protection region that is configured with protection group 1 enables permission access only to the two initiators.

The L4\_AP\_REGION\_I\_H PROT\_GROUP\_ID field determines the region to which the protection group member is attached.

The values of some CONNID\_BIT\_VECTOR and ENABLE fields are exported by the SCM at reset or are user writable (see Table 13-403 for more information).

Table 13-281 and Table 13-282 list the default configuration of the various groups for each L4 interconnect. For each group, some modules/regions are associated with it with default initiator members.

**Table 13-281. L4\_PER Firewall Default Configuration**

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 0 AP registers	AP registers	L4_AP_PROT_GROUP_ROL ES_0_L	No	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_MEM BERS_0_L	No	0xFFFF (all)
Group 1	Reserved	L4_AP_PROT_GROUP_ROL ES_1_L	No	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_MEM BERS_1_L	No	0xFFFF (all)
Group 2	Reserved	L4_AP_PROT_GROUP_ROL ES_2_L	No	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_MEM BERS_2_L	No	0xFFFF (All)
Group 3–6 Free	No modules attached	L4_AP_PROT_GROUP_ROL ES_3_L	Yes	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_ROL ES_4_L		
		L4_AP_PROT_GROUP_ROL ES_5_L		
		L4_AP_PROT_GROUP_ROL ES_6_L		
		L4_AP_PROT_GROUP_MEM BERS_3_L	Yes	0xFFFF (all)
		L4_AP_PROT_GROUP_MEM BERS_4_L		
		L4_AP_PROT_GROUP_MEM BERS_5_L		
		L4_AP_PROT_GROUP_MEM BERS_6_L		
Group 7 Other modules	Other L4_PER modules	L4_AP_PROT_GROUP_ROL ES_7_L	Yes	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_MEM BERS_7_L	Yes	0xFFFF (all)

**Table 13-282. L4\_CFG Firewall Default Configuration**

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 0 AP registers	AP registers	L4_AP_PROT_GROUP_ROL ES_0_L	No	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_ME MBERS_0_L	No	0xFFFF (all)
Group 1	WDTIMER1, GPTIMER12 SAR_RAM3, all FW registers	L4_AP_PROT_GROUP_ROL ES_1_L	No	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_ME MBERS_1_L	No	0xFFFF (all)
Group 3 SAR_RAM2	SAR_RAM2	L4_AP_PROT_GROUP_ROL ES_3_L	No	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_ME MBERS_3_L	No	0xFFFF (all)
Group 4 SAR_RAM4	SAR_RAM4	L4_AP_PROT_GROUP_ROL ES_4_L	No	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_ME MBERS_4_L	No	0xFFFF (all)
Group 5 Free	Reserved	L4_AP_PROT_GROUP_ROL ES_5_L	Yes	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_ME MBERS_5_L	Yes	0xFFFF (all)

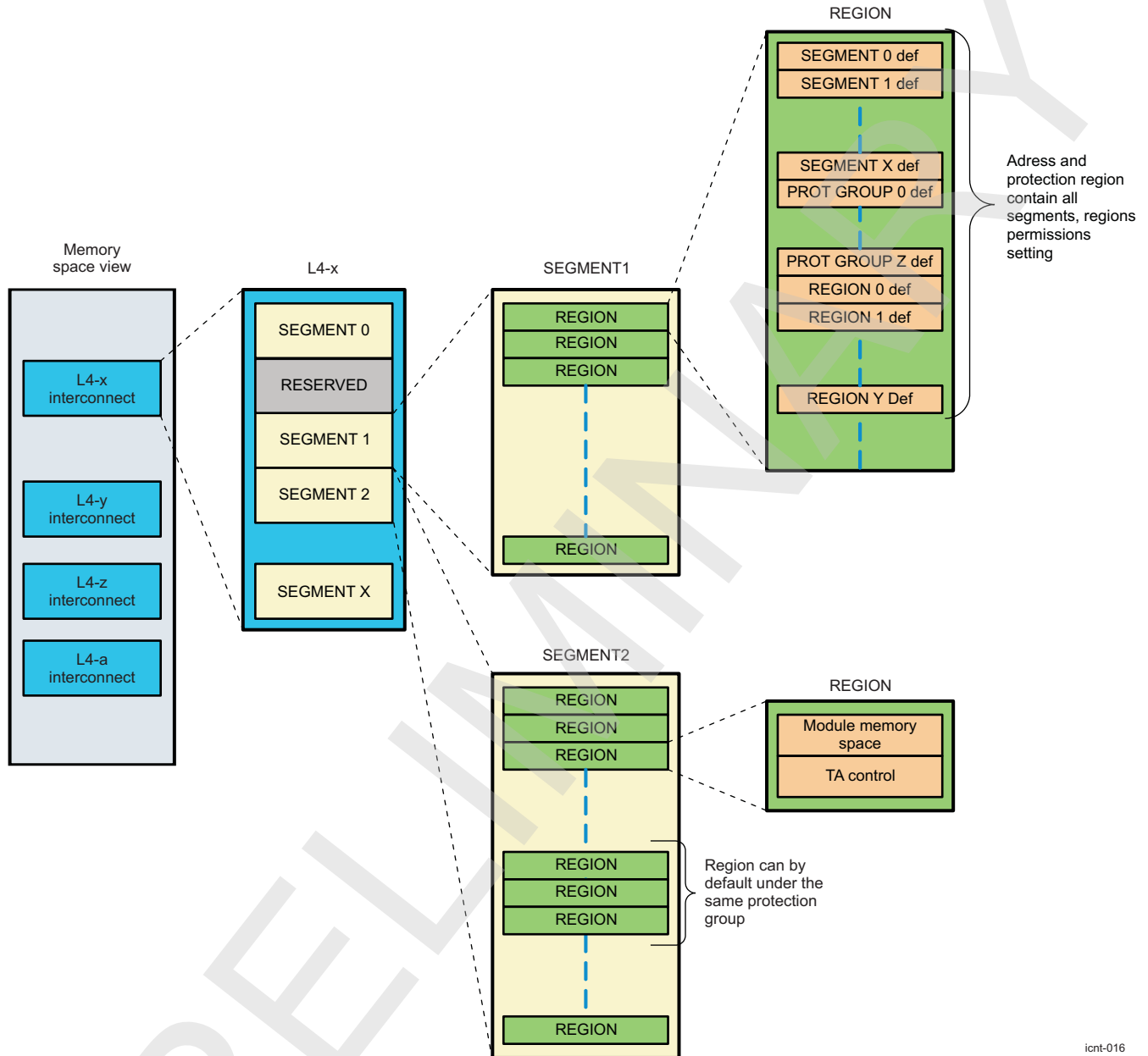
**Table 13-282. L4\_CFG Firewall Default Configuration (continued)**

Group	Default Modules Associated With Group	Register	Modifiability	Default Value
Group 6 Free	No modules attached	L4_AP_PROT_GROUP_ROL ES_6_L	Yes	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_ME MBERS_6_L	Yes	0xFFFF (all)
Group 7 Other modules	Other L4_CFG modules	L4_AP_PROT_GROUP_ROL ES_7_L	Yes	0xFFFF FFFF (all)
		L4_AP_PROT_GROUP_ME MBERS_7_L	Yes	0xFFFF (all)

### 13.3.3.3.2 Segments and Regions

The protection mechanism for L4 interconnects is based on a hierarchical segmentation, as shown in [Figure 13-11](#). By default, some regions are attached to specific protection group members. This specificity lets users set up the permission access to certain types of modules requiring the same access protection without managing region allocation.

Figure 13-11. L4 Segmentation



All interconnect address spaces are covered by regions. [Table 13-283](#) through [Table 13-285](#) list the module mapping with their addresses, region numbers, and default protection group allocated to them.

**NOTE:** Module refers to the configuration registers of the module.

TA refers to the interconnect configuration registers of the TA associated with the module.

Table 13-283. Region Allocations for L4\_PER Interconnect

Module	Region	Description
L4_PER CONFIG	0	Firewall
	1	PER_IA_0
	2	Link agent



**Table 13-283. Region Allocations for L4\_PER Interconnect (continued)**

<b>Module</b>	<b>Region</b>	<b>Description</b>
UART3	3	TA
	4	Module
GPTIMER2	5	TA
	6	Module
GPTIMER3	7	TA
	8	Module
GPTIMER4	9	TA
	10	Module
GPTIMER9	11	TA
	12	Module
DSS	13	TA
	14	Module
GPIO2	15	TA
	16	Module
GPIO3	17	TA
	18	Module
GPIO4	19	TA
	20	Module
GPIO5	21	TA
	22	Module
GPIO6	23	TA
	24	Module
I2C3	36	Module
	25	TA
UART1	26	TA
	27	Module
UART2	28	TA
	29	Module
UART4	30	TA
	31	Module
I2C1	32	TA
	33	Module
I2C2	34	TA
	35	Module
MCBSP4	37	TA
	38	Module
SLIMBUS2	39	TA
	40	Module
ELM	41	TA
	42	Module
GPTIMER10	43	TA
	44	Module
GPTIMER11	45	TA
	46	Module
MCSPI1	49	TA
	50	Module
MCSPI2	51	TA

**Table 13-283. Region Allocations for L4\_PER Interconnect (continued)**

Module	Region	Description
	52	Module
HSMMC1	53	TA
	54	Module
HSMMC3	63	TA
	64	Module
HDQ	67	TA
	68	Module
HSMMC2	69	TA
	70	Module
MCSPI3	71	TA
	72	Module
MCSPI4	73	TA
	74	Module
HSMMC4	75	TA
	76	Module
HSMMC5	77	TA
	78	Module
I2C4	79	TA
	80	Module
L4_PER CONFIG	97	PER_IA_1
	98	PER_IA_2
	99	PER_IA_3

**Table 13-284. Region Allocations for L4\_CFG Interconnect**

Module	Region	Description
L4_CFG CONFIG	0	Firewall
	1	CFG_IA_0
	2	Link agent
CFG_SYSCTRL_GENERAL_CORE	3	TA
	4	Module
CM1	5	TA
	6	Module
sDMA	7	TA
	8	Module
HSI	9	Module
	10	TA
HSUSBTLL	11	TA
	12	Module
SR1	15	TA
	16	Module
SR2	17	TA
	18	Module
SR3	19	TA
	20	Module
CFG_SYSCTRL_PADCONF_CORE	29	TA
	30	Module
CM2	32	TA

**Table 13-284. Region Allocations for L4\_CFG Interconnect (continued)**

Module	Region	Description
	33	Module
DSP subsystem	34	TA
	35	Module
Mailbox	36	TA
	37	Module
SPINLOCK	38	TA
	39	Module
SL2FW	40	TA
	41	Module
OCCRAMFW	48	TA
	49	Module
EMIFFW	50	TA
	51	Module
GPMCFW	52	TA
	53	Module
ISSFW	54	TA
	55	Module
SGXFW	56	TA
	57	Module
CORTEXM3FW	58	TA
	59	Module
IVAHDWF	60	TaA
	61	Module
EMUSSFW	62	TA
	63	Module
ABEFW	64	TA
	65	Module
DSSFW	68	TA
	69	Module
FACEDETECT	74	TA
	75	Module
L4WKUP	76	L4_WKUP regions 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24
	77	Module
	80	L4_WKUP regions 25, 26, 27, 28
	81	L4_WKUP regions 29, 30
	82	L4_WKUP region 13
	83	L4_WKUP region 31
	84	L4_WKUP region 32
	85	L4_WKUP region 33
	86	–
SAR_ROM	91	TA
	92	Module
USBFS	93	TA
	94	Module
USBOTGHS	95	TA
	96	Module

**Table 13-284. Region Allocations for L4\_CFG Interconnect (continued)**

Module	Region	Description
USBHOSTHS	97	TA
	98	Module
USBPHY	100	TA
	101	Module
L4WKUP	102	L4_WKUP region 14
	103	–

**Table 13-285. Region Allocations for L4\_WKUP Interconnect**

Module	Region	Description
L4 WKUP CONFIG	0	AP
	1	CFG_IA_0
	2	Link agent
PRM	3	TA
	4	Module
GPIO1	5	TA
	6	Module
WDTIMER2	7	TA
	8	Module
Timer 1 ms	9	Module
	10	TA
KEYBOARD	11	TA
	12	Module
SAR_RAM	13	TA
	14	Module
SCRM	15	TA
	16	Module
32KTIMER	17	TA
	18	Module
WKUP_SYSCTRL_GENERAL_WKUP	19	TA
	20	Module
WKUP_TA_SYSCTRL_PADCONF_WKUP	23	TA
	24	Module
GPTIMER12	25	TA
	26	Module
WDTIMER1	27	TA
	28	Module
SAR_RAM	31	TA
	32	TA
	33	TA

### 13.3.3.3 L4 Firewall Address and Protection Register Settings

[Table 13-286](#) lists the settings of the AP registers relative to an L4 interconnect firewall. These values are computed based on the physical implementation of each L4 interconnect.

**Table 13-286. L4 Firewall Register Description Overview**

Register Type	Register Name	Bits	Field	Description
Segment	L4_AP_SEGMENT_i_L <sup>(1)</sup>	23:0	BASE	Segment base address
	L4_AP_SEGMENT_i_H <sup>(1)</sup>	4:0	SIZE	Segment size equals to 2 <sup>SIZE</sup>
Protection groups	L4_AP_PROT_GROUP_MEMBERS_k_L <sup>(2)</sup>	15:0	CONNID_BIT_VECTOR	For L4ConnID, see <a href="#">Table 13-280</a> .
	L4_AP_PROT_GROUP_ROLES_k_L <sup>(2)</sup>	31:0	ENABLE	See <a href="#">Section 13.2.3.7.3</a> for REQ_INFO description <sup>(3)</sup>
Region setting	L4_AP_REGION_I_L <sup>(4)</sup>	23:0	BASE	Defines the base address of region with respect to its segment base address
	L4_AP_REGION_I_H <sup>(4)</sup>	27:24	SEGMENT_ID	Segment ID number of the region
		22:20	PROT_GROUP_ID	Protection group member attached to the region
		19:17	BYTE_DATA_WIDTH_EXP	Determines the number of byte in an access
		5:1	SIZE	Size of the region equals to 2 <sup>SIZE</sup>
	0	ENABLE	Enables the region protection	

<sup>(1)</sup> i = 0 to 1 for PER\_AP  
i = 0 to 6 for CFG\_AP

<sup>(2)</sup> k = 0 to 7 for PER\_AP  
k = 0 to 7 for CFG\_AP

<sup>(3)</sup> For L interconnects, only MReqDebug and MReqSupervisor are available.

<sup>(4)</sup> I = 0 to 96 for PER\_AP  
I = 0 to 113 for CFG\_AP

### 13.3.3.4 L4 Error Detection and Reporting

#### 13.3.3.4.1 IA and TA Error Detection and Logging

The L4 interconnect provides mechanisms for handling internally detected errors or errors reported by modules attached to the L4 target ports.

**NOTE:** L4\_IA denotes the IA for all L4 interconnects: L4\_PER, L4\_CFG, and L4\_WKUP.

L4\_TA denotes the TA for all L4 interconnects: L4\_PER, L4\_CFG, and L4\_WKUP.

The L4 interconnects handle four types of errors:

- No target core found or address hole, detected and logged at IA
- Unsupported command, detected and logged at IA
- Protection violation, detected and logged at IA (see [Section 13.3.3.3, L4 Firewalls](#))
- Target does not service a request before a time-out expires. The error is detected and logged at TA (see [Section 13.3.3.4.2, Time-Out](#)).

[Table 13-287](#) lists the value of the L4\_IA\_ERROR\_LOG\_L[25:24] CODE bit field stored when an error occurs.

**Table 13-287. L4 CODE Bit Field Definition**

CODE (bits 1:0)	Error Type	REQ_INFO	Secondary	ConnID	CMD
0	No error				
1	Unsupported command	x	x	x	x
2	Address hole	x	x	x	x
3	Protection violation	x		x	x

- No target core found/address hole: This error indicates that a request was addressed to a hole in the L4 address map. When this error occurs, an in-band error response is returned to the L3 level. The error is also logged into the [L4\\_IA\\_AGENT\\_STATUS\\_L\[27\] INBAND\\_ERROR](#) bit. Additionally, an address hole error code is logged to the [L4\\_IA\\_ERROR\\_LOG\\_L\[25:24\] CODE](#) bit field.
- Unsupported command: This error indicates that the command type of the request is not supported by the accessed target register. The error is logged into the [L4\\_IA\\_AGENT\\_STATUS\\_L\[27\] INBAND\\_ERROR](#) bit. An unsupported command error code is written to the [L4\\_IA\\_ERROR\\_LOG\\_L\[25:24\] CODE](#) bit field for the initiator interface.
- Protection violation: This error indicates that a request is not issued from an allowed initiator member or is issued with the inappropriate ReqInfo qualifiers associated with the target region. This error is reported using an in-band error and is written to the [L4\\_IA\\_AGENT\\_STATUS\\_L\[27\] INBAND\\_ERROR](#) bit. A protection violation error code is saved into the [L4\\_IA\\_ERROR\\_LOG\\_L\[25:24\] CODE](#) bit field for the same initiator interface. A protection violation is also logged in the [L4\\_IA\\_AGENT\\_STATUS\\_L\[31\] PROT\\_ERROR\\_SECONDARY](#) or [\[30\] PROT\\_ERROR\\_PRIMARY](#) bit when in debug or applicative mode, respectively.

The [L4\\_IA\\_ERROR\\_LOG\\_L\[30\] SECONDARY](#) bit indicates whether the error occurred in application or debug.

The [L4\\_IA\\_ERROR\\_LOG\\_H\[15:0\] REQ\\_INFO](#) bit field returns the type access (REQ\_INFO qualifier) that caused the error.

The [L4\\_IA\\_ERROR\\_LOG\\_L\[11:8\] CONNID](#) bit field returns the ID of the initiator that caused the error.

The [L4\\_IA\\_ERROR\\_LOG\\_ADDR\\_L\[31:0\]](#) register logs the address for error conditions.

A [L4\\_IA\\_ERROR\\_LOG\\_L\[31\] MULTI](#) bit is asserted when multiple errors are detected. In this case, the error code corresponds to the first error that occurs.

#### 13.3.3.4.2 Time-Out

A time-out mechanism can be enabled at the interconnect level and on a per-target basis. If the mechanism is enabled for a TA and interconnect and commands are not accepted or responses are not returned within the expected delay, the L4 interconnect generates an error event.

---

**NOTE:** The time-out mechanism is not available on the L4\_WKUP interconnect, but L4\_WKUP timeouts are detected in CFG\_TA\_L4WKUP of the L4\_CFG interconnect.

---

The error is logged in the [L4\\_TA\\_AGENT\\_STATUS\\_L\[8\] REQ\\_TIMEOUT](#) bit. The affected TA enters an error state that causes it to send an error response to any new request targeted at it. To recover from this state, the TA must be reset by the system software. The time-out is counted starting from the moment a command is presented to the target, regardless of the target response to this command.

The L4 interconnect implements a centralized time-base circuit that broadcasts a set of four periodic pulse signals to all connected TAs. The time-base circuit offers four possible sets of four time-base signals. Selection is done by programming the [L4\\_LA\\_NETWORK\\_CONTROL\\_L\[10:8\] TIMEOUT\\_BASE](#) bit field.

The selected time-base signals are available at any TA. Each TA can be programmed to refer to one of these four time-base signals, using the [L4\\_TA\\_AGENT\\_CONTROL\\_L\[10:8\] REQ\\_TIMEOUT](#) bit field. These four signals are referred to as 1X time-base, 4X time-base, 16X time-base, and 64X time-base.

[Table 13-288](#) lists all values in number of L4 clock cycles.

**Table 13-288. L4 Time-Out Link and TA Programming**

<a href="#">L4_LA_NETWORK_CONTROL_L TIMEOUT_BASE[2:0]</a>	<a href="#">L4_TA_AGENT_CONTROL_L REQ_TIMEOUT[2:0]</a>			
	0	1	2	3
0	All L4 time-out features are disabled.			

**Table 13-288. L4 Time-Out Link and TA Programming (continued)**

L4_LA_NETWORK_CONTROL_L TIMEOUT_BASE[2:0]		L4_TA_AGENT_CONTROL_L REQ_TIMEOUT[2:0]				
		0	1	2	3	4
1	Locally disabled		64	256	1024	4096
2			256	1024	4096	16,384
3			1024	4096	16,384	65,536
4			4096	16,384	65,536	262,144

The default reset value is 0x2 for REQ\_TIMEOUT and 0x4 for TIMEOUT\_BASE, implying 16,384 clock cycles.

A time-out condition is detected when the command acceptance or the response is not received after a delay of from one to three time-base periods.

**Example:**

- L4 frequency = 65 MHz
- TIMEOUT\_BASE = 4 in the L4\_LA\_NETWORK\_CONTROL\_L register
- REQ\_TIMEOUT = 2 in the L4\_TA\_AGENT\_CONTROL\_L for TA A
- REQ\_TIMEOUT = 4 in the L4\_TA\_AGENT\_CONTROL\_L for TA B

At agent A, the time-base unit is 16,384 cycles. A time-out is issued when a request to the attached module is not accepted, or no response is sent after a delay of 252 μs to 756 μs.

At agent B, the time-base unit is 262,144 cycles. A time-out is issued when a request to the attached module is not accepted or no response is sent after a delay of 4 ms to 12 ms.

On detection of a time-out condition, the TA logs the error in the L4\_TA\_AGENT\_STATUS\_L[8] REQ\_TIMEOUT bit, and it also reports the error to the IA, which forward it to the L3 interconnects.

After the time-out is detected and logged, the behavior of the attached module is ignored. A new request targeting the module arriving at the timed-out TA receives an error response. If the request is addressed to the agent internal registers, it is processed normally.

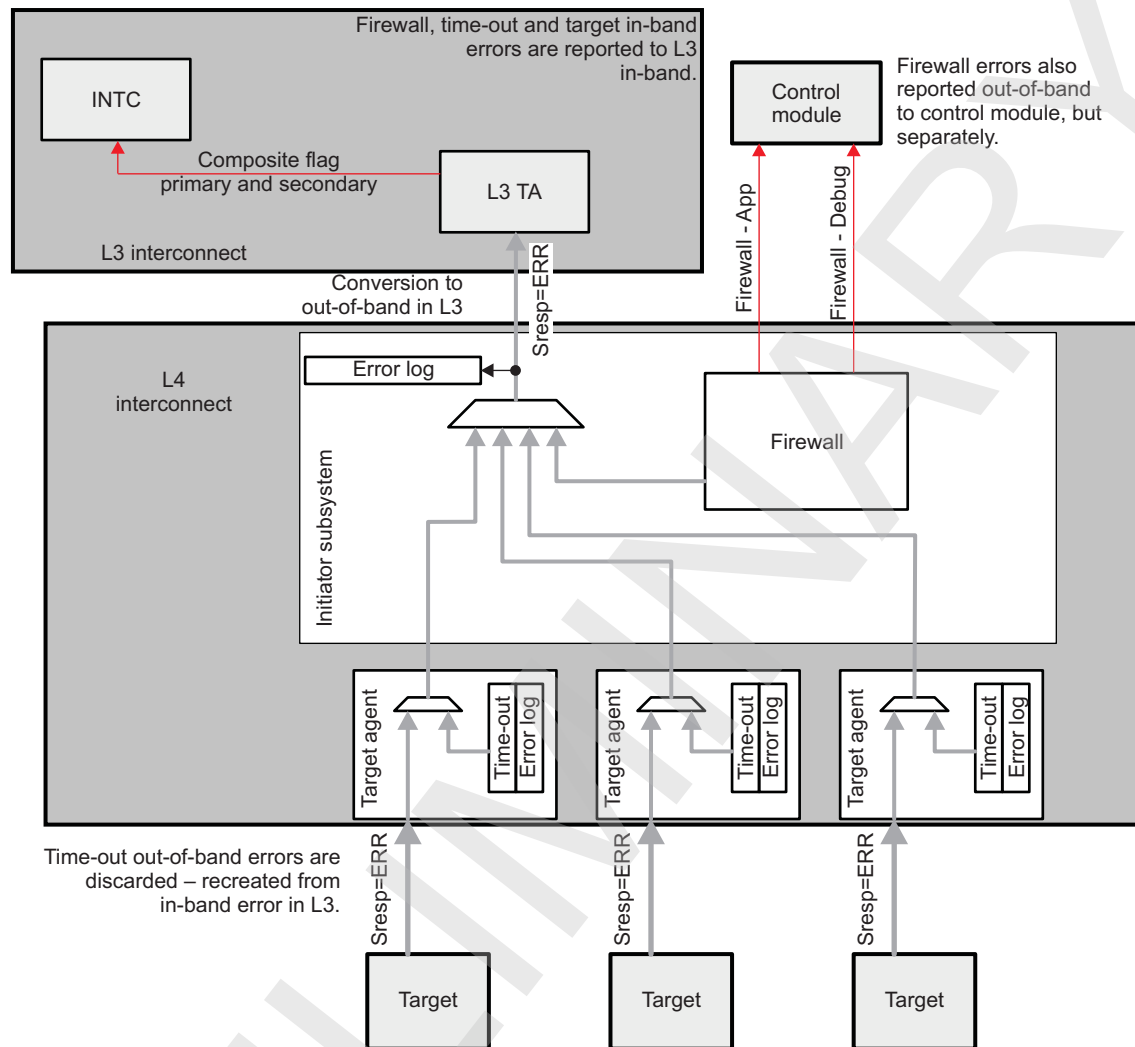
To recover from a time-out error, software is assumed to first reset the faulty module and then the TA using the L4\_TA\_AGENT\_CONTROL\_L[0] OCP\_RESET bit.

**13.3.3.4.3 Error Reporting**

Figure 13-12 shows the error-reporting scheme used in the L4 interconnects. All L4 in-band errors are reported to their respective L3 TA, where errors are converted in an out-of band error signal (the L3 applicative and debug composite flags) going to the L3 Interrupt controllers.

Two levels of mask are present to report the error at INTC level:

- At the applicative and debug composite flag, to enable interrupt reporting, the following bits must be set:
  - L4 CFG in L3\_CLK1\_FLAG\_MASK\_0[3] and L3\_CLK1\_FLAG\_STATUS\_1[3]
  - L4 PER in L3\_CLK2\_FLAG\_MASK\_0[17][16][15][6] and L3\_CLK2\_FLAG\_STATUS\_1[17][16][15][6]
- At the L3 TA level, see Section 13.2, L3 Interconnect.

**Figure 13-12. L4 Error Reporting**

icnt-017

**13.3.3.4.4 Error Recovery**

Writing 1 to the `L4_TA_AGENT_CONTROL_L[0] OCP_RESET` bit initiates the software reset period. Software reset must be asserted for at least 16 cycles of the target module interface clock, which can be a divided clock with respect to the L4 clock.

During the software reset period:

- Requests sent to the target module receive error responses. Therefore, if the faulty request is part of a DMA transfer, it is necessary to stop the DMA to prevent unwanted errors.
- Requests sent to the TA register block are processed as usual.
- The `L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT` status bit is cleared.

Writing 0 to the `L4_TA_AGENT_CONTROL_L[0] OCP_RESET` bit terminates the software reset period.

The attached module must then be reset to complete the recovery.



### 13.3.3.4.5 Firewall Error Logging in the Control Module

When a protection violation occurs, an interrupt is sent to the INTCs (if enabled). An in-band error is sent back to L3 IA and an out-band error can also be logged in the CONTROL.CONTROL\_SEC\_ERR\_STATUS register in the system control module (SCM). These out-band errors are enabled/disabled at the L4 IA level by setting the L4\_IA\_AGENT\_CONTROL\_L[31][30] PROT\_ERROR\_SECONDARY\_REP or PROT\_ERROR\_PRIMARY\_REP bit to 1 for debug and application mode, respectively.

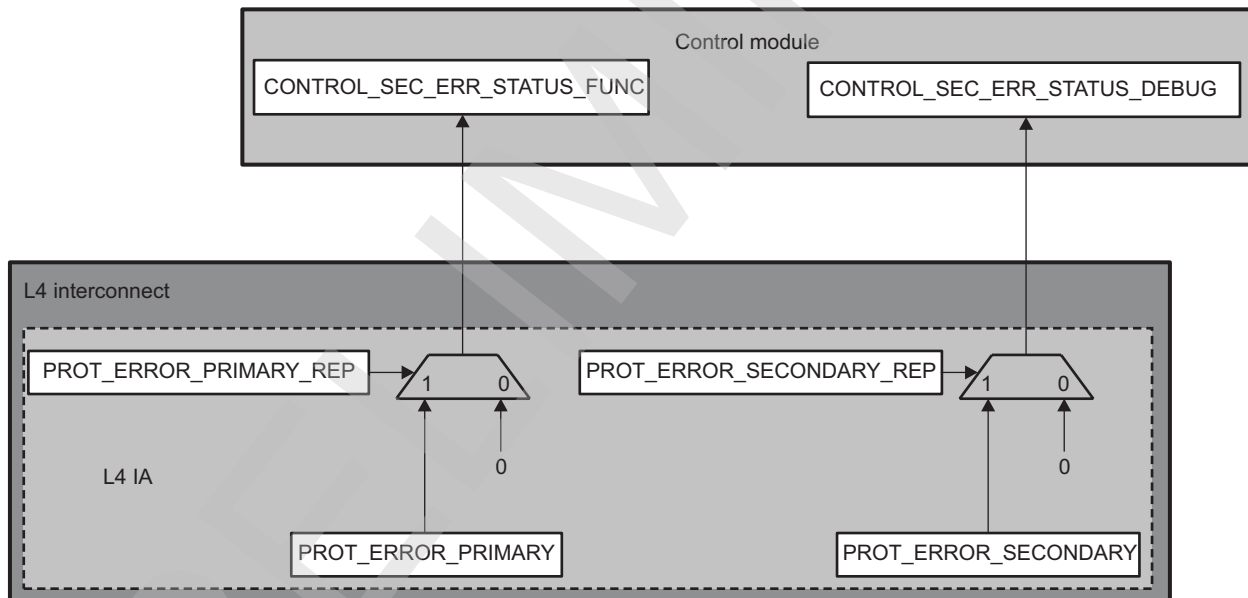
At the control module level, two logging registers are used, depending on the mode:

- In application mode or primary error reporting:
  - CONTROL.CONTROL\_SEC\_ERR\_STATUS\_FUNC[16] = L4\_CFG protection violation
  - CONTROL.CONTROL\_SEC\_ERR\_STATUS\_FUNC[17] = L4\_PER protection violation
- In debug mode or secondary error reporting:
  - CONTROL.CONTROL\_SEC\_ERR\_STATUS\_DEBUG[16] = L4\_CFG protection violation
  - CONTROL.CONTROL\_SEC\_ERR\_STATUS\_DEBUG[17] = L4\_PER protection violation

**NOTE:** The CONTROL.CONTROL\_SEC\_ERR\_STATUS\_xxx registers are only readable.

Figure 13-13 shows the global protection error reporting to at the control module.

**Figure 13-13. Protection Violation Out-of-Band Error Reporting**



icnt-018

## 13.3.4 L4 Interconnect Programming Guide

### 13.3.4.1 L4 Interconnect Low-level Programming Models

This section describes the low-level hardware programming sequences for configuring and using the L4 interconnect module.

#### 13.3.4.1.1 Global Initialization

##### 13.3.4.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the L4 interconnect module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the L4 interconnect. For more information, see [Section 13.3.2, L4 Interconnect Integration](#).

**Table 13-289. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	For more information about the configuration of the module, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Control module	For more information about the configuration of the module, see <a href="#">Chapter 18, Control Module</a> .
MPU INTC	The MPU INTC must be configured to enable the interrupts from L4 interconnect module. See <a href="#">Chapter 17 Interrupt Controllers</a> .
sDMA	For more information about the configuration of the sDMA, see <a href="#">Chapter 16, sDMA</a> .
L3 interconnect	For more information about the interconnect configuration, see <a href="#">Chapter 13, Interconnect</a> .

#### 13.3.4.1.2 Operational Modes Configuration

##### 13.3.4.1.2.1 L4 Interconnect Error Analysis Mode

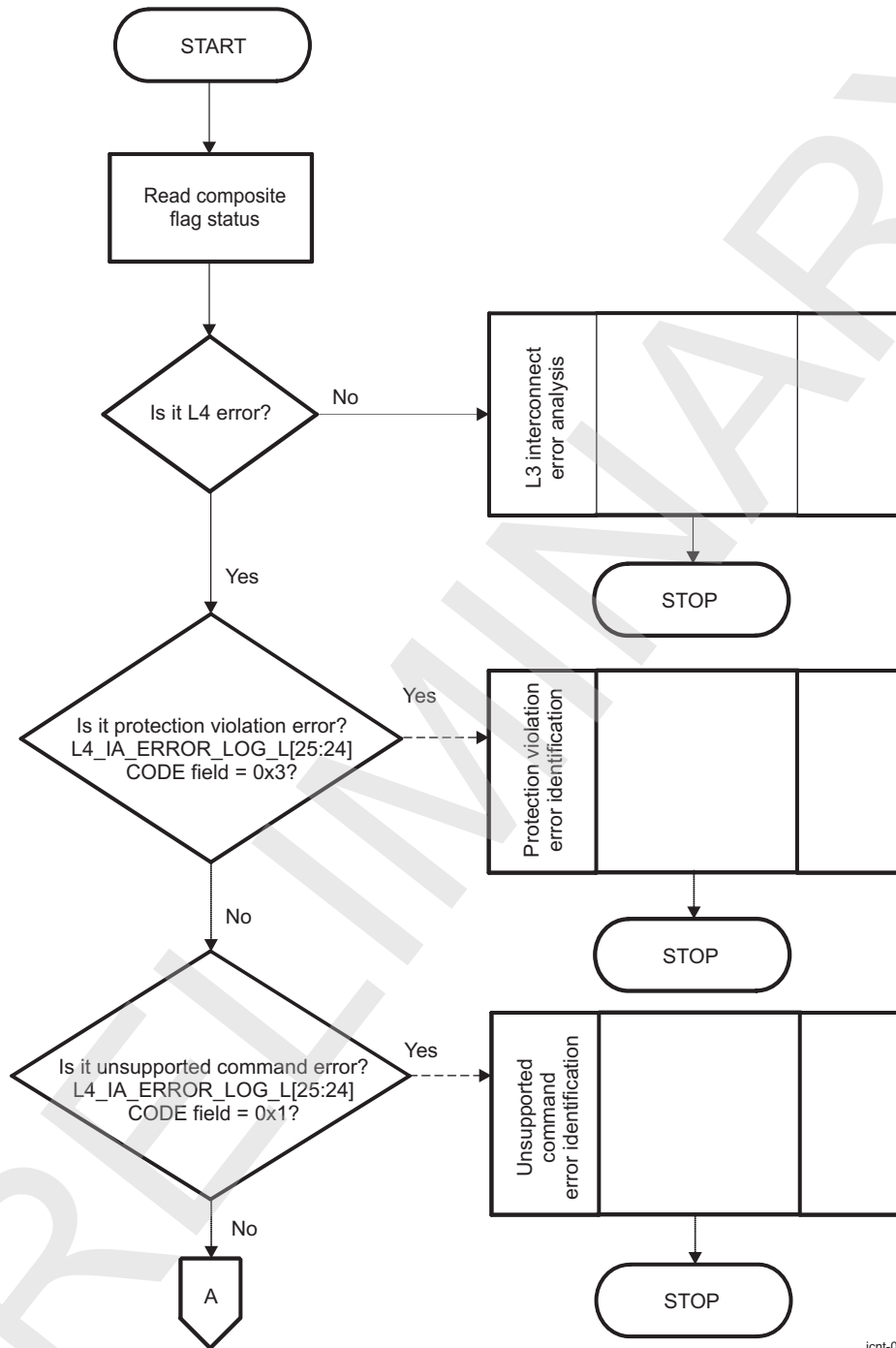
###### 13.3.4.1.2.1.1 Main Sequence: L4 Interconnect Error Analysis Mode

The information required to analyze an error source is logged in several registers. The number of registers to access depends on the error source.

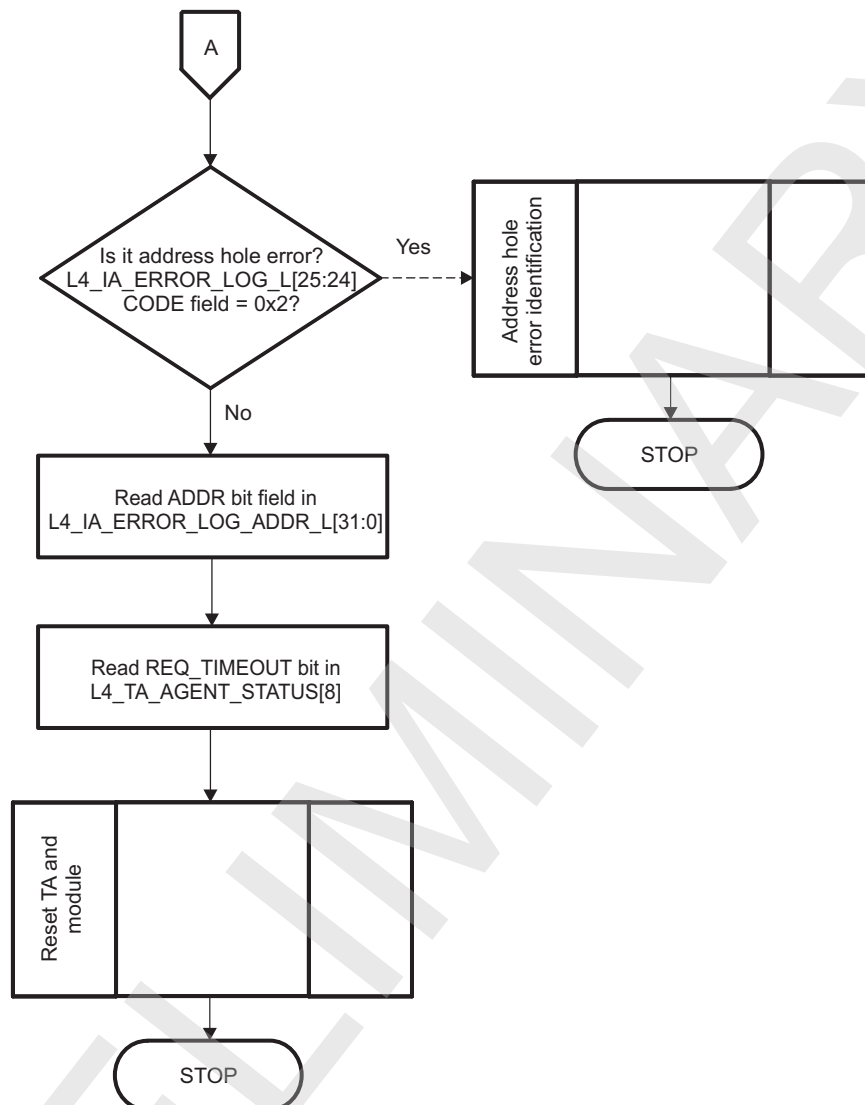
[Figure 13-14](#) and [Figure 13-15](#) show the software sequence required in most cases.

[Table 13-290](#) and [Table 13-291](#) show the main sequence for error analysis mode and its subprocess call summary, respectively.

Figure 13-14. Typical Error Analysis Sequence



icnt-019

**Figure 13-15. Typical Error Analysis Sequence**

icnt-020

**Table 13-290. Main Sequence – Error Analysis Mode**

Register Name	Register Name	Register Name
<a href="#">L4_IA_ERROR_LOG_L</a>	<a href="#">L4_IA_ERROR_LOG_ADDR_L</a>	<a href="#">L4_IA_AGENT_STATUS_L</a>
<a href="#">L4_IA_AGENT_CONTROL_L</a>	CONTROL.CONTROL_SEC_ERR_STAT US_FUNC	CONTROL.CONTROL_SEC_ERR_STAT US_DEBUG
<a href="#">L4_TA_AGENT_STATUS_L</a>		

**Table 13-291. Subprocess Call Summary for Main Sequence – Error Analysis Mode**

Subprocess	Cross-Reference
L3 interconnect error analysis	<a href="#">Section 13.2.3.8, L3 Interconnect Error Handling</a>
L4 interconnect protection violation error identification	<a href="#">Section 13.3.4.1.2.1.2, Subsequence: L4 Interconnect Protection Violation Error Identification</a>
L4 interconnect unsupported command/address hole error identification	<a href="#">Section 13.3.4.1.2.1.3, Subsequence: L4 Interconnect Unsupported Command/Address Hole Error Identification</a>
L4 interconnect reset TA and module	<a href="#">Section 13.3.4.1.2.1.4, Subsequence: L4 Interconnect Reset TA and Module</a>

### 13.3.4.1.2.1.2 Subsequence: L4 Interconnect Protection Violation Error Identification

This procedure describes the protection violation error identification (see [Table 13-292](#)).

**Table 13-292. Protection Violation Error Identification**

Step	Register/Bit Field/Programming Model	Value
Read multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	
Read initiator ID.	L4_IA_ERROR_LOG_L[11:8] CONNID	
Read command that cause the error.	L4_IA_ERROR_LOG_L[2:0] CMD	
Read address of request that caused the error.	L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR	
<b>IF:</b> Is it a primary error?	L4_IA_AGENT_STATUS_L[30] PROT_ERROR_PRIMARY	=0x1
Read status bits.	CONTROL.CONTROL_SEC_ERR_STATUS_FUNC[17:16]	
Write 1 to clear status bits.	CONTROL.CONTROL_SEC_ERR_STATUS_FUNC[17:16]	0x3
Write 1 to clear IA status bit.	L4_IA_AGENT_STATUS_L[30] PROT_ERROR_PRIMARY	0x1
<b>ELSE</b>		
Read status bits.	CONTROL.CONTROL_SEC_ERR_STATUS_DEB_UG[17:16]	
Write 1 to clear status bits.	CONTROL.CONTROL_SEC_ERR_STATUS_DEB_UG[17:16]	0x3
Write 1 to clear IA status bit	L4_IA_AGENT_STATUS_L[31] PROT_ERROR_SECONDARY	0x1
<b>ENDIF</b>		
Write 1 to clear multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	0x1
Write 1 to clear in-band error status.	L4_IA_AGENT_STATUS_L[24] INBAND_ERROR	0x1

### 13.3.4.1.2.1.3 Subsequence: L4 Interconnect Unsupported Command/Address Hole Error Identification

This procedure describes the identification of unsupported command/address hole error (see [Table 13-293](#)).

**Table 13-293. Unsupported Command/Address Hole Error Identification**

Step	Register/Bit Field/Programming Model	Value
Read multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	
Read initiator ID.	L4_IA_ERROR_LOG_L[11:8] CONNID	
Read command that caused the error.	L4_IA_ERROR_LOG_L[2:0] CMD	
Read address of request that caused the error.	L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR	
Read secondary status.	L4_IA_ERROR_LOG_L[30] SECONDARY	
Write 1 to clear secondary status.	L4_IA_ERROR_LOG_L[30] SECONDARY	0x1
Write 1 to clear multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	0x1
Write 1 to clear inband error status.	L4_IA_AGENT_STATUS_L[24] INBAND_ERROR	0x1

### 13.3.4.1.2.1.4 Subsequence: L4 Interconnect Reset TA and Module

This procedure resets the TA and module (see [Table 13-294](#)).

**Table 13-294. Reset TA and Module**

Step	Register/Bit Field/Programming Model	Value
Reset TA.	L4_TA_AGENT_CONTROL_L[0] OCP_RESET	0x1
Wait until target module clock = 16 cycles.		
Write 0 to clear TA time - out status.	L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT	0x0

**Table 13-294. Reset TA and Module (continued)**

Step	Register/Bit Field/Programming Model	Value
Write 0 to clear TA reset.	L4_TA_AGENT_CONTROL_L[0] OCP_RESET	0x0
Reset the attached module. <sup>(1)</sup>		

<sup>(1)</sup> For more information, see the respective module chapter.

### 13.3.4.1.2.2 L4 Interconnect Time-Out Configuration Mode

#### 13.3.4.1.2.2.1 Main Sequence: L4 Interconnect Time-Out Configuration Mode

This procedure describes the time-out configuration sequence (see [Table 13-295](#)).

**Table 13-295. Time-Out Configuration**

Step	Register/Bit Field/Programming Model	Value
Disable time-out.	L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE	0x0
Clear TA time-out error status. <sup>(1)</sup>	L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT	0x1
Set time-out at TA level. <sup>(1)</sup>	L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT	xxx
Set time-out base.	L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE	xxx

<sup>(1)</sup> Must be done for each TA.

### 13.3.4.1.2.3 L4 Interconnect Firewall Configuration Mode

#### 13.3.4.1.2.3.1 Main Sequence: L4 Interconnect Firewall Configuration Mode

This procedure describes the firewall configuration sequence (see [Table 13-296](#)).

**Table 13-296. Firewall Configuration**

Step	Register/Bit Field/Programming Model	Value
Define the members of protection group k. <sup>(1)</sup>	L4_AP_PROT_GROUP_MEMBERS_k_L[15:0] CONNID_BIT_VECTOR	xxx
Define the access type of a protection group k. <sup>(1)</sup>	L4_AP_PROT_GROUP_ROLES_k_L[15:0] ENABLE	xx
Set region affiliation to protection group. <sup>(2)</sup>	L4_AP_REGION_I_L[22:20] PROT_GROUP_ID	xxx

<sup>(1)</sup> Must be done for each protection group.

<sup>(2)</sup> Must be done for each region.

### 13.3.5 L4 Interconnects Register Manual

Table 13-297 through Table 13-299 list all L4 register blocks for IA, TA, AP, and LA. Each module instance is shown with the module register mapping and bit and bit field definitions.

#### 13.3.5.1 L4 Interconnects Instance Summary

**Table 13-297. L4\_PER Instance Summary**

Module Name	L3 Base Address	Size
PER_AP	0x4800 0000	2KB
PER_IA_0	0x4800 0800	2KB
PER_LA	0x4800 1000	4KB
PER_TA_UART3	0x4802 1000	4KB
PER_TA_GPTIMER2	0x4803 3000	4KB
PER_TA_GPTIMER3	0x4803 5000	4KB
PER_TA_GPTIMER4	0x4803 7000	4KB
PER_TA_GPTIMER9	0x4803 F000	4KB
PER_TA_DSS	0x4805 0000	4KB
PER_TA_GPIO2	0x4805 6000	4KB
PER_TA_GPIO3	0x4805 8000	4KB
PER_TA_GPIO4	0x4805 A000	4KB
PER_TA_GPIO5	0x4805 C000	4KB
PER_TA_GPIO6	0x4805 E000	4KB
PER_TA_I2C3	0x4806 1000	4KB
PER_TA_UART1	0x4806 B000	4KB
PER_TA_UART2	0x4806 D000	4KB
PER_TA_UART4	0x4806 F000	4KB
PER_TA_I2C1	0x4807 1000	4KB
PER_TA_I2C2	0x4807 3000	4KB
PER_TA_SLIMBUS2	0x4807 7000	4KB
PER_TA_ELM	0x4807 9000	4KB
PER_TA_GPTIMER10	0x4808 7000	4KB
PER_TA_GPTIMER11	0x4808 9000	4KB
PER_TA_MCBSP4	0x4809 7000	4KB
PER_TA_MCSPI1	0x4809 9000	4KB
PER_TA_MCSPI2	0x4809 B000	4KB
PER_TA_HSMMC1	0x4809 D000	4KB
PER_TA_HSMMC3	0x480A E000	4KB
PER_TA_HDQ	0x480B 3000	4KB
PER_TA_HSMMC2	0x480B 5000	4KB
PER_TA_MCSPI3	0x480B 9000	4KB
PER_TA_MCSPI4	0x480B B000	4KB
PER_TA_HSMMC4	0x480D 2000	4KB
PER_TA_HSMMC5	0x480D 6000	4KB
PER_TA_I2C4	0x4835 1000	4KB

**Table 13-298. L4\_CFG Instance Summary**

Module Name	L3 Base Address	Size
CFG_AP	0x4A00 0000	2KB
CFG_IA_0	0x4A00 0800	2KB

**Table 13-298. L4\_CFG Instance Summary (continued)**

Module Name	L3 Base Address	Size
CFG_LA	0x4A00 1000	4KB
CFG_TA_SYSCTRL_GENERAL_CORE	0x4A00 3000	4KB
CFG_TA_CM1	0x4A00 5000	4KB
CFG_TA_CM2	0x4A00 A000	4KB
CFG_TA_SDMA	0x4A05 7000	4KB
CFG_TA_HSI	0x4A05 C000	4KB
CFG_TA_SAR_ROM	0x4A06 0000	4KB
CFG_TA_HSUSBTLL	0x4A06 3000	4KB
CFG_TA_USBHOSTHS	0x4A06 5000	4KB
CFG_TA_DSP	0x4A06 7000	4KB
CFG_TA_USBFS	0x4A0A A000	4KB
CFG_TA_USBOTGHS	0x4A0A C000	4KB
CFG_TA_USBPHY	0x4A0A E000	4KB
CFG_TA_SR1	0x4A0D A000	4KB
CFG_TA_SR2	0x4A0DC000	4KB
CFG_TA_SR3	0x4A0D E000	4KB
CFG_TA_MAILBOX	0x4A0F 5000	4KB
CFG_TA_SPINLOCK	0x4A0F 7000	4KB
CFG_TA_SYSCTRL_PADCONF_CORE	0x4A10 1000	4KB
CFG_TA_FACEDETECT	0x4A10 B000	4KB
CFG_TA_EMIFFW	0x4A20 D000	4KB
CFG_TA_GPMCFW	0x4A21 1000	4KB
CFG_TA_OCMCRAMFW	0x4A21 3000	4KB
CFG_TA_SGXFW	0x4A21 5000	4KB
CFG_TA_ISSF	0x4A21 7000	4KB
CFG_TA_CORTEXM3FW	0x4A21 9000	4KB
CFG_TA_DSSF	0x4A21 D000	4KB
CFG_TA_SL2FW	0x4A21 F000	4KB
CFG_TA_IVAHDFW	0x4A22 1000	4KB
CFG_TA_EMUSSF	0x4A22 7000	4KB
CFG_TA_ABEFW	0x4A22 9000	4KB
CFG_TA_L4WKUP	0x4A34 0000	4KB

**Table 13-299. L4\_WKUP Instance Summary**

Module Name	L3 Base Address	Size
WKUP_IA_0	0x4A30 0800	2KB
WKUP_LA	0x4A30 1000	4KB
WKUP_TA_32KTIMER	0x4A30 5000	4KB
WKUP_TA_PRM	0x4A30 8000	4KB
WKUP_TA_SCRM	0x4A30 B000	4KB
WKUP_TA_SYSCTRL_GENERAL_WKUP	0x4A30 D000	4KB
WKUP_TA_GPIO1	0x4A31 1000	4KB
WKUP_TA_WDTIMER2	0x4A31 5000	4KB
WKUP_TA_DM_TIMER1MS_1	0x4A31 9000	4KB
WKUP_TA_KEYBOARD	0x4A31 D000	4KB
WKUP_TA_SYSCTRL_PADCONF_WKUP	0x4A31 F000	4KB
WKUP_TA_SAR_RAM	0x4A32 A000	4KB



### 13.3.5.2 L4 Initiator Agent (L4 IA)

#### 13.3.5.2.1 L4 Initiator Agent (L4 IA) Register Summary

Table 13-300 summarizes the L4 IA register mapping.

**Table 13-300. IA Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	PER_IA_0 L3 Physical Address	CFG_IA_0 L3 Physical Address	WKUP_IA_0 L3 Physical Address
L4_IA_COMPONENT_L	R	32	0x0000 0000	0x4800 0800	0x4A00 0800	0x4A30 0800
L4_IA_COMPONENT_H	R	32	0x0000 0004	0x4800 0804	0x4A00 0804	0x4A30 0804
L4_IA_CORE_L	R	32	0x0000 0018	0x4800 0818	0x4A00 0818	0x4A30 0818
L4_IA_CORE_H	R	32	0x0000 001C	0x4800 081C	0x4A00 081C	0x4A30 081C
L4_IA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4800 0820	0x4A00 0820	0x4A30 0820
L4_IA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4800 0824	0x4A00 0824	0x4A30 0824
L4_IA_AGENT_STATUS_L	RW	32	0x0000 0028	0x4800 0828	0x4A00 0828	0x4A30 0828
L4_IA_AGENT_STATUS_H	R	32	0x0000 002C	0x4800 082C	0x4A00 082C	0x4A30 082C
L4_IA_ERROR_LOG_L	RW	32	0x0000 0058	0x4800 0858	0x4A00 0858	0x4A30 0858
L4_IA_ERROR_LOG_H	R	32	0x0000 005C	0x4800 085C	0x4A00 085C	0x4A30 085C
L4_IA_ERROR_LOG_ADDR_L	R	32	0x0000 0060	0x4800 0860	0x4A00 0860	0x4A30 0860
L4_IA_ERROR_LOG_ADDR_H	R	32	0x0000 0064	0x4800 0864	0x4A00 0864	0x4A30 0864

#### 13.3.5.2.2 L4 Initiator Agent (L4 IA) Register Description

Table 13-301 through Table 13-323 describe the L4 IA registers.

**Table 13-301. L4\_IA\_COMPONENT\_L**

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See Table 13-300.
<b>Description</b>	COMPONENT register identifies the component to which this register block belongs. The register contains a component code and revision, which are used to identify the hardware of the component. The COMPONENT register is read-only.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code	R	See <sup>(1)</sup> .
15:0	REV	Component revision code	R	See <sup>(1)</sup> .

<sup>(1)</sup> TI Internal Data

**Table 13-302. Register Call Summary for Register L4\_IA\_COMPONENT\_L**

L4 Interconnects

- [L4 Initiator Agent \(L4 IA\): \[0\]](#)

**Table 13-303. L4\_IA\_COMPONENT\_H**

<b>Address Offset</b>	0x0000 0004
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	COMPONENT register identifies the component to which this register block belongs. The register contains a component code and revision, which are used to identify the hardware of the component. The COMPONENT register is read-only.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000

**Table 13-304. Register Call Summary for Register L4\_IA\_COMPONENT\_H**

L4 Interconnects

- [L4 Initiator Agent \(L4 IA\): \[0\]](#)

**Table 13-305. L4\_IA\_CORE\_L**

<b>Address Offset</b>	0x0000 0018
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Provide information about the core initiator
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CODE																CORE_REV															

Bits	Field Name	Description	Type	Reset
31:16	CORE_CODE	Interconnect core code	R	See <sup>(1)</sup> .
15:0	CORE_REV	Component revision code code	R	See <sup>(1)</sup> .

<sup>(1)</sup> TI Internal Data**Table 13-306. Register Call Summary for Register L4\_IA\_CORE\_L**

L4 Interconnects

- [L4 Initiator Agent \(L4 IA\): \[0\]](#)

**Table 13-307. L4\_IA\_CORE\_H**

<b>Address Offset</b>	0x0000 001C
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Provide information about the core initiator
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VENDOR_CODE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	VENDOR_CODE	Vendor revision core code	R	See <sup>(1)</sup> .

<sup>(1)</sup> TI Internal Data

**Table 13-308. Register Call Summary for Register L4\_IA\_CORE\_H**

L4 Interconnects

- [L4 Initiator Agent \(L4 IA\): \[0\]](#)

**Table 13-309. L4\_IA\_AGENT\_CONTROL\_L**

<b>Address Offset</b>	0x0000 0020
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Enable error reporting on an initiator interface. The error reporting mechanism is enabled when the INBAND_ERROR_REP bit field is set to 1. The out-of-band OCP MError reporting mechanism is enabled when the MERROR_REP bit field is set to 1.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROT_ERROR_SECONDARY_REP	PROT_ERROR_PRIMARY_REP	RESERVED				INBAND_ERROR_REP		RESERVED																							

Bits	Field Name	Description	Type	Reset
31	PROT_ERROR_SECONDARY_REP	Out-of-band reporting of protection mechanism secondary errors	RW	1
30	PROT_ERROR_PRIMARY_REP	Out-of-band reporting of protection mechanism primary errors	RW	1
29:28	RESERVED	Read returns 0.	R	0x0
27	INBAND_ERROR_REP	Setting this field to 1 reports on in-band errors using the INBAND_ERROR log bit of IA.AGENT_STATUS register.	RW	1
26:0	RESERVED	Read returns 0.	R	0x0

**Table 13-310. Register Call Summary for Register L4\_IA\_AGENT\_CONTROL\_L**

L4 Interconnects

- [L4 Error Detection and Reporting: \[0\]](#)
- [L4-Interconnect Low-level Programming Models: \[1\]](#)
- [L4 Initiator Agent \(L4 IA\): \[2\]](#)

**Table 13-311. L4\_IA\_AGENT\_CONTROL\_H**

<b>Address Offset</b>	0x0000 0024
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Enable error reporting on an initiator interface.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

**Table 13-312. Register Call Summary for Register L4\_IA\_AGENT\_CONTROL\_H**

L4 Interconnects

- [L4 Initiator Agent \(L4 IA\): \[0\]](#)

**Table 13-313. L4\_IA\_AGENT\_STATUS\_L**

<b>Address Offset</b>	0x0000 0028
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Stores status information for an initiator. The INBAND_ERROR and MERROR fields are read/write and are implemented as log bits.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROT_ERROR_SECONDARY	PROT_ERROR_PRIMARY	RESERVED		INBAND_ERROR	RESERVED																										

Bits	Field Name	Description	Type	Reset
31	PROT_ERROR_SECONDARY	0x0: Secondary Protection error not present.0x1: Secondary Protection error present	RW W1toClr	0
30	PROT_ERROR_PRIMARY	0x0: Primary Protection error not present.0x1: Primary Protection error present	RW W1toClr	0
29:28	RESERVED	Read returns 0.	R	0x0
27	INBAND_ERROR	0x0 No In-Band error present.0x1 In-Band error present.	RW W1toClr	0
26:0	RESERVED	Read returns 0.	R	0x0

**Table 13-314. Register Call Summary for Register L4\_IA\_AGENT\_STATUS\_L**

L4 Interconnects

- [L4 Error Detection and Reporting](#): [0] [1] [2] [3]
- [L4-Interconnect Low-level Programming Models](#): [4] [5] [6] [7] [8] [9]
- [L4 Initiator Agent \(L4 IA\)](#): [10]

**Table 13-315. L4\_IA\_AGENT\_STATUS\_H**

<b>Address Offset</b>	0x0000 002C
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Stores status information for an initiator.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

**Table 13-316. Register Call Summary for Register L4\_IA\_AGENT\_STATUS\_H**

L4 Interconnects

- [L4 Initiator Agent \(L4 IA\)](#): [0]

**Table 13-317. L4\_IA\_ERROR\_LOG\_L**

<b>Address Offset</b>	0x0000 0058
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Log information about error conditions. The CODE field logs any protection violation or address hole errors detected by the initiator subsystem while decoding a request.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MULTI	SECONDARY	RESERVED				CODE	RESERVED										CONNID				RESERVED				CMD						

Bits	Field Name	Description	Type	Reset
31	MULTI	Multiple errors detected	RW W1toClr	0
30	SECONDARY	Indicates whether protection violation was a primary or secondary error	RW W1toClr	0
29:26	RESERVED	Read returns 0.	R	0x0
25:24	CODE	The error code of an initiator request. 0x00: No errors 0x01: Reserved 0x10: Address hole 0x11: Protection violation	RW W1toClr	0x0
23:14	RESERVED	Read returns 0.	R	0x000
13:8	CONNID	ConnID of request causing the error, refer to <a href="#">Table 13-280</a>	R	0x00
7:3	RESERVED	Read returns 0.	R	0x00
2:0	CMD	Command that caused error	R	0x0

**Table 13-318. Register Call Summary for Register L4\_IA\_ERROR\_LOG\_L**

## L4 Interconnects

- [L4 Error Detection and Reporting: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [L4-Interconnect Low-level Programming Models: \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [L4 Initiator Agent \(L4 IA\): \[18\]](#)

**Table 13-319. L4\_IA\_ERROR\_LOG\_H**

<b>Address Offset</b>	0x0000 005C
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Log information about error conditions.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REQ_INFO															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Read returns 0.	R	0x0000
15:0	REQ_INFO	MReqInfo bits of request that caused the error REQ_INFO[0] = supervisor, REQ_INFO[1] = Debug	R	0x0000

**Table 13-320. Register Call Summary for Register L4\_IA\_ERROR\_LOG\_H**

## L4 Interconnects

- [L4 Error Detection and Reporting: \[0\]](#)
- [L4 Initiator Agent \(L4 IA\): \[1\]](#)

**Table 13-321. L4\_IA\_ERROR\_LOG\_ADDR\_L**

<b>Address Offset</b>	0x0000 0060
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Extended error log (address information)
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Address of request that caused the error. N is the number MAddr bits.	R	0x0000 0000

**Table 13-322. Register Call Summary for Register L4\_IA\_ERROR\_LOG\_ADDR\_L**

## L4 Interconnects

- [L4 Error Detection and Reporting: \[0\]](#)
- [L4-Interconnect Low-level Programming Models: \[1\] \[2\] \[3\]](#)
- [L4 Initiator Agent \(L4 IA\): \[4\]](#)

**Table 13-323. L4\_IA\_ERROR\_LOG\_ADDR\_H**

<b>Address Offset</b>	0x0000 0064
<b>Physical Address</b>	See <a href="#">Table 13-300</a> .
<b>Description</b>	Extended error log (address information)
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

**Table 13-324. Register Call Summary for Register L4\_IA\_ERROR\_LOG\_ADDR\_H**

L4 Interconnects

- [L4 Initiator Agent \(L4 IA\): \[0\]](#)

### 13.3.5.3 L4 Target Agent (L4 TA)

#### 13.3.5.3.1 L4 Target Agent (L4 TA) Register Summary

[Table 13-325](#) through [Table 13-355](#) summarizes the L4 TA mapping of the CFG\_TA, PER\_TA, and WKUP\_TA registers.

**Table 13-325. CFG\_TA Register Mapping Summary 1**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_SYCTRL_GEN ERAL_CORE L3 Physical Address	CFG_TA_CM1 L3 Physical Address	CFG_TA_CM2 L3 Physical Address
<a href="#">L4_TA_COMPO NENT_H</a>	R	32	0x0000 0000	0x4A00 3000	0x4A00 5000	0x4A00 A000
<a href="#">L4_TA_COMPO NENT_L</a>	R	32	0x0000 0004	0x4A00 3000	0x4A00 5000	0x4A00 A000
<a href="#">L4_TA_CORE_L</a>	R	32	0x0000 0018	0x4A00 3018	0x4A00 5018	0x4A00 A018
<a href="#">L4_TA_CORE_H</a>	R	32	0x0000 001C	0x4A00 301C	0x4A00 501C	0x4A00 A01C
<a href="#">L4_TA_AGENT_CONTROL_L</a>	RW	32	0x0000 0020	0x4A00 3020	0x4A00 5020	0x4A00 A020
<a href="#">L4_TA_AGENT_CONTROL_H</a>	R	32	0x0000 0024	0x4A00 3024	0x4A00 5024	0x4A00 A024
<a href="#">L4_TA_AGENT_STATUS_L</a>	R	32	0x0000 0028	0x4A00 3028	0x4A00 5028	0x4A00 A028
<a href="#">L4_TA_AGENT_STATUS_H</a>	R	32	0x0000 002C	0x4A00 302C	0x4A00 502C	0x4A00 A02C

**Table 13-326. CFG\_TA Register Mapping Summary 2**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_SDMA L3 Physical Address	CFG_TA_HSI L3 Physical Address	CFG_TA_SAR_ROM L3 Physical Address
<a href="#">L4_TA_COMPO NENT_H</a>	R	32	0x0000 0000	0x4A05 7000	0x4A05 C000	0x4A06 0000
<a href="#">L4_TA_COMPO NENT_L</a>	R	32	0x0000 0004	0x4A05 7000	0x4A05 C000	0x4A06 0000
<a href="#">L4_TA_CORE_L</a>	R	32	0x0000 0018	0x4A05 7018	0x4A05 C018	0x4A06 0018

**Table 13-326. CFG\_TA Register Mapping Summary 2 (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_SDMA L3 Physical Address	CFG_TA_HSI L3 Physical Address	CFG_TA_SAR_ROM L3 Physical Address
L4_TA_CORE_H	R	32	0x0000 001C	0x4A05 701C	0x4A05 C01C	0x4A06 001C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A05 7020	0x4A05 C020	0x4A06 0020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A05 7024	0x4A05 C024	0x4A06 0024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A05 7028	0x4A05 C028	0x4A06 0028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A05 702C	0x4A05 C02C	0x4A06 002C

**Table 13-327. CFG\_TA Register Mapping Summary 3**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_HSUS BTLL L3 Physical Address	CFG_TA_USBH OSTHS L3 Physical Address	CFG_TA_DSP L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4A06 3000	0x4A06 5000	0x4A06 7000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4A06 3000	0x4A06 5000	0x4A06 7000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A06 3018	0x4A06 5018	0x4A06 7018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A06 301C	0x4A06 501C	0x4A06 701C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A06 3020	0x4A06 5020	0x4A06 7020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A06 3024	0x4A06 5024	0x4A06 7024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A06 3028	0x4A06 5028	0x4A06 7028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A06 302C	0x4A06 502C	0x4A06 702C

**Table 13-328. CFG\_TA Register Mapping Summary 4**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_USBF S L3 Physical Address	CFG_TA_USBO TGHS L3 Physical Address	CFG_TA_USBP HY L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4A0A A000	0x4A0A C000	0x4A0A E000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4A0A A000	0x4A0A C000	0x4A0A E000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A0A A018	0x4A0A C018	0x4A0A E018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A0A A01C	0x4A0A C01C	0x4A0A E01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A0A A020	0x4A0A C020	0x4A0A E020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A0A A024	0x4A0A C024	0x4A0A E024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A0A A028	0x4A0A C028	0x4A0A E028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A0A A02C	0x4A0A C02C	0x4A0A E02C



**Table 13-329. CFG\_TA Register Mapping Summary 5**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_SR1 L3 Physical Address	CFG_TA_SR2 L3 Physical Address	CFG_TA_SR3 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4A0D A000	0x4A0D C000	0x4A0D E000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4A0D A000	0x4A0D C000	0x4A0D E000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A0D A018	0x4A0D C018	0x4A0D E018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A0D A01C	0x4A0D C01C	0x4A0D E01C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4A0D A020	0x4A0D C020	0x4A0D E020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4A0D A024	0x4A0D C024	0x4A0D E024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4A0D A028	0x4A0D C028	0x4A0D E028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4A0D A02C	0x4A0D C02C	0x4A0D E02C

**Table 13-330. CFG\_TA Register Mapping Summary 6**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_MAILBOX L3 Physical Address	CFG_TA_SPINLOCK L3 Physical Address
L4_TA_COMPONE NT_H	R	32	0x0000 0000	0x4A0F 5000	0x4A0F 7000
L4_TA_COMPONE NT_L	R	32	0x0000 0004	0x4A0F 5000	0x4A0F 7000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A0F 5018	0x4A0F 7018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A0F 501C	0x4A0F 701C
L4_TA_AGENT_CO NTROL_L	RW	32	0x0000 0020	0x4A0F 5020	0x4A0F 7020
L4_TA_AGENT_CO NTROL_H	R	32	0x0000 0024	0x4A0F 5024	0x4A0F 7024
L4_TA_AGENT_ST ATUS_L	R	32	0x0000 0028	0x4A0F 5028	0x4A0F 7028
L4_TA_AGENT_ST ATUS_H	R	32	0x0000 002C	0x4A0F 502C	0x4A0F 702C

**Table 13-331. CFG\_TA Register Mapping Summary 7**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_SYSCTRL_ PADCONF_CORE L3 Physical Address
L4_TA_COMPONENT_ H	R	32	0x0000 0000	0x4A10 1000
L4_TA_COMPONENT_ L	R	32	0x0000 0004	0x4A10 1000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A10 1018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A10 101C
L4_TA_AGENT_CONTR OL_L	RW	32	0x0000 0020	0x4A10 1020
L4_TA_AGENT_CONTR OL_H	R	32	0x0000 0024	0x4A10 1024
L4_TA_AGENT_STATU S_L	R	32	0x0000 0028	0x4A10 1028
L4_TA_AGENT_STATU S_H	R	32	0x0000 002C	0x4A10 102C

**Table 13-332. CFG\_TA Register Mapping Summary 8**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_FACEDETECT L3 Physical Address
L4_TA_COMPONENT_H	R	32	0x0000 0000	0x4A10 B000
L4_TA_COMPONENT_L	R	32	0x0000 0004	0x4A10 B000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A10 B018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A10 B01C
L4_TA_AGENT_CONTR_OL_L	RW	32	0x0000 0020	0x4A10 B020
L4_TA_AGENT_CONTR_OL_H	R	32	0x0000 0024	0x4A10 B024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A10 B028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A10 B02C

**Table 13-333. CFG\_TA Register Mapping Summary 9**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_EMIFFW L3 Physical Address	CFG_TA_GPMCFW L3 Physical Address
L4_TA_COMPONENT_H	R	32	0x0000 0000	0x4A20 D000	0x4A21 1000
L4_TA_COMPONENT_L	R	32	0x0000 0004	0x4A20 D000	0x4A21 1000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A20 D018	0x4A21 1018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A20 D01C	0x4A21 101C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A20 D020	0x4A21 1020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A20 D024	0x4A21 1024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A20 D028	0x4A21 1028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A20 D02C	0x4A21 102C

**Table 13-334. CFG\_TA Register Mapping Summary 10**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_OCMC RAMFW L3 Physical Address	CFG_TA_SGXFW L3 Physical Address	CFG_TA_ISSFW L3 Physical Address
L4_TA_COMPONENT_H	R	32	0x0000 0000	0x4A21 3000	0x4A21 5000	0x4A21 7000
L4_TA_COMPONENT_L	R	32	0x0000 0004	0x4A21 3000	0x4A21 5000	0x4A21 7000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A21 3018	0x4A21 5018	0x4A21 7018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A21 301C	0x4A21 501C	0x4A21 701C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A21 3020	0x4A21 5020	0x4A21 7020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A21 3024	0x4A21 5024	0x4A21 7024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A21 3028	0x4A21 5028	0x4A21 7028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A21 302C	0x4A21 502C	0x4A21 702C

**Table 13-335. CFG\_TA Register Mapping Summary 11**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_CORT EXM3FW L3 Physical Address	CFG_TA_DSSFW L3 Physical Address	CFG_TA_SL2FW L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4A21 9000	0x4A21 D000	0x4A21 F000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4A21 9000	0x4A21 D000	0x4A21 F000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A21 9018	0x4A21 D018	0x4A21 F018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A21 901C	0x4A21 D01C	0x4A21 F01C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4A21 9020	0x4A21 D020	0x4A21 F020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4A21 9024	0x4A21 D024	0x4A21 F024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4A21 9028	0x4A21 D028	0x4A21 F028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4A21 902C	0x4A21 D02C	0x4A21 F02C

**Table 13-336. CFG\_TA Register Mapping Summary 12**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_IVAHD FW L3 Physical Address	CFG_TA_EMUS SFW L3 Physical Address	CFG_TA_ABEF W L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4A22 1000	0x4A22 7000	0x4A22 9000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4A22 1000	0x4A22 7000	0x4A22 9000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A22 1018	0x4A22 7018	0x4A22 9018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A22 101C	0x4A22 701C	0x4A22 901C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4A22 1020	0x4A22 7020	0x4A22 9020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4A22 1024	0x4A22 7024	0x4A22 9024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4A22 1028	0x4A22 7028	0x4A22 9028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4A22 102C	0x4A22 702C	0x4A22 902C

**Table 13-337. CFG\_TA Register Mapping Summary 13**

Register Name	Type	Register Width (Bits)	Address Offset	CFG_TA_L4WKUP L3 Physical Address
L4_TA_COMPONENT_ H	R	32	0x0000 0000	0x4A34 0000
L4_TA_COMPONENT_ L	R	32	0x0000 0004	0x4A34 0000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A34 0018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A34 001C
L4_TA_AGENT_CONTR OL_L	RW	32	0x0000 0020	0x4A34 0020
L4_TA_AGENT_CONTR OL_H	R	32	0x0000 0024	0x4A34 0024
L4_TA_AGENT_STATU S_L	R	32	0x0000 0028	0x4A34 0028
L4_TA_AGENT_STATU S_H	R	32	0x0000 002C	0x4A34 002C

**Table 13-338. PER\_TA Register Mapping Summary 1**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_UART 3 L3 Physical Address	PER_TA_GPTIM ER2 L3 Physical Address	PER_TA_GPTIM ER3 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4802 1000	0x4803 3000	0x4803 5000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4802 1000	0x4803 3000	0x4803 5000
L4_TA_CORE_L	R	32	0x0000 0018	0x4802 1018	0x4803 3018	0x4803 5018
L4_TA_CORE_H	R	32	0x0000 001C	0x4802 101C	0x4803 301C	0x4803 501C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4802 1020	0x4803 3020	0x4803 5020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4802 1024	0x4803 3024	0x4803 5024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4802 1028	0x4803 3028	0x4803 5028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4802 102C	0x4803 302C	0x4803 502C

**Table 13-339. PER\_TA Register Mapping Summary 2**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_GPTIM ER4 L3 Physical Address	PER_TA_GPTIM ER9 L3 Physical Address	PER_TA_DSS L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4803 7000	0x4803 F000	0x4805 0000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4803 7000	0x4803 F000	0x4805 0000
L4_TA_CORE_L	R	32	0x0000 0018	0x4803 7018	0x4803 F018	0x4805 0018
L4_TA_CORE_H	R	32	0x0000 001C	0x4803 701C	0x4803 F01C	0x4805 001C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4803 7020	0x4803 F020	0x4805 0020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4803 7024	0x4803 F024	0x4805 0024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4803 7028	0x4803 F028	0x4805 0028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4803 702C	0x4803 F02C	0x4805 002C

**Table 13-340. PER\_TA Register Mapping Summary 3**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_GPIO2 L3 Physical Address	PER_TA_GPIO3 L3 Physical Address	PER_TA_GPIO4 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4805 6000	0x4805 8000	0x4805 A000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4805 6000	0x4805 8000	0x4805 A000
L4_TA_CORE_L	R	32	0x0000 0018	0x4805 6018	0x4805 8018	0x4805 A018
L4_TA_CORE_H	R	32	0x0000 001C	0x4805 601C	0x4805 801C	0x4805 A01C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4805 6020	0x4805 8020	0x4805 A020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4805 6024	0x4805 8024	0x4805 A024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4805 6028	0x4805 8028	0x4805 A028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4805 602C	0x4805 802C	0x4805 A02C

**Table 13-341. PER\_TA Register Mapping Summary 4**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_GPIO5 L3 Physical Address	PER_TA_GPIO6 L3 Physical Address	PER_TA_I2C3 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4805 C000	0x4805 E000	0x4806 1000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4805 C000	0x4805 E000	0x4806 1000
L4_TA_CORE_L	R	32	0x0000 0018	0x4805 C018	0x4805 E018	0x4806 1018
L4_TA_CORE_H	R	32	0x0000 001C	0x4805 C01C	0x4805 E01C	0x4806 101C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4805 C020	0x4805 E020	0x4806 1020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4805 C024	0x4805 E024	0x4806 1024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4805 C028	0x4805 E028	0x4806 1028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4805 C02C	0x4805 E02C	0x4806 102C

**Table 13-342. PER\_TA Register Mapping Summary 5**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_UART 1 L3 Physical Address	PER_TA_UART 2 L3 Physical Address	PER_TA_UART 4 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4806 B000	0x4806 D000	0x4806 F000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4806 B000	0x4806 D000	0x4806 F000
L4_TA_CORE_L	R	32	0x0000 0018	0x4806 B018	0x4806 D018	0x4806 F018
L4_TA_CORE_H	R	32	0x0000 001C	0x4806 B01C	0x4806 D01C	0x4806 F01C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4806 B020	0x4806 D020	0x4806 F020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4806 B024	0x4806 D024	0x4806 F024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4806 B028	0x4806 D028	0x4806 F028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4806 B02C	0x4806 D02C	0x4806 F02C

**Table 13-343. PER\_TA Register Mapping Summary 6**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_I2C1 L3 Physical Address	PER_TA_I2C2 L3 Physical Address
L4_TA_COMPONE NT_H	R	32	0x0000 0000	0x4807 1000	0x4807 3000
L4_TA_COMPONE NT_L	R	32	0x0000 0004	0x4807 1000	0x4807 3000
L4_TA_CORE_L	R	32	0x0000 0018	0x4807 1018	0x4807 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4807 101C	0x4807 301C
L4_TA_AGENT_ CO NTROL_L	RW	32	0x0000 0020	0x4807 1020	0x4807 3020
L4_TA_AGENT_ CO NTROL_H	R	32	0x0000 0024	0x4807 1024	0x4807 3024
L4_TA_AGENT_ ST ATUS_L	R	32	0x0000 0028	0x4807 1028	0x4807 3028
L4_TA_AGENT_ ST ATUS_H	R	32	0x0000 002C	0x4807 102C	0x4807 302C

**Table 13-344. PER\_TA Register Mapping Summary 7**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_SLIMB US2 L3 Physical Address	PER_TA_ELM L3 Physical Address	PER_TA_GPTIM ER10 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4807 7000	0x4807 9000	0x4808 7000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4807 7000	0x4807 9000	0x4808 7000
L4_TA_CORE_L	R	32	0x0000 0018	0x4807 7018	0x4807 9018	0x4808 7018
L4_TA_CORE_H	R	32	0x0000 001C	0x4807 701C	0x4807 901C	0x4808 701C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4807 7020	0x4807 9020	0x4808 7020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4807 7024	0x4807 9024	0x4808 7024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4807 7028	0x4807 9028	0x4808 7028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4807 702C	0x4807 902C	0x4808 702C

**Table 13-345. PER\_TA Register Mapping Summary 8**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_GPTIM ER11 L3 Physical Address	PER_TA_MCBS P4 L3 Physical Address	PER_TA_MCSP1 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4808 9000	0x4809 7000	0x4809 9000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4808 9000	0x4809 7000	0x4809 9000
L4_TA_CORE_L	R	32	0x0000 0018	0x4808 9018	0x4809 7018	0x4809 9018
L4_TA_CORE_H	R	32	0x0000 001C	0x4808 901C	0x4809 701C	0x4809 901C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4808 9020	0x4809 7020	0x4809 9020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4808 9024	0x4809 7024	0x4809 9024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4808 9028	0x4809 7028	0x4809 9028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4808 902C	0x4809 702C	0x4809 902C

**Table 13-346. PER\_TA Register Mapping Summary 9**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_MCSP2 L3 Physical Address	PER_TA_HSMMC1 L3 Physical Address
L4_TA_COMPONE NT_H	R	32	0x0000 0000	0x4809 B000	0x4809 D000
L4_TA_COMPONE NT_L	R	32	0x0000 0004	0x4809 B000	0x4809 D000
L4_TA_CORE_L	R	32	0x0000 0018	0x4809 B018	0x4809 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4809 B01C	0x4809 D01C
L4_TA_AGENT_CO NTROL_L	RW	32	0x0000 0020	0x4809 B020	0x4809 D020
L4_TA_AGENT_CO NTROL_H	R	32	0x0000 0024	0x4809 B024	0x4809 D024
L4_TA_AGENT_ST ATUS_L	R	32	0x0000 0028	0x4809 B028	0x4809 D028
L4_TA_AGENT_ST ATUS_H	R	32	0x0000 002C	0x4809 B02C	0x4809 D02C

**Table 13-347. PER\_TA Register Mapping Summary 10**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_HSMC3 L3 Physical Address
L4_TA_COMPONENT_H	R	32	0x0000 0000	0x480A E000
L4_TA_COMPONENT_L	R	32	0x0000 0004	0x480A E000
L4_TA_CORE_L	R	32	0x0000 0018	0x480A E018
L4_TA_CORE_H	R	32	0x0000 001C	0x480A E01C
L4_TA_AGENT_CONTR_OL_L	RW	32	0x0000 0020	0x480A E020
L4_TA_AGENT_CONTR_OL_H	R	32	0x0000 0024	0x480A E024
L4_TA_AGENT_STATU_S_L	R	32	0x0000 0028	0x480A E028
L4_TA_AGENT_STATU_S_H	R	32	0x0000 002C	0x480A E02C

**Table 13-348. PER\_TA Register Mapping Summary 11**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_HDQ L3 Physical Address	PER_TA_HSMM C2 L3 Physical Address	PER_TA_MCSP1 3 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x480B 3000	0x480B 5000	0x480B 9000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x480B 3000	0x480B 5000	0x480B 9000
L4_TA_CORE_L	R	32	0x0000 0018	0x480B 3018	0x480B 5018	0x480B 9018
L4_TA_CORE_H	R	32	0x0000 001C	0x480B 301C	0x480B 501C	0x480B 901C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x480B 3020	0x480B 5020	0x480B 9020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x480B 3024	0x480B 5024	0x480B 9024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x480B 3028	0x480B 5028	0x480B 9028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x480B 302C	0x480B 502C	0x480B 902C

**Table 13-349. PER\_TA Register Mapping Summary 12**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_MCSP1 4 L3 Physical Address	PER_TA_HSMM C4 L3 Physical Address	PER_TA_HSMM C5 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x480B B000	0x480D 2000	0x480D 6000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x480B B000	0x480D 2000	0x480D 6000
L4_TA_CORE_L	R	32	0x0000 0018	0x480B B018	0x480D 2018	0x480D 6018
L4_TA_CORE_H	R	32	0x0000 001C	0x480B B01C	0x480D 201C	0x480D 601C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x480B B020	0x480D 2020	0x480D 6020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x480B B024	0x480D 2024	0x480D 6024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x480B B028	0x480D 2028	0x480D 6028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x480B B02C	0x480D 202C	0x480D 602C

**Table 13-350. PER\_TA Register Mapping Summary 13**

Register Name	Type	Register Width (Bits)	Address Offset	PER_TA_I2C4 L3 Physical Address
L4_TA_COMPONENT_H	R	32	0x0000 0000	0x4835 1000
L4_TA_COMPONENT_L	R	32	0x0000 0004	0x4835 1000
L4_TA_CORE_L	R	32	0x0000 0018	0x4835 1018
L4_TA_CORE_H	R	32	0x0000 001C	0x4835 101C
L4_TA_AGENT_CONTR_OL_L	RW	32	0x0000 0020	0x4835 1020
L4_TA_AGENT_CONTR_OL_H	R	32	0x0000 0024	0x4835 1024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4835 1028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4835 102C

**Table 13-351. WKUP\_TA Register Mapping Summary 1**

Register Name	Type	Register Width (Bits)	Address Offset	WKUP_TA_32K TIMER L3 Physical Address	WKUP_TA_PR M L3 Physical Address	WKUP_TA_SCR M L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4A30 5000	0x4A30 8000	0x4A30 B000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4A30 5000	0x4A30 8000	0x4A30 B000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A30 5018	0x4A30 8018	0x4A30 B018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A30 501C	0x4A30 801C	0x4A30 B01C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4A30 5020	0x4A30 8020	0x4A30 B020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4A30 5024	0x4A30 8024	0x4A30 B024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4A30 5028	0x4A30 8028	0x4A30 B028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4A30 502C	0x4A30 802C	0x4A30 B02C

**Table 13-352. WKUP\_TA Register Mapping Summary 2**

Register Name	Type	Register Width (Bits)	Address Offset	WKUP_TA_SYS CTRL_ GENERAL_WK UP L3 Physical Address	WKUP_TA_GPI O1 L3 Physical Address	WKUP_TA_WD TIMER2 L3 Physical Address
L4_TA_COMPO NENT_H	R	32	0x0000 0000	0x4A30 D000	0x4A31 1000	0x4A31 5000
L4_TA_COMPO NENT_L	R	32	0x0000 0004	0x4A30 D000	0x4A31 1000	0x4A31 5000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A30 D018	0x4A31 1018	0x4A31 5018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A30 D01C	0x4A31 101C	0x4A31 501C
L4_TA_AGENT_ CONTROL_L	RW	32	0x0000 0020	0x4A30 D020	0x4A31 1020	0x4A31 5020
L4_TA_AGENT_ CONTROL_H	R	32	0x0000 0024	0x4A30 D024	0x4A31 1024	0x4A31 5024
L4_TA_AGENT_ STATUS_L	R	32	0x0000 0028	0x4A30 D028	0x4A31 1028	0x4A31 5028
L4_TA_AGENT_ STATUS_H	R	32	0x0000 002C	0x4A30 D02C	0x4A31 102C	0x4A31 502C



**Table 13-353. WKUP\_TA Register Mapping Summary 3**

Register Name	Type	Register Width (Bits)	Address Offset	WKUP_TA_DM_TI MER1MS_1 L3 Physical Address	WKUP_TA_KEYBO ARD L3 Physical Address
L4_TA_COMPONE NT_H	R	32	0x0000 0000	0x4A31 9000	0x4A31 D000
L4_TA_COMPONE NT_L	R	32	0x0000 0004	0x4A31 9000	0x4A31 D000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A31 9018	0x4A31 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A31 901C	0x4A31 D01C
L4_TA_AGENT_CO NTROL_L	RW	32	0x0000 0020	0x4A31 9020	0x4A31 D020
L4_TA_AGENT_CO NTROL_H	R	32	0x0000 0024	0x4A31 9024	0x4A31 D024
L4_TA_AGENT_ST ATUS_L	R	32	0x0000 0028	0x4A31 9028	0x4A31 D028
L4_TA_AGENT_ST ATUS_H	R	32	0x0000 002C	0x4A31 902C	0x4A31 D02C

**Table 13-354. WKUP\_TA Register Mapping Summary 4**

Register Name	Type	Register Width (Bits)	Address Offset	WKUP_TA_ SYSCTRL_PADCONF_ WKUP L3 Physical Address
L4_TA_COMPONENT_ H	R	32	0x0000 0000	0x4A31 F000
L4_TA_COMPONENT_ L	R	32	0x0000 0004	0x4A31 F000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A31 F018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A31 F01C
L4_TA_AGENT_CONTR OL_L	RW	32	0x0000 0020	0x4A31 F020
L4_TA_AGENT_CONTR OL_H	R	32	0x0000 0024	0x4A31 F024
L4_TA_AGENT_STATU S_L	R	32	0x0000 0028	0x4A31 F028
L4_TA_AGENT_STATU S_H	R	32	0x0000 002C	0x4A31 F02C

**Table 13-355. WKUP\_TA Register Mapping Summary 5**

Register Name	Type	Register Width (Bits)	Address Offset	WKUP_TA_SAR_RAM L3 Physical Address
L4_TA_COMPONENT_ H	R	32	0x0000 0000	0x4A32 A000
L4_TA_COMPONENT_ L	R	32	0x0000 0004	0x4A32 A000
L4_TA_CORE_L	R	32	0x0000 0018	0x4A32 A018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A32 A01C
L4_TA_AGENT_CONTR OL_L	RW	32	0x0000 0020	0x4A32 A020
L4_TA_AGENT_CONTR OL_H	R	32	0x0000 0024	0x4A32 A024
L4_TA_AGENT_STATU S_L	R	32	0x0000 0028	0x4A32 A028
L4_TA_AGENT_STATU S_H	R	32	0x0000 002C	0x4A32 A02C

### 13.3.5.3.2 L4 Target Agent (L4 TA) Register Description

Table 13-356 through Table 13-370 describe the L4 TA registers.

**Table 13-356. L4\_TA\_COMPONENT\_H**

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See Table 13-325 to Table 13-337.
<b>Description</b>	Contains a component code and revision.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 000

**Table 13-357. Register Call Summary for Register L4\_TA\_COMPONENT\_H**

L4 Interconnects

- L4 Target Agent (L4 TA): [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30]

**Table 13-358. L4\_TA\_COMPONENT\_L**

<b>Address Offset</b>	0x0000 0004
<b>Physical Address</b>	See Table 13-325 to Table 13-337.
<b>Description</b>	Contains a component code and revision.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code.	R	See <sup>(1)</sup> .
15:0	REV	Component revision code.	R	See <sup>(1)</sup> .

<sup>(1)</sup> TI Internal Data

**Table 13-359. Register Call Summary for Register L4\_TA\_COMPONENT\_L**

L4 Interconnects

- L4 Target Agent (L4 TA): [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30]

**Table 13-360. L4\_TA\_CORE\_L**

<b>Address Offset</b>	0x0000 0018
<b>Physical Address</b>	See Table 13-325 to Table 13-337.
<b>Description</b>	Contains a component code and revision.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CODE																CORE_REV															

Bits	Field Name	Description	Type	Reset
31:16	CORE_CODE	Interconnect core code	R	See <sup>(1)</sup> .
15:0	CORE_REV	Component revision code code	R	See <sup>(1)</sup> .

<sup>(1)</sup> TI Internal Data

**Table 13-361. Register Call Summary for Register L4\_TA\_CORE\_L**

L4 Interconnects

- L4 Target Agent (L4 TA): [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30]

**Table 13-362. L4\_TA\_CORE\_H**

<b>Address Offset</b>	0x0000 001C
<b>Physical Address</b>	See Table 13-325 to Table 13-337.
<b>Description</b>	Contains a component code and revision.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VENDOR_CODE																							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	VENDOR_CODE	Vendor revision core code	R	See <sup>(1)</sup> .

<sup>(1)</sup> TI Internal Data

**Table 13-363. Register Call Summary for Register L4\_TA\_CORE\_H**

L4 Interconnects

- L4 Target Agent (L4 TA): [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30]

**Table 13-364. L4\_TA\_AGENT\_CONTROL\_L**

<b>Address Offset</b>	0x0000 0020
<b>Physical Address</b>	See Table 13-325 to Table 13-337.
<b>Description</b>	Enable error reporting
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								REQ_TIMEOUT		RESERVED								OCP_RESET					

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24	SERROR_REP	Enable logging of error	R	0x0
23:11	RESERVED	Read returns 0.	UNDEFI NED_TY PE_STRI NG	

Bits	Field Name	Description	Type	Reset
10:8	REQ_TIMEOUT	Time-out Bound. Values are: 0 - No time-out 1 - 1x base cycles. 2 - 4x base cycles. 3 - 16x base cycles. 4 - 64x base cycles.	RW	0x2
7:1	RESERVED	Read returns 0.	R	0x00
0	OCP_RESET	The OCP_RESET field controls the OCP reset signal to the attached core. Setting this bit clears any pending transfers and resets the OCP interface. The bit must be cleared to deassert the OCP reset signal. When the software reset feature is available on a target agent, the target agent OCP must also have a reset signal directed to the target core.	RW	0

**Table 13-365. Register Call Summary for Register L4\_TA\_AGENT\_CONTROL\_L**

## L4 Interconnects

- [L4 Error Detection and Reporting: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [L4-Interconnect Low-level Programming Models: \[7\] \[8\] \[9\] \[10\]](#)
- [L4 Target Agent \(L4 TA\): \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\]](#)

**Table 13-366. L4\_TA\_AGENT\_CONTROL\_H**

<b>Address Offset</b>	0x0000 0024
<b>Physical Address</b>	See <a href="#">Table 13-325</a> to <a href="#">Table 13-337</a> .
<b>Description</b>	Enable clock power management
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EXT_CLOCK	RESERVED														

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Read returns 0.	R	0x000000
8	EXT_CLOCK	When set to 1, the ext_clk_off_i signal on a target agent indicates when the target agent should shut off.	R	0
7:0	RESERVED	Read returns 0.	R	0x00

**Table 13-367. Register Call Summary for Register L4\_TA\_AGENT\_CONTROL\_H**

## L4 Interconnects

- [L4 Target Agent \(L4 TA\): \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\]](#)

**Table 13-368. L4\_TA\_AGENT\_STATUS\_L**

<b>Address Offset</b>	0x0000 0028
<b>Physical Address</b>	See <a href="#">Table 13-325</a> to <a href="#">Table 13-337</a> .
<b>Description</b>	Error reporting
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								REQ_TIMEOUT	RESERVED														

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Read returns 0.	R	0x00
23:9	RESERVED	Read returns 0.	R	0x0001
8	REQ_TIMEOUT	Time-out status: 0x0: No request time-out 0x1: A request time-out has occurred	R 1toCLR	0
7:0	RESERVED	Read returns 0.	R	0x00

**Table 13-369. Register Call Summary for Register L4\_TA\_AGENT\_STATUS\_L**

L4 Interconnects

- [L4 Error Detection and Reporting: \[0\] \[1\] \[2\]](#)
- [L4-Interconnect Low-level Programming Models: \[3\] \[4\]](#)
- [L4 Target Agent \(L4 TA\): \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\]](#)

**Table 13-370. L4\_TA\_AGENT\_STATUS\_H**

<b>Address Offset</b>	0x0000 002C
<b>Physical Address</b>	See <a href="#">Table 13-325</a> to <a href="#">Table 13-337</a> .
<b>Description</b>	Error reporting
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 000

**Table 13-371. Register Call Summary for Register L4\_TA\_AGENT\_STATUS\_H**

L4 Interconnects

- [L4 Target Agent \(L4 TA\): \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\]](#)

### 13.3.5.4 L4 Link Agent (L4 LA)

#### 13.3.5.4.1 L4 Link Agent (L4 LA) Register Summary

[Table 13-372](#) summarizes the L4 LA register mapping.

**Table 13-372. LA Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	PER_LA L3 Physical Address	CFG_LA L3 Physical Address	WKUP_LA L3 Physical Address
L4_LA_COMPO NENT_L	R	32	0x0000 0000	0x4800 1000	0x4A00 1000	0x4A30 1000
L4_LA_COMPO NENT_H	R	32	0x0000 0004	0x4800 1004	0x4A00 1004	0x4A30 1004
L4_LA_NETWO RK_L	R	32	0x0000 0010	0x4800 1010	0x4A00 1010	0x4A30 1010
L4_LA_NETWO RK_H	R	32	0x0000 0014	0x4800 1014	0x4A00 1014	0x4A30 1014
L4_LA_INITIATO R_INFO_L	R	32	0x0000 0018	0x4800 1018	0x4A00 1018	0x4A30 1018
L4_LA_INITIATO R_INFO_H	R	32	0x0000 001C	0x4800 101C	0x4A00 101C	0x4A30 101C
L4_LA_NETWO RK_CONTROL_L	RW	32	0x0000 0020	0x4800 1020	0x4A00 1020	0x4A30 1020
L4_LA_NETWO RK_CONTROL_H	RW	32	0x0000 0024	0x4800 1024	0x4A00 1024	0x4A30 1024

**13.3.5.4.2 L4 Link Agent (L4 LA) Register Description**

Table 13-373 through Table 13-389 describe the L4 LA registers.

**Table 13-373. L4\_LA\_COMPONENT\_L**

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See Table 13-372.
<b>Description</b>	Contain a component code and revision, which are used to identify the hardware of the component.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code.	R	See <sup>(1)</sup> .
15:0	REV	Component revision code.	R	See <sup>(1)</sup> .

<sup>(1)</sup> TI Internal Data

**Table 13-374. Register Call Summary for Register L4\_LA\_COMPONENT\_L**

L4 Interconnects

- L4 Link Agent (L4 LA): [0]

**Table 13-375. L4\_LA\_COMPONENT\_H**

<b>Address Offset</b>	0x0000 0004
<b>Physical Address</b>	See <a href="#">Table 13-372</a> .
<b>Description</b>	Contain a component code and revision, which are used to identify the hardware of the component.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

**Table 13-376. Register Call Summary for Register L4\_LA\_COMPONENT\_H**

- L4 Interconnects
- [L4 Link Agent \(L4 LA\): \[0\]](#)

**Table 13-377. L4\_LA\_NETWORK\_L**

<b>Address Offset</b>	0x0000 0010
<b>Physical Address</b>	See <a href="#">Table 13-372</a> .
<b>Description</b>	Identify the interconnect
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 0000

**Table 13-378. Register Call Summary for Register L4\_LA\_NETWORK\_L**

- L4 Interconnects
- [L4 Link Agent \(L4 LA\): \[0\]](#)

**Table 13-379. L4\_LA\_NETWORK\_H**

<b>Address Offset</b>	0x0000 0014
<b>Physical Address</b>	See <a href="#">Table 13-372</a> .
<b>Description</b>	Identify the interconnect
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																															

Bits	Field Name	Description	Type	Reset
31:0	ID	The ID field uniquely identifies this interconnect.	R	0x00010000

**Table 13-380. Register Call Summary for Register L4\_LA\_NETWORK\_H**

- L4 Interconnects
- [L4 Link Agent \(L4 LA\): \[0\]](#)

**Table 13-381. L4\_LA\_INITIATOR\_INFO\_L**

<b>Address Offset</b>	0x0000 0018
<b>Physical Address</b>	See <a href="#">Table 13-372</a> .
<b>Description</b>	Contain initiator subsystem information.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				PROT_GROUPS				NUMBER_REGIONS								RESERVED								SEGMENTS							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:24	PROT_GROUPS	Number of protection group of in the current L4 0x0: No protection group 0x1: 1 protection group 0x2: 2 protection groups .... 0x8: 8 protection groups 0x9 to 0xF: Reserved	R	see <a href="#">Table 13-383</a>
23:16	NUMBER_REGIONS	Number of regions in the current L4 0x0: Reserved 0x1: 1 region 0x2: 2 regions .... Max regions +1 to 0xFF: Reserved, maximum regions is listed in <a href="#">Table 13-383</a>	R	see <a href="#">Table 13-383</a>
15:4	RESERVED	Read returns 0.	R	0x000
3:0	SEGMENTS	Number of segments in the current L4 0x0: Reserved 0x1: 1 segment 0x2: 2 segments .... 0x8: 8 segments	R	see <a href="#">Table 13-383</a>

**Table 13-382. Register Call Summary for Register L4\_LA\_INITIATOR\_INFO\_L**

L4 Interconnects

- [L4 Firewalls: \[0\]](#)
- [L4 Link Agent \(L4 LA\): \[1\]](#)

**Table 13-383. Reset value for L4\_LA\_INITIATOR\_INFO\_L**

Field Name	L4 PER	L4 CFG	L4 WKUP
PROT_GROUPS	0x8	0x8	0x0
NUMBER_REGIONS	0x55	0x5C	0x20
SEGMENTS	0x2	0x5	0x3



**Table 13-384. L4\_LA\_INITIATOR\_INFO\_H**

<b>Address Offset</b>	0x0000 001C
<b>Physical Address</b>	See <a href="#">Table 13-372</a> .
<b>Description</b>	Contain initiator subsystem information.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								THREADS				RESERVED	CONNID_WIDTH		RESERVED	BYTE_DATA_WIDTH_EXP		RESERVED	ADDR_WIDTH												

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Read returns 0.	R	0x0000
18:16	THREADS	The THREADS field specifies the number of initiator threads connected to the interconnect. The field contains read-only configuration information for the initiator subsystem.	R	see <a href="#">Table 13-386</a>
15	RESERVED	Read returns 0.	R	0
14:12	CONNID_WIDTH	The initiator subsystem ConnID width. The CONNID_WIDTH field contains read-only configuration information for the initiator subsystem.	R	see <a href="#">Table 13-386</a>
11	RESERVED	Read returns 0.	R	0
10:8	BYTE_DATA_WIDTH_EXP	This field specifies the initiator subsystem data width. The BYTE_DATA_WIDTH_EXP field contains read-only configuration information for the initiator subsystem. 0x1: 16-bit data width is specified 0x2: 32-bit data width is specified	R	see <a href="#">Table 13-386</a>
7:5	RESERVED	Read returns 0.	R	0x0
4:0	ADDR_WIDTH	This field specifies the initiator subsystem address width. The ADDR_WIDTH field contains read-only configuration information for the initiator subsystem.	R	see <a href="#">Table 13-386</a>

**Table 13-385. Register Call Summary for Register L4\_LA\_INITIATOR\_INFO\_H**

L4 Interconnects

- [L4 Link Agent \(L4 LA\): \[0\]](#)

**Table 13-386. Reset value for L4\_LA\_INITIATOR\_INFO\_H**

Field Name	L4 PER	L4 CFG	L4 WKUP
THREADS	0x4	0x1	0x1
CONNID_WIDTH	0x4	0x4	0x0
BYTE_DATA_WIDTH_EXP	0x2	0x2	0x2
ADDR_WIDTH	0x18	0x18	0x12

**Table 13-387. L4\_LA\_NETWORK\_CONTROL\_L**

<b>Address Offset</b>	0x0000 0020
<b>Physical Address</b>	See <a href="#">Table 13-372</a> .
<b>Description</b>	Control interconnect minimum timeout values.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TIMEOUT_BASE								RESERVED							

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Read returns 0.	R	0x000000
10:8	TIMEOUT_BASE	The TIMEOUT_BASE field indicates the time-out period (that is, base cycles) for the highest frequency time-base signal sent from the L4 initiator subsystem to all target agents that have time-out enabled. Values for the field are: 0 - Time-out disabled 1 - L4 interconnect clock cycles divided by 64 2 - L4 interconnect clock cycles divided by 256 3 - L4 interconnect clock cycles divided by 1024 4 - L4 interconnect clock cycles divided by 4096	RW	0x4
7:0	RESERVED	Read returns 0.	R	0x00

**Table 13-388. Register Call Summary for Register L4\_LA\_NETWORK\_CONTROL\_L**

L4 Interconnects

- [L4 Error Detection and Reporting: \[0\] \[1\] \[2\]](#)
- [L4-Interconnect Low-level Programming Models: \[3\] \[4\]](#)
- [L4 Link Agent \(L4 LA\): \[5\]](#)

**Table 13-389. L4\_LA\_NETWORK\_CONTROL\_H**

<b>Address Offset</b>	0x0000 0024
<b>Physical Address</b>	See <a href="#">Table 13-372</a> .
<b>Description</b>	Control interconnect global power control
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED								CLOCK_GATE_DISABLE				RESERVED				THREAD0_PRI				RESERVED								EXT_CLOCK				RESERVED							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24	CLOCK_GATE_DISABLE	When set to 1 this field disables all clock gating.	RW	0
23:21	RESERVED	Read returns 0.	R	0x0
20	THREAD0_PRI	Sets thread priority. If the field is set to 0, the default, all initiator threads are treated the same. Setting the THREAD0_PRI field to 1 assigns a higher arbitration priority to thread 0 of the first initiator OCP interface. To avoid starvation, arbitration is imposed by the initiator subsystem. When multiple requests from different initiator threads are dispatched to targets simultaneously, the oldest request is dispatched first. If thread 0 is assigned a higher priority, a request on thread 0 always wins arbitration. Assigning thread 0 of the first initiator OCP the highest priority on a request or response can result in the starvation of other threads.	R	1
19:9	RESERVED	Read returns 0.	R	0x000
8	EXT_CLOCK	When set to 1, the ext_clk_off_i signal on the initiator subsystem instructs the entire L4 to shut off.	R	1
7:0	RESERVED	Read returns 0.	R	0x00

**Table 13-390. Register Call Summary for Register L4\_LA\_NETWORK\_CONTROL\_H**

L4 Interconnects

- L4 Link Agent (L4 LA): [0]

### 13.3.5.5 L4 Address Protection (L4 AP)

#### 13.3.5.5.1 L4 Address Protection (L4 AP) Register Summary

Table 13-391 summarizes the L4 AP register mapping.

**Table 13-391. L4 AP Register Summary**

Register Name	Type	Register Width (Bits)	Address Offset	PER_AP Physical Address	CFG_AP Physical Address
<a href="#">L4_AP_COMPONENT_L</a>	R	32	0x0000 0000	0x4800 0 000	0x4A00 0 000
<a href="#">L4_AP_COMPONENT_H</a>	R	32	0x0000 0004	0x4800 0004	0x4A00 0004
<a href="#">L4_AP_SEGMENT_i_L <sup>(1)</sup></a>	RW	32	0x0000 0100 + (0x08*i)	0x4800 0100 + (0x08*i)	0x4A00 0100 + (0x08*i)
<a href="#">L4_AP_SEGMENT_i_H <sup>(1)</sup></a>	RW	32	0x0000 0104 + (0x08*i)	0x4800 0104 + (0x08*i)	0x4A00 0104 + (0x08*i)
<a href="#">L4_AP_PROT_GROUP_MEMBERS_k_L <sup>(2)</sup></a>	R	32	0x0000 0200 + (0x08*k)	0x4800 0200 + (0x08*k)	0x4A00 0200 + (0x08*k)
<a href="#">L4_AP_PROT_GROUP_MEMBERS_k_H <sup>(2)</sup></a>	R	32	0x0000 0204 + (0x08*k)	0x4800 0204 + (0x08*k)	0x4A00 0204 + (0x08*k)
<a href="#">L4_AP_PROT_GROUP_ROLES_k_L <sup>(2)</sup></a>	R	32	0x0000 0280 + (0x08*k)	0x4800 0280 + (0x08*k)	0x4A00 0280 + (0x08*k)
<a href="#">L4_AP_PROT_GROUP_ROLES_k_H <sup>(2)</sup></a>	R	32	0x0000 0284 + (0x08*k)	0x4800 0284 + (0x08*k)	0x4A00 0284 + (0x08*k)
<a href="#">L4_AP_REGION_l_L <sup>(3)</sup></a>	RW	32	0x0000 0300 + (0x08*l)	0x4800 0300 + (0x08*l)	0x4A00 0300 + (0x08*l)
<a href="#">L4_AP_REGION_l_H <sup>(3)</sup></a>	RW	32	0x0000 0304 + (0x08*l)	0x4800 0304 + (0x08*l)	0x4A00 0304 + (0x08*l)

<sup>(1)</sup> i = 0 to 1 for PER\_AP  
i = 0 to 6 for CFG\_AP

<sup>(2)</sup> k = 0 to 7 for PER\_AP  
k = 0 to 7 for CFG\_AP

<sup>(3)</sup> l = 0 to 96 for PER\_AP  
l = 0 to 113 for CFG\_AP

### 13.3.5.5.2 L4 Address Protection (L4 AP) Register Description

Table 13-392 through Table 13-416 describe the L4 AP registers.

**Table 13-392. L4\_AP\_COMPONENT\_L**

<b>Address Offset</b>	0x000
<b>Physical address</b>	See Table 13-391.
<b>Description</b>	Contains a component code and revision, which are used to identify the hardware of the component.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code	R	See <sup>(1)</sup> .
15:0	REV	Component revision code	R	See <sup>(1)</sup> .

<sup>(1)</sup> TI Internal Data

**Table 13-393. Register Call Summary for Register L4\_AP\_COMPONENT\_L**

L4 Interconnects

- L4 Address Protection (L4 AP): [0]

**Table 13-394. L4\_AP\_COMPONENT\_H**

<b>Address Offset</b>	0x004
<b>Physical address</b>	See Table 13-391.
<b>Description</b>	Contains a component code and revision, which are used to identify the hardware of the component.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															

Bits	Field Name	Description	Type	Reset
31:0	Reserved	Read returns 0	R	0x0000 0000

**Table 13-395. Register Call Summary for Register L4\_AP\_COMPONENT\_H**

L4 Interconnects

- L4 Address Protection (L4 AP): [0]

**Table 13-396. L4\_AP\_SEGMENT\_i\_L**

<b>Address Offset</b>	0x100 + (0x08*i)	<b>Index</b>	i = 0 to 1 for PER_AP i = 0 to 6 for CFG_AP
<b>Physical address</b>	See Table 13-391.		
<b>Description</b>	Define the base address of each segments		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BASE																							

Bits	Field Name	Description	Type	Reset
31:24	Reserved	Read returns 0.	R	0x00
23:0	BASE	The base address of the segment (with 0s from bit 0 to bit SIZE-1).	R	see <a href="#">Table 13-400</a>

**Table 13-397. Register Call Summary for Register L4\_AP\_SEGMENT\_i\_L**

L4 Interconnects

- [L4 Firewalls: \[0\]](#)
- [L4 Address Protection \(L4 AP\): \[1\]](#)

**Table 13-398. L4\_AP\_SEGMENT\_i\_H**

<b>Address Offset</b>	0x104 + (0x08*i)	<b>Index</b>	i = 0 to 1 for PER_AP i = 0 to 6 for CFG_AP
<b>Physical address</b>	See <a href="#">Table 13-391</a> .		
<b>Description</b>	Define the size of each segments		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								SIZE							

Bits	Field Name	Description	Type	Reset
31:5	Reserved	Read returns 0.	R	0x0000000
4:0	SIZE	Segment size is a power of 2, where 2 <sup>SIZE</sup> is the byte size of a segment (all segment registers use the same size).	R	see <a href="#">Table 13-400</a>

**Table 13-399. Register Call Summary for Register L4\_AP\_SEGMENT\_i\_H**

L4 Interconnects

- [L4 Firewalls: \[0\]](#)
- [L4 Address Protection \(L4 AP\): \[1\]](#)

**Table 13-400. Reset Value for L4\_AP\_SEGMENT\_i**

i	L4 PER		L4 CFG	
	BASE	SIZE	BASE	SIZE
0	0x00 0000	0x15	0x00 0000	0x13
1	0x02 0000	0x15	0x08 0000	0x13
2	-	-	0x10 0000	0x13
3	-	-	0x18 0000	0x13
4	-	-	0x20 0000	0x13
5	-	-	0x28 0000	0x13
6	-	-	0x30 0000	0x13

**Table 13-401. L4\_AP\_PROT\_GROUP\_MEMBERS\_k\_L**

<b>Address Offset</b>	0x200 + (0x08*k)	<b>Index</b>	k = 0 to 7 for PER_AP k = 0 to 7 for CFG_AP
<b>Physical address</b>	See <a href="#">Table 13-391</a> .		
<b>Description</b>	Define ConnID bit vectors for a protection group.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																DMM_PAGE_TABLE_WALK	PERF_PROBE	Reserved	HS_USB_HOST_USB	Reserved	MMC1_MMC2	HSI	C2C	DSS	SGX	sDMA	CAM_CORTEXA9_MPU_SS_FACE_DETECT	IVA_HD	DSP_SS	DAP	CORTEXM3_MPU_SS

Bits	Field Name	Description	Type	Reset
31:16	Reserved	Reserved	R	0x000000000000
15	DMM_PAGE_TABLE_WALK	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>
14	PERF_PROBE	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>
13	Reserved	Reserved	RW	See <a href="#">Table 13-403</a>
12	HS_USB_HOST_USB	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>
11	Reserved	Reserved	RW	See <a href="#">Table 13-403</a>
10	MMC1_MMC2	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>
9	HSI	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>
8	C2C	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>
7	DSS	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>
6	SGX	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>
5	sDMA	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>
4	CAM_CORTEXA9_MPU SS_FACE_DETECT	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>

Bits	Field Name	Description	Type	Reset
3	IVA_HD	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a> .
2	DSP_SS	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a> .
1	DAP	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a> .
0	CORTEXM3_MPU_SS	Included in the protection group k 0x0: Not a member 0x1: Member	RW	See <a href="#">Table 13-403</a>

**Table 13-402. Register Call Summary for Register L4\_AP\_PROT\_GROUP\_MEMBERS\_k\_L**

L4 Interconnects

- [L4 Firewalls: \[0\] \[1\] \[2\]](#)
- [L4-Interconnect Low-level Programming Models: \[3\]](#)
- [L4 Address Protection \(L4 AP\): \[4\]](#)

**Table 13-403. Reset Value for L4\_AP\_PROT\_GROUP\_MEMBERS\_k**

k	L4 PER		L4 CFG	
	Type	Reset Value	Type	Reset Value
0	RW	0xFFFF <sup>(1)</sup>	RW	0xFFFF <sup>(1)</sup>
1	RW	0xFFFF <sup>(1)</sup>	RW	0xFFFF <sup>(1)</sup>
2	RW	0xFFFF <sup>(1)</sup>	RW	0x0000 <sup>(1)</sup>
3	RW	0xFFFF	RW	0xFFFF <sup>(1)</sup>
4	RW	0xFFFF	RW	0xFFFF <sup>(1)</sup>
5	RW	0xFFFF	RW	0xFFFF
6	RW	0xFFFF	RW	0xFFFF
7	RW	0xFFFF	RW	0xFFFF

<sup>(1)</sup> Value exported from SCM and valid for GP device only (see *System Control Module*). The values of the other protection registers can be modified during run time.

**Table 13-404. L4\_AP\_PROT\_GROUP\_MEMBERS\_k\_H**

<b>Address Offset</b>	0x204 + (0x08*k)	<b>Index</b>	k = 0 to 7 for PER_AP k = 0 to 7 for CFG_AP
<b>Physical address</b>	See <a href="#">Table 13-391</a> .		
<b>Description</b>	Define ConnID bit vectors for a protection group.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															

Bits	Field Name	Description	Type	Reset
31:0	Reserved	Read returns 0's	R	0x0000 0000

**Table 13-405. Register Call Summary for Register L4\_AP\_PROT\_GROUP\_MEMBERS\_k\_H**

L4 Interconnects

- [L4 Address Protection \(L4 AP\): \[0\]](#)

**Table 13-406. L4\_AP\_PROT\_GROUP\_ROLES\_k\_L**

<b>Address Offset</b>	0x200 + (0x08*k)	<b>Index</b>	k = 0 to 7 for PER_AP k = 0 to 7 for CFG_AP
<b>Physical address</b>	See <a href="#">Table 13-391</a> .		
<b>Description</b>	Define MReqInfo bit vectors for a protection group.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																ENABLE															

Bits	Field Name	Description	Type	Reset
31:16	Reserved	Read returns 0	R	0x0000
15:0	ENABLE	Setting of type acces allowed for the group of initiators.	R	0xFFFF

**Table 13-407. Register Call Summary for Register L4\_AP\_PROT\_GROUP\_ROLES\_k\_L**

L4 Interconnects

- [L4 Firewalls: \[0\] \[1\]](#)
- [L4-Interconnect Low-level Programming Models: \[2\]](#)
- [L4 Address Protection \(L4 AP\): \[3\]](#)

**Table 13-408. Reset Value for L4\_AP\_PROT\_GROUP\_ROLES\_k**

k	L4_PER		L4_CFG	
	Type	Reset Value	Type	Reset Value
0	R	0xFFFF FFFF <sup>(1)</sup>	R	0xFFFF FFFF <sup>(1)</sup>
1	R	0xFFFF FFFF <sup>(1)</sup>	R	0xFFFF FFFF <sup>(1)</sup>
2	R	0xFFFF FFFF <sup>(1)</sup>	R	0x0000 0000 <sup>(1)</sup>
3	RW	0xFFFF FFFF	R	0xFFFF FFFF <sup>(1)</sup>
4	RW	0xFFFF FFFF	R	0xFFFF FFFF <sup>(1)</sup>
5	RW	0xFFFF FFFF	RW	0xFFFF FFFF
6	RW	0xFFFF FFFF	RW	0xFFFF FFFF
7	RW	0xFFFF FFFF	RW	0xFFFF FFFF

<sup>(1)</sup> Value exported from SCM and valid for GP device only (see *System Control Module*). The values of the other protection registers can be modified during run time.

**Table 13-409. L4\_AP\_PROT\_GROUP\_ROLES\_k\_H**

<b>Address Offset</b>	0x204 + (0x08*k)	<b>Index</b>	k = 0 to 7 for PER_AP k = 0 to 7 for CFG_AP
<b>Physical address</b>	See <a href="#">Table 13-391</a> .		
<b>Description</b>	Define ConnID bit vectors for a protection group.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															

Bits	Field Name	Description	Type	Reset
31:0	Reserved	Read returns 0's	R	0x0000 0000

**Table 13-410. Register Call Summary for Register L4\_AP\_PROT\_GROUP\_ROLES\_k\_H**

L4 Interconnects

- [L4 Address Protection \(L4 AP\): \[0\]](#)



**Table 13-411. L4\_AP\_REGION\_I\_L**

<b>Address Offset</b>	0x300 + (0x08*I)	<b>Index</b>	I = 0 to 96 for PER_AP I = 0 to 113 for CFG_AP
<b>Physical address</b>	See <a href="#">Table 13-391</a> .		
<b>Description</b>	Define the base address of the region in respect to the segment it belongs to.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BASE																							

Bits	Field Name	Description	Type	Reset
31:24	Reserved	Read returns 0.	R	0x00
23:0	BASE	Sets the base address of the region relative to its segment base.	R	See <a href="#">Table 13-415</a> to <a href="#">Table 13-416</a>

**Table 13-412. Register Call Summary for Register L4\_AP\_REGION\_I\_L**

L4 Interconnects

- [L4 Firewalls: \[0\]](#)
- [L4-Interconnect Low-level Programming Models: \[1\]](#)
- [L4 Address Protection \(L4 AP\): \[2\]](#)

**Table 13-413. L4\_AP\_REGION\_I\_H**

<b>Address Offset</b>	0x304 + (0x08*I)	<b>Index</b>	I = 0 to 99 for CORE_AP
<b>Physical address</b>	See <a href="#">Table 13-391</a> .		
<b>Description</b>	Define the size, protection group and segment ID of the region		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Reserved	PROT_GROUP_ID			BYTE_DATA_WIDTH_EXP				Reserved								SIZE				ENABLE			

Bits	Field Name	Description	Type	Reset
31:28	Reserved	Read returns 0	R	0x0
27:24	SEGMENT_ID	The segment associated to the region.	R	See <a href="#">Table 13-415</a> and <a href="#">Table 13-416</a> .
23	Reserved	Read returns 0.	R	0
22:20	PROT_GROUP_ID	The protection group associated to the region.	RW	See <a href="#">Table 13-415</a> and <a href="#">Table 13-416</a> .
19:17	BYTE_DATA_WIDTH_EXP	The target data byte width is 2 <sup>(BYTE_DATA_WIDTH_EXP)</sup> bytes.	R	See <a href="#">Table 13-415</a> and <a href="#">Table 13-416</a> .
16:6	Reserved	Read returns 0.	R	0x0
5:1	SIZE	Define the size of the region in bytes. 2 <sup>SIZE</sup> equals the region.	R	See <a href="#">Table 13-415</a> and <a href="#">Table 13-416</a> .
0	ENABLE	0x0: Disable the region, no access allows 0x1: Enable the region, with access as define in registers	R	0x1

**Table 13-414. Register Call Summary for Register L4\_AP\_REGION\_I\_H**

## L4 Interconnects

- L4 Firewalls: [0] [1]
- L4 Address Protection (L4 AP): [2]

**Table 13-415. L4\_AP\_REGION\_I Reset Value for L4 PER**

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	SIZE	BASE
0	0x0	0x0	0x0	0x2	0x0B	0x000000
1	0x1	0x0	0x7	0x2	0x0B	0x000800
2	0x5	0x0	0x7	0x2	0x0C	0x001000
3	0x0	0x0	0x7	0x2	0x0C	0x020000
4	0x0	0x0	0x7	0x2	0x0C	0x021000
5	0x0	0x0	0x7	0x2	0x0C	0x032000
6	0x0	0x0	0x7	0x2	0x0C	0x033000
7	0x0	0x0	0x7	0x2	0x0C	0x034000
8	0x0	0x0	0x7	0x2	0x0C	0x035000
9	0x0	0x0	0x7	0x2	0x0C	0x036000
10	0x0	0x0	0x7	0x2	0x0C	0x037000
11	0x0	0x0	0x7	0x2	0x0C	0x03E000
12	0x0	0x0	0x7	0x2	0x0C	0x03F000
13	0x0	0x0	0x2	0x2	0x10	0x040000
14	0x0	0x0	0x7	0x2	0x0C	0x050000
15	0x0	0x0	0x7	0x2	0x0C	0x055000
16	0x0	0x0	0x7	0x2	0x0C	0x056000
17	0x0	0x0	0x7	0x2	0x0C	0x057000
18	0x0	0x0	0x7	0x2	0x0C	0x058000
19	0x0	0x0	0x7	0x2	0x0C	0x059000
20	0x0	0x0	0x7	0x2	0x0C	0x05A000
21	0x0	0x0	0x7	0x2	0x0C	0x05B000
22	0x0	0x0	0x7	0x2	0x0C	0x05C000
23	0x0	0x0	0x7	0x2	0x0C	0x05D000
24	0x0	0x0	0x7	0x2	0x0C	0x05E000
25	0x0	0x0	0x7	0x1	0x0C	0x060000
26	0x0	0x0	0x7	0x2	0x0C	0x06A000
27	0x0	0x0	0x7	0x2	0x0C	0x06B000
28	0x0	0x0	0x7	0x2	0x0C	0x06C000
29	0x0	0x0	0x7	0x2	0x0C	0x06D000
30	0x1	0x0	0x7	0x2	0x0C	0x06E000
31	0x0	0x0	0x7	0x2	0x0C	0x06F000
32	0x0	0x0	0x7	0x1	0x0C	0x070000
33	0x0	0x0	0x7	0x1	0x0C	0x071000
34	0x0	0x0	0x7	0x1	0x0C	0x072000
35	0x0	0x0	0x7	0x1	0x0C	0x073000
36	0x0	0x0	0x7	0x1	0x0C	0x061000
37	0x0	0x0	0x7	0x2	0x0C	0x074000
38	0x0	0x0	0x7	0x2	0x0C	0x075000
39	0x0	0x0	0x7	0x2	0x0C	0x076000
40	0x0	0x0	0x7	0x2	0x0C	0x077000
41	0x0	0x0	0x7	0x2	0x0C	0x078000

**Table 13-415. L4\_AP\_REGION\_I Reset Value for L4 PER (continued)**

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	SIZE	BASE
42	0x0	0x0	0x7	0x2	0x0C	0x079000
43	0x0	0x0	0x7	0x2	0x0C	0x086000
44	0x0	0x0	0x7	0x2	0x0C	0x087000
45	0x0	0x0	0x7	0x2	0x0C	0x088000
46	0x0	0x0	0x7	0x2	0x0C	0x089000
47	0x0	0x0	0x7	0x2	0x0C	0x096000
48	0x0	0x0	0x7	0x2	0x0C	0x097000
49	0x0	0x0	0x7	0x2	0x0C	0x098000
50	0x0	0x0	0x7	0x2	0x0C	0x099000
51	0x0	0x0	0x7	0x2	0x0C	0x09A000
52	0x0	0x0	0x7	0x2	0x0C	0x09B000
53	0x0	0x0	0x7	0x2	0x0C	0x09C000
54	0x0	0x0	0x7	0x2	0x0C	0x09D000
55	0x0	0x0	0x7	0x2	0x0C	0x09E000
56	0x0	0x0	0x7	0x2	0x0C	0x09F000
57	0x0	0x0	0x1	0x2	0x0C	0x0A0000
58	0x0	0x0	0x7	0x2	0x0C	0x0A1000
59	0x0	0x0	0x1	0x2	0x0C	0x0A2000
60	0x0	0x0	0x7	0x2	0x0C	0x0A3000
61	0x0	0x0	0x1	0x2	0x0D	0x0A8000
62	0x0	0x0	0x7	0x2	0x0C	0x0AA000
63	0x0	0x0	0x7	0x2	0x0C	0x0AD000
64	0x0	0x0	0x7	0x2	0x0C	0x0AE000
65	0x0	0x0	0x7	0x2	0x0C	0x0B2000
66	0x0	0x0	0x7	0x2	0x0C	0x0B3000
67	0x0	0x0	0x7	0x2	0x0C	0x0B4000
68	0x0	0x0	0x7	0x2	0x0C	0x0B5000
69	0x0	0x0	0x7	0x2	0x0C	0x0B8000
70	0x0	0x0	0x7	0x2	0x0C	0x0B9000
71	0x0	0x0	0x7	0x2	0x0C	0x0BA000
72	0x0	0x0	0x7	0x2	0x0C	0x0BB000
73	0x0	0x0	0x7	0x2	0x0C	0x151000
74	0x0	0x0	0x7	0x2	0x0C	0x0D2000
75	0x0	0x0	0x7	0x2	0x0C	0x0D5000
76	0x0	0x0	0x7	0x2	0x0C	0x0D6000
77	0x0	0x1	0x7	0x2	0x0C	0x150000
78	0x0	0x1	0x7	0x2	0x0C	0x151000
79	0x0	0x0	0x7	0x1	0x0C	0x0A2000
80	0x0	0x0	0x7	0x1	0x0C	0x0A3000
81	0x2	0x0	0x7	0x2	0x0A	0x001400
82	0x3	0x0	0x7	0x2	0x0A	0x001800
83	0x4	0x0	0x7	0x2	0x0A	0x001C00
84	0x1	0x0	0x1	0x2	0x0C	0x0A5000

**Table 13-416. L4\_AP\_REGION\_I Reset Value for L4 CFG**

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDT H_EXP	SIZE	BASE
0	0x0	0x0	0x0	0x2	0x0B	0x00000
1	0x1	0x0	0x7	0x2	0x0B	0x00800
2	0x5	0x0	0x7	0x2	0x0C	0x01000
3	0x0	0x0	0x7	0x2	0x0C	0x02000
4	0x0	0x0	0x7	0x2	0x0C	0x03000
5	0x0	0x0	0x7	0x2	0x0C	0x04000
6	0x0	0x0	0x7	0x2	0x0C	0x05000
7	0x0	0x0	0x7	0x2	0x0C	0x56000
8	0x0	0x0	0x7	0x2	0x0C	0x57000
9	0x0	0x0	0x7	0x2	0x0C	0x5C000
10	0x0	0x0	0x7	0x2	0x0E	0x58000
11	0x0	0x0	0x7	0x2	0x0C	0x62000
12	0x0	0x0	0x7	0x2	0x0C	0x63000
13	0x0	0x1	0x7	0x2	0x0C	0x59000
14	0x0	0x1	0x7	0x2	0x0C	0x5A000
15	0x0	0x0	0x7	0x2	0x0E	0x5B000
16	0x0	0x0	0x7	0x2	0x0C	0x5C000
17	0x0	0x0	0x7	0x2	0x0E	0x5D000
18	0x0	0x0	0x7	0x2	0x0C	0x5E000
19	0x0	0x1	0x5	0x2	0x0C	0x60000
20	0x0	0x1	0x7	0x2	0x0C	0x61000
21	0x0	0x2	0x7	0x2	0x0C	0x00000
22	0x0	0x2	0x7	0x2	0x0C	0x01000
23	0x0	0x0	0x7	0x2	0x0C	0x08000
24	0x0	0x0	0x7	0x2	0x0C	0x0A000
25	0x0	0x0	0x7	0x2	0x0C	0x66000
26	0x0	0x0	0x7	0x2	0x0C	0x67000
27	0x0	0x1	0x1	0x2	0x0C	0x74000
28	0x0	0x1	0x7	0x2	0x0C	0x75000
29	0x0	0x1	0x7	0x2	0x0C	0x76000
30	0x0	0x1	0x7	0x2	0x0C	0x77000
31	0x0	0x4	0x1	0x2	0x0C	0x1E000
32	0x0	0x4	0x7	0x2	0x0C	0x1F000
33	0x0	0x4	0x1	0x2	0x0C	0x6C000
34	0x0	0x4	0x7	0x2	0x0C	0x0B000
35	0x0	0x4	0x1	0x2	0x0C	0x04000
36	0x0	0x4	0x7	0x2	0x0C	0x05000
37	0x0	0x4	0x1	0x2	0x0C	0x06000
38	0x0	0x4	0x7	0x2	0x0C	0x07000
39	0x1	0x4	0x1	0x2	0x0D	0x12000
40	0x2	0x1	0x7	0x2	0x0C	0x13000
41	0x0	0x0	0x1	0x2	0x0D	0x0C000
42	0x0	0x0	0x7	0x2	0x0C	0x0D000
43	0x0	0x0	0x1	0x2	0x0C	0x10000
44	0x0	0x0	0x7	0x2	0x0C	0x11000
45	0x0	0x1	0x1	0x2	0x0C	0x16000
46	0x0	0x1	0x7	0x2	0x0C	0x17000

**Table 13-416. L4\_AP\_REGION\_I Reset Value for L4 CFG (continued)**

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDT H_EXP	SIZE	BASE
47	0x0	0x1	0x1	0x2	0x0C	0x14000
48	0x0	0x1	0x7	0x2	0x0C	0x15000
49	0x0	0x4	0x1	0x2	0x0C	0x18000
50	0x0	0x4	0x7	0x2	0x0C	0x19000
51	0x0	0x4	0x1	0x2	0x0C	0x20000
52	0x0	0x4	0x7	0x2	0x0C	0x21000
53	0x0	0x4	0x1	0x2	0x0C	0x26000
54	0x0	0x4	0x7	0x2	0x0C	0x27000
55	0x0	0x4	0x1	0x2	0x0C	0x28000
56	0x0	0x4	0x7	0x2	0x0C	0x29000
57	0x0	0x4	0x1	0x2	0x0C	0x2A000
58	0x0	0x4	0x7	0x2	0x0C	0x1D000
59	0x0	0x4	0x1	0x2	0x0C	0x1C000
60	0x0	0x4	0x7	0x2	0x0C	0x1D000
61	0x0	0x2	0x1	0x2	0x0C	0x02000
62	0x0	0x2	0x7	0x2	0x0C	0x03000
63	0x0	0x2	0x1	0x2	0x0C	0x08000
64	0x0	0x2	0x7	0x2	0x0C	0x09000
65	0x0	0x2	0x1	0x2	0x0C	0x0A000
66	0x0	0x2	0x7	0x2	0x0C	0x0B000
67	0x0	0x6	0x1	0x2	0x11	0x00000
68	0x0	0x6	0x7	0x2	0x0C	0x40000
69	0x0	0x1	0x1	0x2	0x0C	0x36000
70	0x0	0x1	0x7	0x2	0x0C	0x37000
71	0x1	0x6	0x1	0x2	0x0E	0x20000
72	0x2	0x6	0x2	0x2	0x0D	0x24000
73	0x3	0x6	0x1	0x2	0x0C	0x26000
74	0x4	0x6	0x7	0x2	0x0C	0x27000
75	0x5	0x6	0x1	0x2	0x0C	0x27000
76	0x6	0x4	0x4	0x2	0x0C	0x29000
77	0x9	0x4	0x1	0x2	0x10	0x30000
78	0x0	0x4	0x7	0x2	0x0C	0x4D000
79	0x0	0x2	0x7	0x2	0x0C	0x4E000
80	0x0	0x0	0x7	0x2	0x0D	0x5E000
81	0x0	0x0	0x7	0x2	0x0C	0x60000
82	0x0	0x1	0x7	0x2	0x0C	0x29000
83	0x0	0x1	0x7	0x2	0x0C	0x2A000
84	0x0	0x1	0x7	0x2	0x0C	0x2B000
85	0x0	0x1	0x7	0x2	0x0C	0x2C000
86	0x0	0x0	0x7	0x2	0x0C	0x64000
87	0x0	0x0	0x7	0x2	0x0C	0x65000
88	0x0	0x2	0x7	0x2	0x0C	0x2D000
89	0x0	0x4	0x1	0x2	0x0C	0x2E000
90	0x7	0x6	0x7	0x2	0x0D	0x2A000
91	0x8	0x6	0x1	0x2	0x0E	0x2C000

PRELIMINARY

## Chip-to-Chip Interface (C2C)

This chapter describes the C2C interconnection of the device.

Topic	Page
14.1 C2C Overview .....	2816
14.2 C2C Integration .....	2817
14.3 C2C Power, Reset, and Clock Management .....	2818
14.4 C2C L3 Interconnect .....	2844
14.5 C2C SSCM .....	2849
14.6 Intersystem Communication Register Module .....	2851
14.7 C2C Register Manual .....	2859

## 14.1 C2C Overview

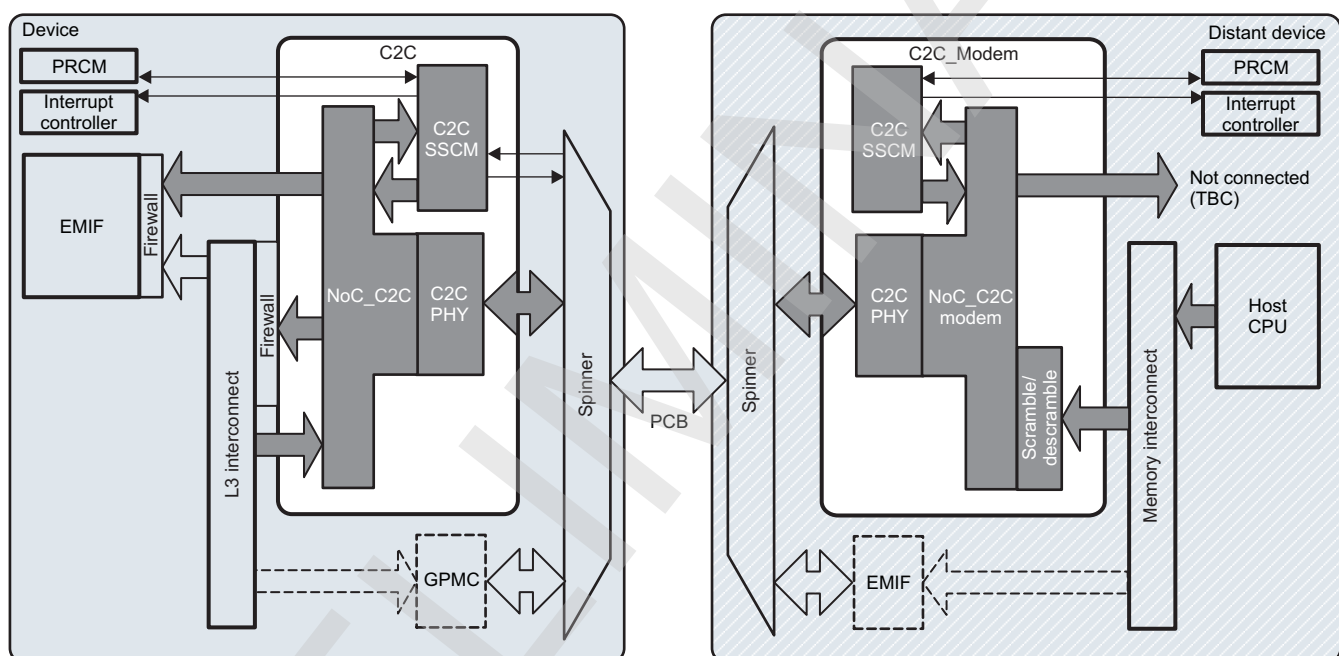
The chip-to-chip (C2C) is a serial, low-latency, peer-to-peer communication protocol that enables the extension of an internal protocol bus to one physical device over a printed circuit board (PCB). The C2C accomplishes this function by serializing local bus transactions in one device. The serialized transactions are transferred between devices through C2C ports, and the remote bus transactions are deserialized.

The C2C includes three main functional submodules:

- The Arteris® interconnect (NoC\_C2C\_Device), including the interface ports and the C2C port
- The physical interface (PHY\_C2C\_Device)
- The subsystem control module (SSCM), which supports the clock division during operating performance point (OPP) changes, wake-up and sleep sequences, and sideband signal protocol encapsulation

Figure 14-1 shows an example of the C2C implementation mirror mode.

**Figure 14-1. C2C Implementation Mirror Mode Example**



c2c-001

The C2C includes the following general features:

- Low-pin-count interface
- Scalability up to 16 inputs/16 outputs (The device supports up to 8 inputs/16 outputs or 16 inputs/8 outputs only.)
- No mandatory 3-state signals
- All signals driven using source synchronous clocking (two dual data rate [DDR] clock signals per direction for TX and RX paths)
- Supports two PHY voltages: 1.2 and 1.8 V
- Supports symmetric (peer-to-peer) and asymmetric (host/peripheral) communication operations
- Protocol support of in-band flow control
- No extra pins needed
- All request packets, response packets, and flow control information multiplexed and sent across the same physical pins
- Supports multiple outstanding transaction reads, writes, and interrupts
- Supports mirror mode to enable self-test with identical device
- Provides signals for system power management



## 14.2 C2C Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Table 14-1 through Table 14-3 summarize the integration of the module in the device.

**Table 14-1. Integration Attributes**

Module Instance	Attributes
	Power Domain
C2C	PD_CORE

**Table 14-2. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
C2C	C2C_L3_ICLK	C2C_L3_ICLK	PRCM	L3 interconnect clock enable
C2C	C2C_L3X2_ICLK	C2C_L3X2_ICLK	PRCM	Functional clock
Resets				
C2C	C2C_WARM_RST	CORE_RST	PRCM	Global warm

**Table 14-3. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
C2C	C2C_SSCM_IRQ0	MA_IRQ_27	Cortex™-A9 INTC	Interrupt request to the Cortex-A9 interrupt controller (INTC)
	C2C_SSCM_IRQ1	MA_IRQ_88	Cortex-A9 INTC	Interrupt request to the Cortex-A9 INTC
	C2C_SSCM_IRQ0	MM_IRQ_46	Cortex™-M3 INTC	Interrupt request to the Cortex-M3 INTC
	C2C_SSCM_IRQ1	MM_IRQ_47	Cortex-M3 INTC	Interrupt request to the Cortex-M3 INTC
	C2C_SSCM_IRQ0	D_IRQ_85	DSP INTC	Interrupt request to the digital signal processor (DSP) INTC
	C2C_SSCM_IRQ1	D_IRQ_86	DSP INTC	Interrupt request to the DSP INTC
DMA Requests				
C2C	C2C_SSCM_GPO 0	S_DMA_20	sDMA	Destination is system DMA (sDMA)
	C2C_SSCM_GPO 1	S_DMA_21	sDMA	Destination is sDMA
	C2C_SSCM_GPO 2	S_DMA_67	sDMA	Destination is sDMA
	C2C_SSCM_GPO 3	S_DMA_68	sDMA	Destination is sDMA

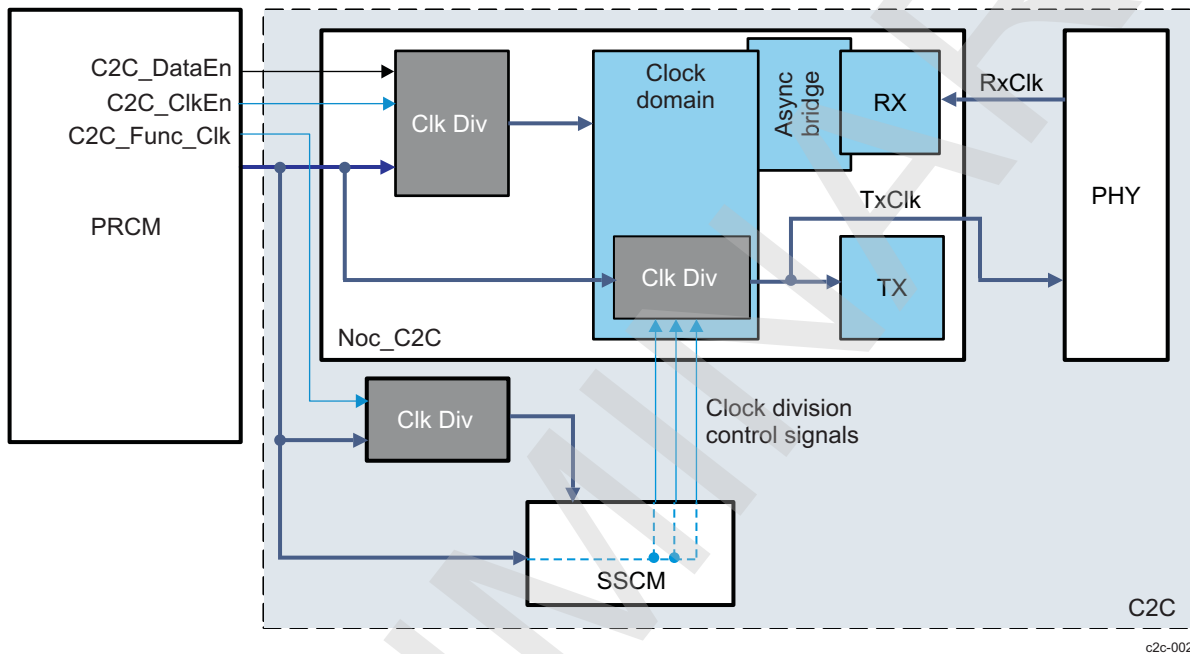
## 14.3 C2C Power, Reset, and Clock Management

### 14.3.1 PRCM Overview

The power, reset, and clock management (PRCM) module provides device-level clock, reset, and power control for other modules, as well as clock and reset signals to the C2C.

Figure 14-2 shows the C2C mode-specific overview of the PRCM.

**Figure 14-2. PRCM Overview**



c2c-002

### 14.3.2 Reset Manager Functional Description

#### 14.3.2.1 Reset Sources

This section describes the reset sources triggering the reset managers in the power and reset manager (PRM).

##### 14.3.2.1.1 Global Reset Sources

Table 14-4 lists the global reset sources of the device. The global reset source signals received by the reset manager trigger the reset of all device modules. For all hardware reset signals, the source of the reset is identified; for the software reset signals, the reset triggering bit is identified.

**Table 14-4. Global Reset Sources**

Type	Name	Source/Control	Description
Hardware	C2C_WARM_RST	C2C module	Global hardware warm reset

#### 14.3.2.2 Reset Domains

A power domain can receive power-on reset (PWRON\_RST) and/or normal reset (RST) signals. These signals reset nonretention logic and behave as follows:

- RST and PWRON\_RST are asserted on any global or local cold reset.
- Only RST is asserted on any global or local warm reset.

A power domain can receive two additional retention logic reset signals: power-on retention reset (PWRON\_RET\_RST) and/or retention reset (RET\_RST). These signals behave as follows:

- RET\_RST and PWRON\_RET\_RST are asserted on any global cold reset or wakeup from OFF state to ON-ACTIVE state.
- Only RET\_RST is asserted on any global warm reset.
- These signals are not asserted on wakeup from RETENTION state.

This section presents the trigger sources and attributes for all reset domains of the device. See [Section 14.3.2.1, Reset Sources](#), for an explanation of each reset trigger source for the device.

[Table 14-5](#) identifies the associated power and reset domains for each module.

**Table 14-5. C2C Power and Reset Domain Association**

Module	Power Domain	Reset Domains
C2C	PD_CORE	CORE_RET_RST, CORE_RST
C2C_FW	PD_CORE	CORE_PWRON_RET_RST, CORE_RST

[Table 14-6](#) lists the reset sources that trigger the reset domains of the device.

**Table 14-6. Reset Sources For the Reset Domains**

Reset Domain	Reset Source	Reset Source Type
AUDIO_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	ALWON_CM1_RST	BAD_DEVICE_RST
GLOBAL_COLD_SW_RST		Global cold
GLOBAL_WARM_SW_RST		Global warm
ICEPICKPOR_RST		Global cold
CORTEXA9_WDT_RST		Global warm
SYS_PWRON_RST		Global cold
SYS_WARMIN_RST		Global warm
VDD_CORE_VOLT_MGR_RST		Global warm
VDD_IVA_VOLT_MGR_RST		Global warm
VDD_MPU_VOLT_MGR_RST		Global warm
C2C_WARM_RST		Global warm
ALWON_CORE_RST		BAD_DEVICE_RST
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm

**Table 14-6. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type	
ALWON_IVA_RST	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
	BAD_DEVICE_RST	Global cold	
	GLOBAL_COLD_SW_RST	Global cold	
	GLOBAL_WARM_SW_RST	Global warm	
	ICEPICKPOR_RST	Global cold	
	CORTEXA9_WDT_RST	Global warm	
	SYS_PWRON_RST	Global cold	
	SYS_WARMIN_RST	Global warm	
	VDD_CORE_VOLT_MGR_RST	Global warm	
	VDD_IVA_VOLT_MGR_RST	Global warm	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
ALWON_MPU_RST	BAD_DEVICE_RST	Global cold	
	GLOBAL_COLD_SW_RST	Global cold	
	GLOBAL_WARM_SW_RST	Global warm	
	ICEPICKPOR_RST	Global cold	
	CORTEXA9_WDT_RST	Global warm	
	SYS_PWRON_RST	Global cold	
	SYS_WARMIN_RST	Global warm	
	VDD_CORE_VOLT_MGR_RST	Global warm	
	VDD_IVA_VOLT_MGR_RST	Global warm	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
	CAM_RST	BAD_DEVICE_RST	Global cold
		GLOBAL_COLD_SW_RST	Global cold
GLOBAL_WARM_SW_RST		Global warm	
ICEPICKPOR_RST		Global cold	
CORTEXA9_WDT_RST		Global warm	
SYS_PWRON_RST		Global cold	
SYS_WARMIN_RST		Global warm	
VDD_CORE_VOLT_MGR_RST		Global warm	
VDD_IVA_VOLT_MGR_RST		Global warm	
VDD_MPU_VOLT_MGR_RST		Global warm	
C2C_WARM_RST		Global warm	
CM2_RET_RST		BAD_DEVICE_RST	Global cold
		GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm	
	ICEPICKPOR_RST	Global cold	
	CORTEXA9_WDT_RST	Global warm	
	SYS_PWRON_RST	Global cold	
	SYS_WARMIN_RST	Global warm	
	VDD_CORE_VOLT_MGR_RST	Global warm	
	VDD_IVA_VOLT_MGR_RST	Global warm	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
	CORE_RET_RST	BAD_DEVICE_RST	Global cold

**Table 14-6. Reset Sources For the Reset Domains (continued)**

<b>Reset Domain</b>	<b>Reset Source</b>	<b>Reset Source Type</b>
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	CORE_RST	BAD_DEVICE_RST
GLOBAL_COLD_SW_RST		Global cold
GLOBAL_WARM_SW_RST		Global warm
ICEPICKPOR_RST		Global cold
CORTEXA9_WDT_RST		Global warm
SYS_PWRON_RST		Global cold
SYS_WARMIN_RST		Global warm
VDD_CORE_VOLT_MGR_RST		Global warm
VDD_IVA_VOLT_MGR_RST		Global warm
VDD_MPU_VOLT_MGR_RST		Global warm
C2C_WARM_RST		Global warm
DLL_RST	BAD_DEVICE_RST	Global cold
	DLL_FREQCHANGE_RST	Local warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
C2C_WARM_RST	Global warm	
DSS_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
DSS_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm

**Table 14-6. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
CORTEXM3_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	RM_CORTEXM3_RSTCTRL_RST3	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
CORTEXM3_RST1	BAD_DEVICE_RST	Global cold
	CORTEXM3_ICECRUSHER1_RST	Local warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	RM_CORTEXM3_RSTCTRL_RST1	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
CORTEXM3_RST2	BAD_DEVICE_RST	Global cold
	CORTEXM3_ICECRUSHER2_RST	Local warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	RM_CORTEXM3_RSTCTRL_RST2	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
CORTEXM3_RST3	BAD_DEVICE_RST	Global cold

**Table 14-6. Reset Sources For the Reset Domains (continued)**

<b>Reset Domain</b>	<b>Reset Source</b>	<b>Reset Source Type</b>
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	RM_CORTEXM3_RSTCTRL_RST3	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	EMU_RST	BAD_DEVICE_RST
GLOBAL_COLD_SW_RST		Global cold
GLOBAL_WARM_SW_RST		Global warm
ICEPICKPOR_RST		Global cold
CORTEXA9_WDT_RST		Global warm
SYS_PWRON_RST		Global cold
SYS_WARMIN_RST		Global warm
VDD_CORE_VOLT_MGR_RST		Global warm
VDD_IVA_VOLT_MGR_RST		Global warm
VDD_MPU_VOLT_MGR_RST		Global warm
C2C_WARM_RST		Global warm
SGX_RST		BAD_DEVICE_RST
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	IVAHD_RST	BAD_DEVICE_RST
GLOBAL_COLD_SW_RST		Global cold
GLOBAL_WARM_SW_RST		Global warm
ICEPICKPOR_RST		Global cold
CORTEXA9_WDT_RST		Global warm
RM_IVAHD_RSTCTRL_RST3		Local warm
SYS_PWRON_RST		Global cold
SYS_WARMIN_RST		Global warm
VDD_CORE_VOLT_MGR_RST		Global warm
VDD_IVA_VOLT_MGR_RST		Global warm
VDD_MPU_VOLT_MGR_RST		Global warm
C2C_WARM_RST		Global warm
IVAHD_SEQ1_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold

**Table 14-6. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	IVAHD_ICECRUSHER1_RST	Local warm
	CORTEXA9_WDT_RST	Global warm
	RM_IVAHD_RSTCTRL_RST1	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	IVAHD_SEQ2_RST	BAD_DEVICE_RST
GLOBAL_COLD_SW_RST		Global cold
GLOBAL_WARM_SW_RST		Global warm
ICEPICKPOR_RST		Global cold
IVAHD_ICECRUSHER2_RST		Local warm
CORTEXA9_WDT_RST		Global warm
RM_IVAHD_RSTCTRL_RST2		Local warm
SYS_PWRON_RST		Global cold
SYS_WARMIN_RST		Global warm
VDD_CORE_VOLT_MGR_RST		Global warm
VDD_IVA_VOLT_MGR_RST		Global warm
VDD_MPU_VOLT_MGR_RST		Global warm
L3_INIT_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
	L3_INIT_RST	BAD_DEVICE_RST
GLOBAL_COLD_SW_RST		Global cold
GLOBAL_WARM_SW_RST		Global warm
ICEPICKPOR_RST		Global cold
CORTEXA9_WDT_RST		Global warm
SYS_PWRON_RST		Global cold
SYS_WARMIN_RST		Global warm
VDD_CORE_VOLT_MGR_RST		Global warm
VDD_IVA_VOLT_MGR_RST		Global warm
VDD_MPU_VOLT_MGR_RST		Global warm
C2C_WARM_RST		Global warm
L4_PER_RET_RST		BAD_DEVICE_RST



**Table 14-6. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
	C2C_WARM_RST	Global warm
L4_PER_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
	VDD_MPU_VOLT_MGR_RST	Global warm
C2C_WARM_RST	Global warm	
CORTEXA9_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_ICECRUSHER_RST	Local warm
	CORTEXA9_WDT_RST	Global warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
VDD_MPU_VOLT_MGR_RST	Global warm	
C2C_WARM_RST	Global warm	
SDMA_RET_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICKPOR_RST	Global cold
	CORTEXA9_WDT_RST	Global warm
	SDMA_RESTORE_RST	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	VDD_CORE_VOLT_MGR_RST	Global warm
	VDD_IVA_VOLT_MGR_RST	Global warm
VDD_MPU_VOLT_MGR_RST	Global warm	
C2C_WARM_RST	Global warm	
DSP_RST	BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold

**Table 14-6. Reset Sources For the Reset Domains (continued)**

Reset Domain	Reset Source	Reset Source Type	
	GLOBAL_WARM_SW_RST	Global warm	
	ICEPICKPOR_RST	Global cold	
	CORTEXA9_WDT_RST	Global warm	
	RM_DSP_RSTCTRL_RST1	Local warm	
	SYS_PWRON_RST	Global cold	
	SYS_WARMIN_RST	Global warm	
	DSP_EMU_RESET_REQ_TR	Local warm	
	VDD_CORE_VOLT_MGR_RST	Global warm	
	VDD_IVA_VOLT_MGR_RST	Global warm	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
	DSP_RET_RST	BAD_DEVICE_RST	Global cold
		GLOBAL_COLD_SW_RST	Global cold
GLOBAL_WARM_SW_RST		Global warm	
ICEPICKPOR_RST		Global cold	
CORTEXA9_WDT_RST		Global warm	
RM_DSP_RSTCTRL_RST2		Local warm	
SYS_PWRON_RST		Global cold	
SYS_WARMIN_RST		Global warm	
VDD_CORE_VOLT_MGR_RST		Global warm	
VDD_IVA_VOLT_MGR_RST		Global warm	
VDD_MPU_VOLT_MGR_RST		Global warm	
C2C_WARM_RST		Global warm	
DSP_SYS_RST		BAD_DEVICE_RST	Global cold
	GLOBAL_COLD_SW_RST	Global cold	
	GLOBAL_WARM_SW_RST	Global warm	
	ICEPICKPOR_RST	Global cold	
	CORTEXA9_WDT_RST	Global warm	
	RM_DSP_RSTCTRL_RST2	Local warm	
	SYS_PWRON_RST	Global cold	
	SYS_WARMIN_RST	Global warm	
	VDD_CORE_VOLT_MGR_RST	Global warm	
	VDD_IVA_VOLT_MGR_RST	Global warm	
	VDD_MPU_VOLT_MGR_RST	Global warm	
	C2C_WARM_RST	Global warm	
	WKUP_RST	BAD_DEVICE_RST	Global cold
GLOBAL_COLD_SW_RST		Global cold	
GLOBAL_WARM_SW_RST		Global warm	
ICEPICKPOR_RST		Global cold	
CORTEXA9_WDT_RST		Global warm	
SYS_PWRON_RST		Global cold	
SYS_WARMIN_RST		Global warm	
VDD_CORE_VOLT_MGR_RST		Global warm	
VDD_IVA_VOLT_MGR_RST		Global warm	
VDD_MPU_VOLT_MGR_RST		Global warm	
C2C_WARM_RST		Global warm	

### 14.3.3 Clock Manager Functional Description

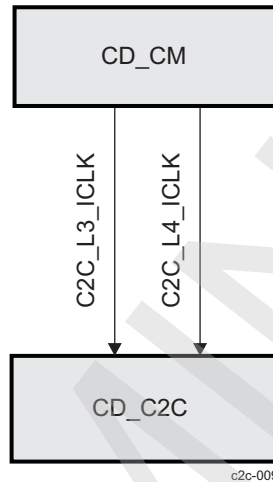
#### 14.3.3.1 CD\_C2C Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits, and their dependencies on other device clock domains.

##### 14.3.3.1.1 Overview

Figure 14-3 shows an overview of the clock domain.

Figure 14-3. CD\_C2C Overview



##### 14.3.3.1.2 Clock Domain Modes

Table 14-7 lists the clock domain modes supported by the clock domain.

Table 14-7. CD\_C2C Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 14-8 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 14-8. CD\_C2C Control and Status Parameters

Parameter Name	Control/Status Bit Field
C2C_L3_ICLK clock status	CM_C2C_CLKSTCTRL[8] CLKACTIVITY_L3_C2C_ICLK
C2C_L4_ICLK clock status	CM_C2C_CLKSTCTRL[9] CLKACTIVITY_L4_C2C_ICLK
Clock domain state transition control	CM_C2C_CLKSTCTRL[1:0] CLKTRCTRL

##### 14.3.3.1.3 Clock Domain Dependency

CD\_C2C has no module wake-up dependency on any other device clock domains.

##### 14.3.3.1.3.1 Static Dependency

Table 14-9 lists the static dependency of the clock domain on other device clock domains.

**Table 14-9. CD\_C2C Static Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_ABE	Disabled	CM_C2C_STATICDEP[3] ABE_STATDEP	Read/write
CD_IVAHD	Disabled	CM_C2C_STATICDEP[2] IVAHD_STATDEP	Read/write
CD_L3_1	Disabled	CM_C2C_STATICDEP[5] L3_1_STATDEP	Read/write
CD_L3_2	Disabled	CM_C2C_STATICDEP[6] L3_2_STATDEP	Read/write
CD_L3_INIT	Disabled	CM_C2C_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Disabled	CM_C2C_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4_PER	Disabled	CM_C2C_STATICDEP[13] L4PER_STATDEP	Read/write
CD_EMIF	Enabled	CM_C2C_STATICDEP[4] MEMIF_STATDEP	Read/write

#### 14.3.3.1.3.2 Dynamic Dependency

Table 14-10 lists the dynamic dependency of the clock domain on other device clock domains.

**Table 14-10. CD\_C2C Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_2	Always enabled	CM_C2C_DYNAMICDEP[6] L3_2_DYNDEP	Read only
CD_EMIF	Always enabled	CM_C2C_DYNAMICDEP[4] MEMIF_DYNDEP	Read only

#### 14.3.3.1.4 Clock Domain Module Attributes

Table 14-11 lists the clocks received by each module of the clock domain and their role (that is, as a functional or interface clock).

**Table 14-11. CD\_C2C Module Clocks Association**

Module	Clock	Clock Type
C2C	C2C_L3_ICLK	Interface
C2C_FW	C2C_L3_ICLK	Interface
	C2C_L4_ICLK	Interface

Table 14-12 lists the supported wake-up request generation capability for each module of the clock domain.

**Table 14-12. CD\_C2C Module Wake-Up Request**

Module	Wake-Up Feature
C2C	Master wake-up request
C2C_FW	None

Table 14-13 lists the supported clock management modes and associated software control bit fields for each module of the power domain.

**Table 14-13. CD\_C2C Module Clock Management Modes and Control**

Module	Clock Mgmt. Protocol	Status Bit Field	Role
C2C	Master/slave	CM_C2C_C2C_CLKCTRL[18] STBYST	Standby status
		CM_C2C_C2C_CLKCTRL[17:1 6] IDLEST	Idle status
C2C_FW	Slave	CM_C2C_C2C_FW_CLKCTRL [17:16] IDLEST	Idle status

Table 14-14 lists the supported clock management modes and associated software control bit fields for each module of the power domain.

**Table 14-14. CD\_C2C Modules Slave Clock Management Modes and Control**

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
C2C	N/A	Available	N/A	CM_C2C_C2C_CLKCTRL[1:0] MODULEMODE	Read only
C2C_FW	N/A	Available	N/A	CM_C2C_C2C_FW_CLKCTRL[1:0] MODULEMODE	Read only

### 14.3.3.2 CD\_CORTEXA9 Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits, and their dependencies on other device clock domains.

#### 14.3.3.2.1 Clock Domain Dependency

CD\_CORTEXA9 has no module wake-up dependency on any other device clock domain.

##### 14.3.3.2.1.1 Static Dependency

Table 14-15 lists the static dependency of the clock domain on other device clock domains.

**Table 14-15. CD\_CORTEXA9 Static Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_C2C	Always disabled	CM_MPU_STATICDEP[18] C2C_STATDEP	Read only

### 14.3.3.3 CD\_L4\_CFG Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits, and their dependencies on other device clock domains.

#### 14.3.3.3.1 Clock Domain Dependency

CD\_L4\_CFG has no static or module wake-up dependency on any other device clock domain.

##### 14.3.3.3.1.1 Dynamic Dependency

Table 14-16 lists the dynamic dependency of the clock domain on other device clock domains.

**Table 14-16. CD\_L4\_CFG Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_C2C	Always enabled	CM_L4CFG_DYNAMICDEP[18] C2C_DYNDEP	Read only

### 14.3.3.4 CD\_L3\_2 Clock Domain

This section identifies the modes supported by the clock domain, the associated control and status bits, and their dependencies on other device clock domains.

#### 14.3.3.4.1 Clock Domain Dependency

CD\_L3\_2 has no static or module wake-up dependency on any other device clock domain.

##### 14.3.3.4.1.1 Dynamic Dependency

Table 14-17 identifies the dynamic dependency of the clock domain on other device clock domains.

**Table 14-17. CD\_L3\_2 Dynamic Dependency Association Parameters**

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_C2C	Always enabled	CM_L3_2_DYNAMICDEP[18] C2C_DYNDEP	Read only

### 14.3.4 Power Management Functional Description

The section highlights the functional concepts of power-domain-level power management in the device.

The PD\_CORE power domain supports the DPS with switching times of less than 5  $\mu$ s.

#### 14.3.4.1 PD\_CORE Description

PD\_CORE contains the following clock domains:

- CD\_C2C
- CD\_DMA
- CD\_CORTEXM3
- CD\_EMIF
- CD\_L3\_1
- CD\_L3\_2
- CD\_L3\_INSTR
- CD\_L4\_CFG

Table 14-18 lists the logic retention capability for each module of the power domain.

**Table 14-18. PD\_CORE C2C Power Attributes**

Module	Logic Retention	DFF Context Status	RFF Context Status
C2C	Partial	RM_C2C_C2C_CONTEXT[0] LOSTCONTEXT_DFF	RM_C2C_C2C_CONTEXT[1] LOSTCONTEXT_RFF
C2C_FW	Partial	RM_C2C_C2C_FW_CONTEXT[0] LOSTCONTEXT_DFF	RM_C2C_C2C_FW_CONTEXT[1] LOSTCONTEXT_RFF

### 14.3.5 PRCM Register Manual

#### 14.3.5.1 PRM Instance Summary

Table 14-19 lists the PRCM registers and their addresses and sizes.

**Table 14-19. PRM Instance Summary**

Module Name	L4 Base Address	Size
INTRCONN_SOCKET_PRM	0x4A30 6000	256 bytes
CKGEN_PRM	0x4A30 6100	256 bytes
MPU_PRM	0x4A30 6300	256 bytes
DSP_PRM	0x4A30 6400	256 bytes
ABE_PRM	0x4A30 6500	256 bytes
ALWAYS_ON_PRM	0x4A30 6600	256 bytes
CORE_PRM	0x4A30 6700	2K bytes
IVAHD_PRM	0x4A30 6F00	256 bytes
CAM_PRM	0x4A30 7000	256 bytes
DSS_PRM	0x4A30 7100	256 bytes
SGX_PRM	0x4A30 7200	256 bytes
L3INIT_PRM	0x4A30 7300	256 bytes
L4PER_PRM	0x4A30 7400	512 bytes
WKUP_PRM	0x4A30 7700	256 bytes
WKUP_CM	0x4A30 7800	256 bytes

**Table 14-19. PRM Instance Summary (continued)**

Module Name	L4 Base Address	Size
EMU_PRM	0x4A30 7900	256 bytes
EMU_CM	0x4A30 7A00	256 bytes
DEVICE_PRM	0x4A30 7B00	256 bytes
INSTR_PRM	0x4A30 7F00	256 bytes

**14.3.5.2 CORE\_PRM Registers**

**14.3.5.2.1 CORE\_PRM Register Summary**

Table 14-20 summarizes the CORE\_PRM registers.

**Table 14-20. CORE\_PRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CORE_PRM L4 Base Address
<a href="#">RM_C2C_C2C_CONTEXT</a>	RW	32	0x0000 0524	0x4A30 6C24
<a href="#">RM_C2C_C2C_FW_CONTEXT</a>	RW	32	0x0000 0534	0x4A30 6C34

**14.3.5.2.2 CORE\_PRM Register Description**

Table 14-21 through Table 14-23 describe the individual CORE\_PRM registers in detail.

**Table 14-21. RM\_C2C\_C2C\_CONTEXT**

<b>Address Offset</b>	0x0000 0524	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 14-20</a> .		
<b>Description</b>	This register contains dedicated C2C context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		LOSTCONTEXT_DFF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specifies whether RFF-based context has been lost due to a previous power transition or other reset source (set upon assertion of CORE_RST signal)  0x0: Context has been maintained. 0x1: Context has been lost.	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specifies whether DFF-based context has been lost due to a previous power transition or other reset source (set upon assertion of CORE_RST signal)  0x0: Context has been maintained. 0x1: Context has been lost.	RW W1toClr	1

**Table 14-22. Register Call Summary for Register RM\_C2C\_C2C\_CONTEXT**

C2C Power, Reset, and Clock Management

- [PD\\_CORE Description: \[0\] \[1\]](#)
- [CORE\\_PRM Register Summary: \[2\]](#)

**Table 14-23. RM\_C2C\_C2C\_FW\_CONTEXT**

<b>Address Offset</b>	0x0000 0534	<b>Instance</b>	CORE_PRM
<b>Physical Address</b>	See <a href="#">Table 14-20</a> .		
<b>Description</b>	This register contains dedicated C2C_FW context statuses. [warm reset insensitive]		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOSTCONTEXT_RFF		LOSTCONTEXT_DFF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	LOSTCONTEXT_RFF	Specifies whether RFF-based context has been lost due to a previous power transition or other reset source (set upon assertion of CORE_RET_RST signal)  0x0: Context has been maintained. 0x1: Context has been lost.	RW W1toClr	1
0	LOSTCONTEXT_DFF	Specifies whether DFF-based context has been lost due to a previous power transition or other reset source (set upon assertion of CORE_RST signal)  0x0: Context has been maintained. 0x1: Context has been lost.	RW W1toClr	1

**Table 14-24. Register Call Summary for Register RM\_C2C\_C2C\_FW\_CONTEXT**

C2C Power, Reset, and Clock Management

- [PD\\_CORE Description: \[0\] \[1\]](#)
- [CORE\\_PRM Register Summary: \[2\]](#)

### 14.3.5.3 CORE\_CM2 Registers

#### 14.3.5.3.1 CORE\_CM2 Register Summary

[Table 14-36](#) summarizes the CORE\_CM2 registers.

**Table 14-25. CORE\_CM2 Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	CORE_CM2 L4 Base Address
<a href="#">CM_C2C_CLKSTCTRL</a>	RW	32	0x0000 0500	0x4A00 8C00
<a href="#">CM_C2C_STATICDEP</a>	RW	32	0x0000 0504	0x4A00 8C04
<a href="#">CM_C2C_DYNAMICDEP</a>	RW	32	0x0000 0508	0x4A00 8C08
<a href="#">CM_C2C_C2C_CLKCTRL</a>	R	32	0x0000 0520	0x4A00 8C20
<a href="#">CM_C2C_C2C_FW_CLKCTRL</a>	R	32	0x0000 0530	0x4A00 8C30



**Table 14-25. CORE\_CM2 Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	CORE_CM2 L4 Base Address
CM_L4CFG_DYNAMICDEP	RW	32	0x0000 0608	0x4A00 8D08

**14.3.5.3.2 CORE\_CM2 Register Description**

Table 14-26 through Table 14-36 describe the individual CORE\_CM2 registers in detail.

**Table 14-26. CM\_C2C\_CLKSTCTRL**

<b>Address Offset</b>	0x0000 0500
<b>Physical Address</b>	See Table 14-25.
<b>Description</b>	This register enables the domain power state transition. It controls the hardware-supervised domain power state transition between On-Active and On-Inactive states. It also holds 1 status bit per clock input of the domain.
<b>Type</b>	RW
<b>Instance</b>	CORE_CM2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																CCLKACTIVITY_L3X2_C2C_ICLK			CLKACTIVITY_L4_C2C_ICLK			CLKACTIVITY_L3_C2C_ICLK			RESERVED							CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10	CCLKACTIVITY_L3X2_C2C_ICLK	This field indicates the state of the L3X2_C2C_ICLK clock in the domain. [warm reset insensitive] Read 0x0: Corresponding clock is definitely gated. Read 0x1: Corresponding clock is running or gating/ungating transition is ongoing.	R	0
9	CLKACTIVITY_L4_C2C_ICLK	This field indicates the state of the L4_C2C_ICLK clock in the domain. [warm reset insensitive] Read 0x0: On-Inactive is definitely gated. Read 0x1: On-Inactive is running or gating/ungating transition is ongoing.	R	0
8	CLKACTIVITY_L3_C2C_ICLK	This field indicates the state of the L3_C2C_ICLK clock in the domain. [warm reset insensitive] Read 0x0: On-Inactive is definitely gated. Read 0x1: On-Inactive is running or gating/ungating transition is ongoing.	R	0
7:2	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	<p>Controls the clock state transition of the C2C clock domain.</p> <p>0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may occur.</p> <p>Read 0x1: Reserved</p> <p>0x2: SW_WKUP: Start a software forced wake-up transition on the domain.</p> <p>0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transitions are based upon hardware conditions.</p>	RW	0x3

**Table 14-27. Register Call Summary for Register CM\_C2C\_CLKSTCTRL**

C2C Power, Reset, and Clock Management

- [Clock Domain Modes: \[0\] \[1\] \[2\]](#)
- [CORE\\_CM2 Register Summary: \[3\]](#)

**Table 14-28. CM\_C2C\_STATICDEP**

<b>Address Offset</b>	0x0000 0504	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 14-25</a> .		
<b>Description</b>	This register controls the static domain dependencies from C2C domain toward target domains. It is relevant only for a domain having system initiator(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																RESERVED				L4PER_STATDEP	L4CFG_STATDEP	RESERVED				L3INIT_STATDEP	L3_2_STATDEP	L3_1_STATDEP	MEMIF_STATDEP	ABE_STATDEP	IVAHD_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13	L4PER_STATDEP	<p>Static dependency toward L4_PER clock domain</p> <p>0x0: Dependency is disabled.</p> <p>0x1: Dependency is enabled.</p>	RW	0
12	L4CFG_STATDEP	<p>Static dependency toward L4_CFG clock domain</p> <p>0x0: Dependency is disabled.</p> <p>0x1: Dependency is enabled.</p>	RW	0
11:8	RESERVED		R	0x0
7	L3INIT_STATDEP	<p>Static dependency toward L3_INIT clock domain</p> <p>0x0: Dependency is disabled.</p> <p>0x1: Dependency is enabled.</p>	RW	0
6	L3_2_STATDEP	<p>Static dependency toward L3_2 clock domain</p> <p>0x0: Dependency is disabled.</p> <p>0x1: Dependency is enabled.</p>	RW	0
5	L3_1_STATDEP	<p>Static dependency toward L3_1 clock domain</p> <p>0x0: Dependency is disabled.</p> <p>0x1: Dependency is enabled.</p>	RW	0

Bits	Field Name	Description	Type	Reset
4	MEMIF_STATDEP	Static dependency toward MEMIF clock domain 0x0: Dependency is disabled. 0x1: Dependency is enabled.	RW	1
3	ABE_STATDEP	Static dependency toward ABE clock domain 0x0: Dependency is disabled. 0x1: Dependency is enabled.	RW	0
2	IVAHD_STATDEP	Static dependency toward IVAHD clock domain 0x0: Dependency is disabled. 0x1: Dependency is enabled.	RW	0
1:0	RESERVED		R	0x0

**Table 14-29. Register Call Summary for Register CM\_C2C\_STATICDEP**

- C2C Power, Reset, and Clock Management
- [Static Dependency](#): [0] [1] [2] [3] [4] [5] [6] [7]
  - [CORE\\_CM2 Register Summary](#): [8]

**Table 14-30. CM\_C2C\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0508	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 14-25</a> .		
<b>Description</b>	This register controls the dynamic domain dependencies from C2C domain toward target domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED										L3_2_DYNDEP	RESERVED	MEMIF_DYNDEP	RESERVED										

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of autosleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:7	RESERVED		R	0x00000
6	L3_2_DYNDEP	Dynamic dependency toward L3_2 clock domain Read 0x1: Dependency is enabled.	R returns 1s	1
5	RESERVED		R	0
4	MEMIF_DYNDEP	Dynamic dependency toward MEMIF clock domain Read 0x1: Dependency is enabled.	R returns 1s	1
3:0	RESERVED		R	0x0

**Table 14-31. Register Call Summary for Register CM\_C2C\_DYNAMICDEP**

- C2C Power, Reset, and Clock Management
- [Dynamic Dependency](#): [0] [1]
  - [CORE\\_CM2 Register Summary](#): [2]

**Table 14-32. CM\_C2C\_C2C\_CLKCTRL**

<b>Address Offset</b>	0x0000 0520		
<b>Physical Address</b>	See <a href="#">Table 14-25</a> .	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the C2C clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED														STBYST		IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18	STBYST	C2C module standby status [warm reset insensitive] Read 0x0: Module is functional (not in standby). Read 0x1: Module is in standby.	R	1
17:16	IDLEST	C2C interface idle status [warm reset insensitive] Read 0x0: C2C interface is in functional state. Read 0x1: C2C interface is in a transitory state. Read 0x2: C2C interface is in Idle state.	R	0x2
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Controls the way mandatory clocks are managed. Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL = 3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 14-33. Register Call Summary for Register CM\_C2C\_C2C\_CLKCTRL**

C2C Power, Reset, and Clock Management

- [Clock Domain Module Attributes: \[0\] \[1\] \[2\]](#)
- [CORE\\_CM2 Register Summary: \[3\]](#)

**Table 14-34. CM\_C2C\_C2C\_FW\_CLKCTRL**

<b>Address Offset</b>	0x0000 0530		
<b>Physical Address</b>	See <a href="#">Table 14-25</a> .	<b>Instance</b>	CORE_CM2
<b>Description</b>	This register manages the C2C_FW clocks.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17:16	IDLEST	Module idle status [warm reset insensitive] Read 0x0: Module is fully functional, including INTRCONN. Read 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion. Read 0x2: Module is in Idle mode (only INTRCONN part). It is functional if using separate functional clock. Read 0x3: Module is disabled and cannot be accessed.	R	0x3
15:2	RESERVED		R	0x0000
1:0	MODULEMODE	Controls the way mandatory clocks are managed Read 0x1: Module is managed automatically by hardware according to clock domain transition. A clock domain sleep transition puts module into idle. A wakeup domain transition puts it back into function. If CLKTRCTRL = 3, any INTRCONN access to module is always granted. Module clocks may be gated according to the clock domain state.	R Rreturns	0x1

**Table 14-35. Register Call Summary for Register CM\_C2C\_C2C\_FW\_CLKCTRL**

C2C Power, Reset, and Clock Management

- [Clock Domain Module Attributes: \[0\] \[1\]](#)
- [CORE\\_CM2 Register Summary: \[2\]](#)

**Table 14-36. CM\_L4CFG\_DYNAMICDEP**

<b>Address Offset</b>	0x0000 0608	<b>Instance</b>	CORE_CM2
<b>Physical Address</b>	See <a href="#">Table 14-25</a> .		
<b>Description</b>	This register controls the dynamic domain dependencies from L4_CFG domain toward target domains. It is relevant only for domain having INTRCONN master port(s).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED				C2C_DYNDP	RESERVED	ALWONCORE_DYNDP	L4WKUP_DYNDP	RESERVED				SDMA_DYNDP	RESERVED	CAM_DYNDP	DSS_DYNDP	L3_INIT_DYNDP	L3_2_DYNDP	L3_1_DYNDP	MEMIF_DYNDP	RESERVED	DSP_DYNDP	RESERVED	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor INTRCONN interface activity for determination of autosleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:19	RESERVED		R	0x00
18	C2C_DYNDP	Dynamic dependency toward C2C clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1
17	RESERVED		R	1
16	ALWONCORE_DYNDP	Dynamic dependency toward ALWONCORE clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1

Bits	Field Name	Description	Type	Reset
15	L4WKUP_DYNDEP	Dynamic dependency toward L4_WKUP clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1
14:12	RESERVED		R	0x0
11	SDMA_DYNDEP	Dynamic dependency toward sDMA clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1
10	RESERVED		R	0
9	CAM_DYNDEP	Dynamic dependency toward ISS clock domain Read 0x0: Dependency is disabled.	R	0
8	DSS_DYNDEP	Dynamic dependency toward DSS clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1
7	L3_INIT_DYNDEP	Dynamic dependency toward L3_INIT clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1
6	L3_2_DYNDEP	Dynamic dependency toward L3_2 clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1
5	L3_1_DYNDEP	Dynamic dependency toward L3_1 clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1
4	MEMIF_DYNDEP	Dynamic dependency toward MEMIF clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1
3:2	RESERVED		R	0x0
1	DSP_DYNDEP	Dynamic dependency toward DSP clock domain Read 0x1: Dependency is enabled.	R Rreturns 1s	1
0	RESERVED		R	0

**Table 14-37. Register Call Summary for Register CM\_L4CFG\_DYNAMICDEP**

C2C Power, Reset, and Clock Management

- [Dynamic Dependency: \[0\]](#)
- [CORE\\_CM2 Register Summary: \[1\]](#)

#### 14.3.5.4 SCRM Instance Summary

[Table 14-38](#) lists the register instances for the system clock and reset manager (SCRM).

**Table 14-38. SCRM Instance Summary**

Module Name	L4 Base Address	Size
SCRM	0x4A30 A000	256 bytes

#### 14.3.5.5 SCRM Registers

##### 14.3.5.5.1 SCRM Register Summary

[Table 14-39](#) summarizes the SCRM registers.

**Table 14-39. SCRM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SCRM L4 Base Address
<a href="#">C2CLKM</a>	RW	32	0x0000 011C	0x4A30 A11C
<a href="#">C2CLKREQ</a>	RW	32	0x0000 0234	0x4A30 A234
<a href="#">C2CRSTCTRL</a>	RW	32	0x0000 041C	0x4A30 A41C

**Table 14-39. SCRM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SCRM L4 Base Address
EXTWARMRSTCTRL	RW	32	0x0000 0500	0x4A30 A500
APEWARMRSTCTRL	RW	32	0x0000 0504	0x4A30 A504
RESERVED	RW	32	0x0000 0508	0x4A30 A508
C2CWARMRSTCTRL	RW	32	0x0000 050C	0x4A30 A50C
C2CWARMRSTST_RE G	RW	32	0x0000 051C	0x4A30 A51C

**14.3.5.5.2 SCRM Register Description**

Table 14-40 through Table 14-52 describe the individual SCRM registers in detail.

**Table 14-40. C2CCLKM**

<b>Address Offset</b>	0x0000 011C	<b>Instance</b>	SCRM
<b>Physical Address</b>	See Table 14-39.		
<b>Description</b>	This register controls the clocks of the external C2C interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SYSCLK		CLK_32KHZ													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads return 0.	R	0x0000 0000
1	SYSCLK	This bit allows to enable and disable the system clock version of the external C2C interface. 0x0: The system clock version for the external C2C interface is disabled. 0x1: The system clock version for the external C2C interface is enabled.	RW	0
0	CLK_32KHZ	This bit allows to enable and disable the 32-kHz clock version of the external C2C interface. 0x0: The 32-kHz clock version for the external C2C interface is disabled. 0x1: The 32-kHz clock version for the external C2C interface is enabled.	RW	0

**Table 14-41. Register Call Summary for Register C2CCLKM**

C2C Power, Reset, and Clock Management

- [SCRM Register Summary: \[0\]](#)

**Table 14-42. C2CLKREQ**

<b>Address Offset</b>	0x0000 0234	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 14-39</a> .		
<b>Description</b>	This register holds qualifiers for the external C2C interface clock request.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACCURACY		POLARITY													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads return 0.	R	0x0000 0000
1	ACCURACY	This bit qualifies the external C2C interface clock request as an accurate clock request.  0x0: An active external C2C interface clock request does not generate an accurate clock request.  0x1: An active external C2C interface clock request generates an accurate clock request.	RW	0
0	POLARITY	This bit defines the active level of the external C2C interface clock request.  0x0: The external C2C interface clock request is active low.  0x1: The external C2C interface clock request is active high.	RW	1

**Table 14-43. Register Call Summary for Register C2CLKREQ**

C2C Power, Reset, and Clock Management

- [SCRM Register Summary: \[0\]](#)

**Table 14-44. C2CRSTCTRL**

<b>Address Offset</b>	0x0000 041C	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 14-39</a> .		
<b>Description</b>	This register controls the release of the external C2C interface reset lines.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WARMRST		COLDRST													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads return 0.	R	0x0000 0000
1	WARMRST	This bit allows to release the warm reset line of the external C2C interface. [warm reset sensitive]  0x0: Clearing this bit releases the warm reset line of the external C2C interface.  0x1: Writing this bit asserts the warm reset line of the external C2C interface.	RW	1



Bits	Field Name	Description	Type	Reset
0	COLDRST	This bit allows to release the cold reset line of the external C2C interface.  0x0: Clearing this bit releases the cold reset line of the external C2C interface.  0x1: Writing this bit asserts the cold reset line of the external C2C interface.	RW	1

**Table 14-45. Register Call Summary for Register C2CRSTCTRL**

C2C Power, Reset, and Clock Management

- [SCRM Register Summary: \[0\]](#)

**Table 14-46. EXTWARMRSTCTRL**

<b>Address Offset</b>	0x0000 0500	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 14-39</a> .		
<b>Description</b>	This register allows selection of the external warm reset source as a global warm reset for the platform.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WARMSRCSELECT			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0.	R	0x0000 0000
0	WARMSRCSELECT	Selects the external warm reset source as a global warm reset  0x0: The external warm reset source does not reset (warm) the other functions (APE, modem, and external C2C interface).  0x1: The external warm reset source resets (warm) the other functions (APE, modem, and external C2C interface).	RW	1

**Table 14-47. Register Call Summary for Register EXTWARMRSTCTRL**

C2C Power, Reset, and Clock Management

- [SCRM Register Summary: \[0\]](#)

**Table 14-48. APEWARMRSTCTRL**

<b>Address Offset</b>	0x0000 0504	<b>Instance</b>	SCRM
<b>Physical Address</b>	See <a href="#">Table 14-39</a> .		
<b>Description</b>	This register allows selection of the APE warm reset source as a global warm reset for the platform.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WARMSRCSELECT			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0.	R	0x0000 0000
0	WARMSRCSELECT	Select the APE warm reset source as a global warm reset.  0x0: The APE warm reset source does not reset (warm) the other functions (external peripherals, modem, and external C2C interface).  0x1: The APE warm reset source resets (warm) the other functions (external peripherals, modem, and external C2C interface).	RW	1

**Table 14-49. Register Call Summary for Register APEWARMRSTCTRL**

C2C Power, Reset, and Clock Management

- [SCRM Register Summary: \[0\]](#)

**Table 14-50. C2CWARMRSTCTRL**

<b>Address Offset</b>	0x0000 050C
<b>Physical Address</b>	See <a href="#">Table 14-39</a> .
<b>Description</b>	This register allows selection of the external C2C interface warm reset source as a global warm reset for the platform.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WARMSRCSELECT			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0.	R	0x0000 0000
0	WARMSRCSELECT	Select the external C2C interface warm reset source as a global warm reset.  0x0: The external C2C interface warm reset source does not reset (warm) the other functions (external peripherals, APE, and modem).  0x1: The external C2C interface warm reset source resets (warm) the other functions (external peripherals, APE, and modem).	RW	0

**Table 14-51. Register Call Summary for Register C2CWARMRSTCTRL**

C2C Power, Reset, and Clock Management

- [SCRM Register Summary: \[0\]](#)

**Table 14-52. C2CWARMRSTST\_REG**

<b>Address Offset</b>	0x0000 051C		
<b>Physical Address</b>	See <a href="#">Table 14-39</a> .	<b>Instance</b>	SCRM
<b>Description</b>	This register logs the source of warm reset on the external C2C interface. Each bit is set upon release of the external C2C interface warm reset and must be cleared by software.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												C2CWARMRSTST	Reserved		

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads return 0.	R	0x0000000
3	C2CWARMRSTST	This bit logs the C2C warm reset source. 0x0: No C2C warm reset occurred. 0x1: A C2C warm reset occurred.	RW W1toClr	0
2 : 0	Reserved	Read returns 0	R	0x0

**Table 14-53. Register Call Summary for Register C2CWARMRSTST\_REG**

C2C Power, Reset, and Clock Management

- [SCRM Register Summary: \[0\]](#)

## 14.4 C2C L3 Interconnect

---

**NOTE:** The L3 interconnect is an instantiation of the Arteris interconnect from Arteris, Inc.

This document contains materials that are ©2003–2009 Arteris, Inc., and that constitute proprietary information of Arteris, Inc.

All materials and trademarks are used under license from Arteris, Inc. For additional information, see the Arteris reference manuals, or contact Arteris, Inc.

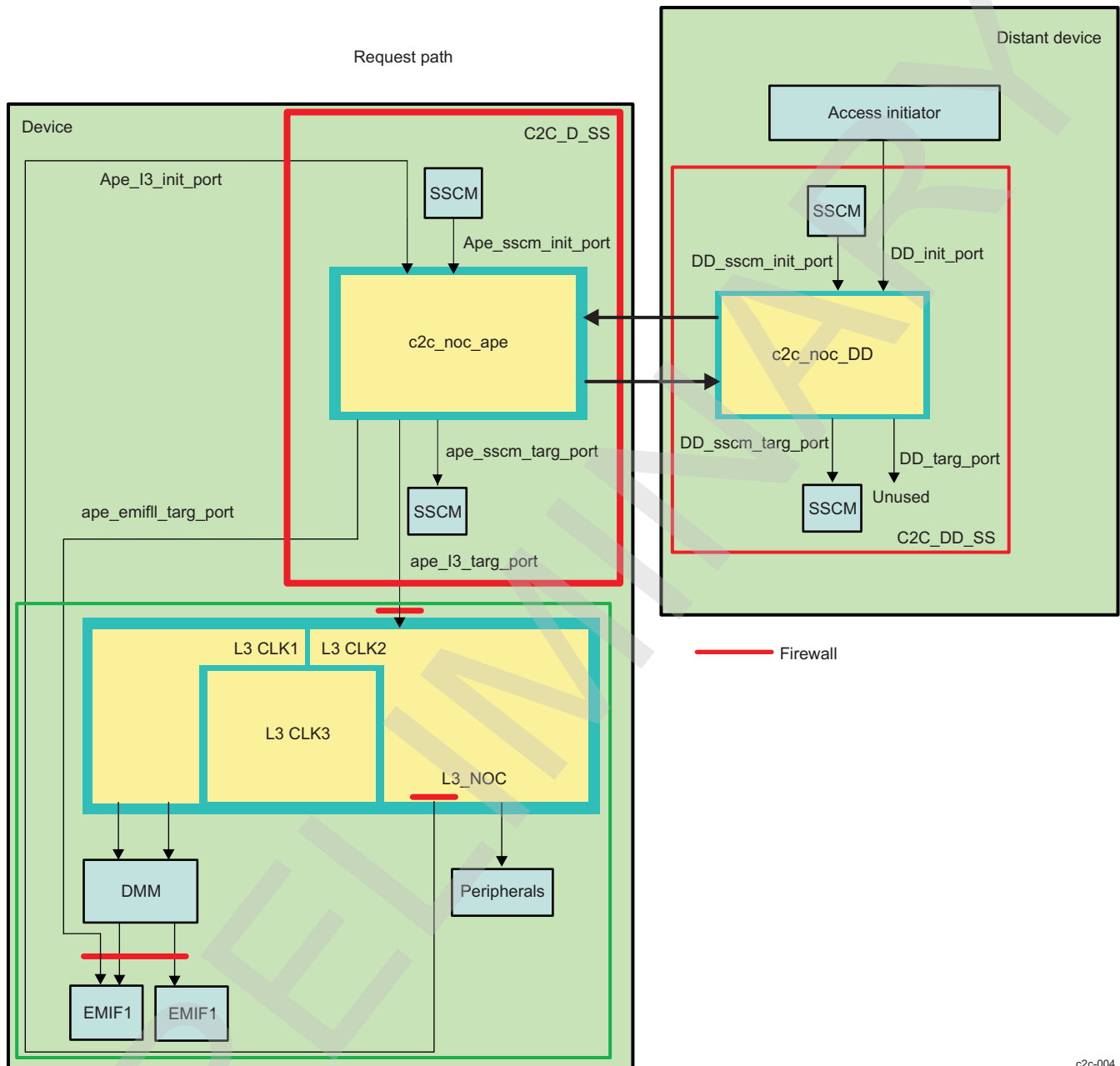
NoC is an abbreviation for network on chip.

---

### 14.4.1 C2C Interconnect Overview

Figure 14-4 shows the L3 interconnect interface.

Figure 14-4. C2C L3 Interface



c2c-004

The C2C L3 interface includes the following main features:

- Network interface units (NIUs): Master NIUs for the initiator agents (IAs) and slave NIUs for the target agents (TAs)
- A partially depleted crossbar exchange network
- A special internal slave NIU to access L3 interconnect configuration registers
- True little-endian platform
- Transaction errors tracking and logging registers
- Provides all signaling support for the chip-level, power-management infrastructure

## 14.4.2 C2C Interconnect Integration

### 14.4.2.1 Clocking, Reset, and Power-Management Scheme

#### 14.4.2.1.1 Clocks

Table 14-54 lists the C2C L3 interconnect clocks. For more information, see [Chapter 3, Power, Reset, and Clock Management](#).

**Table 14-54. Clocks**

Module	Type	Name	Source	Description
C2C L3	Interface	C2C_L3_ICLK	PRCM	Interface/functional clock for the C2C

#### 14.4.2.1.2 Resets

Table 14-55 lists the C2C L3 interconnect resets. For more information, see [Chapter 3, Power, Reset, and Clock Management](#).

**Table 14-55. Resets**

Module	Name	Source	Description
C2C L3	CORE_RST	PRCM	Global warm reset

#### 14.4.2.1.3 Power Management

The L3 interconnect ports belong to the CORE power domain. As part of the system-wide power-management scheme, the L3 ports go into IDLE state after receiving a request from the PRCM module and after all commands are serviced. For more information, see [Chapter 3, Power, Reset, and Clock Management](#).

### 14.4.3 C2C Interconnect Functional Description

#### 14.4.3.1 Slave NIU Configuration

All external subsystem master NIUs can initiate transactions to the L3 applicative slave NIUs listed in [Table 14-56](#) through the C2C port link, as shown in [Table 14-56](#).

**Table 14-56. C2C Connectivity**

Master NIUs	Slave NIUs				
	APE_L3_SN	APE_emifll_SN	APE_SSCM	DD_L3_SN	DD_SSCM
APE_L3_MN			0	0 <sup>(1)</sup>	0 <sup>(2)</sup>
APE_SSCM_MN					0
DD_SSCM_MN			0		
DD_L3_MN	0	0	0 <sup>(2)</sup>		0

<sup>(1)</sup> This connection must be assessed according to system requirements on a case-by-case basis.

<sup>(2)</sup> These connections are present only for debugging purposes.

[Table 14-57](#) lists the mapping of the C2C required information.

**Table 14-57. C2C ReqInfo Mapping**

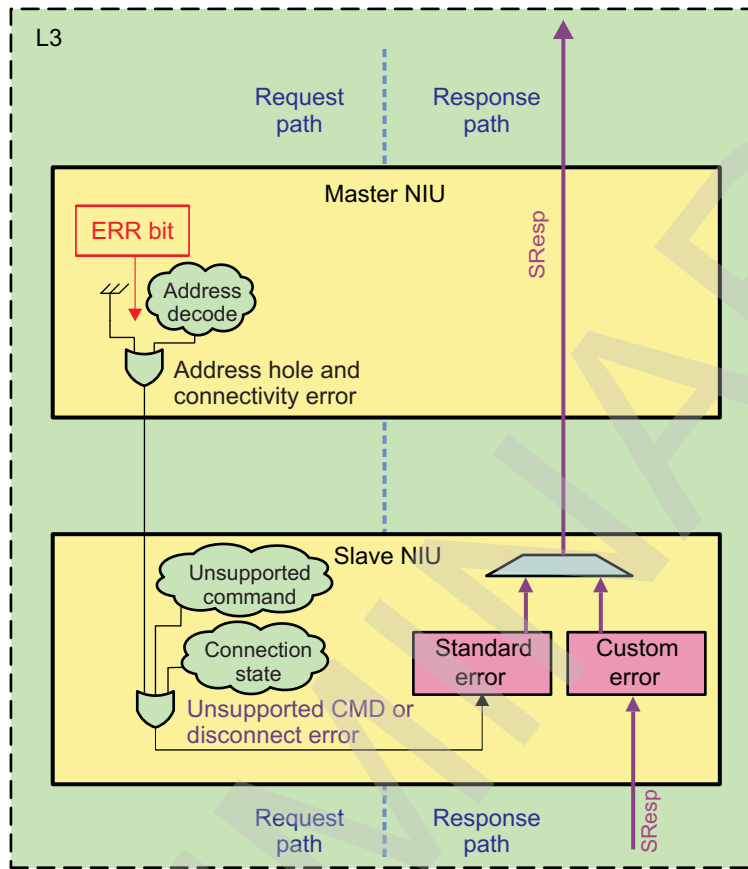
	ReqInfo Name	MReqDebug	MReqType	MReqSupervisor
MNs	APE_L3	x	0	x
	APE_SSCM			
	DD_L3	x	0	x
SNs	APE_L3	x	0	x
	APE_EMIF1	x	0	x
	APE_SSCM			
	DD_SSCM			
	DD_MNs	x	0	x

#### 14.4.3.2 Global Error Routing Scheme

Error logging is not enabled on the C2C because it is already enabled in the L3 interconnect. The C2C L3 only reports in-band errors. For more information about L3 interconnect error logging, see [Section 13.2, L3 Interconnect](#).

[Figure 14-5](#) shows the C2C global error routing scheme.

Figure 14-5. C2C Global Error Routing Scheme



c2c-006

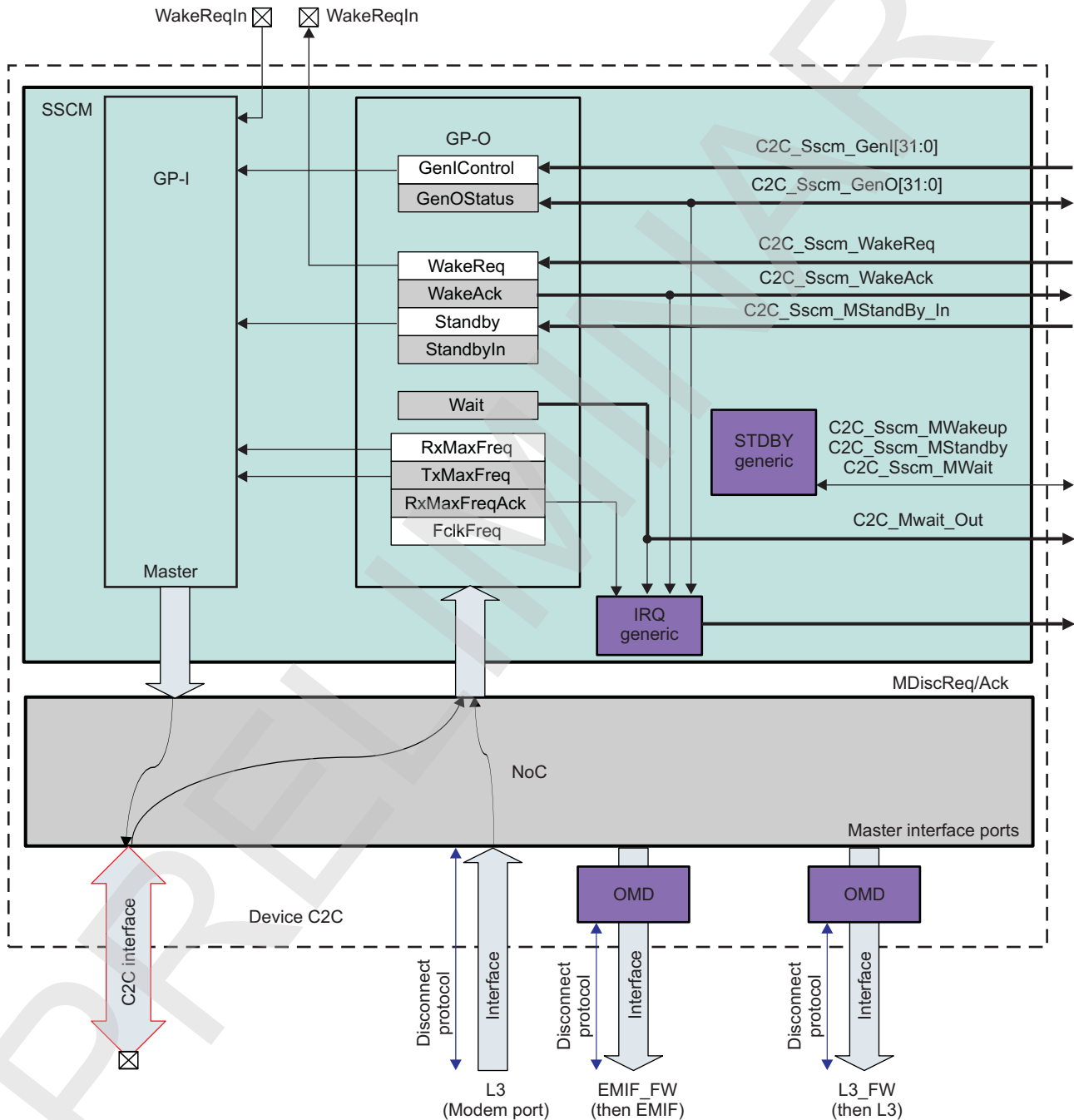


## 14.5 C2C SSCM

### 14.5.1 SSCM Overview

The SSCM supports clock division during OPP changes, the wake-up and sleep sequences, and interrupts signaling. Figure 14-7 shows the C2C SSCM block diagram.

Figure 14-6. C2C SSCM Block Diagram



c2c-003

The C2C SSCM supports:

- Different handshakes for local and distant OPP changes
- Sleep and wake-up sequences initiated locally or from a remote side
- TX clock automatic division

- Transport of general-purpose signals to and from distant C2C
- GPI and GPO submodules

## 14.5.2 SSCM Interrupts

### 14.5.2.1 SSCM GPI Interrupt Mapping

Table 14-58 lists the SSCM GPI interrupt mapping.

**Table 14-58. SSCM GPI Interrupt Mapping**

C2C_SSCM_GPI	Source	Description
C2C_SSCM_GPI[0]	SYS.IRQ1n	External interrupt
C2C_SSCM_GPI[1]	MAIL_U3_C2C_IRQ	Device to DD mailbox interrupt
C2C_SSCM_GPI[2]	MODEM_ICR_IRQ	Distant (Modem) ICR Interrupt
C2C_SSCM_GPI[3]	UART1_IRQ	UART module 1
C2C_SSCM_GPI[4]	UART2_IRQ	UART module 2
C2C_SSCM_GPI[5]	UART3_IRQ	UART module 3
C2C_SSCM_GPI[6]	GPT6_IRQ	General-purpose timer 6
C2C_SSCM_GPI[7]	GPT7_IRQ	General-purpose timer 7
C2C_SSCM_GPI[8]	GPT8_IRQ	General-purpose timer 8
C2C_SSCM_GPI[9]	GPT9_IRQ	General-purpose timer 9
C2C_SSCM_GPI[10]	GPIO1_MPU_IRQ	GPIO module 1
C2C_SSCM_GPI[11]	GPIO2_MPU_IRQ	GPIO module 2
C2C_SSCM_GPI[12]	GPIO3_MPU_IRQ	GPIO module 3
C2C_SSCM_GPI[13]	GPIO4_MPU_IRQ	GPIO module 4
C2C_SSCM_GPI[14]	HS_USB_MC_NINT	Module HS USB OTG
C2C_SSCM_GPI[15]	HS_USB_DMA_NINT	Module HS USB OTG DMA controller interrupt
C2C_SSCM_GPI[16]	FSUSB_IRQ	FS USB-HOST controller interrupt
C2C_SSCM_GPI[17]	FSUSB_SMI_IRQ	FS USB-HOST controller SMI
C2C_SSCM_GPI[18]	L3_APP_IRQ	Application errors on L3
C2C_SSCM_GPI[19]	ONCHI_IRQ	HSUSB MP host interrupt
C2C_SSCM_GPI[20]	EHCI_IRQ	HSUSB MP host interrupt
C2C_SSCM_GPI[21]	TLL_IRQ	TLL interrupt
C2C_SSCM_GPI[22]	SYS.IRQ2n	External interrupt
C2C_SSCM_GPI[23:29]	Reserved	Reserved
C2C_SSCM_GPI[30]	Modem_shutdown_irqModem_wake_irq	PRCM Modem_shutdown_irq
C2C_SSCM_GPI[31]	Modem_wake_irq	PRCM Modem_wake_irq

### 14.5.3 SSCM GPO

The interrupts described in [Section 14.1](#), *C2C Overview*, are active in GPO mode.

## 14.6 Intersystem Communication Register Module

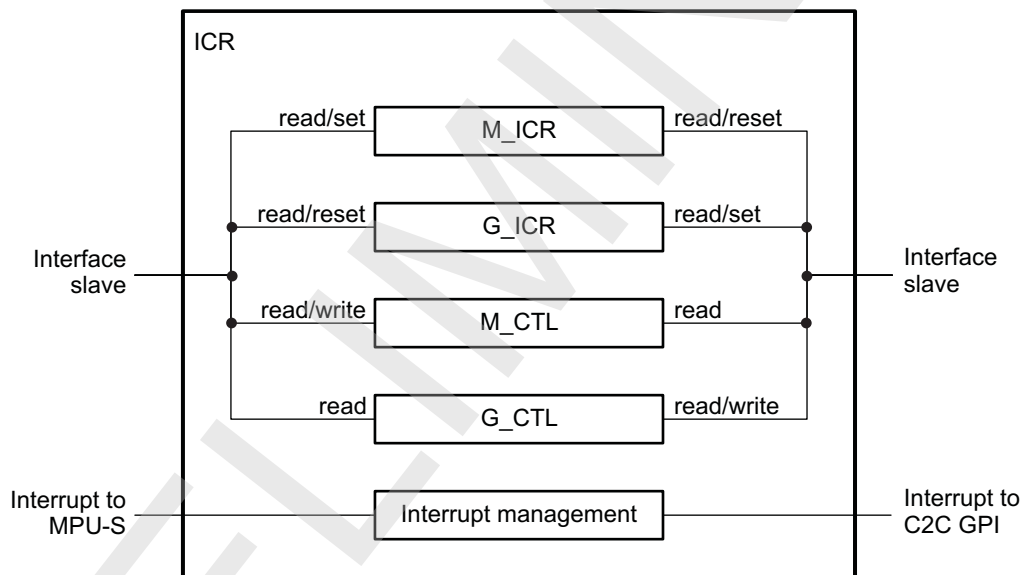
### 14.6.1 intersystem Communication Register Overview

The intersystem communication register (ICR) module synchronizes microprocessor unit (MPU) and modem activity, relying on flag mechanisms that are shared by both systems. Basically, the MPU sets a flag that is reset by the modem and vice versa. For a given direction, all flags are set by the source subsystem and are reset by the destination subsystem when it wants to. Each subsystem can read flags without resetting them. To avoid continuous polling on the ICRs, each subsystem can receive an interrupt when an operation that is related to it (set or reset) completes.

### 14.6.2 ICR Functional Description

The ICR is used when the device is connected to an external initiator device connected on the C2C port. Otherwise, the ICR is not used. The MPU and the external systems communicate with the help of two ICRs. One ICR (**G\_ICR**) is used by the external system to inform the device MPU subsystem of an event and to send commands to the MPU-S. In the same way, the other ICR (**M\_ICR**) is used by the device MPU subsystem to inform the external system of an event and to send commands to the MPU subsystem. The bits definition of the **G\_ICR** and **M\_ICR** registers are defined by software.

Figure 14-7. ICR Block Diagram



c2c-005

#### 14.6.2.1 ICR Clocking Scheme

The ICR function operates from the L4-core interface clock.

#### 14.6.2.2 Flag Management

Flags are used to notify the other subsystem of an event. Software determines the meaning of each flag. The following communication sequence from the MPU-S to the modem-S is typical:

- The MPU-S writes one or more flags in the **M\_ICR** register.
- An interrupt is sent to the modem-S (if enabled).
- The modem-S reads the **M\_ICR** register to check which flags are set.
- The modem-S resets one or more flags in the **M\_ICR** register.
- An interrupt is sent to the MPU-S (if enabled).
- The MPU-S reads the **M\_ICR** register to check which flags are reset.

### 14.6.2.3 Setting and Resetting a Flag

To set a flag, the MPU-S must set the desired bit of the **M\_ICR** register to 1. Setting a bit of **M\_ICR** to 0 does not change its content (that is, if previous value was 1, bit value remains 1 after the write access, [modem-S does the same thing with **G\_ICR** register]).

To reset a flag, the MPU-S must set the desired bit of the **G\_ICR** register to 1 (that is, if previous value of the bit was 1, then it becomes 0 else it remains 0, [modem-S does the same thing with **M\_ICR** register]).

### 14.6.2.4 Interrupt Management

Interrupt behavior is the same on both sides (MPU and modem).

Two kinds of events can generate an interrupt to the MPU:

- The modem sets a flag in **G\_ICR** register.
- The modem resets a flag in **M\_ICR** register.

Interrupt generation is controlled by the **M\_CTL** register.

Interrupt generation mechanism to the modem is exactly the same as on the MPU side:

- The MPU sets a flag in the **M\_ICR** register.
- The MPU resets a flag in the **G\_ICR** register.

Interrupt generation is controlled by the **G\_CTL** register. To acknowledge an interrupt, the MPU must clear the **M\_IRQ\_STATUS** register by setting to 1 the bits that caused the interrupt. Acknowledgment on the modem side follows the same procedure with the **G\_IRQ\_STATUS** register.

## 14.6.3 ICR Register Manual

### 14.6.3.1 ICR Instance Summary

**Table 14-59. C2C Instance Summary**

Module Name	Module Base Address	Size
ICR_MPU	0x4A0B 6000	4KB
ICR_MDM	0x4A0C D000	4KB

14.6.3.2 ICR\_MDM and ICR\_MPU Registers

14.6.3.2.1 ICR Register Summary

Table 14-60 lists the mapping summary of the ICR registers.

Table 14-60. ICR Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ICR_MDM Base Address	ICR_MPU Base Address
ICR_REVISION	R	32	0x0000 0000	0x4A0C D000	0x4A0B 6000
SYS_CONFIG	RW	32	0x0000 0010	0x4A0C D010	0x4A0B 6010
SYS_STATUS	R	32	0x0000 0014	0x4A0C D014	0x4A0B 6014
M_IRQ_STATUS	RW	32	0x0000 0018	0x4A0C D018	0x4A0B 6018
G_IRQ_STATUS	RW	32	0x0000 001C	0x4A0C D01C	0x4A0B 601C
M_ICR	RW	32	0x0000 0020	0x4A0C D020	0x4A0B 6020
G_ICR	RW	32	0x0000 0024	0x4A0C D024	0x4A0B 6024
M_CTL	R	32	0x0000 0028	0x4A0C D028	0x4A0B 6028
G_CTL	RW	32	0x0000 002C	0x4A0C D02C	0x4A0B 602C

14.6.3.2.2 ICR Register Description

Table 14-61 through Table 14-77 describe each ICR register.

Table 14-61. ICR\_REVISION

Address Offset	0x0000 0000	
Physical Address	See Table 14-60.	Instance Table 14-60
Description	This register contains the IP revision code.	
Type	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

Table 14-62. Register Call Summary for Register ICR\_REVISION

Intersystem Communication Register (ICR)

- ICR Register Summary: [0]

Table 14-63. SYS\_CONFIG

Address Offset	0x0000 0010	
Physical Address	See Table 14-60.	Instance Table 14-60
Description	This register allows controlling various parameters of the OCP interface.	
Type	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE	RESERVED	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Read returns 0	R	0x0000000
4:3	IDLEMODE	Slave interface power management, req/ack control 0x0: Force-idle 0x1: No-idle 0x3: Reserved 0x2: Smart-idle	RW	0x0
2	RESERVED	Read returns 0	R	0
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always return 0. 0: Normal mode 1: The module is reset	RW	0
0	AUTOIDLE	Internal OCP clock gating 0: OCP clock is free running 1: Automatic OCP clock gating strategy is applied based on the OCP interface activity	RW	0

**Table 14-64. Register Call Summary for Register SYS\_CONFIG**

Intersystem Communication Register (ICR)

- [ICR Register Summary: \[0\]](#)

**Table 14-65. SYS\_STATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	<a href="#">Table 14-60</a>
<b>Physical Address</b>	See <a href="#">Table 14-60</a> .		
<b>Description</b>	This register provides status information about the module, excluding the interrupt status information.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													RESETDONE		

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	RESETDONE	Internal reset monitoring 0: Internal module reset ongoing 1: Reset completed Read 0x1: Reset completed Read 0x0: Internal module reset in ongoing	R	0

**Table 14-66. Register Call Summary for Register SYS\_STATUS**

Intersystem Communication Register (ICR)

- [ICR Register Summary: \[0\]](#)

**Table 14-67. M\_IRQ\_STATUS**

<b>Address Offset</b>	0x0000 0018
<b>Physical Address</b>	See <a href="#">Table 14-60</a> .
<b>Description</b>	The interrupt status register regroups all the status of the module internal events that can generate an interrupt - Write 1 to a given bit resets this bit.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FLAG_SET		FLAG_RESET													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads returns 0	R	0x0000 0000
1	FLAG_SET	Flag has been set in <a href="#">G_ICR</a> register. Write 0x0: Received command (Rx path) status bit unchanged Write 0x1: Received command (Rx path) status bit is reset Read 0x1: Received command (Rx path) is true ("pending") Read 0x0: Received command (Rx path) false	RW	0
0	FLAG_RESET	Flag has been reset in <a href="#">M_ICR</a> register. Write 0x0: Received command (Rx path) status bit unchanged Write 0x1: Received command (Rx path) status bit is reset Read 0x1: Received command (Rx path) is true ("pending") Read 0x0: Received command (Rx path) false	RW	0

**Table 14-68. Register Call Summary for Register M\_IRQ\_STATUS**

Intersystem Communication Register (ICR)

- [Interrupt Management: \[0\]](#)
- [ICR Register Summary: \[1\]](#)

**Table 14-69. G\_IRQ\_STATUS**

<b>Address Offset</b>	0x0000 001C
<b>Physical Address</b>	See <a href="#">Table 14-60</a> .
<b>Description</b>	The interrupt status register regroups all the status of the module internal events that can generate an interrupt - Write 1 to a given bit resets this bit.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FLAG_SET		FLAG_RESET													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads returns 0	R	0x0000 0000
1	FLAG_SET	Flag has been set in <a href="#">M_ICR</a> register. Write 0x0: Received command (Rx path) status bit unchanged Write 0x1: Received command (Rx path) status bit is reset Read 0x1: Received command (Rx path) is true ("pending") Read 0x0: Received command (Rx path) false	RW W1toClr	0
0	FLAG_RESET	Flag has been reset in <a href="#">G_ICR</a> register. Write 0x0: Received command (Rx path) status bit unchanged Write 0x1: Received command (Rx path) status bit is reset Read 0x1: Received command (Rx path) is true ("pending") Read 0x0: Received command (Rx path) false	RW W1toClr	0

**Table 14-70. Register Call Summary for Register G\_IRQ\_STATUS**

Intersystem Communication Register (ICR)

- [Interrupt Management: \[0\]](#)
- [ICR Register Summary: \[1\]](#)

**Table 14-71. M\_ICR**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	<a href="#">Table 14-60</a>																																																																
<b>Physical Address</b>	See <a href="#">Table 14-60</a> .																																																																		
<b>Description</b>	Flags are used to notify an event to the other subsystem. The meaning of each flag is software dependent.																																																																		
<b>Type</b>	RW																																																																		
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td style="background-color:yellow;">23</td><td style="background-color:yellow;">22</td><td style="background-color:yellow;">21</td><td style="background-color:yellow;">20</td><td style="background-color:yellow;">19</td><td style="background-color:yellow;">18</td><td style="background-color:yellow;">17</td><td style="background-color:yellow;">16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td style="background-color:yellow;">7</td><td style="background-color:yellow;">6</td><td style="background-color:yellow;">5</td><td style="background-color:yellow;">4</td><td style="background-color:yellow;">3</td><td style="background-color:yellow;">2</td><td style="background-color:yellow;">1</td><td style="background-color:yellow;">0</td> </tr> <tr> <td colspan="32">FLAGS</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	FLAGS																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
FLAGS																																																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																															
31:0	FLAGS	Meaning is software dependent.	RW W1toClr	0x0000 0000																																																															

**Table 14-72. Register Call Summary for Register M\_ICR**

Intersystem Communication Register (ICR)

- [ICR Functional Description: \[0\] \[1\]](#)
- [Flag management: \[2\] \[3\] \[4\] \[5\]](#)
- [Setting and Resetting a Flag: \[6\] \[7\] \[8\]](#)
- [Interrupt Management: \[9\] \[10\]](#)
- [ICR Register Summary: \[11\]](#)
- [ICR Register Description: \[12\] \[13\]](#)



**Table 14-73. G\_ICR**

<b>Address Offset</b>	0x0000 0024	
<b>Physical Address</b>	See <a href="#">Table 14-60</a> .	<b>Instance</b> <a href="#">Table 14-60</a>
<b>Description</b>	Flags are used to notify an event to the other subsystem. The meaning of each flag is software dependent.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLAGS																															

Bits	Field Name	Description	Type	Reset
31:0	FLAGS	Meaning is software dependent.	RW W1toSet	0x0000 0000

**Table 14-74. Register Call Summary for Register G\_ICR**

Intersystem Communication Register (ICR)

- [ICR Functional Description: \[0\] \[1\]](#)
- [Setting and Resetting a Flag: \[2\] \[3\]](#)
- [Interrupt Management: \[4\] \[5\]](#)
- [ICR Register Summary: \[6\]](#)
- [ICR Register Description: \[7\] \[8\]](#)

**Table 14-75. M\_CTL**

<b>Address Offset</b>	0x0000 0028	
<b>Physical Address</b>	See <a href="#">Table 14-60</a> .	<b>Instance</b> <a href="#">Table 14-60</a>
<b>Description</b>	Enables interrupts to subsystems.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
																										G_ICR_INTEN	M_ICR_INTEN				

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	G_ICR_INTEN	Enable interrupt to MPU-S 0: No interrupt is generated 1: An interrupt is generated when Modem-S sets a flag to MPU-S	R	0
0	M_ICR_INTEN	Enable interrupt to MPU-S 0: No interrupt is generated 1: An interrupt is generated when Modem-S resets a flag set by MPU-S	R	0

**Table 14-76. Register Call Summary for Register M\_CTL**

Intersystem Communication Register (ICR)

- [Interrupt Management: \[0\]](#)
- [ICR Register Summary: \[1\]](#)

**Table 14-77. G\_CTL**

<b>Address Offset</b>	0x0000 002C		
<b>Physical Address</b>	See <a href="#">Table 14-60</a> .	<b>Instance</b>	<a href="#">Table 14-60</a>
<b>Description</b>	Enables interrupts to subsystems.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																M_ICR_INTEN		G_ICR_INTEN													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		RW	0x0000 0000
1	M_ICR_INTEN	Enable interrupt to Modem-S 0: No interrupt is generated 1: An interrupt is generated when MPU-S sets a flag to Modem-S	RW	0
0	G_ICR_INTEN	Enable interrupt to Modem-S 0: No interrupt is generated 1: An interrupt is generated when MPU-S resets a flag set by Modem-S	RW	0

**Table 14-78. Register Call Summary for Register G\_CTL**

Intersystem Communication Register (ICR)

- [Interrupt Management: \[0\]](#)
- [ICR Register Summary: \[1\]](#)

## 14.7 C2C Register Manual

### 14.7.1 C2C Instance Summary

**Table 14-79. C2C Instance Summary**

Module Name	Base Address	Size
C2C	0x5C000000	64MB

### 14.7.2 C2C Registers

#### 14.7.2.1 C2C Register Summary

Table 14-80 summarizes the C2C registers.

**Table 14-80. C2C Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	C2C L3 Physical Address
C2C_REVISION	R	32	0x0000 0000	0x5C000000
C2C_SYSCONFIG	R	32	0x0000 0004	0x5C000004
C2C_SYSSTATUS	R	32	0x0000 0008	0x5C000008
C2C_PORTCONFIG	RW	32	0x0000 000C	0x5C00000C
C2C_MIRRORMODE	RW	32	0x0000 0010	0x5C000010
C2C_IRQ_RAW_STATUS_0	RW	32	0x0000 0014	0x5C000014
C2C_IRQ_RAW_STATUS_1	RW	32	0x0000 0018	0x5C000018
C2C_IRQ_ENABLE_STATUS_0	RW	32	0x0000 001C	0x5C00001C
C2C_IRQ_ENABLE_STATUS_1	RW	32	0x0000 0020	0x5C000020
C2C_IRQ_ENABLE_SET_0	RW	32	0x0000 0024	0x5C000024
C2C_IRQ_ENABLE_SET_1	RW	32	0x0000 0028	0x5C000028
C2C_IRQ_ENABLE_CLEAR_0	RW	32	0x0000 002C	0x5C00002C
C2C_IRQ_ENABLE_CLEAR_1	RW	32	0x0000 0030	0x5C000030
C2C_FCLK_FREQ	RW	32	0x0000 0040	0x5C000040
C2C_RX_MAX_FREQ	RW	32	0x0000 0044	0x5C000044
C2C_TX_MAX_FREQ	RW	32	0x0000 0048	0x5C000048
C2C_RX_MAX_FREQ_ACK	RW	32	0x0000 004C	0x5C00004C
C2C_WAKE_REQ	RW	32	0x0000 0050	0x5C000050
C2C_WAKE_ACK	RW	32	0x0000 0054	0x5C000054
C2C_STANDBY	RW	32	0x0000 0060	0x5C000060
C2C_STANDBY_IN	RW	32	0x0000 0064	0x5C000064
C2C_WAIT	RW	32	0x0000 0068	0x5C000068
C2C_GENI_CONTROL	RW	32	0x0000 0070	0x5C000070
C2C_GENI_MASK	RW	32	0x0000 0074	0x5C000074
C2C_GENO_STATUS	RW	32	0x0000 0080	0x5C000080
C2C_GENO_INTERRUPT	RW	32	0x0000 0084	0x5C000084
C2C_GENO_LEVEL	RW	32	0x0000 0088	0x5C000088

#### 14.7.2.2 C2C Register Description

Table 14-81 through Table 14-133 describe the individual C2C registers in detail.

**Table 14-81. C2C\_REVISION**

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .
<b>Description</b>	IP Revision identifier (X.Y.R) Used by software to track features, bugs, and compatibility
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

**Table 14-82. Register Call Summary for Register C2C\_REVISION**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-83. C2C\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0004
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .
<b>Description</b>	System configuration register (not used)
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	R	0x0000 0000

**Table 14-84. Register Call Summary for Register C2C\_SYSCONFIG**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-85. C2C\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0008
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .
<b>Description</b>	Reserved
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	R	0x0000 0000

**Table 14-86. Register Call Summary for Register C2C\_SYSSTATUS**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-87. C2C\_PORTCONFIG**

<b>Address Offset</b>	0x0000 000C		
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .	<b>Instance</b>	C2C
<b>Description</b>	This register defines the widths of the TX and RX buses on the C2C interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_WIDTH				TX_WIDTH											

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:4	RX_WIDTH	0000: Bus width is set to 8 bits (default mode). 0001: Bus width is set to 10 bits (possible only if TX = 8). 0010: Bus width is set to 16 bits. Other values are reserved, and should not be used.	RW	0x0
3:0	TX_WIDTH	0000: Bus width is set to 8 bits. 0001: Bus width is set to 10 bits (possible only if RX = 8). 0010: Bus width is set to 16 bits (default mode). Other values are reserved, and shall not be used.	RW	0x2

**Table 14-88. Register Call Summary for Register C2C\_PORTCONFIG**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-89. C2C\_MIRRORMODE**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .	<b>Instance</b>	C2C
<b>Description</b>	This register sets the C2C L3 in mirror mode. See debug section.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	MIRRORMODE														

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	MIRRORMODE	Mirror mode. 0x0: Functional default mode 0x1: Mirror mode for debug	RW	0

**Table 14-90. Register Call Summary for Register C2C\_MIRRORMODE**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-91. C2C\_IRQ\_RAW\_STATUS\_0**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .	<b>Instance</b>	C2C
<b>Description</b>	Interrupt status register (OCP-compliant IRQ line) for C2c_Sscm_Irq(0), regardless of enable settings		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WAIT_IRQ		WAKE_ACK_IRQ		RX_MAX_FREQ_ACK_IRQ		TX_FREQ_IRQ									

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	WAIT_IRQ	Reading 1 indicates WAIT value in WAIT register has been modified by distant C2C. Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0x0
2	WAKE_ACK_IRQ	Reading 1 indicates WAKE_ACK value in the WAKE_ACK register has been modified by distant C2C. Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0x0
1	RX_MAX_FREQ_ACK_IRQ	Reading 1 indicates GPI is not turned on OR RX_MAX_FREQ_ACK value in the RX_MAX_FREQ_ACK register equal RX_MAX_FREQ value in the RX_MAX_FREQ register. Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0x0
0	TX_FREQ_IRQ	Reading 1 indicates that the FCLK_FREQ value in the FCLK_FREQ register has been updated AND the SSCM has finished updating the new division for the TX clock. Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0x0

**Table 14-92. Register Call Summary for Register C2C\_IRQ\_RAW\_STATUS\_0**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-93. C2C\_IRQ\_RAW\_STATUS\_1**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GENO_IRQ_31	GENO_IRQ_30	GENO_IRQ_29	GENO_IRQ_28	GENO_IRQ_27	GENO_IRQ_26	GENO_IRQ_25	GENO_IRQ_24	GENO_IRQ_23	GENO_IRQ_22	GENO_IRQ_21	GENO_IRQ_20	GENO_IRQ_19	GENO_IRQ_18	GENO_IRQ_17	GENO_IRQ_16	GENO_IRQ_15	GENO_IRQ_14	GENO_IRQ_13	GENO_IRQ_12	GENO_IRQ_11	GENO_IRQ_10	GENO_IRQ_9	GENO_IRQ_8	GENO_IRQ_7	GENO_IRQ_6	GENO_IRQ_5	GENO_IRQ_4	GENO_IRQ_3	GENO_IRQ_2	GENO_IRQ_1	GENO_IRQ_0

Bits	Field Name	Description	Type	Reset
31	GENO_IRQ_31	Reading 1 indicates C2c_Sscm_GENO (31) signal and GENO_31 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
30	GENO_IRQ_30	Reading 1 indicates C2c_Sscm_GENO (30) signal and GENO_30 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
29	GENO_IRQ_29	Reading 1 indicates C2c_Sscm_GENO (29) signal and GENO_29 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
28	GENO_IRQ_28	Reading 1 indicates C2c_Sscm_GENO (28) signal and GENO_28 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
27	GENO_IRQ_27	Reading 1 indicates C2c_Sscm_GENO (27) signal and GENO_27 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
26	GENO_IRQ_26	Reading 1 indicates C2c_Sscm_GENO (26) signal and GENO_26 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
25	GENO_IRQ_25	Reading 1 indicates C2c_Sscm_GENO (25) signal and GENO_25 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
24	GENO_IRQ_24	Reading 1 indicates C2c_Sscm_GENO (24) signal and GENO_24 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
23	GENO_IRQ_23	Reading 1 indicates C2c_Sscm_GENO (23) signal and GENO_23 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
22	GENO_IRQ_22	Reading 1 indicates C2c_Sscm_GENO (22) signal and GENO_22 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
21	GENO_IRQ_21	Reading 1 indicates C2c_Sscm_GENO (21) signal and GENO_21 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
20	GENO_IRQ_20	Reading 1 indicates C2c_Sscm_GENO (20) signal and GENO_20 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
19	GENO_IRQ_19	Reading 1 indicates C2c_Sscm_GENO (19) signal and GENO_19 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
18	GENO_IRQ_18	Reading 1 indicates C2c_Sscm_GENO (18) signal and GENO_18 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
17	GENO_IRQ_17	Reading 1 indicates C2c_Sscm_GENO (17) signal and GENO_17 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
16	GENO_IRQ_16	Reading 1 indicates C2c_Sscm_GENO (16) signal and GENO_16 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
15	GENO_IRQ_15	Reading 1 indicates C2c_Sscm_GENO (15) signal and GENO_15 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
14	GENO_IRQ_14	Reading 1 indicates C2c_Sscm_GENO (14) signal and GENO_14 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
13	GENO_IRQ_13	Reading 1 indicates C2c_Sscm_GENO (13) signal and GENO_13 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0



Bits	Field Name	Description	Type	Reset
12	GENO_IRQ_12	Reading 1 indicates C2c_Sscm_GENO (12) signal and GENO_12 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
11	GENO_IRQ_11	Reading 1 indicates C2c_Sscm_GENO (11) signal and GENO_11 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
10	GENO_IRQ_10	Reading 1 indicates C2c_Sscm_GENO (10) signal and GENO_10 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
9	GENO_IRQ_9	Reading 1 indicates C2c_Sscm_GENO (9) signal and GENO_9 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
8	GENO_IRQ_8	Reading 1 indicates C2c_Sscm_GENO (8) signal and GENO_8 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
7	GENO_IRQ_7	Reading 1 indicates C2c_Sscm_GENO (7) signal and GENO_7 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
6	GENO_IRQ_6	Reading 1 indicates C2c_Sscm_GENO (6) signal and GENO_6 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
5	GENO_IRQ_5	Reading 1 indicates C2c_Sscm_GENO (5) signal and GENO_5 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
4	GENO_IRQ_4	Reading 1 indicates C2c_Sscm_GENO (4) signal and GENO_4 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
3	GENO_IRQ_3	Reading 1 indicates C2c_Sscm_GENO (3) signal and GENO_3 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
2	GENO_IRQ_2	Reading 1 indicates C2c_Sscm_GENO (2) signal and GENO_2 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
1	GENO_IRQ_1	Reading 1 indicates C2c_Sscm_GENO (1) signal and GENO_1 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
0	GENO_IRQ_0	Reading 1 indicates C2c_Sscm_GENO (0) signal and GENO_0 value in GENO_STATUS register have been modified by distant C2C.  Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0

**Table 14-94. Register Call Summary for Register C2C\_IRQ\_RAW\_STATUS\_1**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-95. C2C\_IRQ\_ENABLE\_STATUS\_0**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WAIT_IRQ			WAKE_ACK_IRQ		RX_MAX_FREQ_ACK_IRQ		TX_FREQ_IRQ								

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	WAIT_IRQ	Reading 1 indicates WAIT value in WAIT register has been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
2	WAKE_ACK_IRQ	Reading 1 indicates WAKE_ACK value in the WAKE_ACK register has been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
1	RX_MAX_FREQ_ACK_IRQ	Reading 1 indicates GPI is not turned on OR RX_MAX_FREQ_ACK value in the RX_MAX_FREQ_ACK register equal RX_MAX_FREQ value in the RX_MAX_FREQ register.	RW W1toClr	0
0	TX_FREQ_IRQ	Reading 1 indicates that the FCLK_FREQ value in the FCLK_FREQ register has been updated AND the SSCM has finished updating the new division for the Tx clock.	RW W1toClr	0

**Table 14-96. Register Call Summary for Register C2C\_IRQ\_ENABLE\_STATUS\_0**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-97. C2C\_IRQ\_ENABLE\_STATUS\_1**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GENO_IRQ_31	GENO_IRQ_30	GENO_IRQ_29	GENO_IRQ_28	GENO_IRQ_27	GENO_IRQ_26	GENO_IRQ_25	GENO_IRQ_24	GENO_IRQ_23	GENO_IRQ_22	GENO_IRQ_21	GENO_IRQ_20	GENO_IRQ_19	GENO_IRQ_18	GENO_IRQ_17	GENO_IRQ_16	GENO_IRQ_15	GENO_IRQ_14	GENO_IRQ_13	GENO_IRQ_12	GENO_IRQ_11	GENO_IRQ_10	GENO_IRQ_9	GENO_IRQ_8	GENO_IRQ_7	GENO_IRQ_6	GENO_IRQ_5	GENO_IRQ_4	GENO_IRQ_3	GENO_IRQ_2	GENO_IRQ_1	GENO_IRQ_0

Bits	Field Name	Description	Type	Reset
31	GENO_IRQ_31	Reading 1 indicates C2c_Sscm_GENO (31) signal and GENO_31 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
30	GENO_IRQ_30	Reading 1 indicates C2c_Sscm_GENO (30) signal and GENO_30 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
29	GENO_IRQ_29	Reading 1 indicates C2c_Sscm_GENO (29) signal and GENO_29 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
28	GENO_IRQ_28	Reading 1 indicates C2c_Sscm_GENO (28) signal and GENO_28 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
27	GENO_IRQ_27	Reading 1 indicates C2c_Sscm_GENO (27) signal and GENO_27 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
26	GENO_IRQ_26	Reading 1 indicates C2c_Sscm_GENO (26) signal and GENO_26 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
25	GENO_IRQ_25	Reading 1 indicates C2c_Sscm_GENO (25) signal and GENO_25 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
24	GENO_IRQ_24	Reading 1 indicates C2c_Sscm_GENO (24) signal and GENO_24 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
23	GENO_IRQ_23	Reading 1 indicates C2c_Sscm_GENO (23) signal and GENO_23 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
22	GENO_IRQ_22	Reading 1 indicates C2c_Sscm_GENO (22) signal and GENO_22 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
21	GENO_IRQ_21	Reading 1 indicates C2c_Sscm_GENO (21) signal and GENO_21 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
20	GENO_IRQ_20	Reading 1 indicates C2c_Sscm_GENO (20) signal and GENO_20 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
19	GENO_IRQ_19	Reading 1 indicates C2c_Sscm_GENO (19) signal and GENO_19 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
18	GENO_IRQ_18	Reading 1 indicates C2c_Sscm_GENO (18) signal and GENO_18 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
17	GENO_IRQ_17	Reading 1 indicates C2c_Sscm_GENO (17) signal and GENO_17 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
16	GENO_IRQ_16	Reading 1 indicates C2c_Sscm_GENO (16) signal and GENO_16 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
15	GENO_IRQ_15	Reading 1 indicates C2c_Sscm_GENO (15) signal and GENO_15 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
14	GENO_IRQ_14	Reading 1 indicates C2c_Sscm_GENO (14) signal and GENO_14 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
13	GENO_IRQ_13	Reading 1 indicates C2c_Sscm_GENO (13) signal and GENO_13 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
12	GENO_IRQ_12	Reading 1 indicates C2c_Sscm_GENO (12) signal and GENO_12 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
11	GENO_IRQ_11	Reading 1 indicates C2c_Sscm_GENO (11) signal and GENO_11 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
10	GENO_IRQ_10	Reading 1 indicates C2c_Sscm_GENO (10) signal and GENO_10 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
9	GENO_IRQ_9	Reading 1 indicates C2c_Sscm_GENO (9) signal and GENO_9 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
8	GENO_IRQ_8	Reading 1 indicates C2c_Sscm_GENO (8) signal and GENO_8 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
7	GENO_IRQ_7	Reading 1 indicates C2c_Sscm_GENO (7) signal and GENO_7 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
6	GENO_IRQ_6	Reading 1 indicates C2c_Sscm_GENO (6) signal and GENO_6 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
5	GENO_IRQ_5	Reading 1 indicates C2c_Sscm_GENO (5) signal and GENO_5 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
4	GENO_IRQ_4	Reading 1 indicates C2c_Sscm_GENO (4) signal and GENO_4 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
3	GENO_IRQ_3	Reading 1 indicates C2c_Sscm_GENO (3) signal and GENO_3 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
2	GENO_IRQ_2	Reading 1 indicates C2c_Sscm_GENO (2) signal and GENO_2 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
1	GENO_IRQ_1	Reading 1 indicates C2c_Sscm_GENO (1) signal and GENO_1 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
0	GENO_IRQ_0	Reading 1 indicates C2c_Sscm_GENO (0) signal and GENO_0 value in GENO_STATUS register have been modified by distant C2C and interrupt source is not masked. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0

**Table 14-98. Register Call Summary for Register C2C\_IRQ\_ENABLE\_STATUS\_1**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-99. C2C\_IRQ\_ENABLE\_SET\_0**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							WAIT_IRQ_EN	WAKE_ACK_IRQ_EN	RX_MAX_FREQ_ACK_IRQ_EN	TX_FREQ_IRQ_EN					

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	WAIT_IRQ_EN	Interrupt enable for WAIT IRQ. Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
2	WAKE_ACK_IRQ_EN	Interrupt enable for WAKE_ACK IRQ. Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
1	RX_MAX_FREQ_ACK_IRQ_EN	Interrupt enable for RX_MAX_FREQ_ACK IRQ. Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0
0	TX_FREQ_IRQ_EN	Interrupt enable for TX_FREQ IRQ. Writing 1 to this bit sets the bit. Writing 0 has no effect.	RW W1toSet	0

**Table 14-100. Register Call Summary for Register C2C\_IRQ\_ENABLE\_SET\_0**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-101. C2C\_IRQ\_ENABLE\_SET\_1**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GENO_IRQ_31_EN	GENO_IRQ_30_EN	GENO_IRQ_29_EN	GENO_IRQ_28_EN	GENO_IRQ_27_EN	GENO_IRQ_26_EN	GENO_IRQ_25_EN	GENO_IRQ_24_EN	GENO_IRQ_23_EN	GENO_IRQ_22_EN	GENO_IRQ_21_EN	GENO_IRQ_20_EN	GENO_IRQ_19_EN	GENO_IRQ_18_EN	GENO_IRQ_17_EN	GENO_IRQ_16_EN	GENO_IRQ_15_EN	GENO_IRQ_14_EN	GENO_IRQ_13_EN	GENO_IRQ_12_EN	GENO_IRQ_11_EN	GENO_IRQ_10_EN	GENO_IRQ_9_EN	GENO_IRQ_8_EN	GENO_IRQ_7_EN	GENO_IRQ_6_EN	GENO_IRQ_5_EN	GENO_IRQ_4_EN	GENO_IRQ_3_EN	GENO_IRQ_2_EN	GENO_IRQ_1_EN	GENO_IRQ_0_EN

Bits	Field Name	Description	Type	Reset
31	GENO_IRQ_31_EN	Interrupt enable for GENO_IRQ_31 IRQ.	RW W1toSet	0
30	GENO_IRQ_30_EN	Interrupt enable for GENO_IRQ_30 IRQ.	RW W1toSet	0

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
29	GENO_IRQ_29_EN	Interrupt enable for GENO_IRQ_29 IRQ.	RW W1toSet	0
28	GENO_IRQ_28_EN	Interrupt enable for GENO_IRQ_28 IRQ.	RW W1toSet	0
27	GENO_IRQ_27_EN	Interrupt enable for GENO_IRQ_27 IRQ.	RW W1toSet	0
26	GENO_IRQ_26_EN	Interrupt enable for GENO_IRQ_26 IRQ.	RW W1toSet	0
25	GENO_IRQ_25_EN	Interrupt enable for GENO_IRQ_25 IRQ.	RW W1toSet	0
24	GENO_IRQ_24_EN	Interrupt enable for GENO_IRQ_24 IRQ.	RW W1toSet	0
23	GENO_IRQ_23_EN	Interrupt enable for GENO_IRQ_23 IRQ.	RW W1toSet	0
22	GENO_IRQ_22_EN	Interrupt enable for GENO_IRQ_22 IRQ.	RW W1toSet	0
21	GENO_IRQ_21_EN	Interrupt enable for GENO_IRQ_21 IRQ.	RW W1toSet	0
20	GENO_IRQ_20_EN	Interrupt enable for GENO_IRQ_20 IRQ.	RW W1toSet	0
19	GENO_IRQ_19_EN	Interrupt enable for GENO_IRQ_19 IRQ.	RW W1toSet	0
18	GENO_IRQ_18_EN	Interrupt enable for GENO_IRQ_18 IRQ.	RW W1toSet	0
17	GENO_IRQ_17_EN	Interrupt enable for GENO_IRQ_17 IRQ.	RW W1toSet	0
16	GENO_IRQ_16_EN	Interrupt enable for GENO_IRQ_16 IRQ.	RW W1toSet	0
15	GENO_IRQ_15_EN	Interrupt enable for GENO_IRQ_15 IRQ.	RW W1toSet	0
14	GENO_IRQ_14_EN	Interrupt enable for GENO_IRQ_14 IRQ.	RW W1toSet	0
13	GENO_IRQ_13_EN	Interrupt enable for GENO_IRQ_13 IRQ.	RW W1toSet	0
12	GENO_IRQ_12_EN	Interrupt enable for GENO_IRQ_12 IRQ.	RW W1toSet	0
11	GENO_IRQ_11_EN	Interrupt enable for GENO_IRQ_11 IRQ.	RW W1toSet	0
10	GENO_IRQ_10_EN	Interrupt enable for GENO_IRQ_10 IRQ.	RW W1toSet	0
9	GENO_IRQ_9_EN	Interrupt enable for GENO_IRQ_9 IRQ.	RW W1toSet	0
8	GENO_IRQ_8_EN	Interrupt enable for GENO_IRQ_8 IRQ.	RW W1toSet	0
7	GENO_IRQ_7_EN	Interrupt enable for GENO_IRQ_7 IRQ.	RW W1toSet	0
6	GENO_IRQ_6_EN	Interrupt enable for GENO_IRQ_6 IRQ.	RW W1toSet	0
5	GENO_IRQ_5_EN	Interrupt enable for GENO_IRQ_5 IRQ.	RW W1toSet	0
4	GENO_IRQ_4_EN	Interrupt enable for GENO_IRQ_4 IRQ.	RW W1toSet	0
3	GENO_IRQ_3_EN	Interrupt enable for GENO_IRQ_3 IRQ.	RW W1toSet	0
2	GENO_IRQ_2_EN	Interrupt enable for GENO_IRQ_2 IRQ.	RW W1toSet	0
1	GENO_IRQ_1_EN	Interrupt enable for GENO_IRQ_1 IRQ.	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
0	GENO_IRQ_0_EN	Interrupt enable for GENO_IRQ_0 IRQ.	RW W1toSet	0

**Table 14-102. Register Call Summary for Register C2C\_IRQ\_ENABLE\_SET\_1**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-103. C2C\_IRQ\_ENABLE\_CLEAR\_0**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WAIT_IRQ_EN	WAKE_ACK_IRQ_EN	RX_MAX_FREQ_ACK_IRQ_EN	TX_FREQ_IRQ_EN												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	WAIT_IRQ_EN	Interrupt enable for WAIT IRQ.	RW W1toClr	0
2	WAKE_ACK_IRQ_EN	Interrupt enable for WAKE_ACK IRQ.	RW W1toClr	0
1	RX_MAX_FREQ_ACK_IRQ_EN	Interrupt enable for RX_MAX_FREQ_ACK IRQ.	RW W1toClr	0
0	TX_FREQ_IRQ_EN	Interrupt enable for TX_FREQ IRQ.	RW W1toClr	0

**Table 14-104. Register Call Summary for Register C2C\_IRQ\_ENABLE\_CLEAR\_0**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-105. C2C\_IRQ\_ENABLE\_CLEAR\_1**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GENO_IRQ_31_EN	GENO_IRQ_30_EN	GENO_IRQ_29_EN	GENO_IRQ_28_EN	GENO_IRQ_27_EN	GENO_IRQ_26_EN	GENO_IRQ_25_EN	GENO_IRQ_24_EN	GENO_IRQ_23_EN	GENO_IRQ_22_EN	GENO_IRQ_21_EN	GENO_IRQ_20_EN	GENO_IRQ_19_EN	GENO_IRQ_18_EN	GENO_IRQ_17_EN	GENO_IRQ_16_EN	GENO_IRQ_15_EN	GENO_IRQ_14_EN	GENO_IRQ_13_EN	GENO_IRQ_12_EN	GENO_IRQ_11_EN	GENO_IRQ_10_EN	GENO_IRQ_9_EN	GENO_IRQ_8_EN	GENO_IRQ_7_EN	GENO_IRQ_6_EN	GENO_IRQ_5_EN	GENO_IRQ_4_EN	GENO_IRQ_3_EN	GENO_IRQ_2_EN	GENO_IRQ_1_EN	GENO_IRQ_0_EN

Bits	Field Name	Description	Type	Reset
31	GENO_IRQ_31_EN	Interrupt enable for GENO_IRQ_31 IRQ. Writing 1 to this bit clears the bit. Writing 0 has no effect.	RW W1toClr	0
30	GENO_IRQ_30_EN	Interrupt enable for GENO_IRQ_30 IRQ.	RW W1toClr	0
29	GENO_IRQ_29_EN	Interrupt enable for GENO_IRQ_29 IRQ.	RW W1toClr	0
28	GENO_IRQ_28_EN	Interrupt enable for GENO_IRQ_28 IRQ.	RW W1toClr	0
27	GENO_IRQ_27_EN	Interrupt enable for GENO_IRQ_27 IRQ.	RW W1toClr	0
26	GENO_IRQ_26_EN	Interrupt enable for GENO_IRQ_26 IRQ.	RW W1toClr	0
25	GENO_IRQ_25_EN	Interrupt enable for GENO_IRQ_25 IRQ.	RW W1toClr	0
24	GENO_IRQ_24_EN	Interrupt enable for GENO_IRQ_24 IRQ.	RW W1toClr	0
23	GENO_IRQ_23_EN	Interrupt enable for GENO_IRQ_23 IRQ.	RW W1toClr	0
22	GENO_IRQ_22_EN	Interrupt enable for GENO_IRQ_22 IRQ.	RW W1toClr	0
21	GENO_IRQ_21_EN	Interrupt enable for GENO_IRQ_21 IRQ.	RW W1toClr	0
20	GENO_IRQ_20_EN	Interrupt enable for GENO_IRQ_20 IRQ.	RW W1toClr	0
19	GENO_IRQ_19_EN	Interrupt enable for GENO_IRQ_19 IRQ.	RW W1toClr	0
18	GENO_IRQ_18_EN	Interrupt enable for GENO_IRQ_18 IRQ.	RW W1toClr	0
17	GENO_IRQ_17_EN	Interrupt enable for GENO_IRQ_17 IRQ.	RW W1toClr	0
16	GENO_IRQ_16_EN	Interrupt enable for GENO_IRQ_16 IRQ.	RW W1toClr	0
15	GENO_IRQ_15_EN	Interrupt enable for GENO_IRQ_15 IRQ.	RW W1toClr	0
14	GENO_IRQ_14_EN	Interrupt enable for GENO_IRQ_14 IRQ.	RW W1toClr	0
13	GENO_IRQ_13_EN	Interrupt enable for GENO_IRQ_13 IRQ.	RW W1toClr	0
12	GENO_IRQ_12_EN	Interrupt enable for GENO_IRQ_12 IRQ.	RW W1toClr	0
11	GENO_IRQ_11_EN	Interrupt enable for GENO_IRQ_11 IRQ.	RW W1toClr	0
10	GENO_IRQ_10_EN	Interrupt enable for GENO_IRQ_10 IRQ.	RW W1toClr	0
9	GENO_IRQ_9_EN	Interrupt enable for GENO_IRQ_9 IRQ.	RW W1toClr	0
8	GENO_IRQ_8_EN	Interrupt enable for GENO_IRQ_8 IRQ.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
7	GENO_IRQ_7_EN	Interrupt enable for GENO_IRQ_7 IRQ.	RW W1toClr	0
6	GENO_IRQ_6_EN	Interrupt enable for GENO_IRQ_6 IRQ.	RW W1toClr	0
5	GENO_IRQ_5_EN	Interrupt enable for GENO_IRQ_5 IRQ.	RW W1toClr	0
4	GENO_IRQ_4_EN	Interrupt enable for GENO_IRQ_4 IRQ.	RW W1toClr	0
3	GENO_IRQ_3_EN	Interrupt enable for GENO_IRQ_3 IRQ.	RW W1toClr	0
2	GENO_IRQ_2_EN	Interrupt enable for GENO_IRQ_2 IRQ.	RW W1toClr	0
1	GENO_IRQ_1_EN	Interrupt enable for GENO_IRQ_1 IRQ.	RW W1toClr	0
0	GENO_IRQ_0_EN	Interrupt enable for GENO_IRQ_0 IRQ.	RW W1toClr	0

**Table 14-106. Register Call Summary for Register C2C\_IRQ\_ENABLE\_CLEAR\_1**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-107. C2C\_FCLK\_FREQ**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x000000
9:0	FCLK_FREQ	Value of the Functional clock frequency according to OPP written by software. Default value is 400 (MHz).	RW	0x190

**Table 14-108. Register Call Summary for Register C2C\_FCLK\_FREQ**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-109. C2C\_RX\_MAX\_FREQ**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x000000
9:0	RX_MAX_FREQ	Value of the maximum RX frequency according to OPP written by software. Default value is 400 (MHz).	RW	0x190

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x000000
9:0	RX_MAX_FREQ	Value of the requested RX maximum clock frequency according to OPP written by software. Default value is 50 (MHz).	RW	0x032

**Table 14-110. Register Call Summary for Register C2C\_RX\_MAX\_FREQ**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-111. C2C\_TX\_MAX\_FREQ**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TX_MAX_FREQ															

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x000000
9:0	TX_MAX_FREQ	Value of the requested TX maximum clock frequency set by distant C2C. Default value is 50 (MHz).	RW	0x032

**Table 14-112. Register Call Summary for Register C2C\_TX\_MAX\_FREQ**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-113. C2C\_RX\_MAX\_FREQ\_ACK**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_MAX_FREQ_ACK															

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x000000
9:0	RX_MAX_FREQ_ACK	Acknowledged maximum Rx clock frequency. Set by distant C2C. Default value is 50 (MHz).	RW	0x032

**Table 14-114. Register Call Summary for Register C2C\_RX\_MAX\_FREQ\_ACK**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-115. C2C\_WAKE\_REQ**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WAKE_REQ			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	WAKE_REQ	Value of the WAKE_REQ set by software	RW	0

**Table 14-116. Register Call Summary for Register C2C\_WAKE\_REQ**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-117. C2C\_WAKE\_ACK**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WAKE_ACK			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	WAKE_ACK	Value of the WAKE_ACK set by distant C2C	RW	0

**Table 14-118. Register Call Summary for Register C2C\_WAKE\_ACK**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-119. C2C\_STANDBY**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .	<b>Instance</b>	C2C
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STANDBY			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	STANDBY	Value of the STANDBY set by software	RW	1

**Table 14-120. Register Call Summary for Register C2C\_STANDBY**

- C2C Register Manual
- [C2C Register Summary: \[0\]](#)

**Table 14-121. C2C\_STANDBY\_IN**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .	<b>Instance</b>	C2C
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STANDBY			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	STANDBY	Value of the STANDBY set by distant C2C	RW	1

**Table 14-122. Register Call Summary for Register C2C\_STANDBY\_IN**

- C2C Register Manual
- [C2C Register Summary: \[0\]](#)

**Table 14-123. C2C\_WAIT**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .	<b>Instance</b>	C2C
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WAIT			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	WAIT	Value of the WAIT set by distant C2C	RW	1

**Table 14-124. Register Call Summary for Register C2C\_WAIT**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-125. C2C\_GENI\_CONTROL**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	C2C
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GENI_31	GENI_30	GENI_29	GENI_28	GENI_27	GENI_26	GENI_25	GENI_24	GENI_23	GENI_22	GENI_21	GENI_20	GENI_19	GENI_18	GENI_17	GENI_16	GENI_15	GENI_14	GENI_13	GENI_12	GENI_11	GENI_10	GENI_9	GENI_8	GENI_7	GENI_6	GENI_5	GENI_4	GENI_3	GENI_2	GENI_1	GENI_0

Bits	Field Name	Description	Type	Reset
31	GENI_31	If GENI_MASK_31 in GENI_MASK register is reset, GENI_31 is the value of C2C_Sscm_Geni_31 and cannot be modified by software. (signal is not masked) If GENI_MASK_31 is set (signal is masked), GENI_31 can be modified by software.	RW	0
30	GENI_30	f GENI_MASK_30 in GENI_MASK register is reset, GENI_30 is the value of C2C_Sscm_Geni_30 and cannot be modified by software. (signal is not masked) If GENI_MASK_30 is set (signal is masked), GENI_30 can be modified by software.	RW	0
29	GENI_29	f GENI_MASK_29 in GENI_MASK register is reset, GENI_29 is the value of C2C_Sscm_Geni_29 and cannot be modified by software. (signal is not masked) If GENI_MASK_29 is set (signal is masked), GENI_29 can be modified by software.	RW	0
28	GENI_28	f GENI_MASK_28 in GENI_MASK register is reset, GENI_28 is the value of C2C_Sscm_Geni_28 and cannot be modified by software. (signal is not masked) If GENI_MASK_28 is set (signal is masked), GENI_28 can be modified by software.	RW	0
27	GENI_27	f GENI_MASK_27 in GENI_MASK register is reset, GENI_27 is the value of C2C_Sscm_Geni_27 and cannot be modified by software. (signal is not masked) If GENI_MASK_27 is set (signal is masked), GENI_27 can be modified by software.	RW	0
26	GENI_26	f GENI_MASK_26 in GENI_MASK register is reset, GENI_26 is the value of C2C_Sscm_Geni_26 and cannot be modified by software. (signal is not masked) If GENI_MASK_26 is set (signal is masked), GENI_26 can be modified by software.	RW	0
25	GENI_25	f GENI_MASK_25 in GENI_MASK register is reset, GENI_25 is the value of C2C_Sscm_Geni_25 and cannot be modified by software. (signal is not masked) If GENI_MASK_25 is set (signal is masked), GENI_25 can be modified by software.	RW	0
24	GENI_24	f GENI_MASK_24 in GENI_MASK register is reset, GENI_24 is the value of C2C_Sscm_Geni_24 and cannot be modified by software. (signal is not masked) If GENI_MASK_24 is set (signal is masked), GENI_24 can be modified by software.	RW	0

Bits	Field Name	Description	Type	Reset
23	GENI_23	f GENI_MASK_23 in GENI_MASK register is reset, GENI_23 is the value of C2C_Sscm_Geni_23 and cannot be modified by software. (signal is not masked) If GENI_MASK_23 is set (signal is masked), GENI_23 can be modified by software.	RW	0
22	GENI_22	f GENI_MASK_22 in GENI_MASK register is reset, GENI_22 is the value of C2C_Sscm_Geni_22 and cannot be modified by software. (signal is not masked) If GENI_MASK_22 is set (signal is masked), GENI_22 can be modified by software.	RW	0
21	GENI_21	f GENI_MASK_21 in GENI_MASK register is reset, GENI_21 is the value of C2C_Sscm_Geni_21 and cannot be modified by software. (signal is not masked) If GENI_MASK_21 is set (signal is masked), GENI_21 can be modified by software.	RW	0
20	GENI_20	f GENI_MASK_20 in GENI_MASK register is reset, GENI_20 is the value of C2C_Sscm_Geni_20 and cannot be modified by software. (signal is not masked) If GENI_MASK_20 is set (signal is masked), GENI_20 can be modified by software.	RW	0
19	GENI_19	f GENI_MASK_19 in GENI_MASK register is reset, GENI_19 is the value of C2C_Sscm_Geni_19 and cannot be modified by software. (signal is not masked) If GENI_MASK_19 is set (signal is masked), GENI_19 can be modified by software.	RW	0
18	GENI_18	f GENI_MASK_18 in GENI_MASK register is reset, GENI_18 is the value of C2C_Sscm_Geni_18 and cannot be modified by software. (signal is not masked) If GENI_MASK_18 is set (signal is masked), GENI_18 can be modified by software.	RW	0
17	GENI_17	f GENI_MASK_17 in GENI_MASK register is reset, GENI_17 is the value of C2C_Sscm_Geni_17 and cannot be modified by software. (signal is not masked) If GENI_MASK_17 is set (signal is masked), GENI_17 can be modified by software.	RW	0
16	GENI_16	f GENI_MASK_16 in GENI_MASK register is reset, GENI_16 is the value of C2C_Sscm_Geni_16 and cannot be modified by software. (signal is not masked) If GENI_MASK_16 is set (signal is masked), GENI_16 can be modified by software.	RW	0
15	GENI_15	f GENI_MASK_15 in GENI_MASK register is reset, GENI_15 is the value of C2C_Sscm_Geni_15 and cannot be modified by software. (signal is not masked) If GENI_MASK_15 is set (signal is masked), GENI_15 can be modified by software.	RW	0
14	GENI_14	f GENI_MASK_14 in GENI_MASK register is reset, GENI_14 is the value of C2C_Sscm_Geni_14 and cannot be modified by software. (signal is not masked) If GENI_MASK_14 is set (signal is masked), GENI_14 can be modified by software.	RW	0
13	GENI_13	f GENI_MASK_13 in GENI_MASK register is reset, GENI_13 is the value of C2C_Sscm_Geni_13 and cannot be modified by software. (signal is not masked) If GENI_MASK_13 is set (signal is masked), GENI_13 can be modified by software.	RW	0
12	GENI_12	f GENI_MASK_12 in GENI_MASK register is reset, GENI_12 is the value of C2C_Sscm_Geni_12 and cannot be modified by software. (signal is not masked) If GENI_MASK_12 is set (signal is masked), GENI_12 can be modified by software.	RW	0
11	GENI_11	f GENI_MASK_11 in GENI_MASK register is reset, GENI_11 is the value of C2C_Sscm_Geni_11 and cannot be modified by software. (signal is not masked) If GENI_MASK_11 is set (signal is masked), GENI_11 can be modified by software.	RW	0

Bits	Field Name	Description	Type	Reset
10	GENI_10	f GENI_MASK_10 in GENI_MASK register is reset, GENI_10 is the value of C2C_Sscm_Geni_10 and cannot be modified by software. (signal is not masked) If GENI_MASK_10 is set (signal is masked), GENI_10 can be modified by software.	RW	0
9	GENI_9	f GENI_MASK_9 in GENI_MASK register is reset, GENI_9 is the value of C2C_Sscm_Geni_9 and cannot be modified by software. (signal is not masked) If GENI_MASK_9 is set (signal is masked), GENI_9 can be modified by software.	RW	0
8	GENI_8	f GENI_MASK_8 in GENI_MASK register is reset, GENI_8 is the value of C2C_Sscm_Geni_8 and cannot be modified by software. (signal is not masked) If GENI_MASK_8 is set (signal is masked), GENI_8 can be modified by software.	RW	0
7	GENI_7	f GENI_MASK_7 in GENI_MASK register is reset, GENI_7 is the value of C2C_Sscm_Geni_7 and cannot be modified by software. (signal is not masked) If GENI_MASK_7 is set (signal is masked), GENI_7 can be modified by software.	RW	0
6	GENI_6	f GENI_MASK_6 in GENI_MASK register is reset, GENI_6 is the value of C2C_Sscm_Geni_6 and cannot be modified by software. (signal is not masked) If GENI_MASK_6 is set (signal is masked), GENI_6 can be modified by software.	RW	0
5	GENI_5	f GENI_MASK_5 in GENI_MASK register is reset, GENI_5 is the value of C2C_Sscm_Geni_5 and cannot be modified by software. (signal is not masked) If GENI_MASK_5 is set (signal is masked), GENI_5 can be modified by software.	RW	0
4	GENI_4	f GENI_MASK_4 in GENI_MASK register is reset, GENI_4 is the value of C2C_Sscm_Geni_4 and cannot be modified by software. (signal is not masked) If GENI_MASK_4 is set (signal is masked), GENI_4 can be modified by software.	RW	0
3	GENI_3	f GENI_MASK_3 in GENI_MASK register is reset, GENI_3 is the value of C2C_Sscm_Geni_3 and cannot be modified by software. (signal is not masked) If GENI_MASK_3 is set (signal is masked), GENI_3 can be modified by software.	RW	0
2	GENI_2	f GENI_MASK_2 in GENI_MASK register is reset, GENI_2 is the value of C2C_Sscm_Geni_2 and cannot be modified by software. (signal is not masked) If GENI_MASK_2 is set (signal is masked), GENI_2 can be modified by software.	RW	0
1	GENI_1	f GENI_MASK_1 in GENI_MASK register is reset, GENI_1 is the value of C2C_Sscm_Geni_1 and cannot be modified by software. (signal is not masked) If GENI_MASK_1 is set (signal is masked), GENI_1 can be modified by software.	RW	0
0	GENI_0	f GENI_MASK_0 in GENI_MASK register is reset, GENI_0 is the value of C2C_Sscm_Geni_0 and cannot be modified by software. (signal is not masked) If GENI_MASK_0 is set (signal is masked), GENI_0 can be modified by software.	RW	0

**Table 14-126. Register Call Summary for Register C2C\_GENI\_CONTROL**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)



**Table 14-127. C2C\_GENI\_MASK**

Address Offset		0x0000 0074																													
Physical Address		See <a href="#">Table 14-80</a> .										Instance		C2C																	
Description																															
Type		RW																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GENI_MASK_31	GENI_MASK_30	GENI_MASK_29	GENI_MASK_28	GENI_MASK_27	GENI_MASK_26	GENI_MASK_25	GENI_MASK_24	GENI_MASK_23	GENI_MASK_22	GENI_MASK_21	GENI_MASK_20	GENI_MASK_19	GENI_MASK_18	GENI_MASK_17	GENI_MASK_16	GENI_MASK_15	GENI_MASK_14	GENI_MASK_13	GENI_MASK_12	GENI_MASK_11	GENI_MASK_10	GENI_MASK_9	GENI_MASK_8	GENI_MASK_7	GENI_MASK_6	GENI_MASK_5	GENI_MASK_4	GENI_MASK_3	GENI_MASK_2	GENI_MASK_1	GENI_MASK_0

Bits	Field Name	Description	Type	Reset
31	GENI_MASK_31	If GENI_MASK_31 is reset, C2C_Sscm_Geni_31 signal is not masked If GENI_MASK_31 is set, signal is masked.	RW	0
30	GENI_MASK_30	If GENI_MASK_30 is reset, C2C_Sscm_Geni_30 signal is not masked If GENI_MASK_30 is set, signal is masked.	RW	0
29	GENI_MASK_29	If GENI_MASK_29 is reset, C2C_Sscm_Geni_29 signal is not masked If GENI_MASK_29 is set, signal is masked.	RW	0
28	GENI_MASK_28	If GENI_MASK_28 is reset, C2C_Sscm_Geni_28 signal is not masked If GENI_MASK_28 is set, signal is masked.	RW	0
27	GENI_MASK_27	If GENI_MASK_27 is reset, C2C_Sscm_Geni_27 signal is not masked If GENI_MASK_27 is set, signal is masked.	RW	0
26	GENI_MASK_26	If GENI_MASK_26 is reset, C2C_Sscm_Geni_26 signal is not masked If GENI_MASK_26 is set, signal is masked.	RW	0
25	GENI_MASK_25	If GENI_MASK_25 is reset, C2C_Sscm_Geni_25 signal is not masked If GENI_MASK_25 is set, signal is masked.	RW	0
24	GENI_MASK_24	If GENI_MASK_24 is reset, C2C_Sscm_Geni_24 signal is not masked If GENI_MASK_24 is set, signal is masked.	RW	0
23	GENI_MASK_23	If GENI_MASK_23 is reset, C2C_Sscm_Geni_23 signal is not masked If GENI_MASK_23 is set, signal is masked.	RW	0
22	GENI_MASK_22	If GENI_MASK_22 is reset, C2C_Sscm_Geni_22 signal is not masked If GENI_MASK_22 is set, signal is masked.	RW	0
21	GENI_MASK_21	If GENI_MASK_21 is reset, C2C_Sscm_Geni_21 signal is not masked If GENI_MASK_21 is set, signal is masked.	RW	0
20	GENI_MASK_20	If GENI_MASK_20 is reset, C2C_Sscm_Geni_20 signal is not masked If GENI_MASK_20 is set, signal is masked.	RW	0
19	GENI_MASK_19	If GENI_MASK_19 is reset, C2C_Sscm_Geni_19 signal is not masked If GENI_MASK_19 is set, signal is masked.	RW	0
18	GENI_MASK_18	If GENI_MASK_18 is reset, C2C_Sscm_Geni_18 signal is not masked If GENI_MASK_18 is set, signal is masked.	RW	0
17	GENI_MASK_17	If GENI_MASK_17 is reset, C2C_Sscm_Geni_17 signal is not masked If GENI_MASK_17 is set, signal is masked.	RW	0

Bits	Field Name	Description	Type	Reset
16	GENI_MASK_16	If GENI_MASK_16 is reset, C2C_Sscm_Genl_16 signal is not masked If GENI_MASK_16 is set, signal is masked.	RW	0
15	GENI_MASK_15	If GENI_MASK_15 is reset, C2C_Sscm_Genl_15 signal is not masked If GENI_MASK_15 is set, signal is masked.	RW	0
14	GENI_MASK_14	If GENI_MASK_14 is reset, C2C_Sscm_Genl_14 signal is not masked If GENI_MASK_14 is set, signal is masked.	RW	0
13	GENI_MASK_13	If GENI_MASK_13 is reset, C2C_Sscm_Genl_13 signal is not masked If GENI_MASK_13 is set, signal is masked.	RW	0
12	GENI_MASK_12	If GENI_MASK_12 is reset, C2C_Sscm_Genl_12 signal is not masked If GENI_MASK_12 is set, signal is masked.	RW	0
11	GENI_MASK_11	If GENI_MASK_11 is reset, C2C_Sscm_Genl_11 signal is not masked If GENI_MASK_11 is set, signal is masked.	RW	0
10	GENI_MASK_10	If GENI_MASK_10 is reset, C2C_Sscm_Genl_10 signal is not masked If GENI_MASK_10 is set, signal is masked.	RW	0
9	GENI_MASK_9	If GENI_MASK_9 is reset, C2C_Sscm_Genl_9 signal is not masked If GENI_MASK_9 is set, signal is masked.	RW	0
8	GENI_MASK_8	If GENI_MASK_8 is reset, C2C_Sscm_Genl_8 signal is not masked If GENI_MASK_8 is set, signal is masked.	RW	0
7	GENI_MASK_7	If GENI_MASK_7 is reset, C2C_Sscm_Genl_7 signal is not masked If GENI_MASK_7 is set, signal is masked.	RW	0
6	GENI_MASK_6	If GENI_MASK_6 is reset, C2C_Sscm_Genl_6 signal is not masked If GENI_MASK_6 is set, signal is masked.	RW	0
5	GENI_MASK_5	If GENI_MASK_5 is reset, C2C_Sscm_Genl_5 signal is not masked If GENI_MASK_5 is set, signal is masked.	RW	0
4	GENI_MASK_4	If GENI_MASK_4 is reset, C2C_Sscm_Genl_4 signal is not masked If GENI_MASK_4 is set, signal is masked.	RW	0
3	GENI_MASK_3	If GENI_MASK_3 is reset, C2C_Sscm_Genl_3 signal is not masked If GENI_MASK_3 is set, signal is masked.	RW	0
2	GENI_MASK_2	If GENI_MASK_2 is reset, C2C_Sscm_Genl_2 signal is not masked If GENI_MASK_2 is set, signal is masked.	RW	0
1	GENI_MASK_1	If GENI_MASK_1 is reset, C2C_Sscm_Genl_1 signal is not masked If GENI_MASK_1 is set, signal is masked.	RW	0
0	GENI_MASK_0	If GENI_MASK_0 is reset, C2C_Sscm_Genl_0 signal is not masked If GENI_MASK_0 is set, signal is masked.	RW	0

**Table 14-128. Register Call Summary for Register C2C\_GENI\_MASK**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-129. C2C\_GENO\_STATUS**

<b>Address Offset</b>	0x0000 0080																																
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .																<b>Instance</b>	C2C															
<b>Description</b>																																	
<b>Type</b>	RW																																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	GENO_31	GENO_30	GENO_29	GENO_28	GENO_27	GENO_26	GENO_25	GENO_24	GENO_23	GENO_22	GENO_21	GENO_20	GENO_19	GENO_18	GENO_17	GENO_16	GENO_15	GENO_14	GENO_13	GENO_12	GENO_11	GENO_10	GENO_9	GENO_8	GENO_7	GENO_6	GENO_5	GENO_4	GENO_3	GENO_2	GENO_1	GENO_0	

Bits	Field Name	Description	Type	Reset
31	GENO_31	GENO_31 is the value of C2C_Sscm_GenO_31 signal.	RW	0
30	GENO_30	GENO_30 is the value of C2C_Sscm_GenO_30 signal.	RW	0
29	GENO_29	GENO_29 is the value of C2C_Sscm_GenO_29 signal.	RW	0
28	GENO_28	GENO_28 is the value of C2C_Sscm_GenO_28 signal.	RW	0
27	GENO_27	GENO_27 is the value of C2C_Sscm_GenO_27 signal.	RW	0
26	GENO_26	GENO_26 is the value of C2C_Sscm_GenO_26 signal.	RW	0
25	GENO_25	GENO_25 is the value of C2C_Sscm_GenO_25 signal.	RW	0
24	GENO_24	GENO_24 is the value of C2C_Sscm_GenO_24 signal.	RW	0
23	GENO_23	GENO_23 is the value of C2C_Sscm_GenO_23 signal.	RW	0
22	GENO_22	GENO_22 is the value of C2C_Sscm_GenO_22 signal.	RW	0
21	GENO_21	GENO_21 is the value of C2C_Sscm_GenO_21 signal.	RW	0
20	GENO_20	GENO_20 is the value of C2C_Sscm_GenO_20 signal.	RW	0
19	GENO_19	GENO_19 is the value of C2C_Sscm_GenO_19 signal.	RW	0
18	GENO_18	GENO_18 is the value of C2C_Sscm_GenO_18 signal.	RW	0
17	GENO_17	GENO_17 is the value of C2C_Sscm_GenO_17 signal.	RW	0
16	GENO_16	GENO_16 is the value of C2C_Sscm_GenO_16 signal.	RW	0
15	GENO_15	GENO_15 is the value of C2C_Sscm_GenO_15 signal.	RW	0
14	GENO_14	GENO_14 is the value of C2C_Sscm_GenO_14 signal.	RW	0
13	GENO_13	GENO_13 is the value of C2C_Sscm_GenO_13 signal.	RW	0
12	GENO_12	GENO_12 is the value of C2C_Sscm_GenO_12 signal.	RW	0
11	GENO_11	GENO_11 is the value of C2C_Sscm_GenO_11 signal.	RW	0
10	GENO_10	GENO_10 is the value of C2C_Sscm_GenO_10 signal.	RW	0
9	GENO_9	GENO_9 is the value of C2C_Sscm_GenO_9 signal.	RW	0
8	GENO_8	GENO_8 is the value of C2C_Sscm_GenO_8 signal.	RW	0
7	GENO_7	GENO_7 is the value of C2C_Sscm_GenO_7 signal.	RW	0
6	GENO_6	GENO_6 is the value of C2C_Sscm_GenO_6 signal.	RW	0
5	GENO_5	GENO_5 is the value of C2C_Sscm_GenO_5 signal.	RW	0
4	GENO_4	GENO_4 is the value of C2C_Sscm_GenO_4 signal.	RW	0
3	GENO_3	GENO_3 is the value of C2C_Sscm_GenO_3 signal.	RW	0
2	GENO_2	GENO_2 is the value of C2C_Sscm_GenO_2 signal.	RW	0
1	GENO_1	GENO_1 is the value of C2C_Sscm_GenO_1 signal.	RW	0
0	GENO_0	GENO_0 is the value of C2C_Sscm_GenO_0 signal.	RW	0

**Table 14-130. Register Call Summary for Register C2C\_GENO\_STATUS**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-131. C2C\_GENO\_INTERRUPT**

<b>Address Offset</b>	0x0000 0084																														
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .																<b>Instance</b>														
<b>Description</b>																	C2C														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GENO_INT_31	GENO_INT_30	GENO_INT_29	GENO_INT_28	GENO_INT_27	GENO_INT_26	GENO_INT_25	GENO_INT_24	GENO_INT_23	GENO_INT_22	GENO_INT_21	GENO_INT_20	GENO_INT_19	GENO_INT_18	GENO_INT_17	GENO_INT_16	GENO_INT_15	GENO_INT_14	GENO_INT_13	GENO_INT_12	GENO_INT_11	GENO_INT_10	GENO_INT_9	GENO_INT_8	GENO_INT_7	GENO_INT_6	GENO_INT_5	GENO_INT_4	GENO_INT_3	GENO_INT_2	GENO_INT_1	GENO_INT_0

Bits	Field Name	Description	Type	Reset
31	GENO_INT_31	If GENO_INT_31=0, GENO_IRQ_31=1 when GENO_31 toggles; If GENO_INT_31=1: IF GENO_LEVEL_31=0, GENO_IRQ_31 =1 when GENO_31 toggles to 0. IF GENO_LEVEL_31=1, GENO_IRQ_31 =1 when GENO_31 toggles to 1.	RW	0
30	GENO_INT_30	If GENO_INT_30=0, GENO_IRQ_30=1 when GENO_30 toggles; If GENO_INT_30=1: IF GENO_LEVEL_30=0, GENO_IRQ_30 =1 when GENO_30 toggles to 0. IF GENO_LEVEL_30=1, GENO_IRQ_30 =1 when GENO_30 toggles to 1.	RW	0
29	GENO_INT_29	If GENO_INT_29=0, GENO_IRQ_29=1 when GENO_29 toggles; If GENO_INT_29=1: IF GENO_LEVEL_29=0, GENO_IRQ_29 =1 when GENO_29 toggles to 0. IF GENO_LEVEL_29=1, GENO_IRQ_29 =1 when GENO_29 toggles to 1.	RW	0
28	GENO_INT_28	If GENO_INT_28=0, GENO_IRQ_28=1 when GENO_28 toggles; If GENO_INT_28=1: IF GENO_LEVEL_28=0, GENO_IRQ_28 =1 when GENO_28 toggles to 0. IF GENO_LEVEL_28=1, GENO_IRQ_28 =1 when GENO_28 toggles to 1.	RW	0
27	GENO_INT_27	If GENO_INT_27=0, GENO_IRQ_27=1 when GENO_27 toggles; If GENO_INT_27=1: IF GENO_LEVEL_27=0, GENO_IRQ_27 =1 when GENO_27 toggles to 0. IF GENO_LEVEL_27=1, GENO_IRQ_27 =1 when GENO_27 toggles to 1.	RW	0
26	GENO_INT_26	If GENO_INT_26=0, GENO_IRQ_26=1 when GENO_26 toggles; If GENO_INT_26=1: IF GENO_LEVEL_26=0, GENO_IRQ_26 =1 when GENO_26 toggles to 0. IF GENO_LEVEL_26=1, GENO_IRQ_26 =1 when GENO_26 toggles to 1.	RW	0
25	GENO_INT_25	If GENO_INT_25=0, GENO_IRQ_25=1 when GENO_25 toggles; If GENO_INT_25=1: IF GENO_LEVEL_25=0, GENO_IRQ_25 =1 when GENO_25 toggles to 0. IF GENO_LEVEL_25=1, GENO_IRQ_25 =1 when GENO_25 toggles to 1.	RW	0
24	GENO_INT_24	If GENO_INT_24=0, GENO_IRQ_24=1 when GENO_24 toggles; If GENO_INT_24=1: IF GENO_LEVEL_24=0, GENO_IRQ_24 =1 when GENO_24 toggles to 0. IF GENO_LEVEL_24=1, GENO_IRQ_24 =1 when GENO_24 toggles to 1.	RW	0
23	GENO_INT_23	If GENO_INT_23=0, GENO_IRQ_23=1 when GENO_23 toggles; If GENO_INT_23=1: IF GENO_LEVEL_23=0, GENO_IRQ_23 =1 when GENO_23 toggles to 0. IF GENO_LEVEL_23=1, GENO_IRQ_23 =1 when GENO_23 toggles to 1.	RW	0
22	GENO_INT_22	If GENO_INT_22=0, GENO_IRQ_22=1 when GENO_22 toggles; If GENO_INT_22=1: IF GENO_LEVEL_22=0, GENO_IRQ_22 =1 when GENO_22 toggles to 0. IF GENO_LEVEL_22=1, GENO_IRQ_22 =1 when GENO_22 toggles to 1.	RW	0
21	GENO_INT_21	If GENO_INT_21=0, GENO_IRQ_21=1 when GENO_21 toggles; If GENO_INT_21=1: IF GENO_LEVEL_21=0, GENO_IRQ_21 =1 when GENO_21 toggles to 0. IF GENO_LEVEL_21=1, GENO_IRQ_21 =1 when GENO_21 toggles to 1.	RW	0
20	GENO_INT_20	If GENO_INT_20=0, GENO_IRQ_20=1 when GENO_20 toggles; If GENO_INT_20=1: IF GENO_LEVEL_20=0, GENO_IRQ_20 =1 when GENO_20 toggles to 0. IF GENO_LEVEL_20=1, GENO_IRQ_20 =1 when GENO_20 toggles to 1.	RW	0
19	GENO_INT_19	If GENO_INT_19=0, GENO_IRQ_19=1 when GENO_19 toggles; If GENO_INT_19=1: IF GENO_LEVEL_19=0, GENO_IRQ_19 =1 when GENO_19 toggles to 0. IF GENO_LEVEL_19=1, GENO_IRQ_19 =1 when GENO_19 toggles to 1.	RW	0

Bits	Field Name	Description	Type	Reset
18	GENO_INT_18	If GENO_INT_18=0, GENO_IRQ_18=1 when GENO_18 toggles; If GENO_INT_18=1: IF GENO_LEVEL_18=0, GENO_IRQ_18 =1 when GENO_18 toggles to 0. IF GENO_LEVEL_18=1, GENO_IRQ_18 =1 when GENO_18 toggles to 1.	RW	0
17	GENO_INT_17	If GENO_INT_17=0, GENO_IRQ_17=1 when GENO_17 toggles; If GENO_INT_17=1: IF GENO_LEVEL_17=0, GENO_IRQ_17 =1 when GENO_17 toggles to 0. IF GENO_LEVEL_17=1, GENO_IRQ_17 =1 when GENO_17 toggles to 1.	RW	0
16	GENO_INT_16	If GENO_INT_16=0, GENO_IRQ_16=1 when GENO_16 toggles; If GENO_INT_16=1: IF GENO_LEVEL_16=0, GENO_IRQ_16 =1 when GENO_16 toggles to 0. IF GENO_LEVEL_16=1, GENO_IRQ_16 =1 when GENO_16 toggles to 1.	RW	0
15	GENO_INT_15	If GENO_INT_15=0, GENO_IRQ_15=1 when GENO_15 toggles; If GENO_INT_15=1: IF GENO_LEVEL_15=0, GENO_IRQ_15 =1 when GENO_15 toggles to 0. IF GENO_LEVEL_15=1, GENO_IRQ_15 =1 when GENO_15 toggles to 1.	RW	0
14	GENO_INT_14	If GENO_INT_14=0, GENO_IRQ_14=1 when GENO_14 toggles; If GENO_INT_14=1: IF GENO_LEVEL_14=0, GENO_IRQ_14 =1 when GENO_14 toggles to 0. IF GENO_LEVEL_14=1, GENO_IRQ_14 =1 when GENO_14 toggles to 1.	RW	0
13	GENO_INT_13	If GENO_INT_13=0, GENO_IRQ_13=1 when GENO_13 toggles; If GENO_INT_13=1: IF GENO_LEVEL_13=0, GENO_IRQ_13 =1 when GENO_13 toggles to 0. IF GENO_LEVEL_13=1, GENO_IRQ_13 =1 when GENO_13 toggles to 1.	RW	0
12	GENO_INT_12	If GENO_INT_12=0, GENO_IRQ_12=1 when GENO_12 toggles; If GENO_INT_12=1: IF GENO_LEVEL_12=0, GENO_IRQ_12 =1 when GENO_12 toggles to 0. IF GENO_LEVEL_12=1, GENO_IRQ_12 =1 when GENO_12 toggles to 1.	RW	0
11	GENO_INT_11	If GENO_INT_11=0, GENO_IRQ_11=1 when GENO_11 toggles; If GENO_INT_11=1: IF GENO_LEVEL_11=0, GENO_IRQ_11 =1 when GENO_11 toggles to 0. IF GENO_LEVEL_11=1, GENO_IRQ_11 =1 when GENO_11 toggles to 1.	RW	0
10	GENO_INT_10	If GENO_INT_10=0, GENO_IRQ_10=1 when GENO_10 toggles; If GENO_INT_10=1: IF GENO_LEVEL_10=0, GENO_IRQ_10 =1 when GENO_10 toggles to 0. IF GENO_LEVEL_10=1, GENO_IRQ_10 =1 when GENO_10 toggles to 1.	RW	0
9	GENO_INT_9	If GENO_INT_9=0, GENO_IRQ_9=1 when GENO_9 toggles; If GENO_INT_9=1: IF GENO_LEVEL_9=0, GENO_IRQ_9 =1 when GENO_9 toggles to 0. IF GENO_LEVEL_9=1, GENO_IRQ_9 =1 when GENO_9 toggles to 1.	RW	0
8	GENO_INT_8	If GENO_INT_8=0, GENO_IRQ_8=1 when GENO_8 toggles; If GENO_INT_8=1: IF GENO_LEVEL_8=0, GENO_IRQ_8 =1 when GENO_8 toggles to 0. IF GENO_LEVEL_8=1, GENO_IRQ_8 =1 when GENO_8 toggles to 1.	RW	0
7	GENO_INT_7	If GENO_INT_7=0, GENO_IRQ_7=1 when GENO_7 toggles; If GENO_INT_7=1: IF GENO_LEVEL_7=0, GENO_IRQ_7 =1 when GENO_7 toggles to 0. IF GENO_LEVEL_7=1, GENO_IRQ_7 =1 when GENO_7 toggles to 1.	RW	0
6	GENO_INT_6	If GENO_INT_6=0, GENO_IRQ_6=1 when GENO_6 toggles; If GENO_INT_6=1: IF GENO_LEVEL_6=0, GENO_IRQ_6 =1 when GENO_6 toggles to 0. IF GENO_LEVEL_6=1, GENO_IRQ_6 =1 when GENO_6 toggles to 1.	RW	0

Bits	Field Name	Description	Type	Reset
5	GENO_INT_5	If GENO_INT_5=0, GENO_IRQ_5=1 when GENO_5 toggles; If GENO_INT_5=1: IF GENO_LEVEL_5=0, GENO_IRQ_5 =1 when GENO_5 toggles to 0. IF GENO_LEVEL_5=1, GENO_IRQ_5 =1 when GENO_5 toggles to 1.	RW	0
4	GENO_INT_4	If GENO_INT_4=0, GENO_IRQ_4=1 when GENO_4 toggles; If GENO_INT_4=1: IF GENO_LEVEL_4=0, GENO_IRQ_4 =1 when GENO_4 toggles to 0. IF GENO_LEVEL_4=1, GENO_IRQ_4 =1 when GENO_4 toggles to 1.	RW	0
3	GENO_INT_3	If GENO_INT_3=0, GENO_IRQ_3=1 when GENO_3 toggles; If GENO_INT_3=1: IF GENO_LEVEL_3=0, GENO_IRQ_3 =1 when GENO_3 toggles to 0. IF GENO_LEVEL_3=1, GENO_IRQ_3 =1 when GENO_3 toggles to 1.	RW	0
2	GENO_INT_2	If GENO_INT_2=0, GENO_IRQ_2=1 when GENO_2 toggles; If GENO_INT_2=1: IF GENO_LEVEL_2=0, GENO_IRQ_2 =1 when GENO_2 toggles to 0. IF GENO_LEVEL_2=1, GENO_IRQ_2 =1 when GENO_2 toggles to 1.	RW	0
1	GENO_INT_1	If GENO_INT_1=0, GENO_IRQ_1=1 when GENO_1 toggles; If GENO_INT_1=1: IF GENO_LEVEL_1=0, GENO_IRQ_1 =1 when GENO_1 toggles to 0. IF GENO_LEVEL_1=1, GENO_IRQ_1 =1 when GENO_1 toggles to 1.	RW	0
0	GENO_INT_0	If GENO_INT_0=0, GENO_IRQ_0=1 when GENO_0 toggles; If GENO_INT_0=1: IF GENO_LEVEL_0=0, GENO_IRQ_0 =1 when GENO_0 toggles to 0. IF GENO_LEVEL_0=1, GENO_IRQ_0 =1 when GENO_0 toggles to 1.	RW	0

**Table 14-132. Register Call Summary for Register C2C\_GENO\_INTERRUPT**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)

**Table 14-133. C2C\_GENO\_LEVEL**

<b>Address Offset</b>	0x0000 0088																																																
<b>Physical Address</b>	See <a href="#">Table 14-80</a> .																<b>Instance</b>																C2C																
<b>Description</b>																																																	
<b>Type</b>	RW																																																
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
	GENO_LEVEL_31	GENO_LEVEL_30	GENO_LEVEL_29	GENO_LEVEL_28	GENO_LEVEL_27	GENO_LEVEL_26	GENO_LEVEL_25	GENO_LEVEL_24	GENO_LEVEL_23	GENO_LEVEL_22	GENO_LEVEL_21	GENO_LEVEL_20	GENO_LEVEL_19	GENO_LEVEL_18	GENO_LEVEL_17	GENO_LEVEL_16	GENO_LEVEL_15	GENO_LEVEL_14	GENO_LEVEL_13	GENO_LEVEL_12	GENO_LEVEL_11	GENO_LEVEL_10	GENO_LEVEL_9	GENO_LEVEL_8	GENO_LEVEL_7	GENO_LEVEL_6	GENO_LEVEL_5	GENO_LEVEL_4	GENO_LEVEL_3	GENO_LEVEL_2	GENO_LEVEL_1	GENO_LEVEL_0																	
<b>Bits</b>																																																	
<b>Field Name</b>																																																	
<b>Description</b>																																																	
<b>Type</b>																																																	
<b>Reset</b>																																																	
31	GENO_LEVEL_31																IF GENO_LEVEL_31=0, GENO_IRQ_31 =1 when GENO_31 toggles to 0. IF GENO_LEVEL_31=1, GENO_IRQ_31 =1 when GENO_31 toggles to 1.																RW	0															
30	GENO_LEVEL_30																IF GENO_LEVEL_30=0, GENO_IRQ_30 =1 when GENO_30 toggles to 0. IF GENO_LEVEL_30=1, GENO_IRQ_30 =1 when GENO_30 toggles to 1.																RW	0															
29	GENO_LEVEL_29																IF GENO_LEVEL_29=0, GENO_IRQ_29 =1 when GENO_29 toggles to 0. IF GENO_LEVEL_29=1, GENO_IRQ_29 =1 when GENO_29 toggles to 1.																RW	0															

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
28	GENO_LEVEL_28	IF GENO_LEVEL_28=0, GENO_IRQ_28 =1 when GENO_28 toggles to 0. IF GENO_LEVEL_28=1, GENO_IRQ_28 =1 when GENO_28 toggles to 1.	RW	0
27	GENO_LEVEL_27	IF GENO_LEVEL_27=0, GENO_IRQ_27 =1 when GENO_27 toggles to 0. IF GENO_LEVEL_27=1, GENO_IRQ_27 =1 when GENO_27 toggles to 1.	RW	0
26	GENO_LEVEL_26	IF GENO_LEVEL_26=0, GENO_IRQ_26 =1 when GENO_26 toggles to 0. IF GENO_LEVEL_26=1, GENO_IRQ_26 =1 when GENO_26 toggles to 1.	RW	0
25	GENO_LEVEL_25	IF GENO_LEVEL_25=0, GENO_IRQ_25 =1 when GENO_25 toggles to 0. IF GENO_LEVEL_25=1, GENO_IRQ_25 =1 when GENO_25 toggles to 1.	RW	0
24	GENO_LEVEL_24	IF GENO_LEVEL_24=0, GENO_IRQ_24 =1 when GENO_24 toggles to 0. IF GENO_LEVEL_24=1, GENO_IRQ_24 =1 when GENO_24 toggles to 1.	RW	0
23	GENO_LEVEL_23	IF GENO_LEVEL_23=0, GENO_IRQ_23 =1 when GENO_23 toggles to 0. IF GENO_LEVEL_23=1, GENO_IRQ_23 =1 when GENO_23 toggles to 1.	RW	0
22	GENO_LEVEL_22	IF GENO_LEVEL_22=0, GENO_IRQ_22 =1 when GENO_22 toggles to 0. IF GENO_LEVEL_22=1, GENO_IRQ_22 =1 when GENO_22 toggles to 1.	RW	0
21	GENO_LEVEL_21	IF GENO_LEVEL_21=0, GENO_IRQ_21 =1 when GENO_21 toggles to 0. IF GENO_LEVEL_21=1, GENO_IRQ_21 =1 when GENO_21 toggles to 1.	RW	0
20	GENO_LEVEL_20	IF GENO_LEVEL_20=0, GENO_IRQ_20 =1 when GENO_20 toggles to 0. IF GENO_LEVEL_20=1, GENO_IRQ_20 =1 when GENO_20 toggles to 1.	RW	0
19	GENO_LEVEL_19	IF GENO_LEVEL_19=0, GENO_IRQ_19 =1 when GENO_19 toggles to 0. IF GENO_LEVEL_19=1, GENO_IRQ_19 =1 when GENO_19 toggles to 1.	RW	0
18	GENO_LEVEL_18	IF GENO_LEVEL_18=0, GENO_IRQ_18 =1 when GENO_18 toggles to 0. IF GENO_LEVEL_18=1, GENO_IRQ_18 =1 when GENO_18 toggles to 1.	RW	0
17	GENO_LEVEL_17	IF GENO_LEVEL_17=0, GENO_IRQ_17 =1 when GENO_17 toggles to 0. IF GENO_LEVEL_17=1, GENO_IRQ_17 =1 when GENO_17 toggles to 1.	RW	0
16	GENO_LEVEL_16	IF GENO_LEVEL_16=0, GENO_IRQ_16 =1 when GENO_16 toggles to 0. IF GENO_LEVEL_16=1, GENO_IRQ_16 =1 when GENO_16 toggles to 1.	RW	0
15	GENO_LEVEL_15	IF GENO_LEVEL_15=0, GENO_IRQ_15 =1 when GENO_15 toggles to 0. IF GENO_LEVEL_15=1, GENO_IRQ_15 =1 when GENO_15 toggles to 1.	RW	0
14	GENO_LEVEL_14	IF GENO_LEVEL_14=0, GENO_IRQ_14 =1 when GENO_14 toggles to 0. IF GENO_LEVEL_14=1, GENO_IRQ_14 =1 when GENO_14 toggles to 1.	RW	0
13	GENO_LEVEL_13	IF GENO_LEVEL_13=0, GENO_IRQ_13 =1 when GENO_13 toggles to 0. IF GENO_LEVEL_13=1, GENO_IRQ_13 =1 when GENO_13 toggles to 1.	RW	0
12	GENO_LEVEL_12	IF GENO_LEVEL_12=0, GENO_IRQ_12 =1 when GENO_12 toggles to 0. IF GENO_LEVEL_12=1, GENO_IRQ_12 =1 when GENO_12 toggles to 1.	RW	0
11	GENO_LEVEL_11	IF GENO_LEVEL_11=0, GENO_IRQ_11 =1 when GENO_11 toggles to 0. IF GENO_LEVEL_11=1, GENO_IRQ_11 =1 when GENO_11 toggles to 1.	RW	0
10	GENO_LEVEL_10	IF GENO_LEVEL_10=0, GENO_IRQ_10 =1 when GENO_10 toggles to 0. IF GENO_LEVEL_10=1, GENO_IRQ_10 =1 when GENO_10 toggles to 1.	RW	0
9	GENO_LEVEL_9	IF GENO_LEVEL_9=0, GENO_IRQ_9 =1 when GENO_9 toggles to 0. IF GENO_LEVEL_9=1, GENO_IRQ_9 =1 when GENO_9 toggles to 1.	RW	0

Bits	Field Name	Description	Type	Reset
8	GENO_LEVEL_8	IF GENO_LEVEL_8=0, GENO_IRQ_8 =1 when GENO_8 toggles to 0. IF GENO_LEVEL_8=1, GENO_IRQ_8 =1 when GENO_8 toggles to 1.	RW	0
7	GENO_LEVEL_7	IF GENO_LEVEL_7=0, GENO_IRQ_7 =1 when GENO_7 toggles to 0. IF GENO_LEVEL_7=1, GENO_IRQ_7 =1 when GENO_7 toggles to 1.	RW	0
6	GENO_LEVEL_6	IF GENO_LEVEL_6=0, GENO_IRQ_6 =1 when GENO_6 toggles to 0. IF GENO_LEVEL_6=1, GENO_IRQ_6 =1 when GENO_6 toggles to 1.	RW	0
5	GENO_LEVEL_5	IF GENO_LEVEL_5=0, GENO_IRQ_5 =1 when GENO_5 toggles to 0. IF GENO_LEVEL_5=1, GENO_IRQ_5 =1 when GENO_5 toggles to 1.	RW	0
4	GENO_LEVEL_4	IF GENO_LEVEL_4=0, GENO_IRQ_4 =1 when GENO_4 toggles to 0. IF GENO_LEVEL_4=1, GENO_IRQ_4 =1 when GENO_4 toggles to 1.	RW	0
3	GENO_LEVEL_3	IF GENO_LEVEL_3=0, GENO_IRQ_3 =1 when GENO_3 toggles to 0. IF GENO_LEVEL_3=1, GENO_IRQ_3 =1 when GENO_3 toggles to 1.	RW	0
2	GENO_LEVEL_2	IF GENO_LEVEL_2=0, GENO_IRQ_2 =1 when GENO_2 toggles to 0. IF GENO_LEVEL_2=1, GENO_IRQ_2 =1 when GENO_2 toggles to 1.	RW	0
1	GENO_LEVEL_1	IF GENO_LEVEL_1=0, GENO_IRQ_1 =1 when GENO_1 toggles to 0. IF GENO_LEVEL_1=1, GENO_IRQ_1 =1 when GENO_1 toggles to 1.	RW	0
0	GENO_LEVEL_0	IF GENO_LEVEL_0=0, GENO_IRQ_0 =1 when GENO_0 toggles to 0. IF GENO_LEVEL_0=1, GENO_IRQ_0 =1 when GENO_0 toggles to 1.	RW	0

**Table 14-134. Register Call Summary for Register C2C\_GENO\_LEVEL**

C2C Register Manual

- [C2C Register Summary: \[0\]](#)



## Memory Subsystem

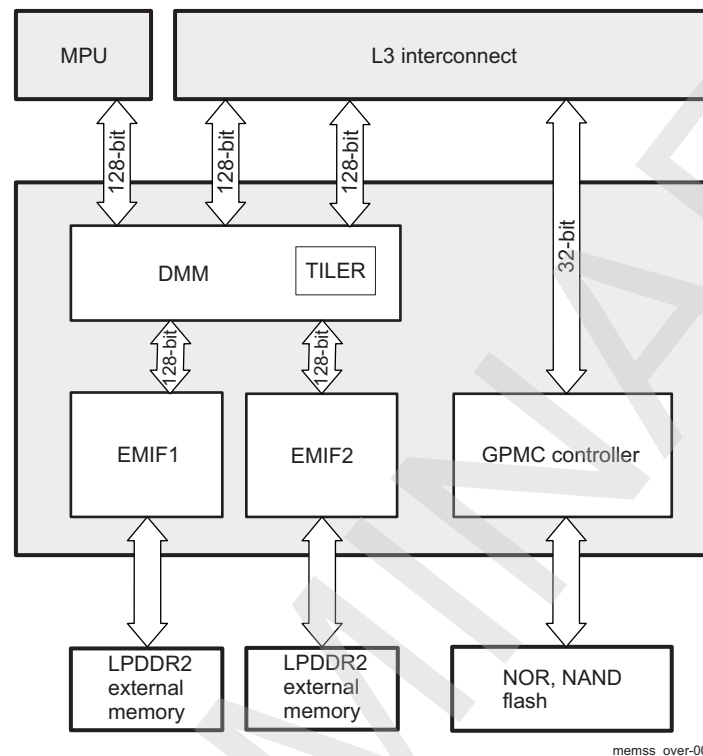
This chapter introduces the memory subsystem of the device.

Topic	Page
15.1 Memory Subsystem Overview .....	2890
15.2 Dynamic Memory Manager .....	2895
15.3 EMIF Controller .....	2977
15.4 General-Purpose Memory Controller .....	3044
15.5 Error Location Module .....	3187
15.6 On-Chip Memory (OCM) Subsystem .....	3216

## 15.1 Memory Subsystem Overview

Figure 15-1 shows a functional diagram of all memory subsystems in the device.

**Figure 15-1. Memory Subsystem Functional Diagram**



### 15.1.1 DMM Overview

The dynamic memory manager (DMM) module is typically located immediately in front of the synchronous dynamic random access memory (SDRAM) memory controller, as shown in Figure 15-1.

In a broad sense, the DMM manages various aspects of memory accesses such as:

- Initiator-indexed priority generation
- Multizone SDRAM memory interleaving configuration
- Block object transfer optimization – tiling
- Centralized low-latency page translation – MMU-like feature

The dynamic qualifier for memory management highlights the software configurability, and hence the runtime nature, of the four aspects of memory management handled by the DMM.

#### 15.1.1.1 DMM Features

From a functional perspective, the role of the DMM is to:

- Add initiator-based priority to any incoming requests
- Perform to/from tiling conversions of tiled requests
- Make on-the-fly basic transforms, such as quadrant rotations and mirroring
- Optionally, provide a low-latency page-based translation to handle memory fragmentation – MMU
- Distribute the traffic on all attached memory controllers according to the interleaving configuration

The DMM main features are:

- Programmable multizone DRAM memory mapping and interleaving configuration
- Programmable initiator-based request priority extension
- Multichannel memory transfer optimization

- Single SDRAM page request generation
- Low-latency interconnect port
- Page-grained address translation to manage memory fragmentation
- Automatic synchronized reloading of the address translation table

### 15.1.2 TILER Overview

The tiling and isometric lightweight engine for rotation (TILER) is a submodule of DMM (see [Figure 15-1](#)) and is therefore described in [Section 15.2, Dynamic Memory Manager](#).

The TILER is intended to improve bidimensional (tiled) block transfer efficiency. It is, therefore, a module aimed at:

- Primarily, efficient handling of 2-dimension (2D) data mapped in tiles, such as video or graphics macroblocks
- Optionally, managing the memory fragmentation and zero-copy physical frame-buffer swapping through a page-grained translation
- Making isometric (distance preserving) transforms, such as 90-, 180-, or 270-degree rotations, with a horizontal or vertical reflection

The lightweight qualifier of this engine highlights its limited support of isometric transforms, as:

- Rotation performs only basic quadrant rotations by some multiple of 90 degrees.
- Reflection is limited to horizontal and vertical flip.
- Translation is restricted to a 4-KB page granularity.

Written differently, the functionality of the TILER is to map a 2D virtually addressed interconnect request into one or more physically addressed interconnect requests by:

- Transforming the virtual address, data, and byte-enable to match the requested 0-, 90-, 180-, or 270-degree orientation in a tiled 2D addressing space
- Optionally translating the oriented tiled address by a page-specific vector to manage memory fragmentation and physical object aliasing

#### 15.1.2.1 TILER Features

The main features of the TILER are:

- Efficiency improvement of 2D block access on SDRAMs
- Optimized interlaced access on tiled frames
- 2D virtual-to-physical address translation of SDRAM bidimensional objects to handle rotation
- Page-grained address translation to manage memory fragmentation and physical buffer aliasing
- Unlimited number of 2D tiled objects supported in any ( 0, 90, 180, or 270 degrees) orientation
- Full bandwidth use by minimizing the size of raster-based initiator buffers
- Optimization of multichannel memory transfers
- Interconnect request granularity balance (in X and Y directions)

### 15.1.3 EMIF Overview

The extended memory interface (EMIF) module is typically located near the DMM module, as shown in [Figure 15-1](#).

The EMIF provides connectivity between the device and LPDDR2-type memory. It manages data bus read/write accesses between external memories, the microprocessor unit (MPU), and the direct memory access (DMA) controller. It acts as the primary interface between SDRAM and all functional blocks, such as the ARM®, IVA-HD, imaging subsystem (ISS), GFX, and DMA controllers.

#### 15.1.3.1 Main Features

The EMIF has the following capabilities:

- Supports JEDEC standard-compliant LPDDR2-SDRAM (S2 and S4) and LPDDR2-NVM devices

- 2-GB SDRAM address range over two chip selects (CSs) (configurable with the DMM; see [Section 15.2, Dynamic Memory Manager](#), for more information)
- Supports two independent CSs, with their corresponding register sets, and independent page tracking
- Both CSs must have the same memory type and size if they are both SDRAM or both nonvolatile memory (NVM). LPDDR2-SDRAM can be used in parallel with LPDDR2-NVM and can have a different size.
- Flexible address muxing scheme which permits choosing different bank-mapping allocation by configuring the bank, column, and row-address decoding ordering
- 16- or 32-bit data path to external SDRAM
- Supports LPDDR2 devices with 1, 2, 4, or 8 internal banks
- Supports the following data bus widths:
  - 128-bit level 3 (L3) interconnect data bus width
  - 16- and 32-bit SDRAM data bus width
- Supports the following CAS latencies: 3, 4, 5, 6, 7, and 8
- Supports 256-, 512-, 1024-, and 2048-word page sizes
- Supports the following burst lengths: 8
- Supports sequential burst type
- SDRAM auto-initialization from reset or configuration change
- Supports bank interleaving across both CSs if same memory type
- Supports self-refresh and precharge power-down modes for low power (per CS, manual entry/exit self-refresh, and deep power-down modes)
- Supports deep power-down mode for low power
- Supports partial array self-refresh and temperature-controlled self-refresh modes for low power. Temperature-controlled self-refresh is supported only for mobile SDRAM having an on-chip temperature sensor.
- Supports prioritized refresh
- Programmable SDRAM refresh rate and backlog counter
- Supports ZQ calibration
- Supports temperature monitoring

#### 15.1.4 GPMC Overview

The general-purpose memory controller (GPMC) is a unified memory controller dedicated to interfacing external memory devices:

- Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices
- Asynchronous, synchronous, and page mode (available only in nonmuxed mode) burst NOR flash devices
- NAND flash
- Pseudo-SRAM devices

##### 15.1.4.1 GPMC Features

The GPMC is the external memory controller of the device. The GPMC data access engine provides a flexible programming model for communication with all standard memories. The GPMC supports various accesses:

- Asynchronous read/write access
- Asynchronous read page access (4, 8, and 16 Word16)
- Synchronous read/write access
- Synchronous read/write burst access without wrap capability (4, 8, and 16 Word16)
- Synchronous read/write burst access with wrap capability (4, 8, and 16 Word16)
- Address/data-multiplexed access
- Little- and big-endian access

The GPMC can communicate with a wide range of external devices:

- External asynchronous or synchronous 8-bit wide memory or device (nonburst device)
- External asynchronous or synchronous 16-bit wide memory or device
- External 16-bit nonmultiplexed device with limited address range (2KB)
- External 16-bit address/data-multiplexed NOR flash device
- External 8-bit and 16-bit NAND flash device
- External 16-bit pseudo-SRAM (pSRAM) device

The main features of the GPMC are:

- 8- or 16-bit-wide data path to external memory device
- Supports up to eight CS regions of programmable size and programmable base addresses in a total address space of 1GB
- Supports transactions controlled by a firewall
- On-the-fly error code detection using the Bose-Chaudhuri-Hocquenghem (BCH) ( $t = 4, 8, \text{ or } 16$ ) or Hamming code to improve the reliability of NAND with a minimum effect on software (NAND flash with 512-byte page size or greater)
- Fully pipelined operation for optimal memory bandwidth use
- External device clock provided from the L3 clock divided by 1, 2, 3, or 4
- Supports programmable autoclock gating when no access is detected
- Independent and programmable control signal timing parameters for setup and hold time on a per-chip basis. Parameters are set according to the memory device timing parameters, with a timing granularity of one L3 clock cycle.
- Flexible internal access time control (WAIT state) and flexible handshake mode using external WAIT pin monitoring
- Support bus keeping
- Support bus turnaround
- Prefetch and write posting engine associated with system DMA (sDMA) to achieve full performance from the AND device with minimum effect on NOR/SRAM concurrent access

---

**NOTE:** Page mode is available only in nonmultiplexed mode.

---

### 15.1.5 ELM Overview

NAND flash memories are dense and nonvolatile, but error-prone. Whenever NAND flash memories are read from, some level of error correction is required. In the case of NAND modules with no internal correction capability, sometimes referred to as bare NAND, the correction process is delegated to the memory controller.

The error location module (ELM) must be used in conjunction with the GPMC. The GPMC probes data read from an external NAND flash and uses them to compute checksum-like information, called syndrome polynomials, on a per-block basis. Each syndrome polynomial gives a status of the read operations for a full block, including 512 bytes of data, parity bits, and an optional spare area data field, with a maximum block size of 1023 bytes. Computation is based on the BCH algorithm. The ELM extracts error addresses from these syndrome polynomials.

Based on the syndrome polynomial value, the ELM can detect errors, compute the number of errors, and give the location of each error bit. The actual data is not required to complete the error correction algorithm. Errors can be reported anywhere in the NAND flash block, including in the parity bits.

The maximum acceptable number of errors that can be corrected depends on a configuration parameter. Error correction levels of 4, 8, and 16 bits are supported. The ELM relies on a static definition of the generator polynomial for each error correction level that corresponds to the generator polynomials defined in the GPMC. A larger number of errors can be detected, but the ELM cannot correct them all. The offending block is then tagged as uncorrectable.

When the error location process completes, an interrupt is triggered to inform a central processing unit (CPU) that its status can be checked. The error locations for the NAND page can be retrieved from the module through register accesses.

#### 15.1.5.1 ELM Features

The ELM features:

- 4, 8, and 16 bits per 512-byte block error location based on BCH algorithm
- Eight simultaneous processing contexts
- Page-based and continuous modes
- Interrupt generation when error location process completes:
  - When the full page has been processed in page mode
  - For each syndrome polynomial in continuous mode

#### 15.1.6 OCM Overview

The on-chip memory (OCM) subsystem consists of the following OCM controllers: one connected to an on-chip ROM (SAR ROM), one connected to an on-chip RAM (SAR RAM), and one connected to an on-chip SRAM (L3 SRAM). Each memory controller has its own dedicated interface to the L3 interconnect.

##### 15.1.6.1 SAR ROM

This on-chip ROM contains 4KB of memory and a linked list of descriptors used by sDMA during the restore context operation (when the device transitions from off to on mode).

The device-embedded SAR ROM has the following characteristics:

- 4-KB ROM
- 32-bit access per cycle
- Support for single- and burst-access transactions

##### 15.1.6.2 SAR RAM

The on-chip SAR RAM contains 8K bytes and is mapped as 4 blocks with irregular region sizes. This memory content is preserved when the device goes into off mode (as long as the wake-up voltage domain remains supplied). It is used as context-saving memory to be written by software so that sDMA restores its saved content when the device transitions from off to on mode.

The device-embedded SAR RAM has the following characteristics:

- Support for single-access transactions
  - Operates at full L4-PER interconnect clock frequency
  - 32-bit access per cycle

##### 15.1.6.3 L3 OCM\_RAM

The on-chip L3 OCM\_RAM contains 56KB of RAM, and partitioning is defined by the L3 firewall logic. The device-embedded L3 OCM\_RAM has the following characteristics:

- Support for single and burst access transactions:
  - Operates at full L3 interconnect clock frequency
  - Fully pipelined, one 32-bit access per cycle
- Restricted access support

## 15.2 Dynamic Memory Manager

### 15.2.1 DMM Overview

This section describes the dynamic memory manager (DMM) and its Tiling and Isometric Lightweight Engine for Rotation (TILER) submodule.

The DMM is introduced in *Memory Overview* in [Chapter 15, Memory Subsystem](#).

On a purely functional perspective, the role of the DMM is to:

- add initiator-based priority to any incoming requests
- perform to-and-fro tiling conversions of "tiled" requests
- making on-the-fly basic itransforms such as quadrant rotations and mirroring
- optionally provide a low-latency page-based translation to handle memory fragmentation - MMU
- distribute the traffic on all attached memory controllers according to the interleaving configuration

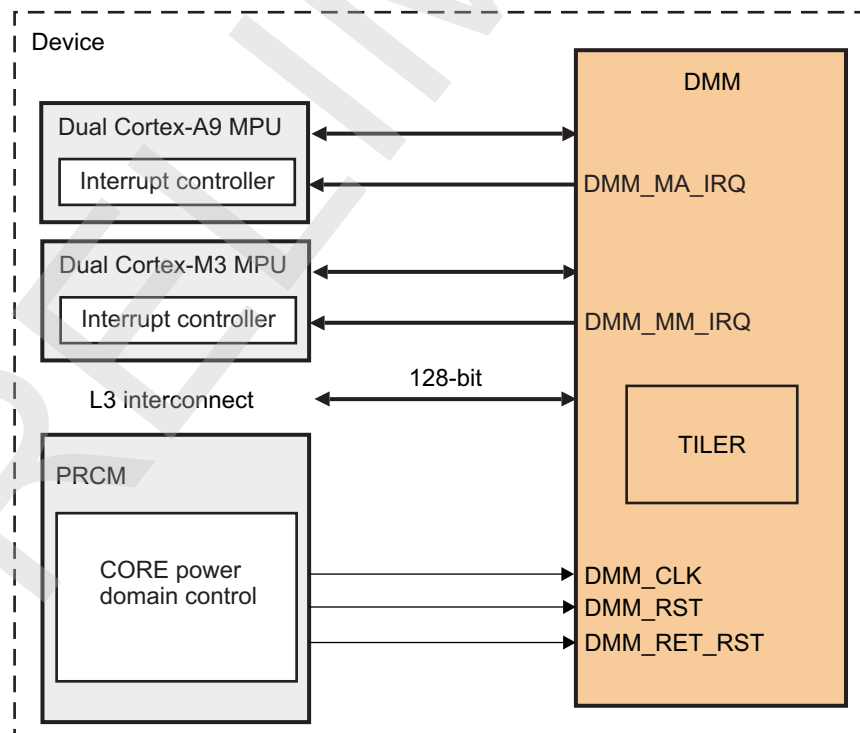
The TILER is also introduced in *Memory Overview* in [Chapter 15, Memory Subsystem](#).

It is a module aimed at:

- primary handling efficiently 2 dimensional data mapped in tiles, such as video or graphics macroblocks
- optionally managing the memory fragmentation and zero-copy physical frame buffers swapping through a page-grained translation
- allows optimised interlaced accesses on tiled frames
- making - distance preserving - transforms, such as 90°, 180°, or 270° rotations, with either a horizontal or vertical reflection
- interleaving the memory accesses among the two memory controllers (also named EMIF, see [Section 15.3, EMIF Controller](#), for more information on it)

Figure 15-2 shows DMM and Tiler overview in the device.

Figure 15-2. DMM and Tiler Overview



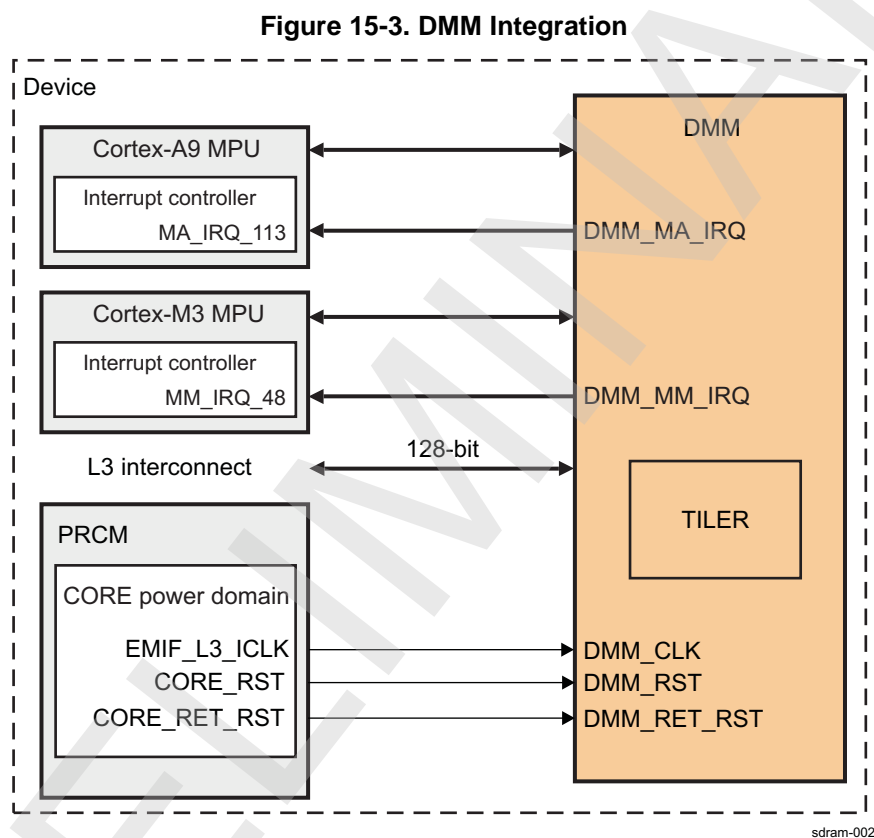
sdram-066

## 15.2.2 DMM Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

- IDLE hardware handshake (the DMM only supports the smart-idle mode)
- No STANDBY hardware handshake
- No wake-up request
- No system direct memory access (sDMA) requests
- One interrupt line for interrupt request (IRQ)
- One functional clock

Figure 15-3 shows DMM integration.



**NOTE:** For more information about the IDLE hardware handshake, see [Chapter 3, Power, Reset, and Clock Management](#).

Table 15-1 and Table 15-3 summarize the DMM integration in the device.

**Table 15-1. DMM Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-up Capability	Interconnect
DMM	PD_CORE	No	N/A

**Table 15-2. DMM Clocks and Resets**

Module Instance	Destination Signal Name	Source Signal Name	Clocks	
			Source	Description



**Table 15-2. DMM Clocks and Resets (continued)**

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DMM	DMM_CLK	EMIF_L3_ICLK	PRCM	DMM interface and functional clock. For information about PRCM clock gating and management, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
<b>Resets</b>				
DMM	DMM_RST	CORE_RST	PRCM	Functional reset. For information about PRCM reset sources and distribution, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
DMM	DMM_RET_RST	CORE_RET_RST	PRCM	Reset for retention flip-flop registers, namely: <a href="#">DMM_SYSCONFIG</a> <a href="#">DMM_LISA_LOCK</a> <a href="#">DMM_LISA_MAP_i</a> <a href="#">DMM_TILER_OR0</a> <a href="#">DMM_TILER_OR1</a> <a href="#">DMM_PAT_VIEW0</a> <a href="#">DMM_PAT_VIEW1</a> <a href="#">DMM_PAT_VIEW_MAP_i</a> <a href="#">DMM_PAT_VIEW_MAP_BASE</a> <a href="#">DMM_PAT_DESCR_i</a> <a href="#">DMM_PAT_AREA_i</a> <a href="#">DMM_PAT_CTRL_i</a> <a href="#">DMM_PAT_DATA_i</a> <a href="#">DMM_PEG_PRIO_k</a> <a href="#">DMM_PEG_PRIO_PAT</a> For information about PRCM reset sources and distribution, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .

**Table 15-3. Hardware Requests**

<b>Interrupt Requests</b>				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
DMM	DMM_MA_IRQ	MA_IRQ_113	Dual Cortex-A9 MPU	DMM interrupt to the Dual Cortex-A9 MPU subsystem
DMM	DMM_MM_IRQ	MM_IRQ_48	Dual Cortex-M3 MPU	DMM interrupt to the Dual Cortex-M3 MPU subsystem
<b>No DMA Requests</b>				

**15.2.2.1 DMM Configuration**

[Table 15-4](#) lists all configured parameters for the DMM. Note that these parameters are read only.

**Table 15-4. DMM TILER Container Geometry**

Scope	Bit field	Configuration Value	Description
LISA	<a href="#">DMM_HWINFO</a> [3:0] TILER_CNT	0x2	2 TILER instance in the DMM
	<a href="#">DMM_HWINFO</a> [11:8] ELLA_CNT	0x1	1 ELLA instance in the DMM
	<a href="#">DMM_HWINFO</a> [19:16] ROBIN_CNT	0x2	2 ROBIN instance in the DMM
	<a href="#">DMM_LISA_HWINFO</a> [4:0] SECTION_CNT	0x2	2 DMM sections
	<a href="#">DMM_LISA_HWINFO</a> [11:8] SDRC_CNT	0x2	2 SDRC controller attached
TILER	<a href="#">DMM_TILER_HWINFO</a> [6:0] OR_CNT	0x10	16 orientation entries
PAT	<a href="#">DMM_PAT_GEOMETRY</a> [4:0] PAGE_SZ	0x0C	4-kiB page granularity
	<a href="#">DMM_PAT_GEOMETRY</a> [19:16] CONT_WDTH	0x8	Container width of 256 pages
	<a href="#">DMM_PAT_GEOMETRY</a> [26:24] CONT_HGHT	0x7	Container height of 128 pages
	<a href="#">DMM_PAT_GEOMETRY</a> [13:8] ADDR_RANGE	0x10	2-GiB PAT output physical address range
	<a href="#">DMM_PAT_HWINFO</a> [6:0] VIEW_CNT	0x10	16 view entries

**Table 15-4. DMM TILER Container Geometry (continued)**

Scope	Bit field	Configurat ion Value	Description
	DMM_PAT_HWINFO[11:8] VIEW_MAP_CNT	0x4	4 view maps
	DMM_PAT_HWINFO[20:16] LUT_CNT	0x1	1 PAT LUT
	DMM_PAT_HWINFO[28:24] ENGINE_CNT	0x2	2 PAT refill engines
PEG	DMM_PEG_HWINFO[6:0] PRIO_CNT	0x10	16 priority entries

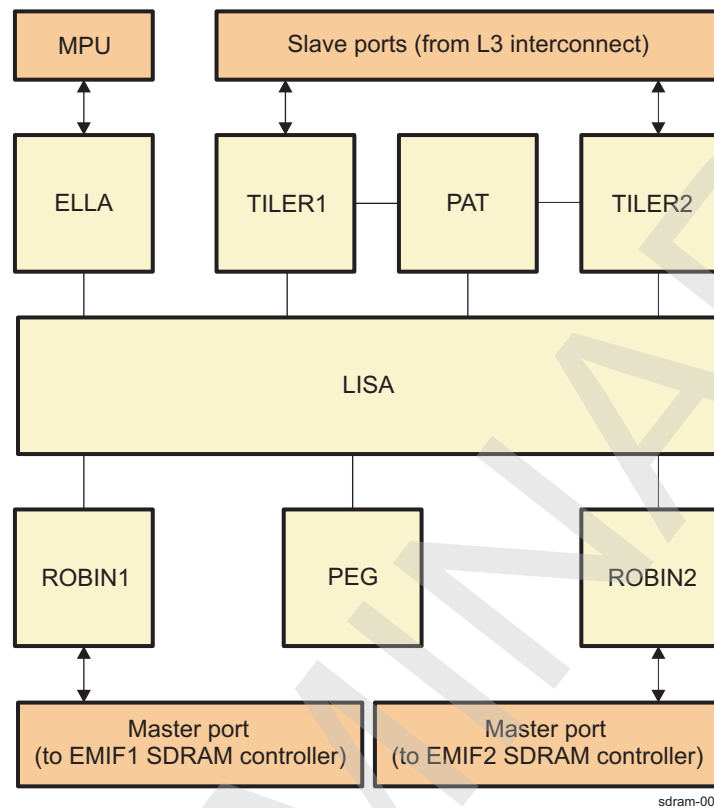
### 15.2.3 DMM Functional Description

#### 15.2.3.1 DMM Block Diagram

Figure 15-4 shows the DMM macro architecture. The DMM consists of six blocks:

- Two Tiling and Isometric Lightweight Engine for Rotation (TILER), having each their own interconnect slave port, for converting requests to-and-fro between the input virtual addressing mode and the output physical tiled addressing mode. Note that, the tiling conversions of requests, write data and responses is fully performed in the TILER blocks.
- Two Re-Ordering Buffer and Initiator Nodes (ROBIN), having each their own interconnect master port, to initiate requests to the SDRAM controller and allow tiled data, tiled response and split response reconstruction. The ROBIN block is only managing the re-ordering buffer and performing the data re-ordering due to the orientation.
- One Extra Low Latency Access (ELLA), having its own interconnect slave port, for providing lower latency accesses to the memory
- A Physical Address Translator (PAT) for managing the memory fragmentation
- A Priority Extension Generator (PEG) to generate priorities required by the SDRAM controller, note that these priorities are not used in the DMM
- A Local Interconnect and Synchronisation Agent (LISA) to synchronise all DMM subsystems and provide access to their configuration registers

Figure 15-4. DMM Block Diagram



sdram-003

**CAUTION**

It is the interconnect duty to ensure that virtually-addressed requests only target a TILER port and that only physically-addressed requests target an ELLA port.

**15.2.3.2 DMM Clock Configuration**

Table 15-5 describes the DMM clocks.

Table 15-5. DMM Clocks

Signal	I/O <sup>(1)</sup>	Description
DMM_CLK	I	Functional and interface clock

<sup>(1)</sup> I = Input; O = Output

The DMM is about SDRAM controller management and naturally sits in the MEMIF clock domain among the SDRAM controllers. The DMM is a fully synchronous module, which uses the clock and clock enable signals provided in the MEMIF clock domain for generating its interface and functional clocks.

To configure DMM\_CLK control and settings, see Table 15-2.

**15.2.3.3 DMM Power Management**

DMM power is supplied by the CORE power domain, and DMM power management complies with system power-management guidelines.

Table 15-6 describes power-management features available for the DMM module.

**Table 15-6. DMM Local Power Management Features**

Feature	Registers	Description
Clock Auto Gating	N/A	Feature not available
Slave Idle Modes	<a href="#">DMM_SYSCONFIG</a> [3:2] SIDLEMODE bit field	Only Smart-idle wakeup mode is available
Clock Activity	N/A	Feature not available
Master Standby Modes	N/A	Feature not available
Global Wake-up Enable	N/A	Feature not available
Wake-up Sources Enable	N/A	Feature not available

Some of the DMM module's registers are impacted by the Save And Restore (SAR) mechanism. For more information about SAR, see [Chapter 3, Power, Reset, and Clock Management](#).

### 15.2.3.4 DMM Interrupt Requests

Errors in PAT area refill registers are reported through the [DMM\\_PAT\\_STATUS\\_i](#)[15:10] ERROR bit field, as shown in [Table 15-7](#).

**Table 15-7. DMM Hardware Status Features**

Feature	Type	Register	Description
PAT error flags	Read only	<a href="#">DMM_PAT_STATUS_i</a> [15:10] ERROR (i = 0 to 3)	Unexpected update of the PAT area refill registers

[Table 15-8](#) lists the event flags, and their masks, that can cause module interrupts.

**Table 15-8. Events**

Interrupt	Event Flag	Event Mask
ERR_LUT_MISS3	<a href="#">DMM_PAT_IRQSTATUS</a> [31] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [31] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [31] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [31] EVENT0
ERR_UPD_DATA3	<a href="#">DMM_PAT_IRQSTATUS</a> [30] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [30] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [30] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [30] EVENT0
ERR_UPD_CTRL3	<a href="#">DMM_PAT_IRQSTATUS</a> [29] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [29] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [29] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [29] EVENT0
ERR_UPD_AREA3	<a href="#">DMM_PAT_IRQSTATUS</a> [28] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [28] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [28] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [28] EVENT0
ERR_INV_DATA3	<a href="#">DMM_PAT_IRQSTATUS</a> [27] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [27] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [27] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [27] EVENT0
ERR_INV_DSC3	<a href="#">DMM_PAT_IRQSTATUS</a> [26] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [26] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [26] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [26] EVENT0
FILL_LST3	<a href="#">DMM_PAT_IRQSTATUS</a> [25] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [25] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [25] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [25] EVENT0
FILL_DSC3	<a href="#">DMM_PAT_IRQSTATUS</a> [24] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [24] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [24] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [24] EVENT0
ERR_LUT_MISS2	<a href="#">DMM_PAT_IRQSTATUS</a> [23] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [23] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [23] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [23] EVENT0
ERR_UPD_DATA2	<a href="#">DMM_PAT_IRQSTATUS</a> [22] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [22] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [22] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [22] EVENT0
ERR_UPD_CTRL2	<a href="#">DMM_PAT_IRQSTATUS</a> [21] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [21] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [21] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [21] EVENT0
ERR_UPD_AREA2	<a href="#">DMM_PAT_IRQSTATUS</a> [20] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [20] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [20] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [20] EVENT0
ERR_INV_DATA2	<a href="#">DMM_PAT_IRQSTATUS</a> [19] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [19] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [19] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [19] EVENT0
ERR_INV_DSC2	<a href="#">DMM_PAT_IRQSTATUS</a> [18] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [18] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [18] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [18] EVENT0
FILL_LST2	<a href="#">DMM_PAT_IRQSTATUS</a> [17] EVENT0 <a href="#">DMM_PAT_IRQSTATUS_RAW</a> [17] EVENT0	<a href="#">DMM_PAT_IRQENABLE_SET</a> [17] EVENT0 <a href="#">DMM_PAT_IRQENABLE_CLR</a> [17] EVENT0

**Table 15-8. Events (continued)**

<b>Interrupt</b>	<b>Event Flag</b>	<b>Event Mask</b>
FILL_DSC2	DMM_PAT_IRQSTATUS[16] EVENT0 DMM_PAT_IRQSTATUS_RAW[16] EVENT0	DMM_PAT_IRQENABLE_SET[16] EVENT0 DMM_PAT_IRQENABLE_CLR[16] EVENT0
ERR_LUT_MISS1	DMM_PAT_IRQSTATUS[15] EVENT0 DMM_PAT_IRQSTATUS_RAW[15] EVENT0	DMM_PAT_IRQENABLE_SET[15] EVENT0 DMM_PAT_IRQENABLE_CLR[15] EVENT0
ERR_UPD_DATA1	DMM_PAT_IRQSTATUS[14] EVENT0 DMM_PAT_IRQSTATUS_RAW[14] EVENT0	DMM_PAT_IRQENABLE_SET[14] EVENT0 DMM_PAT_IRQENABLE_CLR[14] EVENT0
ERR_UPD_CTRL1	DMM_PAT_IRQSTATUS[13] EVENT0 DMM_PAT_IRQSTATUS_RAW[13] EVENT0	DMM_PAT_IRQENABLE_SET[13] EVENT0 DMM_PAT_IRQENABLE_CLR[13] EVENT0
ERR_UPD_AREA1	DMM_PAT_IRQSTATUS[12] EVENT0 DMM_PAT_IRQSTATUS_RAW[12] EVENT0	DMM_PAT_IRQENABLE_SET[12] EVENT0 DMM_PAT_IRQENABLE_CLR[12] EVENT0
ERR_INV_DATA1	DMM_PAT_IRQSTATUS[11] EVENT0 DMM_PAT_IRQSTATUS_RAW[11] EVENT0	DMM_PAT_IRQENABLE_SET[11] EVENT0 DMM_PAT_IRQENABLE_CLR[11] EVENT0
ERR_INV_DSC1	DMM_PAT_IRQSTATUS[10] EVENT0 DMM_PAT_IRQSTATUS_RAW[10] EVENT0	DMM_PAT_IRQENABLE_SET[10] EVENT0 DMM_PAT_IRQENABLE_CLR[10] EVENT0
FILL_LST1	DMM_PAT_IRQSTATUS[9] EVENT0 DMM_PAT_IRQSTATUS_RAW[9] EVENT0	DMM_PAT_IRQENABLE_SET[9] EVENT0 DMM_PAT_IRQENABLE_CLR[9] EVENT0
FILL_DSC1	DMM_PAT_IRQSTATUS[8] EVENT0 DMM_PAT_IRQSTATUS_RAW[8] EVENT0	DMM_PAT_IRQENABLE_SET[8] EVENT0 DMM_PAT_IRQENABLE_CLR[8] EVENT0
ERR_LUT_MISS0	DMM_PAT_IRQSTATUS[7] EVENT0 DMM_PAT_IRQSTATUS_RAW[7] EVENT0	DMM_PAT_IRQENABLE_SET[7] EVENT0 DMM_PAT_IRQENABLE_CLR[7] EVENT0
ERR_UPD_DATA0	DMM_PAT_IRQSTATUS[6] EVENT0 DMM_PAT_IRQSTATUS_RAW[6] EVENT0	DMM_PAT_IRQENABLE_SET[6] EVENT0 DMM_PAT_IRQENABLE_CLR[6] EVENT0
ERR_UPD_CTRL0	DMM_PAT_IRQSTATUS[5] EVENT0 DMM_PAT_IRQSTATUS_RAW[5] EVENT0	DMM_PAT_IRQENABLE_SET[5] EVENT0 DMM_PAT_IRQENABLE_CLR[5] EVENT0
ERR_UPD_AREA0	DMM_PAT_IRQSTATUS[4] EVENT0 DMM_PAT_IRQSTATUS_RAW[4] EVENT0	DMM_PAT_IRQENABLE_SET[4] EVENT0 DMM_PAT_IRQENABLE_CLR[4] EVENT0
ERR_INV_DATA0	DMM_PAT_IRQSTATUS[3] EVENT0 DMM_PAT_IRQSTATUS_RAW[3] EVENT0	DMM_PAT_IRQENABLE_SET[3] EVENT0 DMM_PAT_IRQENABLE_CLR[3] EVENT0
ERR_INV_DSC0	DMM_PAT_IRQSTATUS[2] EVENT0 DMM_PAT_IRQSTATUS_RAW[2] EVENT0	DMM_PAT_IRQENABLE_SET[2] EVENT0 DMM_PAT_IRQENABLE_CLR[2] EVENT0
FILL_LST0	DMM_PAT_IRQSTATUS[1] EVENT0 DMM_PAT_IRQSTATUS_RAW[1] EVENT0	DMM_PAT_IRQENABLE_SET[1] EVENT0 DMM_PAT_IRQENABLE_CLR[1] EVENT0
FILL_DSC0	DMM_PAT_IRQSTATUS[0] EVENT0 DMM_PAT_IRQSTATUS_RAW[0] EVENT0	DMM_PAT_IRQENABLE_SET[0] EVENT0 DMM_PAT_IRQENABLE_CLR[0] EVENT0

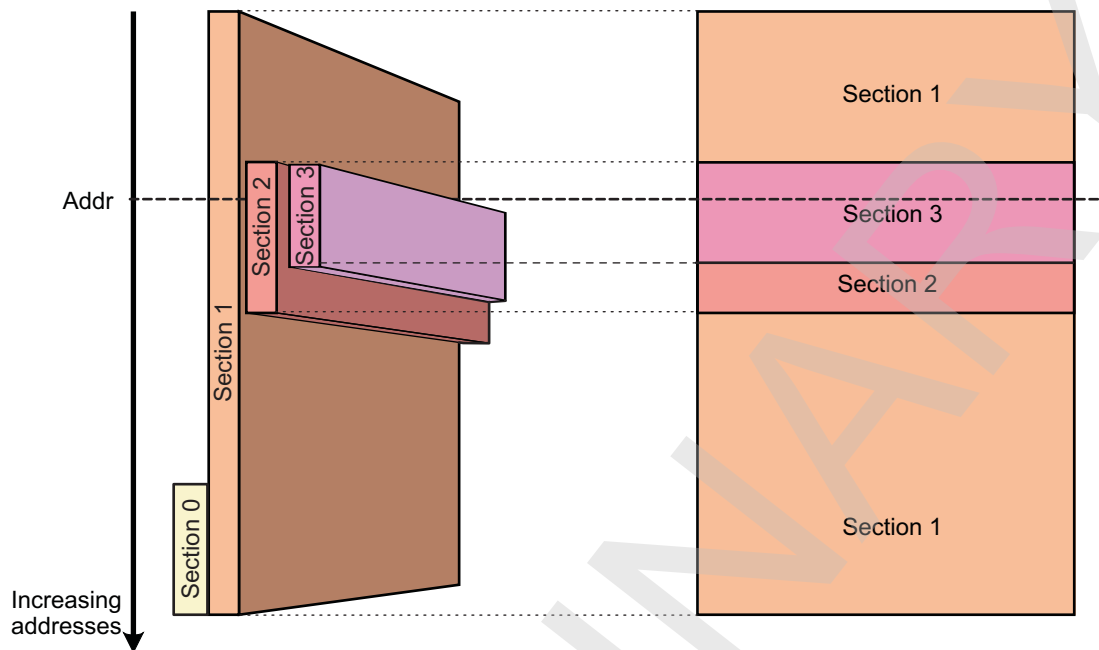
### 15.2.3.5 DMM

#### 15.2.3.5.1 DMM Concepts

The DMM introduces the concepts of dynamic mapping and DMM atomic units.

##### 15.2.3.5.1.1 Dynamic Mapping

The DMM manages its internal memory space as an ordered set of up to four sections. [Figure 15-5](#) shows DMM sections and memory mapping.

**Figure 15-5. DMM Sections and Memory Mapping**

sdram\_004

In the DMM, a section is:

- A segment of 16 MiB to 2 GiB, which is power-of-two in size and aligned to that size in the system map
- An area with a constant interleaving scheme - constant interleaving granularity on a constant set of SDRAM controller (EMIF) targets
- Given a priority equal to its index - the higher the index, the higher the priority.

**NOTE:** Whenever a request hits more than one DMM section, it follows the interleaving scheme of the section having the highest index.

The interleaving configuration of the DMM, its section-based dynamic memory mapping, is shown in the right hand side of [Figure 15-5](#). In this example, a request at the system address Addr will follow the interleaving scheme of section 3, although hitting sections 1, 2 and 3. Similarly, the DMM configuration given in this example prevents any request to use the interleaving scheme of section 0, since section 0 is fully masked by section 1 - which has a higher priority.

Each of the four sections is configured through a [DMM\\_LISA\\_MAP\\_i](#) register, where  $i = 0$  to 3.

**NOTE:** The DMM and SDRAM controller registers ([Table 15-88](#) and EMIF registers, see [Section 15.3, EMIF Controller](#)) are actually declared in two extra static DMM sections of the highest priority so that they cannot be masked by any standard programmable DMM section.

**NOTE:** The DMM atomic size is:

- 1 kiB in non interleaved sections
- equal to the interleaving granularity (128 bytes, 256 bytes or 512 bytes) in interleaved sections

### 15.2.3.5.1.2 Address Mapping

The address mapping inside the DMM is configurable through up to four sections. Each section is defined based on:

- Its system address: the base address of the decoding range for the section.
- Its size: the encoding is the number of bits actually used in the upper 8-bits of the incoming system address.
- Its physical address: the base address of the memory range access in the external memory controller.
- Its address space: the address space used on the external memory controller when hitting this section.
- The target memory controller. The section is not used if this bit field is set to 2'b00.
- Its interleaving definition.

The address decoding is priority-based. In case of overlapping sections, only the highest-order one is hit. Register memory spaces have priority over regular memory sections. In case of four sections, the priority order is therefore:

1. Registers
2. DMM\_LISA\_MAP\_3
3. DMM\_LISA\_MAP\_2
4. DMM\_LISA\_MAP\_1
5. DMM\_LISA\_MAP\_0

All register-related addresses are reserved and fixed in the overall address mapping:

- DMM registers: 0x4E00\_0000 to 0x4FFF\_FFFF
- EMIF1 registers: 0x4C00\_0000 to 0x4CFF\_FFFF
- EMIF2 registers: 0x4D00\_0000 to 0x4DFF\_FFFF

There is no overlapping between register sections.

---

**NOTE:** Section decoding happens after PAT address translation in the case of TILER. In non-bypass mode, the system address considered for TILER accesses is actually the virtual address computed based on the PAT translation tables.

---

The [DMM\\_LISA\\_LOCK](#) register is used to lock the configuration once set. If written to one, the LOCK bit prevents further writes to all [DMM\\_LISA\\_MAP\\_i](#) registers. The LOCK itself cannot be written back to 0. A reset is required to reprogram the sections.

### **15.2.3.5.1.3 Address Translation**

The physical address translation engine (PAT) of the DMM is composed of a 32-k entry physical address translation vector table and of one or two refill engines. The refill engine is a specialised DMA aimed at refilling the content of the physical address translation table.

The address translation mechanism is only available when the incoming request is hitting a page mode or tiled mode container, that is, when the incoming address is targeting the TILER or its aliased view in the system addressing space. Otherwise the physical address translation logic is bypassed so that the resulting physical address corresponds to the input address.

The PAT is supporting multiple address translation schemes, called views, which can be bound to one or more initiator through a view mapping mechanism.

#### **15.2.3.5.1.3.1 PAT View Mappings**

The PAT can have up to 16 groups of initiators sharing a set of 4 PAT views. The connection from an initiator to a PAT view is made through the [DMM\\_PAT\\_VIEW](#) register. Given that each PAT view index is coded on four bits, the [DMM\\_PAT\\_VIEW](#) register is a 64-bit register split in two 32-bit registers named [DMM\\_PAT\\_VIEW0](#) for the first eight PAT view indexes and [DMM\\_PAT\\_VIEW1](#) for the last eight PAT view indexes.

The PAT view index corresponding to the initiator having the value *i* as the four MSB of its L3 ConnID, will be using the view referenced in the entry *i* of the [DMM\\_PAT\\_VIEW](#). For instance the initiator 0xC4 will be using the 13th view index of the [DMM\\_PAT\\_VIEW](#) register, the 5th in the [DMM\\_PAT\\_VIEW1](#) register.

The PAT view index of the initiator  $i$  is found in the  $V_i$  field. The  $W_i$  field is aimed at writing the corresponding  $V_i$ . When writing to the `DMM_PAT_VIEW` registers, the only  $V_i$  view indexes that are updated are those having their corresponding  $W_i$  bit - and byte enable - set. The  $W_i$  bits are always read as 0. For instance, to set the PAT view indexes  $V_3$  and  $V_7$  to respectively 2 and 1, the `DMM_PAT_VIEW0` register shall be written with 5000 6000h.

### 15.2.3.5.1.3.2 PAT View Map Base Address

The PAT view map base address is defining the base address of all PAT translated addresses.

The bit 31 of all PAT translated address is set to `BASE_ADDR`. For instance if the `DMM_PAT_VIEW_MAP_BASE` register is set to 8000 0000h then all PAT translated addresses will have the bit 31 set to 1 so that the translated addresses were ranging from 8000 0000h to FFFF FFFFh. All reserved bits of this DMM PAT base address register are read to 0.

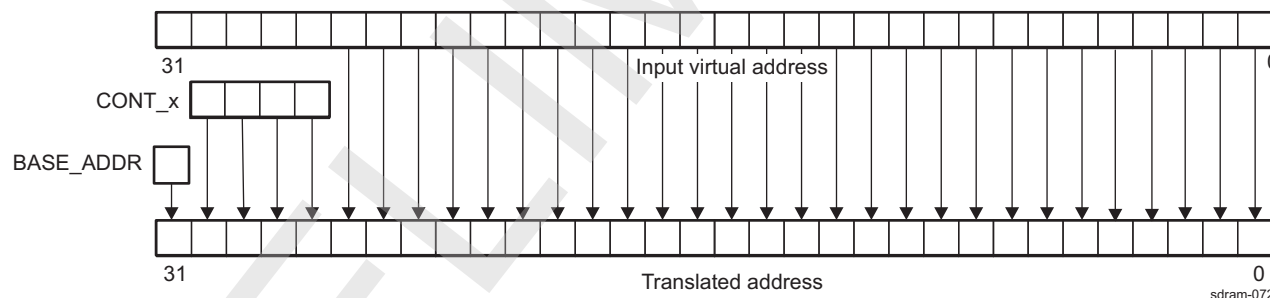
### 15.2.3.5.1.3.3 PAT Views

A PAT view is defining the kind of physical address translation to perform for each mode accesses (page, 8-bit, 16-bit and 32-bit). Each mode of each PAT view can be configured to either use a container-grained translation or a page-mode translation.

#### 15.2.3.5.1.3.3.1 PAT Direct Access Translation

The container-grained translation is named the direct access. In this mode, the translation vector is directly given by the `CONT_x` field corresponding to the accessed mode. A page mode access is using the vector contained in the `DMM_PAT_VIEW_MAP_i[27:24]` `CONT_PAGE` field, a 32-bit mode access is using the vector contained in the `DMM_PAT_VIEW_MAP_i[19:16]` `CONT_32` field, a 16-bit mode access is using the vector contained in the `DMM_PAT_VIEW_MAP_i[11:8]` `CONT_16` field and an 8-bit mode access is using the vector contained in the `DMM_PAT_VIEW_MAP_i[3:0]` `CONT_8` field. See Figure 15-6.

**Figure 15-6. PAT Direct Access Translation**

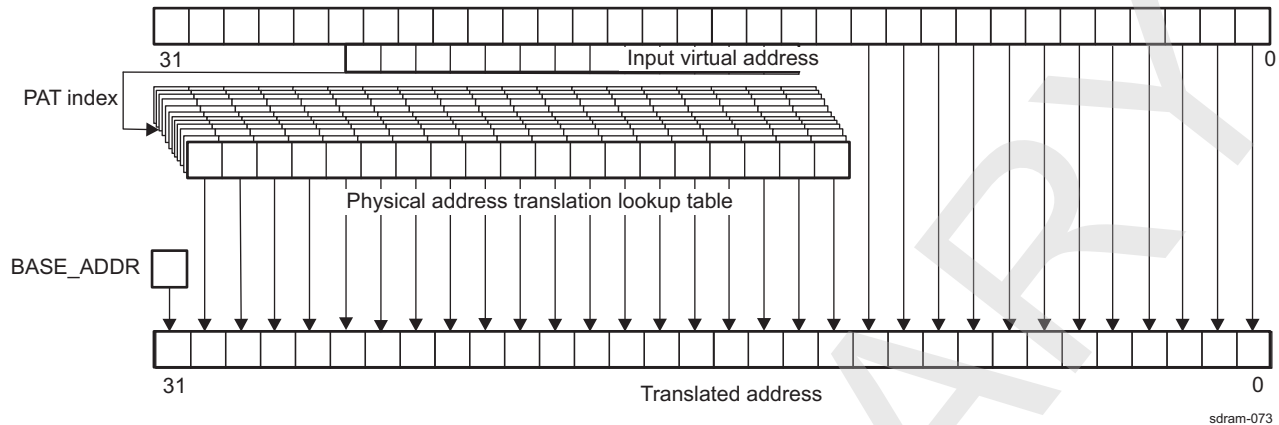


#### 15.2.3.5.1.3.3.2 PAT Indirect Access Translation

The page-grained translation is named the indirect access. In this mode the translation vector is found in the internal 32-k entry physical address translation vector table at the index given by bits 12 to 26 of the input virtual address, and the `DMM_PAT_VIEW_MAP_i` `CONT_x` field is referencing the internal physical address translation table to use. Given that the DMM is only using one of such table, in this mode the `CONT_x` field shall be written as 0. See Figure 15-7.



Figure 15-7. PAT Indirect Access Translation



Each entry of the physical address translation lookup table is a 19-bit vector that will replace bits 12 to 30 of the input virtual address. The PAT index aimed at selecting the vector in the table consists of bits 12 to 26 of the input virtual address.

15.2.3.5.1.3.3.3 PAT View Configuration

There are four different views in a DMM each view has its own `DMM_PAT_VIEW_MAP_i` register aimed at defining the kind of address translation to perform.

The PAT view type of each mode is selected by the `ACCESS_x` field in the `DMM_PAT_VIEW_MAP_i` register, *i* being the index of the considered view. When this field is set to one then the indirect access scheme is used, otherwise the PAT is performing the address translation in a direct way with the corresponding `CONT_x` vector.

When configuring a mode in a view to use the indirect access, the corresponding `CONT_x` field shall be filled with 0.

15.2.3.5.1.3.3.4 PAT Address Translation Lookup Table

The PAT address translation lookup table has the same geometry as the DMM container. It has 128 lines of 256 entries of 19-bit each. See Figure 15-8.

Figure 15-8. Physical Address Translation Table

$E_{0,0}$	$E_{1,0}$	$E_{2,0}$	$E_{3,0}$	$E_{4,0}$	$E_{251,0}$	$E_{252,0}$	$E_{253,0}$	$E_{254,0}$	$E_{255,0}$
$E_{0,1}$	$E_{1,1}$	$E_{2,1}$	$E_{3,1}$	$E_{4,1}$	$E_{251,1}$	$E_{252,1}$	$E_{253,1}$	$E_{254,1}$	$E_{255,1}$
$E_{0,2}$	$E_{1,2}$	$E_{2,2}$	$E_{251,126}$	$E_{4,2}$	$E_{251,2}$	$E_{252,2}$	$E_{253,2}$	$E_{254,2}$	$E_{255,2}$
$E_{0,125}$	$E_{1,125}$	$E_{2,125}$	$E_{3,125}$	$E_{4,127}$	$E_{251,125}$	$E_{252,125}$	$E_{253,125}$	$E_{254,125}$	$E_{255,125}$
$E_{0,126}$	$E_{1,126}$	$E_{2,126}$	$E_{3,126}$	$E_{4,126}$	$E_{251,126}$	$E_{6,126}$	$E_{253,126}$	$E_{254,126}$	$E_{255,126}$
$E_{0,127}$	$E_{1,127}$	$E_{2,127}$	$E_{3,127}$	$E_{4,127}$	$E_{251,127}$	$E_{6,127}$	$E_{253,127}$	$E_{254,127}$	$E_{255,127}$

$E_{x,y}$	4-KiB page entry
-----------	------------------

sdram-074

Each entry of the PAT address corresponds to the page in the DMM container that has the same location. The entry (74, 42) in the table corresponds to the page (74, 42) in any DMM container.

#### 15.2.3.5.1.3.3.5 **Direct Access to the PAT Physical Address Translation Table Vectors**

The PAT physical address translation table is typically refilled with specialised DMA called refill engines. Still some direct read and write access to the content of this table is granted when disabling the use of one or the other refill engine by writing a 1 in the MODE<sub>i</sub> field of the [DMM\\_PAT\\_CONFIG](#) corresponding to the refill engine *i* to disable.

In this mode, often called the debug mode, the data read in or written to the [DMM\\_PAT\\_DATA<sub>i</sub>](#) register corresponds to the vector contained in the physical address translation table which index is found in the X0 and Y0 fields of the [DMM\\_PAT\\_AREA<sub>i</sub>](#) register.

#### 15.2.3.5.1.3.3.6 **Automatic Refill Through the Refill Engines**

See [Section 15.2.4.1](#), *PAT Use Cases*.

### 15.2.3.5.2 **DMM Transaction Flows**

#### 15.2.3.5.2.1 **Non-Tiled Transaction Flow**

Each non-tiled interconnect transaction reaching the DMM, either on a TILER port or on an extra low-latency access port (ELLA), is subject to the same processing.

The TILER and ELLA blocks consider separated request, data and response paths. An overview of each path is given below, and more detailed information is given in [Section 15.2.3.5.3](#), *DMM Internal Macro-Architecture*.

On the request path, the flow consists of:

- Allocating a TILER / ELLA response context to timely generate the appropriate responses
- Splitting 2D requests in a collection of 1D requests - TILER ports only
- Splitting requests at DMM unit boundaries - the split granularity being provided by LISA mapping registers
- Allocating an available buffer in the appropriate ROBIN, for both read and write requests
- In case of a write request, allocating and updating a TILER / ELLA write context to subsequently direct the incoming write data into the relevant re-ordering buffer
- Generating the initiator-indexed priority extension - via the PEG block

On the write data path, the flow consists of forwarding incoming data to the relevant re-ordering buffer in accordance with the corresponding TILER / ELLA write context.

On the response path, responses are returned as soon as:

- At least one response have entered each related buffers, in case of read requests
- All related responses have returned from the SDRAM controller, in case of write requests
- No other previous pending response exists with the same tag

#### 15.2.3.5.2.2 **Tiled Transaction Flow**

Similarly each tiled interconnect transaction reaching the DMM is subject to the same processing.

The TILER blocks consider separated request, data and response paths. An overview of each path is given below, more detailed information being given in [Section 15.2.3.5.3](#), *DMM Internal Macro-Architecture*.

On the request path, the processing phase consists of:

- Decoding the address to qualify whether the request targets the TILER or the memory directly
- Transforming TILER-specific requests to their natural representation - address, width and height are modified accordingly

On the request path, the generation phase consists of:

- Allocating a TILER response context to timely generate the appropriate responses
- Splitting malformed - request's stride is different from the container stride and from the double of this container stride - tiled 2D requests in a collection of 1D requests
- Splitting tiled requests at tile boundaries
- Performing the page-based address translation - via the PAT block
- Allocating an available buffer in the appropriate ROBIN, for both read and write requests
- In case of a write request, allocating and updating a TILER write context to subsequently direct the incoming write data into the relevant re-ordering buffer
- Generating the initiator-indexed priority extension - via the PEG block

On the data path, any incoming data is transformed in accordance with the corresponding TILER write context and sent to the appropriate re-ordering buffer.

On the response path, responses are returned as soon as:

- A minimal number of responses have entered the corresponding buffers, in case of read requests
- All related responses have returned from the SDRAM controllers, in case of write requests
- No other previous pending response exists with the same tag

### **15.2.3.5.3 DMM Internal Macro-Architecture**

This sub-section describes the DMM internal macro-architecture, specifically:

- Input request decoding,
- Physical address translation and synchronised translation table reloading,
- Output request and response generation,
- Memory mapping in the system addressing space - including interleaving,
- Tag management,
- Priority handling,
- Contexts,
- Maximum allowed burst size,
- Reconstruction buffer dimensioning.

#### **15.2.3.5.3.1 ELLA Description**

The Extra Low Latency Access (ELLA) block is basically a TILER block which has been simplified to reduce the latency. In this respect, the ELLA block:

- Is limited to 1D bursts only
- Is not capable of performing tiling conversions
- Does not interact with the PAT block

The ELLA block main role is to split incoming requests at DMM atomic unit boundaries to ensure that requests sent to the SDRAM controller fit in a single SDRAM page. More precisely, the ELLA block is responsible for:

- Allocating an internal response context to timely generate the appropriate responses
- Splitting requests at DMM atomic section boundaries
- Requesting buffer allocation in the appropriate ROBIN
- In case of a write request, allocating and updating an internal write context to subsequently direct the incoming write data into the relevant re-ordering buffer.

#### **15.2.3.5.3.2 LISA Description**

The Local Interconnect and Synchronisation Agent (LISA) is a full crossbar aimed at setting priorities, managing tags and memory mapping.

The interconnect routes:

- TILER requests on the ROBIN initiator nodes
- TILER write data on the ROBIN write buffers

- ROBIN read data to the relevant TILER initiators.

The LISA block registers are [DMM\\_LISA\\_MAP\\_i](#) (i = 0 to 3) and [DMM\\_LISA\\_LOCK](#).

The priority is given to the low-latency port (ELLA). When 2 ROBIN are in use, it interleaves the two ports at a programmable boundary from 128 Bytes or more (configurable with [DMM\\_LISA\\_MAP\\_i\[19:18\]](#) SDRG\_INTL bit field).

### 15.2.3.5.3.3 PAT Description

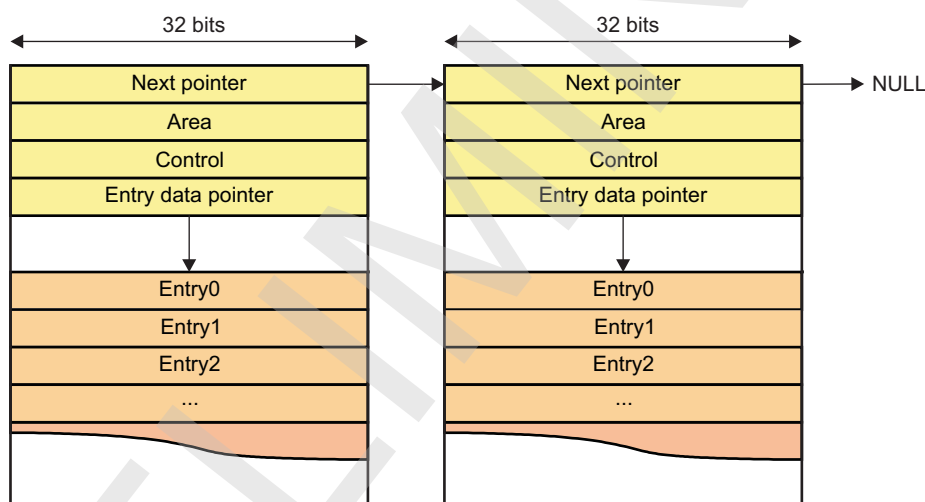
The Physical Address Translator (PAT) block is responsible for mapping physical pages to each TILER container page. The internal address translation memories used in the PAT block are designed with retention flip-flops. It consists of:

- a memory-based LUT which has the same geometry as the container pages
- a refill engine for modifying entries in a given area of the internal LUT

A PAT descriptor is a singly-linked list node as shown [Figure 15-9](#), containing:

- a description of the LUT area to reload
- a description of how to reload the defined LUT area
- a pointer to the location where the corresponding LUT entries are stored

**Figure 15-9. PAT Descriptors**



sdram-065

PAT descriptors are chained and processed until a NULL pointer is encountered. The PAT bank allocation scheme allows the update of 4 consecutive entries of a line in a single cycle whatever the refilling orientation.

The PAT descriptor structure directly maps the [DMM\\_PAT\\_DESCR\\_i](#), [DMM\\_PAT\\_AREA\\_i](#), [DMM\\_PAT\\_CTRL\\_i](#) and [DMM\\_PAT\\_DATA\\_i](#) registers (i = 0 to 3).

The PAT refill engine can be started by:

- filling manually all the necessary registers: the [DMM\\_PAT\\_AREA\\_i](#) register with the (x0, y0) (x1, y1) area definition, the [DMM\\_PAT\\_DATA\\_i](#) with the physical address of the corresponding area entry table, the [DMM\\_PAT\\_CTRL\\_i\[6:4\]](#) DIRECTION with the relevant (S, Y, X) direction of the area refill and the [DMM\\_PAT\\_CTRL\\_i\[0\]](#) START with a 1.
- writing the physical address of a memory-mapped PAT descriptor in the [DMM\\_PAT\\_DESCR\\_i](#) which will update: the [DMM\\_PAT\\_AREA\\_i](#) with the area value of the descriptor, the [DMM\\_PAT\\_DATA\\_i](#) with the data value of the descriptor, the [DMM\\_PAT\\_CTRL\\_i](#) with the ctrl value of the descriptor and the [DMM\\_PAT\\_DESCR\\_i](#) with the next value of the descriptor.
- In any case the data part of the PAT refill only starts when [DMM\\_PAT\\_CTRL\\_i\[0\]](#) START is asserted.

The [DMM\\_PAT\\_STATUS\\_i](#) register can be used to see if the process has completed without errors.

#### **15.2.3.5.3.4 PEG Description**

The priority extension generator (PEG) is a dynamic software-programmable initiator-indexed table of priorities. Its unique role is to bind a priority to an initiator - for initiators that do not generate their own per-transaction priority.

When an interconnect request enters the DMM, its priority is extracted from the PEG lookup table.

The 16 priority entries are software-programmable with [DMM\\_PEG\\_PRIO\\_k](#). A priority of 0 defines the highest priority and a priority of 7 defines the lowest priority. Given an index  $x$ , the priority related to this index is given in the Px field.

Each of these two registers is split in eight 4-bit fields, each field mapping an entry of the lookup table with:

- the 3-bit priority coded on the 3 least significant bits
- a  $\overline{W}$  field-specific active-low local write enable bit, always read as 0, on the most significant bit. The role of the  $\overline{W}$  bit is to allow the modification of a single entry without requiring a read-modify-write sequence. As its  $\overline{W}$  bits are always read as 0, writing back the modified register will update all priority fields of the register.

Although this priority information is generated prior to entering the LISA block, it is not used internally in the local interconnect arbitration, but forwarded to the SDRAM controller, where it indicates the command priority.

It is also possible to give a priority for the internal PAT engine thanks to the [DMM\\_PEG\\_PRIO\\_PAT](#) register.

#### **15.2.3.5.3.5 ROBIN Description**

The re-ordering buffer and initiator node (ROBIN) is a block aimed at providing some working buffering for:

- Converting data and responses to-and-fro between raster and tiled organisations and a master port to connect to the SDRAM controller.

The ROBIN block does:

- Request Forwarding
- Write data and Response buffering
- Keeps write data ordering
- Intra-word tiling and orientation transforms
- Tag handling

#### **15.2.3.5.3.6 TILER Description**

The Tiling and Isometric Lightweight Engine for Rotation (TILER) main functionality is the request conversion due to the tiling. The TILER block is then responsible for:

- Decoding the address to qualify whether the request targets the TILER or the memory directly
- Converting TILER-specific requests to their natural representation - address, width and height are modified accordingly
- Allocating an internal response context to timely generate the appropriate responses
- Splitting tiled requests at tile boundaries and not tiled requests at DMM atomic section boundaries
- Requesting the page-based address translation
- Requesting buffer allocation in the appropriate ROBIN
- In case of a write request, allocating and updating an internal write context to subsequently direct the incoming write data into the relevant re-ordering buffer.

The interdependent tiling and isometric transform concepts introduced in the TILER are fully described in the following section.

### 15.2.3.6 TILER

#### 15.2.3.6.1 TILER Concepts

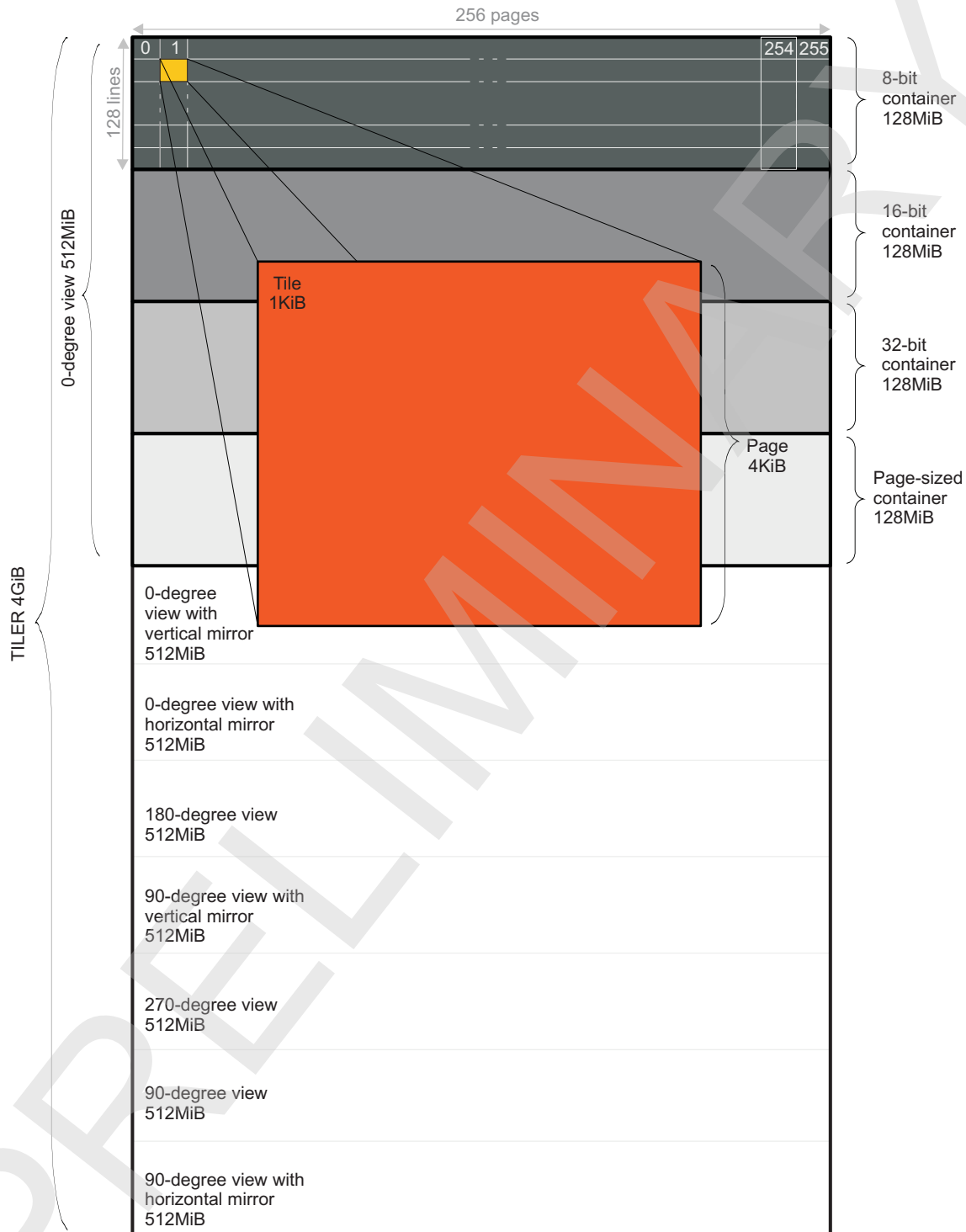
This section describes all concepts behind the TILER transforms, through a top-down approach starting from the main object container.

##### 15.2.3.6.1.1 TILER Rationale

This section is a synthesis of all TILER concepts, giving one rule per TILER structure level.

[Figure 15-10](#) shows the TILER address space structure for tiled modes.

**Figure 15-10. TILER Address Space Structure for Tiled Modes**



sdram-005\_public

**15.2.3.6.1.1.1 The TILER is a 4-GiB Virtual Address Space Composed of Eight Views**

There is one view for each of the eight possible way of scanning a frame-buffer:

1. From left to right then from top to bottom

2. From right to left then from top to bottom
3. From left to right then from bottom to top
4. From right to left then from bottom to top
5. From top to bottom then from left to right
6. From top to bottom then from right to left
7. From bottom to top then from left to right
8. From bottom to top then from right to left

#### **15.2.3.6.1.1.2 A View is a 512-MiB Virtual Address Space Composed of Four Containers**

There is one container per element size to allow correct access patterns in any of the eight possible orientations. The container is the entity where all objects of a given element type are allocated.

The element is the entity of maximum size - 8 bits, 16 bits, 32 bits or page-sized - which is invariant in any orientation.

#### **15.2.3.6.1.1.3 A Container is a 128-MiB Virtual Address Space**

A container is composed of an array of 128 lines of 256 pages of 4 kiB each.

The page defines the granularity of physical memory allocation through a physical address translation unit - MMU.

#### **15.2.3.6.1.1.4 A Page is a 4-kiB Virtual Address Space**

A page is composed of two lines. Each line consists of two tiles.

#### **15.2.3.6.1.1.5 A Tile is a 1-kiB Address Space**

The tile is designed to offer bi-dimensional data locality in a single SDRAM page. In this respect, it has been sized to 1 kiB, that is, to the size of the smallest SDRAM page.

#### **15.2.3.6.1.2 TILER Modes**

The TILER is supporting three major modes - bypass, page and tiled - each having a specific output request generation.

##### **15.2.3.6.1.2.1 Bypass Mode**

As stated in its name, this mode is transparent at the TILER perspective. Still, at the DMM perspective,

- 2D block bursts are broken down on a line-basis in a set of incremental bursts
- Incremental bursts - including those issued by a 2D block burst breakdown - are split at the DMM atomic unit of the section hit by the burst at:
  - at the interleaving granularity of the section - 128 byte, 256 byte or 512 byte - in interleaved sections
  - at 1-kiB boundary in non-interleaved sections

##### **15.2.3.6.1.2.2 Page Mode**

The purpose of this mode is to use the DMM address translation mechanism for non-tiled accesses. In this respect it is similar to the bypass mode,

- 2D block bursts are broken down on a line-basis in a set of incremental bursts
- Incremental bursts - including those issued by a 2D block burst breakdown - are split at:
  - the interleaving granularity of the section - 128 byte, 256 byte or 512 byte - in interleaved sections
  - 1-kiB boundary in non-interleaved sections

##### **15.2.3.6.1.2.3 Tiled Mode**

The tiled mode has two major breakdown algorithms,



- One for well formed 2D block requests - conforming to the orientation, mode and stride given in Table 15-9
- One for 1D incremental requests and ill-formed 2D block requests.

**Table 15-9. Well Formed Tiled Mode 2D Block Requests**

Orientation			Mode		Stride	Description
S	Y	X	M1	M0	(bytes)	
0	x	x	0	0	16384	Plain access to an 8-bit progressive frame in 0° or 180°
					32768	Field access to an 8-bit interlaced frame 0° or 180°
			0	1	32768	Plain access to a 16-bit progressive frame in 0° or 180°
					65536	Field access to a 16-bit interlaced frame 0° or 180°
			1	0	32768	Plain access to a 32-bit progressive frame in 0° or 180°
					65536	Field access to a 32-bit interlaced frame 0° or 180°
1	x	x	0	0	8192	Plain access to an 8-bit progressive frame in 90° or 270°
					16384	Field access to an 8-bit interlaced frame 90° or 270°
			0	1	8192	Plain access to a 16-bit progressive frame in 90° or 270°
					16384	Field access to a 16-bit interlaced frame 90° or 270°
			1	0	16384	Plain access to a 32-bit progressive frame in 90° or 270°
					32768	Field access to a 32-bit interlaced frame 90° or 270°

Similarly to the bypass and page modes, ill formed 2D block requests are broken down on a line basis in a set of incremental bursts. In tiled mode however, these incremental virtual bursts do not translate to 1D physical burst requests.

**15.2.3.6.1.3 Object Container Definition**

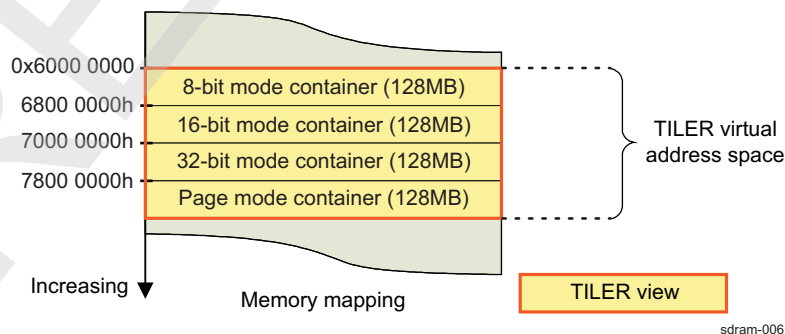
The object container is the unique addressable entry point of the TILER. It is a 128-MiB virtual address space, where all objects of a same kind - and orientation - are allocated.

Four main types of containers are present in the TILER, each one being referred by a mode:

- A 8-bit element mode, for efficiently accessing bi-dimensional arrays of 8-bit data
- A 16-bit element mode, for efficiently accessing bi-dimensional arrays of 16-bit data
- A 32-bit element mode, for efficiently accessing bi-dimensional arrays of 32-bit data
- A page mode, for efficient 1D accesses

Figure 15-11 shows the TILER object containers and views

**Figure 15-11. TILER Object Containers and Views**

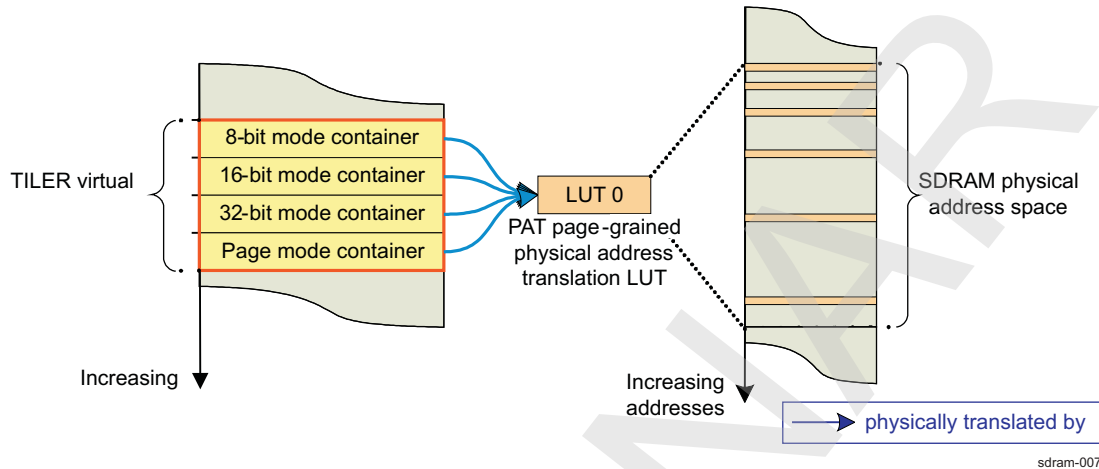


The 512-MiB virtual address space composed of four 128-MiB object containers of different modes is called a view. Given that 8 orientations are available per mode, the TILER is actually managing 32 kinds of containers.

The physical memory footprint of a 512-MiB TILER view is directly subject to the nature of the physical address translator unit - PAT.

A unique physical address translation table - LUT (Look Up Table) is instantiated in the DMM. This table is shared by all TILER modes. Hence, each of the four modes can not be given their own private page-grained physical address translation table - LUT. A maximum of 128 MiB of objects among all TILER modes can be managed as illustrated in [Figure 15-12](#).

**Figure 15-12. TILER Memory Footprint with PAT and Shared Physical Address Translation LUT**



If the four modes have all their own separated virtual 128-MiB object containers, these containers are all mapped to the same piecewise 128-MiB physical address space, and are then not physically separated. Consequently, a memory-related system constraints is that no more than 128-MiB of objects can be available simultaneously in a TILER view.

---

**NOTE:** It is a software duty to ensure that any object allocated in a mode do not physically overlap with any other object, even in another mode.

---

#### 15.2.3.6.1.4 Page Definition

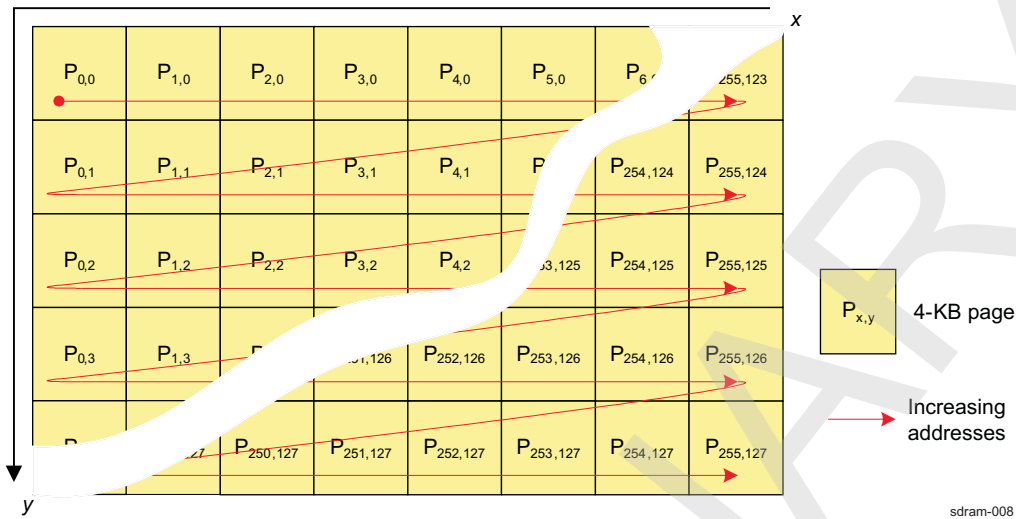
A TILER page defines the granularity of object allocation in virtual TILER containers.

Given that the sub-page structure is mode specific, the page is the smallest granularity common to all modes, making it the granularity to consider in the TILER resource manager for object allocation.

##### 15.2.3.6.1.4.1 Container Geometry with 4-kiB Pages

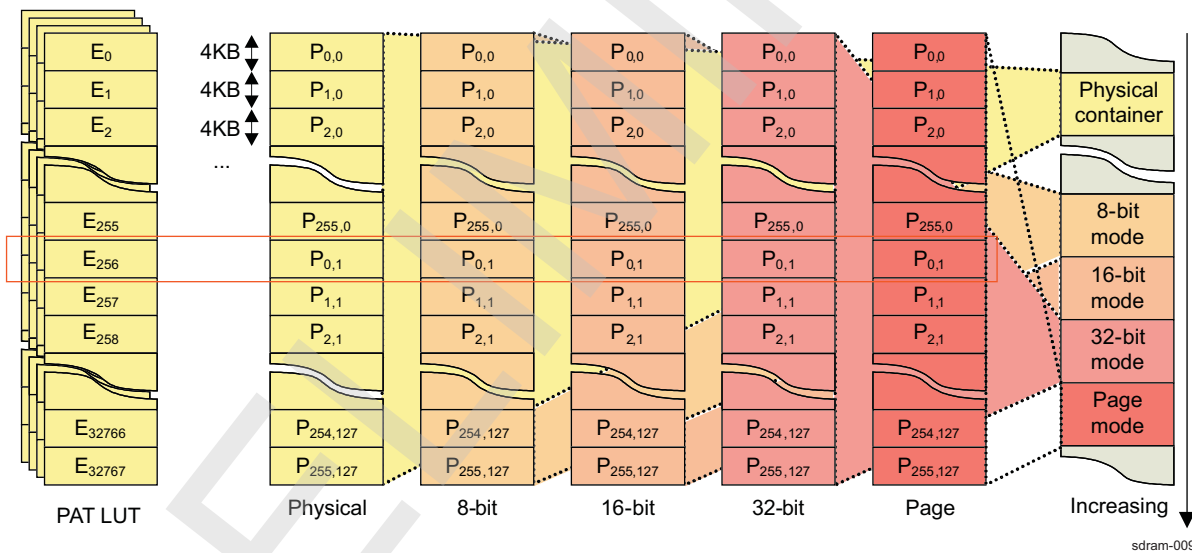
As pages size is 4-kiB, any 128-MiB object container is a set of 32768 pages, organised in an array of 256 columns and 128 rows, as shown in [Figure 15-13](#).

Figure 15-13. Object Container Geometry with 4-kiB Pages



This array of pages is mapped to the system address space as shown in Figure 15-14.

Figure 15-14. TILER Page Mapping when Using 4-kiB Pages



In any 128-MiB object container, the 4-kiB page  $P_{x,y}$  at column  $x$  ( $0 \leq x < 256$ ) and row  $y$  ( $0 \leq y < 128$ ), is found at an offset of  $4096 \cdot (x + 256 \cdot y)$  bytes from the base address of the related object container.

Similarly, the page  $P_{x,y}$  at column  $x$  ( $0 \leq x < 256$ ) and row  $y$  ( $0 \leq y < 128$ ), is translated by the LUT entry  $E_{x+256 \cdot y}$  found at the index  $x + 256 \cdot y$ .

#### 15.2.3.6.1.4.2 Container Geometry and Page Mapping Summary

The TILER has a page size of 4096 bytes. The page  $P_{x,y}$ :

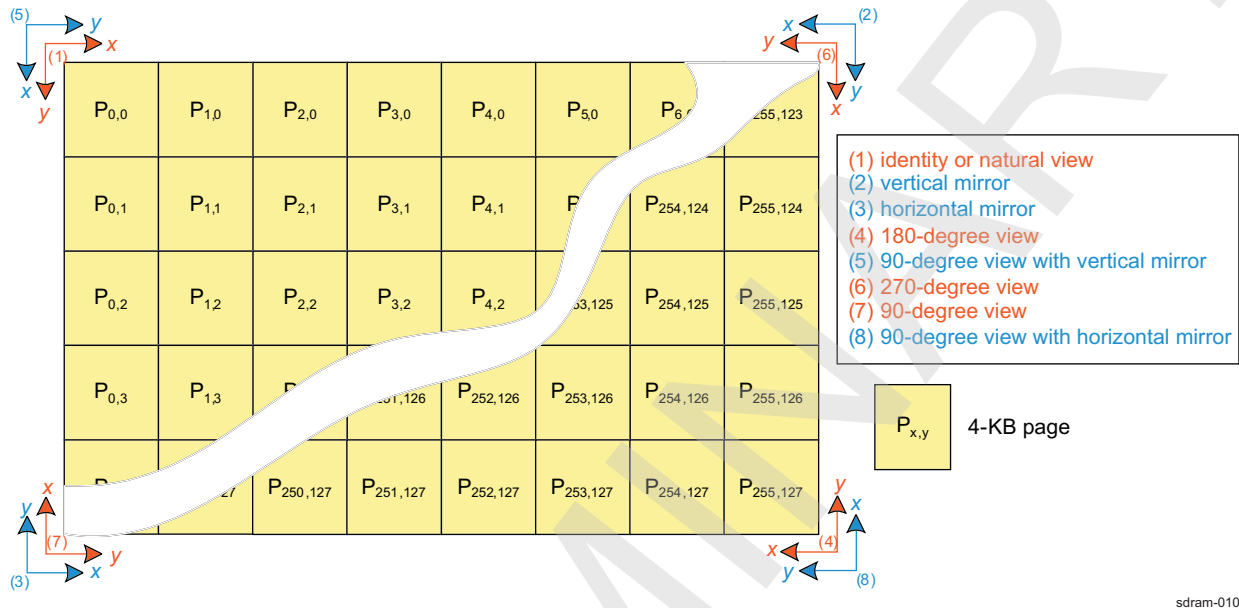
- has  $\max_x = 256$  and  $\max_y = 128$ .
- is found at an offset of:  $4096 \cdot (x + \max_x \cdot y)$  bytes from the base address of the related object container
- is translated by the entry found at the index  $(x + \max_x \cdot y)$  of the LUT table

### 15.2.3.6.1.5 Orientation

This section describes the eight on-the-fly orientation-related isometric transforms, corresponding to all available changes of orthonormal basis in the TILER container bi-dimensional space.

Figure 15-15 shows isometric transforms in the TILER container

**Figure 15-15. Isometric Transforms in the TILER Container**



Mathematically speaking, all these transforms correspond to the composition of a  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , or  $270^\circ$  rotation with an optional reflection. The nature of this orientation is based on the three following binary parameters:

- X to change the direction of the  $x$  axis of the TILER container;
- Y to change the direction of the  $y$  axis of the TILER container;
- S to swap the modified  $x$  and  $y$  axis.

From now on, and in the rest of this document, the term orientation refers to any composition of a "quadrant" rotation with an optional horizontal - flip-flop - or vertical mirroring.

### 15.2.3.6.1.6 Tile Definition

A tile is a subdivision of a page, aimed at:

- Representing a 2D block to better balance the accesses in both directions
- Ensuring that any "tiled" access that fits within a tile is made atomic in the SDRAM controller, and fits in a single SDRAM memory page
- Minimising the number of SDRAM page openings per 2D block transfer

The tile is defined as a 1-kB 2D block, and a 4-kB page as an array of 2 lines of 2 tiles each.

When the considered page is in an interleaved DMM section, it is necessary that:

- the DMM memory interleaving size of tiled accesses is set to 1 kB - a tile size - so that any tiled request that fits within a tile, fits in a single SDRAM memory page
- any request that spans over 2 or 4 tiles, is distributed on a maximum number of SDRAM memory controllers

### 15.2.3.6.1.7 TILER Virtual Addressing

The TILER can be virtually accessed in four different modes, namely 8-bit, 16-bit, 32-bit and page modes. Each mode defines the element granularity to apply isometric transforms, as summarized in Table 15-10.

**Table 15-10. Coding and Description of TILER Modes**

Mode	Name	Granularity (element size)
0	8-bit tiled mode	8 bits
1	16-bit tiled mode	16 bits
2	32-bit tiled mode	32 bits
3	Page mode	4096 bytes

For instance, making a vertical mirroring of a 16-byte horizontal line containing the word 000102030405060708090A0B0C0D0E0Fh, leads to:

- 0F0E0D0C0B0A09080706050403020100h in 8-bit tiled mode
- 0E0F0C0D0A0B08090607040502030001h in 16-bit tiled mode
- 0C0D0E0F08090A0B0405060700010203h in 32-bit tiled mode
- 000102030405060708090A0B0C0D0E0Fh in page mode - unchanged as the element granularity is 4 KB.

Besides, as each of the eight orientations is available for any of the four modes, this gives 32 TILER addressing possibilities, as summarized in [Table 15-11](#).

**Table 15-11. Coding and Description of TILER Orientations**

S	Y	X	Description	Alternate description
0	0	0	0° view	Natural view
0	0	1	0° view with vertical mirror	180° view with horizontal mirror
0	1	0	0° view with horizontal mirror	180° view with vertical mirror
0	1	1	180° view	
1	0	0	90° view with vertical mirror	270° view with horizontal mirror
1	0	1	270° view	
1	1	0	90° view	
1	1	1	90° view with horizontal mirror	270° view with vertical mirror

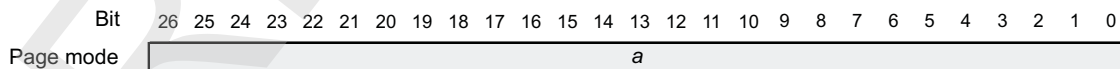
**15.2.3.6.1.7.1 Page Mode Virtual Addressing and Characteristics**

When used in page mode, the 128-MiB TILER space is seen as an orientation-specific sequence of 32768 pages of 4 kiB each. The access sequence inside a page is left unchanged.

Therefore, in page mode, the TILER is accessed similarly to any 128-MiB memory, with a 27-bit byte-based address.

**NOTE:** From now on the address is noted "a".

**Figure 15-16. Page Mode Virtual Addressing**



sdram-027

**15.2.3.6.1.7.2 Tiled Mode Virtual Addressing and Characteristics**

When used in tiled mode, the 128-MiB TILER space is seen as a giant frame-buffer - the container. The addressing and characteristics of this giant frame-buffer depends on:

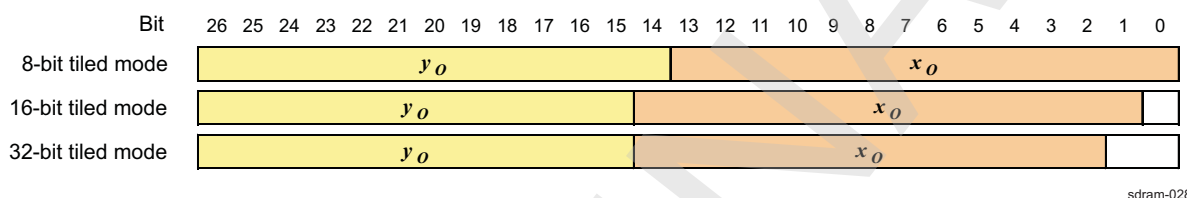
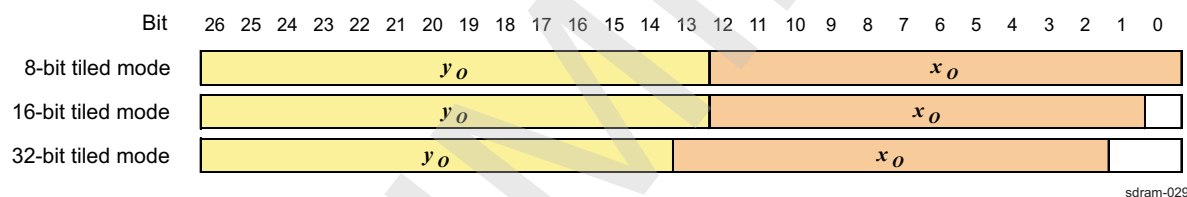
- The tiled mode, which defines the considered atomic element size.
- The orientation, which potentially swaps its x and y axis - and hence the container geometry.

[Table 15-12](#) summarizes tiled mode container characteristics.

**Table 15-12. Tiled Mode Container Characteristics**

Orientation S	$\Upsilon$	$\bar{X}$	Element size (bits)	Width (elements)	Height (elements)	Stride (bytes)	
						Progressive	Interlaced
0	x	x	8	16384	8192	16384	32768
			16	16384	4096	32768	65536
			32	8192	4096	32768	65536
1	x	x	8	8192	16384	8192	16384
			16	4096	16384	8192	16384
			32	4096	8192	16384	32768

As a result, the coordinate  $(x_0, y_0)$  of a pixel in an oriented view is translated in a virtual address as shown in Figure 15-17 and Figure 15-18.

**Figure 15-17. Tiled Mode Addressing in 0° or 180° Orientations (S=0)****Figure 15-18. Tiled Mode Addressing in 90° or 270° Orientations (S=1)**

### 15.2.3.6.1.7.3 Element Ordering in the TILER Container

This section highlights how elements - 8-bit data, 16-bit data, 32-bit data or 4-kB page - are ordered in the container. In other words, this section highlights how the path of incrementing virtual addresses is mapped in the container.

It is interesting to note that whatever the mode, and hence the element size, the sequence for ordering the elements in their related container is strictly similar and only depends on the related orientation. In other words:

- Mode is about element granularity
- Orientation is about change of orthonormal basis for ordering the elements in the mode-specific container.

A corollary to the previous statements is that in a given mode the internal structure of an element is unchanged whatever the orientation. In page mode for instance, the offset of a word inside a page is invariant by orientation; the content of a page is always accessed in the same manner.

In the next sections, the natural container orthonormal basis is referenced as:

$$(\vec{x}_N, \vec{y}_N)$$

sdram-030

and the oriented orthonormal basis as:

$$(\vec{x}_O, \vec{y}_O)$$

sdram-031

15.2.3.6.1.7.3.1 Natural View or 0° View (orientation 0)

This orientation defined by  $S=0$ ,  $\bar{Y}=0$  and  $\bar{X}=0$  and means that the operated change of basis is:

$$\begin{cases} \bar{x}_O = \bar{x}_N \\ \bar{y}_O = \bar{y}_N \end{cases}$$

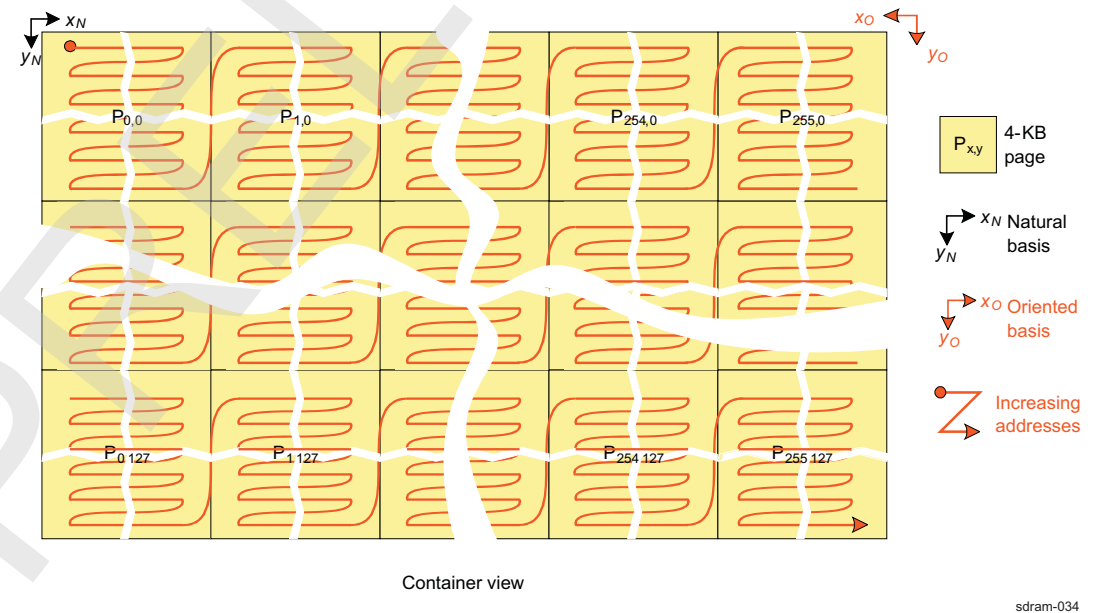
sdram-032

In any TILER mode, the elements are then ordered from left-to-right and then from top-to-bottom in their container, as shown in Figure 15-19 and Figure 15-20.

Figure 15-19. Tiled Mode Ordering of Elements in Natural View



Figure 15-20. Page Mode Ordering of Elements in Natural View



### 15.2.3.6.1.7.3.2 0° View with Vertical Mirror or 180° View with Horizontal Mirror (Orientation 1)

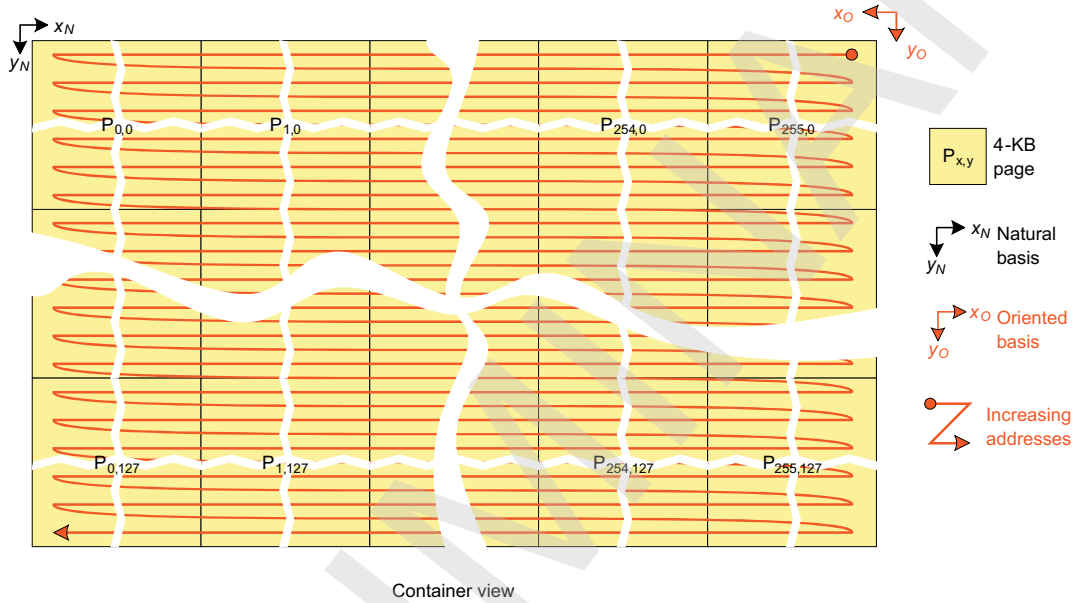
This orientation defined by  $S=0$ ,  $\bar{Y}=0$  and  $\bar{X}=1$  and means that the operated change of basis is:

$$\begin{cases} \bar{x}_O = -\bar{x}_N \\ \bar{y}_O = \bar{y}_N \end{cases}$$

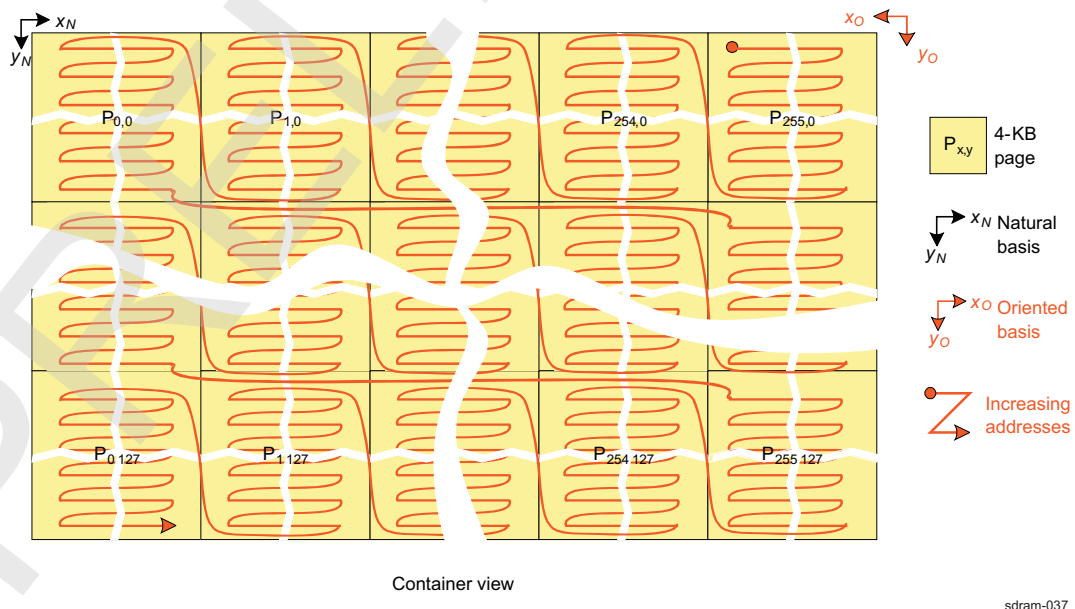
sdram-035

In any TILER mode, the elements are then ordered from right-to-left and then from top-to-bottom in their container, as shown in Figure 15-21 and Figure 15-22.

**Figure 15-21. Tiled Mode Ordering of Elements in 0° View with Vertical Mirror**



**Figure 15-22. Page Mode Ordering of Elements in 0° View with Vertical Mirror**





**15.2.3.6.1.7.3.3 0° View with Horizontal Mirror or 180° View with Vertical Mirror (Orientation 2)**

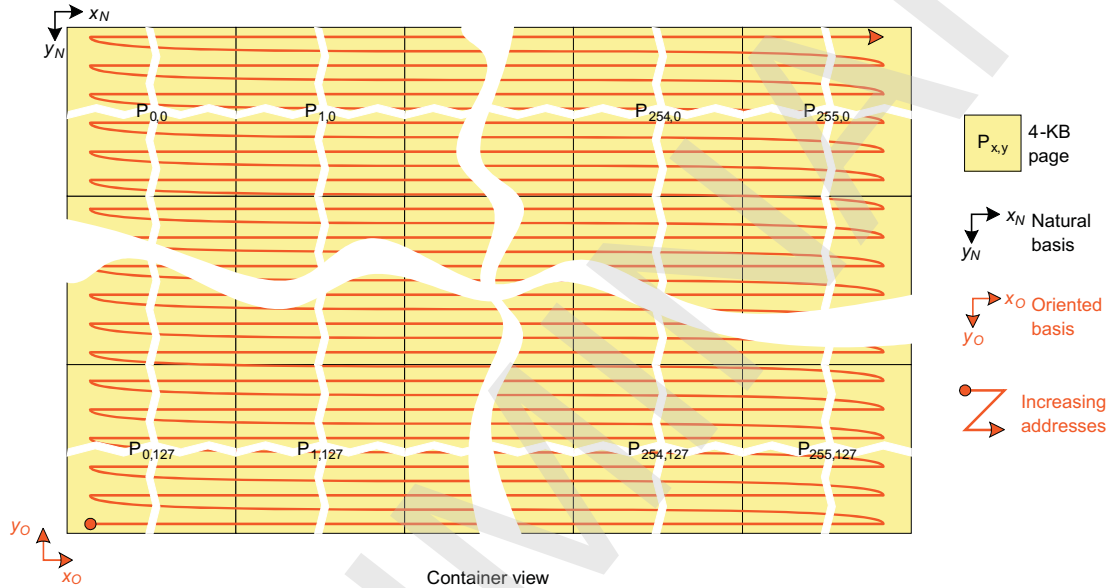
This orientation defined by  $S=0$ ,  $\bar{Y}=1$  and  $\bar{X}=0$  and means that the operated change of basis is:

$$\begin{cases} \bar{x}_O = \bar{x}_N \\ \bar{y}_O = -\bar{y}_N \end{cases}$$

sdram-038

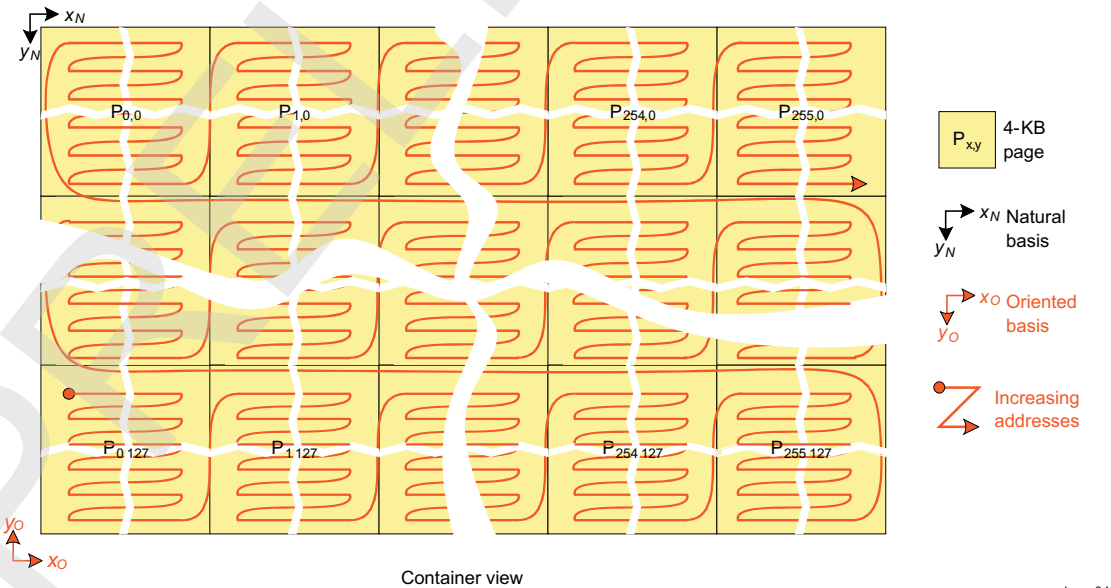
In any TILER mode, the elements are then ordered from left-to-right and then from bottom-to-top in their container, as shown in Figure 15-23 and Figure 15-24.

**Figure 15-23. Tiled Mode Ordering of Elements in 0° View with Horizontal Mirror**



sdram-039

**Figure 15-24. Page Mode Ordering of Elements in 0° View with Horizontal Mirror**



sdram-040

**15.2.3.6.1.7.3.4 180° View (Orientation 3)**

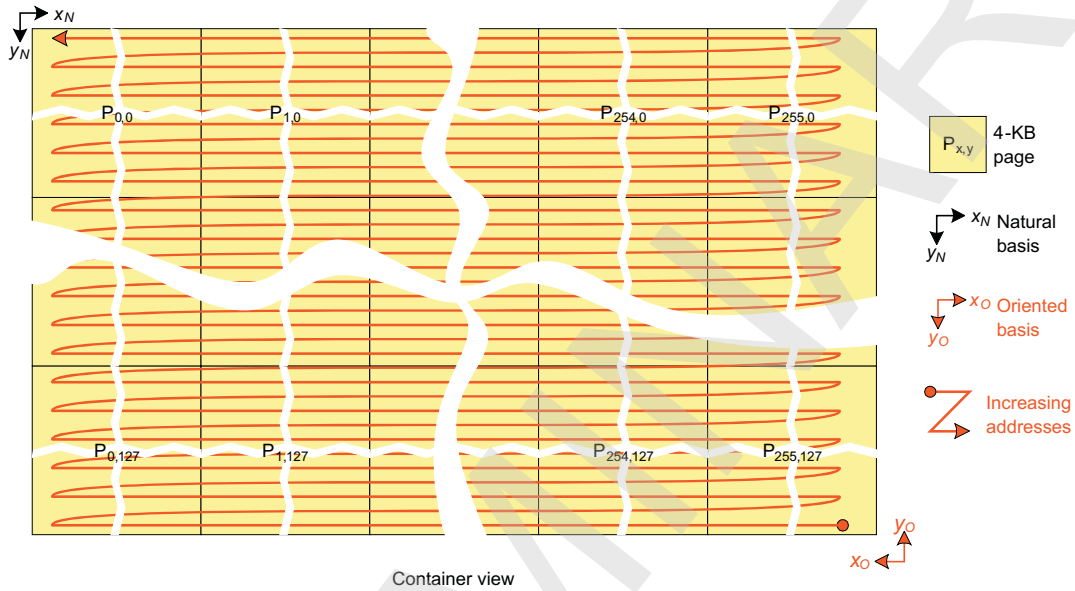
This orientation defined by  $S=0$ ,  $\bar{Y}=1$  and  $\bar{X}=1$  and means that the operated change of basis is:

$$\begin{cases} \bar{x}_O = -\bar{x}_N \\ \bar{y}_O = -\bar{y}_N \end{cases}$$

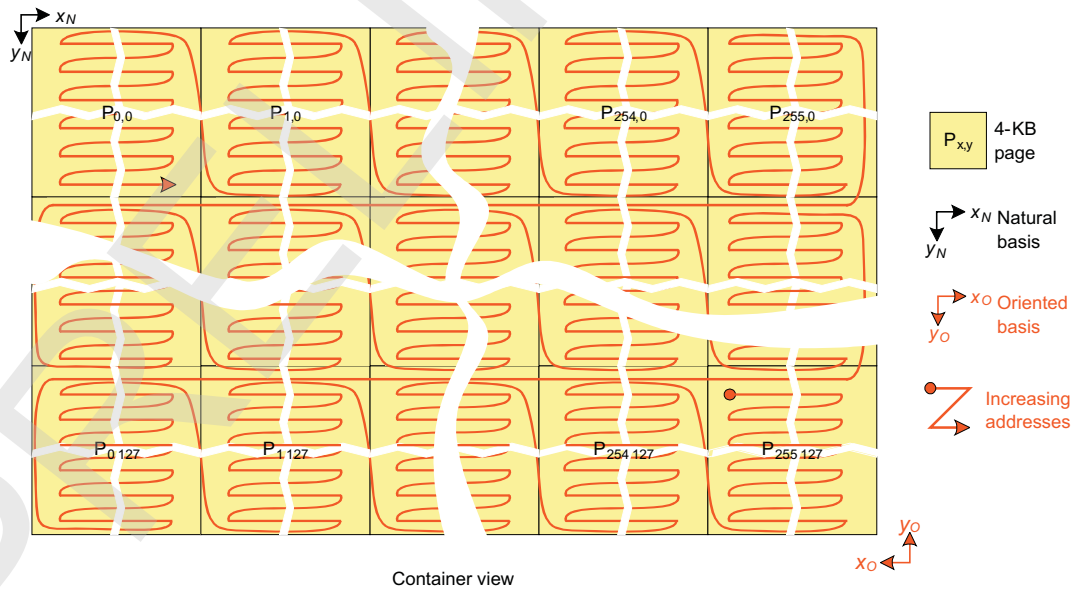
sdram-041

In any TILER mode, the elements are then ordered from right-to-left and then from bottom-to-top in their container, as shown in Figure 15-25 and Figure 15-26.

**Figure 15-25. Tiled Mode Ordering of Elements in 180° View**



**Figure 15-26. Page Mode Ordering of Elements in 180° View**



**15.2.3.6.1.7.3.5 90° View with Vertical Mirror or 270° View with Horizontal Mirror (Orientation 4)**

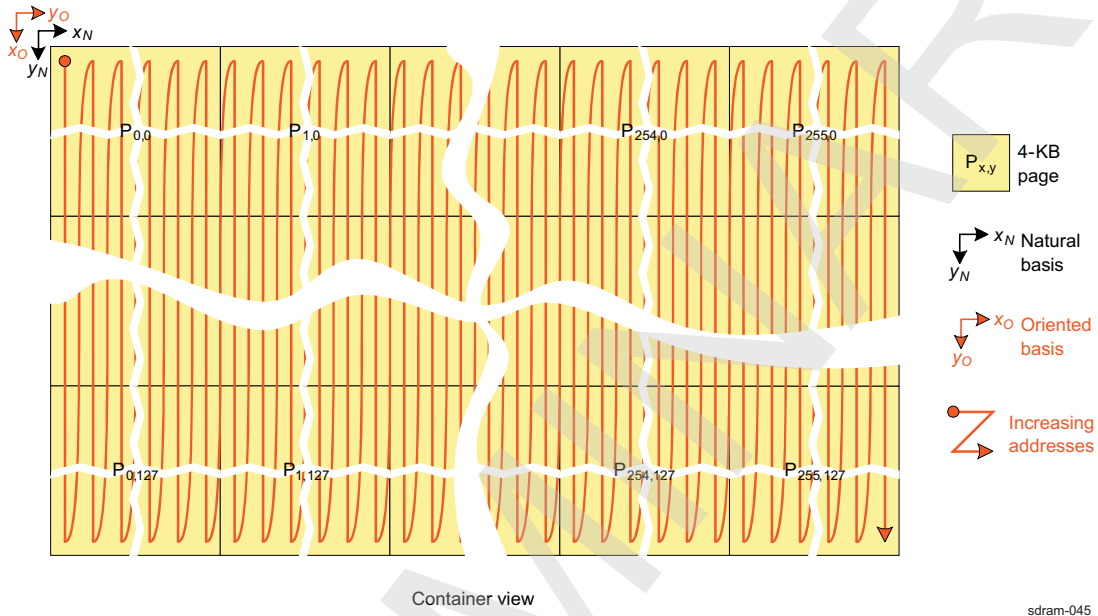
This orientation defined by  $S=1$ ,  $\bar{Y}=0$  and  $\bar{X}=0$  and means that the operated change of basis is:

$$\begin{cases} \bar{x}_O = \bar{y}_N \\ \bar{y}_O = \bar{x}_N \end{cases}$$

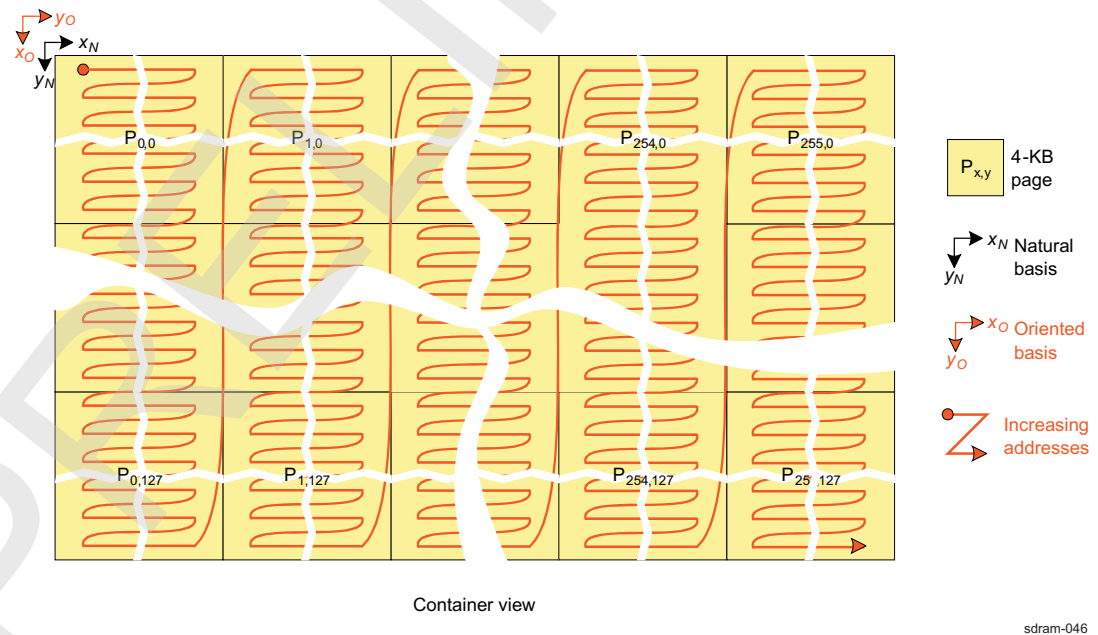
sdram-044

In any TILER mode, the elements are then ordered from top-to-bottom and then from left-to-right in their container, as shown in Figure 15-27 and Figure 15-28.

**Figure 15-27. Tiled Mode Ordering of Elements in 90° View with Vertical Mirror**



**Figure 15-28. Page Mode Ordering of Elements in 90° View with Vertical Mirror**



**15.2.3.6.1.7.3.6 270° View (Orientation 5)**

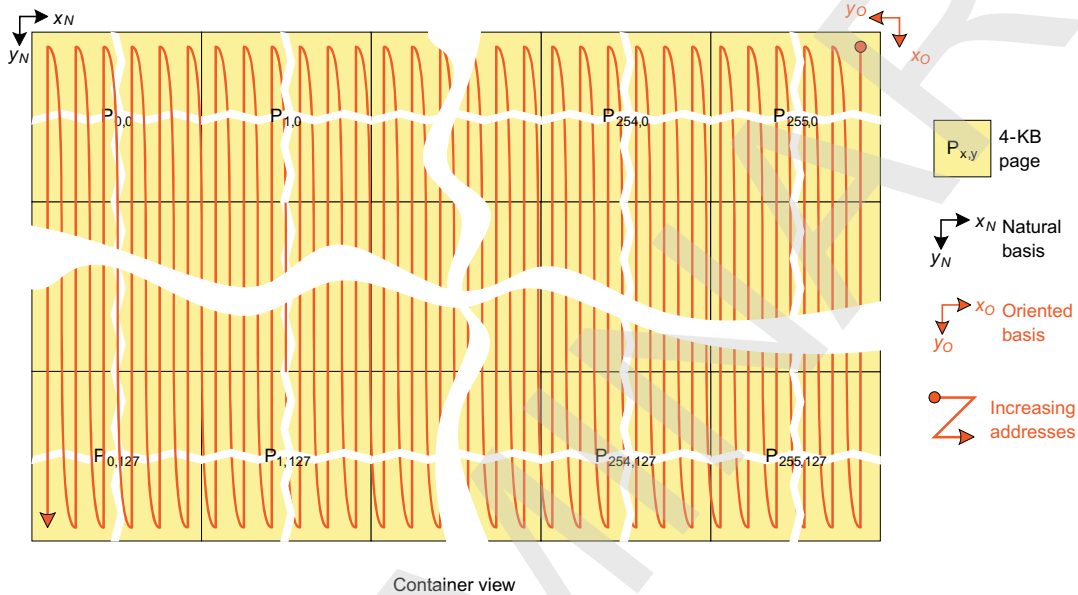
This orientation defined by  $S=1$ ,  $\bar{Y}=0$  and  $\bar{X}=1$  and means that the operated change of basis is:

$$\begin{cases} \bar{x}_O = \bar{y}_N \\ \bar{y}_O = -\bar{x}_N \end{cases}$$

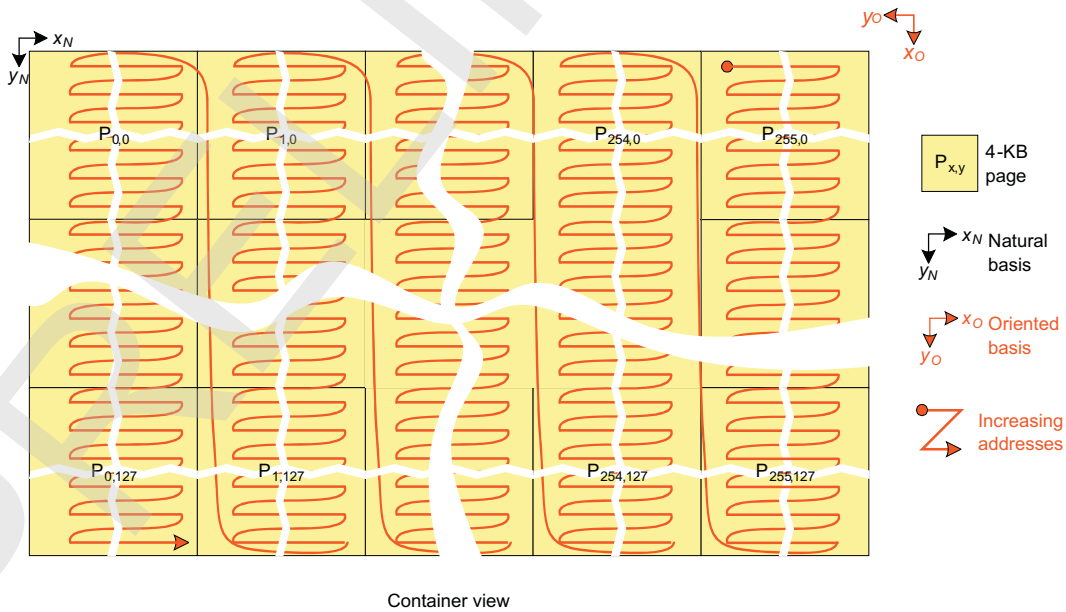
sdram-047

In any TILER mode, the elements are then ordered from top-to-bottom and then from right-to-left in their container, as shown in Figure 15-29 and Figure 15-30.

**Figure 15-29. Tiled Mode Ordering of Elements in 270° View**



**Figure 15-30. Page Mode Ordering of Elements in 270° View**



**15.2.3.6.1.7.3.7 90° View (Orientation 6)**

This orientation defined by  $S=1$ ,  $\bar{Y}=1$  and  $\bar{X}=0$  and means that the operated change of basis is:

$$\begin{cases} \bar{x}_O = -\bar{y}_N \\ \bar{y}_O = \bar{x}_N \end{cases}$$

sdram-050

In any TILER mode, the elements are then ordered from bottom-to-top and then from left-to-right in their container, as shown in Figure 15-31 and Figure 15-32.

Figure 15-31. Tiled Mode Ordering of Elements in 90° View

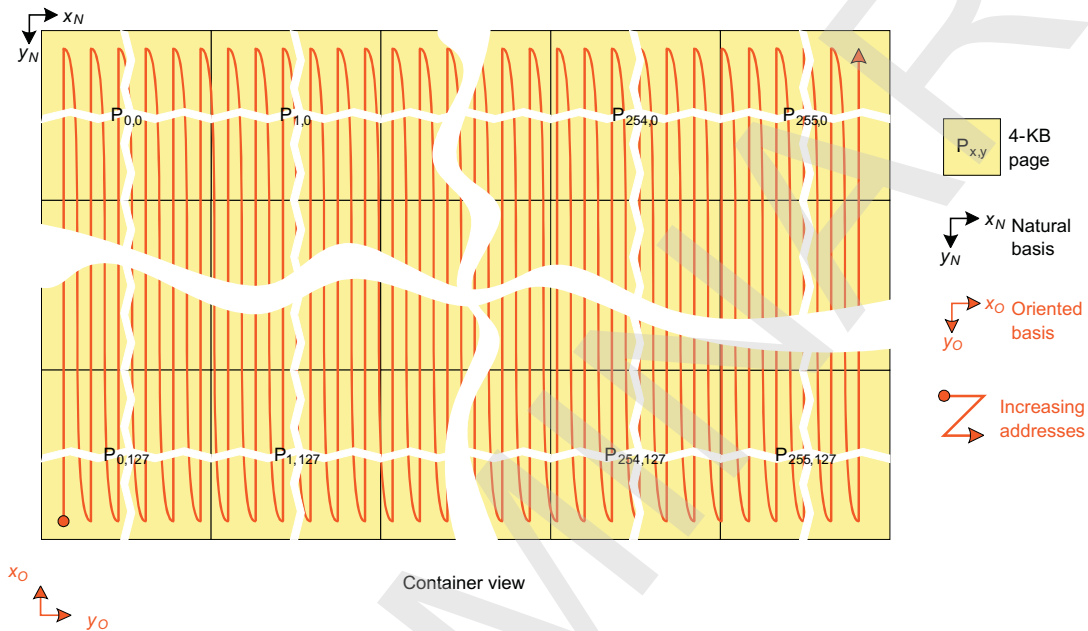
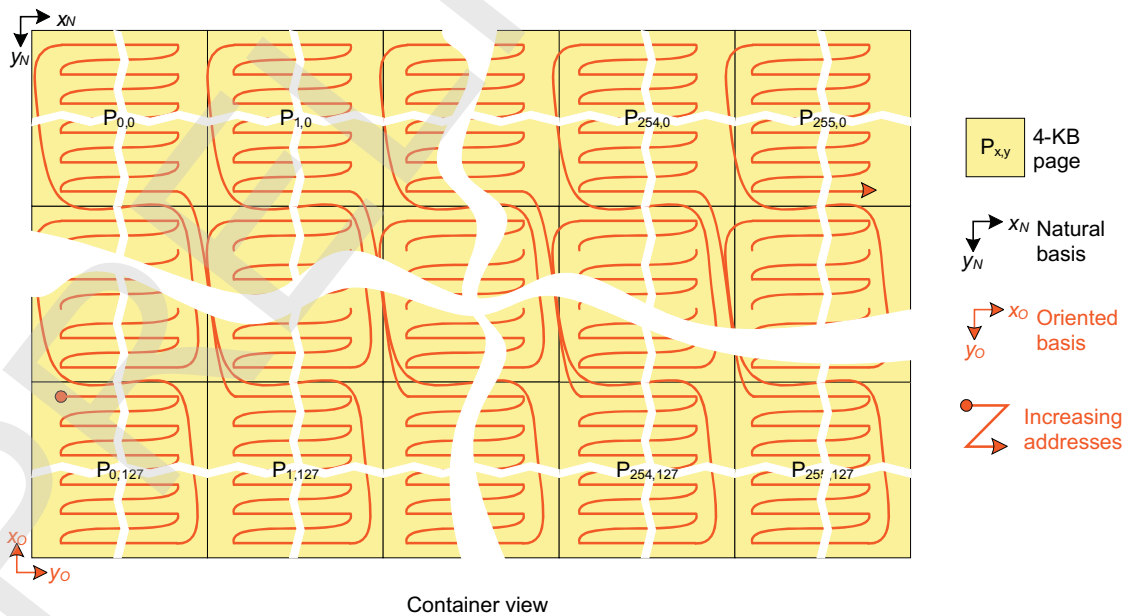


Figure 15-32. Page Mode Ordering of Elements in 90° View



**15.2.3.6.1.7.3.8 90° View with Horizontal Mirror or 270° View with Vertical Mirror (Orientation 7)**

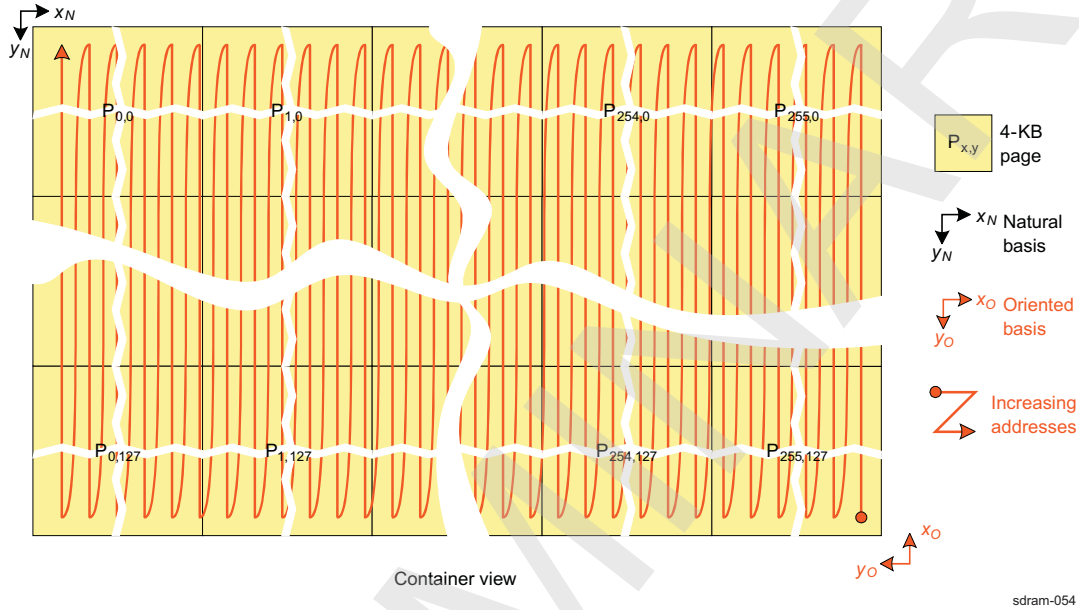
This orientation defined by  $S=1$ ,  $\bar{Y}=1$  and  $\bar{X}=1$  and means that the operated change of basis is:

$$\begin{cases} \bar{x}_O = -\bar{y}_N \\ \bar{y}_O = -\bar{x}_N \end{cases}$$

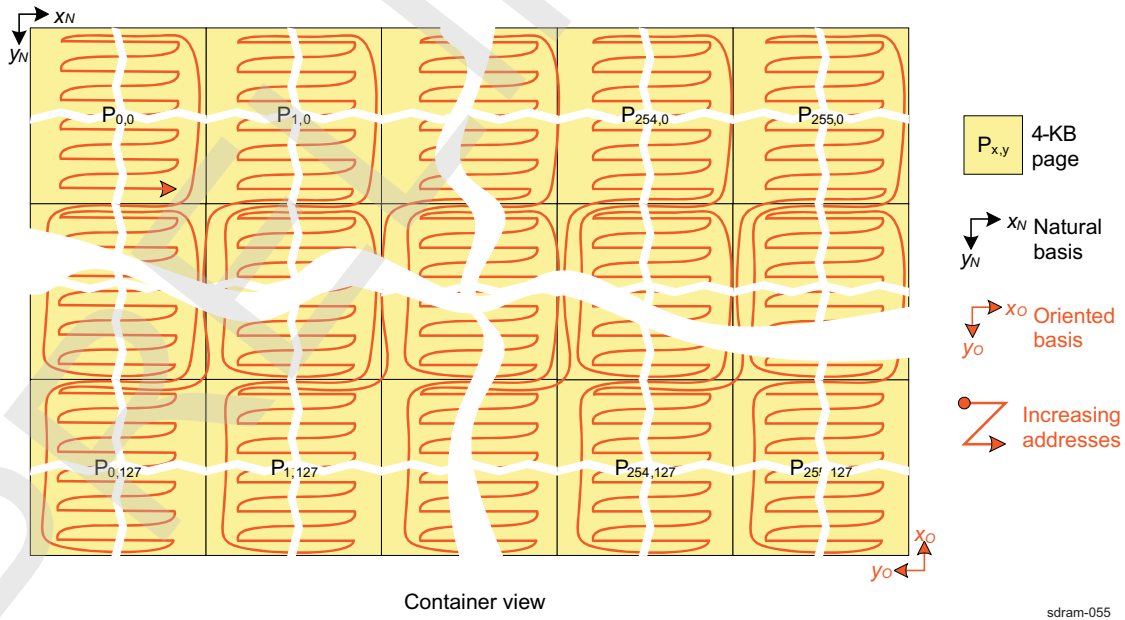
sdram-053

In any TILER mode, the elements are then ordered from bottom-to-top and then from right-to-left in their container, as shown in Figure 15-33 and Figure 15-34.

**Figure 15-33. Tiled Mode Ordering of Elements in 90° View with Horizontal Mirror**



**Figure 15-34. Page Mode Ordering of Elements in 90° View with Horizontal Mirror**



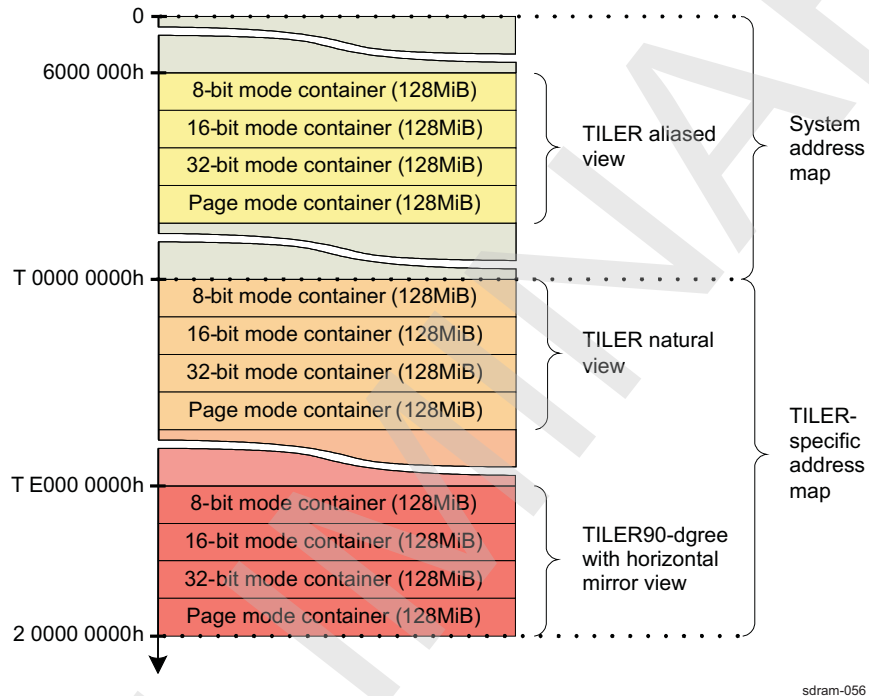
### 15.2.3.6.2 TILER Macro-Architecture

The TILER requires a 4-GiB addressing space to map its 128-MiB physical container in 4 modes and 8

orientations. As its addressing space fills along the 4-GiB global system address map, the TILER addressing space cannot enter as-is in the system address map. Besides, putting in place a register-based mechanism per initiator to specify the orientation of the following accesses and then reduce the TILER addressing space to a single 512-MiB view is not an option, given that most of the bandwidth-hungry initiators require simultaneous accesses to the TILER container in different views.

As a result 32 bits are not enough to address all these requests, since the TILER port must convey virtually-addressed requests to the "oriented" TILER containers, but also physically-addressed requests to the attached SDRAM controllers. A 33rd address bit is necessary to distinguish the two separated address map as illustrated in Figure 15-35.

Figure 15-35. TILER Port Address Map



Still, having separated system and TILER-specific address maps is not sufficient. In a system, many existing and external IP blocks still rely on a 32-bit address, and would then be limited to one or the other 4-GiB addressing space. To overcome this limitation, one 512-MiB view is aliased in the system address map as described in table and fig.

Table 15-13. TILER Aliased View in the L3 Interconnect Mapping

Start address (hex)	End address (hex)	Size
0x6000_0000	0x7FFF_FFFF	512MB

From all incoming requests, TILER requests are filtered as having an address that fits:

- The address format in the TILER-specific address map given in Table 15-14, or
- The address format of the aliased view in the system address map given in Table 15-15, other requests are forwarded directly to the SDRAM controller in a TILER bypass mode.

Table 15-14. Address Format in the TILER-Specific Address Map

32	31	30	29	28	27	26 ... 4	3 ... 0
T	Orientation			Mode		Virtual address	
1	S	Y	X	M1	M0	A26 ... A4	0

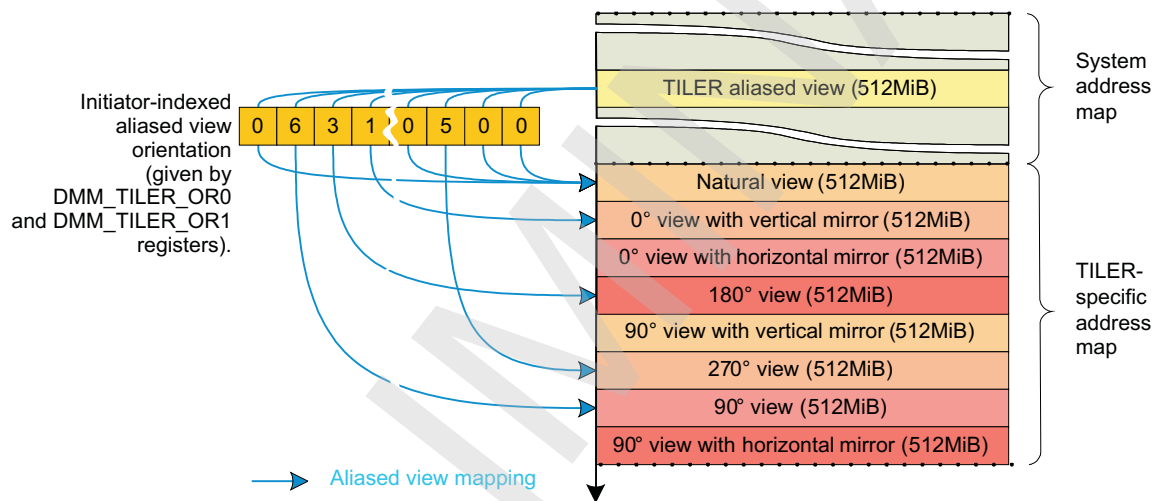
**Table 15-15. Address Format of the TILER Aliased View in the System Address Map**

32	31	30	29	28	27	26 ... 4	3 ... 0
T	TILER aliasing			Mode		Virtual address	
0	0	1	1	M1	M0	A26 ... A4	0

In these address formats:

- The 33rd bit, noted T, is aimed at distinguishing the standard 4-GiB system address map from the 4-GiB TILER-specific address map
- The orientation bits, noted S,  $\bar{Y}$  and  $\bar{X}$ , define the request orientation as specified in [Table 15-11](#)
- The mode bits, noted M1 and M0, define the request mode as specified in [Table 15-10](#)
- The remaining 27 bits, noted A0 to A26, define the mode and orientation specific virtual address as defined in [Figure 15-16](#), [Figure 15-17](#), and [Figure 15-18](#).

The orientation of the aliased TILER view is extracted from an initiator-indexed look-up table, as shown in [Figure 15-36](#).

**Figure 15-36. TILER Aliased View Orientation**

As illustrated in [Figure 15-36](#), an internal initiator-indexed lookup table is storing the current orientation of the aliased view for each initiator.

When an interconnect request hits the TILER aliased view in the system address map, the request initiator orientation is extracted from the lookup table and the request address is translated accordingly to this orientation - the T bit is set and bits [31:29] are replaced with the orientation.

Regarding the dimensioning of this lookup table, given that:

- Initiators accessing simultaneously multiple views use the TILER-specific address map,
- Many initiators either do not need to access any view or can be restricted to only accessing the natural view,

only a limited number of initiators are likely to dynamically modify the orientation of its aliased TILER view. Then, the orientation lookup table is limited to 16 entries. These 16 orientation entries are mapped on the two 32-bit [DMM\\_TILER\\_OR0](#) and [DMM\\_TILER\\_OR1](#) registers.

The first eight entries of the lookup table are mapped in the [DMM\\_TILER\\_OR0](#) register, and the last eight entries in the [DMM\\_TILER\\_OR1](#) register. Therefore, given an index x, the orientation related to this index is given in the ORx field.

Each of these two registers is split in eight 4-bit fields, each field mapping an entry of the lookup table with:

- A S,  $\bar{Y}$ ,  $\bar{X}$  orientation code on the 3 least significant bits
- A  $\bar{W}$  field-specific active-low local write enable bit, always read as 0, on the most significant bit.



The registers fields that correspond to initiators not needing any dynamic configuration of their aliased view orientation must be specified as reserved fields, and only written with zeros.

The role of the  $\bar{W}$  bit is to allow the modification of a single entry without requiring a read-modify-write sequence. This approach is then much friendlier:

- In a system where multiple initiators are capable of modifying their own fields in the registers
- With initiators unable to make the read-modify-write sequence such as DMA.

Still, the  $\bar{W}$  bit is active-low to keep the compatibility with the usual read-modify-write sequence. When reading an aliased view orientation register, as all its  $\bar{W}$  bits are read as 0, if these bits are untouched by the modification - as they should be -, writing back the modified register updates all orientation fields of the register.

### 15.2.3.6.3 TILER Guidelines for Initiators

#### 15.2.3.6.3.1 Buffered Raster-Based Initiators

##### 15.2.3.6.3.1.1 Buffer Size

The necessary minimum buffer size depends on the initiator support in terms of:

- Element size
- Orientation
- Maximum number of elements per line in all supported element sizes and orientations and on the SDRAM memory pre-fetch size.

Let  $N$  be the maximum number of elements per line for a given mode - element size - and orientation,  $P$  be the SDRAM memory pre-fetch size in bytes, and  $max$  be the function returning the maximum of the two parameters, the minimum necessary line buffer size to handle all resolutions in a given mode and orientation is given in [Table 15-16](#).

**Table 15-16. Minimum Buffer Size to Efficiently Handle Lines of up to  $N$  Elements**

	0° or 180° orientation (S=0)	90° or 270° orientation (S=1)
8-bit mode	4 x N bytes	$max(4, P/4) \times N$ bytes
16-bit mode	None	$max(8, P/2) \times N$ bytes
32-bit mode	None	$max(8, P/2) \times N$ bytes

This minimal buffer size can only be reached with an advanced FIFO management scheme. The standard ping-pong buffer requires twice this buffer size.

**NOTE:** Given their nature, field accesses to an interlaced frame-buffer require buffers twice as small as in the standard progressive case.

For instance, an initiator that shall handle only a single progressive frame-buffer in any orientation of:

- Up to 1920x1080 in YUV 4:2:0
  - Up to 1600x1200 in 16-bit RGB565
  - Up to 800x600 in 32-bit ARGB
- requires a line buffer of at least 15360 bytes since:
- $4 \times 1920 + 4 \times 2960 = 15360$  bytes are required for a 3-plane YUV 4:2:0 frame of 1920x1080
  - $4 \times 1920 + 8 \times 960 = 15360$  bytes are required for a 2-plane YUV 4:2:0 frame of 1920x1080
  - $8 \times 1600 = 12800$  bytes are required for a 16-bit RGB565 frame of 1600x1200
  - $8 \times 800 = 6400$  bytes are required for a 32-bit ARGB frame of 800x600
- and a line buffer of at least 30720 bytes when using a simple ping-pong buffer scheme.

### 15.2.3.6.3.1.2 Performance

Table 15-17 gives the ratio of effective data to transferred data at the external memory interface for these buffered raster-based initiators in each mode and orientation, when using an up-to-32-bit DDR, lpDDR or lpDDR2 SDRAM memory - or an up-to-16-bit DDR2 memory.

**Table 15-17. Memory Data Payload for Buffered Raster-Based Initiators on up-to-32-bit DDR, lpDDR, or lpDDR2**

	0° or 180° Orientation (S = 0)		90° or 270° Orientation (S = 1)	
	Progressive	Interlaced	Progressive	Interlaced
8-bit mode	100%	100%	100%	50%
16-bit mode	100%	100%	100%	50%
32-bit mode	100%	100%	100%	50%
Page mode	100%	N/A	100%	100%

When using 32-bit DDR, lpDDR or lpDDR2 SDRAM memories - or even 16-bit DDR, DDR2, lpDDR or lpDDR2 memories -, whenever all previous guidelines are fulfilled, then there is no penalty of any kind for these initiators to access a tiled object in any orientation, except for accesses to an interlaced frame in a 90° or 270° orientation where the memory bandwidth is twice the requested bandwidth.

When using up-to-32-bit DDR, lpDDR, lpDDR2 - or also up-to-16-bit DDR2 - SDRAM memories, all accesses, except interlaced accesses in a 90° or 270° orientation, are equally efficient and offer the full possible memory bandwidth.

Given its progressive nature, the page mode does not introduce any performance hit.

Table 15-18 gives the ratio of effective data to transferred data at the external memory interface for these buffered raster-based initiators in each mode and orientation, when using a 32-bit DDR2 or DDR3 memory.

**Table 15-18. Memory Data Payload for Buffered Raster-Based Initiators on 32-bit DDR2 or DDR3**

	0° or 180° Orientation (S = 0)		90° or 270° Orientation (S = 1)	
	Progressive	Interlaced	Progressive	Interlaced
8-bit mode	100%	50%	100%	50%
16-bit mode	100%	50%	100%	50%
32-bit mode	100%	50%	100%	50%
Page mode	100%	N/A	100%	100%

When using 32-bit DDR2 or DDR3 SDRAM memories, whenever all previous guidelines are fulfilled, then there is no penalty of any kind for these initiators to access a tiled object in any orientation, except for accesses to an interlaced frame in any orientation where the memory bandwidth is twice the requested bandwidth.

When using DDR2 or DDR3 SDRAM memories, all accesses, except interlaced accesses, are equally efficient and offer the full possible memory bandwidth.

Similarly, the page mode does not introduce any performance hit.

## 15.2.4 DMM Use Cases and Tips

### 15.2.4.1 PAT Use Cases

There are five ways of using PAT:

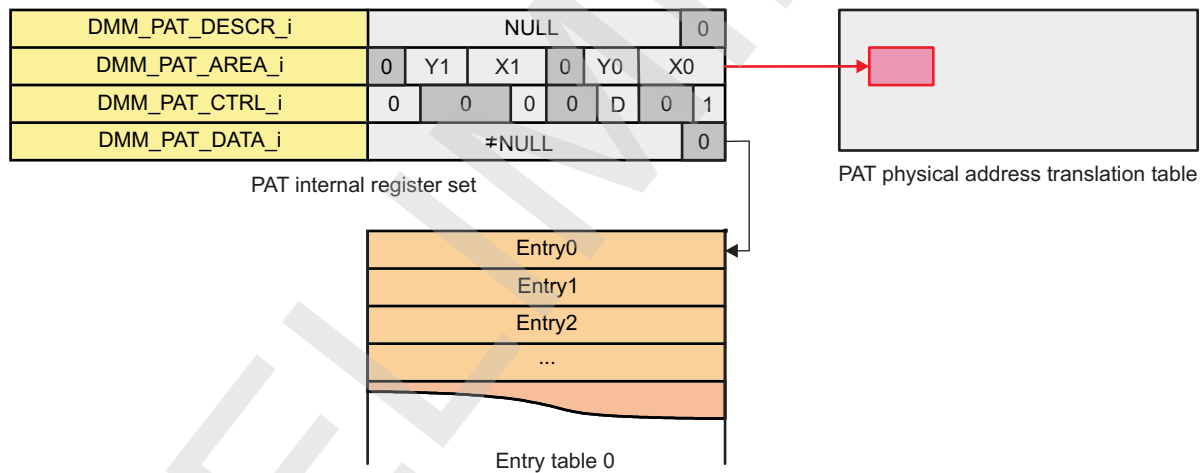
- Simple manual area refill
- Single auto-configured area refill
- Chained auto-configured area refill
- Synchronised auto-configured area refill
- Cyclic synchronised auto-configured area refill

#### 15.2.4.1.1 Simple Manual Area Refill

These steps have to be performed to create the 16-byte aligned memory-mapped entry table containing all entries of the defined area (see Figure 15-37):

- Write the `DMM_PAT_AREA_i` register with the relevant (x0, y0) (x1, y1) area definition
- Write the `DMM_PAT_DATA_i` register with the physical address of the created entry table
- Write the `DMM_PAT_CTRL_i` register with the requested refill direction and assert the `DMM_PAT_CTRL_i[0]` START bit
- The refill is done when the `DMM_PAT_STATUS_i[3]` DONE bit is set
- A new refill can be initiated when the `DMM_PAT_STATUS_i[0]` READY bit is set

Figure 15-37. Simple Manual Area Refill Scheme

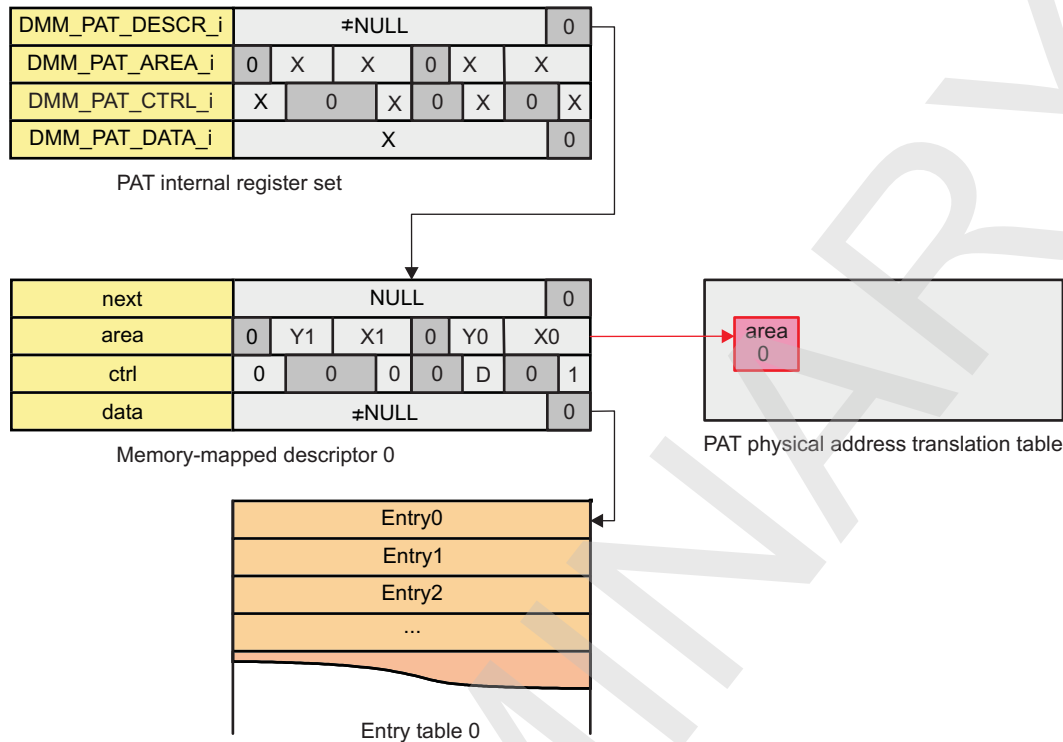


sdram-067

#### 15.2.4.1.2 Single Auto-Configured Area Refill

These steps have to be performed to create the 16-byte aligned memory-mapped entry table containing all entries of the defined area (see Figure 15-38):

- Create a 16-byte aligned memory-mapped descriptor structure where:
  - the next field is set to NULL
  - the area field is set with the relevant (x0, y0) (x1, y1) area definition
  - the ctrl field is set with the requested direction D and the start bit asserted to start refilling as soon as this descriptor enters the PAT refill engine
  - the data field is set to the physical address of the created entry table
- The `DMM_PAT_DESCR_i` register with the physical address of the created descriptor
- The refill is done when the `DMM_PAT_STATUS_i[3]` DONE bit is set
- A new refill can be initiated when the `DMM_PAT_STATUS_i[0]` READY bit is set

**Figure 15-38. Single Auto-Configured Area Refill Scheme**

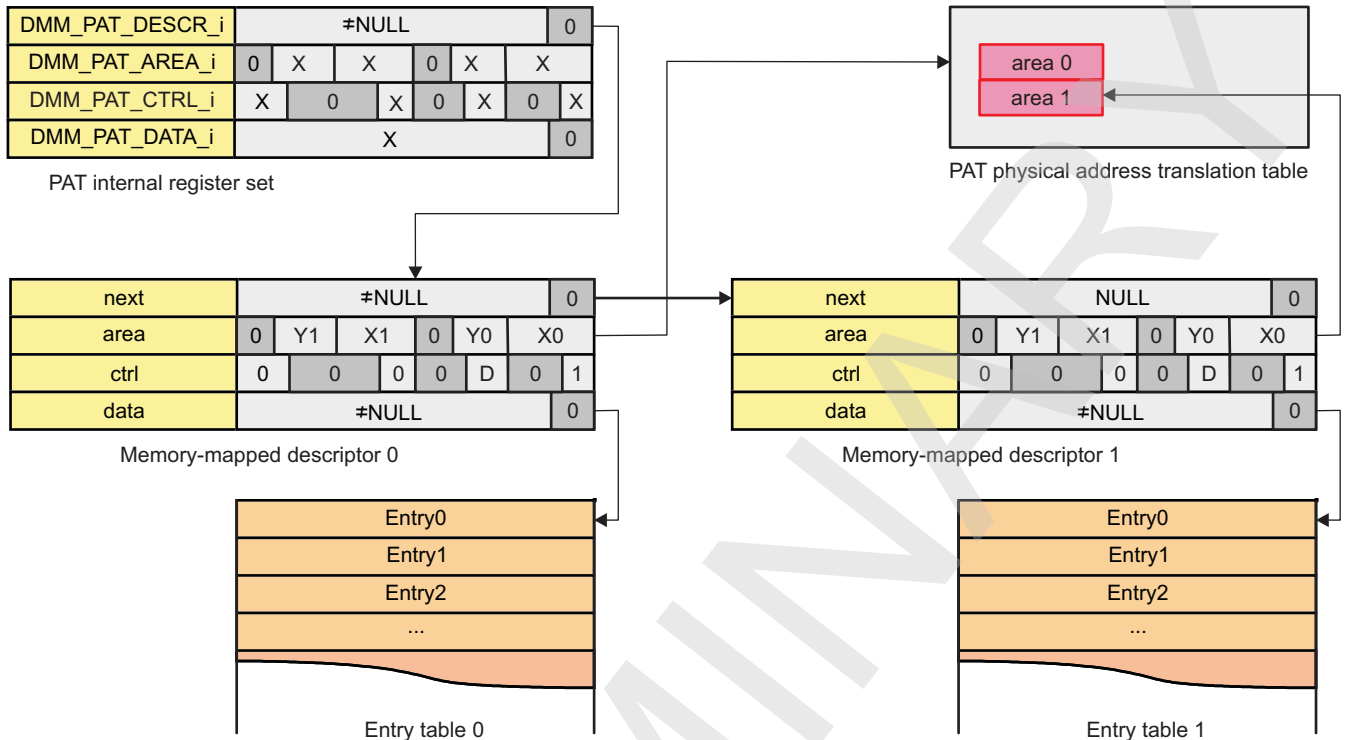
sdram-068

### 15.2.4.1.3 Chained Auto-Configured Area Refill

These steps have to be performed to create one 16-byte aligned memory-mapped entry table per area containing the entries for the corresponding area (see [Figure 15-39](#)):

- Create one 16-byte aligned memory-mapped descriptor structures per area where:
  - the next field is set to the physical address of the next descriptor or NULL for the last one
  - the area field is set with the relevant (x0, y0) (x1, y1) area definition
  - the ctrl field is set with the requested direction D and the START bit asserted to start refilling as soon as the previous area refill is done
  - the data field is set to the physical address of the corresponding entry table
- Write the `DMM_PAT_DESCR_i` register with the physical address of the first created descriptor
- Each area refill is done when the `DMM_PAT_STATUS_i[3]` DONE bit is set
- All area refills are done when the `DMM_PAT_STATUS_i[0]` READY bit is set
- A new refill can be initiated when the `DMM_PAT_STATUS_i[0]` READY bit is set

Figure 15-39. Chained Auto-Configured Area Refill Scheme

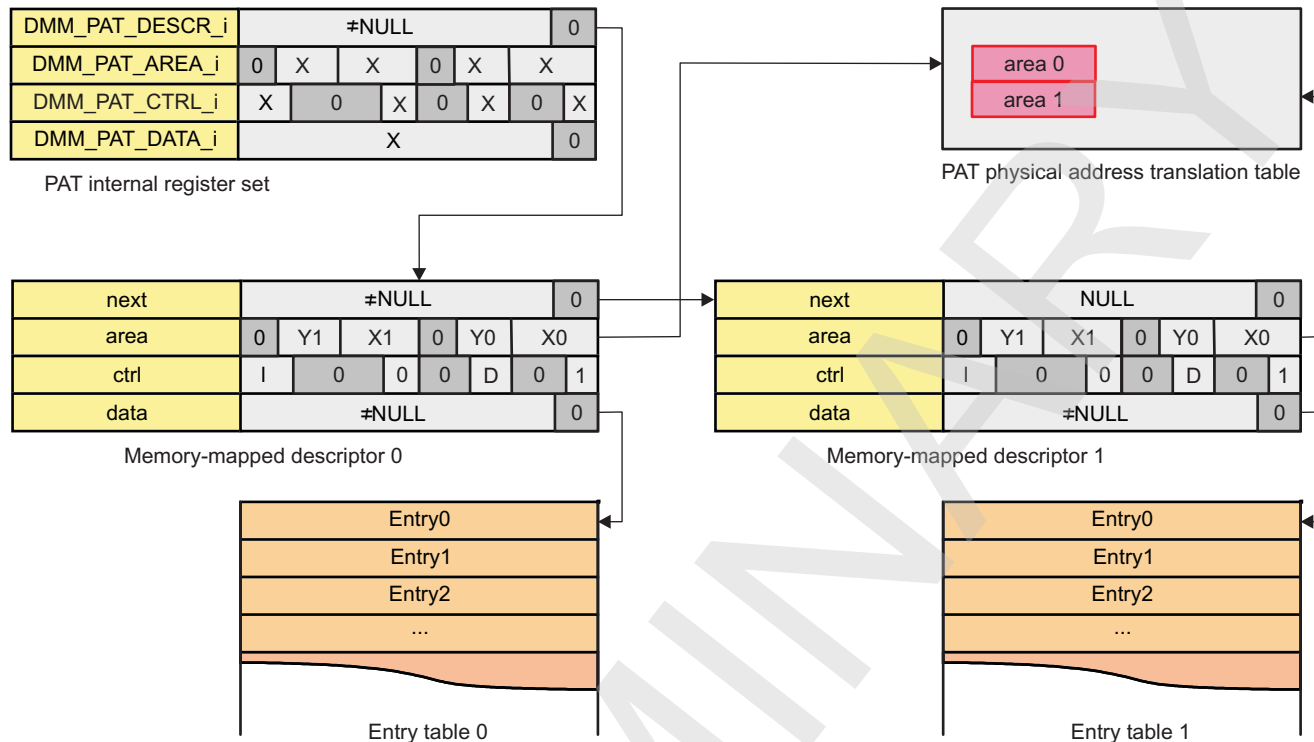


sdram-069

#### 15.2.4.1.4 Synchronized Auto-Configured Area Refill

These steps have to be performed to create one 16-byte aligned memory-mapped entry table per area containing the entries for the corresponding area (see Figure 15-40):

- Create one 16-byte aligned memory-mapped descriptor structures per area where:
  - the `next` field is set to the physical address of the next descriptor or NULL for the last one
  - the `area` field is set with the relevant (x0, y0) (x1, y1) area definition
  - the `ctrl` field is set with the synchronising initiator identifier I, the SYNC bit asserted, the requested direction D, and the START bit asserted to start refilling as soon as the previous area refill is done and initiator I has made one access in the previous area
  - the `data` field is set to the physical address of the corresponding entry table
- Write the `DMM_PAT_DESCR_i` register with the physical address of the first created descriptor
- Each area refill is done when the `DMM_PAT_STATUS_i[3]` DONE bit is set
- All area refills are done when the `DMM_PAT_STATUS_i[0]` READY bit is set
- A new refill can be initiated when the `DMM_PAT_STATUS_i[0]` READY bit is set

**Figure 15-40. Synchronised Auto-Configured Area Refill Scheme**

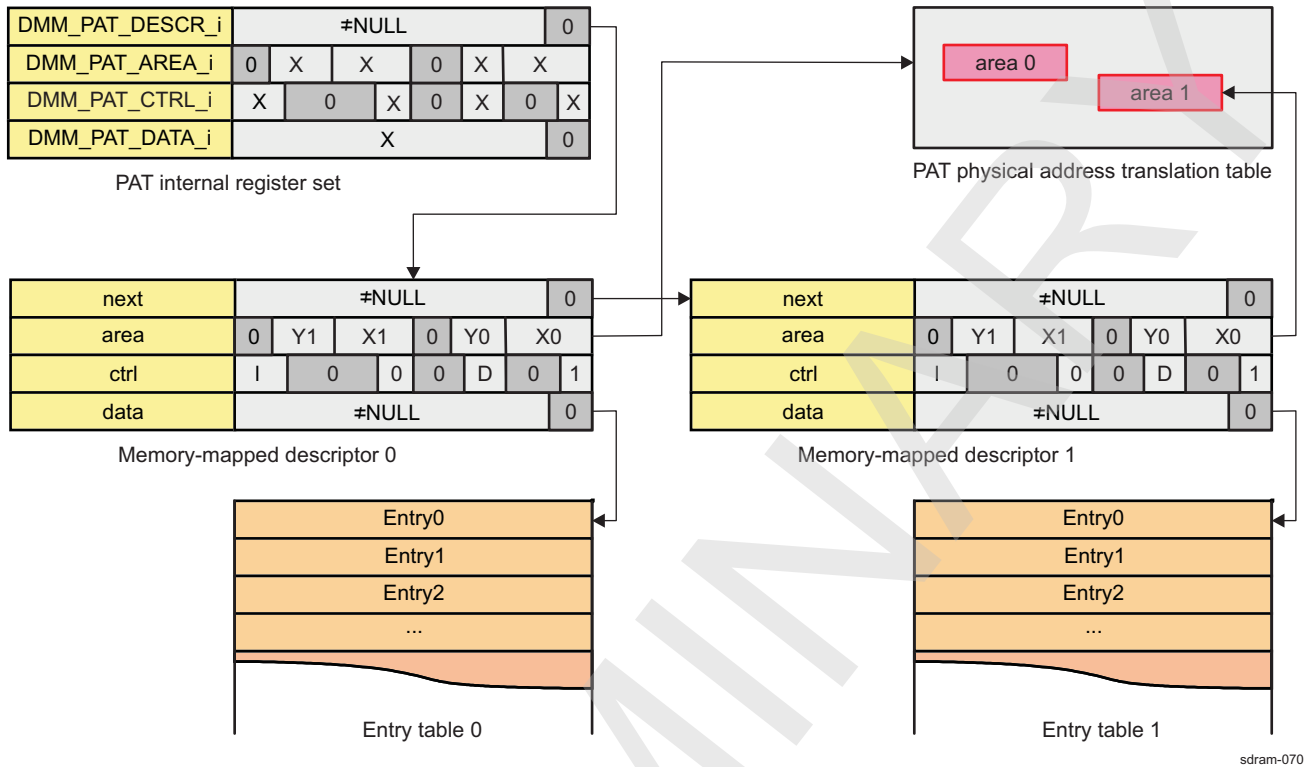
sdram-071

**15.2.4.1.5 Cyclic Synchronized Auto-Configured Area Refill**

These steps have to be performed to create one 16-byte aligned memory-mapped entry table per area containing the entries for the corresponding area (see Figure 15-41):

- Create one 16-byte aligned memory-mapped descriptor structures per area where:
  - the next field is set to the physical address of the next descriptor in the circular list
  - the area field is set with the relevant (x0, y0) (x1, y1) area definition
  - the ctrl field is set with the synchronising initiator identifier I, the SYNC bit asserted, the requested direction D, and the START bit asserted to start refilling as soon as the previous area refill is done and initiator I has made one access in the previous area
  - the data field is set to the physical address of the corresponding entry table
- Write the `DMM_PAT_DESCR_i` register with the physical address of the initial descriptor
- Each area refill is done when the `DMM_PAT_STATUS_i[3]` DONE bit is set
- A new refill can be initiated by writing any value to the `DMM_PAT_DESCR_i` to abort the current one

Figure 15-41. Cyclic Synchronized Auto-Configured Area Refill Scheme



sdr-am-070

**NOTE:** Never use circular lists of descriptors where all descriptors have the `DMM_PAT_CTRL_i[0]` START bit set and no synchronisation. This leads to an endless continuous refill.

### 15.2.4.2 Addressing Management with LISA

In the device, we have: two memory controllers (EMIFs) and one address space per memory controller.

#### 15.2.4.2.1 Case 1: Use of Only One Memory Controller

In this example, we assume that we have 1GB of external memory evenly spread onto two address spaces. We also want the address range for address space 0 to start at offset 0x4000\_0000.

Table 15-19. Address Definition

Address Range	Memory Controller	Memory Controller Address Range
0x8000_0000 to 0x9FFF_FFFF	0x1	0x0000_0000 to 0x0FFF_FFFF
0xA000_0000 to 0xBFFF_FFFF	0x0	0x4000_0000 to 0x5FFF_FFFF

This configuration requires two non-overlapping sections to be set. They may be defined in any order since there is no concern with priority in this case, see Table 15-20:

Table 15-20. Configuration

Bit Field	Section 0 (DMM_LISA_MAP_0)	Section 1 (DMM_LISA_MAP_1)
[31:24] SYS_ADDR	0x80	0xA0
[22:20] SYS_SIZE	0x5	0x5
[19:18] SDRC_INTL	0x0 (not applicable)	0x0 (not applicable)
[17:16] SDRC_ADDRSPC	0x1	0x0
[8] SDRC_MAP	0x1	0x1

**Table 15-20. Configuration (continued)**

Bit Field	Section 0 (DMM_LISA_MAP_0)	Section 1 (DMM_LISA_MAP_1)
[7:0] SDRC_ADDR	0x00	0x40

To check if an address hits a section, use the eight upper address bits of the address and mask them with the hit mask:  $28-2^{\text{SYS\_SIZE}}$ . If the result is equal to SYS\_ADDR, the section is hit.

To define the physical address to be issued to the memory controller, use the eight upper address of the system address, mask them with the address mask:  $2^{\text{SYS\_SIZE}}-1$ , and OR them with SDRC\_ADDR. This will give you the resulting eight upper physical address bits. All lower address bits are forwarded unchanged.

Request to address 0x99AE\_37F0:

- Upper address bits: 0x99
- Hit mask:  $28-25 = 0xE0$
- Masked upper address bits: 0x80 -> hits section 0
- Address mask:  $25-1 = 0x1F$
- Masked upper address bits: 0x19
- OR with SDRC\_ADDR: 0x19
- Physical address: 0x19AE\_37F0

This request will be forwarded to address 0x19AE\_37F0, address space 1 of the memory controller.

Request to address 0xB7FF\_0340:

- Upper address bits: 0xB7
- Hit mask:  $28-25 = 0xE0$
- Masked upper address bits: 0xA0 -> hits section 1
- Address mask:  $25-1 = 0x1F$
- Masked upper address bits: 0x17
- OR with SDRC\_ADDR: 0x57
- Physical address: 0x57FF\_0340

This request will be forwarded to address 0x57FF\_0340, address space 0 of the memory controller.

#### 15.2.4.2.2 Case 2: Use of the 2 Memory Controllers

In the case of 512MB interleaved at 256B boundaries and 256MB non-interleaved on second memory controller, we have this use case:

**Table 15-21. Address Definition**

Address Range	Memory Controller	Memory Controller Address Space	Memory Controller Address Range
0x8000_0000 to 0x9FFF_FFFF	1 and 2, interleaved at 256 bytes boundaries	0 on both controllers	0x0000_0000 to 0x0FFF_FFFF on both controllers
0xA000_0000 to 0xAFFF_FFFF	2 only	0	0x1000_0000 to 0x1FFF_FFFF

Two sections are used to map such a configuration, see [Table 15-22](#):

**Table 15-22. Configuration**

Bit Field	Section 0 (DMM_LISA_MAP_0)	Section 1 (DMM_LISA_MAP_1)
[31:24] SYS_ADDR	0x80	0xA0
[22:20] SYS_SIZE	0x5	0x4
[19:18] SDRC_INTL	0x2	0x0 (not applicable)
[17:16] SDRC_ADDRSPC	0x0	0x0
[8] SDRC_MAP	0x3	0x2
[7:0] SDRC_ADDR	0x00	0x10



Detecting a section hit is the same for the interleaved case as for the single controller case: to check if an address hits a section, use the eight upper address bits of the address and mask them with the hit mask:  $28-2^{\text{SYS\_SIZE}}$ . If the result is equal to SYS\_ADDR, the section is hit.

The physical address generation is modified for the interleaved case though. In case of a 256 bytes interleaving, the first chunk of 256 bytes is mapped to the first controller, the second chunk to the second controller, and so on. This results in system address bit 8 to be decoded as the controller indicator, 0 for the first controller and 1 for the second one (system address bit 7 would be used for interleaving at 128 bytes boundary, and bit 9 for 512 bytes). This bit is not included in the computed physical address, meaning the upper system address bits 9 to 31 are shifted to bits 8 to 30 when generating the physical address.

The rest of the address generation is handled the same as for the single controller case: to define the physical address to be issued to the memory controller, use the eight upper address of the system address, mask them with the address mask:  $2^{\text{SYS\_SIZE}}-1$ , and OR them with SDRG\_ADDR. This will give you the resulting eight upper physical address bits. All lower address bits are forwarded unchanged.

Request to address 0x99AE\_37F0:

- Upper address bits: 0x99
- Hit mask: 28-25 = 0xE0
- Masked upper address bits: 0x80 -> hits section 0
- Address mask: 25-1 = 0x1F
- Masked upper address bits: 0x19
- Full masked address: 0x19AE\_37F0
- Bit 8 is 1 -> targets the second memory controller
- Full masked shifted address (suppressing bit 8): 0x0CD7\_1BF0 (see [Section 15.2.4.2.2.1](#))
- OR upper physical address bits with SDRG\_ADDR: 0x0CD7\_1BF0
- Physical address: 0x0CD7\_1BF0

This request will be forwarded to address 0x0CD7\_1BF0, address space 0, of the second memory controller.

**15.2.4.2.2.1 Address Upper Bits Shifting****Table 15-23. Address Upper Bits Shifting**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
<b>Address Before Interleaving</b>																																
0	0	0	1	1	0	0	1	1	0	1	0	1	1	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0
<b>Address After Interleaving</b>																																
0	0	0	0	1	1	0	0	1	1	0	1	0	1	1	1	0	0	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0

## 15.2.5 DMM Basic Programming Model

The programming model section:

- Describes how objects can be addressed in all TILER modes and orientations
- Explains how the physical containers and physical address translation (PAT) look-up table (LUT) can be shared between different modes
- Does not give an exhaustive description of all TILER and DMM registers, because these are described in [Section 15.2.3.5, DMM](#) and [Section 15.2.3.6, TILER](#).

### 15.2.5.1 Global Initialization

This section identifies the requirements of initializing the surrounding modules when the DMM module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DMM.

**Table 15-24. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
Power, Reset and Clock Management (PRCM)	The module interface and functional clocks must be enabled. See <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Dual Cortex™ - M3 microprocessor unit (MPU)	Cortex-M3 interrupt controller (INTC) configuration must be done to enable the interrupts from DMM module. See <a href="#">Chapter 17, Interrupt Controllers</a> .
Dual Cortex™ - A9 MPU	Cortex-A9 INTC configuration must be done to enable the interrupts from DMM module. See, <a href="#">Chapter 17, Interrupt Controllers</a> .
L3 Interconnect	Data Interface

### 15.2.5.2 DMM Module Global Initialization

This procedure initializes the DMM after a power-on reset.

**Table 15-25. DMM Global Initialization**

Step	Register/ Bit Field / Programming Model	Value
Configure the DMM for Smart-Idle power management mode	<a href="#">DMM_SYSCONFIG</a> [3:2] IDLE_MODE	0x2

### 15.2.5.3 DMM Operational Modes Configuration

#### 15.2.5.3.1 Different Operational Modes

The TILER can be virtually accessed in four different modes, namely 8-bit, 16-bit, 32-bit and page modes. Each mode defines the element granularity to apply isometric transforms, as summarized in [Table 15-28](#)

**Table 15-26. Coding and Description of TILER Modes**

Mode	Name	Granularity (element size)
0	8 - bit tiled mode	8 bits
1	16 - bit tiled mode	16 bits
2	32 - bit tiled mode	32 bits
3	Page mode	4096 bytes

#### 15.2.5.3.2 Configuration Settings and LUT Refill

**Table 15-27. Configuration Settings and LUT Refill**

Step	Register/Bit Field/Programming Model	Value
PAT configuration refill Engine 0	<a href="#">DMM_PAT_CONFIG</a> [0] MODE0	xxx
PAT configuration refill Engine 1	<a href="#">DMM_PAT_CONFIG</a> [1] MODE1	xxx

**Table 15-27. Configuration Settings and LUT Refill (continued)**

Step	Register/Bit Field/Programming Model	Value
Set DMM PAT View register value for each initiator	DMM_PAT_VIEW0 Vx DMM_PAT_VIEW1 Vx	xxx
Set DMM PAT Write enable for the initiators	DMM_PAT_VIEW0 Wx DMM_PAT_VIEW1 Wx	xxx
Choosing container type and access method	DMM_PAT_VIEW_MAP_i	xxx
Define the base address of all view mappings	DMM_PAT_VIEW_MAP_BASE[31] BASE_ADDR	xxx
Set area definition for DMM physical address translator	DMM_PAT_AREA_i	xxx
Set physical address of the current table refill entry data or entry data when in manual mode	DMM_PAT_DATA_i[31:4] ADDR	xxx
Define the direction of this PAT table refill (S Y X), different from 0x011	DMM_PAT_CTRL_i[6:4] DIRECTION	xxx
Start a PAT table refill	DMM_PAT_CTRL_i[0] START	0x1

### 15.2.5.3.3 Interleaving Settings

**Table 15-28. Interleaving Settings**

Step	Register/Bit Field/Programming Model	Value
Set DMM system section address MSB	DMM_LISA_MAP_i[31:24] SYS_ADDR	xxx
Set DMM system section size	DMM_LISA_MAP_i[22:20] SYS_SIZE	xxx
Set SDRAM controller interleaving mode	DMM_LISA_MAP_i[19:18] SDRC_INTL	xxx
Set SDRAM controller address space	DMM_LISA_MAP_i[17:16] SDRC_ADDRSPC	xxx
Set SDRAM controller mapping	DMM_LISA_MAP_i[9:8] SDRC_MAP	xxx
Set SDRAM controller address MSB	DMM_LISA_MAP_i[7:0] SDRC_ADDR	xxx
Enable/disable DMM memory mapping lock	DMM_LISA_LOCK[0] LOCK	xxx

### 15.2.5.3.4 Aliased Tiled View Orientation Settings and LUT Refill

**Table 15-29. Aliased Tiled View Orientation Settings and LUT Refill**

Step	Register/Bit Field/Programming Model	Value
Set DMM TILER orientation for each initiator	DMM_TILER_OR0 ORx DMM_TILER_OR1 ORx	xxx
Set DMM TILER write enable for the initiators	DMM_TILER_OR0 Wx DMM_TILER_OR1 Wx	xxx
Define the base address of all view mappings	DMM_PAT_VIEW_MAP_BASE[31] BASE_ADDR	xxx
DMM PAT initiator for synchronisation	DMM_PAT_CTRL_i[31:28] INITIATOR	xxx
Set DMM PAT table reload synchronisation	DMM_PAT_CTRL_i[16] SYNC	xxx
Set DMM PAT LUT index	DMM_PAT_CTRL_i[9:8] LUT_ID	xxx
Define the direction of this PAT table refill (S Y X)	DMM_PAT_CTRL_i[6:4] DIRECTION	xxx
Start a PAT table refill	DMM_PAT_CTRL_i[0] START	0x1

### 15.2.5.3.5 Priority Settings

**Table 15-30. Priority Settings**

Step	Register/Bit Field/Programming Model	Value
Set priority for each initiator	DMM_PEG_PRIO_k Px	xxx
Set write enable for P_PAT field	DMM_PEG_PRIO_PAT[4]W_PAT	xxx
Set priority for PAT engine	DMM_PEG_PRIO_PAT[2:0]W_PAT	xxx

15.2.5.3.6 Error Handling

Table 15-31. Error Handling

Step	Register/Bit Field/Programming Model	Value
Enable interrupt for selected type of error	DMM_PAT_IRQENABLE_SET[15:9] DMM_PAT_IRQENABLE_SET[7:0]	xxx
when interrupt occurs		
Disable type of error to handle	DMM_PAT_IRQENABLE_CLR[15:9] DMM_PAT_IRQENABLE_CLR[7:0]	xxx
Check error status	DMM_PAT_IRQSTATUS[15:9] DMM_PAT_IRQSTATUS[7:0]	xxx

15.2.5.3.7 PAT Programming Model

The purpose of the PAT is to map the tiled data anywhere in the 4GB physical address range, with a PAGE granularity. (The TILER page is the granularity of physical memory allocation in TILER container. Each page is 4kiB).

A PAT view is defining the kind of physical address translation to perform for each of the page, 8-bit, 16-bit and 32-bit mode accesses. Each mode in each PAT view can be programmed in 2 different modes (direct translation and indirect translation).

15.2.5.3.7.1 PAT in Direct Translation Mode

In this mode, the PAT performs a translation of the 128MB virtual container as a whole in the physical address space (that is, in the SDRAM). This mode is used only for debug or in case of a DMM without PAT module.

15.2.5.3.7.2 PAT in Indirect Translation Mode

This is the most commonly mode used. In this mode, the PAT performs a translation of each 4kB page individually. In this way the 128MB virtual address space can be scattered in the whole 2GB of the physical address space. This is achieved by using a 32678-word look-up table (LUT) which converts each page index (32768 possible values) into 19 address bits representing this page address in the physical memory. The main characteristic of this mode is: no constraint on the use of the physical memory except that it is using a multiple of 4kB areas located at 4kB boundaries in the physical memory.

In this mode the translation vector is found in the internal 32-k entry physical address translation vector table at the index given by bits 12 to 26 of the input virtual address, and the CONT\_x = 0.

Programming sequence:

- Set in DMM\_PAT\_VIEW0 (or DMM\_PAT\_VIEW1 depending on the L3 CONN\_ID of the initiator which is supposed to perform the tiled accesses) with the appropriate value. For example, if the L3 CONN\_ID equals 0, then the register DMM\_PAT\_VIEW0 should be programmed with the value 0x0000\_0001 (PAT view 1 selected for initiator 0)
- DMM\_PAT\_VIEW\_MAP\_BASE[31] = 0x8000\_0000 (should always be programmed at 1 to select the upper 2GB of the physical address space which corresponds to the external memory)
- set all 16bit tiled accesses in indirect translation mode by setting at least the field ACCESS\_16 at 1 (DMM\_PAT\_VIEW\_MAP\_1 = 0x0000\_8000), program DMM\_PAT\_VIEW\_MAP\_1 at 0x8080\_8080.

15.2.5.4 Addressing an Object in Tiled Mode

The purpose of this sub-section is to describe how a frame-buffer with a top-left pixel located at (x0,y0) in the natural view container and a bottom right pixel located at (x1,y1) in the natural view container is addressed in any tiled mode and orientation.

### 15.2.5.4.1 Frame Buffer Addressing

**NOTE:** In this sub-section, the (x,y) coordinates are given in pixel units.

Given a (S,  $\bar{Y}$ ,  $\bar{X}$ ) orientation, the (x,y) coordinates of the first pixel of the frame in the oriented view (with W the width and H the height in pixels of the container in the considered frame mode) is:

(x,y) = (x<sub>or</sub>, y<sub>or</sub>) when S, (y<sub>or</sub>, x<sub>or</sub>) otherwise  
with:

x<sub>or</sub> = x<sub>0</sub> when  $\bar{X}$ , W-1-x<sub>1</sub> otherwise  
y<sub>or</sub> = y<sub>0</sub> when  $\bar{Y}$ , H-1-y<sub>1</sub> otherwise

Given a (M<sub>1</sub>, M<sub>0</sub>) TILER mode where M<sub>1</sub> # 0 or M<sub>0</sub> # 0, the size in bytes of a pixel is 2<sup>2M<sub>1</sub>+M<sub>0</sub></sup>.

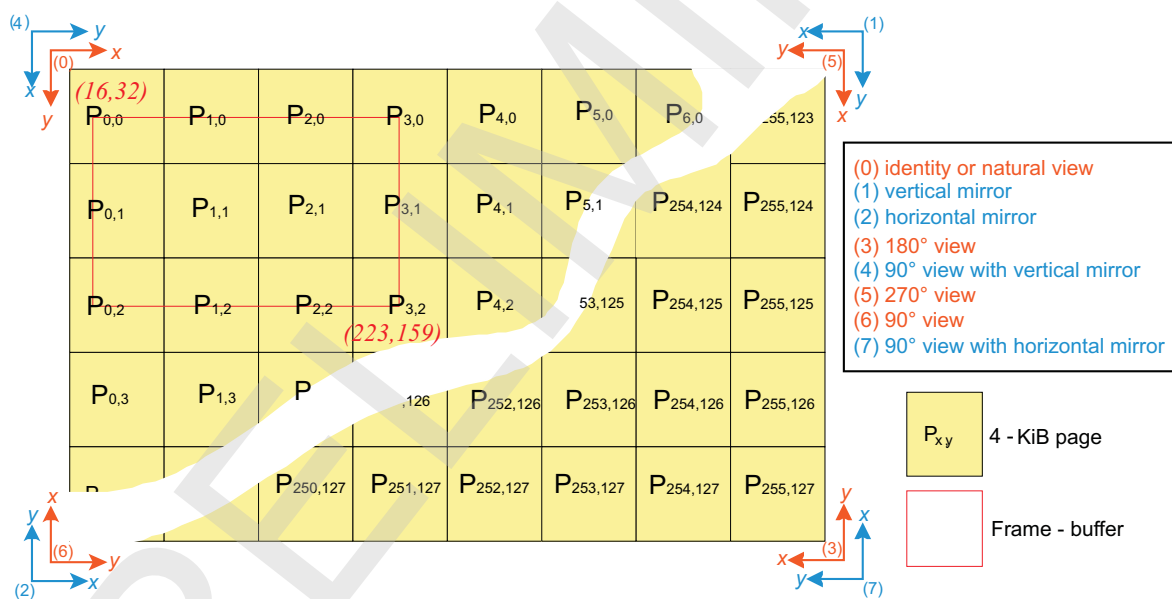
Therefore, given a (S,  $\bar{Y}$ ,  $\bar{X}$ ) orientation and a (M<sub>1</sub>, M<sub>0</sub>) TILER mode where M<sub>1</sub> # 0 or M<sub>0</sub> # 0, the byte offset of the base address of the considered oriented frame in its container is:

base\_address(x<sub>0</sub>, y<sub>0</sub>, (x<sub>1</sub>, y<sub>1</sub>), (S,  $\bar{Y}$ ,  $\bar{X}$ ), (M<sub>1</sub>, M<sub>0</sub>)) = (y<sub>or</sub>W + x<sub>or</sub>) P<sub>5</sub> when S, (x<sub>or</sub>H + y<sub>or</sub>) P<sub>5</sub> otherwise  
with:

x<sub>or</sub> = x<sub>0</sub> when  $\bar{X}$ , W-1-x<sub>1</sub> otherwise  
y<sub>or</sub> = y<sub>0</sub> when  $\bar{Y}$ , H-1-y<sub>1</sub> otherwise  
P<sub>5</sub> = 2<sup>2M<sub>1</sub>+M<sub>0</sub></sup>

In its natural orientation, the TILER container consists of 8192 lines of 16384 pixels of 8 bits, or 4096 lines of 16384 pixels of 16 bits, or 4096 lines of 8192 pixels of 32-bits.

**Figure 15-42. Example of 8-Bit Frame-Buffer Addressing in any Orientation**



sdram-075

This frame-buffer address generation is independent from the page size. For instance, the 8-bit frame-buffer described in Figure 15-42 and ranging from the top-left pixel located at (16, 32) to the bottom-right pixel located at (223, 159) in the natural orientation view, corresponds to the 29-bit view address offsets and full 33-bit TILER addresses given in Table 15-32.

**Table 15-32. 29-Bit View Address Offset and 33-Bit Full TILER Address for an 8-Bit Frame-Buffer**

S	Y	X	x <sub>or</sub>	y <sub>or</sub>	29-bit address offset in view	Full 33-bit TILER address
0	0	0	16	32	00080010h	100080010h
0	0	1	16160	32	00083F20h	120083F20h
0	1	0	16	8032	07D80010h	147D80010h
0	1	1	16160	8032	07D83F20h	167D83F20h

**Table 15-32. 29-Bit View Address Offset and 33-Bit Full TILER Address for an 8-Bit Frame-Buffer (continued)**

S	Y	X	x <sub>or</sub>	y <sub>or</sub>	29-bit address offset in view	Full 33-bit TILER address
1	0	0	16	32	00020020h	180020020h
1	0	1	16160	32	07E40020h	1A7E40020h
1	1	0	16	8032	00021F60h	1C0021F60h
1	1	1	16160	8032	07E41F60h	1E7E41F60h

In this example the TILER is addressed in 8-bit mode, which translates to addresses with all mode bits - bits 27 and 28 - cleared in both the 29-bit address offset in view and full 33-bit TILER address.

**15.2.5.4.2 TILER Page Mapping**

Let:

- c (c = 0, 1 or 2) be the TILER page configuration (0 for 4 kiB, 1 for 16 kiB and 2 for 64 kiB pages)
- (x<sub>0</sub>, y<sub>0</sub>) be the top left pixel and (x<sub>1</sub>, y<sub>1</sub>) be the bottom right pixel of a frame in its natural orientation
- W be the width and H the height in pixels of the container in the considered frame mode

The top left page coordinate (P<sub>x0</sub>, P<sub>y0</sub>) and bottom right page coordinates (P<sub>x1</sub>, P<sub>y1</sub>) corresponding to the frame are given by: P<sub>x0</sub> = (64.x<sub>0</sub>.2<sup>2-c</sup>) / W, P<sub>y0</sub> = (32.y<sub>0</sub>.2<sup>2-c</sup>) / H, P<sub>x1</sub> = (64.x<sub>1</sub>.2<sup>2-c</sup>) / W and P<sub>y1</sub> = (32.y<sub>1</sub>.2<sup>2-c</sup>) / H.

In its natural orientation, the TILER container consists of 8192 lines of 16384 pixels of 8 bits, or 4096 lines of 16384 pixels of 16 bits, or 4096 lines of 8192 pixels of 32-bits.

**NOTE:** The page area type has the exact same structure as the `DMM_PAT_AREA_i` registers. For instance, the 8-bit frame-buffer described in [Figure 15-42](#) and ranging from the top-left pixel located at (16, 32) to the bottom-right pixel located at (223, 159) in the natural orientation view, is mapped from the top-left page located at P0,0 to the bottom-right page located at P3,2.

**15.2.5.5 Addressing an Object in Page Mode**

In page mode, the orientation only modifies the sequence of accessed pages, not their content. In this respect the orientation shall be seen as a way to optimise the 1D object allocation in a TILER container by allowing a 1D object spanning on multiple pages to map a set of adjacent pages in any direction. For instance, a 1.25 MiB object, mapped on 320 pages of 4 kiB can be allocated in a TILER container in any of the eight orientations, see [Section 15.2.3.6.1.7.3, Element Ordering in the TILER Container](#).

The initial page can be chosen freely among all pages. In this respect, the address offset of a page in a TILER container is expressed as:

Let:

- c (c = 0, 1 or 2) be the TILER page configuration (0 for 4 kiB, 1 for 16 kiB and 2 for 64 kiB pages)
- (S, Y, X) be the orientation of the considered 1D object in page mode
- (P<sub>x</sub>, P<sub>y</sub>) be the coordinate of the initial page in the natural orientation view

The byte offset of the base address of the considered initial page in its oriented container is:

base\_address((P<sub>x</sub>, P<sub>y</sub>), (S, Y, X)) = 4096.(64.y<sub>or</sub>.2<sup>2-c</sup> + x<sub>or</sub>).2<sup>2-c</sup> when S, 4096.(32.x<sub>or</sub>.2<sup>2-c</sup> + y<sub>or</sub>).2<sup>2-c</sup> otherwise with:

x<sub>or</sub> = P<sub>x</sub> when X, 64.2<sup>2-c</sup>-1-P<sub>x</sub> otherwise

y<sub>or</sub> = P<sub>y</sub> when Y, 32.2<sup>2-c</sup>-1-P<sub>y</sub> otherwise

For instance, a 1-D object starting from the page P32,63 in the natural orientation view on a DMM using 4-kiB pages, corresponds to the 29-bit view address offsets and full 33-bit TILER addresses given in [Table 15-33](#).

**Table 15-33. 29-Bit View Address Offset and 33-Bit Full TILER Address for a 1D Object**

S	Y	X	x <sub>or</sub>	y <sub>or</sub>	29-bit address offset in view	Full 33-bit TILER address
0	0	0	32	63	1BF20000h	11BF20000h
0	0	1	223	63	1BFDF000h	13BFDF000h
0	1	0	32	64	1C020000h	15C020000h
0	1	1	223	64	1C0DF000h	17C0DF000h
1	0	0	32	63	1903F000h	19903F000h
1	0	1	223	63	1EFBF000h	1BEFBF000h
1	1	0	32	64	19040000h	1D9040000h
1	1	1	223	64	1EFC0000h	1FEFC0000h

In this example the TILER is addressed in page mode, which translates to addresses with mode bits - bits 27 and 28 - set in both the 29-bit address offset in view and full 33-bit TILER address.

### 15.2.5.6 Sharing Containers Between Different Modes

When allocating objects in TILER containers, some care must be taken to ensure that any two objects do not physically overlap.

In this respect, and to ease the object allocation and de-allocation, it is strongly advised to only share a TILER page for different objects that:

- Belong to the same mode and
- That have the exact same lifetime - that are allocated and de-allocated simultaneously.

Two objects of different modes can physically overlap if:

- One physical page is mapped twice in two different locations of the DMM LUT
- The two objects share the DMM LUT and the intersection of their two page set is non empty.

These two issues can be easily avoided by only allocating a physical page once in the DMM LUT.

The second issue is a bit trickier, especially if the allocation of objects in the various TILER containers has to be dynamic. Managing the fragmentation within a flat memory model in a space-constrained system is not straightforward - and is the main reason for the existence of memory management units in most current processors - but adding yet another constraint to the problem - the y dimension - makes it really difficult.

Coming back to the CPU-analogy, the memory fragmentation issue has been mostly solved by:

- Using a virtual memory space much larger than the actual physical memory space,
- Using virtual-to-physical translation system

to allow contiguous virtual memory allocation when:

- Sufficient physical memory is available,
- The larger contiguous physical buffer is smaller than the requested memory allocation.

Similarly, the DMM answer to this object allocation in the TILER containers is two-folds:

1. The DMM has to be dimensioned to have sufficient virtual address space in all modes to permit a static allocation of a pool of virtual buffers in all TILER modes for all TILER-aware initiators
2. Each TILER-aware initiator has to dynamically manage its own pool of virtual buffers.



## 15.2.6 DMM Register Manual

This section provides information about the DMM instance in this product. [Table 15-35](#) provides a summary of the DMM registers. The remaining parts of this section describe the registers within the module instance.

### 15.2.6.1 DMM Instance Summary

**Table 15-34. DMM Instance Summary**

Module Name	Base Address	Size
DMM	0x4E00 0000	32 Mbytes

### 15.2.6.2 DMM Registers

#### 15.2.6.2.1 DMM Register Summary

**Table 15-35. DMM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	DMM L3 Physical Address
<a href="#">DMM_REVISION</a>	R	32	0x0000 0000	0x4E00 0000
<a href="#">DMM_HWINFO</a>	R	32	0x0000 0004	0x4E00 0004
<a href="#">DMM_LISA_HWINFO</a>	R	32	0x0000 0008	0x4E00 0008
<a href="#">DMM_SYSCONFIG</a>	RW	32	0x0000 0010	0x4E00 0010
<a href="#">DMM_LISA_LOCK</a>	RW	32	0x0000 001C	0x4E00 001C
<a href="#">DMM_LISA_MAP_i<sup>(1)</sup></a>	RW	32	0x0000 0040 + (0x4 * i)	0x4E00 0040 + (0x4 * i)
<a href="#">DMM_TILER_HWINFO</a>	R	32	0x0000 0208	0x4E00 0208
<a href="#">DMM_TILER_OR0</a>	RW	32	0x0000 0220	0x4E00 0220
<a href="#">DMM_TILER_OR1</a>	RW	32	0x0000 0224	0x4E00 0224
<a href="#">DMM_PAT_HWINFO</a>	R	32	0x0000 0408	0x4E00 0408
<a href="#">DMM_PAT_GEOMETRY</a>	R	32	0x0000 040C	0x4E00 040C
<a href="#">DMM_PAT_CONFIG</a>	RW	32	0x0000 0410	0x4E00 0410
<a href="#">DMM_PAT_VIEW0</a>	RW	32	0x0000 0420	0x4E00 0420
<a href="#">DMM_PAT_VIEW1</a>	RW	32	0x0000 0424	0x4E00 0424
<a href="#">DMM_PAT_VIEW_MAP_i<sup>(1)</sup></a>	RW	32	0x0000 0440 + (0x4 * i)	0x4E00 0440 + (0x4 * i)
<a href="#">DMM_PAT_VIEW_MAP_BASE</a>	RW	32	0x0000 0460	0x4E00 0460
<a href="#">DMM_PAT_IRQSTATUS_RAW</a>	RW	32	0x0000 0480	0x4E00 0480
<a href="#">DMM_PAT_IRQSTATUS</a>	RW	32	0x0000 0490	0x4E00 0490
<a href="#">DMM_PAT_IRQENABLE_SET</a>	RW	32	0x0000 04A0	0x4E00 04A0
<a href="#">DMM_PAT_IRQENABLE_CLR</a>	RW	32	0x0000 04B0	0x4E00 04B0
<a href="#">DMM_PAT_STATUS_i<sup>(1)</sup></a>	R	32	0x0000 04C0 + (0x4 * i)	0x4E00 04C0 + (0x4 * i)
<a href="#">DMM_PAT_DESCR_i<sup>(1)</sup></a>	RW	32	0x0000 0500 + (0x10 * i)	0x4E00 0500 + (0x10 * i)
<a href="#">DMM_PAT_AREA_i<sup>(1)</sup></a>	RW	32	0x0000 0504 + (0x10 * i)	0x4E00 0504 + (0x10 * i)
<a href="#">DMM_PAT_CTRL_i<sup>(1)</sup></a>	RW	32	0x0000 0508 + (0x10 * i)	0x4E00 0508 + (0x10 * i)
<a href="#">DMM_PAT_DATA_i<sup>(1)</sup></a>	RW	32	0x0000 050C + (0x10 * i)	0x4E00 050C + (0x10 * i)
<a href="#">DMM_PEG_HWINFO</a>	R	32	0x0000 0608	0x4E00 0608
<a href="#">DMM_PEG_PRIO_k<sup>(2)</sup></a>	RW	32	0x0000 0620 + (0x4 * k)	0x4E00 0620 + (0x4 * k)

<sup>(1)</sup> i = 0 to 3 for DMM

<sup>(2)</sup> k = 0 to 7

**Table 15-35. DMM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	DMM L3 Physical Address
DMM_PEG_PRIO_PAT	RW	32	0x0000 0640	0x4E00 0640

**15.2.6.2.2 DMM Register Description****Table 15-36. DMM\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	DMM Revision Number		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Revision Number	R	0x---- ---- <sup>(1)</sup>

<sup>(1)</sup> TI internal data**Table 15-37. Register Call Summary for Register DMM\_REVISION**

DMM Register Manual

- [DMM Register Summary: \[0\]](#)

**Table 15-38. DMM\_HWINFO**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	DMM hardware configuration		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ROBIN_CNT				RESERVED				ELLA_CNT				RESERVED				TILER_CNT							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x000
19:16	ROBIN_CNT	Number of ROBIN in the DMM	R	0x2
15:12	RESERVED	Reserved	R	0x0
11:8	ELLA_CNT	Number of ELLA in the DMM	R	0x1
7:4	RESERVED	Reserved	R	0x0
3:0	TILER_CNT	Number of TILER in the DMM	R	0x2

**Table 15-39. Register Call Summary for Register DMM\_HWINFO**

DMM Integration

- [DMM Configuration: \[0\] \[1\] \[2\]](#)

DMM Register Manual

- [DMM Register Summary: \[3\]](#)

**Table 15-40. DMM\_LISA\_HWINFO**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	DMM
<b>Physical Address</b>	See <a href="#">Table 15-35</a>		
<b>Description</b>	DMM hardware configuration for LISA		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SDRC_CNT				RESERVED			SECTION_CNT								

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11:8	SDRC_CNT	Number of attached SDRAM controllers	R	0x2
7:5	RESERVED	Reserved	R	0x0
4:0	SECTION_CNT	Number of DMM sections	R	0x02

**Table 15-41. Register Call Summary for Register DMM\_LISA\_HWINFO**

DMM Integration

- [DMM Configuration: \[0\] \[1\]](#)

DMM Register Manual

- [DMM Register Summary: \[2\]](#)

**Table 15-42. DMM\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	DMM
<b>Physical Address</b>	See <a href="#">Table 15-35</a>		
<b>Description</b>	DMM clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLE_MODE			RESERVED												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:2	IDLE_MODE	Configuration of the local target state management mode. 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. Backup mode, for debug only. 0x1: No-idle mode: local target never enters idle state. Backup mode, for debug only 0x3: Reserved 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events.	RW	0x2

Bits	Field Name	Description	Type	Reset
1:0	RESERVED	Reserved	RW W0Only	0x0

**Table 15-43. Register Call Summary for Register DMM\_SYSCONFIG**

DMM Integration

- [DMM Integration: \[0\]](#)

DMM Functional Description

- [DMM Power Management: \[1\]](#)

DMM Basic Programming Model

- [DMM Module Global Initialization: \[2\]](#)

DMM Register Manual

- [DMM Register Summary: \[3\]](#)

**Table 15-44. DMM\_LISA\_LOCK**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	DMM
<b>Physical Address</b>	See <a href="#">Table 15-35</a>		
<b>Description</b>	DMM memory mapping lock		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOCK

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Should be written as 0	R	0x0000 0000
0	LOCK	DMM lock map Write 0x0: No effect (clear on reset only) Read 0x0: <a href="#">DMM_LISA_MAP_i</a> un-locked Read 0x1: <a href="#">DMM_LISA_MAP_i</a> locked Write 0x1: Locking <a href="#">DMM_LISA_MAP_i</a> registers	RW	0

**Table 15-45. Register Call Summary for Register DMM\_LISA\_LOCK**

DMM Integration

- [DMM Integration: \[0\]](#)

DMM Functional Description

- [Address Mapping: \[1\]](#)
- [LISA Description: \[2\]](#)

DMM Basic Programming Model

- [Interleaving settings: \[3\]](#)

DMM Register Manual

- [DMM Register Summary: \[4\]](#)

**Table 15-46. DMM\_LISA\_MAP\_i**

<b>Address Offset</b>	0x0000 0040 + (0x4 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	DMM memory mapping n = 0 for the first memory mapping register, n = 1 for the second memory mapping register...		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_ADDR								RESERVED	SYS_SIZE				SDRC_INTL	SDRC_ADDRSPC	RESERVED						SDRC_MAP	SDRC_ADDR									

Bits	Field Name	Description	Type	Reset
31:24	SYS_ADDR	DMM system section address MSB for view mapping n	RW	0x00
23	RESERVED	Reserved	RW W0Only	0
22:20	SYS_SIZE	DMM system section size for view mapping n 0x0: 16-MiB section 0x1: 32-MiB section 0x2: 64-MiB section 0x3: 128-MiB section 0x4: 256-MiB section 0x5: 512-MiB section 0x6: 1-GiB section 0x7: 2-GiB section	RW	0x0
19:18	SDRC_INTL	SDRAM controller interleaving mode 0x0: No interleaving 0x1: 128-byte interleaving 0x2: 256-byte interleaving 0x3: 512-byte interleaving	RW	0x0
17:16	SDRC_ADDRSPC	SDRAM controller address space for view mapping n	RW	0x0
15:10	RESERVED	Reserved	RW W0Only	0x00
9:8	SDRC_MAP	SDRAM controller mapping for view mapping n 0x0: Un-mapped 0x1: Mapped on SDRC 0 only (not interleaved) 0x2: Mapped on SDRC 1 only (not interleaved) 0x3: Mapped on SDRC 0 and SDRC 1 (interleaved)	RW	0
7:0	SDRC_ADDR	SDRAM controller address MSB for view mapping n	RW	0x00

**Table 15-47. Register Call Summary for Register DMM\_LISA\_MAP\_i**

DMM Integration

- [DMM Integration: \[0\]](#)

DMM Functional Description

- [Dynamic Mapping: \[1\]](#)
- [Address Mapping: \[2\]](#)
- [LISA Description: \[3\] \[4\]](#)

DMM Basic Programming Model

- [Interleaving settings: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

**Table 15-47. Register Call Summary for Register DMM\_LISA\_MAP\_i (continued)**

DMM Register Manual

- [DMM Register Summary: \[11\]](#)
- [DMM Register Description: \[12\] \[13\] \[14\]](#)

**Table 15-48. DMM\_TILER\_HWINFO**

<b>Address Offset</b>	0x0000 0208	<b>Instance</b>	DMM
<b>Physical Address</b>	See <a href="#">Table 15-35</a>		
<b>Description</b>	DMM hardware configuration for TILER		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OR_CNT															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4:0	OR_CNT	Number of TILER orientation entries Read 0x1: One orientation entry Read 0x2: Two orientation entries Read 0x4: Four orientation entries Read 0x8: Eight orientation entries Read 0x10: Sixteen orientation entries	R	0x10

**Table 15-49. Register Call Summary for Register DMM\_TILER\_HWINFO**

DMM Integration

- [DMM Configuration: \[0\]](#)

DMM Register Manual

- [DMM Register Summary: \[1\]](#)

**Table 15-50. DMM\_TILER\_OR0**

<b>Address Offset</b>	0x0000 02200	<b>Index</b>	0
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	DMM TILER orientation (initiators 0 to 7)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W7		OR7		W6		OR6		W5		OR5		W4		OR4		W3		OR3		W2		OR2		W1		OR1		W0		OR0	

Bits	Field Name	Description	Type	Reset
31	W7	Write-enable for OR7 bit field Write 0x0: OR7 field is unchanged Write 0x1: OR7 field is updated	RW	0
30:28	OR7	Orientation for initiator 7	RW	0x0
27	W6	Write-enable for OR6 bit field Write 0x0: OR6 field is unchanged Write 0x1: OR6 field is updated	RW	0
26:24	OR6	Orientation for initiator 6	RW	0x0

Bits	Field Name	Description	Type	Reset
23	W5	Write-enable for OR5 bit field Write 0x0: OR5 field is unchanged Write 0x1: OR5 field is updated	RW	0
22:20	OR5	Orientation for initiator 5	RW	0x0
19	W4	Write-enable for OR4 bit field Write 0x0: OR4 field is unchanged Write 0x1: OR4 field is updated	RW	0
18:16	OR4	Orientation for initiator 4	RW	0x0
15	W3	Write-enable for OR3 bit field Write 0x0: OR3 field is unchanged Write 0x1: OR3 field is updated	RW	0
14:12	OR3	Orientation for initiator 3	RW	0x0
11	W2	Write-enable for OR2 bit field Write 0x0: OR2 field is unchanged Write 0x1: OR2 field is updated	RW	0
10:8	OR2	Orientation for initiator 2	RW	0x0
7	W1	Write-enable for OR1 bit field Write 0x0: OR1 field is unchanged Write 0x1: OR1 field is updated	RW	0
6:4	OR1	Orientation for initiator 1	RW	0x0
3	W0	Write-enable for OR0 bit field Write 0x0: OR0 field is unchanged Write 0x1: OR0 field is updated	RW	0
2:0	OR0	Orientation for initiator 0	RW	0x0

**Table 15-51. Register Call Summary for Register DMM\_TILER\_OR0**

DMM Integration

- [DMM Integration: \[0\]](#)

DMM Functional Description

- [TILER Macro-architecture: \[1\] \[2\]](#)

DMM Basic Programming Model

- [Aliased tiled view orientation settings and LUT refill: \[3\] \[4\]](#)

DMM Register Manual

- [DMM Register Summary: \[5\]](#)

**Table 15-52. DMM\_TILER\_OR1**

<b>Address Offset</b>	0x0000 02204	<b>Index</b>	0
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	DMM TILER orientation (initiators 8 to 15)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	OR15			W14	OR14			W13	OR13			W12	OR12			W11	OR11			W10	OR10			W9	OR9			W8	OR8		

Bits	Field Name	Description	Type	Reset
31	W15	Write-enable for OR15 bit field Write 0x0: OR15 field is unchanged Write 0x1: OR15 field is updated	RW	0

Bits	Field Name	Description	Type	Reset
30:28	OR15	Orientation for initiator 15	RW	0x0
27	W14	Write-enable for OR14 bit field Write 0x0: OR14 field is unchanged Write 0x1: OR14 field is updated	RW	0
26:24	OR14	Orientation for initiator 14	RW	0x0
23	W13	Write-enable for OR13 bit field Write 0x0: OR13 field is unchanged Write 0x1: OR13 field is updated	RW	0
22:20	OR13	Orientation for initiator 13	RW	0x0
19	W12	Write-enable for OR12 bit field Write 0x0: OR12 field is unchanged Write 0x1: OR12 field is updated	RW	0
18:16	OR12	Orientation for initiator 12	RW	0x0
15	W11	Write-enable for OR11 bit field Write 0x0: OR11 field is unchanged Write 0x1: OR11 field is updated	RW	0
14:12	OR11	Orientation for initiator 11	RW	0x0
11	W10	Write-enable for OR10 bit field Write 0x0: OR10 field is unchanged Write 0x1: OR10 field is updated	RW	0
10:8	OR10	Orientation for initiator 10	RW	0x0
7	W9	Write-enable for OR9 bit field Write 0x0: OR9 field is unchanged Write 0x1: OR9 field is updated	RW	0
6:4	OR9	Orientation for initiator 9	RW	0x0
3	W8	Write-enable for OR8 bit field Write 0x0: OR8 field is unchanged Write 0x1: OR8 field is updated	RW	0
2:0	OR8	Orientation for initiator 8	RW	0x0

**Table 15-53. Register Call Summary for Register DMM\_TILER\_OR1**

DMM Integration

- [DMM Integration: \[0\]](#)

DMM Functional Description

- [TILER Macro-architecture: \[1\] \[2\]](#)

DMM Basic Programming Model

- [Aliased tiled view orientation settings and LUT refill: \[3\] \[4\]](#)

DMM Register Manual

- [DMM Register Summary: \[5\]](#)



**Table 15-54. DMM\_PAT\_HWINFO**

<b>Address Offset</b>	0x0000 0408		
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	DMM hardware configuration for PAT		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				ENGINE_CNT				RESERVED				LUT_CNT				RESERVED				VIEW_MAP_CNT				RESERVED				VIEW_CNT			

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:24	ENGINE_CNT	Number of PAT refill engines	R	0x04
23:21	RESERVED	Reserved	R	0x0
20:16	LUT_CNT	Number of PAT LUT for page-grained physical address translation	R	0x01
15:12	RESERVED	Reserved	R	0x0
11:8	VIEW_MAP_CNT	Number of internal PAT view mappings. Read 0x1: One view map Read 0x2: Two view maps Read 0x4: Four view maps Read 0x8: Eight view maps	R	0x4
7	RESERVED	Reserved	R	0
6:0	VIEW_CNT	Number of PAT view entries Read 0x1: One view entry Read 0x2: Two view entries Read 0x4: Four view entries Read 0x8: Eight view entries Read 0x10: Sixteen view entries Read 0x20: Thirty-two view entries Read 0x40: Sixty-four view entries	R	0x10

**Table 15-55. Register Call Summary for Register DMM\_PAT\_HWINFO**

DMM Integration

- [DMM Configuration: \[0\] \[1\] \[2\] \[3\]](#)

DMM Register Manual

- [DMM Register Summary: \[4\]](#)

**Table 15-56. DMM\_PAT\_GEOMETRY**

<b>Address Offset</b>	0x0000 040C	<b>Instance</b>	DMM
<b>Physical Address</b>	See <a href="#">Table 15-35</a>		
<b>Description</b>	PAT geometry-related settings		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CONT_HGHT		RESERVED				CONT_WDTH				RESERVED		ADDR_RANGE						RESERVED		PAGE_SZ							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26:24	CONT_HGHT	Container height in pages Read 0x1: Container height of 32 pages Read 0x2: Container height of 64 pages Read 0x4: Container height of 128 pages	R	0x4
23:20	RESERVED	Reserved	R	0x0
19:16	CONT_WDTH	Container width in pages Read 0x2: Container width of 64 pages Read 0x4: Container width of 128 pages Read 0x8: Container width of 256 pages	R	0x8
15:14	RESERVED	Reserved	R	0x0
13:8	ADDR_RANGE	PAT output physical address range Read 0x1: 128-MiB range Read 0x2: 256-MiB range Read 0x4: 512-MiB range Read 0x8: 1-GiB range Read 0x10: 2-GiB range Read 0x20: 4-GiB range	R	0x10
7:5	RESERVED	Reserved	R	0x0
4:0	PAGE_SZ	Page size in 4-kB granularity Read 0x1: 4-kB page Read 0x4: 16-kB page Read 0x10: 64-kB page	R	0x01

**Table 15-57. Register Call Summary for Register DMM\_PAT\_GEOMETRY**

## DMM Integration

- [DMM Configuration: \[0\] \[1\] \[2\] \[3\]](#)

## DMM Register Manual

- [DMM Register Summary: \[4\]](#)

**Table 15-58. DMM\_PAT\_CONFIG**

<b>Address Offset</b>	0x0000 0410		<b>Instance</b>	DMM
<b>Physical Address</b>	See <a href="#">Table 15-35</a>			
<b>Description</b>	This is the PAT configuration register aimed at defining the major PAT configuration of each refill engine.			
<b>Type</b>	RW			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE3	MODE2	MODE1	MODE0												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	MODE3	Mode of refill engine 3 0: Normal mode 1: Direct LUT access	RW	0
2	MODE2	Mode of refill engine 2 0: Normal mode 1: Direct LUT access	RW	0
1	MODE1	Mode of refill engine 1 0: Normal mode 1: Direct LUT access	RW	0
0	MODE0	Mode of refill engine 0 0: Normal mode 1: Direct LUT access	RW	0

**Table 15-59. Register Call Summary for Register DMM\_PAT\_CONFIG**

DMM Functional Description

- [PAT Views: \[0\]](#)

DMM Basic Programming Model

- [Configuration Settings and LUT refill: \[1\] \[2\]](#)

DMM Register Manual

- [DMM Register Summary: \[3\]](#)

**Table 15-60. DMM\_PAT\_VIEW0**

<b>Address Offset</b>	0x0000 0420		<b>Index</b>	
<b>Physical Address</b>	See <a href="#">Table 15-35</a>		<b>Instance</b>	DMM
<b>Description</b>	DMM PAT View register (initiators 0 to 7)			
<b>Type</b>	RW			

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W7	RESERVED	V7	RESERVED	V6	RESERVED	V5	RESERVED	V4	RESERVED	V3	RESERVED	V2	RESERVED	V1	RESERVED	V0	RESERVED	V0	RESERVED	V0	RESERVED	V0	RESERVED	V0	RESERVED	V0	RESERVED	V0	RESERVED	V0	

## Dynamic Memory Manager

www.ti.com

Bits	Field Name	Description	Type	Reset
31	W7	Write-enable for V7 bit field Write 0x0: V7 field is unchanged Write 0x1: V7 field is updated	RW	0
30	RESERVED	Reserved	RW W0Only	0
29:28	V7	PAT view for initiator 7	RW	0x0
27	W6	Write-enable for V6 bit field Write 0x0: V6 field is unchanged Write 0x1: V6 field is updated	RW	0
26	RESERVED	Reserved	RW W0Only	0
25:24	V6	PAT view for initiator 6	RW	0x0
23	W5	Write-enable for V5 bit field Write 0x0: V5 field is unchanged Write 0x1: V5 field is updated	RW	0
22	RESERVED	Reserved	RW W0Only	0
21:20	V5	PAT view for initiator 5	RW	0x0
19	W4	Write-enable for V4 bit field Write 0x0: V4 field is unchanged Write 0x1: V4 field is updated	RW	0
18	RESERVED	Reserved	RW W0Only	0
17:16	V4	PAT view for initiator 4	RW	0x0
15	W3	Write-enable for V3 bit field Write 0x0: V3 field is unchanged Write 0x1: V3 field is updated	RW	0
14	RESERVED	Reserved	RW W0Only	0
13:12	V3	PAT view for initiator 3	RW	0x0
11	W2	Write-enable for V2 bit field Write 0x0: V2 field is unchanged Write 0x1: V2 field is updated	RW	0
10	RESERVED	Reserved	RW W0Only	0
9:8	V2	PAT view for initiator 2	RW	0x0
7	W1	Write-enable for V1 bit field Write 0x0: V1 field is unchanged Write 0x1: V1 field is updated	RW	0
6	RESERVED	Reserved	RW W0Only	0
5:4	V1	PAT view for initiator 1	RW	0x0
3	W0	Write-enable for V0 bit field Write 0x0: V0 field is unchanged Write 0x1: V0 field is updated	RW	0
2	RESERVED	Reserved	RW W0Only	0
1:0	V0	PAT view for initiator 0	RW	0x0

**Table 15-61. Register Call Summary for Register DMM\_PAT\_VIEW0**

DMM Integration	<ul style="list-style-type: none"> <li>DMM Integration: [0]</li> </ul>
DMM Functional Description	<ul style="list-style-type: none"> <li>PAT View Mappings: [1] [2]</li> </ul>
DMM Basic Programming Model	<ul style="list-style-type: none"> <li>Configuration Settings and LUT refill: [3] [4]</li> <li>PAT in indirect translation mode: [5] [6]</li> </ul>
DMM Register Manual	<ul style="list-style-type: none"> <li>DMM Register Summary: [7]</li> </ul>

**Table 15-62. DMM\_PAT\_VIEW1**

<b>Address Offset</b>	0x0000 0424	<b>Index</b>	
<b>Physical Address</b>	See Table 15-35	<b>Instance</b>	DMM
<b>Description</b>	DMM PAT view register (initiators 8 to 15)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	RESERVED	V15	RESERVED	V14	RESERVED	V14	RESERVED	V13	RESERVED	V13	RESERVED	V12	RESERVED	V12	RESERVED	V11	RESERVED	V11	RESERVED	V10	RESERVED	V10	RESERVED	V9	RESERVED	V9	RESERVED	V8	RESERVED	V8	RESERVED

Bits	Field Name	Description	Type	Reset
31	W15	Write-enable for V15 bit field Write 0x0: V15 field is unchanged Write 0x1: V15 field is updated	RW	0
30	RESERVED	Reserved	RW W0Only	0
29:28	V15	PAT view for initiator 15	RW	0x0
27	W14	Write-enable for V14 bit field Write 0x0: V14 field is unchanged Write 0x1: V14 field is updated	RW	0
26	RESERVED	Reserved	RW W0Only	0
25:24	V14	PAT view for initiator 14	RW	0x0
23	W13	Write-enable for V13 bit field Write 0x0: V13 field is unchanged Write 0x1: V13 field is updated	RW	0
22	RESERVED	Reserved	RW W0Only	0
21:20	V13	PAT view for initiator 13	RW	0x0
19	W12	Write-enable for V12 bit field Write 0x0: V12 field is unchanged Write 0x1: V12 field is updated	RW	0
18	RESERVED	Reserved	RW W0Only	0
17:16	V12	PAT view for initiator 12	RW	0x0
15	W11	Write-enable for V11 bit field Write 0x0: V11 field is unchanged Write 0x1: V11 field is updated	RW	0

Bits	Field Name	Description	Type	Reset
14	RESERVED	Reserved	RW W0Only	0
13:12	V11	PAT view for initiator 11	RW	0x0
11	W10	Write-enable for V10 bit field Write 0x0: V10 field is unchanged Write 0x1: V10 field is updated	RW	0
10	RESERVED	Reserved	RW W0Only	0
9:8	V10	PAT view for initiator 10	RW	0x0
7	W9	Write-enable for V9 bit field Write 0x0: V9 field is unchanged Write 0x1: V9 field is updated	RW	0
6	RESERVED	Reserved	RW W0Only	0
5:4	V9	PAT view for initiator 9	RW	0x0
3	W8	Write-enable for V8 bit field Write 0x0: V8 field is unchanged Write 0x1: V8 field is updated	RW	0
2	RESERVED	Reserved	RW W0Only	0
1:0	V8	PAT view for initiator 8	RW	0x0

**Table 15-63. Register Call Summary for Register DMM\_PAT\_VIEW1**

DMM Integration

- [DMM Integration: \[0\]](#)

DMM Functional Description

- [PAT View Mappings: \[1\] \[2\]](#)

DMM Basic Programming Model

- [Configuration Settings and LUT refill: \[3\] \[4\]](#)
- [PAT in indirect translation mode: \[5\]](#)

DMM Register Manual

- [DMM Register Summary: \[6\]](#)

**Table 15-64. DMM\_PAT\_VIEW\_MAP\_i**

<b>Address Offset</b>	0x0000 0440 + (0x4 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	PAT view mapping register n = 0 for the first view mapping register, n = 1 for the second view mapping register...		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_PAGE	RESERVED						CONT_PAGE	ACCESS_32	RESERVED						CONT_32	ACCESS_16	RESERVED						CONT_16	ACCESS_8	RESERVED						CONT_8

Bits	Field Name	Description	Type	Reset
31	ACCESS_PAGE	Kind of access for this page mode container in view mapping n 0x0: Direct access, container base address given in CONT_PAGE 0x1: indirect access through the LUT indexed by CONT_PAGE	RW	0
30:28	RESERVED	Reserved	RW W0Only	0x0
27:24	CONT_PAGE	Container for page mode in view mapping n	RW	0x0
23	ACCESS_32	Kind of access for this 32-bit mode container in view mapping n 0x0: Direct access, container base address given in CONT_32 0x1: indirect access through the LUT indexed by CONT_32	RW	0
22:20	RESERVED	Reserved	RW W0Only	0x0
19:16	CONT_32	Container for 32-bit mode in view mapping n	RW	0x0
15	ACCESS_16	Kind of access for this 16-bit mode container in view mapping n 0x0: Direct access, container base address given in CONT_16 0x1: indirect access through the LUT indexed by CONT_16	RW	0
14:12	RESERVED	Reserved	RW W0Only	0x0
11:8	CONT_16	Container for 16-bit mode in view mapping n	RW	0x0
7	ACCESS_8	Kind of access for this 8-bit mode container in view mapping n 0x0: Direct access, container base address given in CONT_8 0x1: indirect access through the LUT indexed by CONT_8	RW	0
6:4	RESERVED	Reserved	RW W0Only	0x0
3:0	CONT_8	Container for 8-bit mode in view mapping n	RW	0x0

**Table 15-65. Register Call Summary for Register DMM\_PAT\_VIEW\_MAP\_i**

## DMM Integration

- [DMM Integration: \[0\]](#)

## DMM Functional Description

- [PAT Views: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

## DMM Basic Programming Model

- [Configuration Settings and LUT refill: \[8\]](#)

## DMM Register Manual

- [DMM Register Summary: \[9\]](#)

**Table 15-66. DMM\_PAT\_VIEW\_MAP\_BASE**

<b>Address Offset</b>	0x0000 0460		
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	Base address of all view mappings		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31	BASE_ADDR	MSB of the PAT view mapping base address	RW	0
30:0	RESERVED	Reserved	RW W0Only	0x0000 0000

**Table 15-67. Register Call Summary for Register DMM\_PAT\_VIEW\_MAP\_BASE**

DMM Integration

- [DMM Integration: \[0\]](#)

DMM Functional Description

- [PAT View Map Base Address: \[1\]](#)

DMM Basic Programming Model

- [Configuration Settings and LUT refill: \[2\]](#)
- [Aliased tiled view orientation settings and LUT refill: \[3\]](#)
- [PAT in indirect translation mode: \[4\]](#)

DMM Register Manual

- [DMM Register Summary: \[5\]](#)

**Table 15-68. DMM\_PAT\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0480		
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	Per-event raw interrupt status vector. Raw status is set even if the related event is not enabled. Write 1 to set the (raw) status, mostly for debug. n = 0 for the first interrupt status raw register, n = 1 for the second interrupt status raw register...		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ERR_LUT_MISS1	ERR_UPD_DATA1	ERR_UPD_CTRL1	ERR_UPD_AREA1	ERR_INV_DATA1	ERR_INV_DSC1	FILL_LST1	FILL_DSC1	ERR_LUT_MISS0	ERR_UPD_DATA0	ERR_UPD_CTRL0	ERR_UPD_AREA0	ERR_INV_DATA0	ERR_INV_DSC0	FILL_LST0	FILL_DSC0



Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW W0Only	0x0000
15	ERR_LUT_MISS1	Access to a yet-to-be-refilled area event in area 4.n+1 Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
14	ERR_UPD_DATA1	Data register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
13	ERR_UPD_CTRL1	Control register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
12	ERR_UPD_AREA1	Area register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
11	ERR_INV_DATA1	Invalid entry-table pointer error event in area 4.n+1 Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
10	ERR_INV_DSC1	Invalid descriptor pointer error event in area 4.n+1 Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
9	FILL_LST1	End of refill event for the last descriptor in area 4.n+1 Write 0x0: Keep area 1 refill done event Read 0x0: Area 1 is yet-to-be refilled Read 0x1: Area 1 is refilled Write 0x1: Set area 1 refill done event	RW W1toSet	0
8	FILL_DSC1	End of refill event for any descriptor in area 4.n+1 Write 0x0: Keep area 1 refill done event Read 0x0: Area 1 is yet-to-be refilled Read 0x1: Area 1 is refilled Write 0x1: Set area 1 refill done event	RW W1toSet	0
7	ERR_LUT_MISS0	Access to a yet-to-be-refilled area event in area 4.n Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
6	ERR_UPD_DATA0	Data register update whilst refilling error event in area 4.n Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
5	ERR_UPD_CTRL0	Control register update whilst refilling error event in area 4.n Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
4	ERR_UPD_AREA0	Area register update whilst refilling error event in area 4.n Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
3	ERR_INV_DATA0	Invalid entry-table pointer error event in area 4.n Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
2	ERR_INV_DSC0	Invalid descriptor pointer error event in area 4.n Write 0x0: Keep current error event Read 0x0: No such error event Read 0x1: Error event happened Write 0x1: Set error event	RW W1toSet	0
1	FILL_LST0	End of refill event for the last descriptor in area 4.n Write 0x0: Keep area 0 refill done event Read 0x0: Area 0 is yet-to-be refilled Read 0x1: Area 0 is refilled Write 0x1: Set area 0 refill done event	RW W1toSet	0
0	FILL_DSC0	End of refill event for any descriptor in area 4.n Write 0x0: Keep area 0 refill done event Read 0x0: Area 0 is yet-to-be refilled Read 0x1: Area 0 is refilled Write 0x1: Set area 0 refill done event	RW W1toSet	0

**Table 15-69. Register Call Summary for Register DMM\_PAT\_IRQSTATUS\_RAW**

DMM Functional Description

- [DMM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)

DMM Register Manual

- [DMM Register Summary: \[32\]](#)

**Table 15-70. DMM\_PAT\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0490
<b>Physical Address</b>	See <a href="#">Table 15-35</a>
<b>Instance</b>	DMM
<b>Description</b>	Per-event "enabled" interrupt status vector. Enabled status is not set unless the event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). n = 0 for the first interrupt status register, n = 1 for the second interrupt status register...
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ERR_LUT_MISS1	ERR_UPD_DATA1	ERR_UPD_CTRL1	ERR_UPD_AREA1	ERR_INV_DATA1	ERR_INV_DSC1	FILL_LST1	FILL_DSC1	ERR_LUT_MISS0	ERR_UPD_DATA0	ERR_UPD_CTRL0	ERR_UPD_AREA0	ERR_INV_DATA0	ERR_INV_DSC0	FILL_LST0	FILL_DSC0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW W0Only	0x0000
15	ERR_LUT_MISS1	Access to a yet-to-be-refilled area event in area 4.n+1 Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
14	ERR_UPD_DATA1	Data register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
13	ERR_UPD_CTRL1	Control register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
12	ERR_UPD_AREA1	Area register update whilst refilling error event in area 4.n+1 Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
11	ERR_INV_DATA1	Invalid entry-table pointer error event in area 4.n+1 Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
10	ERR_INV_DSC1	Invalid descriptor pointer error event in area 4.n+1 Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
9	FILL_LST1	End of refill event for the last descriptor in area 4.n+1 Write 0x0: Keep current area refill done maskable event Read 0x0: Current area is yet-to-be refilled or this event is masked Read 0x1: Current area is refilled Write 0x1: Clear current area refill done maskable event	RW W1toClr	0
8	FILL_DSC1	End of refill event for any descriptor in area 4.n+1 Write 0x0: Keep current area refill done maskable event Read 0x0: Current area is yet-to-be refilled or this event is masked Read 0x1: Current area is refilled Write 0x1: Clear current area refill done maskable event	RW W1toClr	0
7	ERR_LUT_MISS0	Access to a yet-to-be-refilled area event in area 4.n Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
6	ERR_UPD_DATA0	Data register update whilst refilling error event in area 4.n Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
5	ERR_UPD_CTRL0	Control register update whilst refilling error event in area 4.n Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
4	ERR_UPD_AREA0	Area register update whilst refilling error event in area 4.n Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
3	ERR_INV_DATA0	Invalid entry-table pointer error event in area 4.n Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0
2	ERR_INV_DSC0	Invalid descriptor pointer error event in area 4.n Write 0x0: Keep current maskable error event Read 0x0: No such error event or this event is masked Read 0x1: Error event happened Write 0x1: Clear this maskable error event	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
1	FILL_LST0	End of refill event for the last descriptor in area 4.n Write 0x0: Keep current area refill done maskable event Read 0x0: Current area is yet-to-be refilled or this event is masked Read 0x1: Current area is refilled Write 0x1: Clear current area refill done maskable event	RW W1toClr	0
0	FILL_DSC0	End of refill event for any descriptor in area 4.n Write 0x0: Keep current area refill done maskable event Read 0x0: Current area is yet-to-be refilled or this event is masked Read 0x1: Current area is refilled Write 0x1: Clear current area refill done maskable event	RW W1toClr	0

**Table 15-71. Register Call Summary for Register DMM\_PAT\_IRQSTATUS**

DMM Functional Description

- [DMM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)

DMM Basic Programming Model

- [Error handling: \[32\] \[33\]](#)

DMM Register Manual

- [DMM Register Summary: \[34\]](#)

**Table 15-72. DMM\_PAT\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 04A0
<b>Physical Address</b>	See <a href="#">Table 15-35</a>
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register. n = 0 for the first interrupt enable set register, n = 1 for the second interrupt enable set register...
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ERR_LUT_MISS1	ERR_UPD_DATA1	ERR_UPD_CTRL1	ERR_UPD_AREA1	ERR_INV_DATA1	ERR_INV_DSC1	FILL_LST1	FILL_DSC1	ERR_LUT_MISS0	ERR_UPD_DATA0	ERR_UPD_CTRL0	ERR_UPD_AREA0	ERR_INV_DATA0	ERR_INV_DSC0	FILL_LST0	FILL_DSC0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW W0Only	0x0000
15	ERR_LUT_MISS1	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
14	ERR_UPD_DATA1	Unexpected data register update whilst refilling interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
13	ERR_UPD_CTRL1	Unexpected control register update whilst refilling interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
12	ERR_UPD_AREA1	Unexpected area register update whilst refilling interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
11	ERR_INV_DATA1	Invalid entry-table pointer interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
10	ERR_INV_DSC1	Invalid descriptor pointer interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
9	FILL_LST1	End of refill interrupt source mask for the last descriptor in area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
8	FILL_DSC1	End of refill interrupt source mask for any descriptor in area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
7	ERR_LUT_MISS0	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
6	ERR_UPD_DATA0	Unexpected data register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
5	ERR_UPD_CTRL0	Unexpected control register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
4	ERR_UPD_AREA0	Unexpected area register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
3	ERR_INV_DATA0	Invalid entry-table pointer interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
2	ERR_INV_DSC0	Invalid descriptor pointer interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
1	FILL_LST0	End of refill interrupt source mask for the last descriptor in area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0
0	FILL_DSC0	End of refill interrupt source mask for any descriptor in area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Enable (unmask) this interrupt source	RW W1toSet	0

**Table 15-73. Register Call Summary for Register DMM\_PAT\_IRQENABLE\_SET**

## DMM Functional Description

- [DMM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)

## DMM Basic Programming Model

- [Error handling: \[32\] \[33\]](#)

## DMM Register Manual

- [DMM Register Summary: \[34\]](#)

**Table 15-74. DMM\_PAT\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 04B0
<b>Physical Address</b>	See <a href="#">Table 15-35</a>
<b>Instance</b>	DMM
<b>Description</b>	Per-event interrupt enable bit vector. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register. n = 0 for the first interrupt enable clear register, n = 1 for the second interrupt enable clear register...
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ERR_LUT_MISS1	ERR_UPD_DATA1	ERR_UPD_CTRL1	ERR_UPD_AREA1	ERR_INV_DATA1	ERR_INV_DSC1	FILL_LST1	FILL_DSC1	ERR_LUT_MISS0	ERR_UPD_DATA0	ERR_UPD_CTRL0	ERR_UPD_AREA0	ERR_INV_DATA0	ERR_INV_DSC0	FILL_LST0	FILL_DSC0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW W0Only	0x0000
15	ERR_LUT_MISS1	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n+1  Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
14	ERR_UPD_DATA1	Unexpected data register update whilst refilling interrupt source mask for area 4.n+1  Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
13	ERR_UPD_CTRL1	Unexpected control register update whilst refilling interrupt source mask for area 4.n+1  Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
12	ERR_UPD_AREA1	Unexpected area register update whilst refilling interrupt source mask for area 4.n+1  Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
11	ERR_INV_DATA1	Invalid entry-table pointer interrupt source mask for area 4.n+1  Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0



Bits	Field Name	Description	Type	Reset
10	ERR_INV_DSC1	Invalid descriptor pointer interrupt source mask for area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
9	FILL_LST1	End of refill interrupt source mask for the last descriptor in area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
8	FILL_DSC1	End of refill interrupt source mask for any descriptor in area 4.n+1 Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
7	ERR_LUT_MISS0	Unexpected access to a yet-to-be-refilled area interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
6	ERR_UPD_DATA0	Unexpected data register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
5	ERR_UPD_CTRL0	Unexpected control register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
4	ERR_UPD_AREA0	Unexpected area register update whilst refilling interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
3	ERR_INV_DATA0	Invalid entry-table pointer interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
2	ERR_INV_DSC0	Invalid descriptor pointer interrupt source mask for area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
1	FILL_LST0	End of refill interrupt source mask for the last descriptor in area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0
0	FILL_DSC0	End of refill interrupt source mask for any descriptor in area 4.n Write 0x0: Keep current mask of this interrupt source Read 0x0: This interrupt source is disabled (masked) Read 0x1: This interrupt source is enabled (unmasked) Write 0x1: Disable (mask) this interrupt source	RW W1toClr	0

**Table 15-75. Register Call Summary for Register DMM\_PAT\_IRQENABLE\_CLR**

## DMM Functional Description

- [DMM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)

## DMM Basic Programming Model

- [Error handling: \[32\] \[33\]](#)

## DMM Register Manual

- [DMM Register Summary: \[34\]](#)

**Table 15-76. DMM\_PAT\_STATUS\_i**

<b>Address Offset</b>	0x0000 04C0 + (0x4 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	Status register for each refill engine n = 0 for the first engine status register, n = 1 for the second engine status register...		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CNT								ERROR				RESERVED	BYPASSED	RESERVED	LINKED	DONE	RUN	VALID	READY				

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R	0x00
24:16	CNT	Counter of remaining lines to reload for engine n	R	0x000
15:10	ERROR	Error happened in engine n Read 0x0: No error Read 0x1: Invalid descriptor provided Read 0x2: Invalid data pointer provided Read 0x4: Unexpected area register update whilst refilling Read 0x8: Unexpected control register update whilst refilling Read 0x10: Unexpected data register update whilst refilling Read 0x20: Unexpected access to a yet-to-be-refilled location	R	0x00
9:8	RESERVED	Reserved	R	0x0
7	BYPASSED	Engine n is bypassed. Direct access to the LUT is provided.	R	0
6:5	RESERVED	Reserved	R	0x0
4	LINKED	Area reconfiguration link asserted for engine n	R	0
3	DONE	Area reloading finished for engine n	R	0
2	RUN	Area currently reloading for engine n	R	0
1	VALID	Valid area description for engine n	R	0
0	READY	Area registers ready for engine n	R	1

**Table 15-77. Register Call Summary for Register DMM\_PAT\_STATUS\_i**

DMM Functional Description

- [DMM Interrupt Requests: \[0\] \[1\]](#)
- [PAT Description: \[2\]](#)

DMM Use Cases and Tips

- [Simple manual area refill: \[3\] \[4\]](#)
- [Single auto-configured area refill: \[5\] \[6\]](#)
- [Chained auto-configured area refill: \[7\] \[8\] \[9\]](#)
- [Synchronised auto-configured area refill: \[10\] \[11\] \[12\]](#)
- [Cyclic synchronised auto-configured area refill: \[13\]](#)

DMM Register Manual

- [DMM Register Summary: \[14\]](#)

**Table 15-78. DMM\_PAT\_DESCR\_i**

<b>Address Offset</b>	0x0000 0500 + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	Physical address of the next table refill descriptor n = 0 for the descriptor register of the first engine, n = 1 for the descriptor register of the second engine... Writing to this register aborts the current ongoing area reload and resets the corresponding DMM_PAT_AREA_x, DMM_PAT_CTRL_x and DMM_PAT_DATA_x registers		
<b>Type</b>	RW		
31 30 29 28 27 26 25 24		23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8
		7 6 5 4	3 2 1 0
ADDR			
RESERVED			

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Physical address of the next table refill descriptor of engine n	RW WtoClr	0x0000000
3:0	RESERVED	Reserved	RW W0Only	0x0

**Table 15-79. Register Call Summary for Register DMM\_PAT\_DESCR\_i**

DMM Integration

- [DMM Integration: \[0\]](#)

DMM Functional Description

- [PAT Description: \[1\] \[2\] \[3\]](#)

DMM Use Cases and Tips

- [Single auto-configured area refill: \[4\]](#)
- [Chained auto-configured area refill: \[5\]](#)
- [Synchronised auto-configured area refill: \[6\]](#)
- [Cyclic synchronised auto-configured area refill: \[7\] \[8\]](#)

DMM Register Manual

- [DMM Register Summary: \[9\]](#)

**Table 15-80. DMM\_PAT\_AREA\_i**

<b>Address Offset</b>	0x0000 0504 + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	Area definition for DMM physical address translator n = 0 for the area register of the first engine, n = 1 for the area register of the second engine...		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	Y1							X1							RESERVED	Y0							X0								

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	RW W0Only	0
30:24	Y1	Y-coordinate of the bottom right corner of the PAT area for engine n	RW	0x00
23:16	X1	X-coordinate of the bottom right corner of the PAT area for engine n	RW	0x00
15	RESERVED	Reserved	RW W0Only	0
14:8	Y0	Y-coordinate of the top left corner of the PAT area for engine n	RW	0x00
7:0	X0	X-coordinate of the top left corner of the PAT area for engine n	RW	0x00

**Table 15-81. Register Call Summary for Register DMM\_PAT\_AREA\_i**

DMM Integration

- [DMM Integration: \[0\]](#)

DMM Functional Description

- [PAT Views: \[1\]](#)
- [PAT Description: \[2\] \[3\] \[4\]](#)

DMM Use Cases and Tips

- [Simple manual area refill: \[5\]](#)

**Table 15-81. Register Call Summary for Register DMM\_PAT\_AREA\_i (continued)**

- DMM Basic Programming Model
- [Configuration Settings and LUT refill: \[6\]](#)
  - [TILER page mapping: \[7\]](#)
- DMM Register Manual
- [DMM Register Summary: \[8\]](#)

**Table 15-82. DMM\_PAT\_CTRL\_i**

<b>Address Offset</b>	0x0000 0508 + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	DMM physical address translator control register n = 0 for the control register of the first engine, n = 1 for the control register of the second engine...		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITIATOR								RESERVED								SYNC	RESERVED								DIRECTION	RESERVED	START				

Bits	Field Name	Description	Type	Reset
31:28	INITIATOR	DMM PAT initiator for synchronisation in engine n	RW	0x0
27:17	RESERVED	Reserved	RW W0Only	0x000
16	SYNC	DMM PAT table reload synchronisation for engine n 0x0: Not synchronised 0x1: Synchronised	RW	0
15:7	RESERVED	Reserved	RW W0Only	0x000
6:4	DIRECTION	Direction of this PAT table refill for engine n	RW	0x0
3:1	RESERVED	Reserved	RW W0Only	0x0
0	START	Starting a PAT table refill with engine n	RW	0

**Table 15-83. Register Call Summary for Register DMM\_PAT\_CTRL\_i**

- DMM Integration
- [DMM Integration: \[0\]](#)
- DMM Functional Description
- [PAT Description: \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- DMM Use Cases and Tips
- [Simple manual area refill: \[6\] \[7\]](#)
  - [Cyclic synchronised auto-configured area refill: \[8\]](#)
- DMM Basic Programming Model
- [Configuration Settings and LUT refill: \[9\] \[10\]](#)
  - [Aliased tiled view orientation settings and LUT refill: \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- DMM Register Manual
- [DMM Register Summary: \[16\]](#)

**Table 15-84. DMM\_PAT\_DATA\_i**

<b>Address Offset</b>	0x0000 050C + (0x10 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	Physical address of the current table refill entry data n = 0 for the data register of the first engine, n = 1 for the data register of the second engine...		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	ADDR	Physical address of the current table refill entry data or single actual entry data when in manual mode for engine n	RW	0x00000000
3:0	RESERVED	Reserved	RW W0Only	0x0

**Table 15-85. Register Call Summary for Register DMM\_PAT\_DATA\_i**

## DMM Integration

- [DMM Integration: \[0\]](#)

## DMM Functional Description

- [PAT Views: \[1\]](#)
- [PAT Description: \[2\] \[3\] \[4\]](#)

## DMM Use Cases and Tips

- [Simple manual area refill: \[5\]](#)

## DMM Basic Programming Model

- [Configuration Settings and LUT refill: \[6\]](#)

## DMM Register Manual

- [DMM Register Summary: \[7\]](#)

**Table 15-86. DMM\_PEG\_HWINFO**

<b>Address Offset</b>	0x0000 0608	<b>Instance</b>	DMM
<b>Physical Address</b>	See <a href="#">Table 15-35</a>		
<b>Description</b>	DMM hardware configuration for PEG		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PRIO_CNT															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x00000000
6:0	PRIO_CNT	Number of PEG priority entries Read 0x1: One priority entry Read 0x2: Two priority entries Read 0x4: Four priority entries Read 0x8: Eight priority entries Read 0x10: Sixteen priority entries Read 0x20: Thirty-two priority entries Read 0x40: Sixty-four priority entries	R	0x40

**Table 15-87. Register Call Summary for Register DMM\_PEG\_HWINFO**

- DMM Integration
- [DMM Configuration: \[0\]](#)
- DMM Register Manual
- [DMM Register Summary: \[1\]](#)

**Table 15-88. DMM\_PEG\_PRIO\_k**

<b>Address Offset</b>	0x0000 0620 + (0x4 * k)	<b>Index</b>	k = 0 to 7
<b>Physical Address</b>	Please refer to <a href="#">Table 15-35</a>	<b>Instance</b>	DMM
<b>Description</b>	DMM PEG Priority register n = 0 for the first priority register, n = 1 for the second priority register...		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W7	P7			W6	P6			W5	P5			W4	P4			W3	P3			W2	P2			W1	P1			W0	P0		

Bits	Field Name	Description	Type	Reset
31	W7	Write-enable for P7 bit field Write 0x0: P7 field is unchanged Write 0x1: P7 field is updated	RW	0
30:28	P7	Priority for initiator 8.n+7	RW	0x4
27	W6	Write-enable for P6 bit field Write 0x0: P6 field is unchanged Write 0x1: P6 field is updated	RW	0
26:24	P6	Priority for initiator 8.n+6	RW	0x4
23	W5	Write-enable for P5 bit field Write 0x0: P5 field is unchanged Write 0x1: P5 field is updated	RW	0
22:20	P5	Priority for initiator 8.n+5	RW	0x4
19	W4	Write-enable for P4 bit field Write 0x0: P4 field is unchanged Write 0x1: P4 field is updated	RW	0
18:16	P4	Priority for initiator 8.n+4	RW	0x4
15	W3	Write-enable for P3 bit field Write 0x0: P3 field is unchanged Write 0x1: P3 field is updated	RW	0
14:12	P3	Priority for initiator 8.n+3	RW	0x4
11	W2	Write-enable for P2 bit field Write 0x0: P2 field is unchanged Write 0x1: P2 field is updated	RW	0
10:8	P2	Priority for initiator 8.n+2	RW	0x4
7	W1	Write-enable for P1 bit field Write 0x0: P1 field is unchanged Write 0x1: P1 field is updated	RW	0
6:4	P1	Priority for initiator 8.n+1	RW	0x4
3	W0	Write-enable for P0 bit field Write 0x0: P0 field is unchanged Write 0x1: P0 field is updated	RW	0
2:0	P0	Priority for initiator 8.n	RW	0x4

**Table 15-89. Register Call Summary for Register DMM\_PEG\_PRIO\_k**

DMM Integration	<ul style="list-style-type: none"> <li>• <a href="#">DMM Integration: [0]</a></li> </ul>
DMM Functional Description	<ul style="list-style-type: none"> <li>• <a href="#">PEG Description: [1]</a></li> </ul>
DMM Basic Programming Model	<ul style="list-style-type: none"> <li>• <a href="#">Priority settings: [2]</a></li> </ul>
DMM Register Manual	<ul style="list-style-type: none"> <li>• <a href="#">DMM Register Summary: [3]</a></li> </ul>

**Table 15-90. DMM\_PEG\_PRIO\_PAT**

<b>Address Offset</b>	0x0000 0640	<b>Instance</b>	DMM
<b>Physical Address</b>	See <a href="#">Table 15-35</a>		
<b>Description</b>	DMM PEG Priority register for the internal PAT engine.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																W_PAT		P_PAT													

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	RW W0Only	0x0000000
3	W_PAT	Write-enable for P_PAT bit field Write 0x0: P_PAT field is updated Write 0x1: P_PAT field is unchanged	RW	0
2:0	P_PAT	Priority for PAT engine	RW	0x4

**Table 15-91. Register Call Summary for Register DMM\_PEG\_PRIO\_PAT**

DMM Integration	<ul style="list-style-type: none"> <li>• <a href="#">DMM Integration: [0]</a></li> </ul>
DMM Functional Description	<ul style="list-style-type: none"> <li>• <a href="#">PEG Description: [1]</a></li> </ul>
DMM Basic Programming Model	<ul style="list-style-type: none"> <li>• <a href="#">Priority settings: [2] [3]</a></li> </ul>
DMM Register Manual	<ul style="list-style-type: none"> <li>• <a href="#">DMM Register Summary: [4]</a></li> </ul>



## 15.3 EMIF Controller

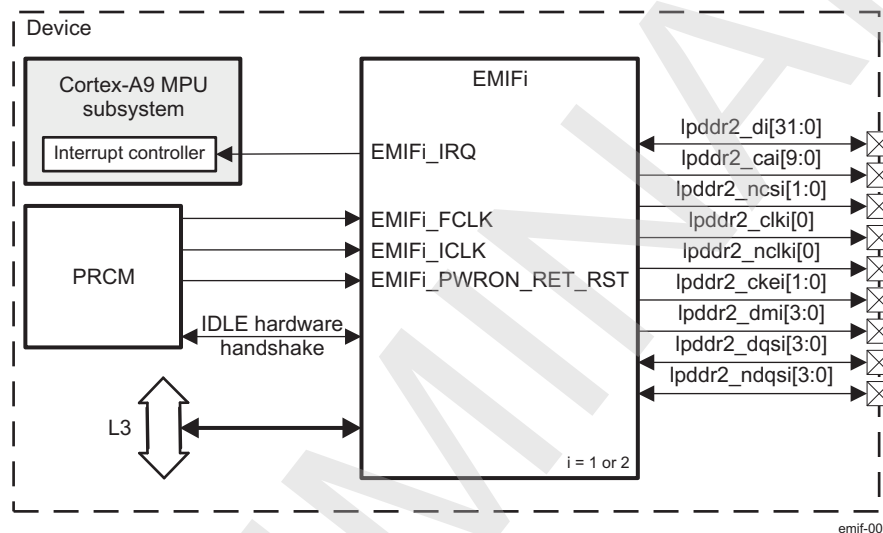
This section describes the features and functions of the external memory interface (EMIF) module of the device.

### 15.3.1 EMIF Module Overview

The EMIF module provides connectivity between the device and the LPDDR2-type memories and manages data bus read/write accesses between external memories, the microprocessor unit (MPU), and the direct memory access (DMA) controller.

The device includes two EMIF controllers. [Figure 15-43](#) shows an overview of the EMIF module.

**Figure 15-43. EMIF Module Overview**



#### 15.3.1.1 EMIF Module Main Features

The EMIF module supports the following features:

- JEDEC standard compliant LPDDR2-SDRAM (S2 and S4) and LPDDR2-NVM devices.
- Two GB SDRAM address range over two chip-selects (1GB per CS ) (configurable with the DMM module, see [Section 15.2, Dynamic Memory Manager](#) for more information)
- Two independent chip-selects with their corresponding register sets and independent page tracking
- Both chip-selects must have the same memory type and size if they are both SDRAM or both NVM. LPDDR2-SDRAM can be used in parallel with LPDDR2-NVM and can have a different size.
- Flexible address muxing scheme which permit to choose different bank mapping allocation by configuring the bank, column and row address decoding ordering
- 16 or -32-bit data path to external SDRAM memory
- LPDDR2 devices with 1, 2, 4 or 8 of internal banks
- Data bus widths:
  - 128-bit L3 interconnect Data Bus Width
  - 16 and 32-bit SDRAM Data Bus Width
- CAS latencies: 3, 4, 5, 6, 7 and 8
- 256, 512, 1024, and 2048-word page sizes
- Burst lengths: 8
- Sequential burst type
- SDRAM auto initialization from reset or configuration change
- Bank Interleaving across both the chip selects if same memory type
- Self Refresh and Precharge Power-down modes for low power (per chip select, manual entry/exit)

Self-Refresh and Deep Power-down modes)

- Deep Power-down mode for low power
- Partial Array Self Refresh and Temperature Controlled Self Refresh modes for low power. Temperature Controlled Self Refresh is only supported for mobile SDRAM having on-chip temperature sensor.
- Prioritized refresh
- Programmable SDRAM refresh rate and backlog counter
- Output impedance (ZQ) calibration
- Temperature monitoring

### 15.3.1.2 Not supported Features

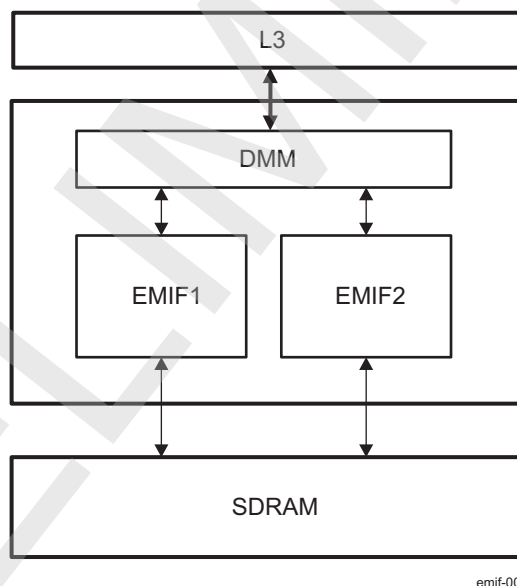
EMIF module does not support :

- Interleave burst type
- Auto precharge
- Active Power-down
- Per-bank refreshes for LPDDR2-SDRAM

### 15.3.1.3 EMIF Overall Architecture

Figure 15-44 shows the EMIF connection with the surrounding modules.

**Figure 15-44. EMIF Overall Architecture**

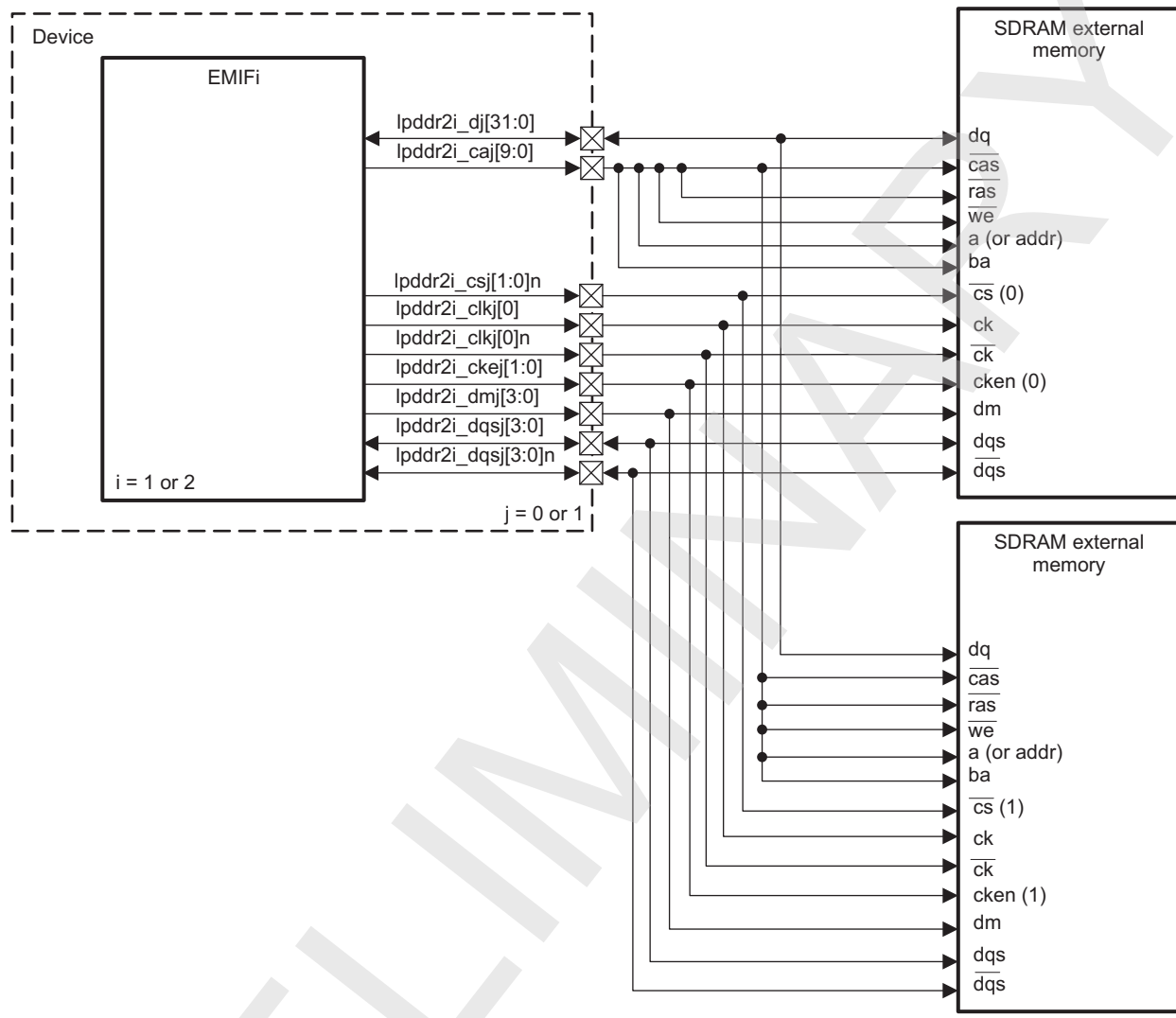


## 15.3.2 EMIF Environment

This section describes the EMIF application fields from an environment point of view (external connections).

Figure 15-45 shows an the EMIF generic configuration.

Figure 15-45. EMIF Generic Configuration



### 15.3.2.1 EMIF Signals

Table 15-92 describes the module signals and specifies their links to functions.

Table 15-92. I/O Signals

Signal	I/O <sup>(1)</sup>	Description
<b>EMIF1</b>		
$lpddr21\_dq[31:0]$	I/O	Data bus
$lpddr21\_ca[9:0]$	O	Double data rate command/address
$lpddr21\_ncs[1:0]$	O	Chip selects 1-0 (active low)
$lpddr21\_clk$	O	Clock
$lpddr21\_nclk$	O	Clock invert
$lpddr21\_cke[1:0]$	O	Clock enables 1-0
$lpddr21\_dm[3:0]$	I/O	Data mask
$lpddr21\_dqs[3:0]$	I/O	Data strobe
$lpddr21\_ndqs[3:0]$	I/O	Data strobe invert

<sup>(1)</sup> I = Input; O = Output

**Table 15-92. I/O Signals (continued)**

Signal	I/O <sup>(1)</sup>	Description
<b>EMIF2</b>		
lpddr22_dq[31:0]	I/O	Data bus
lpddr22_ca[9:0]	O	Double data rate command/address
lpddr22_ncs[1:0]	O	Chip selects 1-0 (active low)
lpddr22_clk	O	Clock
lpddr22_nclk	O	Clock invert
lpddr22_cke[1:0]	O	Clock enables 1-0
lpddr22_dm[3:0]	I/O	Data mask
lpddr22_dqs[3:0]	I/O	Data strobe
lpddr22_ndqs[3:0]	I/O	Data strobe invert

CKE memory pin is dynamically driven by the SDRAM controller according to the memory interface activity. Each chip select has its own CKE pin, so that each chip select can be independently controlled in terms of power-down mode, deep power-down mode and self refresh.

Each LPDDR2 channel, that is, each EMIF controller may be connected to two external LPDDR2 memories. See [Section 15.2, DMM](#) of the device for more information.

### 15.3.3 EMIF Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 15-46 shows the EMIF1 module integration in the device.

Figure 15-46. EMIF1 Integration

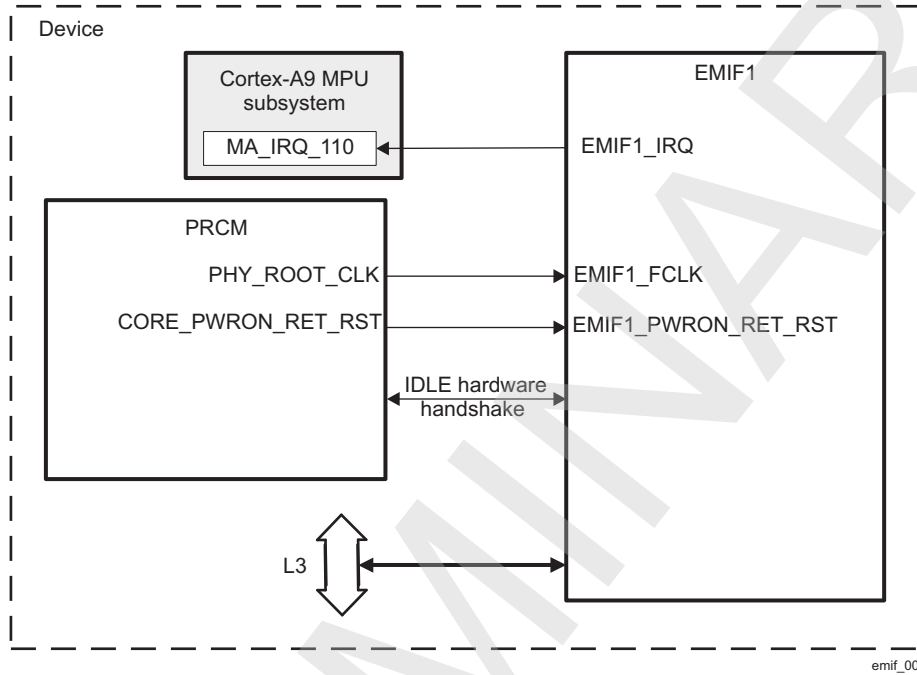
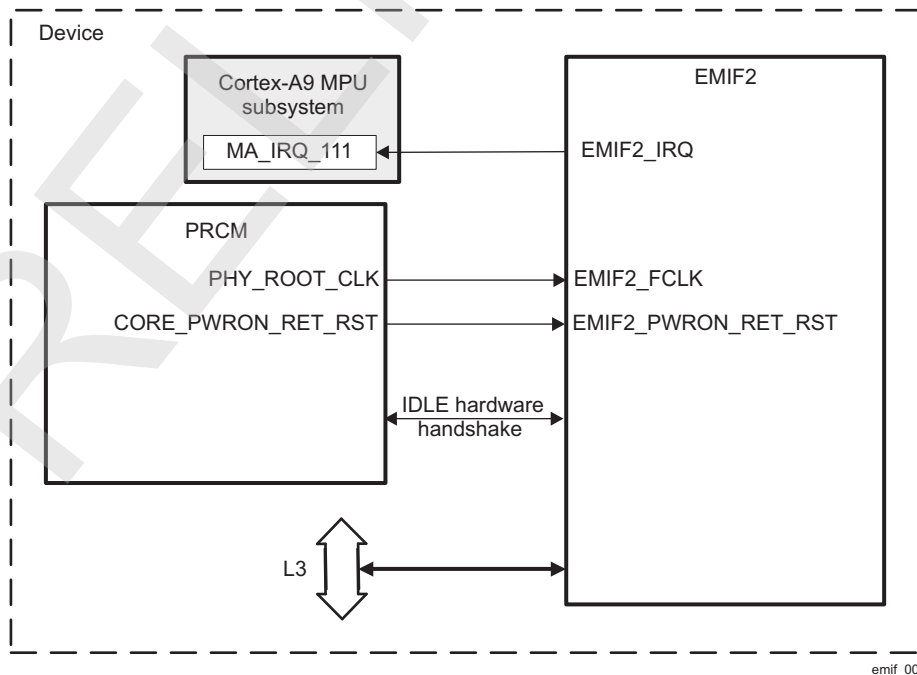


Figure 15-47 shows the EMIF2 module integration in the device.

Figure 15-47. EMIF2 Integration



**NOTE:** For more information about the IDLE hardware handshake, see [Chapter 3, Power, Reset, and Clock Management](#) of the device.

[Table 15-93](#) through [Table 15-95](#) summarize the integration of the module in the device.

**Table 15-93. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-up Capability	Interconnect
EMIF1	PD_CORE	No	L3
EMIF2	PD_CORE	No	L3

**Table 15-94. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
EMIF1	EMIF1_FCLK	PHY_ROOT_CLK	PRCM	Functional and interface clock
	EMIF1_DLL_CLK	DLL_CLK	PRCM	Functional and interface clock
EMIF2	EMIF2_FCLK	PHY_ROOT_CLK	PRCM	Functional and interface clock
	EMIF2_DLL_CLK	DLL_CLK	PRCM	Functional and interface clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
EMIF1	EMIF1_PWRON_RET_RST	CORE_PWRON_RET_RST	PRCM	retention reset
EMIF2	EMIF2_PWRON_RET_RST	CORE_PWRON_RET_RST	PRCM	retention reset

**Table 15-95. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
EMIF1	EMIF1_IRQ	MA_IRQ_110	Cortex-A9 INTC	Interrupt to the MPU Subsystem
EMIF2	EMIF2_IRQ	MA_IRQ_111	Cortex-A9 INTC	Interrupt to the MPU Subsystem
No DMA Requests				

**NOTE:** About interrupt source description, see [Section 15.3.4.5, Interrupt Requests](#).

### 15.3.4 EMIF Functional Description

#### 15.3.4.1 Block Diagram

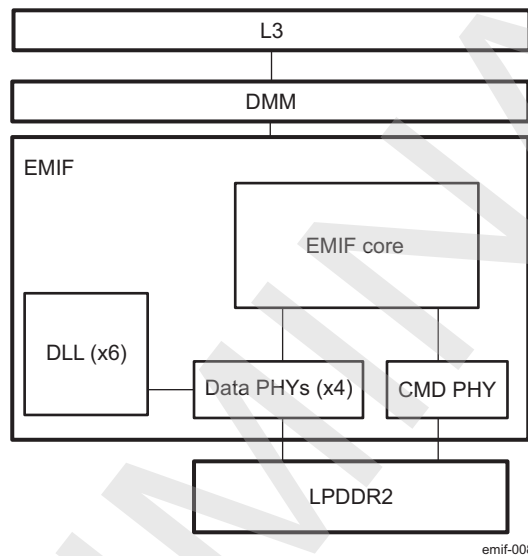
EMIF is a L3-bus peripheral providing an interface to LPDDR2 memories.

Figure 15-48 shows the interconnection between the EMIF module and the other modules.

DLLs are used to delay the input DQS signals during reads, so that these strobe signals can be used to latch incoming data on the DQ pins, as required by the LPDDR2 standard.

PHYS are hard macros that convert single data rate (SDR) signals to double data rate (DDR).

Figure 15-48. EMIF Block Diagram



##### 15.3.4.1.1 L3 Interface

EMIF supports two local interfaces - one that connects to the System Interconnect and the other that connects to a Low-latency Master. These interfaces are used to request all external memory device accesses, to access the EMIF registers, and to transfer all data to/from the EMIF controller. Table 15-96 shows the MAddrSpace mapping to different chip selects.

Table 15-96. MAddrSpace Mapping to Chip Selects

MAddrSpace	Chip Select	Description
0x0	lpddr2i_cs[1:0]n (i = 1 or 2)	SDRAM - which cs actually connects to a SDRAM or a NVM is software configurable.
0x1	lpddr2i_cs[1]n (i = 1 or 2)	LPDDR2-NVM - which cs actually connects to a SDRAM or a NVM is software configurable.
0x2	N/A	Reserved
0x3	N/A	Internal registers

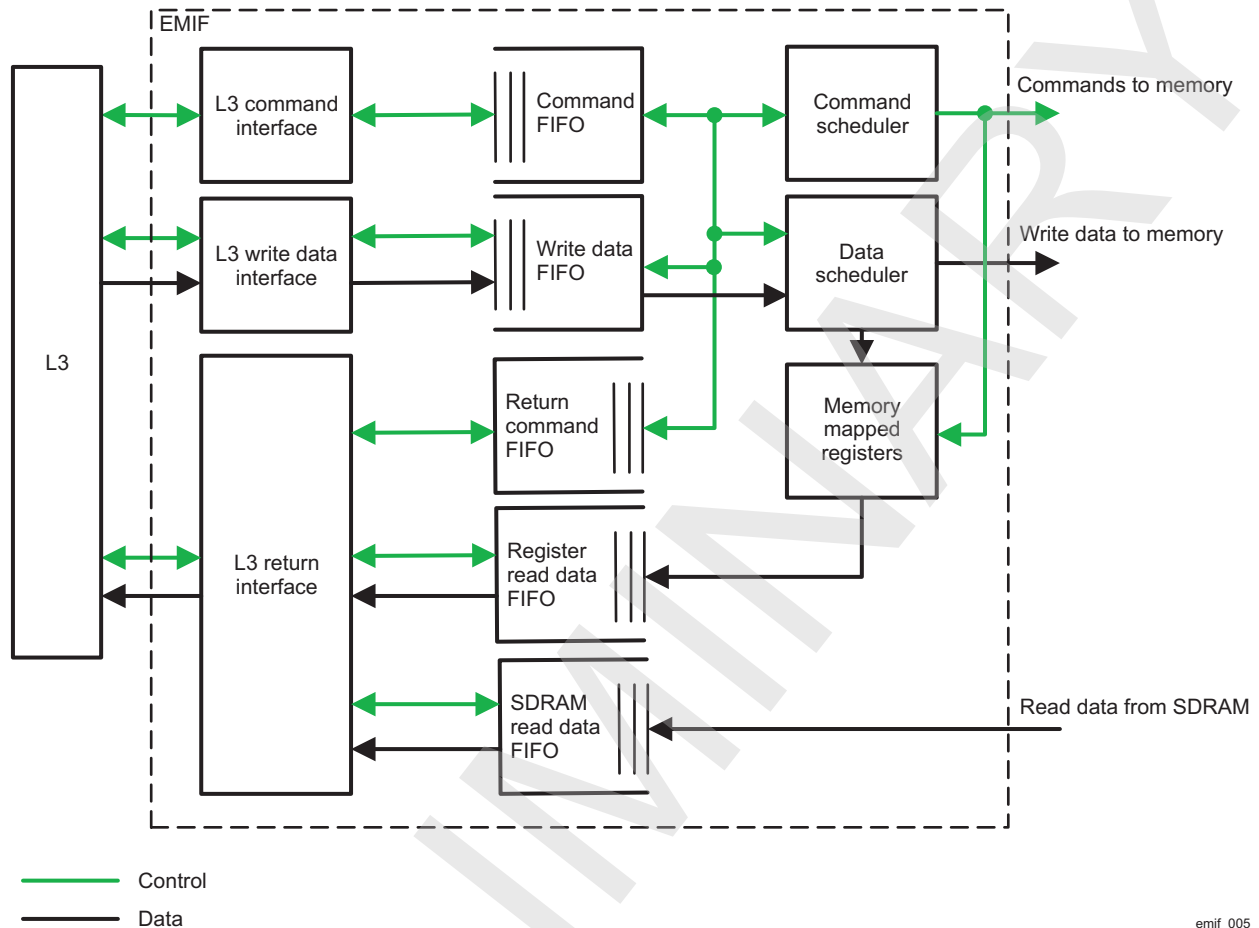
##### 15.3.4.1.2 FIFOs Description

EMIF contains the following FIFOs:

- Command FIFO
- Write Data FIFO
- Return Command FIFO
- Two Read Data FIFOs

Figure 15-49 shows the overall architecture of the EMIF module.

**Figure 15-49. FIFO Block Diagram**



emif\_005

The System (128-bit) L3 interface being the main interface has higher allocation of FIFO slots than the Low-latency (64-bit) L3 interface. The entries allocation is shown in [Table 15-97](#).

**Table 15-97. FIFO Allocation**

Parameter	System L3-Interface entries	Low-Latency L3-Interface entries
Command FIFO	Programmable <sup>(1)</sup>	Programmable <sup>(1)</sup>
Write Data FIFO	20	5
Return Command FIFO	22	17
SDRAM Read Data FIFO	22	17
Register read Data FIFO	2	2

<sup>(1)</sup> The total number of entries in the command FIFO is 10.

The command FIFO is shared between the two L3 interfaces, whereas there are two different FIFOs for each other type, one dedicated to each L3 interface.

The Command FIFO stores all the commands coming in on the L3 command interface. The allocation of entries in the Command FIFO is programmable per L3 interface using the [EMIF\\_L3\\_CONFIG\[27:24\]REG\\_SYS\\_THRESH\\_MAX](#) and [EMIF\\_L3\\_CONFIG\[19:16\]REG\\_LL\\_THRESH\\_MAX](#) bit fields.



If the addition of the [EMIF\\_L3\\_CONFIG\[27:24\]REG\\_SYS\\_THRESH\\_MAX](#) and [EMIF\\_L3\\_CONFIG\[19:16\]REG\\_LL\\_THRESH\\_MAX](#) fields is greater than the number of entries in the Command FIFO, the allocation would depend on the traffic on the interfaces. For example, if both the fields are programmed to 7, then either of the interfaces can use 7 entries in the Command FIFO if the other interface is idle. If both the interfaces have traffic, then the EMIF would start gradually allocating entries to both interfaces until it settles down to 7 entries for the low-latency interface and 3 for system interface since the low-latency interface has higher priority. If low-latency interface has no traffic, the system interface can use 7 entries.

The allocation would also depend on the traffic on the interfaces, if the [EMIF\\_L3\\_CONFIG\[27:24\]REG\\_SYS\\_THRESH\\_MAX](#) and [EMIF\\_L3\\_CONFIG\[19:16\]REG\\_LL\\_THRESH\\_MAX](#) fields are both programmed to be equal to or greater than the number of entries in the Command FIFO. For example, if both the fields are programmed to 10, then the low-latency port can use all the entries if there is no traffic on the system interface. However, if there is no traffic on the low-latency interface, the system interface can use all but one entry. This is because the low-latency interface has higher priority than the system interface. The only way for the system interface to use all command FIFO entries is to set the [EMIF\\_L3\\_CONFIG\[19:16\]REG\\_LL\\_THRESH\\_MAX](#) to zero. Writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1 for that interface.

The Write Data FIFO stores the write data for all the write transactions coming in on the L3 write data interface (when the Write Data FIFO is not full).

The Return Command FIFO stores all the return transactions that are to be issued to the L3 return interface. These include the write status return and the read data return commands.

There are two Read Data FIFOs that store the read data to be sent to the L3 return interface. One Read Data FIFO stores read data from the memory mapped registers. The other Read Data FIFO stores read data from external memory.

A write command is executed only when four L3 words of the corresponding data is received in the Write Data FIFO, or when it is the last data, and if there is space in the Return Command FIFO. The EMIF will schedule a read only if the results can fit into both the Return Command FIFO and the corresponding Read Data FIFOs.

### 15.3.4.1.3 Arbitration

The EMIF looks at all the commands stored in the Command FIFO to schedule commands to the external memory. All commands with the same MConnID on a particular L3 interface will complete in order. The EMIF does not guarantee ordering between commands with different MConnID or between commands from two L3 interfaces. For more information about L3 interconnect terms (MConnID, MCMD, MADDR, ...) see [Section 13.2, L3 Interconnect](#)

However, the EMIF will maintain data coherency. Therefore, the EMIF will block a command, regardless of priority or L3 interface, if that command is to the same block address (2048 bytes) as an older command which has not completed.

Thus, the EMIF might have one pending read or write for each MConnID. Among all pending reads, the EMIF selects all reads that have their corresponding SDRAM banks already open. Similarly, among all pending writes, the EMIF selects all writes that have their corresponding SDRAM banks already open. Accesses to memory mapped registers are treated as accesses that have open banks.

As a result of the above reordering, the EMIF might now have several pending reads and writes that have their corresponding banks open. The EMIF then selects the highest priority read from pending reads, and the highest priority write from pending writes. Commands from the low-latency L3 interface have higher priority over the commands from the system L3 interface. If two or more commands have the highest priority, the EMIF selects the oldest command. As a result, the EMIF might now have the next read and a write command. If the Return Command FIFO and the Read Data FIFO has space and the external bus conflict is resolved, the EMIF performs the final read command before the final write command. If the Return Command FIFO has space but the Read Data FIFO is full, the EMIF performs the final write command before the final read command.

Resolution of external bus conflict means all the SDRAM command to command counters are satisfied and the read to write or write to read turn around time is met.

The EMIF does not support tag interleaving. In other words, for a L3 interface, the EMIF completes executing a L3 command before it switches to another command. The EMIF can however interleave execution between commands from two L3 interfaces.

The data coherency inside the EMIF is only guaranteed in a single level of L3 infrastructure. For example, if a write from a secondary L3 bus segment is blocked by a bridge element, the read from a tertiary bus can still beat the write to the EMIF. In such a case, in order to confirm that a write from master A has landed before a read from master B is performed, master A must wait for the write status from the EMIF before indicating to master B that the data is ready to be read. If master A does not use the L3 wait status, it must perform the following sequence of operations:

1. Perform the required write
  2. Perform a dummy write to the [EMIF\\_MOD\\_ID\\_REV](#) register
  3. Perform a dummy read to the [EMIF\\_MOD\\_ID\\_REV](#) register
  4. Indicate to the master B that the data is ready to be read after completion of read in step 3. The completion of read in step 3 ensures that the previous writes were done
- Apart from reads and writes the EMIF also needs to open and close SDRAM banks, and maintain the refresh counts for a SDRAM. The priority of SDRAM commands with respect to refresh levels are as follows:

1. SDRAM Refresh request when Refresh-Must level is reached (highest priority)
2. L3 request for a read without a higher priority write
3. SDRAM Refresh request when Refresh-Need level is reached
4. L3 request for a write
5. SDRAM Activate commands
6. SDRAM Deactivate commands
7. SDRAM Deep Power-Down request
8. SDRAM clock stop request
9. SDRAM Refresh request when Refresh-May level is reached
10. SDRAM Self-refresh or SDRAM Power-Down request (lowest priority).

While performing the above listed scheduling algorithm, the EMIF might run into following two conditions:

1. A continuous stream of high priority commands can block lower priority commands
2. A continuous stream of SDRAM commands to a row in an open bank can block commands to another row in the same bank

To avoid a continuous blocking effect, the EMIF momentarily raises the priority of the oldest command over all other commands when the number of memory accesses equals to the value programmed in the [EMIF\\_L3\\_CONFIG\[7:0\] REG\\_PR\\_OLD\\_COUNT](#) field.

On top of the above scheduling, the highest priority condition is a reset command. If this condition occurs, the EMIF abandons whatever it is currently doing and begins its start-up sequence. In this case, commands and data stored in the FIFOs are lost. The EMIF also starts its start-up sequence whenever the [EMIF\\_SDRAM\\_CONFIG](#) is written and [EMIF\\_SDRAM\\_REF\\_CTRL\[31\] REG\\_INITREF\\_DIS](#) is set to 0. In this case, commands and data stored in the FIFOs are not lost. The EMIF will ensure that in-flight read or write transactions to the SDRAM are complete before starting the initialization sequence.

All the accesses to a SDRAM are pipelined to maximize the external bus utilization. All of these are done while fulfilling the access timing requirements of a SDRAM.

#### 15.3.4.2 Clock Management

The EMIF can gate off the EMIF\_FCLK. There is an internal mechanism which can stop the EMIF\_FCLK automatically.

EMIF\_FCLK can only be stopped after the SDRAM is put into Self Refresh Mode and the power-idle protocol on the L3 bus is complete.

The EMIF waits for the DLL lock before performing any memory access.

The EMIF\_FCLK frequency can be changed only after putting the external SDRAM in self-refresh mode.

EMIF<sub>i</sub>\_FCLK is divided inside the EMIF module: DDR clock = EMIF<sub>i</sub>\_FCLK / 4.

### 15.3.4.3 Software Reset

**EMIF does not support a software reset.**

EMIF supports the **global warm reset mode** during which the EMIF keeps the the SDRAM content. Upon a request from the PRCM module indicating a need to enter global warm reset mode, the EMIF executes the following :

- completes the on-going access and then puts the SDRAM in Self Refresh mode
- clears all its FIFO contents and deasserts sys\_SCmdAccept and sys\_SDataAccept ports.

When the global warm reset has to terminate :

- The PRCM block deasserts its request signal to the EMIF
- EMIF will exit Self Refresh state once it receives host accesses from the MPU to the external memory.

### 15.3.4.4 System Power Management

#### 15.3.4.4.1 Read Idle Window

The [EMIF\\_READ\\_IDLE\\_CTRL](#) register allows the ability to force idle time between consecutive reads to allow the DDR PHY to recalibrate the DLL delays during gradual voltage and frequency change.

The [EMIF\\_READ\\_IDLE\\_CTRL\[8:0\] REG\\_READ\\_IDLE\\_INTERVAL](#) field determines the maximum interval  $((\text{REG\_READ\_IDLE\_INTERVAL} - 1) * 64 \text{ clock cycles})$  between read idle detections or force read idle time. A value of zero disables the read idle function.

The [EMIF\\_READ\\_IDLE\\_CTRL\[19:16\] REG\\_READ\\_IDLE\\_LEN](#) field determines the minimum size of idle window for the read idle detection as well as the force read idle time. The EMIF will only introduce idle time if it does not see at least  $\text{REG\_READ\_IDLE\_LEN} - 1$  number of idle cycles in  $\text{REG\_READ\_IDLE\_INTERVAL}$ . Any command other than a read command is considered idle for this purpose. Force read idle occurs when the data bus is busy transferring read data and the read idle interval expires.

#### 15.3.4.4.2 LPDDR2-SDRAM Power-Down Mode

The EMIF supports **power-down mode** for low power. The EMIF automatically puts the SDRAM into power-down after the EMIF is idle for [EMIF\\_PWR\\_MGMT\\_CTRL\[3:0\] REG\\_CS\\_TIM](#) number of EMIF\_FCLK cycles and the [EMIF\\_PWR\\_MGMT\\_CTRL\[10:8\] REG\\_LP\\_MODE](#) field is set to 0x3. In power-down mode, the EMIF does not stop the clocks to the SDRAM. The EMIF maintains  $\text{lpddr2i\_cke}j[1:0]$  ( $i = 1 \text{ or } 2 \text{ and } j = 0 \text{ or } 1$ ) low to maintain the power-down state.

When the SDRAM is in power-down, the EMIF services register accesses as normal. The EMIF also services LPDDR2-NVM accesses as normal provided when it is not in power-down (in the case of the SDRAM memories are connected to the same EMIF through two different chip selects).

If the the LPDDR2-SDRAM is in **power-down mode** and :

1. [EMIF\\_PWR\\_MGMT\\_CTRL\[10:8\] REG\\_LP\\_MODE](#) field is changed from 0x3 to some other value or
  2. A SDRAM access is requested or
  3. The Refresh Must level is reached
- then the EMIF will bring the SDRAM out of power-down mode.

For LPDDR2-SDRAM, the EMIF module:

1. Drives  $\text{lpddr2i\_cke}j[1:0]$  ( $i = 1 \text{ or } 2 \text{ and } j = 0 \text{ or } 1$ ) high
2. Waits for [EMIF\\_SDRAM\\_TIM\\_2\[30:28\] REG\\_T\\_XP + 1](#) cycles
3. Enters its **idle state** and can issue any commands.

#### 15.3.4.4.3 LPDDR2-SDRAM Deep Power-Down Mode

For ultimate power savings, the EMIF supports deep power-down mode for LPDDR2-SDRAM.

The SDRAM can be forced into **deep power-down** state through software by setting the [EMIF\\_PWR\\_MGMT\\_CTRL\[11\] REG\\_DPD\\_EN](#) field to 0x1. In this case, the EMIF will continue normal operation until all SDRAM memory access requests have been serviced. At this point the EMIF will issue a DEEP POWER-DOWN command. The EMIF then maintains `lpddr2i_cke[j][1:0]` ( $i = 1$  or  $2$  and  $j = 0$  or  $1$ ) low to maintain the deep power-down state.

Setting the [EMIF\\_PWR\\_MGMT\\_CTRL\[11\] REG\\_DPD\\_EN](#) field to 1 overrides the setting of [EMIF\\_PWR\\_MGMT\\_CTRL\[10:8\] REG\\_LP\\_MODE](#) field. Therefore, if the SDRAM is in Clock Stop, Self Refresh, or power-down mode, and [EMIF\\_PWR\\_MGMT\\_CTRL\[11\] REG\\_DPD\\_EN](#) field is set to 1, the EMIF will exit those modes and go into deep power-down mode.

When the SDRAM is in deep power-down, the EMIF services register accesses as normal. The EMIF also services LPDDR2-NVM accesses as normal provided it is not in power-down (in the case of a SDRAM and a NVM memories are connected to the same EMIF through two different chip selects).

If the [EMIF\\_PWR\\_MGMT\\_CTRL\[11\] REG\\_DPD\\_EN](#) is set to 0x0, the EMIF will bring the SDRAM out of Deep Power-down state.

For LPDDR2-SDRAM, the EMIF controller:

1. Drives `lpddr2i_cke[j][1:0]` ( $i = 1$  or  $2$  and  $j = 0$  or  $1$ ) high
2. Issues a RESET command to LPDDR2 (MRW)
3. Enters its **idle state** and can issue any commands.
4. Software will have to perform initialization as specified in the LPDDR2 Initialization section, see [Section 15.3.4.13](#)

#### 15.3.4.4.4 LPDDR2-NVM Power-Down Mode

The EMIF supports **power-down mode** for LPDDR2-NVM. The EMIF automatically puts the LPDDR2-NVM into power-down after the EMIF is idle for [EMIF\\_PWR\\_MGMT\\_CTRL\[3:0\] REG\\_CS\\_TIM](#) number of EMIF\_FCLK cycles and the [EMIF\\_PWR\\_MGMT\\_CTRL\[10:8\] REG\\_LP\\_MODE](#) field is set to 0x4. The EMIF will maintain `lpddr2i_cke[j][1:0]` ( $i = 1$  or  $2$  and  $j = 0$  or  $1$ ) low to maintain the power-down state.

When LPDDR2-NVM is in power-down, the EMIF services register accesses as normal. The EMIF also services SDRAM accesses as normally provided it is not in self-refresh or power-down (in the case of the NVM memories are connected to the same EMIF through two different chip selects).

When the LPDDR2-NVM is in **power-down mode** and :

1. [EMIF\\_PWR\\_MGMT\\_CTRL\[10:8\] REG\\_LP\\_MODE](#) field is changed from 0x4 to some other value or
  2. A LPDDR2-NVM access is requested or
- then the EMIF will bring the LPDDR2-NVM out of power-down mode.

For LPDDR2-NVM, the EMIF module:

1. Drives `lpddr2i_cke[j][1:0]` ( $i = 1$  or  $2$  and  $j = 0$  or  $1$ ) high
2. Waits for [EMIF\\_SDRAM\\_TIM\\_2\[30:28\] REG\\_T\\_XP + 1](#) cycles
3. Enters its **idle state** and can issue any commands

#### 15.3.4.4.5 LPDDR2-SDRAM Self-Refresh Mode

The EMIF supports **self-refresh mode** for low power. The EMIF automatically puts the SDRAM into Self-refresh after the EMIF is idle for [EMIF\\_PWR\\_MGMT\\_CTRL\[3:0\] REG\\_CS\\_TIM](#) number of EMIF\_FCLK cycles and the [EMIF\\_PWR\\_MGMT\\_CTRL\[10:8\] REG\\_LP\\_MODE](#) field is set to 0x4. In Self-Refresh mode, the EMIF automatically stops the clocks to the SDRAM. The EMIF maintains `lpddr2i_cke[j][1:0]` ( $i = 1$  or  $2$  and  $j = 0$  or  $1$ ) low to maintain the self-refresh state.

When the SDRAM is in self-refresh, the EMIF services register accesses as normal. The EMIF also services LPDDR2-NVM accesses as normal provided it is not in power-down (in the case of a SDRAM and a NVM memories are connected to the same EMIF through two different chip selects).

If the the LPDDR2-SDRAM is in **self refresh mode** and :

1. [EMIF\\_PWR\\_MGMT\\_CTRL\[10:8\] REG\\_LP\\_MODE](#) field is changed from 0x4 to some other value or
2. A SDRAM access is requested or

3. [EMIF\\_SDRAM\\_TIM\\_2\[2:0\]](#) REG\_T\_CKE + 1 cycles have elapsed since the SELF-REFRESH command was issued then the EMIF will bring the SDRAM out of self-refresh mode.

For LPDDR2-SDRAM, the EMIF:

1. Enables clocks
2. Drives `lpddr2i_cke[j][1:0]` (i = 1 or 2 and j = 0 or 1) high
3. Waits for [EMIF\\_SDRAM\\_TIM\\_2\[24:16\]](#) REG\_T\_XSNR + 1 cycles
4. Starts an auto-refresh cycle in the next cycle
5. Enters its idle state and can issue any commands

#### 15.3.4.4.6 Save and Restore

The EMIF supports **save and restore mechanism** to completely switch off power to the EMIF.

See [Chapter 3, Power, Reset, and Clock Management](#) of the device for more information on *off-mode sequence* and *restore power to the EMIF sequence* of operations.

#### 15.3.4.4.7 Retention Mode

EMIF supports **full-retention mode**.

#### 15.3.4.5 Interrupt Requests

The EMIF sources two interrupts: one for the System L3 interface and the other for the Low-latency L3 interface. The pulse and level interrupt signals for a particular interface are both sourced from the same raw interrupt internal to the module.

The EMIF will set the [EMIF\\_IRQSTATUS\\_RAW\\_SYS\[0\]](#) REG\_RAW\_SYS / [EMIF\\_IRQSTATUS\\_RAW\\_LL\[0\]](#) REG\_RAW\_LL bit (depending on which interface sent the errored command) to 1, if an access request to an unsupported MAddrSpace is received. If such an error occurs, it is due to a bad programming of the DMM. See [Section 15.2, Dynamic Memory Manager](#) for more information on addressing.

The interrupts can be cleared once serviced by writing a 1 to the corresponding bits either in the [EMIF\\_IRQSTATUS\\_RAW\\_SYS](#) / [EMIF\\_IRQSTATUS\\_RAW\\_LL](#) or [EMIF\\_IRQSTATUS\\_SYS](#) / [EMIF\\_IRQSTATUS\\_LL](#) registers.

[Table 15-98](#) lists the event flags, and their mask, that can cause module interrupts.

**Table 15-98. Events**

Event Flag	Event Mask	Description
<a href="#">EMIF_IRQSTATUS_RAW_SYS[1]</a> REG_TA_SYS / <a href="#">EMIF_IRQSTATUS_SYS[1]</a> REG_TA_SYS	<a href="#">EMIF_IRQENABLE_SET_SYS[1]</a> REG_EN_TA_SYS / <a href="#">EMIF_IRQENABLE_CLR_SYS[1]</a> REG_EN_TA_SYS	System L3 interrupt for SDRAM temperature alert.
<a href="#">EMIF_IRQSTATUS_RAW_LL[1]</a> REG_TA_LL / <a href="#">EMIF_IRQSTATUS_LL[1]</a> REG_TA_LL	<a href="#">EMIF_IRQENABLE_SET_LL[1]</a> REG_EN_TA_LL / <a href="#">EMIF_IRQENABLE_CLR_LL[1]</a> REG_EN_TA_LL	Low-latency L3 interrupt for SDRAM temperature alert.
<a href="#">EMIF_IRQSTATUS_RAW_SYS[0]</a> REG_ERR_SYS / <a href="#">EMIF_IRQSTATUS_SYS[0]</a> REG_ERR_SYS	<a href="#">EMIF_IRQENABLE_SET_SYS[0]</a> REG_EN_ERR_SYS / <a href="#">EMIF_IRQENABLE_CLR_SYS[0]</a> REG_EN_ERR_SYS	System L3 interrupt for command or address error.
<a href="#">EMIF_IRQSTATUS_RAW_LL[0]</a> REG_ERR_LL / <a href="#">EMIF_IRQSTATUS_LL[0]</a> REG_ERR_LL	<a href="#">EMIF_IRQENABLE_SET_LL[0]</a> REG_EN_ERR_LL / <a href="#">EMIF_IRQENABLE_CLR_LL[0]</a> REG_EN_ERR_LL	Low-latency L3 interrupt for command or address error.

### 15.3.4.6 SDRAM Output Impedance Calibration

The EMIF supports automatic output impedance calibration. The ZQ calibration can be enabled per chip select by setting [EMIF\\_ZQ\\_CONFIG\[31\] REG\\_ZQ\\_CS1EN](#) and [EMIF\\_ZQ\\_CONFIG\[30\] REG\\_ZQ\\_CS0EN](#) bits. The EMIF supports following three types of ZQ calibration commands:

- ZQINIT - ZQ calibration command during initialization
- ZQCS - ZQ calibration short command
- ZQCL - ZQ calibration long command

The EMIF periodically issues a ZQCS command every time [EMIF\\_ZQ\\_CONFIG\[15:0\] REG\\_ZQ\\_REFINTERVAL](#) expires. In other words, the [REG\\_ZQ\\_REFINTERVAL](#) defines the interval between ZQCS commands. The EMIF will wait and block any other command for ([EMIF\\_SDRAM\\_TIM\\_3\[20:15\] REG\\_ZQ\\_ZQCS+1](#)) number of clock DDR clock cycles every time a ZQCS command is issued.

If [EMIF\\_ZQ\\_CONFIG\[28\] REG\\_ZQ\\_SFEXITEN](#) field is set to a 1, the EMIF will issue a ZQCL command every time it exits Self-Refresh, Active Power-Down, and Precharge Power-Down. The EMIF will wait and block any other command for ([EMIF\\_ZQ\\_CONFIG\[17:16\] REG\\_ZQ\\_ZQCL\\_MULT+1](#))\*([EMIF\\_SDRAM\\_TIM\\_3\[20:15\] REG\\_ZQ\\_ZQCS+1](#)) number of clock DDR clock cycles every time a ZQCL command is issued.

If a separate calibration resistor is used per device, the ZQ calibration can be performed simultaneously over both the chip selects. To enable ZQ calibration to be performed simultaneously over both chip selects, the [EMIF\\_ZQ\\_CONFIG\[27\] REG\\_ZQ\\_DUALCALEN](#) field must be set to a 1. If [REG\\_ZQ\\_DUALCALEN](#) is set to a 0, the EMIF will perform ZQ calibration per chip select serially.

### 15.3.4.7 SDRAM Temperature Monitoring

The EMIF supports automatic temperature monitoring to facilitate the software to update the refresh rate according to the LPDDR2 device temperature changes. The temperature monitoring can be enabled per chip select by setting [EMIF\\_TEMP\\_ALERT\\_CONFIG\[31\] REG\\_TA\\_CS1EN](#) and [EMIF\\_TEMP\\_ALERT\\_CONFIG\[30\] REG\\_TA\\_CS0EN](#) bits.

The EMIF periodically polls the temperature of LPDDR2 (issues a MRR command to Mode Register 4) every time [EMIF\\_TEMP\\_ALERT\\_CONFIG\[21:0\] REG\\_TA\\_REFINTERVAL](#) expires. In other words, the [REG\\_TA\\_REFINTERVAL](#) defines the interval between temperature alert polls. If the EMIF sees a 1 on bit 7 of the read data value from MRR, indicating that the temperature has changed, the [EMIF\\_IRQSTATUS\\_SYS\[1\] REG\\_TA\\_SYS](#) bit (for L3 interconnect), respectively, the [EMIF\\_IRQSTATUS\\_LL\[1\] REG\\_TA\\_LL](#) bit for the Low-Latency L3 connected LPDDR2-NVM is set and the module sends an interrupt both on the system and low-latency interrupt lines. On receipt of the interrupt, the software will update the [EMIF\\_SDRAM\\_REF\\_CTRL\[15:0\] REG\\_REFRESH\\_RATE](#) field to the required value as per the temperature change.

If [EMIF\\_TEMP\\_ALERT\\_CONFIG\[28\] REG\\_TA\\_SFEXITEN](#) field is set to a 1, the EMIF will poll for temperature change every time it exits Self-Refresh, Active Power-Down, and Precharge Power-Down.

Since the EMIF is performing a MRR, it needs information on how the LPDDR2 are connected. The [EMIF\\_TEMP\\_ALERT\\_CONFIG\[27:26\] REG\\_TA\\_DEVWDT](#) and [EMIF\\_TEMP\\_ALERT\\_CONFIG\[25:24\] REG\\_TA\\_DEVCNT](#) bits provides the necessary information to the EMIF for MRR data compare. For example, if [REG\\_TA\\_DEVWDT](#) is set to 0 indicating 8-bit devices used, and if [REG\\_TA\\_DEVCNT](#) is set to 2 indicating four devices used to form a 32-bit bus, the mask used for checking would be 4'b1111, that is, the EMIF would expect data on each of the byte lanes on a 32-bit bus.

### 15.3.4.8 SDRAM Refresh Scheduling

The EMIF uses two counters to schedule AUTO REFRESH commands: a 13-bit decrementing **refresh interval counter** and a 4-bit refresh **backlog counter**. The *interval counter* is used to define the rate at which connected SDRAM devices will be refreshed. It is loaded with the [EMIF\\_SDRAM\\_REF\\_CTRL\[15:0\] REG\\_REFRESH\\_RATE](#) field value at reset (only the 13 LSB bits are taken). The interval counter decrements by one each cycle until it reaches zero at which point it reloads from [EMIF\\_SDRAM\\_REF\\_CTRL\[15:0\] REG\\_REFRESH\\_RATE](#) and restarts decrementing. The counter also reloads and restarts decrementing whenever the [EMIF\\_SDRAM\\_REF\\_CTRL\[15:0\] REG\\_REFRESH\\_RATE](#) field is updated.

The refresh *backlog counter* records the number of AUTO REFRESH commands the EMIF currently has outstanding. The backlog counter increments by one each time the interval counter reloads (unless it has reached its maximum value of 15). The backlog counter decrements by one each time the EMIF issues an AUTO REFRESH command (unless it is already at zero). For the range of values that the backlog counter can take, there are four levels of urgency with which the EMIF should perform an auto refresh cycle (in which it issues AUTO REFRESH commands), as follows:

- Refresh May level is reached whenever the backlog count is greater than 0, to indicate that there is a refresh backlog, so if the EMIF is not busy and none of the SDRAM banks are open, it should perform an auto refresh cycle.
- Refresh Release level is reached whenever the backlog count is greater than 4, to indicate that the refresh backlog is getting high, so if the EMIF is not busy it should perform an auto refresh cycle even if any banks are open.
- Refresh Need level is reached whenever the backlog count is greater than 5, to indicate that the refresh backlog is getting high and the EMIF should raise the priority of performing an auto refresh cycle above that of servicing new memory write requests. The EMIF starts servicing new memory accesses after Refresh Release level is cleared.
- Refresh Must level is reached whenever the backlog count is greater than 7, to indicate that the refresh backlog is getting excessive and the EMIF should perform an auto refresh cycle before servicing any new memory access requests. The EMIF starts servicing new memory accesses after Refresh Release level is cleared.

The refresh counters do not operate when SDRAM has been put into self-refresh mode.

The EMIF issues AUTO REFRESH commands within auto refresh cycles. An auto refresh cycle consists of issuing an AUTO REFRESH command and waiting T\_RFC (refer to the SDRAM Timing 3 register, [EMIF\\_SDRAM\\_TIM\\_3](#)) cycles before rechecking the refresh levels. If the Refresh Release level is not reached, the EMIF starts another auto refresh cycle, otherwise it returns to the idle state where it can issue any command.

#### 15.3.4.9 EMIF Access Cycles

By default the EMIF keeps SDRAM chip selects active. To direct a command to just one of the SDRAMs it deasserts the chip select to the other SDRAM for the duration of the command. If [EMIF\\_SDRAM\\_CONFIG\[3\]](#) REG\_EBANK field in the SDRAM Config register is set to 0, chip select 1 will always be driven high except during initialization and for refresh, Power Down, Self Refresh, and Deep Power Down commands.

The EMIF always performs burst accesses to SDRAM. Multiple SDRAM bursts may be needed to service a single L3 burst request. [Table 15-102](#) to [Table 15-103](#) show a few examples on how EMIF accesses SDRAM for a linear incrementing transaction type. T0, T1, etc. are clock cycles. R0 is read starting at column 0, R8 is read starting at column 8, and R16 is read starting at column 16. D0-1 is the data from column 0 and 1, D2-3 is the data from column 2 and 3, and so on.

**Table 15-99. 64-Byte Linear Read Starting at Address 0x0**

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
R0				R8							
				D0-1	D2-3	D4-5	D6-7	D8-9	D10-11	D12-13	D14-15

**Table 15-100. 64-Byte Linear Read Starting at Address 0x8 (LPDDR2-S2)**

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
R2			R8				R16							
				D2-3	D4-5	D6-7	D8-9	D10-11	D12-13	D14-15	D16-17	Unuse d	Unuse d	Unuse d

**Table 15-101. 64-Byte Linear Read Starting at Address 0x8 (LPDDR2-S4)**

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
R2				R8				R16							
			D2-3	D4-5	D6-7	Unuse d	D8-9	D10- 11	D12- 13	D14- 15	D16- 17	Unuse d	Unuse d	Unuse d	

**Table 15-102. 64-Byte Linear Read Starting at Address 0x10**

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
R4		R8						R16					
			D4-5	D6-7	D8-9	D10-11	D12-13	D14-15	D16-17	D18-19	Unused	Unused	

**Table 15-103. 64-Byte Linear Read Starting at Address 0x18**

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
R6		R8						R16					
			D6-7	Unused	D8-9	D10-11	D12-13	D14-15	D16-17	D18-19	D20-21	Unused	

EMIF will use the unused data phases in the above figure by issuing successive read commands if there are reads to open banks pending in the command FIFO.

The write data conversion from single data rate to double data rate is done outside the EMIF.

#### 15.3.4.9.1 SDRAM Read

A read access to a SDRAM is initiated with a READ command. The EMIF provides the internal bank address and the column address on `pad_ca_o` with the READ command. The EMIF registers the first read data from `lpddr2i_dj` ( $i = 1$  or  $2$  and  $j = 0$  or  $1$ ) after  $(\text{read\_latency} + 3)$  DDR clock cycles. Read latency is programmed in the PHY Control 1 register, [EMIF\\_DDR\\_PHY\\_CTRL\\_1](#).

#### 15.3.4.9.2 SDRAM Write

A write access to a SDRAM is initiated with a WRITE command. The EMIF provides the internal bank address and the column address on `pad_ca_o` with the WRITE command.

EMIF delays the write data by the appropriate write latency minus one. Subsequent data words are provided on `lpddr2i_dj` ( $i = 1$  or  $2$  and  $j = 0$  or  $1$ ) along with the appropriate data mask on `lpddr2i_dmj` ( $i = 1$  or  $2$  and  $j = 0$  or  $1$ ) on each successive clock cycles. The DDR PHY then adds one more clock cycle delay to the write data to appropriately center the write data to the outgoing DQS.

#### 15.3.4.10 Turn Around Time

[Table 15-104](#) shows the turn around time that the EMIF introduces on the data bus for various back-to-back accesses. The EMIF takes advantage of the CAS latencies and packs the commands as close as possible on the control bus to introduce the following turn around time on the data bus.

**Table 15-104. Turn Around Time**

Previous Access	Next Access	Turn Around Time (number of DDR clock cycles)
SDRAM read	SDRAM read to same chip select	0
SDRAM read	SDRAM read to different chip select	2
SDRAM read	SDRAM write	2
SDRAM write	SDRAM write	0
SDRAM write	SDRAM read	2



### 15.3.4.11 SDRAM Address Mapping

The EMIF interleaves the internal banks for SDRAM connected to both the chip selects (see [Table 15-105](#) for the SDRAM addressing space), according to the DMM (see [Section 15.2, Dynamic Memory Manager](#) for more information). From the system point of view, the external SDRAM is seen as one block of SDRAM. If two external 64 MB devices are used, a 128 MB memory block is observed. If two external 32 MB devices are used, a 64 MB block is observed.

**Table 15-105. SDRAM Addressing Space**

Module Name	Base Address	Size
EMIF1-CS0-SDRAM	0x8000 0000	0 to 1 GB - programmable in DMM (see <a href="#">Section 15.2, Dynamic Memory Manager</a> )
EMIF2-CS0-SDRAM	0x8000 0000	0 to 1 GB - programmable in DMM (see <a href="#">Section 15.2, Dynamic Memory Manager</a> )
EMIF1-CS1-SDRAM	0xC000 0000	0 to 1 GB - programmable in DMM (see <a href="#">Section 15.2, Dynamic Memory Manager</a> )
EMIF2-CS1-SDRAM	0xC000 0000	0 to 1 GB - programmable in DMM (see <a href="#">Section 15.2, Dynamic Memory Manager</a> )

When addressing SDRAM, if the [EMIF\\_SDRAM\\_CONFIG\[28:27\] REG\\_IBANK\\_POS](#) field is set to 0, the EMIF uses the following 3-bit fields, to determine the mapping from source address to SDRAM row, column, bank and chip select:

- [EMIF\\_SDRAM\\_CONFIG\[6:4\] REG\\_IBANK](#)
- [EMIF\\_SDRAM\\_CONFIG\[3\] REG\\_EBANK](#)
- [EMIF\\_SDRAM\\_CONFIG\[2:0\] REG\\_PAGESIZE](#)

If the [EMIF\\_SDRAM\\_CONFIG\[28:27\] REG\\_IBANK\\_POS](#) field is set to 1, 2, or 3, the EMIF uses the 4 fields to determine the mapping from source address to SDRAM row, column, bank, and chip select:

- [EMIF\\_SDRAM\\_CONFIG\[6:4\] REG\\_IBANK](#)
- [EMIF\\_SDRAM\\_CONFIG\[3\] REG\\_EBANK](#)
- [EMIF\\_SDRAM\\_CONFIG\[2:0\] REG\\_PAGESIZE](#)
- [EMIF\\_SDRAM\\_CONFIG\[9:7\] REG\\_ROWSIZE](#)

In all cases the EMIF considers its SDRAM address space to be a single logical block regardless of the number of physical devices or whether the devices are mapped across 1 or 2 EMIF chip selects.

For [EMIF\\_SDRAM\\_CONFIG\[28:27\] REG\\_IBANK\\_POS = 0](#), [Table 15-106](#) and [Table 15-107](#) show which source address bits, MAddr, map to the SDRAM row, column, bank, and chip select bits for all combinations of [REG\\_IBANK](#), [REG\\_EBANK](#) and [REG\\_PAGESIZE](#).

For [EMIF\\_SDRAM\\_CONFIG\[28:27\] REG\\_IBANK\\_POS != 0](#), [Table 15-109](#) to [Table 15-110](#) shows which source address bits, MAddr, map to the SDRAM row, column, bank, and chip select bits for all combinations of [REG\\_IBANK](#), [REG\\_EBANK](#), [REG\\_PAGESIZE](#), and [REG\\_ROWSIZE](#).

The tables also give the maximum size of the resulting SDRAM space.

Table 15-106. L3 Address to SDRAM Address Mapping for 16-Bit SDRAM (EMIF\_SDRAM\_CONFIG[28:27] REG\_IBANK\_POS=0)

Reach (MBytes)	EMIF_SDRAM_CONFIG[3:0] REG_IBANK	EMIF_SDRAM_CONFIG[6:4] REG_IBANK	EMIF_SDRAM_CONFIG[2:0] REG_ESIZE	MAddr 31	MAddr 30	MAddr 29	MAddr 28	MAddr 27	MAddr 26	MAddr 25	MAddr[24:16]	MAddr 15	MAddr 14	MAddr 13	MAddr 12	MAddr 11	MAddr 10	MAddr 9	MAddr 8:1			
32	0	0	0	-	-	-	-	-	-	-	row	row	row	row	row	row	row	row	col			
64	1	0	0	-	-	-	-	-	-	-	row	row	row	row	row	row	row	cs	col			
64	0	1	0	-	-	-	-	-	-	-	row	row	row	row	row	row	row	bank	col			
128	1	1	0	-	-	-	-	-	-	-	row	row	row	row	row	row	cs	bank	col			
128	0	2	0	-	-	-	-	-	-	-	row	row	row	row	row	row	bank	bank	col			
256	1	2	0	-	-	-	-	-	-	-	row	row	row	row	row	row	cs	bank	bank	col		
256	0	3	0	-	-	-	-	-	-	-	row	row	row	row	row	row	bank	bank	bank	col		
512	1	3	0	-	-	-	-	-	-	-	row	row	row	row	row	cs	bank	bank	bank	col		
64	0	0	1	-	-	-	-	-	-	-	row	row	row	row	row	row	row	row	col	col		
128	1	0	1	-	-	-	-	-	-	-	row	row	row	row	row	row	cs	col	col			
128	0	1	1	-	-	-	-	-	-	-	row	row	row	row	row	row	bank	col	col			
256	1	1	1	-	-	-	-	-	-	-	row	row	row	row	row	cs	bank	col	col			
256	0	2	1	-	-	-	-	-	-	-	row	row	row	row	row	row	bank	bank	col	col		
512	1	2	1	-	-	-	-	-	-	-	row	row	row	row	row	cs	bank	bank	col	col		
512	0	3	1	-	-	-	-	-	-	-	row	row	row	row	row	bank	bank	bank	col	col		
1024	1	3	1	-	-	-	-	-	-	-	row	row	row	row	row	cs	bank	bank	bank	col	col	
128	0	0	2	-	-	-	-	-	-	-	row	row	row	row	row	row	col	col	col			
256	1	0	2	-	-	-	-	-	-	-	row	row	row	row	row	cs	col	col	col			
256	0	1	2	-	-	-	-	-	-	-	row	row	row	row	row	bank	col	col	col			
512	1	1	2	-	-	-	-	-	-	-	row	row	row	row	row	cs	bank	col	col	col		
512	0	2	2	-	-	-	-	-	-	-	row	row	row	row	row	bank	bank	col	col	col		
1024	1	2	2	-	-	-	-	-	-	-	row	row	row	row	row	cs	bank	bank	col	col	col	
1024	0	3	2	-	-	-	-	-	-	-	row	row	row	row	row	bank	bank	bank	col	col	col	
2048	1	3	2	-	-	-	-	-	-	-	row	row	row	row	row	cs	bank	bank	bank	col	col	col
256	0	0	3	-	-	-	-	-	-	-	row	row	row	row	row	row	col	col	col	col		
512	1	0	3	-	-	-	-	-	-	-	row	row	row	row	row	cs	col	col	col	col		
512	0	1	3	-	-	-	-	-	-	-	row	row	row	row	row	bank	col	col	col	col		
1024	1	1	3	-	-	-	-	-	-	-	row	row	row	row	row	cs	bank	col	col	col	col	
1024	0	2	3	-	-	-	-	-	-	-	row	row	row	row	row	bank	bank	col	col	col	col	
2048	1	2	3	-	-	-	-	-	-	-	row	row	row	row	row	cs	bank	bank	col	col	col	col

**Table 15-106. L3 Address to SDRAM Address Mapping for 16-Bit SDRAM (EMIF\_SDRAM\_CONFIG[28:27] REG\_IBANK\_POS=0) (continued)**

Reach (MByte s)	EMIF_SDRAM_CONFIG[3]REG_EBANK	EMIF_SDRAM_CONFIG[6:4]REG_IBANK	EMIF_SDRAM_CONFIG[2:0]REG_ESIZE	MAddr 31	MAddr 30	MAddr 29	MAddr 28	MAddr 27	MAddr 26	MAddr 25	MAddr[24:16]	MAddr 15	MAddr 14	MAddr 13	MAddr 12	MAddr 11	MAddr 10	MAddr 9	MAddr[8:1]
2048	0	3	3	-	row	row	row	row	row	row	row	row	bank	bank	bank	col	col	col	col

**Table 15-107. L3 Address to SDRAM Address Mapping for 32-Bit SDRAM (EMIF\_SDRAM\_CONFIG[28:27] REG\_IBANK\_POS=0)**

Reach (MByte s)	EMIF_SDRAM_CONFIG[3]REG_EBANK	EMIF_SDRAM_CONFIG[6:4]REG_IBANK	EMIF_SDRAM_CONFIG[2:0]REG_ESIZE	MAddr 31	MAddr 30	MAddr 29	MAddr 28	MAddr 27	MAddr 26	MAddr 25	MAddr[24:16]	MAddr 15	MAddr 14	MAddr 13	MAddr 12	MAddr 11	MAddr 10	MAddr 9	MAddr[8:1]
32	0	0	0	-	-	-	-	-	-	row	row	row	row	row	row	row	row	row	col
64	1	0	0	-	-	-	-	-	row	row	row	row	row	row	row	row	row	cs	col
64	0	1	0	-	-	-	-	-	row	row	row	row	row	row	row	row	row	bank	col
128	1	1	0	-	-	-	-	row	row	row	row	row	row	row	row	row	cs	bank	col
128	0	2	0	-	-	-	-	row	row	row	row	row	row	row	row	row	bank	bank	col
256	1	2	0	-	-	-	row	row	row	row	row	row	row	row	row	cs	bank	bank	col
256	0	3	0	-	-	-	row	row	row	row	row	row	row	row	row	bank	bank	bank	col
512	1	3	0	-	-	row	row	row	row	row	row	row	row	row	cs	bank	bank	bank	col
64	0	0	1	-	-	-	-	-	row	row	row	row	row	row	row	row	row	col	col
128	1	0	1	-	-	-	-	row	row	row	row	row	row	row	row	cs	col	col	
128	0	1	1	-	-	-	-	row	row	row	row	row	row	row	row	bank	col	col	
256	1	1	1	-	-	-	row	row	row	row	row	row	row	row	cs	bank	col	col	
256	0	2	1	-	-	-	row	row	row	row	row	row	row	row	row	bank	bank	col	col
512	1	2	1	-	-	row	row	row	row	row	row	row	row	row	cs	bank	bank	col	col
512	0	3	1	-	-	row	row	row	row	row	row	row	row	row	bank	bank	bank	col	col
1024	1	3	1	-	row	row	row	row	row	row	row	row	row	cs	bank	bank	bank	col	col
128	0	0	2	-	-	-	-	row	row	row	row	row	row	row	row	col	col	col	
256	1	0	2	-	-	-	row	row	row	row	row	row	row	row	cs	col	col	col	
256	0	1	2	-	-	-	row	row	row	row	row	row	row	row	bank	col	col	col	
512	1	1	2	-	-	row	row	row	row	row	row	row	row	cs	bank	col	col	col	
512	0	2	2	-	-	row	row	row	row	row	row	row	row	row	bank	bank	col	col	col

**Table 15-107. L3 Address to SDRAM Address Mapping for 32-Bit SDRAM (EMIF\_SDRAM\_CONFIG[28:27] REG\_IBANK\_POS=0) (continued)**

Reach (MBytes)	EMIF_SDRAM_CONFIG[3]REG_EBANK	EMIF_SDRAM_CONFIG[6:4]REG_IBANK	EMIF_SDRAM_CONFIG[2:0]REG_ESIZE	MAddr 31	MAddr 30	MAddr 29	MAddr 28	MAddr 27	MAddr 26	MAddr 25	MAddr[24:16]	MAddr 15	MAddr 14	MAddr 13	MAddr 12	MAddr 11	MAddr 10	MAddr 9	MAddr[8:1]	
1024	1	2	2	-	row	row	row	row	row	row	row	row	row	cs	bank	bank	col	col	col	
1024	0	3	2	-	row	row	row	row	row	row	row	row	row	bank	bank	bank	col	col	col	
2048	1	3	2	row	row	row	row	row	row	row	row	row	row	cs	bank	bank	bank	col	col	col
256	0	0	3	-	-	-	row	row	row	row	row	row	row	row	row	row	col	col	col	col
512	1	0	3	-	-	row	row	row	row	row	row	row	row	row	cs	col	col	col	col	
512	0	1	3	-	-	row	row	row	row	row	row	row	row	row	bank	col	col	col	col	
1024	1	1	3	-	row	row	row	row	row	row	row	row	row	cs	bank	col	col	col	col	
1024	0	2	3	-	row	row	row	row	row	row	row	row	row	bank	bank	col	col	col	col	
2048	1	2	3	row	row	row	row	row	row	row	row	row	row	cs	bank	bank	col	col	col	col
2048	0	3	3	row	row	row	row	row	row	row	row	row	row	bank	bank	bank	col	col	col	col

For [EMIF\\_SDRAM\\_CONFIG\[28:27\] REG\\_IBANK\\_POS = 0](#), the effect of the address-mapping scheme is that as the source address increments across SDRAM page boundaries, the EMIF moves onto the same page in the next bank in the current device (lpddr2i\_cs1[0]n, i= 1 or 2). This movement along the banks of the current device continues until the page has been accessed in all banks in the current device. The EMIF then proceeds to the same page in the next device (if [EMIF\\_SDRAM\\_CONFIG\[3\] REG\\_EBANK = 1](#), lpddr2i\_cs1[1]n, i= 1 or 2) and proceeds through the same page in all its banks before moving over to the next page in the first device (lpddr2i\_cs1[0]n, i= 1 or 2). The EMIF exploits this traversal across internal banks and chip selects while remaining on the same page to maximize the number of open SDRAM banks within the overall SDRAM space.

Thus, the EMIF can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, and can interleave among all of them.

In [Table 15-108](#), [Table 15-109](#), and [Table 15-110](#), the L3 Address to SDRAM Address Mapping is given. In each case, we will have x bits for bank, followed by y bits for row, followed by z bit for CS, followed by s bit for bank and t bit for col in MAddr[31:N] (x, y, z, s and t are given depending on the [EMIF\\_SDRAM\\_CONFIG](#) bit field values).

**Table 15-108. L3 Address to SDRAM Address Mapping for EMIF\_SDRAM\_CONFIG[28:27]  
REG\_IBANK\_POS=1**

MAddr[31:N]				
Bank Address	Row Address	Chip Select	Bank Address	Column Address
Number of bits defined by <b>EMIF_SDRAM_CONFIG</b> [6:4] REG_IBANK	Number of bits defined by <b>EMIF_SDRAM_CONFIG</b> [9:7] REG_ROWSIZE	Number of bits defined by <b>EMIF_SDRAM_CONFIG</b> [3] REG_EBANK	Number of bits defined by <b>EMIF_SDRAM_CONFIG</b> [6:4] REG_IBANK	Number of bits defined by <b>EMIF_SDRAM_CONFIG</b> [2:0] REG_PAGESIZE
= 0 : 0 bits	= 0 : 9 bits	= 0 : 0 bits	= 0 : 0 bits	= 0 : 8 bits
= 1 : 0 bits	= 1 : 10 bits	= 1 : 1 bits	= 1 : 1 bits	= 1 : 9 bits
= 2 : 0 bits	= 2 : 11 bits		= 2 : 2 bits	= 2 : 10 bits
= 3 : 1 bits	= 3 : 12 bits		= 3 : 2 bits	= 3 : 11 bits
	= 4 : 13 bits			
	= 5 : 14 bits			
	= 6 : 15 bits			
	= 7 : 16 bits			

**NOTE:** N=1 for 16-bit SDRAM and N=2 for 32-bit SDRAM

There are 2 times the REG\_IBANK bit field because it gives the number of bank bits depending on their place in the MAddr.

For EMIF\_SDRAM\_CONFIG[28:27] REG\_IBANK\_POS=1, the EMIF interleaves banks the same as EMIF\_SDRAM\_CONFIG[28:27] REG\_IBANK\_POS = 0, but the interleaving of banks within a device (per chip select) is limited to 4 banks. Thus, the EMIF can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, but can only interleave among 8 of them.

**Table 15-109. L3 Address to SDRAM Address Mapping for EMIF\_SDRAM\_CONFIG[28:27] REG\_IBANK\_POS=2**

MAddr[31:N]				
Bank Address	Row Address	Chip Select	Bank Address	Column Address
Number of bits defined by EMIF_SDRAM_CONFIG [6:4] REG_IBANK	Number of bits defined by EMIF_SDRAM_CONFIG [9:7] REG_ROWSIZE	Number of bits defined by EMIF_SDRAM_CONFIG [3] REG_EBANK	Number of bits defined by EMIF_SDRAM_CONFIG [6:4] REG_IBANK	Number of bits defined by EMIF_SDRAM_CONFIG [2:0] REG_PAGESIZE
= 0 : 0 bits	= 0 : 9 bits	= 0 : 0 bits	= 0 : 0 bits	= 0 : 8 bits
= 1 : 0 bits	= 1 : 10 bits	= 1 : 1 bits	= 1 : 1 bits	= 1 : 9 bits
= 2 : 1 bits	= 2 : 11 bits		= 2 : 1 bits	= 2 : 10 bits
= 3 : 2 bits	= 3 : 12 bits		= 3 : 1 bits	= 3 : 11 bits
	= 4 : 13 bits			
	= 5 : 14 bits			
	= 6 : 15 bits			
	= 7 : 16 bits			

**NOTE:** N=1 for 16-bit SDRAM and N=2 for 32-bit SDRAM

For [EMIF\\_SDRAM\\_CONFIG\[28:27\]](#) REG\_IBANK\_POS = 2, the EMIF interleaves banks the same as [EMIF\\_SDRAM\\_CONFIG\[28:27\]](#) REG\_IBANK\_POS = 0, but the interleaving of banks within a device (per chip select) is limited to 2 banks. Thus, the EMIF can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, but can only interleave among 4 of them.

**Table 15-110. L3 Address to SDRAM Address Mapping for EMIF\_SDRAM\_CONFIG[28:27]  
REG\_IBANK\_POS=3**

MAddr[31:N]			
Bank Address	Row Address	Chip Select	Column Address
Number of bits defined by <a href="#">EMIF_SDRAM_CONFIG[6:4]</a> REG_IBANK	Number of bits defined by <a href="#">EMIF_SDRAM_CONFIG[9:7]</a> REG_ROWSIZE	Number of bits defined by <a href="#">EMIF_SDRAM_CONFIG[3]</a> REG_EBANK	Number of bits defined by <a href="#">EMIF_SDRAM_CONFIG[2:0]</a> REG_PAGESIZE
= 0 : 0 bits	= 0 : 9 bits	= 0 : 0 bits	= 0 : 8 bits
= 1 : 1 bits	= 1 : 10 bits	= 1 : 1 bits	= 1 : 9 bits
= 2 : 2 bits	= 2 : 11 bits		= 2 : 10 bits
= 3 : 3 bits	= 3 : 12 bits		= 3 : 11 bits
	= 4 : 13 bits		
	= 5 : 14 bits		
	= 6 : 15 bits		
	= 7 : 16 bits		

---

**NOTE:** N=1 for 16-bit SDRAM and N=2 for 32-bit SDRAM

---

For [EMIF\\_SDRAM\\_CONFIG\[28:27\]](#) REG\_IBANK\_POS = 3, the EMIF cannot interleave banks within a device (per chip select). However, it can still interleave banks between the two chip selects. Thus, the EMIF can keep a maximum of 16 banks (8 internal banks across 2 chip selects) open at a time, but can only interleave among 2 of them.

Since the EMIF interleaves among a lesser number of banks when [EMIF\\_SDRAM\\_CONFIG\[28:27\]](#) REG\_IBANK\_POS != 0, these cases are lower in performance than the [EMIF\\_SDRAM\\_CONFIG\[28:27\]](#) REG\_IBANK\_POS = 0 case. Thus these cases are only recommended to be used along with partial array self-refresh where performance can be traded off for power savings.

#### 15.3.4.12 NVM Address Mapping

NVM are always decoded CS - bank - row - column, and cannot be interleaved.

#### 15.3.4.13 LPDDR2 SDRAM Initialization

On coming out of reset if the [EMIF\\_SDRAM\\_CONFIG\[31:29\]](#) REG\_SDRAM\_TYPE field in the SDRAM Config register is equal to 4 or 5, the EMIF performs a LPDDR2 initialization sequence as follows:

1. Drives [lpddr2i\\_cke\[j\]\[1:0\]](#) (i = 1 or 2 and j = 0 or 1) high and starts to continuously issue NOP commands.
2. After 16 SDRAM refresh rate intervals, issues a PRECHARGE-ALL command. The SDRAM refresh rate is as defined in the [EMIF\\_SDRAM\\_REF\\_CTRL\[15:0\]](#) REG\_REFRESH\_RATE field description.
3. Issues a RESET command.
4. The software then performs the LPDDR2 initialization using the EMIF's [EMIF\\_LPDDR2\\_MODE\\_REG\\_CFG](#) and [EMIF\\_LPDDR2\\_MODE\\_REG\\_DATA](#) registers. The software must enable refreshes by writing a 1 in [EMIF\\_LPDDR2\\_MODE\\_REG\\_CFG\[30\]](#) REG\_REFRESH\_EN field during the last MRW command.

The EMIF also performs the initialization sequence whenever the [EMIF\\_SDRAM\\_CONFIG](#) is written and the LPDDR2 initialization was not performed previously because [SDRAM\\_REF\\_CTRL\[31\]](#) REG\_INITREF\_DIS was set to 0. Once the EMIF performs initialization, rewriting the SDRAM Config register will not cause re-initialization.

---

**NOTE:** The values of the different bit fields in the [EMIF\\_SDRAM\\_CONFIG](#) and [EMIF\\_SDRAM\\_CONFIG\\_2](#) registers are fixed by the System Control Module (SCM) at reset and they can not be modified as they reflect the used hardware LPDDR2 SDRAM and/or NVM memory configurations.

---

The EMIF does not perform any transactions until the LPDDR2 initialization sequence is complete.

When the EMIF comes out of reset, the delay time in step 2 resulting from the 16 refresh rate intervals + 8 cycles is approximately  $16 * \text{EMIF\_SDRAM\_REF\_CTRL}[15:0] \text{ REG\_REFRESH\_RATE} / \text{input frequency}$ .

### 15.3.5 External Memory Interface (EMIF) Programming Guide

#### 15.3.5.1 EMIF Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the EMIF module.

Please, have in mind that the following programming sequences are double as there are two EMIF modules (EMIF1 and EMIF2) with the same register mapping but with different register addresses.

##### 15.3.5.1.1 Global Initialization



**Table 15-111. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
Power, Reset and Clock Management (PRCM)	The module interface and functional clocks must be enabled. See <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Interrupt sub-module of the MPU	The Interrupt requests from the EMIF have to be enabled (unmasked), see <a href="#">Chapter 17, Interrupt Controllers</a> .
DMM (Dynamic Memory Management)	The EMIF is a slave to the DMM module. The DMM should be enabled and configured for communication with LPDDR2 memories. See <a href="#">Section 15.2, DMM</a>

### 15.3.5.1.1.1 EMIF Module Global Initialization

This procedure initializes the EMIF after a power-on reset. It works only with LPDDR2 (Low Power Double Data Rate 2) SDRAM or NVM types. The unused bits in all registers should be written with zeros. The values of the Shadow Registers are loaded in their corresponding registers when the signal "SideAck" is asserted.

**NOTE:** The values of the different bit fields in the [EMIF\\_SDRAM\\_CONFIG](#) and [EMIF\\_SDRAM\\_CONFIG\\_2](#) registers are fixed by the Control Module at reset and they can not be modified as they reflect the used hardware LPDDR2 SDRAM and/or NVM memory configurations.

**Table 15-112. EMIF Global Initialization**

Step	Register/ Bit Field / Programming Model	Value
Disable the Automatic power management mode	<a href="#">EMIF_PWR_MGMT_CTRL</a> [10:8] REG_LP_MODE	0x0
Write the Shadow register of the <a href="#">EMIF_PWR_MGMT_CTRL</a> with the same value.	<a href="#">EMIF_PWR_MGMT_CTRL_SHDW</a>	<a href="#">EMIF_PWR_MGMT_CTRL</a>
Configure the DDR PHY physical layer to the SDRAM (Read Latency for the read data from SDRAM in number of DDR clock cycles)	<a href="#">EMIF_DDR_PHY_CTRL_1</a>	0x-----
Write the Shadow register of the <a href="#">EMIF_DDR_PHY_CTRL_1</a> with the same value.	<a href="#">EMIF_DDR_PHY_CTRL_1_SHDW</a>	<a href="#">EMIF_DDR_PHY_CTRL_1</a>
Configure the DDR PHY physical layer to the SDRAM parameters.	<a href="#">EMIF_DDR_PHY_CTRL_2</a>	0x-----
Write the Shadow register of the <a href="#">EMIF_SDRAM_REF_CTRL</a> with the same value.	<a href="#">EMIF_SDRAM_REF_CTRL_SHDW</a>	<a href="#">EMIF_SDRAM_REF_CTRL</a>
Define the Chip Select to be used and the Mode register address.	<a href="#">EMIF_LPDDR2_MODE_REG_CFG</a>	0x-0000--
Configure the System and Low-Latency L3 Threshold Maxima and the priority counter of an old command.	<a href="#">EMIF_L3_CONFIG</a>	0x0-0-00--
Configure the Performance Counter Config Register	<a href="#">EMIF_PERF_CNT_CFG</a>	0x-00--00-
Configure the Performance Counter Master Region Select Register	<a href="#">EMIF_PERF_CNT_SEL</a>	0x--0---0-
Define timings and clock parameters	EMIF Timing Initialization	
Configure the SDRAM Output Impedance Calibration parameters	EMIF Output Impedance Calibration Mode	
Configure the Temperature Alert parameters	EMIF Temperature Monitoring Mode	
Enable System L3 interrupt for Command or Address Error and/or for Temperature Alert as well.	<a href="#">EMIF_IRQENABLE_SET_SYS</a> [1:0]	0x3
Enable Low-Latency L3 interrupt.	<a href="#">EMIF_IRQENABLE_SET_LL</a> [1:0]	0x3
Specify the dummy-data to be written to the memory for Refresh cycles.	<a href="#">EMIF_LPDDR2_MODE_REG_DATA</a>	0x000000--
Clear all eventual previous indications of treated interrupts – write the 4 registers :	<a href="#">EMIF_IRQSTATUS_SYS</a> , <a href="#">EMIF_IRQSTATUS_LL</a> , <a href="#">EMIF_IRQSTATUS_RAW_SYS</a> , <a href="#">EMIF_IRQSTATUS_RAW_LL</a>	0x0000003

**Table 15-113. Sub-process call summary for Main Sequence - EMIF Global Initialization**

Sub-process Name	Crossreference
EMIF Timing Initialization	<a href="#">Table 15-114 EMIF Timing Initialization</a>
EMIF Output Impedance Calibration Mode	<a href="#">Table 15-115 EMIF Output Impedance Calibration Mode</a>
EMIF Temperature Monitoring Mode	<a href="#">Table 15-123 EMIF Temperature Monitoring Mode</a>

**15.3.5.1.1.2 Sub-sequence - EMIF Timing Initialization****Table 15-114. Sub-sequence - EMIF Timing Initialization**

Step	Register/ Bit Field / Programming Model	Value
Configure the time after which the EMIF being in Idle mode will put the SDRAM in power saving mode.	<a href="#">EMIF_PWR_MGMT_CTRL</a> [3:0] REG_CS_TIM	0x-
Write the Shadow register of the <a href="#">EMIF_PWR_MGMT_CTRL</a> with the same value.	<a href="#">EMIF_PWR_MGMT_CTRL_SHDW</a>	<a href="#">EMIF_PWR_MGMT_CTRL</a>
Configure the LPDDR2-NVM memory timings - 1) Min number of DDR clock cycles from Power-down exit to any command 2) Min number of DDR clock cycles from last Write to Read 3) Min number of DDR clock cycles from Preactive to Activate 4) Min number of DDR clock cycles from last Write transfer to Activate 5) Min number of DDR clock cycles from Activate to Activate for a different bank 6) Min number of DDR clock cycles from Activate to Read or Write	<a href="#">EMIF_LPDDR2_NVM_TIM</a>	0x-----
Write the Shadow register of the <a href="#">EMIF_LPDDR2_NVM_TIM</a> with the same value.	<a href="#">EMIF_LPDDR2_NVM_TIM_SHDW</a>	<a href="#">EMIF_LPDDR2_NVM_TIM</a>
Define the rate at which the connected SDRAM devices are refreshed.	<a href="#">EMIF_SDRAM_REF_CTRL</a> [15:0] REG_REFRESH_RATE	0x---
Write the Shadow register of the <a href="#">EMIF_SDRAM_REF_CTRL</a> with the same value.	<a href="#">EMIF_SDRAM_REF_CTRL_SHDW</a>	<a href="#">EMIF_SDRAM_REF_CTRL</a>
Define timings and clock parameters : 1) Min number of DDR clock cycles for which LPDDR2 must remain in Self Refresh 2) Number of DDR clock cycles for a ZQCS command 3) Number of DDR clock that satisfies t <sub>DQSCKmax</sub> for LPDDR2 4) Min number of DDR clock cycles from Refresh or Load Mode to Refresh or Activate 5) Max number of reg_refresh_rate intervals from Activate to Precharge command	<a href="#">EMIF_SDRAM_TIM_3</a>	0x0000----
Write the Shadow register of the <a href="#">EMIF_SDRAM_TIM_3</a>	<a href="#">EMIF_SDRAM_TIM_3_SHDW</a>	<a href="#">EMIF_SDRAM_TIM_3</a>
Define minimum number of DDR clock cycles from Power-down exit to any command other than a read command	<a href="#">EMIF_SDRAM_TIM_2</a> [30:28]REG_T_XP	0x-
Write the Shadow register of the <a href="#">EMIF_SDRAM_TIM_2</a>	<a href="#">EMIF_SDRAM_TIM_2_SHDW</a>	<a href="#">EMIF_SDRAM_TIM_2</a>
Determine the min size of the Read Idle window for the read idle detection as well as the force read idle time	<a href="#">EMIF_READ_IDLE_CTRL</a> [19:16] REG_READ_IDLE_LEN	0x-
Determine the max interval between read idle detections or force.	<a href="#">EMIF_READ_IDLE_CTRL</a> [8:0] REG_READ_IDLE_INTERVAL	0x---

**Table 15-114. Sub-sequence - EMIF Timing Initialization (continued)**

Step	Register/ Bit Field / Programming Model	Value
Write the Shadow register of the <a href="#">EMIF_READ_IDLE_CTRL</a> with the same value.	<a href="#">EMIF_READ_IDLE_CTRL_SHDW</a>	<a href="#">EMIF_READ_IDLE_CTRL</a>
Configure the Temperature Alert parameters	<a href="#">EMIF_TEMP_ALERT_CONFIG</a>	0x-----
Specify the data to be written to the memory.	<a href="#">EMIF_LPDDR2_MODE_REG_DATA</a>	0x000000--

**15.3.5.1.2 Operational Modes Configuration**

**15.3.5.1.2.1 EMIF Output Impedance Calibration Mode**

**Table 15-115. EMIF Output Impedance Calibration Mode**

Step	Register/ Bit Field / Programming Model	Value
Determine which Chip Select 1 or 0 will be used for automatic output impedance calibration ( <i>Option 1</i> )	<a href="#">EMIF_ZQ_CONFIG</a> [31:30] <a href="#">REG_ZQ_CS1/0EN</a>	0x-
Determine whether both Chip Select 1 and 0 will be used for automatic output impedance calibration simultaneously ( <i>Option 2</i> ).	<a href="#">EMIF_ZQ_CONFIG</a> [29] <a href="#">REG_ZQ_DUALCALEN</a>	0x-
Define the interval (number of refresh periods) between ZQCS commands.	<a href="#">EMIF_ZQ_CONFIG</a> [15:0] <a href="#">REG_ZQ_REFINTERVAL</a>	0x----
Define the number of ZQCL intervals that build the ZQINIT interval.	<a href="#">EMIF_ZQ_CONFIG</a> [19:18] <a href="#">REG_ZQ_ZQINIT_MULT</a>	0x-
Define the number of ZQCS intervals that build the ZQCL interval	<a href="#">EMIF_ZQ_CONFIG</a> [17:16] <a href="#">REG_ZQ_ZQCL_MULT</a>	0x-
Enable the issuing of ZQ-Long Command on Self-Refresh, Active Power-Down, and Precharge Power-Down exit.	<a href="#">EMIF_ZQ_CONFIG</a> [28] <a href="#">REG_ZQ_SFEXITEN</a>	0x1

**15.3.5.1.2.2 EMIF Read**

**Table 15-116. EMIF Read**

Step	Register/ Bit Field / Programming Model	Value
Define the Read Latency of the memory	<a href="#">EMIF_DDR_PHY_CTRL_1</a> [3:0] <a href="#">REG_READ_LATENCY</a>	0x-
Choose Chip Select 0 or 1 to issue mode register command	<a href="#">EMIF_LPDDR2_MODE_REG_CFG</a> [31] <a href="#">REG_CS</a>	0x-
Write the memory address	<a href="#">EMIF_LPDDR2_MODE_REG_CFG</a> [7:0] <a href="#">REG_ADD</a> <a href="#">RESS</a>	0x----
Initiate Read command by reading the register	<a href="#">EMIF_LPDDR2_MODE_REG_DATA</a> [7:0]	Configure the SDRAM Output Impedance Calibration parameters 0x-00-----

**15.3.5.1.2.3 EMIF Write**

**Table 15-117. EMIF Write**

Step	Register/ Bit Field / Programming Model	Value
Choose Chip Select 0 or 1 to issue mode register command	<a href="#">EMIF_LPDDR2_MODE_REG_CFG</a> [31] <a href="#">REG_CS</a>	0x-
Write the memory address	<a href="#">EMIF_LPDDR2_MODE_REG_CFG</a> [7:0] <a href="#">REG_ADD</a> <a href="#">RESS</a>	0x----

**Table 15-117. EMIF Write (continued)**

Step	Register/ Bit Field / Programming Model	Value
Define the minimum number of DDR clock cycles from last Write transfer to Precharge	EMIF_SDRAM_TIM_1[20:17] REG_T_WR	0x-
Supply the Data Byte to be written and initiate Write command by writing the register	EMIF_LPDDR2_MODE_REG_DATA[7:0] REG_VALUE_0	0x--

**15.3.5.1.2.4 EMIF SDRAM Refresh****Table 15-118. EMIF SDRAM Refresh**

Step	Register/ Bit Field / Programming Model	Value
Define the minimum number of DDR clock cycles from Refresh or Load Mode to Refresh or Activate	EMIF_SDRAM_TIM_3[12:4] REG_T_RFC	0x----
Write the Shadow register of the EMIF_SDRAM_TIM_3	EMIF_SDRAM_TIM_3_SHDW	EMIF_SDRAM_TIM_3
Define the rate at which the connected SDRAM devices are refreshed.	EMIF_SDRAM_REF_CTRL[15:0] REG_REFRESH_RATE	0x----
Disable the SDRAM-DDR1/2/3 Refresh	EMIF_SDRAM_REF_CTRL[31] REG_INITREF_DIS	0x1
Write the Shadow register of the EMIF_SDRAM_REF_CTRL	EMIF_SDRAM_REF_CTRL_SHDW	EMIF_SDRAM_REF_CTRL
Choose Chip Select 0 or 1 to issue mode register command.	EMIF_LPDDR2_MODE_REG_CFG[31]	0x-
Enable Refresh for SDRAM-LPDDR2-memory	EMIF_LPDDR2_MODE_REG_CFG[30]	0x1
Dummy write in order to start the LPDDR2 Refresh cycles.	EMIF_LPDDR2_MODE_REG_DATA	0x00000000

**15.3.5.1.2.5 EMIF SDRAM Self-Refresh****Table 15-119. EMIF SDRAM Self-Refresh**

Step	Register/ Bit Field / Programming Model	Value
Define Minimum number of DDR clock cycles from Self-Refresh exit to any command other than a read command	EMIF_SDRAM_TIM_2[24:16] REG_T_XSNR	0x---
Write the Shadow register of the EMIF_SDRAM_TIM_2	EMIF_SDRAM_TIM_2_SHDW	EMIF_SDRAM_TIM_2
Enable the Self-Refresh mode	EMIF_PWR_MGMT_CTRL[10:8] REG_LP_MODE	0x2

**15.3.5.1.2.6 EMIF LPDDR2 (SDRAM or NVM) Power-Down Mode****Table 15-120. EMIF SDRAM Power-Down Mode**

Step	Register/ Bit Field / Programming Model	Value
Define minimum number of DDR clock cycles from Power-down exit to any command other than a read command	EMIF_SDRAM_TIM_2[30:28]REG_T_XP	0x-
Write the Shadow register of the EMIF_SDRAM_TIM_2	EMIF_SDRAM_TIM_2_SHDW	EMIF_SDRAM_TIM_2
Define the number of DDR clock cycles after which the external memory will be put in "Power saving mode" when the EMIF enters in Idle mode.	EMIF_PWR_MGMT_CTRL[3:0] REG_CS_TIM	0x-
Enable (enter) the Power-down mode	EMIF_PWR_MGMT_CTRL[10:8] REG_LP_MODE	0x3

**15.3.5.1.2.7 EMIF LPDDR2-SDRAM Deep Power-Down Mode**
**Table 15-121. EMIF LPDDR2-SDRAM Deep Power-Down Mode**

Step	Register/ Bit Field / Programming Model	Value
Define minimum number of DDR clock cycles from Power-down exit to any command other than a read command	EMIF_SDRAM_TIM_2[30:28]REG_T_XP	0x-
Write the Shadow register of the EMIF_SDRAM_TIM_2	EMIF_SDRAM_TIM_2_SHDW	EMIF_SDRAM_TIM_2
Define the number of DDR clock cycles after which the external memory will be put in "Power saving mode" when the EMIF enters in Idle mode.	EMIF_PWR_MGMT_CTRL[3:0] REG_CS_TIM	0x-
Enable (enter) the Deep Power-Down mode	EMIF_PWR_MGMT_CTRL[11] REG_DPD_EN	0x1

**15.3.5.1.2.8 EMIF Global Warm Reset (Software reset)**
**Table 15-122. EMIF Global Warm Reset (Software reset)**

Step	Register/ Bit Field / Programming Model	Value
IF : External request for Reset	Input signal "EMIFi_GLOBAL_RST"	asserted (Refer to the programming model in the PRCM chapter.)
Follow the Section 15.3.5.1.2.5 EMIF SDRAM Self-Refresh sequence	EMIF_PWR_MGMT_CTRL[10:8] REG_LP_MODE	0x2
ELSE :	Input signal "EMIFi_GLOBAL_RST"	deasserted (Refer to the programming model in the PRCM chapter.)
Disable (exit) the Self-Refresh mode when not needed.	EMIF_PWR_MGMT_CTRL[10:8] REG_LP_MODE	0x0
ENDIF		

**15.3.5.1.2.9 EMIF Temperature Monitoring Mode**
**Table 15-123. EMIF Temperature Monitoring Mode**

Step	Register/ Bit Field / Programming Model	Value
Determine which Chip Select 1 or 0 will be used for Temperature Alert Polling	EMIF_TEMP_ALERT_CONFIG[31:30] REG_TA_CS1/0EN	0x-
Define the interval (number of refresh periods) between temperature alert polls.	EMIF_TEMP_ALERT_CONFIG[21:0] REG_TA_REFINTERVAL	0x-----
Define how wide the physical memory device is	EMIF_TEMP_ALERT_CONFIG[27:26] REG_TA_DEVWDT	0x-
Define which external byte lanes contain a device for temperature monitoring (which devices to be polled).	EMIF_TEMP_ALERT_CONFIG[25:24] REG_TA_DEVCNT	0x-
Enable System L3 interrupt for SDRAM temperature alert.	EMIF_IRQENABLE_SET_SYS[1] REG_EN_TA_SYS	0x1
Enable Low-Latency L3 interrupt for SDRAM temperature alert.	EMIF_IRQENABLE_SET_LL[1] REG_EN_TA_LL	0x1
Enable the issuing of a temperature alert poll on Self-Refresh exit (poll for temperature change every time EMIF exits Self-Refresh, Active Power-Down, and Precharge Power-Down modes).	EMIF_TEMP_ALERT_CONFIG[28] REG_TA_SFEXITEN	0x1

PRELIMINARY

### 15.3.5.1.3 EMIF Events Servicing

#### 15.3.5.1.3.1 Interrupt Servicing

This section covers the event servicing (interrupt treatment) of the EMIF module.

Please, have in mind that the following Interrupt mechanism is double as there are two EMIF modules with different register addresses and 2 Interrupt lines - EMIF1\_IRQ and EMIF2\_IRQ.

**Table 15-124. EMIF Event Servicing**

Step	Register/ Bit Field / Programming Model	Value
Clear all eventual previous indications of treated interrupts – write the 4 registers :	<a href="#">EMIF_IRQSTATUS_SYS</a> , <a href="#">EMIF_IRQSTATUS_LL</a> , <a href="#">EMIF_IRQSTATUS_RAW_SYS</a> , <a href="#">EMIF_IRQSTATUS_RAW_LL</a>	0x0000003
Enable the System L3 interrupts	<a href="#">EMIF_IRQENABLE_SET_SYS</a>	0x0000003
Enable the Low-Latency L3 interrupts	<a href="#">EMIF_IRQENABLE_SET_LL</a>	0x0000003
Wait until an Event occurs...		
Read the status registers :	<a href="#">EMIF_IRQSTATUS_SYS</a> , <a href="#">EMIF_IRQSTATUS_LL</a> , <a href="#">EMIF_IRQSTATUS_RAW_SYS</a> , <a href="#">EMIF_IRQSTATUS_RAW_LL</a>	
IF : the <a href="#">EMIF_IRQSTATUS_SYS</a> [0] =1		
Read the Error Log register and determine the Addressing mode, Command type and the Connection ID of the first errored transaction :	<a href="#">EMIF_L3_ERR_LOG</a>	
Start the Interrupt Handler for System L3 - Command/Address Error	Interrupt Controller of the MPU	
ELSE IF : the <a href="#">EMIF_IRQSTATUS_SYS</a> [1] =1		
Read the Error Log register and determine the Addressing mode, Command type and the Connection ID of the first errored transaction :	<a href="#">EMIF_L3_ERR_LOG</a>	
Start the Interrupt Handler for System L3 - Temperature Alert	Interrupt Controller of the MPU	
Update the REFRESH_RATE to the required value as per the temperature change.	<a href="#">EMIF_SDRAM_REF_CTRL</a> [15:0] REG_REFRESH_RATE	0x----
ELSE IF : the <a href="#">EMIF_IRQSTATUS_RAW_SYS</a> [0] =1		
Read the Error Log register and determine the Addressing mode, Command type and the Connection ID of the first errored transaction :	<a href="#">EMIF_L3_ERR_LOG</a>	
Start the Interrupt Handler for Raw System L3 - Command/Address Error	Interrupt Controller of the MPU	
ELSE IF : the <a href="#">EMIF_IRQSTATUS_RAW_SYS</a> [1] =1		
Read the Error Log register and determine the Addressing mode, Command type and the Connection ID of the first errored transaction :	<a href="#">EMIF_L3_ERR_LOG</a>	
Start the Interrupt Handler for Raw System L3 - Temperature Alert	Interrupt Controller of the MPU	
Update the REFRESH_RATE to the required value as per the temperature change.	<a href="#">EMIF_SDRAM_REF_CTRL</a> [15:0] REG_REFRESH_RATE	0x----
ELSE IF : the <a href="#">EMIF_IRQSTATUS_LL</a> [0] =1		
Read the Error Log register and determine the Addressing mode, Command type and the Connection ID of the first errored transaction :	<a href="#">EMIF_L3_ERR_LOG</a>	
Start the Interrupt Handler for Low-Latency L3 - Command/Address Error	Interrupt Controller of the MPU	

**Table 15-124. EMIF Event Servicing (continued)**

Step	Register/ Bit Field / Programming Model	Value
ELSE IF : the <a href="#">EMIF_IRQSTATUS_LL[1]</a> =1		
Read the Error Log register and determine the Addressing mode, Command type and the Connection ID of the first errored transaction :	<a href="#">EMIF_L3_ERR_LOG</a>	
Start the Interrupt Handler for Low-Latency L3 - Temperature Alert	Interrupt Controller of the MPU	
Update the REFRESH_RATE to the required value as per the temperature change.	<a href="#">EMIF_SDRAM_REF_CTRL[15:0]</a> <a href="#">REG_REFRESH_RATE</a>	0x----
ELSE IF : the <a href="#">EMIF_IRQSTATUS_RAW_LL[0]</a> =1		
Read the Error Log register and determine the Addressing mode, Command type and the Connection ID of the first errored transaction :	<a href="#">EMIF_L3_ERR_LOG</a>	
Start the Interrupt Handler for Raw Low-Latency L3 - Command/Address Error	Interrupt Controller of the MPU	
ELSE : (the <a href="#">EMIF_IRQSTATUS_RAW_LL[1]</a> =1)		
Read the Error Log register and determine the Addressing mode, Command type and the Connection ID of the first errored transaction :	<a href="#">EMIF_L3_ERR_LOG</a>	
Start the Interrupt Handler for Raw Low-Latency L3 - Temperature Alert	Interrupt Controller of the MPU	
Update the REFRESH_RATE to the required value as per the temperature change.	<a href="#">EMIF_SDRAM_REF_CTRL[15:0]</a> <a href="#">REG_REFRESH_RATE</a>	0x----
ENDIF		
Wait until the termination of the corresponding Interrupt Handler...	Interrupt Controller of the MPU	
IF : Address/Command Error was treated		
Clear the LSbit in the relevant IRQ Status registers SYS or LL – write both registers :	<a href="#">EMIF_IRQSTATUS_SYS[0]</a> <a href="#">REG_ERR_SYS</a> and <a href="#">EMIF_IRQSTATUS_RAW_SYS[0]</a> <a href="#">REG_ERR_SYS</a> , or <a href="#">EMIF_IRQSTATUS_LL[0]</a> <a href="#">REG_ERR_LL</a> and <a href="#">EMIF_IRQSTATUS_RAW_LL[0]</a> <a href="#">REG_ERR_LL</a>	0x1
Disable the corresponding Interrupt if needed – write :	<a href="#">EMIF_IRQENABLE_CLR_SYS[0]</a> <a href="#">REG_EN_ERR_SYS</a> or <a href="#">EMIF_IRQENABLE_CLR_LL[0]</a> <a href="#">REG_EN_ERR_LL</a>	0x1
ELSE : (Temperature Alert was proceeded)		
Clear the corresponding bit in the relevant IRQ Status registers SYS or LL – write both registers :	<a href="#">EMIF_IRQSTATUS_SYS[1]</a> <a href="#">REG_TA_SYS</a> and <a href="#">EMIF_IRQSTATUS_RAW_SYS[1]</a> <a href="#">REG_TA_SYS</a> , or <a href="#">EMIF_IRQSTATUS_LL[1]</a> <a href="#">REG_TA_LL</a> and <a href="#">EMIF_IRQSTATUS_RAW_LL[1]</a> <a href="#">REG_TA_LL</a>	0x1
Disable the corresponding Interrupt if needed – write :	<a href="#">EMIF_IRQENABLE_CLR_SYS[1]</a> <a href="#">REG_EN_TA_SYS</a> or <a href="#">EMIF_IRQENABLE_CLR_LL[1]</a> <a href="#">REG_EN_TA_LL</a>	0x1
ENDIF		



## 15.3.6 EMIF Register Manual

### 15.3.6.1 EMIF Instance Summary

**Table 15-125. EMIF Instance Summary**

Module Name	Base Address	Size
EMIF1	0x4C00 0000	16 MB
EMIF2	0x4D00 0000	16 MB

### 15.3.6.2 EMIF Registers

#### 15.3.6.2.1 EMIF Register Summary

**Table 15-126. EMIF Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	EMIF1 L3 Physical Address	EMIF2 L3 Physical Address
EMIF_MOD_ID_RE V	R	32	0x0000 0000	0x4C00 0000	0x4D00 0000
EMIF_STATUS	R	32	0x0000 0004	0x4C00 0004	0x4D00 0004
EMIF_SDRAM_CO NFIG	RW	32	0x0000 0008	0x4C00 0008	0x4D00 0008
EMIF_SDRAM_CO NFIG_2	RW	32	0x0000 000C	0x4C00 000C	0x4D00 000C
EMIF_SDRAM_REF _CTRL	RW	32	0x0000 0010	0x4C00 0010	0x4D00 0010
EMIF_SDRAM_REF _CTRL_SHDW	RW	32	0x0000 0014	0x4C00 0014	0x4D00 0014
EMIF_SDRAM_TIM _1	RW	32	0x0000 0018	0x4C00 0018	0x4D00 0018
EMIF_SDRAM_TIM _1_SHDW	RW	32	0x0000 001C	0x4C00 001C	0x4D00 001C
EMIF_SDRAM_TIM _2	RW	32	0x0000 0020	0x4C00 0020	0x4D00 0020
EMIF_SDRAM_TIM _2_SHDW	RW	32	0x0000 0024	0x4C00 0024	0x4D00 0024
EMIF_SDRAM_TIM _3	RW	32	0x0000 0028	0x4C00 0028	0x4D00 0028
EMIF_SDRAM_TIM _3_SHDW	RW	32	0x0000 002C	0x4C00 002C	0x4D00 002C
EMIF_LPDDR2_NV M_TIM	RW	32	0x0000 0030	0x4C00 0030	0x4D00 0030
EMIF_LPDDR2_NV M_TIM_SHDW	RW	32	0x0000 0034	0x4C00 0034	0x4D00 0034
EMIF_PWR_MGMT _CTRL	RW	32	0x0000 0038	0x4C00 0038	0x4D00 0038
EMIF_PWR_MGMT _CTRL_SHDW	RW	32	0x0000 003C	0x4C00 003C	0x4D00 003C
EMIF_LPDDR2_MO DE_REG_DATA	RW	32	0x0000 0040	0x4C00 0040	0x4D00 0040
EMIF_LPDDR2_MO DE_REG_CFG	RW	32	0x0000 0050	0x4C00 0050	0x4D00 0050
EMIF_L3_CONFIG	RW	32	0x0000 0054	0x4C00 0054	0x4D00 0054
EMIF_L3_CFG_VA L_1	R	32	0x0000 0058	0x4C00 0058	0x4D00 0058
EMIF_L3_CFG_VA L_2	R	32	0x0000 005C	0x4C00 005C	0x4D00 005C

**Table 15-126. EMIF Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	EMIF1 L3 Physical Address	EMIF2 L3 Physical Address
EMIF_PERF_CNT_1	R	32	0x0000 0080	0x4C00 0080	0x4D00 0080
EMIF_PERF_CNT_2	R	32	0x0000 0084	0x4C00 0084	0x4D00 0084
EMIF_PERF_CNT_CFG	RW	32	0x0000 0088	0x4C00 0088	0x4D00 0088
EMIF_PERF_CNT_SEL	RW	32	0x0000 008C	0x4C00 008C	0x4D00 008C
EMIF_PERF_CNT_TIM	R	32	0x0000 0090	0x4C00 0090	0x4D00 0090
EMIF_READ_IDLE_CTRL	RW	32	0x0000 0090	0x4C00 0098	0x4D00 0098
EMIF_READ_IDLE_CTRL_SHDW	RW	32	0x0000 0090	0x4C00 009C	0x4D00 009C
EMIF_IRQSTATUS_RAW_SYS	RW	32	0x0000 00A4	0x4C00 00A4	0x4D00 00A4
EMIF_IRQSTATUS_RAW_LL	RW	32	0x0000 00A8	0x4C00 00A8	0x4D00 00A8
EMIF_IRQSTATUS_SYS	RW	32	0x0000 00AC	0x4C00 00AC	0x4D00 00AC
EMIF_IRQSTATUS_LL	RW	32	0x0000 00B0	0x4C00 00B0	0x4D00 00B0
EMIF_IRQENABLE_SET_SYS	RW	32	0x0000 00B4	0x4C00 00B4	0x4D00 00B4
EMIF_IRQENABLE_SET_LL	RW	32	0x0000 00B8	0x4C00 00B8	0x4D00 00B8
EMIF_IRQENABLE_CLR_SYS	RW	32	0x0000 00BC	0x4C00 00BC	0x4D00 00BC
EMIF_IRQENABLE_CLR_LL	RW	32	0x0000 00C0	0x4C00 00C0	0x4D00 00C0
EMIF_ZQ_CONFIG	RW	32	0x0000 00C8	0x4C00 00C8	0x4D00 00C8
EMIF_TEMP_ALERT_CONFIG	RW	32	0x0000 00CC	0x4C00 00CC	0x4D00 00CC
EMIF_L3_ERR_LOG	R	32	0x0000 00D0	0x4C00 00D0	0x4D00 00D0
EMIF_DDR_PHY_CTRL_1	RW	32	0x0000 00E4	0x4C00 00E4	0x4D00 00E4
EMIF_DDR_PHY_CTRL_1_SHDW	RW	32	0x0000 00E8	0x4C00 00E8	0x4D00 00E8
EMIF_DDR_PHY_CTRL_2	RW	32	0x0000 00EC	0x4C00 00EC	0x4D00 00EC

**NOTE:** The values of the different bit fields in the [EMIF\\_SDRAM\\_CONFIG](#) and [EMIF\\_SDRAM\\_CONFIG2](#) registers are fixed by the Control Module (CM) at reset and they can not be modified as they reflect the used hardware LPDDR2 SDRAM and/or NVM memory configurations.

### 15.3.6.2.2 EMIF Register Description

**Table 15-127. EMIF\_MOD\_ID\_REV**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b> EMIF
<b>Description</b>	Revision number register	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved for future use	R	0x---- ---- <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 15-128. Register Call Summary for Register EMIF\_MOD\_ID\_REV**

EMIF Functional Description

- [Arbitration: \[0\] \[1\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[2\]](#)

**Table 15-129. EMIF\_STATUS**

<b>Address Offset</b>	0x0000 0004	
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b> EMIF
<b>Description</b>	SDRAM Status Register (STATUS)	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_BE		REG_DUAL_CLK_MODE		REG_FAST_INIT		RESERVED																				REG_PHY_DLL_READY		RESERVED			

Bits	Field Name	Description	Type	Reset
31	REG_BE	Big endian mode select for 8 and 16-bit devices, set to 1 for big endian or 0 for little endian operation.	R	0
30	REG_DUAL_CLK_MODE	Dual Clock mode. Defines whether the L3_clk and DDR clock are asynchronous. L3_clk and DDR clock are asynchronous, if set to 1.	R	1
29	REG_FAST_INIT	Fast Init. Defines whether the EMIF fast initialization mode has been enabled. Fast initialization is enabled if set to 1.	R	0
28:3	RESERVED	Reserved	R	0x0000000
2	REG_PHY_DLL_READY	DDR PHY Ready. Reflects the value on the phy_ready port (active high) that defines whether the DDR PHY is ready for normal operation. The DDR PHY is ready for normal operation, if set to 1.	R	1
1:0	RESERVED	Reserved	R	0x0

**Table 15-130. Register Call Summary for Register EMIF\_STATUS**

EMIF Register Manual

- [EMIF Register Summary: \[0\]](#)

**Table 15-131. EMIF\_SDRAM\_CONFIG**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	SDRAM Config Register. A write to this register will cause the EMIF to start the SDRAM initialization sequence if it was not performed previously because <a href="#">EMIF_SDRAM_REF_CTRL[31]</a> <a href="#">REG_INITREF_DIS</a> was a zero.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_SDRAM_TYPE		REG_IBANK_POS		RESERVED												REG_NARROW_MODE		REG_CL				REG_ROWSIZE		REG_IBANK		REG_EBANK	REG_PAGESIZE				

Bits	Field Name	Description	Type	Reset
31:29	REG_SDRAM_TYPE	SDRAM Type selection. Set to 4 for LPDDR2-S4, Set to 5 for LPDDR2-S2 All other value are reserved.	RW	0x- <sup>(1)</sup>
28:27	REG_IBANK_POS	Internal bank position. Set to 0 to assign internal bank address bits from L3 address as shown in <a href="#">Table 15-99</a> and <a href="#">Table 15-100</a> . Set to 1, 2, or 3 to assign internal bank address bits from L3 address as shown in <a href="#">Table 15-101</a> , <a href="#">Table 15-102</a> and <a href="#">Table 15-103</a> .	RW	0x- <sup>(1)</sup>
26:16	RESERVED	reserved	RW	0x0
15:14	REG_NARROW_MODE	SDRAM data bus width. Set to 0 for 32-bit and set to 1 for 16-bit. All other values are reserved.	RW	0x- <sup>(1)</sup>
13:10	REG_CL	CAS Latency. The value of this field defines the CAS latency to be used when accessing connected SDRAM devices. Value of 3, 4, 5, 6, 7, and 8 (CAS latency of 3, 4, 5, 6, 7, and 8) are supported for LPDDR2-SDRAM. All other values are reserved.	RW	0x- <sup>(1)</sup>
9:7	REG_ROWSIZE	Row Size. Defines the number of row address bits of connected SDRAM devices. Set to 0 for 9 row bits, Set to 1 for 10 row bits, Set to 2 for 11 row bits, Set to 3 for 12 row bits, Set to 4 for 13 row bits, Set to 5 for 14 row bits, Set to 6 for 15 row bits, Set to 7 for 16 row bits. This field is only used when <a href="#">EMIF_SDRAM_CONFIG[28:27]</a> <a href="#">REG_IBANK_POS</a> field is set to 1, 2, or 3 or <a href="#">REG_EBANK_POS</a> field in <a href="#">EMIF_SDRAM_CONFIG_2</a> register is set to 1.	RW	0x- <sup>(1)</sup>

<sup>(1)</sup> value fixed by the system control module at reset

Bits	Field Name	Description	Type	Reset
6:4	REG_IBANK	Internal Bank setup. Defines number of banks inside connected SDRAM devices. Set to 0 for 1 bank, Set to 1 for 2 banks, Set to 2 for 4 banks, Set to 3 for 8 banks. All other values are reserved.	RW	0x- <sup>(1)</sup>
3	REG_EBANK	External chip select setup. Defines whether SDRAM accesses will use 1 or 2 chip select lines. Set to 0 to use pad_cs_o_n[0] only. Set to 1 to use pad_cs_o_n[1:0]. This bit will automatically be set to 0 if <a href="#">EMIF_SDRAM_CONFIG_2[30]</a> REG_CS1NVMEN field is set to 1.	RW	0x- <sup>(1)</sup>
2:0	REG_PAGESIZE	Page Size. Defines the internal page size of connected SDRAM devices. Set to 0 for 256-word page (8 column bits), Set to 1 for 512-word page (9 column bits), Set to 2 for 1024-word page (10 column bits), Set to 3 for 2048-word page (11 column bits). All other values are reserved.	RW	0x- <sup>(1)</sup>

**Table 15-132. Register Call Summary for Register EMIF\_SDRAM\_CONFIG**

EMIF Functional Description

- Arbitration: [0]
- EMIF Access Cycles: [1]
- SDRAM Address Mapping: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42]
- LPDDR2 SDRAM Initialization: [43] [44] [45]

External Memory Interface (EMIF) Programming Guide

- EMIF Module Global Initialization: [46]

EMIF Register Manual

- EMIF Register Summary: [47] [48] [49]
- EMIF Register Description: [50]

**Table 15-133. EMIF\_SDRAM\_CONFIG\_2**

<b>Address Offset</b>	0x0000 000C
<b>Physical Address</b>	See <a href="#">Table 15-126</a>
<b>Description</b>	
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	REG_CS1NVMEN	RESERVED	RESERVED	REG_EBANK_POS	RESERVED												REG_RDBNUM	RESERVED	REG_RDBSIZE												

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30	REG_CS1NVMEN	CS1 LPDDR2-NVM enable. Set to 1 if LPDDR2-NVM is connected to CS1. This bit will automatically be set to 0 if reg_sdram_type field in the SDRAM Config register is not set to LPDDR2.	RW	0
29:28	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
27	REG_EBANK_POS	External bank position. Set to 0 to assign external bank address bits from lower OCP address as shown in tables. Set to 1 to assign external bank address bits from higher OCP address bits as shown in tables.	RW	0
26:6	RESERVED		R	0x000000
5:4	REG_RDBNUM	Row Buffer setup. Defines number of row buffers inside connected LPDDR2-NVM devices. Set to 0 for 1 row buffer, set to 1 for 2 row buffers, set to 2 for 4 row buffers, and set to 3 for 8 row buffers. All other values are reserved.	RW	0x0
3	RESERVED		R	0
2:0	REG_RDBSIZE	Row Data Buffer Size. Defines the row data buffer size of connected LPDDR2-NVM devices. Set to 0 for 32 bytes, set to 1 for 64 bytes, set to 2 for 128 bytes, set to 3 for 256 bytes, set to 4 for 512 bytes, set to 5 for 1024 bytes, set to 6 for 2048 bytes, and set to 7 for 4096 bytes.	RW	0x0

**Table 15-134. Register Call Summary for Register EMIF\_SDRAM\_CONFIG\_2**

## EMIF Functional Description

- [LPDDR2 SDRAM Initialization: \[0\]](#)

## External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[1\]](#)

## EMIF Register Manual

- [EMIF Register Summary: \[2\]](#)
- [EMIF Register Description: \[3\] \[4\]](#)

**Table 15-135. EMIF\_SDRAM\_REF\_CTRL**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	SDRAM Refresh Control Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_REFRESH_RATE																							

Bits	Field Name	Description	Type	Reset
31	REG_INITREF_DIS	Initialization and Refresh disable. When set to 1, EMIF will disable SDRAM initialization and refreshes, but will carry out SDRAM write/read transactions.	RW	0
30:16	RESERVED	Reserved	R	0
15:0	REG_REFRESH_RATE	Refresh Rate. Value in this field is used to define the rate at which connected SDRAM devices will be refreshed. SDRAM refresh rate = EMIF rate / reg_refresh_rate where EMIF rate is equal to DDR clock rate. If $reg\_refresh\_rate < (8 * reg\_t\_rfc) + reg\_t\_rp + reg\_t\_rcd + 20$ then it will be loaded with $(8 * reg\_t\_rfc) + reg\_t\_rp + reg\_t\_rcd + 20$ . This is done to avoid lock-up situations when illegal values are programmed.	RW	0x0000

**Table 15-136. Register Call Summary for Register EMIF\_SDRAM\_REF\_CTRL**

EMIF Functional Description

- Arbitration: [0]
- SDRAM Temperature Monitoring: [1]
- SDRAM Refresh Scheduling: [2] [3] [4]
- LPDDR2 SDRAM Initialization: [5] [6]

External Memory Interface (EMIF) Programming Guide

- EMIF Module Global Initialization: [7] [8]
- Sub-sequence - EMIF Timing Initialization: [9] [10] [11]
- EMIF SDRAM Refresh: [12] [13] [14] [15] [16]
- Interrupt Servicing: [17] [18] [19] [20]

EMIF Register Manual

- EMIF Register Summary: [21]
- EMIF Register Description: [22] [23] [24]

**Table 15-137. EMIF\_SDRAM\_REF\_CTRL\_SHDW**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	SDRAM Refresh Control Shadow Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_REFRESH_RATE_SHDW															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	REserved	R	0x0000
15:0	REG_REFRESH_RATE_SHDW	Shadow field for REG_REFRESH_RATE. This field is loaded into <a href="#">EMIF_SDRAM_REF_CTRL</a> [15:0] REG_REFRESH_RATE field when SldleAck is asserted. This register is not auto corrected when the value is invalid.	RW	0x0000

**Table 15-138. Register Call Summary for Register EMIF\_SDRAM\_REF\_CTRL\_SHDW**

External Memory Interface (EMIF) Programming Guide

- EMIF Module Global Initialization: [0]
- Sub-sequence - EMIF Timing Initialization: [1]

EMIF Register Manual

- EMIF Register Summary: [2]

**Table 15-139. EMIF\_SDRAM\_TIM\_1**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	SDRAM Timing 1 Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the SDRAM.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				REG_T_RP				REG_T_RCD				REG_T_WR				REG_T_RAS				REG_T_RC				REG_T_RRD		REG_T_WTR					

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:25	REG_T_RP	Minimum number of DDR clock cycles from Precharge to Activate or Refresh, minus one.	RW	0x0
24:21	REG_T_RCD	Minimum number of DDR clock cycles from Activate to Read or Write, minus one.	RW	0x0
20:17	REG_T_WR	Minimum number of DDR clock cycles from last Write transfer to Precharge, minus one.	RW	0x0
16:12	REG_T_RAS	Minimum number of DDR clock cycles from Activate to Precharge, minus one. reg_t_ras value needs to be bigger than or equal to reg_t_rcd value.	RW	0x0
11:6	REG_T_RC	Minimum number of DDR clock cycles from Activate to Activate, minus one.	RW	000010
5:3	REG_T_RRD	Minimum number of DDR clock cycles from Activate to Activate for a different bank, minus one. For an 8-bank, this field must be equal to ((tFAW/(4*tCK))-1).	RW	100
2:0	REG_T_WTR	Minimum number of DDR clock cycles from last Write to Read, minus one.	RW	11

**Table 15-140. Register Call Summary for Register EMIF\_SDRAM\_TIM\_1**

External Memory Interface (EMIF) Programming Guide

- [EMIF Write: \[0\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[1\]](#)
- [EMIF Register Description: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

**Table 15-141. EMIF\_SDRAM\_TIM\_1\_SHDW**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	SDRAM Timing 1 Shadow Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				REG_T_RP_SHDW				REG_T_RCD_SHDW				REG_T_WR_SHDW				REG_T_RAS_SHDW				REG_T_RC_SHDW				REG_T_RRD_SHDW				REG_T_WTR_SHDW			

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:25	REG_T_RP_SHDW	Shadow field for REG_T_RP. This field is loaded into <a href="#">EMIF_SDRAM_TIM_1</a> [28:25] REG_T_RP field when SldeAck is asserted.	RW	0x0
24:21	REG_T_RCD_SHDW	Shadow field for REG_T_RCD. This field is loaded into <a href="#">EMIF_SDRAM_TIM_1</a> [24:21] REG_T_RCD field when SldeAck is asserted.	RW	0x0
20:17	REG_T_WR_SHDW	Shadow field for REG_T_WR. This field is loaded into <a href="#">EMIF_SDRAM_TIM_1</a> [20:17] REG_T_WR field when SldeAck is asserted.	RW	0x0
16:12	REG_T_RAS_SHDW	Shadow field for REG_T_RAS. This field is loaded into <a href="#">EMIF_SDRAM_TIM_1</a> [16:12] REG_T_RAS field when SldeAck is asserted.	RW	0x00



Bits	Field Name	Description	Type	Reset
11:6	REG_T_RC_SHDW	Shadow field for REG_T_RC. This field is loaded into <a href="#">EMIF_SDRAM_TIM_1</a> [11:6] REG_T_RC field when <code>SldeAck</code> is asserted.	RW	000010
5:3	REG_T_RRD_SHDW	Shadow field for REG_T_RRD. This field is loaded into <a href="#">EMIF_SDRAM_TIM_1</a> [5:3] REG_T_RRD field when <code>SldeAck</code> is asserted.	RW	0x4
2:0	REG_T_WTR_SHDW	Shadow field for REG_T_WTR. This field is loaded into <a href="#">EMIF_SDRAM_TIM_1</a> [2:0] REG_T_WTR field when <code>SldeAck</code> is asserted.	RW	0x3

**Table 15-142. Register Call Summary for Register EMIF\_SDRAM\_TIM\_1\_SHDW**

EMIF Register Manual

- [EMIF Register Summary: \[0\]](#)

**Table 15-143. EMIF\_SDRAM\_TIM\_2**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	SDRAM Timing 2 Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the SDRAM.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	REG_T_XP	RESERVED	REG_T_XSNR													REG_T_XSRD						REG_T_RTP	REG_T_CKE								

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30:28	REG_T_XP	Minimum number of DDR clock cycles from Powerdown exit to any command other than a Read command, minus one.	RW	0x2
27:25	RESERVED	reserved	R	0x0
24:16	REG_T_XSNR	Minimum number of DDR clock cycles from Self-Refresh exit to any command other than a Read command, minus one. REG_T_XSNR and REG_T_XSRD must be programmed with the same value.	RW	0x0A
15:6	REG_T_XSRD	Minimum number of DDR clock cycles from Self-Refresh exit to a Read command, minus one. REG_T_XSNR and REG_T_XSRD must be programmed with the same value.	RW	0000001010
5:3	REG_T_RTP	Minimum number of DDR clock cycles for the last read command to a Precharge command, minus one.	RW	001
2:0	REG_T_CKE	Minimum number of DDR clock cycles between <code>pad_cke_o</code> changes, minus one.	RW	010

**Table 15-144. Register Call Summary for Register EMIF\_SDRAM\_TIM\_2**

EMIF Functional Description

- [LPDDR2-SDRAM Power-Down Mode: \[0\]](#)
- [LPDDR2-NVM Power-Down Mode: \[1\]](#)
- [LPDDR2-SDRAM Self-Refresh Mode: \[2\] \[3\]](#)

**Table 15-144. Register Call Summary for Register EMIF\_SDRAM\_TIM\_2 (continued)**

External Memory Interface (EMIF) Programming Guide

- [Sub-sequence - EMIF Timing Initialization: \[4\] \[5\] \[6\]](#)
- [EMIF SDRAM Self-Refresh: \[7\] \[8\]](#)
- [EMIF LPDDR2 \(SDRAM or NVM\) Power-Down Mode: \[9\] \[10\] \[11\]](#)
- [EMIF LPDDR2-SDRAM Deep Power-Down Mode: \[12\] \[13\] \[14\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[15\]](#)
- [EMIF Register Description: \[16\] \[17\] \[18\] \[19\] \[20\]](#)

**Table 15-145. EMIF\_SDRAM\_TIM\_2\_SHDW**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	SDRAM Timing 2 Shadow Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	REG_T_XP_SHDW	RESERVED	REG_T_XSNR_SHDW				REG_T_XSRD_SHDW										REG_T_RTP_SHDW	REG_T_CKE_SHDW													

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30:28	REG_T_XP_SHDW	Shadow field for REG_T_XP. This field is loaded into <a href="#">EMIF_SDRAM_TIM_2</a> [30:28] REG_T_XP field when SldleAck is asserted.	RW	0x2
27:25	RESERVED	Reserved	RW	0x0
24:16	REG_T_XSNR_SHDW	Shadow field for REG_T_XSNR. This field is loaded into <a href="#">EMIF_SDRAM_TIM_2</a> [24:16] REG_T_XSNR field when SldleAck is asserted.	RW	0x0A
15:6	REG_T_XSRD_SHDW	Shadow field for REG_T_XSRD. This field is loaded into <a href="#">EMIF_SDRAM_TIM_2</a> [15:6] REG_T_XSRD field when SldleAck is asserted.	RW	000001010
5:3	REG_T_RTP_SHDW	Shadow field for REG_T_RTP. This field is loaded into <a href="#">EMIF_SDRAM_TIM_2</a> [5:3] REG_T_RTP field when SldleAck is asserted.	RW	0x1
2:0	REG_T_CKE_SHDW	Shadow field for REG_T_CKE. This field is loaded into <a href="#">EMIF_SDRAM_TIM_2</a> [2:0] REG_T_CKE field when SldleAck is asserted.	RW	0x2

**Table 15-146. Register Call Summary for Register EMIF\_SDRAM\_TIM\_2\_SHDW**

External Memory Interface (EMIF) Programming Guide

- [Sub-sequence - EMIF Timing Initialization: \[0\]](#)
- [EMIF SDRAM Self-Refresh: \[1\]](#)
- [EMIF LPDDR2 \(SDRAM or NVM\) Power-Down Mode: \[2\]](#)
- [EMIF LPDDR2-SDRAM Deep Power-Down Mode: \[3\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[4\]](#)

**Table 15-147. EMIF\_SDRAM\_TIM\_3**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	SDRAM Timing 3 Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the SDRAM.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_T_CKESR		REG_ZQ_ZQCS						REG_T_TDQSKMAX		REG_T_RFC						REG_T_RAS_MAX							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:21	REG_T_CKESR	Minimum number of DDR clock cycles for which LPDDR2 must remain in Self Refresh, minus one.	RW	0x2
20:15	REG_ZQ_ZQCS	Number of DDR clock cycles for a ZQCS command, minus one.	RW	0x08
14:13	REG_T_TDQSKMAX	Number of DDR clock that satisfies tDQSKmax for LPDDR2, minus one.	RW	01
12:4	REG_T_RFC	Minimum number of DDR clock cycles from Refresh or Load Mode to Refresh or Activate, minus one.	RW	0x115
3:0	REG_T_RAS_MAX	Maximum number of reg_refresh_rate intervals from Activate to Precharge command. This field must be equal to ((tRASmax / tREFI)-1) rounded down to the next lower integer. Value for REG_T_RAS_MAX can be calculated as follows: If tRASmax = 120 us and tREFI = 15.7 us, then REG_T_RAS_MAX = ((120/15.7)-1) = 6.64. Round down to the next lower integer. Therefore, the programmed value must be 6.	RW	0x1

**Table 15-148. Register Call Summary for Register EMIF\_SDRAM\_TIM\_3**

EMIF Functional Description

- [SDRAM Output Impedance Calibration: \[0\] \[1\]](#)
- [SDRAM Refresh Scheduling: \[2\]](#)

External Memory Interface (EMIF) Programming Guide

- [Sub-sequence - EMIF Timing Initialization: \[3\] \[4\] \[5\]](#)
- [EMIF SDRAM Refresh: \[6\] \[7\] \[8\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[9\]](#)
- [EMIF Register Description: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

**Table 15-149. EMIF\_SDRAM\_TIM\_3\_SHDW**

<b>Address Offset</b>	0x0000 002C		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	SDRAM Timing 3 Shadow Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_T_CKESR_SHDW				REG_ZQ_ZQCS_SHDW				REG_T_TDQCKMAX_SHDW				REG_T_RFC_SHDW				REG_T_RAS_MAX_SHDW							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:21	REG_T_CKESR_SHDW	Shadow field for reg_t_ckesr. This field is loaded into reg_t_ckesr field in <a href="#">EMIF_SDRAM_TIM_3</a> register when SldleAck is asserted.	RW	0x2
20:15	REG_ZQ_ZQCS_SHDW	Shadow field for reg_zq_zqcs. This field is loaded into reg_zq_zqcs field in <a href="#">EMIF_SDRAM_TIM_3</a> register when SldleAck is asserted.	RW	0x08
14:13	REG_T_TDQCKMAX_SHDW	Shadow field for REG_T_TDQCKMAX. This field is loaded into <a href="#">EMIF_SDRAM_TIM_3</a> [14:13] REG_T_TDQCKMAX field when SldleAck is asserted.	RW	0x1
12:4	REG_T_RFC_SHDW	Shadow field for REG_T_RFC. This field is loaded into <a href="#">EMIF_SDRAM_TIM_3</a> [12:4] REG_T_RFC when SldleAck is asserted.	RW	0x115
3:0	REG_T_RAS_MAX_SHDW	Shadow field for REG_T_RAS_MAX. This field is loaded into <a href="#">EMIF_SDRAM_TIM_3</a> [3:0] REG_T_RAS_MAX field when SldleAck is asserted.	RW	0x1

**Table 15-150. Register Call Summary for Register EMIF\_SDRAM\_TIM\_3\_SHDW**

External Memory Interface (EMIF) Programming Guide

- [Sub-sequence - EMIF Timing Initialization: \[0\]](#)
- [EMIF SDRAM Refresh: \[1\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[2\]](#)

**Table 15-151. EMIF\_LPDDR2\_NVM\_TIM**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	LPDDR2-NVM Timing Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the LPDDR2-NVM.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	REG_NVM_T_XP	RESERVED	REG_NVM_T_WTR	REG_NVM_T_RP	REG_NVM_T_WRA	REG_NVM_T_RRD										REG_NVM_T_RCDMIN															

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30:28	REG_NVM_T_XP	Minimum number of DDR clock cycles from Powerdown exit to any command, minus one.	RW	0x1
27	RESERVED	Reserved	R	0
26:24	REG_NVM_T_WTR	Minimum number of DDR clock cycles from last Write to Read, minus one.	RW	0x1
23:20	REG_NVM_T_RP	Minimum number of DDR clock cycles from Preactive to Activate, minus one.	RW	0x2
19:16	REG_NVM_T_WRA	Minimum number of DDR clock cycles from last Write transfer to Activate, minus one.	RW	0x2
15:8	REG_NVM_T_RRD	Minimum number of DDR clock cycles from Activate to Activate for a different bank, minus one.	RW	0x0C
7:0	REG_NVM_T_RCDMIN	Minimum number of DDR clock cycles from Activate to Read or Write, minus one.	RW	0x0C

**Table 15-152. Register Call Summary for Register EMIF\_LPDDR2\_NVM\_TIM**

- External Memory Interface (EMIF) Programming Guide
- [Sub-sequence - EMIF Timing Initialization: \[0\] \[1\] \[2\]](#)
- EMIF Register Manual
- [EMIF Register Summary: \[3\]](#)
  - [EMIF Register Description: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)

**Table 15-153. EMIF\_LPDDR2\_NVM\_TIM\_SHDW**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	LPDDR2-NVM Timing Shadow Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED				REG_NVM_T_XP_SHDW				RESERVED				REG_NVM_T_WTR_SHDW				REG_NVM_T_RP_SHDW				REG_NVM_T_WRA_SHDW				REG_NVM_T_RRD_SHDW								REG_NVM_T_RCDMIN_SHDW							

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30:28	REG_NVM_T_XP_SHDW	Shadow field for REG_NVM_T_XP. This field is loaded into EMIF_LPDDR2_NVM_TIM[30:28] REG_NVM_T_XP field when SldleAck is asserted.	RW	0x0
27	RESERVED	Reserved	R	0
26:24	REG_NVM_T_WTR_SHDW	Shadow field for REG_NVM_T_WTR. This field is loaded into EMIF_LPDDR2_NVM_TIM[26:24] REG_NVM_T_WTR field when SldleAck is asserted.	RW	0x0
23:20	REG_NVM_T_RP_SHDW	Shadow field for REG_NVM_T_RP. This field is loaded into EMIF_LPDDR2_NVM_TIM[23:20] REG_NVM_T_RP field when SldleAck is asserted.	RW	0x0
19:16	REG_NVM_T_WRA_SHDW	Shadow field for REG_NVM_T_WRA. This field is loaded into EMIF_LPDDR2_NVM_TIM[19:16] REG_NVM_T_WRA field when SldleAck is asserted.	RW	0x0
15:8	REG_NVM_T_RRD_SHDW	Shadow field for REG_NVM_T_RRD. This field is loaded into EMIF_LPDDR2_NVM_TIM[15:8] REG_NVM_T_RRD field when SldleAck is asserted.	RW	0x00
7:0	REG_NVM_T_RCDMIN_SHDW	Shadow field for . This field is loaded into EMIF_LPDDR2_NVM_TIM[7:0] REG_NVM_T_RCDMIN field when SldleAck is asserted.	RW	0x00

**Table 15-154. Register Call Summary for Register EMIF\_LPDDR2\_NVM\_TIM\_SHDW**

External Memory Interface (EMIF) Programming Guide

- [Sub-sequence - EMIF Timing Initialization: \[0\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[1\]](#)

**Table 15-155. EMIF\_PWR\_MGMT\_CTRL**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	Power Management Control Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_PD_TIM				REG_DPD_EN	REG_LP_MODE	REG_SR_TIM				REG_CS_TIM					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	REG_PD_TIM	Power Mangement timer for Power-Down. The EMIF will put the external SDRAM in Power-Down mode after the EMIF is idle for these number of DDR clock cycles and if reg_lp_mode field is set to 4. Set to 0 to immediately enter Power-Down mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.	RW	0x0
11	REG_DPD_EN	Deep Power Down enable. Set to 0 for normal operation. Set to 1 to enter deep power-down mode. This mode will override the reg_lp_mode field setting.	RW	0
10:8	REG_LP_MODE	Automatic Power Management enable. Set to 1 for Clock Stop, set to 2 for Self Refresh, and set to 4 for Power-Down. All other values will disable automatic power management.	RW	0x0
7:4	REG_SR_TIM	Power Mangement timer for Self Refresh. The EMIF will put the external SDRAM in Self Refresh mode after the EMIF is idle for these number of DDR clock cycles and if reg_lp_mode field is set to 2. Set to 0 to immediately enter Self Refresh mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.	RW	0x0
3:0	REG_CS_TIM	Power Mangement timer for Clock Stop. The EMIF will put the external SDRAM in Clock Stop mode after the EMIF is idle for these number of DDR clock cycles and if reg_lp_mode field is set to 1. Set to 0 to immediately enter Clock Stop mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.	RW	0x0

**Table 15-156. Register Call Summary for Register EMIF\_PWR\_MGMT\_CTRL**

## EMIF Functional Description

- [LPDDR2-SDRAM Power-Down Mode: \[0\] \[1\] \[2\]](#)
- [LPDDR2-SDRAM Deep Power-Down Mode: \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [LPDDR2-NVM Power-Down Mode: \[8\] \[9\] \[10\]](#)
- [LPDDR2-SDRAM Self-Refresh Mode: \[11\] \[12\] \[13\]](#)

## External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[14\] \[15\] \[16\]](#)
- [Sub-sequence - EMIF Timing Initialization: \[17\] \[18\] \[19\]](#)
- [EMIF SDRAM Self-Refresh: \[20\]](#)
- [EMIF LPDDR2 \(SDRAM or NVM\) Power-Down Mode: \[21\] \[22\]](#)
- [EMIF LPDDR2-SDRAM Deep Power-Down Mode: \[23\] \[24\]](#)
- [EMIF Global Warm Reset \(Software reset\): \[25\] \[26\]](#)

## EMIF Register Manual

- [EMIF Register Summary: \[27\]](#)

**Table 15-157. EMIF\_PWR\_MGMT\_CTRL\_SHDW**

<b>Address Offset</b>	0x0000 003C		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	Power Management Control Shadow Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_PD_TIM_SHDW		RESERVED						REG_SR_TIM_SHDW		REG_CS_TIM_SHDW					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RO	0x0
15 :12	REG_PD_TIM_SHDW	Shadow field for reg_pd_tim. This field is loaded into reg_pd_tim field in Power Management Control register when SideAck is asserted.	RW	0x0
11:8	RESERVED	Reserved	RO	0x0
7:4	REG_SR_TIM_SHDW	Shadow field for reg_sr_tim. This field is loaded into reg_sr_tim field in Power Management Control register when SideAck is asserted.	RW	0x0
3:0	REG_CS_TIM_SHDW	Shadow field for reg_cs_tim. This field is loaded into reg_cs_tim field in Power Management Control register when SideAck is asserted.	RW	0x0

**Table 15-158. Register Call Summary for Register EMIF\_PWR\_MGMT\_CTRL\_SHDW**

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[0\]](#)
- [Sub-sequence - EMIF Timing Initialization: \[1\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[2\]](#)

**Table 15-159. EMIF\_LPDDR2\_MODE\_REG\_DATA**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	<p>LPDDR2 Mode Reg Data Register</p> <p>A write to this register will cause a Mode Register write command to be sent to the LPDDR2 device with write data as specified in the REG_VALUE_0 field. The address and chip select are taken from the LPDDR2_MODE_REG_CFG register.</p> <p>A read to this register will cause a Mode Register read command to be sent to the LPDDR2 device. The address and chip select are taken from the LPDDR2_MODE_REG_CFG register. The read data will appear in REG_VALUE_0 field.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_VALUE_0															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved for future use.	R	0x00000000
7:0	REG_VALUE_0	Mode register value.	RW	0x00



**Table 15-160. Register Call Summary for Register EMIF\_LPDDR2\_MODE\_REG\_DATA**

EMIF Functional Description

- [LPDDR2 SDRAM Initialization: \[0\]](#)

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[1\]](#)
- [Sub-sequence - EMIF Timing Initialization: \[2\]](#)
- [EMIF Read: \[3\]](#)
- [EMIF Write: \[4\]](#)
- [EMIF SDRAM Refresh: \[5\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[6\]](#)

**Table 15-161. EMIF\_LPDDR2\_MODE\_REG\_CFG**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	LPDDR2 Mode Reg Config Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_CS REG_REFRESH_EN		RESERVED															REG_ADDRESS														

Bits	Field Name	Description	Type	Reset
31	REG_CS	Chip select to issue mode register command. Set to 0 for CS0 and set to 1 for CS1.	RW	0
30	REG_REFRESH_EN	Refresh Enable after MRW write. If a Mode Data register write occurs with this bit set to 1, the refresh operations will commence.	RW	0x0
29:8	RESERVED	Reserved	R	0x000000
7:0	REG_ADDRESS	Mode register address.	RW	0x00

**Table 15-162. Register Call Summary for Register EMIF\_LPDDR2\_MODE\_REG\_CFG**

EMIF Functional Description

- [LPDDR2 SDRAM Initialization: \[0\] \[1\]](#)

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[2\]](#)
- [EMIF Read: \[3\] \[4\]](#)
- [EMIF Write: \[5\] \[6\]](#)
- [EMIF SDRAM Refresh: \[7\] \[8\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[9\]](#)

**Table 15-163. EMIF\_L3\_CONFIG**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	Config Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				REG_SYS_THRESH_MAX				RESERVED				REG_LL_THRESH_MAX				RESERVED								REG_PR_OLD_COUNT							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved for future use.	R	0x0
27:24	REG_SYS_THRESH_MAX	System L3 Threshold Maximum. The number of commands the system interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1. Since the low-latency interface has effectively a higher priority, the only way for the system interface to use all command FIFO entries is to set the REG_LL_THRESH_MAX to zero.	RW	0x7
23:20	RESERVED	Reserved for future use.	R	0x0
19:16	REG_LL_THRESH_MAX	Low-latency L3 Threshold Maximum. The number of commands the low latency interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1.	RW	0x7
15:8	RESERVED	Reserved for future use.	R	0x0
7:0	REG_PR_OLD_COUNT	Priority Raise Old Counter. Number of memory transfers after which the EMIF momentarily raises the priority of old commands in the L3 Command FIFO.	RW	0xFF

**Table 15-164. Register Call Summary for Register EMIF\_L3\_CONFIG**

## EMIF Functional Description

- [FIFOs Description: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Arbitration: \[7\]](#)

## External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[8\]](#)

## EMIF Register Manual

- [EMIF Register Summary: \[9\]](#)

**Table 15-165. EMIF\_L3\_CFG\_VAL\_1**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	Config Value 1 Register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_SYS_BUS_WIDTH		REG_LL_BUS_WIDTH		RESERVED													REG_WR_FIFO_DEPTH				REG_CMD_FIFO_DEPTH										

Bits	Field Name	Description	Type	Reset
31:30	REG_SYS_BUS_WIDTH	System L3 data bus width 0 = 32-bit wide, 1 = 64-bit wide, 2 = 128-bit wide, 3 = Reserved	R	0x2
29:28	REG_LL_BUS_WIDTH	Low-latency L3 data bus width 0 = 32-bit wide, 1 = 64-bit wide, 2 = 128-bit wide, 3 = Reserved	R	0x1
27:16	RESERVED	Reserved for future use.	R	0x000
15:8	REG_WR_FIFO_DEPTH	Write Data FIFO depth	R	0x11
7:0	REG_CMD_FIFO_DEPTH	Command FIFO depth	R	0x0A

**Table 15-166. Register Call Summary for Register EMIF\_L3\_CFG\_VAL\_1**

- EMIF Register Manual
- [EMIF Register Summary: \[0\]](#)

**Table 15-167. EMIF\_L3\_CFG\_VAL\_2**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	Config Value 2 Register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_RREG_FIFO_DEPTH				REG_RSD_FIFO_DEPTH				REG_RCMD_FIFO_DEPTH															

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:16	REG_RREG_FIFO_DEPTH	Register Read Data FIFO depth	R	0x04
15:8	REG_RSD_FIFO_DEPTH	SDRAM Read Data FIFO depth	R	0x27
7:0	REG_RCMD_FIFO_DEPTH	Read Command FIFO depth	R	0x27

**Table 15-168. Register Call Summary for Register EMIF\_L3\_CFG\_VAL\_2**

EMIF Register Manual

- [EMIF Register Summary: \[0\]](#)

**Table 15-169. EMIF\_PERF\_CNT\_1**

<b>Address Offset</b>	0x0000 0080	
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b> EMIF
<b>Description</b>	Performance Counter 1 Register	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_COUNTER1																															

Bits	Field Name	Description	Type	Reset
31:0	REG_COUNTER1	32-bit counter that can be configured as specified in the Performance Counter Config Register and Performance Counter Master Region Select Register.	R	0x0000 0000

**Table 15-170. Register Call Summary for Register EMIF\_PERF\_CNT\_1**

EMIF Register Manual

- [EMIF Register Summary: \[0\]](#)

**Table 15-171. EMIF\_PERF\_CNT\_2**

<b>Address Offset</b>	0x0000 0084	
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b> EMIF
<b>Description</b>	Performance Counter 2 Register	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_COUNTER2																															

Bits	Field Name	Description	Type	Reset
31:0	REG_COUNTER2	32-bit counter that can be configured as specified in the Performance Counter Config Register and Performance Counter Master Region Select Register.	R	0x0000 0000

**Table 15-172. Register Call Summary for Register EMIF\_PERF\_CNT\_2**

EMIF Register Manual

- [EMIF Register Summary: \[0\]](#)

**Table 15-173. EMIF\_PERF\_CNT\_CFG**

<b>Address Offset</b>	0x0000 0088		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	Performance Counter Config Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_CNTR2_MCONNID_EN	REG_CNTR2_REGION_EN	RESERVED										REG_CNTR2_CFG	REG_CNTR1_MCONNID_EN	REG_CNTR1_REGION_EN	RESERVED										REG_CNTR1_CFG						

Bits	Field Name	Description	Type	Reset
31	REG_CNTR2_MCONNID_EN	MConnID filter enable for Performance Counter 2 register.	RW	0
30	REG_CNTR2_REGION_EN	Chip Select filter enable for Performance Counter 2 register.	RW	0
29:20	RESERVED	Reserved for future use	R	0x000
19:16	REG_CNTR2_CFG	Filter configuration for Performance Counter 2. Refer to <a href="#">Table 15-175</a> for details.	RW	0x1
15	REG_CNTR1_MCONNID_EN	MConnID filter enable for Performance Counter 1 register.	RW	0
14	REG_CNTR1_REGION_EN	Chip Select filter enable for Performance Counter 1 register.	RW	0
13:4	RESERVED	Reserved for future use	R	0x000
3:0	REG_CNTR1_CFG	Filter configuration for Performance Counter 1. Refer to <a href="#">Table 15-175</a> for details.	RW	0x0

**Table 15-174. Register Call Summary for Register EMIF\_PERF\_CNT\_CFG**

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[0\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[1\]](#)

**Table 15-175. Performance Counter Filter Configuration**

CNTRn_CFG	CNTRn_REGION_EN	CNTRn_MCONNID_EN	Description
0x0	0x0	0x0 or 0x1	Count total SDRAM accesses
0x1	0x0	0x0 or 0x1	Count total SDRAM activates
0x2	0x0 or 0x1	0x0 or 0x1	Count total reads
0x3	0x0 or 0x1	0x0 or 0x1	Count total writes
0x4	0x0	0x0	Count number of DDR clock cycles L3 Command FIFO is full
0x5	0x0	0x0	Count number of DDR clock cycles L3 Write Data FIFO is full
0x6	0x0	0x0	Count number of DDR clock cycles L3 Read Data FIFO is full
0x7	0x0	0x0	Count number of DDR clock cycles L3 Return Command FIFO is full

**Table 15-175. Performance Counter Filter Configuration (continued)**

CNTRn_CFG	CNTRn_REGION_EN	CNTRn_MCONNID_EN	Description
0x8	0x0 or 0x1	0x0 or 0x1	Count number of priority elevations
0x9	0x0	0x0	Count number of DDR clock cycles that a command was pending
0xA	0x0	0x0	Count number of DDR clock cycles for which the memory data bus was transferring data.
0xB - 0xF	0x0	0x0	Reserved for future use.

**NOTE:** When MReqDebug is set to a 1 for a particular L3 command, the performance counters will not be incremented for that particular command if the CNTRn\_CFG values are equal 0x0, 0x1, 0x2, 0x3, or 0xA.

Table 15-175 shows the possible filter configurations for the two performance counters. These filter configurations can be used in conjunction with a L3 master ID and/or an external chip select to obtain performance statistics for a particular L3 master and/or an external chip select.

**Table 15-176. EMIF\_PERF\_CNT\_SEL**

<b>Address Offset</b>	0x0000 008C	<b>Instance</b>	EMIF
<b>Physical Address</b>	See Table 15-126		
<b>Description</b>	Performance Counter Master Region Select Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_MCONNID2								RESERVED				REG_REGION_SEL2	REG_MCONNID1								RESERVED				REG_REGION_SEL1						

Bits	Field Name	Description	Type	Reset
31:24	REG_MCONNID2	MConnID for Performance Counter 2 register.	RW	0x00
23:18	RESERVED	Reserved for future use.	R	0x00
17:16	REG_REGION_SEL2	MAddrSpace for Performance Counter 2 register.	RW	0x0
15:8	REG_MCONNID1	MConnID for Performance Counter 1 register.	RW	0x00
7:2	RESERVED	Reserved for future use.	R	0x00
1:0	REG_REGION_SEL1	MAddrSpace for Performance Counter 1 register.	RW	0x0

**Table 15-177. Register Call Summary for Register EMIF\_PERF\_CNT\_SEL**

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[0\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[1\]](#)

**Table 15-178. EMIF\_PERF\_CNT\_TIM**

<b>Address Offset</b>	0x0000 0090	
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b> EMIF
<b>Description</b>	Performance Counter Time Register. This is a free running counter.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_TOTAL_TIME																															

Bits	Field Name	Description	Type	Reset
31:0	REG_TOTAL_TIME	32-bit counter that continuously counts number for DDR clock cycles elapsed after EMIF is brought out of reset.	R	0x0000 0000

**Table 15-179. Register Call Summary for Register EMIF\_PERF\_CNT\_TIM**

EMIF Register Manual

- [EMIF Register Summary: \[0\]](#)

**Table 15-180. EMIF\_READ\_IDLE\_CTRL**

<b>Address Offset</b>	0x0000 0098	
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b> EMIF
<b>Description</b>	Read Idle Control Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										REG_READ_IDLE_LEN	RESERVED								REG_READ_IDLE_INTERVAL												

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved for future use.	R	0x0
19:16	REG_READ_IDLE_LEN	The Read Idle Length field determines the minimum size (reg_read_idle_len-1 clock cycles) of Read Idle window for the read idle detection as well as the force read idle time.	RW	0x5
15:9	RESERVED	Reserved for future use.	R	0x0
8:0	REG_READ_IDLE_INTERVAL	The Read Idle Interval field determines the maximum interval ((REG_READ_IDLE_INTERVAL - 1) * 64 clock cycles) between read idle detections or force. A value of zero disables the read idle function.	RW	0x0

**Table 15-181. Register Call Summary for Register EMIF\_READ\_IDLE\_CTRL**

EMIF Functional Description

- [Read Idle Window: \[0\] \[1\] \[2\]](#)

External Memory Interface (EMIF) Programming Guide

- [Sub-sequence - EMIF Timing Initialization: \[3\] \[4\] \[5\] \[6\]](#)

**Table 15-181. Register Call Summary for Register EMIF\_READ\_IDLE\_CTRL (continued)**

EMIF Register Manual

- [EMIF Register Summary: \[7\]](#)
- [EMIF Register Description: \[8\] \[9\]](#)

**Table 15-182. EMIF\_READ\_IDLE\_CTRL\_SHDW**

<b>Address Offset</b>	0x0000 009C	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	Read Idle Control Shadow Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_READ_IDLE_LEN_SHDW				RESERVED								REG_READ_IDLE_INTERVAL_SHDW											

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved for future use.	R	0x0
19:16	REG_READ_IDLE_LEN_SHDW	Shadow field for REG_READ_IDLE_LEN. This field is loaded into <a href="#">EMIF_READ_IDLE_CTRL[19:16]</a> REG_READ_IDLE_LEN field when SIdleAck is asserted.	RW	0x5
15:9	RESERVED	Reserved for future use.	R	0x0
8:0	REG_READ_IDLE_INTERVAL_S HDW	Shadow field for REG_READ_IDLE_INTERVAL. This field is loaded into <a href="#">EMIF_READ_IDLE_CTRL[8:0]</a> REG_READ_IDLE_INTERVAL field when SIdleAck is asserted.	RW	0x0

**Table 15-183. Register Call Summary for Register EMIF\_READ\_IDLE\_CTRL\_SHDW**

External Memory Interface (EMIF) Programming Guide

- [Sub-sequence - EMIF Timing Initialization: \[0\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[1\]](#)



**Table 15-184. EMIF\_IRQSTATUS\_RAW\_SYS**

<b>Address Offset</b>	0x0000 00A4		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	System L3 Interrupt Raw Status Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_TA_SYS		REG_ERR_SYS													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved for future use.	R	0x0000 0000
1	REG_TA_SYS	Raw status of system L3 interrupt for SDRAM temperature alert. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.	RW	0
0	REG_ERR_SYS	Raw status of system L3 interrupt for command or address error. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.	RW	0

**Table 15-185. Register Call Summary for Register EMIF\_IRQSTATUS\_RAW\_SYS**

EMIF Functional Description

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\]](#)

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[4\]](#)
- [Interrupt Servicing: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[11\]](#)

**Table 15-186. EMIF\_IRQSTATUS\_RAW\_LL**

<b>Address Offset</b>	0x0000 00A8		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	Low-Latency L3 Interrupt Raw Status Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_TA_LL		REG_ERR_LL													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved for future use.	R	0x0000 0000
1	REG_TA_LL	Raw status of low-latency L3 interrupt or SDRAM temperature alert. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.	RW	0

Bits	Field Name	Description	Type	Reset
0	REG_ERR_LL	Raw status of low-latency L3 interrupt for command or address error. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.	RW	0

**Table 15-187. Register Call Summary for Register EMIF\_IRQSTATUS\_RAW\_LL**

## EMIF Functional Description

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\]](#)

## External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[4\]](#)
- [Interrupt Servicing: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

## EMIF Register Manual

- [EMIF Register Summary: \[11\]](#)

**Table 15-188. EMIF\_IRQSTATUS\_SYS**

<b>Address Offset</b>	0x0000 00AC		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	System L3 Interrupt Status Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	REG_TA_SYS		REG_ERR_SYS												

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved for future use.	R	0x0000 0000
1	REG_TA_SYS	Enabled status of system L3 interrupt for SDRAM temperature alert. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). Writing a 0 has no effect.	RW	0
0	REG_ERR_SYS	Enabled status of system L3 interrupt interrupt for command or address error. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). Writing a 0 has no effect.	RW	0

**Table 15-189. Register Call Summary for Register EMIF\_IRQSTATUS\_SYS**

## EMIF Functional Description

- [Interrupt Requests: \[0\] \[1\] \[2\]](#)
- [SDRAM Temperature Monitoring: \[3\]](#)

## External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[4\]](#)
- [Interrupt Servicing: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

## EMIF Register Manual

- [EMIF Register Summary: \[11\]](#)

**Table 15-190. EMIF\_IRQSTATUS\_LL**

<b>Address Offset</b>	0x0000 00B0		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	Low-Latency L3 Interrupt Status Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_TA_LL		REG_ERR_LL													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved for future use.	R	0x0000 0000
1	REG_TA_LL	Enabled status of low-latency L3 interrupt for SDRAM temperature alert. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). Writing a 0 has no effect.	RW W1toClr	0
0	REG_ERR_LL	Enabled status of low-latency L3 interrupt for command or address error. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). Writing a 0 has no effect.	RW W1toClr	0

**Table 15-191. Register Call Summary for Register EMIF\_IRQSTATUS\_LL**

EMIF Functional Description

- [Interrupt Requests: \[0\] \[1\] \[2\]](#)
- [SDRAM Temperature Monitoring: \[3\]](#)

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[4\]](#)
- [Interrupt Servicing: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[11\]](#)

**Table 15-192. EMIF\_IRQENABLE\_SET\_SYS**

<b>Address Offset</b>	0x0000 00B4		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	System L3 Interrupt Enable Set Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_EN_TA_SYS		REG_EN_ERR_SYS													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved for future use.	R	0x0000 0000
1	REG_EN_TA_SYS	Enable set for system L3 interrupt for SDRAM temperature alert. . Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0
0	REG_EN_ERR_SYS	Enable set for system L3 interrupt for command or address error. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0

**Table 15-193. Register Call Summary for Register EMIF\_IRQENABLE\_SET\_SYS**

## EMIF Functional Description

- [Interrupt Requests: \[0\] \[1\]](#)

## External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[2\]](#)
- [EMIF Temperature Monitoring Mode: \[3\]](#)
- [Interrupt Servicing: \[4\]](#)

## EMIF Register Manual

- [EMIF Register Summary: \[5\]](#)

**Table 15-194. EMIF\_IRQENABLE\_SET\_LL**

<b>Address Offset</b>	0x0000 00B8	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	Low-Latency L3 Interrupt Enable Set Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_EN_TA_LL		REG_EN_ERR_LL													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved for future use.	R	0x0000 0000
1	REG_EN_TA_LL	Enable set for low-latency L3 interrupt for SDRAM temperature alert. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0
0	REG_EN_ERR_LL	Enable set for low-latency L3 interrupt for command or address error. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0

**Table 15-195. Register Call Summary for Register EMIF\_IRQENABLE\_SET\_LL**

EMIF Functional Description

- [Interrupt Requests: \[0\] \[1\]](#)

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[2\]](#)
- [EMIF Temperature Monitoring Mode: \[3\]](#)
- [Interrupt Servicing: \[4\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[5\]](#)

**Table 15-196. EMIF\_IRQENABLE\_CLR\_SYS**

<b>Address Offset</b>	0x0000 00BC	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	System L3 Interrupt Enable Clear Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_EN_TA_SYS		REG_EN_ERR_SYS													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved for future use	R	0x0000 0000
1	REG_EN_TA_SYS	Enable clear for system L3 interrupt for SDRAM temperature alert. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0
0	REG_EN_ERR_SYS	Enable clear for system L3 interrupt for command or address error. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0

**Table 15-197. Register Call Summary for Register EMIF\_IRQENABLE\_CLR\_SYS**

EMIF Functional Description

- [Interrupt Requests: \[0\] \[1\]](#)

External Memory Interface (EMIF) Programming Guide

- [Interrupt Servicing: \[2\] \[3\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[4\]](#)

**Table 15-198. EMIF\_IRQENABLE\_CLR\_LL**

<b>Address Offset</b>	0x0000 00C0		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	Low-Latency L3 Interrupt Enable Clear Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_EN_TA_LL		REG_EN_ERR_LL													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved for future use.	R	0x0000 0000
1	REG_EN_TA_LL	Enable clear for low-latency L3 interrupt for SDRAM temperature alert. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0
0	REG_EN_ERR_LL	Enable clear for low-latency L3 interrupt for command or address error. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0

**Table 15-199. Register Call Summary for Register EMIF\_IRQENABLE\_CLR\_LL**

EMIF Functional Description

- [Interrupt Requests: \[0\] \[1\]](#)

External Memory Interface (EMIF) Programming Guide

- [Interrupt Servicing: \[2\] \[3\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[4\]](#)

**Table 15-200. EMIF\_ZQ\_CONFIG**

<b>Address Offset</b>	0x0000 00C8		
<b>Physical Address</b>	See <a href="#">Table 15-126</a>	<b>Instance</b>	EMIF
<b>Description</b>	SDRAM Output Impedance Calibration Config Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_ZQ_CS1EN	REG_ZQ_CS0EN	REG_ZQ_DUALCALEN	REG_ZQ_SFEXITEN	RESERVED								REG_ZQ_ZQINIT_MULT	REG_ZQ_ZQCL_MULT	REG_ZQ_REFINTERVAL																	

Bits	Field Name	Description	Type	Reset
31	REG_ZQ_CS1EN	Writing a 1 enables ZQ calibration for CS1.	RW	0x0
30	REG_ZQ_CS0EN	Writing a 1 enables ZQ calibration for CS0.	RW	0x0
29	REG_ZQ_DUALCALEN	ZQ Dual Calibration enable. Allows both ranks to be ZQ calibrated simultaneously. Setting this bit requires both chip selects to have a separate calibration resistor per device.	RW	0x0
28	REG_ZQ_SFEXITEN	ZQCL on Self-Refresh, Active Power-Down, and Precharge Power-Down exit enable. Writing a 1 enables the issuing of ZQCL on Self-Refresh, Active Power-Down, and Precharge Power-Down exit.	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19:18	REG_ZQ_ZQINIT_MULT	Indicates the number of ZQCL intervals that make up a ZQINIT interval, minus one..	RW	0x0
17:16	REG_ZQ_ZQCL_MULT	Indicates the number of ZQCS intervals that make up a ZQCL interval, minus one. ZQCS interval is defined by reg_zq_zqcs in <a href="#">EMIF_SDRAM_TIM_3</a> .	RW	0x0
15:0	REG_ZQ_REFINTERVAL	Number of refresh periods between ZQCS commands. This field supports between one refresh period to 256 ms between ZQCS calibration commands. Refresh period is defined by reg_refresh_rate in <a href="#">EMIF_SDRAM_REF_CTRL</a> register.	RW	0x0000

**Table 15-201. Register Call Summary for Register EMIF\_ZQ\_CONFIG**

EMIF Functional Description

- [SDRAM Output Impedance Calibration: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)

External Memory Interface (EMIF) Programming Guide

- [EMIF Output Impedance Calibration Mode: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[12\]](#)

**Table 15-202. EMIF\_TEMP\_ALERT\_CONFIG**

<b>Address Offset</b>	0x0000 00CC	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	Temperature Alert Config Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
REG_TA_CS1EN	REG_TA_CS0EN	RESERVED	REG_TA_SFEXITEN	REG_TA_DEVWDT	REG_TA_DEVCNT	RESERVED																										
																REG_TA_REFINTERVAL																

Bits	Field Name	Description	Type	Reset
31	REG_TA_CS1EN	Writing a 1 enables temperature alert polling for CS1.	RW	0x0
30	REG_TA_CS0EN	Writing a 1 enables temperature alert polling for CS0.	RW	0x0
29	RESERVED	Reserved	R	0x0
28	REG_TA_SFEXITEN	Temperature Alert Poll on Self-Refresh, Active Power-Down, and Precharge Power-Down exit enable. Writing a 1 enables the issuing of a temperature alert poll on Self-Refresh exit.	RW	0x0

Bits	Field Name	Description	Type	Reset
27:26	REG_TA_DEVWDT	This field indicates how wide a physical device is. It is used in conjunction with the reg_ta_devcnt register to determine which byte lanes contain the temperature alert info. A value of 0 = eight bit wide, 1 = sixteen bit wide, 2 = thirty two bit wide. All others are reserved. If this field is set to 1 and the reg_ta_devcnt field is set to one the byte mask for checking will be 4'b0101.	RW	0x0
25:24	REG_TA_DEVCNT	This field indicates which external byte lanes contain a device for temperature monitoring. A value of 0 = one device, 1 = two devices, 2 = four devices. All other reserved.	RW	0x0
23:22	RESERVED	Reserved	R	0x0
21:0	REG_TA_REFINTERVAL	Number of refresh periods between temperature alert polls. This field supports between one refresh period to 10 seconds between temperature alert polls. Refresh period is defined by reg_refresh_rate in SDRAM Refresh Control register.	RW	0x000000

**Table 15-203. Register Call Summary for Register EMIF\_TEMP\_ALERT\_CONFIG**

EMIF Functional Description

- [SDRAM Temperature Monitoring: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)

External Memory Interface (EMIF) Programming Guide

- [Sub-sequence - EMIF Timing Initialization: \[6\]](#)
- [EMIF Temperature Monitoring Mode: \[7\] \[8\] \[9\] \[10\] \[11\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[12\]](#)

**Table 15-204. EMIF\_L3\_ERR\_LOG**

<b>Address Offset</b>	0x0000 00D0	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	Error Log Register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REG_MADDRSPACE	REG_MBURSTSEQ	REG_MCMD	REG_MCONNID												

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved for future use.	R	0x0000
15:14	REG_MADDRSPACE	Address space of the first errored transaction. 0x0: SDRAM 0x1: LPDDR2-NVM 0x2: reserved 0x3: internal registers	R	0x0
13:11	REG_MBURSTSEQ	Addressing mode of the first errored transaction. (see <a href="#">Chapter 13, Interconnect</a> for more information)	R	0x0
10:8	REG_MCMD	Command type of the first errored transaction. (see <a href="#">Chapter 13, Interconnect</a> for more information)	R	0x0
7:0	REG_MCONNID	Connection ID of the first errored transaction.	R	0x00



**Table 15-205. Register Call Summary for Register EMIF\_L3\_ERR\_LOG**

External Memory Interface (EMIF) Programming Guide

- [Interrupt Servicing: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[8\]](#)

**Table 15-206. EMIF\_DDR\_PHY\_CTRL\_1**

<b>Address Offset</b>	0x0000 00E4	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	DDR PHY Control 1 Register. This register is used to control the DDR PHY. The bit field definitions are DDR PHY specific.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
RESERVED								REG_PHY_FREEZE_DELAY_CODE_POSTAMBLE								REG_PHY_FREEZE_DELAY_CODE_PREAMBLE								REG_DLL_MASTER_SW_CODE_CTRL								REG_DLL_SLAVE_DLY_CTRL								REG_READ_LATENCY			

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	RW	0x0
29:26	REG_PHY_FREEZE_DELAY_CODE_POSTAMBLE	Postamble time respected by the DATA PHY freezing the DLL slave code after reads	RW	0x0
25:22	REG_PHY_FREEZE_DELAY_CODE_PREAMBLE	Preamble time respected by the DATA PHY before freezing the DLL slave code during reads	RW	0x0
21:12	REG_DLL_MASTER_SW_CODE_CTRL	DLL delay code when in software override mode	RW	0x1FF
11:4	REG_DLL_SLAVE_DLY_CTRL	DLL slave delay ratio control	RW	0xFF
3:0	REG_READ_LATENCY	This field defines the read latency for the read data from SDRAM in number of DDR clock cycles. This field is used by the EMIF as well as the PHY. The EMIF will expect the first read data to arrive (reg_read_latency + 3) DDR clock cycles from the read command.	RW	0x0

**Table 15-207. Register Call Summary for Register EMIF\_DDR\_PHY\_CTRL\_1**

EMIF Functional Description

- [SDRAM Read: \[0\]](#)

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[1\] \[2\]](#)
- [EMIF Read: \[3\]](#)

**Table 15-207. Register Call Summary for Register EMIF\_DDR\_PHY\_CTRL\_1 (continued)**

EMIF Register Manual

- [EMIF Register Summary: \[4\]](#)
- [EMIF Register Description: \[5\] \[6\]](#)

**Table 15-208. EMIF\_DDR\_PHY\_CTRL\_1\_SHDW**

<b>Address Offset</b>	0x0000 00E8	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	DDR PHY Control 1 Shadow Register. Shadow field for <a href="#">EMIF_DDR_PHY_CTRL_1</a> .		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				REG_PHY_FREEZE_DELAY_CODE_POSTAMBLE_SHDW				REG_PHY_FREEZE_DELAY_CODE_PREAMBLE_SHDW				REG_DLL_MASTER_SW_CODE_CTRL_SHDW				REG_DLL_SLAVE_DLY_CTRL_SHDW				REG_READ_LATENCY_SHDW											

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	RW	0x0
29:26	REG_PHY_FREEZE_DELAY_CODE_POSTAMBLE_SHDW	Shadow field for REG_PHY_FREEZE_DELAY_CODE_POSTAMBLE	RW	0x0
25:22	REG_PHY_FREEZE_DELAY_CODE_PREAMBLE_SHDW	Shadow field for REG_PHY_FREEZE_DELAY_CODE_PREAMBLE	RW	0x0
21:12	REG_DLL_MASTER_SW_CODE_CTRL_SHDW	Shadow field for REG_DLL_MASTER_SW_CODE_CTRL	RW	0x1FF
11:4	REG_DLL_SLAVE_DLY_CTRL_SHDW	Shadow field for REG_DLL_SLAVE_DLY_CTRL	RW	0xFF
3:0	REG_READ_LATENCY_SHDW	Shadow field for REG_READ_LATENCY. This field is loaded into <a href="#">EMIF_DDR_PHY_CTRL_1[3:0]</a> REG_READ_LATENCY field when SidleAck is asserted.	RW	0x0

**Table 15-209. Register Call Summary for Register EMIF\_DDR\_PHY\_CTRL\_1\_SHDW**

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[0\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[1\]](#)

**Table 15-210. EMIF\_DDR\_PHY\_CTRL\_2**

<b>Address Offset</b>	0x0000 00EC	<b>Instance</b>	EMIF
<b>Physical Address</b>	See <a href="#">Table 15-126</a>		
<b>Description</b>	DDR PHY Control 2 Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_DDR_PHY_CTRL_2																															

Bits	Field Name	Description	Type	Reset
31:0	REG_DDR_PHY_CTRL_2	This register is used to control the DDR PHY. The bit field definitions are DDR PHY specific.	RW	0x0000 0000

**Table 15-211. Register Call Summary for Register EMIF\_DDR\_PHY\_CTRL\_2**

External Memory Interface (EMIF) Programming Guide

- [EMIF Module Global Initialization: \[0\]](#)

EMIF Register Manual

- [EMIF Register Summary: \[1\]](#)

## 15.4 General-Purpose Memory Controller

### 15.4.1 GPMC Overview

The general-purpose memory controller (GPMC) is a unified memory controller dedicated to interfacing external memory devices:

- Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices
- Asynchronous, synchronous, and page mode (only available in non-muxed mode) burst NOR flash devices
- NAND Flash
- Pseudo-SRAM devices

### 15.4.2 GPMC Environment

This section describes the GPMC application fields from an environment point of view (external connections). It describes GPMC connectivity options, and gives three possible interfaces.

#### 15.4.2.1 GPMC Modes

This section shows three GPMC external connections options:

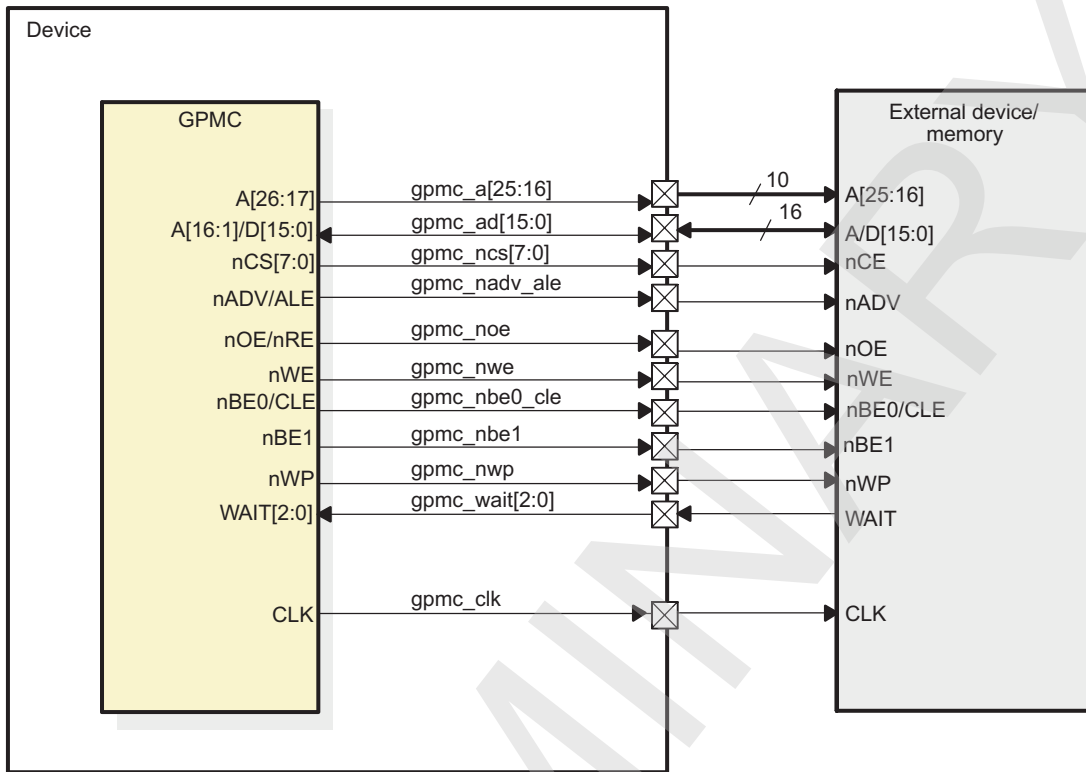
- [Figure 15-50](#) shows a connection between the GPMC and a 16-bit synchronous address/data-multiplexed (or AAD-multiplexed, but this protocol use less address pins) external memory device ([Figure 15-50](#)).
- [Figure 15-50](#) shows a connection between the GPMC and a 16-bit synchronous nonmultiplexed external memory device ([Figure 15-51](#)).
- [Figure 15-52](#) shows a connection between the GPMC and a 16-bit NAND device ([Figure 15-52](#)).

---

**NOTE:** The device does not provide the A0 byte address line required for random-byte addressable 8-bit wide device interfacing (for multiplexed and nonmultiplexed protocol). Hence, an 8-bit device must be connected to the D[7:0] / gpmc\_ad[7:0] data bus (rather than D[15:8] / gpmc\_ad[15:8]) of the GPMC controller. This limits the use of 8-bit wide device interfacing to byte-alias access.

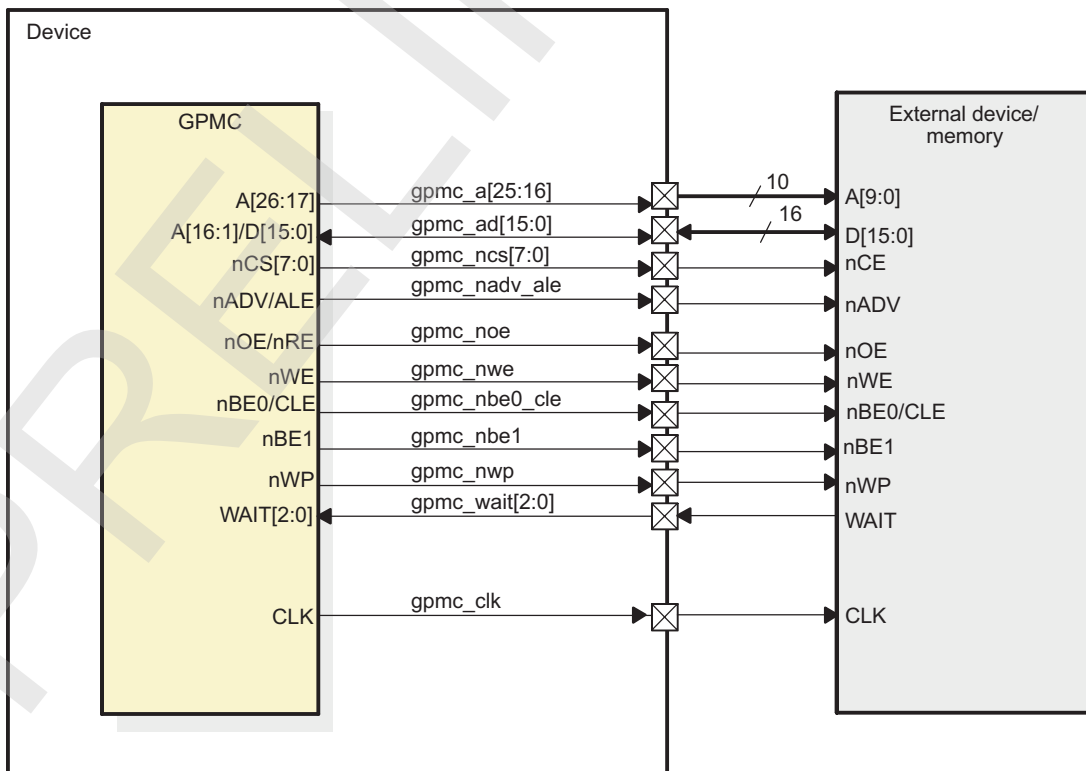
---

Figure 15-50. GPMC to 16-Bit Address/Data-Multiplexed Memory



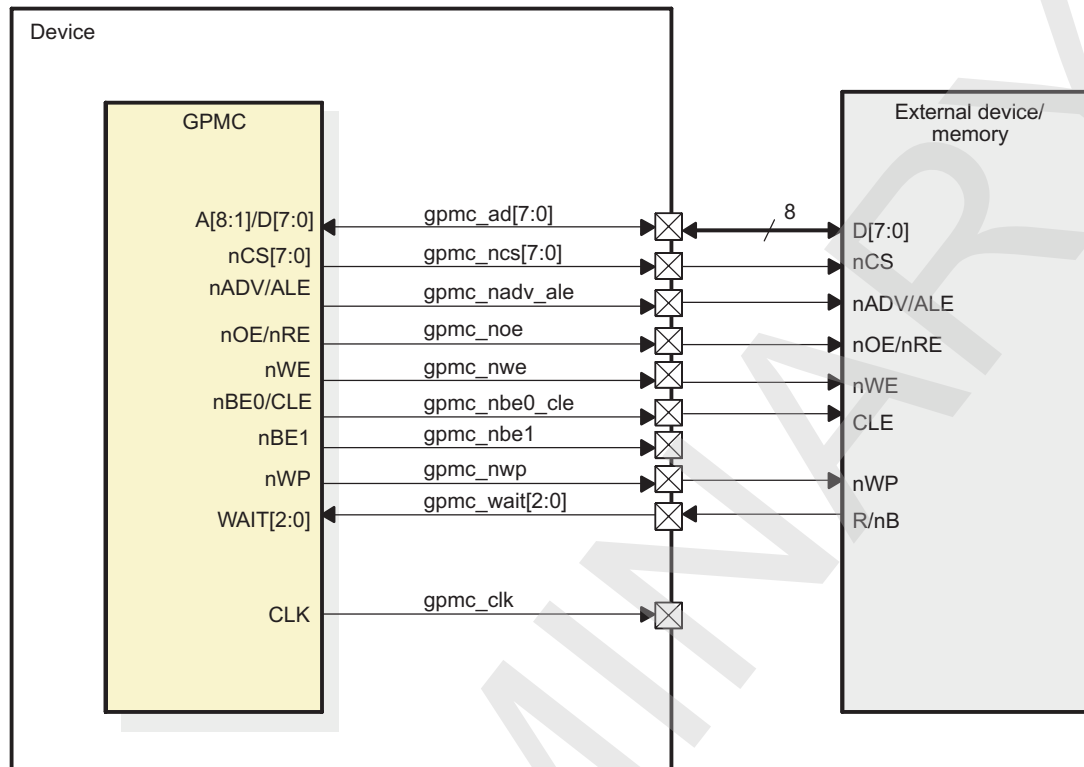
gpmc-002

Figure 15-51. GPMC to 16-Bit Nonmultiplexed Memory



gpmc-045

Figure 15-52. GPMC to 8-Bit NAND Device



gpmc-003

### 15.4.2.2 GPMC Signals

Table 15-212 lists the GPMC subsystem I/O pins.

Table 15-212. GPMC I/O Description

Pin Name	I/O	Description
gpmc_a[25:16]	O	Address
gpmc_ad[15:0]	I/O	Data
gpmc_ncs[7:0]	O	Chip-selects (active low)
gpmc_clk	O	Clock generated for the external memory or device
gpmc_nadv_ale	O	Address valid (active low). Also used as address latch enable (active high) for NAND protocol memories.
gpmc_noe_nre	O	Output enable (active low). Also used as read enable (active low) for NAND protocol memories.
gpmc_nwe	O	Write enable (active low)
gpmc_nbe0_cle	O	Lower-byte enable (active low). Also used as command latch enable for NAND protocol memories.
gpmc_nbe1	O	Upper-byte enable (active low).
gpmc_nwp	O	Write protect (active low)
gpmc_wait[2:0]	I	External wait signal for NOR and NAND protocol memories. The wait signals can be mapped on any of the chip-select.
gpmc_io_dir	O	gpmc_ad[15:0] signal direction control: Low during transmit (for write access: data OUT from GPMC to memory), High during receive (for read access: data IN from memory to GPMC)

Table 15-213 shows the use of address and data GPMC controller pins based on the type of external device.

**Table 15-213. GPMC Pin Multiplexing Options**

GPMC Pin	Multiplexed Address Data 16-Bit Device	Nonmultiplexed Address Data 16-Bit Device With LIMITED- ADDRESS Bit Enabled	16-Bit NAND Device	8-Bit NAND Device
gpmc_a[25]	A26	A10	Not used	Not used
gpmc_a[24]	A25	A9	Not used	Not used
gpmc_a[23]	A24	A8	Not used	Not used
gpmc_a[22]	A23	A7	Not used	Not used
gpmc_a[21]	A22	A6	Not used	Not used
gpmc_a[20]	A21	A5	Not used	Not used
gpmc_a[19]	A20	A4	Not used	Not used
gpmc_a[18]	A19	A3	Not used	Not used
gpmc_a[17]	A18	A2	Not used	Not used
gpmc_a[16]	A17	A1	Not used	Not used
gpmc_ad[15]	A16/D15	D15	D15	Not used
gpmc_ad[14]	A15/D14	D14	D14	Not used
gpmc_ad[13]	A14/D13	D13	D13	Not used
gpmc_ad[12]	A13/D12	D12	D12	Not used
gpmc_ad[11]	A12/D11	D11	D11	Not used
gpmc_ad[10]	A11/D10	D10	D10	Not used
gpmc_ad[9]	A10/D9	D9	D9	Not used
gpmc_ad[8]	A9/D8	D8	D8	Not used
gpmc_ad[7]	A8/D7	D7	D7	D7
gpmc_ad[6]	A7/D6	D6	D6	D6
gpmc_ad[5]	A6/D5	D5	D5	D5
gpmc_ad[4]	A5/D4	D4	D4	D4
gpmc_ad[3]	A4/D3	D3	D3	D3
gpmc_ad[2]	A3/D2	D2	D2	D2
gpmc_ad[1]	A2/D1	D1	D1	D1
gpmc_ad[0]	A1/D0	D0	D0	D0

When using non-multiplexed memories, it is recommended to enable [GPMC\\_CONFIG\[1\] LIMITEDADDRESS](#). This bit forces A[26:11] to 1 on the GPMC I/O side. Thus, only 2Kbytes of memory address space is accessible regardless of the memory size.

With all device types, the GPMC does not drive unnecessary address lines. They stay at their reset value of 0x00.

Address mapping supports address/data-multiplexed 16-bit wide devices:

- The NOR flash memory controller still supports non-multiplexed address and data memory devices.
- Multiplexing mode can be selected through the [GPMC\\_CONFIG1\\_i\[9:8\] MUXADDDATA](#) bit field (i = 0 to 3).
- Asynchronous page mode is not supported for multiplexed address and data devices.

---

**NOTE:** The numbering for A (address) is starting from 1 instead of 0 such as D (data) because the A0 signal is handled within the GPMC but is not driven to any pad. The signal is only relevant to 8-bit accesses and this configuration is not recommended. If such a memory or device is connected, only every second word in the memory is accessible, that is, each 16-bit aligned 8-bit word is mapped but not the non-16-bit aligned ones. Any access to a non-aligned one will actually result in accessing the associated aligned one, at address A1 instead of address A.

---



### 15.4.3 GPMC Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

- No STANDBY hardware handshake
- No wake-up request
- One system direct memory access (sDMA) request
- One interrupt request to the Cortex-A9 MPU Interrupt Controller (MA\_IRQ) and to DSP subsystem (D\_IRQ)
- One clock for functional and interface domains

Figure 15-53 shows GPMC integration.

Figure 15-53. GPMC Integration

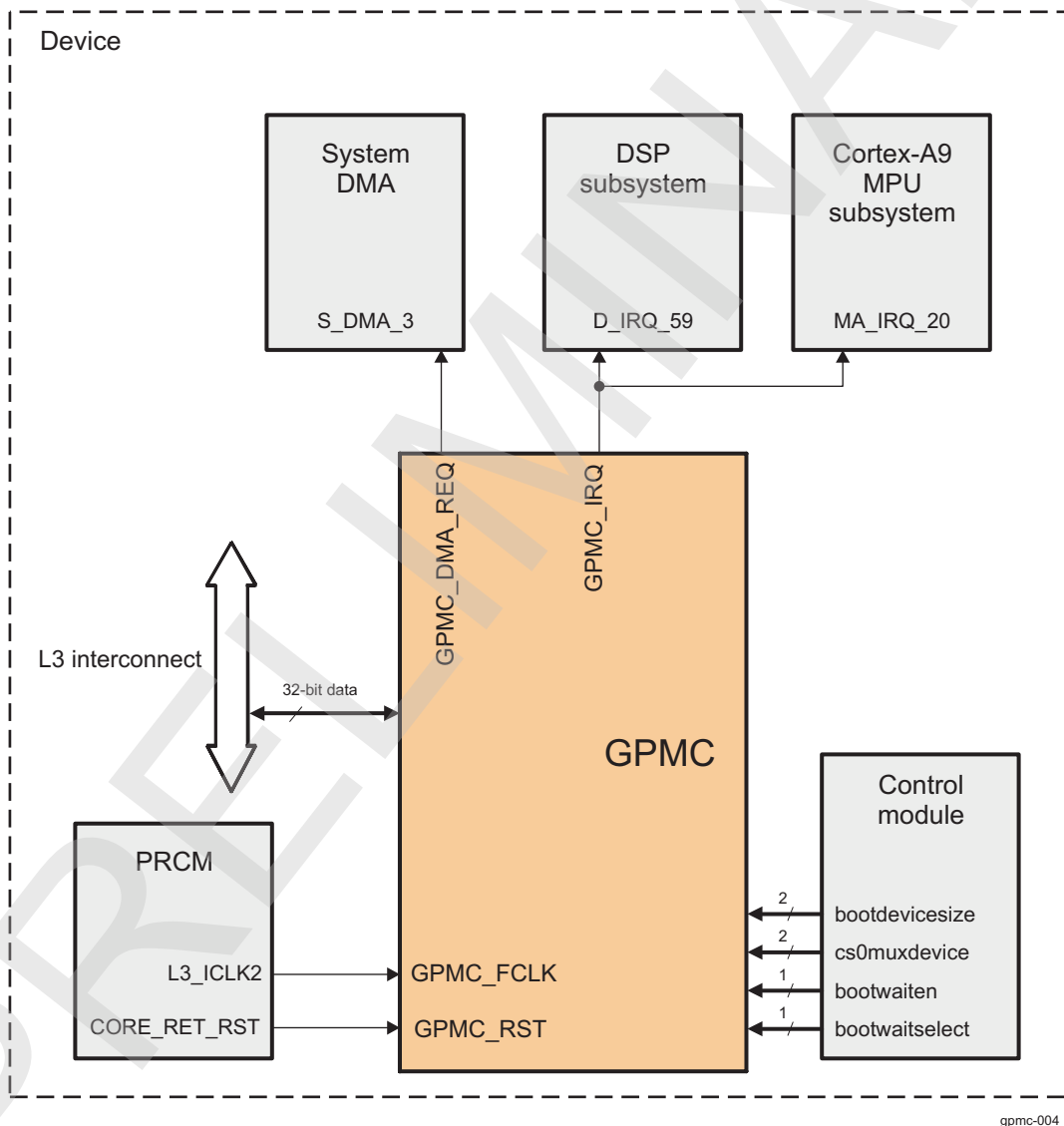


Table 15-214, Table 15-215 and Table 15-216 summarize the integration of the module in the device.

**Table 15-214. GPMC Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-up Capability	Interconnect
GPMC	PD_CORE	No	L3

**NOTE:**

- For information about operation performance point (OPP) configuration description, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

**Table 15-215. GPMC Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
GPMC	GPMC_FCLK	L3_ICLK2	PRCM	Functional clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
GPMC	GPMC_RST	CORE_RET_RST	PRCM	GPMC reset

**NOTE:** For clock description, see [Section 15.4.4.2, GPMC Clock Configuration](#).

**Table 15-216. GPMC Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
GPMC	GPMC_IRQ	D_IRQ_59	DSP	GPMC interrupt to DSP subsystem
	GPMC_IRQ	MA_IRQ_20	Cortex-A9	GPMC interrupt to Cortex-A9 MPU subsystem
DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
GPMC	GPMC_DMA_REQ	S_DMA_3	sDMA	GPMC request from Prefetch Engine to sDMA

**NOTE:**

- For interrupt source description, see [Section 15.4.4.5, GPMC Interrupt Requests](#).
- For DMA source description, see [Section 15.4.4.6, GPMC DMA Requests](#).

### 15.4.4 GPMC Functional Description

The GPMC basic programming model offers maximum flexibility to support various access protocols for each of the eight configurable chip-selects. Use optimal chip-select settings, based on the characteristics of the external device:

- Different protocols can be selected to support generic asynchronous or synchronous random-access devices (NOR flash, SRAM) or to support specific NAND devices.
- The address and the data bus can be multiplexed on the same external bus.
- Read and write access can be independently defined as asynchronous or synchronous.
- System requests (byte, 16-bit word, burst) are performed through single or multiple accesses. External access profiles (single, multiple with optimized burst length, native- or emulated-wrap) are based on external device characteristics (supported protocol, bus width, data buffer size, native-wrap support).
- System burst read or write requests are synchronous-burst (multiple-read or multiple-write). When neither burst nor page mode is supported by external memory or ASIC devices, system burst read or write requests are translated to successive single synchronous or asynchronous accesses (single reads or single writes). 8-bit wide devices are supported only in single synchronous or single asynchronous read or write mode.
- To simulate a programmable internal-wait state, an external wait pin can be monitored to dynamically control external access at the beginning (initial access time) of and during a burst access.

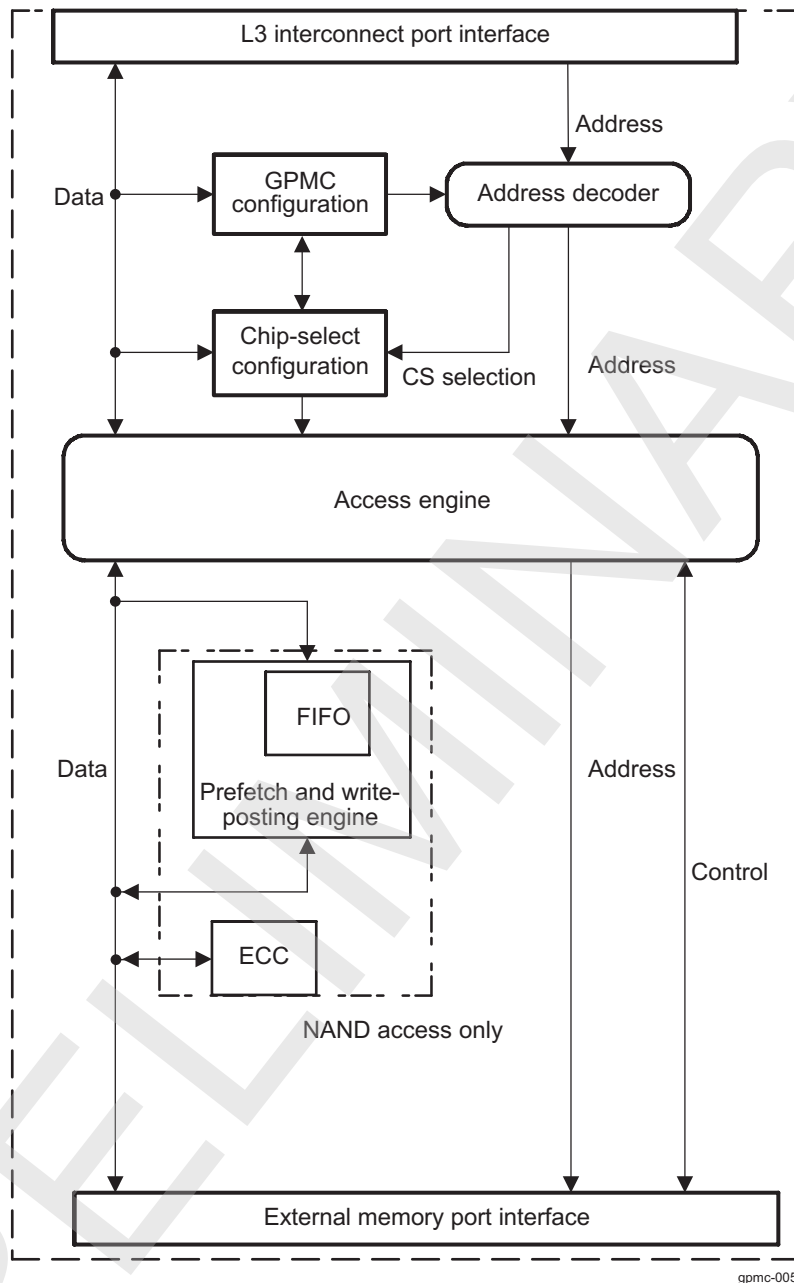
Each control signal is controlled independently for each chip-select. The internal functional clock of the GPMC (GPMC\_FCLK) is used as a time reference to specify the following:

- Read- and write-access duration
- Most GPMC external interface control-signal assertion and deassertion times
- Data-capture time during read access
- External wait-pin monitoring time
- Duration of idle time between accesses, when required

#### 15.4.4.1 GPMC Block Diagram

Figure 15-54 shows the GPMC functional block diagram. The GPMC consists of six blocks:

- L3 interconnect port interface
- Address decoder, GPMC configuration, and chip-select configuration register file
- Access engine
- Prefetch and write-posting engine
- Error correction code engine (ECC)
- External device/memory port interface

**Figure 15-54. GPMC Block Diagram**

The GPMC can access various external devices through the L3 Interconnect. The flexible programming model allows a wide range of attached device types and access schemes.

Based on the programmed configuration bit fields stored in the GPMC registers, the GPMC is able to generate all control signals timing depending on the attached device and access type.

Given the chip-select decoding and its associated configuration registers, the GPMC selects the appropriate device type control signals timing.

#### 15.4.4.2 GPMC Clock Configuration

Table 15-217 describes the GPMC clocks.

**Table 15-217. GPMC Clocks**

Signal	I/O <sup>(1)</sup>	Description
GPMC_FCLK	I	Functional and interface clock
gpmc_clk	O	External clock provided to synchronous external memory devices.

<sup>(1)</sup> I = Input; O = Output

The gpmc\_clk is generated by the GPMC from the internal GPMC\_FCLK clock. The source of the GPMC\_FCLK is described in [Table 15-215](#). The gpmc\_clk is configured via the [GPMC\\_CONFIG1\\_i\[1:0\]](#) GPMCFCLKDIVIDER field (for i = 0 to 3) as shown in [Table 15-218](#).

**Table 15-218. gpmc\_clk Configuration**

Source Clock	<a href="#">GPMC_CONFIG1_i[1:0]</a> GPMCFCLKDIVIDER	gpmc_clk Generated Clock Provided to External Memory Device
GPMC_FCLK	00	GPMC_FCLK
	01	GPMC_FCLK/2
	10	GPMC_FCLK/4
	11	GPMC_FCLK/8

#### 15.4.4.3 GPMC Software Reset

The GPMC can be reset by software through the [GPMC\\_SYSCONFIG\[1\]](#) SOFTRESET bit. Setting the bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Hardware and software resets initialize all GPMC registers and the finite state-machine (FSM) immediately and unconditionally. The [GPMC\\_SYSSTATUS\[0\]](#) RESETDONE bit indicates that the software reset is complete when its value is 1. The software must ensure that the software reset completes before doing GPMC operations.

#### 15.4.4.4 GPMC Power Management

GPMC power is supplied by the CORE power domain, and GPMC power management complies with system power-management guidelines.

[Table 15-219](#) describes power-management features available for the GPMC module.

**NOTE:**

- For information about source clock gating and sleep/wake-up transitions description, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).
- For descriptions of EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see [Section 3.6, Clock Management Functional Description](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**Table 15-219. GPMC Local Power Management Features**

Feature	Registers	Description
Clock Auto Gating	<a href="#">GPMC_SYSCONFIG[0]</a> AUTOIDLE bit	This bit allows a local power optimization inside the module, by gating the GPMC_FCLK clock upon the internal activity.
Slave Idle Modes	<a href="#">GPMC_SYSCONFIG[4:3]</a> SIDLEMODE bit field	Force-idle, No-idle and Smart-idle wakeup modes are available
Clock Activity	N/A	Feature not available
Master Standby Modes	N/A	Feature not available
Global Wake-up Enable	N/A	Feature not available
Wake-up Sources Enable	N/A	Feature not available

### 15.4.4.5 GPMC Interrupt Requests

The GPMC generates one interrupt event as shown in [Figure 15-53](#) :

- The interrupt request goes from GPMC (GPMC\_IRQ) to the Cortex-A9 MPU subsystem : MA\_IRQ\_20 and to the DSP subsystem : D\_IRQ\_59.

[Table 15-220](#) lists the event flags, and their mask, that can cause module interrupts.

**Table 15-220. GPMC Interrupt Events**

Event Flag	Event Mask	Sensitivity	Map to	Description
<a href="#">GPMC_IRQSTATUS</a> [10] WAIT2EDGE DETECTIONSTATUS	<a href="#">GPMC_IRQENABLE</a> [10] WAIT2EDGE DETECTIONENABLE	Edge	D_IRQ_59 and MA_IRQ_20	Wait2 edge detection interrupt: Triggered if a rising or falling edge is detected on the gpmc_wait2 signal. The rising or falling edge detection of Wait2 is selected through <a href="#">GPMC_CONFIG</a> [10] WAIT2PINPOLARITY
<a href="#">GPMC_IRQSTATUS</a> [9] WAIT1EDGE DETECTIONSTATUS	<a href="#">GPMC_IRQENABLE</a> [9] WAIT1EDGE DETECTIONENABLE	Edge	D_IRQ_59 and MA_IRQ_20	Wait1 edge detection interrupt: Triggered if a rising or falling edge is detected on the gpmc_wait1 signal. The rising or falling edge detection of Wait1 is selected through <a href="#">GPMC_CONFIG</a> [9] WAIT1PINPOLARITY
<a href="#">GPMC_IRQSTATUS</a> [8] WAIT0EDGE DETECTIONSTATUS	<a href="#">GPMC_IRQENABLE</a> [8] WAIT0EDGE DETECTIONENABLE	Edge	D_IRQ_59 and MA_IRQ_20	Wait0 edge detection interrupt: Triggered if a rising or falling edge is detected on the gpmc_wait0 signal. The rising or falling edge detection of Wait0 is selected through <a href="#">GPMC_CONFIG</a> [8] WAIT0PINPOLARITY
<a href="#">GPMC_IRQSTATUS</a> [1] TERMINAL COUNTSTATUS	<a href="#">GPMC_IRQENABLE</a> [1] TERMINAL COUNTENABLE	Level	D_IRQ_59 and MA_IRQ_20	Terminal count event: Triggered on prefetch process completion, that is when the number of currently remaining data to be requested reaches zero
<a href="#">GPMC_IRQSTATUS</a> [0] FIFOEVENTSTATUS	<a href="#">GPMC_IRQENABLE</a> [0] FIFOEVENTENABLE	Level	D_IRQ_59 and MA_IRQ_20	FIFO event interrupt: Indicates FIFO levels availability for in Write-Posting mode and prefetch mode. <a href="#">GPMC_PREFETCH_CONFIG1</a> [2] DMAMODE shall be set to 0.

### 15.4.4.6 GPMC DMA Requests

The GPMC generates one DMA event as shown in [Figure 15-53](#) :

- From GPMC (GPMC\_DMA\_REQ) to the system DMA (sDMA) : S\_DMA\_3.

### 15.4.4.7 L3 Interconnect Interface

The GPMC L3 interconnect interface is a pipelined interface including an 16 x 32-bit word write buffer.

Any system host can issue external access requests through the GPMC.

The device system can issue the following requests through this interface:

- One 8-bit / 16-bit / 32-bit interconnect access (read/write)
- Two incrementing 32-bit interconnect accesses (read/write)
- Two wrapped 32-bit interconnect accesses (read/write)
- Four incrementing 32-bit interconnect accesses (read/write)
- Four wrapped 32-bit interconnect accesses (read/write)
- Eight incrementing 32-bit interconnect accesses (read/write)
- Eight wrapped 32-bit interconnect accesses (read/write)

Only linear burst transactions are supported; interleaved burst transactions are not supported. Only power-of-two-length precise bursts 2 x 32, 4 x 32, 8 x 32 or 16 x 32 with the burst base address aligned on the total burst size are supported (this limitation applies to incrementing bursts only).

This interface also provides one interrupt and one DMA request line, for specific event control.

It is recommended to program the ATTACHEDDEVICEPAGELENGTH field ([GPMC\\_CONFIG1\\_i\[24:23\]](#)) according to the effective attached device page length and to enable WRAPBURST bit ([GPMC\\_CONFIG1\\_i\[31\]](#)) if the attached device supports wrapping burst.

However, it is possible to emulate wrapping burst on a non-wrapping memory by providing relevant addresses within the page or splitting transactions. Bursts larger than the memory page length are chopped into multiple bursts transactions. Due to the alignment requirements, a page boundary is never crossed.

#### 15.4.4.8 GPMC Address and Data Bus

The current application supports GPMC connection to NAND devices and to address/data-multiplexed memories or devices. Connection to address/data-nonmultiplexed memories or devices is supported with a limited address range of 2 Kbytes.

Depending on the GPMC configuration of each chip-select, address and data bus lines that are not required for a particular access protocol are not updated (changed from current value) and are not sampled when input (input data bus).

- When the [GPMC\\_CONFIG\[1\]](#) LIMITEDADDRESS bit is set to 1, A26-A11 are not modified during an external memory access. This limits the memory address space to 2K-byte regardless of the memory size.
- For address/data-multiplexed and AAD-multiplexed NOR devices, the address is multiplexed on the data bus.
- 8-bit wide NOR devices do not use GPMC I/O: gpmc\_ad[15:8] for data (they are used for address if needed).
- 16-bit wide NAND devices do not use GPMC I/O: gpmc\_a[25:16].
- 8-bit wide NAND devices do not use GPMC I/O: gpmc\_a[25:16] and GPMC I/O: gpmc\_ad[15:8].

#### CAUTION

Before accessing a chip-select configured with a nonmultiplexed protocol, set the LIMITEDADDRESS bit control.

#### 15.4.4.8.1 GPMC I/O Configuration Setting

---

**NOTE:** In this section and next sections, the *i* in GPMC\_CONFIGx\_i stands for the GPMC chip-select *i* where *i* = 0 to 7.

---

To select a NAND device, program the following register fields:

- [GPMC\\_CONFIG1\\_i\[11:10\]](#) DEVICETYPE field = 0b10
- [GPMC\\_CONFIG1\\_i\[9:8\]](#) MUXADDDATA bit = 0b00

To select an address/data-multiplexed device, program the following register fields:

- [GPMC\\_CONFIG1\\_i\[11:10\]](#) DEVICETYPE field = 0b00
- [GPMC\\_CONFIG1\\_i\[9:8\]](#) MUXADDDATA bit = 0b10

To select an address/address/data-multiplexed device, program the following register fields:

- [GPMC\\_CONFIG1\\_i\[11:10\]](#) DEVICETYPE field = 0b00
- [GPMC\\_CONFIG1\\_i\[9:8\]](#) MUXADDDATA bit = 0b01

To select an address/data-nonmultiplexed device (limited to a 2K-byte address range), program the following register fields:

- [GPMC\\_CONFIG1\\_i\[11:10\]](#) DEVICETYPE field = 0b00
- [GPMC\\_CONFIG1\\_i\[9:8\]](#) MUXADDDATA bit = 0b00
- [GPMC\\_CONFIG\[1\]](#) LIMITEDADDRESS bit = 1

**NOTE:** The `GPMC_CONFIG1[1] LIMITEDADDRESS` bit only applies to address/data-nonmultiplexed devices; it has no effect on other device types (NAND, address/data-multiplexed).

#### 15.4.4.8.2 GPMC CS0 Default Configuration at IC Reset

To ensure a correct external boot with a GPMC access from IC reset time on CS0, several external pins are sampled:

- The `sys_boot[4:0]` pins (device boundary) define the sequence of interfaces and devices to use for booting.
- The `sys_boot[5]` pin defines which group of booting sequences is preferred: memory booting (`sys_boot[5] = 0`) or peripheral booting (`sys_boot[5] = 1`).
- Additional pins are used to configure reset values in the `GPMC_CONFIG1_i` register (where  $i = 0$ ) as detailed below and in [Table 15-221](#):
  - The `bootdevicesize` input pin (at the GPMC boundary) defines the size of the attached device on chip-select 0 and is used to configure the `GPMC_CONFIG1_i[13:12] DEVICESIZE` bits (where  $i = 0$ ). The `BOOT_DEVICE_TYPE` signal is propagated from the system control module (SCM). Its value is fixed for the device at 0x1 at IC reset, which means that only 16-bit memories are supported for booting.
  - The `cs0muxdevice` input pin (at the GPMC boundary) selects whether the attached device to chip-select 0 is an address/data-multiplexed device or not. The input pin is used to configure the `GPMC_CONFIG1_i[9:8] MUXADDDATA` bit (where  $i = 0$ ). The `CS0_MUX_DEVICE` signal is propagated from the SCM. Its value is fixed for the device at 0x2 at IC reset, which means that only address/data-multiplexed memories are supported for booting (that is, in a standalone way from memories connected to CS0).
  - The `bootwaiten` input pin (at the GPMC boundary) enables the monitoring on chip-select 0 of the `WAIT` pin at IC reset release time for read accesses. The input pin is used to configure the `GPMC_CONFIG1_i[22] WAITREADMONITORING` bit (where  $i = 0$ ). The `BOOT_WAIT_ENABLE` signal is propagated from the system control module (SCM). When `sys_boot[5:0] = 0b11 1111`, the `BOOT_WAIT_ENABLE` signal is activated, causing the wait pin to be monitored for read accesses.
  - The `bootwaitselect` input pin selects the `WAIT` signal at IC reset release time between `WAIT0` input pin, `WAIT1` input pin or `WAIT2` input pin. At IC reset release time, these three pins have different polarity.

**Table 15-221. Boot Control Interface Input Signals Description**

Signal Name	Width	Description
<code>bootdevicesize</code>	2	Size of the device attached on CS0 0b00: 8-bit 0b01: 16-bit 0b10: Reserved 0b11: Reserved
<code>cs0muxdevice</code>	2	Multiplexing mode of the device on CS0 0b00: Nonmultiplexed device on CS0 0b01: AAD-multiplexed device on CS0 (address-address-data) 0b10: Address/Data-multiplexed device on CS0 0b11: Reserved
<code>bootwaiten</code>	1	Wait monitoring on CS0 at IC reset release time for read accesses 0: Wait pin is not monitored 1: Wait pin is monitored
<code>bootwaitselect</code>	1	Selection of the wait pin 0: WAIT0 input pin 1: WAIT1 input pin 2: WAIT2 input pin



### CAUTION

Using the internal boot code, the entire CS0 configuration can be modified before the first CS0 access. This modification of internal boot code is necessary for two external devices:

- NAND device attached to CS0
- Nonmultiplexed 2 Kbytes address range device attached to CS0

At reset time, the IC may boot from the internal ROM.

Reset values of the timing control parameters are defined to cope with direct boot on address and data multiplexed NOR Flash device, on non-multiplexed NOR Flash device or on any asynchronous device with large timing margins assuming a low GPMC\_FCLK frequency (for example, 19.2Mhz) at boot time.

For a multiplexed access, the address 16 low-order bits are presented onto gpmc\_ad[15:0] while the high-order bits are presented both onto gpmc\_a[26:16]. If the external chip interface to the memories uses a 16-bit data bus, the high-order address bits are sampled on the address bus.

The reset values of timing parameters used at boot time are:

- CSONTIME = 1
- CSRDOFFTIME = 16
- ADVONTIME = 4
- ADVRDOFFTIME = 5
- OEONTIME = 6
- OEOFFTIME = 16
- RDACCESSTIME = 15
- RDCYCLETIME = 17

For an AAD-multiplexed access, all address bits are passed onto the data bus using two nADV rising edge. The first rising edge latches the address MSB down to bit 17 while the second rising edge latches address bits 16 down to 1. This configuration is only used for 16-bit memories.

The reset values of these timing parameters used at boot time are:

- ADVAADMUXONTIME = 1
- ADVAADMUXRDOFFTIME = 2
- OEAADMUXONTIME = 1
- OEAADMUXRDOFFTIME = 3

#### 15.4.4.9 Address Decoder and Chip-Select Configuration

Addresses are decoded accordingly with the address request of the chip-select and the content of the chip-select base address register file, which includes a set of global GPMC configuration registers and eight sets of chip-select configuration registers.

The GPMC configuration register file is memory-mapped and can be read or written with byte, 16-bit word, or 32-bit word accesses. The register file should be configured as a noncacheable, nonbufferable region to prevent any desynchronization between host execution (write request) and the completion of register configuration (write completed with register updated). [Section 15.4.7, GPMC Register Manual](#) provides the GPMC register locations. For the map of GPMC memory locations, see [Chapter 2, Memory Mapping](#).

After the chip-select is configured, the access engine accesses the external device, drives the external interface control signals, and applies the interface protocol based on user-defined timing parameters and settings.

##### 15.4.4.9.1 Chip-Select Base Address and Region Size

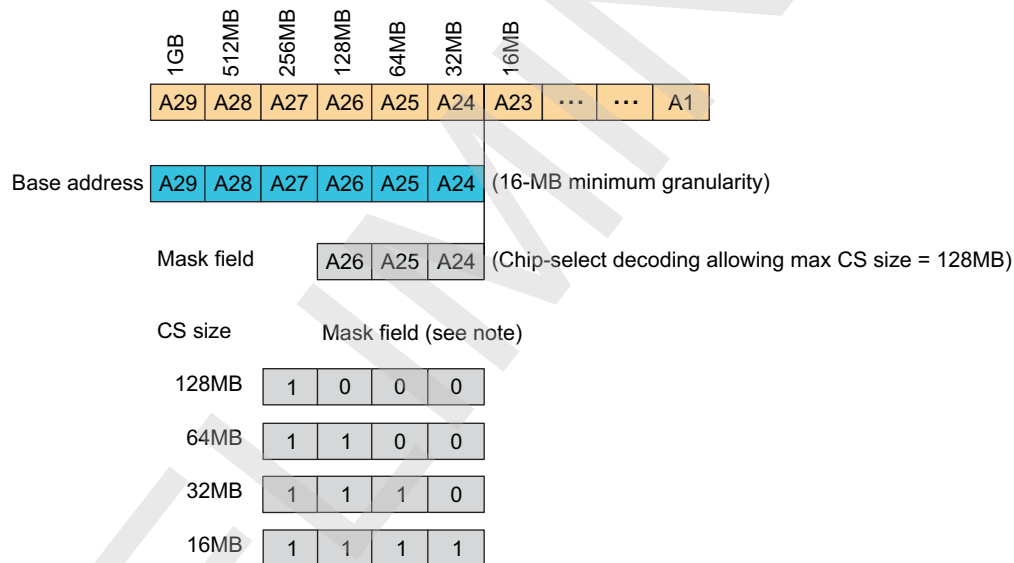
Any external memory or ASIC device attached to the GPMC external interface can be accessed by any device system host within the GPMC 1 Gbyte contiguous address space. For details, see [Chapter 2, Memory Mapping](#).

The GPMC 1G byte address space can be divided into a maximum of eight chip-select regions with programmable base address and programmable CS size. The CS size is programmable from 16M bytes to 128M bytes (must be a power-of-2) and is defined by the mask field. Attached memory smaller than the programmed CS region size is accessed through the entire CS region (aliasing).

Each chip-select has a 6-bit base address encoding and a 4-bit decoding mask, which must be programmed according to the following rules:

- The programmed chip-select region base address must be aligned on the chip-select region size address boundary and is limited to a power-of-2 address value. During access decoding, the register base address value is used for address comparison with the address-bit line mapping as described in [Figure 15-55](#) (with A0 as the device system byte-address line). Base address is programmed through the `GPMC_CONFIG7_i[5:0]` BASEADDRESS bit field
- The register mask is used to exclude some address lines from the decoding. A register mask bit field set to 0 suppresses the associated address line from the address comparison (incoming address bit line is don't care). The register mask value must be limited to the subsequent value, based on the desired chip-select region size. Any other value has an undefined result. When multiple chip-select regions with overlapping addresses are enabled concurrently, access to these chip-select regions is cancelled and a GPMC access error is posted. The mask field is programmed through the `GPMC_CONFIG7_i[11:8]` MASKADDRESS bit field.

**Figure 15-55. Chip-Select Address Mapping and Decoding Mask**



gpmc-006

Chip-select configuration (base and mask address or any protocol and timing settings) must be performed while the associated chip-select is disabled through the `GPMC_CONFIG7_i[6]` CSVALID bit (where *i* stands for the GPMC chip-select value, *i* = 0 to 3). In addition, a chip-select configuration can only be disabled if there is no ongoing access to that chip-select. This requires activity monitoring of the prefetch or write-posting engine if the engine is active on the chip-select. Also, the write buffer state must be monitored to wait for any posted write completion to the chip-select.

Any access attempted to a nonvalid GPMC address region (CSVALID disabled or address decoding outside a valid chip-select region) is not propagated to the external interface and a GPMC access error is posted. In case of chip-selects overlapping, an error is generated and no access will occur on either chip-select.

Chip-select 0 is the only chip-select region enabled after either a power-up or a GPMC reset.

### CAUTION

Although the GPMC interface can drive up to 4 chip-selects, the frequency specified for this interface is for a specific load. If this load is exceeded, the maximum frequency cannot be reached. One solution is to implement a board with buffers, to allow the slowest device to maintain the total load on the lines.

#### 15.4.4.9.2 Access Protocol

##### 15.4.4.9.2.1 Supported Devices

The access protocol of each chip-select can be independently specified through the [GPMC\\_CONFIG1\\_i\[11:10\]](#) DEVICETYPE parameter (where  $i = 0$  to 3) for:

- Random-access synchronous or asynchronous memory like NOR flash, SRAM
- NAND flash asynchronous devices

---

**NOTE:** For more information about the NAND flash GPMC basic programming model and NAND support, see [Section 15.4.4.13, NAND Device Basic Programming Model](#), and [Section 15.4.4.13.1, NAND Memory Device in Byte or Word 16 Stream Mode](#).

---

##### 15.4.4.9.2.2 Access Size Adaptation and Device Width

Each chip-select can be independently configured through the [GPMC\\_CONFIG1\\_i\[13:12\]](#) DEVICESIZE field ( $i = 0$  to 3) to interface with a 16-bit wide device or an 8-bit wide device. System requests with data width greater than the external device data bus width are split into successive accesses according to both the external device data-bus width and little-endian data organization.

---

**NOTE:** The device does not provide the A0 byte address line required for random-byte addressable 8-bit wide device interfacing (for both multiplexed and nonmultiplexed protocol). It limits the use of 8-bit wide device interfacing to byte-alias accesses. This limitation is not applicable to NAND device interfacing (8-bit wide or 16-bit wide devices).

---

##### 15.4.4.9.2.3 Address/Data-Multiplexing Interface

For random synchronous or asynchronous memory interfacing ( $\text{DEVICETYPE} = 0b00$ ), an address- and data-multiplexing protocol can be selected through the [GPMC\\_CONFIG1\\_i\[9:8\]](#) MUXADDDATA bit field ( $i = 0$  to 3). The  $\text{nADV}$  signal must be used as the external device address latch control signal. For the associated chip-select configuration,  $\text{nADV}$  assertion and deassertion time and  $\text{nOE}$  assertion time must be set to the appropriate value to meet the address latch setup/hold time requirements of the external device. See [Section 15.4.3, GPMC Integration](#).

---

**NOTE:** This address/data-multiplexing interface is not applicable to NAND device interfacing. NAND devices require a specific address, command, and data multiplexing protocol. See [Section 15.4.4.13, NAND Device Basic Programming Model](#).

---

#### 15.4.4.9.3 External Signals

##### 15.4.4.9.3.1 WAIT Pin Monitoring Control

GPMC access time can be dynamically controlled using an external `gpmc_wait` pin when the external device access time is not deterministic and cannot be defined and controlled only using the GPMC internal `RDACCESSTIME`, `WRACCESSTIME` and `PAGEBURSTACCESSTIME` wait state generator.

The GPMC features three input wait pins: gpmc\_wait2, gpmc\_wait1 and gpmc\_wait0. These pins allow control of external devices with different wait-pin polarity. They also allow the overlap of wait-pin assertion from different devices without affecting access to devices for which the wait pin is not asserted.

- The [GPMC\\_CONFIG1\\_i\[17:16\]](#) WAITPINSELECT field (i = 0 to 3) selects which input gpmc\_wait pin is used for the device attached to the corresponding chip-select.
- The polarity of the wait pin is defined through the WAITxPINPOLARITY bit of the [GPMC\\_CONFIG](#) register. A wait pin configured to be active low means that low level on the WAIT signal indicates that the data is not ready and that the data bus is invalid. When WAIT is inactive, data is valid.

The GPMC access engine can be configured per CS to monitor the wait pin of the external memory device or not, based on the access type: read or write.

- The [GPMC\\_CONFIG1\\_i\[22\]](#) WAITREADMONITORING bit defines whether the wait pin should be monitored during read accesses or not.
- The [GPMC\\_CONFIG1\\_i\[21\]](#) WAITWRITEMONITORING bit defines whether the wait pin should be monitored during write accesses or not.

The GPMC access engine can be configured to monitor the wait pin of the external memory device asynchronously or synchronously with the GPMC\_CLK clock, depending on the access type: synchronous or asynchronous (the [GPMC\\_CONFIG1\\_i\[29\]](#) READTYPE and [GPMC\\_CONFIG1\\_i\[27\]](#) WRITETYPE bits).

#### 15.4.4.9.3.1.1 Wait Monitoring During an Asynchronous Read Access

When wait-pin monitoring is enabled for read accesses (WAITREADMONITORING), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the wait-deasserted state.

During asynchronous read accesses with wait-pin monitoring enabled, the wait pin must be at a valid level (asserted or deasserted) for at least two GPMC clock cycles before RDACCESSTIME completes, to ensure correct dynamic access-time control through wait-pin monitoring. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

In this context, RDACCESSTIME is used as a WAIT invalid timing window and is set to such a value that the wait pin is at a valid state two GPMC clock cycles before RDACCESSTIME completes.

Similarly, during a multiple-access cycle (for example, asynchronous read page mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-deasserted state. Wait-monitoring pipelining is also applicable to multiple accesses (access within a page).

- WAIT monitored as active freezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as asserted extends the current access time in the page. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as inactive completes the current access time and starts the next access phase in the page. The data bus is considered valid, and data are captured during this clock cycle. In case of a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their related control timing value and according to the CYCLETIME counter status.

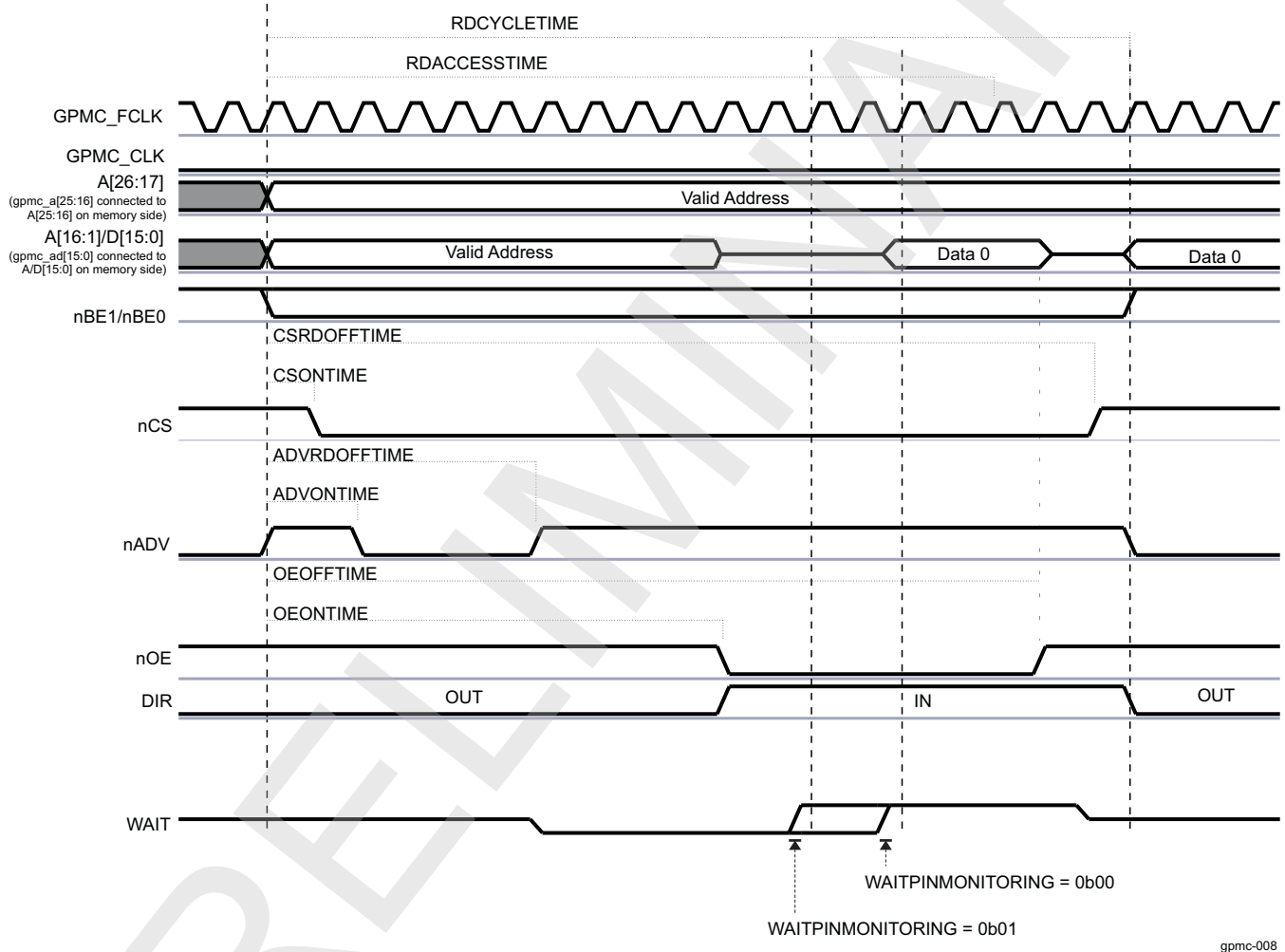
When a delay larger than two GPMC clocks must be observed between wait-pin deactivation time and data valid time (including the required GPMC and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data-capture time and the effective unlock of the CYCLETIME counter. This extra delay can be programmed in the [GPMC\\_CONFIG1\\_i\[19:18\]](#) WAITMONITORINGTIME field (i = 0 to 3).

**NOTE:**

- The WAITMONITORINGTIME parameter does not delay the wait-pin active or inactive detection, nor does it modify the two GPMC clocks pipelined detection delay.
- This extra delay is expressed as a number of GPMC\_CLK clock cycles, even though the access is defined as asynchronous, and no GPMC\_CLK clock is provided to the external device. Still, GPMCFCLKDIVIDER is used as a divider for the GPMC clock, so it must be programmed to define the correct WAITMONITORINGTIME delay.

Figure 15-56 shows wait behavior during an asynchronous single read access.

**Figure 15-56. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1)**



**NOTE:** The WAIT signal is active low. `GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME = 0b00`, or `0b01`.

**15.4.4.9.3.1.2 Wait Monitoring During an Asynchronous Write Access**

When wait-pin monitoring is enabled for write accesses (`GPMC_CONFIG1_i[21] WAITWRITEMONITORING` bit = `0x1`), the WAIT-invalid timing window is defined by the WRACCESSTIME field. WRACCESSTIME must be set so that the wait pin is at a valid state two GPMC clock cycles before WRACCESSTIME completes. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

- WAIT monitored as active freezes the CYCLETIME counter. This informs the GPMC that the data bus is not captured by the external device. The control signals are kept in their current state. The data bus still drives the data.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. This informs that the data bus is correctly captured by the external device. All signals, including the data bus, are controlled according to their related control timing value and to the CYCLETIME counter status.

When a delay larger than two GPMC clock cycles must be observed between wait-pin deassertion time and the effective data write into the external device (including the required GPMC data setup time and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data write time into the external device and the effective unfreezing of the CYCLETIME counter. This extra delay can be programmed in the `GPMC_CONFIG1_i[19:18]` WAITMONITORINGTIME fields ( $i = 0$  to 3).

**NOTE:**

- The WAITMONITORINGTIME parameter does not delay the wait-pin assertion or deassertion detection, nor does it modify the two GPMC clock cycles pipelined detection delay.
- This extra delay is expressed as a number of GPMC\_CLK clock cycles, even though the access is defined as asynchronous, and even though no clock is provided to the external device. Still, `GPMC_CONFIG1_i[1:0]` GPMCFCLKDIVIDER is used as a divider for the GPMC clock and so it must be programmed to define the correct WAITMONITORINGTIME delay.

#### 15.4.4.9.3.1.3 Wait Monitoring During a Synchronous Read Access

During synchronous accesses with wait-pin monitoring enabled, the wait pin is captured synchronously with GPMC\_CLK, using the rising edge of this clock.

The WAIT signal can be programmed to apply to the same clock cycle it is captured in. Alternatively, it can be sampled one or two GPMC\_CLK cycles ahead of the clock cycle it applies to. This pipelining is applicable to the entire burst access, and to all data phase in the burst access. This WAIT pipelining depth is programmed in the `GPMC_CONFIG1_i[19:18]` WAITMONITORINGTIME field (where  $i = 0$  to 3), and is expressed as a number of GPMC\_CLK clock cycles.

In synchronous mode, when wait-pin monitoring is enabled (`GPMC_CONFIG1_i[22]` WAITREADMONITORING bit), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the WAIT deasserted-state detection.

Depending on the programmed WAITMONITORINGTIME value, the wait pin should be at a valid level, either asserted or deasserted:

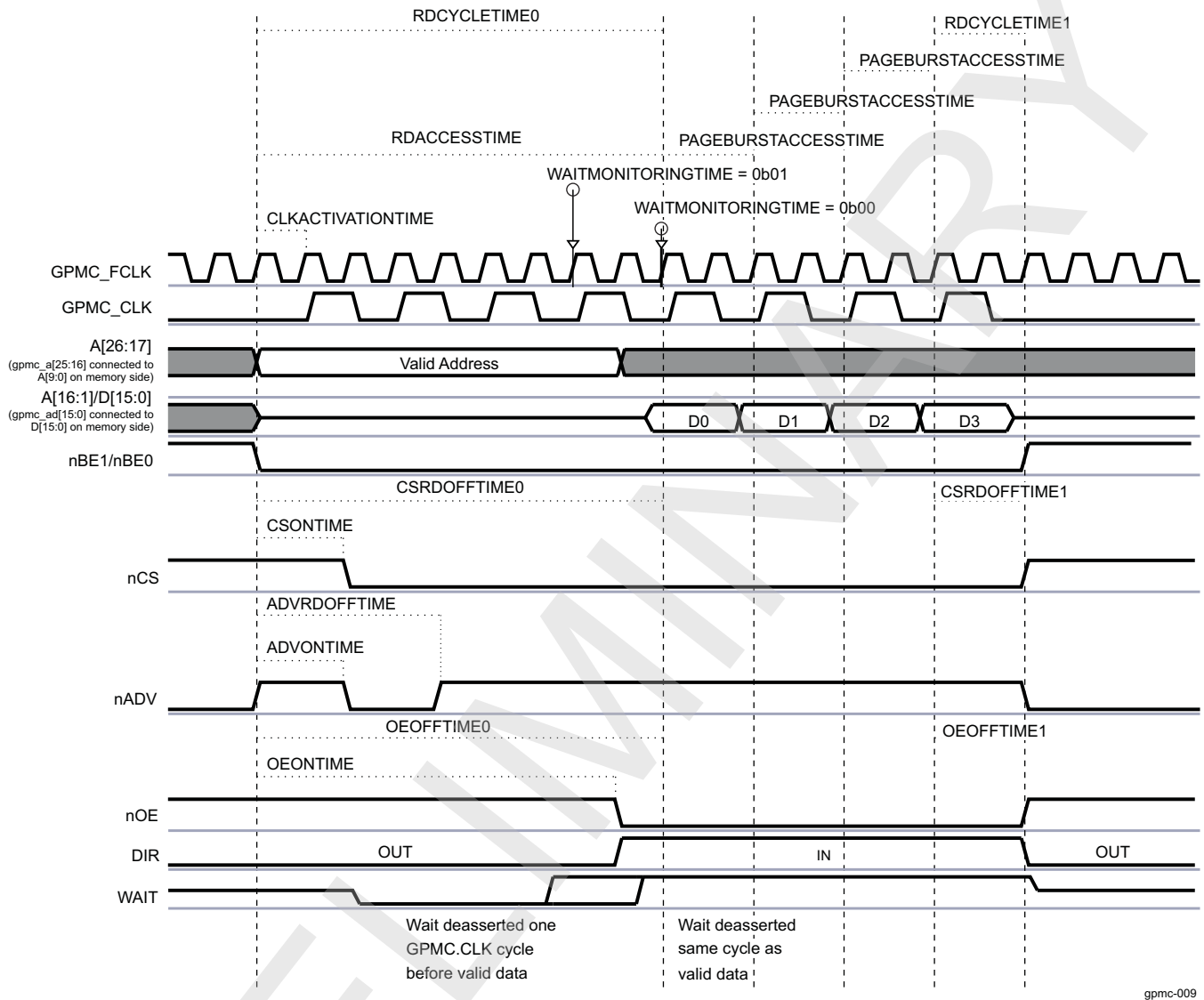
- In the same clock cycle the data is valid if WAITMONITORINGTIME = 0 ( at RDACCESSTIME completion)
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC\_FCLK clock cycles before RDACCESSTIME completion if WAITMONITORINGTIME not equal to 0

Similarly, during a multiple-access cycle (burst mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-inactive state. The Wait pipelining depth programming applies to the whole burst access.

- WAIT monitored as active freezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in a lock state), WAIT monitored as active extends the current access time in the burst. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in lock state), WAIT monitored as inactive completes the current access time and starts the next access phase in the burst. The data bus is considered valid, and data are captured during this clock cycle. In a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their relative control timing value and the CYCLETIME counter status.

Figure 15-57 shows wait behavior during a synchronous read burst access.

Figure 15-57. Wait Behavior During a Synchronous Read Burst Access



gpmc-009

**NOTE:** The WAIT signal is active low.  $WAITMONITORINGTIME = 00, 01$ .

#### 15.4.4.9.3.1.4 Wait Monitoring During a Synchronous Write Access

During synchronous accesses with wait-pin monitoring enabled (the  $WAITWRITEMONITORING$  bit), the wait pin is captured synchronously with  $GPMC\_CLK$ , using the rising edge of this clock.

If enabled, external wait-pin monitoring can be used in combination with  $WRACCESSTIME$  to delay the effective memory device  $GPMC\_CLK$  capture edge.

Wait-monitoring pipelining depth is similar to synchronous read access:

- At  $WRACCESSTIME$  completion if  $WAITMONITORINGTIME = 0$
- In the  $WAITMONITORINGTIME \times (GPMCFCLKDIVIDER + 1)$   $GPMC\_FCLK$  cycles before  $WRACCESSTIME$  completion if  $WAITMONITORINGTIME$  not equal to 0.

Wait-monitoring pipelining definition applies to whole burst accesses:

- WAIT monitored as active freezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as active indicates that the data bus is not being captured by the external device. Control signals are kept in their current state. The data bus is kept in its current state.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as inactive indicates the effective data capture of the bus by the external device and starts the next access of the burst. In case of a single access or if this was the last access in a multiple access cycle, all signals, including the data bus, are controlled according to their related control timing value and the CYCLETIME counter status.

---

**NOTE:** Wait monitoring is supported for all configurations except for [GPMC\\_CONFIG1\\_i\[19:18\]](#) WAITMONITORINGTIME = 0x0 (where i = 0 to 3) for write bursts with a clock divider of 1 or 2 ([GPMC\\_CONFIG1\\_i\[1:0\]](#) GPMCFCLKDIVIDER field equal to 0x0 or 0x1 respectively).

---

#### 15.4.4.9.3.1.5 WAIT with NAND Device

For details about the use of the wait pin for communication with a NAND flash external device, see [Section 15.4.4.13.2, NAND Device-Ready Pin](#).

#### 15.4.4.9.3.1.6 Idle Cycle Control Between Successive Accesses

##### 15.4.4.9.3.1.6.1 Bus Turnaround (BUSTURNAROUND)

To prevent data-bus contention, an access that follows a read access to a slow memory/device must be delayed (in other words, control the nCS/nOE de-assertion to data bus in high-impedance delay).

The bus turnaround is a time-out counter starting after nCS or nOE de-assertion time, whichever occurs first, and delays the next access start-cycle time. The counter is programmed through the [GPMC\\_CONFIG6\\_i\[3:0\]](#) BUSTURNAROUND bit field (where i = 0 to 3).

After a read access to a chip-select with a non zero BUSTURNAROUND, the next access is delayed until the BUSTURNAROUND delay completes, if the next access is one of the following:

- A write access to any chip-select (same or different from the chip-select data was read from)
- A read access to a different chip-select from the chip-select data was read access from
- A read or write access to a chip-select associated with an address/data-multiplexed device

---

**NOTE:** Bus keeping starts after bus turnaround completion so that DIR changes from IN to OUT after bus turnaround. The bus will not have enough time to go into high-impedance even though it could be driven with the same value before bus turnaround timing.

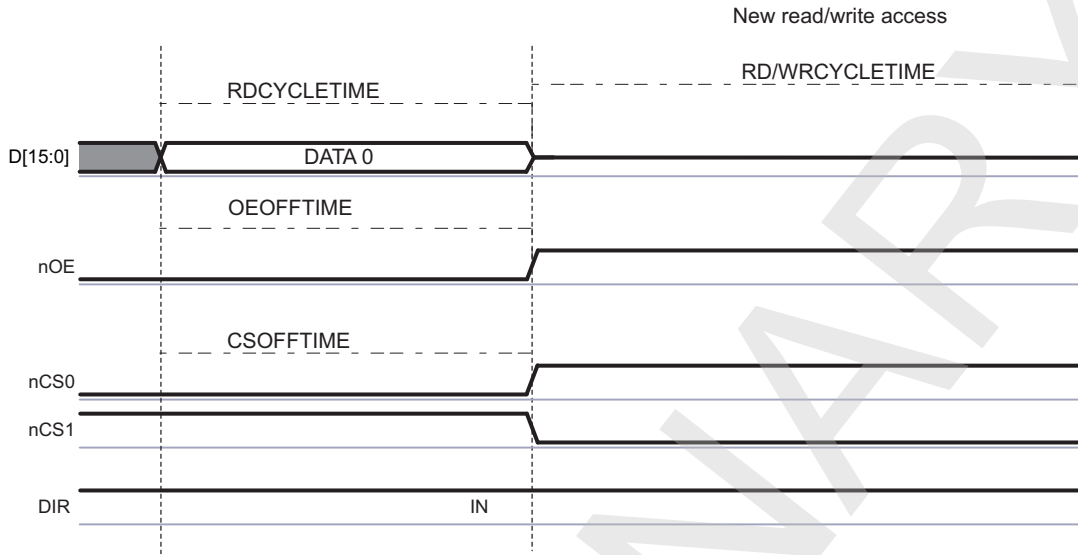
---

BUSTURNAROUND delay runs in parallel with [GPMC\\_CONFIG6\\_i\[3:0\]](#) CYCLE2CYCLEDELAY delays. It should be noted that BUSTURNAROUND is a timing parameter for the ending chip-select access while CYCLE2CYCLEDELAY is a timing parameter for the following chip-select access. The effective minimum delay between successive accesses is driven by these delay timing parameters and by the access type of the following access. See [Figure 15-58](#) to [Figure 15-60](#).

Another way to prevent bus contention is to define an earlier nCS or nOE deassertion time for slow devices or to extend the value of RDCYCLETIME. Doing this prevents bus contention, but affects all accesses of this specific chip-select.

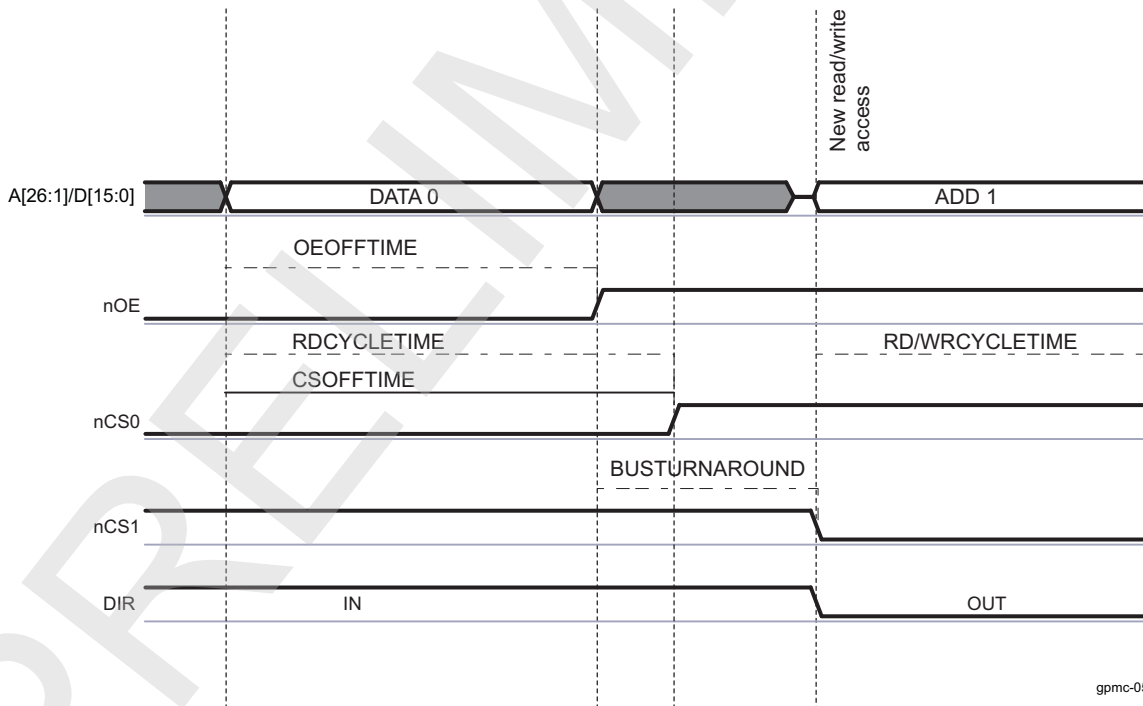


**Figure 15-58. Read to Read for an Address-Data Multiplexed Device, on Different CS, without Bus Turnaround (nCS0 attached to a fast device)**



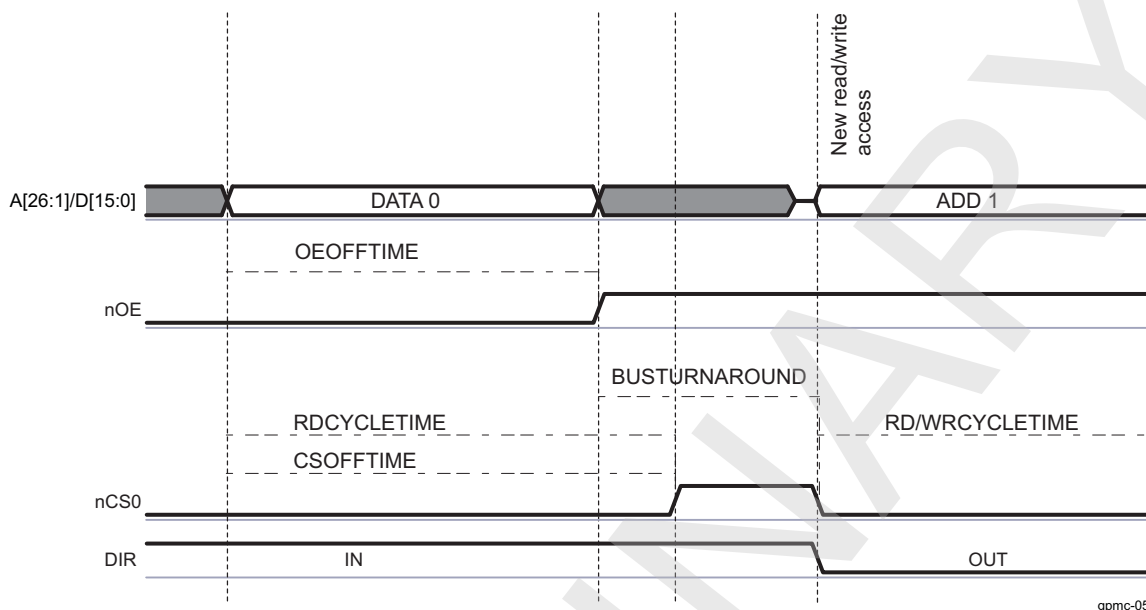
gpmc-049

**Figure 15-59. Read to Read / Write for an Address-Data Multiplexed Device, on Different CS, with Bus Turnaround**



gpmc-050

**Figure 15-60. Read to Read / Write for a Address-Data or AAD-multiplexed device, on the Same CS, with Bus Turnaround**



#### 15.4.4.9.3.1.6.2 Idle Cycles Between Accesses to Same Chip-Select (CYCLE2CYCLESAMECSEN, CYCLE2CYCLEDELAY)

Some devices require a minimum chip-select signal inactive time between accesses. The [GPMC\\_CONFIG6\\_i\[7\]](#) CYCLE2CYCLESAMECSEN bit ( $i = 0$  to 3) enables insertion of a minimum number of GPMC\_FCLK cycles, defined by the [GPMC\\_CONFIG6\\_i\[11:8\]](#) CYCLE2CYCLEDELAY field, between successive accesses of any type (read or write) to the same chip-select.

If CYCLE2CYCLESAMECSEN is enabled, any subsequent access to the same chip-select is delayed until its CYCLE2CYCLEDELAY completes. The CYCLE2CYCLEDELAY counter starts when CSRDOFFTIME/CSWROFFTIME completes.

The same applies to successive accesses occurring during 32-bit word or burst accesses split into successive single accesses when the single-access mode is used ([GPMC\\_CONFIG1\\_i\[30\]](#) READMULTIPLE = 0 or [GPMC\\_CONFIG1\\_i\[28\]](#) WRITEMULTIPLE = 0).

All control signals are kept in their default states during these idle GPMC\_FCLK cycles. This prevents back-to-back accesses to the same chip-select without idle cycles between accesses.

#### 15.4.4.9.3.1.6.3 Idle Cycles Between Accesses to Different Chip-Select (CYCLE2CYCLEDIFFCSEN, CYCLE2CYCLEDELAY)

Because of the pipelined behavior of the system, successive accesses to different chip-selects can occur back-to-back with no idle cycles between accesses. Depending on the control signals ( $nCS$ ,  $nADV/ALE$ ,  $nBE0/CLE$ ,  $nOE/RE$ ,  $nWE$ ) assertion and de-assertion timing parameters and on the IC timing parameters, some control signals assertion times may overlap between the successive accesses to different CS. Similarly, some control signals ( $WE$ ,  $OE/RE$ ) may not respect required transition times.

To work around the overlapping and to observe the required control-signal transitions, a minimum of CYCLE2CYCLEDELAY inactive cycles is inserted between the access being initiated to this chip-select and the previous access ending for a different chip-select. This applies to any type of access (read or write).

If [GPMC\\_CONFIG6\\_i\[6\]](#) CYCLE2CYCLEDIFFCSEN is enabled, the chip-select access is delayed until CYCLE2CYCLEDELAY cycles have expired since the end of a previous access to a different chip-select. CYCLE2CYCLEDELAY count starts at CSRDOFFTIME/CSWROFFTIME completion. All control signals are kept inactive during the idle GPMC\_FCLK cycles.

**NOTE:** CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDIFFCSEN should be set in GPMC\_CONFIG6<sub>i</sub> registers to respectively get idle cycles inserted between accesses on this chip-select and after accesses to a different chip-select.

The CYCLE2CYCLEDELAY delay runs in parallel with the BUSTURNAROUND delay. It should be noted that BUSTURNAROUND is a timing parameter defined for the ending chip-select access, whereas CYCLE2CYCLEDELAY is a timing parameter defined for the starting chip-select access. The effective minimum delay between successive accesses is based on the larger delay timing parameter and on access type combination, since bus turnaround does not apply to all access types. See Section 15.4.4.9.3.1.6.1, *Bus Turnaround (BUSTURNAROUND)* for more details on bus turnaround.

Table 15-222 describes the configuration required for idle cycle insertion.

**Table 15-222. Idle Cycle Insertion Configuration**

1st Access Type	BUSTURN AROUND Timing Parameter	Second Access Type	Chip-Select	Add/Data Multiplexed	CYCLE2 CYCLE SAMECSEN Parameter	CYCLE2 CYCLE DIFFCSEN Parameter	Idle Cycle Insertion Between the Two Accesses
R/W	= 0	R/W	Any	Any	0	x	No idle cycles are inserted if the two accesses are well pipelined.
R	> 0	R	Same	Nonmuxed	x	0	No idle cycles are inserted if the two accesses are well pipelined.
R	> 0	R	Different	Nonmuxed	0	0	BUSTURNAROUND cycles are inserted.
R	> 0	R/W	Any	Muxed	0	0	BUSTURNAROUND cycles are inserted.
R	> 0	W	Any	Any	0	0	BUSTURNAROUND cycles are inserted.
W	> 0	R/W	Any	Any	0	0	No idle cycles are inserted if the two accesses are well pipelined.
R/W	= 0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted.
R/W	= 0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted.
R/W	> 0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is max (BUSTURNAROUND, CYCLE2CYCLEDELAY).
R/W	> 0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is maximum (BUSTURNAROUND, CYCLE2CYCLEDELAY).

**15.4.4.9.3.1.7 Slow Device Support (TIMEPARAGRANULARITY Parameter)**

All access-timing parameters can be multiplied by 2 by setting the GPMC\_CONFIG1<sub>i</sub>[4] TIMEPARAGRANULARITY bit (where i stands for the GPMC chip-select value, i = 0 to 3). Increasing all access timing parameters allows support of slow devices.

**15.4.4.9.3.2 gpmc\_io\_dir Pin**

The gpmc\_io\_dir pin is used to control I/O direction on the GPMC data bus gpmc\_ad[15:0]. Depending on top-level pad multiplexing, this signal can be output and used externally to the device, if required.

The `gpmc_io_dir` pin is low during transmit (OUT) and high during receive (IN).

For write accesses, the `gpmc_io_dir` pin stays OUT from start-cycle time to end-cycle time.

For read accesses, the `gpmc_io_dir` pin goes from OUT to IN at `nOE` assertion time and stays IN until:

- BUSTURNAROUND is enabled:
  - The `gpmc_io_dir` pin goes from IN to OUT at end-cycle time plus programmable bus turnaround time.
- BUSTURNAROUND is disabled:
  - After an asynchronous read access, the `gpmc_io_dir` pin goes from IN to OUT at `RDACCESSTIME + 1 GPMC_FCLK` cycle or when `RDCYCLETIME` completes, whichever occurs last.
  - After a synchronous read access, the `gpmc_io_dir` pin goes from IN to OUT at `RDACCESSTIME + 2 GPMC_FCLK` cycles or when `RDCYCLETIME` completes, whichever occurs last.

Because of the bus-keeping feature of the GPMC, after a read or write access and with no other accesses pending, the default value of the `gpmc_io_dir` pin is OUT (see [Section 15.4.4.10.10, Bus Keeping Support](#)).

In nonmultiplexed devices, the `gpmc_io_dir` pin stays IN between two successive read accesses to prevent unnecessary toggling (nonmultiplexed 16-bit wide devices support is limited to 2 Kbytes).

#### 15.4.4.9.3.3 Reset

No reset signal is sent to the external memory device by the GPMC. For more information about external-device reset, see [Chapter 3, Power, Reset, and Clock Management](#).

The PRCM module provides an input pin, `global_rst_n`, to the GPMC:

- The `global_rst_n` pin is activated during device warm reset and cold reset.
- The `global_rst_n` pin initializes the internal state-machine and the internal configuration registers.

#### 15.4.4.9.3.4 Write Protect Signal (nWP)

When connected to the attached memory device, the write protect signal can enable or disable the lockdown function of the attached memory.

The `gpmc_nwp` output pin value is controlled through the `GPMC_CONFIG[4]` WRITEPROTECT bit, which is common to all CS.

#### 15.4.4.9.3.5 Byte Enable (nBE1/nBE0)

Byte enable signals (`nBE1/nBE0`) are:

- Valid (asserted or nonasserted according to the incoming system request) from access start to access completion for asynchronous and synchronous single accesses
- Asserted low from access start to access completion for asynchronous and synchronous multiple read accesses
- Valid (asserted or nonasserted, according to the incoming system request) synchronously to each written data for synchronous multiple write accesses

#### 15.4.4.9.4 Error Handling

When an error occurs in the GPMC, the error information is stored in the `GPMC_ERR_TYPE` register and the address of the illegal access is stored in the `GPMC_ERR_ADDRESS` register. The GPMC only keeps the first error abort information until the `GPMC_ERR_TYPE` register is reset. Subsequent accesses that cause errors are not logged until the error is cleared by hardware with the `GPMC_ERR_TYPE[0]` ERRORVALID bit.

- `ERRORNOTSUPPADD` occurs when an incoming system request address decoding does not match any valid chip-select region, or if two chip-select regions are defined as overlapped, or if a register file access is tried outside the valid address range of 1 Kbyte.
- `ERRORNOTSUPPMCMD` occurs when an unsupported command request is decoded at the L3 interconnect interface.

- **ERRORTIMEOUT:** A time-out mechanism prevents the system from hanging. The start value of the 9-bit time-out counter is defined in the [GPMC\\_TIMEOUT\\_CONTROL](#) register and enabled with the [GPMC\\_TIMEOUT\\_CONTROL\[0\]](#) TIMEOUTENABLE bit. When enabled, the counter starts at start-cycle time until it reaches 0 and data is not responded to from memory, then a time-out error occurs. When data are sent from memory, this counter is reset to its start value. With multiple accesses (asynchronous page mode or synchronous burst mode), the counter is reset to its start value for each data access within the burst.

The GPMC does not generate interrupts on these errors. True abort to the MPU or interrupt generation is handled at interconnect level.

#### 15.4.4.10 Timing Setting

The GPMC offers the maximum flexibility to support various access protocols. Most of the timing parameters of the protocol access used by the GPMC to communicate with attached memories or devices are programmable on a chip-select basis. Assertion and deassertion times of control signals are defined to match the attached memory or device timing specifications and to get maximum performance during accesses. For more information on GPMC\_CLK and GPMC\_FCLK see [Section 15.4.4.10.6](#), [GPMC\\_CLK](#).

---

**NOTE:** In the following sections, the start access time refer to the time at which the access begins.

---

##### 15.4.4.10.1 Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME)

The [GPMC\\_CONFIG5\\_i\[4:0\]](#) RDCYCLETIME and [GPMC\\_CONFIG5\\_i\[12:8\]](#) WRCYCLETIME fields ( $i = 0$  to 3) define the address bus and byte enables valid times for read and write accesses. To ensure a correct duty cycle of GPMC\_CLK between accesses, RDCYCLETIME and WRCYCLETIME are expressed in GPMC\_FCLK cycles and must be multiples of the GPMC\_CLK cycle. RDCYCLETIME and WRCYCLETIME bit fields can be set with a granularity of 1 or 2 through [GPMC\\_CONFIG1\\_i\[4\]](#) TIMEPARAGRANULARITY.

When either RDCYCLETIME or WRCYCLETIME completes, if they are not already deasserted, all control signals (nCS, nADV/ALE, nOE/RE, nWE, and BE0/CLE) are deasserted to their reset values, regardless of their deassertion time parameters.

An exception to this forced deassertion occurs when a pipelined request to the same chip-select or to a different chip-select is pending. In such a case, it is not necessary to deassert a control signal with deassertion time parameters equal to the cycle-time parameter. This exception to forced deassertion prevents any unnecessary glitches. This requirement also applies to BE signals, thus avoiding an unnecessary BE glitch transition when pipelining requests.

If no inactive cycles are required between successive accesses to the same or to a different chip-select ([GPMC\\_CONFIG6\\_i\[7\]](#) CYCLE2CYCLESAMECSSEN = 0 or [GPMC\\_CONFIG6\\_i\[6\]](#) CYCLE2CYCLEDIFFCSSEN = 0, where  $i = 0$  to 3), and if assertion-time parameters associated with the pipelined access are equal to 0, asserted control signals (nCS, nADV/ALE, nBE0/CLE, nWE, and nOE/RE) are kept asserted. This applies to any read/write to read/write access combination.

If inactive cycles are inserted between successive accesses, that is, [CYCLE2CYCLESAMECSSEN](#) = 1 or [CYCLE2CYCLEDIFFCSSEN](#) = 1, the control signals are forced to their respective default reset values for the number of GPMC\_FCLK cycles defined in [CYCLE2CYCLEDELAY](#):

##### 15.4.4.10.2 nCS: Chip-Select Signal Control Assertion/Deassertion Time (CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADELAY)

The [GPMC\\_CONFIG2\\_i\[3:0\]](#) CSONTIME field (where  $i = 0$  to 3) defines the nCS signal-assertion time relative to the start access time. It is common for read and write accesses.

[GPMC\\_CONFIG2\\_i\[12:8\]](#) CSRDOFFTIME (read access) and [GPMC\\_CONFIG2\\_i\[20:16\]](#) CSWROFFTIME (write access) bit fields define the nCS signal deassertion time relative to start access time.

CSONTIME, CSRDOFFTIME and CSWROFFTIME parameters are applicable to synchronous and asynchronous modes. CSONTIME can be used to control an address and byte enable setup time before chip-select assertion. CSRDOFFTIME and CSWROFFTIME can be used to control an address and byte enable hold time after chip-select deassertion.

nCS signal transitions as controlled through CSONTIME, CSRDOFFTIME, and CSWROFFTIME can be delayed by half a GPMC\_FCLK period by enabling the [GPMC\\_CONFIG2\\_i\[7\]](#) CSEXTRADELAY bit. This half of a GPMC\_FCLK period provides more granularity on the nCS assertion and deassertion time to guarantee proper setup and hold time relative to GPMC\_CLK. CSEXTRADELAY is especially useful in configurations where GPMC\_CLK and GPMC\_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, CSEXTRADELAY applies to all parameters controlling nCS transitions.

The CSEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than the nCS signal-deassertion time, including the extra half-GPMC\_FCLK-period delay.

#### **15.4.4.10.3 nADV/ALE: Address Valid/Address Latch Enable Signal Control Assertion/Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME / ADVEXTRADELAY/ADVAADMUXONTIME/ADVAADMUXRDOFFTIME/ADVAADMUXWROFFTIME)**

The [GPMC\\_CONFIG3\\_i\[3:0\]](#) ADVONTIME field (where i = 0 to 3) defines the nADV/ALE signal-assertion time relative to start access time. It is common to read and write accesses.

[GPMC\\_CONFIG3\\_i\[12:8\]](#) ADVRDOFFTIME (read access) and [GPMC\\_CONFIG3\\_i\[20:16\]](#) ADVWROFFTIME (write access) bit fields define the nADV/ALE signal-deassertion time relative to start access time.

ADVONTIME can be used to control an address and byte enable valid setup time control before nADV/ALE assertion. ADVRDOFFTIME and ADVWROFFTIME can be used to control an address and byte enable valid hold time control after nADV/ALE de-assertion. ADVRDOFFTIME and ADVWROFFTIME are applicable to both synchronous and asynchronous modes.

nADV/ALE signal transitions as controlled through ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME can be delayed by half a GPMC\_FCLK period by enabling the [GPMC\\_CONFIG3\\_i\[7\]](#) ADVEXTRADELAY bit. This half of a GPMC\_FCLK period provides more granularity on nADV/ALE assertion and deassertion time to guarantee proper setup and hold time relative to GPMC\_CLK. The ADVEXTRADELAY configuration parameter is especially useful in configurations where GPMC\_CLK and GPMC\_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, ADVEXTRADELAY applies to all parameters controlling nADV/ALE transitions.

ADVEXTRADELAY must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than nADV/ALE signal-deassertion time, including the extra half-GPMC\_FCLK-period delay.

[GPMC\\_CONFIG3\\_i\[6:4\]](#) ADVAADMUXONTIME, [GPMC\\_CONFIG3\\_i\[26:24\]](#) ADVAADMUXRDOFFTIME, and [GPMC\\_CONFIG3\\_i\[30:28\]](#) ADVAADMUXWROFFTIME parameters have the same functions as ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME, but apply to the first address phase in the AAD-multiplexed protocol. It is the user responsibility to make sure ADVAADMUXxxOFFTIME is programmed to a value lower than or equal to ADVxxOFFTIME. Functionality in AAD-mux mode is undefined if the settings do not comply with this requirement. ADVAADMUXxxOFFTIME can be programmed to the same value as ADVONTIME if no high nADV pulse is needed between the two AAD-mux address phases, which is the typical case in synchronous mode. In this configuration, nADV is kept low until it reaches the correct ADVxxOFFTIME.

See [Section 15.4.4.13, NAND Access Description](#) for more details on ADVONTIME, ADVRDOFFTIME, ADVWROFFTIME, and ADVAADMUXRDOFFTIME, ADVAADMUXWROFFTIME usage for CLE and ALE (Command / Address Latch Enable) usage for a NAND Flash interface.

#### 15.4.4.10.4 **nOE/nRE: Output Enable / Read Enable Signal Control Assertion / Deassertion Time (OEONTIME / OEOFFTIME / OEEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME)**

The [GPMC\\_CONFIG4\\_i\[3:0\]](#) OEONTIME field (where i = 0 to 3) defines the nOE/nRE signal assertion time relative to start access time. It is applicable only to read accesses.

The [GPMC\\_CONFIG4\\_i\[12:8\]](#) OEOFFTIME field defines the nOE/nRE signal deassertion time relative to start access time. It is applicable only to read accesses. nOE/nRE is not asserted during a write cycle.

OEONTIME, OEOFFTIME, OEAADMUXONTIME and OEAADMUXOFFTIME parameters are applicable to synchronous and asynchronous modes. OEONTIME can be used to control an address and byte enable valid setup time control before nOE/nRE assertion. OEOFFTIME can be used to control an address and byte enable valid hold time control after nOE/nRE assertion.

OEAADMUXONTIME and OEAADMUXOFFTIME parameters have the same functions as OEONTIME and OEOFFTIME, but apply to the first OE assertion in the AAD-multiplexed protocol for a read phase, or to the only OE assertion for a write phase. It is the user responsibility to make sure OEAADMUXOFFTIME is programmed to a value lower than OEONTIME. Functionality in AAD-mux mode is undefined if the settings do not comply with this requirement. OEAADMUXOFFTIME shall never be equal to OEONTIME because the AAD-mux protocol requires a second address phase with the nOE signal de-asserted before nOE can be asserted again to define a read command.

The nOE/RE signal transitions as controlled through OEONTIME, OEOFFTIME, OEAADMUXONTIME and OEAADMUXOFFTIME can be delayed by half a GPMC\_FCLK period by enabling the [GPMC\\_CONFIG4\\_i\[7\]](#) OEEXTRADELAY bit. This half of a GPMC\_FCLK period provides more granularity on nOE/RE assertion and deassertion time to guaranty proper setup and hold time relative to GPMC\_CLK. If enabled, OEEXTRADELAY applies to all parameters controlling nOE/nRE transitions.

OEEXTRADELAY must be used carefully, to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program RDCYCLETIME and WRCYCLETIME to be greater than nOE/RE signal-deassertion time, including the extra half-GPMC\_FCLK-period delay.

---

**NOTE:** When the GPMC generates a read access to an address-/data-multiplexed device, it drives the address bus until nOE assertion time.

---

#### 15.4.4.10.5 **nWE: Write Enable Signal Control Assertion / Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY)**

The [GPMC\\_CONFIG4\\_i\[19:16\]](#) WEONTIME field (where i = 0 to 3) defines the nWE signal-assertion time relative to start access time. The [GPMC\\_CONFIG4\\_i\[28:24\]](#) WEOFFTIME field defines the nWE signal-deassertion time relative to start access time. These bit fields only apply to write accesses. nWE is not asserted during a read cycle.

WEONTIME can be used to control an address and byte enable valid setup time control before nWE assertion. WEOFFTIME can be used to control an address and byte enable valid hold time control after nWE assertion.

nWE signal transitions as controlled through WEONTIME, and WEOFFTIME can be delayed by half a GPMC\_FCLK period by enabling the [GPMC\\_CONFIG4\\_i\[23\]](#) WEEXTRADELAY bit. This half of a GPMC\_FCLK period provides more granularity on nWE assertion and deassertion time to guaranty proper setup and hold time relative to GPMC\_CLK. If enabled, WEEXTRADELAY applies to all parameters controlling nWE transitions.

The WEEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the WRCYCLETIME bit field to be greater than the nWE signal-deassertion time, including the extra half-GPMC\_FCLK-period delay.

#### 15.4.4.10.6 **GPMC\_CLK**

GPMC\_CLK is the external clock provided to the attached synchronous memory or device.

- The GPMC\_CLK clock frequency is the GPMC\_FCLK functional clock frequency divided by 1, 2, 3, or 4, depending on the [GPMC\\_CONFIG1\\_i\[1:0\]](#) GPMCFCLKDIVIDER bit field (where i = 0 to 3), with a guaranteed 50-percent duty cycle.

- The GPMC\_CLK clock is only activated when the access in progress is defined as synchronous (read or write access).
- The [GPMC\\_CONFIG1\\_i\[26:25\]](#) CLKACTIVATIONTIME field ( $i = 0$  to 3) defines the number of GPMC\_FCLK cycles from start access time to GPMC\_CLK activation.
- The GPMC\_CLK clock is stopped when cycle time completes and is asserted low between accesses.
- The GPMC\_CLK clock is kept low when access is defined as asynchronous.

**CAUTION**

When cycle time completes, the GPMC\_CLK may be high because of the GPMCFCLKDIVIDER bit field. To ensure correct stoppage of the GPMC\_CLK clock within the 50-percent required duty cycle, it is the user's responsibility to extend the RDCYCLETIME or WRCYCLETIME value.

**NOTE:** To ensure a correct external clock cycle, the following rules must be applied:

- $(RDCYCLETIME \text{ CLKACTIVATIONTIME})$  must be a multiple of  $(GPMCFCLKDIVIDER + 1)$ .
- The PAGEBURSTACCESSTIME value must be a multiple of  $(GPMCFCLKDIVIDER + 1)$ .

**15.4.4.10.7 GPMC\_CLK and Control Signals Setup and Hold**

Control-signal transition (assertion and deassertion) setup and hold values with respect to the GPMC\_CLK edge can be controlled in the following ways:

- For the GPMC\_CLK signal, the [GPMC\\_CONFIG1\\_i\[26:25\]](#) CLKACTIVATIONTIME field ( $i = 0$  to 3) allows setup and hold control of control-signal assertion time.
- The use of a divided GPMC\_CLK allows setup and hold control of control-signal assertion and deassertion times.
- When GPMC\_CLK runs at the GPMC\_FCLK frequency so that GPMC\_CLK edge and control-signal transitions refer to the same GPMC\_FCLK edge, the control-signal transitions can be delayed by half of a GPMC\_FCLK period to provide minimum setup and hold times. This half-GPMC\_FCLK delay is enabled with the CSEXTRADELAY, ADVEXTRADELAY, OEEXTRADELAY, or WEEXTRADELAY parameter. This delay must be used carefully to prevent control-signal overlap between successive accesses to different chip-selects. This implies that the RDCYCLETIME and WRCYCLETIME are greater than the last control-signal deassertion time, including the extra half-GPMC\_FCLK cycle.

**15.4.4.10.8 Access Time (RDACCESSTIME / WRACCESSTIME)**

The read access time and write access time durations can be programmed independently through [GPMC\\_CONFIG5\\_i\[20:16\]](#) RDACCESSTIME and [GPMC\\_CONFIG6\\_i\[28:24\]](#) WRACCESSTIME ( $i = 0$  to 3). This allows nOE and GPMC data capture timing parameters to be independent of nWE and memory device data capture timing parameters. RDACCESSTIME and WRACCESSTIME bit fields can be set with a granularity of 1 or 2 through [GPMC\\_CONFIG1\\_i\[4\]](#) TIMEPARAGRANULARITY.

**15.4.4.10.8.1 Access Time on Read Access**

In asynchronous read mode, for single and paged accesses, [GPMC\\_CONFIG5\\_i\[20:16\]](#) RDACCESSTIME field ( $i = 0$  to 3) defines the number of GPMC\_FCLK cycles from start access time to the GPMC\_FCLK rising edge used for the first data capture. RDACCESSTIME must be programmed to the rounded greater value (in GPMC\_FCLK cycles) of the read access time of the attached memory device.

In synchronous read mode, for single or burst accesses, RDACCESSTIME defines the number of GPMC\_FCLK cycles from start access time to the GPMC\_FCLK rising edge corresponding to the GPMC\_CLK rising edge used for the first data capture.



GPMC\_CLK which is sent to the memory device for synchronization with the GPMC controller, is internally retimed to correctly latch the returned data. GPMC\_CONFIG5\_i[4:0] RDCYCLETIME must be greater than RDACCESSTIME in order to let the GPMC latch the last return data using the internally retimed GPMC\_CLK.

The external WAIT signal can be used in conjunction with RDACCESSTIME to control the effective GPMC data-capture GPMC\_FCLK edge on read access in both asynchronous mode and synchronous mode. For details about wait monitoring, see [Section 15.4.4.9.3.1, Chip-Select Base Address and Region Size](#).

#### **15.4.4.10.8.2 Access Time on Write Access**

In asynchronous write mode, the GPMC\_CONFIG6\_j[28:24] WRACCESSTIME timing parameter is not used to define the effective write access time. Instead, it is used as a WAIT invalid timing window, and must be set to a correct value so that the gpmc\_wait pin is at a valid state two GPMC\_CLK cycles before WRACCESSTIME completes. For details about wait monitoring, see [Section 15.4.4.9.3.1, Chip-Select Base Address and Region Size](#).

In synchronous write mode, for single or burst accesses, WRACCESSTIME defines the number of GPMC\_FCLK cycles from start access time to the GPMC\_CLK rising edge used by the memory device for the first data capture.

The external WAIT signal can be used in conjunction with WRACCESSTIME to control the effective memory device data capture GPMC\_CLK edge for a synchronous write access. For details about wait monitoring, see [Section 15.4.4.9.3.1, Chip-Select Base Address and Region Size](#).

#### **15.4.4.10.9 Page Burst Access Time (PAGEBURSTACCESSTIME)**

GPMC\_CONFIG5\_i[27:24] PAGEBURSTACCESSTIME bit field (i = 0 to 3) can be set with a granularity of 1 or 2 through GPMC\_CONFIG1\_i[4] TIMEPARAGRANULARITY.

##### **15.4.4.10.9.1 Page Burst Access Time on Read Access**

In asynchronous page read mode, the delay between successive word captures in a page is controlled through the PAGEBURSTACCESSTIME bit field. The PAGEBURSTACCESSTIME parameter must be programmed to the rounded greater value (in GPMC\_FCLK cycles) of the read access time of the attached device.

In synchronous burst read mode, the delay between successive word captures in a burst is controlled through the PAGEBURSTACCESSTIME field.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective GPMC data capture GPMC\_FCLK edge on read access. For details about wait monitoring, see [Section 15.4.4.9.3.1, Chip-Select Base Address and Region Size](#).

##### **15.4.4.10.9.2 Page Burst Access Time on Write Access**

Asynchronous page write mode is not supported. PAGEBURSTACCESSTIME is irrelevant in this case.

In synchronous burst write mode, PAGEBURSTACCESSTIME controls the delay between successive memory device word captures in a burst.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective memory-device data capture GPMC\_CLK edge in synchronous write mode. For details about wait monitoring, see [Section 15.4.4.9.3.1, Chip-Select Base Address and Region Size](#).

#### **15.4.4.10.10 Bus Keeping Support**

At the end-cycle time of a read access, if no other access is pending, the GPMC drives the bus with the last data read after RDCYCLETIME completion time to prevent bus floating and reduce power consumption.

After a write access, if no other access is pending, the GPMC keeps driving the data bus after WRCYCLETIME completes with the same data to prevent bus floating and power consumption.

#### 15.4.4.11 NOR Access Description

For each chip-select configuration, the read access can be specified as either asynchronous or synchronous access through the `GPMC_CONFIG1_j[29]` READTYPE bit ( $i = 0$  to 3). For each chip-select configuration, the write access can be specified as either synchronous or asynchronous access through the `GPMC_CONFIG1_j[27]` WRITETYPE bit ( $i = 0$  to 3).

Asynchronous and synchronous read and write access time and related control signals are controlled through timing parameters that refer to `GPMC_FCLK`. The primary difference of synchronous mode is the availability of a configurable clock interface (`GPMC_CLK`) to control the external device. Synchronous mode also affects data-capture and wait-pin monitoring schemes in read access.

For details about asynchronous and synchronous access, see the descriptions of `GPMC_CLK`, `RdAccessTime`, `WrAccessTime`, and wait-pin monitoring.

For more information about timing-parameter settings, see the sample timing diagrams in this chapter.

---

**NOTE:** The address bus and `nBE[1:0]` are fixed for the duration of a synchronous burst read access, but they are updated for each beat of an asynchronous page-read access.

---

##### 15.4.4.11.1 Asynchronous Access Description

This section describes:

- Asynchronous single-read operation on an address/data multiplexed device
- Asynchronous single write operation on an address/data-multiplexed device
- Asynchronous single read operation on an AAD-multiplexed device
- Asynchronous single write operation on an AAD-multiplexed device
- Asynchronous multiple (page) read operation on a non-multiplexed device

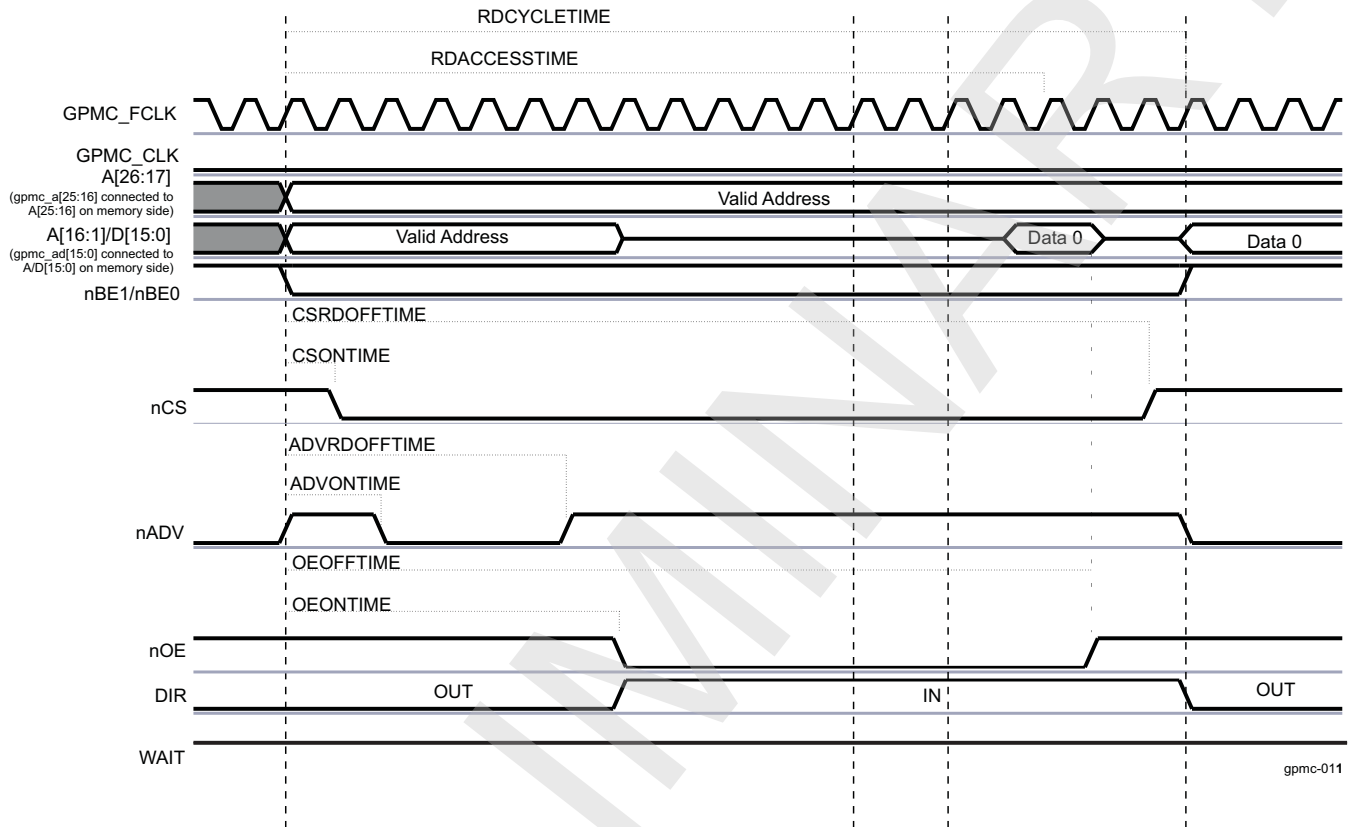
In asynchronous operations `GPMC_CLK` is not provided outside the GPMC and is kept low.

15.4.4.11.1.1 Access on Address/Data Multiplexed Devices

15.4.4.11.1.1.1 Asynchronous Single-Read Operation on an Address/Data Multiplexed Device

Figure 15-61 shows an asynchronous single read operation on an address/data-multiplexed device.

Figure 15-61. Asynchronous Single Read on an Address/Data-Multiplexed Device



See Section 15.4.5.6.1, *GPMC Timing Parameters Formulas* for formulas to calculate timing parameters.

Table 15-256 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until nOE assertion time. For details, see Section 15.4.4.9.2.3, *Address/Data-Multiplexing Interface*.

Address bits (A[16:1] from a GPMC perspective, A[15:0] from an external device perspective) are placed on the address/data bus, and the remaining address bits gpmc\_a[25:16] are placed on the address bus. The address phase ends at nOE assertion, when the DIR signal goes from OUT to IN.

- Chip-select signal nCS:
  - nCS assertion time is controlled by the GPMC\_CONFIG2\_i[3:0] CS ONTIME field. It controls the address setup time to nCS assertion.
  - nCS deassertion time is controlled by the GPMC\_CONFIG2\_i[12:8] CSRD OFFTIME field. It controls the address hold time from nCS deassertion.
- Address valid signal nADV:
  - nADV assertion time is controlled by the GPMC\_CONFIG3\_i[3:0] ADV ONTIME field.
  - nADV deassertion time is controlled by the GPMC\_CONFIG3\_i[12:8] ADV RD OFFTIME field.
- Output enable signal nOE:
  - nOE assertion indicates a read cycle.
  - nOE assertion time is controlled by the GPMC\_CONFIG4\_i[3:0] OE ONTIME field.
  - nOE deassertion time is controlled by the GPMC\_CONFIG4\_i[12:8] OE OFFTIME field.

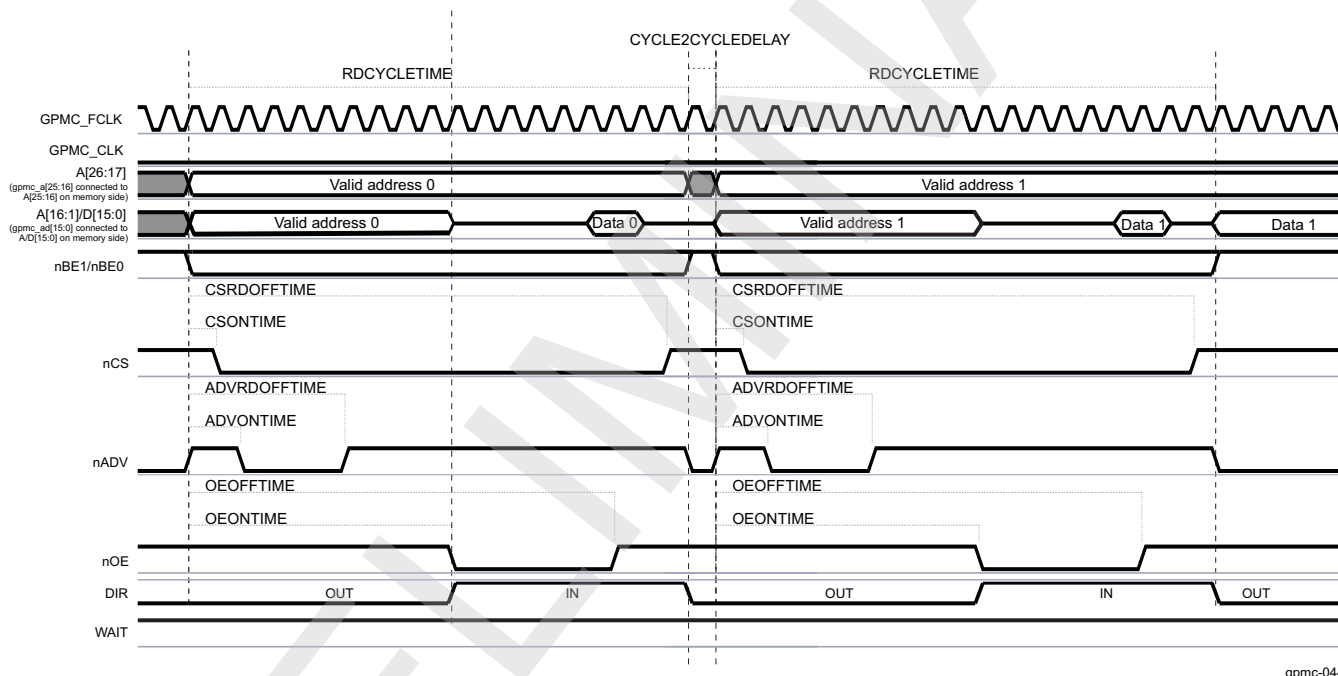
- Read data is latched when RDACCESSTIME completes. Access time is defined in the [GPMC\\_CONFIG5\\_i\[20:16\]](#) RDACCESSTIME field.
- Direction signal DIR: DIR goes from OUT to IN at the same time that nOE is asserted.
- The end of the access is defined by the [GPMC\\_CONFIG5\\_i\[4:0\]](#) RDCYCLETIME parameter.

In the GPMC, when a 16-bit wide device is attached to the controller, a 32-bit word write access is split into two 16-bit word write accesses. For more information about GPMC access size and type adaptation, see [Section 15.4.4.11.5, System Burst Versus External Device Burst Support](#).

Between two successive accesses, if an nCS pulse is needed:

- The [GPMC\\_CONFIG6\\_i\[11:8\]](#) CYCLE2CYCLEDELAY field can be programmed with [GPMC\\_CONFIG6\\_i\[7\]](#) CYCLE2CYCLESAMECSEN enabled.
- The CSWROFFTIME and CSONTIME parameters also allow a chip-select pulse, but this affects all other types of access.

**Figure 15-62. Two Asynchronous Single Read Accesses on an Address/Data Multiplexed Device (32-bit read split into 2 x 16-bit read)**

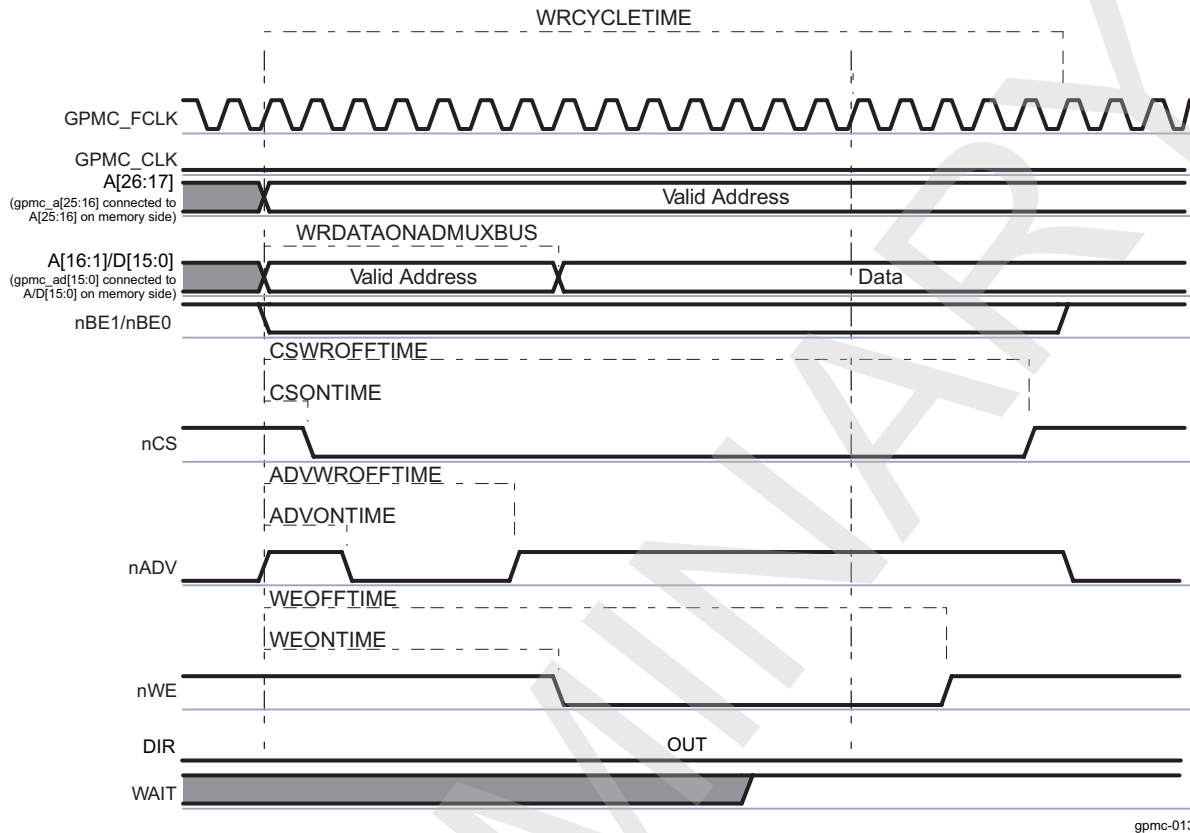


gpmc-044

#### 15.4.4.11.1.2 Asynchronous Single Write Operation on an Address/Data-Multiplexed Device

Figure 15-63 shows an asynchronous single write operation on an address/data-multiplexed device.

**Figure 15-63. Asynchronous Single Write on an Address/Data-Multiplexed Device**



gpmc-013

See [Section 15.4.5.6.1, GPMC Timing Parameters Formulas](#) for formulas to calculate timing parameters.

[Table 15-256](#) lists the timing bit fields to set up in order to configure the GPMC in asynchronous single write mode.

When the GPMC generates a write access to an address/data-multiplexed device, it drives the address bus until nWE assertion time. For more information, see [Section 15.4.4.9.2.3, Address/Data-Multiplexing Interface](#).

The nCS, and nADV signals are controlled in the same way as for asynchronous single read operation on an address/data-multiplexed device.

- Write enable signal nWE:
  - nWE assertion indicates a write cycle.
  - nWE assertion time is controlled by the [GPMC\\_CONFIG4\\_i\[19:16\] WEONTIME](#) field.
  - nWE deassertion time is controlled by the [GPMC\\_CONFIG4\\_i\[28:24\] WEOFFTIME](#) field.
- Direction signal DIR: DIR signal is OUT during the entire access.
- The end of the access is defined by the [GPMC\\_CONFIG5\\_i\[12:8\] WRCYCLETIME](#) parameter.

Address bits A[16:1] (GPMC point of view) are placed on the address/data bus at the start of cycle time, and the remaining address bits A[26:17] are placed on the address bus.

Data is driven on the address/data bus at a [GPMC\\_CONFIG6\\_i\[19:16\] WRDATAONADMUXBUS](#) time.

**NOTE:** Write multiple access in asynchronous mode is not supported. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

After a write operation, if no other access (read or write) is pending, the data bus keeps its previous value. See [Section 15.4.4.10.10, Bus Keeping Support](#).

**15.4.4.11.1.1.3 Asynchronous Multiple (Page) Write Operation on an Address/Data-Multiplexed Device**

Write multiple (page) access in asynchronous mode is not supported for address/data-multiplexed devices.

If `GPMC_CONFIG1_i[28]` `WRITEMULTIPLE` is enabled (0x1) with `GPMC_CONFIG1_i[27]` `WRITETYPE` as asynchronous (0x0), the GPMC processes single asynchronous accesses.

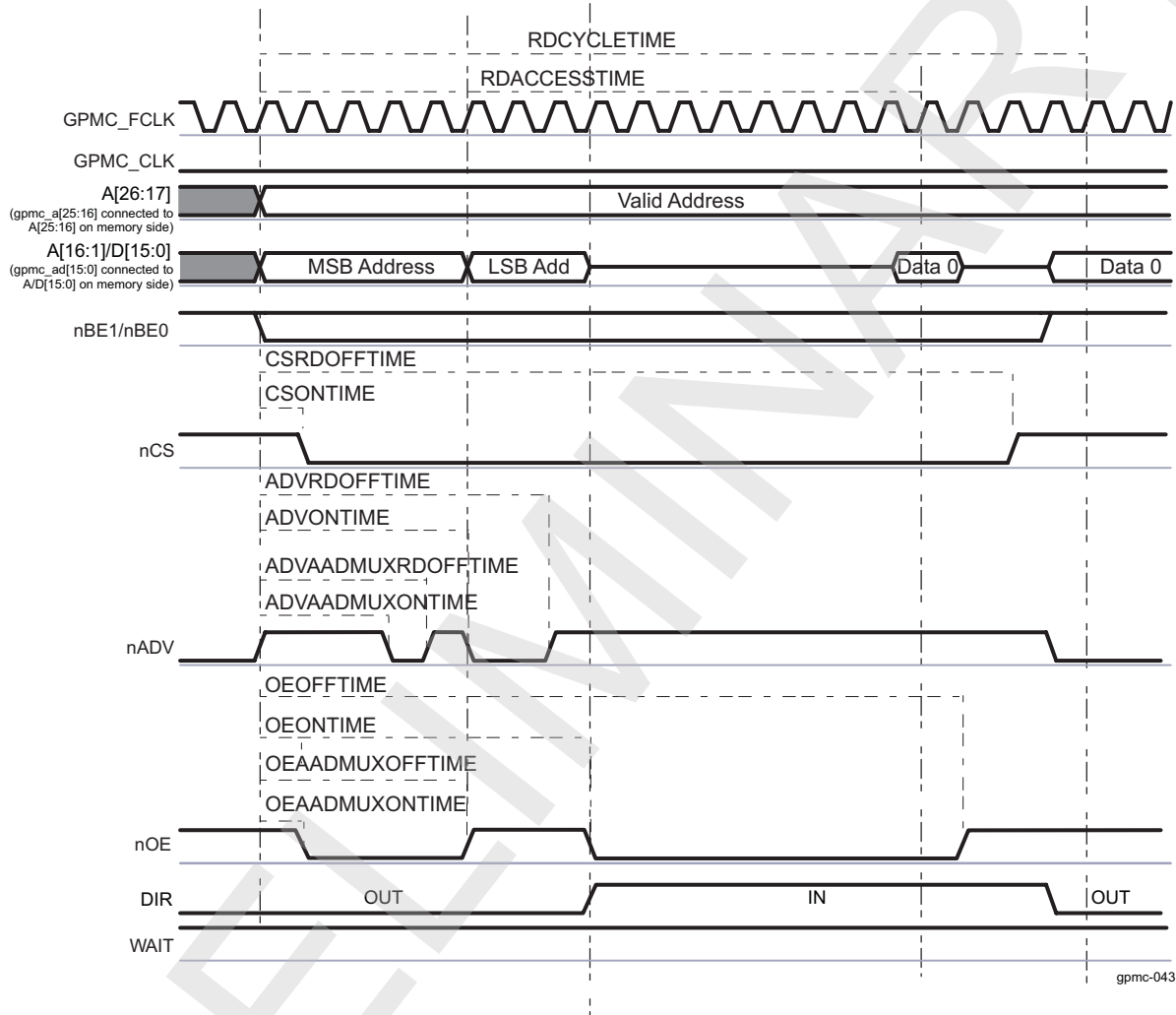
For accesses on non-multiplexed devices refer to [Section 15.4.4.11.3, Asynchronous and Synchronous Accesses in Nonmultiplexed Mode](#).

15.4.4.11.1.2 Access on Address/Address/Data Multiplexed Devices

15.4.4.11.1.2.1 Asynchronous Single Read Operation on an AAD-Multiplexed Device

Figure 15-64 shows an asynchronous single read operation on an AAD-multiplexed device.

Figure 15-64. Asynchronous Single Read on an AAD-Multiplexed Device



See Section 15.4.5.6.1, *GPMC Timing Parameters Formulas* for formulas to calculate timing parameters.

Table 15-256 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single write mode.

When the GPMC generates a read access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The first address phase ends at the first nOE deassertion time. The second phase for LSB address is qualified with nOE driven high. The second address phase ends at the second nOE assertion time, when the DIR signal goes from OUT to IN.

The nCS, and DIR signals are controlled in the same way as for asynchronous single read operation on an address/data-multiplexed device.

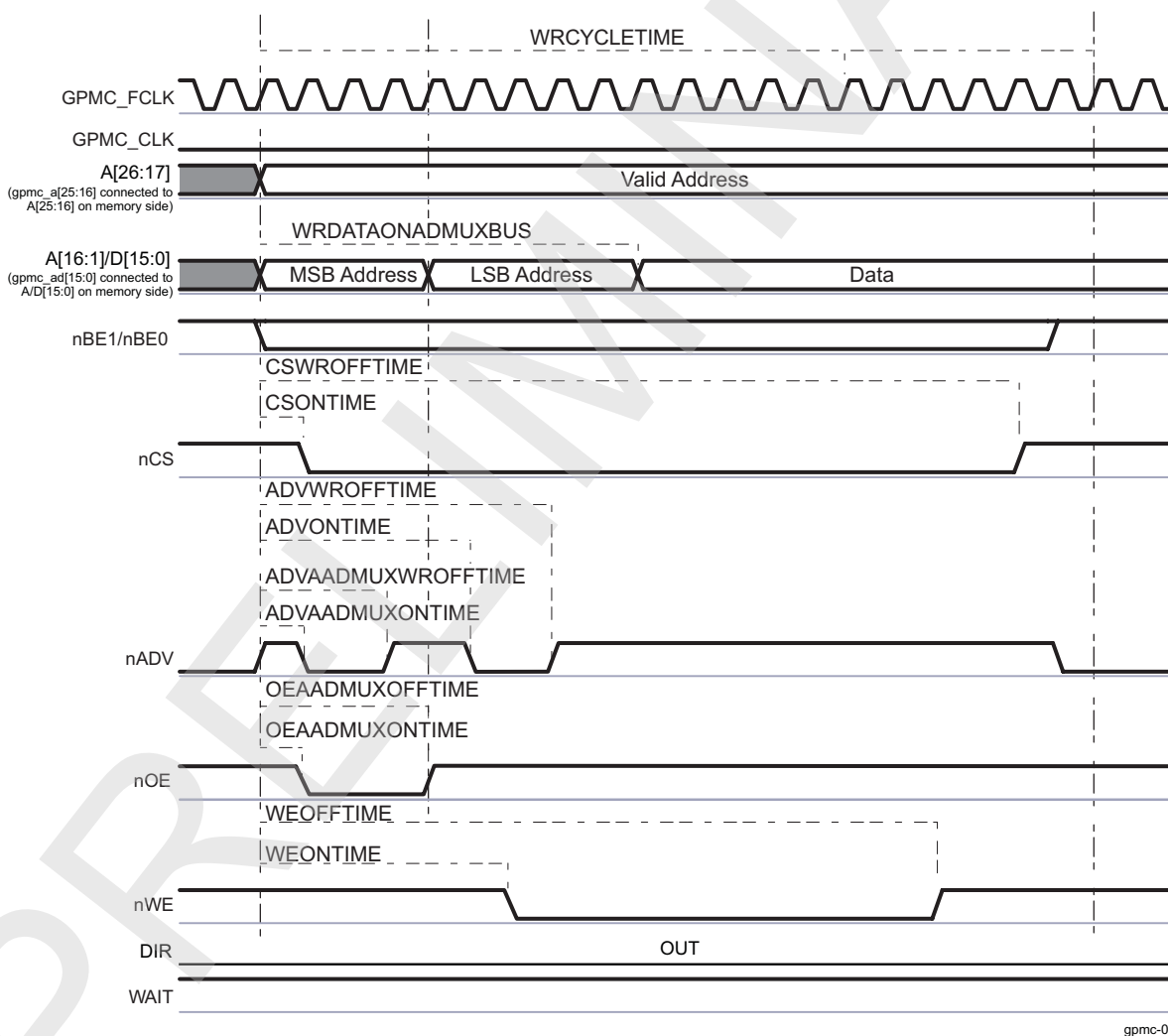
- Address valid signal nADV. nADV is asserted and deasserted twice during a read transaction:
  - nADV first assertion time is controlled by the GPMC\_CONFIG3\_i[6:4] ADVAADMUXONTIME field.
  - nADV first deassertion time is controlled by the GPMC\_CONFIG3\_i[26:24] ADVAADMUXRDOFFTIME field.

- nADV second assertion time is controlled by the [GPMC\\_CONFIG3\\_i\[3:0\]](#) ADVONTIME field.
- nADV second deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[12:8\]](#) ADVRDOFFTIME field.
- Output Enable signal nOE. nOE is asserted and deasserted twice during a read transaction (nOE second assertion indicates a read cycle):
  - nOE first assertion time is controlled by the [GPMC\\_CONFIG4\\_i\[6:4\]](#) OEAADMUXONTIME field.
  - nOE first deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[15:13\]](#) OEAADMUXOFFTIME field.
  - nOE second assertion time is controlled by the [GPMC\\_CONFIG4\\_i\[3:0\]](#) OEONTIME field.
  - nOE second deassertion time is controlled by the [GPMC\\_CONFIG4\\_i\[12:8\]](#) OEOFFTIME field.

#### 15.4.4.11.1.2.2 Asynchronous Single Write Operation on an AAD-Multiplexed Device

Figure 15-65 shows an asynchronous single write operation on an AAD-multiplexed device.

**Figure 15-65. Asynchronous Single Write on an AAD-Multiplexed Device**



See [Section 15.4.5.6.1, GPMC Timing Parameters Formulas](#) for formulas to calculate timing parameters.

[Table 15-256](#) lists the timing bit fields to set up in order to configure the GPMC in asynchronous single write mode.



When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The second phase for LSB address is qualified with nOE driven high. The address phase ends at nWE assertion time.

The nCS, nWE, and DIR signals are controlled in the same way as for asynchronous single write operation on an address/data-multiplexed device. See [Table 15-247](#).

- Address valid signal nADV is asserted and deasserted twice during a write transaction:
  - nADV first assertion time is controlled by the [GPMC\\_CONFIG3\\_i\[6:4\]](#) ADVAADMUXONTIME field.
  - nADV first deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[30:28\]](#) ADVAADMUXWROFFTIME field.
  - nADV second assertion time is controlled by the [GPMC\\_CONFIG3\\_i\[3:0\]](#) ADVONTIME field.
  - nADV second deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[20:16\]](#) ADVWROFFTIME field.
- Output Enable signal nOE is asserted during the address phase of a write transaction:
  - nOE assertion time is controlled by the [GPMC\\_CONFIG4\\_i\[6:4\]](#) OEAADMUXONTIME field.
  - nOE deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[15:13\]](#) OEAADMUXOFFTIME field.

The address bits for the first address phase are driven onto the data bus until nOE deassertion. Data is driven onto the address/data bus at the clock edge defined by the [GPMC\\_CONFIG6\\_i\[19:16\]](#) WRDATAONADMUXBUS parameter.

#### 15.4.4.11.1.2.3 Asynchronous Multiple (Page) Read Operation on an AAD-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for AAD-multiplexed devices.

If [GPMC\\_CONFIG1\\_i\[28\]](#) WRITEMULTIPLE is enabled (0x1) with [GPMC\\_CONFIG1\\_i\[27\]](#) WRITETYPE as asynchronous (0x0), the GPMC processes single asynchronous accesses.

For accesses on non-multiplexed devices refer to [Section 15.4.4.11.3](#), *Asynchronous and Synchronous Accessed in Nonmultiplexed Mode*.

#### 15.4.4.11.2 Synchronous Access Description

This section details read and write synchronous accesses on address/data multiplexed. All information in this section can be applied to any type of memory - non-multiplexed, address and data multiplexed or AAD-multiplexed - with a difference limited to the address phase. For accesses on non-multiplexed devices refer to [Section 15.4.4.11.3](#), *Asynchronous and Synchronous Accessed in Nonmultiplexed Mode*.

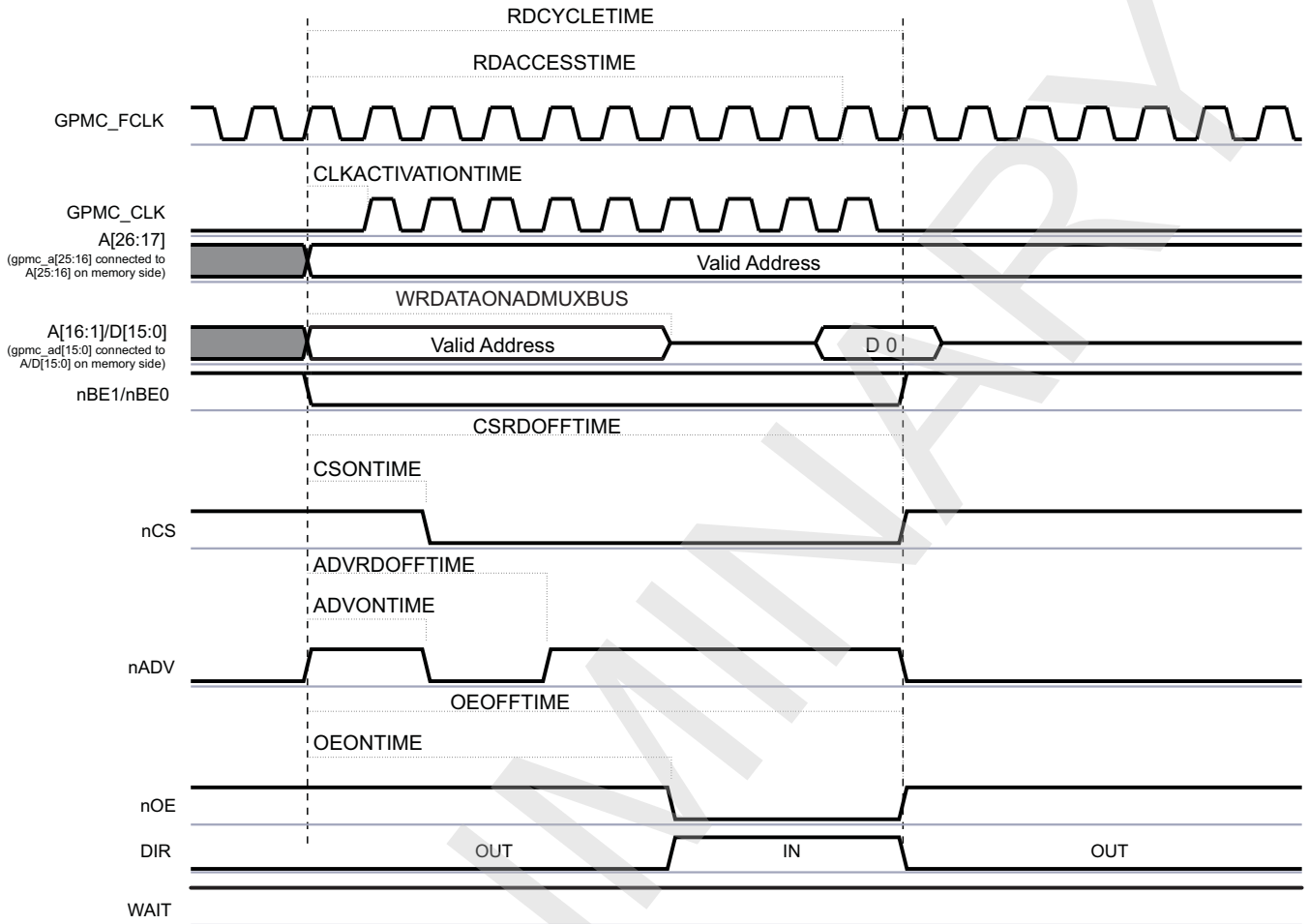
In synchronous operations:

- The GPMC\_CLK clock is provided outside the GPMC when accessing the memory device.
- The GPMC\_CLK clock is derived from the GPMC\_FCLK clock using the [GPMC\\_CONFIG1\\_i\[1:0\]](#) GPMCFCLKDIVIDER field. In the following section i stands for the chip-select number, i = 0 to 3.
- The [GPMC\\_CONFIG1\\_i\[26:25\]](#) CLKACTIVATIONTIME field specifies that the GPMC\_CLK is provided outside the GPMC 0, 1, or 2 GPMC\_FCLK cycles after start access time until RDCYCLETIME or WRCYCLETIME completion.

#### 15.4.4.11.2.1 Synchronous Single Read

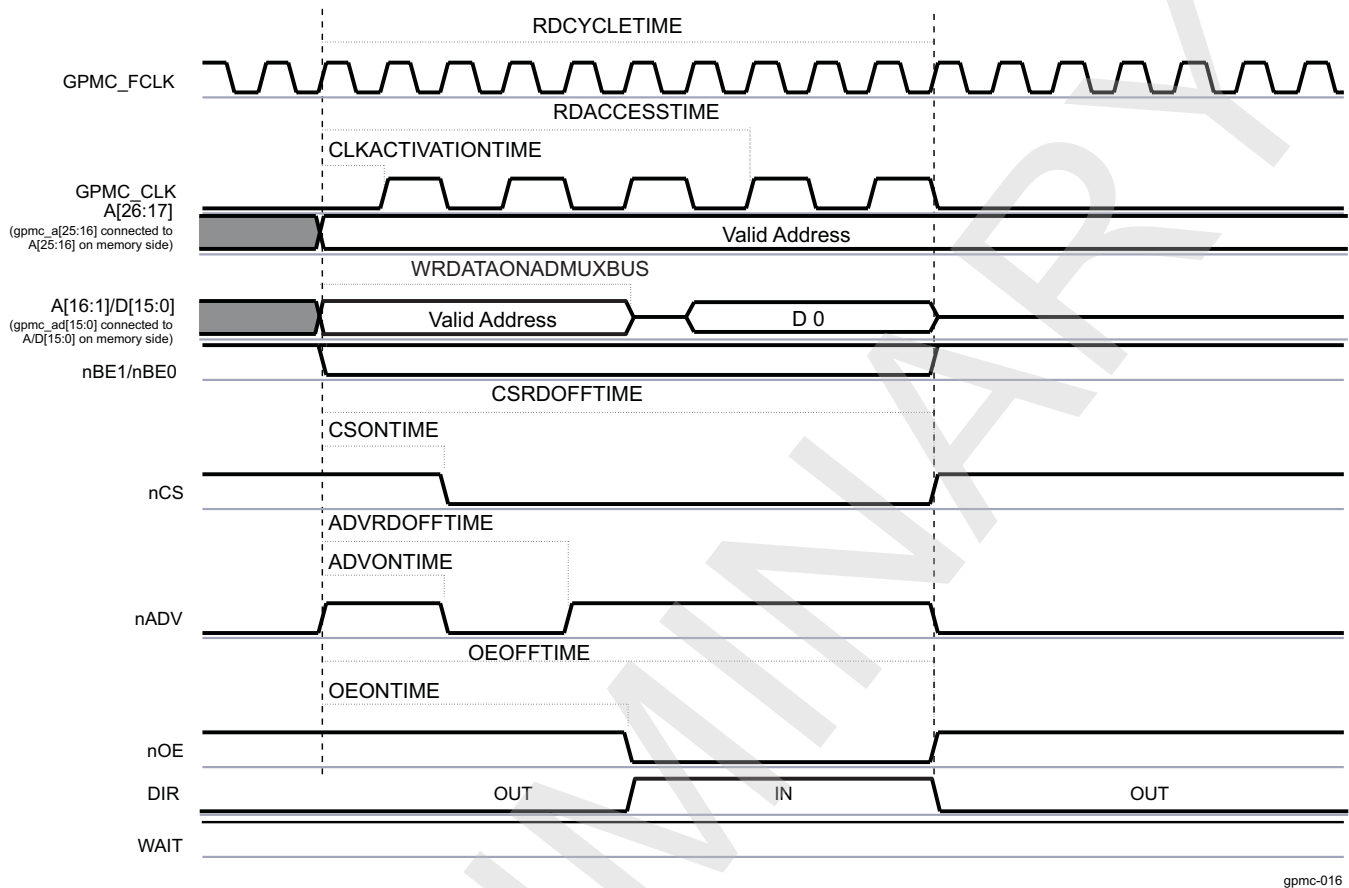
[Figure 15-66](#) and [Figure 15-67](#) show a synchronous single-read operation with GPMCFCLKDIVIDER equal to 0 and 1, respectively.

**Figure 15-66. Synchronous Single Read (GPMCFCLKDIVIDER = 0)**



gpmc-015

Figure 15-67. Synchronous Single Read (GPMCFCLKDIVIDER = 1)



gpmc-016

See Section 15.4.5.6.1, *GPMC Timing Parameters Formulas* for formulas to calculate timing parameters.

Table 15-256 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until nOE assertion time. For details, see Section 15.4.4.9.2.3, *Address/Data-Multiplexing Interface*.

- Chip-select signal nCS:
  - nCS assertion time is controlled by the GPMC\_CONFIG2\_i[3:0] CS ONTIME field and ensures address setup time to nCS assertion.
  - nCS deassertion time is controlled by the GPMC\_CONFIG2\_i[12:8] CSRD OFFTIME field and ensures address hold time to nCS deassertion.
- Address valid signal nADV:
  - nADV assertion time is controlled by the GPMC\_CONFIG3\_i[3:0] ADV ONTIME field.
  - nADV deassertion time is controlled by the GPMC\_CONFIG3\_i[12:8] ADVRD OFFTIME field.
- Output enable signal nOE:
  - nOE assertion indicates a read cycle.
  - nOE assertion time is controlled by the GPMC\_CONFIG4\_i[3:0] OE ONTIME field.
  - nOE deassertion time is controlled by the GPMC\_CONFIG4\_i[12:8] OE OFFTIME field.
- Initial latency for the first read data is controlled by GPMC\_CONFIG5\_i[20:16] RDACCESSTIME or by monitoring the WAIT signal.
- Total access time (GPMC\_CONFIG5\_i[4:0] RDCYCLETIME) corresponds to RDACCESSTIME plus the address hold time from nCS deassertion, plus time from RDACCESSTIME to CSRD OFFTIME.
- Direction signal DIR: DIR goes from OUT to IN at the same time as nOE assertion.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The second phase for LSB address is qualified with nOE driven high. The address phase ends at nWE assertion time.

The nCS, and DIR signals are controlled in the same way as for synchronous single read operation on an address/data-multiplexed device.

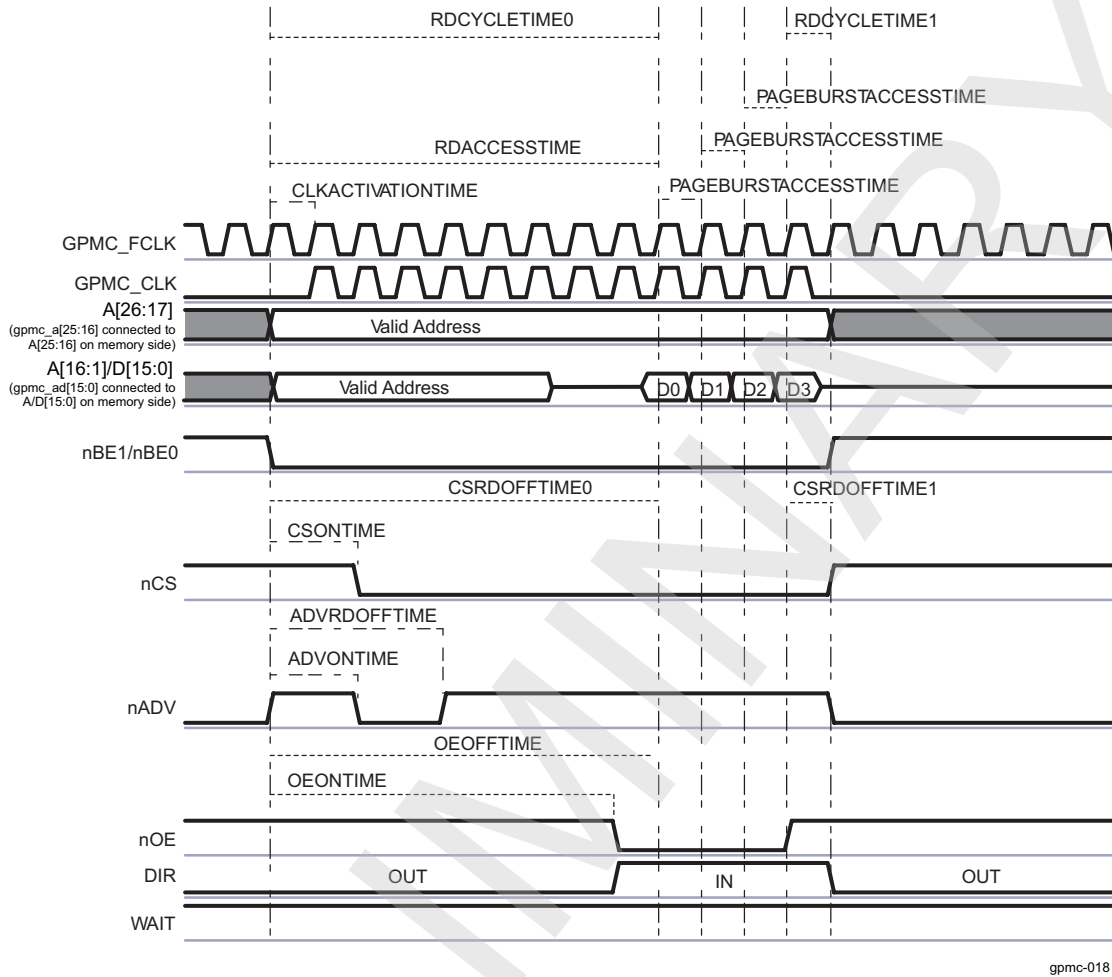
- Address valid signal nADV is asserted and deasserted twice during a read transaction:
  - nADV first assertion time is controlled by the [GPMC\\_CONFIG3\\_i\[6:4\]](#) ADVAADMUXONTIME field.
  - nADV first deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[26:24\]](#) ADVAADMUXRDOFFTIME field.
  - nADV second assertion time is controlled by the [GPMC\\_CONFIG3\\_i\[3:0\]](#) ADVONTIME field.
  - nADV second deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[12:8\]](#) ADVRDOFFTIME field.
- Output Enable signal nOE is asserted and deasserted twice during a read transaction (nOE second assertion indicates a read cycle):
  - nOE first assertion time is controlled by the [GPMC\\_CONFIG4\\_i\[6:4\]](#) OEAADMUXONTIME field.
  - nOE first deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[15:13\]](#) OEAADMUXOFFTIME field.
  - nOE second assertion time is controlled by the [GPMC\\_CONFIG4\\_i\[3:0\]](#) OEONTIME field.
  - nOE second deassertion time is controlled by the [GPMC\\_CONFIG4\\_i\[12:8\]](#) OEOFFTIME field.

After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See [Section 15.4.4.10.10](#), *Bus Keeping Support*.

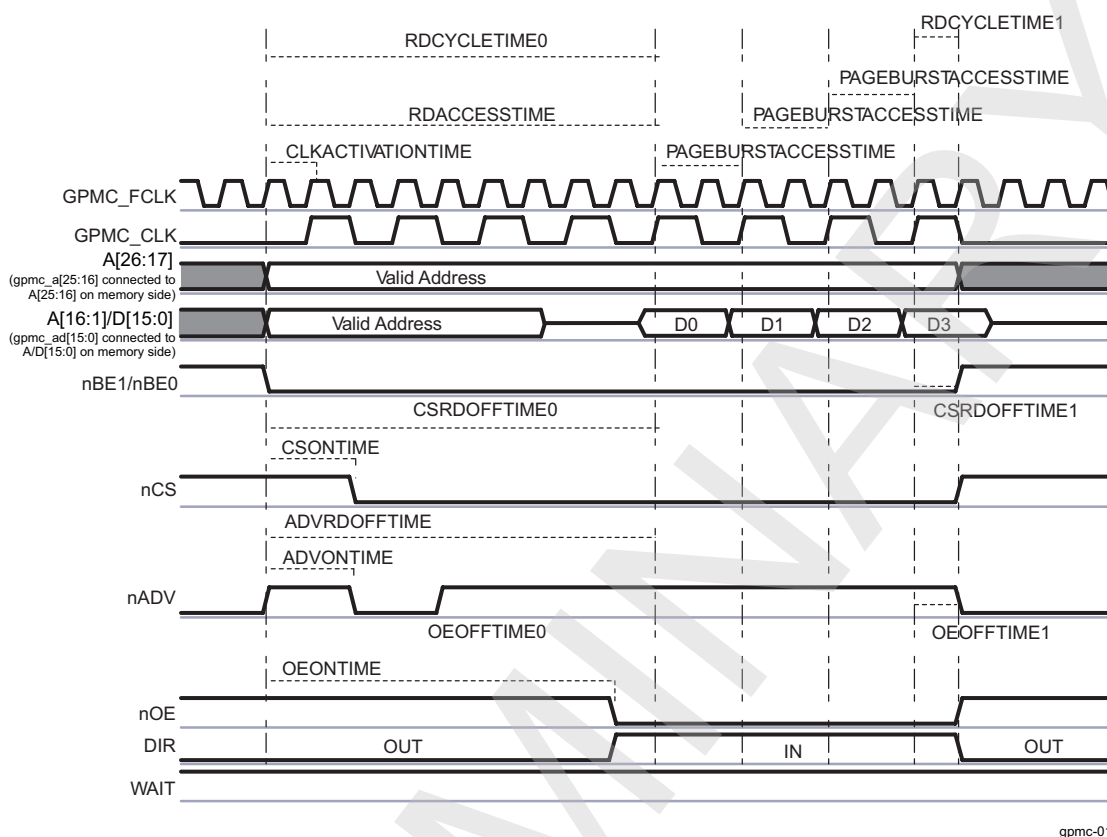
#### 15.4.4.11.2.2 Synchronous Multiple (Burst) Read (4-, 8-, 16-Word16 Burst With Wraparound Capability)

[Figure 15-68](#) and [Figure 15-69](#) show a synchronous multiple read operation with GPMCFCLKDivider equal to 0 and 1, respectively.

Figure 15-68. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0)



gpmc-018

**Figure 15-69. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1)**

gpmc-019

When [GPMC\\_CONFIG5\\_j\[20:16\]](#) RDACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to [GPMC\\_CONFIG5\\_j\[27:24\]](#) PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

The nCS, nADV, nOE and DIR signals are controlled in the same way as for synchronous single read operation. See [Table 15-242](#).

Initial latency for the first read data is controlled by RDACCESSTIME or by monitoring the WAIT signal. Successive read data are provided by the memory device each one or two GPMC\_CLK cycles. The PAGEBURSTACCESSTIME parameter must be set accordingly with [GPMC\\_CONFIG1\\_i\[1:0\]](#) GPMCFCLKDIVIDER and the memory-device internal configuration. Depending on the device page length, the GPMC checks device page crossing during a new burst request and purposely insert initial latency (of RDACCESSTIME) when required.

Total access time [GPMC\\_CONFIG5\\_j\[4:0\]](#) RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time from nCS deassertion. In [Figure 15-69](#), RDCYCLETIME programmed value equals to RDCYCLETIME0 + RDCYCLETIME1.

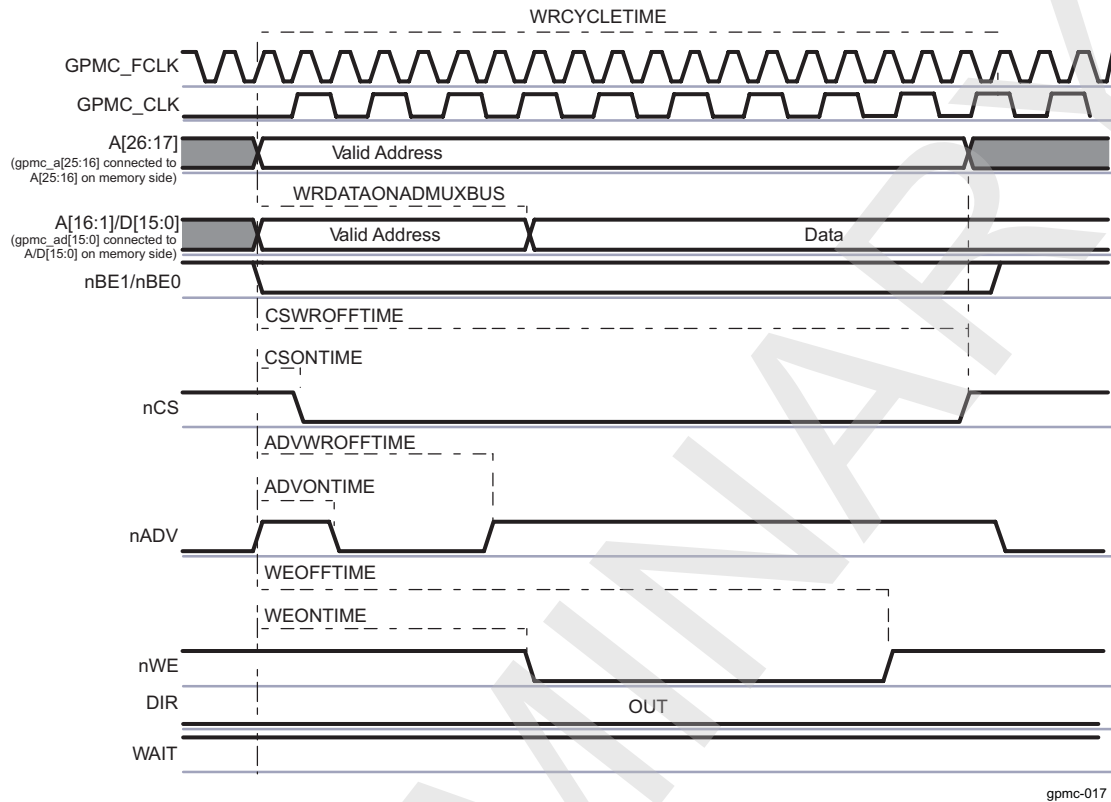
After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See [Section 15.4.4.10.10, Bus Keeping Support](#).

Burst wraparound is enabled through the [GPMC\\_CONFIG1\\_i\[31\]](#) WRAPBURST bit and allows a 4-, 8-, or 16-Word16 linear burst access to wrap within its burst-length boundary through [GPMC\\_CONFIG1\\_i\[24:23\]](#) ATTACHEDDEVICEPAGELENGTH.

### 15.4.4.11.2.3 Synchronous Single Write

Burst write mode is used for synchronous single or burst accesses.

**Figure 15-70. Synchronous Single Write on an Address/Data-Multiplexed Device**



When the GPMC generates a write access to an address/data-multiplexed device, it drives the data bus (with address bits A[16:1]) until GPMC\_CONFIG6\_i[19:16] WRDATAONADMUXBUS time. First data of the burst is driven on the address/data bus at WRDATAONADMUXBUS time.

#### 15.4.4.11.2.4 Synchronous Multiple (Burst) Write

Synchronous burst write mode provides synchronous single or consecutive accesses.

Figure 15-71 shows a synchronous burst write access when the chip-select is configured in address/data-multiplexed mode.

**Figure 15-71. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode**

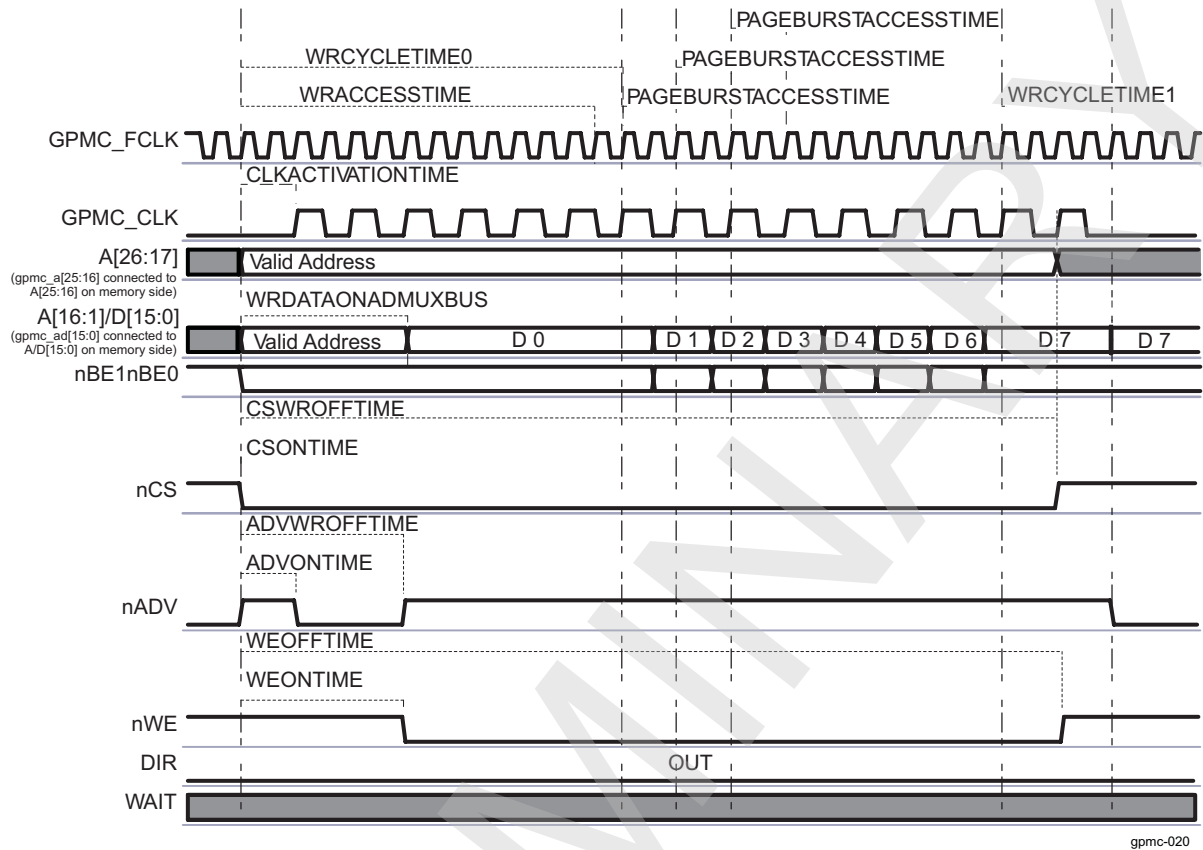
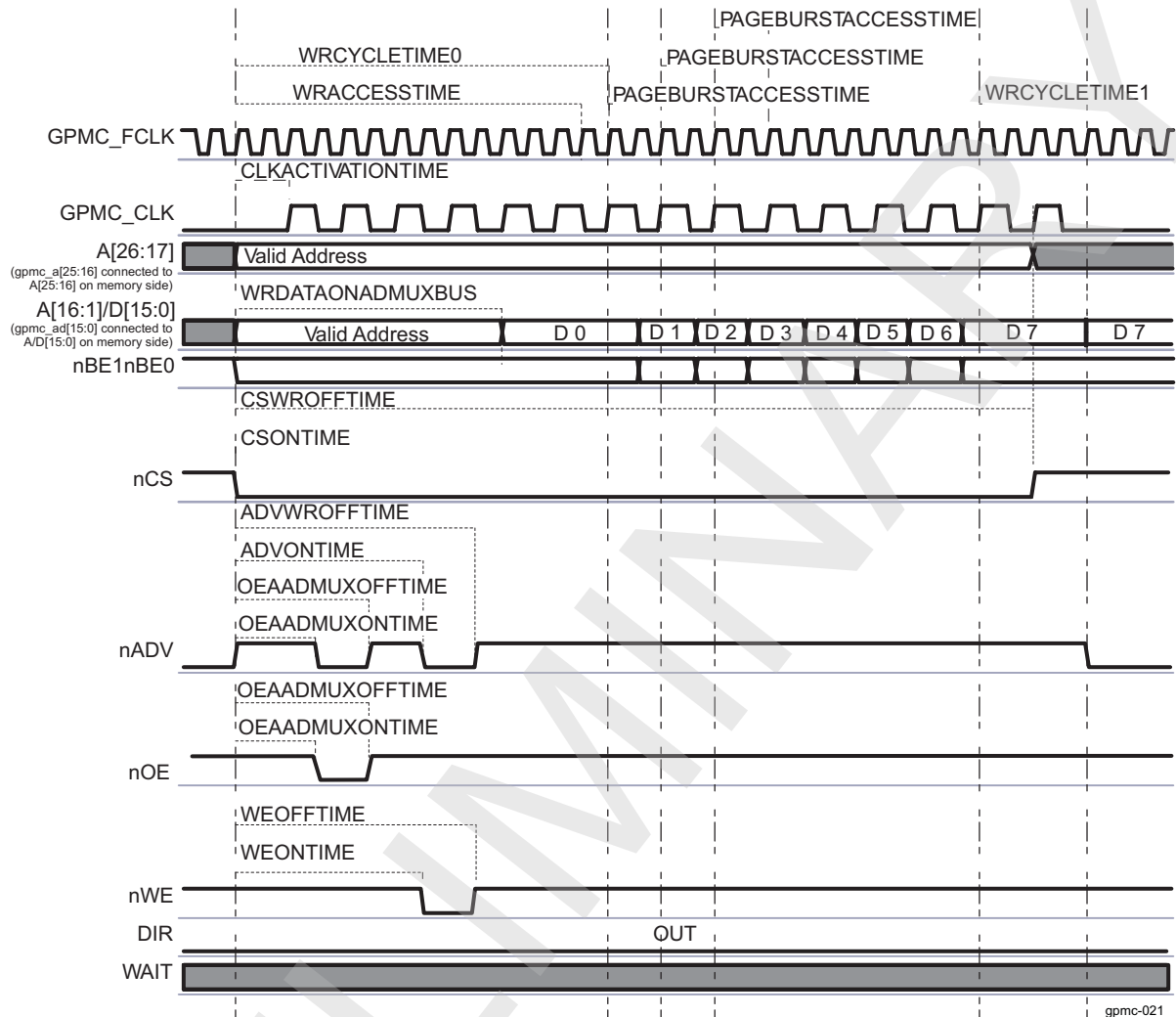


Figure 15-72 shows the same synchronous burst write access when the chip-select is configured in address/address/data-multiplexed (AAD-multiplexed) mode.



Figure 15-72. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode



The first data of the burst is driven on the A/D bus at `GPMC_CONFIG6_i[19:16]` WRDATAONADMUXBUS.

When WRACCESTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to the `GPMC_CONFIG5_i[27:24]` PAGEBURSTACCESTIME multiplied by the number of remaining data transactions.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until nOE assertion time. For details, see [Section 15.4.4.9.2.3, Address/Data-Multiplexing Interface](#).

- Chip-select signal nCS:
  - nCS assertion time is controlled by the `GPMC_CONFIG2_i[3:0]` CSONTIME field (i = 0 to 3) and ensures address setup time to nCS assertion.
  - nCS deassertion time controlled by the `GPMC_CONFIG2_i[20:16]` CSWROFFTIME field and ensures address hold time to nCS deassertion.
- Address valid signal nADV:
  - nADV assertion time is controlled by the `GPMC_CONFIG3_i[3:0]` ADVONTIME field.
  - nADV deassertion time is controlled by the `GPMC_CONFIG3_i[20:16]` ADVWROFFTIME field.
- Write enable signal nWE:
  - nWE assertion indicates a read cycle.
  - nWE assertion time is controlled by the `GPMC_CONFIG4_i[19:16]` WEONTIME field.

- nWE deassertion time is controlled by the [GPMC\\_CONFIG4\\_i\[28:24\]](#) WEOFFTIME field.

---

**NOTE:** The nWE falling edge must not be used to control the time when the burst first data is driven in the address/data bus because some new devices require the nWE signal at low during the address phase.

---

- Direction signal DIR is OUT during the entire access.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The second phase for LSB address is qualified with nOE driven high. The address phase ends at nWE assertion time.

The nCS, and DIR signals are controlled as detailed above.

- Address valid signal nADV is asserted and deasserted twice during a read transaction:
  - nADV first assertion time is controlled by the [GPMC\\_CONFIG3\\_i\[6:4\]](#) ADVAADMUXONTIME field.
  - nADV first deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[26:24\]](#) ADVAADMUXRDOFFTIME field.
  - nADV second assertion time is controlled by the [GPMC\\_CONFIG3\\_i\[3:0\]](#) ADVONTIME field.
  - nADV second deassertion time is controlled by the [GPMC\\_CONFIG3\\_i\[12:8\]](#) ADVRDOFFTIME field.
- Output Enable signal nOE is asserted and deasserted twice during a read transaction (nOE second assertion indicates a read cycle):
  - nOE first assertion time is controlled by the [GPMC\\_CONFIG4\\_i\[6:4\]](#) OEAADMUXONTIME field.
  - nOE first deassertion time is controlled by the [GPMC\\_CONFIG4\\_i\[15:13\]](#) OEAADMUXOFFTIME field.
  - nOE second assertion time is controlled by the [GPMC\\_CONFIG4\\_i\[3:0\]](#) OEONTIME field.
  - nOE second deassertion time is controlled by the [GPMC\\_CONFIG4\\_i\[12:8\]](#) OEOFFTIME field.

First write data is driven by the GPMC at [GPMC\\_CONFIG6\\_i\[19:16\]](#) WRDATAONADMUXBUS, when in address/data mux configuration. The next write data of the burst is driven on the bus at WRACCESSTIME + 1 during [GPMC\\_CONFIG5\\_i\[27:24\]](#) PAGEBURSTACCESSTIME GPMC\_FCLK cycles. The last data of the synchronous burst write is driven until [GPMC\\_CONFIG5\\_i\[12:8\]](#) WRCYCLETIME completes.

- WRACCESSTIME is defined in the [GPMC\\_CONFIG6\\_i\[28:24\]](#) register.
- The PAGEBURSTACCESSTIME parameter must be set accordingly with GPMCFCLKDIVIDER and the memory-device internal configuration.

Total access time [GPMC\\_CONFIG5\\_i\[12:8\]](#) WRCYCLETIME corresponds to WRACCESSTIME plus the address hold time from nCS deassertion. In [Figure 15-71](#), the WRCYCLETIME programmed value equals WRCYCLETIME0 + WRCYCLETIME1. WRCYCLETIME0 and WRCYCLETIME1 delays are not actual parameters and are only a graphical representation of the full WRCYCLETIME value.

After a write operation, if no other access (read or write) is pending, the data bus keeps the previous value. See [Section 15.4.4.10.10, Bus Keeping Support](#).

#### 15.4.4.11.3 Asynchronous and Synchronous Accesses in Nonmultiplexed Mode

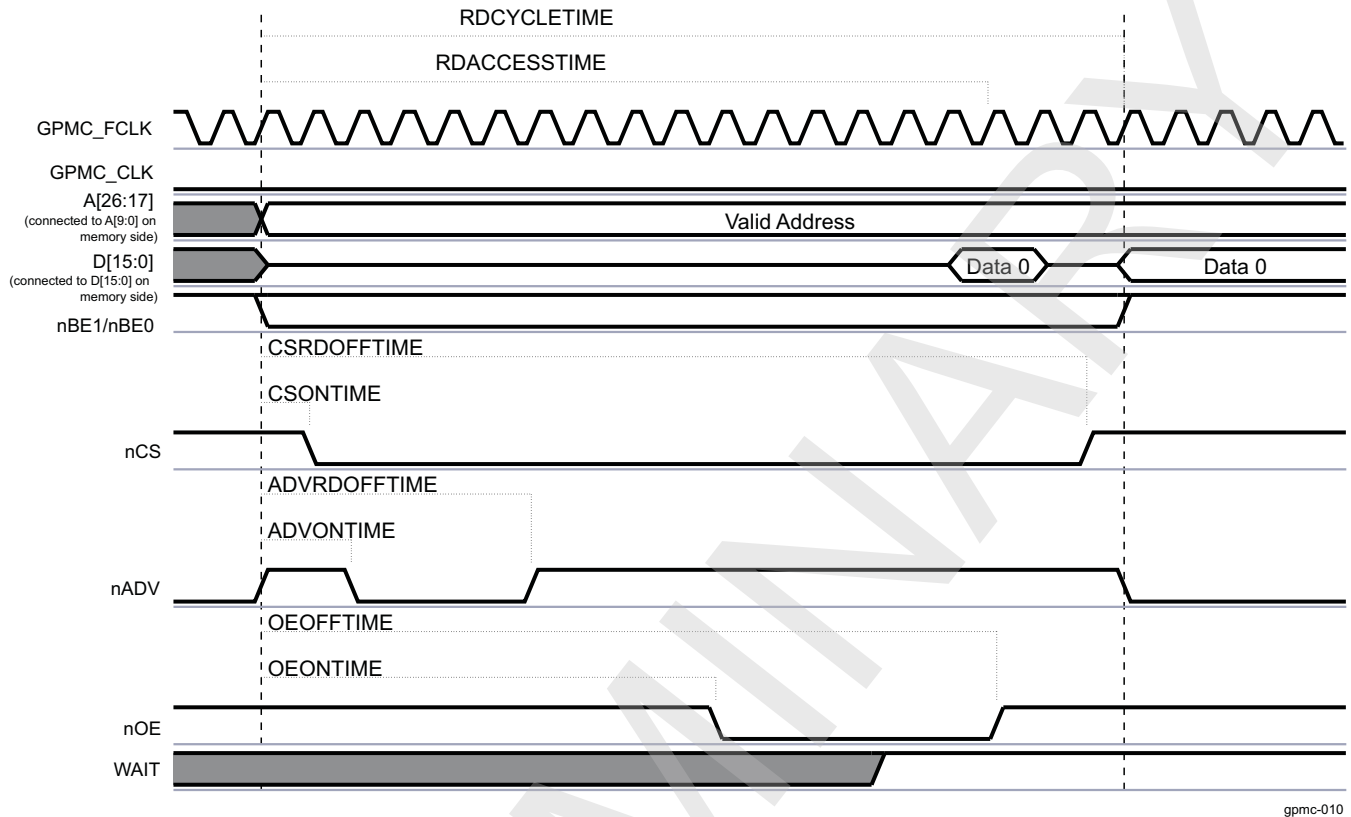
Page mode is only available in non-muxed mode. The non-muxed mode is described in this section even though its use is very limited (address space limited to 2 KBytes).

- Asynchronous single read operation on a nonmultiplexed device
- Asynchronous single write operation on a nonmultiplexed device
- Asynchronous multiple (page mode) read operation on a nonmultiplexed device
- Synchronous operations on a nonmultiplexed device

##### 15.4.4.11.3.1 Asynchronous Single Read Operation on a Nonmultiplexed Device

[Figure 15-73](#) shows an asynchronous single read operation on a nonmultiplexed device.

**Figure 15-73. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device**



The 10-bit address is driven onto the address bus A[10:1] and the 16-bit data is driven onto the data bus D[15:0].

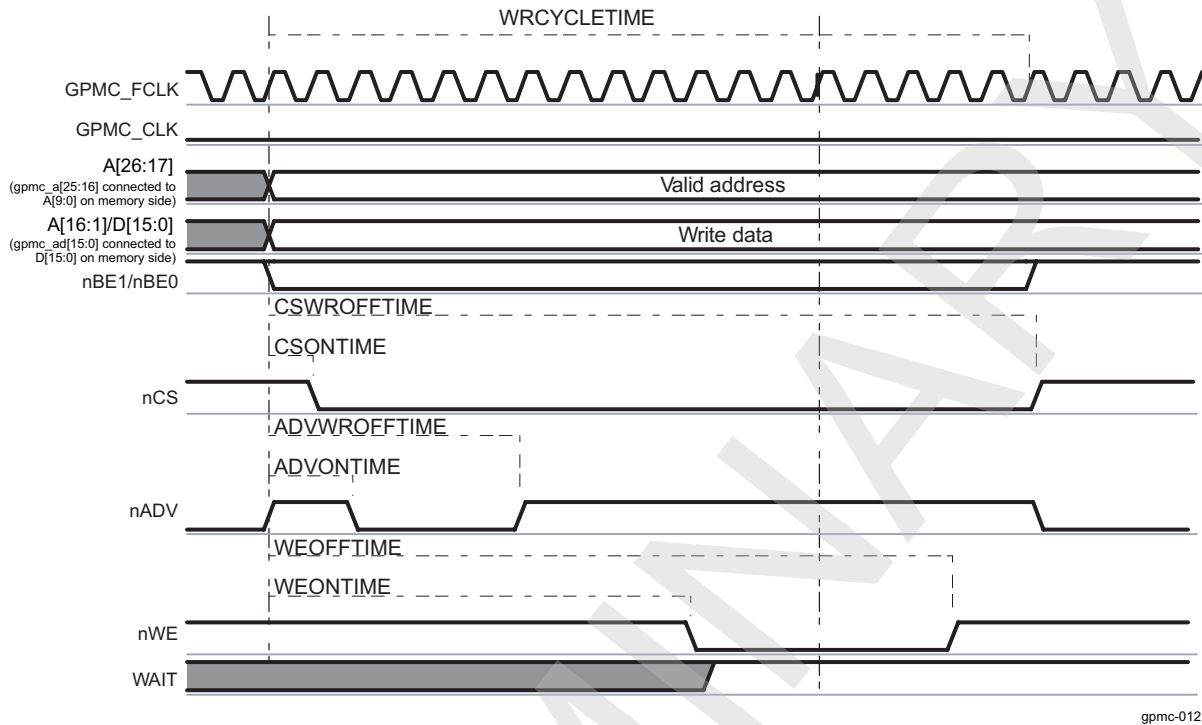
Read data is latched at [GPMC\\_CONFIG1\\_5\[20:16\]](#) RDACCESSTIME completion time. The end of the access is defined by the [GPMC\\_CONFIG1\\_5\[4:0\]](#) RDCYCLETIME parameter.

Set [GPMC\\_CONFIG\[1\] LIMITEDADDRESS](#) to 0x1 (A26-A11 are not modified during an external memory access).

nCS, nADV, nOE and DIR signals are controlled in the same way as address/data multiplexed accesses, see [Table 15-247](#).

#### 15.4.4.11.3.2 Asynchronous Single Write Operation on a Nonmultiplexed Device

[Figure 15-74](#) shows an asynchronous single write operation on a nonmultiplexed device.

**Figure 15-74. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device**

The 10-bit address is driven onto the address bus A[10:1] and the 16-bit data is driven onto the data bus D[15:0].

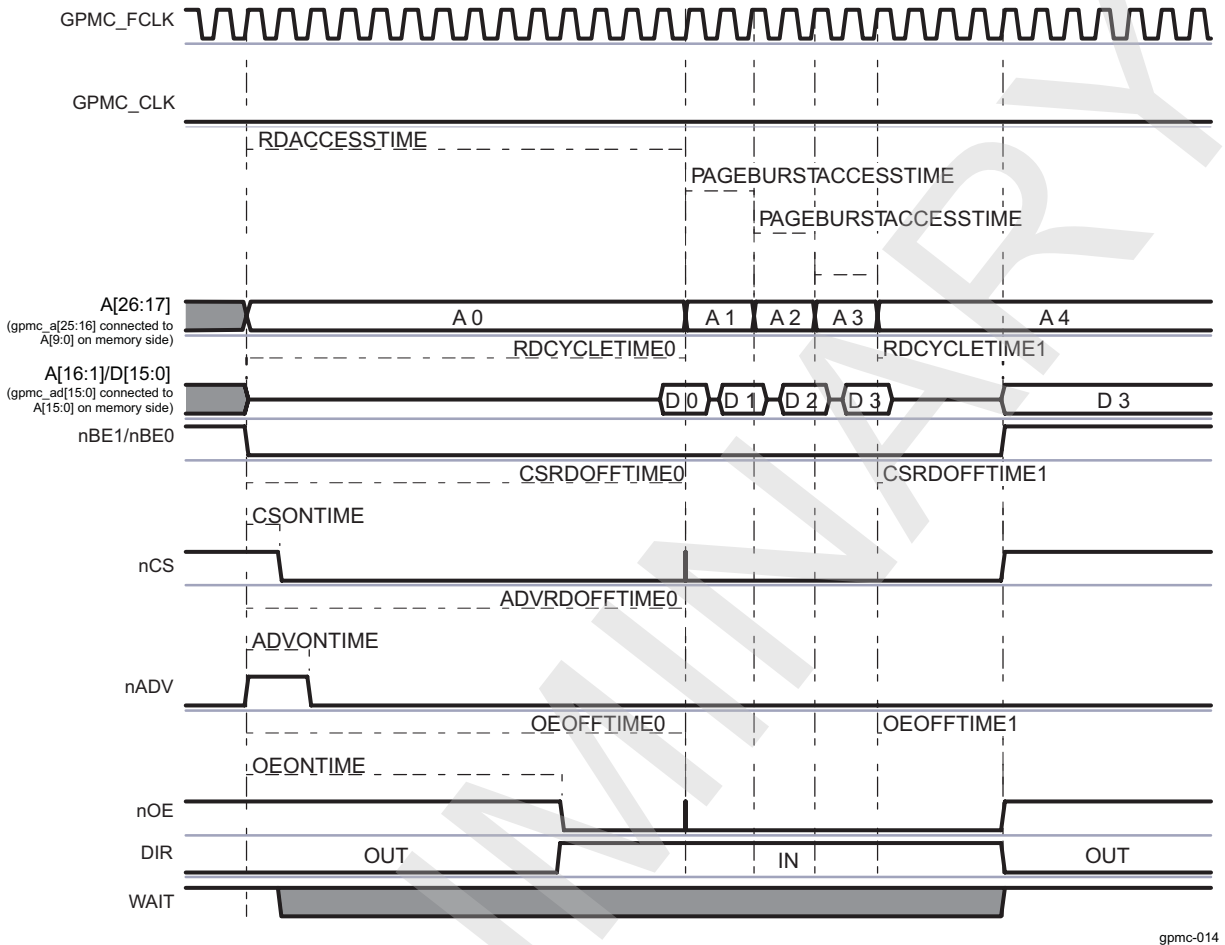
Set `GPMC_CONFIG[1] LIMITEDADDRESS` to 0x1 (A26-A11 are not modified during an external memory access).

nCS, nADV, nWE and DIR signals are controlled in the same way as address/data multiplexed accesses, see [Table 15-247](#).

#### 15.4.4.11.3.3 Asynchronous Multiple (Page Mode) Read Operation on a Nonmultiplexed Device

[Figure 15-75](#) shows an asynchronous multiple read operation on a Nonmultiplexed Device, in which two word32 host read accesses to the GPMC are split into one multiple (page mode of 4 word16) read access to the attached device.

Figure 15-75. Asynchronous Multiple (Page Mode) Read



**NOTE:** The WAIT signal is active low.

nCS, nADV, nOE and DIR signals are controlled in the same way as address/data multiplexed accesses, see [Table 15-247](#).

When RDACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

Read data is latched at [GPMC\\_CONFIG5\\_i\[20:16\]](#) RDACCESSTIME completion time (i = 0 to 3). The end of the access is defined by the [GPMC\\_CONFIG5\\_i\[4:0\]](#) RDCYCLETIME parameter.

Set [GPMC\\_CONFIG\[1\]](#) LIMITEDADDRESS to 0x1 (A26-A11 are not modified during an external memory access).

During consecutive accesses, the GPMC increments the address after each data read completes.

Delay between successive read data in the page is controlled by the [GPMC\\_CONFIG5\\_i\[27:24\]](#) PAGEBURSTACCESSTIME parameter. Depending on the device page length, the GPMC can control device page crossing during a burst request and insert initial RDACCESSTIME latency. Note that page crossing is only possible with a new burst access, meaning a new initial access phase is initiated.

Total access time RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time starting from the nCS deassertion.

- The read cycle time is defined in the [GPMC\\_CONFIG5\\_i\[4:0\]](#) RDCYCLETIME field.
- In [Figure 15-75](#), the RDCYCLETIME programmed value equals RDCYCLETIME0 (before paged accesses) + RDCYCLETIME1 (after paged accesses).

#### 15.4.4.11.3.4 Synchronous Operations on a Nonmultiplexed Device

All information for this section is equivalent to similar operations for address/data- or AAD-multiplexed accesses. The only difference resides in the address phase. See [Section 15.4.5.3, GPMC Configuration in NOR Mode](#).

#### 15.4.4.11.4 Page and Burst Support

Each chip-select can be configured to process system single or burst requests into successive single accesses or asynchronous page/synchronous burst accesses, with appropriate access size adaptation.

Depending on the external device page or burst capability, read and write accesses can be independently configured through the GPMC. The [GPMC\\_CONFIG1\\_i\[30\]](#) READMULTIPLE and [GPMC\\_CONFIG1\\_i\[28\]](#) WRITMULTIPLE bits ( $i = 0$  to 3) are associated with the READTYPE and WRITETYPE parameters.

---

**NOTE:**

- Asynchronous write page mode is not supported.
  - 8-bit wide device support is limited to nonburstable devices (READMULTIPLE and WRITEMULTIPLE are ignored).
  - Not applicable to NAND device interfacing.
- 

#### 15.4.4.11.5 System Burst Versus External Device Burst Support

The device system can issue the following requests to the GPMC:

- Byte, 16-bit word, 32-bit word requests (byte enable controlled). This is always a single request from the interconnect point of view.
- Incrementing fixed-length bursts of two words, four words, and eight words
- Wrapped (critical word access first) fixed-length burst of two, four, or eight words

To process a system request with the optimal protocol, the READMULTIPLE (and READTYPE) and WRITEMULTIPLE (and WRITETYPE) parameters must be set according to the burstable capability (synchronous or asynchronous) of the attached device.

The GPMC access engine issues only fixed-length burst. The maximum length that can be issued is defined per CS by the [GPMC\\_CONFIG1\\_i\[24:23\]](#) ATTACHEDDEVICEPAGELENGTH field ( $i = 0$  to 3). When the ATTACHEDDEVICEPAGELENGTH value is less than the system burst request length (including the appropriate access size adaptation according to the device width), the GPMC splits the system burst request into multiple bursts. Within the specified 4-, 8-, or 16-word value, the ATTACHEDDEVICEPAGELENGTH field value must correspond to the maximum-length burst supported by the memory device configured in fixed-length burst mode (as opposed to continuous burst mode).

To get optimal performance from memory devices that natively support 16 Word16-length-wrapping burst capability (critical word access first), the ATTACHEDDEVICEPAGELENGTH parameter must be set to 16 words and the [GPMC\\_CONFIG1\\_i\[31\]](#) WRAPBURST bit ( $i = 0$  to 3) must be set to 1. Similarly DEVICESPAGELENGTH is set to 4 and 8 for memories supporting respectively 4 and 8 Word16-length-wrapping burst.

When the memory device does not offer (or is not configured to offer) native 16 Word16-length-wrapping burst, the WRAPBURST parameter must be cleared, and the GPMC access engine emulates the wrapping burst by issuing the appropriate burst sequences according to the ATTACHEDDEVICEPAGELENGTH value.

When the memory device does not support native-wrapping burst, there is usually no difference in behavior between a fixed burst length mode and a continuous burst mode configuration (except for a potential power increase from a memory-speculative data prefetch in a continuous burst read). However, even though continuous burst mode is compatible with GPMC behavior, because the GPMC access engine issues only fixed-length burst and does not benefit from continuous burst mode, it is best to configure the memory device in fixed-length burst mode.

The memory device maximum-length burst (configured in fixed-length burst wrap or nonwrap mode) usually corresponds to the memory device data buffer size. Memory devices with a minimum of 16 half-word buffers are the most appropriate (especially with wrap support), but memory devices with smaller buffer size (4 or 8) are also supported, assuming that the `GPMC_CONFIG1_i[24:23]` `ATTACHEDDEVICEPAGELENGTH` field is set accordingly to 4 or 8 words.

The device system issues only requests with addresses or starting addresses for nonwrapping burst requests; that is, the request size boundary is aligned. In case of an eight-word-wrapping burst, the wrapping address always occurs on the eight-words boundary. As a consequence, all words requested must be available from the memory data buffer when the buffer size is equal to or greater than the `ATTACHEDDEVICEPAGELENGTH` value. This usually means that data can be read from or written to the buffer at a constant rate (number of cycles between data) without wait states between data accesses. If the memory does not behave this way (nonzero wait state burstable memory), wait-pin monitoring must be enabled to dynamically control data-access completion within the burst.

---

**NOTE:** When the system burst request length is less than the `ATTACHEDDEVICEPAGELENGTH` value, the GPMC proceeds with the required accesses.

---

#### 15.4.4.12 pSRAM Access Specificities

pSRAM devices are SRAM-pin-compatible low-power memories that contain a self-refreshed DRAM memory array. The `GPMC_CONFIG1_i[11:10]` `DEVICETYPE` field ( $i = 0$  to 3) shall be set to 0b00.

The pSRAM devices uses the NOR protocol. It support the following operations:

- Asynchronous single read
- Asynchronous page read
- Asynchronous single write
- Synchronous single read and write
- Synchronous burst read
- Synchronous burst write (not supported by NOR Flash memory)

pSRAM devices must be powered up and initialized in a predefined manner according to the specifications of the attached device.

pSRAM devices can be programmed to use either mode: fixed or variable latency. pSRAM devices can either automatically schedule autorefresh operations, which force the GPMC to use its `WAIT` signal capability when read or write operations occur during an internal self-refresh operation, or pSRAM devices automatically include the autorefresh operation in the access time. These devices do not require additional `WAIT` signal capability or a minimum `nCS` high pulse width between consecutive accesses to ensure that the correct internal refresh operation is scheduled.

#### 15.4.4.13 NAND Access Description

NAND (8-bit and 16-bit) memory devices using a standard NAND asynchronous address/data-multiplexing scheme can be supported on any chip-select with the appropriate asynchronous configuration settings.

As for any other type of memory compatible with the GPMC interface, accesses to a chip-select allocated to a NAND device can be interleaved with accesses to chip-selects allocated to other external devices.

This interleaved capability limits the system to *chip enable don't care* NAND devices since the chip-select allocated to the NAND device has to be de-asserted if accesses to other chip-selects are requested.

##### 15.4.4.13.1 NAND Memory Device in Byte or 16-bit Word Stream Mode

NAND devices require correct command and address programming before data array read or write accesses. The GPMC does not include specific hardware to translate a random address system request into a NAND-specific multiphase access. In that sense, GPMC NAND support, as opposed to random memory-map device support, is data-stream-oriented (byte or 16-bit word).

The GPMC NAND programming model relies on a software driver for address and command formatting with the correct data address pointer value according to the block and page structure. Because of NAND structure and protocol interface diversity, the GPMC does not support automatic command and address phase programming, and software drivers must access the NAND device ID to ensure that correct command and address formatting are used for the identified device.

NAND device data read and write accesses are achieved through an asynchronous read or write access. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Any chip-select region can be qualified as a NAND region to constrain the nADV/ALE signal as Address Latch Enable (ALE active high, default state value at low) during address program access, and the nBE0/CLE signal as Command Latch Enable (CLE active high, default state value at low) during command program access. GPMC address lines are not used (the previous value is not changed) during NAND access.

#### 15.4.4.13.1.1 Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode

The [GPMC\\_CONFIG7\\_i](#) register (where  $i = 0$  to 3) associated with a NAND device region interfaced in byte or word stream mode can be initialized with a minimum size of 16 Mbytes, because any address location in the chip-select memory region can be used to access a NAND data array. The NAND Flash protocol specifies an address sequence where address bits are passed through the data bus in a series of write accesses with the ALE pin asserted. After this address phase, all operations are streamed and the system requests address is irrelevant.

#### CAUTION

To allow correct command, address, and data-access controls, the [GPMC\\_CONFIG1\\_i](#) register associated with a NAND device region must be initialized in asynchronous read and write modes with the parameters shown in [Table 15-223](#). Failure to comply with these settings corrupts the NAND interface protocol.

**Table 15-223. Chip-Select Configuration for NAND Interfacing**

Bit Field	Register	Value	Comments
WRAPBURST	<a href="#">GPMC_CONFIG1_i[31]</a> <sup>(1)</sup>	0	No wrap
READMULTIPLE	<a href="#">GPMC_CONFIG1_i[30]</a>	0	Single access
READTYPE	<a href="#">GPMC_CONFIG1_i[29]</a>	0	Asynchronous mode
WRITEMULTIPLE	<a href="#">GPMC_CONFIG1_i[28]</a>	0	Single access
WRITETYPE	<a href="#">GPMC_CONFIG1_i[27]</a>	0	Asynchronous mode
CLKACTIVATIONTIME	<a href="#">GPMC_CONFIG1_i[26:25]</a>	0b00	
ATTACHEDDEVICEPAGELENGTH	<a href="#">GPMC_CONFIG1_i[24:23]</a>	Don't care	Single-access mode
WAITREADMONITORING	<a href="#">GPMC_CONFIG1_i[22]</a>	0	Wait not monitored by GPMC access engine
WAITWRITEMONITORING	<a href="#">GPMC_CONFIG1_i[21]</a>	0	Wait not monitored by GPMC access engine
WAITMONITORINGTIME	<a href="#">GPMC_CONFIG1_i[19:18]</a>	Don't care	Wait not monitored by GPMC access engine
WAITPINSELECT	<a href="#">GPMC_CONFIG1_i[17:16]</a>		Select which wait is monitored by edge detectors
DEVICESIZE	<a href="#">GPMC_CONFIG1_i[13:12]</a>	0b00 or 0b01	8- or 16-bit interface
DEVICETYPE	<a href="#">GPMC_CONFIG1_i[11:10]</a>	0b10	NAND device in stream mode
MUXADDDATA	<a href="#">GPMC_CONFIG1_i[9:8]</a>	0b00	Nonmultiplexed mode

<sup>(1)</sup>  $i = 0$  to 3



**Table 15-223. Chip-Select Configuration for NAND Interfacing (continued)**

Bit Field	Register	Value	Comments
TIMEPARAGRANULARITY	<a href="#">GPMC_CONFIG1_i[4]</a>	0	Timing achieved with best GPMC clock granularity
GPMCFCLKDIVIDER	<a href="#">GPMC_CONFIG1_i[1:0]</a>	Don't care	Asynchronous mode

The [GPMC\\_CONFIG1\\_i](#) to [GPMC\\_CONFIG4\\_i](#) register (where  $i = 0$  to  $3$ ) associated with a NAND device region must be initialized with the correct control-signal timing value according to the NAND device timing parameters.

#### 15.4.4.13.1.2 NAND Device Command and Address Phase Control

NAND devices require multiple address programming phases. The MPU software driver is responsible for issuing the correct number of command and address program accesses, according to the device command set and the device address-mapping scheme.

NAND device-command and address-phase programming is achieved through write requests to the [GPMC\\_NAND\\_COMMAND\\_i](#) and [GPMC\\_NAND\\_ADDRESS\\_i](#) register locations ( $i = 0$  to  $3$ ) with the correct command and address values. These locations are mapped in the associated chip-select register region. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Command and address values are not latched during the access and cannot be read back at the register location.

- Only write accesses must be issued to these locations, but the GPMC does not discard any read access. Accessing a NAND device with nOE and CLE or ALE asserted (read access) can produce undefined results.
- Write accesses to the [GPMC\\_NAND\\_COMMAND\\_i](#) register location and to the [GPMC\\_NAND\\_ADDRESS\\_i](#) register location must be posted for faster operations ( $i = 0$  to  $3$ ). The [GPMC\\_CONFIG\[0\]](#) NANDFORCEPOSTEDWRITE bit enables write accesses to these locations as posted, even if they are defined as nonposted.

A write buffer is used to store write transaction information before the external device is accessed:

- Up to eight consecutive posted write accesses can be accepted and stored in the write buffer.
- For nonposted write, the pipeline is one deep.
- An [GPMC\\_STATUS\[0\]](#) EMPTYWRITEBUFFERSTATUS bit stores the empty status of the write buffer.

[GPMC\\_NAND\\_COMMAND\\_i](#) and [GPMC\\_NAND\\_ADDRESS\\_i](#) ( $i = 0$  to  $3$ ) are 32-bit word locations, which means any 32-bit word or 16-bit word access is split into 4- or 2-byte accesses if an 8-bit wide NAND device is attached. For multiple-command phase or multiple-address phase, the software driver can use 32-bit word or 16-bit word access to these registers, but it must account for the splitting and little-endian ordering scheme. When only one byte command or address phase is required, only byte write access to [GPMC\\_NAND\\_COMMAND\\_i](#) and [GPMC\\_NAND\\_ADDRESS\\_i](#) can be used, and any of the four byte locations of the registers are valid.

The same applies to a [GPMC\\_NAND\\_COMMAND\\_i](#) and a [GPMC\\_NAND\\_ADDRESS\\_i](#) ( $i = 0$  to  $3$ ) 32-bit word write access to a 16-bit wide NAND device (split into two 16-bit word accesses). In the case of a 16-bit word write access, the MSByte of the 16-bit word value must be set according to the NAND device requirement (usually 0). Either 16-bit word location or any one of the four byte locations of the registers is valid.

#### 15.4.4.13.1.3 Command Latch Cycle

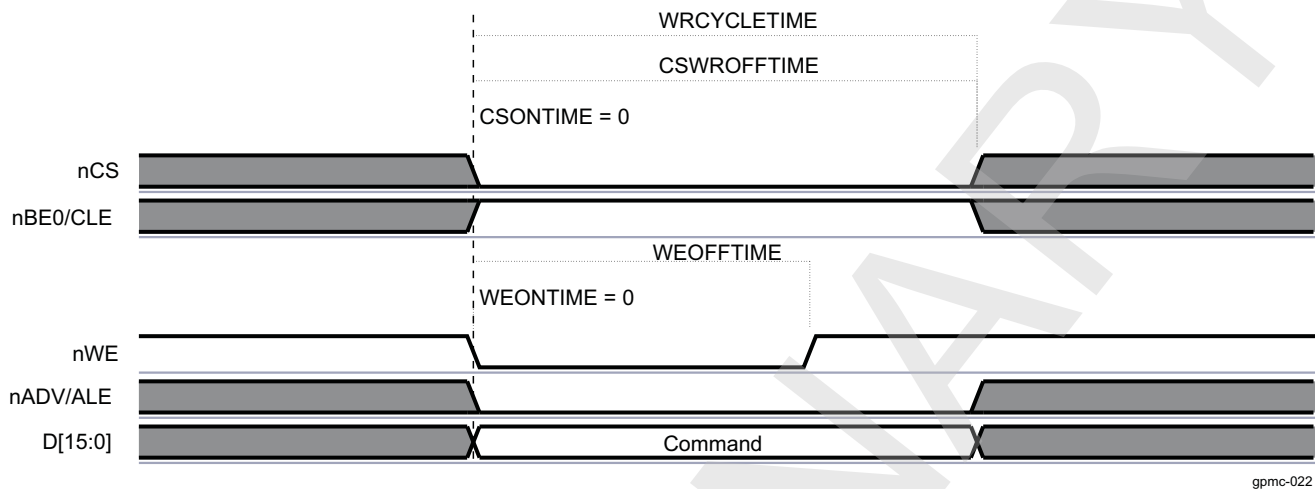
Writing data at the [GPMC\\_NAND\\_COMMAND\\_i](#) location ( $i = 0$  to  $3$ ) places the data as the NAND command value on the bus, using a regular asynchronous write access.

- nCE is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- CLE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- nWE is controlled by the WEONTIME and WEOFFTIME timing parameters.

- ALE and nRE (nOE) are maintained inactive.

Figure 15-76 shows the NAND command latch cycle.

**Figure 15-76. NAND Command Latch Cycle**



**NOTE:** CLE is shared with the nBE0 output signal and has an inverted polarity from BE0. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, nBE0 (also nBE1) must not toggle, because it is shared with CLE.

NAND Flash memories do not use byte enable signals at all.

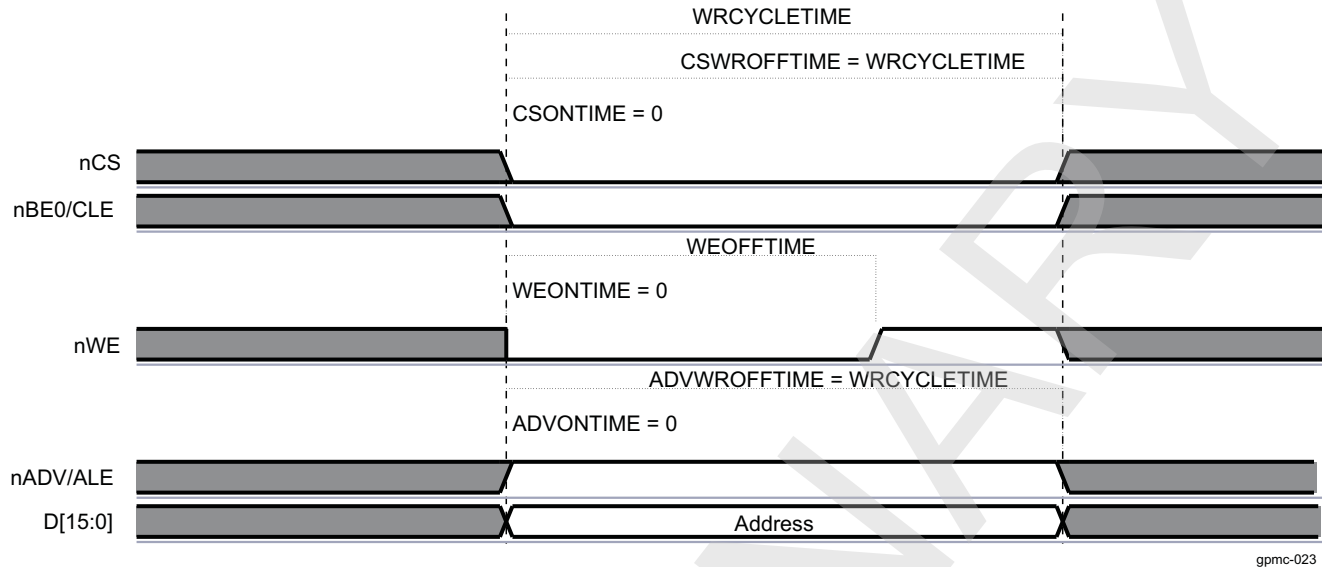
#### 15.4.4.13.1.4 Address Latch Cycle

Writing data at the [GPMC\\_NAND\\_ADDRESS\\_i](#) location ( $i = 0$  to 3) places the data as the NAND partial address value on the bus, using a regular asynchronous write access.

- nCS is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- ALE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- nWE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- CLE and nRE (nOE) are maintained inactive.

Figure 15-77 shows the NAND address latch cycle.

Figure 15-77. NAND Address Latch Cycle



**NOTE:** ALE is shared with the nADV output signal and has an inverted polarity from ADV. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, ALE is kept stable.

#### 15.4.4.13.1.5 NAND Device Data Read and Write Phase Control in Stream Mode

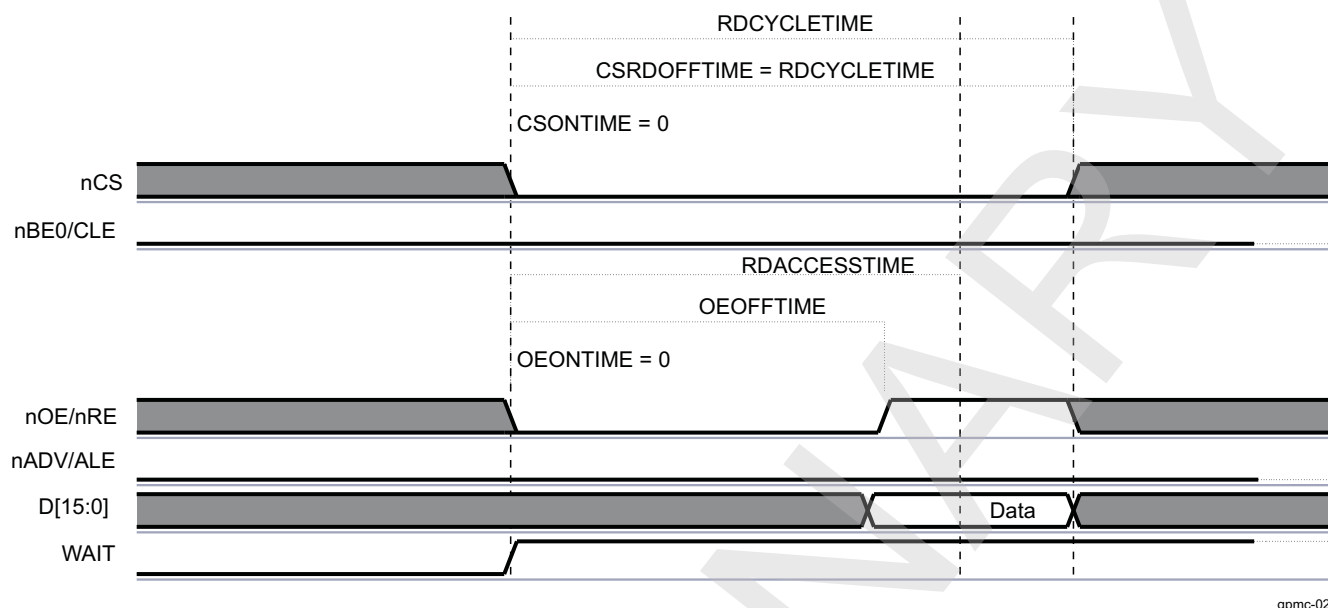
NAND device data read and write accesses are achieved through a read or write request to the chip-select-associated memory region at any address location in the region or through a read or write request to the `GPMC_NAND_DATA_i` location ( $i = 0$  to 3) mapped in the chip-select-associated control register region. `GPMC_NAND_DATA_i` is not a true register, but an address location to enable nRE or nWE signal control. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Reading data from the `GPMC_NAND_DATA_i` location or from any location in the associated chip-select memory region activates an asynchronous read access.

- nCS is controlled by the CSONTIME and CSRDOFFTIME timing parameters.
- nRE is controlled by the OEONTIME and OEOFFTIME timing parameters.
- To take advantage of nRE high-to-data invalid minimum timing value, the RDACCESSTIME can be set so that data are effectively captured after nRE deassertion. This allows optimization of NAND read access cycle time completion. For optimal timing parameter settings, see the NAND device and the device IC timing parameters.

ALE, CLE, and nWE are maintained inactive.

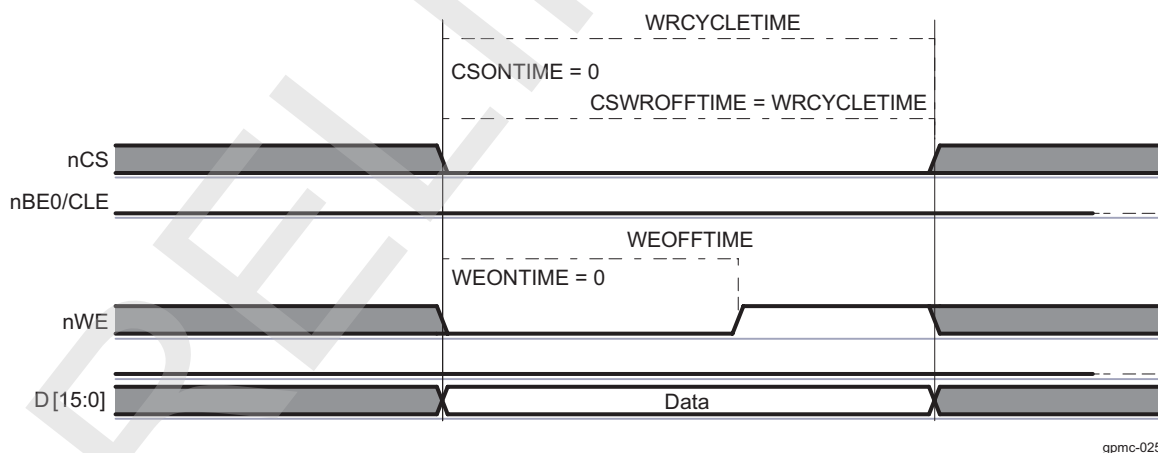
Figure 15-78 shows the NAND data read cycle.

**Figure 15-78. NAND Data Read Cycle**

Writing data to the `GPMC_NAND_DATA_i` location or to any location in the associated chip-select memory region activates an asynchronous write access.

- nCS is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- nWE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE, CLE, and nRE (nOE) are maintained inactive.

Figure 15-79 shows the NAND data write cycle.

**Figure 15-79. NAND Data Write Cycle**

#### 15.4.4.13.1.6 NAND Device General Chip-Select Timing Control Requirement

For most NAND devices, read data access time is dominated by nCS-to-data-valid timing and has faster nRE-to-data-valid timing. Successive accesses with nCS deassertions between accesses are affected by this timing constraint. Because accesses to a NAND device can be interleaved with other chip-select accesses, there is no certainty that nCS always stays low between two accesses to the same chip-select. Moreover, an nCS deassertion time between the same chip-select NAND accesses is likely to be required as follows: the nCS deassertion requires programming CYCLETIME and RDACCESSTIME according to the nCS-to-data-valid critical timing.

To get full performance from NAND read and write accesses, the prefetch engine can dynamically reduce

RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, WRACCESSTIME, CSRDOFFTIME, CSWROFFTIME, ADVRDOFFTIME, ADVWROFFTIME, OEOFFTIME, and WEOFFTIME on back-to-back NAND accesses (to the same memory) and suppress the minimum nCS high pulse width between accesses. For more information about optimal prefetch engine access, see [Section 15.4.4.13.4, Prefetch and Write-Posting Engine](#).

Some NAND devices require minimum write-to-read idle time, especially for device-status read accesses following status-read command programming (write access). If such write-to-read transactions are used, a minimum nCS high pulse width must be set. For this, CYCLE2CYCLESAMEECSEN and CYCLE2CYCLEDELAY must be set according to the appropriate timing requirement to prevent any timing violation.

NAND devices usually have an important nRE high to data bus in tristate mode. This requires a bus turnaround setting (BUSTURNAROUND = 1), so that the next access to a different chip-select is delayed until the BUSTURNAROUND delay completes. Back-to-back NAND read accesses to the same NAND Flash are not affected by the programmed bus turnaround delay.

### **15.4.4.13.1.7 Read and Write Access Size Adaptation**

#### **15.4.4.13.1.7.1 8-bit wide NAND Device**

Host 16-bit word and 32-bit word read and write access requests to a chip-select associated with an 8-bit wide NAND device are split into successive read and write byte accesses to the NAND memory device. Byte access is ordered according to little-endian organization. A NAND 8-bit wide device must be interfaced on the D0D7 interface bus lane. GPMC data accesses are justified on this bus lane when the chip-select is associated with an 8-bit wide NAND device.

#### **15.4.4.13.1.7.2 16-bit wide NAND Device**

Host 32-bit word read and write access requests to a chip-select associated with a 16-bit wide NAND device are split into successive read and write 16-bit word accesses to the NAND memory device. 16-bit word access is ordered according to little-endian organization.

Host byte read and write access requests to a 16-bit wide NAND device are completed as 16-bit accesses on the device itself, because there is no byte-addressing capability on 16-bit wide NAND devices. This means that the NAND device address pointer is incremented on a 16-bit word basis and not on a byte basis. For a read access, only the requested byte is given back to the host, but the remaining byte is not stored or saved by the GPMC, and the next byte or 16-bit word read access gets the next 16-bit word NAND location. For a write access, the invalid byte part of the 16-bit word is driven to FF, and the next byte or 16-bit word write access programs the next 16-bit word NAND location.

Generally, byte access to a 16-bit wide NAND device should be avoided, especially when ECC calculation is enabled. 8-bit or 16-bit ECC-based computations are corrupted by a byte read to a 16-bit wide NAND device, because the nonrequested byte is considered invalid on a read access (not captured on the external data bus; FF is fed to the ECC engine) and is set to FF on a write access.

Host requests (read/write) issued in the chip-select memory region are translated in successive single or split accesses (read/write) to the attached device. Therefore, incrementing 32-bit burst requests are translated in multiple 32-bit sequential accesses following the access adaptation of the 32-bit to 8- or 16-bit device.

### **15.4.4.13.2 NAND Device-Ready Pin**

The NAND memory device provides a ready pin to indicate data availability after a block/page opening and to indicate that data programming is complete. The ready pin can be connected to one of the WAIT GPMC input pins; data read accesses must not be tried when the ready pin is sampled inactive (device is not ready) even if the associated chip-select WAITREADMONITORING bit field is set. The duration of the NAND device busy state after the block/page opening is so long (up to 50 micro second) that accesses occurring when the ready pin is sampled inactive can stall GPMC access and eventually cause a system time-out.

**NOTE:** If a read access to a NAND flash is done using the wait monitoring mode, the device is blocked during a page opening, and so is the GPMC. If the correct settings are used, other chip-selects can be used while the memory processes the page opening command.

To avoid a time-out caused by a block/page opening delay in NAND flash, disable the wait pin monitoring for read and write accesses (that is, set the [GPMC\\_CONFIG1\\_i\[21\]](#) WAITWRITEMONITORING and [GPMC\\_CONFIG1\\_i\[22\]](#) WAITREADMONITORING bits to 0, where  $i = 0$  to 3), and use one of the following methods instead:

- Use software to poll the WAITnSTATUS bit ( $n = 0$  to 3) of the [GPMC\\_STATUS](#) register.
- Configure an interrupt that is generated on the WAIT signal change (through the [GPMC\\_IRQENABLE](#) register bits[11:8]).

Even if the READWAITMONITORING bit is not set, the external memory nR/B pin status is captured in the programmed WAIT bit in the [GPMC\\_STATUS](#) register.

The READWAITMONITORING bit method must be used for other memories than NAND flash, if they require the use of a WAIT signal.

#### 15.4.4.13.2.1 Ready Pin Monitored by Software Polling

The ready signal state can be monitored through the [GPMC\\_STATUS](#) WAITxSTATUS bit ( $x = 0$  or 1). The software must monitor the ready pin only when the signal is declared valid. Refer to the NAND device timing parameters to set the correct software temporization to monitor ready only after the invalid window is complete from the last read command written to the NAND device.

#### 15.4.4.13.2.2 Ready Pin Monitored by Hardware Interrupt

Each gpmc\_wait input pin can generate an interrupt when a wait-to-no-wait transition is detected. Depending on whether the [GPMC\\_CONFIG](#) WAITxPINPOLARITY bits ( $x = 0$  or 1) is active low or active high, the wait-to-no-wait transition is a low-to-high external WAIT signal transition or a high-to-low external WAIT signal transition, respectively.

The wait transition pin detector must be cleared before any transition detection. This is done by writing 1 to the WAITxEDGEDETECTIONSTATUS bit ( $x = 0$  or 3) of the [GPMC\\_IRQSTATUS](#) register according to the gpmc\_wait pin used for the NAND device-ready signal monitoring. To detect a wait-to-no-wait transition, the transition detector requires a wait active time detection of a minimum of two GPMC\_FCLK cycles. Software must incorporate precautions to clear the wait transition pin detector before wait (busy) time completes.

A wait-to-no-wait transition detection can issue a GPMC interrupt if the WAITxEDGEDETECTIONENABLE bit in the [GPMC\\_IRQENABLE](#) register is set and if the WAITxEDGEDETECTIONSTATUS bit field in the [GPMC\\_IRQSTATUS](#) register is set.

The WAITMONITORINGTIME field does not affect wait-to-no-wait transition time detection.

It is also possible to poll the WAITxEDGEDETECTIONSTATUS bit field in the [GPMC\\_IRQSTATUS](#) register according to the gpmc\_wait pin used for the NAND device ready signal monitoring.

#### 15.4.4.13.3 ECC Calculator

The General Purpose Memory Controller includes an Error Code Correction (ECC) calculator circuitry that enables on the fly ECC calculation during data read or data program (that is, write) operations. The page size supported by the ECC calculator in one calculation/context is 512 bytes.

The user can choose from two different algorithms with different error correction capabilities through the [GPMC\\_ECC\\_CONFIG\[16\]](#) ECCALGORITHM bit:

1. Hamming code for 1-bit error code correction on 8- or 16-bit NAND Flash organized with page size greater than 512 bytes
2. BCH (Bose-Chaudhuri-Hocquenghem) code for 4- to 16-bit error correction

The GPMC does not directly handle the error code correction itself. During writes, the GPMC computes parity bits. During reads, the GPMC provides enough information for the processor to correct errors without reading the data buffer all over again.

The Hamming code ECC is based on a 2-dimensional (row and column) bit parity accumulation. This parity accumulation is either accomplished on the programmed number of bytes or 16-bit words read from the memory device, or written to the memory device in stream mode.

Because the ECC engine includes only one accumulation context, it can be allocated to only one chip-select at a time through the [GPMC\\_ECC\\_CONFIG\[3:1\]](#) ECCCS bit field. Even if two CS use different ECC algorithms, one the Hamming code and the other a BCH code, they must define separate ECC contexts because some of the ECC registers are common to all types of algorithms.

#### **15.4.4.13.3.1 Hamming Code**

All references to Error Code Correction (ECC) in this subsection refer to the 1-bit error correction Hamming code.

The ECC is based on a two-dimensional (row and column) bit parity accumulation known as Hamming Code. The parity accumulation is done for a programmed number of bytes or 16-bit word read from the memory device or written to the memory device in stream mode.

There is no automatic error detection or correction, and it is the software NAND driver responsibility to read the multiple ECC calculation results, compare them to the expected code value, and take the appropriate corrective actions according to the error handling strategy (ECC storage in spare byte, error correction on read, block invalidation).

The ECC engine includes a single accumulation context. It can be allocated to a single designated chip-select at a time and parallel computations on different chip-selects are not possible. Since it is allocated to a single chip-select, the ECC computation is not affected by interleaved GPMC accesses to other chip-selects and devices. The ECC accumulation is sequentially processed in the order of data read from or written to the memory on the designated chip-select. The ECC engine does not differentiate read accesses from write accesses and does not differentiate data from command or status information. It is the software responsibility to make sure only relevant data are passed to the NAND flash memory while the ECC computation engine is active.

The starting NAND page location must be programmed first, followed by an ECC accumulation context reset with an ECC enabling, if required. The NAND device accesses discussed in the following sections must be limited to data read or write until the specified number of ECC calculations is completed.

##### **15.4.4.13.3.1.1 ECC Result Register and ECC Computation Accumulation Size**

The GPMC includes up to nine ECC result registers ([GPMC\\_ECCj\\_RESULT](#),  $j = 1$  to 9) to store ECC computation results when the specified number of bytes or 16-bit words has been computed.

The ECC result registers are used sequentially; one ECC result is stored in one ECC result register on the list, the next ECC result is stored in the next ECC result register on the list, and so forth, until the last ECC computation. The [GPMC\\_ECCj\\_RESULT](#) register value is valid only when the programmed number of bytes or 16-bit words has been accumulated, which means that the same number of bytes or 16-bit words has been read from or written to the NAND device in sequence.

The [GPMC\\_ECC\\_CONTROL\[3:0\]](#) ECCPOINTER field must be set to the correct value to select the ECC result register to be used first in the list for the incoming ECC computation process. The ECCPointer can be read to determine which ECC register is used in the next ECC result storage for the ongoing ECC computation. The [GPMC\\_ECCj\\_RESULT](#) register value ( $j = 1$  to 9) can be considered valid when ECCPOINTER equals  $j + 1$ . When [GPMC\\_ECCj\\_RESULT](#) (where  $j = 9$ ) is updated, ECCPOINTER is frozen at 10, and ECC computing is stopped (ECCENABLE = 0).

The ECC accumulator must be reset before any ECC computation accumulation process. The [GPMC\\_ECC\\_CONTROL\[8\]](#) ECCCLEAR bit must be set to 1 (nonpersistent bit) to clear the accumulator and all ECC result registers.

For each ECC result (each [GPMC\\_ECCj\\_RESULT](#) register,  $j = 1$  to 9), the number of bytes or 16-bit words used for ECC computing accumulation can be selected from between two programmable values.

The ECCjRESULTSISE bits (j = 1 to 9) in the [GPMC\\_ECC\\_SIZE\\_CONFIG](#) register select which programmable size value (ECCSIZE0 or ECCSIZE1) must be used for this ECC result (stored in [GPMC\\_ECCj\\_RESULT](#)).

The ECCSIZE0 and ECCSIZE1 fields allow selection of the number of bytes or 16-bit words used for ECC computation accumulation. Any even values from 2 to 512 are allowed.

Flexibility in the number of ECCs computed and the number of bytes or 16-bit words used in the successive ECC computations enables different NAND page error-correction strategies. Usually based on 256 or 512 bytes and on 128 or 256 16-bit word, the number of ECC results required is a function of the NAND device page size. Specific ECC accumulation size can be used when computing the ECC on the NAND spare byte.

For example, with a 2 Kbyte data page 8-bit wide NAND device, eight ECCs accumulated on 256 bytes can be computed and added to one extra ECC computed on the 24 spare bytes area where the eight ECC results used for comparison and correction with the computed data page ECC are stored. The GPMC then provides nine [GPMC\\_ECCj\\_RESULT](#) registers (j= 1 to 9) to store the results. In this case, ECCSIZE0 is set to 256, and ECCSIZE1 is set to 24; the ECC[1:8]RESULTSISE bits are set to 0, and the ECC9RESULTSISE bit is set to 1.

#### 15.4.4.13.3.1.2 ECC Enabling

The [GPMC\\_ECC\\_CONFIG](#)[3:1] ECCCS field selects the allocated chip-select. The [GPMC\\_ECC\\_CONFIG](#)[0] ECCENABLE bit enables ECC computation on the next detected read or write access to the selected chip-select.

The ECCPOINTER, ECCCLEAR, ECCSIZE, ECCjRESULTSISE (where j = 1 to 9), ECC16B, and ECCCS fields must not be changed or cleared while an ECC computation is in progress.

The ECC accumulator and ECC result register must not be changed or cleared while an ECC computation is in progress.

[Table 15-224](#) describes the ECC enable settings.

**Table 15-224. ECC Enable Settings**

Bit Field	Register	Value	Comments
ECCCS	<a href="#">GPMC_ECC_CONFIG</a>	0-3	Selects the chip-select where ECC is computed
ECC16B	<a href="#">GPMC_ECC_CONFIG</a>	0/1	Selects column number for ECC calculation
ECCCLEAR	<a href="#">GPMC_ECC_CONTROL</a>	0-7	Clears all ECC result registers
ECCPOINTER	<a href="#">GPMC_ECC_CONTROL</a>	0-7	A write to this bit field selects the ECC result register where the first ECC computation is stored. Set to 1 by default.
ECCSIZE1	<a href="#">GPMC_ECC_SIZE_CONFIG</a>	0x00-0xFF	Defines ECCSIZE1
ECCSIZE0	<a href="#">GPMC_ECC_SIZE_CONFIG</a>	0x00-0xFF	Defines ECCSIZE0
ECCjRESULTSISE (j from 1 to 9)	<a href="#">GPMC_ECC_SIZE_CONFIG</a>	0/1	Selects the size of ECCn result register
ECCENABLE	<a href="#">GPMC_ECC_CONFIG</a>	1	Enables the ECC computation

#### 15.4.4.13.3.1.3 ECC Computation

The ECC algorithm is a multiple parity bit accumulation computed on the odd and even bit streams extracted from the byte or Word 16 streams. The parity accumulation is split into row and column accumulations, as shown in [Figure 15-80](#) and [Figure 15-81](#). The intermediate row and column parities are used to compute the upper level row and column parities. Only the final computation of each parity bit is used for ECC comparison and correction.

P1o = bit7 XOR bit5 XOR bit3 XOR bit1 on each byte of the data stream

P1e = bit6 XOR bit4 XOR bit2 XOR bit0 on each byte of the data stream

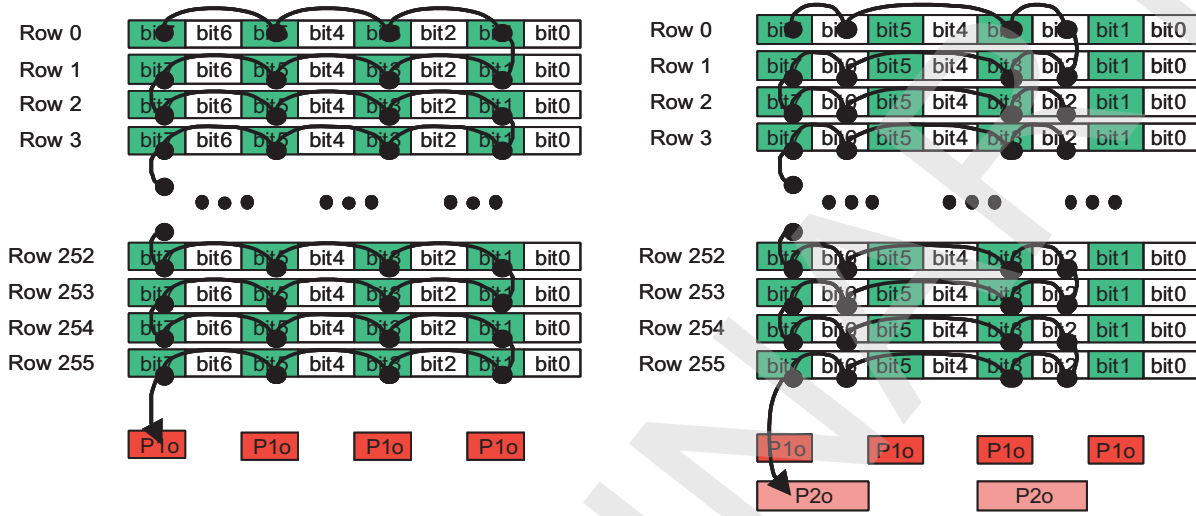
P2o = bit7 XOR bit6 XOR bit3 XOR bit2 on each byte of the data stream

P2e = bit5 XOR bit4 XOR bit1 XOR bit0 on each byte of the data stream



P4o = bit7 XOR bit6 XOR bit5 XOR bit4 on each byte of the data stream  
 P4e = bit3 XOR bit2 XOR bit1 XOR bit0 on each byte of the data stream  
 Each column parity bit is XORed with the previous accumulated value.

Figure 15-80. Hamming Code Accumulation Algorithm (1/2)



gpmc-026

For line parities, the bits of each new data are XORed together, and line parity bits are computed as described below:

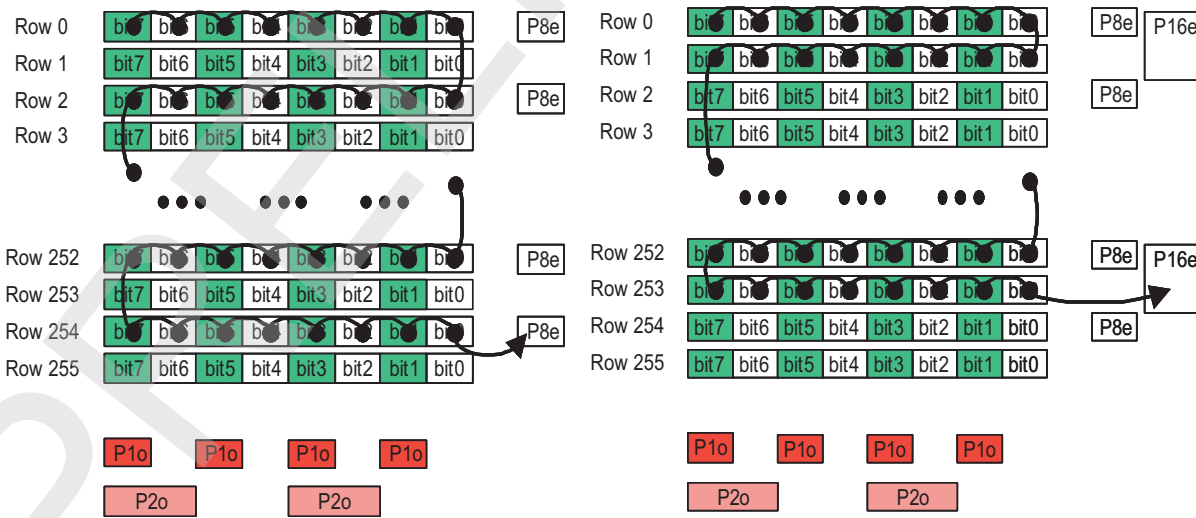
$$P8e = \text{row0 XOR row2 XOR row4 XOR ... XOR row254}$$

$$P8o = \text{row1 XOR row3 XOR row5 XOR ... XOR row255}$$

$$P16e = \text{row0 XOR row1 XOR row4 XOR row5 XOR ... XOR row252 XOR row 253}$$

$$P16o = \text{row2 XOR row3 XOR row6 XOR row7 XOR ... XOR row254 XOR row 255}$$

Figure 15-81. Hamming Code Accumulation Algorithm (2/2)



gpmc-027

Unused parity bits in the result registers are set to 0.

Figure 15-82 shows ECC computation for a 256-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and sixteen row parity bits (P8o-P16o-P32o--P1024o for odd parities, and P8e-P16e-P32e--P1024e for even parities).

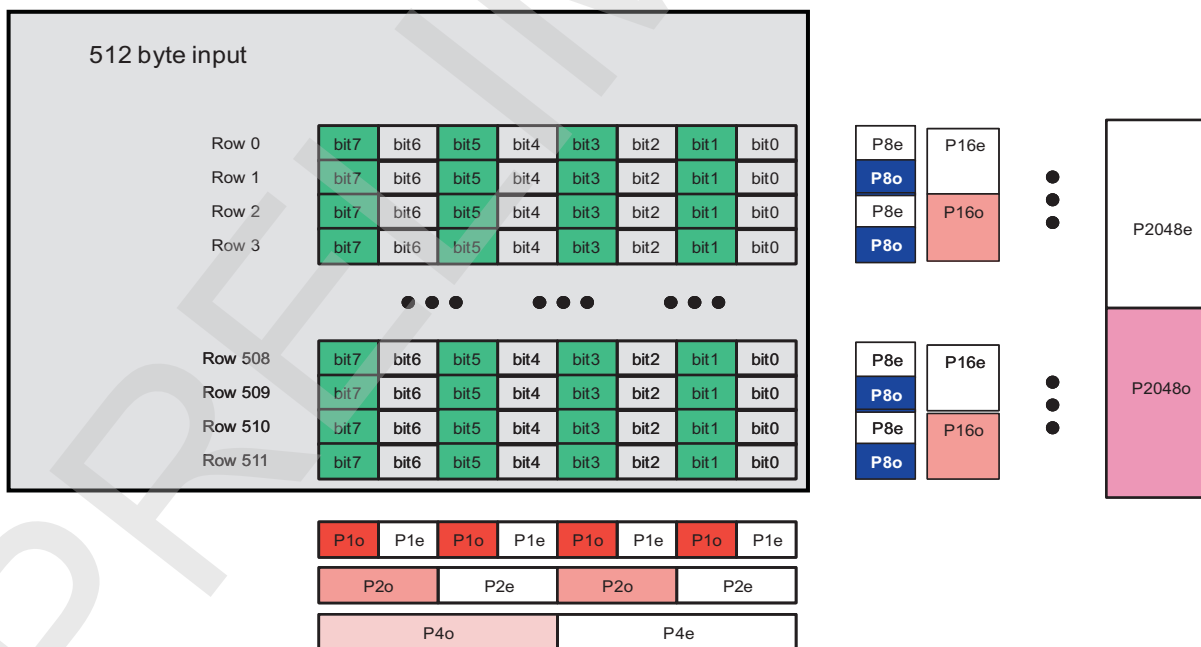
Figure 15-82. ECC Computation for a 256-Byte Data Stream (Read or Write)



gpmc-028

Figure 15-83 shows ECC computation for a 512-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and eighteen row parity bits (P8o-P16o-P32o--P1024o- - P2048o for odd parities, and P8e-P16e-P32e--P1024e- P2048e for even parities).

Figure 15-83. ECC Computation for a 512-Byte Data Stream (Read or Write)



gpmc-029

For a 2 Kbytes page, four 512 bytes ECC calculations plus one for the spare area are required. Results are stored in the [GPMC\\_ECCj\\_RESULT](#) registers (j = 1 to 9).

**15.4.4.13.3.1.4 ECC Comparison and Correction**

To detect an error, the computed ECC result must be XORed with the parity value stored in the spare area of the accessed page.

- If the result of this logical XOR is all 0s, no error is detected and the read data is correct.
- If every second bit in the parity result is a 1, one bit is corrupted and is located at bit address (P2048o, P1024o, P512o, P256o, P128o, P64o, P32o, P16o, P8o, P4o, P2o, P1o). The software must correct the corresponding bit.
- If only one bit in the parity result is 1, it is an ECC error and the read data is correct.

**15.4.4.13.3.1.5 ECC Calculation Based on 8-Bit Word**

The 8-bit based ECC computation is used for 8-bit wide NAND device interfacing.

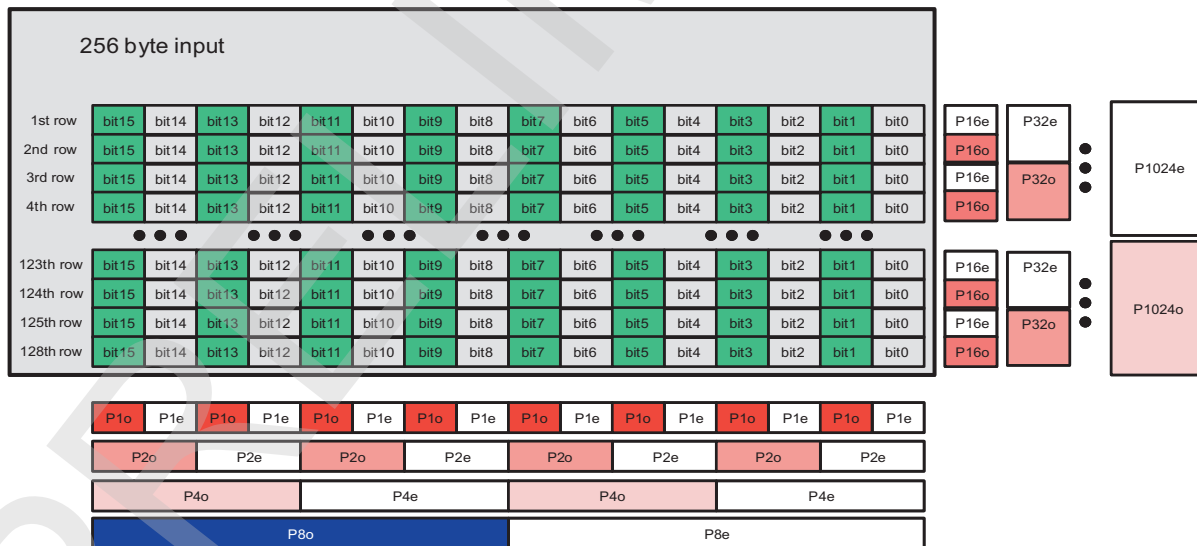
The 8-bit based ECC computation can be used for 16-bit wide NAND device interfacing to get backward compatibility on the error-handling strategy used with 8-bit wide NAND devices. In this case, the 16-bit wide data read from or written to the NAND device is fragmented into 2 bytes. According to little-endian access, the least significant bit (LSB) of the 16-bit wide data is ordered first in the byte stream used for 8-bit based ECC computation.

**15.4.4.13.3.1.6 ECC Calculation Based on 16-Bit Word**

ECC computation based on a 16-bit word is used for 16-bit wide NAND device interfacing. This ECC computation is not supported when interfacing an 8-bit wide NAND device, and the [GPMC\\_ECC\\_CONFIG\[7\]](#) ECC16B bit must be set to 0 when interfacing an 8-bit wide NAND device.

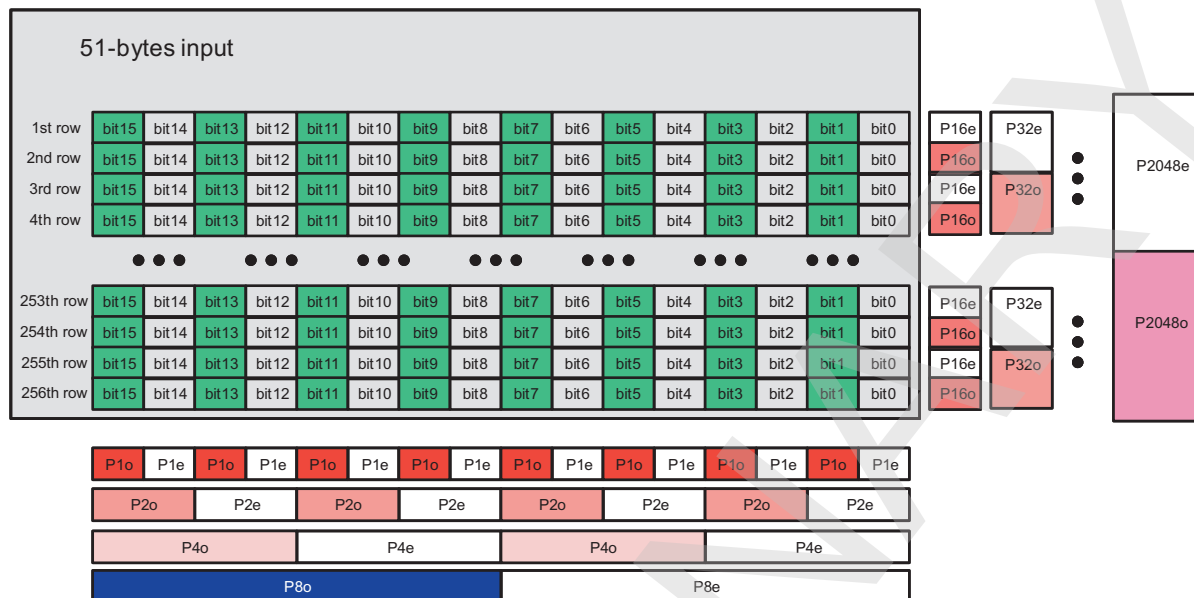
The parity computation based on 16-bit words affects the row and column parity mapping. The main difference is that the odd and even parity bits P8o and P8e are computed on rows for an 8-bit based ECC while there are computed on columns for a 16-bit based ECC. [Figure 15-84](#) and [Figure 15-85](#) show a 128 Word 16 ECC computation scheme and a 256 16-bit word ECC computation scheme.

**Figure 15-84. 128 Word16 ECC Computation**



gpmc-030

Figure 15-85. 256 Word16 ECC Computation



gpmc-031

#### 15.4.4.13.3.2 BCH Code (Bose-Chaudhuri-Hocquenghem)

All references to Error Code Correction (ECC) in this subsection refer to the 4- to 16-bit error correction BCH code.

##### 15.4.4.13.3.2.1 Requirements

- Read and write accesses to a NAND flash take place by whole pages, in a predetermined sequence: first the data byte page itself, then some spare bytes, including the BCH ECC (and other information). The NAND IC can cache a full page, including spares, for read and write accesses. Typical page write sequence:
  - Sequential write to NAND cache of main data + spare data, for a page. ECC is calculated on the fly. Calculated ECC may be inserted on the fly in the spares, or replaced by dummy accesses.
  - When the calculated ECC is replaced by dummy accesses, it must be written to the cache in a second, separate phase. The ECC module is disabled during that time.
  - NAND writes its cache line (page) to the array.
 Typical page read sequence:
  - Sequential read of a page. ECC is calculated on the fly.
  - ECC module buffers status determines the presence of errors.
- Accesses to several memories may be interleaved by the GPMC, but only one of those memories can be a NAND using the BCH engine at a time; in other words, only one BCH calculation (for example, for a single page) can be on-going at any time. Note also that the sequential nature of NAND accesses guarantees that the data is always written / read out in the same order. BCH-relevant accesses are selected by the GPMCs chip-select.
- Each page may hold up to 4 Kbytes of data, spare bytes not included. This means up to 8 x 512-byte BCH messages. Since all the data is written / read out first, followed by the BCH ECC, this means that the BCH engine must be able to hold 8 104-bit remainders or syndromes (or smaller, 52-bit ones) at the same time. The BCH module has the capacity to store all remainders internally. After the page start, an internal counter is used to detect the 512-byte sector boundaries. On those boundaries, the current remainder is stored and the divider reset for the next calculation. At the end of the page, the BCH module contains all remainders.
- NAND access cycles hold 8 or 16 bits of data each (1 or 2 bytes); Each NAND cycle takes at least 4 cycles of the GPMCs internal clock. This means the NAND flash timing parameters must define a

RDCYCLETIME and a WRCYCLETIME of at least 4 clock cycles after optimization when using the BCH calculator.

- The spare area is assumed to be large enough to hold the BCH ECC, that is, to have at least a message of 13 bytes available per 512-byte sector of data. The zone of unused spare area by the ECC may or may not be protected by the same ECC scheme, by extending the BCH message beyond 512 bytes (maximum codeword is 1023-byte long, ECC included, which leaves a lot of space to cover some spares bytes).

**15.4.4.13.3.2.2 Memory-Mapping of the BCH Codeword**

BCH encoding considers a block of data to protect as a polynomial message  $M(x)$ . In our standard case, 512 bytes of data (that is,  $2^{12}$  bits = 4096 bits) are seen as a polynomial of degree  $2^{12} - 1 = 4095$ , with parameters ranging from  $M_0$  to  $M_{4095}$ . For 512 bytes of data, 52 bits are required for 4-bit error correction, and 104 bits are required for 8-bit error correction and 207 bits are required for 16-bit error correction. The ECC is a remainder polynomial  $R(x)$  of degree 103 (or 51, depending on the selected mode). The complete codeword  $C(x)$  is the concatenation of  $M(x)$  and  $R(x)$  as shown in Table 15-225.

**Table 15-225. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits)**

	Message $M(x)$			ECC $R(x)$		
Bit number	$M_{4095}$	...	$M_0$	$R_{103}$	...	$R_0$

If the message is extended by the addition of spare bytes to be protected by the same ECC, the principle is still valid. For example, a 3-byte extension of the message gives a polynomial message  $M(x)$  of degree  $((512 + 3) * 8) - 1 = 4119$ , for a total of  $3+13 = 16$  spare bytes of spare, all protected as part of the same codeword.

The message and the ECC bits are manipulated and mapped in the GPMC byte-oriented system. The ECC bits are stored in `GPMC_BCH_RESULT0_i`, `GPMC_BCH_RESULT1_i`, `GPMC_BCH_RESULT2_i`, and `GPMC_BCH_RESULT3_i` (where  $i = 0$  to 3).

**15.4.4.13.3.2.2.1 Memory-Mapping of the Data Message**

The data message mapping shall follow the following rules:

- Bit endianness within a byte is little-endian, that is, the bytes LS bit is also the lowest-degree polynomial parameter: a byte  $b_7$ - $b_0$  (with  $b_0$  the LS bit) represents a segment of polynomial  $b_7 * x^{(7+i)} + b_6 * x^{(6+i)} + \dots + b_0 * x^i$
- The message is mapped in the NAND starting with the highest-order parameters, that is, in the lowest addresses of a NAND page.
- Byte endianness within the NANDs 16-bit words is big endian. This means that the same message mapped in 8- and 16-bit memories has the same content at the same byte address.

**NOTE:** The BCH module has no visibility over actual addresses. The most important point is the sequence of data word the BCH sees. However, the NAND page is always scanned incrementally in read and write accesses, and this produces the mapping patterns described below.

The following tables represent the mapping of the same 512-byte vector (typically a BCH message) in the NANDs memory space. Note that the byte 'address' is only an offset modulo 512 (0x200), since the same page may contain several contiguous 512-byte sectors (BCH blocks). The LSB and MSB are respectively the bits  $M_0$  and  $M(2^{12}-1)$  of the codeword mapping given above. In both cases the data vectors are aligned, that is, their boundaries coincide with the RAMs data word boundaries.

**Table 15-226. Aligned Message Byte Mapping in 8-bit NAND**

Byte offset	8-bit word
0x000	(msb) Byte 511 (0x1FF)
0x001	Byte 510 (0x1FE)
...	...

**Table 15-226. Aligned Message Byte Mapping in 8-bit NAND (continued)**

Byte offset	8-bit word
0x1FF	Byte 0 (0x0) (lsb)

**Table 15-227. Aligned Message Byte Mapping in 16-bit NAND**

Byte offset	16-bit words MSB	16-bit words LSB
0x000	Byte 510 (0x1FE)	(msb) Byte 511 (0x1FF)
0x002	Byte 508 (0x1FC)	Byte 509 (0x1FD)
...	...	...
0x1FE	Byte 0 (0x0)	(lsb) Byte 1 (0x1)

The following tables show the mapping in memory of arbitrarily-sized messages, starting on access (byte or 16-bit word) boundaries for more clarity. Note that message may actually start and stop on arbitrary nibbles. A nibble is a 4-bit entity. The unused nibbles are not discarded, and they can still be used by the BCH module, but as part of the next message section (for example, on another sectors ECC).

**Table 15-228. Aligned Nibble Mapping of Message in 8-bit NAND**

Byte offset	8-bit word	
	4-bit most significant Nibble	4-bit less significant Nibble
1	(msb) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
...	...	...
S/2 - 2	Nibble 3	Nibble 2
S/2 - 1	Nibble 1	Nibble 0 (lsb)

**Table 15-229. Misaligned Nibble Mapping of Message in 8-bit NAND**

Byte offset	8-bit word	
	4-bit most significant Nibble	4-bit less significant Nibble
1	(msb) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
...	...	...
(S+1)/2 - 2	Nibble 2	Nibble 1
(S+1)/2 - 1	Nibble 0 (lsb)	

**Table 15-230. Aligned Nibble Mapping of Message in 16-bit NAND**

Byte offset	16-bit word			
	4-bit most significant Nibble		4-bit less significant Nibble	
0	Nibble S-3	Nibble S-4	(msb) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...	...	...	...
S/2 - 4	Nibble 5	Nibble 4	Nibble 7	Nibble 6
S/2 - 2	Nibble 1	Nibble 0 (lsb)	Nibble 3	Nibble 2

**Table 15-231. Misaligned Nibble Mapping of Message in 16-bit NAND (1 Unused Nibble)**

Byte offset	16-bit word			
	4-bit most significant Nibble		4-bit less significant Nibble	
0	Nibble S-3	Nibble S-4	(msb) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...	...	...	...
(S+1)/2 - 4	Nibble 4	Nibble 3	Nibble 6	Nibble 5
(S+1)/2 - 2	Nibble 0 (lsb)		Nibble 2	Nibble 1

**Table 15-232. Misaligned Nibble Mapping of Message in 16-bit NAND (2 Unused Nibbles)**

Byte offset	16-bit word			
	4-bit most significant Nibble		4-bit less significant Nibble	
0	Nibble S-3	Nibble S-4	(msb) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...	...	...	...
(S+2)/2 - 4	Nibble 3	Nibble 2	Nibble 5	Nibble 4
(S+2)/2 - 2			Nibble 1	Nibble 0 (lsb)

**Table 15-233. Misaligned Nibble Mapping of Message in 16-bit NAND (3 Unused Nibbles)**

Byte offset	16-bit word			
	4-bit most significant Nibble		4-bit less significant Nibble	
0	Nibble S-3	Nibble S-4	(msb) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...	...	...	...
(S+3)/2 - 4	Nibble 2	Nibble 1	Nibble 4	Nibble 3
(S+3)/2 - 2			Nibble 0 (lsb)	

Note that many other cases exist than the ones represented above, for example, where the message does not start on a word boundary.

#### 15.4.4.13.3.2.2.2 Memory-Mapping of the ECC

The ECC (or remainder) is presented by the BCH module as a single 104-bit (or 52-bit), little-endian vector. It is up to the software to fetch those 13 bytes (or 6 bytes) from the modules interface, then store them to the NANDs spare area (page write) or to an intermediate buffer for comparison with the stored ECC (page read). There are no constraints on the ECC mapping inside the spare area: it is a softwarecontrolled operation.

However, it is advised to maintain a coherence in the respective formats of the message or the ECC remainder once they have been read out of the NAND. The error correction algorithm works from the complete codeword (concatenated message and remainder) once an error as been detected. The creation of this codeword should be made as straightforward as possible.

There are cases where the same NAND access contains both data and the ECC protecting that data. This is the case when the data/ECC boundary (which can be on any nibble) does not coincide with an access boundary. The ECC is calculated on-the-fly following the write. In that case, the write must also contain part of the ECC because it is impossible to insert the ECC on-the-fly. Instead:

- During the initial page write (BCH encoding), the ECC is replaced by dummy bits. The BCH encoder is by definition turned OFF during the ECC section, so the BCH result is unmodified.
- During a second phase, the ECC is written to the correct location, next to the actual data.
- The completed line buffer is then written to the NAND array.

#### 15.4.4.13.3.2.2.3 Wrapping Modes

For a given wrapping mode, the module automatically goes through a specific number of sections, as data is being fed into the module. For each section, the BCH core can be enabled (in which case the data is fed to the BCH divider) or not (in which case the BCH simply counts to the end of the section). When enabled, the data is added to the ongoing calculation for a given sector number (for example, number 0).

Wrapping modes are described below. To get a better understanding and see the real-life read and write sequences implemented with each mode, see [Section 15.4.4.13.3.2.3, Supported NAND Page Mappings and ECC Schemes](#).

For each mode:

- a sequence describes the mode in pseudo-language, with for each section the size and the buffer used for ECC processing (if ON). The programmable lengths are size, size0 and size1.
- a checksum condition is given. If the checksum condition is not respected for a given mode, the modules behavior is unpredictable. S is the number of sectors in the page; size0 and size1 are the section sizes programmed for the mode, in nibbles.

Note that wrapping modes 8, 9, 10, and 11 insert a 1-nibble padding where the BCH processing is OFF. This is intended for  $t = 4$  ECC, where ECC is 6 bytes long and the ECC area is expected to include (at least) 1 unused nibble to remain byte-aligned.

#### 15.4.4.13.3.2.2.4 Manual Mode (0x0)

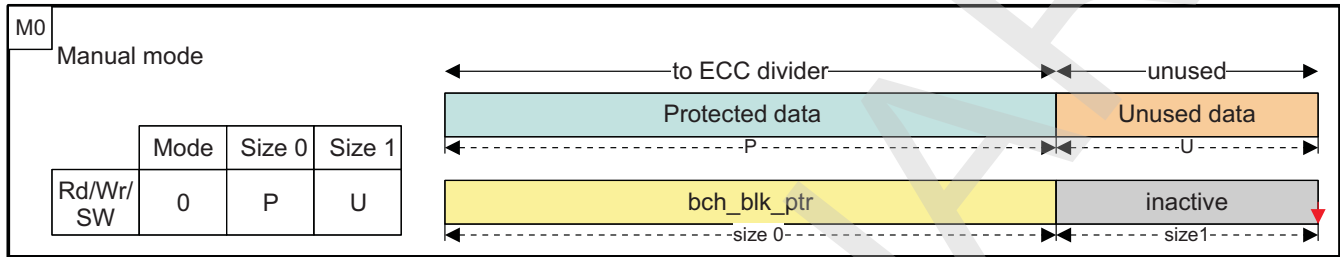
This mode is intended for short sequences, added manually to a given buffer through the software data port input. A complete page may be built out of several such sequences.



To process an arbitrary sequence of 4-bit nibbles, accesses to the software data port shall be made, containing the appropriate data. If the sequence end does not coincide with an access boundary (for example, to process 5 nibbles = 20 bits in 16-bit access mode) and those nibbles need to be skipped, a number of unused nibbles shall be programmed in size1 (in the same example: 5 nibbles to process + 3 to discard = 8 nibbles = exactly 2 x 16-bit accesses: we must program size0 = 5, size1 = 3).

**NOTE:** In the following figures size and size0 are the same parameter.

**Figure 15-86. Manual Mode Sequence and Mapping**



gpmc-032

Section processing sequence:

- One time with buffer
  - size0 nibbles of data, processing ON
  - size1 nibbles of unused data, processing OFF

Checksum: size0 + size1 nibbles must fit in a whole number of accesses.

In the following sections, S is the number of sectors in the page.

**15.4.4.13.3.2.2.5 Mode 0x1**

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + size1)

**15.4.4.13.3.2.2.6 Mode 0xA (10)**

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - 1 nibble pad spare, processing OFF
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + 1 + size1)

**15.4.4.13.3.2.2.7 Mode 0x2**

Page processing sequence:

- Repeat with buffer 0 to S-1

- 512-byte data, processing ON
  - Repeat with buffer 0 to S-1
    - size0 nibbles spare, processing OFF
    - size1 nibbles spare, processing ON
- Checksum: Spare area size (nibbles) = S - (size0 + size1)

#### 15.4.4.13.3.2.2.8 Mode 0x3

- Page processing sequence:
- Repeat with buffer 0 to S-1
    - 512-byte data, processing ON
  - One time with buffer 0
    - size0 nibbles spare, processing ON
  - Repeat with buffer 0 to S-1
    - size1 nibbles spare, processing ON
- Checksum: Spare area size (nibbles) = size0 + (S - size1)

#### 15.4.4.13.3.2.2.9 Mode 0x7

- Page processing sequence:
- Repeat with buffer 0 to S-1
    - 512-byte data, processing ON
  - One time with buffer 0
    - size0 nibbles spare, processing ON
  - Repeat S times (no buffer used)
    - size1 nibbles spare, processing OFF
- Checksum: Spare area size (nibbles) = size0 + (S - size1)

#### 15.4.4.13.3.2.2.10 Mode 0x8

- Page processing sequence:
- Repeat with buffer 0 to S-1
    - 512-byte data, processing ON
  - One time with buffer 0
    - size0 nibbles spare, processing ON
  - Repeat with buffer 0 to S-1
    - 1 nibble padding spare, processing OFF
    - size1 nibbles spare, processing ON
- Checksum: Spare area size (nibbles) = size0 + (S - (1+size1))

#### 15.4.4.13.3.2.2.11 Mode 0x4

- Page processing sequence:
- Repeat with buffer 0 to S-1
    - 512-byte data, processing ON
  - One time (no buffer used)
    - size0 nibbles spare, processing OFF
  - Repeat with buffer 0 to S-1
    - size1 nibbles spare, processing ON
- Checksum: Spare area size (nibbles) = size0 + (S - size1)

**15.4.4.13.3.2.2.12 Mode 0x9**

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - (1+size1))

**15.4.4.13.3.2.2.13 Mode 0x5**

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + size1)

**15.4.4.13.3.2.2.14 Mode 0xB (11)**

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + 1 + size1)

**15.4.4.13.3.2.2.15 Mode 0x6**

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + size1)

**15.4.4.13.3.2.3 Supported NAND Page Mappings and ECC Schemes**

The following rules apply throughout the entire mapping description:

- Main data area (sectors) size is hardcoded to 512 bytes.
- Spare area size is programmable.

- All page sections (of main area data bytes, protected spare bytes, unprotected spare bytes, and ECC) are defined as explained in [Section 15.4.4.13.3.2.1](#), *Memory-Mapping of the Data Message*.

Each one of the following sections shows a NAND page mapping example (per-sector spare mappings, pooled spare mapping, per-sector spare mapping, with ECC separated at the end of the page).

In the mapping diagrams, sections that belong to the same BCH codeword have the same color (blue or green); unprotected sections are not covered (orange) by the BCH scheme.

Below each mapping diagram, a write (encoding) and read (decoding: syndrome generation) sequence is given, with the number of the active buffers at each point in time (yellow). In the inactive zones (grey), no computing is taking place but the data counter is still active.

In [Figure 15-87](#) to [Figure 15-89](#), tables on the left summarize the mode, size0, size1 parameters to program for respectively write and read processing of a page, with the given mapping, where :

- P is the size of spare byte section Protected by the ECC (in nibbles)
- U is the size of spare byte section Unprotected by the ECC (in nibbles)
- E is the size of the ECC itself (in nibbles)
- S is the number of Sectors per page (2 in the current diagrams)

Each time the processing of a BCH block is complete (ECC calculation for write/encoding, syndrome generation for read/decoding, indicated by red arrows), the update pointer is pulsed. Note that the processing for block 0 can be the first or the last to complete, depending on the NAND page mapping and operation (read or write). All examples show a page size of 1kByte + spares, that is, S = 2 sectors of 512 bytes. The same principles can be extended to larger pages by adding more sectors.

The actual BCH codeword size is used during the error location work to restrict the search range: by definition, errors can only happen in the codeword that was actually written to the NAND, and not in the mathematical codeword of  $n = 2^{13} - 1 = 8191$  bits. That codeword (higher-order bits) is all-zero and implicit during computations.

The actual BCH codeword size depends on the mode, on the programmed sizes and on the sector number (all sizes in nibbles):

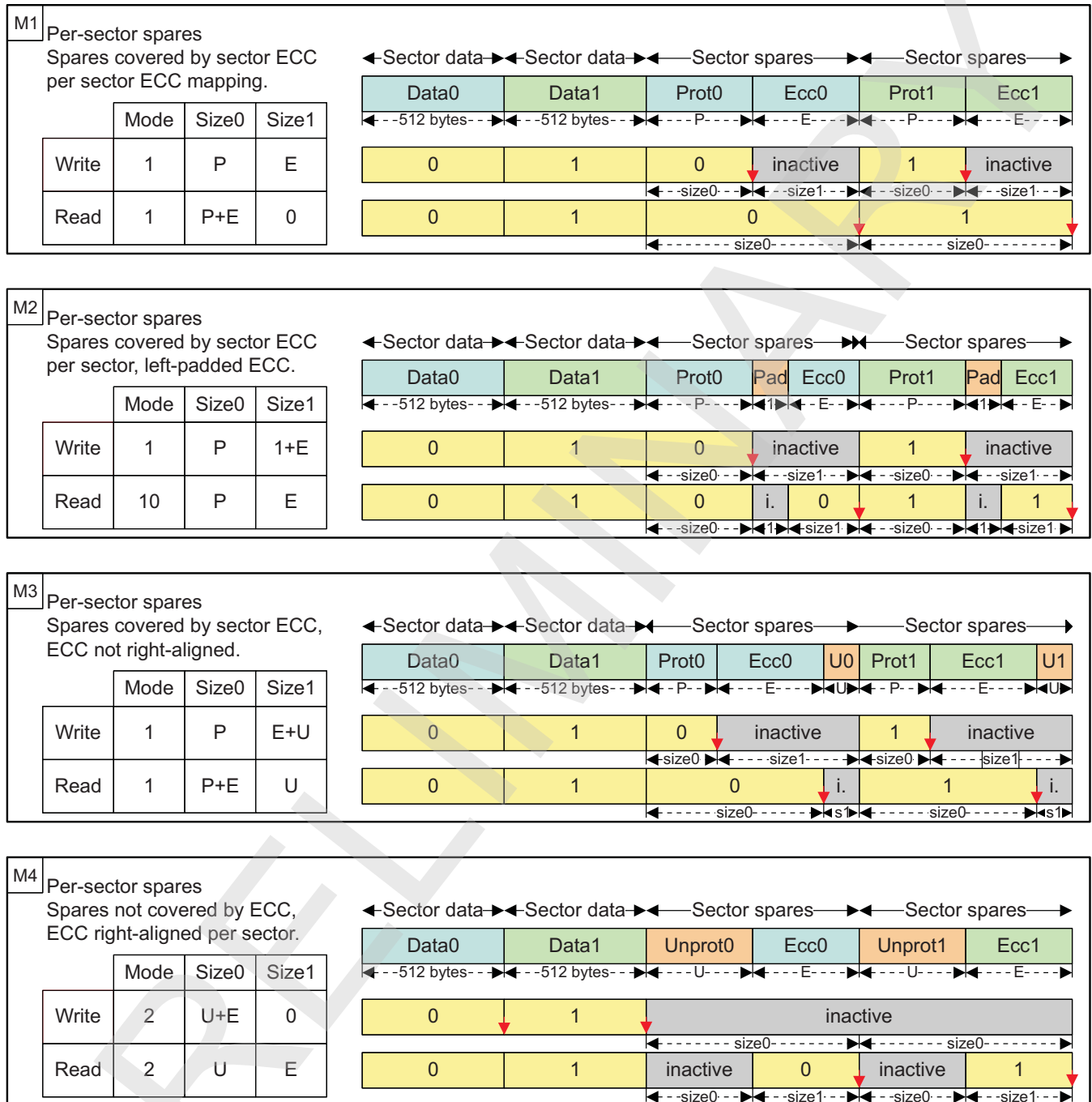
- Spares mapped and protected per sector (below: see M1-M2-M3-M9-M10):
  - all sectors: (512) + P + E
- Spares pooled and protected by sector 0 (below: see M5-M6):
  - sector 0 codeword: (512) + P + E
  - other sectors: (512) + E
- Unprotected spares (below: see M4-M7-M8-M11-M12):
  - all codewords (512) + E

#### 15.4.4.13.3.2.3.1 *Per-Sector Spare Mappings*

In these schemes, each 512-byte sector of the main area has its own dedicated section of the spare area. The spare area of each sector is composed of :

- ECC, which must be located after the data it protects
- other data, which may or may not be protected by the sectors ECC

Figure 15-87. NAND Page Mapping and ECC: Per-Sector Schemes



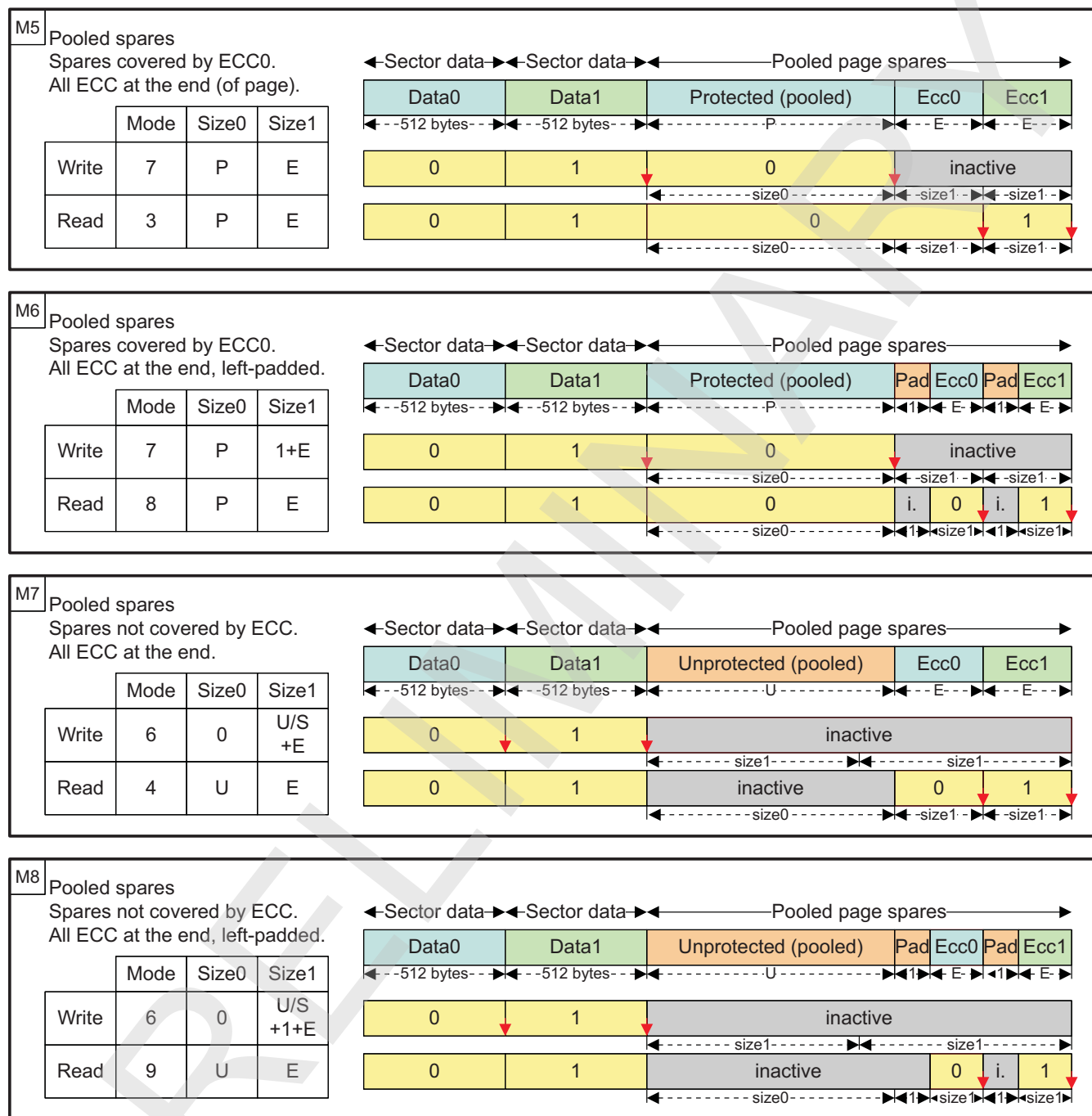
gpmc-033

### 15.4.4.13.3.2.3.2 Pooled Spare Mapping

In the schemes below, the spare area is pooled for the page.

- The ECC of each sector is aligned at the end of the spare area.
- The non-ECC spare data may or may not be covered by the ECC of sector 0

**Figure 15-88. NAND Page Mapping and ECC: Pooled Spare Schemes**



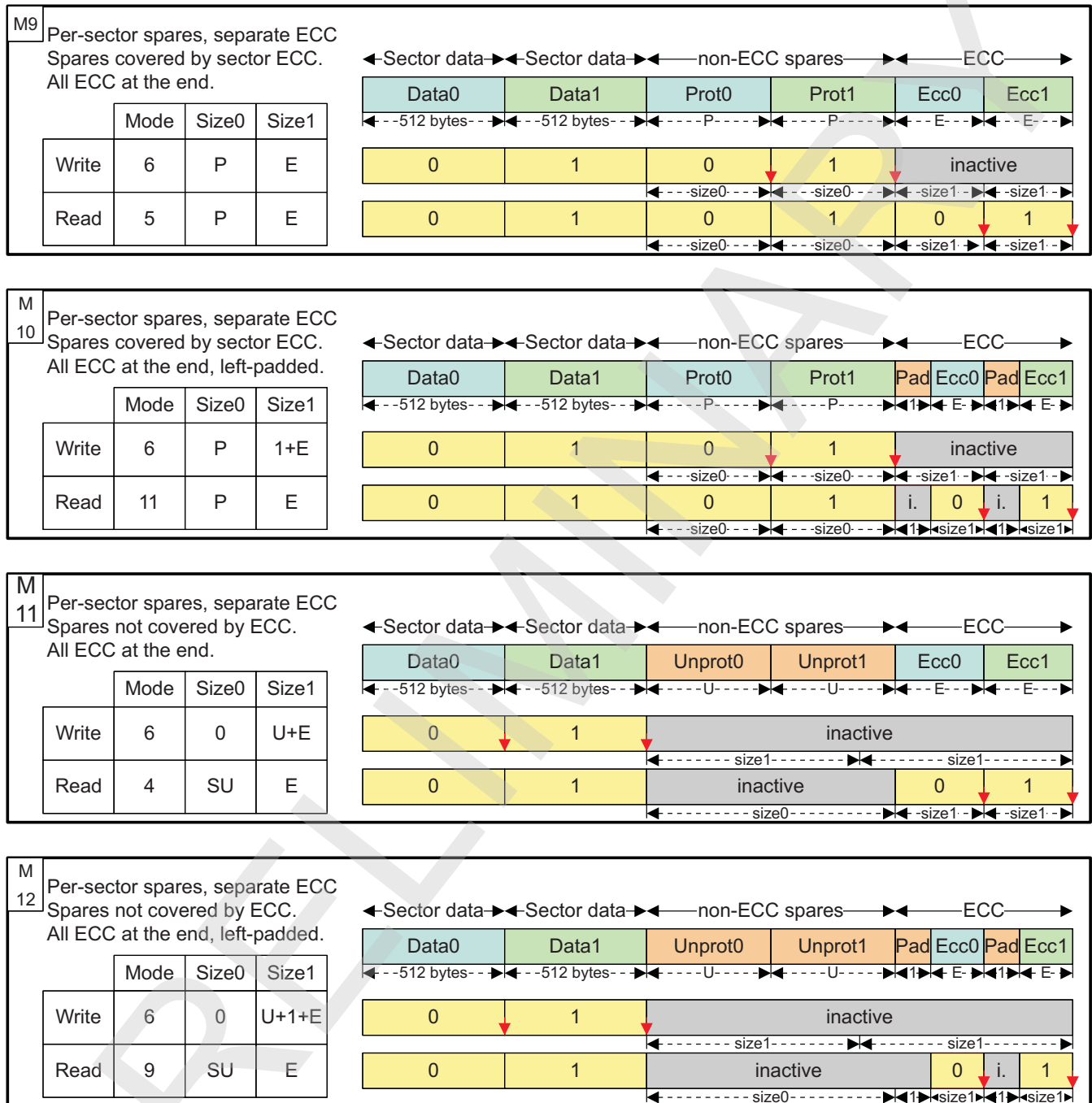
gpmc-034

**15.4.4.13.3.2.3.3 Per-Sector Spare Mapping, with ECC Separated at the End of the Page**

In these schemes, each 512-byte sector of the main area is associated with 2 sections of the spare area.

- ECC section, all aligned at the end of the page
- other data section, aligned before the ECCs, each of which may or may not be protected by its sectors ECC

Figure 15-89. NAND Page Mapping and ECC: Per-Sector Schemes, with Separate ECC



gpmc\_035

#### 15.4.4.13.4 Prefetch and Write-Posting Engine

NAND device data access cycles are usually much slower than the MPU system frequency; such NAND read or write accesses issued by the processor will impact the overall system performance, especially considering long read or write sequences required for NAND page loading or programming. To minimize this effect on system performance, the GPMC includes a prefetch and write-posting engine, which can be used to read from or write to any chip-select location in a buffered manner.

The prefetch and write-posting engine is a simplified embedded-access requester that presents requests to the access engine on a user-defined chip-select target. The access engine interleaves these requests with any request coming from the L3 interface; as a default the prefetch and write-posting engine has the lowest priority.

The prefetch and write-posting engine is dedicated to data-stream access (as opposed to random data access); thus, it is primarily dedicated to NAND support. The engine does not include an address generator; the request is limited to chip-select target identification. It includes a 64-byte FIFO associated with a DMA request synchronization line, for optimal DMA-based use.

The prefetch and write-posting engine uses an embedded 64 bytes (32 16-bit word) FIFO to prefetch data from the NAND device in read mode (prefetch mode) or to store host data to be programmed into the NAND device in write mode (write-posting mode). The FIFO draining and filling (read and write) can be controlled either by the MPU through interrupt synchronization (an interrupt is triggered whenever a programmable threshold is reached) or the sDMA through DMA request synchronization, with a programmable request byte size in both prefetch or posting mode.

The prefetch and write-posting engine includes a single memory pool. Therefore, only one mode, read or write, can be used at any given time. In other words, the prefetch and write-posting engine is a single-context engine that can be allocated to only one chip-select at a time for a read prefetch or a write-posting process.

The engine does not support atomic command and address phase programming and is limited to linear memory read or write access. In consequence, it is limited to NAND data-stream access. The engine relies on the MPU NAND software driver to control block and page opening with the correct data address pointer initialization, before the engine can read from or write to the NAND memory device.

Once started, the engine data reads and writes sequencing is solely based on FIFO location availability and until the total programmed number of bytes is read or written.

Any host-concurrent accesses to a different chip-select are correctly interleaved with ongoing engine accesses. The engine has the lowest priority access so that host accesses to a different chip-select do not suffer a large latency.

A round-robin arbitration scheme can be enabled to ensure minimum bandwidth to the prefetch and write-posting engine in the case of back-to-back direct memory requests to a different chip-select. If the [GPMC\\_PREFETCH\\_CONFIG1\[23\] PFPWENROUNDROBIN](#) bit is enabled, the arbitration grants the prefetch and write posting engine access to the GPMC bus for a number of requests programmed in the [GPMC\\_PREFETCH\\_CONFIG1\[19:16\] PFPWWEIGHTEDPRIO](#) field.

The prefetch/write-posting engine read or write request is routed to the access engine with the chip-select destination ID. After the required arbitration phase, the access engine processes the request as a single access with the data access size equal to the device size specified in the corresponding chip-select configuration.

---

**NOTE:** The destination chip-select configuration must be set to the NAND protocol-compatible configuration for which address lines are not used (the address bus is not changed from its current value). Selecting a different chip-select configuration can produce undefined behavior.

---

#### 15.4.4.13.4.1 General Facts About the Engine Configuration

The engine can be configured only if the [GPMC\\_PREFETCH\\_CONTROL\[0\] STARTENGINE](#) bit is de-asserted.

The engine must be correctly configured in prefetch or write-posting mode and must be linked to a NAND chip-select before it can be started. The chip-select is linked using the [GPMC\\_PREFETCH\\_CONFIG1\[26:24\] ENGINECSSELECTOR](#) field.

In both prefetch and write-posting modes, the engine respectively uses byte or 16-bit word access requests for an 8- or 16-bit wide NAND device attached to the linked chip-select. The FIFOTHRESHOLD and TRANSFERCOUNT fields must be programmed accordingly as a number of bytes or a number of 16-bit word.



When the [GPMC\\_PREFETCH\\_CONFIG1\[7\]](#) ENABLEENGINE bit is set, the FIFO entry on the L3 interconnect port side is accessible at any address in the associated chip-select memory region. When the ENABLEENGINE bit is set, any host access to this chip-select is rerouted to the FIFO input. Directly accessing the NAND device linked to this chip-select from the host is still possible through the [GPMC\\_NAND\\_COMMAND\\_i](#), [GPMC\\_NAND\\_ADDRESS\\_i](#), and [GPMC\\_NAND\\_DATA\\_i](#) registers (where i = 0 to 3).

The FIFO entry on the L3 interconnect port can be accessed with Byte, 16-bit word, or 32-bit word access size, according to little-endian format, even though the FIFO input is 32-bit wide.

The FIFO control is made easier through the use of interrupts or DMA requests associated with the FIFOTHRESHOLD bit field. The [GPMC\\_PREFETCH\\_STATUS\[30:24\]](#) FIFOPointer field monitors the number of available bytes to be read in prefetch mode or the number of free empty slots which can be written in write-posting mode. The [GPMC\\_PREFETCH\\_STATUS\[13:0\]](#) COUNTVALUE field monitors the number of remaining bytes to be read or written by the engine according to the TRANSFERCOUNT value. The FIFOPointer and COUNTVALUE bit fields are always expressed as a number of bytes even if a 16-bit wide NAND device is attached to the linked chip-select.

In prefetch mode, when the FIFOPointer equals 0, that is, the FIFO is empty, a host read access receives the byte last read from the FIFO as its response. In case of 32-bit word or 16-bit word read accesses, the last byte read from the FIFO is copied the required number of times to fit the requested word size. In write-posting mode, when the FIFOPointer equals 0, that is, the FIFO is full, a host write overwrites the last FIFO byte location. There is no underflow or overflow error reporting in the GPMC.

#### 15.4.4.13.4.2 Prefetch Mode

The prefetch mode is selected when the [GPMC\\_PREFETCH\\_CONFIG1\[0\]](#) ACCESSMODE bit is cleared.

The MPU NAND software driver must issue the block and page opening (READ) command with the correct data address pointer initialization before the engine can be started to read from the NAND memory device. The engine is started by asserting the [GPMC\\_PREFETCH\\_CONTROL\[0\]](#) STARTENGINE bit. The STARTENGINE bit automatically clears when the prefetch process completes.

If required, the ECC calculator engine must be initialized (that is, reset, configured, and enabled) before the prefetch engine is started, so that the ECC is correctly computed on all data read by the prefetch engine.

When the [GPMC\\_PREFETCH\\_CONFIG1\[3\]](#) SYNCHROMODE bit is cleared, the prefetch engine starts requesting data as soon as the STARTENGINE bit is set. If using this configuration, the host must monitor the NAND device-ready pin so that it only sets the STARTENGINE bit when the NAND device is in a ready state, meaning data is valid for prefetching.

When the SYNCHROMODE bit is set, the prefetch engine starts requesting data when an active to inactive wait signal transition is detected. The transition detector must be cleared before any transition detection; see [Section 15.4.4.13.2.2, Ready Pin Monitored by Hardware Interrupt](#). The [GPMC\\_PREFETCH\\_CONFIG1\[5:4\]](#) WAITPINSELECTOR field selects which gpmc\_wait pin edge detector triggers the prefetch engine in this synchronized mode.

If the STARTENGINE bit is set after the NAND address phase (page opening command), the engine is effectively started only after the actual NAND address phase completion. To prevent GPMC stall during this NAND address phase, set the STARTENGINE bit field before NAND address phase completion when in synchronized mode. The prefetch engine will start when an active to inactive wait signal transition is detected. The STARTENGINE bit is automatically cleared on prefetch process completion.

The prefetch engine issues a read request to fill the FIFO with the amount of data specified by [GPMC\\_PREFETCH\\_CONFIG2\[13:0\]](#) TRANSFERCOUNT field.

**Table 15-234. Prefetch Mode Configuration**

Bit Field	Register	Value	Comments
STARTENGINE	<a href="#">GPMC_PREFETCH_CONTROL[0]</a>	0	Prefetch engine can be configured only if STARTENGINE is set to 0.
ENGINECSSELECTOR	<a href="#">GPMC_PREFETCH_CONFIG1[26:24]</a>	0 to 3	Selects the chip-select associated with a NAND device where the prefetch engine is active.

**Table 15-234. Prefetch Mode Configuration (continued)**

Bit Field	Register	Value	Comments
ACCESSMODE	<a href="#">GPMC_PREFETCH_CONFIG1</a> [0]	0	Selects prefetch mode
FIFOTHRESHOLD	<a href="#">GPMC_PREFETCH_CONFIG1</a> [14:8]		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	<a href="#">GPMC_PREFETCH_CONFIG2</a> [13:0]		Selects the number of bytes to be read or written by the engine to the selected chip-select
SYNCHROMODE	<a href="#">GPMC_PREFETCH_CONFIG1</a> [3]	0/1	Selects when the engine starts the access to the chip-select
WAITPINSELECT	<a href="#">GPMC_PREFETCH_CONFIG1</a> [17:16]	0 to 1	Selects wait pin edge detector (if <a href="#">GPMC_PREFETCH_CONFIG1</a> [3] SYNCHROMODE = 0x1)
ENABLEOPTIMIZEDACCESS	<a href="#">GPMC_PREFETCH_CONFIG1</a> [27]	0/1	See <a href="#">Section 15.4.4.13.4.6, Optimizing NAND Access Using the Prefetch and Write-Posting Engine</a> .
CYCLEOPTIMIZATION	<a href="#">GPMC_PREFETCH_CONFIG1</a> [30:28]		Number of clock cycle removed to timing parameters
ENABLEENGINE	<a href="#">GPMC_PREFETCH_CONFIG1</a> [7]	1	Engine enabled
STARTENGINE	<a href="#">GPMC_PREFETCH_CONTROL</a> [0]	1	Starts the prefetch engine

#### 15.4.4.13.4.3 FIFO Control in Prefetch Mode

The FIFO can be drained directly by the MPU or by an sDMA channel.

In MPU draining mode, the FIFO status can be monitored through the [GPMC\\_PREFETCH\\_STATUS](#)[30:24] FIFOPINTER field or through the [GPMC\\_PREFETCH\\_STATUS](#)[16] FIFOTHRESHOLDSTATUS bit. The FIFOPINTER indicates the current number of available data to be read; FIFOTHRESHOLDSTATUS set to 1 indicates that at least FIFOTHRESHOLD bytes are available from the FIFO.

An interrupt can be triggered by the GPMC if the [GPMC\\_IRQENABLE](#)[0] FIFOEVENTENABLE bit is set. The FIFO interrupt event is logged, and the [GPMC\\_IRQSTATUS](#)[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, the MPU must read all the available bytes, or at least enough bytes to get below the programmed FIFO threshold, and the FIFOEVENTSTATUS bit must be cleared to enable further interrupt events. The FIFOEVENTSTATUS bit must always be reset prior to asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt generation must be enabled after enabling the STARTENGINE bit.

Prefetch completion can be monitored through the [GPMC\\_PREFETCH\\_STATUS](#)[13:0] COUNTVALUE field. COUNTVALUE indicates the number of currently remaining data to be requested according to the TRANSFERCOUNT value. An interrupt can be triggered by the GPMC when the prefetch process is complete (that is, COUNTVALUE equals 0) if the [GPMC\\_IRQENABLE](#)[1] TERMINALCOUNTEVENTENABLE bit is set. At prefetch completion, the TERMINALCOUNT interrupt event is also logged, and the [GPMC\\_IRQSTATUS](#)[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTATUS bit. The TERMINALCOUNTSTATUS bit must always be cleared prior to asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

---

**NOTE:** The COUNTVALUE value is only valid when the prefetch engine is active (started), and an interrupt is only triggered when COUNTVALUE reaches 0, that is, when the prefetch engine automatically goes from an active to an inactive state.

---

The number of bytes to be prefetched (programmed in TRANSFERCOUNT) must be a multiple of the programmed FIFOTHRESHOLD to trigger the correct number of interrupts allowing a deterministic and transparent FIFO control. If this guideline is respected, the number of ISR accesses is always required and the FIFO is always empty after the last interrupt is triggered. In other cases, the TERMINALCOUNT interrupt must be used to read the remaining bytes in the FIFO (the number of remaining bytes being lower than the FIFOTHRESHOLD value).

In DMA draining mode, the [GPMC\\_PREFETCH\\_CONFIG1\[2\]](#) DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes are ready to be read from the FIFO. The DMA channel owning this DMA request must be programmed so that the number of bytes programmed in FIFOTHRESHOLD is read from the FIFO during the DMA request process. The DMA request is kept active until this number of bytes has effectively been read from the FIFO, and no other DMA request can be issued until the ongoing active request is complete.

In prefetch mode, the TERMINALCOUNT event is also a source of DMA requests if the number of bytes to be prefetched is not a multiple of FIFOTHRESHOLD, the remaining bytes in the FIFO can be read by the DMA channel using the last DMA request. This assumes that the number of remaining bytes to be read is known and controlled through the DMA channel programming model.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive to active prefetch (the STARTENGINE bit is set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that the out-of-date active DMA request does not trigger spurious DMA transfers.

#### 15.4.4.13.4.4 Write-Posting Mode

The write-posting mode is selected when the [GPMC\\_PREFETCH\\_CONFIG1\[0\]](#) ACCESSMODE bit is set.

The MPU NAND software driver must issue the correct address pointer initialization command (page program) before the engine can start writing data into the NAND memory device. The engine starts when the [GPMC\\_PREFETCH\\_CONTROL\[0\]](#) STARTENGINE bit is set to 1. The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion (adding ECC handling, if required).

If used, the ECC calculator engine must be started (configured, reset, and enabled) before the posting engine is started so that the ECC parities are properly calculated on all data written by the prefetch engine to the associated chip-select.

In write-posting mode, the [GPMC\\_PREFETCH\\_CONFIG1\[3\]](#) SYNCHROMODE bit must be cleared so that posting starts as soon as the STARTENGINE bit is set and the FIFO is not empty.

If the STARTENGINE bit is set after the NAND address phase (page program command), the STARTENGINE setting is effective only after the actual NAND command completion. To prevent GPMC stall during this NAND command phase, set the STARTENGINE bit field before the NAND address completion and ensure that the associated DMA channel is enabled after the NAND address phase.

The posting engine issues a write request when valid data are available from the FIFO and until the programmed [GPMC\\_PREFETCH\\_CONFIG2\[13:0\]](#) TRANSFERCOUNT accesses have been completed.

The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion. The closing program command phase must only be issued when the full NAND page has been written into the NAND flash write buffer, including the spare area data and the ECC parities, if used.

**Table 15-235. Write-Posting Mode Configuration**

Bit Field	Register	Value	Comments
STARTENGINE	<a href="#">GPMC_PREFETCH_CONTROL[0]</a>	0	Write-posting engine can be configured only if STARTENGINE is set to 0.
ENGINECSSELECTOR	<a href="#">GPMC_PREFETCH_CONFIG1[26:24]</a>	0 to 3	Selects the chip-select associated with a NAND device where the prefetch engine is active
ACCESSMODE	<a href="#">GPMC_PREFETCH_CONFIG1[0]</a>	1	Selects write-posting mode
FIFOTHRESHOLD	<a href="#">GPMC_PREFETCH_CONFIG1[14:8]</a>		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	<a href="#">GPMC_PREFETCH_CONFIG2[13:0]</a>		Selects the number of bytes to be read or written by the engine from/to the selected chip-select

**Table 15-235. Write-Posting Mode Configuration (continued)**

Bit Field	Register	Value	Comments
SYNCHROMODE	GPMC_PREFETCH_CONFIG1[3]	0	Engine starts the access to chip-select as soon as STARTENGINE is set.
ENABLEOPTIMIZEDACCESS	GPMC_PREFETCH_CONFIG1[27]	0/1	See <a href="#">Section 15.4.4.13.4.6, Optimizing NAND Access Using the Prefetch and Write-Posting Engine.</a>
CYCLOPTIMIZATION	GPMC_PREFETCH_CONFIG1[30:28]		
ENABLEENGINE	GPMC_PREFETCH_CONFIG1[7]	1	Engine enabled
STARTENGINE	GPMC_PREFETCH_CONTROL[0]	1	Starts the prefetch engine

#### 15.4.4.13.4.5 FIFO Control in Write-Posting Mode

The FIFO can be filled directly by the MPU or by an sDMA channel.

In MPU filling mode, the FIFO status can be monitored through the FIFOPINTER or through the [GPMC\\_PREFETCH\\_STATUS](#)[16] FIFOTHRESHOLDSTATUS bit. FIFOPINTER indicates the current number of available free byte places in the FIFO, and the FIFOTHRESHOLDSTATUS bit, when set, indicates that at least FIFOTHRESHOLD free byte places are available in the FIFO.

An interrupt can be issued by the GPMC if the [GPMC\\_IRQENABLE](#)[0] FIFOEVENTENABLE bit is set. When the interrupt is fired, the [GPMC\\_IRQSTATUS](#)[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, the MPU must write enough bytes to fill the FIFO, or enough bytes to get below the programmed threshold, and the FIFOEVENTSTATUS bit must be cleared to get further interrupt events. The FIFOEVENTSTATUS bit must always be cleared prior to asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt must be enabled after enabling the STARTENGINE bit.

The posting completion can be monitored through the [GPMC\\_PREFETCH\\_STATUS](#)[13:0] COUNTVALUE field. COUNTVALUE indicates the current number of remaining data to be written based on the TRANSFERCOUNT value. An interrupt is issued by the GPMC when the write-posting process completes (that is, COUNTVALUE equal to 0) if the [GPMC\\_IRQENABLE](#)[1] TERMINALCOUNTEVENTENABLE bit is set. When the interrupt is fired, the [GPMC\\_IRQSTATUS](#)[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTATUS bit. The TERMINALCOUNTSTATUS bit must always be cleared prior to asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

---

**NOTE:** The COUNTVALUE value is only valid if the write-posting engine is active and started, and an interrupt is only issued when COUNTVALUE reaches 0, that is, when the posting engine automatically goes from active to inactive.

---

In DMA filling mode, the DMAMode bit field in the [GPMC\\_PREFETCH\\_CONFIG1](#)[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes-free places are available in the FIFO. The DMA channel owning this DMA request must be programmed so that a number of bytes equal to the value programmed in the FIFOTHRESHOLD bit field are written into the FIFO during the DMA access. The DMA request remains active until the associated number of bytes has effectively been written into the FIFO, and no other DMA request can be issued until the ongoing active request has been completed.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive to active prefetch (STARTENGINE set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that an out-of-date active DMA request does not trigger spurious DMA transfers.

In write-posting mode, the DMA or the MPU fill the FIFO with no consideration to the associated byte enables. Any byte stored in the FIFO is written into the memory device.

#### 15.4.4.13.4.6 Optimizing NAND Access Using the Prefetch and Write-Posting Engine

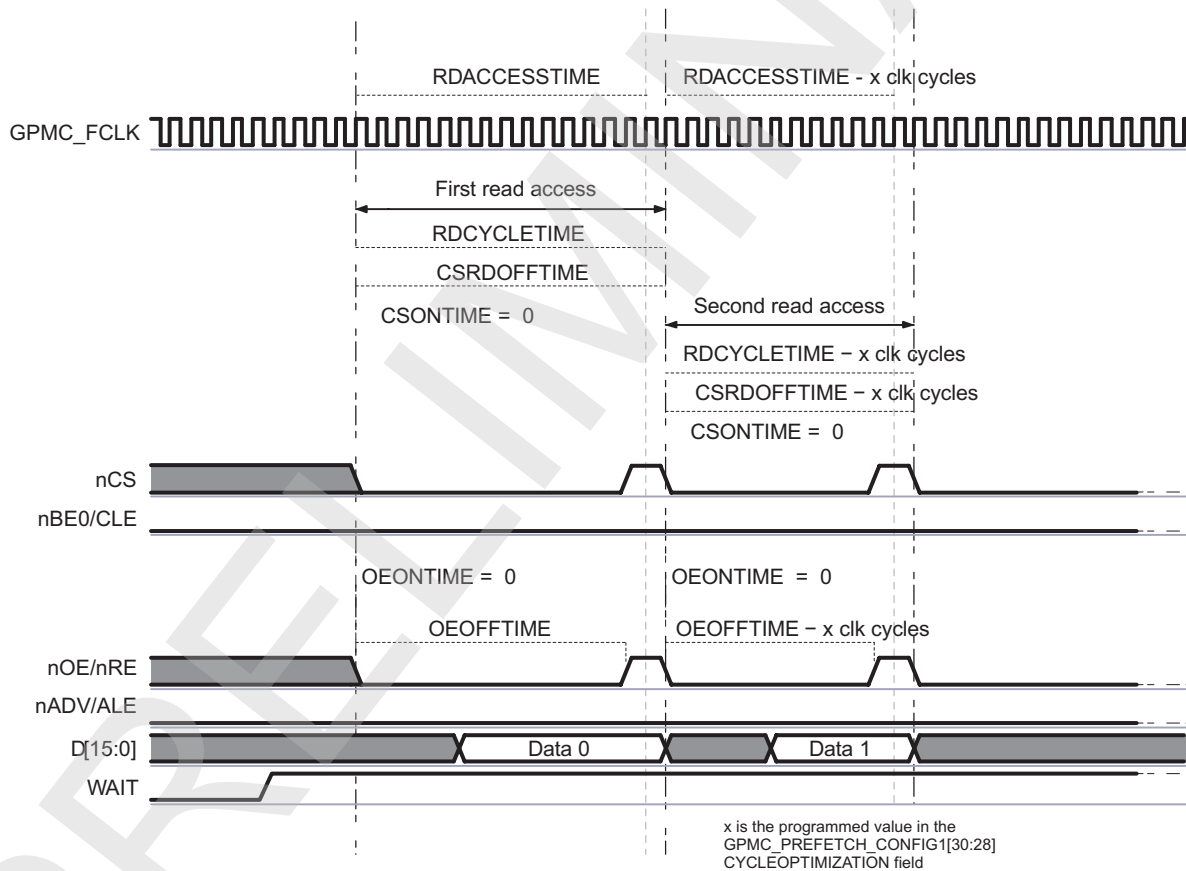
Access time to a NAND memory device can be optimized for back-to-back accesses if the associated nCS

signal is not deasserted between accesses. The GPMC access engine can track prefetch engine accesses to optimize the access timing parameter programmed for the allocated chip-select, if no accesses to other chip-selects (that is, interleaved accesses) occur. Similarly, the access engine also eliminates the CYCLE2CYCLEDELAY even if CYCLE2CYCLESAMECSSEN is set. This capability is limited to the prefetch and write-posting engine accesses, and MPU accesses to a NAND memory device (through the defined chip-select memory region or through the GPMC\_NAND\_DATA\_i location, i = 0 to 3) are never optimized.

The GPMC\_PREFETCH\_CONFIG1[27] ENABLEOPTIMIZEDACCESS bit must be set to enable optimized accesses. To optimize access time, the GPMC\_PREFETCH\_CONFIG1[30:28] CYCLEOPTIMIZATION field defines the number of GPMC\_FCLK cycles to be suppressed from the RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, WRACCESSTIME, CSOFFTIME, ADVOFFTIME, OEOFFTIME, and WEOFFTIME timing parameters.

Figure 15-90 highlights that, in the case of back-to-back accesses to the NAND flash through the prefetch engine, CYCLE2CYCLESAMECSSEN is forced to 0 when using optimized accesses. The first access uses the regular timing settings for this chip-select. All accesses after this one use settings reduced by x clock cycles, x being defined by the GPMC\_PREFETCH\_CONFIG1[30:28] CYCLEOPTIMIZATION field.

Figure 15-90. NAND Read Cycle Optimization Timing Description



gpmc-036

#### 15.4.4.13.4.7 Interleaved Accesses Between Prefetch and Write-Posting Engine and Other Chip-Selects

Any on-going read or write access from the prefetch and write-posting engine is completed before an access to any other chip-select can be initiated. As a default, the arbiter uses a fixed-priority algorithm, and the prefetch and write-posting engine has the lowest priority. The maximum latency added to access starting time in this case equals the RDCYCLETIME or WRCYCLETIME (optimized or not) plus the requested BUSTURNAROUND delay for bus turnaround completion programmed for the chip-select to which the NAND device is connected to.

Alternatively, a round-robin arbitration can be used to prioritize accesses to the external bus. This arbitration scheme is enabled by setting the [GPMC\\_PREFETCH\\_CONFIG1\[23\]](#) PFPWENROUNDROBIN bit. When a request to another chip-select is received while the prefetch and write-posting engine is active, priority is given to the new request. The request processed thereafter is the prefetch and write-posting engine request, even if another interconnect request is passed in the mean time. The engine keeps control of the bus for an additional number of requests programmed in the [GPMC\\_PREFETCH\\_CONFIG1\[19:16\]](#) PFPWWEIGHTEDPRIO bit field. Control is then passed to the direct interconnect request.

As an example, the round-robin arbitration scheme is selected with PFPWWEIGHTEDPRIO set to 0x2. Considering the prefetch and write-posting engine and the interconnect interface are always requesting access to the external interface, the GPMC grants priority to the direct interconnect access for one request. The GPMC then grants priority to the engine for three requests, and finally back to the direct interconnect access, until the arbiter is reset when one of the two initiators stops initiating requests.

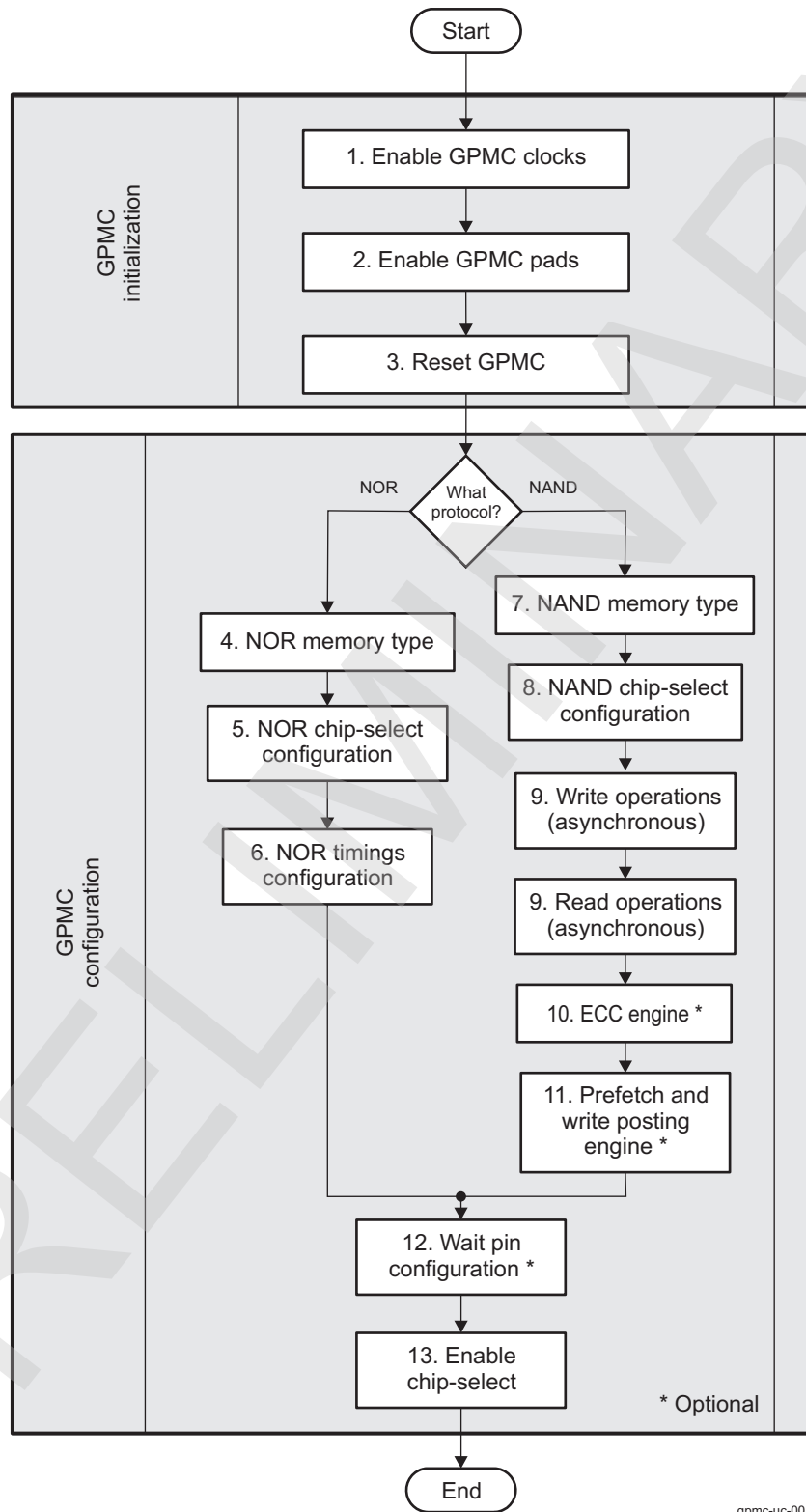
## 15.4.5 GPMC Basic Programming Model

### 15.4.5.1 GPMC High Level Programming Model Overview

The goal of the basic high-level programming model is to introduce a top down approach to users that need to configure the GPMC module.

[Figure 15-91](#), [Table 15-236](#), [Table 15-237](#) and [Table 15-238](#) show a programming model top level diagram for the GPMC, and the description of each step. Each block of the diagram is detailed in one of the following subsections through a set of registers to configure.

Figure 15-91. Programming Model Top Level Diagram



gpmc-uc-001

**Table 15-236. GPMC Initialization**

Step	Description
Enable GPMC clocks	See <a href="#">Table 15-239</a>
Enable GPMC pads	See <a href="#">Table 15-240</a>
Reset GPMC	See <a href="#">Table 15-241</a>

**Table 15-237. GPMC Configuration in NOR mode**

Step	Description
NOR Memory Type	See <a href="#">Table 15-242</a>
NOR Chip-Select Configuration	See <a href="#">Table 15-243</a>
NOR Timings Configuration	See <a href="#">Table 15-244</a>
Wait Pin Configuration	See <a href="#">Table 15-252</a>
Enable Chip-Select	See <a href="#">Table 15-253</a>

**Table 15-238. GPMC Configuration in NAND mode**

Step	Description
NAND Memory Type	See <a href="#">Table 15-247</a>
NAND Chip-Select Configuration	See <a href="#">Table 15-248</a>
Write Operations (Asynchronous)	See <a href="#">Table 15-249</a>
Read Operations (Asynchronous)	See <a href="#">Table 15-249</a>
ECC Engine	See <a href="#">Table 15-250</a>
Prefetch and Write-Posting Engine	See <a href="#">Table 15-251</a>
Wait Pin Configuration	See <a href="#">Table 15-252</a>
Enable Chip-Select	See <a href="#">Table 15-253</a>

### 15.4.5.2 GPMC Initialization

[Table 15-269](#) describes the settings required to prepare the GPMC, that is enabling its clock, its pads, and proceeding to a GPMC reset.

**NOTE:** In the tables of this section, 'x' in Value column stands for 'depends on configuration'.

**Table 15-239. Enable GPMC Clocks**

Sub-process Name	Register / Bit Field	Value
Enable the gpmc_clk pad	CONTROL.CONTROL_PADCONF_GPMC_NCS7[31:16] MUXMODE1	x
Configure the GPMC_CLK IO cell drive strength	CONTROL.CONTROL_PROG_IO0[6] GPMC_CLK	x
Set the CORE_L3_ICLK ratio vs CORE_CLK	PRCM.CM_CLKSEL_CORE[1:0] CLKSEL_L3	x
Set idle mode configuration to prevent CORE_L3_ICLK clock gating	GPMC.GPMC_SYSCONFIG[4:3] IDLEMODE	x

**Table 15-240. Enable GPMC Pads**

Sub-process Name	Register / Bit Field	Value
Configure the gpmc_a1 pad	CONTROL.CONTROL_PADCONF_SDRD_DQS3	x
Configure gpmc_a2 and gpmc_a3 pads	CONTROL.CONTROL_PADCONF_GPMC_A2	x
Configure gpmc_a4 and gpmc_a5 pads	CONTROL.CONTROL_PADCONF_GPMC_A4	x
Configure gpmc_a6 and gpmc_a7 pads	CONTROL.CONTROL_PADCONF_GPMC_A6	x
Configure gpmc_a8 and gpmc_a9 pads	CONTROL.CONTROL_PADCONF_GPMC_A8	x
Configure gpmc_a10 and gpmc_d0 pads	CONTROL.CONTROL_PADCONF_GPMC_A10	x



**Table 15-240. Enable GPMC Pads (continued)**

Sub-process Name	Register / Bit Field	Value
Configure gpmc_d1 and gpmc_d2 pads	CONTROL.CONTROL_PADCONF_GPMC_D1	x
Configure gpmc_d3 and gpmc_d4 pads	CONTROL.CONTROL_PADCONF_GPMC_D3	x
Configure gpmc_d5 and gpmc_d6 pads	CONTROL.CONTROL_PADCONF_GPMC_D5	x
Configure gpmc_d7 and gpmc_d8 pads	CONTROL.CONTROL_PADCONF_GPMC_D7	x
Configure gpmc_d9 and gpmc_d10 pads	CONTROL.CONTROL_PADCONF_GPMC_D9	x
Configure gpmc_d11 and gpmc_d12 pads	CONTROL.CONTROL_PADCONF_GPMC_D11	x
Configure gpmc_d13 and gpmc_d14 pads	CONTROL.CONTROL_PADCONF_GPMC_D13	x
Configure gpmc_d15 and gpmc_ncs0 pads	CONTROL.CONTROL_PADCONF_GPMC_D15	x
Configure gpmc_ncs1 and gpmc_ncs2 pads	CONTROL.CONTROL_PADCONF_GPMC_NCS1	x
Configure gpmc_ncs3 and gpmc_ncs4 pads	CONTROL.CONTROL_PADCONF_GPMC_NCS3	x
Configure gpmc_ncs5 and gpmc_ncs6 pads	CONTROL.CONTROL_PADCONF_GPMC_NCS5	x
Configure gpmc_ncs7 and gpmc_io_dir pads	CONTROL.CONTROL_PADCONF_GPMC_NCS7	x
Configure gpmc_nadv_ale and gpmc_noe pads	CONTROL.CONTROL_PADCONF_GPMC_NADV_ALE	x
Configure gpmc_nwe and gpmc_nbe0_cle pads	CONTROL.CONTROL_PADCONF_GPMC_NWE	x
Configure gpmc_nbe1 and gpmc_nwp pads	CONTROL.CONTROL_PADCONF_GPMC_NBE1	x
Configure gpmc_wait0 and gpmc_wait1 pads	CONTROL.CONTROL_PADCONF_GPMC_WAIT0	x
Configure gpmc_wait2 and gpmc_wait3 pads	CONTROL.CONTROL_PADCONF_GPMC_WAIT2	x
Configure the GPMC I/O cells drive strength	CONTROL.CONTROL_PROG_IO0	x

**Table 15-241. Reset GPMC**

Sub-process Name	Register / Bit Field	Value
Start a software reset	GPMC_SYSCONFIG[1] SOFTRESET	0x1
Wait until	GPMC_SYSSTATUS[0] RESETDONE =	0x1

### 15.4.5.3 GPMC Configuration in NOR Mode

This section gives a generic configuration for parameters related to the NOR memory connected to the GPMC.

**Table 15-242. NOR Memory Type**

Sub-process Name	Register / Bit Field	Value
Set the NOR protocol	GPMC_CONFIG1_i[11:10] DEVICETYPE	0x0
Set a device size	GPMC_CONFIG1_i[13:12] DEVICESIZE	x
Select an address and data multiplexing protocol	GPMC_CONFIG1_i[9] MUXADDDATA	x
Set the attached device page length	GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH	x
Set the wrapping burst capabilities	GPMC_CONFIG1_i[31] WRAPBURST	x
Select a timing signals latencies factor	GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY	x
Select an output clock frequency <sup>(1)</sup>	GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER	x
Choose an output clock activation time <sup>(1)</sup>	GPMC_CONFIG1_i[26:25] CLKACTIVATIONTIME	x
Set a single or multiple access for read operations <sup>(1)</sup>	GPMC_CONFIG1_i[30] READMULTIPLE	x
Set a synchronous or asynchronous mode for read operations	GPMC_CONFIG1_i[29] READTYPE	x
Set a single or multiple access for write operations	GPMC_CONFIG1_i[28] WRITEMULTIPLE	x
Set a synchronous or asynchronous mode for write operations	GPMC_CONFIG1_i[27] WRITETYPE	x

<sup>(1)</sup> Only applies to synchronous configurations (or non-multiplexed async for the multiple access one).

**Table 15-243. NOR Chip-Select Configuration**

Sub-process Name	Register / Bit Field	Value
Select the chip-select base address	GPMC_CONFIG7_[5:0] BASEADDRESS	x
Select the chip-select mask address	GPMC_CONFIG7_[11:8] MASKADDRESS	x

**Table 15-244. NOR Timings Configuration**

Sub-process Name	Register / Bit Field	Value
Configure adequate timing parameters in various memory modes	See <a href="#">Section 15.4.5.6, GPMC Timing Parameters</a>	

**Table 15-245. Wait Pin Configuration**

Sub-process Name	Register / Bit Field	Value
Enable or disable wait pin monitoring for read operations	GPMC_CONFIG1_[22] WAITREADMONITORING	x
Enable or disable wait pin monitoring for write operations	GPMC_CONFIG1_[21] WAITWRITEMONITORING	x
Select a wait pin monitoring time	GPMC_CONFIG1_[19:18] WAITMONITORINGTIME	x
Choose the input wait pin for the chip-select	GPMC_CONFIG1_[17:16] WAITPINSELECT	x

**Table 15-246. Enable Chip-Select**

Sub-process Name	Register / Bit Field	Value
When all parameters are configured, enable the chip-select	GPMC_CONFIG7_[6] CSVALID	x

#### 15.4.5.4 GPMC Configuration in NAND mode

This section gives a generic configuration for parameters related to the NAND memory connected to the GPMC.

**Table 15-247. NAND Memory Type**

Sub-process Name	Register / Bit Field	Value
Set the NAND protocol	GPMC_CONFIG1_[11:10] DEVICETYPE	0x2
Set a device size	GPMC_CONFIG1_[13:12] DEVICESIZE	x
Set the address and data multiplexing protocol to non-multiplexed attached device	GPMC_CONFIG1_[9] MUXADDDATA	0x0
Select a timing signals latencies factor	GPMC_CONFIG1_[4] TIMEPARAGRANULARITY	x
Set a synchronous or asynchronous mode and a single or multiple access for read and write operations	See <a href="#">Section 15.4.5.5, Set Memory Access</a>	x

**Table 15-248. NAND Chip-Select Configuration**

Sub-process Name	Register / Bit Field	Value
Select the chip-select base address	GPMC_CONFIG7_[5:0] BASEADDRESS	x
Select the chip-select minimum granularity (16M bytes)	GPMC_CONFIG7_[11:8] MASKADDRESS	x

**Table 15-249. Asynchronous Read and Write Operations**

Sub-process Name	Register / Bit Field	Value
Configure adequate timing parameters in asynchronous modes	See <a href="#">Section 15.4.5.6, GPMC Timing Parameters</a>	

**Table 15-250. ECC Engine**

Sub-process Name	Register / Bit Field	Value
Select the ECC result register where the first ECC computation is stored (Only applies to Hamming)	GPMC_ECC_CONTROL[3:0] ECCPOINTER	x <sup>(1)</sup>

<sup>(1)</sup> This parameter depends on the numbers of sectors in a page.

**Table 15-250. ECC Engine (continued)**

Sub-process Name	Register / Bit Field	Value
Clear all ECC result registers	GPMC_ECC_CONTROL[8] ECCCLEAR	Write '1' to clear.
Define ECCSIZE0 and ECCSIZE1	GPMC_ECC_SIZE_CONFIG[19:12] ECCSIZE0 and [29:22] ECCSIZE1	x <sup>(2)</sup>
Select the size of each of the 9 result registers (size specified by ECCSIZE0 or ECCSIZE1)	GPMC_ECC_SIZE_CONFIG[j-1] ECCjRESULTSIZ where j = 1 to 9	x
Select the chip-select where ECC is computed	GPMC_ECC_CONFIG[3:1] ECCCS	x
Select the Hamming code or BCH code ECC algorithm in use	GPMC_ECC_CONFIG[16] ECCALGORITHM	x
Select word size for ECC calculation	GPMC_ECC_CONFIG[7] ECC16B	x
If the BCH code is used, Set an error correction capability and Select a number of sectors to process	GPMC_ECC_CONFIG[13:12] ECCBCHTSEL and GPMC_ECC_CONFIG[6:4] ECCTOPSECTOR	x
Enable the ECC computation	GPMC_ECC_CONFIG[0] ECCENABLE	0x1

<sup>(2)</sup> Depends on the size of each sector in the NAND page.

**Table 15-251. Prefetch and Write-Posting Engine**

Sub-process Name	Register / Bit Field	Value
Disable the engine before configuration	GPMC_PREFETCH_CONTROL[0] STARTENGINE	0x0
Select the chip-select associated with a NAND device where the prefetch engine is active	GPMC_PREFETCH_CONFIG1[26:24] ENGINECSSELECTOR	x
Select access direction through prefetch engine, read or write.	GPMC_PREFETCH_CONFIG1[0] ACCESSMODE	x
Select the threshold used to issue a DMA request	GPMC_PREFETCH_CONFIG1[14:8] FIFOTHRESHOLD	x
Select either DMA synchronized mode or software manual mode.	GPMC_PREFETCH_CONFIG1[2] DMAMODE	x
Select if the engine immediately starts accessing the memory upon STARTENGINE assertion or if hardware synchronization based on a WAIT signal is used.	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode	GPMC_PREFETCH_CONFIG1[5:4] WAITPINSELECTOR	x
Enter a number of clock cycles removed to timing parameters (For all back-to-back accesses to the NAND flash but not the first one)	GPMC_PREFETCH_CONFIG1[30:28] CYCLOPTIMIZATION	x
Enable the prefetch postwrite engine	GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE	0x1
Select the number of bytes to be read or written by the engine to the selected chip-select	GPMC_PREFETCH_CONFIG2[13:0] TRANSFERCOUNT	x
Start the prefetch engine	GPMC_PREFETCH_CONTROL[0] STARTENGINE	0x1

**Table 15-252. Wait Pin Configuration**

Sub-process Name	Register / Bit Field	Value
Selects when the engine starts the access to CS	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode	GPMC_PREFETCH_CONFIG1[5:4] WAITPINSELECTOR	x

**Table 15-253. Enable Chip-Select**

Sub-process Name	Register / Bit Field	Value
When all parameters are configured, enable the chip-select	GPMC_CONFIG7_[6] CSVALID	x

### 15.4.5.5 Set Memory Access

This section details the bit field to configure to set the GPMC in various memory modes.

**Table 15-254. Mode Parameters Check List Table**

Register	Bit	Bit field name	Asynchronous				Synchronous			
			Single Read Access	Single Write Access	Multiple Read (Page) Access	Multiple Write (Page) Access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access
<a href="#">GPMC_CONFIG1_i</a>	30	READMULTIPLE	0x0	-	0x1 <sup>(1)</sup>	N/S	0x0	-	0x1	-
<a href="#">GPMC_CONFIG1_i</a>	29	READTYPE	0x0	-	0x0 <sup>(1)</sup>	N/S	0x1	-	0x1	-
<a href="#">GPMC_CONFIG1_i</a>	28	WRITEMULTIPLE	-	0x0	- <sup>(1)</sup>	N/S	-	0x0	-	0x1
<a href="#">GPMC_CONFIG1_i</a>	27	WRITETYPE	-	0x0	- <sup>(1)</sup>	N/S	-	0x1	-	0x1

<sup>(1)</sup> Multiple read is not supported in address/data- and AAD-multiplexed modes. Multiple read is supported in Nonmultiplexed mode.

**Table 15-255. Access Type Parameters Check List Table**

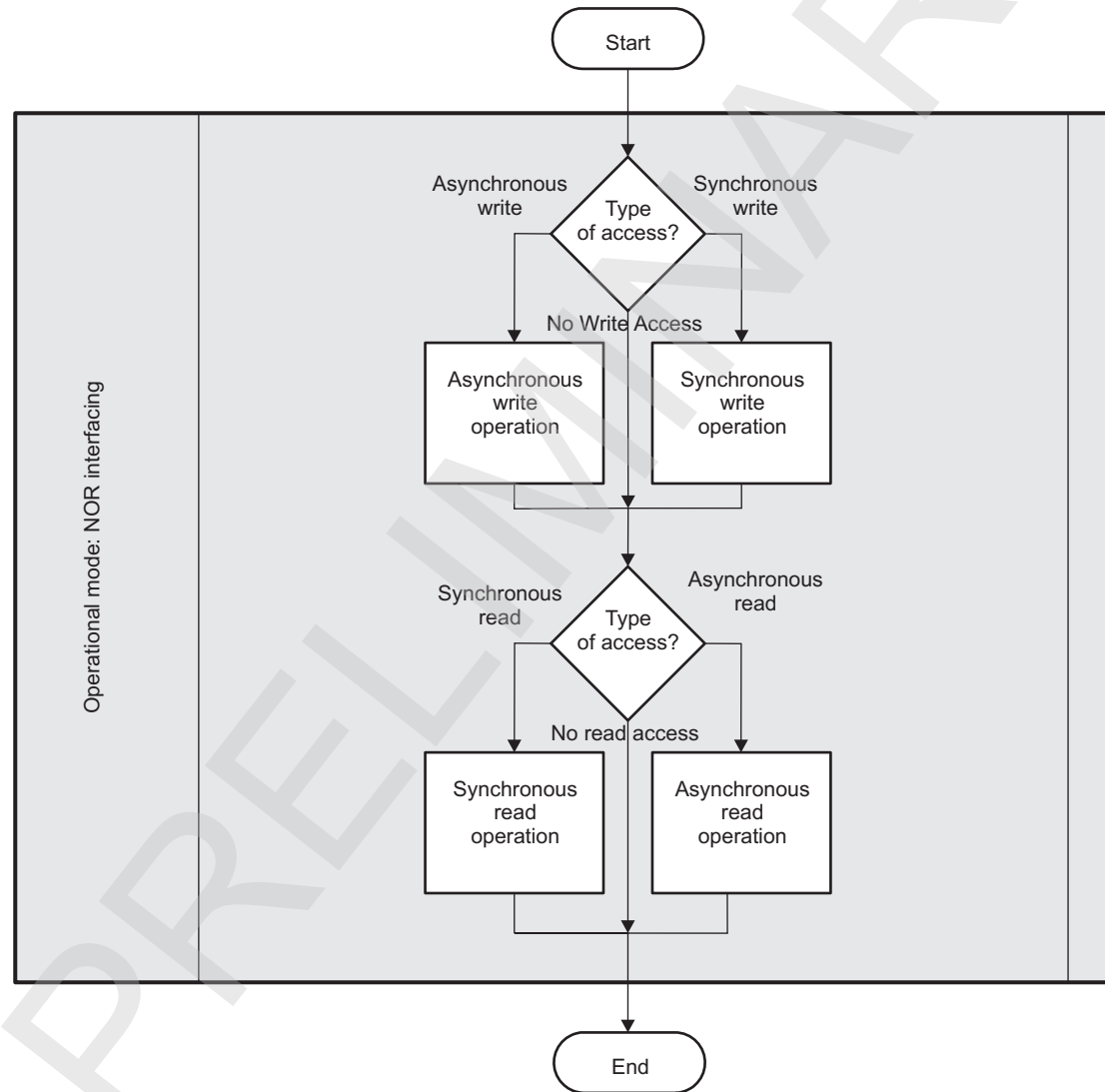
Register	Bit	Bit field name	Access Type		
			Non-mux	Address /Data Mux	AAD-mux
<a href="#">GPMC_CONFIG1_i</a>	9:8	MUXADDDATA	0x0	0x2	0x1
<a href="#">GPMC_CONFIG</a>	1	LIMITEDADDRESS <sup>(1)</sup>	0x1	0x0	0x0

<sup>(1)</sup> This parameter applies to all chip-selects. There is no per-CS configuration for this parameter.

15.4.5.6 GPMC Timing Parameters

Figure 15-92 shows a programming model diagram for the NOR interfacing timing parameters.

Figure 15-92. NOR Interfacing Timing Parameters Diagram



gpmc-uc-002

Table 15-256 lists bit fields to configure adequate timing parameter in various memory modes.

**Table 15-256. Timing Parameters**

Register	Bit	Bit field name	Asynchronous			Synchronous				Access Type		
			Single Read Access	Single Write Access	Multiple Read (Page) access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access	Non-multiplexed	Address/Data-multiplexed	AAD-multiplexed
GPMC_CONFIG1_j	9	MUXADDDATA	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_j	29	READTYPE	y		y	y		y		y	y	y
GPMC_CONFIG1_j	30	READMULTIPLE	y		y	y		y		y	y	y
GPMC_CONFIG1_j	27	WRITETYPE		y			y		y	y	y	y
GPMC_CONFIG1_j	28	WRITEMULTIPLE		y			y		y	y	y	y
GPMC_CONFIG1_j	31	WRAPBURST						y	y	y	y	y
GPMC_CONFIG1_j	26:25	CLKACTIVATIONTIME				y	y	y	y	y	y	y
GPMC_CONFIG1_j	19:18	WAITMONITORINGTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_j	4	TIMEPARAGRANULARITY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_j	20:16	CSWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG2_j	12:8	CSRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG2_j	7	CSEXTRADelay	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_j	3:0	CSONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_j	30:28	ADVAADMUXWROFFTIME		y			y		y			y
GPMC_CONFIG3_j	30:29	ADVAADMUXRDOFFTIME	y		y	y		y				y
GPMC_CONFIG3_j	6:4	ADVAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG3_j	20:16	ADVWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG3_j	12:8	ADVRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG3_j	7	ADVEXTRADelay	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_j	3:0	ADVONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG4_j	15:13	OEAADMUXOFFTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_j	6:4	OEAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_j	28:24	WEOFFTIME		y			y		y	y	y	y
GPMC_CONFIG4_j	23	WEEXTRADelay		y			y		y	y	y	y
GPMC_CONFIG4_j	19:16	WEONTIME		y			y		y	y	y	y
GPMC_CONFIG4_j	12:8	OEOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG4_j	7	OEEEXTRADelay	y		y	y		y		y	y	y
GPMC_CONFIG4_j	3:0	OEONTIME	y		y	y		y		y	y	y
GPMC_CONFIG5_j	27:24	PAGEBURSTACCESSTIME			y			y	y	y	y	y

**Table 15-256. Timing Parameters (continued)**

			Asynchronous			Synchronous				Access Type		
<a href="#">GPMC_CONFIG5_j</a>	20:16	RDACCESSTIME	y		y	y	y			y	y	y
<a href="#">GPMC_CONFIG5_j</a>	12:8	WRCYCLETIME		y			y		y	y	y	y
<a href="#">GPMC_CONFIG5_j</a>	4:0	RDCYCLETIME	y		y	y	y			y	y	y
<a href="#">GPMC_CONFIG6_j</a>	28:24	WRACCESSTIME		y			y		y	y	y	y
<a href="#">GPMC_CONFIG6_j</a>	19:16	WRDATAONADMUXBUS		y			y		y		y	y
<a href="#">GPMC_CONFIG6_j</a>	11:8	CYCLE2CYCLEDELAY	y	y	y	y	y	y	y	y	y	y
<a href="#">GPMC_CONFIG6_j</a>	7	CYCLE2CYCLESAMECSEN	y	y	y	y	y	y	y	y	y	y
<a href="#">GPMC_CONFIG6_j</a>	6	CYCLE2CYCLEDIFFCSEN	y	y	y	y	y	y	y	y	y	y
<a href="#">GPMC_CONFIG6_j</a>	3:0	BUSTURNAROUND	y	y	y	y	y	y	y	y	y	y
<a href="#">GPMC_CONFIG7_j</a>	6	CSVALID	y	y	y	y	y	y	y	y	y	y

PRELIMINARY

### 15.4.5.6.1 GPMC Timing Parameters Formulas

This section intends to help the user to calculate the GPMC timing bit fields values. Formulas are not listed exhaustively.

The section details:

- NAND Flash Interface Timing Parameters Formulas
- Synchronous NOR Flash Timing Parameters Formulas
- Asynchronous NOR Flash Timing Parameters Formulas

#### 15.4.5.6.1.1 NAND Flash Interface Timing Parameters Formulas

This section lists formulas to use in order to calculate NAND timing parameters. This is the case when `GPMC_CONFIG1_i[11:10] DEVICETYPE = 0x2`. [Table 15-257](#) details NAND timing parameters.

**Table 15-257. NAND Formulas Description Table**

Configuration Parameter	Unit	Description
A	ns	Pulse duration - gpmc_nwe valid time
B	ns	Delay time - gpmc_ncs valid to gpmc_nwe valid
C	ns	Delay time - gpmc_nbe0_cle/gpmc_nadv_ale high to gpmc_nwe valid
D	ns	Delay time - gpmc_ad[15:0] valid to gpmc_nwe valid
E	ns	Delay time - gpmc_nwe invalid to gpmc_ad[15:0] invalid
F	ns	Delay time - gpmc_nwe invalid to gpmc_nbe0_cle/gpmc_nadv_ale invalid
G	ns	Delay time - gpmc_nwe invalid to gpmc_ncs invalid
H	ns	Cycle time - Write cycle time
I	ns	Delay time - gpmc_ncs valid to gpmc_noe valid
J	ns	Setup time - gpmc_ad[15:0] valid to gpmc_noe invalid
K	ns	Pulse duration - gpmc_noe valid time
L	ns	Cycle time - Read cycle time
M	ns	Delay time - gpmc_noe invalid to gpmc_ncs invalid

The configuration parameters are calculated through the following formulas.

$$A = (WEOffTime - WEOnTime) * (TimeParaGranularity + 1) * GPMC\_FCLK \text{ period}$$

$$B = ((WEOnTime - CSOnTime) * (TimeParaGranularity + 1) + 0.5 * (WEEExtraDelay - CSEExtraDelay)) * GPMC\_FCLK \text{ period}$$

$$C = ((WEOnTime - ADVOnTime) * (TimeParaGranularity + 1) + 0.5 * (WEEExtraDelay - ADVExtraDelay)) * GPMC\_FCLK \text{ period}$$

$$D = (WEOnTime * (TimeParaGranularity + 1) + 0.5 * WEEExtraDelay) * GPMC\_FCLK \text{ period}$$

$$E = (WrCycleTime - WEOffTime * (TimeParaGranularity + 1) - 0.5 * WEEExtraDelay) * GPMC\_FCLK \text{ period}$$

$$F = (ADVWrOffTime - WEOffTime * (TimeParaGranularity + 1) + 0.5 * (ADVExtraDelay - WEEExtraDelay)) * GPMC\_FCLK \text{ period}$$

$$G = (CSWrOffTime - WEOffTime * (TimeParaGranularity + 1) + 0.5 * (CSEExtraDelay - WEEExtraDelay)) * GPMC\_FCLK \text{ period}$$

$$H = WrCycleTime * (1 + TimeParaGranularity) * GPMC\_FCLK \text{ period}$$

$$I = ((OEOnTime - CSOnTime) * (TimeParaGranularity + 1) + 0.5 * (OEEExtraDelay - CSEExtraDelay)) * GPMC\_FCLK \text{ period}$$

$$J = ((AccessTime - OEOffTime) * (TimeParaGranularity + 1) - 0.5 * OEEExtraDelay) * GPMC\_FCLK \text{ period}$$

$$K = (OEOffTime - OEOnTime) * (1 + TimeParaGranularity) * GPMC\_FCLK \text{ period}$$

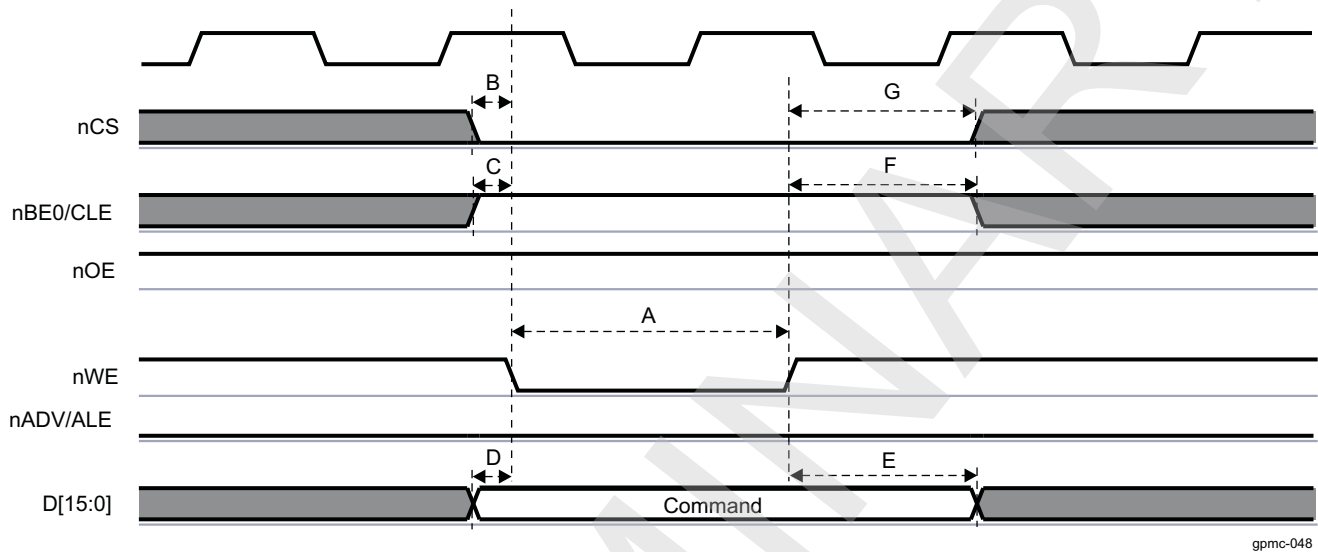
$$L = RdCycleTime * (1 + TimeParaGranularity) * GPMC\_FCLK \text{ period}$$



$$M = (\text{CSRdOffTime} - \text{OEOffTime} * (\text{TimeParaGranularity} + 1) + 0.5 * (\text{CSExtraDelay} - \text{OEEExtraDelay}) * \text{GPMC\_FCLK period}$$

Figure 15-93 shows a command latch cycle timing simplified example where formulas are associated with signal waves.

Figure 15-93. NAND Command Latch Cycle Timing Simplified Example



15.4.5.6.1.2 Synchronous NOR Flash Timing Parameters Formulas

This section lists all formulas to use in order to calculate synchronous NOR timing parameters. This is the case when GPMC\_CONFIG1\_[11:10] DEVICETYPE = 0x0 and when READTYPE or WRITETYPE are set to synchronous mode.

Table 15-258. Synchronous NOR Formulas Description Table

Configuration Parameter	Unit	Description
A	ns	Pulse duration - gpmc_ncs low
B	ns	Delay time - address bus valid to gpmc_clk first edge Delay time - gpmc_nbe0_cle/gpmc_nbe1 valid to gpmc_clk first edge
C	ns	Pulse duration - gpmc_nbe0_cle/gpmc_nbe1 low
D	ns	Delay time - gpmc_clk rising edge to gpmc_nbe0_cle/gpmc_nbe1 invalid Delay time - gpmc_clk rising edge to gpmc_nadv_ale invalid
E	ns	Delay time - gpmc_clk rising edge to gpmc_ncs invalid Delay time - gpmc_clk rising edge to gpmc_noe invalid
F	ns	Delay time - gpmc_clk rising edge to gpmc_ncs transition
G	ns	Delay time - gpmc_clk rising edge to gpmc_nadv_ale transition
H	ns	Delay time - gpmc_clk rising edge to gpmc_noe transition
I	ns	Delay time - gpmc_clk rising edge to gpmc_nwe transition
J	ns	Delay time - gpmc_clk rising edge to gpmc_ad data bus transition Delay time - gpmc_clk rising edge to gpmc_nbe0_cle/gpmc_nbe1 transition
K	ns	Pulse duration - gpmc_nadv_ale low
L	ns	Delay time - gpmc_wait invalid to first data latching gpmc_clk edge

The configuration parameters are calculated through the following formulas.

1. For single read accesses:
  - A = (CSRDOFFTIME - CSONTIME) \* (TIMEPARAGRANULARITY + 1) \* GPMC\_FCLK period
  - C = RDCYCLETIME \* (TIMEPARAGRANULARITY + 1) \* GPMC\_FCLK period
  - D = (RDCYCLETIME - ACCESSTIME) \* GPMC\_FCLK period
  - E = (CSRDOFFTIME - ACCESSTIME) \* GPMC\_FCLK period
2. For burst read accesses (where n is the page burst access number):
  - A = (CSRDOFFTIME - CSONTIME + (n - 1) \* PAGEBURSTACCESSTIME) \* (TIMEPARAGRANULARITY + 1) \* GPMC\_FCLK period
  - C = (RDCYCLETIME + (n - 1) \* PAGEBURSTACCESSTIME) \* (TIMEPARAGRANULARITY + 1) \* GPMC\_FCLK period
  - D = (RDCYCLETIME - (ACCESSTIME + (n - 1) \* PAGEBURSTACCESSTIME)) \* GPMC\_FCLK period
  - E = (CSRDOFFTIME - (ACCESSTIME + (n - 1) \* PAGEBURSTACCESSTIME)) \* GPMC\_FCLK period
3. For burst write accesses (where n is the page burst access number):
  - A = (CSWROFFTIME - CSONTIME + (n - 1) \* PAGEBURSTACCESSTIME) \* (TIMEPARAGRANULARITY + 1) \* GPMC\_FCLK period
  - C = (WRCYCLETIME + (n - 1) \* PAGEBURSTACCESSTIME) \* (TIMEPARAGRANULARITY + 1) \* GPMC\_FCLK period
  - D = (WRCYCLETIME - (ACCESSTIME + (n - 1) \* PAGEBURSTACCESSTIME)) \* GPMC\_FCLK period
  - E = (CSWROFFTIME - (ACCESSTIME + (n - 1) \* PAGEBURSTACCESSTIME)) \* GPMC\_FCLK period
4. For all accesses:
  - For nCS falling edge (CS activated):
    - Case where `GPMC_CONFIG1_j[1:0] GPMCFCLKDIVIDER = 0x0`:  
F = 0.5 \* CSEXTRADELAY \* GPMC\_FCLK period
    - Case where `GPMCFCLKDIVIDER = 0x1`:  
F = 0.5 \* CSEXTRADELAY \* GPMC\_FCLK period, when (CLKACTIVATIONTIME and CSONTIME are odd) or (CLKACTIVATIONTIME and CSONTIME are even)  
F = (1 + 0.5 \* CSEXTRADELAY) \* GPMC\_FCLK period otherwise.
    - Case where `GPMCFCLKDIVIDER = 0x2`:  
F = 0.5 \* CSEXTRADELAY \* GPMC\_FCLK period, when (CSONTIME - CLKACTIVATIONTIME) is a multiple of 3  
F = (1 + 0.5 \* CSEXTRADELAY) \* GPMC\_FCLK period, when (CSONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3  
F = (2 + 0.5 \* CSEXTRADELAY) \* GPMC\_FCLK period, when (CSONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3
  - For nCS rising edge (CS deactivated) in reading mode:
    - Case where `GPMC_CONFIG1_j[1:0] GPMCFCLKDIVIDER = 0x0`:  
F = 0.5 \* CSEXTRADELAY \* GPMC\_FCLK period
    - Case where `GPMCFCLKDIVIDER = 0x1`:  
F = 0.5 \* CSEXTRADELAY \* GPMC\_FCLK period, when (CLKACTIVATIONTIME and CSRDOFFTIME are odd) or (CLKACTIVATIONTIME and CSRDOFFTIME are even)  
F = (1 + 0.5 \* CSEXTRADELAY) \* GPMC\_FCLK period otherwise.
    - Case where `GPMCFCLKDIVIDER = 0x2`:  
F = 0.5 \* CSEXTRADELAY \* GPMC\_FCLK period, when (CSRDOFFTIME - CLKACTIVATIONTIME) is a multiple of 3  
F = (1 + 0.5 \* CSEXTRADELAY) \* GPMC\_FCLK period, when (CSRDOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3  
F = (2 + 0.5 \* CSEXTRADELAY) \* GPMC\_FCLK period, when (CSRDOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3
  - For nCS rising edge (CS deactivated) in writing mode:
    - Case where `GPMC_CONFIG1_j[1:0] GPMCFCLKDIVIDER = 0x0`:  
F = 0.5 \* CSEXTRADELAY \* GPMC\_FCLK period
    - Case where `GPMCFCLKDIVIDER = 0x1`:  
F = 0.5 \* CSEXTRADELAY \* GPMC\_FCLK period, when (CLKACTIVATIONTIME and CSWROFFTIME are odd) or (CLKACTIVATIONTIME and CSWROFFTIME are even)  
F = (1 + 0.5 \* CSEXTRADELAY) \* GPMC\_FCLK period otherwise.
    - Case where `GPMCFCLKDIVIDER = 0x2`:  
F = 0.5 \* CSEXTRADELAY \* GPMC\_FCLK period, when (CSWROFFTIME -

CLKACTIVATIONTIME) is a multiple of 3

$F = (1 + 0.5 * CSEXTRADELAY) * GPMC\_FCLK$  period, when (CSWROFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

$F = (2 + 0.5 * CSEXTRADELAY) * GPMC\_FCLK$  period, when (CSWROFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For nADV falling edge (nADV activated):

- Case where  $GPMC\_CONFIG1\_j[1:0]$  GPMCFCLKDIVIDER = 0x0:  
 $G = 0.5 * ADVEXTRADELAY * GPMC\_FCLK$  period
- Case where GPMCFCLKDIVIDER = 0x1:  
 $G = 0.5 * ADVEXTRADELAY * GPMC\_FCLK$  period, when (CLKACTIVATIONTIME and ADVONTIME are odd) or (CLKACTIVATIONTIME and ADVONTIME are even)  
 $G = (1 + 0.5 * ADVEXTRADELAY) * GPMC\_FCLK$  period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:  
 $G = 0.5 * ADVEXTRADELAY * GPMC\_FCLK$  period, when (ADVONTIME - CLKACTIVATIONTIME) is a multiple of 3  
 $G = (1 + 0.5 * ADVEXTRADELAY) * GPMC\_FCLK$  period, when (ADVONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3  
 $G = (2 + 0.5 * ADVEXTRADELAY) * GPMC\_FCLK$  period, when (ADVONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For nADV rising edge (nADV deactivated) in reading mode:

- Case where  $GPMC\_CONFIG1\_j[1:0]$  GPMCFCLKDIVIDER = 0x0:  
 $G = 0.5 * ADVEXTRADELAY * GPMC\_FCLK$  period
- Case where GPMCFCLKDIVIDER = 0x1:  
 $G = 0.5 * ADVEXTRADELAY * GPMC\_FCLK$  period, when (CLKACTIVATIONTIME and ADVRDOFFTIME are odd) or (CLKACTIVATIONTIME and ADVRDOFFTIME are even)  
 $G = (1 + 0.5 * ADVEXTRADELAY) * GPMC\_FCLK$  period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:  
 $G = 0.5 * ADVEXTRADELAY * GPMC\_FCLK$  period, when (ADVRDOFFTIME - CLKACTIVATIONTIME) is a multiple of 3  
 $G = (1 + 0.5 * ADVEXTRADELAY) * GPMC\_FCLK$  period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3  
 $G = (2 + 0.5 * ADVEXTRADELAY) * GPMC\_FCLK$  period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For nADV rising edge (nADV deactivated) in writing mode:

- Case where  $GPMC\_CONFIG1\_j[1:0]$  GPMCFCLKDIVIDER = 0x0:  
 $G = 0.5 * ADVEXTRADELAY * GPMC\_FCLK$  period
- Case where GPMCFCLKDIVIDER = 0x1:  
 $G = 0.5 * ADVEXTRADELAY * GPMC\_FCLK$  period, when (CLKACTIVATIONTIME and ADVWROFFTIME are odd) or (CLKACTIVATIONTIME and ADVWROFFTIME are even)  
 $G = (1 + 0.5 * ADVEXTRADELAY) * GPMC\_FCLK$  period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:  
 $G = 0.5 * ADVEXTRADELAY * GPMC\_FCLK$  period, when (ADVWROFFTIME - CLKACTIVATIONTIME) is a multiple of 3  
 $G = (1 + 0.5 * ADVEXTRADELAY) * GPMC\_FCLK$  period, when (ADVWROFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3  
 $G = (2 + 0.5 * ADVEXTRADELAY) * GPMC\_FCLK$  period, when (ADVWROFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

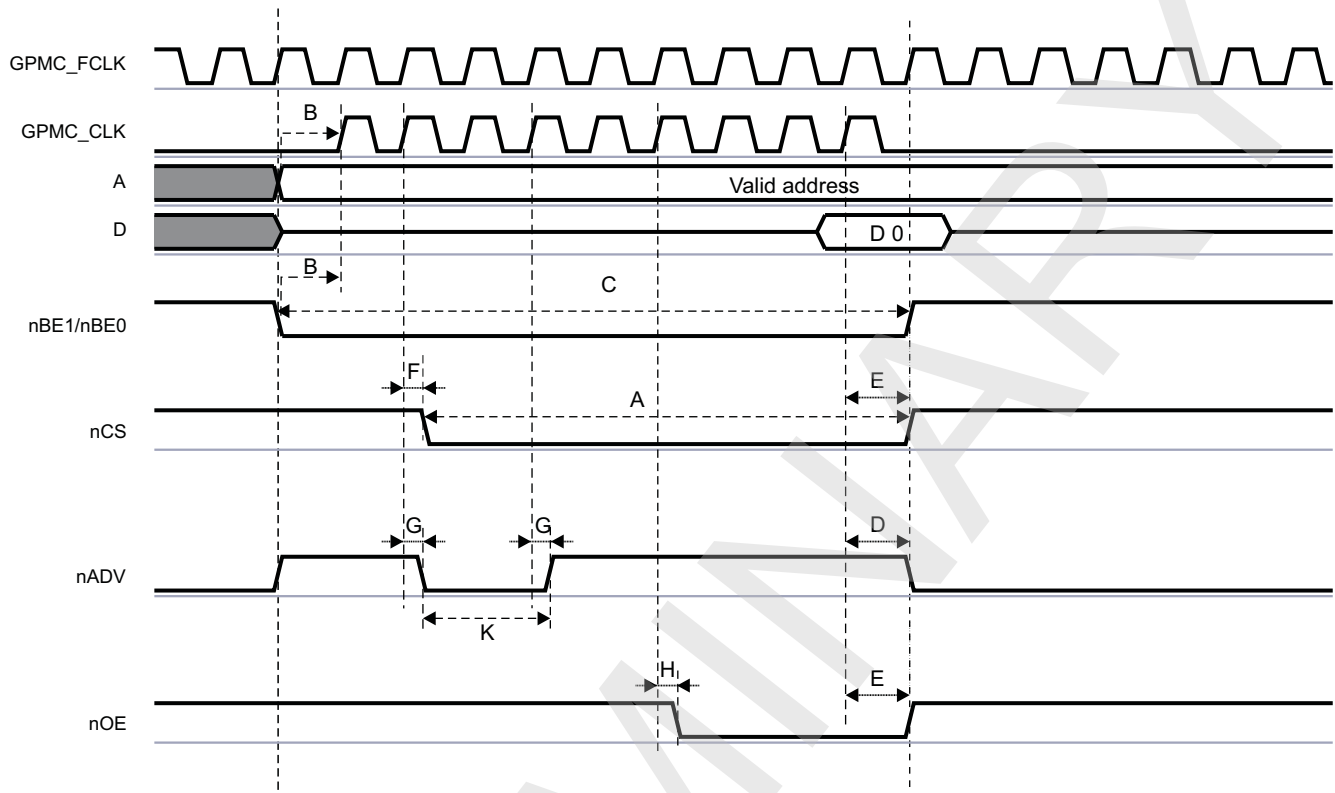
For nOE falling edge (nOE activated):

- Case where  $GPMC\_CONFIG1\_j[1:0]$  GPMCFCLKDIVIDER = 0x0:  
 $H = 0.5 * OEEXTRADELAY * GPMC\_FCLK$  period
- Case where GPMCFCLKDIVIDER = 0x1:  
 $H = 0.5 * OEEXTRADELAY * GPMC\_FCLK$  period, when (CLKACTIVATIONTIME and OEONTIME are odd) or (CLKACTIVATIONTIME and OEONTIME are even)  
 $H = (1 + 0.5 * OEEXTRADELAY) * GPMC\_FCLK$  period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:  
 $H = 0.5 * OEEXTRADELAY * GPMC\_FCLK$  period, when (OEONTIME - CLKACTIVATIONTIME) is a multiple of 3  
 $H = (1 + 0.5 * OEEXTRADELAY) * GPMC\_FCLK$  period, when (OEONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

- $H = (2 + 0.5 * OEEXTRADELAY) * GPMC\_FCLK$  period, when  $(OEONTIME - CLKACTIVATIONTIME - 2)$  is a multiple of 3
- For nOE rising edge (nOE deactivated):
- Case where  $GPMC\_CONFIG1\_j[1:0]$  GPMCFCLKDIVIDER = 0x0:  
 $H = 0.5 * OEEXTRADELAY * GPMC\_FCLK$  period
  - Case where GPMCFCLKDIVIDER = 0x1:  
 $H = 0.5 * OEEXTRADELAY * GPMC\_FCLK$  period, when  $(CLKACTIVATIONTIME$  and  $OEOFFTIME$  are odd) or  $(CLKACTIVATIONTIME$  and  $OEOFFTIME$  are even)  
 $H = (1 + 0.5 * OEEXTRADELAY) * GPMC\_FCLK$  period otherwise.
  - Case where GPMCFCLKDIVIDER = 0x2:  
 $H = 0.5 * OEEXTRADELAY * GPMC\_FCLK$  period, when  $(OEOFFTIME - CLKACTIVATIONTIME)$  is a multiple of 3  
 $H = (1 + 0.5 * OEEXTRADELAY) * GPMC\_FCLK$  period, when  $(OEOFFTIME - CLKACTIVATIONTIME - 1)$  is a multiple of 3  
 $H = (2 + 0.5 * OEEXTRADELAY) * GPMC\_FCLK$  period, when  $(OEOFFTIME - CLKACTIVATIONTIME - 2)$  is a multiple of 3
- For nWE falling edge (nWE activated):
- Case where  $GPMC\_CONFIG1\_j[1:0]$  GPMCFCLKDIVIDER = 0x0:  
 $I = 0.5 * WEEXTRADELAY * GPMC\_FCLK$  period
  - Case where GPMCFCLKDIVIDER = 0x1:  
 $I = 0.5 * WEEXTRADELAY * GPMC\_FCLK$  period, when  $(CLKACTIVATIONTIME$  and  $WEONTIME$  are odd) or  $(CLKACTIVATIONTIME$  and  $WEONTIME$  are even)  
 $I = (1 + 0.5 * WEEXTRADELAY) * GPMC\_FCLK$  period otherwise.
  - Case where GPMCFCLKDIVIDER = 0x2:  
 $I = 0.5 * WEEXTRADELAY * GPMC\_FCLK$  period, when  $(WEONTIME - CLKACTIVATIONTIME)$  is a multiple of 3  
 $I = (1 + 0.5 * WEEXTRADELAY) * GPMC\_FCLK$  period, when  $(WEONTIME - CLKACTIVATIONTIME - 1)$  is a multiple of 3  
 $I = (2 + 0.5 * WEEXTRADELAY) * GPMC\_FCLK$  period, when  $(WEONTIME - CLKACTIVATIONTIME - 2)$  is a multiple of 3
- For nWE rising edge (nWE deactivated):
- Case where  $GPMC\_CONFIG1\_j[1:0]$  GPMCFCLKDIVIDER = 0x0:  
 $I = 0.5 * WEEXTRADELAY * GPMC\_FCLK$  period
  - Case where GPMCFCLKDIVIDER = 0x1:  
 $I = 0.5 * WEEXTRADELAY * GPMC\_FCLK$  period, when  $(CLKACTIVATIONTIME$  and  $WEOFFTIME$  are odd) or  $(CLKACTIVATIONTIME$  and  $WEOFFTIME$  are even)  
 $I = (1 + 0.5 * WEEXTRADELAY) * GPMC\_FCLK$  period otherwise.
  - Case where GPMCFCLKDIVIDER = 0x2:  
 $I = 0.5 * WEEXTRADELAY * GPMC\_FCLK$  period, when  $(WEOFFTIME - CLKACTIVATIONTIME)$  is a multiple of 3  
 $I = (1 + 0.5 * WEEXTRADELAY) * GPMC\_FCLK$  period, when  $(WEOFFTIME - CLKACTIVATIONTIME - 1)$  is a multiple of 3  
 $I = (2 + 0.5 * WEEXTRADELAY) * GPMC\_FCLK$  period, when  $(WEOFFTIME - CLKACTIVATIONTIME - 2)$  is a multiple of 3
- For gpmc\_nadv low pulse duration:
- Read operation:  
 $K = (ADVRDOFFTIME - ADVONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC\_FCLK$  period
  - Write operation:  
 $K = (ADVWROFFTIME - ADVONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC\_FCLK$  period
- For gpmc\_wait invalid to first data latching gpmc\_clk edge:
- $L = WAITMONITORINGTIME * (GPMCFCLKDIVIDER + 1) * GPMC\_FCLK$  period +  $GPMC\_CLK$  period

Figure 15-94 shows a synchronous NOR single read simplified example where formulas are associated with signal waves.

Figure 15-94. Synchronous NOR Single Read Simplified Example



gpmc-046

15.4.5.6.1.3 Asynchronous NOR Flash Timing Parameters Formulas

This section lists all formulas to use in order to calculate asynchronous NOR timing parameters. This is the case when GPMC\_CONFIG1\_i[11:10] DEVICETYPE = 0x0 and when READTYPE or WRITETYPE are set to asynchronous mode.

Table 15-259. Asynchronous NOR Formulas Description Table

Configuration Parameter	Unit	Description
A	ns	Pulse duration - gpmc_ncs low
B	ns	Delay time - gpmc_ncs valid to gpmc_nadv_ale invalid
C	ns	Delay time - gpmc_ncs valid to gpmc_noe invalid (single read)
D	ns	Pulse duration - address bus valid - 2nd, 3rd and 4th accesses
E	ns	Delay time - gpmc_ncs valid to gpmc_nwe valid
F	ns	Delay time - gpmc_ncs valid to gpmc_nwe invalid
G	ns	Address invalid duration between 2 successive R/W accesses
H	ns	Setup time - read data valid before gpmc_noe high
I	ns	Delay time - gpmc_ncs valid to gpmc_noe invalid (burst read)
J	ns	Delay time - address bus valid to gpmc_ncs valid Delay time - data bus valid to gpmc_ncs valid Delay time - gpmc_nbe0_cle/gpmc_nbe1 valid to gpmc_ncs valid
K	ns	Delay time - gpmc_ncs valid to gpmc_nadv_ale valid
L	ns	Delay time - gpmc_ncs valid to gpmc_noe valid
M	ns	Delay time - gpmc_ncs valid to first data latching edge
N	ns	Pulse duration - gpmc_nbe0_cle/gpmc_nbe1 valid time

**Table 15-259. Asynchronous NOR Formulas Description Table (continued)**

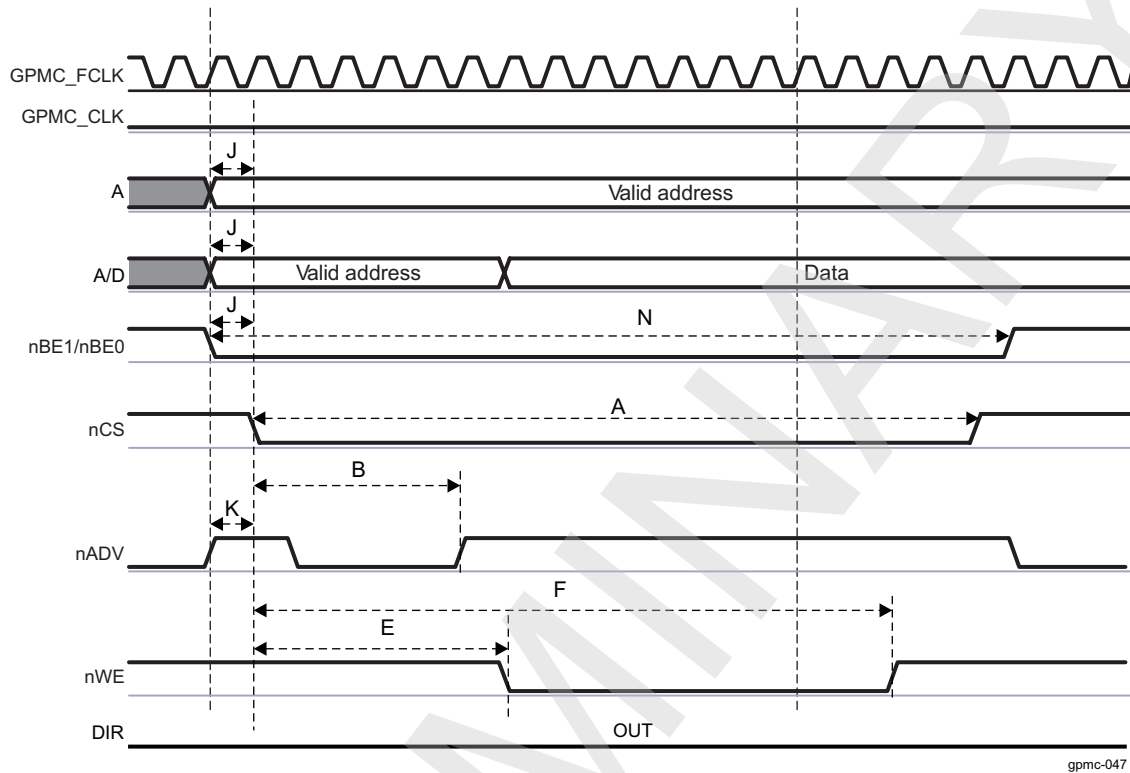
Configuration Parameter	Unit	Description
O	ns	Delay time - gpmc_ncs valid to gpmc_nadv_ale valid

The configuration parameters are calculated through the following formulas. Note that these formulas are not exhaustive.

- gpmc\_ncs low pulse:  
For single read:  $A = (\text{CSRDOFFTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) * \text{GPMC\_FCLK period}$   
For burst read:  $A = (\text{CSRDOFFTIME} - \text{CSONTIME} + (\text{N} - 1) * \text{PAGEBURSTACCESSTIME}) * (\text{TIMEPARAGRANULARITY} + 1) * \text{GPMC\_FCLK period}$ , where N = page burst access number  
For single write:  $A = (\text{CSWROFFTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) * \text{GPMC\_FCLK period}$   
For burst write:  $A = (\text{CSWROFFTIME} - \text{CSONTIME} + (\text{N} - 1) * \text{PAGEBURSTACCESSTIME}) * (\text{TIMEPARAGRANULARITY} + 1) * \text{GPMC\_FCLK period}$ , where N = page burst access number
- gpmc\_ncs valid to gpmc\_nadv\_ale invalid delay:  
For reading:  $B = ((\text{ADVRDOFFTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * (\text{ADVEXTRADELAY} - \text{CSEXTRADELAY})) * \text{GPMC\_FCLK period}$   
For writing:  $B = ((\text{ADVWROFFTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * (\text{ADVEXTRADELAY} - \text{CSEXTRADELAY})) * \text{GPMC\_FCLK period}$
- $C = ((\text{OEOFFTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * (\text{OEEXTRADELAY} - \text{CSEXTRADELAY})) * \text{GPMC\_FCLK period}$
- $D = \text{PAGEBURSTACCESSTIME} * (\text{TIMEPARAGRANULARITY} + 1) * \text{GPMC\_FCLK period}$
- $E = ((\text{WEONTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * (\text{WEEXTRADELAY} - \text{CSEXTRADELAY})) * \text{GPMC\_FCLK period}$
- $F = ((\text{WEOFFTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * (\text{WEEXTRADELAY} - \text{CSEXTRADELAY})) * \text{GPMC\_FCLK period}$
- $G = \text{CYCLE2CYCLEDELAY} * \text{GPMC\_FCLK period}$
- $H = ((\text{OEOFFTIME} - \text{ACCESSTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * \text{OEEXTRADELAY}) * \text{GPMC\_FCLK period}$
- $I = ((\text{OEOFFTIME} + (\text{N} - 1) * \text{PAGEBURSTACCESSTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * (\text{OEEXTRADELAY} - \text{CSEXTRADELAY})) * \text{GPMC\_FCLK period}$ , where N = page burst access number
- $J = (\text{CSONTIME} * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * \text{CSEXTRADELAY}) * \text{GPMC\_FCLK period}$
- $K = ((\text{ADVONTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * (\text{ADVEXTRADELAY} - \text{CSEXTRADELAY})) * \text{GPMC\_FCLK period}$
- $L = ((\text{OEONTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * (\text{OEEXTRADELAY} - \text{CSEXTRADELAY})) * \text{GPMC\_FCLK period}$
- $M = ((\text{ACCESSTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) - 0.5 * \text{CSEXTRADELAY}) * \text{GPMC\_FCLK period}$
- gpmc\_nbe0\_cle/gpmc\_nbe1 pulse:  
For single read:  $N = \text{RDCYCLETIME} * (\text{TIMEPARAGRANULARITY} + 1) * \text{GPMC\_FCLK period}$   
For burst read:  $N = (\text{RDCYCLETIME} + (\text{N} - 1) * \text{PAGEBURSTACCESSTIME}) * (\text{TIMEPARAGRANULARITY} + 1) * \text{GPMC\_FCLK period}$ , where N = page burst access number  
For burst write:  $N = (\text{WRCYCLETIME} + (\text{N} - 1) * \text{PAGEBURSTACCESSTIME}) * (\text{TIMEPARAGRANULARITY} + 1) * \text{GPMC\_FCLK period}$ , where N = page burst access number
- $O = ((\text{WRCYCLETIME} + (\text{N} - 1) * \text{PAGEBURSTACCESSTIME} - \text{CSONTIME}) * (\text{TIMEPARAGRANULARITY} + 1) + 0.5 * (\text{ADVEXTRADELAY} - \text{CSEXTRADELAY})) * \text{GPMC\_FCLK period}$

Figure 15-95 shows an asynchronous NOR single write simplified example where formulas are associated with signal waves.

**Figure 15-95. Asynchronous NOR Single Write Simplified Example**



**NOTE:** Write multiple access is not supported in asynchronous mode. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

## 15.4.6 GPMC Use Cases And Tips

### 15.4.6.1 How to Set GPMC Timing Parameters for Typical Accesses

#### 15.4.6.1.1 External Memory Attached to the GPMC Module

As discussed in the introduction to this chapter, the GPMC module supports the following external memory types:

- Asynchronous or synchronous, 8-bit or 16-bit-width memory or device
- 16-bit address/data-multiplexed or not multiplexed NOR flash device
- 8- or 16-bit NAND flash device

The following examples show how to calculate GPMC timing parameters by showing a typical parameter setup for the access to be performed.

The example is based on a 512-Mb multiplexed NOR flash memory with the following characteristics:

- Type: NOR flash (address/data-multiplexed mode)
- Size: 512M bits
- Data Bus: 16 bits wide
- Speed: 104 MHz clock frequency
- Read access time: 80 ns

#### 15.4.6.1.2 Typical GPMC Setup

Table 15-260 lists some of the I/Os of the GPMC module.

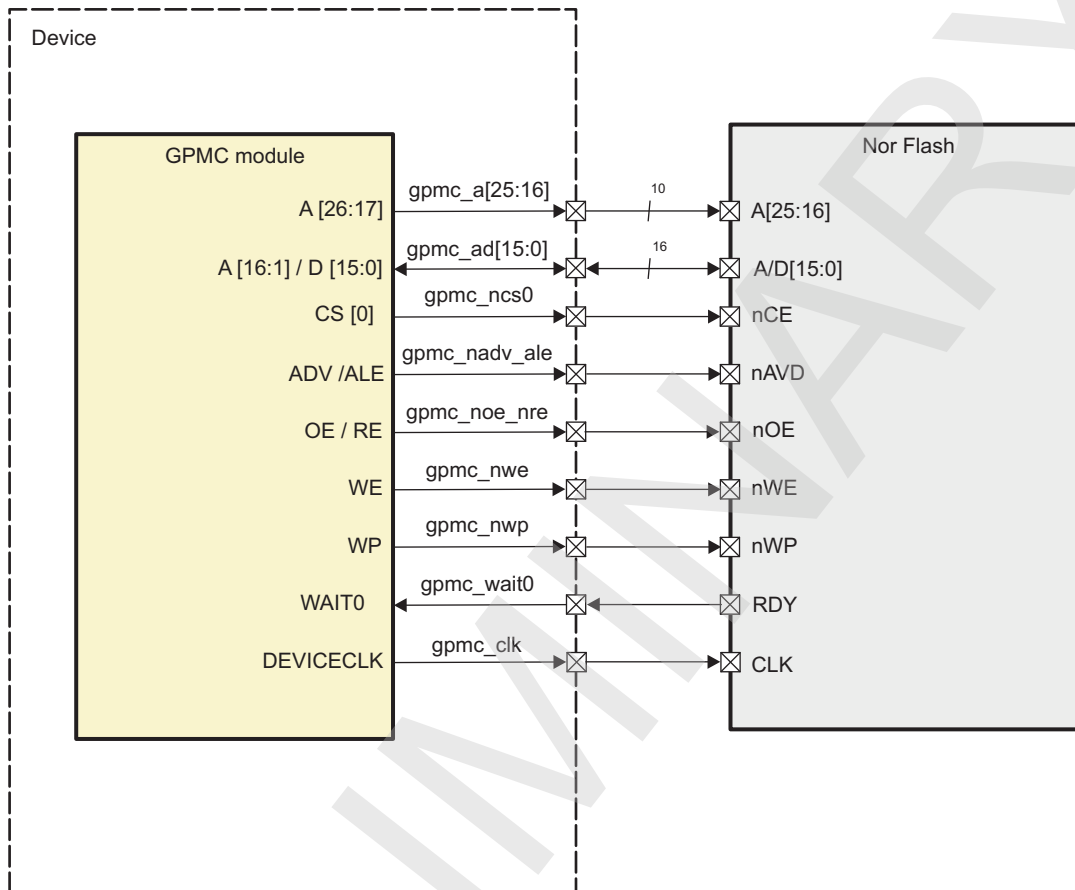
**Table 15-260. GPMC Signals**

Signal Name	I/O	Description
GPMC_FCLK	Internal	Functional and interface clock. Acts as the time reference.
gpmc_clk	O	External clock provided to the external device for synchronous operations
gpmc_a[25:16]	O	Address
gpmc_ad[15: 0]	I/O	Data-multiplexed with addresses A[16:1] on memory side
gpmc_ncsx	O	Chip-select (where x = 0, or 1)
gpmc_nadv_ale	O	Address valid enable
gpmc_noe_nre	O	Output enable (read access only)
gpmc_nwe	O	Write enable (write access only)
gpmc_wait[2:0]	I	Ready signal from memory device. Indicates when valid burst data is ready to be read



Figure 15-96 shows the typical connection between the GPMC module and an attached NOR Flash memory.

Figure 15-96. GPMC Connection to an External NOR Flash Memory



gpmc-037

The following sections demonstrate how to calculate GPMC parameters for three access types:

- Synchronous burst read
- Asynchronous read
- Asynchronous single write

15.4.6.1.2.1 GPMC Configuration for Synchronous Burst Read Access

The clock runs at 104 MHz ( f = 104 MHz; T = 9, 615 ns).

Table 15-261 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 15-262 shows how to calculate timings for the GPMC using the memory parameters.

Figure 15-97 shows the synchronous burst read access.

Table 15-261. Useful Timing Parameters on the Memory Side

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCES	nCS setup time to clock	0
tACS	Address setup time to clock	3
tIACC	Synchronous access time	80

**Table 15-261. Useful Timing Parameters on the Memory Side (continued)**

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tBACC	Burst access time valid clock to output delay	5,2
tCEZ	Chip-select to High-Z	7
tOEZ	Output enable to High-Z	7
tAVC	nADV setup time	6
tAVD	nAVD pulse	6
tACH	Address hold time from clock	3

The following terms, which describe the timing interface between the controller and its attached device, are used to calculate the timing parameters on the GPMC side:

- Read Access time (GPMC side): Time required to activate the clock + read access time requested on the memory side + data setup time required for optimal capture of a burst of data
- Data setup time (GPMC side): Ensures a good capture of a burst of data (as opposed to taking a burst of data out). One word of data is processed in one clock cycle ( $T = 9,615$  ns). The read access time between 2 bursts of data is tBACC = 5,2 ns. Therefore, data setup time is a clock period - tBACC = 4,415 ns of data setup.
- Access completion (GPMC side): (Different from page burst access time) Time required between the last burst access and access completion: nCS/nOE hold time (nCS and nOE must be released at the end of an access. These signals are held to allow the access to complete).
- Read cycle time (GPMC side): Read Access time + access completion
- Write cycle time for burst access: Not supported for NOR flash memory

**Table 15-262. Calculating GPMC Timing Parameters**

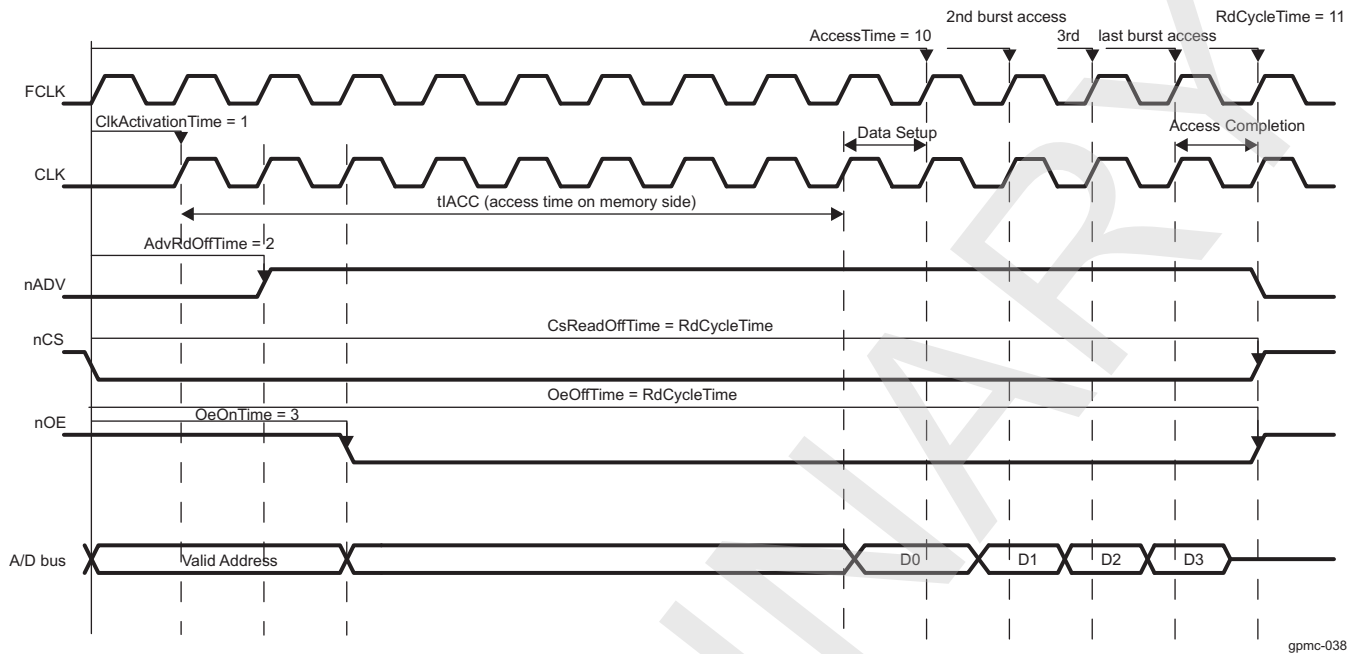
Parameter Name on GPMC Side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
GPMC FCLK Divider	-	-	-	GPMCFCLKDIVIDER = 0x0
ClkActivation Time	min ( tCES, tACS)	3	1	CLKACTIVATIONTIME = 0x1
RdAccessTime	roundmax (ClkActivationTime + tIACC + DataSetupTime)	94,03: (9,615 + 80 + 4,415)	10 : roundmax (94,03 / 9,615)	ACCESSTIME = 0x0A
PageBurst AccessTime	roundmax (tBACC)	roundmax (5,2)	1	PAGEBURSTACCESSTIME = 0x1
RdCycleTime	AccessTime + max ( tCEZ, tOEZ)	101, 03: (94, 03 + 7)	11	RDCYCLETIME = 0x0B
CsOnTime	tCES	0	0	CSONTIME = 0x0
CsReadOffTime	RdCycleTime	-	11	CSRDOFFTIME = 0x0B
AdvOnTime	tAVC <sup>(1)</sup>	0	0	ADVONTIME = 0x0
AdvRdOffTime	tAVD + tAVC <sup>(2)</sup>	12	2	ADVRDOFFTIME = 0x02
OeOnTime <sup>(3)</sup>	(ClkActivationTime + tACH) < OeOnTime < (ClkActivationTime + tIACC)	-	3 for instance.	OEONTIME = 0x3
OeOffTime	RdCycleTime	-	11	OEOFFTIME = 0x0B

<sup>(1)</sup> The external clock provided to the NOR flash is not yet available.

<sup>(2)</sup> AdvRdOffTime - AdvOnTime = tAVD; thus, AdvRdOffTime = tAVD + AdvOnTime = tAVD + tAVC.

<sup>(3)</sup> OeOnTime must guarantee that addresses are available. It must not exceed the availability of the first burst of data.

**Figure 15-97. Synchronous Burst Read Access (Timing Parameters in Clock Cycles)**



**15.4.6.1.2.2 GPMC Configuration for Asynchronous Read Access**

The clock runs at 104 MHz ( f = 104 MHz; T = 9, 615 ns).

Table 15-263 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 15-264 shows how to calculate timings for the GPMC using the memory parameters.

Figure 15-98 shows the asynchronous read access.

**Table 15-263. AC Characteristics for Asynchronous Read Access**

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCE	Read Access time from nCS low	80
tAAVDS	Address setup time to rising edge of nADV	3
tAVDP	nADV low time	6
tCAS	nCS setup time to nADV	0
tOE	Output enable to output valid	6
tOEZ	Output enable to High-Z	7

Use the following formula to calculate the RdCycleTime parameter for this typical access:

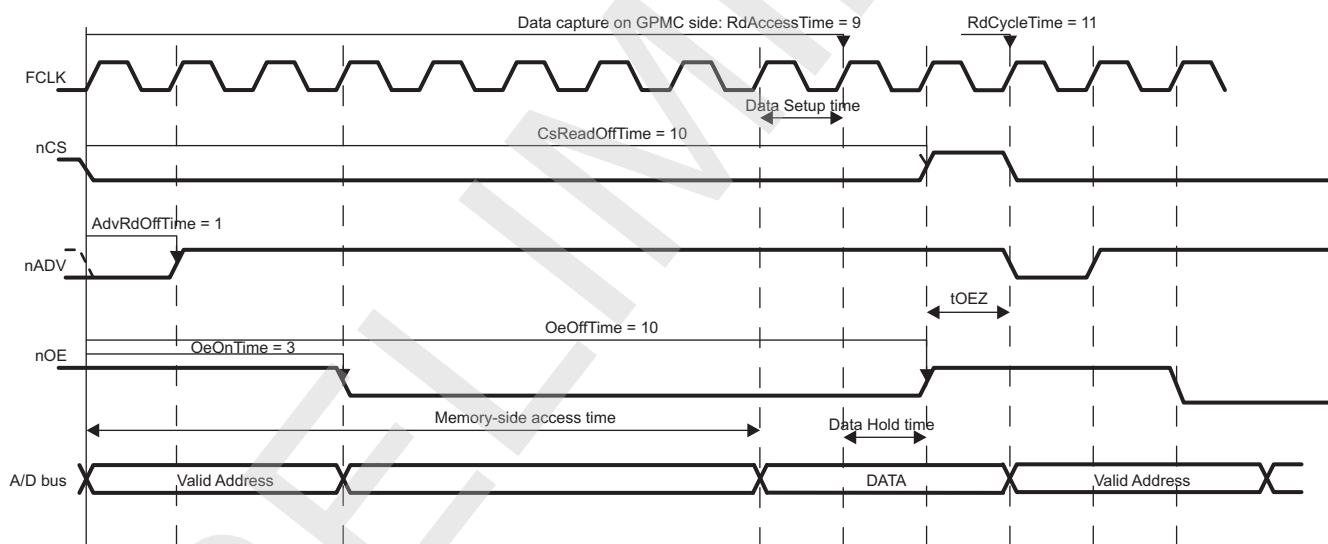
$$RdCycleTime = RdAccessTime + AccessCompletion = RdAccessTime + 1 \text{ clock cycle} + tOEZ:$$

- First, on the memory side, the external memory makes the data available to the output bus. This is the memory-side read access time defined in Table 15-263: the number of clock cycles between the address capture (nADV rising edge) and the data valid on the output bus. The GPMC requires some hold time to allow the data to be captured correctly and the access to be finished.
- To read the data correctly, the GPMC must be configured to meet the data setup time requirement of the memory; the GPMC module captures the data on the next rising edge. This is access time on the GPMC side.
- There must also be a data hold time for correctly reading the data (checking that there is no nOE/nCS deassertion while reading the data). This data hold time is 1 clock cycle (that is, AccessTime + 1).

- To complete the access, nOE/nCS signals are driven to High-Z. AccessTime + 1 + tOEZ is the read cycle time.
- Addresses can now be relatched and a new read cycle begun.

**Table 15-264. GPMC Timing Parameters for Asynchronous Read Access**

Parameter Name on GPMC side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
ClkActivationTime		n/a (asynchronous mode)		
AccessTime	round max (tCE)	80	9	ACCESSTIME = 0x09
PageBurstAccess Time	n/a (single access)			
RdCycleTime	AccessTime + 1 cycle + tOEZ	96, 615	11	RDCYCLETIME = 0x0B
CsOnTime	tCAS	0	0	CSONTIME = 0x0
CsReadOffTime	AccessTime + 1 cycle	89, 615	10	CSRDOFFTIME = 0x0A
AdvOnTime	tAAVDS	3	1	ADVONTIME = 0x1
AdvRdOffTime	tAAVDS + tAVDP	9	1	ADVRDOFFTIME = 0x01
OeOnTime	OeOnTime >= AdvRdOffTime (multiplexed mode)	-	3 for instance	OEONTIME = 0x3
OeOffTime	AccessTime + 1 cycle	89, 615	10	OEOFFTIME = 0x0A

**Figure 15-98. Asynchronous Single Read Access (Timing Parameters in Clock Cycles)**

gpmc-039

#### 15.4.6.1.2.3 GPMC Configuration for Asynchronous Single Write Access

The clock runs at 104 MHz: (f = 104 MHz; T = 9, 615 ns).

Table 15-266 shows how to calculate timings for the GPMC using the memory parameters.

Table 15-265 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Figure 15-99 shows the synchronous burst write access.

**Table 15-265. AC Characteristics for Asynchronous Single Write ( Memory Side)**

AC Characteristics on the Memory Side	Description	Duration (ns)
tWC	Write cycle time	60
tAVDP	nADV low time	6

**Table 15-265. AC Characteristics for Asynchronous Single Write ( Memory Side) (continued)**

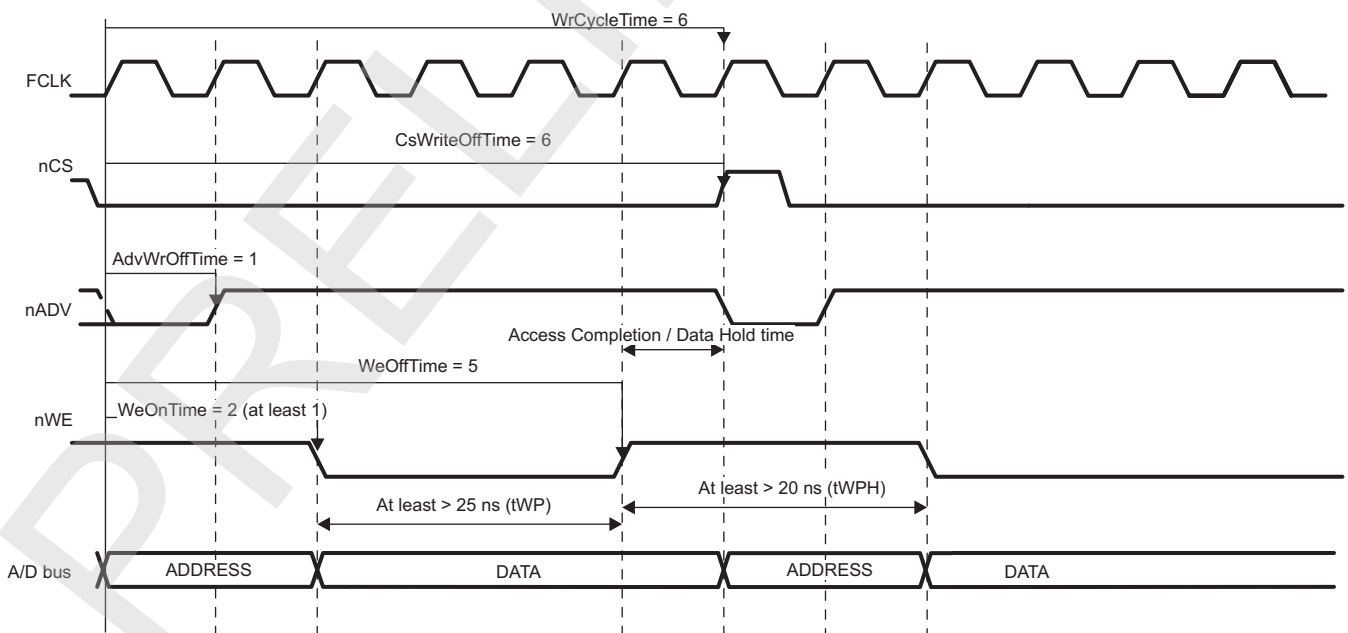
AC Characteristics on the Memory Side	Description	Duration (ns)
tWP	Write pulse width	25
tWPH	Write pulse width high	20
tCS	nCS setup time to nWE	3
tCAS	nCS setup time to nADV	0
tAVSC	nADV setup time	3

For asynchronous single write access, write cycle time is  $WrCycleTime = WeOffTime + AccessCompletion = WeOffTime + 1$ . For the AccessCompletion, the GPMC requires 1 cycle of data hold time (nCS deassertion).

**Table 15-266. GPMC Timing parameters for Asynchronous Single Write**

Parameter Name on GPMC side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Registers Configuration
ClkActivationTime		n/a (asynchronous mode)		
AccessTime	Applicable only to WAITMONITORING (the value is the same as for read access)			
PageBurstAccessTime		n/a (single access)		
WrCycleTime	$WeOffTime + AccessCompletion$	57, 615	6	WRCYCLETIME = 0x06
CsOnTime	tCAS	0	0	CSONTIME = 0x0
CsWrOffTime	$WeOffTime + 1$	57, 615	6	CSWROFFTIME = 0x06
AdvOnTime	tAVSC	3	1	ADVONTIME = 0x1
AdvWrOffTime	$tAVSC + tAVDP$	9	1	ADVWROFFTIME = 0x01
WeOnTime	tCS	3	1	WEONTIME = 0x1
WeOffTime	$tCS + tWP + tWPH$	48	5	WEOFFTIME = 0x05

**Figure 15-99. Asynchronous Single Write Access (Timing Parameters in Clock Cycles)**



gpmc-040

### 15.4.6.2 How to choose a suitable memory to use with the GPMC

This section is intended to help the user select a suitable memory device to interface with the GPMC controller.

#### 15.4.6.2.1 Supported Memories or Devices

NAND flash and NOR flash architectures are the two flash technologies. The GPMC supports various types of external memory or device, basically any one that supports NAND or NOR protocols:

- 8- and 16-bit width asynchronous or synchronous memory or device (8-bit: non burst device only)
- 16-bit address and data multiplexed NOR flash devices (pSRAM, OneNAND™, ...)
- 8- and 16-bit NAND flash device

**NOTE:** Non-multiplexed NOR flash devices are supported by the GPMC but their usage is highly limited. As only ten address pins are available on the GPMC interface, the maximum device size supported is 2KB.

##### 15.4.6.2.1.1 Memory Pin Multiplexing

This section highlights the interfacing differences of the GPMC supported memories.

**Table 15-267. Supported memories interfaces**

Function	16-bit Address/Data muxed pSRAM or NOR Flash <sup>(1)</sup>	OneNAND	16-bit NAND	8-bit NAND
gpmc_a[25]	A26			
gpmc_a[24]	A25			
gpmc_a[23]	A24			
gpmc_a[22]	A23			
gpmc_a[21]	A22			
gpmc_a[20]	A21			
gpmc_a[19]	A20			
gpmc_a[18]	A19			
gpmc_a[17]	A18			
gpmc_a[16]	A17			
gpmc_ad[15]	D15 or A16		IO15	
gpmc_ad[14]	D14 or A15		IO14	
gpmc_ad[13]	D13 or A14		IO13	
gpmc_ad[12]	D12 or A13		IO12	
gpmc_ad[11]	D11 or A12		IO11	
gpmc_ad[10]	D10 or A11		IO10	
gpmc_ad[9]	D9 or A10		IO9	
gpmc_ad[8]	D8 or A9		IO8	
gpmc_ad[7]	D7 or A8			IO7
gpmc_ad[6]	D6 or A7			IO6
gpmc_ad[5]	D5 or A6			IO5
gpmc_ad[4]	D4 or A5			IO4
gpmc_ad[3]	D3 or A4			IO3
gpmc_ad[2]	D2 or A3			IO2
gpmc_ad[1]	D1 or A2			IO1
gpmc_ad[0]	D0 or A1			IO0
gpmc_clk	CLK			

<sup>(1)</sup> Addresses seen from the device side. When interfacing to the external IC, A1 is connected to the memory A0, A2 to the memory A1, and so on...

**Table 15-267. Supported memories interfaces (continued)**

Function	16-bit Address/Data muxed pSRAM or NOR Flash <sup>(1)</sup>	OneNAND	16-bit NAND	8-bit NAND
gpmc_ncs0	nCS0 (Chip Select)		nCE0 (Chip Enable)	
gpmc_ncs1	nCS1		nCE1	
gpmc_ncs2	nCS2		nCE2	
gpmc_ncs3	nCS3		nCE3	
gpmc_ncs4	nCS4		nCE4	
gpmc_ncs5	nCS5		nCE5	
gpmc_ncs6	nCS6		nCE6	
gpmc_ncs7	nCS7		nCE7	
gpmc_nadv_ale	nADV (Address Valid)		ALE (Address Latch Enable)	
gpmc_noe	nOE (Output Enable)		nRE (Read Enable)	
gpmc_nwe	nWE (Write Enable)		nWE (Write Enable)	
gpmc_nbe0_cle	nBE0 (Byte Enable)		CLE (Command Latch Enable)	
gpmc_nbe1	nBE1			
gpmc_nwp	nWP (Write Protect)		nWP (Write Protect)	
gpmc_wait0	WAIT0		R/nB0 (Ready/Busy)	
gpmc_wait1	WAIT1		R/nB1	
gpmc_wait2	WAIT2		R/nB2	

#### 15.4.6.2.1.2 NAND Interface Protocol

NAND flash architecture, introduced in 1989, is a flash technology. NAND is a page-oriented memory device, that is, read and write accesses are done by pages. NAND achieves great density by sharing common areas of the storage transistor, which creates strings of serially connected transistors (in NOR devices, each transistor stands alone). Thanks to its high density NAND is best suited to devices requiring high capacity data storage, such as pictures, music, or data files. NAND non-volatility, makes of it a good storage solution for many applications where mobility, low power, and speed are key factors. Low pin count and simple interface are other advantages of NAND.

Table 15-268 summarizes the NAND interface signals level applied to external device or memories.

**Table 15-268. NAND Interface Bus Operations Summary**

Bus operation	CLE	ALE	nCE	nWE <sup>(1)</sup>	nRE <sup>(1)</sup>	nWP
Read (cmd input)	H	L	L	RE	H	x
Read (add input)	L	H	L	RE	H	x
Write (cmd input)	H	L	L	RE	H	H
Write (add input)	L	H	L	RE	H	H
Data input	L	L	L	RE	H	H
Data output	L	L	L	H	FE	x
Busy (during read)	x	x	H <sup>(2)</sup>	H <sup>(2)</sup>	H <sup>(2)</sup>	x
Busy (during program)	x	x	x	x	x	H
Busy (during erase)	x	x	x	x	x	H
Write protect	x	x	x	x	x	L
Stand-by	x	x	H	x	x	H/L <sup>(3)</sup>

<sup>(1)</sup> RE stands for rising edge, FE stands for falling edge

<sup>(2)</sup> Can be either nCE high, or WE and nRE high.

<sup>(3)</sup> nWP should be biased to CMOS high or CMOS low for standby

### 15.4.6.2.1.3 NOR Interface Protocol

NOR flash architecture, introduced in 1988, is a flash technology. Unlike NAND which is a sequential access device, NOR is directly addressable, that is, is designed to be a random access device. NOR is best suited to devices used to store and run code or firmware, usually in small capacities. While NOR has fast read capabilities it has slow write and erase functions compared to NAND architecture.

Table 15-269 summarizes the NOR interface signals level applied to external device or memories.

**Table 15-269. NOR Interface Bus Operations Summary**

Bus operation	CLK	nADV	nCS	nOE	nWE	WAIT	DQ[15:0]
Read (asynchronous)	x	L	L	L	H	Asserted	Output
Read (synchronous)	Running	L	L	L	H	Driven	Output
Read (burst suspend)	Halted	x	L	H	H	Active	Output
Write	x	L	L	H	L	Asserted	Input
Output disable	x	x	L	H	H	Asserted	High-Z
Standby	x	x	H	x	x	High-Z	High-Z

### 15.4.6.2.1.4 Other Technologies

Other supported device type interact with the GPMC through the NOR interface protocol.

OneNAND™ is a high density and low-power memory device. OneNAND™ is based on single- or multi-level-cell NAND core with SRAM and logic, and interfaces as a synchronous NOR Flash, plus has synchronous write capability. It reads faster than conventional NAND and write faster than conventional NOR flash. Hence, it is appropriate for both mass storage and code storage.

pSRAM stands for pseudo-static random access memory. pSRAM is a low-power memory device for mobile applications. pSRAM is based on the DRAM cell with internal refresh and address control features, and interfaces as a synchronous NOR Flash, plus has synchronous write capability.

### 15.4.6.2.1.5 Supported Protocols

The GPMC supports the following interface protocols when communicating with external memory or external devices:

- Asynchronous read/write access
- Asynchronous read page access (4-8-16 Word16)
- Synchronous read/write access
- Synchronous read burst access without wrap capability (4-8-16 Word16)
- Synchronous read burst access with wrap capability (4-8-16 Word16)

### 15.4.6.2.2 GPMC Features and Settings

This section lists GPMC features and settings:

- Supported device type: up to four NAND or NOR protocol external memories or devices
- Operating Voltage: 1.8V;
- Maximum GPMC addressing capability: 1 GByte divided into eight chip-selects
- Maximum supported memory size: 256 MBytes (must be a power-of-2)
- Minimum supported memory size: 16 MBytes (must be a power-of-2). Aliasing occurs when addressing smaller memories.
- Data path to external memory or device: 8- and 16-bit wide
- Burst and page access: burst of 4-8-16 Word16
- Supports bus keeping
- Supports bus turn around



## 15.4.7 GPMC Register Manual

This section provides information about the GPMC instance in this product. [Table 15-271](#) provides a summary of the GPMC registers. The remaining parts of this section describe the registers within the module instance.

### 15.4.7.1 GPMC Register Summary

**Table 15-270. GPMC Instance Summary**

Module Name	Base Address	Size
GPMC	0x5000 0000	32 Mbytes

**Table 15-271. GPMC Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPMC L3 Physical Address
<a href="#">GPMC_REVISION</a>	R	32	0x0000 0000	0x5000 0000
<a href="#">GPMC_SYSCONFIG</a>	RW	32	0x0000 0010	0x5000 0010
<a href="#">GPMC_SYSSTATUS</a>	R	32	0x0000 0014	0x5000 0014
<a href="#">GPMC_IRQSTATUS</a>	RW	32	0x0000 0018	0x5000 0018
<a href="#">GPMC_IRQENABLE</a>	RW	32	0x0000 001C	0x5000 001C
<a href="#">GPMC_TIMEOUT_CONTROL</a>	RW	32	0x0000 0040	0x5000 0040
<a href="#">GPMC_ERR_ADDRESS</a>	RW	32	0x0000 0044	0x5000 0044
<a href="#">GPMC_ERR_TYPE</a>	RW	32	0x0000 0048	0x5000 0048
<a href="#">GPMC_CONFIG</a>	RW	32	0x0000 0050	0x5000 0050
<a href="#">GPMC_STATUS</a>	RW	32	0x0000 0054	0x5000 0054
<a href="#">GPMC_CONFIG1_i<sup>(1)</sup></a>	RW	32	0x0000 0060 + (0x0000 0030 * i)	0x5000 0060 + (0x0000 0030 * i)
<a href="#">GPMC_CONFIG2_i<sup>(1)</sup></a>	RW	32	0x0000 0064 + (0x0000 0030 * i)	0x5000 0064 + (0x0000 0030 * i)
<a href="#">GPMC_CONFIG3_i<sup>(1)</sup></a>	RW	32	0x0000 0068 + (0x0000 0030 * i)	0x5000 0068 + (0x0000 0030 * i)
<a href="#">GPMC_CONFIG4_i<sup>(1)</sup></a>	RW	32	0x0000 006C + (0x0000 0030 * i)	0x5000 006C + (0x0000 0030 * i)
<a href="#">GPMC_CONFIG5_i<sup>(1)</sup></a>	RW	32	0x0000 0070 + (0x0000 0030 * i)	0x5000 0070 + (0x0000 0030 * i)
<a href="#">GPMC_CONFIG6_i<sup>(1)</sup></a>	RW	32	0x0000 0074 + (0x0000 0030 * i)	0x5000 0074 + (0x0000 0030 * i)
<a href="#">GPMC_CONFIG7_i<sup>(1)</sup></a>	RW	32	0x0000 0078 + (0x0000 0030 * i)	0x5000 0078 + (0x0000 0030 * i)
<a href="#">GPMC_NAND_COMMAND_i<sup>(1)</sup></a>	W	32	0x0000 007C + (0x0000 0030 * i)	0x5000 007C + (0x0000 0030 * i)
<a href="#">GPMC_NAND_ADDRESS_i<sup>(1)</sup></a>	W	32	0x0000 0080 + (0x0000 0030 * i)	0x5000 0080 + (0x0000 0030 * i)
<a href="#">GPMC_NAND_DATA_i<sup>(1)</sup></a>	RW	32	0x0000 0084 + (0x0000 0030 * i)	0x5000 0084 + (0x0000 0030 * i)
<a href="#">GPMC_PREFETCH_CONFIG1</a>	RW	32	0x0000 01E0	0x5000 01E0
<a href="#">GPMC_PREFETCH_CONFIG2</a>	RW	32	0x0000 01E4	0x5000 01E4
<a href="#">GPMC_PREFETCH_CONTROL</a>	RW	32	0x0000 01EC	0x5000 01EC
<a href="#">GPMC_PREFETCH_STATUS</a>	RW	32	0x0000 01F0	0x5000 01F0
<a href="#">GPMC_ECC_CONFIG</a>	RW	32	0x0000 01F4	0x5000 01F4
<a href="#">GPMC_ECC_CONTROL</a>	RW	32	0x0000 01F8	0x5000 01F8
<a href="#">GPMC_ECC_SIZE_CONFIG</a>	RW	32	0x0000 01FC	0x5000 01FC
<a href="#">GPMC_ECC_j_RESULT<sup>(2)</sup></a>	RW	32	0x0000 0200 + (0x0000 0004 * j)	0x5000 0200 + (0x0000 0004 * j)
<a href="#">GPMC_BCH_RESULT0_i<sup>(1)</sup></a>	RW	32	0x0000 0240 + (0x0000 0010 * i)	0x5000 0240 + (0x0000 0010 * i)
<a href="#">GPMC_BCH_RESULT1_i<sup>(1)</sup></a>	RW	32	0x0000 0244 + (0x0000 0010 * i)	0x5000 0244 + (0x0000 0010 * i)
<a href="#">GPMC_BCH_RESULT2_i<sup>(1)</sup></a>	RW	32	0x0000 0248 + (0x0000 0010 * i)	0x5000 0248 + (0x0000 0010 * i)
<a href="#">GPMC_BCH_RESULT3_i<sup>(1)</sup></a>	RW	32	0x0000 024C + (0x0000 0010 * i)	0x5000 024C + (0x0000 0010 * i)
<a href="#">GPMC_BCH_RESULT4_i<sup>(1)</sup></a>	RW	32	0x0000 0300 + (0x0000 0010 * i)	0x5000 0300 + (0x0000 0010 * i)
<a href="#">GPMC_BCH_RESULT5_i<sup>(1)</sup></a>	RW	32	0x0000 0304 + (0x0000 0010 * i)	0x5000 0304 + (0x0000 0010 * i)
<a href="#">GPMC_BCH_RESULT6_i<sup>(1)</sup></a>	RW	32	0x0000 0308 + (0x0000 0010 * i)	0x5000 0308 + (0x0000 0010 * i)
<a href="#">GPMC_BCH_SWDATA</a>	RW	32	0x0000 02D0	0x5000 02D0

<sup>(1)</sup> i = 0 to 3 for GPMC

<sup>(2)</sup> j = 0 to 8 for GPMC

### 15.4.7.2 GPMC Register Descriptions

**NOTE:** All GPMC registers are aligned to 32-bit address boundaries. All register file accesses, except to [GPMC\\_NAND\\_DATA\\_i](#) register, are little endian. If the [GPMC\\_NAND\\_DATA\\_i](#) register location is accessed, the endianness is access-dependent.

In the whole section *i* corresponds to the chip-select number, *i* = 0 to 3.

**Table 15-272. GPMC\_REVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 0000		
<b>Description</b>	This register contains the IP revision code.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	T1 internal data

**Table 15-273. Register Call Summary for Register GPMC\_REVISION**

GPMC Register Manual

- [GPMC Register Summary: \[0\]](#)

**Table 15-274. GPMC\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 0010		
<b>Description</b>	This register controls the various parameters of the Interconnect.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIDLEMODE	RESERVED	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0s for future compatibility. Reads return 0s.	RW	0x00000000
4:3	SIDLEMODE	0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module 0x3: Do not use	RW	0x0
2	RESERVED	Write 0 for future compatibility Reads returns 0	RW	0x0
1	SOFTRESET	Software reset. Set this bit to 1 triggers a module reset. This bit is automatically reset by hardware. During reads, it always returns 0. 0x0: Normal mode 0x1: The module is reset	RW	0x0

Bits	Field Name	Description	Type	Reset
0	AUTOIDLE	Internal Interface clock gating strategy 0x0: Interface clock is free-running 0x1: Automatic Interface clock gating strategy is applied, based on the Interconnect activity	RW	0x0

**Table 15-275. Register Call Summary for Register GPMC\_SYSCONFIG**

GPMC Functional Description

- [GPMC Software Reset: \[0\]](#)
- [GPMC Power Management: \[1\] \[2\]](#)

GPMC Basic Programming Model

- [GPMC Initialization: \[3\] \[4\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[5\]](#)

**Table 15-276. GPMC\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 0014		
<b>Description</b>	This register provides status information about the module, excluding the interrupt status information		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											RESETDONE				

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reads returns 0	R	0x000000
7:1	RESERVED	Reads returns 0 (reserved for Interconnect-socket status information)	R	0x00
0	RESETDONE	Internal reset monitoring 0x0: Internal module reset in ongoing 0x1: Reset completed	R	0x-

**Table 15-277. Register Call Summary for Register GPMC\_SYSSTATUS**

GPMC Functional Description

- [GPMC Software Reset: \[0\]](#)

GPMC Basic Programming Model

- [GPMC Initialization: \[1\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[2\]](#)

**Table 15-278. GPMC\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 0018		
<b>Description</b>	This interrupt status register regroups all the status of the module internal events that can generate an interrupt.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																WAIT2EDGEDETECTIONSTATUS			WAIT1EDGEDETECTIONSTATUS			WAIT0EDGEDETECTIONSTATUS			RESERVED					TERMINALCOUNTSTATUS		FIFOEVENTSTATUS

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
10	WAIT2EDGEDETECTIONSTATUS	Status of the Wait2 Edge Detection interrupt Read 0x0: A transition on WAIT2 input pin has not been detected Write 0x0: WAIT2EDGEDETECTIONSTATUS bit unchanged Read 0x1: A transition on WAIT2 input pin has been detected Write 0x1: WAIT2EDGEDETECTIONSTATUS bit is reset	RW	0x0
9	WAIT1EDGEDETECTIONSTATUS	Status of the Wait1 Edge Detection interrupt Read 0x0: A transition on WAIT1 input pin has not been detected Write 0x0: WAIT1EDGEDETECTIONSTATUS bit unchanged Read 0x1: A transition on WAIT1 input pin has been detected Write 0x1: WAIT1EDGEDETECTIONSTATUS bit is reset	RW	0x0
8	WAIT0EDGEDETECTIONSTATUS	Status of the Wait0 Edge Detection interrupt Read 0x0: A transition on WAIT0 input pin has not been detected Write 0x0: WAIT0EDGEDETECTIONSTATUS bit unchanged Read 0x1: A transition on WAIT0 input pin has been detected Write 0x1: WAIT0EDGEDETECTIONSTATUS bit is reset	RW	0x0
7:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
1	TERMINALCOUNTSTATUS	Status of the TerminalCountEvent interrupt Read 0x0: Indicates that CountValue is greater than 0 Write 0x0: TERMINALCOUNTSTATUS bit unchanged Read 0x1: Indicates that CountValue is equal to 0 Write 0x1: TERMINALCOUNTSTATUS bit is reset	RW	0x0

Bits	Field Name	Description	Type	Reset
0	FIFOEVENTSTATUS	<p>Status of the FIFOEvent interrupt</p> <p>Read 0x0: Indicates than less than <a href="#">GPMC_PREFETCH_STATUS</a>[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and less than FIFOTHRESHOLD bytes free places are available in write-posting mode.</p> <p>Write 0x0: FIFOEVENTSTATUS bit unchanged</p> <p>Read 0x1: Indicates than at least <a href="#">GPMC_PREFETCH_STATUS</a>[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and at least FIFOTHRESHOLD bytes free places are available in write-posting mode.</p> <p>Write 0x1: FIFOEVENTSTATUS bit is reset</p>	RW	0x0

**Table 15-279. Register Call Summary for Register GPMC\_IRQSTATUS**

GPMC Functional Description

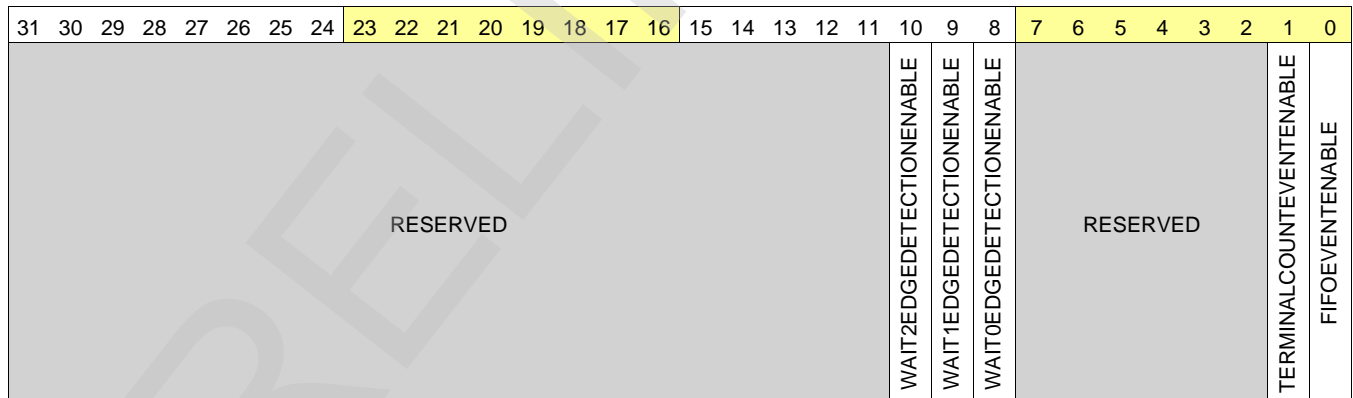
- [GPMC Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Ready Pin Monitored by Hardware Interrupt: \[5\] \[6\] \[7\]](#)
- [FIFO Control in Prefetch Mode: \[8\] \[9\]](#)
- [FIFO Control in Write-Posting Mode: \[10\] \[11\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[12\]](#)

**Table 15-280. GPMC\_IRQENABLE**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 001C		
<b>Description</b>	The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
10	WAIT2EDGEDETECTION ENABLE	<p>Enables the Wait2 Edge Detection interrupt</p> <p>0x0: Wait2EdgeDetection interrupt is masked</p> <p>0x1: Wait2EdgeDetection event generates an interrupt if occurs</p>	RW	0
9	WAIT1EDGEDETECTION ENABLE	<p>Enables the Wait1 Edge Detection interrupt</p> <p>0x0: Wait1EdgeDetection interrupt is masked</p> <p>0x1: Wait1EdgeDetection event generates an interrupt if occurs</p>	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WAIT0EDGEDETECTION ENABLE	Enables the Wait0 Edge Detection interrupt 0x0: Wait0EdgeDetection interrupt is masked 0x1: Wait0EdgeDetection event generates an interrupt if occurs	RW	0x0
7:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
1	TERMINALCOUNTEVENT ENABLE	Enables TerminalCountEvent interrupt issuing in prefetch or write-posting mode 0x0: TerminalCountEvent interrupt is masked 0x1: TerminalCountEvent interrupt is not masked	RW	0x0
0	FIFOEVENTENABLE	Enables the FIFOEvent interrupt 0x0: FIFOEvent interrupt is masked 0x1: FIFOEvent interrupt is not masked	RW	0x0

**Table 15-281. Register Call Summary for Register GPMC\_IRQENABLE**

## GPMC Functional Description

- [GPMC Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [NAND Device-Ready Pin: \[5\]](#)
- [Ready Pin Monitored by Hardware Interrupt: \[6\]](#)
- [FIFO Control in Prefetch Mode: \[7\] \[8\]](#)
- [FIFO Control in Write-Posting Mode: \[9\] \[10\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[11\]](#)

**Table 15-282. GPMC\_TIMEOUT\_CONTROL**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 0040		
<b>Description</b>	The <a href="#">GPMC_TIMEOUT_CONTROL</a> register allows the user to set the start value of the timeout counter		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TIMEOUTSTARTVALUE								RESERVED		TIMEOUTENABLE					

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000
12:4	TIMEOUTSTARTVALUE	Start value of the time-out counter 0x000: Zero GPMC_FCLK cycle 0x001: One GPMC_FCLK cycle ... 0x1FF: 511 GPMC_FCLK cycles	RW	0x1FF
3:1	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
0	TIMEOUTENABLE	Enable bit of the TimeOut feature 0x0: TimeOut feature is disabled 0x1: TimeOut feature is enabled	RW	0x0

**Table 15-283. Register Call Summary for Register GPMC\_TIMEOUT\_CONTROL**

GPMC Functional Description

- [Error Handling: \[0\] \[1\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[2\]](#)
- [GPMC Register Descriptions: \[3\]](#)

**Table 15-284. GPMC\_ERR\_ADDRESS**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 0044		
<b>Description</b>	The <a href="#">GPMC_ERR_ADDRESS</a> register stores the address of the illegal access when an error occurs		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ILLEGALADD															

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
30:0	ILLEGALADD	Address of illegal access A30: 0 for memory region, 1 for GPMC register region A29-A0: 1 GBytes max	R	0x00000000

**Table 15-285. Register Call Summary for Register GPMC\_ERR\_ADDRESS**

GPMC Functional Description

- [Error Handling: \[0\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[1\]](#)
- [GPMC Register Descriptions: \[2\]](#)

**Table 15-286. GPMC\_ERR\_TYPE**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 0048		
<b>Description</b>	The <a href="#">GPMC_ERR_TYPE</a> register stores the type of error when an error occurs		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ILLEGALCMD		RESERVED		ERRORNOTSUPPADD	ERRORNOTSUPPMCMD	ERRORTIMEOUT	RESERVED	ERRORVALID							





Bits	Field Name	Description	Type	Reset
9	WAIT1PINPOLARITY	Selects the polarity of input pin WAIT1 0x0: WAIT1 active low 0x1: WAIT1 active high	RW	0x1
8	WAIT0PINPOLARITY	Selects the polarity of input pin WAIT0 0x0: WAIT0 active low 0x1: WAIT0 active high	RW	0x0
7:5	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
4	WRITEPROTECT	Controls the WP output pin level 0x0: WP output pin is low 0x1: WP output pin is high	RW	0x0
3:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
1	LIMITEDADDRESS	Limited Address device support 0x0: No effect 0x1: A26-A11 are not modified during an external memory access.	RW	0x0
0	NANDFORCEPOSTEDWRITE	Enables the Force Posted Write feature to NAND Cmd/Add/Data location 0x0: Disables Force Posted Write 0x1: Enables Force Posted Write	RW	0x0

**Table 15-289. Register Call Summary for Register GPMC\_CONFIG**

GPMC Environment

- [GPMC Signals: \[0\]](#)

GPMC Functional Description

- [GPMC Interrupt Requests: \[1\] \[2\] \[3\]](#)
- [GPMC Address and Data Bus: \[4\]](#)
- [GPMC I/O Configuration Setting: \[5\] \[6\]](#)
- [WAIT Pin Monitoring Control: \[7\]](#)
- [Write Protect Signal \(nWP\): \[8\]](#)
- [Asynchronous Single Read Operation on a Nonmultiplexed Device: \[9\] \[10\] \[11\]](#)
- [Asynchronous Single Write Operation on a Nonmultiplexed Device: \[12\]](#)
- [Asynchronous Multiple \(Page Mode\) Read Operation on a Nonmultiplexed Device: \[13\]](#)
- [NAND Device Command and Address Phase Control: \[14\]](#)
- [Ready Pin Monitored by Hardware Interrupt: \[15\]](#)

GPMC Basic Programming Model

- [Set Memory Access: \[16\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[17\]](#)

**Table 15-290. GPMC\_STATUS**

Address Offset	0x0000 0054	Instance	GPMC
Physical Address	0x5000 0054		
Description	The status register provides global status bits of the GPMC		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																WAIT2STATUS			WAIT1STATUS			WAIT0STATUS			RESERVED							EMPTYWRITEBUFFERSTATUS

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
10	WAIT2STATUS	Is a copy of input pin WAIT2. (Reset value is WAIT2 input pin sampled at IC reset) 0x0: WAIT2 asserted (inactive state) 0x1: WAIT2 de-asserted	R	0x-
9	WAIT1STATUS	Is a copy of input pin WAIT1. (Reset value is WAIT1 input pin sampled at IC reset) 0x0: WAIT1 asserted (inactive state) 0x1: WAIT1 de-asserted	R	0x-
8	WAIT0STATUS	Is a copy of input pin WAIT0. (Reset value is WAIT0 input pin sampled at IC reset) 0x0: WAIT0 asserted (inactive state) 0x1: WAIT0 de-asserted	R	0x-
7:1	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x00
0	EMPTYWRITEBUFFERSTATUS	Stores the empty status of the write buffer 0x0: Write Buffer is not empty 0x1: Write Buffer is empty	R	0x1

**Table 15-291. Register Call Summary for Register GPMC\_STATUS**

## GPMC Functional Description

- [NAND Device Command and Address Phase Control: \[0\]](#)
- [NAND Device-Ready Pin: \[1\] \[2\]](#)
- [Ready Pin Monitored by Software Polling: \[3\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[4\]](#)

**Table 15-292. GPMC\_CONFIG1\_i**

<b>Address Offset</b>	0x0000 0060 + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0060 + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	The configuration register 1 sets signal control parameters per chip-select		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRAPBURST	READMULTIPLE	READTYPE	WRITEMULTIPLE	WRITETYPE	CLKACTIVATIONTIME	ATTACHEDDEVICEPAGELENGTH	WAITREADMONITORING	WAITWRITEMONITORING	RESERVED	WAITMONITORINGTIME	WAITPINSELECT	RESERVED	DEVICESTYPE	DEVICETYPE	MUXADDRESS	RESERVED	TIMEPARAGRANULARITY	RESERVED	GPMCCLKDIVIDER												

Bits	Field Name	Description	Type	Reset
31	WRAPBURST	Enables the wrapping burst capability. Must be set if the attached device is configured in wrapping burst 0x0: Synchronous wrapping burst not supported 0x1: Synchronous wrapping burst supported	RW	0x0
30	READMULTIPLE	Selects the read single or multiple access 0x0: Single access 0x1: Multiple access (burst if synchronous, page if asynchronous)	RW	0x0
29	READTYPE	Selects the read mode operation 0x0: Read Asynchronous 0x1: Read Synchronous	RW	0x0
28	WRITEMULTIPLE	Selects the write single or multiple access 0x0: Single access 0x1: Multiple access (burst if synchronous, considered as single if asynchronous)	RW	0x0
27	WRITETYPE	Selects the write mode operation 0x0: Write Asynchronous 0x1: Write Synchronous	RW	0x0
26:25	CLKACTIVATIONTIME	Output GPMC_CLK activation time 0x0: First rising edge of GPMC_CLK at start access time 0x1: First rising edge of GPMC_CLK one GPMC_FCLK cycle after start access time 0x2: First rising edge of GPMC_CLK two GPMC_FCLK cycles after start access time 0x3: Reserved	RW	0x0
24:23	ATTACHEDDEVICEPAGELENGTH	Specifies the attached device page (burst) length 0x0: 4 Words 0x1: 8 Words 0x2: 16 Words 0x3: Reserved (1 Word = Interface size)	RW	0x0
22	WAITREADMONITORING	Selects the Wait monitoring configuration for Read accesses (Reset value is <i>bootwaiten</i> input pin sampled at IC reset) 0x0: Wait pin is not monitored for read accesses 0x1: Wait pin is monitored for read accesses	RW	0x-
21	WAITWRITEMONITORING	Selects the Wait monitoring configuration for Write accesses 0x0: Wait pin is not monitored for write accesses 0x1: Wait pin is monitored for write accesses	RW	0x0
20	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0

## General-Purpose Memory Controller

www.ti.com

Bits	Field Name	Description	Type	Reset
19:18	WAITMONITORINGTIME	Selects input pin Wait monitoring time 0x0: Wait pin is monitored with valid data 0x1: Wait pin is monitored one GPMC_CLK cycle before valid data 0x2: Wait pin is monitored two GPMC_CLK cycle before valid data 0x3: Reserved	RW	0x0
17:16	WAITPINSELECT	Selects the input WAIT pin for this chip-select (Reset value is <i>bootwaitselect</i> input pin sampled at IC reset for CS0 and 0 for CS1-7) 0x0: Wait input pin is WAIT0 0x1: Wait input pin is WAIT1 0x2: Wait input pin is WAIT2 0x3: Reserved	RW	0x-
15:14	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x0
13:12	DEVICESTYPE	Selects the device size attached (Reset value is <i>bootdevicesize</i> input pin sampled at IC reset for CS0 and 0x1 for CS1 to CS7) 0x0: 8 bit 0x1: 16 bit 0x2: Reserved 0x3: Reserved	RW	0x-
11:10	DEVICETYPE	Selects the attached device type 0x0: NOR Flash like, asynchronous and synchronous devices 0x1: Reserved 0x2: NAND Flash like devices, stream mode 0x3: Reserved	RW	0x0
9:8	MUXADDDATA	Enables the address and data multiplexed protocol (Reset value is <i>cs0muxdevice</i> input pin sampled at IC reset for CS0 and 0 for CS1-CS7) 0x0: Non-multiplexed attached device 0x1: AAD-mux protocol device 0x2: Address and data multiplexed attached device 0x3: Reserved	RW	0x-
7:5	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
4	TIMEPARAGRANULARITY	Signals timing latencies scalar factor (RD/WRCYCLETIME, RD/WRACCESSTIME, PAGEBURSTACCESSTIME, CSONTIME, CSRD/WROFFTIME, ADVONTIME, ADVRD/WROFFTIME, OEONTIME, OEOFFTIME, WEONTIME, WEOFFTIME, CYCLE2CYCLEDELAY, BUSTURNAROUND, TIMEOUTSTARTVALUE) 0x0: x1 latencies 0x1: x2 latencies	RW	0x0
3:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
1:0	GPMCFCLKDIVIDER	Divides the GPMC_FCLK clock 0x0: GPMC_CLK frequency = GPMC_FCLK frequency 0x1: GPMC_CLK frequency = GPMC_FCLK frequency / 2 0x2: GPMC_CLK frequency = GPMC_FCLK frequency / 3 0x3: GPMC_CLK frequency = GPMC_FCLK frequency / 4	RW	0x0

**Table 15-293. Register Call Summary for Register GPMC\_CONFIG1\_i**

GPMC Environment	
<ul style="list-style-type: none"> <li>GPMC Signals: [0]</li> </ul>	
GPMC Functional Description	
<ul style="list-style-type: none"> <li>GPMC Clock Configuration: [1] [2]</li> <li>L3 Interconnect Interface: [3] [4]</li> <li>GPMC I/O Configuration Setting: [5] [6] [7] [8] [9] [10] [11] [12]</li> <li>GPMC CS0 Default Configuration at IC Reset: [13] [14] [15] [16]</li> <li>Supported Devices: [17]</li> <li>Access Size Adaptation and Device Width: [18]</li> <li>Address/Data-Multiplexing Interface: [19]</li> <li>WAIT Pin Monitoring Control: [20] [21] [22] [23] [24]</li> <li>Wait Monitoring During an Asynchronous Read Access: [25] [26]</li> <li>Wait Monitoring During an Asynchronous Write Access: [27] [28] [29]</li> <li>Wait Monitoring During a Synchronous Read Access: [30] [31]</li> <li>Wait Monitoring During a Synchronous Write Access: [32] [33]</li> <li>Idle Cycle Control Between Successive Accesses: [34] [35]</li> <li>Slow Device Support (TIMEPARAGRANULARITY Parameter): [36]</li> <li>Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME): [37]</li> <li>GPMC_CLK: [38] [39]</li> <li>GPMC_CLK and Control Signals Setup and Hold: [40]</li> <li>Access Time (RDACCESSTIME / WRACCESSTIME): [41]</li> <li>Page Burst Access Time (PAGEBURSTACCESSTIME): [42]</li> <li>NOR Access Description: [43] [44]</li> <li>Asynchronous Multiple (Page) Write Operation on an Address/Data-Multiplexed Device: [45] [46]</li> <li>Asynchronous Multiple (Page) Read Operation on an AAD-Multiplexed Device: [47] [48]</li> <li>Synchronous Access Description: [49] [50]</li> <li>Synchronous Multiple (Burst) Read (4-, 8-, 16-Word16 Burst With Wraparound Capability): [51] [52] [53]</li> <li>Page and Burst Support: [54] [55]</li> <li>System Burst Versus External Device Burst Support: [56] [57] [58]</li> <li>pSRAM Access Specificities: [59]</li> <li>Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode: [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77]</li> <li>NAND Device-Ready Pin: [78] [79]</li> </ul>	
GPMC Basic Programming Model	
<ul style="list-style-type: none"> <li>GPMC Configuration in NOR Mode: [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95]</li> <li>GPMC Configuration in NAND mode: [96] [97] [98] [99]</li> <li>Set Memory Access: [100] [101] [102] [103] [104]</li> <li>GPMC Timing Parameters: [105] [106] [107] [108] [109] [110] [111] [112] [113]</li> <li>NAND Flash Interface Timing Parameters Formulas: [114]</li> <li>Synchronous NOR Flash Timing Parameters Formulas: [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125]</li> <li>Asynchronous NOR Flash Timing Parameters Formulas: [126]</li> </ul>	
GPMC Register Manual	
<ul style="list-style-type: none"> <li>GPMC Register Summary: [127]</li> </ul>	

**Table 15-294. GPMC\_CONFIG2\_i**

<b>Address Offset</b>	0x0000 0064 + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0064 + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	CS signal timing parameter configuration		
<b>Type</b>	RW		

## General-Purpose Memory Controller

www.ti.com

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSWROFFTIME				RESERVED		CSRDOFFTIME				CSEXTRADelay	RESERVED		CSONTIME										

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x000
20:16	CSWROFFTIME	CS i de-assertion time from start cycle time for write accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
15:13	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
12:8	CSRDOFFTIME	CS i de-assertion time from start cycle time for read accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
7	CSEXTRADelay	CS i Add Extra Half GPMC_FCLK cycle 0x0: CS i Timing control signal is not delayed 0x1: CS i Timing control signal is delayed of half GPMC_FCLK clock cycle	RW	0x0
6:4	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
3:0	CSONTIME	CS i assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x1

Table 15-295. Register Call Summary for Register GPMC\_CONFIG2\_i

## GPMC Functional Description

- [nCS: Chip-Select Signal Control Assertion/Deassertion Time \(CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADelay\): \[0\] \[1\] \[2\] \[3\]](#)
- [Asynchronous Single-Read Operation on an Address/Data Multiplexed Device: \[4\] \[5\]](#)
- [Synchronous Single Read: \[6\] \[7\]](#)
- [Synchronous Multiple \(Burst\) Write: \[8\] \[9\]](#)

## GPMC Basic Programming Model

- [GPMC Timing Parameters: \[10\] \[11\] \[12\] \[13\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[14\]](#)

Table 15-296. GPMC\_CONFIG3\_i

<b>Address Offset</b>	0x0000 0068 + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0068 + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	nADV signal timing parameter configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED				ADVAADMUXWROFFTIME				RESERVED				ADVAADMUXRDOFFTIME				RESERVED				ADVWROFFTIME				RESERVED				ADVRDOFFTIME				ADVEXTRADELAY		ADVAADMUXONTIME			ADVONTIME		

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
30:28	ADVAADMUXWROFFTIME	nADV de-assertion for first address phase when using the AAD-mux protocol 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x2
27	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
26:24	ADVAADMUXRDOFFTIME	nADV assertion for first address phase when using the AAD-mux protocol 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x2
23:21	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
20:16	ADVWROFFTIME	nADV de-assertion time from start cycle time for write accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x02
15:13	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
12:8	ADVRDOFFTIME	nADV de-assertion time from start cycle time for read accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x02
7	ADVEXTRADELAY	nADV Add Extra Half GPMC_FCLK cycle 0x0: nADV Timing control signal is not delayed 0x1: nADV Timing control signal is delayed of half GPMC_FCLK clock cycle	RW	0x0
6:4	ADVAADMUXONTIME	nADV assertion for first address phase when using the AAD-mux protocol 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x1
3:0	ADVONTIME	nADV assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x1

**Table 15-297. Register Call Summary for Register GPMC\_CONFIG3\_i**

GPMC Functional Description
<ul style="list-style-type: none"> <li>nADV/ALE: Address Valid/Address Latch Enable Signal Control Assertion/Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME / ADVEXTRADELAY/ADVAADMUXONTIME/ADVAADMUXROFFTIME/ADVAADMUXWROFFTIME): [0] [1] [2] [3] [4] [5] [6]</li> <li>Asynchronous Single-Read Operation on an Address/Data Multiplexed Device: [7] [8]</li> <li>Asynchronous Single Read Operation on an AAD-Multiplexed Device: [9] [10] [11] [12] [13]</li> <li>Asynchronous Single Write Operation on an AAD-Multiplexed Device: [14] [15] [16] [17] [18]</li> <li>Synchronous Single Read: [19] [20] [21] [22] [23] [24] [25]</li> <li>Synchronous Multiple (Burst) Write: [26] [27] [28] [29] [30] [31]</li> </ul>
GPMC Basic Programming Model
<ul style="list-style-type: none"> <li>GPMC Timing Parameters: [32] [33] [34] [35] [36] [37] [38]</li> </ul>
GPMC Register Manual
<ul style="list-style-type: none"> <li>GPMC Register Summary: [39]</li> </ul>

**Table 15-298. GPMC\_CONFIG4\_i**

<b>Address Offset</b>	0x0000 006C + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 006C + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	nWE and nOE signals timing parameter configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WEEXTRADELAY	RESERVED				WEONTIME	OEAADMUXOFFTIME				OEOFFTIME	OEEXTRADELAY	OEAADMUXONTIME				OEONTIME							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x3
28:24	WEOFFTIME	nWE de-assertion time from start cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
23	WEEXTRADELAY	nWE Add Extra Half GPMC_FCLK cycle 0x0: nWE Timing control signal is not delayed 0x1: nWE Timing control signal is delayed of half GPMC_FCLK clock cycle	RW	0x0
22:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
19:16	WEONTIME	nWE assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x5
15:13	OEAADMUXOFFTIME	nOE de-assertion time for the first address phase in an AAD-mux access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x3



Bits	Field Name	Description	Type	Reset
12:8	OEOFFTIME	nOE de-assertion time from start cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
7	OEXTRADELAY	nOE Add Extra Half GPMC_FCLK cycle 0x0: nOE Timing control signal is not delayed 0x1: nOE Timing control signal is delayed of half GPMC_FCLK clock cycle	RW	0x0
6:4	OEAADMUXONTIME	nOE assertion time for the first address phase in an AAD-mux access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x1
3:0	OEONTIME	nOE assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x6

**Table 15-299. Register Call Summary for Register GPMC\_CONFIG4\_i**

GPMC Functional Description

- nOE/nRE: Output Enable / Read Enable Signal Control Assertion / Deassertion Time (OEONTIME / OEOFFTIME / OEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME): [0] [1] [2]
- nWE: Write Enable Signal Control Assertion / Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY): [3] [4] [5]
- Asynchronous Single-Read Operation on an Address/Data Multiplexed Device: [6] [7]
- Asynchronous Single Write Operation on an Address/Data-Multiplexed Device: [8] [9]
- Asynchronous Single Read Operation on an AAD-Multiplexed Device: [10] [11] [12]
- Asynchronous Single Write Operation on an AAD-Multiplexed Device: [13]
- Synchronous Single Read: [14] [15] [16] [17] [18]
- Synchronous Multiple (Burst) Write: [19] [20] [21] [22] [23] [24]
- Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode: [25]

GPMC Basic Programming Model

- GPMC Timing Parameters: [26] [27] [28] [29] [30] [31] [32] [33]

GPMC Register Manual

- GPMC Register Summary: [34]

**Table 15-300. GPMC\_CONFIG5\_i**

<b>Address Offset</b>	0x0000 0070 + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0070 + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	RdAccessTime and CycleTime timing parameters configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				PAGEBURSTACCSSTIME				RESERVED				RDACCESSTIME				RESERVED				WRCYCLETIME				RESERVED				RDCYCLETIME			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
27:24	PAGEBURSTACCESSTIME	Delay between successive words in a multiple access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x1
23:21	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
20:16	RDACCESSTIME	Delay between start cycle time and first data valid 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x0F
15:13	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x0
12:8	WRCYCLETIME	Total write cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x11
7:5	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
4:0	RDCYCLETIME	Total read cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x11

**Table 15-301. Register Call Summary for Register GPMC\_CONFIG5\_i**

## GPMC Functional Description

- [Read Cycle Time and Write Cycle Time \(RDCYCLETIME / WRCYCLETIME\): \[0\] \[1\]](#)
- [Access Time \(RDACCESSTIME / WRACCESSTIME\): \[2\]](#)
- [Access Time on Read Access: \[3\] \[4\]](#)
- [Page Burst Access Time \(PAGEBURSTACCESSTIME\): \[5\]](#)
- [Asynchronous Single-Read Operation on an Address/Data Multiplexed Device: \[6\] \[7\]](#)
- [Asynchronous Single Write Operation on an Address/Data-Multiplexed Device: \[8\]](#)
- [Synchronous Single Read: \[9\] \[10\]](#)
- [Synchronous Multiple \(Burst\) Read \(4-, 8-, 16-Word/16 Burst With Wraparound Capability\): \[11\] \[12\] \[13\]](#)
- [Synchronous Multiple \(Burst\) Write: \[14\] \[15\] \[16\] \[17\]](#)
- [Asynchronous Multiple \(Page Mode\) Read Operation on a Nonmultiplexed Device: \[18\] \[19\] \[20\] \[21\]](#)

## GPMC Basic Programming Model

- [GPMC Timing Parameters: \[22\] \[23\] \[24\] \[25\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[26\]](#)

**Table 15-302. GPMC\_CONFIG6\_i**

<b>Address Offset</b>	0x0000 0074 + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0074 + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	WrAccessTime, WrDataOnADmuxBus, Cycle2Cycle and BusTurnAround parameters configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	WRACCESSTIME						RESERVED	WRDATAONADMUXBUS				RESERVED	CYCLE2CYCLEDELAY				CYCLE2CYCLESAMECSEN	CYCLE2CYCLEDIFFCSEN	RESERVED	BUSTURNAROUND										

Bits	Field Name	Description	Type	Reset
31	RESERVED	TI Internal use - Do not modify.	RW	0x1
30:29	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
28:24	WRACCESSTIME	Delay from start access time to the GPMC_FCLK rising edge corresponding the the GPMC_CLK rising edge used by the attached memory for the first data capture 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x0F
23:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
19:16	WRDATAONADMUXBUS	Specifies on which GPMC_FCLK rising edge the first data of the synchronous burst write is driven in the add/data mux bus	RW	0x7
15:12	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
11:8	CYCLE2CYCLEDELAY	Chip-select high pulse delay between successive accesses 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x0
7	CYCLE2CYCLESAMECSEN	Add CYCLE2CYCLEDELAY between successive accesses to the same CS (any access type) 0x0: No delay between the two accesses 0x1: Add CYCLE2CYCLEDELAY	RW	0x0
6	CYCLE2CYCLEDIFFCSEN	Add CYCLE2CYCLEDELAY between successive accesses to a different CS (any access type) 0x0: No delay between the two accesses 0x1: Add CYCLE2CYCLEDELAY	RW	0x0
5:4	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x0
3:0	BUSTURNAROUND	Bus turn around latency between successive accesses to the same CS (read to write) or to a different CS (read to read and read to write) 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x0

**Table 15-303. Register Call Summary for Register GPMC\_CONFIG6\_i**

## GPMC Functional Description

- [Idle Cycle Control Between Successive Accesses: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Read Cycle Time and Write Cycle Time \(RDCYCLETIME / WRCYCLETIME\): \[6\] \[7\]](#)
- [Access Time \(RDACCESSTIME / WRACCESSTIME\): \[8\]](#)
- [Access Time on Write Access: \[9\]](#)
- [Asynchronous Single-Read Operation on an Address/Data Multiplexed Device: \[10\] \[11\]](#)
- [Asynchronous Single Write Operation on an Address/Data-Multiplexed Device: \[12\]](#)
- [Asynchronous Single Write Operation on an AAD-Multiplexed Device: \[13\]](#)
- [Synchronous Single Write: \[14\]](#)
- [Synchronous Multiple \(Burst\) Write: \[15\] \[16\] \[17\]](#)

## GPMC Basic Programming Model

- [GPMC Timing Parameters: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[24\]](#)

**Table 15-304. GPMC\_CONFIG7\_i**

<b>Address Offset</b>	0x0000 0078 + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0078 + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	CS address mapping configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MASKADDRESS		RESERVED	CSVALID	BASEADDRESS											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000
11:8	MASKADDRESS	CS mask address. 0x1000: Chip-select size of 128 Mbytes 0x1100: Chip-select size of 64 Mbytes 0x1110: Chip-select size of 32 Mbytes 0x1111: Chip-select size of 16 Mbytes Other values must be avoided as they create holes in the chip-select address space.	RW	0xF
7	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
6	CSVALID	CS enable 0x0: CS disabled 0x1: CS enabled	RW	See <sup>(1)</sup>
5:0	BASEADDRESS	CSi base address where i = 0 to 7 (16M bytes minimum granularity) bits [5:0] corresponds to A29, A28, A27, A26, A25, and A24. See <a href="#">Figure 15-55</a>	RW	0x00

<sup>(1)</sup> Reset value is 0x1 for CS0 and 0x0 for CS1 to CS7

**Table 15-305. Register Call Summary for Register GPMC\_CONFIG7\_i**

## GPMC Functional Description

- [Chip-Select Base Address and Region Size: \[0\] \[1\] \[2\]](#)
- [Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode: \[3\]](#)

**Table 15-305. Register Call Summary for Register GPMC\_CONFIG7\_i (continued)**

- GPMC Basic Programming Model
- [GPMC Configuration in NOR Mode: \[4\] \[5\] \[6\]](#)
  - [GPMC Configuration in NAND mode: \[7\] \[8\] \[9\]](#)
  - [GPMC Timing Parameters: \[10\]](#)
- GPMC Register Manual
- [GPMC Register Summary: \[11\]](#)

**Table 15-306. GPMC\_NAND\_COMMAND\_i**

<b>Address Offset</b>	0x0000 007C + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 007C + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	This register is not a true register, just an address location.		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_COMMAND																															

Bits	Field Name	Description	Type	Reset
31:0	GPMC_NAND_COMMAND	This register is not a true register, just an address location. Writing data at the <a href="#">GPMC_NAND_COMMAND_i</a> location places the data as the NAND command value on the bus, using a regular asynchronous write access.	W	n/a

**Table 15-307. Register Call Summary for Register GPMC\_NAND\_COMMAND\_i**

- GPMC Functional Description
- [NAND Device Command and Address Phase Control: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
  - [Command Latch Cycle: \[5\]](#)
  - [General Facts About the Engine Configuration: \[6\]](#)
- GPMC Register Manual
- [GPMC Register Summary: \[7\]](#)
  - [GPMC Register Descriptions: \[8\]](#)

**Table 15-308. GPMC\_NAND\_ADDRESS\_i**

<b>Address Offset</b>	0x0000 0080 + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0080 + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	This register is not a true register, just an address location.		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_ADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	GPMC_NAND_ADDRESS	This register is not a true register, just an address location. Writing data at the <a href="#">GPMC_NAND_ADDRESS_i</a> location places the data as the NAND partial address value on the bus, using a regular asynchronous write access.	W	n/a

**Table 15-309. Register Call Summary for Register GPMC\_NAND\_ADDRESS\_i**

## GPMC Functional Description

- [NAND Device Command and Address Phase Control: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Address Latch Cycle: \[5\]](#)
- [General Facts About the Engine Configuration: \[6\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[7\]](#)
- [GPMC Register Descriptions: \[8\]](#)

**Table 15-310. GPMC\_NAND\_DATA\_i**

<b>Address Offset</b>	0x0000 0084 + (0x0000 0030 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0084 + (0x0000 0030 * i)	<b>Instance</b>	GPMC
<b>Description</b>	This register is not a true register, just an address location.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_DATA																															

Bits	Field Name	Description	Type	Reset
31:0	GPMC_NAND_DATA	This register is not a true register, just an address location. Reading data from the <a href="#">GPMC_NAND_DATA_i</a> location or from any location in the associated chip-select memory region activates an asynchronous read access.	W	n/a

**Table 15-311. Register Call Summary for Register GPMC\_NAND\_DATA\_i**

## GPMC Functional Description

- [NAND Device Data Read and Write Phase Control in Stream Mode: \[0\] \[1\] \[2\] \[3\]](#)
- [General Facts About the Engine Configuration: \[4\]](#)
- [Optimizing NAND Access Using the Prefetch and Write-Posting Engine: \[5\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[6\]](#)
- [GPMC Register Descriptions: \[7\] \[8\] \[9\]](#)

**Table 15-312. GPMC\_PREFETCH\_CONFIG1**

<b>Address Offset</b>	0x0000 01E0	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 01E0		
<b>Description</b>	Prefetch engine configuration 1		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CYCLOPTIMIZATION		ENABLEOPTIMIZEDACCESS	ENGINECSSELECTOR	PPFWENROUNDROBIN	RESERVED		PPFWWEIGHTDPRI				RESERVED	FIFOTHRESHOLD					ENABLEENGINE	RESERVED	WAITPINSELECTOR	SYNCHROMODE	DMAMODE	RESERVED	ACCESSMODE							

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
30:28	CYCLEOPTIMIZATION	Define the number of GPMC_FCLK cycles to be subtracted from RDCYCLETIME, WRCYCLETIME, ACCESSTIME, CSRDOFFTIME, CSWROFFTIME, ADVRDOFFTIME, ADVWROFFTIME, OEOFFTIME, WEOFFTIME 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x0
27	ENABLEOPTIMIZEDACCESS	Enables access cycle optimization 0x0: Access cycle optimization is disabled 0x1: Access cycle optimization is enabled	RW	0x0
26:24	ENGINECSSELECTOR	Selects the CS where Prefetch Postwrite engine is active 0x0: CS0 0x1: CS1 0x2: CS2 0x3: CS3 0x4: CS4 0x5: CS5 0x6: CS6 0x7: CS7	RW	0x0
23	PPWENROUNDROBIN	Enables the PFPW RoundRobin arbitration 0x0: Prefetch Postwrite engine round robin arbitration is disabled 0x1: Prefetch Postwrite engine round robin arbitration is enabled	RW	0x0
22:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
19:16	PPWWEIGHTEDPRIO	When an arbitration occurs between a DMA and a PFPW engine access, the DMA is always serviced. If the PFPWEnRoundRobin is enabled, 0x0: the next access is granted to the PFPW engine, 0x1: the two next accesses are granted to the PFPW engine, ... 0xF: the 16 next accesses are granted to the PFPW engine.	RW	0x0
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
14:8	FIFOTHRESHOLD	Selects the maximum number of bytes read from the FIFO or written to the FIFO by the host on a DMA or interrupt request 0x00: 0 byte 0x01: 1 byte ... 0x40: 64 bytes	RW	0x40
7	ENABLEENGINE	Enables the Prefetch Postwrite engine 0x0: Prefetch Postwrite engine is disabled 0x1: Prefetch Postwrite engine is enabled	RW	0x0
6	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
5:4	WAITPINSELECTOR	Select which wait pin edge detector should start the engine in synchronized mode 0x0: Selects Wait0 EdgeDetection 0x1: Selects Wait1 EdgeDetection 0x2: Selects Wait2 EdgeDetection 0x3: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
3	SYNCHROMODE	Selects when the engine starts the access to CS 0x0: Engine starts the access to CS as soon as STARTENGINE is set 0x1: Engine starts the access to CS as soon as STARTENGINE is set AND wait to non wait edge detection on the selected wait pin	RW	0x0
2	DMAMODE	Selects interrupt synchronization or DMA request synchronization 0x0: Interrupt synchronization is enabled. Only interrupt line will be activated on FIFO threshold crossing. 0x1: DMA request synchronization is enabled. A DMA request protocol is used.	RW	0x0
1	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
0	ACCESSMODE	Selects prefetch read or write-posting accesses 0x0: Prefetch read mode 0x1: Write-posting mode	RW	0x0

**Table 15-313. Register Call Summary for Register GPMC\_PREFETCH\_CONFIG1**

## GPMC Functional Description

- [GPMC Interrupt Requests: \[0\]](#)
- [Prefetch and Write-Posting Engine: \[1\] \[2\]](#)
- [General Facts About the Engine Configuration: \[3\] \[4\]](#)
- [Prefetch Mode: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\]](#)
- [FIFO Control in Prefetch Mode: \[17\]](#)
- [Write-Posting Mode: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\]](#)
- [FIFO Control in Write-Posting Mode: \[27\]](#)
- [Optimizing NAND Access Using the Prefetch and Write-Posting Engine: \[28\] \[29\] \[30\]](#)
- [Interleaved Accesses Between Prefetch and Write-Posting Engine and Other Chip-Selects: \[31\] \[32\]](#)

## GPMC Basic Programming Model

- [GPMC Configuration in NAND mode: \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[43\]](#)

**Table 15-314. GPMC\_PREFETCH\_CONFIG2**

<b>Address Offset</b>	0x0000 01E4	<b>Instance</b>	GPMC																																																												
<b>Physical Address</b>	0x5000 01E4																																																														
<b>Description</b>	Prefetch engine configuration 2																																																														
<b>Type</b>	RW																																																														
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="12">TRANSFERCOUNT</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																TRANSFERCOUNT											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
RESERVED																TRANSFERCOUNT																																															
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																											
31:14	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000																																																											
13:0	TRANSFERCOUNT	Selects the number of bytes to be read or written by the engine to the selected CS 0x0000: 0 byte 0x0001: 1 byte ... 0x2000: 8 Kbytes	RW	0x0000																																																											



**Table 15-315. Register Call Summary for Register GPMC\_PREFETCH\_CONFIG2**

GPMC Functional Description

- [Prefetch Mode: \[0\] \[1\]](#)
- [Write-Posting Mode: \[2\] \[3\]](#)

GPMC Basic Programming Model

- [GPMC Configuration in NAND mode: \[4\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[5\]](#)

**Table 15-316. GPMC\_PREFETCH\_CONTROL**

<b>Address Offset</b>	0x0000 01EC	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 01EC		
<b>Description</b>	Prefetch engine control		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STARTENGINE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000000
0	STARTENGINE	Resets the FIFO pointer and starts the engine Read 0x0: Engine is stopped Write 0x0 stops the engine Read 0x1: Engine is running Write 0x1 resets the FIFO pointer to 0x0 in prefetch mode and 0x40 in postwrite mode and starts the engine	RW	0x0

**Table 15-317. Register Call Summary for Register GPMC\_PREFETCH\_CONTROL**

GPMC Functional Description

- [General Facts About the Engine Configuration: \[0\]](#)
- [Prefetch Mode: \[1\] \[2\] \[3\]](#)
- [Write-Posting Mode: \[4\] \[5\] \[6\]](#)

GPMC Basic Programming Model

- [GPMC Configuration in NAND mode: \[7\] \[8\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[9\]](#)

**Table 15-318. GPMC\_PREFETCH\_STATUS**

<b>Address Offset</b>	0x0000 01F0	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 01F0		
<b>Description</b>	Prefetch engine status		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED								RESERVED								RESERVED								RESERVED								COUNTVALUE							
RESERVED								RESERVED								RESERVED								RESERVED								COUNTVALUE							

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
30:24	FIFOPOINTER	Number of available bytes to be read or number of free empty byte places to be written 0x00: 0 byte available to be read or 0 free empty place to be written ... 0x40: 64 bytes available to be read or 64 empty places to be written	R	0x00
23:17	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
16	FIFOTHRESHOLDSTATUS	Set when FIFOPointer exceeds FIFOThreshold value 0x0: FIFOPointer smaller or equal to FIFOThreshold. Writing to this bit has no effect 0x1: FIFOPointer greater than FIFOThreshold. Writing to this bit has no effect	R	0x0
15:14	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
13:0	COUNTVALUE	Number of remaining bytes to be read or to be written by the engine according to the TransferCount value 0x0000: 0 byte remaining to be read or to be written 0x0001: 1 byte remaining to be read or to be written ... 0x2000: 8 Kbytes remaining to be read or to be written	R	0x0000

**Table 15-319. Register Call Summary for Register GPMC\_PREFETCH\_STATUS**

## GPMC Functional Description

- [General Facts About the Engine Configuration: \[0\] \[1\]](#)
- [FIFO Control in Prefetch Mode: \[2\] \[3\] \[4\]](#)
- [FIFO Control in Write-Posting Mode: \[5\] \[6\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[7\]](#)
- [GPMC Register Descriptions: \[8\] \[9\]](#)

**Table 15-320. GPMC\_ECC\_CONFIG**

<b>Address Offset</b>	0x0000 01F4	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 01F4		
<b>Description</b>	ECC configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECCALGORITHM	RESERVED	ECCBCHTSEL	ECCWRAPMODE	ECC16B	ECCTOPSECTOR	ECCCS	ECCENABLE								

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0000
16	ECCALGORITHM	ECC algorithm used 0x0: Hamming code 0x1: BCH code	RW	0x0
15:14	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
13:12	ECCBCHTSEL	Error correction capability used for BCH 0x0: Up to 4 bits error correction (t = 4) 0x1: Up to 8 bits error correction (t = 8) 0x2: Up to 16 bits error correction (t = 16) 0x3: Reserved	RW	0x1
11:8	ECCWRAPMODE	Spare area organization definition for the BCH algorithm. See the BCH syndrome/parity calculator module functional specification for more details	RW	0x0
7	ECC16B	Selects an ECC calculated on 16 columns 0x0: ECC calculated on 8 columns 0x1: ECC calculated on 16 columns	RW	0x0
6:4	ECCTOPSECTOR	Number of sectors to process with the BCH algorithm 0x0: 1 sector (512kB page) 0x1: 2 sectors ... 0x3: 4 sectors (2kB page) ... 0x7: 8 sectors (4kB page)	RW	0x3
3:1	ECCCS	Selects the CS where ECC is computed 0x0: Chip-select 0 0x1: Chip-select 1 0x2: Chip-select 2 0x3: Chip-select 3 Other: Reserved	RW	0x0
0	ECCENABLE	Enables the ECC feature 0x0: ECC disabled 0x1: ECC enabled	RW	0x0

**Table 15-321. Register Call Summary for Register GPMC\_ECC\_CONFIG**

GPMC Functional Description

- [ECC Calculator: \[0\] \[1\]](#)
- [ECC Enabling: \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [ECC Calculation Based on 16-Bit Word: \[7\]](#)

**Table 15-321. Register Call Summary for Register GPMC\_ECC\_CONFIG (continued)**

GPMC Basic Programming Model

- [GPMC Configuration in NAND mode: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[14\]](#)
- [GPMC Register Descriptions: \[15\] \[16\] \[17\]](#)

**Table 15-322. GPMC\_ECC\_CONTROL**

<b>Address Offset</b>	0x0000 01F8	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 01F8		
<b>Description</b>	ECC control		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECCCLEAR	RESERVED				ECCPOINTER										

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
8	ECCCLEAR	Clear all ECC result registers Reads returns 0 Write 0x1 to this field clear all ECC result registers Write 0x0 is ignored	RW	0x0
7:4	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
3:0	ECCPOINTER	Selects ECC result register (Reads to this field give the dynamic position of the ECC pointer - Writes to this field select the ECC result register where the first ECC computation will be stored); Other enums: writing other values disables the ECC engine (ECCENABLE bit of <a href="#">GPMC_ECC_CONFIG</a> set to 0)  0x0: Writing 0x0 disables the ECC engine (ECCENABLE bit of <a href="#">GPMC_ECC_CONFIG</a> set to 0)  0x1: ECC result register 1 selected 0x2: ECC result register 2 selected 0x3: ECC result register 3 selected 0x4: ECC result register 4 selected 0x5: ECC result register 5 selected 0x6: ECC result register 6 selected 0x7: ECC result register 7 selected 0x8: ECC result register 8 selected 0x9: ECC result register 9 selected	RW	0x0

**Table 15-323. Register Call Summary for Register GPMC\_ECC\_CONTROL**

GPMC Functional Description

- [ECC Result Register and ECC Computation Accumulation Size: \[0\] \[1\]](#)
- [ECC Enabling: \[2\] \[3\]](#)

GPMC Basic Programming Model

- [GPMC Configuration in NAND mode: \[4\] \[5\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[6\]](#)

**Table 15-324. GPMC\_ECC\_SIZE\_CONFIG**

<b>Address Offset</b>	0x0000 01FC	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 01FC		
<b>Description</b>	ECC size		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				ECCSIZE1				RESERVED				ECCSIZE0				RESERVED				ECC9RESULTSIZ	ECC8RESULTSIZ	ECC7RESULTSIZ	ECC6RESULTSIZ	ECC5RESULTSIZ	ECC4RESULTSIZ	ECC3RESULTSIZ	ECC2RESULTSIZ	ECC1RESULTSIZ			

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
29:22	ECCSIZE1	Defines ECC size 1 0x00: 2 Bytes 0x01: 4 Bytes 0x02: 6 Bytes 0x03: 8 Bytes ... 0xFF: 512 Bytes	RW	0xFF
21:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
19:12	ECCSIZE0	Defines ECC size 0 0x00: 2 Bytes 0x01: 4 Bytes 0x02: 6 Bytes 0x03: 8 Bytes ... 0xFF: 512 Bytes	RW	0xFF
11:9	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
8	ECC9RESULTSIZ	Selects ECC size for ECC 9 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
7	ECC8RESULTSIZ	Selects ECC size for ECC 8 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
6	ECC7RESULTSIZ	Selects ECC size for ECC 7 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
5	ECC6RESULTSIZ	Selects ECC size for ECC 6 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
4	ECC5RESULTSIZ	Selects ECC size for ECC 5 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
3	ECC4RESULTSIZ	Selects ECC size for ECC 4 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
2	ECC3RESULTSIZ	Selects ECC size for ECC 3 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0

Bits	Field Name	Description	Type	Reset
1	ECC2RESULTSIZ	Selects ECC size for ECC 2 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
0	ECC1RESULTSIZ	Selects ECC size for ECC 1 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0

**Table 15-325. Register Call Summary for Register GPMC\_ECC\_SIZE\_CONFIG**

## GPMC Functional Description

- [ECC Result Register and ECC Computation Accumulation Size: \[0\]](#)
- [ECC Enabling: \[1\] \[2\] \[3\]](#)

## GPMC Basic Programming Model

- [GPMC Configuration in NAND mode: \[4\] \[5\]](#)

## GPMC Register Manual

- [GPMC Register Summary: \[6\]](#)

**Table 15-326. GPMC\_ECCj\_RESULT**

<b>Address Offset</b>	0x0000 0200 + (0x0000 0004 * (j - 1))	<b>Index</b>	j = 1 to 9
<b>Physical Address</b>	0x5000 0200 + (0x0000 0004 * (j - 1))	<b>Instance</b>	GPMC
<b>Description</b>	ECC result register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				P2048o	P1024o	P512o	P256o	P128o	P64o	P32o	P16o	P8o	P4o	P2o	P1o	RESERVED				P2048e	P1024e	P512e	P256e	P128e	P64e	P32e	P16e	P8e	P4e	P2e	P1e

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
27	P2048o	Odd row parity bit 2048, only used for ECC computed on 512 Bytes	R	0x0
26	P1024o	Odd row parity bit 1024	R	0x0
25	P512o	Odd row parity bit 512	R	0x0
24	P256o	Odd row parity bit 256	R	0x0
23	P128o	Odd row parity bit 128	R	0x0
22	P64o	Odd row parity bit 64	R	0x0
21	P32o	Odd row parity bit 32	R	0x0
20	P16o	Odd row parity bit 16	R	0x0
19	P8o	Odd row parity bit 8	R	0x0
18	P4o	Odd Column Parity bit 4	R	0x0
17	P2o	Odd Column Parity bit 2	R	0x0
16	P1o	Odd Column Parity bit 1	R	0x0
15:12	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
11	P2048e	Even row parity bit 2048, only used for ECC computed on 512 Bytes	R	0x0
10	P1024e	Even row parity bit 1024	R	0x0
9	P512e	Even row parity bit 512	R	0x0
8	P256e	Even row parity bit 256	R	0x0
7	P128e	Even row parity bit 128	R	0x0
6	P64e	Even row parity bit 64	R	0x0

Bits	Field Name	Description	Type	Reset
5	P32e	Even row parity bit 32	R	0x0
4	P16e	Even row parity bit 16	R	0x0
3	P8e	Even row parity bit 8	R	0x0
2	P4e	Even column parity bit 4	R	0x0
1	P2e	Even column parity bit 2	R	0x0
0	P1e	Even column parity bit 1	R	0x0

**Table 15-327. Register Call Summary for Register GPMC\_ECCj\_RESULT**

GPMC Functional Description

- [ECC Result Register and ECC Computation Accumulation Size: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [ECC Computation: \[7\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[8\]](#)

**Table 15-328. GPMC\_BCH\_RESULT0\_i**

<b>Address Offset</b>	0x0000 0240 + (0x0000 0010 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0240 + (0x0000 0010 * i)	<b>Instance</b>	GPMC
<b>Description</b>	BCH ECC result (bits 0 to 31)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_0																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_0	BCH ECC result (bits 0 to 31)	RW	0x00000000

**Table 15-329. Register Call Summary for Register GPMC\_BCH\_RESULT0\_i**

GPMC Functional Description

- [Memory-Mapping of the BCH Codeword: \[0\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[1\]](#)

**Table 15-330. GPMC\_BCH\_RESULT1\_i**

<b>Address Offset</b>	0x0000 0244 + (0x0000 0010 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0244 + (0x0000 0010 * i)	<b>Instance</b>	GPMC
<b>Description</b>	BCH ECC result (bits 32 to 63)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_1																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_1	BCH ECC result (bits 32 to 63)	RW	0x00000000

**Table 15-331. Register Call Summary for Register GPMC\_BCH\_RESULT1\_i**

GPMC Functional Description

- [Memory-Mapping of the BCH Codeword: \[0\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[1\]](#)

**Table 15-332. GPMC\_BCH\_RESULT2\_i**

<b>Address Offset</b>	0x0000 0248 + (0x0000 0010 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0248 + (0x0000 0010 * i)	<b>Instance</b>	GPMC
<b>Description</b>	BCH ECC result (bits 64 to 95)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_2																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_2	BCH ECC result (bits 64 to 95)	RW	0x00000000

**Table 15-333. Register Call Summary for Register GPMC\_BCH\_RESULT2\_i**

GPMC Functional Description

- [Memory-Mapping of the BCH Codeword: \[0\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[1\]](#)

**Table 15-334. GPMC\_BCH\_RESULT3\_i**

<b>Address Offset</b>	0x0000 024C + (0x0000 0010 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 024C + (0x0000 0010 * i)	<b>Instance</b>	GPMC
<b>Description</b>	BCH ECC result (bits 96 to 127)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_3																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_3	BCH ECC result (bits 96 to 127)	RW	0x00000000

**Table 15-335. Register Call Summary for Register GPMC\_BCH\_RESULT3\_i**

GPMC Functional Description

- [Memory-Mapping of the BCH Codeword: \[0\]](#)

GPMC Register Manual

- [GPMC Register Summary: \[1\]](#)

**Table 15-336. GPMC\_BCH\_RESULT4\_i**

<b>Address Offset</b>	0x0000 0300 + (0x0000 0010 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0300 + (0x0000 0010 * i)	<b>Instance</b>	GPMC
<b>Description</b>	BCH ECC result (bits 128 to 159)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_4																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_4	BCH ECC result (bits 128 to 159)	RW	0x00000000



**Table 15-337. Register Call Summary for Register GPMC\_BCH\_RESULT4\_i**

GPMC Register Manual

- [GPMC Register Summary: \[0\]](#)

**Table 15-338. GPMC\_BCH\_RESULT5\_i**

<b>Address Offset</b>	0x0000 0304 + (0x0000 0010 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0304 + (0x0000 0010 * i)	<b>Instance</b>	GPMC
<b>Description</b>	BCH ECC result (bits 160 to 191)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_5																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_5	BCH ECC result (bits 160 to 191)	RW	0x00000000

**Table 15-339. Register Call Summary for Register GPMC\_BCH\_RESULT5\_i**

GPMC Register Manual

- [GPMC Register Summary: \[0\]](#)

**Table 15-340. GPMC\_BCH\_RESULT6\_i**

<b>Address Offset</b>	0x0000 0308 + (0x0000 0010 * i)	<b>Index</b>	i = 0 to 3
<b>Physical Address</b>	0x5000 0308 + (0x0000 0010 * i)	<b>Instance</b>	GPMC
<b>Description</b>	BCH ECC result (bits 192 to 207)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCH_RESULT_6															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Read returns 0s.	R	0x0000
15:0	BCH_RESULT_6	BCH ECC result (bits 192 to 207)	RW	0x0000

**Table 15-341. Register Call Summary for Register GPMC\_BCH\_RESULT6\_i**

GPMC Register Manual

- [GPMC Register Summary: \[0\]](#)

**Table 15-342. GPMC\_BCH\_SWDATA**

<b>Address Offset</b>	0x0000 02D0	<b>Instance</b>	GPMC
<b>Physical Address</b>	0x5000 02D0		
<b>Description</b>	This register is used to directly pass data to the BCH ECC calculator without accessing the actual NAND flash interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCH_DATA															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Read returns 0s.	R	0x0000
15:0	BCH_DATA	Data to be included in the BCH calculation. Only bits 0 to 7 are taken into account if the calculator is configured to use 8 bits data ( <a href="#">GPMC_ECC_CONFIG</a> [7] ECC16B = 0)	RW	0x0000

**Table 15-343. Register Call Summary for Register GPMC\_BCH\_SWDATA**

GPMC Register Manual

- [GPMC Register Summary: \[0\]](#)

## 15.5 Error Location Module

### 15.5.1 Error Location Module Overview

Non-managed NAND flash memories can be dense and nonvolatile in their own nature, but error-prone. When reading from NAND flash memories, some level of error-correction is required. In the case of NAND modules with no internal correction capability, sometimes referred to as *bare NANDs*, the correction process is delegated to the memory controller.

The general-purpose memory controller (GPMC) probes data read from an external NAND flash and uses this to compute checksum-like information, called syndrome polynomials, on a per-block basis. Each syndrome polynomial gives a status of the read operations for a full block, including 512 bytes of data, parity bits, and an optional spare-area data field, with a maximum block size of 1023 bytes. Computation is based on a Bose-Chaudhuri-Hocquenghem (BCH) algorithm. The error-location module (ELM) extracts error addresses from these syndrome polynomials.

Based on the syndrome polynomial value, the ELM can detect errors, compute the number of errors, and give the location of each error bit. The actual data is not required to complete the error-correction algorithm. Errors can be reported anywhere in the NAND flash block, including in the parity bits.

The maximum acceptable number of errors that can be corrected depends on a programmable configuration parameter. 4-, 8-, and 16-bit error-correction levels are supported. The ELM relies on a static and fixed definition of the generator polynomial for each error-correction level that corresponds to the generator polynomials defined in the GPMC (there are three fixed polynomial for the three correction error levels). A larger number of errors than the programmed error-correction level may be detected, but the ELM cannot correct them all. The offending block is then tagged as *uncorrectable* in the associated computation exit status register. If the computation is successful, that is, if the number of errors detected does not exceed the maximum value authorized for the chosen correction capability, the exit status register contains the information on the number of detected errors.

When the error-location process completes, an interrupt is triggered to inform the central processing unit (CPU) that its status can be checked. The number of detected errors and their locations in the NAND block can be retrieved from the module through register accesses.

#### 15.5.1.1 ELM Features

The ELM has the following features:

- 4, 8, and 16 bits per 512-byte block error-location based on BCH algorithms
- Eight simultaneous processing contexts
- Page-based and continuous modes
- Interrupt generation on error-location process completion:
  - When the full page has been processed in page mode
  - For each syndrome polynomial in continuous mode

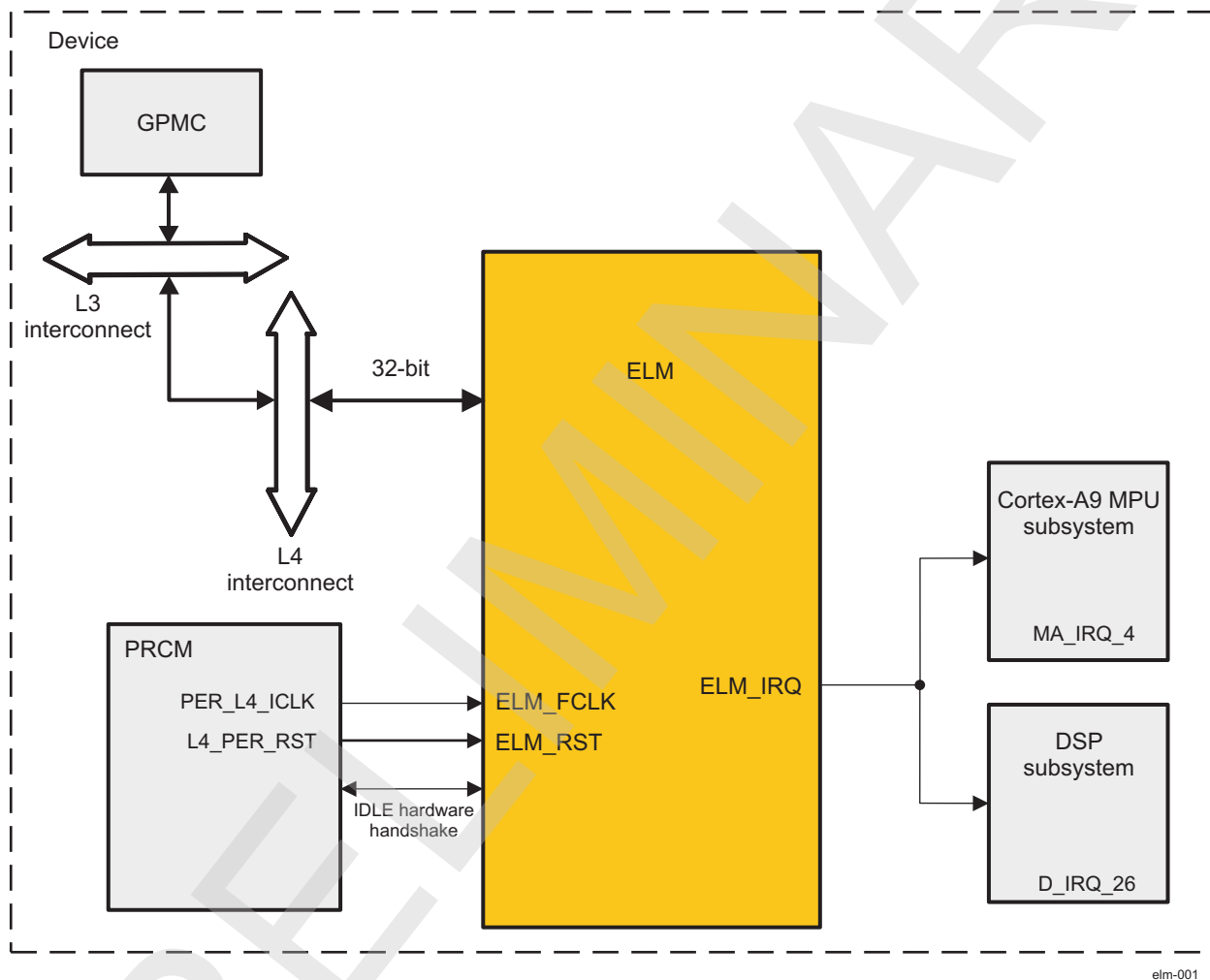
### 15.5.2 ELM Integration

The ELM extracts error addresses from generated syndrome polynomials.

The ELM is used with the GPMC. Syndrome polynomials generated on-the-fly when reading a NAND flash page and stored in GPMC registers are passed to the ELM. The microprocessor unit (MPU) can then correct the data block by flipping the bits to which the ELM error-location outputs point.

Figure 15-100 shows the integration of the ELM subsystem in the device.

**Figure 15-100. ELM Integration**



**Table 15-344. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
ELM	PD_L4_PER	No	NA

**Table 15-345. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
ELM	ELM_FCLK	PER_L4_ICLK	PRCM	Functional clock For information about PRCM clock gating and management, see <a href="#">Section 3.6.8, CD_L4_PER Clock Domain</a> in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .

Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
ELM	ELM_RST	L4_PER_RST	PRCM	Power domain hardware reset For information about PRCM reset sources and distribution, see <a href="#">Section 3.5.4, Reset Domains</a> in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .

**Table 15-346. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
ELM	ELM_IRQ	MA_IRQ_4	Cortex-A9	BCH error-location module interrupt For information about Cortex-A9 interrupt control, see <a href="#">Section 17.3.2, Interrupt Requests to Cortex-A9 MPU INTC</a> .
		D_IRQ_26	DSP	BCH error-location module interrupt For information about DSP interrupt control, see <a href="#">Interrupts Requests</a> .

**15.5.3 ELM Functional Description**

The ELM is designed around the error-location engine, which handles the computation based on the input syndrome polynomials.

The ELM maps the error-location engine to a standard interconnect interface by using a set of registers to control inputs and outputs.

**15.5.3.1 ELM Software Reset**

To perform a software reset, write a 1 to the [ELM\\_SYSCONFIG\[1\] SOFTRESET](#) bit. The [ELM\\_SYSSTATUS\[0\] RESETDONE](#) bit indicates that the software reset is complete when its value is 1. When the software reset completes, the [ELM\\_SYSCONFIG\[1\] SOFTRESET](#) bit is automatically reset.

**15.5.3.2 ELM Power Management**

[Table 15-347](#) describes the power-management features available to the ELM module.

**NOTE:**

- For information about source clock gating and sleep/wake-up transitions description, see the [Section 3.6.8, CD\\_L4\\_PER Clock Domain](#) in [Chapter 3, Power, Reset, and Clock Management](#).
- For a general description of EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see the [Section 3.6 Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#)

**Table 15-347. Local Power Management Features**

Feature	Registers	Description
Clock autogating	<a href="#">ELM_SYSCONFIG</a> [0] AUTOGATING bit	This bit allows a local power optimization inside the module by gating the ELM_FCLK clock upon the interface activity.
Slave idle modes	<a href="#">ELM_SYSCONFIG</a> [4:3] SIDLEMODE bit field	Force-idle, No-idle, and Smart-idle modes are available.
Clock activity	<a href="#">ELM_SYSCONFIG</a> [8] CLOCKACTIVITY bit	The clock can be switched-off or maintained during the wake-up period.
Master Standby modes	N/A	
Global Wake-up Enable	N/A	
Wake-up Sources Enable	N/A	

**CAUTION**

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the ELM CLOCKACTIVITY and ELM clock PRCM control bits. For a description of the ClockActivity feature, see [Chapter 3, Power, Reset, and Clock Management](#).

**15.5.3.3 ELM Interrupt Requests**

[Table 15-348](#) lists the event flags, and their masks, that can cause module interrupts.

**Table 15-348. Events**

Event Flag	Event Mask	Map to	Description
<a href="#">ELM_IRQSTATUS</a> [8] PAGE_VALID	<a href="#">ELM_IRQENABLE</a> [8] PAGE_MASK	ELM_IRQ	Page interrupt
<a href="#">ELM_IRQSTATUS</a> [7] LOC_VALID_7	<a href="#">ELM_IRQENABLE</a> [7] LOCATION_MASK_7	ELM_IRQ	Error-location interrupt for syndrome polynomial 7
<a href="#">ELM_IRQSTATUS</a> [6] LOC_VALID_6	<a href="#">ELM_IRQENABLE</a> [6] LOCATION_MASK_6	ELM_IRQ	Error-location interrupt for syndrome polynomial 6
<a href="#">ELM_IRQSTATUS</a> [5] LOC_VALID_5	<a href="#">ELM_IRQENABLE</a> [5] LOCATION_MASK_5	ELM_IRQ	Error-location interrupt for syndrome polynomial 5
<a href="#">ELM_IRQSTATUS</a> [4] LOC_VALID_4	<a href="#">ELM_IRQENABLE</a> [4] LOCATION_MASK_4	ELM_IRQ	Error-location interrupt for syndrome polynomial 4
<a href="#">ELM_IRQSTATUS</a> [3] LOC_VALID_3	<a href="#">ELM_IRQENABLE</a> [3] LOCATION_MASK_3	ELM_IRQ	Error-location interrupt for syndrome polynomial 3
<a href="#">ELM_IRQSTATUS</a> [2] LOC_VALID_2	<a href="#">ELM_IRQENABLE</a> [2] LOCATION_MASK_2	ELM_IRQ	Error-location interrupt for syndrome polynomial 2
<a href="#">ELM_IRQSTATUS</a> [1] LOC_VALID_1	<a href="#">ELM_IRQENABLE</a> [1] LOCATION_MASK_1	ELM_IRQ	Error-location interrupt for syndrome polynomial 1
<a href="#">ELM_IRQSTATUS</a> [0] LOC_VALID_0	<a href="#">ELM_IRQENABLE</a> [0] LOCATION_MASK_0	ELM_IRQ	Error-location interrupt for syndrome polynomial 0

**15.5.3.4 Processing Initialization**

[ELM\\_LOCATION\\_CONFIG](#) global setting parameters must be set before using the error-location engine. The [ELM\\_LOCATION\\_CONFIG](#)[1:0] ECC\_BCH\_LEVEL bit defines the error-correction level used (4-, 8-, or 16-bit error-correction). The [ELM\\_LOCATION\\_CONFIG](#)[26:16] ECC\_SIZE bit field defines the maximum buffer length beyond which the engine processing no longer looks for errors.

The CPU can choose to use the ELM in continuous mode or page mode. If all [ELM\\_PAGE\\_CTRL](#)[i] SECTOR\_i bits are reset (i is the syndrome polynomial number, i = 0 to 7), continuous mode is used. In any other case, page mode is implicitly selected.

- Continuous mode: Each syndrome polynomial is processed independently – results for a syndrome can be retrieved and acknowledged at any time, whatever the status of the other seven processing contexts.
- Page mode: Syndrome polynomials are grouped into atomic entities – only one page can be processed at any given time, even if all eight contexts are not used for this page. Unused contexts are lost and cannot be affected to any other processing. The full page must be acknowledged and cleared before moving to the next page.

For completion interrupts to be generated correctly, all `ELM_IRQENABLE[i] LOCATION_MASK_i` bits (i = 0 to 7) must be forced to 0 when in page mode, and set to 1 in continuous mode. Additionally, the `ELM_IRQENABLE[8] PAGE_MASK` bit must be set to 1 when in page mode.

The CPU initiates error-location processing by writing a syndrome polynomial into one of the eight possible register sets. Each of these register sets includes seven registers: `ELM_SYNDROME_FRAGMENT_0_i` to `ELM_SYNDROME_FRAGMENT_6_i`. The first six registers can be written in any order, but `ELM_SYNDROME_FRAGMENT_6_i` must be written last because it includes the validity bit, which instructs the ELM that this syndrome polynomial must be processed (the `ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID` bit).

As soon as one validity bit is asserted (`ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x1`, with i = 0 to 7), error-location processing can start for the corresponding syndrome polynomial. The associated `ELM_LOCATION_STATUS_i` and `ELM_ERROR_LOCATION_0_i` to `ELM_ERROR_LOCATION_15_i` registers are not reset (i = 0 to 7). The software must not consider them until the corresponding `ELM_IRQSTATUS[i] LOC_VALID_i` bit is set.

### 15.5.3.5 Processing Sequence

While the error-location engine is busy processing one syndrome polynomial, further syndrome polynomials can be written. They are processed when the current processing completes.

The engine completes early when:

- No error is detected; that is, when the `ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE` bit is set to 1 and the `ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS` bit field is set to 0x0.
- Too many errors are detected; that is, when the `ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE` bit is set to 0 while the `ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS` bit field is set with the value output by the error-location engine. The reported number of errors is not ensured if `ECC_CORRECTABLE` is 0.

If the engine completes early, the associated error-location registers `ELM_ERROR_LOCATION_0_i` to `ELM_ERROR_LOCATION_15_i` are not updated (i = 0 to 7).

In all other cases, the engine goes through the entire error-location process. Each time an error-location is found, it is logged in the associated `ECC_ERROR_LOCATION` bit field. The first error detected is logged in the `ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION` bit field; the second in the `ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION` bit field, and so on.

**Table 15-349. ELM\_LOCATION\_STATUS\_i Value Decoding Table**

ECC_CORRECTABLE Value	ECC_NB_ERRORS Value	Status	Number of Errors Detected	Action Required
1	0	OK	0	None
1	≠ 0	OK	ECC_NB_ERRORS	Correct the data buffer read based on the <code>ELM_ERROR_LOCATION_0_i</code> to <code>ELM_ERROR_LOCATION_15_i</code> results.
0	Any	Failed	Unknown	Software-dependant

### 15.5.3.6 Processing Completion

When the processing for a given syndrome polynomial completes, its `ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID` bit is reset. It must not be set again until the exit status registers, `ELM_LOCATION_STATUS_i` (i = 0 to 7), for this processing are checked. Failure to comply with this rule leads to potential loss of the first polynomial process data output.

The error-location engine signals the process completion to the ELM. When this event is detected, the corresponding `ELM_IRQSTATUS[i] LOC_VALID_i` bit is set ( $i = 0$  to  $7$ ). The processing-exit status is available from the associated `ELM_LOCATION_STATUS_i` register, and error locations are stored in order in the `ECC_ERROR_LOCATION` fields. The software must only read valid error-location registers based on the number of errors detected and located.

Immediately after the error-location engine completes, a new syndrome polynomial can be processed, if any is available, as reported by the `ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID` validity bit, depending on the configured error-correction level. If several syndrome polynomials are available, a round-robin arbitration is used to select one for processing.

In continuous mode (that is, all bits in `ELM_PAGE_CTRL` are reset), an interrupt is triggered whenever a `ELM_IRQSTATUS[i] LOC_VALID_i` bit is asserted. The CPU must read the `ELM_IRQSTATUS` register to determine which polynomial is processed and retrieve the exit status and error locations (`ELM_LOCATION_STATUS_i` and `ELM_ERROR_LOCATION_0_i` to `ELM_ERROR_LOCATION_15_i`). When done, the CPU must clear the corresponding `ELM_IRQSTATUS[i] LOC_VALID_i` bit by writing it to 1. Other status bits must be written to 0 so that other interrupts are not unintentionally cleared. When using this mode, the `ELM_IRQSTATUS[8] PAGE_VALID` interrupt is never triggered.

In page mode, the module does not trigger interrupts for the processing completion of each polynomial because the `ELM_IRQENABLE[i] LOCATION_MASK_i` bits are cleared. A page is defined using the `ELM_PAGE_CTRL` register. Each `SECTOR_i` bit set means the corresponding polynomial  $i$  is part of the page processing. A page is fully processed when all tagged polynomials have been processed, as logged in the `ELM_IRQSTATUS[i] LOC_VALID_i` bit fields. The module triggers an `ELM_IRQSTATUS[8] PAGE_VALID` interrupt whenever it detects that the full page has been processed. To make sure the next page can be correctly processed, all status bits in the `ELM_IRQSTATUS` register must be cleared by using a single atomic-write access.

---

**NOTE:** Do not modify page setting parameters in the `ELM_PAGE_CTRL` register unless the engine is idle, no polynomial input is valid, and all interrupts have been cleared.

---

Because no polynomial-level interrupt is triggered in page mode, polynomials cleared in the `ELM_PAGE_CTRL[i] SECTOR_i` bit fields ( $i = 0$  to  $7$ ) are processed as usual, but are essentially ignored. The CPU must manually poll the `ELM_IRQSTATUS` bits to check for their status.

## 15.5.4 ELM Basic Programming Model

### 15.5.4.1 ELM Low Level Programming Model

#### 15.5.4.1.1 Processing Initialization

**Table 15-350. ELM Processing Initialization**

Step	Register/ Bit Field / Programming Model	Value
Resets the module	<code>ELM_SYSCONFIG[1] SOFTRESET</code>	0x1
Wait until reset is done.	<code>ELM_SYSSTATUS[0] RESETDONE</code>	0x1
Configure the slave interface power management.	<code>ELM_SYSCONFIG[4:3] SIDLEMODE</code>	Set value
Defines the error-correction level used	<code>ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL</code>	Set value
Defines the maximum buffer length	<code>ELM_LOCATION_CONFIG[26:16] ECC_SIZE</code>	Set value
Sets the ELM in continuous mode or page mode	<code>ELM_PAGE_CTRL</code>	Set value
<b>If</b> continuous mode is used	All <code>ELM_PAGE_CTRL[i] SECTOR_i</code> ( $i = 0$ to $7$ )	0x0
Enables interrupt for syndrome polynomial $i$	<code>ELM_IRQENABLE[i] LOCATION_MASK_i</code>	0x1
<b>else</b> (page mode is used)	One syndrome polynomial $i$ is set <code>ELM_PAGE_CTRL[i] SECTOR_i</code> ( $i = 0$ to $7$ )	0x1
Disable all interrupts for syndrome polynomial and enable <code>PAGE_MASK</code> interrupt.	All <code>ELM_IRQENABLE[i] LOCATION_MASK_i = 0x0</code> and <code>ELM_IRQENABLE[8] PAGE_MASK = 0x1</code>	Set value
<b>endif</b>		Set value



**Table 15-350. ELM Processing Initialization (continued)**

Step	Register/ Bit Field / Programming Model	Value
Set the input syndrome polynomial i.	ELM_SYNDROME_FRAGMENT_0_i	Set value
	ELM_SYNDROME_FRAGMENT_1_i	Set value
	ELM_SYNDROME_FRAGMENT_5_i	Set value
	ELM_SYNDROME_FRAGMENT_6_i	Set value
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID	0x1

**15.5.4.1.2 Read Results**

The engine goes through the entire error-location process and results can be read. [Table 15-351](#) and [Table 15-352](#) describe the processing completion for continuous and page modes, respectively.

**Table 15-351. ELM Processing Completion for Continuous Mode**

Step	Register/ Bit Field / Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Read for which i the error-location process is complete.	ELM_IRQSTATUS[i] LOC_VALID_i	0x1
<b>if</b> the process fails (too many errors) It is software dependant.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x0
<b>else</b> (process successful, the engine completes)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS	
Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers. It is the software responsibility to correct errors in the data buffer.	ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION	
<b>endif</b>		
Clear the corresponding i interrupt.	ELM_IRQSTATUS[i] LOC_VALID_i	0x1

A new syndrome polynomial can be processed after the end of processing ([ELM\\_SYNDROME\\_FRAGMENT\\_6\\_i\[16\]](#) SYNDROME\_VALID = 0x0) and after the exit status register check ([ELM\\_LOCATION\\_STATUS\\_i](#)).

**Table 15-352. ELM Processing Completion for Page Mode**

Step	Register/ Bit Field / Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTATUS[8] PAGE_VALID	0x1
<b>Repeat</b> the following actions the necessary number of times. That is, once for each valid defined block in the page.		
Read the process exit status.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	
<b>if</b> the process fails (too many errors) It is software dependant.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x0
<b>else</b> (process successful, the engine completes)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS	

**Table 15-352. ELM Processing Completion for Page Mode (continued)**

Step	Register/ Bit Field / Programming Model	Value
Read the error-location bit addresses for syndrome polynomial <i>i</i> of the ECC_NB_ERRORS first registers.	ELM_ERROR_LOCATION_0_ <i>i</i> [12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_ <i>i</i> [12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_ <i>i</i> [12:0] ECC_ERROR_LOCATION	
<b>endif</b>		
<b>End Repeat</b>		
Clear the ELM_IRQSTATUS register.	ELM_IRQSTATUS	0x1FF

Next page can be correctly processed after a page is fully processed, when all tagged polynomials have been processed (ELM\_IRQSTATUS[*i*] LOC\_VALID\_*i* = 0x1 for all syndrome polynomials *i* used in the page).

#### 15.5.4.2 Use Case: ELM Used in Continuous Mode

In this example, the ELM module is programmed for an 8-bit error-correction capability in continuous mode. After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, a non-zero polynomial syndrome is reported from the GPMC (Polynomial syndrome 0 is used in the ELM):

- P = 0x0A16ABE115E44F767BFB0D0980

**Table 15-353. Use Case: Continuous Mode**

Step	Register/ Bit Field / Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2
Defines the error-correction level used: 8 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x1
Defines the maximum buffer length: 528 bytes (2x528 = 1056)	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in continuous mode	ELM_PAGE_CTRL	0
Enables interrupt for syndrome polynomial 0	ELM_IRQENABLE[0] LOCATION_MASK_0	0x1
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_ <i>i</i> (i=0)	0xFB0D0980
	ELM_SYNDROME_FRAGMENT_1_ <i>i</i> (i=0)	0xE44F767B
	ELM_SYNDROME_FRAGMENT_2_ <i>i</i> (i=0)	0x16ABE115
	ELM_SYNDROME_FRAGMENT_3_ <i>i</i> (i=0)	0x0000000A
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_ <i>i</i> [16] SYNDROME_VALID (i=0)	0x1
Wait until process is complete for syndrome polynomial 0: IRQ_ELM is generated or poll the status register.		
Read that error-location process is complete for syndrome polynomial 0.	ELM_IRQSTATUS[0] LOC_VALID_0	0x1
Read the process exit status: All errors were successfully located.	ELM_LOCATION_STATUS_ <i>i</i> [8] ECC_CORRECTABLE (i=0)	0x1
Read the number of errors: Four errors detected.	ELM_LOCATION_STATUS_ <i>i</i> [4:0] ECC_NB_ERRORS (i=0)	0x4
Read the error-location bit addresses for syndrome polynomial 0 of the 4 first registers: Errors are located in the data buffer at decimal addresses 431, 1062, 1909, 3452.	ELM_ERROR_LOCATION_0_ <i>i</i> (i=0)	0x1AF
	ELM_ERROR_LOCATION_1_ <i>i</i> (i=0)	0x426
	ELM_ERROR_LOCATION_2_ <i>i</i> (i=0)	0x775
	ELM_ERROR_LOCATION_3_ <i>i</i> (i=0)	0xD7C

**Table 15-353. Use Case: Continuous Mode (continued)**

Step	Register/ Bit Field / Programming Model	Value
Clear the corresponding interrupt for polynomial 0.	ELM_IRQSTATUS[0] LOC_VALID_0	0x1

The NAND flash data in the sector are seen as a polynomial of degree 4223 (number of bits in a 528 byte buffer minus 1), with each data bit being a coefficient in the polynomial. When reading from a NAND flash using the GPMC module, computation of the polynomial syndrome assumes that the first NAND word read at address 0x0 contains the highest-order coefficient in the message. Furthermore, in the 16-bit NAND word, bits are ordered from bit 7 to bit 0, then from bit 15 to bit 8. Based on this convention, an address table of the data buffer can be built. NAND memory addresses in Table 15-354 are given in decimal format.

**Table 15-354. 16-bit NAND Sector Buffer Address Map**

NAND Memory Address	Message bit addresses in the memory word															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	4215	4214	4213	4212	4211	4210	4209	4208	4223	4222	4221	4220	4219	4218	4217	4216
1	4175	4174	4173	4172	4171	4170	4169	4168	4183	4182	4181	4180	4179	4178	4177	4176
...																
47	3463	3462	3461	3460	3459	3458	3457	3456	3471	3470	3469	3468	3467	3466	3465	3464
48	3447	3446	3445	3444	3443	3442	3441	3440	3455	3454	3453	3452	3451	3450	3449	3448
49	3431	3430	3429	3428	3427	3426	3425	3424	3439	3438	3437	3436	3435	3434	3433	3432
50	3415	3414	3413	3412	3411	3410	3409	3408	3423	3422	3421	3420	3419	3418	3417	3416
...																
255	135	134	133	132	131	130	129	128	143	142	141	140	139	138	137	136
256	119	118	117	116	115	114	113	112	127	126	125	124	123	122	121	120
257	103	102	101	100	99	98	97	96	111	110	109	108	107	106	105	104
258	87	86	85	84	83	82	81	80	95	94	93	92	91	90	89	88
259	71	70	69	68	67	66	65	64	79	78	77	76	75	74	73	72
260	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56
261	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40
262	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
263	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8

The table can now be used to determine which bits in the buffer were incorrect and must be flipped. In this example, the first bit to be flipped is bit 4 from the 49th byte read from memory. It is up to the processor to correctly map this word to the copied buffer and to flip this bit. The same process must be repeated for all detected errors.

**15.5.4.3 Use Case: ELM Used in Page Mode**

In this example, the ELM module is programmed for an 16-bit error-correction capability in page mode. After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, four non-zero polynomial syndromes are reported from the GPMC (Polynomial syndrome 0, 1, 2, and 3 are used in the ELM):

- P0 = 0xE8B0 12ADDB5A318E05BE B0693DB28330B5CC A329AA05E0B718EF
- P1 = 0xBAD0 49A0D932C22E6669 0948DF08BE093336 79C6BA10E5F935EB
- P2 = 0x69D9 B86ABCD5EC3697FA A6498FEE54556EA0 1579EF7D60BA3189
- P3 = 0x0

**Table 15-355. Use Case: Page Mode**

Step	Register/ Bit Field / Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1

**Table 15-355. Use Case: Page Mode (continued)**

Step	Register/ Bit Field / Programming Model	Value
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2
Defines the error-correction level used: 16 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x2
Defines the maximum buffer length: 528 bytes	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in page mode (4 blocks in a page)	ELM_PAGE_CTRL[0] SECTOR_0	0x1
	ELM_PAGE_CTRL[1] SECTOR_1	0x1
	ELM_PAGE_CTRL[2] SECTOR_2	0x1
	ELM_PAGE_CTRL[3] SECTOR_3	0x1
Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt.	ELM_IRQENABLE	0x100
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_i (i=0)	0xE0B718EF
	ELM_SYNDROME_FRAGMENT_1_i (i=0)	0xA329AA05
	ELM_SYNDROME_FRAGMENT_2_i (i=0)	0x8330B5CC
	ELM_SYNDROME_FRAGMENT_3_i (i=0)	0xB0693DB2
	ELM_SYNDROME_FRAGMENT_4_i (i=0)	0x318E05BE
	ELM_SYNDROME_FRAGMENT_5_i (i=0)	0x12ADDB5A
	ELM_SYNDROME_FRAGMENT_6_i (i=0)	0xE8B0
Set the input syndrome polynomial 1.	ELM_SYNDROME_FRAGMENT_0_i (i=1)	0xE5F935EB
	ELM_SYNDROME_FRAGMENT_1_i (i=1)	0x79C6BA10
	ELM_SYNDROME_FRAGMENT_2_i (i=1)	0xBE093336
	ELM_SYNDROME_FRAGMENT_3_i (i=1)	0x0948DF08
	ELM_SYNDROME_FRAGMENT_4_i (i=1)	0xC22E6669
	ELM_SYNDROME_FRAGMENT_5_i (i=1)	0x49A0D932
	ELM_SYNDROME_FRAGMENT_6_i (i=1)	0xBAD0
Set the input syndrome polynomial 2.	ELM_SYNDROME_FRAGMENT_0_i (i=2)	0x60BA3189
	ELM_SYNDROME_FRAGMENT_1_i (i=2)	0x1579EF7D
	ELM_SYNDROME_FRAGMENT_2_i (i=2)	0x54556EA0
	ELM_SYNDROME_FRAGMENT_3_i (i=2)	0xA6498FEE
	ELM_SYNDROME_FRAGMENT_4_i (i=2)	0xEC3697FA
	ELM_SYNDROME_FRAGMENT_5_i (i=2)	0xB86ABCD5
	ELM_SYNDROME_FRAGMENT_6_i (i=2)	0x69D9
Set the input syndrome polynomial 3.	ELM_SYNDROME_FRAGMENT_0_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_1_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_2_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_3_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_4_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_5_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_6_i (i=3)	0x0
Initiates the computation process for syndrome polynomial 0	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=0)	0x1
Initiates the computation process for syndrome polynomial 1	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=1)	0x1
Initiates the computation process for syndrome polynomial 2	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=2)	0x1
Initiates the computation process for syndrome polynomial 3	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=3)	0x1
Wait until process is complete for syndrome polynomial 0, 1, 2, and 3: Wait until the ELM_IRQ interrupt is generated or poll the status register.		

**Table 15-355. Use Case: Page Mode (continued)**

Step	Register/ Bit Field / Programming Model	Value
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTATUS[8] PAGE_VALID	0x1
Read the process exit status for syndrome polynomial 0: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=0)	0x1
Read the process exit status for syndrome polynomial 1: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=1)	0x1
Read the process exit status for syndrome polynomial 2: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=2)	0x1
Read the process exit status for syndrome polynomial 3: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=3)	0x1
Read the number of errors for syndrome polynomial 0: 4 errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=0)	0x4
Read the number of errors for syndrome polynomial 1: 2 errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=1)	0x2
Read the number of errors for syndrome polynomial 2: 1 error detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=2)	0x1
Read the number of errors for syndrome polynomial 3: 0 errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=3)	0x0
Read the error-location bit addresses for syndrome polynomial 0 of the 4 first registers:	ELM_ERROR_LOCATION_0_i (i=0)	0x1FE
	ELM_ERROR_LOCATION_1_i (i=0)	0x617
	ELM_ERROR_LOCATION_2_i (i=0)	0x650
	ELM_ERROR_LOCATION_3_i (i=0)	0xA83
Read the error-location bit addresses for syndrome polynomial 1 of the 2 first registers:	ELM_ERROR_LOCATION_0_i (i=1)	0x4
	ELM_ERROR_LOCATION_1_i (i=1)	0x1036
Read the errors location bit addresses for syndrome polynomial 2 of the first registers:	ELM_ERROR_LOCATION_0_i (i=1)	0x3E8
Clear the ELM_IRQSTATUS register.	ELM_IRQSTATUS	0x1FF

### 15.5.5 ELM Register Manual

#### 15.5.5.1 ELM Instance Summary

**Table 15-356. ELM Instance Summary**

Module Name	Base Address	Size
ELM	0x4807 8000	4KB

#### 15.5.5.2 ELM Registers

##### 15.5.5.2.1 ELM Register Summary

**Table 15-357. ELM Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	ELM L3 Physical Address
ELM_REVISION	R	32	0x0000 0000	0x4807 8000

**Table 15-357. ELM Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	ELM L3 Physical Address
ELM_SYSCONFIG	RW	32	0x0000 0010	0x4807 8010
ELM_SYSSTATUS	R	32	0x0000 0014	0x4807 8014
ELM_IRQSTATUS	RW	32	0x0000 0018	0x4807 8018
ELM_IRQENABLE	RW	32	0x0000 001C	0x4807 801C
ELM_LOCATION_CONFIG	RW	32	0x0000 0020	0x4807 8020
ELM_PAGE_CTRL	RW	32	0x0000 0080	0x4807 8080
ELM_SYNDROME_FRAGMENT_0_i <sup>(1)</sup>	RW	32	0x0000 0400 + (0x40 * i)	0x4807 8400 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_1_i <sup>(1)</sup>	RW	32	0x0000 0404 + (0x40 * i)	0x4807 8404 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_2_i <sup>(1)</sup>	RW	32	0x0000 0408 + (0x40 * i)	0x4807 8408 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_3_i <sup>(1)</sup>	RW	32	0x0000 040C + (0x40 * i)	0x4807 840C + (0x40 * i)
ELM_SYNDROME_FRAGMENT_4_i <sup>(1)</sup>	RW	32	0x0000 0410 + (0x40 * i)	0x4807 8410 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_5_i <sup>(1)</sup>	RW	32	0x0000 0414 + (0x40 * i)	0x4807 8414 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_6_i <sup>(1)</sup>	RW	32	0x0000 0418 + (0x40 * i)	0x4807 8418 + (0x40 * i)
ELM_LOCATION_STATUS_i <sup>(2)</sup>	R	32	0x0000 0800 + (0x100 * i)	0x4807 8800 + (0x100 * i)
ELM_ERROR_LOCATION_0_i <sup>(2)</sup>	R	32	0x0000 0880 + (0x100 * i)	0x4807 8880 + (0x100 * i)
ELM_ERROR_LOCATION_1_i <sup>(2)</sup>	R	32	0x0000 0884 + (0x100 * i)	0x4807 8884 + (0x100 * i)
ELM_ERROR_LOCATION_2_i <sup>(2)</sup>	R	32	0x0000 0888 + (0x100 * i)	0x4807 8888 + (0x100 * i)
ELM_ERROR_LOCATION_3_i <sup>(2)</sup>	R	32	0x0000 088C + (0x100 * i)	0x4807 888C + (0x100 * i)
ELM_ERROR_LOCATION_4_i <sup>(2)</sup>	R	32	0x0000 0890 + (0x100 * i)	0x4807 8890 + (0x100 * i)
ELM_ERROR_LOCATION_5_i <sup>(2)</sup>	R	32	0x0000 0894 + (0x100 * i)	0x4807 8894 + (0x100 * i)
ELM_ERROR_LOCATION_6_i <sup>(2)</sup>	R	32	0x0000 0898 + (0x100 * i)	0x4807 8898 + (0x100 * i)
ELM_ERROR_LOCATION_7_i <sup>(2)</sup>	R	32	0x0000 089C + (0x100 * i)	0x4807 889C + (0x100 * i)
ELM_ERROR_LOCATION_8_i <sup>(2)</sup>	R	32	0x0000 08A0 + (0x100 * i)	0x4807 88A0 + (0x100 * i)
ELM_ERROR_LOCATION_9_i <sup>(2)</sup>	R	32	0x0000 08A4 + (0x100 * i)	0x4807 88A4 + (0x100 * i)
ELM_ERROR_LOCATION_10_i <sup>(2)</sup>	R	32	0x0000 08A8 + (0x100 * i)	0x4807 88A8 + (0x100 * i)
ELM_ERROR_LOCATION_11_i <sup>(2)</sup>	R	32	0x0000 08AC + (0x100 * i)	0x4807 88AC + (0x100 * i)
ELM_ERROR_LOCATION_12_i <sup>(2)</sup>	R	32	0x0000 08B0 + (0x100 * i)	0x4807 88B0 + (0x100 * i)
ELM_ERROR_LOCATION_13_i <sup>(2)</sup>	R	32	0x0000 08B4 + (0x100 * i)	0x4807 88B4 + (0x100 * i)
ELM_ERROR_LOCATION_14_i <sup>(2)</sup>	R	32	0x0000 08B8 + (0x100 * i)	0x4807 88B8 + (0x100 * i)
ELM_ERROR_LOCATION_15_i <sup>(2)</sup>	R	32	0x0000 08BC + (0x100 * i)	0x4807 88BC + (0x100 * i)

<sup>(1)</sup> i = 0 to 7 for ELM<sup>(2)</sup> i = 0 to 7 for ELM**15.5.5.2.2 ELM Register Description****Table 15-358. ELM\_REVISION**

<b>Address Offset</b>	0x0000 0000																																																																	
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b> ELM																																																																
<b>Description</b>	This register contains the IP revision code. (A write or reset of to this register has no effect.)																																																																	
<b>Type</b>	R																																																																	
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">REVISION</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	REVISION																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
REVISION																																																																		

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

**Table 15-359. Register Call Summary for Register ELM\_REVISION**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-360. ELM\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b> ELM
<b>Description</b>	This register allows controlling various parameters of the OCP interface.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLOCKACTIVITYOCP	RESERVED		SIDLEMODE		RESERVED	SOFTRESET	AUTOGATING								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	CLOCKACTIVITYOCP	OCP Clock activity when module is in IDLE mode (during wake up mode period) 0x0: OCP Clock can be switch-off 0x1: OCP Clock is maintained during wake up period	RW	0
7:5	RESERVED	Reserved	R	0x0
4:3	SIDLEMODE	Slave interface power management (IDLE req/ack control) 0x0: FORCE Idle. IDLE request is acknowledged unconditionally and immediately (Default <i>Dumb</i> mode for safety) 0x1: NO idle. IDLE request is never acknowledged. 0x2: SMART Idle. The acknowledgment to an IDLE request is given based on the internal activity. 0x3: Reserved - do not use	RW	0x2
2	RESERVED	Reserved	R	0
1	SOFTRESET	Module software reset This bit is automatically reset by hardware (During reads, it always returns 0.) It has same effect as the OCP hardware reset. 0x0: Normal mode 0x1: Start soft reset sequence.	RW	0
0	AUTOGATING	Internal OCP clock gating strategy (no module visible impact other than saving power) 0x0: OCP clock is free-running 0x1: Automatic internal OCP clock gating strategy is applied based on the OCP interface activity.	RW	1

**Table 15-361. Register Call Summary for Register ELM\_SYSCONFIG**

## ELM Functional Description

- [ELM Software Reset: \[0\] \[1\]](#)
- [ELM Power Management: \[2\] \[3\] \[4\]](#)

## ELM Basic Programming Model

- [Processing Initialization: \[5\] \[6\]](#)
- [Use case: ELM used in continuous mode: \[7\] \[8\]](#)
- [Use case: ELM used in page mode: \[9\] \[10\]](#)

## ELM Register Manual

- [ELM Register Summary: \[11\]](#)

**Table 15-362. ELM\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	ELM
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .		
<b>Description</b>	Internal reset monitoring (OCP domain) Undefined since: From hardware perspective, the reset state is 0. From software user perspective, when the accessible module is 1.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESETDONE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Internal reset monitoring (OCP domain) Undefined since: From hardware perspective, the reset state is 0. From software user perspective, when the accessible module is 1.  Read 0x0: Reset is on-going Read 0x1: Reset is done (completed)	R	1

**Table 15-363. Register Call Summary for Register ELM\_SYSSTATUS**

## ELM Functional Description

- [ELM Software Reset: \[0\]](#)

## ELM Basic Programming Model

- [Processing Initialization: \[1\]](#)
- [Use case: ELM used in continuous mode: \[2\]](#)
- [Use case: ELM used in page mode: \[3\]](#)

## ELM Register Manual

- [ELM Register Summary: \[4\]](#)



**Table 15-364. ELM\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0018		
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Interrupt status. This register doubles as a status register for the error-location processes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PAGE_VALID	LOC_VALID_7	LOC_VALID_6	LOC_VALID_5	LOC_VALID_4	LOC_VALID_3	LOC_VALID_2	LOC_VALID_1	LOC_VALID_0							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	PAGE_VALID	Error-location status for a full page, based on the mask definition Read 0x0: Error locations invalid for all polynomials enabled in the ECC_INTERRUPT_MASK register Read 0x1: All error locations valid Write 0x0: No effect Write 0x1: Clear interrupt	RW	0
7	LOC_VALID_7	Error-location status for syndrome polynomial 7 Read 0x0: No syndrome processed or process in progress Read 0x1: Error-location process completed Write 0x0: No effect Write 0x1: Clear interrupt	RW W1toClr	0
6	LOC_VALID_6	Error-location status for syndrome polynomial 6	RW W1toClr	0
5	LOC_VALID_5	Error-location status for syndrome polynomial 5	RW W1toClr	0
4	LOC_VALID_4	Error-location status for syndrome polynomial 4	RW W1toClr	0
3	LOC_VALID_3	Error-location status for syndrome polynomial 3	RW W1toClr	0
2	LOC_VALID_2	Error-location status for syndrome polynomial 2	RW W1toClr	0
1	LOC_VALID_1	Error-location status for syndrome polynomial 1	RW W1toClr	0
0	LOC_VALID_0	Error-location status for syndrome polynomial 0	RW W1toClr	0

**Table 15-365. Register Call Summary for Register ELM\_IRQSTATUS**

ELM Functional Description

- [ELM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [Processing Initialization: \[9\]](#)
- [Processing Completion: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)

ELM Basic Programming Model

- [Read results: \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]](#)
- [Use case: ELM used in continuous mode: \[25\] \[26\]](#)
- [Use case: ELM used in page mode: \[27\] \[28\] \[29\]](#)

ELM Register Manual

- [ELM Register Summary: \[30\]](#)

**Table 15-366. ELM\_IRQENABLE**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	ELM
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .		
<b>Description</b>	Interrupt enable		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																								PAGE_MASK	LOCATION_MASK_7	LOCATION_MASK_6	LOCATION_MASK_5	LOCATION_MASK_4	LOCATION_MASK_3	LOCATION_MASK_2	LOCATION_MASK_1	LOCATION_MASK_0

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	PAGE_MASK	Page interrupt mask bit 0: Disable interrupt 1: Enable interrupt	RW	0
7	LOCATION_MASK_7	Error-location interrupt mask bit for syndrome polynomial 7	RW	0
6	LOCATION_MASK_6	Error-location interrupt mask bit for syndrome polynomial 6	RW	0
5	LOCATION_MASK_5	Error-location interrupt mask bit for syndrome polynomial 5	RW	0
4	LOCATION_MASK_4	Error-location interrupt mask bit for syndrome polynomial 4	RW	0
3	LOCATION_MASK_3	Error-location interrupt mask bit for syndrome polynomial 3	RW	0
2	LOCATION_MASK_2	Error-location interrupt mask bit for syndrome polynomial 2	RW	0
1	LOCATION_MASK_1	Error-location interrupt mask bit for syndrome polynomial 1	RW	0
0	LOCATION_MASK_0	Error-location interrupt mask bit for syndrome polynomial 0 0: Disable interrupt 1: Enable interrupt	RW	0

**Table 15-367. Register Call Summary for Register ELM\_IRQENABLE**

## ELM Functional Description

- [ELM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [Processing Initialization: \[9\] \[10\]](#)
- [Processing Completion: \[11\]](#)

## ELM Basic Programming Model

- [Processing Initialization: \[12\] \[13\] \[14\]](#)
- [Use case: ELM used in continuous mode: \[15\]](#)
- [Use case: ELM used in page mode: \[16\]](#)

## ELM Register Manual

- [ELM Register Summary: \[17\]](#)

**Table 15-368. ELM\_LOCATION\_CONFIG**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	ELM
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .		
<b>Description</b>	ECC algorithm parameters		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ECC_SIZE								RESERVED								ECC_BCH_LEVEL							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26:16	ECC_SIZE	Maximum size of the buffers for which the error-location engine is used, in number of nibbles (4-bits entities)	RW	0x000
15:2	RESERVED	Reserved	R	0x0000
1:0	ECC_BCH_LEVEL	Error correction level 0x0: 4 bits 0x1: 8 bits 0x2: 16 bits 0x3: Reserved	RW	0x0

**Table 15-369. Register Call Summary for Register ELM\_LOCATION\_CONFIG**

ELM Functional Description

- [Processing Initialization: \[0\] \[1\] \[2\]](#)

ELM Basic Programming Model

- [Processing Initialization: \[3\] \[4\]](#)
- [Use case: ELM used in continuous mode: \[5\] \[6\]](#)
- [Use case: ELM used in page mode: \[7\] \[8\]](#)

ELM Register Manual

- [ELM Register Summary: \[9\]](#)

**Table 15-370. ELM\_PAGE\_CTRL**

<b>Address Offset</b>	0x0000 0080	<b>Instance</b>	ELM
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .		
<b>Description</b>	Page definition		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SECTOR_7	SECTOR_6	SECTOR_5	SECTOR_4	SECTOR_3	SECTOR_2	SECTOR_1	SECTOR_0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7	SECTOR_7	Set to 1 if syndrome polynomial 7 is part of the page in page mode. Must be 0 in continuous mode.	RW	0

Bits	Field Name	Description	Type	Reset
6	SECTOR_6	Set to 1 if syndrome polynomial 6 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
5	SECTOR_5	Set to 1 if syndrome polynomial 5 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
4	SECTOR_4	Set to 1 if syndrome polynomial 4 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
3	SECTOR_3	Set to 1 if syndrome polynomial 3 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
2	SECTOR_2	Set to 1 if syndrome polynomial 2 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
1	SECTOR_1	Set to 1 if syndrome polynomial 1 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
0	SECTOR_0	Set to 1 if syndrome polynomial 0 is part of the page in page mode. Must be 0 in continuous mode.	RW	0

**Table 15-371. Register Call Summary for Register ELM\_PAGE\_CTRL**

## ELM Functional Description

- [Processing Initialization: \[0\]](#)
- [Processing Completion: \[1\] \[2\] \[3\] \[4\]](#)

## ELM Basic Programming Model

- [Processing Initialization: \[5\] \[6\] \[7\]](#)
- [Use case: ELM used in continuous mode: \[8\]](#)
- [Use case: ELM used in page mode: \[9\] \[10\] \[11\] \[12\]](#)

## ELM Register Manual

- [ELM Register Summary: \[13\]](#)

**Table 15-372. ELM\_SYNDROME\_FRAGMENT\_0\_i**

<b>Address Offset</b>	0x0000 0400 + (0x40 * i)	<b>Index</b>	i = 0 to 7																																																																
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM																																																																
<b>Description</b>	Input syndrome polynomial bits 0 to 31.																																																																		
<b>Type</b>	RW																																																																		
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td style="background-color: yellow;">23</td><td style="background-color: yellow;">22</td><td style="background-color: yellow;">21</td><td style="background-color: yellow;">20</td><td style="background-color: yellow;">19</td><td style="background-color: yellow;">18</td><td style="background-color: yellow;">17</td><td style="background-color: yellow;">16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td style="background-color: yellow;">7</td><td style="background-color: yellow;">6</td><td style="background-color: yellow;">5</td><td style="background-color: yellow;">4</td><td style="background-color: yellow;">3</td><td style="background-color: yellow;">2</td><td style="background-color: yellow;">1</td><td>0</td> </tr> <tr> <td colspan="32">SYNDROME_0</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	SYNDROME_0																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
SYNDROME_0																																																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																															
31:0	SYNDROME_0	Syndrom bits 0 to 31	RW	0x0000 0000																																																															

**Table 15-373. Register Call Summary for Register ELM\_SYNDROME\_FRAGMENT\_0\_i**

## ELM Functional Description

- [Processing Initialization: \[0\]](#)

## ELM Basic Programming Model

- [Processing Initialization: \[1\]](#)
- [Use case: ELM used in continuous mode: \[2\]](#)
- [Use case: ELM used in page mode: \[3\] \[4\] \[5\] \[6\]](#)

## ELM Register Manual

- [ELM Register Summary: \[7\]](#)

**Table 15-374. ELM\_SYNDROME\_FRAGMENT\_1\_i**

<b>Address Offset</b>	0x0000 0404 + (0x40 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Input syndrome polynomial bits 32 to 63.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_1																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_1	Syndrome bits 32 to 63	RW	0x0000 0000

**Table 15-375. Register Call Summary for Register ELM\_SYNDROME\_FRAGMENT\_1\_i**

ELM Basic Programming Model

- [Processing Initialization: \[0\]](#)
- [Use case: ELM used in continuous mode: \[1\]](#)
- [Use case: ELM used in page mode: \[2\] \[3\] \[4\] \[5\]](#)

ELM Register Manual

- [ELM Register Summary: \[6\]](#)

**Table 15-376. ELM\_SYNDROME\_FRAGMENT\_2\_i**

<b>Address Offset</b>	0x0000 0408 + (0x40 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Input syndrome polynomial bits 64 to 95.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_2																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_2	Syndrome bits 64 to 95	RW	0x0000 0000

**Table 15-377. Register Call Summary for Register ELM\_SYNDROME\_FRAGMENT\_2\_i**

ELM Basic Programming Model

- [Use case: ELM used in continuous mode: \[0\]](#)
- [Use case: ELM used in page mode: \[1\] \[2\] \[3\] \[4\]](#)

ELM Register Manual

- [ELM Register Summary: \[5\]](#)

**Table 15-378. ELM\_SYNDROME\_FRAGMENT\_3\_i**

<b>Address Offset</b>	0x0000 040C + (0x40 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Input syndrome polynomial bits 96 to 127		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_3																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_3	Syndrome bits 96 to 127	RW	0x0000 0000

**Table 15-379. Register Call Summary for Register ELM\_SYNDROME\_FRAGMENT\_3\_i**

ELM Basic Programming Model

- [Use case: ELM used in continuous mode: \[0\]](#)
- [Use case: ELM used in page mode: \[1\] \[2\] \[3\] \[4\]](#)

ELM Register Manual

- [ELM Register Summary: \[5\]](#)

**Table 15-380. ELM\_SYNDROME\_FRAGMENT\_4\_i**

<b>Address Offset</b>	0x0000 0410 + (0x40 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Input syndrome polynomial bits 128 to 159.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_4																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_4	Syndrome bits 128 to 159	RW	0x0000 0000

**Table 15-381. Register Call Summary for Register ELM\_SYNDROME\_FRAGMENT\_4\_i**

ELM Basic Programming Model

- [Use case: ELM used in page mode: \[0\] \[1\] \[2\] \[3\]](#)

ELM Register Manual

- [ELM Register Summary: \[4\]](#)

**Table 15-382. ELM\_SYNDROME\_FRAGMENT\_5\_i**

<b>Address Offset</b>	0x0000 0414 + (0x40 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Input syndrome polynomial bits 160 to 191.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_5																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_5	Syndrome bits 160 to 191	RW	0x0000 0000

**Table 15-383. Register Call Summary for Register ELM\_SYNDROME\_FRAGMENT\_5\_i**

ELM Basic Programming Model

- [Processing Initialization: \[0\]](#)
- [Use case: ELM used in page mode: \[1\] \[2\] \[3\] \[4\]](#)

ELM Register Manual

- [ELM Register Summary: \[5\]](#)

**Table 15-384. ELM\_SYNDROME\_FRAGMENT\_6\_i**

<b>Address Offset</b>	0x0000 0418 + (0x40 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Input syndrome polynomial bits 192 to 207.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SYNDROME_VALID	SYNDROME_6														

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000
16	SYNDROME_VALID	Syndrome valid bit 0x0: This syndrome polynomial should not be processed. 0x1: This syndrome polynomial must be processed.	RW	0
15:0	SYNDROME_6	Syndrome bits 192 to 207	RW	0x0000

**Table 15-385. Register Call Summary for Register ELM\_SYNDROME\_FRAGMENT\_6\_i**

ELM Functional Description

- [Processing Initialization: \[0\] \[1\] \[2\] \[3\]](#)
- [Processing Completion: \[4\] \[5\]](#)

ELM Basic Programming Model

- [Processing Initialization: \[6\] \[7\]](#)
- [Read results: \[8\]](#)
- [Use case: ELM used in continuous mode: \[9\]](#)
- [Use case: ELM used in page mode: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)

ELM Register Manual

- [ELM Register Summary: \[18\]](#)

**Table 15-386. ELM\_LOCATION\_STATUS\_i**

<b>Address Offset</b>	0x0000 0800 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Exit status for the syndrome polynomial processing		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_CORRECTABLE	RESERVED	ECC_NB_ERRORS													

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	ECC_CORRECTABLE	Error-location process exit status 0x0: ECC error-location process failed. Number of errors and error locations are invalid. 0x1: All errors were successfully located. Number of errors and error locations are valid.	R	0
7:5	RESERVED	Reserved	R	0x0
4:0	ECC_NB_ERRORS	Number of errors detected and located	R	0x00

**Table 15-387. Register Call Summary for Register ELM\_LOCATION\_STATUS\_i**

## ELM Functional Description

- [Processing Initialization: \[0\]](#)
- [Processing Sequence: \[1\] \[2\] \[3\] \[4\]](#)
- [Processing Completion: \[5\] \[6\] \[7\]](#)

## ELM Basic Programming Model

- [Read results: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [Use case: ELM used in continuous mode: \[16\] \[17\]](#)
- [Use case: ELM used in page mode: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\]](#)

## ELM Register Manual

- [ELM Register Summary: \[26\]](#)

**Table 15-388. ELM\_ERROR\_LOCATION\_0\_i**

<b>Address Offset</b>	0x0000 0880 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-389. Register Call Summary for Register ELM\_ERROR\_LOCATION\_0\_i**

## ELM Functional Description

- [Processing Initialization: \[0\]](#)
- [Processing Sequence: \[1\] \[2\] \[3\]](#)
- [Processing Completion: \[4\]](#)

## ELM Basic Programming Model

- [Read results: \[5\] \[6\]](#)
- [Use case: ELM used in continuous mode: \[7\]](#)
- [Use case: ELM used in page mode: \[8\] \[9\] \[10\]](#)

## ELM Register Manual

- [ELM Register Summary: \[11\]](#)



**Table 15-390. ELM\_ERROR\_LOCATION\_1\_i**

<b>Address Offset</b>	0x0000 0884 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-391. Register Call Summary for Register ELM\_ERROR\_LOCATION\_1\_i**

ELM Functional Description

- [Processing Sequence: \[0\]](#)

ELM Basic Programming Model

- [Read results: \[1\] \[2\]](#)
- [Use case: ELM used in continuous mode: \[3\]](#)
- [Use case: ELM used in page mode: \[4\] \[5\]](#)

ELM Register Manual

- [ELM Register Summary: \[6\]](#)

**Table 15-392. ELM\_ERROR\_LOCATION\_2\_i**

<b>Address Offset</b>	0x0000 0888 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-393. Register Call Summary for Register ELM\_ERROR\_LOCATION\_2\_i**

ELM Basic Programming Model

- [Use case: ELM used in continuous mode: \[0\]](#)
- [Use case: ELM used in page mode: \[1\]](#)

ELM Register Manual

- [ELM Register Summary: \[2\]](#)

**Table 15-394. ELM\_ERROR\_LOCATION\_3\_i**

<b>Address Offset</b>	0x0000 088C + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-395. Register Call Summary for Register ELM\_ERROR\_LOCATION\_3\_i**

ELM Basic Programming Model

- [Use case: ELM used in continuous mode: \[0\]](#)
- [Use case: ELM used in page mode: \[1\]](#)

ELM Register Manual

- [ELM Register Summary: \[2\]](#)

**Table 15-396. ELM\_ERROR\_LOCATION\_4\_i**

<b>Address Offset</b>	0x0000 0890 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-397. Register Call Summary for Register ELM\_ERROR\_LOCATION\_4\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-398. ELM\_ERROR\_LOCATION\_5\_i**

<b>Address Offset</b>	0x0000 0894 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-399. Register Call Summary for Register ELM\_ERROR\_LOCATION\_5\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-400. ELM\_ERROR\_LOCATION\_6\_i**

<b>Address Offset</b>	0x0000 0898 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-401. Register Call Summary for Register ELM\_ERROR\_LOCATION\_6\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-402. ELM\_ERROR\_LOCATION\_7\_i**

<b>Address Offset</b>	0x0000 089C + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-403. Register Call Summary for Register ELM\_ERROR\_LOCATION\_7\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-404. ELM\_ERROR\_LOCATION\_8\_i**

<b>Address Offset</b>	0x0000 08A0 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-405. Register Call Summary for Register ELM\_ERROR\_LOCATION\_8\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-406. ELM\_ERROR\_LOCATION\_9\_i**

<b>Address Offset</b>	0x0000 08A4 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-407. Register Call Summary for Register ELM\_ERROR\_LOCATION\_9\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-408. ELM\_ERROR\_LOCATION\_10\_i**

<b>Address Offset</b>	0x0000 08A8 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-409. Register Call Summary for Register ELM\_ERROR\_LOCATION\_10\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-410. ELM\_ERROR\_LOCATION\_11\_i**

<b>Address Offset</b>	0x0000 08AC + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-411. Register Call Summary for Register ELM\_ERROR\_LOCATION\_11\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-412. ELM\_ERROR\_LOCATION\_12\_i**

<b>Address Offset</b>	0x0000 08B0 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-413. Register Call Summary for Register ELM\_ERROR\_LOCATION\_12\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-414. ELM\_ERROR\_LOCATION\_13\_i**

<b>Address Offset</b>	0x0000 08B4 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-415. Register Call Summary for Register ELM\_ERROR\_LOCATION\_13\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-416. ELM\_ERROR\_LOCATION\_14\_i**

<b>Address Offset</b>	0x0000 08B8 + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-417. Register Call Summary for Register ELM\_ERROR\_LOCATION\_14\_i**

ELM Register Manual

- [ELM Register Summary: \[0\]](#)

**Table 15-418. ELM\_ERROR\_LOCATION\_15\_i**

<b>Address Offset</b>	0x0000 08BC + (0x100 * i)	<b>Index</b>	i = 0 to 7
<b>Physical Address</b>	See <a href="#">Table 15-357</a> .	<b>Instance</b>	ELM
<b>Description</b>	Error-location register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

**Table 15-419. Register Call Summary for Register ELM\_ERROR\_LOCATION\_15\_i**

ELM Functional Description

- [Processing Initialization: \[0\]](#)
- [Processing Sequence: \[1\] \[2\]](#)
- [Processing Completion: \[3\]](#)

ELM Basic Programming Model

- [Read results: \[4\] \[5\]](#)

ELM Register Manual

- [ELM Register Summary: \[6\]](#)

PRELIMINARY

## 15.6 On-Chip Memory (OCM) Subsystem

### 15.6.1 OCM Subsystem Overview

There are three on-chip memories in the device.

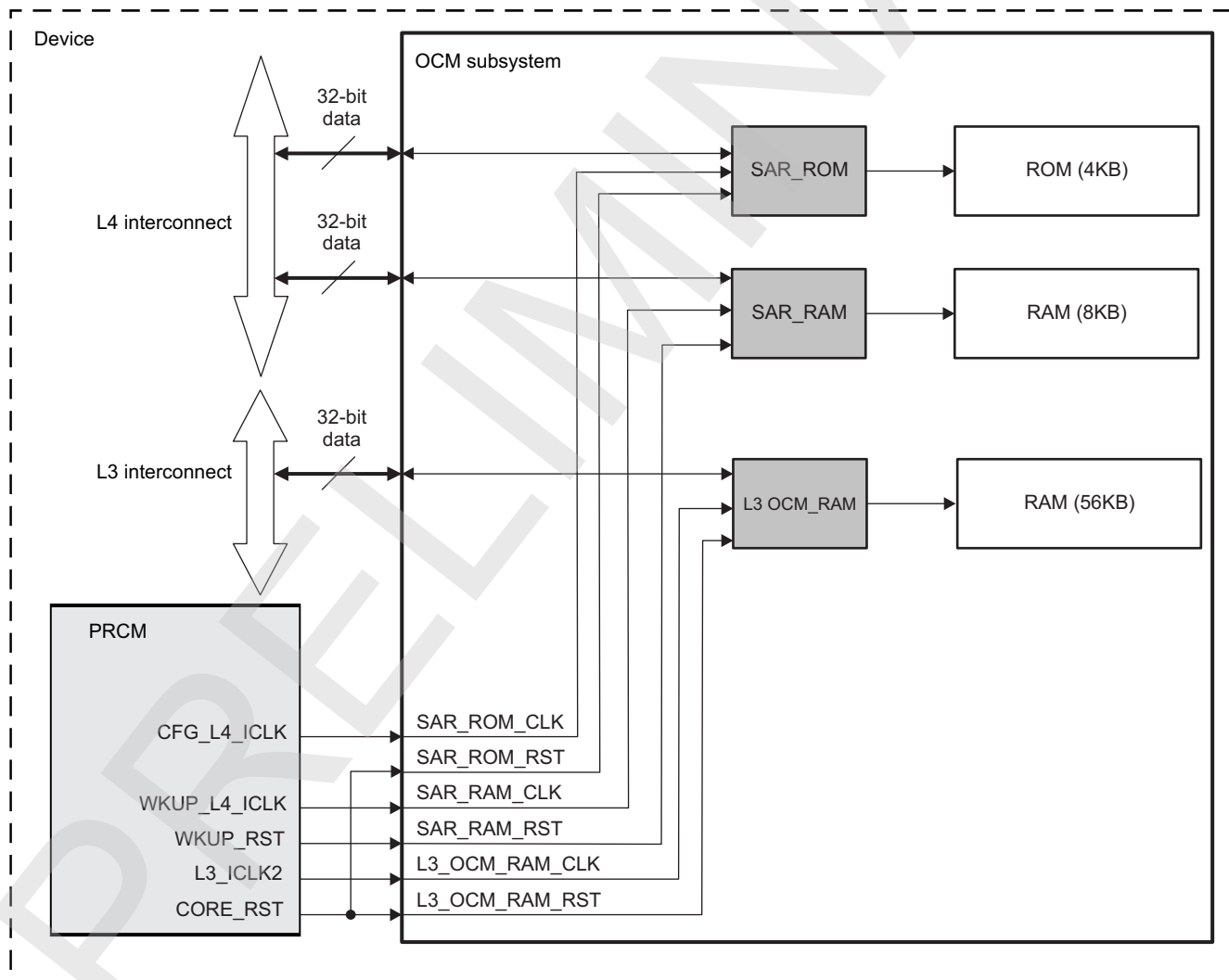
- A 4K-byte SAR read-only memory (ROM)
- A 8K-byte SAR random-access memory (RAM)
- A 56K-byte L3 static random-access memory (L3 OCM\_RAM)

There are other on-chip memories located inside other subsystems (Cortex-A9 MPU, DSP IVA-HD, ABE, Cortex-M3 MPU) that are not described in the following sections. Refer to each subsystem chapter for more details.

### 15.6.2 OCM Subsystem Integration

Figure 15-101 shows the integration of the OCM subsystem to the device.

**Figure 15-101. OCM Subsystem Integration to the Device**



Ocm-004

Table 15-420 through Table 15-421 summarize the integration of the module in the device.



**Table 15-420. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
L3 OCM_RAM	PD_CORE	No	NA
SAR_RAM	PD_WKUP	No	NA
SAR_ROM	PD_CORE	No	NA

**Table 15-421. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
L3 OCM_RAM	L3 OCM_RAM_CLK	L3_ICLK2	PRCM	L3 OCM_RAM Clock For information about PRCM clock gating and management, see the <i>CORE Power Domain Clocks Controls</i> section in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
SAR_RAM	SAR_RAM_CLK	WKUP_L4_ICLK	PRCM	SAR RAM clock For information about PRCM clock gating and management, see the <i>Wake-Up Power Domain Clocks Controls</i> section in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
SAR_ROM	SAR_ROM_CLK	CFG_L4_ICLK	PRCM	SAR ROM clock For information about PRCM clock gating and management, see the <i>CORE Power Domain Clocks Controls</i> section in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
L3 OCM_RAM	L3 OCM_RAM_RST	CORE_RST	PRCM	L3 OCM_RAM reset signal For information about PRCM reset sources and distribution, see the <i>CORE Power Domain Reset Description</i> section in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
SAR_RAM	SAR_RAM_RST	WKUP_RST	PRCM	SAR RAM reset signal For information about PRCM reset sources and distribution, see the <i>WAKE-UP Power Domain Reset Description</i> section in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
SAR_ROM	SAR_ROM_RST	CORE_RST	PRCM	SAR ROM reset signal For information about PRCM reset sources and distribution, see the <i>CORE Power Domain Reset Description</i> section in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .

### 15.6.3 OCM Subsystem Functional Description

#### 15.6.3.1 SAR ROM

This on-chip ROM contains 4K bytes of memory and a linked list of descriptors used by the system DMA (sDMA) during the restore context operation (when device transitions from OFF to ON modes).

The device-embedded SAR ROM has the following characteristics:

- 4K-bytes ROM
- 32-bit access per cycle
- Support for single and burst access transactions

The memory space of the embedded ROM starts at 0x4A05 E000 and ends at 0x4A05 FFFF (8K bytes are reserved, but only 4K bytes are used).

### 15.6.3.2 SAR RAM

The on-chip SAR RAM contains 8K bytes and is mapped as 4 blocks with irregular region sizes. This memory content is preserved when the device goes into OFF mode (as long as the wake-up voltage domain remains supplied). It is used as context-saving memory to be written by the software, so that the sDMA restores its saved content when the device transitions from OFF to ON mode. Locations not used by the sDMA can be used as a scratch-pad memory by the software. It is mapped on the wake-up voltage domain.

The device-embedded SAR RAM has the following characteristics:

- Support for single-access transactions
  - Operates at full L4-PER interconnect clock frequency
  - 32-bit access per cycle

The SAR RAM space is at the following location:

- SAR space 1 (4KB): Starts at 0x4A32 6000 and ends at 0x4A32 6FFF
- SAR space 2 (1KB): Starts at 0x4A32 7000 and ends at 0x4A32 73FF
- SAR space 3 (2KB): Starts at 0x4A32 8000 and ends at 0x4A32 87FF
- SAR space 4 (1KB): Starts at 0x4A32 9000 and ends at 0x4A32 93FF

### 15.6.3.3 L3 OCM\_RAM

The on-chip L3 OCM\_RAM contains 56K bytes of RAM, partitioning is defined by the L3 firewall logic.

The device-embedded L3 OCM\_RAM has the following characteristics:

- Support for single and burst access transactions:
  - Operates at full L3 interconnect clock frequency
  - Fully pipelined, one 32-bit access per cycle
- Restricted access support based on:
  - A region-based partitioning (refer to L3 firewall description)
  - The module owner of the access, with respect to its read and write permissions to that region
  - The transaction attributes of the access, with respect to the region permission properties. See [Section 13.2, L3 Interconnect](#) for details.

The memory space of the embedded RAM starts at 0x4030 0000 and ends at 0x4030 DFFF.

This chapter describes the system direct memory access (sDMA) module.

Topic	Page
16.1 sDMA Module Overview .....	3220
16.2 sDMA Controller Environment .....	3222
16.3 sDMA Module Integration .....	3224
16.4 sDMA Functional Description .....	3229
16.5 sDMA Basic Programming Model .....	3262
16.6 sDMA Register Manual .....	3269

## 16.1 sDMA Module Overview

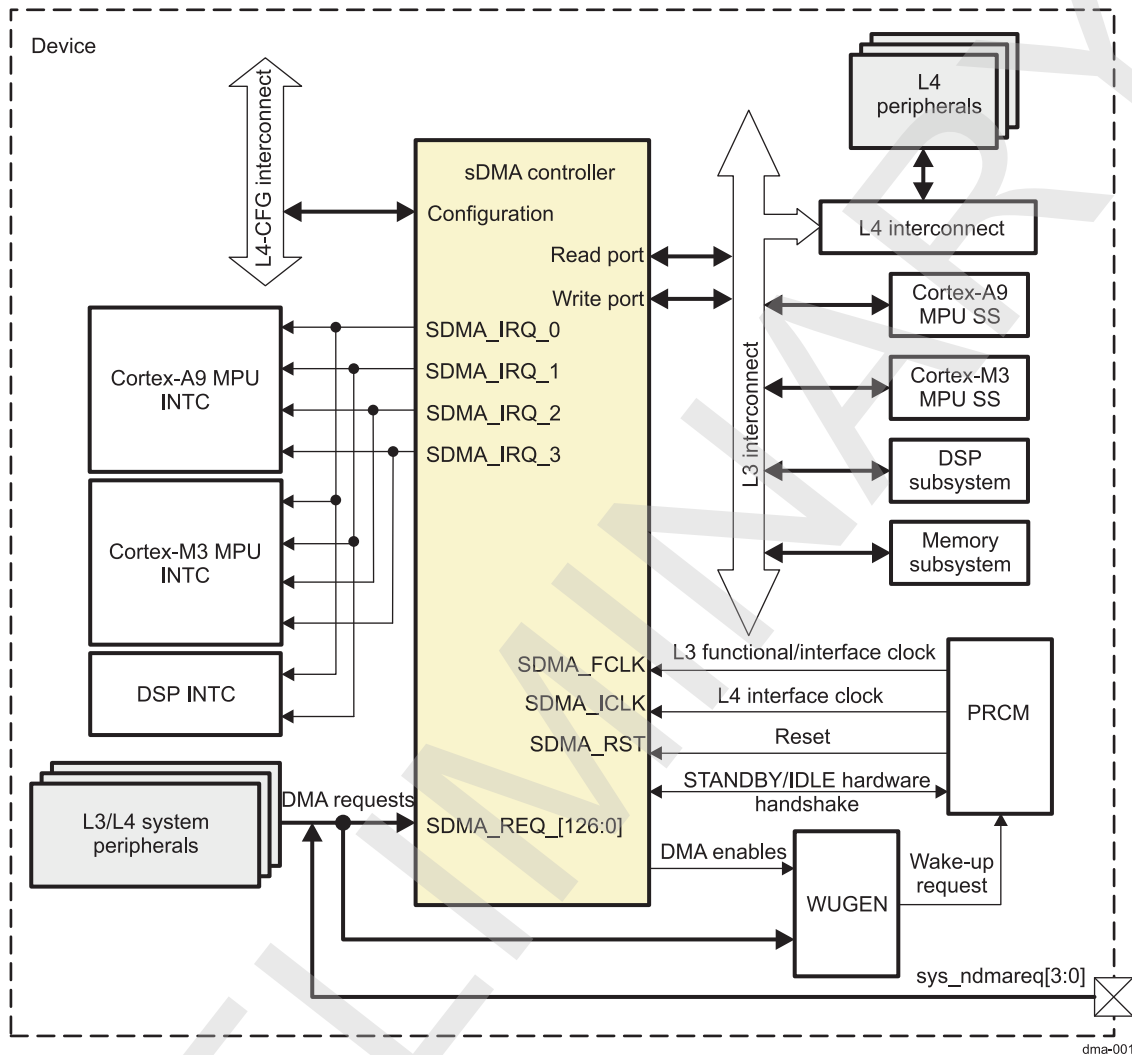
The system direct memory access (sDMA), also called DMA4, performs high-performance data transfers between memories and peripheral devices without microprocessor unit (MPU) or digital signal processor (DSP) support during transfer. A DMA transfer is programmed through a logical DMA channel, which allows the transfer to be optimally tailored to the requirements of the application.

The DMA controller includes the following main features:

- Data transfer support in either direction between:
  - Memory and memory
  - Memory and peripheral device
- 32 logical DMA channels supporting:
  - Multiple concurrent transfers
  - Independent transfer profile for each channel
  - 8-bit, 16-bit, or 32-bit data element transfer size
  - Software-triggered or hardware-synchronized transfers
  - Flexible source and destination address generation
  - Burst read and write
  - Chained multiple-channel transfers
  - Endianism conversion
  - Draining
  - Linked-list support for descriptor types 1, 2, and 3
- First-come, first-serve DMA scheduling with fixed priority
- Up to 127 DMA requests
- Constant fill
- Transparent copy
- Four programmable interrupt request output lines
- FIFO depth: 256 x 64-bit
- Data buffering
- FIFO budget allocation
- Power-management support
- Auto-idle power-saving support
- Implementation of retention flip-flops (RFFs) to support dynamic power switching (DPS) between system power modes without MPU involvement. For more information about DPS, see [Section 3.1.2.3, Dynamic Power Switching \(DPS\)](#).

Figure 16-1 shows an overview of the sDMA module.

Figure 16-1. sDMA Overview



The sDMA module has three ports: one read, one write, and one configuration port and provides multiple logical channel support. A dynamically allocated FIFO queue memory pool provides buffering between the read and write ports, which are multithreaded (two threads for the write port and four threads for the read port); this means that each transaction is flagged by a thread ID (0, 1, 2, or 3) in the request direction and in the response direction. This allows the read port to have four outstanding requests at a time. The write port has two threads budget available.

The MPU (or DSP) configures the sDMA through the L4-CFG interconnect.

## 16.2 sDMA Controller Environment

The sDMA controller supports external DMA requests through the sys\_ndmareq[3:0] pins (see [Table 16-1](#)). A logical channel can be configured to respond to an external synchronization request.

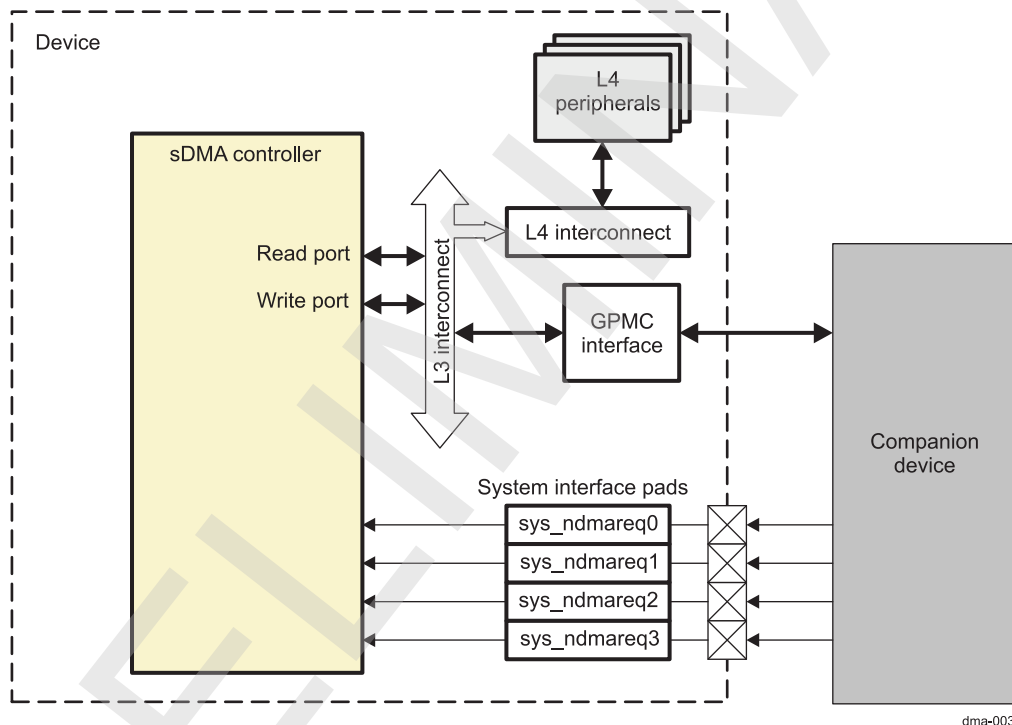
**Table 16-1. External sDMA Request Signals**

Signal Name	I/O <sup>(1)</sup>	Description
sys_ndmareq0	I	External DMA request 0 (system expansion)
sys_ndmareq1	I	External DMA request 1 (system expansion)
sys_ndmareq2	I	External DMA request 2 (system expansion)
sys_ndmareq3	I	External DMA request 3 (system expansion)

<sup>(1)</sup> I = Input, O = Output

[Figure 16-2](#) shows an example of how to use the external hardware DMA request pins in the sDMA environment.

**Figure 16-2. Example of External DMA Requests Use**



dma-003

An external device can use the external DMA request pins to start a logical channel transfer over the general-purpose memory controller (GPMC) interface. The transfer can be a memory-to-memory transfer in which the source memory is in the external device.

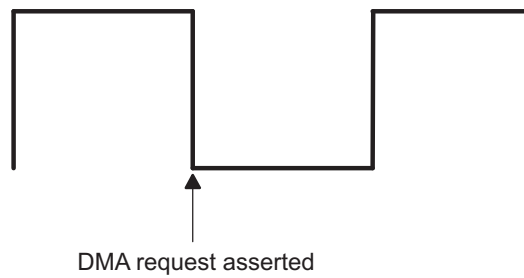
By default, the external DMA request signals are not available on external pins after a cold reset. For more information about multiplexing out the four signal lines to pins, see [Chapter 18, Control Module](#).

The external hardware DMA request line schemes can be either edge-sensitive or transition-sensitive. The default scheme for external requests is transition-sensitive. The sensitivity selection can be configured in the control module. For more information, see [Chapter 18, Control Module](#).

All internal peripherals of the device use the transition-sensitive scheme.

[Figure 16-3](#) shows the DMA request captured on a falling edge in the edge-sensitive scheme.

**Figure 16-3. Edge-Sensitive DMA Request Scheme**

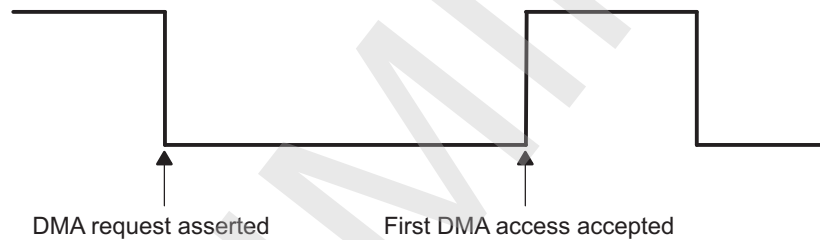


dma-013

For a transition-sensitive DMA request (see [Figure 16-4](#)), the line must be maintained low (asserted) until the first DMA access is complete, after which the line must be maintained high (deasserted) for greater than one clock cycle (DMA\_L3\_ICLK):

- When the deassertion time is less than one clock cycle, the sDMA might not detect the deassertion.
- When the channel is enabled one cycle after a DMA request is disabled, the channel detects the DMA request and starts the corresponding transfer.
- When the channel is enabled two cycles after the DMA request is disabled, the channel does not detect the DMA request.

**Figure 16-4. Transition-Sensitive DMA Request Scheme**



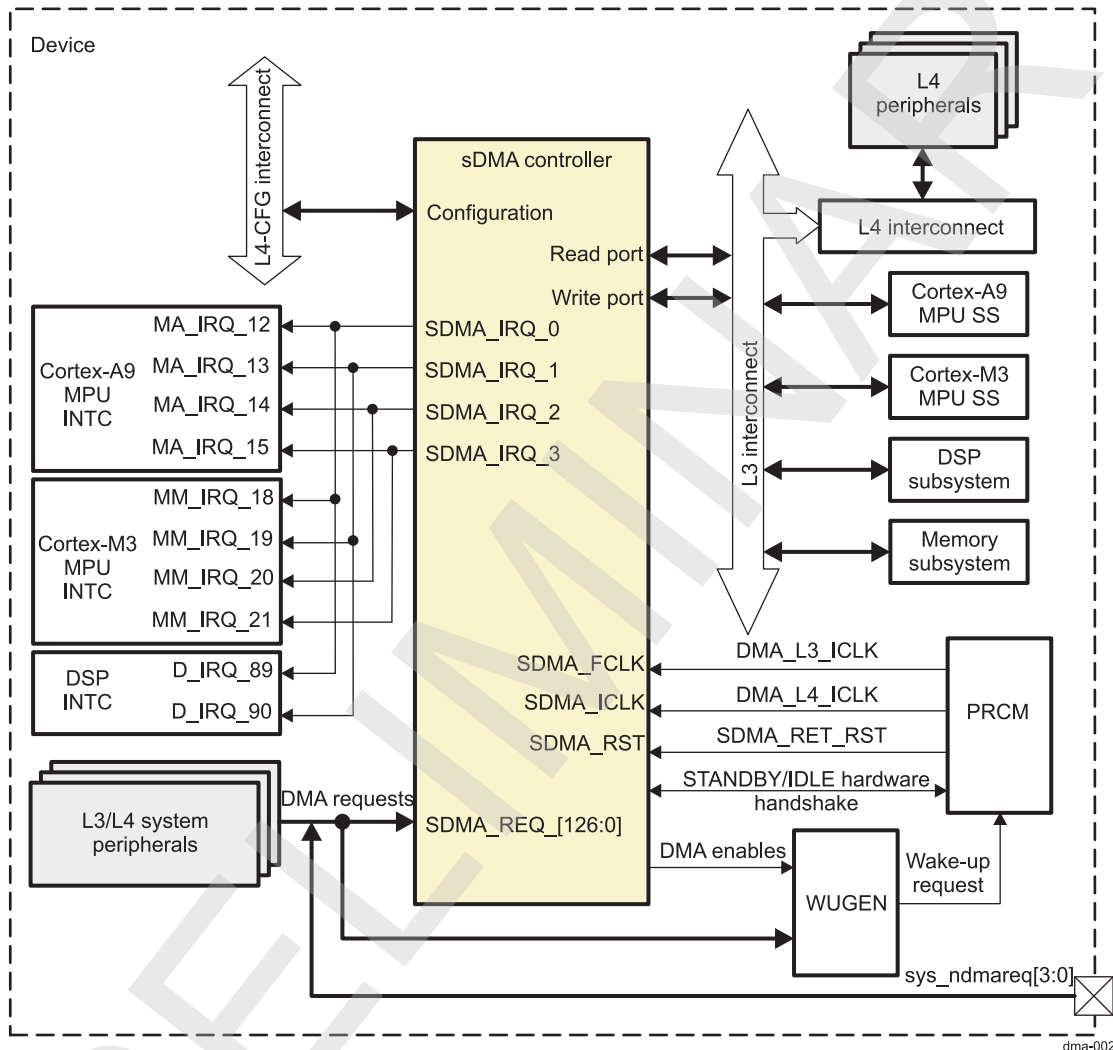
dma-012

## 16.3 sDMA Module Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 16-5 shows the sDMA controller integration.

**Figure 16-5. sDMA Controller Integration**



**NOTE:** For more information about the wake-up generator (WUGEN), the STANDBY/IDLE hardware handshake, and the wake-up request, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

Table 16-2 through Table 16-4 summarize the integration of the module in the device.

**Table 16-2. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
sDMA	PD_CORE	L3 and L4-CFG



**Table 16-3. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
sDMA	SDMA_FCLK	DMA_L3_ICLK	PRCM	It is both a functional clock for all internal logic and for the two master read and write ports. For information about PRCM clock gating and management, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
sDMA	SDMA_ICLK	DMA_L4_ICLK	PRCM	Interface clock. It supports the configuration port. For information about PRCM clock gating and management, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
sDMA	SDMA_RST	SDMA_RET_RST	PRCM	Hardware reset. It initializes all internal logic of the sDMA module, all global registers and some of the per-channel registers, implemented in flip-flops. However, all remaining per-channel registers are memory-based, and, therefore, are not reset (have undefined values). Thus, when programming a channel for the first time, all bits that have undefined reset values must be configured before enabling the channel. For information about PRCM reset sources and distribution, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .

**Table 16-4. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
sDMA	SDMA_IRQ_0	MA_IRQ_12	Cortex-A9 MPU INTC	sDMA interrupt request 0. For information about the Cortex-A9 MPU INTC, see <a href="#">Section 17.3.2</a> .
	SDMA_IRQ_1	MA_IRQ_13	Cortex-A9 MPU INTC	sDMA interrupt request 1
	SDMA_IRQ_2	MA_IRQ_14	Cortex-A9 MPU INTC	sDMA interrupt request 2
	SDMA_IRQ_3	MA_IRQ_15	Cortex-A9 MPU INTC	sDMA interrupt request 3
sDMA	SDMA_IRQ_0	MM_IRQ_18	Cortex-M3 MPU INTC	sDMA interrupt request 0. For information about the Cortex-M3 MPU INTC, see <a href="#">Section 17.3.3</a> .
	SDMA_IRQ_1	MM_IRQ_19	Cortex-M3 MPU INTC	sDMA interrupt request 1
	SDMA_IRQ_2	MM_IRQ_20	Cortex-M3 MPU INTC	sDMA interrupt request 2
	SDMA_IRQ_3	MM_IRQ_21	Cortex-M3 MPU INTC	sDMA interrupt request 3
sDMA	SDMA_IRQ_0	D_IRQ_89	DSP INTC	sDMA interrupt request 0. For information about the DSP INTC, see .
	SDMA_IRQ_1	D_IRQ_90	DSP INTC	sDMA interrupt request 1

**NOTE:** For a description of the interrupt source, see [Section 16.4.2, sDMA Controller Interrupt Requests](#).

### 16.3.1 DMA Requests to the sDMA Controller

Table 16-5 lists the sDMA controller request mapping.

**Table 16-5. sDMA Controller Request Mapping**

DMA Request Line	Source	Description
S_DMA_0	Reserved	Reserved
S_DMA_1	sys_dma_req0	External DMA request 0 (system expansion)
S_DMA_2	sys_dma_req1	External DMA request 1 (system expansion)
S_DMA_3	GPMC_DMA	GPMC data transmit request from prefetch engine
S_DMA_4	Reserved	Reserved
S_DMA_5	DSS_DISPC_DMA	Frame update request
S_DMA_6	sys_dma_req2	External DMA request 2 (system expansion)
S_DMA_7	MCASP1_AXEVT	McASP1 transmit request
S_DMA_8	ISS_DMA0	Imaging subsystem request 0
S_DMA_9	ISS_DMA1	Imaging subsystem request 1
S_DMA_10	Reserved	Reserved
S_DMA_11	ISS_DMA2	Imaging subsystem request 2
S_DMA_12	ISS_DMA3	Imaging subsystem request 3
S_DMA_13	DSS_RFBI_DMA	Display subsystem RFBI request
S_DMA_14	SPI3_DMA_TX0	McSPI3 transmit request channel 0
S_DMA_15	SPI3_DMA_RX0	McSPI3 receive request channel 0
S_DMA_16	MCBSP2_DMA_TX	McBSP2 transmit request
S_DMA_17	MCBSP2_DMA_RX	McBSP2 receive request
S_DMA_18	MCBSP3_DMA_TX	McBSP3 transmit request
S_DMA_19	MCBSP3_DMA_RX	McBSP3 receive request
S_DMA_20	C2C_SSCM_GPO0	C2C GPO line 0
S_DMA_21	C2C_SSCM_GPO1	C2C GPO line 1
S_DMA_22	SPI3_DMA_TX1	McSPI3 transmit request channel 1
S_DMA_23	SPI3_DMA_RX1	McSPI3 receive request channel 1
S_DMA_24	I2C3_DMA_TX	I2C3 transmit request
S_DMA_25	I2C3_DMA_RX	I2C3 receive request
S_DMA_26	I2C1_DMA_TX	I2C1 transmit request
S_DMA_27	I2C1_DMA_RX	I2C1 receive request
S_DMA_28	I2C2_DMA_TX	I2C2 transmit request
S_DMA_29	I2C2_DMA_RX	I2C2 receive request
S_DMA_30	MCBSP4_DMA_TX	McBSP4 transmit request
S_DMA_31	MCBSP4_DMA_RX	McBSP4 receive request
S_DMA_32	MCBSP1_DMA_TX	McBSP1 transmit request
S_DMA_33	MCBSP1_DMA_RX	McBSP1 receive request
S_DMA_34	SPI1_DMA_TX0	McSPI1 transmit request channel 0
S_DMA_35	SPI1_DMA_RX0	McSPI1 receive request channel 0
S_DMA_36	SPI1_DMA_TX1	McSPI1 transmit request channel 1
S_DMA_37	SPI1_DMA_RX1	McSPI1 receive request channel 1
S_DMA_38	SPI1_DMA_TX2	McSPI1 transmit request channel 2
S_DMA_39	SPI1_DMA_RX2	McSPI1 receive request channel 2

**Table 16-5. sDMA Controller Request Mapping (continued)**

DMA Request Line	Source	Description
S_DMA_40	SPI1_DMA_TX3	McSPI1 transmit request channel 3
S_DMA_41	SPI1_DMA_RX3	McSPI1 receive request channel 3
S_DMA_42	SPI2_DMA_TX0	McSPI2 transmit request channel 0
S_DMA_43	SPI2_DMA_RX0	McSPI2 receive request channel 0
S_DMA_44	SPI2_DMA_TX1	McSPI2 transmit request channel 1
S_DMA_45	SPI2_DMA_RX1	McSPI2 receive request channel 1
S_DMA_46	MMC2_DMA_TX	MMC2 transmit request
S_DMA_47	MMC2_DMA_RX	MMC2 receive request
S_DMA_48	UART1_DMA_TX	UART1 transmit request
S_DMA_49	UART1_DMA_RX	UART1 receive request
S_DMA_50	UART2_DMA_TX	UART2 transmit request
S_DMA_51	UART2_DMA_RX	UART2 receive request
S_DMA_52	UART3_DMA_TX	UART3 transmit request
S_DMA_53	UART3_DMA_RX	UART3 receive request
S_DMA_54	UART4_DMA_TX	UART4 transmit request
S_DMA_55	UART4_DMA_RX	UART4 receive request
S_DMA_56	MMC4_DMA_TX	MMC4 transmit request
S_DMA_57	MMC4_DMA_RX	MMC4 receive request
S_DMA_58	MMC5_DMA_TX	MMC5 transmit request
S_DMA_59	MMC5_DMA_RX	MMC5 receive request
S_DMA_60	MMC1_DMA_TX	MMC1 transmit request
S_DMA_61	MMC1_DMA_RX	MMC1 receive request
S_DMA_62	Reserved	Reserved
S_DMA_63	sys_dma_req3	External DMA request 3 (system expansion)
S_DMA_64	MCPDM_DMA_UP	McPDM uplink request
S_DMA_65	MCPDM_DMA_DOWN	McPDM downlink request
S_DMA_66	DMIC_DMA	DMIC DMA request
S_DMA_67	C2C_SSCM_GPO2	C2C GPO line 2
S_DMA_68	C2C_SSCM_GPO3	C2C GPO line 3
S_DMA_69	SPI4_DMA_TX0	McSPI4 transmit request channel 0
S_DMA_70	SPI4_DMA_RX0	McSPI4 receive request channel 0
S_DMA_71	DSS_DSI1_DMA0	Display subsystem DSI1 request 0
S_DMA_72	DSS_DSI1_DMA1	Display subsystem DSI1 request 1
S_DMA_73	DSS_DSI1_DMA2	Display subsystem DSI1 request 2
S_DMA_74	DSS_DSI1_DMA3	Display subsystem DSI1 request 3
S_DMA_75	DSS_HDMI_DMA	Display subsystem HDMI audio request
S_DMA_76	MMC3_DMA_TX	MMC3 transmit request
S_DMA_77	MMC3_DMA_RX	MMC3 receive request
S_DMA_78	Reserved	Reserved
S_DMA_79	Reserved	Reserved
S_DMA_80	DSS_DSI2_DMA0	Display subsystem DSI2 request 0
S_DMA_81	DSS_DSI2_DMA1	Display subsystem DSI2 request 1
S_DMA_82	DSS_DSI2_DMA2	Display subsystem DSI2 request 2
S_DMA_83	DSS_DSI2_DMA3	Display subsystem DSI2 request 3
S_DMA_84	SLIMBUS1_DMA_TX0	SLIMBUS1 transmit request channel 0
S_DMA_85	SLIMBUS1_DMA_TX1	SLIMBUS1 transmit request channel 1
S_DMA_86	SLIMBUS1_DMA_TX2	SLIMBUS1 transmit request channel 2

**Table 16-5. sDMA Controller Request Mapping (continued)**

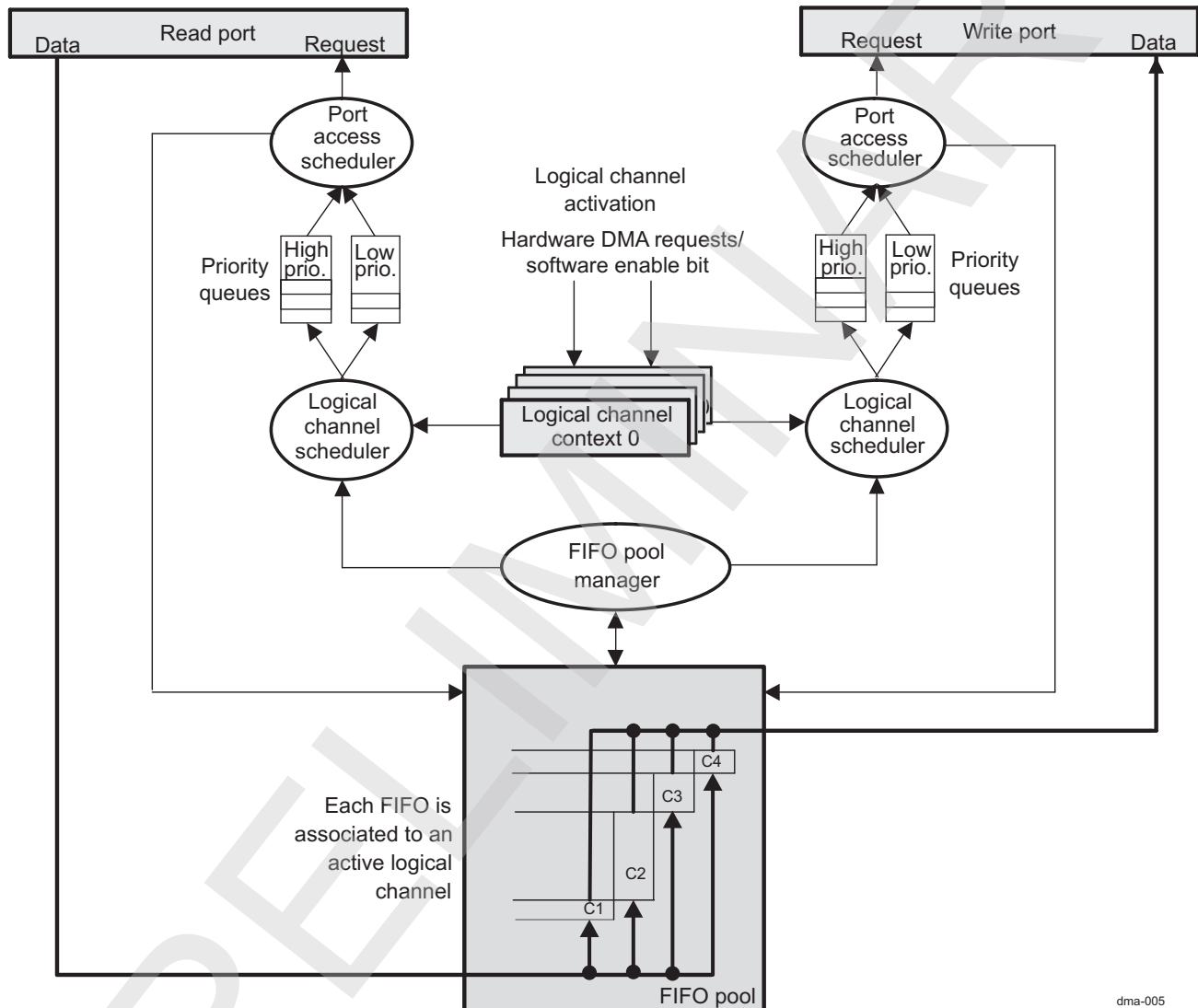
DMA Request Line	Source	Description
S_DMA_87	SLIMBUS1_DMA_TX3	SLIMBUS1 transmit request channel 3
S_DMA_88	SLIMBUS1_DMA_RX0	SLIMBUS1 receive request channel 0
S_DMA_89	SLIMBUS1_DMA_RX1	SLIMBUS1 receive request channel 1
S_DMA_90	SLIMBUS1_DMA_RX2	SLIMBUS1 receive request channel 2
S_DMA_91	SLIMBUS1_DMA_RX3	SLIMBUS1 receive request channel 3
S_DMA_92	SLIMBUS2_DMA_TX0	SLIMBUS2 transmit request channel 0
S_DMA_93	SLIMBUS2_DMA_TX1	SLIMBUS2 transmit request channel 1
S_DMA_94	SLIMBUS2_DMA_TX2	SLIMBUS2 transmit request channel 2
S_DMA_95	SLIMBUS2_DMA_TX3	SLIMBUS2 transmit request channel 3
S_DMA_96	SLIMBUS2_DMA_RX0	SLIMBUS2 receive request channel 0
S_DMA_97	SLIMBUS2_DMA_RX1	SLIMBUS2 receive request channel 1
S_DMA_98	SLIMBUS2_DMA_RX2	SLIMBUS2 receive request channel 2
S_DMA_99	SLIMBUS2_DMA_RX3	SLIMBUS2 receive request channel 3
S_DMA_100	ABE_DMA0	Audio back-end – request FIFO 0
S_DMA_101	ABE_DMA1	Audio back-end – request FIFO 1
S_DMA_102	ABE_DMA2	Audio back-end – request FIFO 2
S_DMA_103	ABE_DMA3	Audio back-end – request FIFO 3
S_DMA_104	ABE_DMA4	Audio back-end – request FIFO 4
S_DMA_105	ABE_DMA5	Audio back-end – request FIFO 5
S_DMA_106	ABE_DMA6	Audio back-end – request FIFO 6
S_DMA_107	ABE_DMA7	Audio back-end – request FIFO 7
S_DMA_108	Reserved	Reserved
S_DMA_109	Reserved	Reserved
S_DMA_110	Reserved	Reserved
S_DMA_111	Reserved	Reserved
S_DMA_112	Reserved	Reserved
S_DMA_113	Reserved	Reserved
S_DMA_114	Reserved	Reserved
S_DMA_115	Reserved	Reserved
S_DMA_116	Reserved	Reserved
S_DMA_117	Reserved	Reserved
S_DMA_118	Reserved	Reserved
S_DMA_119	Reserved	Reserved
S_DMA_120	Reserved	Reserved
S_DMA_121	Reserved	Reserved
S_DMA_122	Reserved	Reserved
S_DMA_123	I2C4_DMA_TX	I2C4 transmit request
S_DMA_124	I2C4_DMA_RX	I2C4 receive request
S_DMA_125	Reserved	Reserved
S_DMA_126	Reserved	Reserved

## 16.4 sDMA Functional Description

The sDMA module provides high-performance data transfers between memories and peripheral devices with low processor use. A DMA transfer is programmed through a logical DMA channel, which allows the transfer to be optimally tailored to the requirements of the application.

Figure 16-6 shows the sDMA controller top-level block diagram.

Figure 16-6. sDMA Controller Top-Level Block Diagram



dma-005

### 16.4.1 sDMA Controller Power Management

Table 16-6 describes power-management features available for the sDMA controller.

**NOTE:**

- For information about source clock gating and sleep/wake-up transitions, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).
- For a description of the EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**Table 16-6. Local Power Management Features**

Feature	Registers	Description
Clock auto gating	<a href="#">DMA4_OCP_SYSCONFIG</a> [0] AUTOIDLE bit	This bit allows local power optimization inside the module by gating the SDMA_ICLK clock upon the interface activity.
Slave idle modes	<a href="#">DMA4_OCP_SYSCONFIG</a> [4:3] SIDLEMODE bit field	Force-idle, no-idle and smart-idle modes are available.
Clock activity	<a href="#">DMA4_OCP_SYSCONFIG</a> [9:8] CLOCKACTIVITY bit field	For configuration details, see <a href="#">Table 16-7</a> .
Master standby modes	<a href="#">DMA4_OCP_SYSCONFIG</a> [13:12] MIDLEMODE bit field	Force-standby, no-standby and smart-standby modes are available.
Global Wake-up Enable	N/A	Feature not available
Wake-up Sources Enable	N/A	Feature not available

**Table 16-7. Clock Activity Settings**

SDMA_CLOCKACTIVITY Values	Clock State When Module is in IDLE State	
	SDMA_ICLK	SDMA_FCLK
00	Off	Off
10	Off	On
01	On	Off
11	On	On

**CAUTION**

Because the PRCM module cannot read CLOCKACTIVITY settings through hardware, software must ensure consistent programming between the SDMA\_CLOCKACTIVITY and sDMA clock PRCM control bits. For a description of the ClockActivity feature, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**16.4.2 sDMA Controller Interrupt Requests**

DMA4 has four interrupt lines ( $L_j$ , where  $j = 0, 1, 2, 3$ ). Each logical channel can request an interrupt over any line. The attachment of a channel interrupt event to one of these four external lines is programmable. Software determines whether it attaches a channel interrupt to a single IRQ line or to multiple IRQ lines.

There are two different registers per interrupt line:

- The [DMA4\\_IRQSTATUS\\_Lj](#) CH\_31\_0\_Lj field shows the status of the different sources of interrupt. If the [DMA4\\_IRQENABLE\\_Lj](#) bit is 1, channel  $i$  is the source of interrupt in line  $j$ . In contrast to the [DMA4\\_CSRi](#) registers, the [DMA4\\_IRQSTATUS\\_Lj](#) registers are updated regardless of the corresponding bits in the [DMA4\\_IRQENABLE\\_Lj](#) registers.
- The [DMA4\\_IRQENABLE\\_Lj](#) CH\_31\_0\_Lj\_EN field masks/unmasks the channel interrupt. If the [DMA4\\_IRQENABLE\\_Lj](#) bit is set to 0, channel interrupt  $i$  of the line  $j$  is masked.

Each logical channel can generate a number of different interrupt events when enabled (that is, set to 1) in the [DMA4\\_CICRi](#) register. Each status bit is updated in the [DMA4\\_CSRi](#) register only when the corresponding enable bit is enabled in the [DMA4\\_CICRi](#) register.

To determine an interrupt source when an interrupt rises on an interrupt line  $L_j$ :

- Identify the channel ( $LCH_i$ ) generating the interrupt.  
Read the [DMA4\\_IRQSTATUS\\_Lj.LCHi](#) ( $LCH_0$  to  $LCH_{31}$ ). If  $LCH_i = 1$ , channel  $i$  is the originator of the interrupt.
- Identify the interrupt event.  
Read the  $LCH_i$  [DMA4\\_CSRi](#). For example, if the drop event (the [DMA4\\_CSRi\[1\]](#) DROP bit) is 1, there will be a request collision.

The interrupt event status bit in the [DMA4\\_CSRi](#) register is immediately reset after it is written to 1. The interrupt status bit in the [DMA4\\_IRQSTATUS\\_Lj](#) register is cleared after it is written to 1.

### 16.4.2.1 Interrupt Generation

The sDMA module has four interrupt request output lines, SDMA\_IRQ\_0 to SDMA\_IRQ\_3. One or more logical channels can be programmed to generate an interrupt request on any of these lines when any one of the maskable DMA events listed in [Table 16-8](#) occurs.

**Table 16-8. Logical DMA Channel Events**

Event	Description
End of packet	A packet transfer completed.
End of block	A block transfer completed.
End of frame	A frame transfer completed.
End of super block	A super block transfer completed.
Half of frame	Half of the current frame transferred.
Start of last frame	The first element of the last frame transferred.
Transaction error	A transaction error is returned by the interconnect in either the read or write port.
Address error	An attempt was made to perform a DMA access to an address not aligned on an ES boundary. Condition to occur: if DMA4_CEN[23:0] CHANNEL_ELMNT_NBR = 0x000000 or DMA4_CFN[15:0] CHANNEL_FRAME_NBR = 0x0000 or DMA4_CSDP[1:0] DATA_TYPE = 0x3.
Supervisor transaction error	An error occurred, for example, when an unauthorized initiator (that is not a supervisor ) tries to use a supervisor transfer.
Drain end	Drain is completed ( <a href="#">DMA4_CCRi</a> [10] WR_ACTIVE becomes 0).
Drop error	A drop event interrupt is generated when a DMA request is being serviced while a second one is asserted and a third one arrives before the second DMA request is serviced.

The logical DMA channels that generate an interrupt on a particular IRQ output are specified through the [DMA4\\_IRQENABLE\\_Lj](#) register (where *j* is the IRQ number: 0, 1, 2, or 3). The events that generate an interrupt for a particular channel can be configured through the channel [DMA4\\_CICRi](#) register.

When an interrupt is detected, the logical DMA channel generating the event can first be identified by reading the [DMA4\\_IRQSTATUS\\_Lj](#) register. The event causing the interrupt then can be identified by reading the interrupt status via the relevant DMA channel [DMA4\\_CSRi](#) register.

### 16.4.3 Logical Channel Transfer Overview

As [Figure 16-6](#) shows, the sDMA module has one read port and one write port operating independently of each other. Buffering is provided between the read and write ports through a FIFO queue memory pool that is shared dynamically between the active logical channels.

- Logical channel synchronization
  - A logical channel is described as hardware-synchronized when the DMA transfers are triggered by DMA requests from a hardware device. Alternatively, a logical channel is described as nonsynchronized when the DMA transfer is triggered by software.
- Logical channel activation
  - A logical channel becomes active as follows:
    - For hardware-synchronized transfers, when the logical channel is enabled and the hardware DMA request line is asserted
    - For software-triggered (nonsynchronized) transfers, as soon as software enables the logical channel
- Logical channel transfer composition
  - A DMA transfer is divided automatically into a number of transactions. Depending on the logical channel context configured, the transfer size, the start address alignment, the addressing mode, and the configured maximum burst size, each transaction can be either a single access or a burst of accesses.
- Logical channel scheduling

When several logical channels are active at the same time, schedulers manage the read and write ports. The scheduling of logical channel transfers is similar for both read and write ports. When a logical channel becomes active, it is added to the tail of a scheduling queue. If more than one logical channel becomes active at the same time, the one with the lower number is queued first. This mechanism provides a first-come, first-serve scheduling scheme between the concurrently active logical channels.

In addition, each read and write port has a high-priority queue and a low-priority queue. The priority bits (WRITE\_PRIORITY and READ\_PRIORITY) in the logical channel DMA4\_CCRi register determine if a logical channel is queued as high or low priority. A software-configurable 8-bit priority counter gives weighting to the priority write queue. For every N (1 to 255) schedules from the priority write queue, one is scheduled from the regular write queue. A channel that is scheduled goes to the end of the queue after it completes its turn on the port. The relative weighting of the scheduling of the high-priority queue to the low priority queue is programmable from 1:1 to 1:256 through the DMA global channel register using the DMA4\_GCR[23:16] ARBITRATION\_RATE bit field.

---

**NOTE:** The DMA4\_GCR[23:16] ARBITRATION\_RATE field does not depend on the DMA4\_GCR[13:12] HI\_THREAD\_RESERVED field. The ARBITRATION\_RATE field depends on the DMA4\_CCRi[26] WRITE\_PRIORITY bit and the DMA4\_CCRi[6] READ\_PRIORITY bit.

---

- Read/write port access scheduling policy

When either the read or write port becomes available, the port access scheduler selects the next logical channel for which to perform a DMA transaction from either the high- or low-priority queue.

When the current DMA transaction (single or burst access) is complete and the full DMA transfer is not finished, the logical channel returns to the tail of the queue. Because the port access scheduling is on a per-transaction basis, a logical channel can be queued repeatedly this way several times during its block transfer.

The sDMA module can have up to four outstanding read transactions and two outstanding write transactions in the system interconnect; four read and two write thread IDs exist. For an arbitration cycle to occur, these two conditions must be met:

- At least one channel is requesting.
- At least one free thread ID is available.

On an arbitration cycle, the scheduler grants the highest priority channel that has an active request, allocates the thread ID, and tags this thread as busy. At a given time, a channel cannot be allocated for more than one thread ID.

---

**NOTE:** If more than one channel is active, each channel is given a thread ID for the current service only, not for the whole channel transfer.

---

When only one channel is enabled, only one thread is allocated for the channel. In such a situation the channel can have maximum of four outstanding commands (without getting the responses) without rescheduling the channel at the end of each transaction. Each command can be either single access (8-bit, 16-bit or 32-bit) or burst access (2 x M, 4 x M, 8 x M or 16 x M, where M can be 8-bit, 16-bit or 32-bit).

When nonburst alignment is at the beginning of the transfer, the channel is rescheduled for each smaller access until burst-aligned. When the end of the transfer is not burst-aligned, the channel is rescheduled for each of the remaining smaller accesses.

For a logical channel transfer completion, when the last access is written to the destination, the logical channel becomes inactive. If enabled, an interrupt request is generated (see [Section 16.4.2.1, Interrupt Generation](#)).

#### 16.4.4 FIFO Queue Memory Pool

A FIFO queue memory pool provides buffering between the read and write ports. The hardware allocates the space dynamically to a number of FIFO queues, and each queue is associated with an active logical channel.



To avoid a memory pool overflow, if there are fewer entries in the FIFO queue memory pool than are required for the maximum configured source burst size of the next logical channel to be scheduled, the logical channel is returned to the tail of the queue, and the port access scheduler continues to search the queue until it finds a logical channel that can be scheduled.

The maximum FIFO depth that can be allocated to each individual logical channel can be limited globally through the `DMA4_GCR[7:0] MAX_CHANNEL_FIFO_DEPTH` bit field. This value should be configured to allow a fair allocation of the memory pool between the active channels.

A logical channel is scheduled if it has not yet reached its allocation limit, even if the access to be performed will exceed this limit. This means that the effective number of entries used by a particular logical channel is limited to the configured maximum entries per channel + channel maximum configured burst size (in words) 1.

### 16.4.5 Addressing Modes

A DMA transfer block consists of a number of frames (FN). Each frame consists of a number of elements (EN), and each element can have a size of 8, 16, or 32 bits (ES), as follows:

$$\text{transfer block size} = \text{FN} \times \text{EN} \times \text{ES}$$

The FN, EN, and ES are common for both the source and destination. However, the way in which the data is represented (addressing profile/mode) is independently programmable for the source and destination devices, using one of these four addressing modes:

- Constant: The address remains the same for consecutive element accesses.
- Post-increment: The address increases by the ES, even across consecutive frames.
- Single-index: The address increases by the ES plus the element index (EI) value minus 1 (even across consecutive frames).
- Double-index: The address increases by the ES plus the EI value minus 1 within a frame. When a full frame is transferred, the address increases by the ES plus the frame index (FI) value minus 1.

The ES, EI, and FI values are expressed in bytes. The EI and FI values can be positive or negative.

When calculating the EI and FI values, it is critical to note that, after an element is accessed, the logical channel address pointer equals the address of the last byte (highest address) of the accessed element. The correct value for the EI or FI should be such that, when added to the logical channel address pointer, results in the address of the first byte (lowest address) of the next element to be accessed.

The EI and FI values must be configured so that the address of each element in the transfer is aligned on an ES boundary.

Consequently, the single-index addressing mode with EI = 1 or double-index addressing mode with EI = 1 and FI = 1 is equivalent to post-increment addressing.

---

**NOTE:** The source and destination start addresses must also be aligned on an ES boundary.

---

When the address of an element to be accessed is not aligned on an ES boundary, the transfer is stopped and a misaligned address error interrupt occurs, if enabled (see [Section 16.4.2.1, Interrupt Generation](#)).

The `DMA4_CFNi` register configures the FN in a block.

The `DMA4_CENi` register configures the EN.

The `DMA4_CSDPi` register configures the ES.

The `DMA4_CSSAi` and `DMA4_CDSAi` registers configure the source and destination start addresses.

The `DMA4_CCRi` register configures the source and destination addressing modes.

The `DMA4_CSEIi`, `DMA4_CSFli`, `DMA4_CDEIi`, and `DMA4_CDFli` registers configure the source EI, source FI, destination EI, and destination FI, respectively.

The addressing profiles are expressed as equations as follows:

Equation 1. Constant addressing:

$$A(n + 1) = A(n)$$

Equation 2. Post-increment addressing:

$$A(n + 1) = A(n) + ES$$

Equation 3. Single-indexed addressing:

$$A(n + 1) = A(n) + ES + (EI - 1)$$

Equation 4. Double-indexed addressing:

When not at the end of a frame or transfer (that is, when the element counter  $\neq 0$ ):

$$A(n + 1) = A(n) + ES + (EI - 1)$$

When at the end of a frame but not at the end of the transfer (that is, when the element counter = 0 and the frame counter  $\neq 0$ ):

$$A(n + 1) = A(n) + ES + (FI - 1)$$

Calculate the element and frame index as follows:

Equation 5. Element index

$$EI = [(Stride EI - 1) * ES] + 1$$

Equation 6. Frame index

$$FI = [(Stride FI - 1) * ES] + 1$$

where:

A(n): Byte address of the element  $n$  within the transfer.

ES is in bytes, ES{1, 2, 4}.

EI is in bytes, specified in a configuration register, 32768 EI 32767.

Stride EI: The difference in the number of elements between the start of the current element  $n$  to the start of next element,  $n+1$ .

Element counter: A counter that is (re)initiated with the number of elements per frame or per transfer. Decreased by 1 for each element transferred. The initial value is configured in the register DMA channel element number, DMA4\_CENi.

FI is in bytes, specified in a configuration register, 2147483648 FI 2147483647.

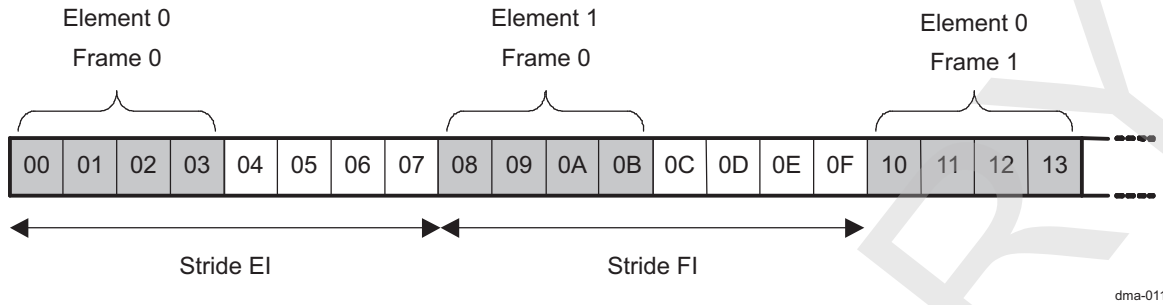
Stride FI: The difference in the number of elements between the start of the last element of the current frame and the beginning of the first element of the next frame.

Frame counter: A counter that is (re)initiated with the FN per transfer. Decreased by 1 for each frame transferred. The initial value is configured in the register DMA channel frame number, DMA4\_CFNi.

Figure 16-7 shows how a stride EI and FI are defined. When handling complex configurations, using strides can make it easier to calculate EI and FI because you can calculate in elements instead of bytes. (This approach is used in the 90-degree clockwise image rotation example shown in Figure 16-11.) The double-index addressing example shown in Figure 16-7 uses ES = 4, EN = 2, EI = 5, FI = 5, and FN = 2.

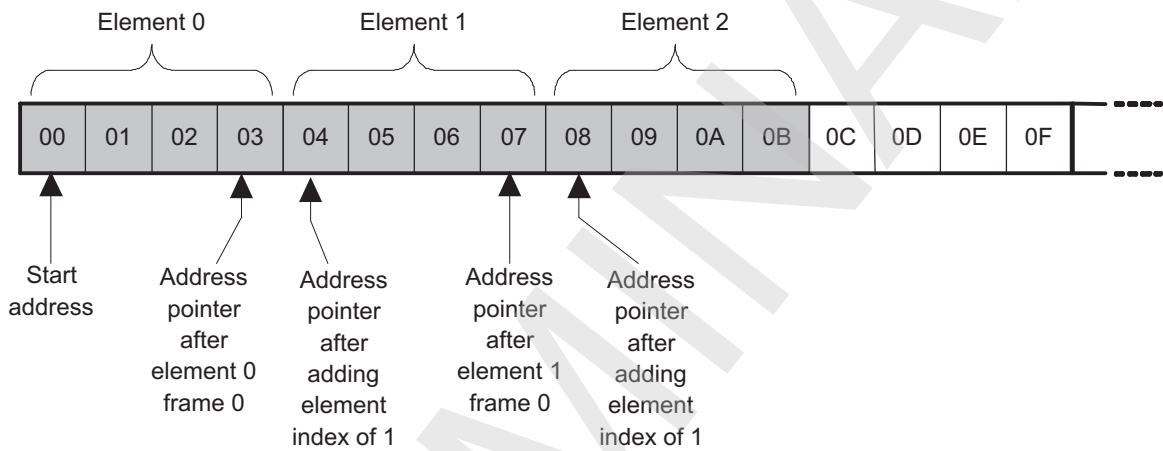
Figure 16-7 through Figure 16-10 show examples of addressing mode configurations. Table 16-9 lists parameter values for the examples.

**Figure 16-7. Example Showing Double-Index Addressing, Elements, Frames, and Strides**



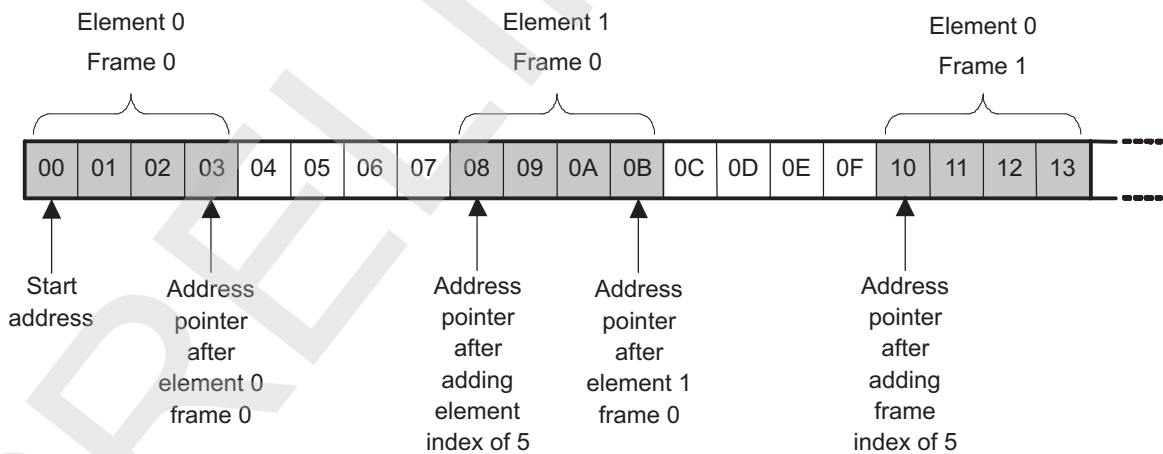
dma-011

**Figure 16-8. Addressing Mode Example (a)**

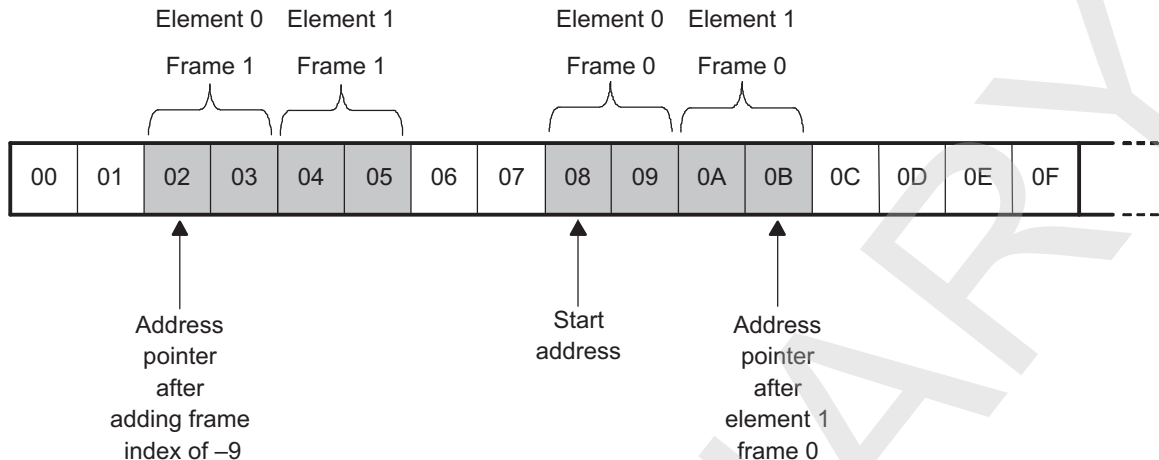


dma-010

**Figure 16-9. Addressing Mode Example (b)**



dma-009

**Figure 16-10. Addressing Mode Example (c)**

dma-008

**Table 16-9. Parameter Values for Addressing Mode Examples (a), (b), and (c)**

Parameter	Example (a)	Example (b)	Example (c)
Addressing mode	Single index (or post-increment)	Double index	Double index
Start address	0	0	8
ES	4 (32-bit)	4 (32-bit)	2 (16-bit)
EN	3	2	2
EI	1	5	1
FN	1	2	2
Frame index	N/A	5	-9

Double indexing can occur either on source (read) or destination (write). Equations for rotation of  $xx$  degrees on destination are obtained by taking equations for rotation of  $(360 - xx)$  degrees on source, and swapping the width ( $x$ ) and height ( $y$ ) of the image in them. The opposite is also true. Table 16-10 lists the equations for 90-, 180-, and 270-degree rotations.

**Table 16-10. Equations for Rotation**

		90 Rotation	180 Rotation	270 Rotation
Double indexing on destination (write)	Base address	$ES*(y-1)$	$ES*(x*y-1)$	$ES*y*(x-1)$
	EI	$ES*(y-1) + 1$	$1-2*ES$	$1-ES*(y + 1)$
	FI	$1 ES*[(x-1)*y + 2]$	$1-2*ES$	$1+ES*(x-1)*y$
Double indexing on source (read)	Base address	$ES*x*(y-1)$	$ES*(x*y-1)$	$ES*(x-1)$
	EI	$1-ES*(x + 1)$	$1-2*ES$	$ES*(x-1) + 1$
	FI	$1+ES*(y-1)*x$	$1-2*ES$	$1 ES*[(y-1)*x + 2]$

Table 16-11 and Figure 16-11 show the configuration required to perform a 90-degree clockwise rotation of a 240 x 160 pixel, 32-bit image. The EI, frame size, and FI values are configured so that the image is rotated line by line starting at the left end of the top line.

**NOTE:** The FI value for the destination is negative so that the first pixel of each subsequent line of the source image is written to the correct location at the destination.

Equation 5 and Equation 6 calculate the destination, FI and EI. The example assumes that the image lines are stored at consecutive addresses in memory, meaning that both EI and FI on the source side are 1.

**Rotations:**

Section 16.5.7, 90-Degree Clockwise Image Rotation, describes how to program an example of a 90-degree clockwise image rotation.

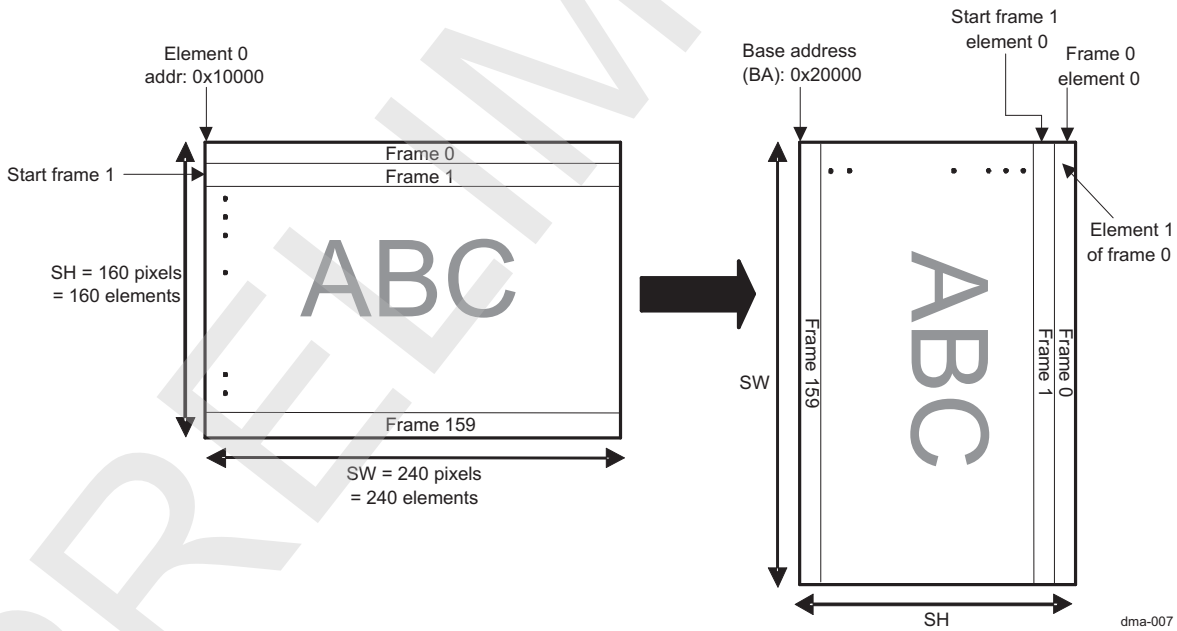
Observe that:

- One pixel = one element
- One line = one DMA frame
- Pixel size = element size = ES

**Table 16-11. Example Parameter Values for a 90-Degree Clockwise Image Rotation**

Parameter	Source Value	Destination Value
Bits per pixel	32	32
ES	4	4
Image width	SW	SH
Image height	SH	SW
Stride elements (stride EI)	1 element	SH
Stride frames (stride FI)	1 element	$-[(SW-1)*SH + 1] = 38,241$ elements
Start address	0x100000	$0x200000 + (SH - 1) \times ES = 0x20027C$
EN	SW	SW
EI	$[(Stride EI - 1) * ES] + 1 = 1$	$[(Stride EI - 1) * ES] + 1 = 637$
FN	SH	SH
FI	$[(Stride FI - 1) * ES] + 1 = 1$	$[(Stride FI - 1) * ES] + 1 = 152,967$

**Figure 16-11. Example of a 90-Degree Clockwise Image Rotation**



**16.4.6 Packed Accesses**

To pack data means to group data to match the bus size, thus optimizing a transfer. When the logical channel ES is less than the DMA module read/write port size, and the addressing profile supports it (post-increment mode or single- or double-index mode with EI = 1), the number of elements to transfer in each read/write port access can be maximized by specifying that the source or destination is packed through the channel DMA4\_CSDPi register. Thus:

- For a read/write port size of 32 bits, the source or destination can be configured as packed for transfer ESs of 8 bits (four elements per access) and 16 bits (two elements per access).

- For a read/write port size of 64 bits, the source or destination can be configured as packed for transfer ESs of 8 bits (eight elements per access), 16 bits (four elements per access), and 32 bits (two elements per access).

Depending on the start address and transfer length, the first or last packed access can be only partially filled. This is indicated to the source or destination using the byte-enable signals.

### 16.4.7 Burst Transactions

Transfer performance can be improved so that the source or destination and addressing profile supports it. This can be achieved by configuring the logical channel to perform burst transactions consisting of multiple instead of single accesses. The channel can be programmed to use burst sizes equal to 16, 32, or 64 bytes through the `DMA4_CSDPi` register, with the read burst size programmable independently of the write burst size. Typically, the optimal burst size is 64 bytes (16 accesses for a 32-bit read/write port size or 8 accesses for a 64-bit read/write port size).

To obtain the maximum benefit from burst transactions, the source and destination start addresses must be aligned with the burst size. If this is not the case, the start of the transfer can consist of a number of smaller (single or burst) transactions until the first burst size boundary is reached.

Similarly, if the end of the transfer is not aligned on a burst size boundary, the final part of the transfer can consist of a number of smaller transactions.

---

**NOTE:** If post-incrementing is used, data must be packed to DMA data-port width, to use burst.

---

### 16.4.8 Endianism Conversion

The source and destination are each specified as little-endian or big-endian through the `DMA4_CSDPi` register for the particular logical channel. If the endianism of the source and destination differ, and if the logical channel ES is less than the sDMA module read/write port size, an endianism conversion is applied to the data before it is written to the destination.

When transferring data between a source and a destination with different endianism, it is important to specify an ES that equals the type of data being transferred to preserve the correct data image at the destination.

In the system, endianism conversion can be performed in more than one place. It is possible to instruct the source and/or destination to lock the endianism (that is, to not perform a conversion) through the logical DMA channel `DMA4_CSDPi` register.

### 16.4.9 Transfer Synchronization

A logical channel can be programmed for either software-triggered or hardware synchronized transfers.

#### 16.4.9.1 Software Synchronization

A transfer is software-triggered when the logical channel is set up and started by software. To specify a software-triggered transfer, set the channel DMA `DMA4_CCRi[4:0]` and `DMA4_CCRi[20:19]` bit fields to 0. The transfer starts as soon as the DMA `DMA4_CCRi[7]` bit is set (when it enters the scheduling process).

#### 16.4.9.2 Hardware Synchronization

A transfer is hardware-synchronized if the logical channel activation is driven by hardware requests from the source or destination target. A hardware-synchronized transfer is specified by configuring the DMA request line number in the channel `DMA4_CCRi` register to a value that corresponds to the DMA request line from the source or destination that generates the DMA requests. The DMA request numbers to be configured are specified in the DMA request mapping (see [Table 16-8](#)).

Specify the DMA request number in the `DMA4_CCRi[4:0]` SYNCHRO\_CONTROL and `DMA4_CCRi[20:19]` SYNCHRO\_CONTROL\_UPPER bit fields. After the `DMA4_CCRi[7]` ENABLE bit is set, the logical channel becomes enabled but not activated (it does not enter the scheduling process), which means that channel registers are not updated until the first DMA request is received.

**NOTE:** The channel synchronization control registers are 1-based. For example, to enable the S\_DMA\_1 request, the DMA4\_CCRi[4:0] SYNCHRO\_CONTROL bit field must be set to 0x2 (DMA request number + 1).

**NOTE: DMA Request Line**

A DMA request line must not be shared between concurrently enabled DMA channels. However, a DMA request line can be shared among several chained logical channels.

For hardware synchronization, the amount of data to be transferred for each assertion of the DMA request line is configured through the frame synchronization (FS) and block synchronization (BS) bits in the logical channel DMA4\_CCRi register and the DMA4\_CCRi[5] FS and DMA4\_CCRi[18] BS bits, respectively.

The amount of data can be any of the following:

- A single element transfer: A complete element defined by data type. For example, 8/16/32 bits are transferred in response to a DMA request.
- A full frame: A complete frame of several elements is transferred in response to a DMA request.
- A full block (a full channel transfer): A complete block of several frames is transferred in response to a DMA request.
- A full packet (a full channel transfer): A complete packet of several elements is transferred in response to a DMA request.

Packets allow the size of each part of the full DMA transfer to be configured independently of the organization of the data to be transferred (typically a number of elements). This can be useful when the source or destination has a buffer (such as a FIFO queue) with a size unrelated to the frame size of the transfer. The packet size then can be set to the size of the buffer.

Packet transfer must be used only where the source or destination is addressed in constant addressing mode, because FI registers are reused to specify the packet size.

To support the burst mode, the logical channel must also be configured in target-port packed access mode.

The packet size is configured based on the DMA4\_CCRi[24] SEL\_SRC\_DST\_SYNC bit through either the channel DMA4\_CDFIi register (source synchronized) or the DMA4\_CSFIi register (destination synchronized).

When the logical channel transfer block is not an exact multiple of the packet size, the final packet consists of the remaining elements in the transfer, using burst or single accesses to complete the block transfer.

The maximum transfer size, regardless of the packet size, is always as follows:

$$\text{Block\_size} = \text{Number\_of\_Frame\_in\_Block} * \text{Number\_of\_Element\_in\_Frame} * \text{Element\_Size}$$

- Synchronized at the source

The DMA module optimizes the transfer with respect to the number and size of burst transactions for the given source and destination addressing profiles and configured maximum burst sizes. When writing to the destination is slower than reading from the source, data is buffered in the channel FIFO queue. If the transfer is packet-synchronized at the source, the end-of-packet interrupt is disabled (see [Section 16.4.13, Reprogramming an Active Channel](#)).

For a source synchronized transfer, buffering can be enabled or disabled by setting the DMA4\_CCRi[25] BUFFERING\_DISABLE bit. For a packet source synchronization with buffering disabled and the packed/burst across the packet boundary, the last packed/burst write transaction is split in optimized smaller accesses to complete the packet transfer size. However, for a packet source synchronized transfer with buffering enabled and with the packed/burst across the packet boundary, the DMA module waits for the next DMA request(s) to read enough data to issue an atomic packed/burst write transaction (assuming that the address is packed/burst aligned).

**NOTE:** Buffering is not performed between frames, even if it is enabled. If the packed/burst is across the frame boundary, the last packed/burst write transaction is split in optimized smaller accesses to complete the frame transfer size.

- Synchronized at the destination

The performance of a hardware-synchronized transfer can be improved by using the prefetch mode, enabled through the channel [DMA4\\_CCRi\[23\]](#) PREFETCH bit. Data is prefetched on the read port side before the DMA request received and buffered in the FIFO queue. Up to a full transfer block can be prefetched, although this can be limited by the specified maximum channel FIFO queue depth (see [Section 16.4.4, FIFO Queue Memory Pool](#)).

Buffering disable is not allowed for a destination-synchronized transfer.

**NOTE:** Behavior is undefined when prefetch is enabled and a transfer is synchronized to the source.

Regardless of whether buffering is enabled, the last transaction in the frame or in the block is write nonposted (WNP) even if the write mode is specified as write last nonposted (WLNP; the [DMA4\\_CSDPi\[17:16\]](#) WRITE\_MODE bit field = 0x2). However, in a packet synchronization mode, the last transaction of each packet in the transfer is WNP only if the buffering disable is on (even if the write mode is specified as WLNP).

Regardless of whether buffering is enabled, the packet interrupt is not generated in the packet source synchronized mode.

#### CAUTION

The [DMA4\\_CCRi\[25\]](#) BUFFERING\_DISABLE bit must be filled with an allowed value, as specified in [Table 16-12](#).

**Table 16-12. Buffering Disable**

	<b>BUFFERING_DISABLE</b> (0: Buffering enable, 1: Buffering disable)	
Destination synchronized	0	Allowed
	1	Not allowed
Source synchronized	0	Allowed
	1	Allowed

- Synchronized transfer monitoring using CDAC ([DMA4\\_CDACi](#)):

Context is restored only when the channel becomes active on a DMA request (not at software enable). The channel is software-enabled first, and then a DMA request is asserted followed by the first context restore.

The CDAC register is writable; thus, the CDAC can be initialized to monitor the transfer and determine whether the transfer is started (for more information, see [Section 16.5.4, Synchronized Transfer Monitoring Using CDAC](#)).

**NOTE:** For 16-bit transactions, start reading from or writing to the LSByte first to enable the register update. This is not an issue for 32-bit read-write transactions.

### 16.4.10 Thread Budget Allocation

When several concurrent channels are latency critical and hardware synchronized, a specific latency cannot be ensured until the target is served. This situation occurs when the number of concurrent channels is greater than the number of available threads.



---

**NOTE:** Four threads are available on the read port, and two threads are available on the write port.

---

For a hardware-synchronized transfer (memory to peripheral), a minimum bandwidth for a latency-critical transfer must be ensured to avoid collisions between two hardware requests.

Because it is latency critical, the software user is responsible for the following:

- Programming the synchronized channel as a high-priority channel
- Reserving one or several threads for high-priority channels

The proposed implementation is as follows (see [Section 16.5.5, Concurrent Software and Hardware Synchronization](#)):

Prevent the regular channel queue from exceeding more than a programmable (3, 2, or 1) number of threads on the read port and no more than one thread on the write port. This number can be set in the global register [DMA4\\_GCR\[13:12\]](#).

The thread reservation is programmable for maximum use of thread resources for concurrent, low-priority channel transfer. Programmability can also allow a partial throughput control by limiting in software the number of concurrent outstanding requests that break the pipelining.

Depending on the [DMA4\\_GCR \[13:12\]](#) value, the following threadID on the read/write ports are allocated for a high-priority channel:

Read port priority thread reservation:

- [DMA4\\_GCR\[13:12\] = 0x0](#) => No ThreadID is reserved for high-priority channels.
- [DMA4\\_GCR\[13:12\] = 0x1](#) => Read ThreadID 0 is reserved for high-priority channels.
- [DMA4\\_GCR\[13:12\] = 0x2](#) => Read ThreadID 0 and Read ThreadID 1 are reserved for high-priority channels.
- [DMA4\\_GCR\[13:12\] = 0x3](#) => Read ThreadID 0, Read ThreadID 1, and Read ThreadID 2 are reserved for high-priority channels.

Write port priority thread reservation:

- [DMA4\\_GCR\[13:12\] = 0x0](#) => No ThreadID is reserved for high-priority channels
- [DMA4\\_GCR\[13:12\] = 0x1](#) => Write ThreadID 0 is reserved for high-priority channels.
- [DMA4\\_GCR\[13:12\] = 0x2](#) => Write ThreadID 0 is reserved for high-priority channels.
- [DMA4\\_GCR\[13:12\] = 0x3](#) => Write ThreadID 0 is reserved for high-priority channels.

Regardless of whether the enabled channels are high priority, only the setting of the [DMA4\\_GCR\[13:12\]](#) value forces the thread reservation to these values. Set the appropriate value to avoid losing threads using only regular channels.

To have an independent read and write priority context, a per-channel bit ([DMA4\\_CCRI\[26\]](#)) is added for write priority, and the previous priority bit becomes read priority bit ([DMA4\\_CCRI\[6\]](#)).

---

**NOTE:** The device has one priority bit per logical channel, not one priority bit per port.

---

### 16.4.11 FIFO Budget Allocation

To avoid fully occupying the FIFO with a high-priority transfer while low-priority channels wait in the arbitration queue, two separate FIFO budgets are specified: one for high-priority channels and one for low-priority channels. This is defined in the [DMA4\\_GCR](#) register, allowing the user to share the FIFO budget between the low- and high-priority channels. The amount of the FIFO allocated by the low- and high-priority channels is fixed by the value set in the [DMA4\\_GCR\[15:14\]](#) HI\_LO\_FIFO\_BUDGET field. The maximum channel FIFO depth is limited by the HI\_LO\_FIFO\_BUDGET field as follows:

If the channel is low priority:

- When HI\_LO\_FIFO\_BUDGET = 0x1, then low priority cannot exceed 75 percent of the total FIFO.
- When HI\_LO\_FIFO\_BUDGET = 0x2, then low priority cannot exceed 25 percent of the total FIFO.
- When HI\_LO\_FIFO\_BUDGET = 0x3, then low priority cannot exceed 50 percent of the total FIFO.

If channel is high priority

- When HI\_LO\_FIFO\_BUDGET = 0x1, then high priority cannot exceed 25 percent of the total FIFO.
- When HI\_LO\_FIFO\_BUDGET = 0x2, then high priority cannot exceed 75 percent of the total FIFO.
- When HI\_LO\_FIFO\_BUDGET = 0x3, then high priority cannot exceed 50 percent of the total FIFO.

The user is responsible for performing the following equation:

- For a high-priority channel: (Per\_Channel\_Maximum FIFO Depth + 1) x Number of High Channel =< High Budget FIFO
- For a low-priority channel: (Per\_Channel\_Maximum FIFO Depth + 1) x Number of Low Channel =< Low Budget FIFO

---

**NOTE:** Ensure that *Number of High Channel* means *Number of Active High-Priority Channel* and that *Number of Low Channel* means *Number of Active Low-Priority Channel*.

---

#### 16.4.12 Chained Logical Channel Transfers

Chaining multiple logical channels permits transfers consisting of multiple parts to be executed without repeated software intervention. This results in better performance than the alternative of software setting up and starting each transfer separately. Each part of a chained transfer can have the data addressed in a different manner that permits the programming of a variety of complex transfers. For example:

- Interlaced video data with one logical channel configured to transfer the even lines and another logical channel configured to transfer the odd lines
- Protocol headers with a separate DMA4 channel configured to transfer each field in the header

Channels can be chained through each channel [DMA4\\_CLNK\\_CTRLi](#) register. When the transfer for the first channel completes, the next channel in the chain is enabled. The number of channels in the chain that are configured for hardware-synchronized transfers is flexible (although typically it might be all, none, or just the first one). The DMA request line number should be set to 0 to specify that any or all of the channels in a chain are software-triggered or nonsynchronized.

The last channel in a chain can be chained to the first channel to create a continuously looping chain. The continuously looping transfer can be stopped on the fly at a specific channel by disabling the [DMA4\\_CLNK\\_CTRLi\[15\]](#) ENABLE\_LNK bit. The looping transfer stops after the specified channel transfer is complete.

---

**NOTE: DMA Request Line**

A DMA request line must not be shared between concurrently enabled DMA channels. However, a DMA request line can be shared between several chained logical channels.

---

For more information on the programming model, see [Section 16.5, sDMA Basic Programming Model](#).

#### 16.4.13 Reprogramming an Active Channel

A currently active logical DMA channel can be disabled through the [DMA4\\_CCRi\[7\]](#) ENABLE bit. When an ongoing transaction is complete and the read-active and write-active bits in the [DMA4\\_CCRi](#) register ([DMA4\\_CCRi\[9\]](#) RD\_ACTIVE and [DMA4\\_CCRi\[10\]](#) WR\_ACTIVE) are reset, the channel can be reprogrammed for a new transfer.

#### 16.4.14 Packet Synchronization

A packet transfer notion is related to the behavior of some peripherals, which have certain buffering capability and requires to transfer the buffer content once an element number threshold is reached (a hardware DMA request is generated). To associate a frame synchronization to each DMA request is possible, but this limits the maximum transfer size. Indeed the maximum transfer size is proportional to the FIFO depth of the peripheral:

maximum\_transfer\_size = peripheral\_FIFO\_depth x number\_of\_frame\_in\_block

The packet synchronization allows to dissociate the transfer size from the FIFO depth of the peripheral. Only Constant addressing mode is allowed on a read port or a write port if source target or destination target is packet synchronized respectively.

*Example:*

Consider a camera interface with a FIFO\_depth of 128 words and a FIFO\_element\_number\_threshold of 128, and a picture to transfer with a size 320 lines by 240 columns. If frame synchronization is associated with each DMA request then the maximum transfer size that can be performed is  $128 \times 2^{16}$  words. In this case, a frame is 128-word long, which does not fit the size of a line, and it is not possible to generate an interrupt at the end of line. However, by introducing the packet transfer notion, which is related to the peripheral FIFO behavior/structure, the maximum transfer size (maximum\_transfer\_size =  $2^{24} \times 2^{16}$  words) is independent of both peripheral\_FIFO\_depth and FIFO\_element\_number\_threshold. This allows a long-enough transfer within one channel context and rotation operation on a large image format.

The main features of DMA Packet transfer are as follows:

- DMA Packet\_Data\_Size for each DMA Request: the Peripheral\_element\_number\_threshold (the number of elements in a packet) shares the DMA4\_CSFl<sub>i</sub> and DMA4\_CDFl<sub>i</sub> configuration registers. If the peripheral is the source target, the addressing mode is constant, and the DMA4\_CSFll[15:0] bit field is used to specify the packet data size in the DMA4\_CSFll register. The user must set the DMA4\_CCRi[24] SEL\_SRC\_DST\_SYNC bit to 1. If the peripheral is the destination target, the addressing mode is constant, the DMA4\_CDFll[15:0], is used to specify the packet data size (PKT\_ELNT\_NBR), and the bit field [31:16] is unused. To specify the packet data size in the DMA4\_CDFll register, the user must set the DMA4\_CCRi[24] SEL\_SRC\_DST\_SYNC bit to 0.

---

**NOTE:** The packet size can be a submultiple or non-submultiple of a frame size. If DMA Packet\_Data\_Size is aligned on the DMA channel block data size boundary, then DMA transfers the last data in the channel block boundary, and stops at the block boundary for the last packet DMA Request. If the Packet\_Data\_size is not aligned on the block boundary, the remaining data smaller than a packet size are transferred using burst or single accesses to complete the block.

---

- DMA Packet\_Data\_Transfer does not affect DMA channel capabilities in term of packing and bursting. The Packet synchronization mode is active when DMA4\_CCRi[5] FS = DMA4\_CCRi[18] BS = 1. Then
  - if DMA4\_CCRi[24] SEL\_SRC\_DST\_SYNC=0, the DMA4\_CDFll[15:0] bit field gives the number of elements in the packet and the DMA4\_CDFll[31:16] bit field is unused for the packet size.
  - if DMA4\_CCRi[24] SEL\_SRC\_DST\_SYNC=1, the DMA4\_CSFll[15:0] bit field gives the number of element in the packet and the DMA4\_CSFll[31:16] bit field is unused for the packet size.

---

**NOTE:** The maximum transfer size, regardless of the packet size, is always:  
 Block\_size = Number\_of\_Frame\_in\_Block x Number\_of\_Element\_in\_Frame x Element\_Size.

If DMA channel packet/burst access is across the packet boundary, the DMA hardware automatically splits this packing/burst access into multiple smaller accesses that are aligned on the packet boundary. Otherwise, the DMA transfers data as a usual packing/burst access.

---

### 16.4.15 Graphics Acceleration Support

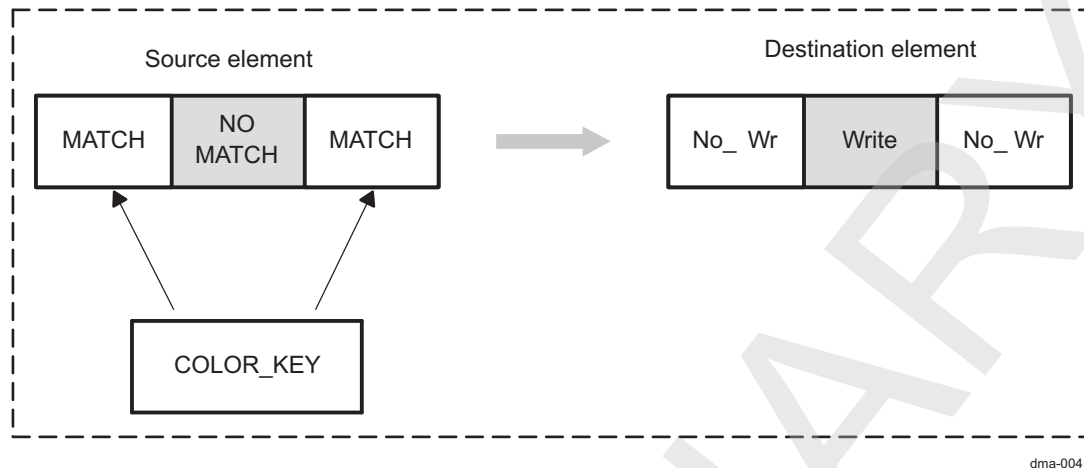
The sDMA supports two graphic acceleration features:

- Transparent copy
- Constant fill

Only one of these features can be enabled at a time through the DMA4\_CCRi register for the particular logical DMA channel.

The transparent copy feature enables specification of a particular color through the DMA4\_COLORi register, so that when it is recognized in the data from the source, it is not copied to the corresponding location in the destination, but instead leaves the data in the corresponding location in the destination as it is.

Figure 16-12 shows the 2-D graphic transparent color block diagram.

**Figure 16-12. 2-D Graphic Transparent Color Block Diagram**

The constant fill feature provides the ability to specify a particular color through the [DMA4\\_COLORi](#) register for every specified location in the destination. In this case, the transfer consists only of writing to the destination without reading from a source.

Both features support 8 bpp (bits per pixel), 16 bpp, and 24 bpp, depending on what is specified as the DMA transfer ES via the [DMA4\\_CSDPi](#) register. An ES of 32 bits corresponds to 24 bpp. During a 32-bit (24 bpp) transfer, the 8 most-significant bits (MSB) ([31:24]) are 0. Both features are compatible with packed and burst transactions.

#### 16.4.16 Supervisor Modes

A logical DMA channel can be configured to operate in supervisor mode through the [DMA4\\_CCRi\[22\]](#) SUPERVISOR bit. This must be done using supervisor access. Once a channel is configured in supervisor mode, the channel configuration is protected from nonsupervisor accesses. All DMA transactions on a supervisor channel are supervisor transactions.

#### 16.4.17 Posted and Nonposted Writes

A logical channel can be configured in its [DMA4\\_CSDPi\[17:16\]](#) bits to use one of three write access handshake modes for the destination:

- Nonposted write: Each write must complete before transfer can continue or complete.
- Posted write: Transfer continues without waiting for each write to complete (might improve performance with slow devices).
- Posted with final write nonposted: Transfer continues without waiting for each write to complete, but final write is completed before transfer can complete.

#### 16.4.18 Disabling a Channel During Transfer

When a channel is disabled during a transfer, the channel undergoes an abort, unless it is hardware source synchronized with buffering enabled ([DMA4\\_CCRi\[25\]](#) BUFFERING\_DISABLE = 0). If this is the case, the FIFO is drained to prevent the loss of data. See [Section 16.4.19, FIFO Draining Mechanism](#), for details about this feature.

#### 16.4.19 FIFO Draining Mechanism

When a source synchronized channel is disabled during a transfer, then the current hardware request (element/packet/frame/block) service is completed and the channel [DMA4\\_CCRi\[9\]](#) RD\_ACTIVE bit is set to 0, which means the channel is not active on the read port. The remaining data in the corresponding disabled channel FIFO is drained onto the write port and transferred to the programmed destination as in normal transfer.

At the end of the draining the `DMA4_CCRi[10] WR_ACTIVE` bit is set to 0 (channel is no longer active on the write port) and if the `DMA4_CICRI[12] DRAIN_END_IE` is set to 1, the `DMA4_CSRI[12] DRAIN_END` status bit is updated and an interrupt is generated.

Once a channel is disabled during a transfer, it must wait for `DMA4_CCRi[9] RD_ACTIVE` and `DMA4_CCRi[10] WR_ACTIVE` to become 0 before being reenabled for a new transfer. The FIFO drain for a channel occurs only in the following cases:

- If the channel is a source synchronized channel and `DMA4_CCRi[25] BUFFERING_DISABLE = 0` and
- If the channel is not a solid fill channel and
- If the channel is not a transparent and copy channel and
- If the channel is a hardware, synchronized channel

---

**NOTE:** For a self-linked or chain-linked channel, the user must disable the `DMA4_CLNK_CTRLi[15] ENABLE_LINK` bit before disabling the channel.

---

In all other cases, the channel undergoes an abort.

## 16.4.20 *Linked List*

### 16.4.20.1 Overview

The sDMA supports the logical transfer descriptor loader feature. A transfer descriptor represents a set of values that maps to a set of logical channel configuration registers.

A logical channel transfer descriptor can be loaded by DMA from memories, and then successive transfer descriptors can be autonomously loaded based on a linked-list scheme. This enables DMA4 scatter gather transfers with minimum MPU support by removing successive channel configuration processing and associated interrupt handling overheads. It also optimizes DMA4 channel resources by enabling efficient transfer serialization on a single logical channel versus concurrent (multiple) logical channel use.

Different types of transfer descriptors are supported (full or partial logical channel configuration registers are set). This optimizes the memory size required for storing a long linked list, because parameter changes are limited to only a few logical channel configuration registers.

### 16.4.20.2 Link-List Transfer Profile

A linked-list transfer can be seen as a super-block transfer (where the block is composed of FN frames and each frame includes EN elements). The block size (FN x EN x ES) can be changed in the linked list by loading an updated transfer descriptor.

The end of the super block is signaled in the last descriptor associated with the last block. Generally, for a given link-list transfer, the logical channel is set at the beginning of the transfer and the logical channel configurations for the subsequent blocks are slightly changed. Thus, the descriptor can be limited to an update of only few parameters, such as FN or EN. This assumes that the content of unmodified registers is preserved when a new descriptor is loaded.

A transfer descriptor is composed of a set of channel configuration register values with the addition of the next-descriptor pointer register (`DMA4_CNDPi`) and a channel-descriptor parameter register (`DMA4_CDPi`). The next-descriptor pointer is the 32-bit address pointer from where the next transfer descriptor is to be loaded. The next-descriptor pointer is mapped depending on the descriptor type (1, 2, or 3).

### 16.4.20.3 Descriptors

A transfer descriptor is a set of values that maps to a set of logical channel configuration registers. The descriptor contains the parameters associated with a transfer profile (transfer size, source or destination addresses, etc). Four different types of transfer descriptors are supported to optimize the memory size required to store a long linked list and to minimize MPU use to create and maintain the descriptor list.

A transfer descriptor is a list of 32-bit values. A descriptor must be 32-bit aligned in memory. Only the 30 least-significant bits (LSBs) of the next-descriptor address pointer are updated from the descriptor, and the DMA4 forces the 2 LSBs to 0 on generation of the pointer address. The descriptor size is variable depends on the descriptor type and the `Nxt_Dv` and `Nxt_Sv` bit fields.

Transfer descriptor bit mapping is the same as DMA4 logical-channel configuration register bit mapping, with the following exceptions:

- `Src_Element_index` and `Dst_Element_index` are concatenated in the same 32-bit location.
- `DMA4_CICRi` (interrupt event mask)
- CFN (frame number)
- Bit fields
  - P: Corresponds to the `PAUSE_LINK_LIST` bit:
    - When set to 1 in the descriptor, the channel is suspended when the descriptor load completes.
    - The user must not set the `PAUSE_LINK_LIST` bit through the configuration port. Otherwise, behavior is undefined.
    - When set to 0 (through the configuration port) after pause, the linked-list channel resumes its transfer (either descriptor load or data load).
  - B: Corresponds to the end-of-block enable bit (`BLOCK_IE`) of the `DMA4_CICRi` register; valid only for type 3. This value is don't care for descriptor types 1 and 2, where `DMA4_CICRi` is fully specified.
  - `Nxt_Dv`, `Nxt_Sv`: Mapped in the `DMA4_CDPi` register. They indicate one of the following possibilities:
    - Next descriptor contains an updated destination or source address.
    - Next descriptor does not update the source or destination address, but increments the last source or destination address (from the end of the last transfer).
    - The next source address and/or destination address are the last valid ones in the configuration memory. This means that the corresponding location in the configuration memory is not updated (assuming that they were initialized at least once in the past). This is also called wrapping addressing.
  - `Next_Descriptor_Type`: Specifies the next descriptor type that corresponds to the `NEXT_DESCRIPTOR_TYPE` bit field in the `DMA4_CDPi` register

#### 16.4.20.3.1 Type 1

A type 1 descriptor includes the overall channel configuration register value to be loaded (global registers are not part of the type 1 descriptor). This descriptor is used primarily when major changes are required:

- Channel read or write access profiles must be modified, for example, bursting and packing (included in the `DMA4_CSDPi` register)
- Attach a new DMA request to the same channel or change the priority or change the access privilege (included in the `DMA4_CCRi` register)
- Enable solid or transparent color fill (included in the `DMA4_CCRi` and `DMA4_COLORi` registers)
- Enable a channel link (included in the `DMA4_CLNK_CTRLi` register)

Table 16-13 shows a type 1 descriptor.

Table 16-13. Type 1

	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
Ptr+ 0x2C	CCR																																									
Ptr+ 0x28	CLNK_CTRL																																									
Ptr+ 0x24	CSDP																																									
Ptr+ 0x20	COLOR																																									
Ptr+ 0x1C	Src_Frame_index/Src_Packet_size																																									
Ptr+ 0x18	Dst_Frame_index/Dst_Packet_size																																									
Ptr+ 0x14	Src_Element_index																Dst_Element_index																									
Ptr+ 0x10	CICR (interrupt events mask)																CFN frame number																									
Ptr+ 0xC	Destination_Start_Address																																									
Ptr+ 0x8	Source_Start_Address																																									
Ptr+ 0x4	N_type	B	Dv	Sv	Element_number																																					
Ptr	Next_descriptor_address_pointer																																R	P								

16.4.20.3.2 Type 2

A type 2 descriptor includes the overall logical channel transfer address register and transfer format register to be loaded. This descriptor enables 2D addressing linked-list transfer (for example, a multimedia application where 2D objects are moved in a link). Table 16-14 shows a type 2 descriptor with source and destination address updates. Table 16-15 shows a type 2 descriptor with one source or destination address update.

Table 16-14. Type 2 With Source and Destination Address Updates

	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0
Ptr+ 0x1C	Src_Frame_index/Src_Packet_size																																									
Ptr+ 0x18	Dst_Frame_index/Dst_Packet_size																																									
Ptr+ 0x14	Src_Element_index																Dst_Element_index																									
Ptr+ 0x10	CICR (interrupt events Mask)																CFN frame number																									
Ptr+ 0xC	Destination_Start_Address																																									
Ptr+ 0x8	Source_Start_Address																																									
Ptr+ 0x4	N_type	B	Dv	Sv	Element_number																																					
Ptr	Next_descriptor_address_pointer																																R	P								

**Table 16-15. Type 2 With Source or Destination Address Update**

	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0
Ptr+ 0x18	Src_Frame_index/Src_Packet_size																															
Ptr+ 0x14	Dst_Frame_index/Dst_Packet_size																															
Ptr+ 0x10	Src_Element_index														Dst_Element_index																	
Ptr+ 0xC	CICR (interrupt events Mask)														CFN frame number																	
Ptr+ 0x8	Source_Start_Address or Destination_Start_Address																															
Ptr+ 0x4	N_type	B	Dv	Sv	Element_number																											
Ptr	Next_descriptor_address_pointer																										R sv	P				

**16.4.20.3.3 Type 3**

A type 3 descriptor is limited to a few logical channel transfer address registers and transfer format registers to be loaded. This descriptor enables simple 1D addressing link transfer (for example, scatter-gather or ping-pong memory movement using a linked list). [Table 16-16](#) shows a type 3 descriptor with source and destination address updates. [Table 16-17](#) shows a type 3 descriptor with one source or address destination update.

**Table 16-16. Type 3 With Source and Destination Address Updates**

	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0
Ptr+ 0xC	Destination_Start_Address																															
Ptr+ 0x8	Source_Start_Address																															
Ptr+ 0x4	N_type	B	Dv	Sv	Element_number																											
Ptr	Next_descriptor_address_pointer																										R sv	P				

**Table 16-17. Type 3 With Source or Destination Address Update**

	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	1 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0
Ptr+ 0x8	Source_Start_Address or Destination_Start_Address																															
Ptr+ 0x4	N_type	B	Dv	Sv	Element_number																											
Ptr	Next_descriptor_address_pointer																										R sv	P				

**16.4.20.4 Linked-List Control and Monitoring****16.4.20.4.1 Transfer Mode Setting**

Four descriptor types are available in [DMA4\\_CDPi\[9:8\] TRANSFER\\_MODE](#) to distinguish the different transfer modes:

- [DMA4\\_CDPi\[9:8\] TRANSFER\\_MODE = 00](#): The current channel is using normal mode.
- [DMA4\\_CDPi\[9:8\] TRANSFER\\_MODE = 01](#): The current channel is using link-list channel mode for a type 1, 2, or 3 descriptor.

The reset value is normal mode ([DMA4\\_CDPi\[9:8\] TRANSFER\\_MODE = 0](#)).



#### 16.4.20.4.2 Starting a Linked List

Like a nonlinked-list transfer, a link transfer starts under host control by enabling the associated logical channel (set the `DMA4_CCRi[7]` ENABLE bit to 1). The `DMA4_CDPi[10]` FAST bit sets the start mode of the link-list transfer:

In nonfast-start mode, the logical channel configuration is fully initialized so that the transfer can start without descriptor loading.

In fast-start mode, the descriptor pointer and other inputs are given. The channel starts by loading the descriptor and then starts the data transfer phase.

#### 16.4.20.4.3 Monitoring a Linked-List Progression

In addition to the `DMA4_CCENi` (remaining elements) and `DMA4_CCFNi` (remaining frames) registers that are used to monitor the transfer progress, a per-channel register, `DMA4_CCDNi` (channel current active descriptor number), monitors which descriptor in the list is active. The user must initialize the `DMA4_CCDNi` register to 0 during the initial configuration. When the `DMA4_CCDNi` register is updated, the `DMA4_CCFNi` and the `DMA4_CCENi` registers are updated. The user must also initialize the `DMA4_CCFNi` and `DMA4_CCENi` registers to 0xFFFF and to 0xFFFFFFFF, respectively, to track the effective transfer start of synchronized transfer.

#### 16.4.20.4.4 Interrupt During Linked-List Execution

Any logical channel source of interrupt can be triggered during a linked-list execution, if the interrupt source is enabled during the initial configuration in `CICR`. The `DMA4_CICRi` register can also be updated during the linked-list execution if descriptor types 1 and 2 are used.

The use of an interrupt event in a link execution can be difficult, because the link can progress in parallel with interrupt service routine (ISR) execution. This makes it difficult to synchronize them unless system assumptions are used. The most appropriate synchronization model is to get an interrupt-only on linked-list completion, when the last transfer block is complete. This prevents the interrupt from occurring during the link execution. An end-of-super-block interrupt event available in the `DMA4_CICRi` and `DMA4_CSRi` registers can be enabled at initial configuration or when using descriptor types 1 and 2. To prevent the use of descriptor type 1 or 2 to update `BLOCK_IE` (full `DMA4_CICRi` update), a dedicated `BLOCK_IE` bit field is also available in a type 3 descriptor.

#### 16.4.20.4.5 Pause a Linked List

When the channel is suspended, it remains enabled.

The pause behaves differently, depending on the transfer mode:

- Normal transfer mode: If the user sets the `DMA4_CDPi[7]` PAUSE\_LINK\_LIST bit to 1, the channel completes the current read and write transactions and then suspends the channel. The channel can be resumed by setting the channel `DMA4_CDPi[7]` PAUSE\_LINK\_LIST bit to 0.
- Linked-list type 1, 2, or 3 mode: The user must not set the `DMA4_CDPi[7]` PAUSE\_LINK\_LIST bit through the configuration port; otherwise, transfer behavior is undefined.
  - A PAUSE\_LINK\_LIST bit (P) is set to 1 in the descriptor.
    - The channel is suspended after the descriptor load, translation, and configuration memory update are complete.
    - The linked list can be resumed by resetting the `DMA4_CDPi[7]` PAUSE\_LINK\_LIST bit (through the configuration port).

#### 16.4.20.4.6 Stop a Linked List (Abort or Drain)

The channel can be stopped for a drain or an abort. These cases are exclusive.

##### 16.4.20.4.6.1 Drain

- Drain conditions:

A channel is a drain candidate if it is a hardware-source-synchronized transfer with `DMA4_CCRi[25]` BUFFERING\_DISABLE = 0 and should not be doing any of the graphics operation (transparent copy

or solid-color fill).

- Drain trigger:  
A drain candidate channel is drained if it is disabled (`DMA4_CCRi[7] ENABLE = 0`) or if it receives a transaction error on the read port.
- Drain behavior with a type 1, 2, or 3 descriptor. Drain trigger can occur in two situations:
  - During descriptor loading: Any ongoing current transaction is complete and the channel is aborted.
  - During data loading: The read is completed at the boundary of the request (element/frame/packet/block boundary), the FIFO is drained to the destination, and then a `DRAIN_END` interrupt can be asserted.

#### 16.4.20.4.6.2 Abort

- Abort condition:  
A channel is an abort candidate if it is software-synchronized, hardware-destination-synchronized, solid color-fill, transparent-color fill, or hardware-source-synchronized with `DMA4_CCRi[25] BUFFERING_DISABLE = 1`.
- Abort trigger:  
A channel is an abort candidate if it is disabled (`DMA4_CCRi[7] ENABLE = 0`), if it receives a transaction error on the read or write port, or if there is a `MISALIGNMENT_ERROR`.
- Abort behavior with a type 1, 2, or 3 descriptor:  
If an abort trigger occurs, the channel aborts immediately after completion of current read/write transactions and then the FIFO is cleaned up.  
In type 1, 2, or 3, if an abort trigger or drain trigger occurs during the descriptor load phase, the channel aborts.

#### 16.4.20.4.7 Status Bit Behavior

This section describes the behavior of the `DMA4_CSRi[6] SYNC`, `DMA4_CCRi[9] RD_ACTIVE` and `DMA4_CCRi[10] WR_ACTIVE` status bits:

- For a hardware-synchronized channel in linked-list mode, the `DMA4_CSRi[6] SYNC` bit becomes active (`DMA4_CSRi[6] SYNC = 1`) when the first data load transaction is scheduled and remains active until the last data load transaction in the block (not super block) is descheduled (`DMA4_CSRi[6] SYNC = 0`). The `SYNC` bit is not active during the descriptor load phase.
- The `DMA4_CCRi[9] RD_ACTIVE` bit is active during the data load phase and the descriptor load phase. It becomes active when the first read transaction is scheduled. It becomes inactive:
  - When (during the descriptor load phase) the last descriptor write request is descheduled
  - When (during the data load phase) the last read transaction in the block (not super block) is descheduled for software-synchronized transfer or destination-synchronized transfer with prefetch enabled
  - When (during the data load phase) the last read transaction in the request (element/frame/packet/block sync) is descheduled for hardware-source-synchronized transfer or hardware-destination-synchronized transfer without prefetch
- The `DMA4_CCRi[10] WR_ACTIVE` bit is active only during the data load phase. It becomes active when the first write transaction is scheduled and becomes inactive:
  - Until the last write transaction in the block (not super block) is descheduled and the FIFO is cleaned up for software-synchronized transfer
  - Until the last write transaction in the request (element/frame/packet/block sync) is descheduled and the FIFO is cleaned up for hardware-source-synchronized transfer (with `DMA4_CCRi[25] BUFFERING_DISABLE = 0`) or hardware-destination-synchronized transfer.

#### 16.4.20.4.8 Linked-List Channel Linking

Channel linking for inter and intra super blocks is supported for type 1, 2, and 3 descriptors.

Assume that CHx and CHz are linked-list channels using generic descriptors. If CHx is composed of N descriptors and CHz is composed of M descriptors, then in nonfast mode,

CHx: CHx[Data1]-> CHx[DES1] -> . -> CHx[DESN]->CHx[DataN + 1]

CHz: CHz[Data1]-> CHz[DES1] -> . -> CHz[DESM]->CHz[DataM + 1]

It is possible to link CHx to CHz or CHx to itself after the completion of the CHx transfer (end of super block). To do this, the user must set the [DMA4\\_CLNK\\_CTRLi\[15\] ENABLE\\_LNK](#) bit to 1 and the [DMA4\\_CLNK\\_CTRLi\[4:0\] NEXTLCH\\_ID](#) bit to z (or to x for self linking) through the last descriptor using a type 1 descriptor. The sequence is:

CHx: CHx[Data1]-> CHx[DES1] -> . -> CHx[DESN]->CHx[DataN+1] -> CHz: CHz[Data1]-> CHz[DES1] -> . -> CHz[DESM]->CHz[DataM+1]

It is also possible to link CHx to CHz during the CHx transfer and before the end of super block. The user must set the [DMA4\\_CLNK\\_CTRLi\[15\] ENABLE\\_LNK](#) bit to 1 and the [DMA4\\_CLNK\\_CTRLi\[4:0\] NEXTLCH\\_ID](#) bit to z through descriptor p (CHx[DESp]) using a type 1 descriptor. The sequence is:

CHx: CHx[Data1]-> CHx[DES1] ->.-> CHx[DESp]->CHx[Data(p + 1)] -> CHz[Data1]-> CHz[DES1] -> .

The user must continue the linking until channels CHx and CHz complete their super-block transfers; otherwise, the channels remain enabled.

---

**NOTE:** In channel linking, the head of a chain can be in fast mode or nonfast mode. All channels that are not in the head of the chain can be in nonfast mode only. In self-linking, the channel cannot be in fast mode.

---



---

**NOTE:** If channel CHx links to CHz in the middle of the superblock transfer (remember link bit can be set through Type-1 descriptor load), CHx will get disabled after the corresponding data load and enables the channel CHz.

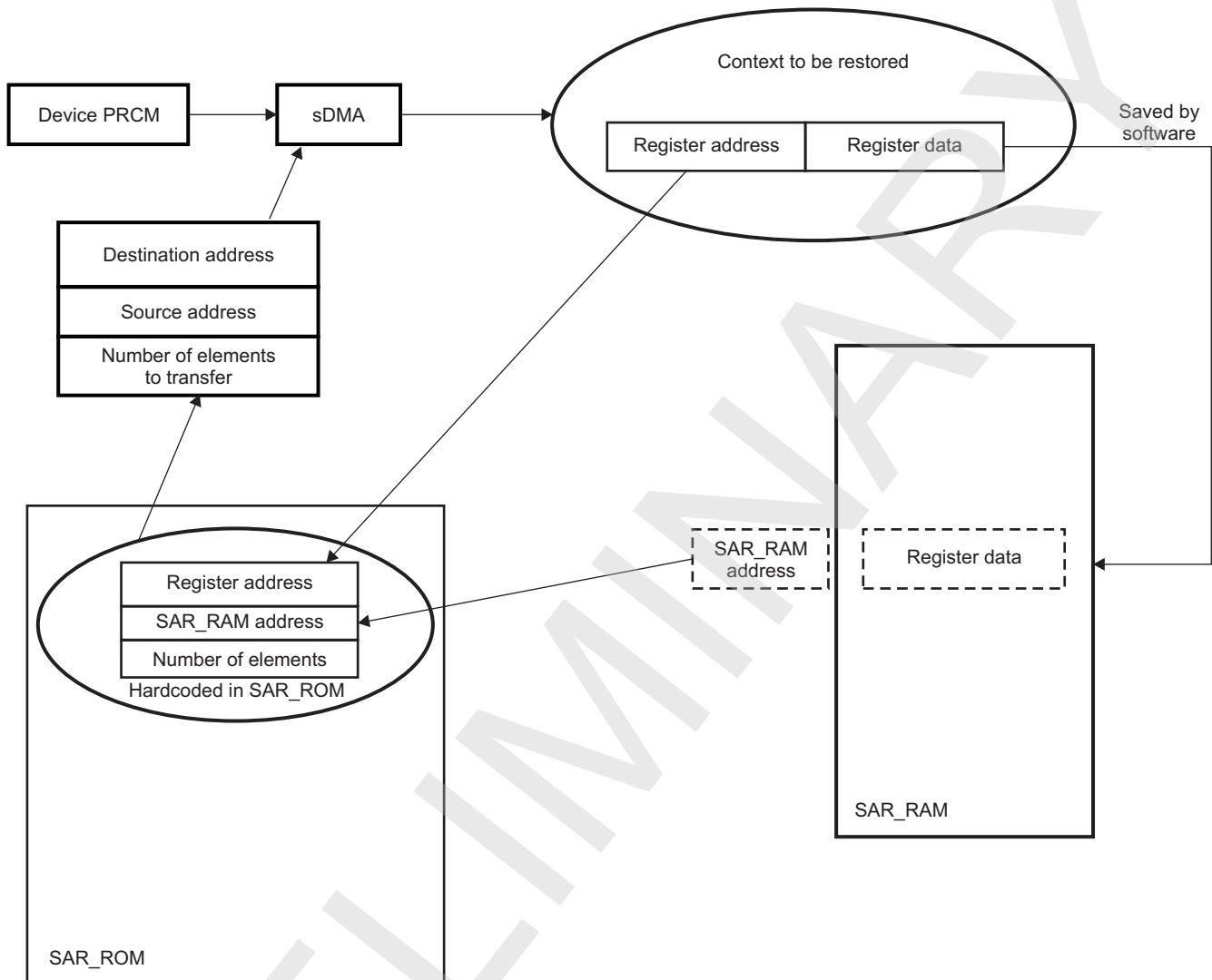
---

### 16.4.21 Auto-Restore Feature

The auto-restore feature is introduced mainly to accelerate the restore process, which otherwise is complex as well as time and resource consuming. In many system applications, it is required that the modem wake up and use only a few modules to wake up, without the main MPU (Cortex-A9 MPU) intervention. The MPU still might be in OFF mode; however, some modules might need to come out of OFF mode. The sDMA can be used to restore the register contents of these modules. The auto-restore feature consists in saving the registers data into a memory (SAR RAM sitting in the WKUP domain) that remains powered even during OFF-mode then it consists in retrieving automatically data when returning from OFF-mode.

The restore feature is enabled by default and cannot be disabled.

Figure 16-13 shows an overview of the save and restore mechanism using SAR RAM and SAR ROM.

**Figure 16-13. Save and Restore Mechanism Overview Using SAR RAM and SAR ROM**


dma-014

#### 16.4.21.1 Save Process

The save process is done entirely by software; the sDMA module normally is not involved. The software must save the data to be restored in a dedicated location in SAR RAM. To reduce the save sequence execution time, the sDMA can be used with linked-list descriptors. These save descriptors are symmetrical with regards to descriptors that lay in the SAR ROM.

The register content to be saved can be the register value before going into OFF-mode or a value that is required on wake up. This means that the restored register value can be different from the last value of the register before going into OFF-mode.

The following registers are impacted:

- CM1 and CM2 configuration  
Bits 0 of the CM\_SHADOW\_FREQ\_CONFIG1 register and the CM\_SHADOW\_FREQ\_CONFIG2 register are self-clearing and must be set at restore time. Thus, these data must always be overwritten in the SAR RAM.
- Because USBHOSTHS and USBTLL restore needs a particular sequencing, the software must overwrite data read from the following registers implied in phase2a and phase 2b:
  - **CM\_L3INIT\_HSUSBHOST\_CLKCTRL\_RESTORE**

- CM\_L3INIT\_HSUSBTLL\_CLKCTRL\_RESTORE
- CM\_SDMA\_STATICDEP\_RESTORE

The USBHOSTHS and USBTLL modules must be put in a particular mode when saving (and restoring) their context. In the case of context saving, the software must follow the following sequence:

- Configure the L3INIT clock domain to SW\_WKUP to ensure module transitions.
- Disable both modules.
- Assert the SAR\_MODE pins.
- Enable back both modules. Optional clocks must be kept gated.
- Save both modules context.
- Disable both modules.
- Deassert SAR\_MODE pins.
- At this point, the software might enable back both modules and get out L3INIT clock domain from the SW\_WKUP state.

### 16.4.21.2 Restore Process

As explained earlier, every time modem wakes up, the MPU must not necessarily wake up. This needs an automated way of restore. E.g .when mobile is in roaming, and not in active use, it needs to change base stations. For this task, the MPU need not wake up, only a few peripherals/subsystems need to be in ON mode. Before executing any task, we need to restore their register contents.

#### 16.4.21.2.1 PRCM and sDMA interaction During Restore Phase

Restore is performed during three phases. At the end of each phase, the sDMA stops and waits for the PRCM to perform the next phase transfer. For more information, see the *Device Off Mode Wake-Up Sequences* section in [Chapter 3, Power, Reset and Clock Management](#).

#### 16.4.21.2.2 sDMA Descriptors Description

The restore phase is automatically done by the sDMA. The sDMA gets the register address (**Destination address**), the **number of elements** to transfer and the address of the data to transfer (**Source address**) from SAR ROM. As mentioned previously the register data is stored in the SAR RAM.

SAR ROM content:

The SAR ROM contains two types of descriptors that the DMA-4 should use:

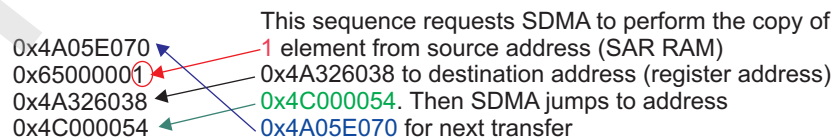
- Type-1 descriptors
- Type-3 descriptors with both source and destination address update

**Type-1** descriptor (48 bytes) is to be used at the beginning of each phase. Indeed, in addition of source and destination address and number of elements, it contains the required data that initializes the sDMA. [Table 16-13](#) shows the Type-1 descriptor structure.

**Type-3** descriptors (16 bytes) are to be used for subsequent sDMA transfers and contain mainly source and destination address and number of elements. [Table 16-16](#) shows the Type-3 descriptor structure.

[Figure 16-14](#) shows an example of SAR ROM code (Type-3 Descriptor):

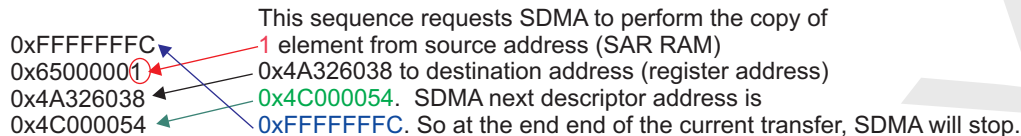
**Figure 16-14. Type-3 SAR ROM Code Example**



dma-015

[Figure 16-15](#) shows an example of Last DMA transfer (Type-3 Descriptor):

To indicate to the DMA that this is the last transfer, then the next descriptor address must be set to 0xFFFFFFF0 as in the example below :

**Figure 16-15. Type-3 SAR ROM Last transfer Example**

dma-016

### 16.4.21.3 SAR RAM

The SAR RAM contains the **data** which needs to be maintained during OFF-mode. The SAR RAM is split into 3 banks:

**Table 16-18. SAR RAM Allocation**

Access Mode	Bank	Address Range
public	1 (SAR_RAM_1)	0x4A32_6000 0x4A32_6FFF (4kB)
public, for IO Pad	2 (SAR_RAM_2)	0x4A32_7000 0x4A32_73FF (1kB)
public	3 (SAR_RAM_3)	0x4A32_8000 0x4A32_87FF (2kB)

### 16.4.21.4 SAR ROM

The SAR ROM contains the list of registers that need to be saved before OFF-mode and so restored after OFF-mode and contains also the location where the register data is stored in SAR RAM. The SAR ROM start address is 0x4A05\_E000.

When exiting OFF-mode, PRCM automatically redirects sDMA to address:

- 0x4A05\_E000 for phase 1 (in SAR ROM)
- 0x4A05\_E030 for phase 2a (in SAR ROM)
- 0x4A05\_E060 for phase 2b (in SAR ROM)

The SAR ROM is build so that sDMA will perform all the transfers for each phase. At the end of each phase, sDMA will stop. PRCM will then request to perform the next phase transfer depending on wakeup source.

#### Remarks:

- Phase 1 is always to happen, the RESTORE mechanism is enabled by default and can not be disabled.
- Phase 2 (2a and 2b) is to happen depending on the wake-up source from OFF-mode. Phase 2 will not be performed if C2C is the wake-up source, but it will be performed for all the other wake-up sources. So it is mandatory to correctly fill the SAR RAM locations used for USB restore, even USB is not used because the restore sequence will happen.
- USB host RESTORE is to be performed before USB TLL RESTORE

**NOTE:** Some consecutive address registers cannot be saved/restored with burst. This is the case for:

- CLK1\_DSS\_BW\_REGULATOR.L3\_BW\_R\_BANDWIDTH and CLK1\_DSS\_BW\_REGULATOR.L3\_BW\_R\_WATERMARK registers
- CLK2\_XXX\_BW\_REGULATOR.L3\_BW\_R\_BANDWIDTH and CLK2\_XXX\_BW\_REGULATOR.L3\_BW\_R\_WATERMARK registers, where XXX stands for ISS, IVAHD and SGX.

To avoid sDMA transfer failure, the transfer of the above register is performed using a single sDMA transfer.

### 16.4.21.5 SAR RAM Memory Allocation Reserved for Save and Restore

The memory size of the SAR RAM reserved for the Save and Restore mechanism is:

- 57% (2344 bytes) reserved for the SAR mechanism from SAR\_RAM\_1. The whole SAR\_RAM\_1 size is 4KB.
- 40% (408 bytes) reserved for the SAR mechanism from SAR\_RAM\_2. The whole SAR\_RAM\_2 size is 1KB.
- 51% (1052 bytes) reserved for the SAR mechanism from SAR\_RAM\_3. The whole SAR\_RAM\_3 size is 2KB.

The free part of the SAR RAM can be used by the user software to store any data that needs to be in always-on memory.

#### 16.4.21.6 Register Mapping in SAR RAM

Table 16-19 shows the SAR\_RAM\_1 memory mapping.

**Table 16-19. SAR\_RAM\_1 Memory Mapping**

Order	Register Name	Register Start Addr	Register End Addr	Size	SAR_RAM_1 Addr
1	EMIF1.EMIF_SDRAM_CONFIG	0x4C000008		4	0x4A326000
2	EMIF1.EMIF_LPDDR2_NVM_CONFIG	0x4C00000C		4	0x4A326004
3	EMIF1.EMIF_SDRAM_REF_CTRL	0x4C000010		4	0x4A326008
4	EMIF1.EMIF_SDRAM_REF_CTRL_SHDW	0x4C000014		4	0x4A32600C
5	EMIF1.EMIF_SDRAM_TIM_1	0x4C000018		4	0x4A326010
6	EMIF1.EMIF_SDRAM_TIM_1_SHDW	0x4C00001C		4	0x4A326014
7	EMIF1.EMIF_SDRAM_TIM_2	0x4C000020		4	0x4A326018
8	EMIF1.EMIF_SDRAM_TIM_2_SHDW	0x4C000024		4	0x4A32601C
9	EMIF1.EMIF_SDRAM_TIM_3	0x4C000028		4	0x4A326020
10	EMIF1.EMIF_SDRAM_TIM_3_SHDW	0x4C00002C		4	0x4A326024
11	EMIF1.EMIF_LPDDR2_NVM_TIM	0x4C000030		4	0x4A326028
12	EMIF1.EMIF_LPDDR2_NVM_TIM_SHDW	0x4C000034		4	0x4A32602C
13	EMIF1.EMIF_PWR_MGMT_CTRL	0x4C000038		4	0x4A326030
14	EMIF1.EMIF_PWR_MGMT_CTRL_SHDW	0x4C00003C		4	0x4A326034
15	EMIF1.EMIF_L3_CONFIG	0x4C000054		4	0x4A326038
16	EMIF1.EMIF_PERF_CNT_CFG	0x4C000088		4	0x4A32603C
17	EMIF1.EMIF_PERF_CNT_SEL	0x4C00008C		4	0x4A326040
18	EMIF1.EMIF_READ_IDLE_CTRL	0x4C000098		4	0x4A326044
19	EMIF1.EMIF_READ_IDLE_CTRL_SHDW	0x4C00009C		4	0x4A326048
20	EMIF1.EMIF_IRQENABLE_SET_SYS	0x4C0000B4		4	0x4A32604C
21	EMIF1.EMIF_IRQENABLE_SET_LL	0x4C0000B8		4	0x4A326050
22	EMIF1.EMIF_ZQ_CONFIG	0x4C0000C8		4	0x4A326054
23	EMIF1.EMIF_TEMP_ALERT_CONFIG	0x4C0000CC		4	0x4A326058
24	EMIF1.EMIF_DDR_PHY_CTRL_1	0x4C0000E4		4	0x4A32605C
25	EMIF1.EMIF_DDR_PHY_CTRL_1_SHDW	0x4C0000E8		4	0x4A326060
26	EMIF1.EMIF_DDR_PHY_CTRL_2	0x4C0000EC		4	0x4A326064
27	EMIF2.EMIF_SDRAM_CONFIG	0x4D000008		4	0x4A326068
28	EMIF2.EMIF_LPDDR2_NVM_CONFIG	0x4D00000C		4	0x4A32606C
29	EMIF2.EMIF_SDRAM_REF_CTRL	0x4D000010		4	0x4A326070
30	EMIF2.EMIF_SDRAM_REF_CTRL_SHDW	0x4D000014		4	0x4A326074
31	EMIF2.EMIF_SDRAM_TIM_1	0x4D000018		4	0x4A326078
32	EMIF2.EMIF_SDRAM_TIM_1_SHDW	0x4D00001C		4	0x4A32607C
33	EMIF2.EMIF_SDRAM_TIM_2	0x4D000020		4	0x4A326080
34	EMIF2.EMIF_SDRAM_TIM_2_SHDW	0x4D000024		4	0x4A326084
35	EMIF2.EMIF_SDRAM_TIM_3	0x4D000028		4	0x4A326088

**Table 16-19. SAR\_RAM\_1 Memory Mapping (continued)**

Order	Register Name	Register Start Addr	Register End Addr	Size	SAR_RAM_1 Addr
36	EMIF2.EMIF_SDRAM_TIM_3_SHDW	0x4D00002C		4	0x4A32608C
37	EMIF2.EMIF_LPDDR2_NVM_TIM	0x4D000030		4	0x4A326090
38	EMIF2.EMIF_LPDDR2_NVM_TIM_SHDW	0x4D000034		4	0x4A326094
39	EMIF2.EMIF_PWR_MGMT_CTRL	0x4D000038		4	0x4A326098
40	EMIF2.EMIF_PWR_MGMT_CTRL_SHDW	0x4D00003C		4	0x4A32609C
41	EMIF2.EMIF_L3_CONFIG	0x4D000054		4	0x4A3260A0
42	EMIF2.EMIF_PERF_CNT_CFG	0x4D000088		4	0x4A3260A4
43	EMIF2.EMIF_PERF_CNT_SEL	0x4D00008C		4	0x4A3260A8
44	EMIF2.EMIF_READ_IDLE_CTRL	0x4D000098		4	0x4A3260AC
45	EMIF2.EMIF_READ_IDLE_CTRL_SHDW	0x4D00009C		4	0x4A3260B0
46	EMIF2.EMIF_IRQENABLE_SET_SYS	0x4D0000B4		4	0x4A3260B4
47	EMIF2.EMIF_IRQENABLE_SET_LL	0x4D0000B8		4	0x4A3260B8
48	EMIF2.EMIF_ZQ_CONFIG	0x4D0000C8		4	0x4A3260BC
49	EMIF2.EMIF_TEMP_ALERT_CONFIG	0x4D0000CC		4	0x4A3260C0
50	EMIF2.EMIF_DDR_PHY_CTRL_1	0x4D0000E4		4	0x4A3260C4
51	EMIF2.EMIF_DDR_PHY_CTRL_1_SHDW	0x4D0000E8		4	0x4A3260C8
52	EMIF2.EMIF_DDR_PHY_CTRL_2	0x4D0000EC		4	0x4A3260CC
53	RESTORE_CM2. CM_MEMIF_CLKSTCTRL_RESTORE	0x4A009E0C		4	0x4A3260D0
54	RESTORE_CM1. CM_CLKSEL_CORE_RESTORE	0x4A004E00		4	0x4A3260D4
55	RESTORE_CM1. CM_DIV_M2_DPLL_CORE_RESTORE	0x4A004E04		4	0x4A3260D8
56	RESTORE_CM1. CM_DIV_M3_DPLL_CORE_RESTORE	0x4A004E08		4	0x4A3260DC
57	RESTORE_CM1. CM_DIV_M4_DPLL_CORE_RESTORE	0x4A004E0C		4	0x4A3260E0
58	RESTORE_CM1. CM_DIV_M5_DPLL_CORE_RESTORE	0x4A004E10		4	0x4A3260E4
59	RESTORE_CM1. CM_DIV_M6_DPLL_CORE_RESTORE	0x4A004E14		4	0x4A3260E8
60	RESTORE_CM1. CM_DIV_M7_DPLL_CORE_RESTORE	0x4A004E18		4	0x4A3260EC
61	RESTORE_CM1. CM_CLKSEL_DPLL_CORE_RESTORE	0x4A004E1C		4	0x4A3260F0
62	RESTORE_CM1. CM_SSC_DELTAMSTEP_DPLL_CORE_RESTORE	0x4A004E20		4	0x4A3260F4
63	RESTORE_CM1. CM_SSC_MODFREQDIV_DPLL_CORE_RESTORE	0x4A004E24		4	0x4A3260F8
64	RESTORE_CM1. CM_CLKMODE_DPLL_CORE_RESTORE	0x4A004E28		4	0x4A3260FC
65	RESTORE_CM1. CM_SHADOW_FREQ_CONFIG2_RESTOR E	0x4A004E2C		4	0x4A326100
66	RESTORE_CM1. CM_SHADOW_FREQ_CONFIG1_RESTOR E	0x4A004E30		4	0x4A326104
67	RESTORE_CM1. CM_AUTOIDLE_DPLL_CORE_RESTORE	0x4A004E34		4	0x4A326108
68	RESTORE_CM1. CM_MPU_CLKSTCTRL_RESTORE	0x4A004E38		4	0x4A32610C



**Table 16-19. SAR\_RAM\_1 Memory Mapping (continued)**

Order	Register Name	Register Start Addr	Register End Addr	Size	SAR_RAM_1 Addr
69	RESTORE_CM1. CM_CM1_PROFILING_CLKCTRL_RESTO RE	0x4A004E3C		4	0x4A3260110
70	RESTORE_CM1. CM_DYN_DEP_PRESCAL_RESTORE	0x4A004E40		4	0x4A326114
71	RESTORE_CM2. CM_L3_1_CLKSTCTRL_RESTORE	0x4A009E00		4	0x4A326118
72	RESTORE_CM2. CM_L3_2_CLKSTCTRL_RESTORE	0x4A009E04		4	0x4A32611C
73	RESTORE_CM2. CM_L4CFG_CLKSTCTRL_RESTORE	0x4A009E08		4	0x4A326120
74	RESTORE_CM2. CM_MEMIF_CLKSTCTRL_RESTORE	0x4A009E0C		4	0x4A326124
75	RESTORE_CM2. CM_L4PER_CLKSTCTRL_RESTORE	0x4A009E10		4	0x4A326128
76	RESTORE_CM2. CM_L3INIT_CLKSTCTRL_RESTORE	0x4A009E14		4	0x4A32612C
77	RESTORE_CM2. CM_L3INSTR_L3_3_CLKCTRL_RESTORE	0x4A009E18		4	0x4A326130
78	RESTORE_CM2. CM_L3INSTR_L3_INSTR_CLKCTRL_REST ORE	0x4A009E1C		4	0x4A326134
79	RESTORE_CM2. CM_L3INSTR_INTRCONN_WP1_CLKCTRL _RESTORE	0x4A009E20		4	0x4A326138
80	RESTORE_CM2. CM_CM2_PROFILING_CLKCTRL_RESTO RE	0x4A009E24		4	0x4A32613C
81	RESTORE_CM2. CM_C2C_STATICDEP_RESTORE	0x4A009E28		4	0x4A326140
82	RESTORE_CM2. CM_L3_1_DYNAMICDEP_RESTORE	0x4A009E2C		4	0x4A326144
83	RESTORE_CM2. CM_L3_2_DYNAMICDEP_RESTORE	0x4A009E30		4	0x4A326148
84	RESTORE_CM2. CM_C2C_DYNAMICDEP_RESTORE	0x4A009E34		4	0x4A32614C
85	RESTORE_CM2. CM_L4CFG_DYNAMICDEP_RESTORE	0x4A009E38		4	0x4A326150
86	RESTORE_CM2. CM_L4PER_DYNAMICDEP_RESTORE	0x4A009E3C		4	0x4A326154
87	C2C. C2C_PORTCONFIG	0x5C00000C		4	0x4A326158
88	C2C. C2C_MIRRORMODE	0x5C000010		4	0x4A32615C
89	C2C. C2C_IRQ_ENABLE_SET_1	0x5C000028		4	0x4A326160
90	C2C. C2C_FCLK_FREQ	0x5C000040		4	0x4A326164
91	C2C. C2C_RX_MAX_FREQ	0x5C000044		4	0x4A326168
92	C2C. C2C_GENI_CONTROL	0x5C000070		4	0x4A32616C
93	C2C. C2C_GENI_MASK	0x5C000074		4	0x4A326170
94	C2C.C2C_GENO_INTERRUPT	0x5C000084		4	0x4A326174
95	C2C.C2C_GENO_LEVEL	0x5C000088		4	0x4A326178
97	SYSCTRL_PADCONF_CORE.CONTROL_P ADCONF_GLOBAL	0x4A1005A0	0x4A1005DB	60	0x4A32617C
98	SYSCTRL_PADCONF_CORE.CONTROL_P BIASLITE	0x4A100600	0x4A100637	56	0x4A3261B8
99	SYSCTRL_PADCONF_CORE.CONTROL_L PDDR2IO1_0	0x4A100638	0x4A100657	32	0x4A3261F0

**Table 16-19. SAR\_RAM\_1 Memory Mapping (continued)**

Order	Register Name	Register Start Addr	Register End Addr	Size	SAR_RAM_1 Addr
100	SYSCTRL_PADCONF_CORE.CONTROL_B US_HOLD	0x4A100658		4	0x4A326210
101	SYSCTRL_PADCONF_CORE.CONTROL_C 2C	0x4A10065C		4	0x4A326214
102	SYSCTRL_PADCONF_CORE.CONTROL_E FUSE_1	0x4A100700		4	0x4A326218
103	SYSCTRL_PADCONF_CORE.CONTROL_E FUSE_2	0x4A100704		4	0x4A32621C
104	SYSCTRL_PADCONF_CORE.CONTROL_E FUSE_4	0x4A10070C		4	0x4A326220
239	DMM.DMM_LISA_MAP_0	0x4E000040		4	0x4A326224
240	DMM.DMM_LISA_MAP_1	0x4E000044		4	0x4A326228
241	DMM.DMM_LISA_MAP_2	0x4E000048		4	0x4A32622C
242	DMM.DMM_LISA_MAP_3	0x4E00004C		4	0x4A326230
243	DMM.DMM_LISA_LOCK	0x4E00001C		4	0x4A326234
244	DMM.DMM_TILER_OR0	0x4E000220		4	0x4A326238
245	DMM.DMM_TILER_OR1	0x4E000224		4	0x4A32623C
246	DMM.DMM_PAT_VIEW0	0x4E000420		4	0x4A326240
247	DMM.DMM_PAT_VIEW1	0x4E000424		4	0x4A326244
248	DMM.DMM_PAT_VIEW_MAP_0	0x4E000440		4	0x4A326248
249	DMM.DMM_PAT_VIEW_MAP_1	0x4E000444		4	0x4A32624C
250	DMM.DMM_PAT_VIEW_MAP_2	0x4E000448		4	0x4A326250
251	DMM.DMM_PAT_VIEW_MAP_3	0x4E00044C		4	0x4A326254
252	DMM.DMM_PAT_VIEW_MAP_BASE	0x4E000460		4	0x4A326258
253	DMM.DMM_PAT_IRQENABLE_SET	0x4A32625C		4	0x4A32625C
254	DMM.DMM_PAT_DESCR_0	0x4E000500		4	0x4A326260
255	DMM.DMM_PAT_AREA_0	0x4E000504		4	0x4A326264
256	DMM.DMM_PAT_CTRL_0	0x4E000508		4	0x4A326268
257	DMM.DMM_PAT_DATA_0	0x4E00050C		4	0x4A32626C
258	DMM.DMM_PAT_DESCR_1	0x4E000510		4	0x4A326270
259	DMM.DMM_PAT_AREA_1	0x4E000514		4	0x4A326274
260	DMM.DMM_PAT_CTRL_1	0x4E000518		4	0x4A326278
261	DMM.DMM_PAT_DATA_1	0x4E00051C		4	0x4A32627C
262	DMM.DMM_PAT_DESCR_2	0x4E000520		4	0x4A326280
263	DMM.DMM_PAT_AREA_2	0x4E000524		4	0x4A326284
264	DMM.DMM_PAT_CTRL_2	0x4E000528		4	0x4A326288
265	DMM.DMM_PAT_DATA_2	0x4E00052C		4	0x4A32628C
266	DMM.DMM_PAT_DESCR_3	0x4E000530		4	0x4A326290
267	DMM.DMM_PAT_AREA_3	0x4E000534		4	0x4A326294
268	DMM.DMM_PAT_CTRL_3	0x4E000538		4	0x4A326298
269	DMM.DMM_PAT_DATA_3	0x4E00053C		4	0x4A32629C
270	DMM.DMM_PEG_PRIO0	0x4E000460		4	0x4A3262A0
271	DMM.DMM_PEG_PRIO1	0x4E000464		4	0x4A3262A4
272	DMM.DMM_PEG_PRIO_PAT	0x4E000640		4	0x4A3262A8
273	CLK1_FLAGMUX_CLK1.L3_FLAGMUX_MA SK0	0x44000508		4	0x4A3262AC
274	CLK1_FLAGMUX_CLK1.L3_FLAGMUX_MA SK1	0x44000510		4	0x4A3262B0

**Table 16-19. SAR\_RAM\_1 Memory Mapping (continued)**

Order	Register Name	Register Start Addr	Register End Addr	Size	SAR_RAM_1 Addr
275	CLK1_DSS_BW_REGULATOR.L3_BW_R_BANDWIDTH	0x44000708		4	0x4A3262B4
276	CLK1_DSS_BW_REGULATOR.L3_BW_R_WATERMARK	0x4400070C		4	0x4A3262B8
277	CLK1_RATE_ADAPT_RESP_32TO128_CLK1.L3_RA_CNF	0x44000808		4	0x4A3262BC
278	CLK2_FLAGMUX_CLK2.L3_FLAGMUX_MASK0	0x44801008		4	0x4A3262C0
279	CLK2_FLAGMUX_CLK2.L3_FLAGMUX_MASK1	0x44801010		4	0x4A3262C4
280	CLK2_RATE_ADAPT_RESP_32TO128_CLK2.L3_RA_CNF	0x44801208		4	0x4A3262C8
281	CLK2_ISS_BW_REGULATOR.L3_BW_R_BANDWIDTH	0x44801308		4	0x4A3262CC
282	CLK2_ISS_BW_REGULATOR.L3_BW_R_WATERMARK	0x4480130C		4	0x4A3262D0
284	CLK2_IVAHD_BW_REGULATOR.L3_BW_R_BANDWIDTH	0x44801408		4	0x4A3262D4
284	CCLK2_IVAHD_BW_REGULATOR.L3_BW_R_WATERMARK	0x4480140C		4	0x4A3262D8
285	CLK2_SGX_BW_REGULATOR.L3_BW_R_BANDWIDTH	0x44801508		4	0x4A3262DC
286	CLK2_SGX_BW_REGULATOR.L3_BW_R_WATERMARK	0x4480150C		4	0x4A3262E0
287	CLK3_FLAGMUX_CLK3.L3_FLAGMUX_MASK0	0x45000208		4	0x4A3262E4
288	CLK3_FLAGMUX_CLK3.L3_FLAGMUX_MASK1	0x45000210		4	0x4A3262E8
292	RESTORE_CM2.CM_L3INIT_USB_HOST_CLKCTRL_RESTORE	0x4A009E54		4	0x4A3262EC
293	RESTORE_CM2.CM_L3INIT_USB_TLL_CLKCTRL_RESTORE	0x4A009E58		4	0x4A3262F0
294	RESTORE_CM2.CM_SDMA_STATICDEP_RESTORE	0x4A009E5C		4	0x4A3262F4
296	USBTLLHS_config.USBTLL_SAR_CNTX_0	0x4A062400		4	0x4A3262F8
297	USBTLLHS_config.USBTLL_SAR_CNTX_1	0x4A062404		4	0x4A3262FC
298	USBTLLHS_config.USBTLL_SAR_CNTX_2	0x4A062408		4	0x4A326300
299	USBTLLHS_config.USBTLL_SAR_CNTX_3	0x4A06240C		4	0x4A326304
300	USBTLLHS_config.USBTLL_SAR_CNTX_4	0x4A062410		4	0x4A326308
301	USBTLLHS_config.USBTLL_SAR_CNTX_5	0x4A062414		4	0x4A32630C
302	USBTLLHS_config.USBTLL_SAR_CNTX_6	0x4A062418		4	0x4A326310
303	HSUSBHOST.UHH_SYSCONFIG	0x4A064010		4	0x4A326310
304	HSUSBHOST.UHH_HOSTCONFIG	0x4A064040		4	0x4A326310
305	HSUSBHOST.UHH_SAR_CNTX_0	0x4A064100	0x4A0646FF	1536	0x4A32631C

Table 16-20 shows the SAR\_RAM\_2 memory mapping:

**Table 16-20. SAR\_RAM\_2 Memory Mapping**

Order	Register Name	Register Start Address	Register End Address	Size	SAR RAM Address
96	SYSCTRL_PADCONF_CORE.CONTROL_CORE_PAD0_GPMC_AD0_PAD1_GPMC_AD1	0x4A100040	0x4A1001D7	408	0x4A327000

Table 16-21 shows the SAR\_RAM\_3 memory mapping:

**Table 16-21. SAR\_RAM\_3 memory mapping**

Order	Register Name	Register Start Address	Register End Address	Size	SAR RAM Address
138	PER_AP.L4_AP_PROT_GROUP_MEMBER_S_3_L	0x48000218		4	0x4A32826C
139	PER_AP.L4_AP_PROT_GROUP_MEMBER_S_4_L	0x48000220		4	0x4A328270
140	PER_AP.L4_AP_PROT_GROUP_MEMBER_S_5_L	0x48000228		4	0x4A328274
141	PER_AP.L4_AP_PROT_GROUP_MEMBER_S_6_L	0x48000230		4	0x4A328278
142	PER_AP.L4_AP_PROT_GROUP_MEMBER_S_7_L	0x48000238		4	0x4A32827C
143	PER_AP.L4_AP_PROT_GROUP_ROLES_3	0x48000298	0x4800029F	8	0x4A328280
144	PER_AP.L4_AP_PROT_GROUP_ROLES_4	0x480002A0	0x480002A7	8	0x4A328288
145	PER_AP.L4_AP_PROT_GROUP_ROLES_5	0x480002A8	0x480002AF	8	0x4A328290
146	PER_AP.L4_AP_PROT_GROUP_ROLES_6	0x480002B0	0x480002B7	8	0x4A328298
147	PER_AP.L4_AP_PROT_GROUP_ROLES_7	0x480002B8	0x480002BF	8	0x4A3282A0
148	PER_AP.L4_AP_REGION_0_L	0x48000304		4	0x4A3282A8
149	PER_AP.L4_AP_REGION_3_L	0x4800031C		4	0x4A3282AC
150	PER_AP.L4_AP_REGION_5_L	0x4800032C		4	0x4A3282B0
151	PER_AP.L4_AP_REGION_7_L	0x4800033C		4	0x4A3282B4
152	PER_AP.L4_AP_REGION_9_L	0x4800034C		4	0x4A3282B8
153	PER_AP.L4_AP_REGION_11_L	0x4800035C		4	0x4A3282BC
154	PER_AP.L4_AP_REGION_13_L	0x4800036C		4	0x4A3282C0
155	PER_AP.L4_AP_REGION_15_L	0x4800037C		4	0x4A3282C4
156	PER_AP.L4_AP_REGION_17_L	0x4800038C		4	0x4A3282C8
157	PER_AP.L4_AP_REGION_19_L	0x4800039C		4	0x4A3282CC
158	PER_AP.L4_AP_REGION_21_L	0x480003AC		4	0x4A3282D0
159	PER_AP.L4_AP_REGION_23_L	0x480003BC		4	0x4A3282D4
160	PER_AP.L4_AP_REGION_25_L	0x480003CC		4	0x4A3282D8
161	PER_AP.L4_AP_REGION_26_L	0x480003D4		4	0x4A3282DC
162	PER_AP.L4_AP_REGION_28_L	0x480003E4		4	0x4A3282E0
163	PER_AP.L4_AP_REGION_30_L	0x480003F4		4	0x4A3282E4
164	PER_AP.L4_AP_REGION_32_L	0x48000404		4	0x4A3282E8
165	PER_AP.L4_AP_REGION_34_L	0x48000414		4	0x4A3282EC
166	PER_AP.L4_AP_REGION_37_L	0x4800042C		4	0x4A3282F0
167	PER_AP.L4_AP_REGION_39_L	0x4800043C		4	0x4A3282F4
168	PER_AP.L4_AP_REGION_41_L	0x4800044C		4	0x4A3282F8
169	PER_AP.L4_AP_REGION_43_L	0x4800045C		4	0x4A3282FC
170	PER_AP.L4_AP_REGION_45_L	0x4800046C		4	0x4A328300
171	PER_AP.L4_AP_REGION_47_L	0x4800047C		4	0x4A328304
172	PER_AP.L4_AP_REGION_49_L	0x4800048C		4	0x4A328308
173	PER_AP.L4_AP_REGION_51_L	0x4800049C		4	0x4A32830C
174	PER_AP.L4_AP_REGION_53_L	0x480004AC		4	0x4A328310
175	PER_AP.L4_AP_REGION_55_L	0x480004BC		4	0x4A328314

**Table 16-21. SAR\_RAM\_3 memory mapping (continued)**

Order	Register Name	Register Start Address	Register End Address	Size	SAR RAM Address
176	PER_AP.L4_AP_REGION_57_L	0x480004CC		4	0x4A328318
177	PER_AP.L4_AP_REGION_59_L	0x480004DC		4	0x4A32831C
178	PER_AP.L4_AP_REGION_61_L	0x480004EC		4	0x4A328320
179	PER_AP.L4_AP_REGION_63_L	0x480004FC		4	0x4A328324
180	PER_AP.L4_AP_REGION_65_L	0x4800050C		4	0x4A328328
181	PER_AP.L4_AP_REGION_67_L	0x4800051C		4	0x4A32832C
182	PER_AP.L4_AP_REGION_69_L	0x4800052C		4	0x4A328330
183	PER_AP.L4_AP_REGION_71_L	0x4800053C		4	0x4A328334
184	PER_AP.L4_AP_REGION_73_L	0x4800054C		4	0x4A328338
185	PER_AP.L4_AP_REGION_75_L	0x4800055C		4	0x4A32833C
186	PER_AP.L4_AP_REGION_77_L	0x4800056C		4	0x4A328340
187	PER_AP.L4_AP_REGION_79_L	0x4800057C		4	0x4A328344
188	PER_AP.L4_AP_REGION_84_L	0x480005A4		4	0x4A328348
189	CFG_AP.L4_AP_PROT_GROUP_MEMBE RS_6_L	0x4A000230		4	0x4A32834C
190	CFG_AP.L4_AP_PROT_GROUP_MEMBE RS_7_L	0x4A000238		4	0x4A328350
191	CFG_AP.L4_AP_PROT_GROUP_ROLES_ 6	0x4A0002B0	0x4A0002B7	8	0x4A328354
192	CFG_AP.L4_AP_PROT_GROUP_ROLES_ 7	0x4A0002B8	0x4A0002BF	8	0x4A32835C
193	CFG_AP.L4_AP_REGION_0_L	0x4A000304		4	0x4A328360
194	CFG_AP.L4_AP_REGION_3_L	0x4A00031C		4	0x4A328364
195	CFG_AP.L4_AP_REGION_5_L	0x4A00032C		4	0x4A328368
196	CFG_AP.L4_AP_REGION_7_L	0x4A00033C		4	0x4A32836C
197	CFG_AP.L4_AP_REGION_10_L	0x4A000354		4	0x4A328370
198	CFG_AP.L4_AP_REGION_11_L	0x4A00035C		4	0x4A328374
199	CFG_AP.L4_AP_REGION_13_L	0x4A00036C		4	0x4A328378
200	CFG_AP.L4_AP_REGION_15_L	0x4A00037C		4	0x4A32837C
201	CFG_AP.L4_AP_REGION_17_L	0x4A00038C		4	0x4A3283
202	CFG_AP.L4_AP_REGION_21_L	0x4A0003AC		4	0x4A3282A8
203	CFG_AP.L4_AP_REGION_23_L	0x4A0003BC		4	0x4A3282A8
204	CFG_AP.L4_AP_REGION_25_L	0x4A0003CC		4	0x4A3282A8
205	CFG_AP.L4_AP_REGION_27_L	0x4A0003DC		4	0x4A3282A8
206	CFG_AP.L4_AP_REGION_29_L	0x4A0003EC		4	0x4A3282A8
207	CFG_AP.L4_AP_REGION_31_L	0x4A0003FC		4	0x4A3282A8
208	CFG_AP.L4_AP_REGION_33_L	0x4A00040C		4	0x4A3282A8
209	CFG_AP.L4_AP_REGION_35_L	0x4A00041C		4	0x4A3282A8
210	CFG_AP.L4_AP_REGION_37_L	0x4A00042C		4	0x4A3282A8
211	CFG_AP.L4_AP_REGION_39_L	0x4A00043C		4	0x4A3282A8
212	CFG_AP.L4_AP_REGION_41_L	0x4A00044C		4	0x4A3282A8
213	CFG_AP.L4_AP_REGION_43_L	0x4A00045C		4	0x4A3282A8
214	CFG_AP.L4_AP_REGION_45_L	0x4A00046C		4	0x4A3282A8
215	CFG_AP.L4_AP_REGION_47_L	0x4A00047C		4	0x4A3282A8
216	CFG_AP.L4_AP_REGION_49_L	0x4A00048C		4	0x4A3282A8
217	CFG_AP.L4_AP_REGION_51_L	0x4A00049C		4	0x4A3282A8
218	CFG_AP.L4_AP_REGION_53_L	0x4A0004AC		4	0x4A3282A8
219	CFG_AP.L4_AP_REGION_55_L	0x4A0004BC		4	0x4A3282A8

**Table 16-21. SAR\_RAM\_3 memory mapping (continued)**

Order	Register Name	Register Start Address	Register End Address	Size	SAR RAM Address
220	CFG_AP.L4_AP_REGION_57_L	0x4A0004CC		4	0x4A3282A8
221	CFG_AP.L4_AP_REGION_59_L	0x4A0004DC		4	0x4A3282A8
222	CFG_AP.L4_AP_REGION_61_L	0x4A0004EC		4	0x4A3282A8
223	CFG_AP.L4_AP_REGION_63_L	0x4A0004FC		4	0x4A3282A8
224	CFG_AP.L4_AP_REGION_65_L	0x4A00050C		4	0x4A3282A8
225	CFG_AP.L4_AP_REGION_67_L	0x4A00051C		4	0x4A3282A8
226	CFG_AP.L4_AP_REGION_69_L	0x4A00052C		4	0x4A3282A8
227	CFG_AP.L4_AP_REGION_71_L	0x4A00053C		4	0x4A3282A8
228	CFG_AP.L4_AP_REGION_73_L	0x4A00054C		4	0x4A3282A8
229	CFG_AP.L4_AP_REGION_75_L	0x4A00055C		4	0x4A3282A8
230	CFG_AP.L4_AP_REGION_77_L	0x4A00056C		4	0x4A3282A8
231	CFG_AP.L4_AP_REGION_78_L	0x4A000574		4	0x4A3282A8
232	CFG_AP.L4_AP_REGION_80_L	0x4A000584		4	0x4A3282A8
233	CFG_AP.L4_AP_REGION_82_L	0x4A000594		4	0x4A3282A8
234	CFG_AP.L4_AP_REGION_84_L	0x4A0005A4		4	0x4A3282A8
235	CFG_AP.L4_AP_REGION_86_L	0x4A0005B4		4	0x4A3282A8
236	CFG_AP.L4_AP_REGION_88_L	0x4A0005C4		4	0x4A3282A8
237	CFG_AP.L4_AP_REGION_90_L	0x4A0005D4		4	0x4A3282A8
238	CFG_AP.L4_AP_REGION_91_L	0x4A0005DC		4	0x4A3282A8

## 16.5 sDMA Basic Programming Model

### 16.5.1 Setup Configuration

After a hardware reset, program all fields in the logical channel registers to default values for any channels used, because most fields are undefined following reset.

Before programming any DMA transfers, the priority arbitration rate and the maximum FIFO depth must be configured through the [DMA4\\_GCR](#) register, and any required interrupts must be enabled through the [DMA4\\_IRQENABLE\\_Lj](#) registers and the logical channel [DMA4\\_CICRi](#) registers.

Software clears the [DMA4\\_CSRi](#) register and the IRQSTATUS bit for the different interrupt lines before enabling the channel.

### 16.5.2 Software-Triggered (Nonsynchronized) Transfer

To program a software-triggered DMA transfer:

1. Configure the transfer parameters in the logical DMA channel registers:

- [DMA4\\_CSDPi](#):
  - Transfer ES (8 bits, 16 bits, or 32 bits) in DMA register bits [DMA4\\_CSDPi\[1:0\]](#)
  - Read and write port access types (single/burst), DMA register bits [DMA4\\_CSDPi\[8:7\]](#) and [DMA4\\_CSDPi\[15:14\]](#)
  - Source and destination endianism, DMA register bits [DMA4\\_CSDPi\[21\]](#) and [DMA4\\_CSDPi\[19\]](#)
  - Write mode (posted or nonposted) and DMA register bits [DMA4\\_CSDPi\[17:16\]](#)
  - Source or destination packed or nonpacked (if the ES is less than the read/write port size), DMA register bits [DMA4\\_CSDPi\[6\]](#) and [DMA4\\_CSDPi\[13\]](#)
- [DMA4\\_CENi](#): EN
- [DMA4\\_CFNi](#): FN per transfer block
- [DMA4\\_CSSAi](#) and [DMA4\\_CDSAi](#): Source and destination start address (aligned with transfer ES)
- [DMA4\\_CCRi](#):
  - Read and write port addressing modes, DMA register bits [DMA4\\_CCRi\[13:12\]](#) and

- DMA4\_CCRi[15:14]
    - Priority bit for both read and write ports, DMA register bits DMA4\_CCRi[6] and DMA4\_CCRi[26]
    - DMA request number (set to 0 for a software-triggered transfer) and DMA register bits DMA4\_CCRi[4:0] = 0 and DMA4\_CCRi[20:19] = 0
  - DMA4\_CSEIi, DMA4\_CSFli, DMA4\_CDEIi, and DMA4\_CDFIi: Source and destination element and frame indexes (depending on addressing mode)
2. Start the transfer through the enable bit in the channel DMA4\_CCRi register and DMA register bit DMA4\_CCRi[7]

The example below perform a DMA transfer on channel 10 of a 240\*160 picture from RAM to RAM (0x80C00000 to 0x80F00000) :

```

UWORD32 RegVal = 0;
DMA4_t *DMA4;
DMA4 = (DMA4_t *)malloc(sizeof(DMA4_t));
/* Init. parameters */
DMA4->DataType = 0x2; // DMA4_CSDPi[1:0]
DMA4->ReadPortAccessType = 0; // DMA4_CSDPi[8:7]
DMA4->WritePortAccessType = 0; // DMA4_CSDPi[15:14]
DMA4->SourceEndiansim = 0; // DMA4_CSDPi[21]
DMA4->DestinationEndianism = 0; // DMA4_CSDPi[19]
DMA4->WriteMode = 0; // DMA4_CSDPi[17:16]
DMA4->SourcePacked = 0; // DMA4_CSDPi[6]
DMA4->DestinationPacked = 0; // DMA4_CSDPi[13]
DMA4->NumberOfElementPerFrame = 240; // DMA4_CENi
DMA4->NumberOfFramePerTransferBlock = 160; // DMA4_CFNi
DMA4->SourceStartAddress = 0x80C00000; // DMA4_CSSAi
DMA4->DestinationStartAddress = 0x80F00000; // DMA4_CDSAi
DMA4->SourceElementIndex = 1; // DMA4_CSEIi
DMA4->SourceFrameIndex = 1; // DMA4_CSFli
DMA4->DestinationElementIndex = 1; // DMA4_CDEIi
DMA4->DestinationFrameIndex = 1; // DMA4_CDFIi
DMA4->ReadPortAccessMode = 1; // DMA4_CCRi[13:12]
DMA4->WritePortAccessMode = 1; // DMA4_CCRi[15:14]
DMA4->ReadPriority = 0; // DMA4_CCRi[6]
DMA4->WritePriority = 0; // DMA4_CCRi[23]
DMA4->ReadRequestNumber = 0; // DMA4_CCRi[4:0]
DMA4->WriteRequestNumber = 0; // DMA4_CCRi[20:19]
/* 1) Configure the transfer parameters in the logical DMA registers */
/*-----*/
/* a) Set the data type CSDP[1:0], the Read/Write Port access type CSDP[8:7]/[15:14], the
Source/dest endiansim CSDP[21]/CSDP[19], write mode CSDP[17:16], source/dest packed or non-packed
CSDP[6]/CSDP[13]*/

// Read CSDP
RegVal = DMA4_CSDP_CH10;
// Build reg
RegVal = ((RegVal & ~ 0x3) | DMA4->DataType );
RegVal = ((RegVal & ~(0x3 << 7)) | (DMA4->ReadPortAccessType << 7));
RegVal = ((RegVal & ~(0x3 << 14)) | (DMA4->WritePortAccessType << 14));
RegVal = ((RegVal & ~(0x1 << 21)) | (DMA4->SourceEndiansim << 21));
RegVal = ((RegVal & ~(0x1 << 19)) | (DMA4->DestinationEndianism << 19));
RegVal = ((RegVal & ~(0x3 << 16)) | (DMA4->WriteMode << 16));
RegVal = ((RegVal & ~(0x1 << 6)) | (DMA4->SourcePacked << 6));
RegVal = ((RegVal & ~(0x1 << 13)) | (DMA4->DestinationPacked << 13));
// Write CSDP
DMA4_CSDP_CH10 = RegVal;
/* b) Set the number of element per frame CEN[23:0]*/
DMA4_CEN_CH10 = DMA4->NumberOfElementPerFrame;
/* c) Set the number of frame per block CFN[15:0]*/
DMA4_CFN_CH10 = DMA4->NumberOfFramePerTransferBlock;
/* d) Set the Source/dest start address index CSSA[31:0]/CDSA[31:0]*/
DMA4_CSSA_CH10 = DMA4->SourceStartAddress; // address start
DMA4_CDSA_CH10 = DMA4->DestinationStartAddress; // address dest
/* e) Set the Read Port addressing mode CCR[13:12], the Write Port addressing mode CCR[15:14],
read/write priority CCR[6]/CCR[26], the current LCH CCR[20:19]=00 and CCR[4:0]=00000*/
// Read CCR
RegVal = DMA4_CCR_CH10;
// Build reg
RegVal = ((RegVal & ~(0x3 << 12)) | (DMA4->ReadPortAccessMode << 12));
RegVal = ((RegVal & ~(0x3 << 14)) | (DMA4->WritePortAccessMode << 14));
RegVal = ((RegVal & ~(0x1 << 6)) | (DMA4->ReadPriority << 6));
    
```

```

RegVal = ((RegVal & ~(0x1 << 26)) | (DMA4->WritePriority << 26));
RegVal&= 0xFFCFFFE0 ;
// Write CCR
DMA4_CCR_CH10 = RegVal;
/* f)- Set the source element index CSEI[15:0]*/
DMA4_CSEI_CH10 = DMA4->SourceElementIndex;
/* g)- Set the source frame index CSFI[15:0]*/
DMA4_CSFI_CH10 = DMA4->SourceFrameIndex ;
/* h)- Set the destination element index CDEI[15:0]*/
DMA4_CDEI_CH10 = DMA4->DestinationElementIndex;
/* i)- Set the destination frame index CDFI[31:0]*/
DMA4_CDFI_CH10 = DMA4->DestinationFrameIndex;
/* 2) Start the DMA transfer by Setting the enable bit CCR[7]=1 */
/*-----*/
//write enable bit
DMA4_CCR_CH10 |= 1 << 7; /* start */

```

### 16.5.3 Hardware-Synchronized Transfer

To monitor a hardware synchronized DMA transfer, initialize the [DMA4\\_CDACi](#) register before the software enable.

To configure an LCh to synchronize by element, packet, frame, or block, the frame synchronization [DMA4\\_CCRi\[5\]](#) FS bit and the block synchronization [DMA4\\_CCRi\[18\]](#) BS bit register must be programmed. For all the following synchronized transfers (element, packet, frame or block synchronized transfers) User must set first : [DMA4\\_CCRi\[24\]](#) SEL\_SRC\_DST\_SYNC to 1 when the source triggers on the DMA request and [DMA4\\_CCRi\[24\]](#) SEL\_SRC\_DST\_SYNC to 0 when the Destination triggers on the DMA request. Note: User must take care when setting the [DMA4\\_CCRi\[23\]](#) PREFETCH bit it is in conjunction with [DMA4\\_CCRi\[24\]](#) SEL\_SRC\_DST\_SYNC bit.

- To configure an LCh to transfer one element per DMA request:
  1. Set the number of DMA request associated to the current LCH in the [DMA4\\_CCRi\[20:19\]](#) SYNCHRO\_CONTROL\_UPPER and [DMA4\\_CCRi\[4:0\]](#) SYNCHRO bit field.
  2. Set the data type, also referenced as element size (ES), in the [DMA4\\_CSDPi\[1:0\]](#) DATA\_TYPE bit field.
  3. Set the Read Port access type (single or burst access) in the [DMA4\\_CSDPi\[8:7\]](#) SRC\_BURST\_EN bit field.
  4. Set the Write Port access type (single or burst access) in the [DMA4\\_CSDPi\[15:14\]](#) DST\_BURST\_EN bit field.
  5. Set the Read Port addressing mode in the [DMA4\\_CCRi\[13:12\]](#) SRC\_AMODE bit field.
  6. Set the Write Port addressing mode in the [DMA4\\_CCRi\[15:14\]](#) DST\_AMODE bit field.
  7. Set the Read start address in the [DMA4\\_CSSAi\[31:0\]](#) SRC\_START\_ADRS bit field.
  8. Set the Write start address in the [DMA4\\_CDSAi\[31:0\]](#) DST\_START\_ADRS bit field.
  9. Set both FS and BS to 0 in [DMA4\\_CCRi\[5\]](#) FS and [DMA4\\_CCRi\[18\]](#) BS.
  10. Set to 1 the channel enable bit [DMA4\\_CCRi\[7\]](#) EN.
- To configure an LCh to transfer one frame per DMA request:
  1. Set the number of DMA request associated to the current LCH in the [DMA4\\_CCRi\[20:19\]](#) SYNCHRO\_CONTROL\_UPPER and [DMA4\\_CCRi\[4:0\]](#) SYNCHRO bit field.
  2. Set the data type, also referenced as element size (ES), in the [DMA4\\_CSDPi\[1:0\]](#) DATA\_TYPE bit field.
  3. Set the number of element per frame in the [DMA4\\_CENi\[23:0\]](#) CHANNEL\_ELMNT\_NBR bit field.
  4. Set the Read Port access type (single or burst access) in the [DMA4\\_CSDPi\[8:7\]](#) SRC\_BURST\_EN bit field.
  5. Set the Write Port access type (single or burst access) in the [DMA4\\_CSDPi\[15:14\]](#) DST\_BURST\_EN bit field.
  6. Set the Read Port addressing mode in the [DMA4\\_CCRi\[13:12\]](#) SRC\_AMODE bit field.
  7. Set the Write Port addressing mode in the [DMA4\\_CCRi\[15:14\]](#) DST\_AMODE bit field.
  8. Set the Read start address in the [DMA4\\_CSSAi\[31:0\]](#) SRC\_START\_ADRS bit field.
  9. Set the Write start address in the [DMA4\\_CDSAi\[31:0\]](#) DST\_START\_ADRS bit field.



10. Set FS to 1 and BS to 0, respectively, in `DMA4_CCRi[5]` FS and `DMA4_CCRi[18]` BS.
11. Set to 1 the channel enable bit `DMA4_CCRi[7]` EN.
- To configure an LCh to transfer one block per DMA request:
  1. Set the number of DMA request associated to the current LCH in the `DMA4_CCRi[20:19]` SYNCHRO\_CONTROL\_UPPER and `DMA4_CCRi[4:0]` SYNCHRO bit field.
  2. Set the data type, also referenced as element size (ES), in the `DMA4_CSDPi[1:0]` DATA\_TYPE bit field.
  3. Set the number of element per frame in the `DMA4_CENi[23:0]` CHANNEL\_ELMNT\_NBR bit field.
  4. Set in the `DMA4_CFNi[15:0]` CHANNEL\_FRAME\_NBR bit field the number of frame (transfers), to take place before the LCH gets disabled.
  5. Set the Read Port access type (single or burst access) in the `DMA4_CSDPi[8:7]` SRC\_BURST\_EN bit field.
  6. Set the Write Port access type (single or burst access) in the `DMA4_CSDPi[15:14]` DST\_BURST\_EN bit field.
  7. Set the Read Port addressing mode in the `DMA4_CCRi[13:12]` SRC\_AMODE bit field.
  8. Set the Write Port addressing mode in the `DMA4_CCRi[15:14]` DST\_AMODE bit field.
  9. Set the Read start address in the `DMA4_CSSAi[31:0]` SRC\_START\_ADRS bit field.
  10. Set the Write start address in the `DMA4_CDSAi[31:0]` DST\_START\_ADRS bit field.
  11. Set FS to 0 and BS to 1, respectively, in `DMA4_CCRi[5]` FS and `DMA4_CCRi[18]` BS.
  12. Set to 1 the channel enable bit `DMA4_CCRi[7]` EN.
- To configure an LCh to transfer one packet per DMA request:
  1. Set the number of DMA request associated to the current LCH in the `DMA4_CCRi[20:19]` SYNCHRO\_CONTROL\_UPPER and `DMA4_CCRi[4:0]` SYNCHRO bit field.
  2. Set the data type, also referenced as element size (ES), in the `DMA4_CSDPi[1:0]` DATA\_TYPE bit field.
  3. Set the number of elements per packet to transfer: If the packet requestor is in the source, set `DMA4_CCRi [24]` SEL\_SRC\_DST\_SYNC to 1 and set the packet element number in the `DMA4_CSFli` register and set the addressing mode of source to constant addressing in `DMA4_CCRi[13:12]` SRC\_AMODE bit field; else, if the packet requestor is in the destination, set the `DMA4_CCRi[24]` SEL\_SRC\_DST\_SYNC to 0 and set the packet element number in the `DMA4_CDFli` register and set the addressing mode of destination to constant addressing in `DMA4_CCRi[15:14]` DST\_AMODE bit field.
  4. Set the number of elements per frame in the `DMA4_CENi[23:0]` CHANNEL\_ELMNT\_NBR bit field.
  5. Set in the `DMA4_CFNi[15:0]` CHANNEL\_FRAME\_NBR bit field the number of frames (transfers), to take place before the LCH gets disabled.
  6. Set the element number in the packet in the `DMA4_CSFli[15:0]` PKT\_ELNT\_NBR, if constant addressing or post-incremented addressing modes are used in the source side. However, the number of element in the packet is set in the `DMA4_CDFli[15:0]` PKT\_ELNT\_NBR if constant addressing mode is used in the destination side.
  7. Set the Read Port access type (single or burst access) in the `DMA4_CSDPi[8:7]` SRC\_BURST\_EN bit field.
  8. Set the Write Port access type (single or burst access) in the `DMA4_CSDPi[15:14]` DST\_BURST\_EN bit field.
  9. Set the Read Port addressing mode in the `DMA4_CCRi[13:12]` SRC\_AMODE bit field.
  10. Set the Write Port addressing mode in the `DMA4_CCRi[15:14]` DST\_AMODE bit field.
  11. Set the Read start address in the `DMA4_CSSAi[31:0]` SRC\_START\_ADRS bit field.
  12. Set the Write start address in the `DMA4_CDSAi[31:0]` DST\_START\_ADRS bit field.
  13. Set FS to 1 and BS to 1, respectively, in `DMA4_CCRi[5]` FS and `DMA4_CCRi[18]` BS.
  14. Set to 1 the channel enable bit `DMA4_CCRi[7]` EN.

---

**NOTE:** It is possible to stop a transfer by disabling the channel by resetting the `DMA4_CCRi[7]` ENABLE bit.

---

### 16.5.4 Synchronized Transfer Monitoring Using CDAC

The `DMA4_CDACi` register is writable and non-initialized (value undefined). It can be initialized to monitor a transfer by applying the following programming model:

1. Write 0 in the `DMA4_CDACi` register.
2. Enable the channel.
3. If a time-out occurs, read `DMA4_CDACi` register.
4. If `DMA4_CDACi` != 0 (it is the value configured in `DMA4_CDACi`):
 

This indicates that the corresponding transfer has started. User can then rely on `DMA4_CCENi` and `DMA4_CCFNi` element and frame counters.

Else, if `DMA4_CDACi` = 0 (it is the value configured in the `DMA4_CDACi`):

This indicates that the corresponding transfer did not start.

### 16.5.5 Concurrent Software and Hardware Synchronization

This section describes thread allocation only; it does not describe the entire transfer. Because synchronized transfers are latency critical, you must allocate a thread at least on the synchronized target side.

Even for multiple concurrent channels, thread reservation ensures that when a hardware DMA request arrives, the read/write scheduler finds available thread(s) to initiate a channel schedule and issue a read/write transaction.

Consider six concurrent channels:

- Channels 0, 1, 2, and 3 are dedicated to memory-memory transfer; they are software triggered and not synchronized.
- Channel 4 is dedicated to memory-peripheral transfer, hardware triggered, and synchronized on the write side.
- Channel 5 is dedicated to peripheral-memory transfer, hardware triggered, and synchronized on the read side.

To perform thread reservation:

1. Allow thread reservation for priority channel 4 and channel 5:
 

Reserve one thread (Read ThreadID 0) on the read port and one thread (Write ThreadID 0) on the write port: set the `DMA4_GCR[13:12]` `HI_THREAD_RESERVED` bit field to 0x1.
2. Specify channel priority:
 

Channel 4 is a write high priority channel: set `DMA4_CCRi[26]` `WRITE_PRIORITY` = 1.

Channel 5 is a read high priority channel: set `DMA4_CCRi[6]` `READ_PRIORITY` = 1.

### 16.5.6 Chained Transfer

A chained DMA transfer can be programmed as follows:

1. Configure the transfer parameters for each logical DMA channel in the chain as in step 1 for either the synchronized or non-synchronized transfers describe in [Section 16.5.5, Concurrent Software and Hardware Synchronization](#).
2. For each channel in the chain, configure the `DMA4_CLNK_CTRLi` register as follows:
  - Next logical DMA channel number (for a looping chained transfer link last channel to first channel number), in the `DMA4_CLNK_CTRLi[4:0]` `NEXTLCH_ID` bit field.
  - Include the logical channel to the chain and enable link by setting the `DMA4_CLNK_CTRLi[15]` `ENABLE_LNK` bit.
  - For a non-looping chain, the last logical channel in the chain must have the `DMA4_CLNK_CTRLi[15]` `ENABLE_LNK` bit set to 0 to indicate the end of the chain.
3. Enable the transfer through the enable bit in the first logical channel `DMA4_CCRi[7]` `ENABLE` bit. All other channels in the chain must be disabled. Each channel is enabled automatically in turn when the previous logical channel transfer completes. A non-synchronized transfer starts immediately; a hardware-synchronized transfer starts when the DMA request line corresponding to the first DMA channel in the chain is asserted.

To stop a looping chained transfer, disable the `DMA4_CLNK_CTRLi[15] ENABLE_LNK` bit (by setting it to 0x0), of the final channel transfer.

In the RAM-to-RAM copy example, to copy in loop, it is possible to link channel 10 on itself. The following line can be added in the channel configuration:

```
/* g) Set link for loop */
DMA4_CLNK_CTRL_CH10 = 0x0000800A;
```

### 16.5.7 90-Degree Clockwise Image Rotation

The 90-degree clockwise image rotation example described in [Section 16.4.5, Addressing Modes](#), can be programmed as follows:

1. Configure the transfer parameters in the logical DMA channel registers:

- **DMA4\_CSDPi:**
  - Transfer ES = 32-bit (32 bpp), `DMA4_CSDPi[1:0] DATA_TYPE` bit field
  - Read and write port access types = maximum burst size supported by memory device, `DMA4_CSDPi[8:7] SRC_BURST_EN` and `DMA4_CSDPi[15:14] DST_BURST_EN` bit fields
  - Source and destination endianness, `DMA4_CSDPi[21] SRC_ENDIAN` and `DMA4_CSDPi[19] DST_ENDIAN` bits
  - Write mode = posted with last element nonposted, `DMA4_CSDPi[17:16] WRITE_MODE` bit field
  - Source and destination packed = Yes (although destination writes do not benefit because  $EI > 1$ ), `DMA4_CSDPi[6] SRC_PACKED` and `DMA4_CSDPi[13] DST_PACKED` bits
- **DMA4\_CENi:** EN = 240
- **DMA4\_CFNi:** FN per transfer block = 160
- **DMA4\_CSSAi:** Source start address = 0x100000
- **DMA4\_CDSAi:** destination start address = 0x20013E
- **DMA4\_CCRi:**
  - Read and write port addressing modes = double-index addressing mode for both or post-increment addressing on source and double-index addressing on destination, `DMA4_CCRi[13:12] SRC_AMODE` and `DMA4_CCRi[15:14] DST_AMODE` bit fields
  - Low or high priority, `DMA4_CCRi[6] READ_PRIORITY` bit
  - DMA request number = 0 (for software-triggered transfer), `DMA4_CCRi[4:0] SYNCHRO_CONTROL` and `DMA4_CCRi[20:19] SYNCHRO_CONTROL_UPPER` bit fields
- **DMA4\_CSEIi:** Source EI = 1
- **DMA4\_CSFII:** Source frame index = 1
- **DMA4\_CDEIi:** destination EI = 637
- **DMA4\_CDFII:** destination frame index = 152967

2. Start the transfer through the enable bit in the channel `DMA4_CCRi` register.

The following parameters are used to perform this rotation from 0x80C00000 RAM address to 0x80F00000, with the same code used in [Section 16.5.2, Software-Triggered \(Nonsynchronized\) Transfer](#).

```
/* Init. parameters */
DMA4->DataType = 0x2; // DMA4_CSDPi[1:0]
DMA4->ReadPortAccessType = 0x3; // DMA4_CSDPi[8:7]
DMA4->WritePortAccessType = 0x3; // DMA4_CSDPi[15:14]
DMA4->SourceEndiansim = 0; // DMA4_CSDPi[21]
DMA4->DestinationEndianness = 0; // DMA4_CSDPi[19]
DMA4->WriteMode = 0x2; // DMA4_CSDPi[17:16]
DMA4->SourcePacked = 0x1; // DMA4_CSDPi[6]
DMA4->DestinationPacked = 0x1; // DMA4_CSDPi[13]
DMA4->NumberOfElementPerFrame = 240; // DMA4_CENi
DMA4->NumberOfFramePerTransferBlock = 160; // DMA4_CFNi
DMA4->SourceStartAddress = 0x80C00000; // DMA4_CSSAi
DMA4->DestinationStartAddress = 0x80F00000; // DMA4_CDSAi
DMA4->SourceElementIndex = 1; // DMA4_CSEIi
DMA4->SourceFrameIndex = 1; // DMA4_CSFII
DMA4->DestinationElementIndex = 637; // DMA4_CDEIi
DMA4->DestinationFrameIndex = -152967; // DMA4_CDFII
DMA4->ReadPortAccessMode = 0x3; // DMA4_CCRi[13:12]
DMA4->WritePortAccessMode = 0x3; // DMA4_CCRi[15:14]
DMA4->ReadPriority = 0; // DMA4_CCRi[6]
```

```
DMA4->WritePriority = 0; // DMA4_CCRi[23]
DMA4->ReadRequestNumber = 0; // DMA4_CCRi[4:0]
DMA4->WriteRequestNumber = 0; // DMA4_CCRi[20:19]
```

### 16.5.8 Graphic Operations

- Transparent copy:
  1. Set the [DMA4\\_CCRi\[17\]](#) TRANSPARENT\_COPY\_ENABLE bit to 1
  2. Set the [DMA4\\_CCRi\[16\]](#) CONST\_FILL\_ENABLE bit to 0
  3. Set the value of the key color in the [DMA4\\_COLORi\[15:0\]](#) COLOR\_KEY bit field

To perform this graphic operation, the following lines can be added to the example of [Section 16.5.2, Software-Triggered \(Nonsynchronized\) Transfer](#).

```
DMA4_CCR_CH10 &= ~(0x1 << 16);
DMA4_CCR_CH10 |= 0x1 << 17;
DMA4_COLOR_CH10 = 0x00000003;
```

- Solid Color fill:
  1. Set the [DMA4\\_CCRi\[16\]](#) CONST\_FILL\_ENABLE bit to 1
  2. Set the [DMA4\\_CCRi\[17\]](#) TRANSPARENT\_COPY\_ENABLE bit to 0
  3. Set the value of key the color in the [DMA4\\_COLORi\[15:0\]](#) SOLID\_COLOR bit field

To perform this graphic operation, the following lines can be added to the example of [Section 16.5.2, Software-Triggered \(Nonsynchronized\) Transfer](#).

```
DMA4_CCR_CH10 &= ~(0x1 << 17);
DMA4_CCR_CH10 |= 0x1 << 16;
DMA4_COLOR_CH10 = 0x00000003;
```

### 16.5.9 Linked-List Programming Guidelines

- With the exception of the [DMA4\\_CCRi\[7\]](#) ENABLE bit and the [DMA4\\_CDPi\[7\]](#) PAUSE\_LINK\_LIST bit during a linked-list transfer (descriptor load phase or data load phase), avoid programming any register through the configuration port.
- Before enabling any linked-list transfer, ensure that all global registers and all registers in the descriptor are initialized. Some static channel registers (registers that are not updated by the descriptor to be loaded) must also be initialized correctly:
  - For type 2, the following registers must be initialized with consistent values:
    - All global registers
    - [DMA4\\_CCRi](#)
    - [DMA4\\_CSDPi](#)
    - [DMA4\\_CLNK\\_CTRLi](#)
  - For type 3, the following registers must be initialized with consistent values:
    - All global registers
    - [DMA4\\_CCRi](#)
    - [DMA4\\_CSDPi](#)
    - [DMA4\\_CLNK\\_CTRLi](#)
    - [DMA4\\_CICRi](#)
    - [DMA4\\_CFNi](#)
- In case of a linked list with descriptor types 2 and 3, the content of the [DMA4\\_CCRi](#) register must not change during super-block life.
- The PAUSE\_LINK\_LIST bit must not be set in the initialization phase.

## 16.6 sDMA Register Manual

### 16.6.1 sDMA Instance Summary

**Table 16-22. sDMA Instance Summary**

Module Name	Base Address	Size
sDMA	0x4A05 6000	4KB

### 16.6.2 sDMA Registers

#### 16.6.2.1 sDMA Register Summary

Index  $i$  represents the logical channel number (where  $i = 0$  to 31). The offset address for some registers is calculated from channel  $c$  number. For example, register DMA4\_CCR10 (channel 10) has an offset address of  $10 \times 0x60 = 0x3C0$ , and so a physical address of  $0x4A05\ 6080 + 0x3C0 = 0x4A05\ 6440$ .

Index  $j$  represents the interrupt line number (where  $j = 0$  to 3). The offset address for some registers is calculated from channel  $c$  number. For example, register DMA4\_IRQSTATUS\_L3 (line 3) has an offset address of  $3 \times 0x4 = 0xC$ , and so a physical address of  $0x4A05\ 6008 + 0xC = 0x4A05\ 6014$ .

**Table 16-23. sDMA Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	sDMA L3 Base Address
DMA4_REVISION	R	32	0x0000 0000	0x4A05 6000
DMA4_IRQSTATUS_Lj <sup>(1)</sup>	RW	32	0x0000 0008 + (0x4 * j)	0x4A05 6008 + (0x4 * j)
DMA4_IRQENABLE_Lj <sup>(1)</sup>	RW	32	0x0000 0018 + (0x4 * j)	0x4A05 6018 + (0x4 * j)
DMA4_SYSSTATUS	R	32	0x0000 0028	0x4A05 6028
DMA4_OCP_SYSCONFIG	RW	32	0x0000 002C	0x4A05 602C
DMA4_CAPS_0	RW	32	0x0000 0064	0x4A05 6064
DMA4_CAPS_2	R	32	0x0000 006C	0x4A05 606C
DMA4_CAPS_3	R	32	0x0000 0070	0x4A05 6070
DMA4_CAPS_4	RW	32	0x0000 0074	0x4A05 6074
DMA4_GCR	RW	32	0x0000 0078	0x4A05 6078
DMA4_CCRi <sup>(2)</sup>	RW	32	0x0000 0080 + (0x60 * i)	0x4A05 6080 + (0x60 * i)
DMA4_CLNK_CTRLi <sup>(2)</sup>	RW	32	0x0000 0084 + (0x60 * i)	0x4A05 6084 + (0x60 * i)
DMA4_CICRi <sup>(2)</sup>	RW	32	0x0000 0088 + (0x60 * i)	0x4A05 6088 + (0x60 * i)
DMA4_CSRi <sup>(2)</sup>	RW	32	0x0000 008C + (0x60 * i)	0x4A05 608C + (0x60 * i)
DMA4_CSDPi <sup>(2)</sup>	RW	32	0x0000 0090 + (0x60 * i)	0x4A05 6090 + (0x60 * i)
DMA4_CENi <sup>(2)</sup>	RW	32	0x0000 0094 + (0x60 * i)	0x4A05 6094 + (0x60 * i)
DMA4_CFNi <sup>(2)</sup>	RW	32	0x0000 0098 + (0x60 * i)	0x4A05 6098 + (0x60 * i)
DMA4_CSSAi <sup>(2)</sup>	RW	32	0x0000 009C + (0x60 * i)	0x4A05 609C + (0x60 * i)
DMA4_CDSAi <sup>(2)</sup>	RW	32	0x0000 00A0 + (0x60 * i)	0x4A05 60A0 + (0x60 * i)

<sup>(1)</sup>  $j = 0$  to 3

<sup>(2)</sup>  $i = 0$  to 31

**Table 16-23. sDMA Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	sDMA L3 Base Address
DMA4_CSEIi <sup>(2)</sup>	RW	32	0x0000 00A4 + (0x60 * i)	0x4A05 60A4 + (0x60 * i)
DMA4_CSFli <sup>(2)</sup>	RW	32	0x0000 00A8 + (0x60 * i)	0x4A05 60A8 + (0x60 * i)
DMA4_CDEIi <sup>(2)</sup>	RW	32	0x0000 00AC + (0x60 * i)	0x4A05 60AC + (0x60 * i)
DMA4_CDFli <sup>(2)</sup>	RW	32	0x0000 00B0 + (0x60 * i)	0x4A05 60B0 + (0x60 * i)
DMA4_CSACi <sup>(2)</sup>	R	32	0x0000 00B4 + (0x60 * i)	0x4A05 60B4 + (0x60 * i)
DMA4_CDACi <sup>(2)</sup>	RW	32	0x0000 00B8 + (0x60 * i)	0x4A05 60B8 + (0x60 * i)
DMA4_CCENi <sup>(2)</sup>	RW	32	0x0000 00BC + (0x60 * i)	0x4A05 60BC + (0x60 * i)
DMA4_CCFNi <sup>(2)</sup>	RW	32	0x0000 00C0 + (0x60 * i)	0x4A05 60C0 + (0x60 * i)
DMA4_COLORi <sup>(2)</sup>	RW	32	0x0000 00C4 + (0x60 * i)	0x4A05 60C4 + (0x60 * i)
DMA4_CDPi <sup>(3)</sup>	RW	32	0x0000 00D0 + (0x60 * i)	0x4A05 60D0 + (0x60 * i)
DMA4_CNDPi <sup>(3)</sup>	RW	32	0x0000 00D4 + (0x60 * i)	0x4A05 60D4 + (0x60 * i)
DMA4_CCDNi <sup>(3)</sup>	RW	32	0x0000 00D8 + (0x60 * i)	0x4A05 60D8 + (0x60 * i)

<sup>(3)</sup> i = 0 to 31

### 16.6.2.2 sDMA Register Description

**NOTE:** Some registers have no reset value (marked with -) because of hardware implementation in memory. Software must ensure the correct programming of these registers, if needed.

Shadow registers are used to read run-time registers such as CCEN, CCFN, CDAC, and CSAC. Typically, when accessed in 8-bit or 16-bit access for two consecutive accesses, the value of the previous registers can change. A shadow register holds the entire value to let the next access recover the remaining 24 bits or 16 bits.

For non-32-bit transactions, start reading or writing from the LSByte first to enable the register update. There is no issue for 32-bit read-write transactions.

**Table 16-24. DMA4\_REVISION**

<b>Address Offset</b>	0x0000 0000																																																																	
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b> sDMA																																																																
<b>Description</b>	This register contains the DMA revision code																																																																	
<b>Type</b>	R																																																																	
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">REVISION</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	REVISION																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
REVISION																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	REVISION	Reserved, Write 0's for future compatibility. Read returns 0	R	See <sup>(1)</sup>																																																														

<sup>(1)</sup> TI internal Data

**Table 16-25. Register Call Summary for Register DMA4\_REVISION**

sDMA Register Manual

- [SDMA Register Summary: \[0\]](#)

**Table 16-26. DMA4\_IRQSTATUS\_Lj**

<b>Address Offset</b>	0x0000 0008 + (0x4 * j)	<b>Index</b>	j = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	The interrupt status register regroups all the status of the DMA4 channels that can generate an interrupt over line Lj.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH_31_0_Lj																															

Bits	Field Name	Description	Type	Reset
31:0	CH_31_0_Lj	Channel 31 Interrupt on Lj: When an interrupt is seen on the line Lj the status of a interrupting channel i is read in the bit field i.  Read 0x0: Channel Interrupt Lj false Write 0x0: Channel Interrupt Lj status bit unchanged Write 0x1: Channel Interrupt Lj status bit is reset Read 0x1: Channel Interrupt Lj true (pending)	RW	0x0000 0000

**Table 16-27. Register Call Summary for Register DMA4\_IRQSTATUS\_Lj**

sDMA Functional Description

- [SDMA Controller Interrupt Requests: \[0\] \[1\] \[2\] \[3\]](#)
- [Interrupt Generation: \[4\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[5\]](#)

**Table 16-28. DMA4\_IRQENABLE\_Lj**

<b>Address Offset</b>	0x0000 0018 + (0x4 * j)	<b>Index</b>	j = 0 to 3
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on line Lj		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH_31_0_Lj_EN																															

Bits	Field Name	Description	Type	Reset
31:0	CH_31_0_Lj_EN	Channel Interrupt on Lj mask/unmask : to Mask/Unmask a channel i interrupt on Lj the user writes 0/1 on the bit field i.  0x0: Channel Interrupt Lj is masked 0x1: Channel Interrupt Lj generates an interrupt when it occurs	RW	0x0000 0000

**Table 16-29. Register Call Summary for Register DMA4\_IRQENABLE\_Lj**

sDMA Functional Description

- [SDMA Controller Interrupt Requests: \[0\] \[1\] \[2\] \[3\]](#)
- [Interrupt Generation: \[4\]](#)

**Table 16-29. Register Call Summary for Register DMA4\_IRQENABLE\_Lj (continued)**

sDMA Basic Programming Model

- [Setup Configuration: \[5\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[6\]](#)

**Table 16-30. DMA4\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0028	
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b> sDMA
<b>Description</b>	The register provides status information about the module excluding the interrupt status information (see interrupt status register)	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RESETDONE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved for module-specific status information	R	0x0000 0000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is on-going Read 0x1: Reset completed	R	1

**Table 16-31. Register Call Summary for Register DMA4\_SYSSTATUS**

sDMA Register Manual

- [SDMA Register Summary: \[0\]](#)

**Table 16-32. DMA4\_OCP\_SYSCONFIG**

<b>Address Offset</b>	0x0000 002C	
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b> sDMA
<b>Description</b>	DMA system configuration register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MIDLEMODE	RESERVED	CLOCKACTIVITY	RESERVED	EMUFREE	SIDLEMODE	RESERVED	RESERVED	AUTOIDLE							



Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0x00000
13:12	MIDLEMODE	Read write power management, standby/wait control 0x0: Force-standby: MStandby is asserted only when all the DMA channels are disabled 0x1: No-Standby: MStandby is never asserted 0x2: Smart-Standby: MStandby is asserted if at least one of the following two conditions is satisfied: 1. All the channels are disabled, OR 2. There is no non-synchronized channel enabled AND [if hardware synchronized channel is enabled, then no DMA request input is asserted and no requests are pending to be serviced]. 0x3: Reserved	RW	0x0
11:10	RESERVED	Reserved for clocks activities extension	RW	0x0
9:8	CLOCKACTIVITY	Clocks activities during wake-up Bit 8: Interface clock 0x0: Interface clock can be switched-off Bit 9: Functional clock 0x0: Functional clock can be switched-off	R	0x0
7:6	RESERVED	Write 0's for future compatibility. Read returns 0	RW	0x0
5	EMUFREE	Enable sensitivity to MSuspend 0x0: DMA4 freezes its internal logic upon MSuspend assertion 0x1: DMA4 ignores the MSuspend input	RW	0
4:3	SIDLEMODE	Configuration port power management, Idle req/ack control 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Idle acknowledge is given by DMA4 if all of the conditions are true: 1. All the channels are disabled. 2. If hardware synchronized channel is enabled, then no DMA request input is asserted and no requests are pending to be serviced. 3. All transactions are completed on all the DMA ports. 4.No interrupts are pending to be serviced. 0x3: Reserved. Do not use	RW	0x0
2	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0
1	RESERVED	Reserved for non-GP devices	RW	0
0	AUTOIDLE	Internal interface clock gating strategy 0x0: Interface clock is free running 0x1: Automatic interface clock gating strategy is applied, based on the interface activity.	RW	0

**Table 16-33. Register Call Summary for Register DMA4\_OCP\_SYSCONFIG**

sDMA Functional Description

- [SDMA Controller Power Management: \[0\] \[1\] \[2\] \[3\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[4\]](#)

**Table 16-34. DMA4\_CAPS\_0**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	sDMA
<b>Physical Address</b>	See <a href="#">Table 16-23</a>		
<b>Description</b>	DMA Capabilities Register 0 LSW		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LINK_LIST_CPBLTY_TYPE4	LINK_LIST_CPBLTY_TYPE123	CONST_FILL_CPBLTY	TRANSPARENT_BLT_CPBLTY	RESERVED																			

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Write 0's for future compatibility. Read returns 0	RW	0x000
21	LINK_LIST_CPBLTY_TYPE4	Link List capability for type4 descriptor capability	R	0
20	LINK_LIST_CPBLTY_TYPE123	Link List capability for type123 descriptor capability	R	1
19	CONST_FILL_CPBLTY	Constant_Fill_Capability Read 0x0: No LCH supports constant fill copy Read 0x1: any LCH supports constant fill copy	R	1
18	TRANSPARENT_BLT_CPBLTY	Transparent_BLT_Capability Read 0x0: No LCH supports transparent BLT copy Read 0x1: any LCH supports transparent BLT copy	R	1
17:0	RESERVED	Write 0's for future compatibility. Read returns 0	RW	0x00000

**Table 16-35. Register Call Summary for Register DMA4\_CAPS\_0**

sDMA Register Manual

- [SDMA Register Summary: \[0\]](#)

**Table 16-36. DMA4\_CAPS\_2**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	sDMA
<b>Physical Address</b>	See <a href="#">Table 16-23</a>		
<b>Description</b>	DMA Capabilities Register 2		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SEPARATE_SRC_AND_DST_INDEX_CPBLTY	DST_DOUBLE_INDEX_ADRS_CPBLTY	DST_SINGLE_INDEX_ADRS_CPBLTY	DST_POST_INCRMNT_ADRS_CPBLTY	DST_CONST_ADRS_CPBLTY	SRC_DOUBLE_INDEX_ADRS_CPBLTY	SRC_SINGLE_INDEX_ADRS_CPBLTY	SRC_POST_INCREMENT_ADRS_CPBLTY	SRC_CONST_ADRS_CPBLTY							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Write 0's for future compatibility. Read returns 0	R	0x000000
8	SEPARATE_SRC_AND_DST_IN DEX_CPBLTY	Separate_source/destination_index_capability  Read 0x0: Does not support separate src/dst index for 2D addressing  Read 0x1: Supports separate src/dst index for 2D addressing	R	1
7	DST_DOUBLE_INDEX_ADRS_C PBLTY	Destination_double_index_address_capability  Read 0x0: Does not support double index address mode on the destination port  Read 0x1: Supports double index address mode on the destination port	R	1
6	DST_SINGLE_INDEX_ADRS_C PBLTY	Destination_single_index_address_capability  Read 0x0: Does not support single index address mode on the destination port  Read 0x1: Supports single index address mode on the destination port	R	1
5	DST_POST_INCRMNT_ADRS_ CPBLTY	Destination_post_increment_address_capability  Read 0x0: Does not supports post-increment address mode in the destination port  Read 0x1: Supports post-increment address mode in the destination port	R	1
4	DST_CONST_ADRS_CPBLTY	Destination_constant_address_capability  Read 0x0: Does not supports constant address mode in the destination port  Read 0x1: Supports constant address mode in the destination port	R	1
3	SRC_DOUBLE_INDEX_ADRS_ CPBLTY	Source_double_index_address_capability  Read 0x0: Does not support double index address mode on the source port  Read 0x1: Supports double index address mode on the source port	R	1
2	SRC_SINGLE_INDEX_ADRS_C PBLTY	Source_single_index_address_capability  Read 0x0: Does not support single index address mode on the source port  Read 0x1: Supports single index address mode in the source port	R	1

Bits	Field Name	Description	Type	Reset
1	SRC_POST_INCREMENT_ADR S_CPBLTY	Source_post_increment_address_capability  Read 0x0: Does not supports post-increment address mode in the source port  Read 0x1: Supports post-increment address mode in the source port	R	1
0	SRC_CONST_ADRS_CPBLTY	Source_constant_address_capability  Read 0x0: Does not supports constant address mode in the source port  Read 0x1: Supports constant address mode in the source port	R	1

**Table 16-37. Register Call Summary for Register DMA4\_CAPS\_2**

sDMA Register Manual

- [SDMA Register Summary: \[0\]](#)

**Table 16-38. DMA4\_CAPS\_3**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	sDMA
<b>Physical Address</b>	See <a href="#">Table 16-23</a>		
<b>Description</b>	DMA Capabilities Register 3		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BLOCK_SYNCHR_CPBLTY		PKT_SYNCHR_CPBLTY		CHANNEL_CHANINIG_CPBLTY		CHANNEL_INTERLEAVE_CPBLTY		RESERVED		FRAME_SYNCHR_CPBLTY		ELMNT_SYNCHR_CPBLTY			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0's for future compatibility. Read returns 0	R	0x000000
7	BLOCK_SYNCHR_CPBLTY	Block_synchronization_capability  Read 0x0: Does not support synchronization transfer on block boundary  Read 0x1: Supports synchronization transfer on block boundary	R	1
6	PKT_SYNCHR_CPBLTY	Packet_synchronization_capability  Read 0x0: Does not support synchronization transfer on packet boundary  Read 0x1: Supports synchronization transfer on packet boundary	R	1
5	CHANNEL_CHANINIG_CPBLTY	Channel_Chaninig_capability  Read 0x0: Does not support Channel Chaninig capability  Read 0x1: Supports Channel Chaninig capability	R	1

Bits	Field Name	Description	Type	Reset
4	CHANNEL_INTERLEAVE_CAPBLTY	Channel_interleave_capability Read 0x0: Does not support Channel interleave capability Read 0x1: Supports Channel_interleave capability	R	1
3:2	RESERVED		R	0x0
1	FRAME_SYNCHR_CPBLTY	Frame_synchronization_capability Read 0x0: Does not support synchronization transfer on Frame boundary Read 0x1: Supports synchronization transfer on Frame boundary	R	1
0	ELMNT_SYNCHR_CPBLTY	Element_synchronization_capability Read 0x0: Does not support synchronization transfer on Element boundary Read 0x1: Supports synchronization transfer on Element boundary	R	1

**Table 16-39. Register Call Summary for Register DMA4\_CAPS\_3**

sDMA Register Manual

- [SDMA Register Summary: \[0\]](#)

**Table 16-40. DMA4\_CAPS\_4**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	sDMA
<b>Physical Address</b>	See <a href="#">Table 16-23</a>		
<b>Description</b>	DMA Capabilities Register 4		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
RESERVED																EOSB_INTERRUPT_CPBLTY		RESERVED		DRAIN_END_INTERRUPT_CPBLTY		MISALIGNED_ADRS_ERR_INTERRUPT_CPBLTY		SUPERVISOR_ERR_INTERRUPT_CPBLTY		RESERVED		TRANS_ERR_INTERRUPT_CPBLTY		PKT_INTERRUPT_CPBLTY		SYNC_STATUS_CPBLTY		BLOCK_INTERRUPT_CPBLTY		LAST_FRAME_INTERRUPT_CPBLTY		FRAME_INTERRUPT_CPBLTY		HALF_FRAME_INTERRUPT_CPBLTY		EVENT_DROP_INTERRUPT_CPBLTY		RESERVED	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00000
14	EOSB_INTERRUPT_CPBLTY	End of Super Block detection capability.	R	1
13	RESERVED	Reserved for non-GP devices	R	1
12	DRAIN_END_INTERRUPT_CPBLTY	Drain End detection capability.	R	1
11	MISALIGNED_ADRS_ERR_INTERRUPT_CPBLTY	Misaligned error detection capability.	R	1

Bits	Field Name	Description	Type	Reset
10	SUPERVISOR_ERR_INTERRUPT_CPBLTY	Supervisor error detection capability.	R	1
9	RESERVED	Reserved for non-GP devices	R	1
8	TRANS_ERR_INTERRUPT_CPBLTY	Transaction error detection capability.	R	1
7	PKT_INTERRUPT_CPBLTY	End of Packet detection capability. Read 0x0: Does not support end of packet interrupt generation capability Read 0x1: Supports end of packet interrupt generation capability	R	1
6	SYNC_STATUS_CPBLTY	Sync_status_capability Read 0x0: Does not support synchronized transfer status bit generation Read 0x1: Supports synchronized transfer status bit generation	R	1
5	BLOCK_INTERRUPT_CPBLTY	End of block detection capability. Read 0x0: Does not support end of block interrupt generation capability Read 0x1: Supports end of block interrupt generation capability	R	1
4	LAST_FRAME_INTERRUPT_CPBLTY	Start of last frame detection capability. Read 0x0: Does not support last frame interrupt generation capability Read 0x1: Supports last frame interrupt generation capability	R	1
3	FRAME_INTERRUPT_CPBLTY	End of frame detection capability. Read 0x0: Does not support end of frame interrupt generation capability Read 0x1: Supports end of frame interrupt generation capability	R	1
2	HALF_FRAME_INTERRUPT_CPBLTY	Detection capability of the half of frame end. Read 0x0: Does not support half of frame interrupt generation capability Read 0x1: Supports half of frame interrupt generation capability	R	1
1	EVENT_DROP_INTERRUPT_CPBLTY	Request collision detection capability. Read 0x0: Does not support event drop interrupt generation capability Read 0x1: Supports event drop interrupt generation capability	R	1
0	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0

**Table 16-41. Register Call Summary for Register DMA4\_CAPS\_4**

sDMA Register Manual

- [SDMA Register Summary: \[0\]](#)

**Table 16-42. DMA4\_GCR**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	sDMA
<b>Physical Address</b>	See <a href="#">Table 16-23</a>		
<b>Description</b>	<a href="#">DMA4_GCR</a>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED								ARBITRATION_RATE								HI_LO_FIFO_BUDGET		HI_THREAD_RESERVED		RESERVED						MAX_CHANNEL_FIFO_DEPTH							
CHANNEL_ID_GATE																																	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00
24	CHANNEL_ID_GATE	Gates the Channel ID bus monitoring on both Read and Write ports 0x0: Gates the Channel ID qualifiers on both Read and Write Ports 0x1: Does not gate the Channel ID qualifiers on both Read and Write Ports	RW	0x0
23:16	ARBITRATION_RATE	Arbitration switching rate between prioritized and regular channel queues	RW	0x01
15:14	HI_LO_FIFO_BUDGET	Allow to have a separate Global FIFO budget for high and low priority channels. For Hi priority Channel: (Per_channel_Maximum FIFO depth + 1) x Number of active High priority Channel =< High Budget FIFO For Low priority channel: (Per_channel_Maximum FIFO depth + 1) x Number of active Low priority Channel =< Low Budget FIFO 0x0: no fixed budget for neither higher nor lower priority channel 0x1: 75% of FIFO for low priority and 25% for high priority channels 0x2: 25% of FIFO for low priority and 75% for high priority channels 0x3: 50% of FIFO for low priority and 50% for high priority channels	RW	0x0
13:12	HI_THREAD_RESERVED	Allow thread reservation for high priority channel on both read and write ports. 0x0: No ThreadID is reserved on the Read Port for high priority channels. No ThreadID is reserved on the Write Port for high priority channels. 0x1: Read Port ThreadID 0 is reserved for high priority channels. Write Port ThreadID 0 is reserved for high priority channels. 0x2: Read port ThreadID 0 and ThreadID 1 are reserved for high priority channels. Write Port ThreadID 0 is reserved for high priority channels. 0x3: Read PortThreadID 0, ThreadID 1 and ThreadID 2 are reserved for high priority channels. Write Port ThreadID 0 is reserved for high priority channels.	RW	0x0
11:8	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0

Bits	Field Name	Description	Type	Reset
7:0	MAX_CHANNEL_FIFO_DEPTH	Maximum FIFO depth allocated to one logical channel. Maximum FIFO depth can not be 0x0. It should be at least 0x1 or greater. Note that If channel limit is less than destination burst size enough data will not be accumulated in the data FIFO and it will never be sent out on the WR port. The burst size should be less than the FIFO limit specified in this bit field.	RW	0x10

**Table 16-43. Register Call Summary for Register DMA4\_GCR**

## sDMA Functional Description

- [Logical Channel Transfer Overview: \[0\] \[1\] \[2\]](#)
- [FIFO Queue Memory Pool: \[3\]](#)
- [Thread Budget Allocation: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [FIFO Budget Allocation: \[15\] \[16\]](#)

## sDMA Basic Programming Model

- [Setup Configuration: \[17\]](#)
- [Concurrent Software and Hardware Synchronization: \[18\]](#)

## sDMA Register Manual

- [SDMA Register Summary: \[19\]](#)
- [SDMA Register Description: \[20\]](#)

**Table 16-44. DMA4\_CCRi**

<b>Address Offset</b>	0x0000 0080 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Control Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	WRITE_PRIORITY	BUFFERING_DISABLE	SEL_SRC_DST_SYNC	PREFETCH	SUPERVISOR	RESERVED	SYNCHRO_CONTROL_UPPER	BS	TRANSPARENT_COPY_ENABLE	CONST_FILL_ENABLE	DST_AMODE	SRC_AMODE	RESERVED	WR_ACTIVE	RD_ACTIVE	SUSPEND_SENSITIVE	ENABLE	READ_PRIORITY	FS	SYNCHRO_CONTROL	SYNCHRO_CONTROL	SYNCHRO_CONTROL	SYNCHRO_CONTROL	SYNCHRO_CONTROL	SYNCHRO_CONTROL	SYNCHRO_CONTROL	SYNCHRO_CONTROL

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0
29:27	RESERVED	Reserved for non-GP devices	RW	0x0
26	WRITE_PRIORITY	Channel priority on the Write side  0x0: Channel has low priority on the Write side during the arbitration process  0x1: Channel has high priority on Write sided during the arbitration process	RW	0



Bits	Field Name	Description	Type	Reset
25	BUFFERING_DISABLE	This bit allows to disable the default buffering functionality when transfer is source synchronized.  0x0: buffering is enable across element/packet when source is synchronized to element, packet, frame or bolcks  0x1: buffering is disabled across element/packet when source is synchronized to element, packet, frame or bolcks	RW	-
24	SEL_SRC_DST_SYNC	Specifyes that element, packet, frame or block transfer (depending on CCR.bs and CCR.fs) is triggered by the source or the destination on the DMA request  0x0: Transfer is triggered by the destination. If synch on packet the packet element number is specified in the CDFI register  0x1: Transfer is triggered by the source. If synchronized on packet the packet element number is specified in the CSFI register	RW	-
23	PREFETCH	Enables the prefetch mode  0x0: Prefetch mode is disabled. When Sel_Src_Dst_Sync=1 transfers are buffred and pipelined between DMA requests  0x1: Prefetch mode is enabled. Prefetch mode is active only when destination is synchronised. It is SW user responsibility not to have at the same time Prefetch=1 when Sel_Src_Dst_Sync=1. This mode is not supported	RW	0
22	SUPERVISOR	Enables the supervisor mode  0x0: Supervisor mode is disabled  0x1: Supervisor mode is enabled	RW	0
21	RESERVED	Reserved for non GP devices	RW	0
20:19	SYNCHRO_CONTROL_UPPER	Channel Synchronization control upper (used in conjunction with the 5 bits of synchro channel <a href="#">DMA4_CCRi[4:0]</a> ) Used in conjunction, as two msb, with the five bits of the synchro channel bit field.	RW	0b00
18	BS	Block synchronization This bit used in conjunction with the fs to see how the DMA request is serviced in a synchronized transfer	RW	-
17	TRANSPARENT_COPY_ENABLE	Transparent copy enable  0x0: Transparent copy mode is disabled  0x1: Transparent copy mode is enabled	RW	-
16	CONST_FILL_ENABLE	Constant fill enable  0x0: Constant fill mode is disabled  0x1: Constant fill mode is enabled	RW	0
15:14	DST_AMODE	Selects the addressing mode on the Write Port of a channel.  0x0: Constant address mode  0x1: Post-incremented address mode  0x2: Single index address mode  0x3: Double index address mode	RW	0bxx
13:12	SRC_AMODE	Selects the addressing mode on the Read Port of a channel.  0x0: Constant address mode  0x1: Post-incremented address mode  0x2: Single index address mode  0x3: Double index address mode	RW	0bxx
11	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0

Bits	Field Name	Description	Type	Reset
10	WR_ACTIVE	Indicates if the channel write context is active or not Read 0x0: Channel is not active on the write port Read 0x1: Channel is currently active on the write port	R	0
9	RD_ACTIVE	Indicates if the channel read context is active or not Read 0x0: Channel is not active on the read port Read 0x1: Channel is currently active on the read port	R	0
8	SUSPEND_SENSITIVE	Logical channel suspend enable bit 0x0: The channel ignores the MSuspend even if EMUFree is set to 0. 0x1: If EMUFree is set to 0 and MSuspend comes in then all current OCP services (single transaction or burst transaction as specified in the corresponding CSDP register) have to be completed before stopping processing any more transactions	RW	0
7	ENABLE	Logical channel enable. It is SW responsibility to clear the CSR register and the IRQSTATUS bit for the different interrupt lines before enabling the channel. 0x0: The logical channel is disabled 0x1: The logical channel is enabled	RW	0
6	READ_PRIORITY	Channel priority on the read side 0x0: Channel has low priority on the Read side during the arbitration process 0x1: Channel has high priority on read sided during the arbitration process	RW	0
5	FS	Frame synchronization This bit used in conjunction with the BS to see how the DMA request is serviced in a synchronized transfer FS=0 and BS=0: An element is transferred once a DMA request is made. FS=0 and BS=1: An entire block is transferred once a DMA request is made. FS=1 and BS=0: An entire frame is transferred once a DMA request is made. FS=1 and BS=1: A packet is transferred once a DMA request is made. All these different transfers can be interleaved on the port with other DMA requests.	RW	-
4:0	SYNCHRO_CONTROL	Channel synchronization control This bit field used in conjunction with the second_level_synchro_control_upper (as two msb) 0000000 : 1s reserved for non synchronized LCH transfer xxxxxxx (from 1 to 127) There are 127 possible DMA request to assign to any LCH. <b>Note:</b> The channel synchronization control registers are 1-based. For example, to enable the S_DMA_1 request, DMA4_CCR[4:0] SYNCHRO_CONTROL must be set to 0x2 (DMA request number + 1).	RW	0b00000

**Table 16-45. Register Call Summary for Register DMA4\_CCRi**

sDMA Functional Description

- [Interrupt Generation: \[0\]](#)
- [Logical Channel Transfer Overview: \[1\] \[2\] \[3\]](#)
- [Addressing Modes: \[4\]](#)
- [Software Synchronization: \[5\] \[6\] \[7\]](#)
- [Hardware Synchronization: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)
- [Thread Budget Allocation: \[20\] \[21\]](#)
- [Reprogramming an Active Channel: \[22\] \[23\] \[24\] \[25\]](#)
- [Packet Synchronization: \[26\] \[27\] \[28\] \[29\] \[30\]](#)
- [Graphics Acceleration Support: \[31\]](#)
- [Supervisor Modes: \[32\]](#)
- [Disabling a Channel During Transfer: \[33\]](#)
- [FIFO Draining Mechanism: \[34\] \[35\] \[36\] \[37\] \[38\]](#)
- [Type 1: \[39\] \[40\]](#)
- [Starting a Linked List: \[41\]](#)
- [Drain: \[42\] \[43\]](#)
- [Abort: \[44\] \[45\]](#)
- [Status Bit Behavior: \[46\] \[47\] \[48\] \[49\] \[50\]](#)

sDMA Basic Programming Model

- [Software-Triggered \(Nonsynchronized\) Transfer: \[51\] \[52\] \[53\] \[54\] \[55\] \[56\] \[57\] \[58\] \[59\]](#)
- [Hardware-Synchronized Transfer: \[60\] \[61\] \[62\] \[63\] \[64\] \[65\] \[66\] \[67\] \[68\] \[69\] \[70\] \[71\] \[72\] \[73\] \[74\] \[75\] \[76\] \[77\] \[78\] \[79\] \[80\] \[81\] \[82\] \[83\] \[84\] \[85\] \[86\] \[87\] \[88\] \[89\] \[90\] \[91\] \[92\] \[93\] \[94\] \[95\] \[96\] \[97\] \[98\]](#)
- [Concurrent Software and Hardware Synchronization: \[99\] \[100\]](#)
- [Chained Transfer: \[101\]](#)
- [90-Degree Clockwise Image Rotation: \[102\] \[103\] \[104\] \[105\] \[106\] \[107\] \[108\]](#)
- [Graphic Operations: \[109\] \[110\] \[111\] \[112\]](#)
- [Linked-List Programming Guidelines: \[113\] \[114\] \[115\] \[116\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[117\]](#)
- [SDMA Register Description: \[118\]](#)

**Table 16-46. DMA4\_CLNK\_CTRLi**

<b>Address Offset</b>	0x0000 0084 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Link Control Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																ENABLE_LNK	RESERVED										NEXTLCH_ID					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15	ENABLE_LNK	Enables or disable the channel linking.  0x0: Channel linking mode is disabled When set on the fly to 0 the current channel will complete the transfer and stops the chain linking  0x1: Channel linking mode is enabled. The logical channel defined in the NextLCH_ID is enabled at the end of the current transfer	RW	0
14:5	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x000

Bits	Field Name	Description	Type	Reset
4:0	NEXTLCH_ID	Defines the NextLCh_ID, which is used to build logical channel chaining queue.	RW	0bxxxxx

**Table 16-47. Register Call Summary for Register DMA4\_CLNK\_CTRLi**

## sDMA Functional Description

- [Chained Logical Channel Transfers: \[0\] \[1\]](#)
- [FIFO Draining Mechanism: \[2\]](#)
- [Type 1: \[3\]](#)
- [Linked-List Channel Linking: \[4\] \[5\] \[6\] \[7\]](#)

## sDMA Basic Programming Model

- [Chained Transfer: \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [Linked-List Programming Guidelines: \[13\] \[14\]](#)

## sDMA Register Manual

- [SDMA Register Summary: \[15\]](#)

**Table 16-48. DMA4\_CICRi**

<b>Address Offset</b>	0x0000 0088 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Interrupt Control Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0															
RESERVED																SUPER_BLOCK_IE	RESERVED	DRAIN_IE	MISALIGNED_ERR_IE	SUPERVISOR_ERR_IE	RESERVED	TRANS_ERR_IE	PKT_IE	RESERVED	BLOCK_IE	LAST_IE	FRAME_IE	HALF_IE	DROP_IE	RESERVED																

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00000
14	SUPER_BLOCK_IE	Enables the end of super block interrupt	RW	-
13	RESERVED	Reserved for non-GP devices	RW	1
12	DRAIN_IE	Enables the end of draining interrupt	RW	0
11	MISALIGNED_ERR_IE	Enables the address misaligned error event interrupt 0x0: Disables the misaligned address error event interrupt 0x1: Enables the misaligned address error event interrupt	RW	-
10	SUPERVISOR_ERR_IE	Enables the supervisor transaction error event interrupt 0x0: Disables the supervisor transaction error event interrupt 0x1: Enables the supervisor transaction error event interrupt	RW	1
9	RESERVED	Reserved for non-GP devices	RW	1
8	TRANS_ERR_IE	Enables the transaction error event interrupt 0x0: Disables the transaction error event interrupt 0x1: Enables the transaction error event interrupt	RW	-

Bits	Field Name	Description	Type	Reset
7	PKT_IE	Enables the end of Packet interrupt 0x0: Disables the end of Packet transfer interrupt 0x1: Enables the end of Packet transfer interrupt	RW	-
6	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0
5	BLOCK_IE	Enables the end of block interrupt 0x0: Disables the end of block interrupt 0x1: Disables the end of block interrupt	RW	-
4	LAST_IE	Last frame interrupt enable (start of last frame) 0x0: Disables the last frame interrupt 0x1: Enables the last frame interrupt	RW	-
3	FRAME_IE	Frame interrupt enable (end of frame) 0x0: Disables the end of frame interrupt 0x1: Enables the end of frame interrupt	RW	-
2	HALF_IE	Enables or disables the half frame interrupt. 0x0: Disables the half frame interrupt 0x1: Enables the half frame interrupt	RW	-
1	DROP_IE	Synchronization event drop interrupt enable (request collision) 0x0: Disables the event drop interrupt 0x1: Enables the event drop interrupt	RW	0
0	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0

**Table 16-49. Register Call Summary for Register DMA4\_CICRi**

sDMA Functional Description

- [SDMA Controller Interrupt Requests: \[0\] \[1\]](#)
- [Interrupt Generation: \[2\]](#)
- [FIFO Draining Mechanism: \[3\]](#)
- [Descriptors: \[4\] \[5\] \[6\]](#)
- [Interrupt During Linked-List Execution: \[7\] \[8\] \[9\]](#)

sDMA Basic Programming Model

- [Setup Configuration: \[10\]](#)
- [Linked-List Programming Guidelines: \[11\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[12\]](#)

**Table 16-50. DMA4\_CSRi**

<b>Address Offset</b>	0x0000 008C + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Status Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
RESERVED																RESERVED	SUPER_BLOCK	RESERVED	DRAIN_END	MISALIGNED_ADRS_ERR	SUPERVISOR_ERR	RESERVED	TRANS_ERR	PKT	SYNC	BLOCK	LAST	FRAME	HALF	DROP	RESERVED											

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
16:15	RESERVED	Reserved for debug (Monitor descriptor/data load phase), Write 0's for future compatibility, Read returns 0	RW	0x0
14	SUPER_BLOCK	End of Super block event	RW	0
13	RESERVED	Reserved for non-GP devices	RW	0
12	DRAIN_END	End of channel draining	RW	0
11	MISALIGNED_ADRS_ERR	Misaligned address error event Read 0x0: No address error Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: An address error has been occurred	RW	0
10	SUPERVISOR_ERR	Supervisor transaction error event Read 0x0: No supervisor transaction error Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: A supervisor transaction error has been occurred	RW	0
9	RESERVED	Reserved for non-GP devices	RW	0
8	TRANS_ERR	Transaction error event Read 0x0: No transaction error Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: A transaction error has been occurred	RW	0
7	PKT	End of Packet transfer Read 0x0: The current packet transfer has not been finished Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The current packet has been transferred	RW	0
6	SYNC	Synchronization status of a channel. Read 0x0: Logical channel is not scheduled or servicing a non synchronized DMA request. Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: Logical channel is servicing a synchronized DMA request	RW	0
5	BLOCK	End of block event Read 0x0: The current block transfer has not been finished Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The current block has been transferred	RW	0
4	LAST	Last frame (start of last frame) Read 0x0: The start of the last frame to transfer is not reached Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The start of the last frame to transfer is reached	RW	0

Bits	Field Name	Description	Type	Reset
3	FRAME	End of frame event Read 0x0: The end of current transferred frame is not reached Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The end of current transferred frame is reached	RW	0
2	HALF	Half of frame event. Read 0x0: The half of current transferred frame is not reached Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: The half of current transferred frame is reached	RW	0
1	DROP	Synchronization event drop occurred during the transfer Read 0x0: No synchronization collision Write 0x0: Status bit unchanged Write 0x1: Status bit is reset Read 0x1: A synchronization collision has been occurred	RW	0
0	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0

**Table 16-51. Register Call Summary for Register DMA4\_CSRi**

sDMA Functional Description

- [SDMA Controller Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Interrupt Generation: \[5\]](#)
- [FIFO Draining Mechanism: \[6\]](#)
- [Interrupt During Linked-List Execution: \[7\]](#)
- [Status Bit Behavior: \[8\] \[9\] \[10\] \[11\]](#)

sDMA Basic Programming Model

- [Setup Configuration: \[12\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[13\]](#)

**Table 16-52. DMA4\_CSDPi**

<b>Address Offset</b>	0x0000 0090 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Source Destination Parameters		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SRC_ENDIAN	SRC_ENDIAN_LOCK	DST_ENDIAN	DST_ENDIAN_LOCK	WRITE_MODE	DST_BURST_EN	DST_PACKED	RESERVED				SRC_BURST_EN	SRC_PACKED	RESERVED				DATA_TYPE						

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x000
21	SRC_ENDIAN	Channel source endianness control 0x0: Source has Little Endian type 0x1: Source has Big Endian type	RW	-
20	SRC_ENDIAN_LOCK	Endianness Lock 0x0: Endianness adapt 0x1: Endianness lock	RW	-
19	DST_ENDIAN	Channel Destination endianness control 0x0: Destination has Little Endian type 0x1: Destination has Big Endian type	RW	-
18	DST_ENDIAN_LOCK	Endianness Lock 0x0: Endianness adapt 0x1: Endianness lock	RW	-
17:16	WRITE_MODE	Used to enable writing mode without posting or with posting 0x0: Write None Posted (WRNP) 0x1: Write (Posted) 0x2: All transaction are mapped on the Write command as posted except for the last transaction in the transfer mapped on a Write None Posted 0x3: Undefined	RW	0bxx
15:14	DST_BURST_EN	Used to enable bursting on the Write Port. Smaller burst size than the programmed burst size is also allowed 0x0: single access 0x1: 16 bytes or 4x32-bit / 2x64-bit burst access 0x2: 32 bytes or 8x32-bit / 4x64-bit burst access 0x3: 64 bytes or 16x32-bit / 8x64-bit burst access	RW	0b00
13	DST_PACKED	Destination receives packed data. 0x0: The destination target is non packed 0x1: The destination target is packed	RW	-
12:9	RESERVED	Write the reset value. Read returns reset value	RW	0x-
8:7	SRC_BURST_EN	Used to enable bursting on the Read Port. Smaller burst size than the programmed burst size is also allowed 0x0: single access 0x1: 16 bytes or 4x32-bit / 2x64-bit burst access 0x2: 32 bytes or 8x32-bit / 4x64-bit burst access 0x3: 64 bytes or 16x32-bit / 8x64-bit burst access	RW	0bxx
6	SRC_PACKED	Source provides packed data. 0x0: The source target is non packed 0x1: The source target is packed	RW	-
5:2	RESERVED	Write the reset value. Read returns reset value	RW	0x-
1:0	DATA_TYPE	Defines the type of the data moved in the channel. 0x0: 8 bits scalar 0x1: 16 bits scalar 0x2: 32 bits scalar 0x3: Reserved	RW	0bxx



**Table 16-53. Register Call Summary for Register DMA4\_CSDPi**

sDMA Functional Description

- [Addressing Modes](#): [0]
- [Packed Accesses](#): [1]
- [Burst Transactions](#): [2]
- [Endianism Conversion](#): [3] [4]
- [Hardware Synchronization](#): [5]
- [Graphics Acceleration Support](#): [6]
- [Posted and Nonposted Writes](#): [7]
- [Type 1](#): [8]

sDMA Basic Programming Model

- [Software-Triggered \(Nonsynchronized\) Transfer](#): [9] [10] [11] [12] [13] [14] [15] [16] [17]
- [Hardware-Synchronized Transfer](#): [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29]
- [90-Degree Clockwise Image Rotation](#): [30] [31] [32] [33] [34] [35] [36] [37] [38]
- [Linked-List Programming Guidelines](#): [39] [40]

sDMA Register Manual

- [SDMA Register Summary](#): [41]

**Table 16-54. DMA4\_CENi**

<b>Address Offset</b>	0x0000 0094 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Element Number		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CHANNEL_ELMNT_NBR																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00
23:0	CHANNEL_ELMNT_NBR	Number of elements within a frame (unsigned) to transfer	RW	0x-----

**Table 16-55. Register Call Summary for Register DMA4\_CENi**

sDMA Functional Description

- [Addressing Modes](#): [0] [1]

sDMA Basic Programming Model

- [Software-Triggered \(Nonsynchronized\) Transfer](#): [2]
- [Hardware-Synchronized Transfer](#): [3] [4] [5]
- [90-Degree Clockwise Image Rotation](#): [6]

sDMA Register Manual

- [SDMA Register Summary](#): [7]

**Table 16-56. DMA4\_CFNi**

<b>Address Offset</b>	0x0000 0098 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Frame Number		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CHANNEL_FRAME_NBR																							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15:0	CHANNEL_FRAME_NBR	Number of frames within the block to be transferred (unsigned)	RW	0x----

**Table 16-57. Register Call Summary for Register DMA4\_CFNi**

sDMA Functional Description

- [Addressing Modes: \[0\] \[1\]](#)

sDMA Basic Programming Model

- [Software-Triggered \(Nonsynchronized\) Transfer: \[2\]](#)
- [Hardware-Synchronized Transfer: \[3\] \[4\]](#)
- [90-Degree Clockwise Image Rotation: \[5\]](#)
- [Linked-List Programming Guidelines: \[6\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[7\]](#)

**Table 16-58. DMA4\_CSSAi**

<b>Address Offset</b>	0x0000 009C + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Source Start Address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC_START_ADRS																															

Bits	Field Name	Description	Type	Reset
31:0	SRC_START_ADRS	32 bits of the source start address	RW	0x-----

**Table 16-59. Register Call Summary for Register DMA4\_CSSAi**

sDMA Functional Description

- [Addressing Modes: \[0\]](#)

sDMA Basic Programming Model

- [Software-Triggered \(Nonsynchronized\) Transfer: \[1\]](#)
- [Hardware-Synchronized Transfer: \[2\] \[3\] \[4\] \[5\]](#)
- [90-Degree Clockwise Image Rotation: \[6\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[7\]](#)

**Table 16-60. DMA4\_CDSAi**

<b>Address Offset</b>	0x0000 00A0 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Destination Start Address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST_START_ADRS																															

Bits	Field Name	Description	Type	Reset
31:0	DST_START_ADRS	32 bits of the destination start address	RW	0x-----

**Table 16-61. Register Call Summary for Register DMA4\_CDSAi**

sDMA Functional Description
<ul style="list-style-type: none"> <li>Addressing Modes: [0]</li> </ul>
sDMA Basic Programming Model
<ul style="list-style-type: none"> <li>Software-Triggered (Nonsynchronized) Transfer: [1]</li> <li>Hardware-Synchronized Transfer: [2] [3] [4] [5]</li> <li>90-Degree Clockwise Image Rotation: [6]</li> </ul>
sDMA Register Manual
<ul style="list-style-type: none"> <li>SDMA Register Summary: [7]</li> </ul>

**Table 16-62. DMA4\_CSEIi**

<b>Address Offset</b>	0x0000 00A4 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See Table 16-23	<b>Instance</b>	sDMA
<b>Description</b>	Channel Source Element Index (Signed)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CHANNEL_SRC_ELMNT_INDEX																							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15:0	CHANNEL_SRC_ELMNT_INDEX	Channel source element index	RW	0x----

**Table 16-63. Register Call Summary for Register DMA4\_CSEIi**

sDMA Functional Description
<ul style="list-style-type: none"> <li>Addressing Modes: [0]</li> </ul>
sDMA Basic Programming Model
<ul style="list-style-type: none"> <li>Software-Triggered (Nonsynchronized) Transfer: [1]</li> <li>90-Degree Clockwise Image Rotation: [2]</li> </ul>
sDMA Register Manual
<ul style="list-style-type: none"> <li>SDMA Register Summary: [3]</li> </ul>

**Table 16-64. DMA4\_CSFli**

<b>Address Offset</b>	0x0000 00A8 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See Table 16-23	<b>Instance</b>	sDMA
<b>Description</b>	Channel Source Frame Index (Signed) or 16-bit Packet size		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH_SRC_FRM_INDEX_OR_16BIT_PKT_ELNT_NBR																															

Bits	Field Name	Description	Type	Reset
31:0	CH_SRC_FRM_INDEX_OR_16BIT_PKT_ELNT_NBR	Channel source frame index value if source address is in double index mode. Or if fs=bs=1 and DMA_CCR[SEL_SRC_DST_SYNC]=1; the bit field [15:0] gives the number of element in packet. The field [31:16] is unused for the packet size.	RW	0x-----

**Table 16-65. Register Call Summary for Register DMA4\_CSFl*i***

## sDMA Functional Description

- [Addressing Modes: \[0\]](#)
- [Hardware Synchronization: \[1\]](#)
- [Packet Synchronization: \[2\] \[3\] \[4\] \[5\] \[6\]](#)

## sDMA Basic Programming Model

- [Software-Triggered \(Nonsynchronized\) Transfer: \[7\]](#)
- [Hardware-Synchronized Transfer: \[8\] \[9\]](#)
- [90-Degree Clockwise Image Rotation: \[10\]](#)

## sDMA Register Manual

- [SDMA Register Summary: \[11\]](#)

**Table 16-66. DMA4\_CDEl*i***

<b>Address Offset</b>	0x0000 00AC + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Destination Element Index (Signed)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CHANNEL_DST_ELMNT_INDEX															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15:0	CHANNEL_DST_ELMNT_INDEX	Channel destination element index	RW	0x----

**Table 16-67. Register Call Summary for Register DMA4\_CDEl*i***

## sDMA Functional Description

- [Addressing Modes: \[0\]](#)

## sDMA Basic Programming Model

- [Software-Triggered \(Nonsynchronized\) Transfer: \[1\]](#)
- [90-Degree Clockwise Image Rotation: \[2\]](#)

## sDMA Register Manual

- [SDMA Register Summary: \[3\]](#)

**Table 16-68. DMA4\_CDFl*i***

<b>Address Offset</b>	0x0000 00B0 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Destination Frame Index (Signed) or 16-bit Packet size		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH_DST_FRM_IDX_OR_16BIT_PKT_ELNT_NBR																															

Bits	Field Name	Description	Type	Reset
31:0	CH_DST_FRM_IDX_OR_16BIT_PKT_ELNT_NBR	Channel destination frame index value if destination address is in double index mode. Or if fs=bs=1 and DMA_CCR[SEL_SRC_DST_SYNC]=0; the bit field [15:0] gives the number of element in packet. The field [31:16] is unused for the packet size..	RW	0x-----

**Table 16-69. Register Call Summary for Register DMA4\_CDFIi**

- sDMA Functional Description
- [Addressing Modes: \[0\]](#)
  - [Hardware Synchronization: \[1\]](#)
  - [Packet Synchronization: \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- sDMA Basic Programming Model
- [Software-Triggered \(Nonsynchronized\) Transfer: \[7\]](#)
  - [Hardware-Synchronized Transfer: \[8\] \[9\]](#)
  - [90-Degree Clockwise Image Rotation: \[10\]](#)
- sDMA Register Manual
- [SDMA Register Summary: \[11\]](#)

**Table 16-70. DMA4\_CSACi**

<b>Address Offset</b>	0x0000 00B4 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Source Address Value. User has to access this register only in 32-bit access. If accessed in 8-bit or 16bit data may be corrupted.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC_ELMNT_ADRS																															

Bits	Field Name	Description	Type	Reset
31:0	SRC_ELMNT_ADRS	Current source address counter value	R	0x-----

**Table 16-71. Register Call Summary for Register DMA4\_CSACi**

- sDMA Register Manual
- [SDMA Register Summary: \[0\]](#)

**Table 16-72. DMA4\_CDACi**

<b>Address Offset</b>	0x0000 00B8 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Destination Address Value. User has to access this register only in 32-bit access. If accessed in 8-bit or 16bit data may be corrupted.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST_ELMNT_ADRS																															

Bits	Field Name	Description	Type	Reset
31:0	DST_ELMNT_ADRS	Current destination address counter value	RW	0x-----

**Table 16-73. Register Call Summary for Register DMA4\_CDACi**

- sDMA Functional Description
- [Hardware Synchronization: \[0\]](#)
- sDMA Basic Programming Model
- [Hardware-Synchronized Transfer: \[1\]](#)
  - [Synchronized Transfer Monitoring Using CDAC: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- sDMA Register Manual
- [SDMA Register Summary: \[9\]](#)

**Table 16-74. DMA4\_CCENi**

<b>Address Offset</b>	0x0000 00BC + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Current Transferred Element Number in the current frame. User has to access this register only in 32-bit access. If accessed in 8-bit or 16bit data may be corrupted.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CURRENT_ELMNT_NBR																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x00
23:0	CURRENT_ELMNT_NBR	Channel current transferred element number in the current frame	RW	0x-----

**Table 16-75. Register Call Summary for Register DMA4\_CCENi**

sDMA Functional Description

- [Monitoring a Linked-List Progression: \[0\] \[1\] \[2\]](#)

sDMA Basic Programming Model

- [Synchronized Transfer Monitoring Using CDAC: \[3\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[4\]](#)

**Table 16-76. DMA4\_CCFNi**

<b>Address Offset</b>	0x0000 00C0 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel Current Transferred Frame Number in the current transfer. User has to access this register only in 32-bit access. If accessed in 8-bit or 16bit data may be corrupted.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CURRENT_FRAME_NBR																							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x0000
15:0	CURRENT_FRAME_NBR	Channel current transferred frame number in the current transfer	RW	0x----

**Table 16-77. Register Call Summary for Register DMA4\_CCFNi**

sDMA Functional Description

- [Monitoring a Linked-List Progression: \[0\] \[1\] \[2\]](#)

sDMA Basic Programming Model

- [Synchronized Transfer Monitoring Using CDAC: \[3\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[4\]](#)

**Table 16-78. DMA4\_COLORi**

<b>Address Offset</b>	0x0000 00C4 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	Channel DMA COLOR KEY /SOLID COLOR		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CH_BLT_FRGRND_COLOR_OR_SOLID_COLOR_PTRN																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0x--
23:0	CH_BLT_FRGRND_COLOR_OR_SOLID_COLOR_PTRN	Color key or solid color pattern: The pattern is replicated according to the data type. If the data-type is 8-bit the pattern is replicated 4 times to fill the register in order to enhance processing when data is packed at the graphic module input. The same reasoning for 16-bit data-type.	RW	0x-----

**Table 16-79. Register Call Summary for Register DMA4\_COLORi**

sDMA Functional Description

- [Graphics Acceleration Support: \[0\] \[1\]](#)
- [Type 1: \[2\]](#)

sDMA Basic Programming Model

- [Graphic Operations: \[3\] \[4\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[5\]](#)

**Table 16-80. DMA4\_CDPi**

<b>Address Offset</b>	0x0000 00D0 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	This register controls the various parameters of the link list mechanism		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FAST	TRANSFER_MODE	PAUSE_LINK_LIST	NEXT_DESCRIPTOR_TYPE	SRC_VALID	DEST_VALID										

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0x00000
10	FAST	Sets the fast-start mode for linked list descriptor types 1, 2 and 3  0x0: No fast-start mode 0x1: Fast-start mode is enabled.	RW	0x0

Bits	Field Name	Description	Type	Reset
9:8	TRANSFER_MODE	Enable linked-list transfer mode 0x0: Normal transfer mode is used. 0x1: Linked-list channel mode for type 1, 2, or 3 descriptor is used. 0x2: Undefined 0x3: Undefined	RW	0x0
7	PAUSE_LINK_LIST	Suspend the linked-list transfer at completion of the current block transfer. 0x0: Linked list is active. 0x1: Linked list is suspended at the boundary of next descriptor loading.	RW	0x0
6:4	NEXT_DESCRIPTOR_TYPE	Next Descriptor Type 0x0: Undefined 0x1: Next descriptor is of type 1. 0x2: Next descriptor is of type 2. 0x3: Next descriptor is of type 3. 0x4: Undefined 0x5: Undefined 0x6: Undefined 0x7: Undefined	RW	0x-
3:2	SRC_VALID	Source address valid 0x0: The source address is not present in the next descriptor and continuous incrementing is enabled. 0x1: The source address must be reloaded in the next descriptor transfer. 0x2: The source start address is not present in the next descriptor. But will reload the one from configuration memory which belongs to the previous descriptor. 0x3: Undefined addressing mode	RW	0x-
1:0	DEST_VALID	Destination address valid 0x0: The destination address is not present in the next descriptor and continuous incrementing is enabled. 0x1: The destination address must be reloaded in the next descriptor transfer. 0x2: The destination start address is not present in the next descriptor. But will reload the one from configuration memory which belongs to the previous descriptor. 0x3: Undefined addressing mode	RW	0x-

**Table 16-81. Register Call Summary for Register DMA4\_CDPi**

## sDMA Functional Description

- [Link-List Transfer Profile: \[0\]](#)
- [Descriptors: \[1\] \[2\]](#)
- [Transfer Mode Setting: \[3\] \[4\] \[5\] \[6\]](#)
- [Starting a Linked List: \[7\]](#)
- [Pause a Linked List: \[8\] \[9\] \[10\] \[11\]](#)

## sDMA Basic Programming Model

- [Linked-List Programming Guidelines: \[12\]](#)

## sDMA Register Manual

- [SDMA Register Summary: \[13\]](#)



**Table 16-82. DMA4\_CNDPi**

<b>Address Offset</b>	0x0000 00D4 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>	This register contains the Next descriptor Address Pointer for the link list Mechanism		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_DESCRIPTOR_POINTER																RESERVED															

Bits	Field Name	Description	Type	Reset
31:2	NEXT_DESCRIPTOR_POINTER	This register contains the Next descriptor Address Pointer for the link list Mechanism	RW	0bxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx
1:0	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0x0

**Table 16-83. Register Call Summary for Register DMA4\_CNDPi**

sDMA Functional Description

- [Link-List Transfer Profile: \[0\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[1\]](#)

**Table 16-84. DMA4\_CCDNi**

<b>Address Offset</b>	0x0000 00D8 + (0x60 * i)	<b>index:</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 16-23</a>	<b>Instance</b>	sDMA
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CURRENT_DESCRIPTOR_NBR															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility, Reads return 0	RW	0x0000
15:0	CURRENT_DESCRIPTOR_NBR	This register when read contains the current active descriptor number in the link list. This register is Read/write to allow user initialization.	RW	0x----

**Table 16-85. Register Call Summary for Register DMA4\_CCDNi**

sDMA Functional Description

- [Monitoring a Linked-List Progression: \[0\] \[1\] \[2\]](#)

sDMA Register Manual

- [SDMA Register Summary: \[3\]](#)

PRELIMINARY

## Interrupt Controllers

This chapter describes the interrupt controllers (INTCs) in the device.

Topic	Page
<b>17.1 Interrupt Controllers Overview .....</b>	<b>3300</b>
<b>17.2 Interrupt Controllers Environment .....</b>	<b>3302</b>
<b>17.3 Interrupt Controllers Integration .....</b>	<b>3303</b>
<b>17.4 Interrupt Controllers Functional Description .....</b>	<b>3309</b>
<b>17.5 Interrupt Controllers Register Manual .....</b>	<b>3310</b>

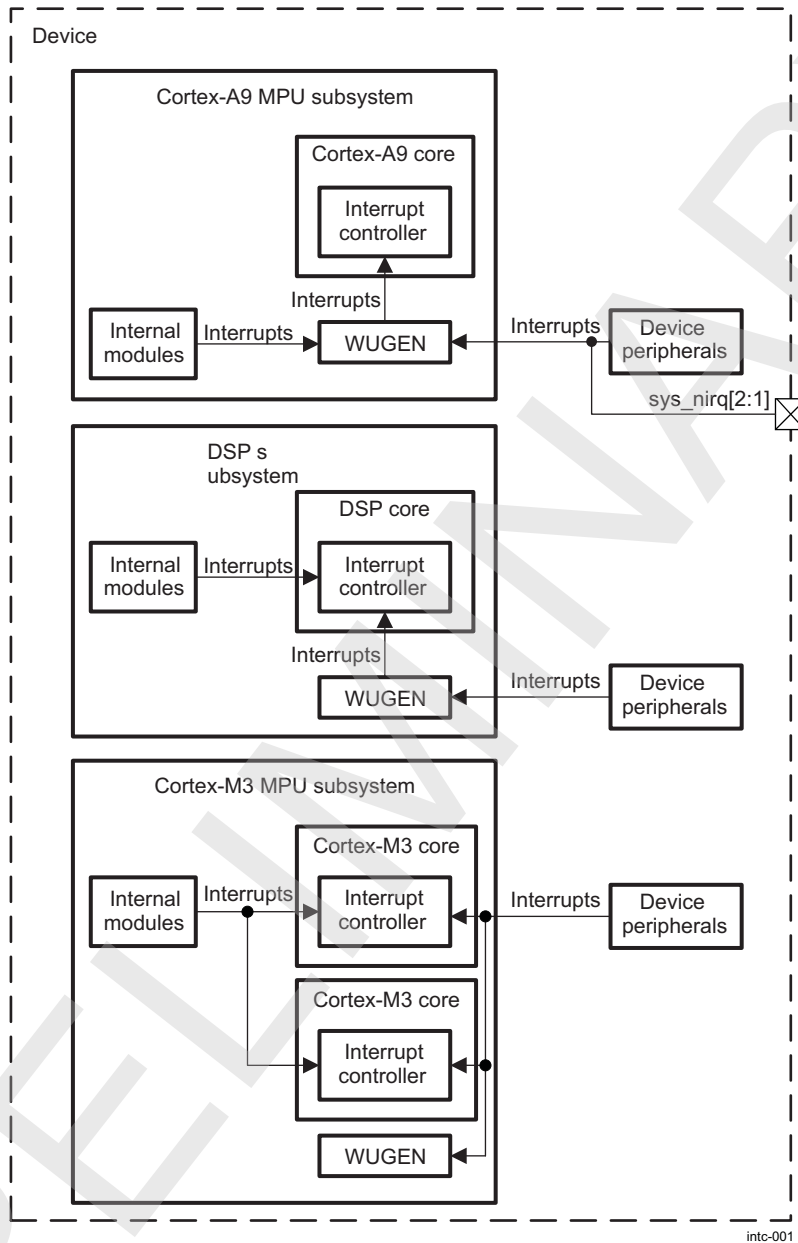
## 17.1 Interrupt Controllers Overview

The device provides three interrupt controller (INTC) modules, which handle interrupts at the device level:

- Cortex™-A9 MPU subsystem (MPUSS) interrupt controller (Cortex-A9 MPU INTC): This INTC module is a single functional unit that is integrated in the Cortex-A9 multiprocessor core alongside Cortex-A9 processors. It provides:
  - 128 hardware interrupt inputs
  - Generation of interrupts by software
  - Prioritization of interrupts
  - Masking of any interrupts
  - Distribution of the interrupts to the target Cortex-A9 processor(s)
  - Tracking the status of interrupts
  - Wake-up generator (WUGEN) module on external and local interrupts
- DSP subsystem (DSPSS) interrupt controller: This module is a specific combination of wake-up generator (WUGEN) and the C64x+ DSP INTC. It is used in the device but not described in detail in this chapter. For detailed information about this module, see [Chapter 5, DSP Subsystem](#).
- Cortex-M3 MPUSS interrupt controller (Cortex-M3 MPU INTC): This module is also called Nested Vectored Interrupt Controller (NVIC) and is integrated within each Cortex-M3 so the different interrupt lines are directly connected to each core. The interrupt mapping is the same for the two cores to facilitate parallel processing. The NVIC supports:
  - 64 external interrupts (in addition to 16 Cortex-M3 internal interrupts), which are dynamically prioritized with 16 levels of priority defined for each core
  - Low-latency exception and interrupt handling
  - Prioritization and handling of exceptions
  - Control of the local power management
  - Debug accesses to the processor core
  - Wake-up generator (WUGEN) module on external interrupts, coming from device peripherals

Figure 17-1 shows the internal interrupt scheme of the device.

Figure 17-1. Interrupt Controllers Overview



**NOTE:** For more information about the WUGEN module in the Cortex-A9 MPUSS (including integration, functional and register description), see [Chapter 4, Cortex-A9 MPU Subsystem](#). For more information about the WUGEN module in the Cortex-M3 MPUSS (including integration, functional and register description), see [Chapter 7, Cortex-M3 MPU Subsystem](#).

## 17.2 Interrupt Controllers Environment

This section describes the INTC application fields from the environment point of view (external connections).

Figure 17-2 shows the relationship between the module and external interrupts.

**Figure 17-2. Interrupts From External Devices**

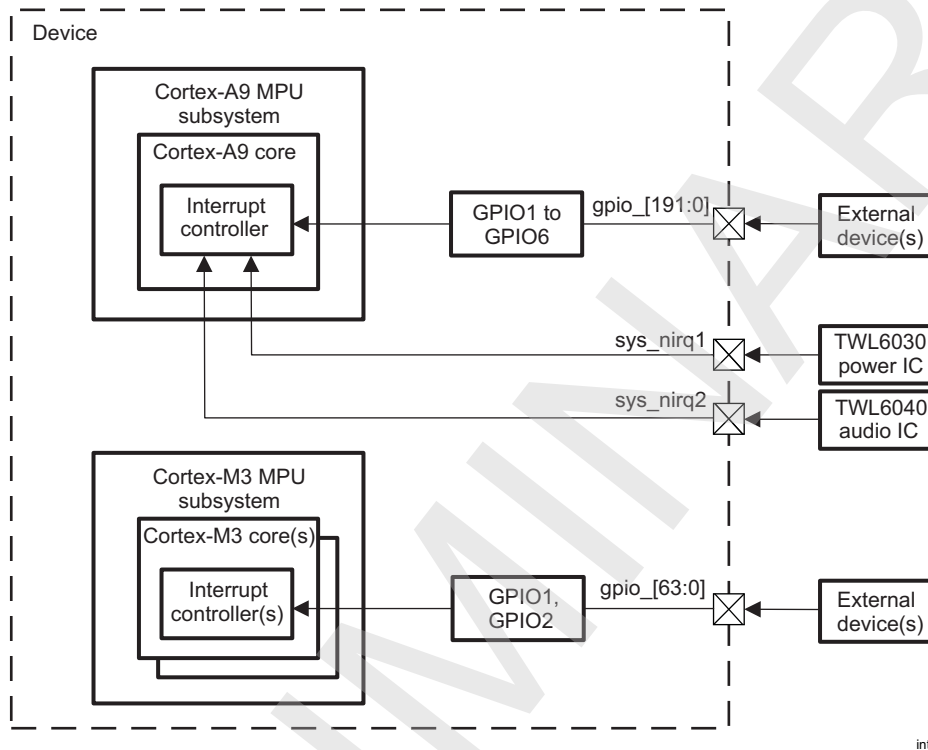


Table 17-1 describes the I/O signals that can be used by external devices to generate interrupts to the Cortex-A9 MPU INTC or Cortex-M3 MPU INTC.

**Table 17-1. Interrupt Controllers I/O Signals**

Signal Name	I/O <sup>(1)</sup>	Reset Value	Description	Cortex-A9 MPU INTC	Cortex-M3 MPU INTC
sys_nirq1	I	1	TWL6030 power IC can use this pin to generate a system wake-up event to Cortex-A9 MPU INTC.	x	-
sys_nirq2	I	1	TWL6040 audio IC can use this pin to generate a system wake-up event to Cortex-A9 MPU INTC.	x	-
gpio_[31:0]	I/O	-	External devices can use GPIO modules to generate interrupts to the Cortex-A9 MPU or Cortex-M3 MPU. There are six dedicated interrupt lines (for GPIO1 to GPIO6) to the Cortex-A9 MPU INTC and two dedicated interrupt lines (for GPIO1 and GPIO2) to the Cortex-M3 MPU INTC. Each GPIO module can generate a single interrupt whenever there is at least one event in any one of the configured 32 GPIO inputs. For more information about GPIO interrupt generation, see <a href="#">Chapter 25, General-Purpose Interface</a> .	x	x
gpio_[63:32]	I/O	-		x	x
gpio_[95:64]	I/O	-		x	-
gpio_[127:96]	I/O	-		x	-
gpio_[159:128]	I/O	-		x	-
gpio_[191:160]	I/O	-		x	-

<sup>(1)</sup> I = Input, O = Output

## 17.3 Interrupt Controllers Integration

The Cortex-A9 MPU INTC is integrated inside the Cortex-A9 MPU core. For more information about Cortex-A9 MPU INTC integration, see [Chapter 4, Cortex-A9 MPU Subsystem Integration](#).

The Cortex-M3 MPU INTC is integrated inside each Cortex-M3 MPU core. For more information about Cortex-M3 MPU INTC integration, see [Chapter 7, Dual Cortex-M3 MPU Subsystem Integration](#).

### 17.3.1 Interrupts Mapping

**NOTE:** All device peripherals interrupts (external to the Cortex-A9 MPU, DSP, and Cortex-M3 MPU subsystems) are active-high, level sensitive.

#### CAUTION

A single interrupt source can be physically mapped to multiple INTCs (Cortex-A9 MPU, DSP, and Cortex-M3 MPU subsystems). With multiple-mapped interrupts, it is strongly recommended to unmask each interrupt source in only one INTC at a time.

### 17.3.2 Interrupt Requests to Cortex-A9 MPU INTC

[Table 17-2](#) shows the Cortex-A9 MPU INTC interrupt mapping.

**Table 17-2. Cortex-A9 MPU INTC Interrupt Mapping**

Interrupt Line (Destination)	Interrupt Line (Source)	Module Name	Description
MA_IRQ_0	L2_CACHE_IRQ	CORTEXA9	L2 cache controller interrupt <sup>(1)</sup>
MA_IRQ_1	CTI_IRQ_0	CORTEXA9	Cross-trigger module 0 (CTI0) interrupt <sup>(1)</sup>
MA_IRQ_2	CTI_IRQ_1	CORTEXA9	Cross-trigger module 1 (CTI1) interrupt <sup>(1)</sup>
MA_IRQ_3	Reserved	Reserved	Reserved
MA_IRQ_4	ELM_IRQ	ELM	Error location process completion <sup>(2)</sup>
MA_IRQ_5	Reserved	Reserved	Reserved
MA_IRQ_6	Reserved	Reserved	Reserved
MA_IRQ_7	sys_nirq1	External	External interrupt 1 (active low)
MA_IRQ_8	Reserved	Reserved	Reserved
MA_IRQ_9	L3_DBG_IRQ	L3	L3 interconnect debug error
MA_IRQ_10	L3_APP_IRQ	L3	L3 interconnect application error
MA_IRQ_11	PRCM_MPU_IRQ	PRCM	PRCM interrupt
MA_IRQ_12	SDMA_IRQ_0	sDMA	sDMA interrupt 0 <sup>(4)</sup>
MA_IRQ_13	SDMA_IRQ_1	sDMA	sDMA interrupt 1 <sup>(4)</sup>
MA_IRQ_14	SDMA_IRQ_2	sDMA	sDMA interrupt 2 <sup>(3)</sup>
MA_IRQ_15	SDMA_IRQ_3	sDMA	sDMA interrupt 3 <sup>(3)</sup>
MA_IRQ_16	MCBSP4_IRQ	MCBSP4	MCBSP4 interrupt <sup>(2)</sup>
MA_IRQ_17	MCBSP1_IRQ	MCBSP1	MCBSP1 interrupt <sup>(2)</sup>
MA_IRQ_18	SR_MPU_IRQ	SR_MPU	SmartReflex MPU interrupt
MA_IRQ_19	SR_CORE_IRQ	SR_CORE	SmartReflex Core interrupt
MA_IRQ_20	GPMC_IRQ	GPMC	GPMC interrupt <sup>(2)</sup>
MA_IRQ_21	SGX_IRQ	SGX	2D/3D graphics module interrupt
MA_IRQ_22	MCBSP2_IRQ	MCBSP2	MCBSP2 interrupt <sup>(2)</sup>
MA_IRQ_23	MCBSP3_IRQ	MCBSP3	MCBSP3 interrupt <sup>(2)</sup>
MA_IRQ_24	ISS_IRQ5	Imaging subsystem	Imaging subsystem interrupt 5 <sup>(3)</sup>

**Table 17-2. Cortex-A9 MPU INTC Interrupt Mapping (continued)**

Interrupt Line (Destination)	Interrupt Line (Source)	Module Name	Description
MA_IRQ_25	DSS_DISPC_IRQ	Display controller	Display controller interrupt <sup>(4)</sup>
MA_IRQ_26	MAIL_U0_MPU_IRQ	MAILBOX	Mailbox user 0 interrupt
MA_IRQ_27	C2C_SSCM_IRQ0	C2C	C2C status interrupt <sup>(4)</sup>
MA_IRQ_28	DSP_MMU_IRQ	DSP	DSP MMU interrupt
MA_IRQ_29	GPIO1_MPU_IRQ	GPIO1	GPIO1 MPU interrupt <sup>(3)</sup>
MA_IRQ_30	GPIO2_MPU_IRQ	GPIO2	GPIO2 MPU interrupt <sup>(3)</sup>
MA_IRQ_31	GPIO3_MPU_IRQ	GPIO3	GPIO3 MPU interrupt
MA_IRQ_32	GPIO4_MPU_IRQ	GPIO4	GPIO4 MPU interrupt
MA_IRQ_33	GPIO5_MPU_IRQ	GPIO5	GPIO5 MPU interrupt
MA_IRQ_34	GPIO6_MPU_IRQ	GPIO6	GPIO6 MPU interrupt
MA_IRQ_35	Reserved	Reserved	Reserved
MA_IRQ_36	WDT3_IRQ	WDTIMER3	WDTIMER3 overflow
MA_IRQ_37	GPT1_IRQ	GPTIMER1	GPTIMER1 interrupt
MA_IRQ_38	GPT2_IRQ	GPTIMER2	GPTIMER2 interrupt
MA_IRQ_39	GPT3_IRQ	GPTIMER3	GPTIMER3 interrupt <sup>(3)</sup>
MA_IRQ_40	GPT4_IRQ	GPTIMER4	GPTIMER4 interrupt <sup>(3)</sup>
MA_IRQ_41	GPT5_IRQ	GPTIMER5	GPTIMER5 interrupt <sup>(2)</sup>
MA_IRQ_42	GPT6_IRQ	GPTIMER6	GPTIMER6 interrupt <sup>(2)</sup>
MA_IRQ_43	GPT7_IRQ	GPTIMER7	GPTIMER7 interrupt <sup>(2)</sup>
MA_IRQ_44	GPT8_IRQ	GPTIMER8	GPTIMER8 interrupt <sup>(2)</sup>
MA_IRQ_45	GPT9_IRQ	GPTIMER9	GPTIMER9 interrupt <sup>(3)</sup>
MA_IRQ_46	GPT10_IRQ	GPTIMER10	GPTIMER10 interrupt
MA_IRQ_47	GPT11_IRQ	GPTIMER11	GPTIMER11 interrupt <sup>(3)</sup>
MA_IRQ_48	MCSP14_IRQ	MCSP14	MCSP14 interrupt
MA_IRQ_49	Reserved	Reserved	Reserved
MA_IRQ_50	Reserved	Reserved	Reserved
MA_IRQ_51	Reserved	Reserved	Reserved
MA_IRQ_52	Reserved	Reserved	Reserved
MA_IRQ_53	DSS_DSI1_IRQ	MIPI DSI1	Display Subsystem DSI1 interrupt <sup>(4)</sup>
MA_IRQ_54	Reserved	Reserved	Reserved
MA_IRQ_55	Reserved	Reserved	Reserved
MA_IRQ_56	I2C1_IRQ	I2C1	I2C1 interrupt <sup>(3)</sup>
MA_IRQ_57	I2C2_IRQ	I2C2	I2C2 interrupt <sup>(3)</sup>
MA_IRQ_58	HDQ_IRQ	HDQ	HDQ/1wire interrupt
MA_IRQ_59	MMC5_IRQ	HSMMC5	MMC5 interrupt <sup>(3)</sup>
MA_IRQ_60	Reserved	Reserved	Reserved
MA_IRQ_61	I2C3_IRQ	I2C3	I2C3 interrupt <sup>(3)</sup>
MA_IRQ_62	I2C4_IRQ	I2C4	I2C4 interrupt <sup>(3)</sup>
MA_IRQ_63	Reserved	Reserved	Reserved
MA_IRQ_64	Reserved	Reserved	Reserved
MA_IRQ_65	MCSP11_IRQ	MCSP11	MCSP11 interrupt <sup>(4)</sup>
MA_IRQ_66	MCSP12_IRQ	MCSP12	MCSP12 interrupt <sup>(3)</sup>
MA_IRQ_67	HSI_P1_MPU_IRQ	HSI	HSI Port 1 interrupt <sup>(3)</sup>
MA_IRQ_68	HSI_P2_MPU_IRQ	HSI	HSI Port 2 interrupt <sup>(3)</sup>
MA_IRQ_69	FDIF_IRQ_3	Face Detect	Face detect interrupt 3
MA_IRQ_70	UART4_IRQ	UART4	UART module 4 interrupt
MA_IRQ_71	HSI_DMA_MPU_IRQ	HSI	HSI DMA engine MPU request <sup>(3)</sup>



**Table 17-2. Cortex-A9 MPU INTC Interrupt Mapping (continued)**

<b>Interrupt Line (Destination)</b>	<b>Interrupt Line (Source)</b>	<b>Module Name</b>	<b>Description</b>
MA_IRQ_72	UART1_IRQ	UART1	UART1 interrupt
MA_IRQ_73	UART2_IRQ	UART2	UART2 interrupt
MA_IRQ_74	UART3_IRQ	UART3	UART3 interrupt <sup>(4)</sup>
MA_IRQ_75	PBIAS_IRQ	PRCM	Merged interrupt for PBIASlite1 and 2
MA_IRQ_76	HSUSB_OHCI_IRQ	HSUSBHOST	HSUSB MP host interrupt OHCI controller
MA_IRQ_77	HSUSB_EHCI_IRQ	HSUSBHOST	HSUSB MP host interrupt EHCI controller <sup>(3)</sup>
MA_IRQ_78	HSUSB_TLL_IRQ	HSUSBTLL	HSUSB MP TLL interrupt <sup>(3)</sup>
MA_IRQ_79	Reserved	Reserved	Reserved
MA_IRQ_80	WDT2_IRQ	WDTIMER2	WDTIMER2 interrupt
MA_IRQ_81	Reserved	Reserved	Reserved
MA_IRQ_82	Reserved	Reserved	Reserved
MA_IRQ_83	MMC1_IRQ	HSMMC1	MMC1 interrupt <sup>(4)</sup>
MA_IRQ_84	DSS_DSI2_IRQ	MIPI DSI2	Display subsystem DSI2 interrupt <sup>(4)</sup>
MA_IRQ_85	Reserved	Reserved	Reserved
MA_IRQ_86	MMC2_IRQ	HSMMC2	MMC2 interrupt <sup>(4)</sup>
MA_IRQ_87	MPU_ICR_IRQ	ICR	ICR interrupt
MA_IRQ_88	C2C_SSCM_IRQ1	C2C	C2C GPI interrupt <sup>(4)</sup>
MA_IRQ_89	FSUSB_IRQ	FSUSB	FS-USB - host controller Interrupt <sup>(3)</sup>
MA_IRQ_90	FSUSB_SMI_IRQ	FSUSB	FS-USB - host controller SMI Interrupt <sup>(3)</sup>
MA_IRQ_91	MCSPi3_IRQ	MCSPi3	MCSPi3 interrupt
MA_IRQ_92	HSUSB_OTG_IRQ	HSUSBOTG	HSUSB OTG controller interrupt <sup>(3)</sup>
MA_IRQ_93	HSUSB_OTG_DMA_IRQ	HSUSBOTG	HSUSB OTG DMA interrupt <sup>(3)</sup>
MA_IRQ_94	MMC3_IRQ	HSMMC3	MMC3 interrupt <sup>(3)</sup>
MA_IRQ_95	Reserved	Reserved	Reserved
MA_IRQ_96	MMC4_IRQ	HSMMC4	MMC4 interrupt <sup>(3)</sup>
MA_IRQ_97	SLIMBUS1_IRQ	SLIMBUS1	SLIMBUS1 interrupt <sup>(2)</sup>
MA_IRQ_98	SLIMBUS2_IRQ	SLIMBUS2	SLIMBUS2 interrupt <sup>(2)</sup>
MA_IRQ_99	ABE_MPU_IRQ	ABE subsystem	Audio back-end interrupt
MA_IRQ_100	CORTEXM3_MMU_IRQ	CORTEXM3	Cortex-M3 MMU interrupt
MA_IRQ_101	DSS_HDMI_IRQ	HDMI	Display subsystem HDMI interrupt <sup>(4)</sup>
MA_IRQ_102	SR_IVA_IRQ	SR_IVA	SmartReflex IVA interrupt
MA_IRQ_103	IVAHD_IRQ2	IVAHD	Sync interrupt from ICONT2 (vDMA) <sup>(4)</sup>
MA_IRQ_104	IVAHD_IRQ1	IVAHD	Sync interrupt from ICONT1 <sup>(4)</sup>
MA_IRQ_105	Reserved	Reserved	Reserved
MA_IRQ_106	Reserved	Reserved	Reserved
MA_IRQ_107	IVAHD_MAILBOX_IRQ_0	IVAHD	IVAHD mailbox interrupt 0
MA_IRQ_108	Reserved	Reserved	Reserved
MA_IRQ_109	MCASP1_AXINT	MCASP1	McASP1 transmit interrupt <sup>(2)</sup>
MA_IRQ_110	EMIF1_IRQ	EMIF1	EMIF1 interrupt
MA_IRQ_111	EMIF2_IRQ	EMIF2	EMIF2 interrupt
MA_IRQ_112	MCPDM_IRQ	MCPDM	MCPDM interrupt <sup>(2)</sup>
MA_IRQ_113	DMM_IRQ	DMM	DMM interrupt <sup>(3)</sup>
MA_IRQ_114	DMIC_IRQ	DMIC	DMIC interrupt <sup>(2)</sup>
MA_IRQ_115	Reserved	Reserved	Reserved
MA_IRQ_116	Reserved	Reserved	Reserved
MA_IRQ_117	Reserved	Reserved	Reserved
MA_IRQ_118	Reserved	Reserved	Reserved

**Table 17-2. Cortex-A9 MPU INTC Interrupt Mapping (continued)**

Interrupt Line (Destination)	Interrupt Line (Source)	Module Name	Description
MA_IRQ_119	sys_nirq2	External	External interrupt 2 (active low)
MA_IRQ_120	KBD_CTL_IRQ	KEYBOARD	Keyboard controller interrupt
MA_IRQ_121	Reserved	Reserved	Reserved
MA_IRQ_122	Reserved	Reserved	Reserved
MA_IRQ_123	Reserved	Reserved	Reserved
MA_IRQ_124	Reserved	Reserved	Reserved
MA_IRQ_125	Reserved	Reserved	Reserved
MA_IRQ_126	Reserved	Reserved	Reserved
MA_IRQ_127	Reserved	Reserved	Reserved

- (1) Internally generated within the Cortex-A9 MPU subsystem
- (2) Shared with DSP INTC
- (3) Shared with Cortex-M3 MPU INTC
- (4) Shared with DSP and Cortex-M3 MPU INTCs

**NOTE:** All Cortex-A9 MPUSS internal interrupts are active-high, level-sensitive.

### 17.3.3 Interrupt Requests to Cortex-M3 MPU INTC

Table 17-3 shows the Cortex-M3 MPU INTC interrupt mapping.

**Table 17-3. Cortex-M3 MPU INTC Interrupt Mapping**

Interrupt Line (Destination)	Interrupt Line (Source)	Module Name	Description
MM_IRQ_0	XLATE_MMU_FAULT	CORTEXM3	L2 cache MMU interrupt <sup>(1)</sup>
MM_IRQ_1	SHARED_CACHE_MMU_CPU_INT	CORTEXM3	Shared cache or MMU maintenance completed <sup>(1)</sup>
MM_IRQ_2	CTM_TIM_EVENT_1	CORTEXM3	CTM timer event (timer #1) <sup>(1)</sup>
MM_IRQ_3	HWSEM_M3_IRQ	CORTEXM3	Semaphore interrupt <sup>(1)</sup>
MM_IRQ_4	IC_NEMUINTR	CORTEXM3	ICECrusher (one to each core) <sup>(1)</sup>
MM_IRQ_5	IMP_FAULT	CORTEXM3	Imprecise fault (from interconnect) <sup>(1)</sup>
MM_IRQ_6	CTM_TIM_EVENT_2	CORTEXM3	CTM timer event (timer #2) <sup>(1)</sup>
MM_IRQ_7	DSS_DISPC_IRQ	Display controller	Display controller interrupt <sup>(4)</sup>
MM_IRQ_8	DSS_DSI1_IRQ	MIPI DSI1	Display subsystem DSI1 interrupt <sup>(4)</sup>
MM_IRQ_9	DSS_DSI2_IRQ	MIPI DSI2	Display subsystem DSI2 interrupt <sup>(4)</sup>
MM_IRQ_10	DSS_HDMI_IRQ	HDMI	Display subsystem HDMI interrupt <sup>(4)</sup>
MM_IRQ_11	ISS_IRQ0	Imaging subsystem	Imaging subsystem interrupt 0
MM_IRQ_12	ISS_IRQ1	Imaging subsystem	Imaging subsystem interrupt 1
MM_IRQ_13	ISS_IRQ2	Imaging subsystem	Imaging subsystem interrupt 2
MM_IRQ_14	ISS_IRQ3	Imaging subsystem	Imaging subsystem interrupt 3
MM_IRQ_15	ISS_IRQ4	Imaging subsystem	Imaging subsystem interrupt 4 <sup>(2)</sup>
MM_IRQ_16	ISS_IRQ5	Imaging subsystem	Imaging subsystem interrupt 5 <sup>(3)</sup>
MM_IRQ_17	FDIF_IRQ_1	Face detect	Face Detect interrupt 1
MM_IRQ_18	SDMA_IRQ_0	sDMA	sDMA interrupt 0 <sup>(4)</sup>
MM_IRQ_19	SDMA_IRQ_1	sDMA	sDMA interrupt 1 <sup>(4)</sup>
MM_IRQ_20	SDMA_IRQ_2	sDMA	sDMA interrupt 2 <sup>(3)</sup>
MM_IRQ_21	SDMA_IRQ_3	sDMA	sDMA interrupt 3 <sup>(3)</sup>
MM_IRQ_22	IVAHD_MAILBOX_IRQ_2	IVAHD	IVAHD mailbox interrupt 2

**Table 17-3. Cortex-M3 MPU INTC Interrupt Mapping (continued)**

<b>Interrupt Line (Destination)</b>	<b>Interrupt Line (Source)</b>	<b>Module Name</b>	<b>Description</b>
MM_IRQ_23	IWAHD_IRQ2	IWAHD	Sync interrupt from ICONT2 (vDMA) <sup>(4)</sup>
MM_IRQ_24	IWAHD_IRQ1	IWAHD	Sync interrupt from ICONT1 <sup>(4)</sup>
MM_IRQ_25	I2C1_IRQ	I2C1	I2C1 interrupt <sup>(3)</sup>
MM_IRQ_26	I2C2_IRQ	I2C2	I2C2 interrupt <sup>(3)</sup>
MM_IRQ_27	I2C3_IRQ	I2C3	I2C3 interrupt <sup>(3)</sup>
MM_IRQ_28	I2C4_IRQ	I2C4	I2C4 interrupt <sup>(3)</sup>
MM_IRQ_29	UART3_IRQ	UART3	UART3 interrupt <sup>(4)</sup>
MM_IRQ_30	Reserved	Reserved	Reserved
MM_IRQ_31	PRCM_M3_IRQ	PRCM	PRCM interrupt
MM_IRQ_32	Reserved	Reserved	Reserved
MM_IRQ_33	HSI_DMA_MPU_IRQ	HSI	HSI DMA engine MPU interrupt <sup>(3)</sup>
MM_IRQ_34	MAIL_U2_M3_IRQ	MAILBOX	Mailbox user 2 interrupt
MM_IRQ_35	GPIO1_MPU_IRQ	GPIO1	GPIO1 MPU interrupt <sup>(3)</sup>
MM_IRQ_36	GPIO2_MPU_IRQ	GPIO2	GPIO2 MPU interrupt <sup>(3)</sup>
MM_IRQ_37	GPT3_IRQ	GPTIMER3	GPTIMER3 interrupt <sup>(3)</sup>
MM_IRQ_38	GPT4_IRQ	GPTIMER4	GPTIMER4 interrupt <sup>(3)</sup>
MM_IRQ_39	GPT9_IRQ	GPTIMER9	GPTIMER9 interrupt <sup>(3)</sup>
MM_IRQ_40	GPT11_IRQ	GPTIMER11	GPTIMER11 interrupt <sup>(3)</sup>
MM_IRQ_41	MCSP11_IRQ	MCSP11	MCSP11 interrupt <sup>(4)</sup>
MM_IRQ_42	MCSP12_IRQ	MCSP12	MCSP12 interrupt <sup>(3)</sup>
MM_IRQ_43	Reserved	Reserved	Reserved
MM_IRQ_44	Reserved	Reserved	Reserved
MM_IRQ_45	Reserved	Reserved	Reserved
MM_IRQ_46	C2C_SSCM_IRQ0	C2C	C2C Status interrupt <sup>(4)</sup>
MM_IRQ_47	C2C_SSCM_IRQ1	C2C	C2C GPI interrupt <sup>(4)</sup>
MM_IRQ_48	DMM_IRQ	DMM	DMM interrupt <sup>(3)</sup>
MM_IRQ_49	Reserved	Reserved	Reserved
MM_IRQ_50	MMC1_IRQ	HSMMC1	MMC1 interrupt <sup>(4)</sup>
MM_IRQ_51	MMC2_IRQ	HSMMC2	MMC2 interrupt <sup>(4)</sup>
MM_IRQ_52	MMC3_IRQ	HSMMC3	MMC3 interrupt <sup>(3)</sup>
MM_IRQ_53	MMC4_IRQ	HSMMC4	MMC4 interrupt <sup>(3)</sup>
MM_IRQ_54	MMC5_IRQ	HSMMC5	MMC5 interrupt <sup>(3)</sup>
MM_IRQ_55	Reserved	Reserved	Reserved
MM_IRQ_56	FSUSB_SMI_IRQ	FSUSB	FS-USB - host controller SMI interrupt <sup>(3)</sup>
MM_IRQ_57	HSUSB_EHCI_IRQ	HSUSBHOST	HSUSB MP host interrupt EHCI controller <sup>(3)</sup>
MM_IRQ_58	HSUSB_TLL_IRQ	HSUSBTLL	HSUSB MP TLL interrupt <sup>(3)</sup>
MM_IRQ_59	FSUSB_IRQ	FSUSB	FS-USB - host controller Interrupt <sup>(3)</sup>
MM_IRQ_60	HSUSB_OTG_IRQ	HSUSBOTG	HSUSB OTG controller interrupt <sup>(3)</sup>
MM_IRQ_61	HSUSB_OTG_DMA_IRQ	HSUSBOTG	HSUSB OTG DMA interrupt <sup>(3)</sup>
MM_IRQ_62	HSI_P1_MPU_IRQ	HSI	HSI Port 1 interrupt <sup>(3)</sup>
MM_IRQ_63	HSI_P2_MPU_IRQ	HSI	HSI Port 2 interrupt <sup>(3)</sup>

(1) Internally generated within the Cortex-M3 MPU subsystem

(2) Shared with DSP INTC

(3) Shared with Cortex-A9 MPU INTC

(4) Shared with DSP and Cortex-A9 MPU INTCs

**NOTE:** MM\_IRQ\_0 triggers exception #16 inside the Cortex-M3 core(s).

---

**NOTE:** All Cortex-M3 MPUSS internal interrupts are edge interrupts, except the one generated from the MMU (XLATE\_MMU\_FAULT), which is level interrupt.

---

## 17.4 Interrupt Controllers Functional Description

### 17.4.1 Cortex-A9 MPU INTC Functional Description

Cortex-A9 MPU INTC is configured to have the following interrupt sources:

- 16 Interprocessor interrupts (IPI): ID[15:0]
- 2 private Timer/Watchdog interrupts: ID[30:29]
- 2 legacy nFIQ & nIRQ: one per CPU, bypasses the interrupt distributor logic and directly drives interrupt requests into CPU if used in legacy mode (else treated like other interrupts lines with ID28 and ID31 respectively)
- 128 hardware interrupts: ID[159:32] (rising-edge or high-level sensitive). The interrupt mapping of these interrupts is presented in [Table 17-2, Interrupt Requests to Cortex-A9 MPU INTC](#).

#### CAUTION

Although the hardware interrupts can be configured to be rising-edge or high-level sensitive in the INTC, the device supports only active-high, level-sensitive configuration.

For more information about the functionality of the Cortex-A9 MPU INTC, see *ARM Cortex-A9 MP Core Technical Reference Manual*.

### 17.4.2 Cortex-M3 MPU INTC Functional Description

The Cortex-M3 MPU INTC has the following configuration in the device:

- 64 hardware interrupts/16 levels of priority for each Cortex-M3 core. The interrupt mapping of these interrupts is presented in [Table 17-3, Interrupt Requests to Cortex-M3 MPU INTC](#).
- The calibration register, which is used to reload the SYSTICK timer to generate a fixed interval, is not supported in the Cortex-M3 core within the Cortex-M3 MPUSS. The TENMS field is not available due to the different frequencies during voltage scaling.
- The SYSRESETREQ reset feature is not supported at the Cortex-M3 processor level. Writing to this bit has no effect.

For more information about the functionality of the Cortex-M3 MPU INTC, see *ARM Cortex-M3 Technical Reference Manual*.

## 17.5 Interrupt Controllers Register Manual

### 17.5.1 Interrupt Controllers Instance Summary

For information about the Cortex-A9 MPU INTC and Cortex-M3 MPU INTC instances, see [Chapter 4, Cortex-A9 MPU Subsystem](#) and [Chapter 7, Dual Cortex-M3 Subsystem](#).

### 17.5.2 Cortex-A9 MPU INTC Registers

For information about the Cortex-A9 MPU INTC registers and their description, see *ARM Cortex-A9 MP Core Technical Reference Manual*.

### 17.5.3 Cortex-M3 MPU INTC Registers

For information about the Cortex-M3 MPU INTC registers and their description, see *ARM Cortex-M3 Technical Reference Manual*.

---

---

---

## Control Module

---

---

---

This chapter describes the system control module for the device.

Topic	Page
<b>18.1 Control Module Overview .....</b>	<b>3312</b>
<b>18.2 Control Module Environment .....</b>	<b>3314</b>
<b>18.3 Control Module Integration .....</b>	<b>3315</b>
<b>18.4 Control Module Functional Description .....</b>	<b>3318</b>
<b>18.5 Control Module Programming Guide .....</b>	<b>3425</b>
<b>18.6 Control Module Register Manual .....</b>	<b>3432</b>

## 18.1 Control Module Overview

The control module allows software control of the various static modes supported by the device. It is composed of two control submodules: General Control Module and Device (Padconfiguration) Control Module.

The General Control Module has the following features:

- General status and control
- Hardware observability input/output (I/O) multiplexing
- System boot decoding logic
- Standard eFuse logic

The Device Control Module provides the following features:

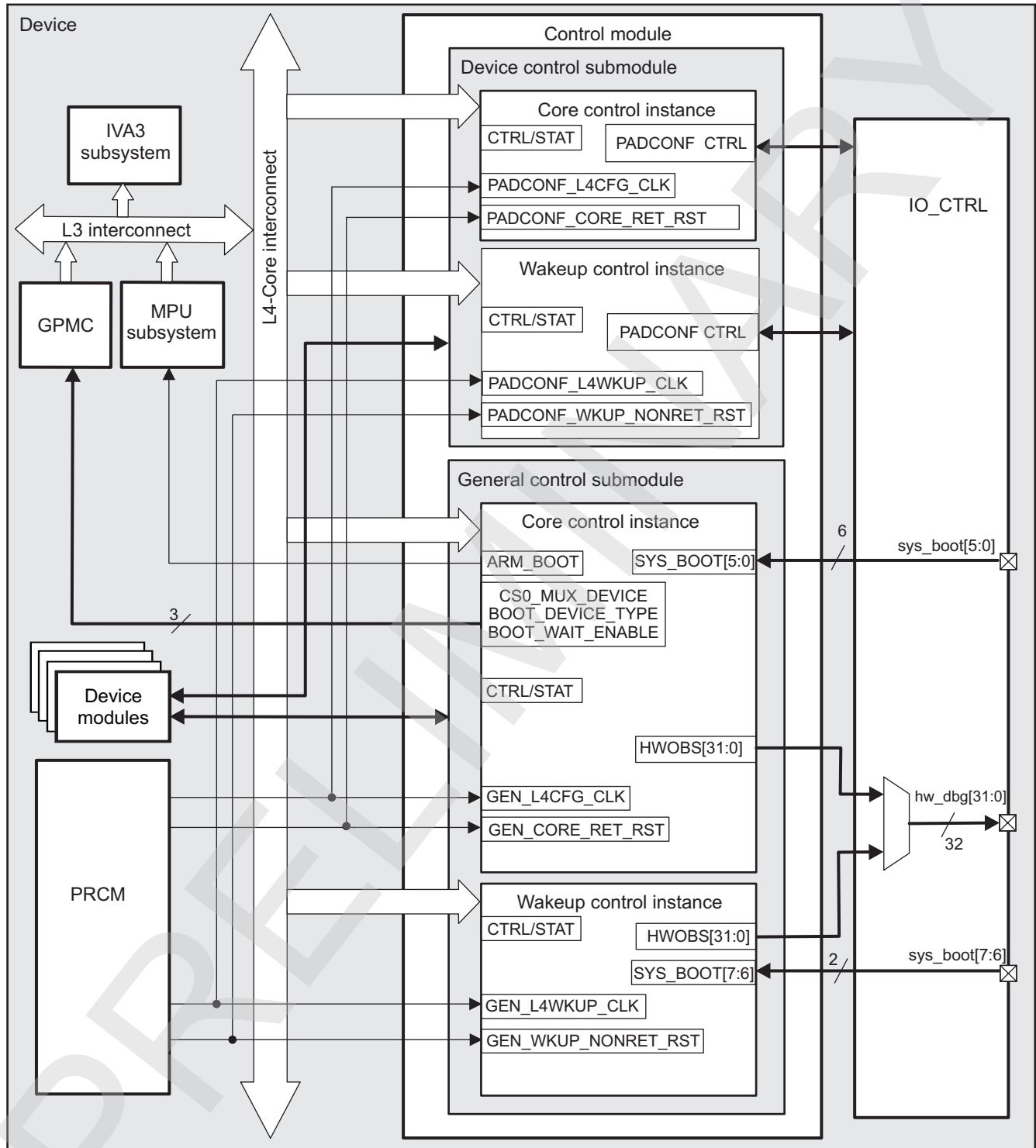
- Pad functional I/O multiplexing
- Pad pullup / pulldown configuration
- Pad groups associated signal integrity controls
- Complex I/O configuration
- Analog function control
- Device specific eFuse registers

Both control submodules are additionally split into two instances: the control core instance within the CORE power domain, and the control wake-up instance within the WKUP power domain. For more details on the four different control module instances, see [Section 18.3, Control Module Integration](#).

[Figure 18-1](#) shows the overview of the control module.



Figure 18-1. Control Module Overview



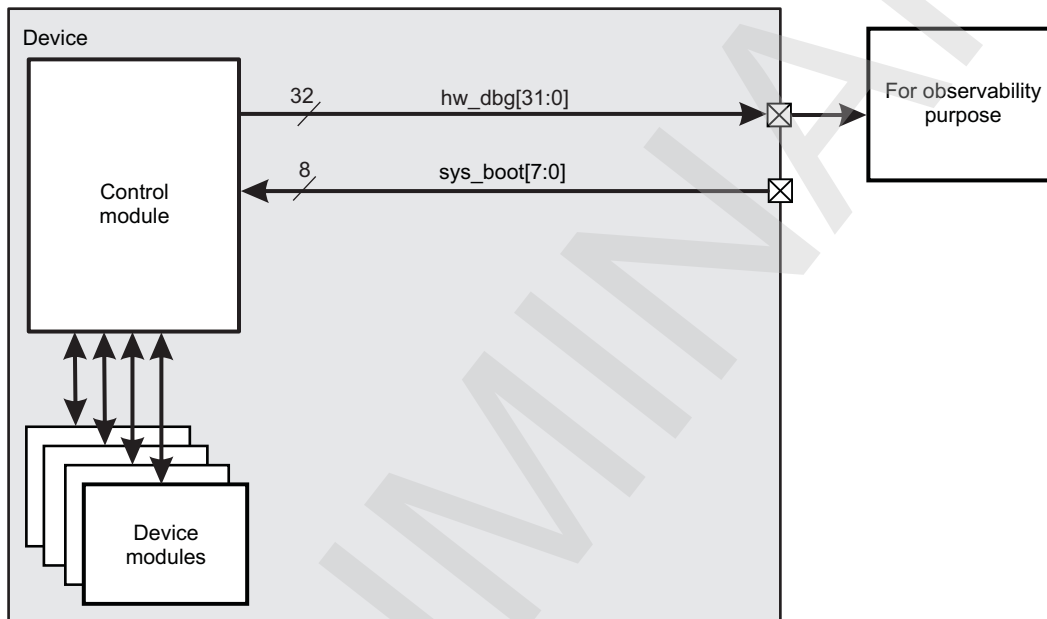
sysctrl-001

## 18.2 Control Module Environment

The control module integrates the system boot decoding logic accessible from eight external input pins (`sys_boot[7:0]`).

The control module controls the multiplexing of device internal module signals routed to the external pins (`hw_dbg[31:0]`) for hardware debug purposes. See [Section 18.4.11, Hardware Observability](#) and [Section 18.4.8, PAD Functional Multiplexing and Configuration](#) for more details. Using the correct pad configuration, the control module maps these pins at the device boundary to observe the internal signals from the power, reset and clock management (PRCM) modules signals, direct memory access (DMA) requests, and interrupts. [Figure 18-2](#) shows an overview of the control module environment.

**Figure 18-2. Control Module Environment**



sysctrl-002

### 18.2.1 Control Module Signals

[Table 18-1](#) lists the control module booting inputs and hardware observability outputs configured to observe device module debug signals. It describes the module signals and specifies their links to functions.

**Table 18-1. Control Module I/O Description**

Signal	I/O <sup>(1)</sup>	Description	Reset Value <sup>(2)</sup>
<code>hw_dbg[31:0]</code>	O	Internal hardware observability signals 0 to 31	HiZ
<code>sys_boot[7:0]</code>	I	Boot mode configuration signals 0 to 7	HiZ

<sup>(1)</sup> I = Input; O = Output

<sup>(2)</sup> HiZ = High impedance

### 18.3 Control Module Integration

The control module of the device is on the L4-Core interconnect. It consists of four module instances:

- General Core Control module (SYSCTRL\_GENERAL\_CORE)
- General Wakeup Control module (SYSCTRL\_GENERAL\_WKUP)
- Device Core Control module (SYSCTRL\_PADCONF\_CORE)
- Device Wakeup Control module (SYSCTRL\_PADCONF\_WKUP)

For more information about the different control module instances, see [Table 18-3](#).

Each of these four module instances has an associated register set and shares one functional clock and one reset signal with the *PRCM* module. These signals are described in [Table 18-2](#).

**Table 18-2. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SYSCTRL_GENERAL_WKUP	GEN_L4WKUP_CLK	WKUP_L4_ICLK2	PRCM	L4_WKUP interface/functional clock propagated to the General Wakeup Control module
SYSCTRL_GENERAL_CORE	GEN_L4CFG_CLK	CFG_L4_ICLK	PRCM	L4_CFG interface/functional clock propagated to the General Core Control module
SYSCTRL_PADCONF_WKUP	PADCONF_L4WKUP_CLK	WKUP_L4_ICLK2	PRCM	L4_WKUP interface/functional clock propagated to the Device Wakeup Control module
SYSCTRL_PADCONF_CORE	PADCONF_L4CFG_CLK	CFG_L4_ICLK	PRCM	L4_CFG interface/functional clock propagated to the Device Core Control module
Resets				
SYSCTRL_GENERAL_WKUP	GEN_WKUP_NONRET_RST	WKUP_PWRON_RST	PRCM	Internal power-on reset affecting General Wakeup Control module
SYSCTRL_GENERAL_CORE	GEN_CORE_RET_RST	CORE_PWRON_RET_RST	PRCM	Internal power-on reset affecting the General Core Control module
SYSCTRL_PADCONF_WKUP	PADCONF_WKUP_NONRET_RST	WKUP_PWRON_RST	PRCM	Internal power-on reset affecting the Device Wakeup Control module
SYSCTRL_PADCONF_CORE	PADCONF_CORE_RET_RST	CORE_PWRON_RET_RST	PRCM	Internal power-on reset affecting the Device Core Control module

The control module is reset by the internal power-on reset which is active low. The PRCM provides a retention power-on reset signal (CORE\_PWRON\_RET\_RST) duplicated to each of the two core control module instances, and a nonretention power-on reset signal (WKUP\_PWRON\_RST) duplicated to each of the two wake-up control module instances in the control module. For further details see [Section 3.5.4](#), *Reset Domains*, in [Chapter 3, Power, Reset, and Clock Management](#). The control module does not respond to a warm reset or to an L4-Core reset.

The main sequential logic within the system control module is accessible in a register file through the L4-Core. The PRCM module provides the CFG\_L4\_ICLK functional clock signal (L4\_CFG -clock domain) separately to each of the two core control submodules. See [Section 3.6.3.2.4, CM2 Clock Generator](#) in [Chapter 3, Power, Reset, and Clock Management](#) for further details on this signal.

The PRCM module also supplies the WKUP\_L4\_ICLK2 functional clock signal (L4\_WKUP -clock domain) to each of the two wake-up control module instances. See [Section 3.6.3.1, PRM Clock Source](#) in [Chapter 3, Power, Reset, and Clock Management](#) for further details on this signal.

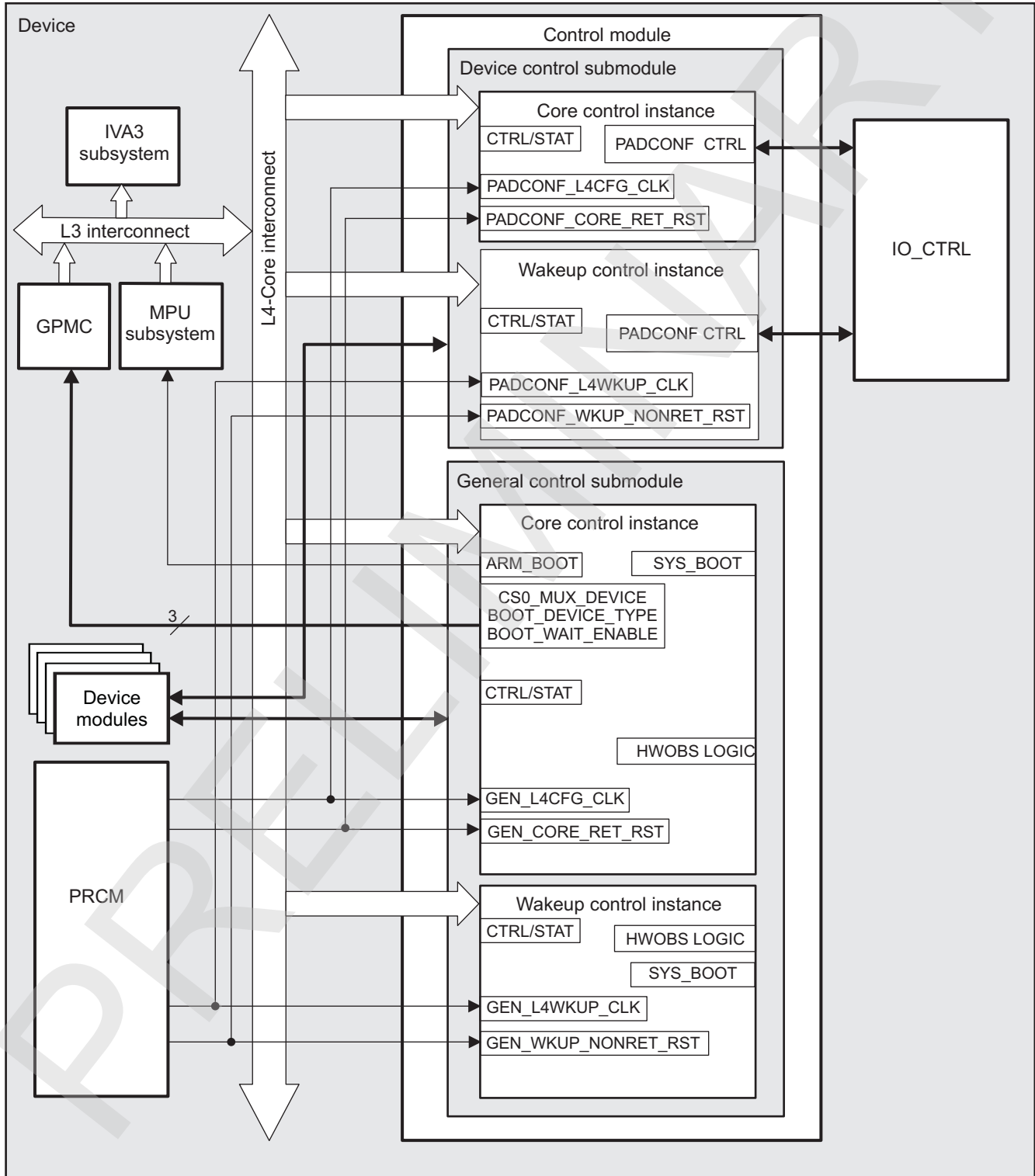
The signal integration of the control module features:

- No IDLE hardware handshake
- No STANDBY hardware handshake
- No SAVE and RESTORE hardware handshake
- No Wake-up request generation
- No DMA request generation
- No interrupt request (IRQ) generation

- Two functional clocks, one within the CFG- and the other within WKUP- clock domains
- One retention and one nonretention internal power-on based resets

Figure 18-3 summarizes the integration of the module in the device.

**Figure 18-3. Control Module Integration**



sysctrl-003

**Table 18-3. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
SYSCTRL_GENERAL_WKUP	PD_WKUP	L4_WKUP
SYSCTRL_GENERAL_CORE	PD_CORE	L4_CFG
SYSCTRL_PADCONF_WKUP	PD_WKUP	L4_WKUP
SYSCTRL_PADCONF_CORE	PD_CORE	L4_CFG

The software sets the configuration registers to the desired values depending on the configuration of the requested device. Static device configuration registers can be set by the software at any time and are effective immediately.

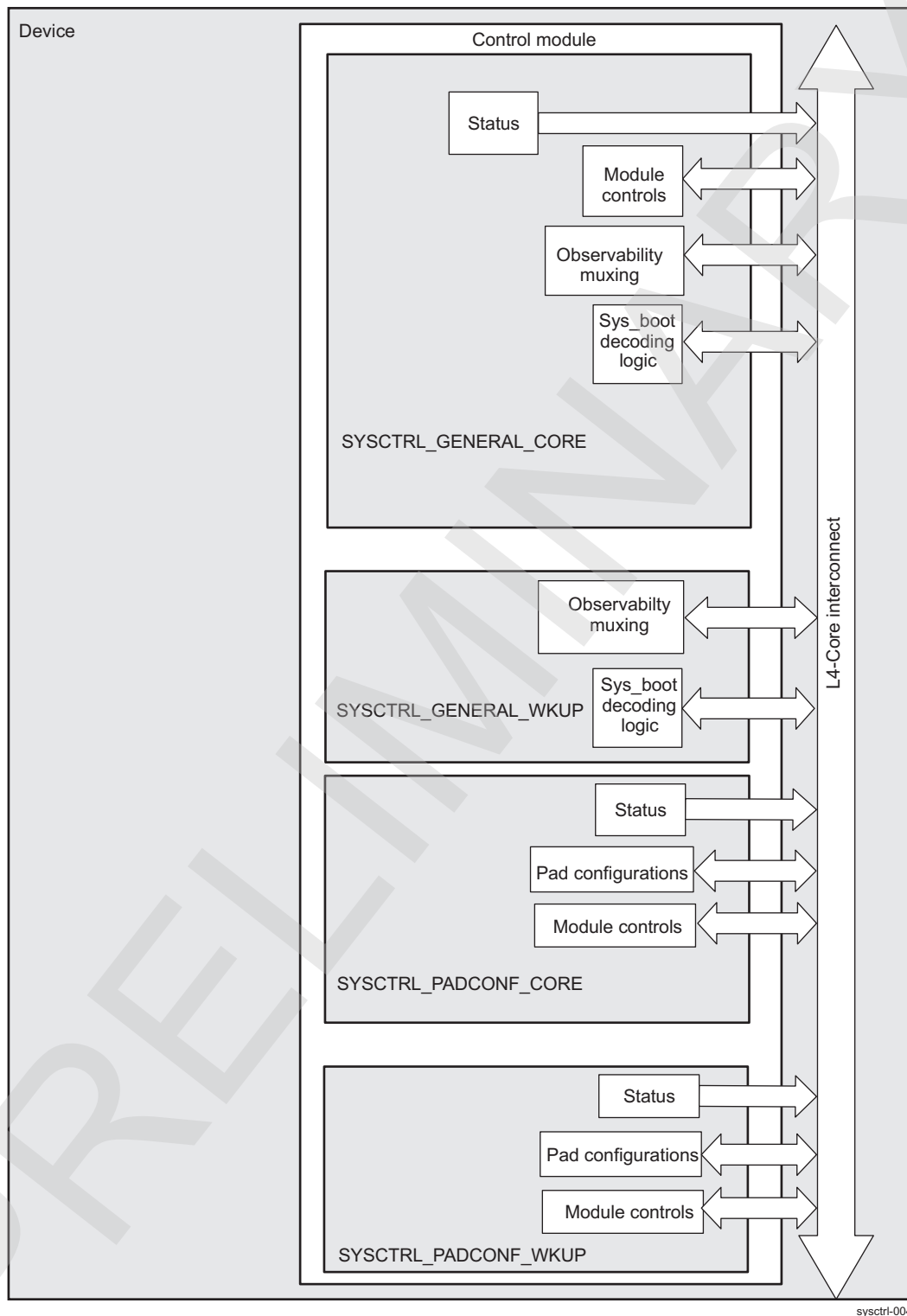
## 18.4 Control Module Functional Description

### 18.4.1 Control Module Block Diagram

The control module controls the settings of various device modules through register configuration and internal signals. It also controls the pad configuration, pin functional multiplexing and the routing of internal signals (such as PRCM signals or DMA requests) to output pins configured for hardware observability.

Figure 18-4 is the control module block diagram.

Figure 18-4. Control Module Block Diagram



Each of the control submodules primarily implements a bank of registers accessible (read/write) by the software along with some read only registers carrying status information.

Read/write registers available in the General Control Module can be divided into the following classes:

- Hardware observability registers (32-bit- read/write register)

- Status and configuration registers
- Boot registers
- eFuse ROM registers
- Miscellaneous functional registers

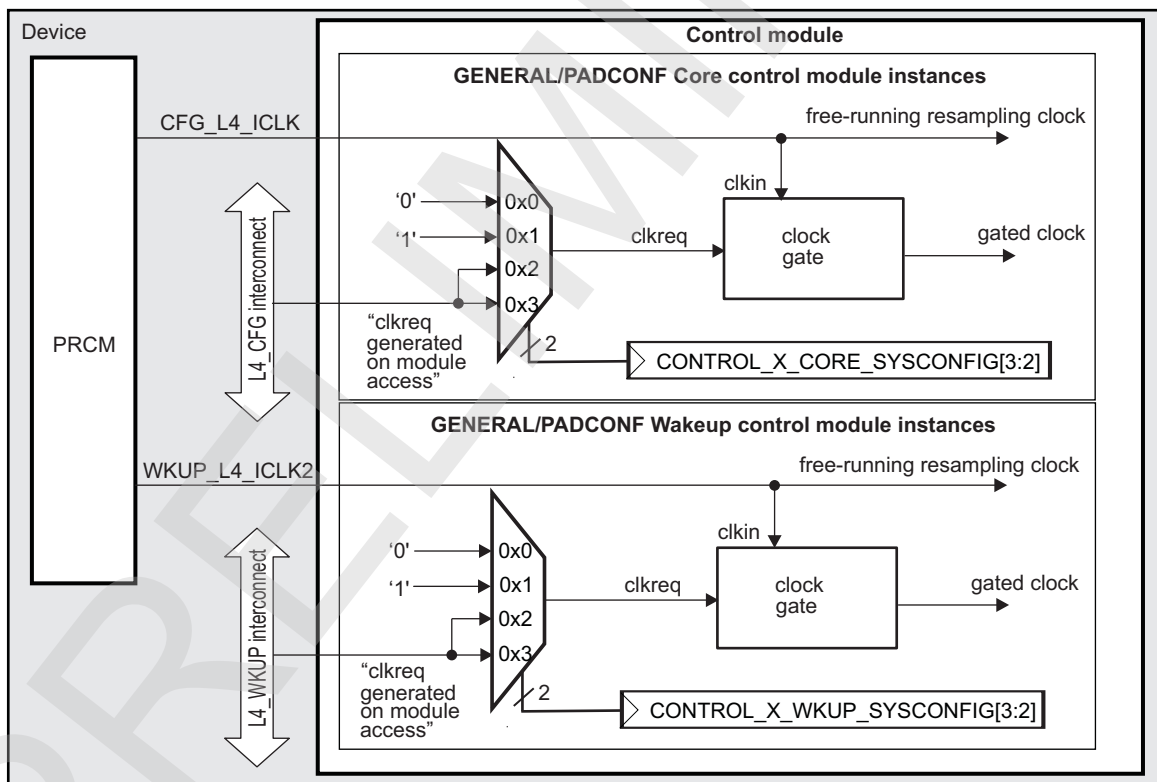
Read/write registers available in the Device(Padconfiguration) Control Module can be divided into the following classes:

- Pad configuration registers (32-bit- registers, one register for two pads)
- Pad groups (not individual per pad) associated registers for signal integrity parameters control (32-bit read/write registers )
- I/O cells associated specific control registers
- Device specific status and configuration registers
- Miscellaneous functional registers

### 18.4.2 Control Module Clock Configuration

Figure 18-5 shows the control module internal clock distribution schema. Each of the CFG\_L4\_ICLK and WKUP\_L4\_ICLK2 source clocks is internally distributed into two clock paths: the first one directly mirrors the source clock and is an always-running clock. It is used to synchronize and resample some of the module inputs. The other clock path can be internally gated within the module.

**Figure 18-5. Control Module Internal Clock Paths**



sysctrl-005

The clock gating strategy is selected through the bit field CONTROL\_SYSCONFIG[3:2] IP\_SYSCONFIG\_IDLEMODE available in each of the four Control Module instances. The different possibilities for clock management, are described in Table 18-4. There are no PRCM software controls for the Control Module clocks. There are however clock activity status bits located in PRCM. See Section 3.6.14.2, *Clock Domain Modes* and Section 3.6.4, *CD\_WKUP Clock Domain* in Chapter 3, *Power, Reset, and Clock Management*.



**Table 18-4. Internal Clock Gating Options**

Feature	CONTROL_SYSCONFIG[3:2] IP_SYSCONFIG_IDLEMODE value	Description
Local clock control	0b00	Clock is manually gated
Local clock control	0b01	Clock is manually enabled
Local clock autogating	0b10,0b11	Clock is automatically gated when there is no access to the Control Module through L4-interconnect

### 18.4.3 Control Module Software Reset

The control module is not sensitive to software reset.

### 18.4.4 Control Module Power Management

#### 18.4.4.1 Retention

The General- and Device Core Control module instances are fully built with retention flip-flop (RFF) and are powered in the core domain, so they can be in either active or retention mode. The core instances lose their content when the CORE power domain is in OFF mode. In contrast to the core control instances, the control wake-up instances are always powered and active. [Section 3.9, Device Low-Power States](#) in [Chapter 3, Power, Reset, and Clock Management](#) provides more information about the device low power states.

#### 18.4.4.2 Power Management Protocols

The control module, which is slave on the L4-interconnect, supports neither STANDBY nor IDLE mode protocols.

#### 18.4.4.3 Save and Restore (SAR) Mechanism

The control core module instances lose their contents when the CORE power domain is in OFF mode. To prevent this, before going to OFF mode, the core register context is copied by software into a dedicated SAR\_RAM sitting in the WKUP voltage domain. The save-and-restore (SAR) mechanism, controlled by the PRCM module, allows them to restore registers to a predefined value (not reset value, but value defined by the boot code). Thus core domain registers will have a correct configuration/state immediately after core power is on again without requesting a software operation to reconfigure them.

Because the SAR mechanism is managed by software-accessible registers, no hardware-specific requirements exist. For more details on SAR, see [Section 3.9.3, Device Off State Management](#) in [Chapter 3, Power, Reset, and Clock Management](#)

### 18.4.5 Hardware Requests

The control module does not generate interrupt, dma, or wake-up requests.

### 18.4.6 Control Module Initialization

The control module responds only to the internal power-on reset and device type. At power-on, reset values for the registers define the *safe state* for the chip. In the initialization mode, only modules to be used at boot time are associated to the pads. Other module inputs are internally tied and outputs pads are turned off. After power-on reset, the software is in charge of setting the pad functional multiplexing and configuration registers to the desired values according to the requested chip configuration. See [Chapter 27, Initialization](#), for further information.

General-purpose (GP) devices include features that are inaccessible or unavailable. These inaccessible registers define the default or fixed device configuration or behavior.

The [CONTROL\\_STATUS\[7:0\]](#) SYS\_BOOT reflects the state of the sys\_boot pins captured at power-on reset.

## 18.4.7 Control Module Instances

### 18.4.7.1 General Core Control Module Instance

The SYSCTRL\_GENERAL\_CORE instance contains most of the control logic providing hardware observability features for different DPLL clockout and DPLL -specific signals, DMA requests , IRQs, and other internal hardware signals. The General Core Control provides the system boot decoding logic and has control over the standard E-fuse settings. The system configuration registers, specific function controls, some peripheral control and status registers are also mapped there.

The register sets available within the SYSCTRL\_GENERAL\_CORE are:

- Observability control registers - [CONTROL\\_HWOBS\\_CONTROL](#), [CONTROL\\_DEBOBS\\_FINAL\\_MUX\\_SEL](#), [CONTROL\\_CONF\\_SDMA\\_REQ\\_SELx](#), [CONTROL\\_CONF\\_CLK\\_SELx](#) (x=0 to 2), [CONTROL\\_CONF\\_DPLL\\_X\\_SEL](#) ( where X is FREQLOCK, TINITZ, PHASELOCK, TENABLE, TENABLEDIV, BYPASSACK, IDLE)
- Part of the debug view channel selection registers - [CONTROL\\_CORE\\_CONF\\_DEBUG\\_SEL\\_TST\\_X](#)
- MMU address translation related registers - [CONTROL\\_CORTEX\\_M3\\_MMUADDRTRANSLTR](#), [CONTROL\\_CORTEX\\_M3\\_MMUADDRLOGICTR](#)
- Standard fuse registers - [CONTROL\\_STD\\_FUSE\\_X](#)
- Clock gating configuration register - [CONTROL\\_GEN\\_CORE\\_SYSCONFIG](#)
- Control status register - [CONTROL\\_STATUS](#)
- Peripheral controls - [CONTROL\\_TEMP\\_SENSOR](#), [CONTROL\\_DSS\\_CONTROL](#), [CONTROL\\_USBOTGHS\\_CONTROL](#), [CONTROL\\_LDOSRAM\\_MPU\\_VOLTAGE\\_CTRL](#), etc.

### 18.4.7.2 General Wakeup Control Module Instance

Associated registers are:

- Clock gating configuration register - [CONTROL\\_GEN\\_WKUP\\_SYSCONFIG](#)
- Register with spare bits - [CONTROL\\_OCPREG\\_SPARE](#)
- Part of the debug view channel selection registers - [CONTROL\\_WKUP\\_CONF\\_DEBUG\\_SEL\\_TST\\_X](#)

### 18.4.7.3 Device Core Control Module Instance

The SYSCTRL\_PADCONF\_CORE instance provides the device specific PAD control features like pullup and pulldown I/O configuration and pad functional multiplexing. Wake-up control/status features are also subject of the Device Core Control.

The registers associated to this module are:

- Clock gating configuration register - [CONTROL\\_PADCONF\\_CORE\\_SYSCONFIG](#)
- Different peripheral pad mode configuration and wake-up status/configuration - [CONTROL\\_CORE\\_PAD0\\_X\\_PAD1\\_Y](#) , [CONTROL\\_PADCONF\\_WAKEUPEVENTx](#) , [CONTROL\\_PADCONF\\_GLOBAL](#), [CONTROL\\_SMARTxIO](#), [CONTROL\\_C2CIO\\_PADCONF\\_0](#), etc.
- Peripheral pad voltage-level control registers - [CONTROL\\_PADCONF\\_MODE](#)
- Miscellaneous controls for different device interface modules - [CONTROL\\_X](#) registers (including [CONTROL\\_EFUSE](#) registers)

### 18.4.7.4 Device Wakeup Control Module Instance

Configuration registers for all pads linked to the wake-up voltage domain are in the WKUP device control module. The registers instantiated here are:

- Clock gating configuration register - [CONTROL\\_PADCONF\\_WKUP\\_SYSCONFIG](#)
- [CONTROL\\_WKUP\\_PAD0\\_X\\_PAD1\\_Y](#) registers for configuring FREF clock oscillator pads, reset signals and power mode pad configurations

---

**NOTE:** SYSCTRL\_PADCONF\_X alias notifies a Device Control module Instance.

---

Table 18-112 shows the memory size and base addresses assigned to the different control module instances

### 18.4.8 PAD Functional Multiplexing and Configuration

After power-on reset, the software has in charge to set the pad functional multiplexing and configuration registers to the desired values according to the requested chip configuration. The configuration is controlled by pads or by group of pads. Each configurable pin has its own configuration register for pullup/down control and for the assignment to a given module.

Two sets of 32-bit registers provide the PAD functional multiplexing and configuration feature of the control module. The first set of registers CONTROL\_CORE\_PAD0\_X\_PAD1\_Y provides the configuration of pads in the CORE power domain. The second set of registers CONTROL\_WKUP\_PAD0\_X\_PAD1\_Y configures pads in the WKUP power domain.

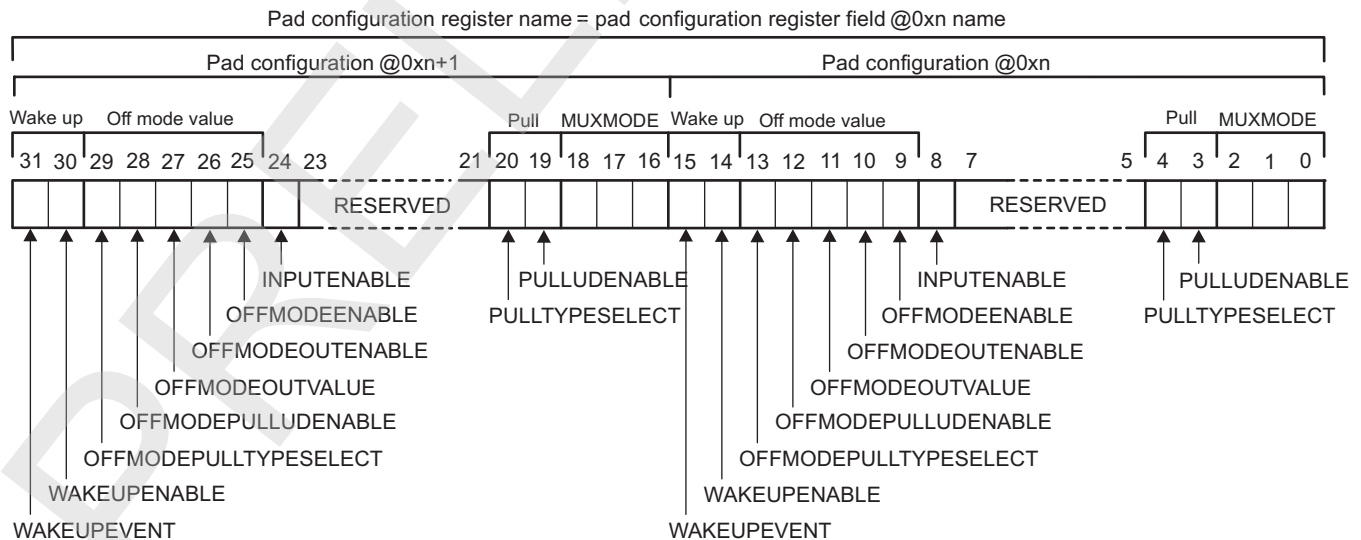
The pads configuration as defined in this module is used in the IO\_CTRL logic along with the gating and protection features. The Gating feature is implemented for the output pads. The Protection feature is implemented for input lines. These features implement additional control and multiplexing capabilities hard-wired in the IO\_CTRL logic and driven by internal device state and/or logic.

The gating feature allows forcing an output pad to a fixed value (Z, pullup, pulldown, 1, 0) or to a given internal signal. Each gating command is defined with a logic equation and the output state/value to be set on the pad when the logic equation is "True".

The protection feature provides control on each input line of the module when no pad is assigned to this input (protected value).When POR is released, the value on the pad is driven by the default configuration of control module. Most of time, this configuration is aligned with default value selected on the I/O cell. The protected value can be logical level (0 or 1) or a signal from a pad; it is selected when none of the pad\_x.MuxMod is selecting a chip input.

Each 32-bit PADCONF register controls two pads: the first pad is controlled by the 16 least significant bits (LSBs), and the second pad is controlled by the 16 most significant bits (MSBs), hence one pad configuration register provides control for two different pins. After power-on reset, the software sets the pad functional multiplexing and configuration registers to the requested device pad configurations. Data written in these registers directly command the multiplexing of the pad configuration logic. Figure 18-6 describes the associated to each pin 16-bit-wide pad configuration register field.

Figure 18-6. Pad Configuration Register Functionality



**NOTE:** These registers can be accessed using 8-, 16-, and 32-bit operations.

The functional bits of a pad configuration register field are divided into the following five fields:

- MUXMODE (3 bits) defines the multiplexing mode applied to the pin. A mode corresponds to the selection of the functionality mapped on the pin with seven (0 to 6) possible functional modes for each pin.
- PULL (2 bits) for combinational pullup/pulldown configuration:
  - PULLTYPESELECT: Pullup/pulldown type selection for the pin
  - PULLUDENABLE: Enables Pullup/pulldown feature for the pin
- INPUTENABLE (1 bit) drives an input enable signal to the I/O CTRL
  - INPUTENABLE = 0: Input disable. Pin is configured in output only mode
  - INPUTENABLE = 1: Input enable. Pin is configured in bidirectional mode
- Off mode values (1 bit) override the pin state when the OFFMODEENABLE bit in CONTROL\_CORE\_PAD0\_X\_PAD1\_Y is set and off mode is active. This feature allows having separate configurations (5 bits) for the pins when in off mode:
  - OFFMODEENABLE: Off mode pin state override control. Set to 1 to enable the feature and to 0 to disable it
  - OFFMODEOUTENABLE: Off mode output enable value. Set to 0 to enable the feature and to 1 to disable it
  - OFFMODEOUTVALUE: Off mode output value.
  - OFFMODEPULLUDENABLE: Off mode pullup/pulldown feature enable
  - OFFMODEPULLTYPESELECT: Off mode pullup/pulldown type selection
- Wake-up bits (2 bits):
  - WAKEUPENABLE: Enable wake-up detection on input. It is also the off mode input enable value.
  - WAKEUPEVENT: Wake-up event status for the pin.

**CAUTION**

Software must configure the OFF mode pads. It must ensure that the input/output capability is enabled for each pin.

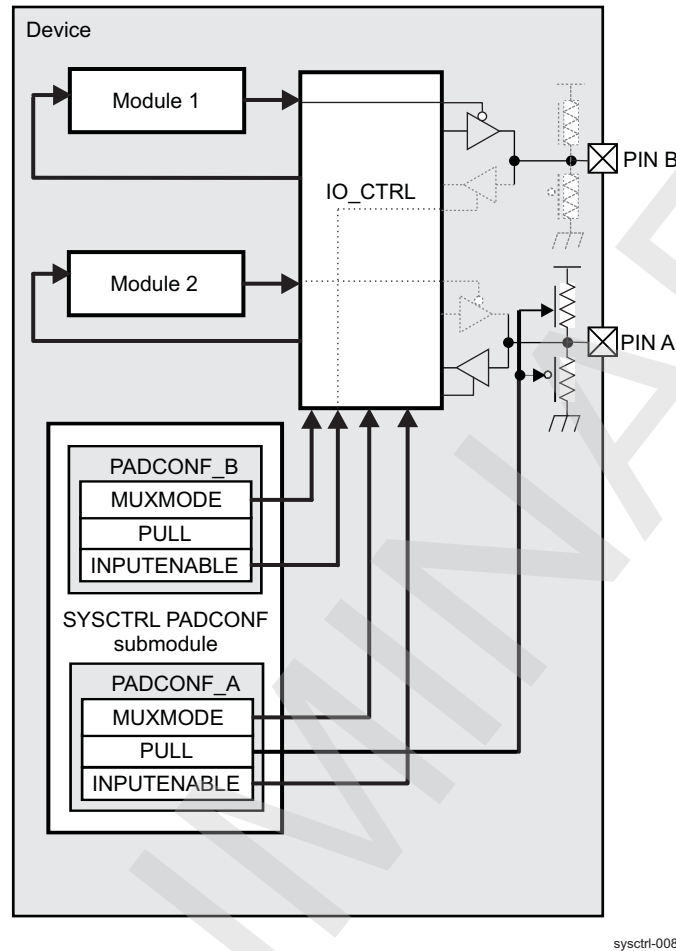
**CAUTION**

The OFFOUTENABLE and OFFOUTVALUE bits are functional only if the pad configuration supports output mode on at least one MUXMODE. For a pad that supports only the input feature, the OFFOUTENABLE and OFFOUTVALUE bits cannot be configured (they are don't care and read always returns 0).

**NOTE:** Because off-mode has impact only on pads in CORE domain, OFFMODE-related bits are not present in WKUP power domain padconfiguration registers.

Figure 18-7 shows the pad configuration functionality when off mode is inactive. For more information about off mode, see [Section 18.4.8.4, Off Mode](#).

Figure 18-7. Pad Configuration Diagram



The I/O pads are software-controlled by:

- Writing to the CONTROL\_(CORE/WKUP)\_PAD0\_X\_PAD1\_Y registers in the Device Control Module for input/output and pullup / pulldown configuration.
  - Writing to the GPIOi.GPIO\_OE registers in the GPIO module for input/output configuration.
- For more information about how to configure the I/O pads, see [Section 18.5.1.2.5, Pad Configuration Programming Points](#).

For more information about the GPIO module, see [Section 25.4.8, General-Purpose Interface Data Input/Output Capabilities](#).

**NOTE:** For a correct configuration of each pin direction (input, output, bidirectional), the CONTROL\_(CORE/WKUP)\_PAD0\_X\_PAD1\_Y and the GPIOi.GPIO\_OE registers must be written.

[Table 18-5](#) lists the bit directions of the CONTROL\_(CORE/WKUP)\_PAD0\_X\_PAD1\_Y registers.

Table 18-5. Bit Meanings for CONTROL\_PADCONF\_x Registers

Pad Configuration Bit	Bit Meaning	
	0	1
PULLUDENABLE	Not activated	Activated
PULLTYPESELECT	Pulldown	Pullup
INPUTENABLE	Input enable signal inactive	Input enable signal active

**Table 18-5. Bit Meanings for CONTROL\_PADCONF\_x Registers (continued)**

Pad Configuration Bit	Bit Meaning	
	0	1
OFFMODEENABLE	Off mode values are invalid.	Off mode values are valid.
OFFMODEOUTENABLE	Output	Input
OFFMODEOUTVALUE	Low	High
OFFMODEPULLUDENABLE	Not activated	Activated
OFFMODEPULLTYPESELECT	Pulldown	Pullup
WAKEUPENABLE	Disable I/O wake-up function.	Enable I/O wake-up function.
WAKEUPEVENT	Wake-up event not detected	Detect wake-up event.

### 18.4.8.1 Mode Selection

Table 18-6 lists the multiplexing modes and settings.

**Table 18-6. Mode Selection**

MUXMODE	Selected Mode
0b000	Primary mode = Mode 0
0b001	Mode 1
0b010	Mode 2
0b011	Mode 3
0b100	Mode 4
0b101	Mode 5
0b110	Mode 6
0b111	Safe mode = Mode 7

The MUXMODE field in the CONTROL\_CORE\_PAD0\_X\_PAD1\_Y and CONTROL\_WKUP\_PAD0\_X\_PAD1\_Y registers defines the multiplexing mode applied to the pad. Modes are referred to by their decimal (from 0 to 7) or binary (from 0b000 to 0b111) representation. Functional modes are defined from 0b000 to 0b110; mode 0b111 is referred to as the safe mode.

For most pads, the reset value for the MUXMODE field in the CONTROL\_CORE\_PAD0\_X\_PAD1\_Y and CONTROL\_WKUP\_PAD0\_X\_PAD1\_Y registers is 0b111. The exceptions are pads to be used at boot time to transfer data from selected peripherals to the external flash memory.

Mode 0 is the primary mode. When mode 0 is set, the function mapped to the pin corresponds to the name of the pin.

Mode 1 to mode 6 are possible modes for alternate functions. On each pin, some modes are used effectively for alternate functions, while other modes are unused and correspond to no functional configuration.

The safe mode avoids any risk of electrical contention by configuring the pin as an input with no functional interface mapped to it. The safe mode is used mainly as the default mode for all pins containing no mandatory interface at the release of power-on reset.

---

**NOTE:** When a pad is set into a mux mode which is not defined by pin multiplexing, this pad is actually set undriven (HiZ) with potential pullup/down.

---

For more information about the configurable mode on each pin, see [Table 18-9](#) and [Table 18-8](#).

---

**NOTE:** Pad names are signal names available in mode 0.

---

### 18.4.8.2 Pull Selection

Whichever pull value is configured, pulls are automatically disabled when a pin is configured as an output (see [Table 18-7](#)).

**Table 18-7. Pull Selection**

PULL		Pin Behavior
PULLTYPESELECT	PULLUDENABLE	
0b0	0b0	Pulldown selected but not activated
0b0	0b1	Pulldown selected and activated if pin is not configured as OUTPUT
0b1	0b0	Pullup selected but not activated
0b1	0b1	Pullup selected and activated if pin is not configured as OUTPUT

For more information on the pull available on each pin, see [Table 18-9](#) and [Table 18-8](#).

### 18.4.8.3 Pad Multiplexing Register Fields

[Table 18-9](#) and [Table 18-8](#) provide for each pad configuration register field the address offset and associated signal name for each multiplexing mode (as set by bit field MUXMODE). Mode 0 is always defined. Modes with no signal name are undefined for the given pad. For more information about default pad configuration settings (power-on reset values) for pads in the CORE power domain, see [Section 18.6.7](#), *SYSCTRL\_PADCONF\_CORE Register Description*; for pads in the WKUP power domain, see [Section 18.6.9](#), *SYSCTRL\_PADCONF\_WKUP Register Description*.

---

**NOTE:**

- [Table 18-9](#) lists the pad configuration registers instantiated in the CORE power domain that drive the pads in the CORE power domain. [Table 18-8](#) lists the pad configuration registers instantiated in the WKUP power domain that drive the pads in the WKUP power domain.
  - In [Table 18-9](#) and [Table 18-8](#), an empty cell indicates that the mode or pull is not available for this pin.
-

**Table 18-8. Device Wakeup Control Module Pad Configuration Register Fields**

<b>SYSCTRL_PADCONF_WKUP.[Register name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
CONTROL_WKUP_PAD0_GPIO_WK0_PAD1_GPIO_WK1[15:0]	0x040	reserved			gpio_wk0			hw_dbg1	safe_mode
CONTROL_WKUP_PAD0_GPIO_WK0_PAD1_GPIO_WK1[31:16]	0x040	reserved			gpio_wk1			hw_dbg2	safe_mode
CONTROL_WKUP_PAD0_GPIO_WK2_PAD1_GPIO_WK3[15:0]	0x044	reserved			gpio_wk2			hw_dbg3	safe_mode
CONTROL_WKUP_PAD0_GPIO_WK2_PAD1_GPIO_WK3[31:16]	0x044	reserved			gpio_wk3			hw_dbg4	safe_mode
CONTROL_WKUP_PAD0_GPIO_WK4_PAD1_SR_SCL[15:0]	0x048	reserved			gpio_wk4			hw_dbg5	safe_mode
CONTROL_WKUP_PAD0_GPIO_WK4_PAD1_SR_SCL[31:16]	0x048	sr_scl							
CONTROL_WKUP_PAD0_SR_SDA_PAD1_FREF_XTAL_IN[15:0]	0x04C	sr_sda							
CONTROL_WKUP_PAD0_SR_SDA_PAD1_FREF_XTAL_IN[31:16]	0x04C	fref_xtal_in				c2c_wakereqin			
CONTROL_WKUP_PAD0_FREQ_SLICER_IN_PAD1_FREQ_CLK_IOREQ[15:0]	0x050	fref_slicer_in			gpi_wk5	c2c_wakereqin			safe_mode
CONTROL_WKUP_PAD0_FREQ_SLICER_IN_PAD1_FREQ_CLK_IOREQ[31:16]	0x050	fref_clk_ioreq							
CONTROL_WKUP_PAD0_FREQ_CLK0_OUT_PAD1_FREQ_CLK3_REQ[15:0]	0x054	fref_clk0_out	fref_clk1_req	Reserved	gpio_wk6		sdmmc2_dat7	hw_dbg6	safe_mode
CONTROL_WKUP_PAD0_FREQ_CLK0_OUT_PAD1_FREQ_CLK3_REQ[31:16]	0x054	fref_clk3_req	fref_clk1_req	Reserved	gpio_wk30	c2c_wakereqin	sdmmc2_dat4	hw_dbg7	safe_mode
CONTROL_WKUP_PAD0_FREQ_CLK3_OUT_PAD1_FREQ_CLK4_REQ[15:0]	0x058	fref_clk3_out	fref_clk2_req	Reserved	gpio_wk31	c2c_wakereqout	sdmmc2_dat5	hw_dbg8	safe_mode
CONTROL_WKUP_PAD0_FREQ_CLK3_OUT_PAD1_FREQ_CLK4_REQ[31:16]	0x058	fref_clk4_req	fref_clk5_out		gpio_wk7		sdmmc2_dat6	hw_dbg9	



**Table 18-8. Device Wakeup Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_WKUP.[Register name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
CONTROL_WKUP_PAD0_FREF_CLK4_OUT_PAD1_SY S_32K[15:0]	0x05C	fref_clk4_out			gpio_wk8			hw_dbg10	
CONTROL_WKUP_PAD0_FREF_CLK4_OUT_PAD1_SY S_32K[31:16]	0x05C	sys_32k							
CONTROL_WKUP_PAD0_SYS_NRESPWRON_PAD1_S YS_NRESWARM[15:0]	0x060	sys_nrespwron							
CONTROL_WKUP_PAD0_SYS_NRESPWRON_PAD1_S YS_NRESWARM[31:16]	0x060	sys_nreswarm							
CONTROL_WKUP_PAD0_SYS_PWR_REQ_PAD1_SYS _PWRON_RESET_OUT[15:0]	0x064	sys_pwr_req							
CONTROL_WKUP_PAD0_SYS_PWR_REQ_PAD1_SYS _PWRON_RESET_OUT[31:16]	0x064	sys_pwron_reset_out			gpio_wk29			hw_dbg11	
CONTROL_WKUP_PAD0_SYS_BOOT6_PAD1_SYS_B OOT7[15:0]	0x068	sys_boot6	dpm_emu18		gpio_wk9	c2c_wakereqout		hw_dbg12	safe_mode
CONTROL_WKUP_PAD0_SYS_BOOT6_PAD1_SYS_B OOT7[31:16]	0x068	sys_boot7	dpm_emu19		gpio_wk10			hw_dbg13	safe_mode
CONTROL_WKUP_PAD0_JTAG_NTRST_PAD1_JTAG_ TCK[15:0]	0x06C	jtag_ntrst							
CONTROL_WKUP_PAD0_JTAG_NTRST_PAD1_JTAG_ TCK[31:16]	0x06C	jtag_tck							safe_mode
CONTROL_WKUP_PAD0_JTAG_RTCK_PAD1_JTAG_T MS_TMISC[15:0]	0x070	jtag_rtck							
CONTROL_WKUP_PAD0_JTAG_RTCK_PAD1_JTAG_T MS_TMISC[31:16]	0x070	jtag_tms_tmsc							safe_mode
CONTROL_WKUP_PAD0_JTAG_TDI_PAD1_JTAG_TD O[15:0]	0x074	jtag_tdi							

**Table 18-8. Device Wakeup Control Module Pad Configuration Register Fields (continued)**

SYSCTRL_PADCONF_WKUP.[Register name]	Address Offset	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_WKUP_PAD0_JTAG_TDI_PAD1_JTAG_TDO[31:16]	0x074	jtag_tdo							

**Table 18-9. Device Core Control Module Pad Configuration Register Fields**

SYSCTRL_PADCONF_CORE.[Register Name]	Address Offset	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_CORE_PAD0_GPMC_AD0_PAD1_GPMC_AD1[15:0]	0x040	gpmc_ad0	sdmmc2_dat0						
CONTROL_CORE_PAD0_GPMC_AD0_PAD1_GPMC_AD1[31:16]	0x040	gpmc_ad1	sdmmc2_dat1						
CONTROL_CORE_PAD0_GPMC_AD2_PAD1_GPMC_AD3[15:0]	0x044	gpmc_ad2	sdmmc2_dat2						
CONTROL_CORE_PAD0_GPMC_AD2_PAD1_GPMC_AD3[31:16]	0x044	gpmc_ad3	sdmmc2_dat3						
CONTROL_CORE_PAD0_GPMC_AD4_PAD1_GPMC_AD5[15:0]	0x048	gpmc_ad4	sdmmc2_dat4	sdmmc2_dir_dat0					
CONTROL_CORE_PAD0_GPMC_AD4_PAD1_GPMC_AD5[31:16]	0x048	gpmc_ad5	sdmmc2_dat5	sdmmc2_dir_dat1					
CONTROL_CORE_PAD0_GPMC_AD6_PAD1_GPMC_AD7[15:0]	0x04C	gpmc_ad6	sdmmc2_dat6	sdmmc2_dir_cmd					
CONTROL_CORE_PAD0_GPMC_AD6_PAD1_GPMC_AD7[31:16]	0x04C	gpmc_ad7	sdmmc2_dat7	sdmmc2_clk_fdbk					
CONTROL_CORE_PAD0_GPMC_AD8_PAD1_GPMC_AD9[15:0]	0x050	gpmc_ad8	kpd_row0	c2c_data15	gpio_32		sdmmc1_dat0		
CONTROL_CORE_PAD0_GPMC_AD8_PAD1_GPMC_AD9[31:16]	0x050	gpmc_ad9	kpd_row1	c2c_data14	gpio_33		sdmmc1_dat1		
CONTROL_CORE_PAD0_GPMC_AD10_PAD1_GPMC_AD11[15:0]	0x054	gpmc_ad10	kpd_row2	c2c_data13	gpio_34		sdmmc1_dat2		

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_GPMC_AD10_PAD1_GPMC_AD11[31:16]</a>	0x054	gpmc_ad11	kpd_row3	c2c_data12	gpio_35		sdmmc1_dat3		
<a href="#">CONTROL_CORE_PAD0_GPMC_AD12_PAD1_GPMC_AD13[15:0]</a>	0x058	gpmc_ad12	kpd_col0	c2c_data11	gpio_36		sdmmc1_dat4		
<a href="#">CONTROL_CORE_PAD0_GPMC_AD12_PAD1_GPMC_AD13[31:16]</a>	0x058	gpmc_ad13	kpd_col1	c2c_data10	gpio_37		sdmmc1_dat5		
<a href="#">CONTROL_CORE_PAD0_GPMC_AD14_PAD1_GPMC_AD15[15:0]</a>	0x05C	gpmc_ad14	kpd_col2	c2c_data9	gpio_38		sdmmc1_dat6		
<a href="#">CONTROL_CORE_PAD0_GPMC_AD14_PAD1_GPMC_AD15[31:16]</a>	0x05C	gpmc_ad15	kpd_col3	c2c_data8	gpio_39		sdmmc1_dat7		
<a href="#">CONTROL_CORE_PAD0_GPMC_A16_PAD1_GPMC_A17[15:0]</a>	0x060	gpmc_a16	kpd_row4	c2c_datain0	gpio_40	venc_656_data0			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_A16_PAD1_GPMC_A17[31:16]</a>	0x060	gpmc_a17	kpd_row5	c2c_datain1	gpio_41	venc_656_data1			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_A18_PAD1_GPMC_A19[15:0]</a>	0x064	gpmc_a18	kpd_row6	c2c_datain2	gpio_42	venc_656_data2			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_A18_PAD1_GPMC_A19[31:16]</a>	0x064	gpmc_a19	kpd_row7	c2c_datain3	gpio_43	venc_656_data3			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_A20_PAD1_GPMC_A21[15:0]</a>	0x068	gpmc_a20	kpd_col4	c2c_datain4	gpio_44	venc_656_data4			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_A20_PAD1_GPMC_A21[31:16]</a>	0x068	gpmc_a21	kpd_col5	c2c_datain5	gpio_45	venc_656_data5			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_A22_PAD1_GPMC_A23[15:0]</a>	0x06C	gpmc_a22	kpd_col6	c2c_datain6	gpio_46	venc_656_data6			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_A22_PAD1_GPMC_A23[31:16]</a>	0x06C	gpmc_a23	kpd_col7	c2c_datain7	gpio_47	venc_656_data7			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_A24_PAD1_GPMC_A25[15:0]</a>	0x070	gpmc_a24	kpd_col8	c2c_clkout0	gpio_48				safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
CONTROL_CORE_PAD0_G PMC_A24_PAD1_GPMC_A2 5[31:16]	0x070	gpmc_a25		c2c_ clkout1	gpio_49				safe_mode
CONTROL_CORE_PAD0_G PMC_NCS0_PAD1_GPMC_ NCS1[15:0]	0x074	gpmc_ncs0			gpio_50	sys_ndmareq0			
CONTROL_CORE_PAD0_G PMC_NCS0_PAD1_GPMC_ NCS1[31:16]	0x074	gpmc_ncs1		c2c_ dataout6	gpio_51				safe_mode
CONTROL_CORE_PAD0_G PMC_NCS2_PAD1_GPMC_ NCS3[15:0]	0x078	gpmc_ncs2	kpd_row8	c2c_ dataout7	gpio_52				safe_mode
CONTROL_CORE_PAD0_G PMC_NCS2_PAD1_GPMC_ NCS3[31:16]	0x078	gpmc_ncs3	gpmc_dir	c2c_ dataout4	gpio_53				safe_mode
CONTROL_CORE_PAD0_G PMC_NWP_PAD1_GPMC_C LK[15:0]	0x07C	gpmc_nwp	dsi1_te0		gpio_54	sys_ndmareq1			
CONTROL_CORE_PAD0_G PMC_NWP_PAD1_GPMC_C LK[31:16]	0x07C	gpmc_clk			gpio_55	sys_ndmareq2	sdmmc1_ cmd		
CONTROL_CORE_PAD0_G PMC_NADV_ALE_PAD1_GP MC_NOE[15:0]	0x080	gpmc_nadv_ale	dsi1_te1		gpio_56	sys_ndmareq3	sdmmc1_ clk		
CONTROL_CORE_PAD0_G PMC_NADV_ALE_PAD1_GP MC_NOE[31:16]	0x080	gpmc_noe	sdmmc2_ clk						
CONTROL_CORE_PAD0_G PMC_NWE_PAD1_GPMC_N BE0_CLE[15:0]	0x084	gpmc_nwe	sdmmc2_ cmd						
CONTROL_CORE_PAD0_G PMC_NWE_PAD1_GPMC_N BE0_CLE[31:16]	0x084	gpmc_nbe0_cle	dsi2_te0		gpio_59				
CONTROL_CORE_PAD0_G PMC_NBE1_PAD1_GPMC_ WAIT0[15:0]	0x088	gpmc_nbe1		c2c_ dataout5	gpio_60				safe_mode
CONTROL_CORE_PAD0_G PMC_NBE1_PAD1_GPMC_ WAIT0[31:16]	0x088	gpmc_wait0	dsi2_te1		gpio_61				
CONTROL_CORE_PAD0_G PMC_WAIT1_PAD1_GPMC_ WAIT2[15:0]	0x08C	gpmc_wait1		c2c_ dataout2	gpio_62				safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_GPMC_WAIT1_PAD1_GPMC_WAIT2[31:16]</a>	0x08C	gpmc_wait2	usbc1_ icusb_txen	c2c_ dataout3	gpio_100	sys_ndmareq0			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_NCS4_PAD1_GPMC_NCS5[15:0]</a>	0x090	gpmc_ncs4	dsi1_te0	c2c_clkin0	gpio_101	sys_ndmareq1			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_NCS4_PAD1_GPMC_NCS5[31:16]</a>	0x090	gpmc_ncs5	dsi1_te1	c2c_clkin1	gpio_102	sys_ndmareq2			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_NCS6_PAD1_GPMC_NCS7[15:0]</a>	0x094	gpmc_ncs6	dsi2_te0	c2c_ dataout0	gpio_103	sys_ndmareq3			safe_mode
<a href="#">CONTROL_CORE_PAD0_GPMC_NCS6_PAD1_GPMC_NCS7[31:16]</a>	0x094	gpmc_ncs7	dsi2_te1	c2c_ dataout1	gpio_104				safe_mode
<a href="#">CONTROL_CORE_PAD0_GPI063_PAD1_GPI064[15:0]</a>	0x098	reserved			gpio_63				safe_mode
<a href="#">CONTROL_CORE_PAD0_GPI063_PAD1_GPI064[31:16]</a>	0x098	reserved			gpio_64				safe_mode
<a href="#">CONTROL_CORE_PAD0_GPI065_PAD1_GPI066[15:0]</a>	0x09C	reserved			gpio_65				safe_mode
<a href="#">CONTROL_CORE_PAD0_GPI065_PAD1_GPI066[31:16]</a>	0x09C	reserved			gpio_66				safe_mode
<a href="#">CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[15:0]</a>	0x0A0	csi21_dx0			gpi_67				safe_mode
<a href="#">CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0[31:16]</a>	0x0A0	csi21_dy0			gpi_68				safe_mode
<a href="#">CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[15:0]</a>	0x0A4	csi21_dx1			gpi_69				safe_mode
<a href="#">CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1[31:16]</a>	0x0A4	csi21_dy1			gpi_70				safe_mode
<a href="#">CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[15:0]</a>	0x0A8	csi21_dx2			gpi_71				safe_mode
<a href="#">CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2[31:16]</a>	0x0A8	csi21_dy2			gpi_72				safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_C SI21_DX3_PAD1_CSI21_DY3[15:0]</a>	0x0AC	csi21_dx3			gpi_73				safe_mode
<a href="#">CONTROL_CORE_PAD0_C SI21_DX3_PAD1_CSI21_DY3[31:16]</a>	0x0AC	csi21_dy3			gpi_74				safe_mode
<a href="#">CONTROL_CORE_PAD0_C SI21_DX4_PAD1_CSI21_DY4[15:0]</a>	0x0B0	csi21_dx4			gpi_75				safe_mode
<a href="#">CONTROL_CORE_PAD0_C SI21_DX4_PAD1_CSI21_DY4[31:16]</a>	0x0B0	csi21_dy4			gpi_76				safe_mode
<a href="#">CONTROL_CORE_PAD0_C SI22_DX0_PAD1_CSI22_DY0[15:0]</a>	0x0B4	csi22_dx0			gpi_77				safe_mode
<a href="#">CONTROL_CORE_PAD0_C SI22_DX0_PAD1_CSI22_DY0[31:16]</a>	0x0B4	csi22_dy0			gpi_78				safe_mode
<a href="#">CONTROL_CORE_PAD0_C SI22_DX1_PAD1_CSI22_DY1[15:0]</a>	0x0B8	csi22_dx1			gpi_79				safe_mode
<a href="#">CONTROL_CORE_PAD0_C SI22_DX1_PAD1_CSI22_DY1[31:16]</a>	0x0B8	csi22_dy1			gpi_80				safe_mode
<a href="#">CONTROL_CORE_PAD0_C AM_SHUTTER_PAD1_CAM_STROBE[15:0]</a>	0x0BC	cam_shutter			gpio_81				safe_mode
<a href="#">CONTROL_CORE_PAD0_C AM_SHUTTER_PAD1_CAM_STROBE[31:16]</a>	0x0BC	cam_strobe			gpio_82				safe_mode
<a href="#">CONTROL_CORE_PAD0_C AM_GLOBALRESET_PAD1_USBB1_ULPITLL_CLK[15:0]</a>	0x0C0	cam_globalreset			gpio_83				safe_mode
<a href="#">CONTROL_CORE_PAD0_C AM_GLOBALRESET_PAD1_USBB1_ULPITLL_CLK[31:16]</a>	0x0C0	usbb1_ulpitll_clk	hsi1_cawake		gpio_84	usbb1_ulpiphy_clk		hw_dbg20	safe_mode
<a href="#">CONTROL_CORE_PAD0_U SBB1_ULPITLL_STP_PAD1_USBB1_ULPITLL_DIR[15:0]</a>	0x0C4	usbb1_ulpitll_stp	hsi1_cadata	mcbasp4_clkr	gpio_85	usbb1_ulpiphy_stp	usbb1_mm_rxdp	hw_dbg21	safe_mode
<a href="#">CONTROL_CORE_PAD0_U SBB1_ULPITLL_STP_PAD1_USBB1_ULPITLL_DIR[31:16]</a>	0x0C4	usbb1_ulpitll_dir	hsi1_caflag	mcbasp4_fsr	gpio_86	usbb1_ulpiphy_dir		hw_dbg22	safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

SYSCTRL_PADCONF_COR E.[Register Name]	Address Offset	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_CORE_PAD0_U SBB1_ULPITLL_NEXT_PAD1_USBB1_ULPITLL_DAT0[15:0]	0x0C8	usbb1_ulpitll_nxt	hsi1_acready	mcbsp4_fsx	gpio_87	usbb1_ulpiphy_nxt	usbb1_mm_rxdm	hw_dbg23	safe_mode
CONTROL_CORE_PAD0_U SBB1_ULPITLL_NEXT_PAD1_USBB1_ULPITLL_DAT0[31:16]	0x0C8	usbb1_ulpitll_dat0	hsi1_acwake	mcbsp4_clkx	gpio_88	usbb1_ulpiphy_dat0	usbb1_mm_txen	hw_dbg24	safe_mode
CONTROL_CORE_PAD0_U SBB1_ULPITLL_DAT1_PAD1_USBB1_ULPITLL_DAT2[15:0]	0x0CC	usbb1_ulpitll_dat1	hsi1_acdata	mcbsp4_dx	gpio_89	usbb1_ulpiphy_dat1	usbb1_mm_txdm	hw_dbg25	safe_mode
CONTROL_CORE_PAD0_U SBB1_ULPITLL_DAT1_PAD1_USBB1_ULPITLL_DAT2[31:16]	0x0CC	usbb1_ulpitll_dat2	hsi1_aclflag	mcbsp4_dr	gpio_90	usbb1_ulpiphy_dat2	usbb1_mm_txse0	hw_dbg26	safe_mode
CONTROL_CORE_PAD0_U SBB1_ULPITLL_DAT3_PAD1_USBB1_ULPITLL_DAT4[15:0]	0x0D0	usbb1_ulpitll_dat3	hsi1_caready		gpio_91	usbb1_ulpiphy_dat3	usbb1_mm_rxcv	hw_dbg27	safe_mode
CONTROL_CORE_PAD0_U SBB1_ULPITLL_DAT3_PAD1_USBB1_ULPITLL_DAT4[31:16]	0x0D0	usbb1_ulpitll_dat4	dmtimer8_pwm_evt	abe_mcbasp3_dr	gpio_92	usbb1_ulpiphy_dat4		hw_dbg28	safe_mode
CONTROL_CORE_PAD0_U SBB1_ULPITLL_DAT5_PAD1_USBB1_ULPITLL_DAT6[15:0]	0x0D4	usbb1_ulpitll_dat5	dmtimer9_pwm_evt	abe_mcbasp3_dx	gpio_93	usbb1_ulpiphy_dat5		hw_dbg29	safe_mode
CONTROL_CORE_PAD0_U SBB1_ULPITLL_DAT5_PAD1_USBB1_ULPITLL_DAT6[31:16]	0x0D4	usbb1_ulpitll_dat6	dmtimer10_pwm_evt	abe_mcbasp3_clkx	gpio_94	usbb1_ulpiphy_dat6	abe_dmic_din3	hw_dbg30	safe_mode
CONTROL_CORE_PAD0_U SBB1_ULPITLL_DAT7_PAD1_USBB1_HSIC_DATA[15:0]	0x0D8	usbb1_ulpitll_dat7	dmtimer11_pwm_evt	abe_mcbasp3_fsx	gpio_95	usbb1_ulpiphy_dat7	abe_dmic_clk3	hw_dbg31	safe_mode
CONTROL_CORE_PAD0_U SBB1_ULPITLL_DAT7_PAD1_USBB1_HSIC_DATA[31:16]	0x0D8	usbb1_hsic_data			gpio_96				safe_mode
CONTROL_CORE_PAD0_U SBB1_HSIC_STROBE_PAD1_USBC1_ICUSB_DP[15:0]	0x0DC	usbb1_hsic_strobe			gpio_97				safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
CONTROL_CORE_PAD0_U SBB1_HSIC_STROBE_PAD1 _USBC1_ICUSB_DP[31:16]	0x0DC	usbc1_ icusb_dp			gpio_98				safe_mode
CONTROL_CORE_PAD0_U SBC1_ICUSB_DM_PAD1_S DMMC1_CLK[15:0]	0x0E0	usbc1_ icusb_dm			gpio_99				safe_mode
CONTROL_CORE_PAD0_U SBC1_ICUSB_DM_PAD1_S DMMC1_CLK[31:16]	0x0E0	sdmmc1_ clk		dpm_emu19	gpio_100				safe_mode
CONTROL_CORE_PAD0_S DMMC1_CMD_PAD1_SDMM C1_DAT0[15:0]	0x0E4	sdmmc1_ cmd		uart1_rx	gpio_101				safe_mode
CONTROL_CORE_PAD0_S DMMC1_CMD_PAD1_SDMM C1_DAT0[31:16]	0x0E4	sdmmc1_ dat0		dpm_emu18	gpio_102				safe_mode
CONTROL_CORE_PAD0_S DMMC1_DAT1_PAD1_SDM MC1_DAT2[15:0]	0x0E8	sdmmc1_ dat1		dpm_emu17	gpio_103				safe_mode
CONTROL_CORE_PAD0_S DMMC1_DAT1_PAD1_SDM MC1_DAT2[31:16]	0x0E8	sdmmc1_ dat2		dpm_emu16	gpio_104	jtag_ tms_tmsc			safe_mode
CONTROL_CORE_PAD0_S DMMC1_DAT3_PAD1_SDM MC1_DAT4[15:0]	0x0EC	sdmmc1_ dat3		dpm_emu15	gpio_105	jtag_tck			safe_mode
CONTROL_CORE_PAD0_S DMMC1_DAT3_PAD1_SDM MC1_DAT4[31:16]	0x0EC	sdmmc1_ dat4			gpio_106				safe_mode
CONTROL_CORE_PAD0_S DMMC1_DAT5_PAD1_SDM MC1_DAT6[15:0]	0x0F0	sdmmc1_ dat5			gpio_107				safe_mode
CONTROL_CORE_PAD0_S DMMC1_DAT5_PAD1_SDM MC1_DAT6[31:16]	0x0F0	sdmmc1_ dat6			gpio_108				safe_mode
CONTROL_CORE_PAD0_S DMMC1_DAT7_PAD1_ABE_ MCBSP2_CLKX[15:0]	0x0F4	sdmmc1_ dat7			gpio_109				safe_mode
CONTROL_CORE_PAD0_S DMMC1_DAT7_PAD1_ABE_ MCBSP2_CLKX[31:16]	0x0F4	abe_ mcbbsp2_clkx	mcspi2_clk	abe_ mccasp_ahclkx	gpio_110	usbb2_ mm_rxdm			safe_mode
CONTROL_CORE_PAD0_A BE_MCBSP2_DR_PAD1_AB E_MCBSP2_DX[15:0]	0x0F8	abe_ mcbbsp2_dr	mcspi2_ somi	abe_ mccasp_axr	gpio_111	usbb2_ mm_rxdp			safe_mode



**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_ABE_MCBSP2_DR_PAD1_ABE_MCBSP2_DX[31:16]</a>	0x0F8	abe_mcbbsp2_dx	mcspi2_simo	abe_mcasp_amute	gpio_112	usbb2_mm_rxcv			safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_MCBSP2_FSX_PAD1_ABE_MCBSP1_CLKX[15:0]</a>	0x0FC	abe_mcbbsp2_fsx	mcspi2_cs0	abe_mcasp_afsx	gpio_113	usbb2_mm_txen			safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_MCBSP2_FSX_PAD1_ABE_MCBSP1_CLKX[31:16]</a>	0x0FC	abe_mcbbsp1_clkx	abe_slimbus1_clock		gpio_114				safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_MCBSP1_DR_PAD1_ABE_MCBSP1_DX[15:0]</a>	0x100	abe_mcbbsp1_dr	abe_slimbus1_data		gpio_115				safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_MCBSP1_DR_PAD1_ABE_MCBSP1_DX[31:16]</a>	0x100	abe_mcbbsp1_dx	sdmmc3_dat2	abe_mcasp_aclkx	gpio_116				safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_MCBSP1_FSX_PAD1_ABE_PDM_UL_DATA[15:0]</a>	0x104	abe_mcbbsp1_fsx	sdmmc3_dat3	abe_mcasp_amutein	gpio_117				safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_MCBSP1_FSX_PAD1_ABE_PDM_UL_DATA[31:16]</a>	0x104	abe_pdm_ul_data	abe_mcbbsp3_dr						safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_PDM_DL_DATA_PAD1_ABE_PDM_FRAME[15:0]</a>	0x108	abe_pdm_dl_data	abe_mcbbsp3_dx						safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_PDM_DL_DATA_PAD1_ABE_PDM_FRAME[31:16]</a>	0x108	abe_pdm_frame	abe_mcbbsp3_clkx						safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_PDM_LB_CLK_PAD1_ABE_CLKS[15:0]</a>	0x10C	abe_pdm_lb_clk	abe_mcbbsp3_fsx						safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_PDM_LB_CLK_PAD1_ABE_CLKS[31:16]</a>	0x10C	abe_clks			gpio_118				safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_DMIC_CLK1_PAD1_ABE_DMIC_DIN1[15:0]</a>	0x110	abe_dmic_clk1			gpio_119	usbb2_mm_txse0	uart4_cts		safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_DMIC_CLK1_PAD1_ABE_DMIC_DIN1[31:16]</a>	0x110	abe_dmic_din1			gpio_120	usbb2_mm_txdat	uart4_rts		safe_mode
<a href="#">CONTROL_CORE_PAD0_ABE_DMIC_DIN2_PAD1_ABE_DMIC_DIN3[15:0]</a>	0x114	abe_dmic_din2	slimbus2_clock	abe_mcasp_axr	gpio_121		dmtimer11_pwm_evt		safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCONFIG_PADCONF_CORE.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_ABE_DMIC_DIN2_PAD1_ABE_DMIC_DIN3[31:16]</a>	0x114	abe_dmic_din3	slimbus2_data	abe_dmic_clk2	gpio_122		dmtimer9_pwm_evt		safe_mode
<a href="#">CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS[15:0]</a>	0x118	uart2_cts	sdmmc3_clk		gpio_123				safe_mode
<a href="#">CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS[31:16]</a>	0x118	uart2_rts	sdmmc3_cmd		gpio_124				safe_mode
<a href="#">CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX[15:0]</a>	0x11C	uart2_rx	sdmmc3_dat0		gpio_125				safe_mode
<a href="#">CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX[31:16]</a>	0x11C	uart2_tx	sdmmc3_dat1		gpio_126				safe_mode
<a href="#">CONTROL_CORE_PAD0_HDQ_SIO_PAD1_I2C1_SCL[15:0]</a>	0x120	hdq_sio	i2c3_sccb	i2c2_sccb	gpio_127				safe_mode
<a href="#">CONTROL_CORE_PAD0_HDQ_SIO_PAD1_I2C1_SCL[31:16]</a>	0x120	i2c1_scl							
<a href="#">CONTROL_CORE_PAD0_I2C1_SDA_PAD1_I2C2_SCL[15:0]</a>	0x124	i2c1_sda							
<a href="#">CONTROL_CORE_PAD0_I2C1_SDA_PAD1_I2C2_SCL[31:16]</a>	0x124	i2c2_scl	uart1_rx		gpio_128				safe_mode
<a href="#">CONTROL_CORE_PAD0_I2C2_SDA_PAD1_I2C3_SCL[15:0]</a>	0x128	i2c2_sda	uart1_tx		gpio_129				safe_mode
<a href="#">CONTROL_CORE_PAD0_I2C2_SDA_PAD1_I2C3_SCL[31:16]</a>	0x128	i2c3_scl			gpio_130				safe_mode
<a href="#">CONTROL_CORE_PAD0_I2C3_SDA_PAD1_I2C4_SCL[15:0]</a>	0x12C	i2c3_sda			gpio_131				safe_mode
<a href="#">CONTROL_CORE_PAD0_I2C3_SDA_PAD1_I2C4_SCL[31:16]</a>	0x12C	i2c4_scl			gpio_132				safe_mode
<a href="#">CONTROL_CORE_PAD0_I2C4_SDA_PAD1_MCSPI1_CLK[15:0]</a>	0x130	i2c4_sda			gpio_133				safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

SYSCTRL_PADCONF_COR E.[Register Name]	Address Offset	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_CORE_PAD0_I2C4_SDA_PAD1_MCSP1_CLK[31:16]	0x130	mcspi1_clk			gpio_134				safe_mode
CONTROL_CORE_PAD0_MCSP11_SOMI_PAD1_MCSP11_SIMO[15:0]	0x134	mcspi1_somi			gpio_135				safe_mode
CONTROL_CORE_PAD0_MCSP11_SOMI_PAD1_MCSP11_SIMO[31:16]	0x134	mcspi1_simo			gpio_136				safe_mode
CONTROL_CORE_PAD0_MCSP11_CS0_PAD1_MCSP11_CS1[15:0]	0x138	mcspi1_cs0			gpio_137				safe_mode
CONTROL_CORE_PAD0_MCSP11_CS0_PAD1_MCSP11_CS1[31:16]	0x138	mcspi1_cs1	uart1_rx		gpio_138				safe_mode
CONTROL_CORE_PAD0_MCSP11_CS2_PAD1_MCSP11_CS3[15:0]	0x13C	mcspi1_cs2	uart1_cts	slimbus2_clock	gpio_139				safe_mode
CONTROL_CORE_PAD0_MCSP11_CS2_PAD1_MCSP11_CS3[31:16]	0x13C	mcspi1_cs3	uart1_rts	slimbus2_data	gpio_140				safe_mode
CONTROL_CORE_PAD0_UART3_CTS_RCTX_PAD1_UART3_RTS_SD[15:0]	0x140	uart3_cts_rctx	uart1_tx		gpio_141				safe_mode
CONTROL_CORE_PAD0_UART3_CTS_RCTX_PAD1_UART3_RTS_SD[31:16]	0x140	uart3_rts_sd			gpio_142				safe_mode
CONTROL_CORE_PAD0_UART3_RX_IRRX_PAD1_UART3_TX_IRTX[15:0]	0x144	uart3_rx_irrx	dmtimer8_pwm_evt		gpio_143				safe_mode
CONTROL_CORE_PAD0_UART3_RX_IRRX_PAD1_UART3_TX_IRTX[31:16]	0x144	uart3_tx_irtx	dmtimer9_pwm_evt		gpio_144				safe_mode
CONTROL_CORE_PAD0_SDMMC5_CLK_PAD1_SDMMC5_CMD[15:0]	0x148	sdmmc5_clk	mcspi2_clk	usbc1_icusb_dp	gpio_145		sdmmc2_clk		safe_mode
CONTROL_CORE_PAD0_SDMMC5_CLK_PAD1_SDMMC5_CMD[31:16]	0x148	sdmmc5_cmd	mcspi2_simo	usbc1_icusb_dm	gpio_146		sdmmc2_cmd		safe_mode
CONTROL_CORE_PAD0_SDMMC5_DAT0_PAD1_SDMMC5_DAT1[15:0]	0x14C	sdmmc5_dat0	mcspi2_somi	usbc1_icusb_rcv	gpio_147		sdmmc2_dat0		safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_S DMMC5_DAT0_PAD1_SDM MC5_DAT1[31:16]</a>	0x14C	sdmmc5_dat1		usbc1_ icusb_txen	gpio_148		sdmmc2_dat1		safe_mode
<a href="#">CONTROL_CORE_PAD0_S DMMC5_DAT2_PAD1_SDM MC5_DAT3[15:0]</a>	0x150	sdmmc5_dat2	mcspi2_cs1		gpio_149		sdmmc2_dat2		safe_mode
<a href="#">CONTROL_CORE_PAD0_S DMMC5_DAT2_PAD1_SDM MC5_DAT3[31:16]</a>	0x150	sdmmc5_dat3	mcspi2_cs0		gpio_150		sdmmc2_dat3		safe_mode
<a href="#">CONTROL_CORE_PAD0_M CSPI4_CLK_PAD1_MCSP14_SIMO[15:0]</a>	0x154	mcspi4_clk	sdmmc4_clk	kpd_col6	gpio_151				safe_mode
<a href="#">CONTROL_CORE_PAD0_M CSPI4_CLK_PAD1_MCSP14_SIMO[31:16]</a>	0x154	mcspi4_simo	sdmmc4_cmd	kpd_col7	gpio_152				safe_mode
<a href="#">CONTROL_CORE_PAD0_M CSPI4_SOMI_PAD1_MCSP14_CS0[15:0]</a>	0x158	mcspi4_somi	sdmmc4_dat0	kpd_row6	gpio_153				safe_mode
<a href="#">CONTROL_CORE_PAD0_M CSPI4_SOMI_PAD1_MCSP14_CS0[31:16]</a>	0x158	mcspi4_cs0	sdmmc4_dat3	kpd_row7	gpio_154				safe_mode
<a href="#">CONTROL_CORE_PAD0_U ART4_RX_PAD1_UART4_TX [15:0]</a>	0x15C	uart4_rx	sdmmc4_dat2	kpd_row8	gpio_155				safe_mode
<a href="#">CONTROL_CORE_PAD0_U ART4_RX_PAD1_UART4_TX [31:16]</a>	0x15C	uart4_tx	sdmmc4_dat1	kpd_col8	gpio_156				safe_mode
<a href="#">CONTROL_CORE_PAD0_U SBB2_ULPITLL_CLK_PAD1_USBB2_ULPITLL_STP[15:0]</a>	0x160	usb2_ulpitll_clk	usb2_ulpiphy_clk	sdmmc4_cmd	gpio_157	hsi2_cawake			safe_mode
<a href="#">CONTROL_CORE_PAD0_U SBB2_ULPITLL_CLK_PAD1_USBB2_ULPITLL_STP[31:16]</a>	0x160	usb2_ulpitll_stp	usb2_ulpiphy_stp	sdmmc4_clk	gpio_158	hsi2_cadata	dispc2_data23		safe_mode
<a href="#">CONTROL_CORE_PAD0_U SBB2_ULPITLL_DIR_PAD1_USBB2_ULPITLL_NXT[15:0]</a>	0x164	usb2_ulpitll_dir	usb2_ulpiphy_dir	sdmmc4_dat0	gpio_159	hsi2_caflag	dispc2_data22		safe_mode
<a href="#">CONTROL_CORE_PAD0_U SBB2_ULPITLL_DIR_PAD1_USBB2_ULPITLL_NXT[31:16]</a>	0x164	usb2_ulpitll_nxt	usb2_ulpiphy_nxt	sdmmc4_dat1	gpio_160	hsi2_acready	dispc2_data21		safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_U_SBB2_ULPITLL_DAT0_PAD1_USBB2_ULPITLL_DAT1[15:0]</a>	0x168	usbb2_ulpitll_dat0	usbb2_ulpiPHY_dat0	sdmmc4_dat2	gpio_161	hsi2_acwake	dispc2_data20	usbb2_mm_txen	safe_mode
<a href="#">CONTROL_CORE_PAD0_U_SBB2_ULPITLL_DAT0_PAD1_USBB2_ULPITLL_DAT1[31:16]</a>	0x168	usbb2_ulpitll_dat1	usbb2_ulpiPHY_dat1	sdmmc4_dat3	gpio_162	hsi2_acdata	dispc2_data19	usbb2_mm_txdat	safe_mode
<a href="#">CONTROL_CORE_PAD0_U_SBB2_ULPITLL_DAT2_PAD1_USBB2_ULPITLL_DAT3[15:0]</a>	0x16C	usbb2_ulpitll_dat2	usbb2_ulpiPHY_dat2	sdmmc3_dat2	gpio_163	hsi2_aclag	dispc2_data18	usbb2_mm_txse0	safe_mode
<a href="#">CONTROL_CORE_PAD0_U_SBB2_ULPITLL_DAT2_PAD1_USBB2_ULPITLL_DAT3[31:16]</a>	0x16C	usbb2_ulpitll_dat3	usbb2_ulpiPHY_dat3	sdmmc3_dat1	gpio_164	hsi2_caready	dispc2_data15	rfbi_data15	safe_mode
<a href="#">CONTROL_CORE_PAD0_U_SBB2_ULPITLL_DAT4_PAD1_USBB2_ULPITLL_DAT5[15:0]</a>	0x170	usbb2_ulpitll_dat4	usbb2_ulpiPHY_dat4	sdmmc3_dat0	gpio_165	mcspi3_somi	dispc2_data14	rfbi_data14	safe_mode
<a href="#">CONTROL_CORE_PAD0_U_SBB2_ULPITLL_DAT4_PAD1_USBB2_ULPITLL_DAT5[31:16]</a>	0x170	usbb2_ulpitll_dat5	usbb2_ulpiPHY_dat5	sdmmc3_dat3	gpio_166	mcspi3_cs0	dispc2_data13	rfbi_data13	safe_mode
<a href="#">CONTROL_CORE_PAD0_U_SBB2_ULPITLL_DAT6_PAD1_USBB2_ULPITLL_DAT7[15:0]</a>	0x174	usbb2_ulpitll_dat6	usbb2_ulpiPHY_dat6	sdmmc3_cmd	gpio_167	mcspi3_simo	dispc2_data12	rfbi_data12	safe_mode
<a href="#">CONTROL_CORE_PAD0_U_SBB2_ULPITLL_DAT6_PAD1_USBB2_ULPITLL_DAT7[31:16]</a>	0x174	usbb2_ulpitll_dat7	usbb2_ulpiPHY_dat7	sdmmc3_clk	gpio_168	mcspi3_clk	dispc2_data11	rfbi_data11	safe_mode
<a href="#">CONTROL_CORE_PAD0_U_SBB2_HSIC_DATA_PAD1_U_SBB2_HSIC_STROBE[15:0]</a>	0x178	usbb2_hsic_data			gpio_169				safe_mode
<a href="#">CONTROL_CORE_PAD0_U_SBB2_HSIC_DATA_PAD1_U_SBB2_HSIC_STROBE[31:16]</a>	0x178	usbb2_hsic_strobe			gpio_170				safe_mode
<a href="#">CONTROL_CORE_PAD0_K_PD_COL3_PAD1_KPD_COL4[15:0]</a>	0x17C	kpd_col3	kpd_col0		gpio_171				safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_KPD_COL3_PAD1_KPD_COL4[31:16]</a>	0x17C	kpd_col4	kpd_col1		gpio_172				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_COL5_PAD1_KPD_COL0[15:0]</a>	0x180	kpd_col5	kpd_col2		gpio_173				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_COL5_PAD1_KPD_COL0[31:16]</a>	0x180	kpd_col0	kpd_col3		gpio_174				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_COL1_PAD1_KPD_COL2[15:0]</a>	0x184	kpd_col1	kpd_col4		gpio_0				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_COL1_PAD1_KPD_COL2[31:16]</a>	0x184	kpd_col2	kpd_col5		gpio_1				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_ROW3_PAD1_KPD_ROW4[15:0]</a>	0x188	kpd_row3	kpd_row0		gpio_175				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_ROW3_PAD1_KPD_ROW4[31:16]</a>	0x188	kpd_row4	kpd_row1		gpio_176				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_ROW5_PAD1_KPD_ROW0[15:0]</a>	0x18C	kpd_row5	kpd_row2		gpio_177				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_ROW5_PAD1_KPD_ROW0[31:16]</a>	0x18C	kpd_row0	kpd_row3		gpio_178				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_ROW1_PAD1_KPD_ROW2[15:0]</a>	0x190	kpd_row1	kpd_row4		gpio_2				safe_mode
<a href="#">CONTROL_CORE_PAD0_KPD_ROW1_PAD1_KPD_ROW2[31:16]</a>	0x190	kpd_row2	kpd_row5		gpio_3				safe_mode
<a href="#">CONTROL_CORE_PAD0_USBA0_OTG_CE_PAD1_USBA0_OTG_DP[15:0]</a>	0x194	usba0_otg_ce							
<a href="#">CONTROL_CORE_PAD0_USBA0_OTG_CE_PAD1_USBA0_OTG_DP[31:16]</a>	0x194	usba0_otg_dp	uart3_rx_irrx	uart2_rx					safe_mode
<a href="#">CONTROL_CORE_PAD0_USBA0_OTG_DM_PAD1_FREQ_CLK1_OUT[15:0]</a>	0x198	usba0_otg_dm	uart3_tx_irtx	uart2_tx					safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

SYSCTRL_PADCONF_COR E.[Register Name]	Address Offset	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
CONTROL_CORE_PAD0_U SBA0_OTG_DM_PAD1_FRE F_CLK1_OUT[31:16]	0x198	fref_clk1_out			gpio_181				safe_mode
CONTROL_CORE_PAD0_F REF_CLK2_OUT_PAD1_SY S_NIRQ1[15:0]	0x19C	fref_clk2_out			gpio_182				safe_mode
CONTROL_CORE_PAD0_F REF_CLK2_OUT_PAD1_SY S_NIRQ1[31:16]	0x19C	sys_nirq1							safe_mode
CONTROL_CORE_PAD0_S YS_NIRQ2_PAD1_SYS_BO OT0[15:0]	0x1A0	sys_nirq2			gpio_183				safe_mode
CONTROL_CORE_PAD0_S YS_NIRQ2_PAD1_SYS_BO OT0[31:16]	0x1A0	sys_boot0			gpio_184				safe_mode
CONTROL_CORE_PAD0_S YS_BOOT1_PAD1_SYS_BO OT2[15:0]	0x1A4	sys_boot1			gpio_185				safe_mode
CONTROL_CORE_PAD0_S YS_BOOT1_PAD1_SYS_BO OT2[31:16]	0x1A4	sys_boot2			gpio_186				safe_mode
CONTROL_CORE_PAD0_S YS_BOOT3_PAD1_SYS_BO OT4[15:0]	0x1A8	sys_boot3			gpio_187				safe_mode
CONTROL_CORE_PAD0_S YS_BOOT3_PAD1_SYS_BO OT4[31:16]	0x1A8	sys_boot4			gpio_188				safe_mode
CONTROL_CORE_PAD0_S YS_BOOT5_PAD1_DPM_EM U0[15:0]	0x1AC	sys_boot5			gpio_189				safe_mode
CONTROL_CORE_PAD0_S YS_BOOT5_PAD1_DPM_EM U0[31:16]	0x1AC	dpm_emu0			gpio_11			hw_dbg0	safe_mode
CONTROL_CORE_PAD0_D PM_EMU1_PAD1_DPM_EM U2[15:0]	0x1B0	dpm_emu1			gpio_12			hw_dbg1	safe_mode
CONTROL_CORE_PAD0_D PM_EMU1_PAD1_DPM_EM U2[31:16]	0x1B0	dpm_emu2	usba0_ulpiphy_clk		gpio_13		dispc2_fid	hw_dbg2	safe_mode
CONTROL_CORE_PAD0_D PM_EMU3_PAD1_DPM_EM U4[15:0]	0x1B4	dpm_emu3	usba0_ulpiphy_stp		gpio_14	rffi_data10	dispc2_data10	hw_dbg3	safe_mode

**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_D PM_EMU3_PAD1_DPM_EM U4[31:16]</a>	0x1B4	dpm_emu4	usba0_ulpiphy_dir		gpio_15	rubi_data9	dispc2_data9	hw_dbg4	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU5_PAD1_DPM_EM U6[15:0]</a>	0x1B8	dpm_emu5	usba0_ulpiphy_nxt		gpio_16	rubi_te_vsync0	dispc2_data16	hw_dbg5	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU5_PAD1_DPM_EM U6[31:16]</a>	0x1B8	dpm_emu6	usba0_ulpiphy_dat0	uart3_tx_irtx	gpio_17	rubi_hsync0	dispc2_data17	hw_dbg6	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU7_PAD1_DPM_EM U8[15:0]</a>	0x1BC	dpm_emu7	usba0_ulpiphy_dat1	uart3_rx_irrx	gpio_18	rubi_cs0	dispc2_hsync	hw_dbg7	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU7_PAD1_DPM_EM U8[31:16]</a>	0x1BC	dpm_emu8	usba0_ulpiphy_dat2	uart3_rts_sd	gpio_19	rubi_re	dispc2_pclk	hw_dbg8	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU9_PAD1_DPM_EM U10[15:0]</a>	0x1C0	dpm_emu9	usba0_ulpiphy_dat3	uart3_cts_rctx	gpio_20	rubi_we	dispc2_vsync	hw_dbg9	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU9_PAD1_DPM_EM U10[31:16]</a>	0x1C0	dpm_emu10	usba0_ulpiphy_dat4		gpio_21	rubi_a0	dispc2_de	hw_dbg10	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU11_PAD1_DPM_EM U12[15:0]</a>	0x1C4	dpm_emu11	usba0_ulpiphy_dat5		gpio_22	rubi_data8	dispc2_data8	hw_dbg11	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU11_PAD1_DPM_EM U12[31:16]</a>	0x1C4	dpm_emu12	usba0_ulpiphy_dat6		gpio_23	rubi_data7	dispc2_data7	hw_dbg12	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU13_PAD1_DPM_EM U14[15:0]</a>	0x1C8	dpm_emu13	usba0_ulpiphy_dat7		gpio_24	rubi_data6	dispc2_data6	hw_dbg13	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU13_PAD1_DPM_EM U14[31:16]</a>	0x1C8	dpm_emu14	reserved	uart1_rx	gpio_25	rubi_data5	dispc2_data5	hw_dbg14	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU15_PAD1_DPM_EM U16[15:0]</a>	0x1CC	dpm_emu15	reserved		gpio_26	rubi_data4	dispc2_data4	hw_dbg15	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU15_PAD1_DPM_EM U16[31:16]</a>	0x1CC	dpm_emu16	dmtimer8_pwm_evt	dsi1_te0	gpio_27	rubi_data3	dispc2_data3	hw_dbg16	safe_mode
<a href="#">CONTROL_CORE_PAD0_D PM_EMU17_PAD1_DPM_EM U18[15:0]</a>	0x1D0	dpm_emu17	dmtimer9_pwm_evt	dsi1_te1	gpio_28	rubi_data2	dispc2_data2	hw_dbg17	safe_mode



**Table 18-9. Device Core Control Module Pad Configuration Register Fields (continued)**

<b>SYSCTRL_PADCONF_COR E.[Register Name]</b>	<b>Address Offset</b>	<b>Mode 0</b>	<b>Mode 1</b>	<b>Mode 2</b>	<b>Mode 3</b>	<b>Mode 4</b>	<b>Mode 5</b>	<b>Mode 6</b>	<b>Mode 7</b>
<a href="#">CONTROL_CORE_PAD0_DPM_EMU17_PAD1_DPM_EMU18</a> [31:16]	0x1D0	dpm_emu18	dmtimer10_pwm_evt	dsi2_te0	gpio_190	rfbi_data1	dispc2_data1	hw_dbg18	safe_mode
<a href="#">CONTROL_CORE_PAD0_DPM_EMU19</a> [31:0]	0x1D4	dpm_emu19	dmtimer11_pwm_evt	dsi2_te1	gpio_191	rfbi_data0	dispc2_data0	hw_dbg19	safe_mode

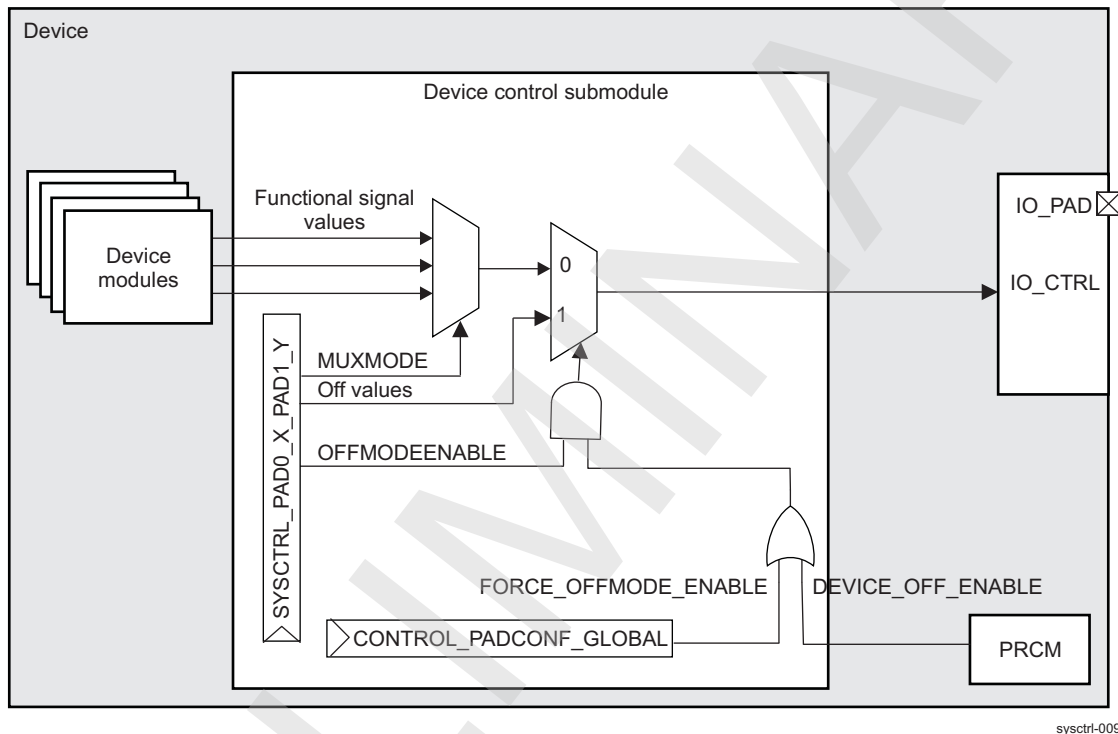
#### 18.4.8.4 Off Mode PAD Control Overview

When Off mode is selected, which is true when `PRM_DEVICE_OFF_CTRL[0] DEVICE_OFF_ENABLE = 0b1` from PRCM or `CONTROL_PADCONF_GLOBAL[31] FORCE_OFFMODE_ENABLE = 0b1` (force bit to test the value of off mode), the off mode values field `CONTROL_CORE_PAD0_X_PAD1_Y[27,11] OFFMODEOUTVALUE` overrides the pad state when `OFFMODEENABLE` bit `CONTROL_CORE_PAD0_X_PAD1_Y[25,9]` is set.

The `CONTROL_CORE_PAD0_X_PAD1_Y[26,10] OFFMODEOUTENABLE` bit is active low.

Figure 18-8 shows the off mode pad control.

**Figure 18-8. Off Mode Pad control Overview**



If off mode is active and the `OFFMODEENABLE` bit is set to 0 (disable OFF mode override feature), the pad keeps the AND value of the configuration (input/output, PU/PD) it had before going into off mode:

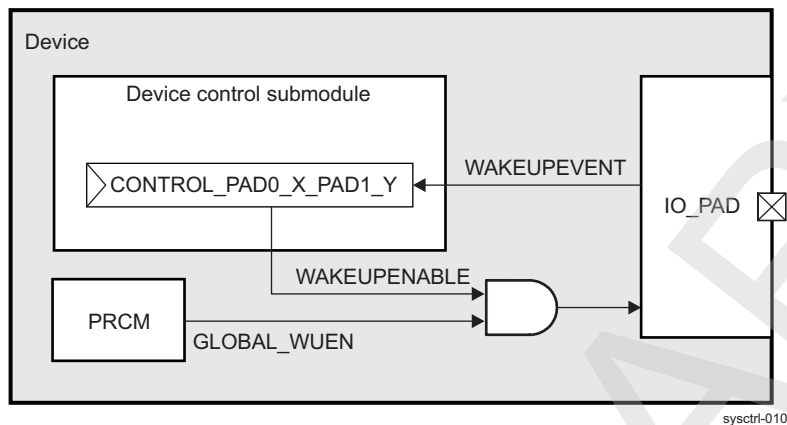
- For an input, the pad is isolated and the pull remains active.
- For an output, the value is latched before going into off mode, to drive the same value in off mode.

##### 18.4.8.4.1 Wake-Up Event Detection

In off mode, wake-up event detection can also be enabled on an input pad. The pad wake-up event is latched in the `WAKEUPEVENT` bit `CONTROL_CORE_PAD0_X_PAD1_Y[31,15]` and `CONTROL_WKUP_PAD0_X_PAD1_Y[31,15]`.

The off mode I/O pads wake-up scheme is enabled by setting the bit `PRM_IO_PMCTRL[16] GLOBAL_WUEN` from PRCM. The wake-up scheme status is transmitted by the `WKUP_ENABLE` signal. The wake-up event detection capability of each I/O pad of the device is individually enabled or disabled by writing `WAKEUPENABLE` bits - `CONTROL_CORE_PAD0_X_PAD1_Y[30,14]` and `CONTROL_WKUP_PAD0_X_PAD1_Y[30,14]`.

Figure 18-9. Wake-Up Event Detection Overview



**NOTE:** When wake-up detection is enabled for a pad, the pad must be configured as input to avoid contention between the device output buffer and an external driver. If this pin is already configured as an input in active mode, hardware automatically retains the same input state; no software action is required.

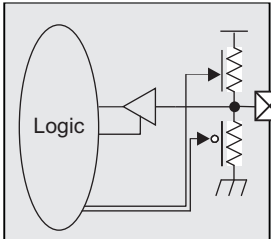
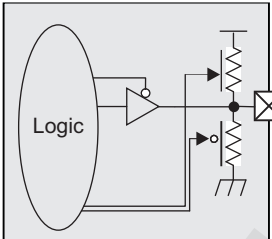
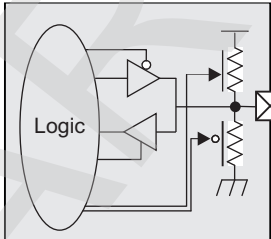
If this pin is configured as an output in active mode, the OFF override function must be enabled to set the pin as an input during off mode: in the CONTROL.CONTROL\_PADCONF register, set OFFMODEENABLE to 0x1 to enable the OFF override function, and set OFFMODEOUTENABLE to 0x1 to switch this pin to input mode.

The registers [CONTROL\\_PADCONF\\_WAKEUPEVENT\\_0](#) through [CONTROL\\_PADCONF\\_WAKEUPEVENT\\_6](#) are dedicated 32 bit registers for a separate capture of the WAKEUPEVENT from different PADs. Duplication of the wake-up event bits CONTROL\_CORE\_PAD0\_X\_PAD1\_Y[31,15] and CONTROL\_WKUP\_PAD0\_X\_PAD1\_Y[31,15] inside these 7 additional registers helps reduce the software latency in identifying which PAD has triggered a wake-up event.

### 18.4.8.4.2 I/O Power Optimization

To optimize I/O power, it is important to avoid unconnected or incorrectly pulled pins. According to the type of pins, the way to reduce power consumption can differ. Table 18-10 shows the three available pin (or ball) types.

**Table 18-10. Pin Types**

Input	Output	Bidirectional
 <p>sysctrl-011</p>	 <p>sysctrl-012</p>	 <p>sysctrl-013</p>

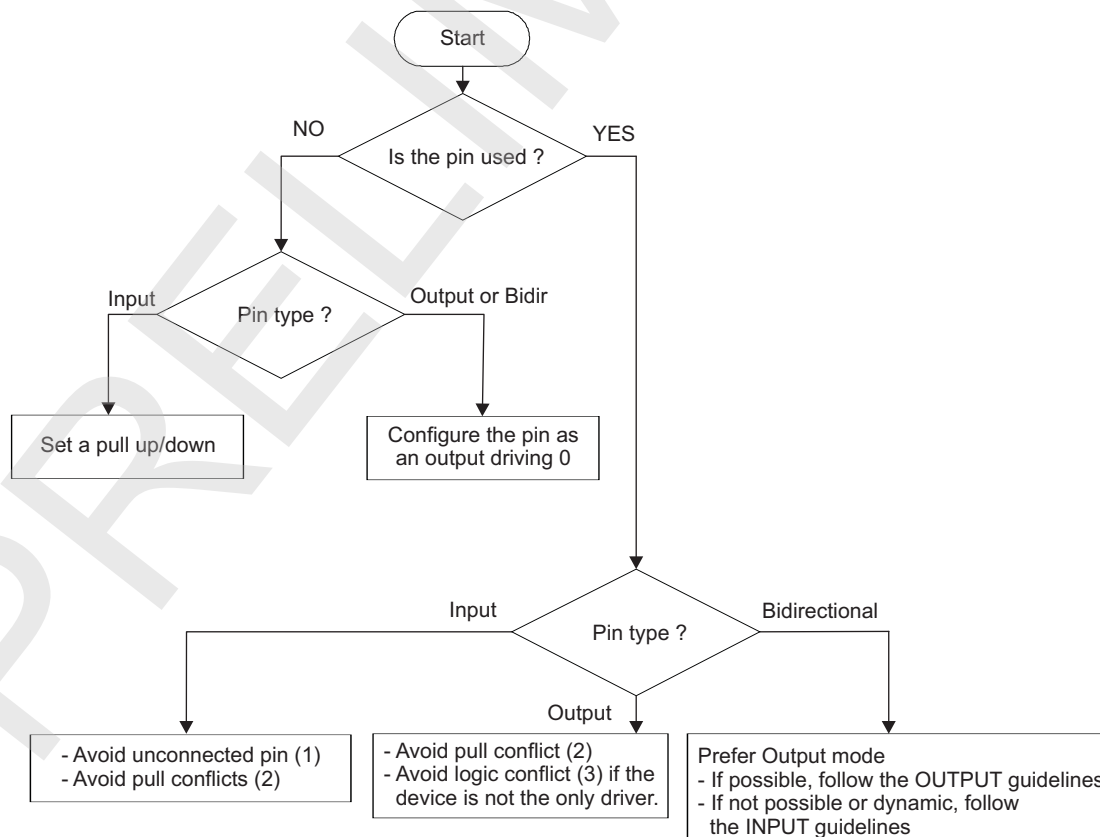
The configuration differs according to the I/O cell types. The following advices can be useful to avoid extra current leakage:

- For input pins, use a pullup/down when possible.
- For output pins, check existing pulls to avoid conflicts.
- For bidirectional pins, reconfigure the pin as an output driving 0 when possible.

Some I/O configurations involve modifications during the software setup of I/Os and sometimes require several hardware updates.

Figure 18-10 shows how to optimize the power consumption of pads.

**Figure 18-10. I/O Power Optimization Flow Chart**



sysctrl-014

The following notes give additional explanation about the pin configuration.

1. To avoid unconnected pins, the configuration depends on its use:
  - If the pin is not driven externally, a pullup / pulldown is required.
  - Otherwise, a pullup/down is not necessary.
2. Pull conflicts occur when there are different pulls on the same line. To correctly configure the pin, avoid external and internal pull together.
3. Logic conflicts consist in different electrical levels at the same time on one line. This can occur when several devices are connected to the same line. The two possible cases are:
  - If no external device drives the line, configure the pin to drive a 0.
  - If another device drives the line, either the same value must be driven or the pin must be disconnected (high-impedance state - HiZ).

---

**NOTE:** It is advised to use high impedance logical state either on the device or the external component when the line is driven by both components.

---

### 18.4.9 Extended-Drain I/O and PBIAS Cell

The device mainly supports 1.8-V I/O voltage on its interfaces, with the exception of MMC/SD/SDIO1 interface, which supports both 1.8-V and 3.0-V voltages. The need for embedded extended-drain I/Os on MMC/SD/SDIO1 interface imposes the use of embedded PBIAS cell MMC1\_PBIAS to provide a 1.8-V or 3.0-V reference.

A second PBIAS cell (PBIAS1) is integrated in the device. One of its associated extended-drain I/O cells operates signals of the gpio\_wk0 through gpio\_wk2 I/Os.

A separate extended-drain I/O cell (USBC1\_ICUSB extended-drain I/O cell) is shared between usbc1\_icusb\_dp, usbc1\_icusb\_dm (Muxmode = 0x0), and gpio98 and gpio99 (MuxMode = 0x3) pads. The USBC1\_ICUSB I/O cell and the gpio\_wk\_0-gpio\_wk\_2 extended-drain I/O cells are biased by the same PBIAS1 output voltage.

---

**NOTE:** With the appropriate PBIAS cell configuration, the different I/Os muxed on the sdmmc1, usbc1\_icusb and gpio\_wk0 through gpio\_wk2 pads can operate at 1.8V or 3V-power supply.

---

The MMC1\_PBIAS cell and its associated MMC1 extended-drain I/O is software-controlled by bits in the [CONTROL\\_PBIASLITE](#) and [CONTROL\\_MMC1](#) registers of the device core control module. For descriptions of these registers, see [Table 18-570](#) and [CONTROL\\_MMC1](#) in [Section 18.6.6](#), [SYSCTRL\\_PADCONF\\_CORE Register](#). PBIAS1 and associated GPIO\_WKx and USBC1\_ICUSB\_x I/Os are software-controlled by bits in the [CONTROL\\_PBIASLITE](#), [CONTROL\\_MMC1](#), and [CONTROL\\_GPIOWK](#) registers. For descriptions of these registers, see [Table 18-570](#), [CONTROL\\_MMC1](#) in [Section 18.6.6](#), [SYSCTRL\\_PADCONF\\_CORE Register Summary](#), and [CONTROL\\_GPIOWK](#) in [Section 18.6.9](#), [SYSCTRL\\_PADCONF\\_WKUP Register Description](#).

[Figure 18-11](#) shows the functional block diagram with the connections between the PBIAS cells and the extended-drain I/O cells.

Figure 18-11. Functional Block Diagram

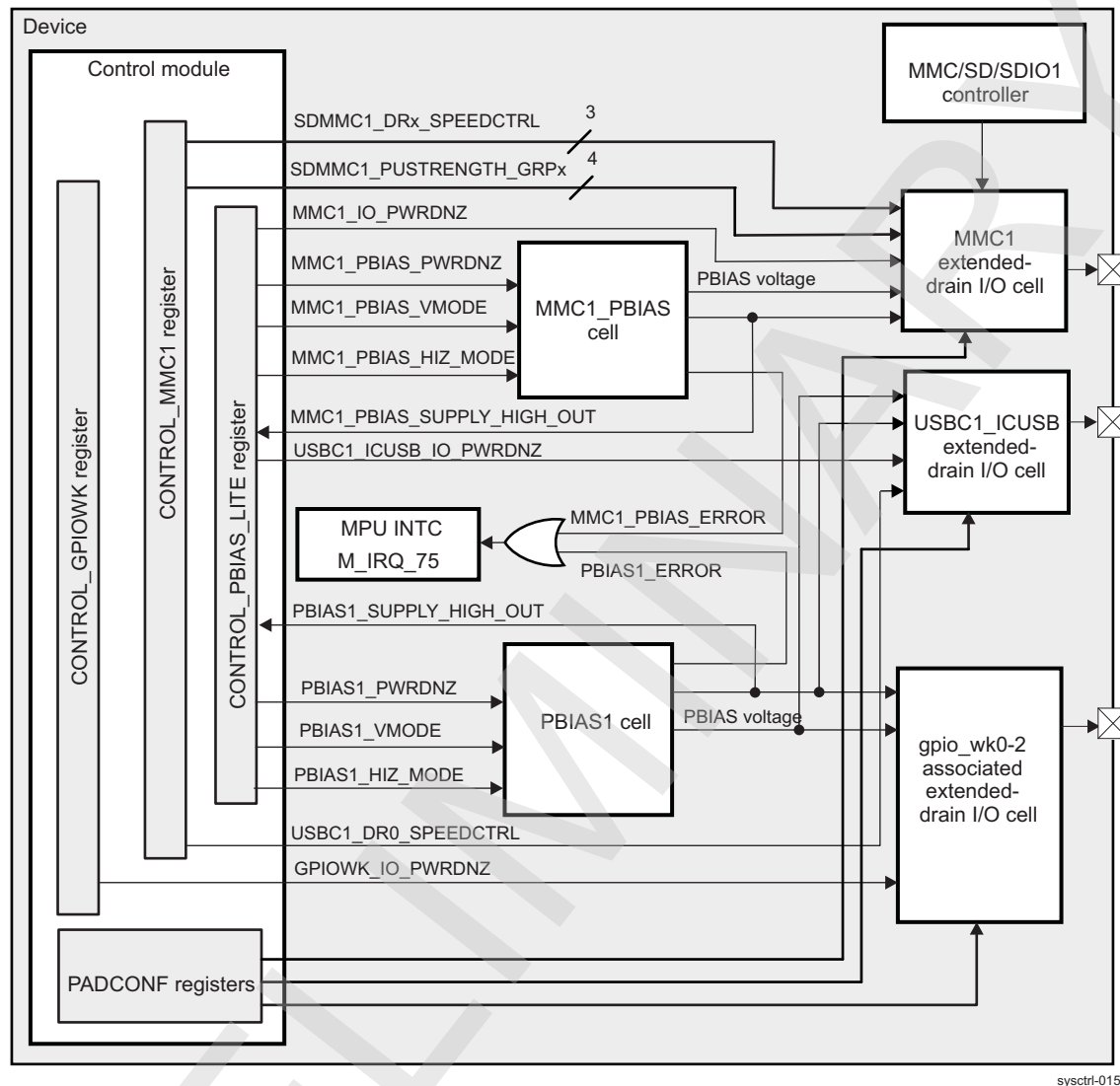


Table 18-11. PBIAS Cell and Extended-Drain I/O Pin CONTROL\_PBIAS\_LITE Bit control

Control Signals for Different PBIAS and Extended-Drain I/O Cells	Description
MMC1_PBIAS_PWRDNZ	Protects the MMC1_PBIAS cell when the SDMMC1_VDDS supply voltage is not stable. Software must keep this signal to 0b0 whenever this voltage is ramping up/down or changing.  When this bit is at 0, the PBIAS output voltage is equal to SDMMC1_VDDS voltage and the PBIAS supply detector is disabled.
MMC1_PBIAS_VMODE	Sets the SDMMC1_VDDS voltage level (MMC1_PBIAS side) according to which MMC1_PBIAS cell generates appropriate biasing levels for the MMC1 extended-drain I/O. The default state of this bit is HIGH, indicating that the voltage level SDMMC1_VDDS is 3.0 V.
MMC1_PBIAS_HIZ_MODE	When MMC1_PBIAS_HIZ_MODE is set to 1 the PBIAS output is in high impedance. MMC1_PBIAS_ERROR is automatically set to 1, when MMC1_PBIAS_HIZ_MODE = 1. When MMC1_PBIAS_HIZ_MODE = 0, MMC1_PBIAS cell is in normal operation mode.
MMC1_PBIAS_SUPPLY_HIGH_OUT	Output from the internal MMC1_PBIAS voltage supply detector, which describes whether the SDMMC1_VDDS supply is 3.0 V or 1.8 V.

**Table 18-11. PBIAS Cell and Extended-Drain I/O Pin CONTROL\_PBIAS\_LITE Bit control  
(continued)**

<b>Control Signals for Different PBIAS and Extended-Drain I/O Cells</b>	<b>Description</b>
MMC1_PBIAS_ERROR	Indicates whether the software-programmed MMC1_PBIAS_VMODE level matches the MMC1_PBIAS_SUPPLY_HI_OUT output signal during normal operation.
MMC1_IO_PWRDNZ	Protects the MMC1 I/O extended-drain cell when the SDMMC1_VDDS voltage is not stable. Software must keep this signal to 0b0 whenever the SDMMC1_VDDS supply voltage is ramping up/down or changing. When this bit is at 0, the pad is floating.
SDMMC1_DRx_SPEEDCTRL, x = (0..2)	Speed control signals for three different SDMMC1 pad groups identified within the MMC1 I/O cell
SDMMC1_PUSTRNGTH_GRPx, x = (0..3)	Pullup strength controls for four different SDMMC1 pad groups identified within the MMC1 I/O cell
PBIAS1_PWRDNZ	Protects the PBIAS1 cell when the SIM_VDDS supply voltage is not stable. Software must keep PWRDNZ signal to 0b0 whenever this supply voltage is ramping up/down or changing.  When this bit is at 0, the PBIAS1 output voltage is equal to SIM_VDDS voltage and the PBIAS supply detector is disabled.
PBIAS1_VMODE	Sets the SIM_VDDS voltage level (PBIAS1 side) according to which PBIAS1 cell generates appropriate biasing levels for the GPIOWK0GPIOWK2 and USBC1_ICUSB associated extended-drain I/Os. The default state of this bit is HIGH, indicating that the voltage level SIM_VDDS is 3.0 V.
PBIAS1_HIZ_MODE	When PBIAS1_HIZ_MODE is set to 1 the PBIAS1 output is in HIGH impedance. PBIAS1_ERROR is automatically set to 1, when PBIAS1_HIZ_MODE = 1.  When PBIAS1_HIZ_MODE = 0, PBIAS1 is in normal operation mode.
PBIAS1_SUPPLY_HIGH_OUT	Output from the internal PBIAS1 voltage supply detector, which describes whether the SIM_VDDS supply is 3.0 V or 1.8 V.
PBIAS1_ERROR	Indicates whether the software-programmed PBIAS1_VMODE level matches the PBIAS1_SUPPLY_HI_OUT output signal during normal operation
GPIOWK_IO_PWRDNZ	Protects the GPIOWK0GPIOWK2 associated extended-drain I/O cell when the SIM_VDDS voltage is not stable.  Software must keep this signal to 0b0 whenever the SIM_VDDS supply voltage is ramping up/down or changing. When this bit is at 0, the pad is floating.
PAD_GPIO_WK1_LOW	This signal should be set to 0b0 to disable the strong pulldown available after reset on the gpio_wk1 pad. This is to ensure normal use of the gpio_wk1 pad.
0.22 PAD_GPIO_WK2_LOW	This signal should be set to 0b0 to disable the strong pulldown available after reset on the gpio_wk2 pad. This is to ensure normal use of the gpio_wk2 pad.
0.23 USBC1_ICUSB_IO_PWRDNZ	Protects the USBC1_ICUSB extended-drain I/O cell when the SIM_VDDS voltage is not stable. Software must keep this signal to 0b0 whenever the SIM_VDDS supply voltage is ramping up/down or changing.  When this bit is at 0, the pad is floating.
0.24 USBC1_DR0_SPEEDCTRL	Speed controls for the ICUSB pads associated with the USBC1_ICUSB associated I/O cell

**NOTE:** When HIZ\_MODE = 1, the PBIAS output (pbias\_mmc1 pin) must be tied to a defined external potential to avoid stress on transistors.

Table 18-12 lists the power supplies for PBIAS and the extended-drain I/O cells.

**Table 18-12. Power Supplies**

<b>Name</b>	<b>Description</b>
VDD2	Core voltage supply
SDMMC1_VDDS	MMC1 I/O cell supply voltage nominal 1.8 V/3.0 V
SIM_VDDS	USBC1_ICUSB and GPIOWK I/O cell supply voltage nominal 1.8 V/3.0 V
VDDS	1.8-V supply for the input buffer

### 18.4.9.1 PBIAS Cell

The MMC1\_PBIAS cell provides a bias for the the MMC/SD/SDIO1 extended-drain I/O cell used with high voltage for the MMC/SD/SDIO1 interface. This PBIAS cell provides a voltage reference (PBIAS voltage) for biasing extended drain in the MMC/SD/SDIO1 cell. In addition to generating the bias voltage, the cell can detect the supply voltage (SDMMC1\_VDDSD) value (1.8 V or 3.0 V) and update, with its status, the [CONTROL\\_PBIASLITE](#)[24] MMC1\_PBIASLITE\_SUPPLY\_HI\_OUT bit. However, this is possible when voltage has reached its steady state (the internal PBIAS voltage detector is only enabled when the MMC1\_PBIAS\_PWRDNZ signal is set to 0b1).

The PBIAS1 cell provides a reference voltage for biasing the gpiowk0gpiowk2 associated extended-drain I/O cell and the USBC1\_ICUSB extended-drain I/O cell. In addition to generating the bias voltage, the cell can detect the supply voltage (SIM\_VDDSD) value (1.8 V or 3.0 V) and update, with its status, the [CONTROL\\_PBIASLITE](#)[30] PBIASLITE1\_SUPPLY\_HI\_OUT bit. However, this is possible when voltage has reached its steady state (the internal PBIAS voltage detector is enabled only when the PBIAS1\_PWRDNZ signal is set to 0b1).

#### CAUTION

A PBIAS cell lets the peripheral associated to its corresponding extended-drain I/O cells support 1.8-V and 3.0-V voltages. A PBIAS cell is not a part of a peripheral, but a part of the device I/Os to which this peripheral is internally connected. These device I/Os are not exclusive to one peripheral; through I/O multiplexing they can be connected to other internal signals. It is necessary to configure the PBIAS to enable the I/Os, regardless of how I/O multiplexing is configured for these device I/Os. In other words, independently of which signal is internally connected to a device I/O powered by SDMMC1\_VDDSD or SIM\_VDDSD, the corresponding PBIAS must be configured.

### 18.4.9.2 Extended-Drain I/O

The following MMC/SD/SDIO1 interface signals (selected when MuxMode = 0) use the device I/Os associated with the MMC1 extended-drain I/O cell:

- sdmmc1\_clk
- sdmmc1\_cmd
- sdmmc1\_dati (where i = 0 to 7)

The MMC1 I/O cell is powered externally through the vdds\_sdmmc1 pad.

---

**NOTE:** The MMC1 I/O cell associated device I/Os are also shared with other signals. gpio100gpio109, etc. are available when MuxMode is different than 0x0. If signals other than sdmmc1\_x are selected, the MMC1 I/O cell must also be configured.

---

For more information about signal multiplexing, see [Table 18-9](#).

For GP device I/Os, which are part of the GPIOWK pads associated extended-drain cell, Muxmode = 3 as well as Muxmode=6 are functional.

With Muxmode set to 0x3, the following signals are present:

- gpio\_wk0
- gpio\_wk1
- gpio\_wk2

With Muxmode set to 0x6, the output signals hw\_dbg1, hw\_dbg2 and hw\_dbg3 are present at the gpio\_wk0, gpio\_wk1 and gpio\_wk2 pads, respectively.

This PBIAS1 associated extended-drain I/O cell is externally powered through the vdds\_sim pad.



**NOTE:** The GPIOWK associated I/O cell should be properly configured when gpiowk0gpiowk2 pads are used. For more information about signal multiplexing, see [Table 18-9](#).

The following USBC1\_ICUSB interface signals (selected through MuxMode = 0) use the device I/Os associated with the USBC1\_ICUSB extended-drain I/O cell:

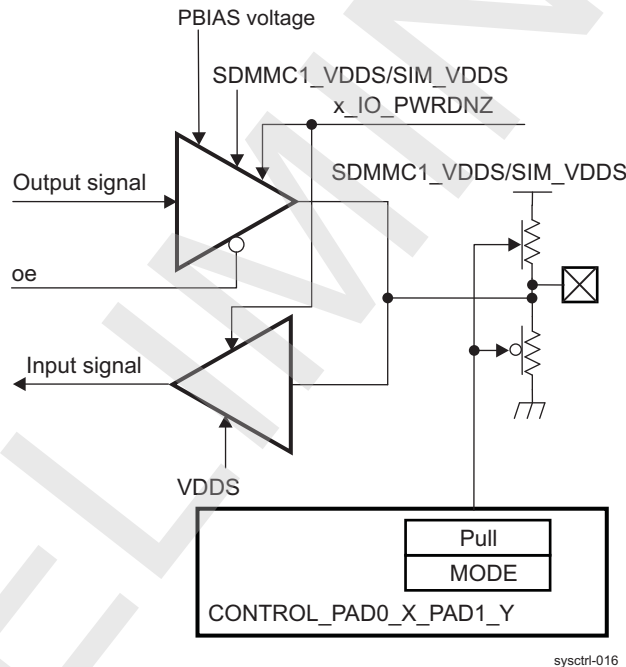
- usbc1\_icusb\_dp
- usbc1\_icusb\_dm

The USBC1\_ICUSB I/O cell uses the same external power supply voltage as the GPIOWK cell (vdds\_sim pad).

**NOTE:** The USBC1\_ICUSB I/O cell associated device I/Os are also shared with gpio\_98 and gpio\_99 signals, when MuxMode is equal to 0x3. USBC1IC\_ICUSB cell must be configured to allow for gpio\_98 - gpio\_99 operation. For more information about signal multiplexing, see [Table 18-9](#).

Figure 18-12 describes the extended-drain I/O cells.

**Figure 18-12. Extended-Drain I/O**



The extended-drain I/O cells have the following I/O signals:

- Output signal, input signal, and oe, which comes from the selected I/O module with the correctly configured MUXMODE field within the corresponding CONTROL\_CORE\_PAD0\_X\_PAD1\_Y or CONTROL\_WKUP\_PAD0\_X\_PAD1\_Y register.

**NOTE:** Extended-drain I/Os are muxed I/Os with mode and pullup/pulldown configurations programmable in the control module.

- The PBIAS voltage is a voltage reference for biasing the extended-drain in the MMC1, USBC1\_ICSUB, and GPIOWK I/O cells.
- The PWRDNZ signals are used to protect the I/O cells when the SDMMC1\_VDDS/SIM\_VDDS voltage is not stable.

**CAUTION**

Software must keep the PWRDNZ related signals to 0b0 whenever the SDMMC1\_VDDDS or SIM\_VDDDS signal is ramping up/down or changing. When it is 0, the pad is floating (the pad may not reflect the state of the output signal, and the input signal may not reflect the state of the pad).

- The [CONTROL\\_PBIASLITE](#)[24] MMC1\_PBIASLITE\_SUPPLY\_HI\_OUT bit is a status bit on the SDMMC1\_VDDDS value and is used to inform the MMC1 I/O cell about the value of the SDMMC1\_VDDDS signal.
- The [CONTROL\\_PBIASLITE](#)[30] PBIASLITE1\_SUPPLY\_HI\_OUT bit is a status bit on the SIM\_VDDDS value and is used to inform the gpiowk0gpiowk2 associated I/O cell and USBC1\_ICUSB I/O cell about the value of the SIM\_VDDDS signal.

For more information about the software configurations of the extended drain I/Os and PBIAS cells, see [Section 18.5.1.2.2, Extended Drain I/Os and PBIAS Cells Programming Guide](#).

**NOTE:** The device can be supplied by an external power IC. TI provides such a global solution to its customers with the TWL60xx power IC.

If the device is associated with the TWL60xx power IC, before using an extended-drain I/O interface, software must program the TWL60xx to enable the VMMC1 (for the MMC/SD/SDIO1 or muxed I/O modules) or VSIM (for the GPIOWK0, GPIOWK2, USBC1\_ICUSB, or muxed I/Os) and the LDO to provide a 1.8-V/3.0-V voltage. This is done by software through the I<sup>2</sup>C interface that links the device and TWL60xx IC.

**CAUTION**

The power supply voltage generated from the companion power IC and applied on the vdds\_sdmmc1 or vdds\_sim device pins must be programmed according to the recommendations for the currently selected I/O module power supply range.

If the application does not want the I/O interface running at 3.0 V, software users must then assert the VMODE signal to low for 1.8-V activity: in this case, the PBIAS is connected to ground.

[Table 18-13](#) lists the control signal with the corresponding control bits from the [CONTROL\\_PBIASLITE](#), [CONTROL\\_MMC1](#), and [CONTROL\\_GPIOWK](#) registers to configure the PBIAS and the extended-drain I/O cells. These signals can be software-controlled.

**Table 18-13. PBIAS/Extended-Drain I/O Cells Control Signals**

Control signals for Different PBIAS and Extended-Drain I/O Cells	Bits used for PBIAS Cell and I/O Cell Configurations for Different I/O Modules	Reset Value
MMC1_PBIAS_PWRDNZ	<a href="#">CONTROL_PBIASLITE</a> [22] MMC1_PBIASLITE_PWRDNZ	0
MMC1_PBIAS_VMODE	<a href="#">CONTROL_PBIASLITE</a> [21] MMC1_PBIASLITE_VMODE	1
MMC1_PBIAS_HIZ_MODE	<a href="#">CONTROL_PBIASLITE</a> [25] MMC1_PBIASLITE_HIZ_MODE	0
MMC1_PBIAS_SUPPLY_HIGH_OUT	<a href="#">CONTROL_PBIASLITE</a> [24] MMC1_PBIASLITE_SUPPLY_HI_OUT	0
MMC1_PBIAS_ERROR	<a href="#">CONTROL_PBIASLITE</a> [23] MMC1_PBIASLITE_VMODE_ERROR	0
MMC1_IO_PWRDNZ	<a href="#">CONTROL_PBIASLITE</a> [26] MMC1_PWRDNZ	0
SDMMC1_DRx_SPEEDCTRL, x = (0..2)	<a href="#">CONTROL_MMC1</a> [27] SDMMC1_DR0_SPEEDCTRL	0
	<a href="#">CONTROL_MMC1</a> [26] SDMMC1_DR1_SPEEDCTRL	0
	<a href="#">CONTROL_MMC1</a> [25] SDMMC1_DR2_SPEEDCTRL	0

**Table 18-13. PBIAS/Extended-Drain I/O Cells Control Signals (continued)**

Control signals for Different PBIAS and Extended-Drain I/O Cells	Bits used for PBIAS Cell and I/O Cell Configurations for Different I/O Modules	Reset Value
SDMMC1_PUSTRENGTH_GRPx, x = (0..3)	CONTROL_MMC1[31] SDMMC1_PUSTRENGTH_GRP0	1
	CONTROL_MMC1[30] SDMMC1_PUSTRENGTH_GRP1	0
	CONTROL_MMC1[29] SDMMC1_PUSTRENGTH_GRP2	1
	CONTROL_MMC1[28] SDMMC1_PUSTRENGTH_GRP3	1
PBIAS1_PWRDNZ	CONTROL_PBIASLITE[28] PBIASLITE1_PWRDNZ	0
PBIAS1_VMODE	CONTROL_PBIASLITE[27] PBIASLITE1_VMODE	1
PBIAS1_HIZ_MODE	CONTROL_PBIASLITE[31] PBIASLITE1_HIZ_MODE	0
PBIAS1_SUPPLY_HIGH_OUT	CONTROL_PBIASLITE[30] PBIASLITE1_SUPPLY_HI_OUT	0
PBIAS1_ERROR	CONTROL_PBIASLITE[29] PBIASLITE1_VMODE_ERROR	0
GPIOWK_IO_PWRDNZ	CONTROL_GPIOWK[28] GPIOWK_IO_PWRDNZ	0
PAD_GPIO_WK1_LOW	CONTROL_GPIOWK[31] PAD_GPIO_WK1_LOW	1
PAD_GPIO_WK2_LOW	CONTROL_GPIOWK[29] PAD_GPIO_WK2_LOW	1
USBC1_ICUSB_IO_PWRDNZ	CONTROL_PBIASLITE[20] USBC1_ICUSB_PWRDNZ	0
USBC1_DR0_SPEEDCTRL	CONTROL_MMC1[24] USBC1_DR0_SPEEDCTRL	0

The two PBIAS cells support two ranges of I/O SDMMC1\_VDDS/ SIM\_VDDS: 1.8 V, typical for low-voltage applications, and 3.0 V, typical for high-voltage applications. For each supply voltage range, the cell generates suitable bias voltage (PBIAS) for extended-drain PMOS devices.

**CAUTION**

The PBIAS cell must be programmed according to peripheral power supply voltage. See [Table 18-14](#).

**Table 18-14. Voltage Configuration<sup>(1)</sup>**

PBIASLITEVMODE Configuration	SDMMC1_VDDS/SIM_VDDSVoltage	Type of Operation
1.8 V	1.8 V	Normal 1.8-V operation
1.8 V	3.0 V	Damaging configuration <sup>(2)</sup>
3.0 V	1.8 V	Degraded functionality <sup>(2)</sup>
3.0 V	3.0 V	Normal 3.0-V operation

<sup>(1)</sup> For damaging configuration, hardware system protection is provided to prevent deterioration of the associated extended-drain I/Os.

<sup>(2)</sup> It is forbidden to use these modes.

The PBIAS output voltage is the same as SDMMC1\_VDDS / SIM\_VDDS voltage when the corresponding PWRDNZ bit is low. Once the SDMMC1\_VDDS / SIM\_VDDS supply is stable, software can release the PWRDNZ (setting it high). This then starts up the PBIAS cell work to generate the PBIAS voltage. During the completion process the corresponding I/Os cannot be used for transmitting data. If VMODE is set correctly, (compared with supply detector output and found to be the same), then PBIAS output voltage is generated based on the value of VMODE.

**NOTE:** In the case of a damaging configuration, hardware system protection prevents deterioration of the associated extended-drain I/Os.

**CAUTION**

The following requirements are critical for an extended-drain I/O cell:

- The VMODE bit must be defined before the PWRNDZ bit is made HIGH (cell is brought out of power-down mode).
- The default state of VMODE bit is HIGH (indicates 3.0-V operation).
- PWRNDZ bit must be kept LOW whenever the SDMMC1\_VDDS/SIM\_VDDS supply is ramping up/down or changing. This could be damaging.
- Supply detector is disabled when PWRDNZ = 0 and in this case supply detector output (SUPPLY\_HI\_OUT) is initialized to 0.

**18.4.9.3 PBIAS Error Generation**

If VMODE and supply detector outputs are unequal. It takes 4 microseconds for the comparator to process the inputs and generate VMODE\_ERROR flag.

Table 18-15 summarizes the generation of the PBIAS error signals - MMC1\_PBIAS\_ERROR of the MMC1\_PBIAS cell and PBIAS1\_ERROR of the PBIAS1 cell. They depend on the various CONTROL\_PBIASLITE bits combinations. The shaded row is a potential condition that could cause a reliability issue if not detected. To prevent this reliability issue, the PBIAS voltage is tied to SDMMC1\_VDDS/SIM\_VDDS when the PBIAS error signal is HIGH. This disables the IOs.

**CAUTION**

Although asserting VMODE\_ERROR to 1 provides that PBIAS voltage is equal to SDMMC1\_VDDS/SIM\_VDDS, this does not completely protect the IO cells. It is also recommended that all the I/Os are powered down, setting the corresponding I/O cells PWRDNZ mode bits to 0b0 by software as soon as VMODE\_ERROR=0b1.

**Table 18-15. PBIAS Error Signal Truth Table**

VMODE	PWRDNZ	HIZ_MODE	SUPPLY_HI_OUT	VMODE_ERROR
0	0	X	0	0
0	1	0	0	0
0	1	0	1	1
X	1	1	X	1
1	0	X	0	0
1	1	0	0	1
1	1	0	1	0

**NOTE:** VMODE\_ERROR =0b1: VMODE level is not the same as SUPPLY\_HI\_OUT or high impedance mode (HiZ) is selected.

VMODE\_ERROR = 0b0: VMODE level is the same as SUPPLY\_HI\_OUT or VMODE not considered (PWRDNZ=0b0)

The MMC1\_PBIAS\_ERROR and PBIAS1\_ERROR signals are ORed and the outputs mapped to the MPU subsystem interrupt controller (INTC). The interrupt line MA\_IRQ\_75 is activated upon error occurrence. For more information about the Cortex-A9 MPU interrupt controller registers and their description, see the *ARM Cortex-A9 Technical Reference Manual*.

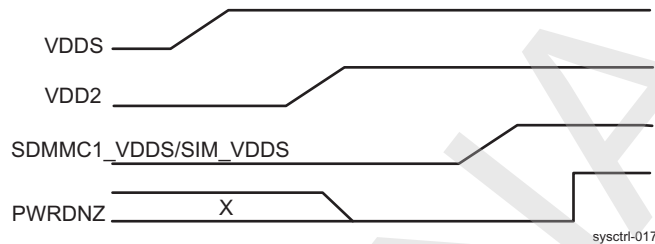
The CONTROL\_PBIASLITE[29] PBIASLITE1\_VMODE\_ERROR bit and the CONTROL\_PBIASLITE[23] MMC1\_PBIASLITE\_VMODE\_ERROR bit also indicates whether this kind of error occurs.

#### 18.4.9.4 Critical Timing Requirements

It is crucial that the PBIAS and I/O cell related PWRDNZ bits are deasserted (made 1 from 0) only after SDMMC1\_VDDS/SIM\_VDDS power supply voltage is stable in the device. The cells support only the case when 1.8-V input buffer supply voltage VDDS ramps up before VDD2 (core voltage). This power-up sequence is automatically provided upon power-on reset by hardware in the device. Care must be taken that the externally provided SDMMC1\_VDDS/SIM\_VDDS voltage (1.8 V/ 3 V) be powered up, only after both VDD2 and VDDS have reached their steady state.

Figure 18-13 shows the only possible power-up sequence, as well as the expected behavior of PWRNDZ mode related bits with regard to supplying voltages ramp up.

**Figure 18-13. VDDS Ramps Up Before VDD2**



#### 18.4.10 Band Gap Voltage and Temperature Sensor

The device supplies a voltage reference and a temperature sensor feature which are gathered in the band gap voltage and temperature sensor (VBGAPTS) module. The band gap provides current and voltage reference for its internal circuits and also to other analog IP blocks. The analog-to-digital converter (ADC) produces an output value that is proportional to the silicon temperature.

Main features of the BGAPTS module are:

- A constant voltage reference output: 0.5 V
- Four constant current reference outputs of 1microAmp.
- Analog supply is a nominal 1.8 V.
- Small ADC with 8-bit digital output
- Off mode compatible
- Thermal shutdown comparator output

The band-gap and the temperature sensor are software-controlled by bits in the register [CONTROL\\_TEMP\\_SENSOR](#) within the SYSCTRL\_GENERAL\_CORE module instance. See [Table 18-192](#), for the description of this register.

Figure 18-14 shows the functional block diagram of the band gap and the temperature sensor module.

Figure 18-14. Functional Block Diagram

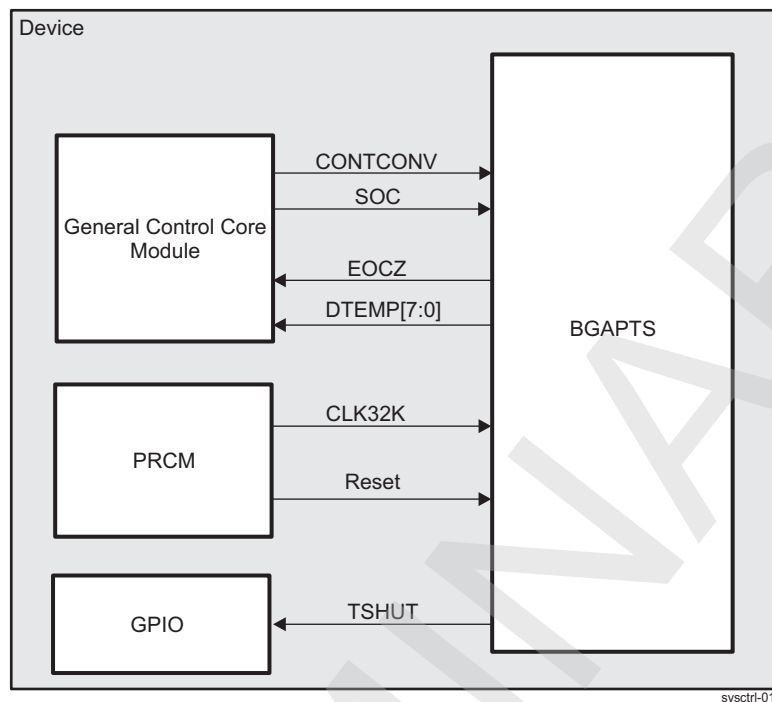


Table 18-16 describes the input and output signals for the BGAPTS module.

Table 18-16. Band Gap Voltage and Temperature Sensor Signal Description

Pin	I/O <sup>(1)</sup>	Description
SOC	I	Start Of Conversion signal. A transition to high starts a new ADC conversion cycle.
EOCZ	O	End of Conversion signal. When low, the signal indicates that the value of DTEMP[7:0] is valid.
DTEMP[7:0]	O	Temperature data from the temperature sensor. This value is valid when EOCZ is low.
CONTCONV	I	Configures the temperature sensor in continuous conversion mode. 0: ADC single conversion mode 1: ADC continuous conversion mode
TSHUT	O	Thermal shutdown comparator output. The signal is low during normal operation and goes high during a thermal shutdown event.
CLRZ	I	When low the temperature sensor is in reset mode. This signal must be LOW when OFFMODE is HIGH.
CLK32K	I	32-kHz functional clock from the CORE power domain used by the temperature sensor during temperature conversion.

<sup>(1)</sup> I=Input, O=Output

All the configuration signals are available through the [CONTROL\\_TEMP\\_SENSOR](#) register.

**NOTE:** VBGAPTS CLK32K is used only for the temperature sensor. When ADC is not converting, this clock can be switched OFF (CLK32K = 0). In this case, DTEMP[7:0] retains the temperature code obtained at the previous conversion.

#### 18.4.10.1 Band Gap Voltage Reference

The band gap voltage reference feature provides several constant voltage and current references for its internal circuits, but also for other analog modules.

The voltage reference signal is a 0.5-V output which is internally used by the module and that is exported for external modules or for tests.

The current reference signals consist in four lines, each of which generates a 1- $\mu$ A current with a 1.8-V voltage. These lines can be used to bias other modules.

### 18.4.10.2 Temperature Sensor

The temperature sensor feature is used to convert the temperature of the device into a decimal value coded on 8 bits. An internal ADC is used for conversion. The recommended operating temperatures must be in the range ( $-40^{\circ}\text{C}$  to  $125^{\circ}\text{C}$ ) for standard conversion, and can go up to  $160^{\circ}\text{C}$  when using the thermal shutdown comparator output (TSHUT signal). This comparator output is asserted high when temperature of the chip is higher than  $147^{\circ}\text{C}$ . This signal is internally connected to a GPIO pin. See [Chapter 25, General-Purpose Interface](#), for more information.

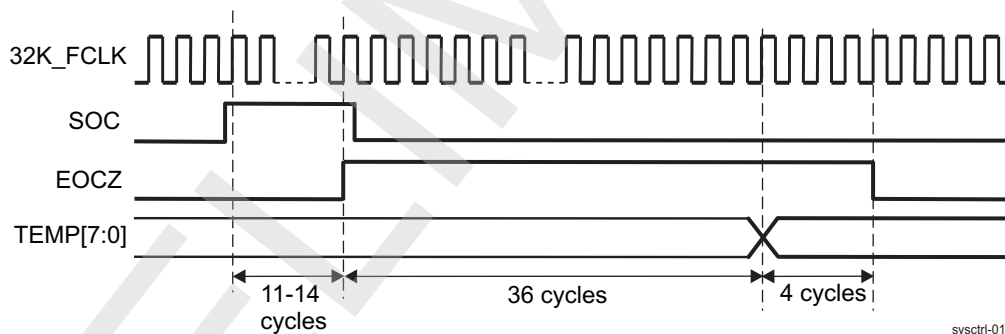
The temperature sensor offers two operating modes: a single conversion mode and a continuous conversion mode.

#### 18.4.10.2.1 Single Conversion Mode (CONTCONV = 0)

When the ADC is idle ([CONTROL\\_TEMP\\_SENSOR\[8\]](#) BGAP\_TEMP\_SENSOR\_EOCZ = 0) and the [CONTROL\\_TEMP\\_SENSOR\[10\]](#) BGAP\_TEMP\_SENSOR\_CONTCONV bit is set to 0, it is possible to ask for a single temperature conversion. To initiate a new conversion, the start of conversion (SOC) signal should be asserted and maintained high until the end of conversion (EOCZ) signal goes high, after which the SOC should be made low by writing 0 to the [CONTROL\\_TEMP\\_SENSOR\[9\]](#) BGAP\_TEMP\_SENSOR\_SOC bit. Conversion completion is indicated by a negative edge on EOCZ (the [CONTROL\\_TEMP\\_SENSOR\[8\]](#) BGAP\_TEMP\_SENSOR\_EOCZ bit).

The timing sequence for a single temperature conversion is shown in [Figure 18-15](#).

**Figure 18-15. Single Conversion Mode (CONTCONV = 0)**



#### 18.4.10.2.2 Continuous Conversion Mode (CONTCONV = 1)

When the ADC is idle ([CONTROL\\_TEMP\\_SENSOR\[8\]](#) BGAP\_TEMP\_SENSOR\_EOCZ = 0) and the [CONTROL\\_TEMP\\_SENSOR\[10\]](#) BGAP\_TEMP\_SENSOR\_CONTCONV bit is set to 1, it is possible to ask for a continuous temperature conversion. To initiate a new conversion, the SOC signal should be asserted and maintained high until the EOCZ signal goes high, after which the SOC should be made low by writing 0 to the [CONTROL\\_TEMP\\_SENSOR\[9\]](#) BGAP\_TEMP\_SENSOR\_SOC bit. The ADC starts converting continuously, and when a conversion is complete, data is written to the [CONTROL\\_TEMP\\_SENSOR\[7:0\]](#) BGAP\_TEMP\_SENSOR\_DTEMP bit field and EOCZ goes low after four clock cycles.

The timing sequence for a continuous temperature conversion is shown in [Section 18.4.11.1](#).

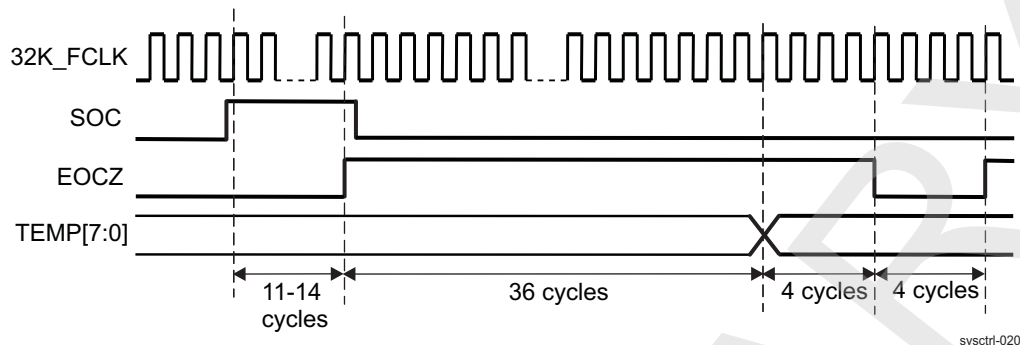
**Figure 18-16. Continuous Conversion Mode (CONTCONV = 1)****18.4.10.2.3 ADC Codes Versus Temperature**

Table 18-17 gives the temperature range corresponding to each value of the `CONTROL_TEMP_SENSOR[7:0]` `BGAP_TEMP_SENSOR_DTEMP` bit field.

**Table 18-17. ADC Code Versus Temperature**

ADC Code	Temperature °C		ADC Code	Temperature °C		ADC Code	Temperature °C		ADC Code	Temperature °C	
	From	To		From	To		From	To		From	To
0	-40	-40	32	-5	-3.5	64	50	52	96	105	107
1	-40	-40	33	-3.5	-1.5	65	52	53.5	97	107	109
2	-40	-40	34	-1.5	0	66	53.5	55	98	109	111
3	-40	-40	35	0	2	67	55	57	99	111	113
4	-40	-40	36	2	3.5	68	57	58.5	100	113	115
5	-40	-40	37	3.5	5	69	58.5	60	101	115	117
6	-40	-40	38	5	6.5	70	60	62	102	117	118.5
7	-40	-40	39	6.5	8.5	71	62	64	103	118.5	120
8	-40	-40	40	8.5	10	72	64	66	104	120	122
9	-40	-40	41	10	12	73	66	68	105	122	123.5
10	-40	-40	42	12	13.5	74	68	70	106	123.5	125
11	-40	-40	43	13.5	15	75	70	71.5	107	125	125
12	-40	-40	44	15	17	76	71.5	73.5	108	125	125
13	-40	-38	45	17	19	77	73.5	75	109	125	125
14	-38	-35	46	19	21	78	75	77	110	125	125
15	-35	-34	47	21	23	79	77	78.5	111	125	125
16	-34	-32	48	23	25	80	78.5	80	112	125	125
17	-32	-30	49	25	27	81	80	82	113	125	125
18	-30	-28	50	27	28.5	82	82	83.5	114	125	125
19	-28	-26	51	28.5	30	83	83.5	85	115	125	125
20	-26	-24	52	30	32	84	85	87	116	125	125
21	-24	-22	53	32	33.5	85	87	88.5	117	125	125
22	-22	-20	54	33.5	35	86	88.5	90	118	125	125
23	-20	-18.5	55	35	37	87	90	92	119	125	125
24	-18.5	-17	56	37	38.5	88	92	93.5	120	125	125
25	-17	-15	57	38.5	40	89	93.5	95	121	125	125
26	-15	-13.5	58	40	42	90	95	97	122	125	125
27	-13.5	-12	59	42	43.5	91	97	98.5	123	125	125
28	-12	-10	60	43.5	45	92	98.5	100	124	125	125



**Table 18-17. ADC Code Versus Temperature (continued)**

ADC Code	Temperature °C		ADC Code	Temperature °C		ADC Code	Temperature °C		ADC Code	Temperature °C	
29	-10	-8	61	45	47	93	100	102	125	125	125
30	-8	-6.5	62	47	48.5	94	102	103.5	126	125	125
31	-6.5	-5	63	48.5	50	95	103.5	105	127	125	125

**NOTE:** ADC code values in the subrange 128–255 are reserved for future use.

### 18.4.11 Hardware Observability

#### 18.4.11.1 Device Internal Signals Observability Overview

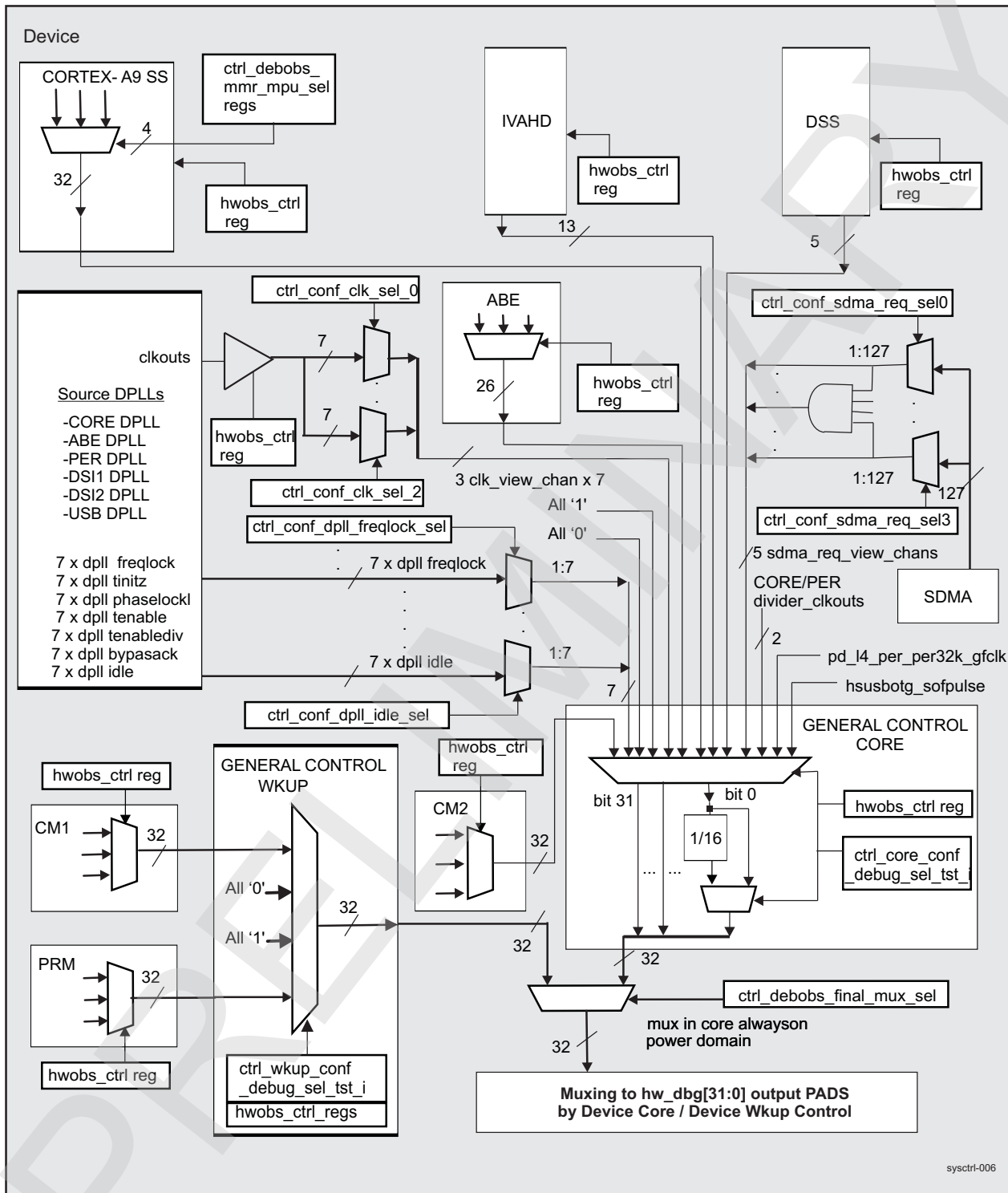
The General Control Module provides observability features covering different module internal signals related to:

- Power and reset management protocol inside different power domains
- Power domain states / transition indication
- Interrupt requests by different modules
- DMA requests by different modules
- Clocks and clock management

The associated signal multiplexing function is controlled through the hardware observability I/O multiplexing registers (32bits read/write registers). The selected signals to be observed are multiplexed to the po\_hw\_dbg[31:0] bus.

Figure 18-17 shows an overview of observability multiplexing, which minimizes the number of signals exchanged at the device boundary.

Figure 18-17. Overview of the Hardware Observability Functionality



#### 18.4.11.2 Observability Gating Capabilities

The programmer has the option to gate all 32 signals of the observability bus simultaneously, keeping

them either tied to 1, programming `CONTROL_HWOBS_CONTROL` [1] `HWOBS_ALL_ONE_MODE` = 0b1, or keeping them tied to 0, programming `CONTROL_HWOBS_CONTROL`[2] `HWOBS_ALL_ZERO_MODE` = 0b1. This can be useful to check the path from hardware observability to external pads. These two bits should be set to 0 to have the observability feature enabled for all observability pads.

Additionally the bit `CONTROL_HWOBS_CONTROL`[0] `HWOBS_MACRO_ENABLE` provides gating feature only to observability signals coming from the following modules (identified in MACRO group): CORTEX-A9 MPU subsystem, IVAHD, display subsystem, CM2 and ABE. When `CONTROL_HWOBS_CONTROL`[0] `HWOBS_MACRO_ENABLE` = 0b0, the associated with MACRO group observability outputs are gated and forced to 0b0.

---

**NOTE:** Do not set the observability gating bits: `CONTROL_HWOBS_CONTROL`[1] `HWOBS_ALL_ONE_MODE` and `CONTROL_HWOBS_CONTROL`[2] `HWOBS_ALL_ZERO_MODE` simultaneously in 1 to avoid undefined states of the observability bus.

---

The clock frequency of the `po_hw_dbg(0)` signal can be optionally divided by 1, 2, 4, 8 or 16 through programming the bit field `CONTROL_HWOBS_CONTROL`[7:3] `HWOBS_CLK_DIV_SEL`.

---

**NOTE:** For enabling observability over `po_hw_dbg(0)` bus line, bit field `CONTROL_HWOBS_CONTROL`[7:3] `HWOBS_CLKDIV_SEL` must be set different than 0. The default value (0) is not enough to see the core or the wkup observability value at pin `hw_dbg0`.

---

The different observable clockout signals can be simultaneously gated at DPLLs outputs through setting the bit `CONTROL_HWOBS_CONTROL`[8] `HWOBS_CLKOBS_GATE_ENABLE` = 0b1. This feature is useful to save power.

### 18.4.11.3 Observability Signals Multiplexing at General Core/Wake-Up Control Level

Most of the observability multiplexers (MACRO/DPLL/sDMA associated) are instantiated inside the General Core Control module, being part of the switchable CORE domain. There are also signal multiplexers instantiated inside General Wakeup Control module (CM1/PRM dedicated), being part of the WAKEUP domain. The 32-bit `CONTROL_DEBOBS_FINAL_MUX_SEL`[31:0] SELECT bit field, instantiated inside the always-on core domain, selects a signal to be observed between the outputs at CORE and the WAKEUP multiplexing level. A bit from this register set to 1 selects to route a signal from CORE domain to a certain `po_hw_dbgn` (n=0,..31) line, and when set to 0 selects to route a signal from WAKEUP domain to the same line. The output of this top level multiplexing is further passed to padconfiguration muxing I/O level inside Device Core Control Module boundaries.

The module and view-channel selection/reduction layer in the CORE domain is configured through the 4-bit field MODE in the 32 x 32-bit registers `CONTROL_CORE_CONF_DEBUG_SEL_TST_0` through `CONTROL_CORE_CONF_DEBUG_SEL_TST_31`. The following modules can be selected as source of observability signals: Cortex-A9 MPU subsystem, IVAHD, display subsystem, CM2 and ABE (MACRO signal sources).

Part of the `CONTROL_CORE_CONF_DEBUG_SEL_TST` registers are responsible for selection of sDMA view channels, different DPLL-clock view channels as well as other signal view channels.

- 3 x DPLL clockout signal view channels
- 7 x DPLL (other than clkout signal) view channels
- 5 x sDMA signal view channels
- 3 x additional clockout view channels monitoring signals sourced from CORE and PER domain clock dividers -`coredivider_clkout3`; `perdivider_clkout4` and `per32k_gfclk`.

[Table 18-18](#) summarizes the internal signals multiplexing schema for each of the 32 `CONTROL_CORE_CONF_DEBUG_SEL_TST` registers.

**Table 18-18. CORE Power Domain Observability Signals Multiplexing**

REG_NAME	REG_ID	REG_OFFSET	REG_RESET	INPUT0	INPUT1	INPUT2	INPUT3	INPUT4	INPUT5	INPUT6	INPUT7	INPUT8	INPUT9	INPUT10	INPUT11	INPUT12
Reg Name	Reg	Reg	Mode value upon power-on reset	Mode 0x0	Mode 0x1	Mode 0x2	Mode 0x3	Mode 0x4	Mode 0x5	Mode 0x6	Mode 0x7	Mode 0x8	Mode 0x9	Mode 0xA	Mode 0xB	Mode 0xC
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_0</a>	0	0x0480	0x0	hwobs_int_mpu(0)	clk_view_0	Reserved	Reserved	This information is not available in the public domain.	hwobs_int_cm2(0)	hwobs_int_ivahd(0)	Reserved	hwobs_int_abe(0)	This information is not available in the public domain.	Reserved	Reserved	This information is not available in the public domain.
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_1</a>	1	0x0484	0x0	hwobs_int_mpu(1)	clk_view_1	Reserved	Reserved	Reserved	hwobs_int_cm2(1)	hwobs_int_ivahd(1)	Reserved	hwobs_int_abe(1)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_2</a>	2	0x0488	0x0	hwobs_int_mpu(2)	clk_view_2	Reserved	Reserved	Reserved	hwobs_int_cm2(2)	hwobs_int_ivahd(2)	Reserved	hwobs_int_abe(2)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_3</a>	3	0x048C	0x0	hwobs_int_mpu(3)	dppll_freqlock	Reserved	Reserved	Reserved	hwobs_int_cm2(3)	hwobs_int_ivahd(3)	Reserved	hwobs_int_abe(3)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_4</a>	4	0x0490	0x0	hwobs_int_mpu(4)	dppll_tinitz	Reserved	Reserved	Reserved	hwobs_int_cm2(4)	hwobs_int_ivahd(4)	Reserved	hwobs_int_abe(4)	This information is not available in the public domain.	Reserved	Reserved	N/A

**Table 18-18. CORE Power Domain Observability Signals Multiplexing (continued)**

REG_NAME	REG_ID	REG_OFFSET	REG_RESET	INPUT0	INPUT1	INPUT2	INPUT3	INPUT4	INPUT5	INPUT6	INPUT7	INPUT8	INPUT9	INPUT10	INPUT11	INPUT12
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_5</a>	5	0x0494	0x0	hwobs_int_mpu(5)	dppll_phaselock	Reserved	Reserved	Reserved	hwobs_int_cm2(5)	hwobs_int_ivahd(5)	Reserved	hwobs_int_abe(5)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_6</a>	6	0x0498	0x0	hwobs_int_mpu(6)	Reserved	Reserved	Reserved	Reserved	hwobs_int_cm2(6)	hwobs_int_ivahd(6)	Reserved	hwobs_int_abe(6)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_7</a>	7	0x049C	0x0	hwobs_int_mpu(7)	dppll_tenable	sdma_req_vie_w_0	Reserved	Reserved	hwobs_int_cm2(7)	hwobs_int_ivahd(7)	Reserved	hwobs_int_abe(7)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_8</a>	8	0x04A0	0x0	hwobs_int_mpu(8)	dppll_tenablediv	sdma_req_vie_w_1	Reserved	Reserved	hwobs_int_cm2(8)	hwobs_int_ivahd(8)	Reserved	hwobs_int_abe(8)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_9</a>	9	0x04A4	0x0	hwobs_int_mpu(9)	dppll_bypassack	sdma_req_vie_w_2	Reserved	Reserved	hwobs_int_cm2(9)	hwobs_int_ivahd(9)	Reserved	hwobs_int_abe(9)	This information is not available in the public domain.	Reserved	hwobs_dss_vencvtdetgp	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_10</a>	10	0x04A8	0x0	hwobs_int_mpu(10)	dppll_idle	sdma_req_vie_w_3	Reserved	Reserved	hwobs_int_cm2(10)	hwobs_int_ivahd(10)	Reserved	hwobs_int_abe(10)	This information is not available in the public domain.	Reserved	hwobs_dss_dacpwrndnd_aczcvbs_y	N/A

Table 18-18. CORE Power Domain Observability Signals Multiplexing (continued)

REG_NAME	REG_ID	REG_OFFSET	REG_RESET	INPUT0	INPUT1	INPUT2	INPUT3	INPUT4	INPUT5	INPUT6	INPUT7	INPUT8	INPUT9	INPUT10	INPUT11	INPUT12
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_11</a>	11	0x04AC	0x0	hwobs_int_mpu(11)	hwobs_hsusbotg_sofpulse	sdma_req_view_all	Reserved	Reserved	hwobs_int_cm2(11)	hwobs_int_ivahd(11)	Reserved	hwobs_int_abe(11)	This information is not available in the public domain.	Reserved	hwobs_dssdacpwrndnaczcroma	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_12</a>	12	0x04B0	0x0	hwobs_int_mpu(12)	Reserved	hwobs_coredivider_clkout3	Reserved	Reserved	hwobs_int_cm2(12)	hwobs_int_ivahd(12)	Reserved	hwobs_int_abe(12)	This information is not available in the public domain.	Reserved	hwobs_dssdacpwrndnaccvbs	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_13</a>	13	0x04B4	0x0	hwobs_int_mpu(13)	Reserved	hwobs_perdivider_clkout4	Reserved	Reserved	hwobs_int_cm2(13)	hwobs_int_ivahd(13)	Reserved	hwobs_int_abe(13)	This information is not available in the public domain.	Reserved	hwobs_dssdacpwrndnbgz	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_14</a>	14	0x04B8	0x0	hwobs_int_mpu(14)	Reserved	hwobs_pd_l4_per32k_gfclk	Reserved	Reserved	hwobs_int_cm2(14)	hwobs_int_ivahd(14)	Reserved	hwobs_int_abe(14)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_15</a>	15	0x04BC	0x0	hwobs_int_mpu(15)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(15)	hwobs_int_ivahd(15)	Reserved	hwobs_int_abe(15)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_16</a>	16	0x04C0	0x0	hwobs_int_mpu(16)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(16)	hwobs_int_ivahd(16)	Reserved	hwobs_int_abe(16)	This information is not available in the public domain.	Reserved	Reserved	N/A

**Table 18-18. CORE Power Domain Observability Signals Multiplexing (continued)**

REG_NAME	REG_ID	REG_OFFSET	REG_RESET	INPUT0	INPUT1	INPUT2	INPUT3	INPUT4	INPUT5	INPUT6	INPUT7	INPUT8	INPUT9	INPUT10	INPUT11	INPUT12
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_17</a>	17	0x04C4	0x0	hwobs_int_mpu(17)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(17)	hwobs_int_ivahd(17)	Reserved	hwobs_int_abe(17)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_18</a>	18	0x04C8	0x0	hwobs_int_mpu(18)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(18)	hwobs_int_ivahd(18)	Reserved	hwobs_int_abe(18)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_19</a>	19	0x04CC	0x0	hwobs_int_mpu(19)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(19)	hwobs_int_ivahd(19)	Reserved	hwobs_int_abe(19)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_20</a>	20	0x04D0	0x0	hwobs_int_mpu(20)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(20)	hwobs_int_ivahd(20)	Reserved	hwobs_int_abe(20)	This information is not available in the public domain.	Reserved	Reserved	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_21</a>	21	0x04D4	0x0	hwobs_int_mpu(21)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(21)	hwobs_int_ivahd(21)	Reserved	hwobs_int_abe(21)	This information is not available in the public domain.	Reserved	N/A	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_22</a>	22	0x04D8	0x0	hwobs_int_mpu(22)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(22)	hwobs_int_ivahd(22)	Reserved	hwobs_int_abe(22)	This information is not available in the public domain.	Reserved	N/A	N/A

**Table 18-18. CORE Power Domain Observability Signals Multiplexing (continued)**

REG_NAME	REG_ID	REG_OFFSET	REG_RESET	INPUT0	INPUT1	INPUT2	INPUT3	INPUT4	INPUT5	INPUT6	INPUT7	INPUT8	INPUT9	INPUT10	INPUT11	INPUT12
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_23</a>	23	0x04DC	0x0	hwobs_int_mpu(23)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(23)	hwobs_int_ivahd(23)	Reserved	hwobs_int_abe(23)	This information is not available in the public domain.	Reserved	N/A	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_24</a>	24	0x04E0	0x0	hwobs_int_mpu(24)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(24)	hwobs_int_ivahd(24)	Reserved	hwobs_int_abe(24)	This information is not available in the public domain.	Reserved	N/A	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_25</a>	25	0x04E4	0x0	hwobs_int_mpu(25)	Reserved	0	Reserved	Reserved	hwobs_int_cm2(25)	hwobs_int_ivahd(25)	Reserved	hwobs_int_abe(25)	This information is not available in the public domain.	Reserved	N/A	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_26</a>	26	0x04E8	0x0	hwobs_int_mpu(26)	Reserved	0	Reserved	0	hwobs_int_cm2(26)	hwobs_int_ivahd(26)	Reserved	hwobs_int_abe(26)	This information is not available in the public domain.	Reserved	N/A	N/A
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_27</a>	27	0x04EC	0x0	hwobs_int_mpu(27)	Reserved	0	Reserved	0	hwobs_int_cm2(27)	hwobs_int_ivahd(27)	Reserved	hwobs_int_abe(27)	This information is not available in the public domain.	Reserved	0	sdma_req_view_0
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_28</a>	28	0x04F0	0x0	hwobs_int_mpu(28)	Reserved	0	Reserved	0	hwobs_int_cm2(28)	hwobs_int_ivahd(28)	Reserved	hwobs_int_abe(28)	This information is not available in the public domain.	Reserved	0	sdma_req_view_1



**Table 18-18. CORE Power Domain Observability Signals Multiplexing (continued)**

REG_NAME	REG_ID	REG_OFFSET	REG_RESET	INPUT0	INPUT1	INPUT2	INPUT3	INPUT4	INPUT5	INPUT6	INPUT7	INPUT8	INPUT9	INPUT10	INPUT11	INPUT12
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_29</a>	29	0x04F4	0x0	hwobs_int_mpu(29)	Reserved	0	Reserved	0	hwobs_int_cm2(29)	hwobs_int_ivahd(29)	Reserved	hwobs_int_abe(29)	This information is not available in the public domain.	Reserved	0	sdma_req_view_2
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_30</a>	30	0x04F8	0x0	hwobs_int_mpu(30)	Reserved	0	Reserved	0	hwobs_int_cm2(30)	hwobs_int_ivahd(30)	Reserved	hwobs_int_abe(30)	This information is not available in the public domain.	Reserved	0	sdma_req_view_3
<a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_31</a>	31	0x04FC	0x0	hwobs_int_mpu(31)	Reserved	0	Reserved	0	hwobs_int_cm2(31)	hwobs_int_ivahd(31)	Reserved	hwobs_int_abe(31)	This information is not available in the public domain.	Reserved	0	sdma_req_view_all

There is a third 1:N selection layer, which is applied over several signals to route one at a time to a certain view channel output:

- The CONF\_CLK\_SEL0[2:0] MULT through CONF\_CLK\_SEL2[2:0] MULT bit fields control multiplexing of DPLL CLKOUT signals over three parallel DPLL clockout view channels. The signal source of a certain view channel can be selected to be one of 6 DPLLs in the device -g.h. CORE DPLL, ABE DPLL, PER DPLL, DSI1 DPLL, DSI2 DPLL and USB DPLL.
- The register bit fields associated with the 7 different DPLL signal view channels are summarized in [Table 18-19](#). Again, the 3-bit bit field MULT in each of the 7 different registers makes it possible to choose the source of the dpll signal to be one of the following device DPLLs : CORE DPLL, ABE DPLL, PER DPLL, DSI1 DPLL, DSI2 DPLL and USB DPLL .
- There are four independent system dma request view channels, each of which has a dedicated register ([CONTROL\\_CONF\\_SDMA\\_REQ\\_SEL0](#) through [CONTROL\\_CONF\\_SDMA\\_REQ\\_SEL3](#)), with a 7-bit bit field MULT to select one from a total of 127 possible system dma request signals for observation purposes. See [Chapter 16](#), *sDMA Controller* for more details. The fifth channel, sdma\_req\_view\_all, is acquired through applying logical AND reduction over the four currently selected sDMA view channel output signals).

**Table 18-19. DPLL Signal Observability Controls**

Feature	Type	Register/Bit Field/Observability Control	Description
Different domain dpll freqlock observ.	Control	<a href="#">CONTROL_CONF_DPLL_FREQLOCK_SEL</a> [2:0] MULT	Select DPLL freqlockview channel(1:8)
Different domain dpll phaselock observ.	Control	<a href="#">CONTROL_CONF_DPLL_PHASELOCK_SEL</a> [2:0] MULT	Select DPLL phaselockview channel(1:8)

**Table 18-19. DPLL Signal Observability Controls (continued)**

Feature	Type	Register/Bit Field/Observability Control	Description
Different domain dpll tinitz observ.	Control	<a href="#">CONTROL_CONF_DPLL_TINITZ_SEL</a> [2:0] MULT	Selects to observe DPLL tinitzview channel (1:8)
Different domain dpll tenable observ.	Control	<a href="#">CONTROL_CONF_DPLL_TENABLE_SEL</a> [2:0] MULT	Selects to observe DPLL tenableview channel (1:8)
Different domain dpll tenablediv observ.	Control	<a href="#">CONTROL_CONF_DPLL_TENABLEDIV_SEL</a> [2:0] MULT	selects to observe DPLL tenabledivview channel (1:8)
Different domain dpll bypassack observ.	Control	<a href="#">CONTROL_CONF_DPLL_BYPASSACK_SEL</a> [2:0] MULT	selects to observe DPLL bypassackview channel (1:8)
Different domain dpll idle observ.	Control	<a href="#">CONTROL_CONF_DPLL_IDLE_SEL</a> [2:0] MULT	selects to observe DPLL idlview channel (1:8)

The [CONTROL\\_WKUP\\_CONF\\_DEBUG\\_SEL\\_TST\\_i](#)[0] MODE bit field (i=0..31) provides the option to choose between the observation of device Power and Reset Manager signals or observation of Clock Manager 1 related signals. If [CONTROL\\_WKUP\\_CONF\\_DEBUG\\_SEL\\_TST\\_i](#)[0] =0b1, the hwobs\_int\_prm\_i signal is selected. Otherwise the hwobs\_int\_cm1\_i signal is outputted.

#### 18.4.11.4 Observability Signals Multiplexing at Device Core Control Level

The pads used for the hardware debug must be properly configured by selecting the hardware debug function (hw\_dbgn) of the pad. To map the hw\_dbg0 through hw\_dbg19 signals to their associated external device pins, set the related MUXMODE fields within [CONTROL\\_CORE\\_PAD0\\_SYS\\_BOOT5\\_PAD1\\_DPM\\_EMU0](#) through [CONTROL\\_CORE\\_PAD0\\_DPM\\_EMU19](#) padconfiguration registers to 0b110 (Mode 6). To make the remaining signals: hw\_dbg20 through the hw\_dbg31 available at their corresponding chip outputs, set the MUXMODE fields within [CONTROL\\_CORE\\_PAD0\\_CAM\\_GLOBALRESET\\_PAD1\\_USBB1\\_ULPITLL\\_CLK](#) through [CONTROL\\_CORE\\_PAD0\\_USBB1\\_ULPITLL\\_DAT7\\_PAD1\\_USBB1\\_HSIC\\_DATA](#) padconfiguration registers to 0b110 (Mode 6).

To have hw\_dbg1 through hw\_dbg13 signals multiplexed on WAKEUP domain- pads, one should set MuxMode=0x6 in the following SYSCONF\_PADCONF\_WKUP registers:

- [CONTROL\\_WKUP\\_PAD0\\_GPIO\\_WK0\\_PAD1\\_GPIO\\_WK1](#) through [CONTROL\\_WKUP\\_PAD0\\_GPIO\\_WK2\\_PAD1\\_GPIO\\_WK3](#)[31:16] to have hw\_dbg1 - hw\_dbg4 signals output on the gpio\_wk0 - gpio\_wk3 device pads, respectively.
- [CONTROL\\_WKUP\\_PAD0\\_GPIO\\_WK4\\_PAD1\\_SR\\_SCL](#) [15:0] to have hw\_dbg5 signal output on the gpio\_wk4 device pad.
- [CONTROL\\_WKUP\\_PAD0\\_FREF\\_CLK0\\_OUT\\_PAD1\\_FREF\\_CLK3\\_REQ](#) through [CONTROL\\_WKUP\\_PAD0\\_FREF\\_CLK3\\_OUT\\_PAD1\\_FREF\\_CLK4\\_REQ](#) to have hw\_dbg6 - hw\_dbg9 signal output on the fref\_clk0\_out, fref\_clk3\_req, fref\_clk3\_out and fref\_clk4\_req device pads, respectively.
- [CONTROL\\_WKUP\\_PAD0\\_FREF\\_CLK4\\_OUT\\_PAD1\\_SYS\\_32K](#) [15:0] to have hw\_dbg10 signal output on the fref\_clk4\_out device pad.
- [CONTROL\\_WKUP\\_PAD0\\_SYS\\_PWR\\_REQ\\_PAD1\\_SYS\\_PWRON\\_RESET\\_OUT](#) [31:16] to have hw\_dbg11 signal output on the sys\_pwrn\_reset\_out device pad.
- [CONTROL\\_WKUP\\_PAD0\\_SYS\\_BOOT6\\_PAD1\\_SYS\\_BOOT7](#) to have hw\_dbg12 and hw\_dbg13 signals output on the sys\_boot6 and sys\_boot7 device pads, respectively.

See [Table 18-6](#) for more information.

See [Table 18-111](#), *ACTIVE PADS Configuration Points*, for more details on hardware observability configuration.

#### 18.4.11.5 Observability Signals Sourced at Module Level

Observable signals coming from within different device subsystems are mapped to General Core Control and General Wakeup Control observability inputs. Additional multiplexing is also implemented inside some of the observable modules (See [Figure 18-17](#), *Overview of the Hardware Observability Functionality*).

##### 18.4.11.5.1 Dual Cortex-A9 Subsystem Observable Signals

[Table 18-20](#) through [Table 18-27](#) show Cortex-A9 MPUSS observable signals available in 8 multiplexing modes. The mode is selected at General Control Module level through programming the bit field [CONTROL\\_DEBOBS\\_MMR\\_MPU](#)[3:0] SELECT.

**Table 18-20. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 0**

<b>CONTROL_DEBOBS_MMR_MPU[3:0] SELECT = 0b000</b>			
<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[31:24]	Intr [7:0]	hwobs_int_mpu(31) downto hwobs_int_mpu(24) correspond to Cortex-A9 MPUSS interrupt requests on lines MA_IRQ_7 downto MA_IRQ_0. See <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> , for further details.	-
hwobs_int_mpu[23]	mpu_rstn	MPU warm reset	-
hwobs_int_mpu[22]	mpu_pwrn_rstn	MPU power-on reset	-
hwobs_int_mpu[21:15]	Reserved	-	-
hwobs_int_mpu[14]	CPU0 Standby	Standby request asserted from CPU0 L1STBY ctrl to the local PRM Wrapper	
hwobs_int_mpu[13]	CPU1 Standby	Standby request asserted from CPU1 L1STBY ctrl to the local PRM Wrapper	
hwobs_int_mpu[12]	DPLLPhaseLock	Module in Phase lock condition	Active high
hwobs_int_mpu[11]	DPLL Clock / 16	MPU DPLL clock divided by 16	
hwobs_int_mpu[10]	DPLLFreqLock	Module in Freq lock condition	
hwobs_int_mpu[9]	PRESETN	Debug Reset	
hwobs_int_mpu[8]	ATRESETN	Trace Reset	
hwobs_int_mpu[7]	DPLLInitZ	-	
hwobs_int_mpu[6:5]	Reserved	-	
hwobs_int_mpu[4]	nCPURESET[0]	Cortex-A9 CPU reset (per CPU0)	
hwobs_int_mpu[3]	clk_mgclk/ 16	-	
hwobs_int_mpu[2]	PROFILING_EVEN2PRM[2:0]	Output to PRM for profiling local CPU power status	
hwobs_int_mpu[1]			
hwobs_int_mpu[0]			

**Table 18-21. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 1**

<b>CONTROL_DEBOBS_MMR_MPU[3:0] SELECT = 0b001</b>			
<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[31:10]	Reserved	-	-
hwobs_int_mpu[9]	WDTRESET	Watch Dog Status Reset	-
hwobs_int_mpu[8]	Clk_mgenb/8	-	-
hwobs_int_mpu[7]	DPLLInitZ	MPUSS DPLL Initialization	-
hwobs_int_mpu[6:5]	Reserved	-	-
hwobs_int_mpu[4]	nCPURESET[1]	Cortex A9 CPU Reset ( per CPU1)	
hwobs_int_mpu[3]	clk_mgclk/ 16	-	

**Table 18-21. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 1 (continued)**

<b>CONTROL_DEBOBS_MMR_MPU[3:0] SELECT = 0b001</b>			
<b>SYCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[2]	PROFILING_EVEN2PRM[5:3]	Output to PRM for profiling local CPU power status.	
hwobs_int_mpu[1]			
hwobs_int_mpu[0]			

**Table 18-22. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 2**

<b>CONTROL_DEBOBS_MMR_MPU[3:0] SELECT = 0b010</b>			
<b>SYCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[27]	top_mstandby_0	Standby handshake to local PRCM (CPU0)	-
hwobs_int_mpu[26]	top_mwait_0	standby wakeup handshake for CPU0	
hwobs_int_mpu[25]	top_mwakeup_0	wakeup request from CPU0	
hwobs_int_mpu[24]	top_mstandby_1	Standby handshake to local PRCM (CPU1)	
hwobs_int_mpu[23]	top_mwait_1	standby wakeup handshake for CPU1	-
hwobs_int_mpu[22]	top_mwakeup_1	wakeup request from CPU1	
hwobs_int_mpu[21]	MSTANDBY	Global MStandby handshake to Global PRCM	
hwobs_int_mpu[20]	MWAIT	Global MWait handshake to Global PRCM	
hwobs_int_mpu[19]	MWAKEUP	Global Mwakeup handshake to Global PRCM	
hwobs_int_mpu[18]	EMUASSERTNRESET[0]	Command for assertion of a local reset to CPU Core0	
hwobs_int_mpu[17:8]	Reserved	-	-
hwobs_int_mpu[7]	DPLLInitZ	MPUSS Dpll initialization	
hwobs_int_mpu[6]	Aclk / 8		
hwobs_int_mpu[5]	Aclk_50p / 8		
hwobs_int_mpu[4]	L2_Cache_Ctrl_IDLE	IDLE state of L2-cache controller	
hwobs_int_mpu[3]	clk_mgclk/ 16	-	
hwobs_int_mpu[2]	PROFILING_EVEN2PRM[8:6]	Output to PRM for profiling local CPU power status.	-
hwobs_int_mpu[1]			
hwobs_int_mpu[0]			

**Table 18-23. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 3**

<b>CONTROL_DEBOBS_MMR_MPU[3:0] SELECT = 0b011</b>			
<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[31:16]	Intr [31:16]	hwobs_int_mpu(31) downto hwobs_int_mpu(16) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_31 downto MA_IRQ_16. See <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> ,for further details.	-
hwobs_int_mpu[15:8]	Intr [15:8]	hwobs_int_mpu(15) downto hwobs_int_mpu(8) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_15 downto MA_IRQ_8. See <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> ,for further details.	-
hwobs_int_mpu[7]	DPLLInitZ	MPUSS DPLL Initialization	-
hwobs_int_mpu[6]	PODPLLBYPASSACK	Bypass status signal.	1 o/p clock in bypass
hwobs_int_mpu[5]	PIDPLLTENABLEDIV	Core RegM2/N2 load enable (L-H).	-
hwobs_int_mpu[4]	PIDPLLTENABLE	Core RegM2/N2 load enable (L-H).	-
hwobs_int_mpu[3]	clk_mgclk/ 16	-	-
hwobs_int_mpu[2]	PROFILING_EVEN2PRM[11:9]	Output to PRM for profiling local CPU power status.	-
hwobs_int_mpu[1]			
hwobs_int_mpu[0]			

**Table 18-24. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 4**

<b>CONTROL_DEBOBS_MMR_MPU[3:0] SELECT = 0b100</b>			
<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[31:24]	Intr [55:48]	hwobs_int_mpu(31) downto hwobs_int_mpu(24) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_55 downto MA_IRQ_48. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> ,for further details.	-
hwobs_int_mpu[23:16]	Intr [47:40]	hwobs_int_mpu(23) downto hwobs_int_mpu(16) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_47 downto MA_IRQ_40. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> ,for further details.	-

**Table 18-24. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 4 (continued)**

<b>CONTROL_DEBOBS_MMR_MPU[3:0] SELECT = 0b100</b>			
<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[15:8]	Intr [39:32]	hwobs_int_mpu(15) downto hwobs_int_mpu(8) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_39 downto MA_IRQ_32. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> , for further details.	-
hwobs_int_mpu[7]	PIDPLLDLE	Observes PLL to Idle mode setting. Sets CLKOUT to the bypass mode clock as selected by ULOWCLKEN.	1: MPUSS DPLL is set to IDLE mode
hwobs_int_mpu[6]	DPLLWRCK	-	-
hwobs_int_mpu[5]	DPLLCLKOUT	MPUSS DPLL clockout	-
hwobs_int_mpu[4]	EMUASSERTNRESET[1]	Command for assertion of a local reset to CPU Core1	-
hwobs_int_mpu[3]	clk_mgclk/ 16	-	-
hwobs_int_mpu[2]	LPRM_EMU_i/f CPU0[2:0]	LatchReset Bus Ack	-
hwobs_int_mpu[1]		ForceActive Ack	
hwobs_int_mpu[0]		InhibitSleep Ack	

**Table 18-25. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 5**

<b>CONTROL_DEBOBS_MMR_MPU[3:0] SELECT = 0b101</b>			
<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[31:16]	Intr [79:64]	hwobs_int_mpu(31) downto hwobs_int_mpu(16) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_79 downto MA_IRQ_64. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> , for further details.	-
hwobs_int_mpu[15:8]	Intr [63:56]	hwobs_int_mpu(15) downto hwobs_int_mpu(8) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_63 downto MA_IRQ_56. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> , for further details.	-
hwobs_int_mpu[7:4]	Reserved	-	-
hwobs_int_mpu[3]	clk_mgclk/ 16	-	-

**Table 18-25. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 5 (continued)****CONTROL\_DEBOBS\_MMR\_MPU[3:0] SELECT = 0b101**

<b>SYSTRM_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYSTRM_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[2]	LPRM_EMU_i/f CPU0[5:3]	CPU0 EMUWAITINRESET	-
hwobs_int_mpu[1]		CPU0 EMURELEASEFROMWIR	-
hwobs_int_mpu[0]		CPU0 EMUBLOCKRESET	-

**Table 18-26. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 6****CONTROL\_DEBOBS\_MMR\_MPU[3:0] SELECT = 0b110**

<b>SYSTRM_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYSTRM_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[31:24]	Intr [103:96]	hwobs_int_mpu(31) downto hwobs_int_mpu(24) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_103 downto MA_IRQ_96. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> , for further details.	-
hwobs_int_mpu[23:16]	Intr [95:88]	hwobs_int_mpu(23) downto hwobs_int_mpu(16) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_95 downto MA_IRQ_88. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> , for further details.	-
hwobs_int_mpu[15:8]	Intr [87:80]	hwobs_int_mpu(15) downto hwobs_int_mpu(8) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_87 downto MA_IRQ_80. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> , for further details.	-
hwobs_int_mpu[7:5]	TESTMODE	Reserved	-
hwobs_int_mpu[4]	DPLLBYPASS	DPLL bypass command	-
hwobs_int_mpu[3]	clk_mgclk/ 16	-	-
hwobs_int_mpu[2]	LPRM_EMU_i/f CPU1[2:0]	LatchReset Bus Ack	-
hwobs_int_mpu[1]		ForceActive Ack	-
hwobs_int_mpu[0]		InhibitSleep Ack	-



**Table 18-27. Observability Signals Multiplexing at Cortex-A9 MPUSS Level - Mode 7**

<b>CONTROL_DEBOBS_MMR_MPU[3:0] SELECT = 0b111</b>			
<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS</b>	<b>Cortex-A9 MPUSS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_mpu[31]	Intr [127: 112]	hwobs_int_mpu(31) downto hwobs_int_mpu(16) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_127 downto MA_IRQ_112. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> for further details.	-
hwobs_int_mpu[15]	Intr [111: 104]	hwobs_int_mpu(15) downto hwobs_int_mpu(8) correspond to Cortex-A9 MPUSS interrupt requests on lines: MA_IRQ_111 downto MA_IRQ_104. Refer to <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> for further details.	-
hwobs_int_mpu[7:4]	Reserved	-	-
hwobs_int_mpu[3]	clk_mgclk/ 16	-	-
hwobs_int_mpu[2]	LPRM_EMU_i/f CPU1[5:3]	CPU1 EMUWAITINRESET	-
hwobs_int_mpu[1]		CPU1 EMURELEASEFROMWIR	-
hwobs_int_mpu[0]		CPU1 EMUBLOCKRESET	-

**18.4.11.5.2 IVAHD Subsystem Observable Signals**

Table 18-28 shows the full set of 13 IVAHD observable signals from inside the IVAHD subsystem. No observability signals multiplexing is implemented at IVAHD level.

**Table 18-28. Observability Signals at IVAHD Subsystem Level**

<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside IVAHD Subsystem</b>	<b>IVAHD SS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_ivahd[31:13]	unused	General control module IVAHD inputs tied to low	tied to low
hwobs_int_ivahd[12]	MAILBOX_poidleack[1]	Mailbox idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[11]	IVAHD_INTC_poidleack[1]	IVAHD local interconnect idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[10]	SMSET_poidleack[1]	SMSET idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[9]	MSGIF_poidleack[1]	MSGIF idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[8]	ECD3_poidleack[1]	ECD3 idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[7]	MC3_poidleack[1]	MC3 idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[6]	IPE3_poidleack[1]	iPE3 idle acknowledge (idle domain 0)	-

**Table 18-28. Observability Signals at IVAHD Subsystem Level (continued)**

<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside IVAHD Subsystem</b>	<b>IVAHD SS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal description</b>	<b>Logical value meaning</b>
hwobs_int_ivahd[5]	CALC3_poidleack[1]	CALC3 idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[4]	ILF3_poidleack[1]	iLF3 idle acknowledge s (idle domain 0)	-
hwobs_int_ivahd[3]	IME3_poidleack[1]	iME3 idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[2]	VDMA_poidleack[1]	vDMA idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[1]	ICONT2_poidleack[1]	ICONT2 idle acknowledge (idle domain 0)	-
hwobs_int_ivahd[0]	ICONT1_poidleack[1]	ICONT1 idle acknowledge (idle domain 0)	-

#### 18.4.11.5.3 Display Subsystem Observable Signals

The DSS observable signals described in [Table 18-29](#) are directly mapped to DSS observability inputs of the SYSCTRL\_GENERAL\_CORE Module, and no additional multiplexing is performed inside the DSS subsystem wrapper.

**Table 18-29. Observability Signals at DSS Subsystem Level**

<b>SYSCTRL_GENERAL_CORE observability inputs tied to signals inside DSS Subsystem</b>	<b>DSS observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal Description</b>	<b>Logical value meaning</b>
hwobs_dssvenctvdetgp	DSS_VENC_TVDETP	DSS VENC tvdetgp	Active high pulse to detect the TV connection
hwobs_dssdacpwrndaczcvbs_y	DSS_DAC_PWR_DND ACZCVBS/Y	DSS DAC pwrnd dac zcvbs_y - Power down for DAC Y	0:DAC Y put in power-down mode 1:DAC Y put in ACTIVE state
hwobs_dssdacpwrndaczchroma	DSS_DAC_PWR_DND ACZCHROMA	DSS DAC pwrnd dac zchroma - Power down for DAC CHROMA;	0:DAC CHROMA put in power-down mode 1:DAC CHROMA put in ACTIVE state
hwobs_dssdacpwrndaccvbs	DSS_DAC_PWR_DND ACCVBS	DSS DAC pwrnd dac zcvbs - Power down for DAC CVBS;	0:DAC CVBS put in power-down mode 1:DAC CVBS put in ACTIVE state
hwobs_dssdacpwrndbgz	DSS_DAC_PWR_DNB GZ	DSS DAC pwrnd bgz - Band Gap power down for DAC ;	0: DAC Bandgap put in power-down mode 1: DAC Bandgap put in ACTIVE state

#### 18.4.11.5.4 DSP Subsystem Observable Signals

This information is not available in the public domain

#### 18.4.11.5.5 ABE Subsystem Observable Signals

The Audio - backend observable signals described in [Table 18-30](#) are directly mapped to the ABE dedicated observability inputs - hwobs\_int\_abe[31:0] of the SYSCTRL\_GENERAL\_CORE Module, and no additional multiplexing is performed inside the ABE .

**Table 18-30. Observability Signals at ABE Subsystem Level**

SYSCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS	SYSCTRL_GENERAL_CORE observability inputs tied to signals inside Cortex-A9 MPUSS	Signal Description	Logical value meaning
hwobs_int_abe[31:26]	-	-	-
hwobs_int_abe[25:24]	OPP_DIV	OPP DIVIDER bit[1:0]	-
hwobs_int_abe[23]	AESS_to_MPU_IRQ	IRQ sent from AESS to MCU. Refer to <a href="#">Chapter 12, ABE</a> .	-
hwobs_int_abe[22]	AESS_to_DSPSS_IRQ	IRQ sent from AESS to DSP. Refer to <a href="#">, Interrupt Requests, and Chapter 12, ABE</a> .	-
hwobs_int_abe[21]	Reserved	-	-
hwobs_int_abe[20]	GPT_WDT_PWM_INP	PWM received from the GPTIMER or watchdog (OR)	-
hwobs_int_abe[19]	AESS_DMAREQ_ALL	(logical OR will all dma req out)	High while outgoing DMA request from Audio Engine have not been acknowledged by the host.
hwobs_int_abe[18]	AESS_DMAREQ_PER	(logical OR with all dma req coming from the peripheral)	High when incoming dmarequest from Audio Engine are still to be served.
hwobs_int_abe[17]	AESS_IDLEPO	the PO_IDLE AESS output	High when the Audio Engine execute a wait until an activation of the PI_WAKEUP pin has been done.
hwobs_int_abe[16:0]	RESERVED	-	-

**18.4.11.5.6 USBOTGHS Subsystem Observable Signals**

One of the General Wakeup Control observability inputs can be mapped to the USBOTGHS "start of frame signal" - hsubotg\_sofpulse through setting the [CONTROL\\_CORE\\_CONF\\_DEBUG\\_SEL\\_TST\\_11\[3:0\]](#) MODE =0x1.

**18.4.11.5.7 CM2 Susbsystem Observable Signals**

The SYSCTRL\_GENERAL\_CORE observability inputs - hwobs\_int\_cm2(i) , (i=0..31) are mapped to the signals debugcm2(i), (i=0..31) coming out of the PRCM submodule -Clock Manager 2 (CM2).

There are local multiplexing controls at the CM2 level that reside in the CM2\_DEBUG\_CFG register. The CM2 observability output signals are driven on the debugcm2(0) through debugcm2(7) lines. A certain mux set is selected by programming the CM2\_DEBUG\_CFG[7:0] SEL0 bit field. The remaining debugcm2(8) through debugcm2(31) lines are reserved for future use. For more information about the CM2 observability source select field, see [Section 3.11.30.1, INTRCONN\\_SOCKET\\_CM2 Register Summary](#) in [Chapter 3, Power, Reset, and Clock Management](#).

[Table 18-31](#) and [Table 18-32](#) describe CM2 observable signals specific to initiators and targets, respectively. [Table 18-33](#) describes other CM2 observable signals.

**Table 18-31. CM2 Observable Signals Specific to Initiators**

CM2_DEBUG_CFG [7:0] SEL0 local, hwobs mux-set selector	debugcm2(0)	debugcm2(1)	debugcm2(2)	debugcm2(3)	debugcm2(4)	debugcm2(5)	debugcm2(6)	debugcm2(7)
0x0	ifsm_ccptx_ccpt_stby_I0.MStandby	ifsm_ccptx_ccpt_stby_I0.init_Wait	ifsm_ccptx_ccpt_stby_I0.module_mwakeup	reserved	reserved	reserved	-	-
0x1	reserved	reserved	reserved	ifsm_dss_dss_stby_pm_I0.MStandby	ifsm_dss_dss_stby_pm_I0.init_Wait	ifsm_dss_dss_stby_pm_I0.module_mwakeup	-	-
0x2	ifsm_cortex_m3_pm_master_I0.MStandby	ifsm_cortex_m3_pm_master_I0.init_Wait	ifsm_cortex_m3_pm_master_I0.module_mwakeup	reserved	reserved	reserved	-	-
0x3	ifsm_fdif_standby_prot_I0.MStandby	ifsm_fdif_standby_prot_I0.init_Wait	ifsm_fdif_standby_prot_I0.module_mwakeup	ifsm_gfx_stby_prot_I0.MStandby	ifsm_gfx_stby_prot_I0.init_Wait	ifsm_gfx_stby_prot_I0.module_mwakeup	-	-
0x4	ifsm_hsi_hsi_standby_I0.MStandby	ifsm_hsi_hsi_standby_I0.init_Wait	ifsm_hsi_hsi_standby_I0.module_mwakeup	ifsm_iss_standby_iss_I0.MStandby	ifsm_iss_standby_iss_I0.init_Wait	ifsm_iss_standby_iss_I0.module_mwakeup	-	-
0x5	ifsm_ivahd_master_I0.MStandby	ifsm_ivahd_master_I0.init_Wait	ifsm_ivahd_master_I0.module_mwakeup	ifsm_mmc1_stby_prot_I0.MStandby	ifsm_mmc1_stby_prot_I0.init_Wait	ifsm_mmc1_stby_prot_I0.module_mwakeup	-	-
0x6	ifsm_mmc2_stby_prot_I0.MStandby	ifsm_mmc2_stby_prot_I0.init_Wait	ifsm_mmc2_stby_prot_I0.module_mwakeup	reserved	reserved	reserved	-	-
0x7	reserved	reserved	reserved	reserved	reserved	reserved	-	-
0x8	reserved	reserved	reserved	reserved	reserved	reserved	-	-
0x9	ifsm_sdma_stby_prot_I0.MStandby	ifsm_sdma_stby_prot_I0.init_Wait	ifsm_sdma_stby_prot_I0.module_mwakeup	reserved	reserved	reserved	-	-
0xA	reserved	reserved	reserved	ifsm_usb_host_fs_standby_prot_I0.MStandby	ifsm_usb_host_fs_standby_prot_I0.init_Wait	ifsm_usb_host_fs_standby_prot_I0.module_mwakeup	-	-
0xB	ifsm_usb_host_usbhosths_masterif_pwr_I0.MStandby	ifsm_usb_host_usbhosths_masterif_pwr_I0.init_Wait	ifsm_usb_host_usbhosths_masterif_pwr_I0.module_mwakeup	ifsm_usb_otg_usbotghs_master_I0.MStandby	ifsm_usb_otg_usbotghs_master_I0.init_Wait	ifsm_usb_otg_usbotghs_master_I0.module_mwakeup	-	-

**Table 18-32. CM2 Observable Signals Specific to Targets**

CM2_DEBUG_CFG [7:0] SEL0, local hwobs mux-set selector	debugcm2(0)	debugcm2(1)	debugcm2(2)	debugcm2(3)	debugcm2(4)	debugcm2(5)	debugcm2(6)	debugcm2(7)
0xC	reserved	reserved	reserved	reserved	reserved	reserved	reserved	-
0xD	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0xE	tfsm_ccptx_ccpt_idle_I0. idleReq	tfsm_ccptx_ccpt_idle_I0. SidleAck	tfsm_ccptx_ccpt_idle_I0. module_swakeup	tfsm_ccptx_ccpt_idle_I0. fclken_IP	tfsm_cm2_profiling_I0. IdleReq	tfsm_cm2_profiling_I0. SidleAck	tfsm_cm2_profiling_I0. module_swakeup	tfsm_cm2_profiling_I0. fclken_IP
0xF	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x10	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x11	reserved	reserved	reserved	reserved	tfsm_dmtimer10_dmtimer_dmc1ms _power_I0. IdleReq	tfsm_dmtimer10_dmtimer_dmc1ms _power_I0. SidleAck	tfsm_dmtimer10_dmtimer_dmc1ms _power_I0. module_swakeup	tfsm_dmtimer10_dmtimer_dmc1ms _power_I0. fclken_IP
0x12	tfsm_dmtimer11_dmtimer_dmc_power_I0. idleReq	tfsm_dmtimer11_dmtimer_dmc_power_I0. SidleAck	tfsm_dmtimer11_dmtimer_dmc_power_I0. module_swakeup	tfsm_dmtimer11_dmtimer_dmc_power_I0. fclken_IP	tfsm_dmtimer2_dmtimer_dmc1ms_power_I0. idleReq	tfsm_dmtimer2_dmtimer_dmc1ms_power_I0. SidleAck	tfsm_dmtimer2_dmtimer_dmc1ms_power_I0. module_swakeup	tfsm_dmtimer2_dmtimer_dmc1ms_power_I0. fclken_IP
0x13	tfsm_dmtimer3_dmtimer_dmc_power_I0. idleReq	tfsm_dmtimer3_dmtimer_dmc_power_I0. SidleAck	tfsm_dmtimer3_dmtimer_dmc_power_I0. module_swakeup	tfsm_dmtimer3_dmtimer_dmc_power_I0. fclken_IP	tfsm_dmtimer4_dmtimer_dmc_power_I0. idleReq	tfsm_dmtimer4_dmtimer_dmc_power_I0. SidleAck	tfsm_dmtimer4_dmtimer_dmc_power_I0. module_swakeup	tfsm_dmtimer4_dmtimer_dmc_power_I0. fclken_IP
0x14	tfsm_dmtimer9_dmtimer_dmc_power_I0. idleReq	tfsm_dmtimer9_dmtimer_dmc_power_I0. SidleAck	tfsm_dmtimer9_dmtimer_dmc_power_I0. module_swakeup	tfsm_dmtimer9_dmtimer_dmc_power_I0. fclken_IP	-	-	-	-
0x15	tfsm_dss_dss_idle_pm_I0. idleReq	tfsm_dss_dss_idle_pm_I0. SidleAck	tfsm_dss_dss_idle_pm_I0. module_swakeup(0)	tfsm_dss_dss_idle_pm_I0. module_swakeup(1)	tfsm_dss_dss_idle_pm_I0. module_swakeup(2)	tfsm_dss_dss_idle_pm_I0. module_swakeup(3)	tfsm_dss_dss_idle_pm_I0. module_swakeup(4)	tfsm_dss_dss_idle_pm_I0. fclken_IP
0x16	tfsm_cortex_m3_pm_slave_I0. idleReq	tfsm_cortex_m3_pm_slave_I0. SidleAck	tfsm_cortex_m3_pm_slave_I0. module_swakeup	tfsm_cortex_m3_pm_slave_I0. fclken_IP	tfsm_elm_idle_I0. idleReq	tfsm_elm_idle_I0. SidleAck	tfsm_elm_idle_I0. module_swakeup	tfsm_elm_idle_I0. fclken_IP

**Table 18-32. CM2 Observable Signals Specific to Targets (continued)**

CM2_DEBUG_CFG [7:0] SEL0, local hwobs mux-set selector	debugcm2(0)	debugcm2(1)	debugcm2(2)	debugcm2(3)	debugcm2(4)	debugcm2(5)	debugcm2(6)	debugcm2(7)
0x17	reserved	reserved	reserved	reserved	tfsm_emif1_ l3idleprm_I0. idleReq	tfsm_emif1_ l3idleprm_I0. SidleAck	tfsm_emif1_ l3idleprm_I0. module_swakeup	tfsm_emif1_ l3idleprm_I0. fclken_IP
0x18	tfsm_emif2_ l3idleprm_I0. idleReq	tfsm_emif2_ l3idleprm_I0. SidleAck	tfsm_emif2_ l3idleprm_I0. module_swakeup	tfsm_emif2_ l3idleprm_I0. fclken_IP	reserved	reserved	reserved	reserved
0x19	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x1A	tfsm_fdif_ idle_prot_I0. idleReq	tfsm_fdif_ idle_prot_I0. SidleAck	tfsm_fdif_ idle_prot_I0. module_swakeup	tfsm_fdif_ idle_prot_I0. fclken_IP	tfsm_gfx_ idle_prot_I0. idleReq	tfsm_gfx_ idle_prot_I0. SidleAck	tfsm_gfx_ idle_prot_I0. module_swakeup	tfsm_gfx_ idle_prot_I0. fclken_IP
0x1B	tfsm_gpio2_ gpio_v2_ power_I0. idleReq	tfsm_gpio2_ gpio_v2_ power_I0. SidleAck	tfsm_gpio2_ gpio_v2_ power_I0. module_swakeup(0)	tfsm_gpio2_ gpio_v2_ power_I0. module_swakeup(1)	tfsm_gpio2_ gpio_v2_ power_I0. fclken_IP	-	-	-
0x1C	tfsm_gpio3_ gpio_v2_ power_I0. idleReq	tfsm_gpio3_ gpio_v2_ power_I0. SidleAck	tfsm_gpio3_ gpio_v2_ power_I0. module_swakeup(0)	tfsm_gpio3_ gpio_v2_ power_I0. module_swakeup(1)	tfsm_gpio3_ gpio_v2_ power_I0. fclken_IP	-	-	-
0x1D	tfsm_gpio4_ gpio_v2_ power_I0. idleReq	tfsm_gpio4_ gpio_v2_ power_I0. SidleAck	tfsm_gpio4_ gpio_v2_ power_I0. module_swakeup(0)	tfsm_gpio4_ gpio_v2_ power_I0. module_swakeup(1)	tfsm_gpio4_ gpio_v2_ power_I0. fclken_IP	-	-	-
0x1E	tfsm_gpio5_ gpio_v2_ power_I0. idleReq	tfsm_gpio5_ gpio_v2_ power_I0. SidleAck	tfsm_gpio5_ gpio_v2_ power_I0. module_swakeup(0)	tfsm_gpio5_ gpio_v2_ power_I0. module_swakeup(1)	tfsm_gpio5_ gpio_v2_ power_I0. fclken_IP	-	-	-
0x1F	tfsm_gpio6_ gpio_v2_ power_I0. idleReq	tfsm_gpio6_ gpio_v2_ power_I0. SidleAck	tfsm_gpio6_ gpio_v2_ power_I0. module_swakeup(0)	tfsm_gpio6_ gpio_v2_ power_I0. module_swakeup(1)	tfsm_gpio6_ gpio_v2_ power_I0. fclken_IP	-	-	-
0x20	tfsm_gpmc_ idle_I0. idleReq	tfsm_gpmc_ idle_I0. SidleAck	tfsm_gpmc_ idle_I0. module_swakeup	tfsm_gpmc_ idle_I0. fclken_IP	tfsm_hdq1w_ hdq1w_idle_I0. idleReq	tfsm_hdq1w_ hdq1w_idle_I0. SidleAck	tfsm_hdq1w_ hdq1w_idle_I0. module_swakeup	tfsm_hdq1w_ hdq1w_idle_I0. fclken_IP
0x21	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved

**Table 18-32. CM2 Observable Signals Specific to Targets (continued)**

CM2_DEBUG_CFG [7:0] SEL0, local hwobs mux-set selector	debugcm2(0)	debugcm2(1)	debugcm2(2)	debugcm2(3)	debugcm2(4)	debugcm2(5)	debugcm2(6)	debugcm2(7)
0x22	tfsm_hsi_hsi_idle_I0. idleReq	tfsm_hsi_hsi_idle_I0. SidleAck	tfsm_hsi_hsi_idle_I0. module_swakeup(0)	tfsm_hsi_hsi_idle_I0. module_swakeup(1)	tfsm_hsi_hsi_idle_I0. module_swakeup(2)	tfsm_hsi_hsi_idle_I0. fclken_IP	-	-
0x23	tfsm_hw_semaphore_hwsem_idle_I0. idleReq	tfsm_hw_semaphore_hwsem_idle_I0. SidleAck	tfsm_hw_semaphore_hwsem_idle_I0. module_swakeup	tfsm_hw_semaphore_hwsem_idle_I0. fclken_IP	-	-	-	-
0x24	tfsm_i2c1_mshsi2c_power_I0. idleReq	tfsm_i2c1_mshsi2c_power_I0. SidleAck	tfsm_i2c1_mshsi2c_power_I0. module_swakeup(0)	tfsm_i2c1_mshsi2c_power_I0. module_swakeup(1)	tfsm_i2c1_mshsi2c_power_I0. fclken_IP	-	-	-
0x25	tfsm_i2c2_mshsi2c_power_I0. idleReq	tfsm_i2c2_mshsi2c_power_I0. SidleAck	tfsm_i2c2_mshsi2c_power_I0. module_swakeup(0)	tfsm_i2c2_mshsi2c_power_I0. module_swakeup(1)	tfsm_i2c2_mshsi2c_power_I0. fclken_IP	-	-	-
0x26	tfsm_i2c3_mshsi2c_power_I0. idleReq	tfsm_i2c3_mshsi2c_power_I0. SidleAck	tfsm_i2c3_mshsi2c_power_I0. module_swakeup(0)	tfsm_i2c3_mshsi2c_power_I0. module_swakeup(1)	tfsm_i2c3_mshsi2c_power_I0. fclken_IP	-	-	-
0x27	tfsm_i2c4_mshsi2c_power_I0. idleReq	tfsm_i2c4_mshsi2c_power_I0. SidleAck	tfsm_i2c4_mshsi2c_power_I0. module_swakeup(0)	tfsm_i2c4_mshsi2c_power_I0. module_swakeup(1)	tfsm_i2c4_mshsi2c_power_I0. fclken_IP	-	-	-
0x28	tfsm_i2c5_mshsi2c_power_I0. idleReq	tfsm_i2c5_mshsi2c_power_I0. SidleAck	tfsm_i2c5_mshsi2c_power_I0. module_swakeup(0)	tfsm_i2c5_mshsi2c_power_I0. module_swakeup(1)	tfsm_i2c5_mshsi2c_power_I0. fclken_IP	-	-	-
0x29	tfsm_iss_idle_iss_I0. idleReq	tfsm_iss_idle_iss_I0. SidleAck	tfsm_iss_idle_iss_I0. module_swakeup	tfsm_iss_idle_iss_I0. fclken_IP	tfsm_ivahd_idle0_I0. idleReq	tfsm_ivahd_idle0_I0. SidleAck	tfsm_ivahd_idle0_I0. module_swakeup	tfsm_ivahd_idle0_I0. fclken_IP
0x2A	reserved	reserved	reserved	reserved	tfsm_l3_interconnect_1_idle_prot_I0. idleReq	tfsm_l3_interconnect_1_idle_prot_I0. SidleAck	tfsm_l3_interconnect_1_idle_prot_I0. module_swakeup	tfsm_l3_interconnect_1_idle_prot_I0. fclken_IP

**Table 18-32. CM2 Observable Signals Specific to Targets (continued)**

CM2_DEBUG_CFG [7:0] SEL0, local hwobs mux-set selector	debugcm2(0)	debugcm2(1)	debugcm2(2)	debugcm2(3)	debugcm2(4)	debugcm2(5)	debugcm2(6)	debugcm2(7)
0x2B	tfsml3_interconnect_2_idle_prot_I0. idleReq	tfsml3_interconnect_2_idle_prot_I0. SidleAck	tfsml3_interconnect_2_idle_prot_I0. module_wakeup	tfsml3_interconnect_2_idle_prot_I0. fclken_IP	tfsml3_interconnect_3_idle_prot_I0. idleReq	tfsml3_interconnect_3_idle_prot_I0. SidleAck	tfsml3_interconnect_3_idle_prot_I0. module_wakeup	tfsml3_interconnect_3_idle_prot_I0. fclken_IP
0x2C	tfsml4_cfg_interconnect_idle_prot_I0. idleReq	tfsml4_cfg_interconnect_idle_prot_I0. SidleAck	tfsml4_cfg_interconnect_idle_prot_I0. module_wakeup	tfsml4_cfg_interconnect_idle_prot_I0. fclken_IP	tfsml4_per_interconnect_idle_prot_I0. idleReq	tfsml4_per_interconnect_idle_prot_I0. SidleAck	tfsml4_per_interconnect_idle_prot_I0. module_wakeup	tfsml4_per_interconnect_idle_prot_I0. fclken_IP
0x2D	tfsmlmailbox_mbx_idle_I0. idleReq	tfsmlmailbox_mbx_idle_I0. SidleAck	tfsmlmailbox_mbx_idle_I0. module_wakeup	tfsmlmailbox_mbx_idle_I0. fclken_IP	-	-	-	-
0x2E	tfsmlmccasp2_cfgidle_p_cfg_I0. idleReq	tfsmlmccasp2_cfgidle_p_cfg_I0. SidleAck	tfsmlmccasp2_cfgidle_p_cfg_I0. module_wakeup(0)	tfsmlmccasp2_cfgidle_p_cfg_I0. module_wakeup(1)	tfsmlmccasp2_cfgidle_p_cfg_I0. fclken_IP			
0x2F	reserved	reserved	reserved	reserved	reserved	-	-	-
0x30	tfsmlmcbbsp4_mcbssplp_power_I0. idleReq	tfsmlmcbbsp4_mcbssplp_power_I0. SidleAck	tfsmlmcbbsp4_mcbssplp_power_I0. module_wakeup	tfsmlmcbbsp4_mcbssplp_power_I0. fclken_IP	tfsmlmccspi1_spi_power_I0. idleReq	tfsmlmccspi1_spi_power_I0. SidleAck	tfsmlmccspi1_spi_power_I0. module_wakeup	tfsmlmccspi1_spi_power_I0. fclken_IP
0x31	tfsmlmccspi2_spi_power_I0. idleReq	tfsmlmccspi2_spi_power_I0. SidleAck	tfsmlmccspi2_spi_power_I0. module_wakeup	tfsmlmccspi2_spi_power_I0. fclken_IP	tfsmlmccspi3_spi_power_I0. idleReq	tfsmlmccspi3_spi_power_I0. SidleAck	tfsmlmccspi3_spi_power_I0. module_wakeup	tfsmlmccspi3_spi_power_I0. fclken_IP
0x32	tfsmlmccspi4_spi_power_I0. idleReq	tfsmlmccspi4_spi_power_I0. SidleAck	tfsmlmccspi4_spi_power_I0. module_wakeup	tfsmlmccspi4_spi_power_I0. fclken_IP	reserved	reserved	reserved	reserved
0x33	tfsmlmmc1_idle_prot_I0. idleReq	tfsmlmmc1_idle_prot_I0. SidleAck	tfsmlmmc1_idle_prot_I0. module_wakeup	tfsmlmmc1_idle_prot_I0. fclken_IP	tfsmlmmc2_idle_prot_I0. idleReq	tfsmlmmc2_idle_prot_I0. SidleAck	tfsmlmmc2_idle_prot_I0. module_wakeup	tfsmlmmc2_idle_prot_I0. fclken_IP
0x34	reserved	reserved	reserved	reserved	tfsmlmmc_sd3_mmchs_idle_I0. idleReq	tfsmlmmc_sd3_mmchs_idle_I0. SidleAck	tfsmlmmc_sd3_mmchs_idle_I0. module_wakeup	tfsmlmmc_sd3_mmchs_idle_I0. fclken_IP



**Table 18-32. CM2 Observable Signals Specific to Targets (continued)**

CM2_DEBUG_CFG [7:0] SEL0, local hwobs mux-set selector	debugcm2(0)	debugcm2(1)	debugcm2(2)	debugcm2(3)	debugcm2(4)	debugcm2(5)	debugcm2(6)	debugcm2(7)
0x35	tfsm_mmc_sd4_mmchs_idle_I0. idleReq	tfsm_mmc_sd4_mmchs_idle_I0. SidleAck	tfsm_mmc_sd4_mmchs_idle_I0. module_swakeup	tfsm_mmc_sd4_mmchs_idle_I0. fclken_IP	tfsm_mmc_sd5_mmchs_idle_I0. idleReq	tfsm_mmc_sd5_mmchs_idle_I0. SidleAck	tfsm_mmc_sd5_mmchs_idle_I0. module_swakeup	tfsm_mmc_sd5_mmchs_idle_I0. fclken_IP
0x36	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x37	reserved	reserved	reserved	reserved	tfsm_ocmc_ram_ocmcram_idle_I0. idleReq	tfsm_ocmc_ram_ocmcram_idle_I0. SidleAck	tfsm_ocmc_ram_ocmcram_idle_I0. module_swakeup	tfsm_ocmc_ram_ocmcram_idle_I0. fclken_IP
0x38	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x39	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x3A	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x3B	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x3C	reserved	reserved	reserved	reserved	tfsm_sdma_idle_prot_I0. idleReq	tfsm_sdma_idle_prot_I0. SidleAck	tfsm_sdma_idle_prot_I0. module_swakeup	tfsm_sdma_idle_prot_I0. fclken_IP
0x3D	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x3E	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x3F	tfsm_smartreflex_core_sr_idle_I0. idleReq	tfsm_smartreflex_core_sr_idle_I0. SidleAck	tfsm_smartreflex_core_sr_idle_I0. module_swakeup	tfsm_smartreflex_core_sr_idle_I0. fclken_IP	tfsm_smartreflex_iva_sr_idle_I0. idleReq	tfsm_smartreflex_iva_sr_idle_I0. SidleAck	tfsm_smartreflex_iva_sr_idle_I0. module_swakeup	tfsm_smartreflex_iva_sr_idle_I0. fclken_IP
0x40	tfsm_smartreflex_mpu_sr_idle_I0. idleReq	tfsm_smartreflex_mpu_sr_idle_I0. SidleAck	tfsm_smartreflex_mpu_sr_idle_I0. module_swakeup	tfsm_smartreflex_mpu_sr_idle_I0. fclken_IP	reserved	reserved	reserved	reserved
0x41	tfsm_uart1_uartirdacir_power_I0. idleReq	tfsm_uart1_uartirdacir_power_I0. SidleAck	tfsm_uart1_uartirdacir_power_I0. module_swakeup	tfsm_uart1_uartirdacir_power_I0. fclken_IP	tfsm_uart2_uartirdacir_power_I0. idleReq	tfsm_uart2_uartirdacir_power_I0. SidleAck	tfsm_uart2_uartirdacir_power_I0. module_swakeup	tfsm_uart2_uartirdacir_power_I0. fclken_IP
0x42	tfsm_uart3_uartirdacir_power_I0. idleReq	tfsm_uart3_uartirdacir_power_I0. SidleAck	tfsm_uart3_uartirdacir_power_I0. module_swakeup	tfsm_uart3_uartirdacir_power_I0. fclken_IP	tfsm_uart4_uartirdacir_power_I0. idleReq	tfsm_uart4_uartirdacir_power_I0. SidleAck	tfsm_uart4_uartirdacir_power_I0. module_swakeup	tfsm_uart4_uartirdacir_power_I0. fclken_IP
0x43	reserved	reserved	reserved	reserved	tfsm_usb_host_fs_idle_prot_I0. idleReq	tfsm_usb_host_fs_idle_prot_I0. SidleAck	tfsm_usb_host_fs_idle_prot_I0. module_swakeup	tfsm_usb_host_fs_idle_prot_I0. fclken_IP

**Table 18-32. CM2 Observable Signals Specific to Targets (continued)**

CM2_DEBUG_CFG [7:0] SEL0, local hwobs mux-set selector	debugcm2(0)	debugcm2(1)	debugcm2(2)	debugcm2(3)	debugcm2(4)	debugcm2(5)	debugcm2(6)	debugcm2(7)
0x44	tfsm_usb_host_usbhosths_regif_pwr_I0. idleReq	tfsm_usb_host_usbhosths_regif_pwr_I0. SidleAck	tfsm_usb_host_usbhosths_regif_pwr_I0. module_swakeup	tfsm_usb_host_usbhosths_regif_pwr_I0. fclken_IP	tfsm_usb_otg_usbotghs_slave_I0. idleReq	tfsm_usb_otg_usbotghs_slave_I0. SidleAck	tfsm_usb_otg_usbotghs_slave_I0. module_swakeup	tfsm_usb_otg_usbotghs_slave_I0. fclken_IP
0x45	tfsm_usb_tll_usb_tll_power_I0. idleReq	tfsm_usb_tll_usb_tll_power_I0. SidleAck	tfsm_usb_tll_usb_tll_power_I0. module_swakeup	tfsm_usb_tll_usb_tll_power_I0. fclken_IP	reserved	reserved	reserved	reserved
0x46	reserved	reserved	reserved	reserved	-	-	-	-

**Table 18-33. CM2 - Other Observable Signals**

CM2_DEBUG_CFG [7:0] SEL0, local hwobs mux-set selector	debugcm2(0)	debugcm2(1)	debugcm2(2)	debugcm2(3)	debugcm2(4)	debugcm2(5)	debugcm2(6)	debugcm2(7)
0xA7	-	-	-	-	forceactive_cortex_m3_cpu1	forceactive_cortex_m3_cpu2	inhibitsleep_cortex_m3_cpu1	inhibitsleep_cortex_m3_cpu2

#### 18.4.11.5.8 CM1 Subsystem Observable Signals

The General Wakeup Control observability inputs - hwobs\_int\_cm1(i) , (i=0..31) are mapped to the signals debugcm1(i), (i=0..31) coming out of the PRCM submodule - Clock Manager 1 (CM1).

There are local multiplexing controls at the CM1 level that reside in the CM1\_DEBUG\_CFG register. The CM1 observability output signals are driven on the debugcm1(0) through debugcm1(7) lines. A certain mux set is selected by programming the CM1\_DEBUG\_CFG[6:0] SEL0 bit field. The remaining debugcm1(8) through debugcm1(31) lines are reserved for future use. For more information about the CM1 observability source select field, see [Section 3.11.22.1, INTRCONN\\_SOCKET\\_CM1 Register Summary](#) in [Chapter 3, Power, Reset, and Clock Management](#).

[Table 18-34](#) and [Table 18-35](#) list CM1 observable signals specific to initiators and targets, respectively. [Table 18-36](#) lists other CM1 observable signals.

**Table 18-34. CM1 Observable Signals Specific to Initiators**

CM1_DEBUG_CFG [6:0] SEL0 local hwobs mux-set selector	debugcm1(0)	debugcm1(1)	debugcm1(2)	debugcm1(3)	debugcm1(4)	debugcm1(5)	debugcm1(6)	debugcm1(7)
0x0	ifsm_aess_standby_prot_I0. MStandby	ifsm_aess_standby_prot_I0. init_Wait	ifsm_aess_standby_prot_I0. module_mwakeup	ifsm_cortex_a9_mpu_pm_I0. MStandby	ifsm_cortex_a9_mpu_pm_I0. init_Wait	ifsm_cortex_a9_mpu_pm_I0. module_mwakeup (0)	ifsm_cortex_a9_mpu_pm_I0. module_mwakeup (1)	ifsm_cortex_a9_mpu_pm_I0. module_mwakeup (2)
0x1	ifsm_dspss_stdby_prot_I0. MStandby	ifsm_dspss_stdby_prot_I0. init_Wait	ifsm_dspss_stdby_prot_I0. module_mwakeup	See <a href="#">Table 18-35</a>	See <a href="#">Table 18-35</a>	See <a href="#">Table 18-35</a>	See <a href="#">Table 18-35</a>	See <a href="#">Table 18-35</a>

**Table 18-35. CM1 Observable Signals Specific to Targets**

CM1_DEBUG_CFG [6:0] SEL0 local hwobs mux-set selector	debugcm1(0)	debugcm1(1)	debugcm1(2)	debugcm1(3)	debugcm1(4)	debugcm1(5)	debugcm1(6)	debugcm1(7)
0x1	See <a href="#">Table 18-34</a>	See <a href="#">Table 18-34</a>	See <a href="#">Table 18-34</a>	tfsm_aess_idle_prot_I0. idleReq	tfsm_aess_idle_prot_I0. SidleAck	tfsm_aess_idle_prot_I0. module_swakeup	tfsm_aess_idle_prot_I0. fclken_IP	-
0x2	tfsm_cm1_profiling_I0. idleReq	tfsm_cm1_profiling_I0. SidleAck	tfsm_cm1_profiling_I0. module_swakeup	tfsm_cm1_profiling_I0. fclken_IP	-	-	-	-
0x3	tfsm_dmic_idle_prot_I0. idleReq	tfsm_dmic_idle_prot_I0. SidleAck	tfsm_dmic_idle_prot_I0. module_swakeup(0)	tfsm_dmic_idle_prot_I0. module_swakeup(1)	tfsm_dmic_idle_prot_I0. fclken_IP	-	-	-
0x4	tfsm_l4_audio_interconnect_idle_prot_I0. idleReq	tfsm_l4_audio_interconnect_idle_prot_I0. SidleAck	tfsm_l4_audio_interconnect_idle_prot_I0. module_swakeup	tfsm_l4_audio_interconnect_idle_prot_I0. fclken_IP	-	-	-	-
0x5	tfsm_mcaspcfgidle_p_cfg_I0. idleReq	tfsm_mcaspcfgidle_p_cfg_I0. SidleAck	tfsm_mcaspcfgidle_p_cfg_I0. module_swakeup(0)	tfsm_mcaspcfgidle_p_cfg_I0. module_swakeup(1)	tfsm_mcaspcfgidle_p_cfg_I0. fclken_IP	-	-	-
0x6	tfsm_mcbbsp1_mcbssplp_power_I0. idleReq	tfsm_mcbbsp1_mcbssplp_power_I0. SidleAck	tfsm_mcbbsp1_mcbssplp_power_I0. module_swakeup	tfsm_mcbbsp1_mcbssplp_power_I0. fclken_IP	tfsm_mcbbsp2_mcbssplp_power_I0. idleReq	tfsm_mcbbsp2_mcbssplp_power_I0. SidleAck	tfsm_mcbbsp2_mcbssplp_power_I0. module_swakeup	tfsm_mcbbsp2_mcbssplp_power_I0. fclken_IP

**Table 18-35. CM1 Observable Signals Specific to Targets (continued)**

CM1_DEBUG_CFG [6:0] SEL0 local hwobs mux-set selector	debugcm1(0)	debugcm1(1)	debugcm1(2)	debugcm1(3)	debugcm1(4)	debugcm1(5)	debugcm1(6)	debugcm1(7)
0x7	tfs_mcbssp3_mcbssplp_power_I0. idleReq	tfs_mcbssp3_mcbssplp_power_I0. SidleAck	tfs_mcbssp3_mcbssplp_power_I0. module_swakeup	tfs_mcbssp3_mcbssplp_power_I0. fclken_IP	tfs_m_cortex_a9_dummyidle_I0. idleReq	tfs_m_cortex_a9_dummyidle_I0. SidleAck	tfs_m_cortex_a9_dummyidle_I0. module_swakeup	tfs_m_cortex_a9_dummyidle_I0. fclken_IP
0x8	tfs_m_pdm_idle_prot_I0. idleReq	tfs_m_pdm_idle_prot_I0. SidleAck	tfs_m_pdm_idle_prot_I0. module_swakeup(0)	tfs_m_pdm_idle_prot_I0. module_swakeup(1)	tfs_m_pdm_idle_prot_I0. fclken_IP	-	-	-
0x9	reserved	reserved	reserved	reserved	reserved	-	-	-
0xA	tfs_m_dspss_idle_prot_I0. idleReq	tfs_m_dspss_idle_prot_I0. SidleAck	tfs_m_dspss_idle_prot_I0. module_swakeup	tfs_m_dspss_idle_prot_I0. fclken_IP	tfs_m_timer5_dmtimer_dmc_power_I0. idleReq	tfs_m_timer5_dmtimer_dmc_power_I0. SidleAck	tfs_m_timer5_dmtimer_dmc_power_I0. module_swakeup	tfs_m_timer5_dmtimer_dmc_power_I0. fclken_IP
0xB	tfs_m_timer6_dmtimer_dmc_power_I0. idleReq	tfs_m_timer6_dmtimer_dmc_power_I0. SidleAck	tfs_m_timer6_dmtimer_dmc_power_I0. module_swakeup	tfs_m_timer6_dmtimer_dmc_power_I0. fclken_IP	tfs_m_timer7_dmtimer_dmc_power_I0. idleReq	tfs_m_timer7_dmtimer_dmc_power_I0. SidleAck	tfs_m_timer7_dmtimer_dmc_power_I0. module_swakeup	tfs_m_timer7_dmtimer_dmc_power_I0. fclken_IP
0xC	tfs_m_timer8_dmtimer_dmc_power_I0. idleReq	tfs_m_timer8_dmtimer_dmc_power_I0. SidleAck	tfs_m_timer8_dmtimer_dmc_power_I0. module_swakeup	tfs_m_timer8_dmtimer_dmc_power_I0. fclken_IP	tfs_m_wdt3_watchdog_power_I0. idleReq	tfs_m_wdt3_watchdog_power_I0. SidleAck	tfs_m_wdt3_watchdog_power_I0. module_swakeup	tfs_m_wdt3_watchdog_power_I0. fclken_IP

**Table 18-36. CM1 - Other Observable Signals**

CM1_DEBUG_CFG [6:0] SEL0 local hwobs mux-set selector	debugcm1(0)	debugcm1(1)	debugcm1(2)	debugcm1(3)	debugcm1(4)	debugcm1(5)	debugcm1(6)	debugcm1(7)
0x37	forceactive_cortex_m3_cpu1	forceactive_cortex_m3_cpu2	forceactive_ivahd_seq1	forceactive_ivahd_seq2	forceactive_mpu	forceactive_dspss	inhibitsleep_cortex_m3_cpu1	inhibitsleep_cortex_m3_cpu2
0x38	inhibitsleep_ivahd_seq1	inhibitsleep_ivahd_seq2	inhibitsleep_mpu	inhibitsleep_dspss	-	-	-	-

### 18.4.11.5.9 PRM Subsystem Observable Signals

The General Wakeup Control observability inputs - hwobs\_int\_prm(i), (i=0..31) are mapped to the signals debugprm(i), (i=0..31) coming out from the Power Reset Manager (PRM) submodule.

There are local multiplexing controls at the PRM level that reside in the PRM\_DEBUG\_CFG register. The PRM observability output signals are driven on the debugprm(0) through debugprm(7) lines. A certain mux set is selected by programming the PRM\_DEBUG\_CFG[6:0] SEL0 bit field. The remaining debugprm(8) through debugprm(31) lines are reserved for future use. For more information about the PRM observability source select field, see [Section 3.11.2.1, INTRCONN\\_SOCKET\\_PRM Register Summary](#) in [Chapter 3, Power, Reset, and Clock Management](#).

Table 18-37 through Table 18-40 list various PRM observable signals.

**Table 18-37. PRM - Observable Signals Specific to Targets**

PRM_DEBUG_CFG [6:0] SEL0, local hwobs mux-set selector	debugprm(0)	debugprm(1)	debugprm(2)	debugprm(3)	debugprm(4)	debugprm(5)	debugprm(6)	debugprm(7)
0x11	tfsm_gpio1_gpio_v2_power_I0. idleReq	tfsm_gpio1_gpio_v2_power_I0. SideAck	tfsm_gpio1_gpio_v2_power_I0. module_swakeup(0)	tfsm_gpio1_gpio_v2_power_I0. module_swakeup(1)	tfsm_gpio1_gpio_v2_power_I0. fclken_IP	-	-	-
0x12	tfsm_keyboard_kbd_power_I0. idleReq	tfsm_keyboard_kbd_power_I0. SideAck	tfsm_keyboard_kbd_power_I0. module_swakeup	tfsm_keyboard_kbd_power_I0. fclken_IP	tfsm_prm_profiling_I0. idleReq	tfsm_prm_profiling_I0. SideAck	tfsm_prm_profiling_I0. module_swakeup	tfsm_prm_profiling_I0. fclken_IP
0x13	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
0x14	tfsm_synctimer_synctimer32k_power_I0. idleReq	tfsm_synctimer_synctimer32k_power_I0. SideAck	tfsm_synctimer_synctimer32k_power_I0. module_swakeup	tfsm_synctimer_synctimer32k_power_I0. fclken_IP	reserved to non-GP devices	reserved to non-GP devices	reserved to non-GP devices	reserved to non-GP devices
0x15	tfsm_timer1_dmtimer_dmc1ms_power_I0. idleReq	tfsm_timer1_dmtimer_dmc1ms_power_I0. SideAck	tfsm_timer1_dmtimer_dmc1ms_power_I0. module_swakeup	tfsm_timer1_dmtimer_dmc1ms_power_I0. fclken_IP	reserved to non-GP devices	reserved to non-GP devices	reserved to non-GP devices	reserved to non-GP devices
0x16	reserved to non-GP devices	reserved to non-GP devices	reserved to non-GP devices	reserved to non-GP devices	tfsm_wdt2_watchdog_power_I0. idleReq	tfsm_wdt2_watchdog_power_I0. SideAck	tfsm_wdt2_watchdog_power_I0. module_swakeup	tfsm_wdt2_watchdog_power_I0. fclken_IP

**Table 18-37. PRM - Observable Signals Specific to Targets (continued)**

PRM_DEBUG_CFG [6:0] SELO, local hwobs mux-set selector	debugprm(0)	debugprm(1)	debugprm(2)	debugprm(3)	debugprm(4)	debugprm(5)	debugprm(6)	debugprm(7)
0x17	tfsn_wkup_l4_interconnect_idle_prot_I0. idleReq	tfsn_wkup_l4_interconnect_idle_prot_I0. SideAck	tfsn_wkup_l4_interconnect_idle_prot_I0. module_swakeup	tfsn_wkup_l4_interconnect_idle_prot_I0. fclken_IP	reserved	reserved	reserved	reserved

**Table 18-38. Power State Controller PGOODOUT/RET Signals**

PRM_DEBUG_CFG [6:0] SELO local hwobs mux-set selector	debugprm(0)	debugprm(1)	debugprm(2)	debugprm(3)	debugprm(4)	debugprm(5)	debugprm(6)	debugprm(7)
0x2F	-	-	-	-	-	-	pscon_audio_I0. pgoodout	pscon_audio_I0. ret
0x30	pscon_cam_I0. pgoodout	pscon_core_I0. pgoodout	pscon_core_I0. ret	Reserved to Non-GP devices	pscon_dss_I0. pgoodout	pscon_dss_I0. ret	reserved	pscon_gfx_I0. pgoodout
0x31	pscon_ivahd_I0. pgoodout	pscon_ivahd_I0. ret	pscon_l3_init_I0. pgoodout	pscon_l3_init_I0. ret	pscon_l4_per_I0. pgoodout	pscon_l4_per_I0. ret	pscon_mpu_I0. pgoodout	pscon_mpu_I0. ret
0x32	pscon_std_efuse_I0. pgoodout	pscon_dspss_I0. pgoodout	pscon_dspss_I0. ret	-	-	-	-	-

**Table 18-39. Reset Signals in PRM Module**

PRM_DEBUG_CFG [6:0] SELO local hwobs mux-set selector	debugprm(0)	debugprm(1)	debugprm(2)	debugprm(3)	debugprm(4)	debugprm(5)	debugprm(6)	debugprm(7)
0x38	reserved	reserved	inv_glbl_sw_rst_custom. ro_rst_globalcold_sw_n	reserved	sys_pwron_arst_n	inv_glbl_sw_rst_custom. ro_rst_globalwarm_sw_n	reserved	reserved
0x39	mpu_wdt_rst_n	reserved to Non-GP devices	sys_warmin_arst_n	c2c_warm_rst	audio_pwron_rst_n	audio_rst_n	always_on_cm1_pwron_rst_n	always_on_cm1_rst_n

**Table 18-39. Reset Signals in PRM Module (continued)**

PRM_DEBUG_CFG [6:0] SELO local hwobs mux-set selector	debugprm(0)	debugprm(1)	debugprm(2)	debugprm(3)	debugprm(4)	debugprm(5)	debugprm(6)	debugprm(7)
0x3A	always_on_core_pwrn_rst	always_on_core_rst_n	always_on_iva_rst_n	always_on_mpu_rst_n	cam_rst_n	cm2_pwrn_ret_rst_n	cm2_ret_rst_n	core_pwrn_ret_rst_n
0x3B	core_ret_rst_n	core_rst_n	reserved to Non-GP devices	dppl_rst_n	dppl_iva_pwrn_rst	dppl_l3_init_pwrn_ret_rst	dppl_mpu_pwrn_rst	dss_ret_rst_n
0x3C	dss_rst_n	cortex_m3_pwrn_rst_n	cortex_m3_ret_rst_n	cortex_m3_rst1_n	cortex_m3_rst2_n	cortex_m3_rst3_n	emif_ddr_phy_pwrn_rst_n	reserved
0x3D	reserved	reserved	gfx_rst_n	ivahd_pwrn_rst_n	ivahd_rst_n	ivahd_seq1_rst_n	ivahd_seq2_rst_n	l3_init_pwrn_rst_n
0x3E	l3_init_ret_rst_n	l3_init_rst_n	l4_per_pwrn_ret_rst_n	l4_per_ret_rst_n	l4_per_rst_n	mpu_pwrn_rst_n	mpu_rst_n	sdma_ret_rst_n
0x3F	std_efuse_pwrn_rst_n	dspss_dsp_rst_n	dspss_pwrn_rst_n	dspss_ret_rst_n	dspss_sys_rst_n	wkup_pwrn_rst_n	wkup_rst_n	wkup_sys_pwrn_rst_n
0x40	rstmngr_mpu_l2rstdisable_io.reset_data	cortex_m3_rstdone	mpu_pwrn_rstdone	dspss_rstdone	reserved	reserved	reserved	reserved
0x41	reserved	reserved	reserved	reserved	rstactst_audio_rstactst_io.logStatus	rstactst_cam_rstactst_io.logStatus	rstactst_core_rstactst_io.logStatus	Reserved to non-GP devices
0x42	rstactst_dss_rstactst_io.logStatus	reserved	rstactst_gfx_rstactst_io.logStatus	rstactst_ivahd_rstactst_io.logStatus	rstactst_l3_init_rstactst_io.logStatus	rstactst_l4_per_rstactst_io.logStatus	rstactst_mpu_rstactst_io.logStatus	rstactst_dspss_rstactst_io.logStatus
0x43	rstactst_globalwarm_rstactst_io.logStatus	Reserved to Non-GP devices	Reserved to Non-GP devices	rstactst_globalcold_rstactst_io.logStatus	reserved	reserved	reserved	reserved

**Table 18-40. PRM - Other Observable Signals**

PRM_DEBUG_CFG [6:0] SEL0 local hwobs mux-set selector	debugprm(0 )	debugprm(1)	debugprm(2)	debugprm(3)	debugprm(4)	debugprm(5)	debugprm(6)	debugprm(7)
0x44	reserved	forceactive_ cortex_m3_ cpu1	forceactive_ cortex_m3_ cpu2	forceactive_ ivahd_seq1	forceactive_ ivahd_seq2	forceactive_ mpu	forceactive_ dspss	-
0x45	inhibitsleep_ cortex_m3_ _cpu1	inhibitsleep_ cortex_m3_ _cpu2	inhibitsleep_ ivahd_seq1	inhibitsleep_ ivahd_seq2	inhibitsleep_ mpu	inhibitsleep_ dspss	reserved	reserved



#### 18.4.11.5.10 DPLL Clockview Channel Observable Signals Multiplexing

Table 18-41 shows multiplexing of the different DPLL clock outputs for each of the three available different clk\_view\_0, clk\_view\_1 and clk\_view\_2 channels.

**Table 18-41. Different DPLL Observable Clock output signals**

SYSCTRL_GENERAL_CORE observability inputs tied to signals inside different module DPLLs	Different DPLL observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs	Signal Description	DPLL CLKOUT selection per clk_view channel i CONTROL_CONF_CLK_SELi [2:0] MULT value, [i=0..2]
hwobs_coredpll_clkout	DPLL_TESTCLKOUT	CORE DPLL	0x0
hwobs_abedpll_clkout	PODPLLTESTCLKOUT	ABE DPLL	0x1
hwobs_perdppll_clkout	DPLL_TESTCLKOUT	PERIPH DPLL	0x2
hwobs_dsi1dppll_clkout	PODPLLTESTCLKOUT	DSI1 DPLL	0x3
hwobs_dsi2dppll_clkout	DPLL_TESTCLKOUT	DSI2 DPLL	0x4
reserved	reserved	-	0x5
hwobs_usbdppll_clkout	PODPLLTESTCLKOUT	USB DPLL	0x6
reserved	reserved	-	0x7

#### 18.4.11.5.11 Other CORE DPLL observable signals

Table 18-42 shows mapping of (other than CLKOUT) CORE dppll observability signals to the SYSCTRL\_GENERAL\_CORE dppll observing inputs. The selection of CORE DPLL as the source of the additional DPLL signals is done at SYSCTRL\_GENERAL\_CORE level through setting bit field MULT=0x0 inside all of the eight registers - CONTROL\_CONF\_DPLL\_FREQLOCK\_SEL through CONTROL\_CONF\_DPLL\_IDLE\_SEL.

**Table 18-42. Other CORE DPLL Observability Signals**

SYSCTRL_GENERAL_CORE observability inputs tied to CORE DPLL observable signals	Other CORE DPLL observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs	Signal Description	Logical value meaning
hwobs_coredpll_freqlock	COREdppll_freqlock	CORE DPLL frequency lock check	-
hwobs_coredpll_tinitz	COREdppll_tinitz	check CORE DPLL status, lock / reload time	-
hwobs_coredpll_phaselock	COREdppll_phaselock	CORE DPLL phase lock check	-
hwobs_coredpll_tenable	COREdppll_tenable	CORE DPLL Core RegM/N load enable input	-
hwobs_coredpll_tenablediv	COREdppll_tenablediv	CORE DPLL Core RegM2/N2 load enable input	-
hwobs_coredpll_bypassack	COREdppll_bypassack	CORE DPLL Bypass status signal	-
hwobs_coredpll_idle	COREdppll_idle	CORE DPLL Idle mode input	-

#### 18.4.11.5.12 Other ABE DPLL observable signals

Table 18-43 shows mapping of (other than CLKOUT) CORE dppll observability signals to the SYSCTRL\_GENERAL\_CORE dppll observing inputs. The selection of ABE DPLL as the source of the additional DPLL signals is done at SYSCTRL\_GENERAL\_CORE level through setting bit field MULT=0x1 inside all of the eight registers - CONTROL\_CONF\_DPLL\_FREQLOCK\_SEL through CONTROL\_CONF\_DPLL\_IDLE\_SEL.

**Table 18-43. Other ABE DPLL Observability Signals**

<b>SYSCTRL_GENERAL_CORE observability inputs tied to ABE DPLL observable signals</b>	<b>Other ABE DPLL observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal Description</b>	<b>Logical value meaning</b>
hwobs_abedpll_freqlock	dspdll_freqlock	ABE DPLL frequency lock check	-
hwobs_abedpll_tinitz	dspdll_tinitz	check ABE DPLL status, lock / relock time	-
hwobs_abedpll_phaselock	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_abedpll_tenable	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_abedpll_tenablediv	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_abedpll_bypassack	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_abedpll_idle	dspDpllPhaseLock	dspDPLL phase lock check	-

**18.4.11.5.13 Other PERIPH DPLL observable signals**

Table 18-44 shows mapping of (other than CLKOUT) CORE dpll observability signals to the SYSCTRL\_GENERAL\_CORE dpll observing inputs. The selection of PERIPH DPLL as the source of the additional DPLL signals is done at SYSCTRL\_GENERAL\_CORE level through setting bit field MULT=0x2 inside all of the eight registers - [CONTROL\\_CONF\\_DPLL\\_FREQLOCK\\_SEL](#) through [CONTROL\\_CONF\\_DPLL\\_IDLE\\_SEL](#).

**Table 18-44. Other PERIPH DPLL Observability Signals**

<b>SYSCTRL_GENERAL_CORE observability inputs tied to PERIPH DPLL observable signals</b>	<b>Other PERIPH DPLL observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal Description</b>	<b>Logical value meaning</b>
hwobs_perdpll_freqlock	PERIPHdpll_freqlock	PERIPH DPLL frequency lock check	-
hwobs_perdpll_tinitz	PERIPHdpll_tinitz	check PERIPH DPLL status, lock / relock time	-
hwobs_perdpll_phaselock	PERIPHDpllPhaseLock	PERIPH DPLL phase lock check	-
hwobs_perdpll_tenable	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_perdpll_tenablediv	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_perdpll_bypassack	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_perdpll_idle	PERIPHDpllPhaseLock	PERIPH DPLL phase lock check	-

**18.4.11.5.14 Other DSI1 DPLL observable signals**

Table 18-45 shows mapping of (other than CLKOUT) CORE dpll observability signals to the SYSCTRL\_GENERAL\_CORE dpll observing inputs. The selection of DSI1 DPLL as the source of the additional DPLL signals is done at SYSCTRL\_GENERAL\_CORE level through setting bit field MULT=0x3 inside all of the eight registers - [CONTROL\\_CONF\\_DPLL\\_FREQLOCK\\_SEL](#) through [CONTROL\\_CONF\\_DPLL\\_IDLE\\_SEL](#).

**Table 18-45. Other DSI1 DPLL Observability Signals**

<b>SYSCTRL_GENERAL_CORE observability inputs tied to DSI1 DPLL observable signals</b>	<b>Other DSI1 DPLL observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal Description</b>	<b>Logical value meaning</b>
hwobs_dsi1dpll_freqlock	dspdpll_freqlock	dspDPLL frequency lock check	-
hwobs_dsi1dpll_tinitz	dspdpll_tinitz	check dspDPLL status, lock / relock time	-
hwobs_dsi1dpll_phaselock	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_dsi1dpll_tenable	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_dsi1dpll_tenablediv	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_dsi1dpll_bypassack	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_dsi1dpll_idle	dspDpllPhaseLock	dspDPLL phase lock check	-

**18.4.11.5.15 Other DSI2 DPLL observable signals**

Table 18-46 shows mapping of (other than CLKOUT) CORE dpll observability signals to the SYSCTRL\_GENERAL\_CORE dpll observing inputs. The selection of DSI2 DPLL as the source of the additional DPLL signals is done at SYSCTRL\_GENERAL\_CORE level through setting bit field MULT=0x4 inside all of the eight registers - [CONTROL\\_CONF\\_DPLL\\_FREQLOCK\\_SEL](#) through [CONTROL\\_CONF\\_DPLL\\_IDLE\\_SEL](#).

**Table 18-46. Other DSI2 DPLL Observability Signals**

<b>SYSCTRL_GENERAL_CORE observability inputs tied to DSI2 DPLL observable signals</b>	<b>Other DSI2 DPLL observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal Description</b>	<b>Logical value meaning</b>
hwobs_dsi2dpll_freqlock	PERIPHDpll_freqlock	PERIPH DPLL frequency lock check	-
hwobs_dsi2dpll_tinitz	PERIPHDpll_tinitz	check PERIPH DPLL status, lock / relock time	-
hwobs_dsi2dpll_phaselock	PERIPHDpllPhaseLock	PERIPH DPLL phase lock check	-
hwobs_dsi2dpll_tenable	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_dsi2dpll_tenablediv	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_dsi2dpll_bypassack	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_dsi2dpll_idle	PERIPHDpllPhaseLock	PERIPH DPLL phase lock check	-

**18.4.11.5.16 Other USB DPLL observable signals**

Table 18-47 shows mapping of (other than CLKOUT) CORE dpll observability signals to the SYSCTRL\_GENERAL\_CORE dpll observing inputs. The selection of USB DPLL as the source of the additional DPLL signals is done at SYSCTRL\_GENERAL\_CORE level through setting bit field MULT=0x6 inside all of the eight registers - [CONTROL\\_CONF\\_DPLL\\_FREQLOCK\\_SEL](#) through [CONTROL\\_CONF\\_DPLL\\_IDLE\\_SEL](#).

**Table 18-47. Other USB DPLL Observability Signals**

<b>SYSCTRL_GENERAL_CORE observability inputs tied to USB DPLL observable signals</b>	<b>Other USB DPLL observable signals mapped to SYSCTRL_GENERAL_CORE observability inputs</b>	<b>Signal Description</b>	<b>Logical value meaning</b>
hwobs_usbdpll_freqlock	dspdll_freqlock	dspDPLL frequency lock check	-
hwobs_usbdpll_tinitz	dspdll_tinitz	check dspDPLL status, lock / relock time	-
hwobs_usbdpll_phaselock	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_usbdpll_tenable	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_usbdpll_tenablediv	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_usbdpll_bypassack	dspDpllPhaseLock	dspDPLL phase lock check	-
hwobs_usbdpll_idle	dspDpllPhaseLock	dspDPLL phase lock check	-

### 18.4.12 Functional Register Description

#### 18.4.12.1 DSP Boot Register

This information is not available in the public domain.

#### 18.4.12.2 PBIASLITE Control Register

Table 18-48 describes the [CONTROL\\_PBIASLITE](#) register, which controls some settings of PBIAS LITE cells for the MMC I/O interface.

**Table 18-48. PBIAS Control Register**

<b>Physical Address</b>	<b>Register Name</b>	<b>Description</b>	<b>Access</b>
0x4A10 0600	<a href="#">CONTROL_PBIASLITE</a>	Control settings for PBIAS cell	R/W

For more information about the PBIAS cells, see [Section 18.4.9, Extended-Drain I/O and PBIAS Cell](#).

#### 18.4.12.3 Temperature Sensor Control Register

Table 18-49 describes the [CONTROL\\_TEMP\\_SENSOR](#) register, which controls the temperature sensor.

**Table 18-49. Temperature Sensor Register**

<b>Physical Address</b>	<b>Register Name</b>	<b>Description</b>	<b>Access</b>
0x4A00 232C	<a href="#">CONTROL_TEMP_SENSOR</a>	Temperature sensor control register	R/W

#### 18.4.12.4 CSI Receiver Control Register

To enable the physical layer of the camera receiver, the programmer manipulates the bits within the [CONTROL\\_CAMERA\\_RX](#) register.

**Table 18-50. CSI Receiver control Register**

<b>Physical Address</b>	<b>Register Name</b>	<b>Description</b>	<b>Access</b>
0x4A10 0608	<a href="#">CONTROL_CAMERA_RX</a>	Module dedicated configurations	R/W

### 18.4.12.5 Protection Status Registers

Table 18-51 lists the status registers.

**Table 18-51. Protection Status Registers**

Physical Address	Register Name	Description	Access
0x4A00 22D0	<a href="#">CONTROL_SEC_ERR_STATUS_FUNC</a>	Protection firewall error status register	Public (R)
0x4A00 22D4	<a href="#">CONTROL_SEC_ERR_STATUS_DEBUG</a>	Protection firewall error status register debug	Public (RW)

These registers do not depend on the device type.

The [CONTROL\\_PROT\\_ERR\\_STATUS\\_FUNC](#) can be read in public mode, but cannot be written. The [CONTROL\\_PROT\\_ERR\\_STATUS\\_DEBUG](#) register is both read and write accessible in GP devices.

These bits are cleared when the L3 and L4 firewall embedded error log registers are cleared. All bits in these registers reflect device internal events related to the device protection.

On a specific event (signal rising edge), the corresponding bit is set. On a rising edge, the input signal must stay high for at least two interface clocks periods to be recognized. Software must clear each bit after reviewing the events.

When a protection violation occurs, the following bits are set:

- In application mode
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[01] = L3RAM protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[02] = GPMC protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[03] = EMIF protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[04] = IVAHD protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[05] = Dual Cortex™-M3 protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[06] = SL2 protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[12] = C2C protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[13] = SGX protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[14] = DSS protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[15] = ISS protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[16] = L4\_PER protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[17] = L4\_CFG protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[18] = DEBUGSS protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[19] = L4 AudioBE protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_FUNC](#)[20] = C2C INIT protection violation
- In debug mode
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[01] = L3RAM protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[02] = GPMC protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[03] = EMIF protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[04] = IVAHD protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[05] = Dual Cortex™-M3 protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[06] = SL2 protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[12] = C2C protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[13] = SGX protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[14] = DSS protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[15] = ISS protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[16] = L4\_PER protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[17] = L4\_CFG protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[18] = DEBUGSS protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[19] = L4 AudioBE protection violation
  - [CONTROL\\_SEC\\_ERR\\_STATUS\\_DEBUG](#)[20] = C2C INIT protection violation

For more information about L4 firewall protection violation report mechanisms, see [Section 13.3.3.4.5, Firewall Error Logging in the Control Module](#). For more information about L3 firewall protection violation report mechanisms, see [Section 13.2.3.8, L3 Interconnect Error Analysis](#).

### 18.4.12.6 Signal Integrity Parameter Control Registers with Pad Group Assignment

#### 18.4.12.6.1 Signal Integrity Parameter Controls Overview

Most of the I/O cells associated to the device pads are configurable to deliver their carried signals to the targeted sink with maximum signal integrity. Configuration of these I/Os is done by pad groups assignment and not individually per pad.

The different I/O-specific requirements determine usage of different type of I/O cells for the device I/Os. Software controls associated with the signal integrity parameters are implemented at the SYSCTRL\_PADCONF\_CORE and SYSCTRL\_PADCONF\_WKUP level.

Following types of I/O cells associated with pad groups can be identified:

- LPDDR2 high-speed I/O cells with impedance(I), slew rate(SR) and weak driver (WD) settings
- High-speed I/O cells with far-end load (LB) settings
- Low -speed I/O cells with combined slew rate (SC) and load vs transmission line (TL) length -(LB) settings
- I/O cells with combined mode (MB) controls and load compensation (LB) settings
- 50-Ω output buffer I/O cells with mode (MB) controls and load compensation (LB) settings
- I/O cells with control (DS0) of the output driver impedance
- C2C I/O cells with LB0 control adjustable according to interface requirement / voltage mode
- I/O cells with pullup strength (PUSTRENGTH) and speed controls
- I2Cx I/O cells with pullup resistance vs capacitance load settings
- JTAG I/O cells with bus-holder enable control

The pad group assigned signal integrity controls are provided through the registers described in [Table 18-52](#).

**Table 18-52. Signal Integrity Parameter Control Registers**

Physical Address	Register Name	Description	Access
0x4A10 05A8	<a href="#">CONTROL_SMART1IO_PADCONF_0</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05AC	<a href="#">CONTROL_SMART1IO_PADCONF_1</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05B0	<a href="#">CONTROL_SMART2IO_PADCONF_0</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05B4	<a href="#">CONTROL_SMART2IO_PADCONF_1</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05B8	<a href="#">CONTROL_SMART3IO_PADCONF_0</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05BC	<a href="#">CONTROL_SMART3IO_PADCONF_1</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05C0	<a href="#">CONTROL_SMART3IO_PADCONF_2</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05C4	<a href="#">CONTROL_USBB_HSIC</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05C8	<a href="#">CONTROL_SMART3IO_PADCONF_3</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05CC	<a href="#">CONTROL_SMART2IO_PADCONF_2</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05D0	<a href="#">CONTROL_SMART1IO_PADCONF_2</a>	Signal integrity pad group assignment control register	R/W
0x4A10 05D4	<a href="#">CONTROL_SMART1IO_PADCONF_3</a>	Signal integrity pad group assignment control register	R/W

**Table 18-52. Signal Integrity Parameter Control Registers (continued)**

Physical Address	Register Name	Description	Access
0x4A10 05D8	<a href="#">CONTROL_C2CIO_PADCONF_0</a>	Signal integrity pad group assignment control register	R/W
0x4A10 0604	<a href="#">CONTROL_I2C_0</a>	Signal integrity pad group assignment control register	R/W
0x4A10 0628	<a href="#">CONTROL_MMC1</a>	Signal integrity pad group assignment control register	R/W
0x4A10 0638	<a href="#">CONTROL_LPDDR2IO1_0</a>	Signal integrity pad group assignment control register	R/W
0x4A10 063C	<a href="#">CONTROL_LPDDR2IO1_1</a>	Signal integrity pad group assignment control register	R/W
0x4A10 0640	<a href="#">CONTROL_LPDDR2IO1_2</a>	Signal integrity pad group assignment control register	R/W
0x4A10 0644	<a href="#">CONTROL_LPDDR2IO1_3</a>	Signal integrity pad group assignment control register	R/W
0x4A10 0648	<a href="#">CONTROL_LPDDR2IO2_0</a>	Signal integrity pad group assignment control register	R/W
0x4A10 064C	<a href="#">CONTROL_LPDDR2IO2_1</a>	Signal integrity pad group assignment control register	R/W
0x4A10 0650	<a href="#">CONTROL_LPDDR2IO2_2</a>	Signal integrity pad group assignment control register	R/W
0x4A10 0654	<a href="#">CONTROL_LPDDR2IO2_3</a>	Signal integrity pad group assignment control register	R/W
0x4A31 E5A0	<a href="#">CONTROL_SMART1NOPMIO_PADCONF_0</a>	Signal integrity pad group assignment control register	R/W
0x4A31 E5A4	<a href="#">CONTROL_SMART1NOPMIO_PADCONF_1</a>	Signal integrity pad group assignment control register	R/W
0x4A31 E5B0	<a href="#">CONTROL_SMART3NOPMIO_PADCONF_0</a>	Signal integrity pad group assignment control register	R/W
0x4A31 E5B4	<a href="#">CONTROL_SMART3NOPMIO_PADCONF_1</a>	Signal integrity pad group assignment control register	R/W
0x4A31 E604	<a href="#">CONTROL_I2C_2</a>	Signal integrity pad group assignment control register	R/W
0x4A31 E608	<a href="#">CONTROL_JTAG</a>	Signal integrity pad group assignment control register	R/W

#### 18.4.12.6.2 High-speed I/O Buffers with Impedance, Slew Rate and Weak Driver Settings

The LPDDR2IO (EMIF I/F) buffers are based on 1.2V- High-Speed Un-terminated Logic and require excellent driver impedance control to achieve optimal timing -EMI trade-offs. A specific type of device I/O cell with programmable features are used to meet this requirement. They support 1,2-V LPDDR2 operation with target speed of 400MHz. These IO cells are also provided with some additional features to support High Speed Inter -Chip (HSIC) interface.

Three types of software configurable I/O cells are integrated to support high-speed unterminated logic :

- Single-ended I/O cells
- Two-pad differential I/O cells
- Vref generation cell

Two-pad differential I/O cells are provided with differential receiver and complementary transmitter. They are used to support LPDDR2 clock (lpddr21\_ck,lpddr21\_nck, lpddr22\_ck, lpddr22\_nck) pads and LPDDR2 strobe ( lpddr21\_dqsi, lpddr21\_ndqsi, lpddr22\_dqsi, lpddr22\_ndqsi ,where i=0 to 3) pads.

A Vref-generation cell is integrated in the device to supply a Vref based receiver.

These I/O cells are integrated with the following controls located in the SYSCTRL\_PADCONF\_CORE module:

- ❑ Driver output impedance (I) controls

- SR controls
- WD controls to support HSIC interface and avoid floating pad
- Process-only impedance compensation using 5-bit binary coded schema separately for pullup p<1:5> and pulldown n<1:5> impedance
- Internal and external reference voltage source selection for the Vref generation cell
- Vref cell decoupling capacitor selection

**NOTE:** I/Os are non-failsafe, which means that voltage must never be applied at pad before the internal IO supply voltage VDDS is powered-up.

Programmable bits for impedance and SR control are provided when the I/Os are set to operate in driver mode (outputs).

The bits I[2:0] are used to program desired impedance value of the driver in the range 30  $\Omega$  to 80  $\Omega$ . [Table 18-53](#) describes the driver impedance settings.

**Table 18-53. Driver Impedance Output Settings**

I2	I1	I0	Drive Setting Name	Output Impedance (Ron)	Drive Strength  Ioh , Iol	Example: Ron	Example:  Ioh , Iol  for Rref=50 $\Omega$	Notes
0	0	0	Drv5	1.6*Rref	0.625*Iout	80 $\Omega$	5 mA	
0	0	1	Drv6	1.33*Rref	0.75*Iout	67 $\Omega$	6 mA	
0	1	0	Drv7	1.14*Rref	0.875*Iout	57 $\Omega$	7 mA	
0	1	1	Drv8	Rref	Iout	50 $\Omega$	8 mA	
1	0	0	Drv9	0.88*Rref	1.125*Iout	44 $\Omega$	9 mA	
1	0	1	Drv10	0.8*Rref	1.250*Iout	40 $\Omega$	10 mA	
1	1	0	Drv11	0.73*Rref	1.375*Iout	36 $\Omega$	11 mA	
1	1	1	Drv12	0.67*Rref	1.5*Iout	34 $\Omega$	11.6 mA	

**NOTE:**

- Reference output impedance is considered to be the production trim setting for Drv8 mode, which corresponds to Rout = 50  $\Omega$  / Iout = 8 mA.
- [Table 18-53](#) is valid for both pull-up and pull-down outputs.
- The drive strength column shows the equivalent output current Iout setting. Output currents IOH, IOL are measured at VOH, VOL of VDDS- 0.4 V and 0.4 V respectively.

To achieve optimal noise/ speed trade off, the slew rate of the output signal can also be programmed using the slew rate (turn-on time) control bits SR[1:0]. The SR settings shown in [Table 18-54](#) don't affect the driver DC drive -strength. They only control the driver turn -on time.

**Table 18-54. Driver Slew Rate Settings**

SR1	SR0	Turn-on time level	Turn on time for Drv8 setting (Value is the average for E-fuse setting)
0	0	Fastest	125 ps
0	1	Faster	500 ps
1	0	Fast	325 ps
1	1	Slow	750 ps

Weak pull-up, pull-down or a keeper on pad is enabled through bits wd0 and wd1. Weak pull-up, pull-down option is used to define the pad state (high or low) when no signal is driving the pad. The weak keeper option is used to maintain the previous output value, when nothing is driving the pad. [Table 18-55](#) describes the WD controls.



**NOTE:** The WD[1:0] bit field controls play role of the PULLTYPESELECT and PULLUDENABLE bits within the standard pad configuration registers, additionally providing the keeper function to the I/Os with the state WD=0b11. For LPDDR2 I/Os the WD controls do not apply during OFF-mode.

In contrast to the LPDDR2 I/Os, the usbb1\_hsic and usbb2\_hsic pads are additionally provided with bits to select weak pullup/pulldown and keeper function during OFF mode. For a given USBBx\_HSIC pad (where x=1,2) the associated bit to enable this option is OFFMODE\_WD\_ENABLE and the actual controls become OFFMODE\_WD[1:0] bits.

**Table 18-55. Weak Driver PUPD /Latch mode**

WD1	WD0	Operation
0	0	Pull logic disabled
0	1	Weak pullup enabled
1	0	Weak pulldown enabled
1	1	Weak keeper enabled

The E-fuse impedance compensation settings are exported upon power-on reset to 10 bits in the [CONTROL\\_EFUSE\\_2](#) register of the Device Control Module, as follows:

-[CONTROL\\_EFUSE\\_2\[23\]](#) LPDDR2\_PTV\_N1 through [CONTROL\\_EFUSE\\_2\[19\]](#) LPDDR2\_PTV\_N5 (pull-down impedance trimming)

-[CONTROL\\_EFUSE\\_2\[18\]](#) LPDDR2\_PTV\_P1 through [CONTROL\\_EFUSE\\_2\[14\]](#) LPDDR2\_PTV\_P5 (pull-up impedance trimming)

**CAUTION**

Do not override the E-fuse impedance compensation settings in the [CONTROL\\_EFUSE\\_2](#) register

The Voltage reference cell which is used with LPDDR2IO Vref based receiver has the following associated control bit fields in the [CONTROL\\_LPDDR2IO1\\_3](#) and [CONTROL\\_LPDDR2IO2\\_3](#) registers:

- Internal reference voltage drive selection through bits : VREF\_TAP0 and VREF\_TAP1 bits
- Selection of coupling capacitor between the internal reference node and power supply ports -( VDDS and VSS ) through bits: VREF\_CCAP0 and VREF\_CCAP1 bits

[Table 18-56](#) describes the possible options for internal reference voltage drive selection.

**Table 18-56. Internal reference configuration**

VREF_TAP1	VREF_TAP0	Description
0	0	Pad/bias2 Connected to internal reference VDDS/2, for 2- $\mu$ A current load
0	1	Pad/bias2 Connected to internal reference VDDS/2, for 4- $\mu$ A current load
1	0	Pad/bias2 Connected to internal reference VDDS/2, for 6- $\mu$ A current load
1	1	Pad/bias2 Connected to internal reference VDDS/2, for 8- $\mu$ A current load

In internal reference mode the Vref cell can also be used to supply reference voltage to other external devices.

Bits to enable / disable internal voltage reference for CA- and DQ- lines in manual and auto modes are available to user in registers [CONTROL\\_LPDDR2IO1\\_3](#) and [CONTROL\\_LPDDR2IO2\\_3](#) respectively. When auto and manual modes are enabled, manual controls take higher priority over the voltage reference.

**CAUTION**

While using the Vref cell in internal reference mode, nothing else should be driving the pad externally. Although the pad can be connected to supply Vref signal to other chips.

The possible options for coupling capacitor configuration is listed in [Table 18-57](#) :

**Table 18-57. Vref cell decoupling cap selection table**

VREF_CCAPO	VREF_CCAP1	Capacitor
0	0	No capacitor connected
0	1	Capacitor between internal reference node and VDD5
1	0	Capacitor between internal reference node and VSS
1	1	Capacitor between internal reference node and VDD5 as well as internal reference node and VSS

**18.4.12.6.3 Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings**

[Table 18-58](#) provides the low-speed I/O cells SR control modes selectable through SC[1:0] bit field. These settings act like coarse settings for the signal performance behaviour of the low-speed I/Os. The SR settings determine three target maximal frequencies of I/Os operation at maximum load (see [Table 18-58](#).) The different slew control modes (0-2) requirements are described in [Table 18-59](#), [Table 18-61](#) and [Table 18-62](#).

[Table 18-60](#) lists the low-speed I/O cells - far-end capacitance load vs TL length settings selectable through bits LB[1] and LB[0], respectively. The four possible LB-settings described in [Table 18-60](#) are fully applicable and have maximum effect only in the case when slew control mode 0 is selected. These settings fine tune the signal performance. When SR control modes 1 and 2 are selected, then sensitivity to LB[1:0] setting is minimal, however LB[1:0] =11 achieves the best I/O performance.

**Table 18-58. Slew control mode (coarse) settings**

SC[1] bit	SC[0] bit	Slew Rate Control Mode	Maximum frequency achieved at maximum load
0	0	0	20MHz
0	1	1	60MHz
1	0	2	40MHz

**Table 18-59. Signal performance for slew control mode 0 - SC[1:0]=00**

Slew rate mode setting - SC[1:0]=00				
Parameter	Min	Max	Units	Comments
Transmission line Impedance	30	70	$\Omega$	-
Equivalent Load Range = ( Transmission line load + Far end load)	4	60	pF	Max load comes from dual transmission line
Transmission line(TL) length	2	20	cm	
Far end Load Range	2	10	pF	Assuming that characteristic impedance of TL is 50 $\Omega$

**Table 18-60. Load bit (fine) settings**

LB[1] (Far-end capacitance control)	LB[0] (TL- length control)	Effective TL lengt (cm)	Far-end capacitance (pF) (per TL)
0	0	2-20	1-10
0	1	20-40 (dual TL)	1-10
1	0	2-20	10-20
1	1	20-40 (dual TL)	10-20

**NOTE:** The LB modes in [Table 18-60](#) are fully applicable and have maximum effect only in the case of slew control mode 0 (SC[1:0]=00). For slew control modes 1 and 2, LB[1:0]=0b11 is recommended.

**Table 18-61. Signal performance for slew control mode 1 - SC[1:0]=01**

Slew rate mode setting - SC[1:0]=01				
Parameter	Min	Max	Units	Comments
Transmission line Impedance	30	70	$\Omega$	-
Equivalent Load Range = ( Transmission line load + Far-end load)	2	21	pF	This includes Far-end load and printed wiring board (PWB) trace
Transmission line(TL) length	1	6	cm	-
Far-end Load Range	1	15	pF	Assuming that characteristic impedance of TL is 50 $\Omega$

**NOTE:** When slew rate mode 1 is selected, it's recommended to set LB[1:0]=0b11 to achieve best signal performance.

**Table 18-62. Signal performance for slew control mode 2 - SC[1:0]=10**

Slew rate mode setting - SC[1:0]=10				
Parameter	Min	Max	Units	Comments
Transmission line impedance	30	70	$\Omega$	-
Equivalent Load Range = ( Transmission line load + Far end load)	7	33	pF	This includes Far-end load and printed wiring board (PWB) trace
Transmission line(TL) length	2	8	cm	
Far end Load Range	5	25	pF	assuming that characteristic impedance of TL is 50 $\Omega$

**NOTE:** When slew rate mode 2 is selected, it's recommended to set LB[1:0]=0b11 to achieve best signal performance.

#### 18.4.12.6.4 I/Os with Combined Mode and Load Settings

I/Os of this type has an output buffer with an internal PAD voltage SR control mechanism. Typical driver output impedance is 25  $\Omega$ . The corresponding I/O cell is controlled from two mode bits ( MB0 and MB1 ) and a dedicated bit ( LB0 ) for load compensation. [Table 18-63](#) describes the mode bit settings for each mode. The load bit programs the far end load for the I/O cell associated pads. [Table 18-63](#) also lists the load bit settings, and [Table 18-64](#) lists the I/O cell features in different modes.

**Table 18-63. Software controls for I/Os with combined Mode and Load settings**

Control Bits for I/Os with combined Mode vs Load settings			
MODE	MB1	MB0	LB0
MODE1	0	1	0
MODE2	1	0	0 (farend load < 15pf) 1 (farend load >= 15pf)
MODE3	1	1	0 (farend load < 4pf) 1 (farend load >= 4pf)
MODE4	1	1	(farend load < 4pf) 1 (farend load >= 4pf)

**Table 18-64. I/O cell Features in Different Modes**

Interface	Mode1	Mode2	Mode3	Mode4
	SDMMC/SPI	DMIC / GPIO / Asyn Ctrl signal	HSI	ULPI
Interface frequency (MHz)	48	4	200	60
Clload_max (pF)	10	35	5	5
Max TX.line length (cm)	10	20	5	6
Clload_min (pF)	2	5	3	2
Min TX.line length(cm)	2	2	2	2

**NOTE:** MODE4 is achieved by using the same mode setting as that of MODE3. The MODE4 design is limited by stringent delay requirement, even though the frequency is lower than MODE3.

A process-only compensation using 4-bit binary coded schema is used separately for pullup and pulldown output parameter variations. Only MODE2 doesn't use such compensation. The I/Os with combined mode and load controls ( 25- $\Omega$  output buffer ) share the same compensation settings with the I/Os with driver output impedance (DS) settings as well as with the C2CI/O buffers. They are exported from E-fuse upon power-on reset to the following bits in the [CONTROL\\_EFUSE\\_2](#) register:

- [CONTROL\\_EFUSE\\_2](#)[31] EFUSE\_SMART2TEST\_P0 through [CONTROL\\_EFUSE\\_2](#)[28] EFUSE\_SMART2TEST\_P3 (pullup output parameters trimming)
- [CONTROL\\_EFUSE\\_2](#)[27] EFUSE\_SMART2TEST\_N0 through [CONTROL\\_EFUSE\\_2](#)[24] EFUSE\_SMART2TEST\_N3 (pulldown output parameters trimming)

#### CAUTION

Do not override the SMART2TEST\_P / SMART2TEST\_N - E-fuse compensation settings in the [CONTROL\\_EFUSE\\_2](#) register

**18.4.12.6.5 50-Ω Output Buffer I/Os with Combined Mode and Load Settings**

I/Os of this type has an output buffer with an internal PAD voltage slew rate control mechanism. Driver output impedance is typically 50 Ω. The corresponding I/O cell is controlled from two mode bits MB0 and MB1 and a dedicated bit LB0 for load compensation. They are located within a number of the device signal integrity control registers - CONTROL\_SMART3IO\_PADCONF\_x (x=0,1,2), CONTROL\_SMART3IO\_PADCONF\_3, CONTROL\_SMART3NOPMIO\_PADCONF\_0, CONTROL\_SMART3NOPMIO\_PADCONF\_1 registers. MODE1 for this type of I/O cells meets different specific requirements than MODE1 for buffers described in Section 18.4.12.6.4. The recommended mode and load bit settings are listed in table Table 18-65

**Table 18-65. Software controls for 50-Ω output buffer I/O cells with combined Mode and Load settings**

Control Bits for I/Os with 50-Ω output buffer and combined Mode vs Load settings				
MODE	MB1	MB0	LB0	
MODE1	1	1	1	
MODE2	0	1	0	
MODE3	1	0	1	
MODE4	1	0	0	

The different mode features are described in Table 18-66.

**Table 18-66. Different Mode features**

Interface	Mode1	Mode2	Mode3	Mode4
Interface frequency (MHz)	60	48	24	25
Cload_max (pF)	5	5	20	11
Cload_min (pF)	2	2	16	2

A process-only compensation using 4-bit binary coded schema is used separately for pull-up and pull-down output parameter variations. Only MODE1 and MODE2 make use of Efuse process-only compensation to reduce spread in timing parameters. The I/Os with 50-Ω output buffer and combined mode and load controls share the same compensation settings with the I/Os with driver output impedance (DS) settings as well as with the C2CI/O buffers. They are exported from E-fuse upon power-on reset to the following bits in the CONTROL\_EFUSE\_2 register:

- CONTROL\_EFUSE\_2[31] EFUSE\_SMART2TEST\_P0 through CONTROL\_EFUSE\_2[28] EFUSE\_SMART2TEST\_P3 (pull-up output parameters trimming)
- CONTROL\_EFUSE\_2[27] EFUSE\_SMART2TEST\_N0 through CONTROL\_EFUSE\_2[24] EFUSE\_SMART2TEST\_N3 (pull-down output parameters trimming)

**CAUTION**

Do not override the SMART2TEST\_P / SMART2TEST\_N - E-fuse compensation settings in the CONTROL\_EFUSE\_2 register

**18.4.12.6.6 I/O cells with Configurable Output Driver Impedance**

This type of I/O cells has an output buffer with an internal PAD voltage slew rate control mechanism. It operates in 25 Ω or 50 Ω impedance mode depending on the drive strength bit - DS0 setting. The drive strength control logic is explained in Table 18-67. The performance characteristics at different power modes 1.2/1.8 V are shown in Table 18-68. These controls are located within the CONTROL\_SMART2IO\_PADCONF\_2 register.

**Table 18-67. Driver Impedance Control**

MODE	DS0	Impedance
MODE1	1	25 Ω
MODE2	0	50 Ω

**Table 18-68. I/O cell performance for different impedances**

Control Bits for I/Os with configurable driver output impedance			
DS0 setting	TL impedance ( $\Omega$ )	Load (pF)	Max. Freq. (MHz)
0	40-60	2-5	75
1	30-45	2-5	100

A process-only compensation using 4-bit binary coded schema is used separately for pull-up and pull-down output parameter variations. Only MODE1 and MODE2 make use of Efuse process-only compensation to reduce spread in timing parameters. The I/Os with driver impedance (DS) control share the same compensation settings with the I/Os with combined mode and load controls (both with 25- and 50- $\Omega$  output ) as well as with the C2C/I/O buffers. They are exported from E-fuse upon power-on reset to the following bits in the [CONTROL\\_EFUSE\\_2](#) register:

-[CONTROL\\_EFUSE\\_2\[31\]](#) EFUSE\_SMART2TEST\_P0 through [CONTROL\\_EFUSE\\_2\[28\]](#) EFUSE\_SMART2TEST\_P3 (pull-up output parameters trimming)

-[CONTROL\\_EFUSE\\_2\[27\]](#) EFUSE\_SMART2TEST\_N0 through [CONTROL\\_EFUSE\\_2\[24\]](#) EFUSE\_SMART2TEST\_N3 (pull-down output parameters trimming)

#### 18.4.12.6.7 C2C/I/O cells with load setting adjustable according to interface requirement / voltage mode

This IO cell can be configured to be used in different interfaces, like C2C, GPMC and SDMMC2 using a single-bit (LB0) selection. Like LPDDR2IO this IO cell is provided with selectable Lvcmos and Vref based receiver option. A vref pad is provided to supply reference voltage to the receiver in the 1.2V mode. To save power at lower frequency operation, lvcmos receiver can be selected, and two bits , for each of the two C2C pad groups are used to manually enable/disable the LVC MOS receiver - CMOSEN\_C2C\_FRM\_CTRL in the [CONTROL\\_C2CIO\\_PADCONF\\_0](#) register. Tables [Table 18-69](#) through [Table 18-71](#) explain the logic of the mode selection bit for use in different interfaces .

**Table 18-69. I/O cell performance for C2C I/F at 1.2/1.8 V**

Control Bits of the C2C I/O cell in C2C application						
Interface /Voltage Mode	LB0 bit setting	Interface summary	Fmax	TL impedance	TX Far-end load end	TL length
C2C 1.2 V	1	Interface with LPDDR1 modem. Modem multiplex its own LPDDR1 I/F with its own C2C I/F	200 MHz (400Mbps)	40-55 $\Omega$	2-4 pF	2 cm
C2C 1.8 V	1	Interface with LPDDR2 modem. Modem multiplex its own LPDDR2 I/F with its own C2C I/F	200 MHz (400 Mbps)	40-55 $\Omega$	2-4 pF	2 cm

**Table 18-70. I/O cell performance for GPMC I/F at 1.2/1.8 V**

Control Bits of the C2C I/O cell in GPMC application						
Interface /Voltage Mode	LB0 bit setting	Interface summary	Fmax	TL impedance	TX Far-end load end	TL length
GPMC 1.2 V	0	100MHz I/F [Read] 83MHz Interface [Write]	100 MHz	30-50 $\Omega$	5-10 pF	4 cm
GPMC 1.8 V	1	100MHz I/F [Read] 83MHz Interface [Write]	100 MHz	30-50 $\Omega$	5-10 pF	4 cm

**Table 18-71. I/O cell performance for SDMMC2 I/F at 1.2/1.8 V**

Control Bits of the C2C I/O cell in GPMC application						
Interface /Voltage Mode	LB0 bit setting	Interface summary	Fmax	TL impedance	TX Far-end load end	TL length
SDMMC2 1.2 V	0	48 MHz I/F	48 MHz SDR 96 MHz DDR	30-55 $\Omega$	2-5 pF	2-5 cm
SDMMC2 1.8 V	0	48 MHz I/F	48 MHz SDR 96 MHz DDR	30-55 $\Omega$	2-5 pF	2-5 cm

The Voltage reference cell which is used with C2CIO Vref based receiver has the following associated control bit fields in the [CONTROL\\_C2CIO\\_PADCONF\\_0](#) register:

- Selection of coupling capacitor between the internal reference node and power supply ports - VDDS and VSS through bit field: C2C\_VREF\_CCAP.

In the internal reference mode the Vref cell can also be used to supply reference voltage to other external devices. At Vdds=1.8V mode, the Internal reference generator will be shut off and internal reference node will be connected to zero.

Bits to enable / disable the internal voltage reference in manual and auto modes are available to user in registers [CONTROL\\_C2CIO\\_PADCONF\\_0](#) respectively. When auto and manual modes are enabled, manual controls take higher priority over the voltage reference.

#### CAUTION

While using the Vref cell in internal reference mode, nothing else should be driving the pad externally. Although the pad can be connected to supply Vref signal to other chips.

The possible options for coupling capacitor configuration is listed in [Table 18-72](#) :

**Table 18-72. Vref cell decoupling cap selection table**

VREF_CCAP0	VREF_CCAP1	Capacitor
0	0	No capacitor connected
0	1	Capacitor between internal reference node and VDDS
1	0	Capacitor between internal reference node and VSS
1	1	Capacitor between internal reference node and VDDS as well as internal reference node and VSS

A process-only compensation using 4-bit binary coded schema is used separately for pull-up and pull-down output parameter variations. The C2CI/O buffers share the same compensation settings with the I/Os with combined mode and load controls (both with 25- and 50- $\Omega$  output ) and I/Os with driver output impedance (DS) settings. They are exported from E-fuse upon power-on reset to the following bits in the [CONTROL\\_EFUSE\\_2](#) register:

- [CONTROL\\_EFUSE\\_2](#)[31] EFUSE\_SMART2TEST\_P0 through [CONTROL\\_EFUSE\\_2](#)[28] EFUSE\_SMART2TEST\_P3 (pull-up output parameters trimming)
- [CONTROL\\_EFUSE\\_2](#)[27] EFUSE\_SMART2TEST\_N0 through [CONTROL\\_EFUSE\\_2](#)[24] EFUSE\_SMART2TEST\_N3 (pull-down output parameters trimming)

#### 18.4.12.6.8 SDMMC1 Pull-up Strength Control

The SDMMC1\_PUSTRENGTH\_GRPi (where i=1 to 3) control bits are added to program the pullup strength for the MMC/SD/SDIO1 I/Os group. [Table 18-73](#) lists the supported pullup resistance ranges.

**Table 18-73. SDMMC1 Pullup Strength Controls**

Pustrength controls	
SDMMC1_PUSTRENGTH_GRPi = 0b0	SDMMC1_PUSTRENGTH_GRPi=0b1
Pullup with 50-100 kΩ strength selected	Pullup with 10-50 kΩ strength selected

#### 18.4.12.6.9 I2Cx I/Os Group Pullupresx Controls and Load Range Settings

Table 18-74 provides examples of pull-up resistance vs load range I2Cx I/O signal integrity control capabilities. Three possible operation modes of the I2C associated I/O cells are shown: non-I2C, Full speed (FS) and High speed (HS) modes (see Section 23.1, *Multimaster Highspeed I2C controller*). Non-I2C mode means that the MuxMode field in I2Cx dedicated PADCONFS is set to select signals different than I2C signals, see Section 18.4.8.3.

**Table 18-74. Example I2Cx Pullupresx vs. I2C LB Parameter Settings in Different Modes**

Non I2C mode	FS I2C bus mode	HS I2C p2p mode
nmode=1 hsmode=0 pullupresx=1 lb[1:0]=00	nmode=0 hsmode=0 pullupresx=1 lb[1:0]=00 400kHz, external PU	nmode=0 hsmode=1 pullupresx=0 lb[1:0]=00 3.4 MHz, cap. load 5-12 pF

The I2Cx buffer contains internal pull-up resistors for both Fast and High-Speed modes meeting I2C rise time specifications for loads up to 80 pF in HS mode and 150 pF in FAST mode. LB[1:0] signals can be used to select appropriate pullup resistor for a given load range. Table 18-75 lists the configuration options.

Active high GLFENB[0] bit is implemented for I2C receivers to enable glitch-free operation.

**Table 18-75. Internal Pullup Resistor in Fast/HS Mode**

LB1	LB0	Load Range HS Mode (pF)	Nominal Resistance in HS Mode (Ω)	Load Range Fast Mode (pF)	Nominal Resistance in Fast Mode (Ω)
0	0	5-12	1.66K	5-15	4.5K
0	1	12-25	920	15-50	2.1K
1	0	25-50	500	50-150	860
1	1	50-80	300	-	-

If a suitable external pullup resistor is used, I<sup>2</sup>C I/Os can meet I<sup>2</sup>C rise/fall time specifications in 1.2 / 1.8V mode under all modes up to 400 pF. The type of the pullup resistor (external or internal) for a certain I2Cx I/O group (x=1 to 4) is selected through the I2Cx\_SCL\_PULLUPRESX and I2Cx\_SDA\_PULLUPRESX bits. Internal pullups are activated when I2Cx\_SCL\_PULLUPRESX / I2Cx\_SDA\_PULLUPRESX =0b0. I2C functionality is not ensured if the weak internal pullup or pulldown (the 100 μA nominal PU-PD) is activated on any of the I/Os connected to the bus.

#### 18.4.12.6.10 Device Interfaces Signal Group Controls Mapping

Table 18-76 through Table 18-103 describes the mapping of the different signal integrity control bit fields in the different signal integrity control register described in the Table 18-52.

**NOTE:** The same I/O cell and I/O cell controls assigned to pad signal available when Muxmode = 0 are shared between signals available on the same pad when Muxmode is selected different than 0. For more information, see Section 18.4.8.3, *Pad Multiplexing Register Fields*.



**Table 18-76. EMIF (LPDDR2IO) Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

EMIF (LPDDR2IO) Interface I/O Controls		
Bit Fields for Pad Group Controls	Pads (Muxmode = 0) in Group	Type of I/O Group-Associated Control Bit Fields
<a href="#">CONTROL_LPDDR2IO1_0</a> [31:30] LPDDR2IO1_GR4_SR <a href="#">CONTROL_LPDDR2IO1_0</a> [29:27] LPDDR2IO1_GR4_I <a href="#">CONTROL_LPDDR2IO1_0</a> [26:25] LPDDR2IO1_GR4_WD	lpddr21_dq24–lpddr21_dq31; lpddr21_dm3	High-speed IO Buffers with Impedance (I), Slew Rate(SR) and Weak Driver settings (WD) (See <a href="#">Section 18.4.12.6.2</a> for more information)
<a href="#">CONTROL_LPDDR2IO1_0</a> [23:22] LPDDR2IO1_GR3_SR <a href="#">CONTROL_LPDDR2IO1_0</a> [21:19] LPDDR2IO1_GR3_I <a href="#">CONTROL_LPDDR2IO1_0</a> [18:17] LPDDR2IO1_GR3_WD	lpddr21_dq16–lpddr21_dq23; lpddr21_dm2	
<a href="#">CONTROL_LPDDR2IO1_0</a> [15:14] LPDDR2IO1_GR2_SR <a href="#">CONTROL_LPDDR2IO1_0</a> [13:11] LPDDR2IO1_GR2_I <a href="#">CONTROL_LPDDR2IO1_0</a> [10:9] LPDDR2IO1_GR2_WD	lpddr21_dq8–lpddr21_dq15; lpddr21_dm1	
<a href="#">CONTROL_LPDDR2IO1_0</a> [7:6] LPDDR2IO1_GR1_SR <a href="#">CONTROL_LPDDR2IO1_0</a> [5:3] LPDDR2IO1_GR1_I <a href="#">CONTROL_LPDDR2IO1_0</a> [2:1] LPDDR2IO1_GR1_WD	lpddr21_dq0–lpddr21_dq7; lpddr21_dm0	
<a href="#">CONTROL_LPDDR2IO1_1</a> [31:30] LPDDR2IO1_GR8_SR <a href="#">CONTROL_LPDDR2IO1_1</a> [29:27] LPDDR2IO1_GR8_I <a href="#">CONTROL_LPDDR2IO1_1</a> [26:25] LPDDR2IO1_GR8_WD	lpddr21_dqs2; lpddr21_ndqs2	
<a href="#">CONTROL_LPDDR2IO1_1</a> [23:22] LPDDR2IO1_GR7_SR <a href="#">CONTROL_LPDDR2IO1_1</a> [21:19] LPDDR2IO1_GR7_I <a href="#">CONTROL_LPDDR2IO1_1</a> [18:17] LPDDR2IO1_GR7_WD	lpddr21_dqs1; lpddr21_ndqs1	
<a href="#">CONTROL_LPDDR2IO1_1</a> [15:14] LPDDR2IO1_GR6_SR <a href="#">CONTROL_LPDDR2IO1_1</a> [13:11] LPDDR2IO1_GR6_I <a href="#">CONTROL_LPDDR2IO1_1</a> [10:9] LPDDR2IO1_GR6_WD	lpddr21_dqs0; lpddr21_ndqs0	
<a href="#">CONTROL_LPDDR2IO1_1</a> [7:6] LPDDR2IO1_GR5_SR <a href="#">CONTROL_LPDDR2IO1_1</a> [5:3] LPDDR2IO1_GR5_I <a href="#">CONTROL_LPDDR2IO1_1</a> [2:1] LPDDR2IO1_GR5_WD	lpddr21_ca0–lpddr21_ca9	
<a href="#">CONTROL_LPDDR2IO1_2</a> [31:30] LPDDR2IO1_GR11_SR <a href="#">CONTROL_LPDDR2IO1_2</a> [29:27] LPDDR2IO1_GR11_I <a href="#">CONTROL_LPDDR2IO1_2</a> [26:25] LPDDR2IO1_GR11_WD	lpddr21_ncs0; lpddr21_ncs1; lpddr21_cke0; lpddr21_cke1	
<a href="#">CONTROL_LPDDR2IO1_2</a> [23:22] LPDDR2IO1_GR10_SR <a href="#">CONTROL_LPDDR2IO1_2</a> [21:19] LPDDR2IO1_GR10_I <a href="#">CONTROL_LPDDR2IO1_2</a> [18:17] LPDDR2IO1_GR10_WD	lpddr21_ck; lpddr21_nck	

**Table 18-76. EMIF (LPDDR2IO) Signal Group Parameter Controls to Different Interface I/O Pads Mapping (continued)**

EMIF (LPDDR2IO) Interface I/O Controls		
Bit Fields for Pad Group Controls	Pads (Muxmode = 0) in Group	Type of I/O Group-Associated Control Bit Fields
<a href="#">CONTROL_LPDDR2IO1_2</a> [15:14] LPDDR2IO1_GR9_SR <a href="#">CONTROL_LPDDR2IO1_2</a> [13:11] LPDDR2IO1_GR9_I <a href="#">CONTROL_LPDDR2IO1_2</a> [10:9] LPDDR2IO1_GR9_WD	lpddr21_dqs3; lpddr21_ndqs3	High-speed IO Buffers with Impedance (I), Slew Rate(SR) and Weak Driver settings (WD) (See <a href="#">Section 18.4.12.6.2</a> for more information)
<a href="#">CONTROL_LPDDR2IO2_0</a> [31:30] LPDDR2IO2_GR4_SR <a href="#">CONTROL_LPDDR2IO2_0</a> [29:27] LPDDR2IO2_GR4_I <a href="#">CONTROL_LPDDR2IO2_0</a> [26:25] LPDDR2IO2_GR4_WD	lpddr22_dq24–lpddr22_dq31; lpddr22_dm3	
<a href="#">CONTROL_LPDDR2IO2_0</a> [23:22] LPDDR2IO2_GR3_SR <a href="#">CONTROL_LPDDR2IO2_0</a> [21:19] LPDDR2IO2_GR3_I <a href="#">CONTROL_LPDDR2IO2_0</a> [18:17] LPDDR2IO2_GR3_WD	lpddr22_dq16–lpddr22_dq23; lpddr22_dm2	
<a href="#">CONTROL_LPDDR2IO2_0</a> [15:14] LPDDR2IO2_GR2_SR <a href="#">CONTROL_LPDDR2IO2_0</a> [13:11] LPDDR2IO2_GR2_I <a href="#">CONTROL_LPDDR2IO2_0</a> [10:9] LPDDR2IO2_GR2_WD	lpddr22_dq8–lpddr22_dq15; lpddr22_dm1	
<a href="#">CONTROL_LPDDR2IO2_0</a> [7:6] LPDDR2IO2_GR1_SR <a href="#">CONTROL_LPDDR2IO2_0</a> [5:3] LPDDR2IO2_GR1_I <a href="#">CONTROL_LPDDR2IO2_0</a> [2:1] LPDDR2IO2_GR1_WD	lpddr22_dq0–lpddr22_dq7; lpddr22_dm0	
<a href="#">CONTROL_LPDDR2IO2_1</a> [31:30] LPDDR2IO2_GR8_SR <a href="#">CONTROL_LPDDR2IO2_1</a> [29:27] LPDDR2IO2_GR8_I <a href="#">CONTROL_LPDDR2IO2_1</a> [26:25] LPDDR2IO2_GR8_WD	lpddr22_dqs2; lpddr22_ndqs2	
<a href="#">CONTROL_LPDDR2IO2_1</a> [23:22] LPDDR2IO2_GR7_SR <a href="#">CONTROL_LPDDR2IO2_1</a> [21:19] LPDDR2IO2_GR7_I <a href="#">CONTROL_LPDDR2IO2_1</a> [18:17] LPDDR2IO2_GR7_WD	lpddr22_dqs1; lpddr22_ndqs1	
<a href="#">CONTROL_LPDDR2IO2_1</a> [15:14] LPDDR2IO2_GR6_SR <a href="#">CONTROL_LPDDR2IO2_1</a> [13:11] LPDDR2IO2_GR6_I <a href="#">CONTROL_LPDDR2IO2_1</a> [10:9] LPDDR2IO2_GR6_WD	lpddr22_dqs0; lpddr22_ndqs0	
<a href="#">CONTROL_LPDDR2IO2_1</a> [7:6] LPDDR2IO2_GR5_SR <a href="#">CONTROL_LPDDR2IO2_1</a> [5:3] LPDDR2IO2_GR5_I <a href="#">CONTROL_LPDDR2IO2_1</a> [2:1] LPDDR2IO2_GR5_WD	lpddr22_ca0–lpddr22_ca9	
<a href="#">CONTROL_LPDDR2IO2_2</a> [31:30] LPDDR2IO2_GR11_SR <a href="#">CONTROL_LPDDR2IO2_2</a> [29:27] LPDDR2IO2_GR11_I <a href="#">CONTROL_LPDDR2IO2_2</a> [26:25] LPDDR2IO2_GR11_WD	lpddr22_ncs0; lpddr22_ncs1; lpddr22_cke0; lpddr22_cke1	High-speed IO Buffers with Impedance (I), Slew Rate(SR) and Weak Driver settings (WD) (See <a href="#">Section 18.4.12.6.2</a> for more information)

**Table 18-76. EMIF (LPDDR2IO) Signal Group Parameter Controls to Different Interface I/O Pads Mapping (continued)**

EMIF (LPDDR2IO) Interface I/O Controls		
Bit Fields for Pad Group Controls	Pads (Muxmode = 0) in Group	Type of I/O Group-Associated Control Bit Fields
<a href="#">CONTROL_LPDDR2IO2_2</a> [23:22] LPDDR2IO2_GR10_SR <a href="#">CONTROL_LPDDR2IO2_2</a> [21:19] LPDDR2IO2_GR10_I <a href="#">CONTROL_LPDDR2IO2_2</a> [18:17] LPDDR2IO2_GR10_WD	lpddr22_ck; lpddr22_nck	
<a href="#">CONTROL_LPDDR2IO2_2</a> [15:14] LPDDR2IO2_GR9_SR <a href="#">CONTROL_LPDDR2IO2_2</a> [13:11] LPDDR2IO2_GR9_I <a href="#">CONTROL_LPDDR2IO2_2</a> [10:9] LPDDR2IO2_GR9_WD	lpddr22_dqs3; lpddr22_ndqs3	

**Table 18-77. GPMC I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

GPMC interface I/O Controls		
Bit Fields for Pad Group Controls	Pads (Muxmode = 0) in Group	Type of I/O-Group-Associated Control Bit Fields
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [13] SDMMC2_DR0_LB0	gpmc_ad0–gpmc_ad7; gpmc_noe; gpmc_nwe	C2CIO cell I/Os with single-LB0 control (See <a href="#">Section 18.4.12.6.7</a> for more information )
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [28] GPIO_DR0_LB0	gpmc_ad8–gpmc_ad11	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [15] KPD_DR4_LB0	gpmc_ad12–gpmc_ad15	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [27] GPIO_DR1_LB0	gpmc_a16–gpmc_a19	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [14] KPD_DR5_LB0	gpmc_a20–gpmc_a23	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [26] GPIO_DR2_LB0	gpmc_a24 ; gpmc_nbe0_cle	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [17]GPMC_DR8_LB0	gpmc_a25	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [25] GPMC_DR0_LB0	gpmc_ncs0	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [31] C2C_DR0_LB0	gpmc_ncs1	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [16] GPMC_DR9_LB0	gpmc_ncs2	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [23] GPMC_DR10_LB0	gpmc_ncs3	
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [13:12] GPMC_DR1_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [8] GPMC_DR1_LB	gpmc_nwp	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [24] GPMC_DR11_LB0	gpmc_nadv_ale	C2CIO cell I/Os with single-LB0 control (See <a href="#">Section 18.4.12.6.7</a> for more information )
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [20] GPMC_DR5_LB0	gpmc_clk	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [21] GPMC_DR4_LB0	gpmc_nbe1	
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [17:16] GPMC_DR3_SC; <a href="#">CONTROL_SMART1IO_PADCONF_1</a> [17:16] GPMC_DR3_LB	gpmc_wait0	Low speed I/Os LB[1: 0]+SC[1:0] controls (See <a href="#">Section 18.4.12.6.3</a> for more information )

**Table 18-77. GPMC I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping  
(continued)**

GPMC interface I/O Controls		
Bit Fields for Pad Group Controls	Pads (Muxmode = 0) in Group	Type of I/O-Group-Associated Control Bit Fields
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [22] GPMC_DR2_LB0	gpmc_wait1	C2CIO cell I/Os with single-LB0 control (See <a href="#">Section 18.4.12.6.7</a> for more information )
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [19] GPMC_DR6_LB0	gpmc_wait2	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [18] GPMC_DR7_LB0	gpmc_ncs4, gpmc_ncs5	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [30] C2C_DR1_LB0	gpmc_ncs6	
<a href="#">CONTROL_C2CIO_PADCONF_0</a> [29] C2C_DR2_LB0	gpmc_ncs7	

**Table 18-78. CAMERA I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

CAMERA interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [29:28] CAM_DR0_SC <a href="#">CONTROL_SMART1IO_PADCONF_1</a> [29:28] CAM_DR0_LB	cam_shutter; cam_strobe; cam_globalreset	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )

**Table 18-79. USBB1 I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

USB B1 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [11] USB B1_DR0_DS	usbb1_ulpitll_clk	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [21:20] HSI_DR0_MB <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [10] HSI_DR0_LB	usbb1_ulpitll_stp	I/Os with MB[1:0] + LB[0] settings (See <a href="#">Section 18.4.12.6.4</a> for more information ) with single-LB0 control
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [1:0] USB B1_DR2_MB <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [4] USB B1_DR2_LB	usbb1_ulpitll_dir	
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [21:20] HSI_DR1_MB <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [26] HSI_DR1_LB	usbb1_ulpitll_nxt; usbb1_ulpitll_dat0	I/Os with MB[1:0] + LB[0] settings (See <a href="#">Section 18.4.12.6.4</a> for more information ) with single-LB0 control
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [19:18] HSI_DR2_MB <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [25] HSI_DR2_LB	usbb1_ulpitll_dat1; usbb1_ulpitll_dat2	
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [17:16] HSI_DR3_MB <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [24] HSI_DR3_LB	usbb1_ulpitll_dat3	
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [29:28] GPIO_DR3_MB <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [30] GPIO_DR3_LB	usbb1_ulpitll_dat4; usbb1_ulpitll_dat5	

**Table 18-79. USBB1 I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping (continued)**

USB B1 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [27:26] GPIO_DR4_MB <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [29] GPIO_DR4_LB	usbb1_ulpitll_dat6; usbb1_ulpitll_dat7	
<a href="#">CONTROL_USBB_HSIC</a> [26:25] USB B1_DR1_SR <a href="#">CONTROL_USBB_HSIC</a> [24:22] USB B1_DR1_I <a href="#">CONTROL_USBB_HSIC</a> [21:20] USB B1_HSIC_DATA_WD <a href="#">CONTROL_USBB_HSIC</a> [13] USB B1_HSIC_DATA_OFFMODE_WD_ENA BLE <a href="#">CONTROL_USBB_HSIC</a> [12:11] USB B1_HSIC_DATA_OFFMODE_WD <a href="#">CONTROL_USBB_HSIC</a> [19:18] USB B1_HSIC_STROBE_WD <a href="#">CONTROL_USBB_HSIC</a> [10] USB B1_HSIC_STROBE_OFFMODE_WD_E NABLE <a href="#">CONTROL_USBB_HSIC</a> [9:8] USB B1_HSIC_STROBE_OFFMODE_WD	usbb1_hsic_data; usbb1_hsic_strobe	High-speed IO Buffers with Impedance (I), Slew Rate(SR) and Weak Driver (WD) settings (See <a href="#">Section 18.4.12.6.2</a> for more information)

**Table 18-80. USB B2 I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

USB B2 interface I/O controls		
Bit Fields for pad group controls	Pads(Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [12] USB B2_DR0_DS	usbb2_ulpitll_clk;	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [23] SDMMC4_DR0_DS	usbb2_ulpitll_stp; usbb2_ulpitll_dir	
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [27] HSI2_DR0_DS	usbb2_ulpitll_nxt; usbb2_ulpitll_dat0	
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [22] SDMMC4_DR1_DS	usbb2_ulpitll_dat1	
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [26] HSI2_DR1_DS	usbb2_ulpitll_dat2	
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [25] HSI2_DR2_DS	usbb2_ulpitll_dat3	
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [24] SDMMC3_DR0_DS	usbb2_ulpitll_dat4	
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [20] SPI3_DR1_DS	usbb2_ulpitll_dat5	
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [21] SPI3_DR0_DS	usbb2_ulpitll_dat6; usbb2_ulpitll_dat7	

**Table 18-80. USB2 I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping (continued)**

USB2 interface I/O controls		
Bit Fields for pad group controls	Pads(Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_USBB_HSIC</a> [31:30] USB2_DR1_SR <a href="#">CONTROL_USBB_HSIC</a> [29:27] USB2_DR1_I <a href="#">CONTROL_USBB_HSIC</a> [17:16] USB2_HSIC_DATA_WD <a href="#">CONTROL_USBB_HSIC</a> [7] USB2_HSIC_DATA_OFFMODE_WD_ENA BLE <a href="#">CONTROL_USBB_HSIC</a> [6:5] USB2_HSIC_DATA_OFFMODE_WD <a href="#">CONTROL_USBB_HSIC</a> [15:14] USB2_HSIC_STROBE_WD <a href="#">CONTROL_USBB_HSIC</a> [4] USB2_HSIC_STROBE_OFFMODE_WD_E NABLE <a href="#">CONTROL_USBB_HSIC</a> [3:2] USB2_HSIC_STROBE_OFFMODE_WD	usbb2_hsic_data; usbb2_hsic_strobe	High-speed IO Buffers with Impedance (I), Slew Rate(SR) and Weak Driver (WD) settings (See <a href="#">Section 18.4.12.6.2</a> for more information)

**Table 18-81. USBC1 I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

USBC1 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_PBIASLITE</a> [20] USBC1_ICUSB_PWRDNZ	usbc1_icusb_dp; usbc1_icusb_dm	Power-down mode for gpio_wk pads associated extended drain I/O cell + speed control. PUSTRENGTH control is hardware tied to 0 - (50k -100k).
<a href="#">CONTROL_MMC1</a> [24] USBC1_DR0_SPEEDCTRL		Speed control for the usbc1 I/Os. See <a href="#">Table 18-13</a> for more information.

**Table 18-82. KEYPAD I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

KEYPAD interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART1IO_PADCONF_2</a> [31:30] KPD_DR0_SC <a href="#">CONTROL_SMART1IO_PADCONF_3</a> [31:30] KPD_DR0_LB	kpd_col3; kpd_col4; kpd_col5	Low speed I/Os LB[1: 0]+SC[1:0] controls (See <a href="#">Section 18.4.12.6.3</a> for more information )
<a href="#">CONTROL_SMART1IO_PADCONF_2</a> [29:28] KPD_DR1_SC <a href="#">CONTROL_SMART1IO_PADCONF_3</a> [29:28] KPD_DR1_LB	kpd_col0; kpd_col1; kpd_col2	
<a href="#">CONTROL_SMART1IO_PADCONF_2</a> [27:26] KPD_DR2_SC <a href="#">CONTROL_SMART1IO_PADCONF_3</a> [27:26] KPD_DR2_LB	kpd_row3; kpd_row4; kpd_row5	
<a href="#">CONTROL_SMART1IO_PADCONF_2</a> [25:24] KPD_DR3_SC <a href="#">CONTROL_SMART1IO_PADCONF_3</a> [25:24] KPD_DR3_LB	kpd_row0; kpd_row1; kpd_row2	

**Table 18-83. HDQ I/O Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

HDQ interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART1IO_PADCONF_2</a> [23:22] HDQ_DR0_SC <a href="#">CONTROL_SMART1IO_PADCONF_3</a> [23:22] HDQ_DR0_LB	hdq_sio	Low speed I/Os LB[1: 0]+SC[1:0] controls (See <a href="#">Section 18.4.12.6.3</a> for more information )

**Table 18-84. GPIO Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

GPIO interface I/O controls		
Bit Fields for pad group controls	Signals (Muxmode=3) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART1NOPMIO_PADCONF_0</a> [13:12] GPIOWK4_DR0_SC <a href="#">CONTROL_SMART1NOPMIO_PADCONF_1</a> [13:12] GPIOWK4_DR0_LB	gpio_wk4	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [13:12] GPIO_63_64_DR0_SC <a href="#">CONTROL_SMART1IO_PADCONF_1</a> [13:12] GPIO_63_64_DR0_LB	gpio_63 - gpio_64	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )

**Table 18-85. SDMMC1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

SDMMC1 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_MMC1</a> [31] SDMMC1_PUSTRENGTH_GRP0;	sdmmc1_clk	Pad pullup strength control. See <a href="#">Section 18.4.12.6.8</a> , <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.
<a href="#">CONTROL_MMC1</a> [27] SDMMC1_DR0_SPEEDCTRL		Pad speed control. See <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.
<a href="#">CONTROL_MMC1</a> [30] SDMMC1_PUSTRENGTH_GRP1;	sdmmc1_cmd	Pad pullup strength control. See <a href="#">Section 18.4.12.6.8</a> , <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.
<a href="#">CONTROL_MMC1</a> [27] SDMMC1_DR0_SPEEDCTRL		Pad speed control. See <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.
<a href="#">CONTROL_MMC1</a> [29] SDMMC1_PUSTRENGTH_GRP2;	sdmmc1_dat0; sdmmc1_dat1	Pad pullup strength control. See <a href="#">Section 18.4.12.6.8</a> , <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.
<a href="#">CONTROL_MMC1</a> [27] SDMMC1_DR0_SPEEDCTRL		Pad speed control. See <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.
<a href="#">CONTROL_MMC1</a> [29] SDMMC1_PUSTRENGTH_GRP2;	sdmmc1_dat2; sdmmc1_dat3	Pad pullup strength control. See <a href="#">Section 18.4.12.6.8</a> , <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.
<a href="#">CONTROL_MMC1</a> [25] SDMMC1_DR2_SPEEDCTRL		Pad speed control. See <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.

**Table 18-85. SDMMC1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping (continued)**

SDMMC1 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_MMC1</a> [28] SDMMC1_PUSTRENGTH_GRP3;	sdmmc1_dat4 - sdmmc1_dat7	Pad pullup strength control. See <a href="#">Section 18.4.12.6.8</a> , <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.
<a href="#">CONTROL_MMC1</a> [26] SDMMC1_DR1_SPEEDCTRL		Pad speed control. See <a href="#">Section 18.4.9.2</a> and <a href="#">Section 18.5.1.2.2</a> for more details.

**Table 18-86. SDMMC5 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

SDMMC5 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [31:30] SPI2_DR1_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [15] SPI2_DR1_LB	sdmmc5_clk; sdmmc5_cmd; sdmmc5_dat0	I/Os with MB[1:0] + LB[0] settings. See <a href="#">Section 18.4.12.6.4</a> for more information.
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [29:28] SPI2_DR2_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [14] SPI2_DR2_LB	sdmmc5_dat1	I/Os with MB[1:0] + LB[0] settings. See <a href="#">Section 18.4.12.6.4</a> for more information.
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [25:24] GPIO_DR5_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [28] GPIO_DR5_LB	sdmmc5_dat2	I/Os with MB[1:0] + LB[0] settings. See <a href="#">Section 18.4.12.6.4</a> for more information.
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [23:22] GPIO_DR6_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [27] GPIO_DR6_LB	sdmmc5_dat3	I/Os with MB[1:0] + LB[0] settings. See <a href="#">Section 18.4.12.6.4</a> for more information.

**Table 18-87. ABE McBSP1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

McBSP1 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_3</a> [31:30] SLIMBUS1_DR0_MB <a href="#">CONTROL_SMART3IO_PADCONF_3</a> [19] SLIMBUS1_DR0_LB	abe_mcbbsp1_clkx	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3IO_PADCONF_3</a> [29:28] SLIMBUS1_DR1_MB <a href="#">CONTROL_SMART3IO_PADCONF_3</a> [15] SLIMBUS1_DR1_LB	abe_mcbbsp1_dr	
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [9:8] SDMMC3_DR0_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [5] SDMMC3_DR0_LB	abe_mcbbsp1_dx; abe_mcbbsp1_fsx	



**Table 18-88. ABE McBSP2 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

McBSP2 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [15:14] MCBSP2_DR0_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [23] MCBSP2_DR0_LB;	abe_mcbbsp2_clkx; abe_mcbbsp2_dr; abe_mcbbsp2_dx	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [1:0] SPI2_DR0_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [16] SPI2_DR0_LB;	abe_mcbbsp2_fsx	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )

**Table 18-89. ABE McPDM Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

McPDM interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [7:6] PDM_DR0_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [7] PDM_DR0_LB;	abe_pdm_ul_data; abe_pdm_dl_data; abe_pdm_frame; abe_pdm_lb_clk	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )

**Table 18-90. ABE DMIC Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

DMIC interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [31:30] DMIC_DR0_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [31] DMIC_DR0_LB;	abe_dmic_clk1; abe_dmic_din1	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3IO_PADCONF_3</a> [27:26] SLIMBUS2_DR0_MB; <a href="#">CONTROL_SMART3IO_PADCONF_3</a> [14] SLIMBUS2_DR0_LB;	abe_dmic_din2	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3IO_PADCONF_3</a> [23:22] SLIMBUS2_DR2_MB; <a href="#">CONTROL_SMART3IO_PADCONF_3</a> [17] SLIMBUS2_DR2_LB;	abe_dmic_din3	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )

**Table 18-91. ABE Shared Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

ABE modules shared I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [31:30] ABE_DR0_SC; <a href="#">CONTROL_SMART2IO_PADCONF_1</a> [31:30] ABE_DR0_LB;	abe_clks	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )

**Table 18-92. UART2 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

UART2 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [27:26] UART2_DR0_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [13] UART2_DR0_LB;	uart2_cts; uart2_rts; uart2_rx	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [25:24] UART2_DR1_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [12] UART2_DR1_LB;	uart2_tx	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )

**Table 18-93. UART3 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

UART3 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [7:6] UART3_DR0_SC; <a href="#">CONTROL_SMART1IO_PADCONF_1</a> [7:6] UART3_DR0_LB;	uart3_cts_rctx;	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [5:4] UART3_DR1_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [5:4] UART3_DR1_LB;	uart3_rts_sd; uart3_rx_irrx; uart3_tx_irtx	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )

**Table 18-94. UART4 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

UART4 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [23:22] UART4_DR0_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [11] UART4_DR0_LB;	uart4_rx; uart4_tx;	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )

**Table 18-95. McSPI1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

McSPI1 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [11:10] MCSP11_DR0_SC; <a href="#">CONTROL_SMART1IO_PADCONF_1</a> [11:10] MCSP11_DR0_LB;	mcspi1_clk; mcspi1_somi; mcspi1_simo; mcspi1_cs0	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [9:8] UART1_DR0_SC; <a href="#">CONTROL_SMART1IO_PADCONF_1</a> [9:8] UART1_DR0_LB;	mcspi1_cs1	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )
<a href="#">CONTROL_SMART3IO_PADCONF_3</a> [25:24] SLIMBUS2_DR1_MB; <a href="#">CONTROL_SMART3IO_PADCONF_3</a> [18] SLIMBUS2_DR1_LB;	mcspi1_cs2	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )

**Table 18-95. McSPI1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping (continued)**

McSPI1 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_3</a> [21:20] SLIMBUS2_DR3_MB; <a href="#">CONTROL_SMART3IO_PADCONF_3</a> [16] SLIMBUS2_DR3_LB;	mcspi1_cs3	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )

**Table 18-96. McSPI4 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

McSPI4 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [13:12] MCSPI4_DR0_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [22] MCSPI4_DR0_LB;	mcspi4_clk; mcspi4_simo; mcspi4_somi;	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3IO_PADCONF_0</a> [11:10] MCSPI4_DR1_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [21] MCSPI4_DR1_LB;	mcspi4_cs0	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )

**Table 18-97. I2C1 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

I2C1 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_I2C_0</a> [3] I2C1_SCL_GLFENB	i2c1_scl	Enable glitch-free operation for I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0</a> [2:1] I2C1_SCL_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_0</a> [0] I2C1_SCL_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0</a> [19] I2C1_SDA_GLFENB	i2c1_sda	Enable glitch-free operation for I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0</a> [18:17] I2C1_SDA_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_0</a> [16] I2C1_SDA_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.

**Table 18-98. I2C2 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

I2C2 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_I2C_0[7]</a> I2C2_SCL_GLFENB	i2c2_scl	Enable glitch-free operation for I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0[6:5]</a> I2C2_SCL_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_0[4]</a> I2C2_SCL_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0[23]</a> I2C2_SDA_GLFENB	i2c2_sda	Enable glitch-free operation for I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0[22:21]</a> I2C2_SDA_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_0[20]</a> I2C2_SDA_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.

**Table 18-99. I2C3 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

I2C3 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_I2C_0[11]</a> I2C3_SCL_GLFENB	i2c3_scl	Enable glitch-free operation for I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0[10:9]</a> I2C3_SCL_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_0[8]</a> I2C3_SCL_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0[27]</a> I2C3_SDA_GLFENB	i2c3_sda	Enable glitch-free operation for I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0[26:25]</a> I2C3_SDA_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_0[24]</a> I2C3_SDA_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.

**Table 18-100. I2C4 Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

I2C4 interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_I2C_0[15]</a> I2C4_SCL_GLFENB	i2c4_scl	Enable glitch-free operation for I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0[14:13]</a> I2C4_SCL_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_0[12]</a> I2C4_SCL_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0[31]</a> I2C4_SDA_GLFENB	i2c4_sda	Enable glitch-free operation for I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_0[30:29]</a> I2C4_SDA_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_0[28]</a> I2C4_SDA_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.

**Table 18-101. SR Signal Group Parameter Controls to Different Interface I/O Pads Mapping**

SR interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_I2C_2[31]</a> SR_SCL_GLFENB	sr_scl	Enable glitch-free operation for Smart Reflex I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_2[30:29]</a> SR_SCL_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_2[28]</a> SR_SCL_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_2[27]</a> SR_SDA_GLFENB	sr_sda	Enable glitch-free operation for Smart Reflex I2C receiver. See <a href="#">Section 18.4.12.6.9</a> for more information.
<a href="#">CONTROL_I2C_2[26:25]</a> SR_SDA_LOAD_BITS		LB[1:0] - load bit controls to select internal pullup resistor value . See <a href="#">Table 18-75</a> for more information.
<a href="#">CONTROL_I2C_2[24]</a> SR_SDA_PULLUPRESX		Bit to enable internal pull-up resistor option. See <a href="#">Section 18.4.12.6.9</a> for more information.

**Table 18-102. System Signals Group Parameter Controls to Different Interface I/O Pads Mapping**

System interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART1NOPMIO_PADCONF_0</a> [31:30] FREF_DR0_SC; <a href="#">CONTROL_SMART1NOPMIO_PADCONF_1</a> [31:30] FREF_DR0_LB;	fref_clk_ioreq	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )
<a href="#">CONTROL_SMART3NOPMIO_PADCONF_0</a> [31:30] FREF_DR1_MB; <a href="#">CONTROL_SMART3NOPMIO_PADCONF_1</a> [31] FREF_DR1_LB0;	fref_clk0_out	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [5:4] FREF_DR2_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [6] FREF_DR2_LB;	fref_clk1_out	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3IO_PADCONF_1</a> [3:2] FREF_DR3_MB; <a href="#">CONTROL_SMART3IO_PADCONF_2</a> [9] FREF_DR3_LB;	fref_clk2_out	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3NOPMIO_PADCONF_0</a> [23:22] FREF_DR4_MB; <a href="#">CONTROL_SMART3NOPMIO_PADCONF_1</a> [27] FREF_DR4_LB0;	fref_clk3_req	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3NOPMIO_PADCONF_0</a> [29:28] FREF_DR5_MB; <a href="#">CONTROL_SMART3NOPMIO_PADCONF_1</a> [30] FREF_DR5_LB0;	fref_clk3_out	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3NOPMIO_PADCONF_0</a> [27:26] FREF_DR6_MB; <a href="#">CONTROL_SMART3NOPMIO_PADCONF_1</a> [29] FREF_DR6_LB0;	fref_clk4_req	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART3NOPMIO_PADCONF_0</a> [25:24] FREF_DR7_MB; <a href="#">CONTROL_SMART3NOPMIO_PADCONF_1</a> [28] FREF_DR7_LB0;	fref_clk4_out	50-Ω output buffer I/Os with MB[1:0] + LB[0] settings. (See <a href="#">Section 18.4.12.6.5</a> for more information )
<a href="#">CONTROL_SMART1NOPMIO_PADCONF_0</a> [19:18] GPIO_DR7_SC; <a href="#">CONTROL_SMART1NOPMIO_PADCONF_1</a> [19:18] GPIO_DR7_LB;	sys_pwron_reset_out	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [23:22] GPIO_DR8_SC; <a href="#">CONTROL_SMART1IO_PADCONF_1</a> [23:22] GPIO_DR8_LB;	sys_nirq2	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )
<a href="#">CONTROL_SMART1IO_PADCONF_0</a> [21:20] GPIO_DR9_SC; <a href="#">CONTROL_SMART1IO_PADCONF_1</a> [21:20] GPIO_DR9_LB;	sys_boot0 -sys_boot5	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )

**Table 18-102. System Signals Group Parameter Controls to Different Interface I/O Pads Mapping (continued)**

System interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART1NOPMIO_PADCONF_0</a> [15:14] DPM_DR0_SC; <a href="#">CONTROL_SMART1NOPMIO_PADCONF_1</a> [15:14] DPM_DR0_LB;	sys_boot6; sys_boot7	Low speed I/Os LB[1: 0]+SC[1:0] controls, (See <a href="#">Section 18.4.12.6.3</a> for more information )

**Table 18-103. DPM Signals Group Parameter Controls to Different Interface I/O Pads Mapping**

System interface I/O controls		
Bit Fields for pad group controls	Pads (Muxmode=0) in group	Type of the I/O- group-associated control bit fields
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [31] DPM_DR1_DS;	dpm_emu0; dpm_emu1	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [15] USBA0_DR0_DS;	dpm_emu2	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [14] USBA0_DR1_DS;	dpm_emu3 - dpm_emu5	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [19] UART3_DR2_DS;	dpm_emu6	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [18] UART3_DR3_DS;	dpm_emu7	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [17] UART3_DR4_DS;	dpm_emu8	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [16] UART3_DR5_DS;	dpm_emu9	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [13] USBA_DR2_DS;	dpm_emu10 - dpm_emu13	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [28] GPIO_DR10_DS;	dpm_emu14 ; dpm_emu15	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [30] DPM_DR2_DS;	dpm_emu16; dpm_emu17	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)
<a href="#">CONTROL_SMART2IO_PADCONF_2</a> [29] DPM_DR3_DS;	dpm_emu18; dpm_emu19	I/O cell with single-bit drive strength (DS) control for driver output impedance (See <a href="#">Section 18.4.12.6.6</a> for more information)

#### 18.4.12.7 Dual Voltage-Supplied Peripheral Controls

Some of the peripheral I/Os have the option to be supplied with a 1,2-V or 1,8-V power supply (VDDS through chip external pads). The power supply level to the dual voltage I/Os of different peripherals is configurable through bit fields in register [CONTROL\\_CORE\\_PADCONF\\_MODE](#) in the CORE domain - (EMIF, GPMC, C2C, CAMERA, SDMMC2 I/Os) and [CONTROL\\_WKUP\\_PADCONF\\_MODE](#) in the WKUP domain - (EMIF and FREF generator).



## 18.5 Control Module Programming Guide

### 18.5.1 Control Module Low-Level Programming Models

This section covers the low-level programming sequences for configuration and use of the control module.

#### 18.5.1.1 Global Initialization

##### 18.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the control module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the environment and integration of the control module. See [Section 18.2](#), *Control Module Environment* and [Section 18.3](#), *Control Module Integration* for further information.

**Table 18-104. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	The two clocks, CFG_L4_ICLK and WKUP_L4_ICLK2 coming from PRCM are not software-controlled by PRCM. See <a href="#">Section 3.6.3.2.4</a> , <i>CM2 Clock Generator</i> and <a href="#">Section 3.6.3.1</a> , <i>PRM Clock Source</i> in <a href="#">Chapter 3</a> , <i>Power, Reset, and Clock Management</i> for more information about the two clocks.
L4 interconnect	For information about L4-CFG interconnect configuration, see <a href="#">Section 13.3</a> , <i>L4 Interconnects</i> in <a href="#">Chapter 13</a> , <i>Interconnects</i> .

##### 18.5.1.1.2 Control Module Global Initialization

###### 18.5.1.1.2.1 Software Reset

The Control Module is not sensitive to a software reset.

###### 18.5.1.1.2.2 Clock Gating Configuraton

As described in [Section 18.4.2](#), *Control Module Clock Configuration*, the programmer can choose clock-gating strategy individually for each of the four Control Module instances. [Table 18-105](#) explains the necessary steps.

**Table 18-105. Configure Clock Gating for Different Control Module Instances**

Step	Register/ bit field / Programming Model	Value
1. Set clock gating for SYCTRL_PADCONF_CORE	<a href="#">CONTROL_PADCONF_CORE_SYSCONFIG</a> [3:2]	0x-
2. Set clock gating for SYCTRL_PADCONF_WKUP	<a href="#">CONTROL_PADCONF_WKUP_SYSCONFIG</a> [3:2]	0x-
3. Set clock gating for SYCTRL_GENERAL_CORE	<a href="#">CONTROL_GEN_CORE_SYSCONFIG</a> [3:2]	0x-
4. Set clock gating for SYCTRL_GENERAL_WKUP	<a href="#">CONTROL_GEN_WKUP_SYSCONFIG</a> [3:2]	0x-

#### 18.5.1.2 Operational Modes Configuration

##### 18.5.1.2.1 Video Driver Configuration

This section gives information about all modules and features in the high-tier device.

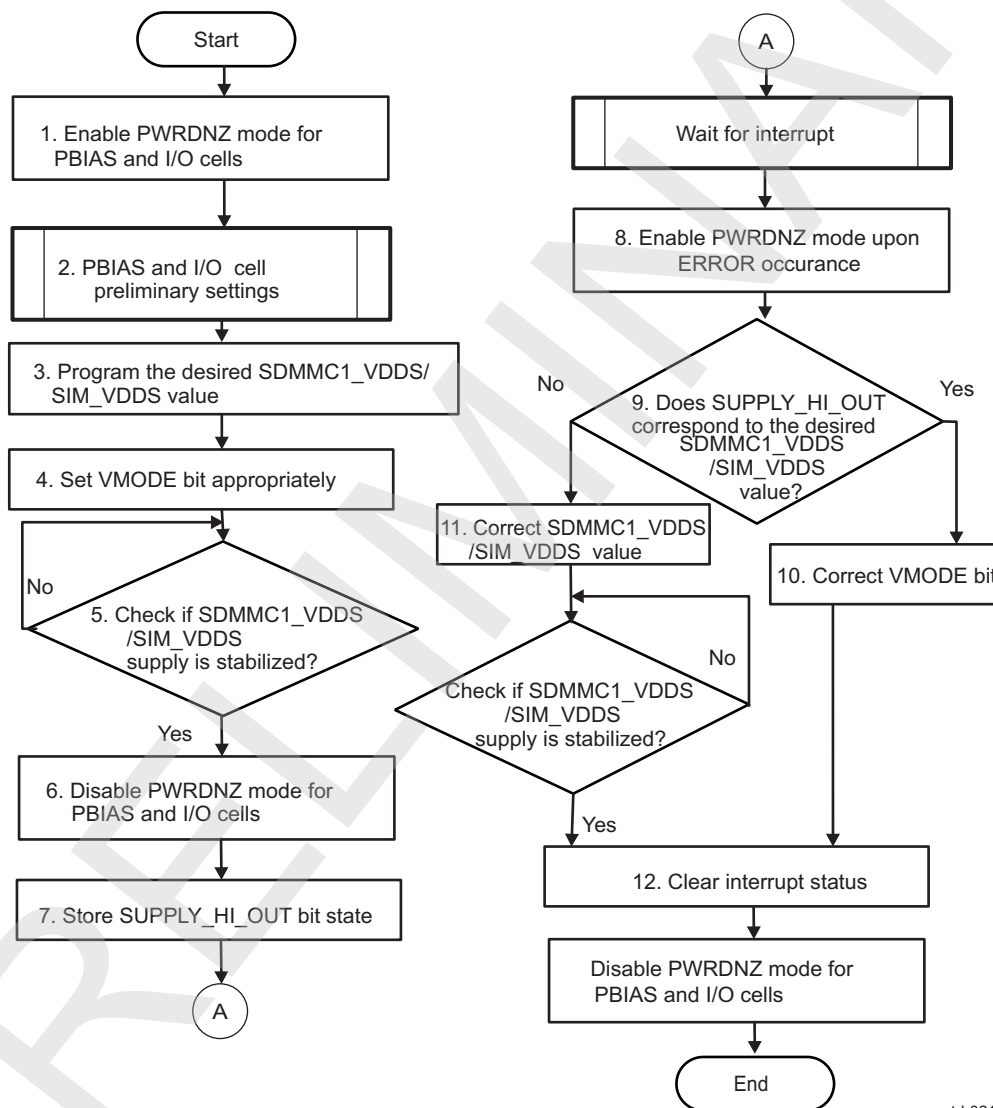
For more details on video DAC, see [Section 10.6](#), *Video Encoder*.

**Table 18-106. Control Module AVDAC Configuration**

Step	Register/ bit field / Programming Model	Value
1.Video buffer is bypassed / not bypassed	CONTROL_AVDAC[30] AVDAC_TVOUTBYPASS	0b-
2.Enable/disable ac coupling for TV output	CONTROL_AVDAC[31] AVDAC_ACEN	0b-

### 18.5.1.2.2 Extended-Drain I/Os and PBIAS Cells Programming Guide

This section describes the programming flow to go from 3.0 V to 1.8 V, and vice versa when operating with the extended-drain I/Os and PBIAS cells.

**Figure 18-18. Configuring PBIAS Cell to Operate in Different Modes, Interrupt Method**

sysctrl-021

**Table 18-107. Register Settings for Configuring the MMC1\_PBIAS and MMC1 I/O Cells**

Procedure Name	Description	Register Settings
1. Enable PWRDNZ mode for MMC1_PBIAS cell , and MMC/SD/SDIO1 I/O cell.	Software must keep MMC1_PBIAS cell and MMC1_IO cell PWRDNZ signals Software must keep MMC1_PBIAS cell and MMC1_IO cell PWRDNZ signals LOW whenever SDMMC1_VDDS is ramping up/down. or changing .This is for cell protection purposes.	1. Set <a href="#">CONTROL_PBIASLITE</a> [22] MMC1_PBIASLITE_PWRDNZ = 0b0. 2. Set <a href="#">CONTROL_PBIASLITE</a> [26] MMC1_PWRDNZ = 0b0.
2. Preliminary MMC1_PBIAS and MMC1 IO cell settings	Enable MMC1_PBIAS cell output and MMC1_PBIAS error associated interrupt Select desired I/Os within the MMC1 I/O cell and set appropriately corresponding pad configuration fields Set the group speed and pull-up strength controls for the MMC/SD/SDIO1 pads.	1. Set <a href="#">CONTROL_PBIASLITE</a> [25]MMC1_PBIASLITE_HIZ_MODE = 0b0. 2. See <a href="#">Chapter 17, Interrupt Controller</a> , for more information about setting MA_IRQ_75 line associated interrupt request PBIAS_IRQ. 3. For I/O multiplexing and pad configuration settings, see <a href="#">Section 18.5.1.2.5, Pad Configuration Programming Points</a> . 4. Set the speed control bits for the MMC1 I/O cell pads: <ul style="list-style-type: none"> <li>• <a href="#">CONTROL_MMC1</a>[27] SDMMC1_DR0_SPEEDCTRL = 0b-</li> <li>• <a href="#">CONTROL_MMC1</a>[26] SDMMC1_DR1_SPEEDCTRL = 0b-</li> <li>• <a href="#">CONTROL_MMC1</a>[28] SDMMC1_DR2_SPEEDCTRL = 0b-</li> </ul> 5. Set the pullup strength control bits for the MMC1 I/O cell pad groups: <ul style="list-style-type: none"> <li>• <a href="#">CONTROL_MMC1</a>[31] SDMMC1_PUSTRENGTH_GRP0=0b-</li> <li>• <a href="#">CONTROL_MMC1</a>[30] SDMMC1_PUSTRENGTH_GRP1=0b-</li> <li>• <a href="#">CONTROL_MMC1</a>[29] SDMMC1_PUSTRENGTH_GRP2=0b-</li> <li>• <a href="#">CONTROL_MMC1</a>[28] SDMMC1_PUSTRENGTH_GRP3=0b-</li> </ul>
3. Program the desired SDMMC1_VDDS voltage for MMC/SD/SDIO1 I/Os.	Program external power supply IC to generate appropriate SDMMC1_VDDS voltage:(3-V / 1.8-V).	See <a href="#">Section 23.1, Multimaster Highspeed I2C</a> , for more information on I <sup>2</sup> C interface operation.
4. Set VMODE bit appropriately.	PBIAS voltage generation is based on VMODE value, set according to to externally applied SDMMC1_VDDS voltage.	Set <a href="#">CONTROL_PBIASLITE</a> [21] MMC1_PBIASLITE_VMODE = 0b- <sup>(1)</sup>
5. Check if SDMMC1_VDDS supply is stabilized?	Check the state of the externally generated SDMMC1_VDDS voltage, which is provided through communication between the device and external power supply chip over I <sup>2</sup> C interface.	See <a href="#">Section 23.1, Multimaster Highspeed I2C</a> , for more information on I <sup>2</sup> C interface settings.
6. Disable PWRDNZ (power down) mode for MMC1_PBIAS and associated extended-drain MMC1 I/O cell.	This starts up the MMC1_PBIAS cell work to generate the PBIAS voltage.	1. Set <a href="#">CONTROL_PBIASLITE</a> [22] MMC1_PBIASLITE_PWRDNZ = 0b1. 2. Set <a href="#">CONTROL_PBIASLITE</a> [26] MMC1_PWRDNZ = 0b1.
7. Store SUPPLY_HI_OUT bit state.	Done as a preliminary step for error handling analysis. (Whenever PWRDNZ = 0b0 , SUPPLY_HI_OUT is unconditionally set to 0b0.)	Store <a href="#">CONTROL_PBIASLITE</a> [24] MMC1_PBIASLITE_SUPPLY_HI_OUT bit in memory.
8.Enable PWRDNZ mode upon error occurrence.	It is recommended that all the I/Os are powered down (PWRDNZ = 0) as soon as VMODE_ERROR is high.	1. Set <a href="#">CONTROL_PBIASLITE</a> [26] MMC1_PWRDNZ = 0b0 . 2. Set <a href="#">CONTROL_PBIASLITE</a> [22]MMC1_PBIASLITE_PWRDNZ = 0b0.
9. Does SUPPLY_HI_OUT correspond to the desired SDMMC1_VDDS?	Test if error comes from incorrectly set SDMMC1_VDDS value or incorrectly set VMODE bit.	Check the stored in memory SUPPLY_HI_OUT bit.
10. Correct VMODE bit	If SUPPLY_HI_OUT level corresponds to desired SDMMC1_VDDS value, then correct the value of the VMODE bit.	Modify value of <a href="#">CONTROL_PBIASLITE</a> [21] MMC1_PBIASLITE_VMODE.
11. Correct SDMMC1_VDDS value.	If SUPPLY_HI_OUT is different from expected , correct and set SDMMC1_VDDS value over I <sup>2</sup> C again	See <a href="#">Section 23.1, Multimaster Highspeed I2C</a> , for more information on I <sup>2</sup> C interface settings.

<sup>(1)</sup> Should be set HIGH when external MMC1\_VDDS source programmed to be 3V, or LOW when external MMC1\_VDDS source programmed to be 1.8 V. Reset default value corresponds to 3-V .

**Table 18-107. Register Settings for Configuring the MMC1\_PBIAS and MMC1 I/O Cells (continued)**

Procedure Name	Description	Register Settings
12. Clear Interrupt status.	Clear the IRQ_MA_75 interrupt status flag.	See <a href="#">Chapter 17, Interrupt Controller</a> , for more information about setting MA_IRQ_75 line associated interrupt request PBIAS_IRQ.

**Table 18-108. Register Settings for Configuring the PBIAS1 Cell, USBC1\_ICUSB and gpio\_wk0gpio\_wk2 Associated I/O Cells**

Procedure Name	Description	Register Settings
1. Enable PWRDNZ (power down) mode for PBIAS1 cell ,USBC1_ICUSB I/O cell and gpio_wk0-gpio_wk2 associated I/O cell.	Software must keep PBIAS1 cell and the associated USBC1_ICUSB and gpio_wk0-gpio_wk2 I/O cells PWRDNZ signals LOW whenever SIM_VDDS is ramping up/down or changing. This is for cell protection purposes.	<ol style="list-style-type: none"> <li>1. Set <a href="#">CONTROL_PBIASLITE</a>[28] PBIASLITE1_PWRDNZ = 0b0.</li> <li>2. Set <a href="#">CONTROL_PBIASLITE</a>[20] USBC1_ICUSB_PWRDNZ = 0b0.</li> <li>3. Set <a href="#">CONTROL_GPIOWK</a>[28] GPIOWK_IO_PWRDNZ = 0b0.</li> </ol>
2. Preliminary MMC1_PBIAS and MMC1 IO cell settings	Enable MMC1_PBIAS cell output and MMC1_PBIAS error associated interrupt Select desired I/Os within the USBC1_ICUSB and GPIOWK I/O cells and set appropriately their corresponding pad configuration fields Set the speed and pull-down controls for the pads (if applicable).	<ol style="list-style-type: none"> <li>1. Set <a href="#">CONTROL_PBIASLITE</a>[31] PBIASLITE1_HIZ_MODE = 0b0.</li> <li>2. See <a href="#">Chapter 17, Interrupt Controller</a>, for more information about setting MA_IRQ_75 line associated interrupt request PBIAS_IRQ.</li> <li>3. For I/O multiplexing and pad configuration settings, see <a href="#">Section 18.5.1.2.5, Pad Configuration Programming Points</a>.</li> <li>4. Set the speed control bits for USBC1_ICUSB I/O cell pads <a href="#">CONTROL_MMC1</a>[24] USBC1_DR0_SPEEDCTRL = 0b- (No speedctrl available for GPIOWK associated I/O cell).</li> <li>5. Disable strong pulldowns on the gpio_wk1 and gpio_wk2 pads available by default after reset: <ul style="list-style-type: none"> <li>• <a href="#">CONTROL_GPIOWK</a>[31] PAD_GPIO_WK1_LOW = 0b0</li> <li>• <a href="#">CONTROL_GPIOWK</a>[29] PAD_GPIO_WK2_LOW = 0b0</li> </ul> </li> </ol>
3. Program the desired SIM_VDDS voltage for the USBC1_ICUSB / gpio_wk0-gpio_wk2 or( hw_dbg1 - hw_dbg3 in WKUP domain) I/Os .	Program external power supply IC to generate appropriate SIM_VDDS voltage:(3-V/1.8-V)	See <a href="#">Section 23.1, Multimaster Highspeed I2C</a> , for more information on I <sup>2</sup> C interface operation.
4. Set VMODE bit appropriately.	PBIAS voltage generation is based on VMODE value, set according to to externally applied SIM_VDDS voltage.	Set <a href="#">CONTROL_PBIASLITE</a> [27] PBIASLITE1_VMODE = 0b-. <sup>(1)</sup>
5. Check if SIM_VDDS supply is stabilized?	Check the state of the externally generated SIM_VDDS voltage. This is provided through communication between device and companion power supply chip over I2C interface.	See <a href="#">Section 23.1, Multimaster Highspeed I2C</a> , for more information on I <sup>2</sup> C interface settings.
6. Disable PWRDNZ (power down) mode for PBIAS1 cell and its associated extended-drain USBC1_ICUSB and gpio_wk0-gpio_wk2 I/O cells.	This starts up the the PBIAS1 cell work to generate the PBIAS voltage.	<ol style="list-style-type: none"> <li>1. Set <a href="#">CONTROL_PBIASLITE</a>[28] PBIASLITE1_PWRDNZ = 0b1.</li> <li>2. Set <a href="#">CONTROL_PBIASLITE</a>[20 ] USBC1_ICUSB_PWRDNZ = 0b1.</li> <li>3. Set <a href="#">CONTROL_GPIOWK</a>[28] GPIOWK_IO_PWRDNZ = 0b1.</li> </ol>
7. Store SUPPLY_HI_OUT bit state.	Done as a preliminary step for error handling analysis. (Whenever PWRDNZ = 0b0 , SUPPLY_HI_OUT is unconditionally set to 0b0.)	Store <a href="#">CONTROL_PBIASLITE</a> [30] PBIASLITE1_SUPPLY_HI_OUT bit in memory.
8.Enable PWRDNZ mode upon error occurrence.	It is recommended that all the I/Os are powered down (PWRDNZ = 0) as soon as VMODE_ERROR is high.	<ol style="list-style-type: none"> <li>1. Set <a href="#">CONTROL_PBIASLITE</a>[28] PBIASLITE1_PWRDNZ = 0b0.</li> <li>2. Set <a href="#">CONTROL_PBIASLITE</a>[20] USBC1_ICUSB_PWRDNZ = 0b0.</li> <li>3. Set <a href="#">CONTROL_GPIOWK</a>[28] GPIOWK_IO_PWRDNZ = 0b0.</li> </ol>

<sup>(1)</sup> Should be set HIGH when external SIM\_VDDS source programmed to be 3V, or LOW when external SIM\_VDDS source programmed to be 1.8 V. Reset default value corresponds to 3-V . Reset default value corresponds to 3-V .

**Table 18-108. Register Settings for Configuring the PBIAS1 Cell, USBC1\_ICUSB and gpio\_wk0gpio\_wk2 Associated I/O Cells (continued)**

Procedure Name	Description	Register Settings
9. Does SUPPLY_HI_OUT correspond to the desired SIM_VDDS level?	Test if error comes from incorrectly set SIM_VDDS value or incorrectly set VMODE bit.	Check the stored in memory SUPPLY_HI_OUT bit.
10. Correct VMODE bit.	If SUPPLY_HI_OUT level corresponds to desired SIM_VDDS value, then correct the value of VMODE bit.	Modify value of <a href="#">CONTROL_PBIASLITE[27]</a> PBIASLITE1_VMODE.
11. Correct SIM_VDDS value.	If SUPPLY_HI_OUT is different than expected, correct and set SIM_VDDS value over I <sup>2</sup> C again.	See <a href="#">Section 23.1, Multimaster Highspeed I2C</a> , for more information on I <sup>2</sup> C interface settings.
12. Clear Interrupt status.	Clear the IRQ_MA_75 interrupt status flag.	See <a href="#">Chapter 17, Interrupt Controller</a> , for more information about setting MA_IRQ_75 line associated interrupt request PBIAS_IRQ.

### 18.5.1.2.3 Hardware Observability Settings

This section describes the procedures to control hardware observability features of the device. To configure hardware observability feature, the programmer must perform the steps listed in [Table 18-109](#).

**Table 18-109. Configure Hardware Observability Features**

Step	Register/Bit Field/Programming Model	Value
1. Select hwobs(0) signal frequency divider value	<a href="#">CONTROL_HWOBS_CONTROL[7:3]</a> HWOBS_CLKDIV_SEL	0x- <sup>(1)</sup>
2. Enable/disable hwobs gating and alternatively set all hwobs ports to either 0 or 1.	1. <a href="#">CONTROL_HWOBS_CONTROL[2]</a> HWOBS_ALL_ZERO_MODE 2. <a href="#">CONTROL_HWOBS_CONTROL[1]</a> HWOBS_ALL_ONE_MODE	0x- <sup>(2)</sup>
3. Enable/disable hwobs port gating only for signals coming from IPs identified in the MACRO group.	<a href="#">CONTROL_HWOBS_CONTROL[0]</a> HWOBS_MACRO_ENABLE	0x- <sup>(3)</sup>
4. Associate a certain hwobs channel to either CORE or WKUP power domain	Configure each bit from the bit field <a href="#">CONTROL_DEBOBS_FINAL_MUX_SEL[31:0]</a> SELECT appropriately for its corresponding hw_dbg channel	0x-
5. Choose the source of a signal or signal view channel in the CORE power domain	Configure each of the bit fields <a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_0[3:0]</a> MODE through <a href="#">CONTROL_CORE_CONF_DEBUG_SEL_TST_31[3:0]</a> MODE appropriately for its corresponding hw_dbg channel	0x-
6. Choose the source of a signal within the WKUP power domain	Configure each bit <a href="#">CONTROL_WKUP_CONF_DEBUG_SEL_TST_i [0]</a> MODE, (where i=0 to 31) appropriately for its corresponding hw_dbg channel i (where i=0 to 31)	0x-
7. Choose to associate a certain DPLL signal to a given DPLL signal view channel	1. <a href="#">CONTROL_CONF_DPLL_FREQLOCK_SEL[2:0]</a> MULT 2. <a href="#">CONTROL_CONF_DPLL_TINITZ_SEL[2:0]</a> MULT 3. <a href="#">CONTROL_CONF_DPLL_PHASELOCK_SEL[2:0]</a> MULT 4. <a href="#">CONTROL_CONF_DPLL_TENABLE_SEL[2:0]</a> MULT 5. <a href="#">CONTROL_CONF_DPLL_TENABLEDIV_SEL[2:0]</a> MULT 6. <a href="#">CONTROL_CONF_DPLL_BYPASSACK_SEL[2:0]</a> MULT 7. <a href="#">CONTROL_CONF_DPLL_IDLE_SEL[2:0]</a> MULT	0x-
8. Choose to gate all the observable clocks at DPLL outputs simultaneously	<a href="#">CONTROL_HWOBS_CONTROL[8]</a> HWOBS_CLKOBS_GATE_ENABLE	0x-
9. Choose to associate a certain clockout signal to a given clockview channel	<a href="#">CONTROL_CONF_CLK_SEL0[2:0]</a> MULT through <a href="#">CONTROL_CONF_CLK_SEL2[2:0]</a> MULT	0x-
10. Choose to associate an sDMA signal to a given sDMA view channel	<a href="#">CONTROL_CONF_SDMA_REQ_SEL0[6:0]</a> MULT through <a href="#">CONTROL_CONF_SDMA_REQ_SEL3[6:0]</a> MULT	0x-

<sup>(1)</sup> To configure hwobs(0) channel to operate normally , HWOBS\_CLKDIV\_SEL bit value should be set different from 0.

<sup>(2)</sup> Set both bits to 0 to enable hardware observability. Set only one of them to 1 to gate hwobs ports according to output state required.

<sup>(3)</sup> The modules which belong to this group are: Cortex-A9 MPU subsystem, IVAHD, display subsystem , CM2, and ABE.

**Table 18-109. Configure Hardware Observability Features (continued)**

Step	Register/Bit Field/Programming Model	Value
11. Set the appropriate pad configuration bit fields to associate hw_dbgn pad function to output pins and configure additional output pad features	See <a href="#">Section 18.4.11, Hardware Observability</a> and <a href="#">Section 18.4.8, Pad Functional Multiplexing and Configuration</a> , for more information.	See <a href="#">Section 18.4.8, PAD Functional Multiplexing and Configuration</a> for more information

See [Section 18.4.11, Hardware Observability](#), for more details on different functionalities and registers associated with hardware observability features.

#### 18.5.1.2.4 Off Mode Preliminary Settings

This section describes configurations of PADs to work in off mode. The following actions must be performed once, and remain valid for all device OFF <-> ON transitions.

**Table 18-110. Setting PADS Off Mode Through Padconfiguration Registers**

Step	Register/ bit field / Programming Model	Value
1. Set OFFMODEPULLTYPESELECT bit	CONTROL_CORE_PAD0_X_PAD1_Y[29,13] Select pullup/down type when PAD in off mode	0b-
2. Set OFFMODEPULLUDENABLE bit	CONTROL_CORE_PAD0_X_PAD1_Y[28,12] Enable/disable pullup/down feature when PAD in off mode	0b-
3. Set OFFMODEOUTVALUE bit	CONTROL_CORE_PAD0_X_PAD1_Y[27,11] Set off mode pad override value	0b-
4. Set OFFMODEENABLE bit (enable/disable PAD overriding capability in OFF mode)	CONTROL_CORE_PAD0_X_PAD1_Y[25,9] Set pad to keep its previous state (it had before going to off mode) or force pad output to a certain override value (OFFMODEOUTVALUE) when PAD in off mode	0b-
5. Set OFFMODEOUTENABLE bit	CONTROL_CORE_PAD0_X_PAD1_Y[26,10] Enable/disable output in off mode	0b-
6. Program a valid device ACTIVE pads configuration	See <a href="#">Table 18-111</a> for more details on ACTIVE pads configuration	
7. Set clock gating strategy	1. CONTROL_PADCONF_CORE_SYSCONFIG[3:2]IP_SYSCONFIG 2. CONTROL_PADCONF_WKUP_SYSCONFIG[3:2]IP_SYSCONFIG	0x2,0x3
8. Enable/disable the wake-up detection capability <sup>(1)</sup>	CONTROL_CORE_PAD0_X_PAD1_Y[30,14]	0b-

<sup>(1)</sup> When the wake-up detection is enabled for a pad, this pad is configured as input. Therefore, do not forget to write 0b1 in the OFFOUTENABLE bit to disable the output capability.

#### 18.5.1.2.5 Pad Configuration Programming Points

This section describes through an example the configurations of PADs necessary for working in active mode.

To configure the pad, ensure that the following steps are done:

- Identify signals required on the interface based on the target application.  
**Example:** To configure the UART2 interface on balls, the required signals are uart2\_tx, uart2\_rts, uart2\_cts, and uart2\_rx.
- Choose the pads used for those signals. Some signals could be available on several pads and/or may be multiplexed with other signals needed for another application. See [Section 18.4.8.3, Pad Multiplexing Register Fields](#).

**Example:** Unlike the UART2 interface signals - uart2\_rx and uart2\_tx, each of which is available on two pads, the uart2\_cts and uart2\_rts signals are available only on one pad each (MUXMODE=0b000). However the uart2\_cts is multiplexed with the MMC I/O signal sdmmc3\_clk and the uart2\_rts signals is multiplexed with the MMC I/O signal sdmmc3\_cmd on the same pad. Assume that the MMC I/O 3 interface is also required in the system. Therefore, for the MMC I/O interface signals, pads must be

used where those signals are not multiplexed with a UART2 signal if possible. In this case the solution is to use sdmmc3\_clk and sdmmc3\_cmd on another two pads of the device - usbb2\_ulpitll\_dat7 and usbb2\_ulpitll\_dat6, configuring (MUXMODE=0b010).

- Identify the pad configuration registers associated with the pads to be used in the application. See [Section 18.4.8.3, Pad Multiplexing Register Fields](#).

**Example:** Under the previous hypothesis, the pad configuration registers to program are:

- uart2\_rts: [CONTROL\\_CORE\\_PAD0\\_UART2\\_CTS\\_PAD1\\_UART2\\_RTS\[31:16\]](#)
- uart2\_cts: [CONTROL\\_CORE\\_PAD0\\_UART2\\_CTS\\_PAD1\\_UART2\\_RTS\[15:0\]](#)
- uart2\_tx: [CONTROL\\_CORE\\_PAD0\\_UART2\\_RX\\_PAD1\\_UART2\\_TX\[31:16\]](#)
- uart2\_rx: [CONTROL\\_CORE\\_PAD0\\_UART2\\_RX\\_PAD1\\_UART2\\_TX\[15:0\]](#)

**Table 18-111. ACTIVE PADS Configuration Points**

Step	Register/ bit field / Programming Model	Value
1. Configure MUXMODE field in pad configuration registers associated with the pads used	1. <a href="#">CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS[18:16]</a> UART2_RTS_MUXMODE	0b000 0b000
	2. <a href="#">CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS[2:0]</a> UART2_CTS_MUXMODE	0b000 (0b010)
	3. <a href="#">CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX[18:16]</a> UART2_TX_MUXMODE	0b000 (0b010)
	4. <a href="#">CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX[2:0]</a> UART2_RX_MUXMODE	
2. Select the type of the internal pullup/down resistor <sup>(1)</sup>	1. <a href="#">CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX[4]</a> UART2_RX_PULLTYPESELECT	0b1 (pullup selected)
	2. <a href="#">CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS[2:0]</a> UART2_CTS_PULLTYPESELECT	0b1 (pullup selected)
3. Enable/Disable pullup/down feature of the pad <sup>(2)</sup>	1. <a href="#">CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX[3]</a> UART2_RX_PULLUDENABLE	0b1 (Pull feature enabled)
	2. <a href="#">CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS[3]</a> UART2_CTS_PULLUDENABLE	0b1 (Pull feature enabled)
4. Set the direction of the pin signal	1. <a href="#">CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX[8]</a> UART2_RX_INPUTENABLE	0b1 (input pin)
	2. <a href="#">CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS[8]</a> UART2_CTS_INPUTENABLE	0b1 (input pin)
	3. <a href="#">CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX[24]</a> UART2_TX_INPUTENABLE	0b0 (output pin)
	4. <a href="#">CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS[24]</a> UART2_RTS_INPUTENABLE	0b0 (output pin)

<sup>(1)</sup> It is recommended to use internal pullup/down resistor if pad is used as input and external pullup/down is not used.

<sup>(2)</sup> Pullup/down feature is automatically disabled if pad is used as an output, regardless of the value of PULLUDENABLE.

**NOTE:** The order for setting the previous pad configuration bits is not important.

## 18.6 Control Module Register Manual

### 18.6.1 Control Module Instance Summary

Table 18-112 lists the base address and address space for the control module instances.

**Table 18-112. Control Module Instance Summary**

Module instance	Base Address	Size	Description
SYSCTRL_GENERAL_C ORE	0x4A00 2000	4KB	Module
SYSCTRL_GENERAL_W KUP	0x4A30 C000	4KB	Module
SYSCTRL_PADCONF_C ORE	0x4A10 0000	4KB	Module
SYSCTRL_PADCONF_W KUP	0x4A31 E000	4KB	Module

**NOTE:** All module registers are 8-, 16-, or 32-bit accessible through the L4 interconnect (little-endian encoding).

### 18.6.2 SYSCTRL\_GENERAL\_CORE Register Summary

**Table 18-113. SYSCTRL\_GENERAL\_CORE Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_GENERAL_C ORE Physical Address
CONTROL_GEN_CORE_REVISION	R	32	0x00000000	0x4A002000
CONTROL_GEN_CORE_HWINFO	R	32	0x00000004	0x4A002004
CONTROL_GEN_CORE_SYSCONFIG	RW	32	0x00000010	0x4A002010
CONTROL_STD_FUSE_DIE_ID_0	R	32	0x0000 0200	0x4A00 2200
CONTROL_ID_CODE	R	32	0x0000 0204	0x4A00 2204
CONTROL_STD_FUSE_DIE_ID_1	R	32	0x0000 0208	0x4A00 2208
CONTROL_STD_FUSE_DIE_ID_2	R	32	0x0000 020C	0x4A00 220C
CONTROL_STD_FUSE_DIE_ID_3	R	32	0x0000 0210	0x4A00 2210
CONTROL_STD_FUSE_PROD_ID_0	R	32	0x0000 0214	0x4A00 2214
CONTROL_STD_FUSE_PROD_ID_1	R	32	0x0000 0218	0x4A00 2218
CONTROL_STD_FUSE_USB_CONF	R	32	0x0000 021C	0x4A00 221C
RESERVED	R	32	0x0000 0220	0x4A00 2220
CONTROL_STD_FUSE_OPP_VDD_WKUP	R	32	0x0000 0228	0x4A00 2228
CONTROL_STD_FUSE_OPP_VDD_IVA_0	R	32	0x0000 022C	0x4A00 222C
CONTROL_STD_FUSE_OPP_VDD_IVA_1	R	32	0x0000 0230	0x4A00 2230
CONTROL_STD_FUSE_OPP_VDD_IVA_2	R	32	0x0000 0234	0x4A00 2234
CONTROL_STD_FUSE_OPP_VDD_IVA_3	R	32	0x0000 0238	0x4A00 2238
CONTROL_STD_FUSE_OPP_VDD_IVA_4	R	32	0x0000 023C	0x4A00 223C
CONTROL_STD_FUSE_OPP_VDD_MPU_0	R	32	0x0000 0240	0x4A00 2240
CONTROL_STD_FUSE_OPP_VDD_MPU_1	R	32	0x0000 0244	0x4A00 2244
CONTROL_STD_FUSE_OPP_VDD_MPU_2	R	32	0x0000 0248	0x4A00 2248
CONTROL_STD_FUSE_OPP_VDD_MPU_3	R	32	0x0000 024C	0x4A00 224C
CONTROL_STD_FUSE_OPP_VDD_MPU_4	R	32	0x0000 0250	0x4A00 2250
CONTROL_STD_FUSE_OPP_VDD_CORE_0	R	32	0x0000 0254	0x4A00 2254
CONTROL_STD_FUSE_OPP_VDD_CORE_1	R	32	0x0000 0258	0x4A00 2258
CONTROL_STD_FUSE_OPP_VDD_CORE_2	R	32	0x0000 025C	0x4A00 225C



**Table 18-113. SYSCTRL\_GENERAL\_CORE Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_GENERAL_CORE Physical Address
CONTROL_STD_FUSE_OPP_BGAP	R	32	0x0000 0260	0x4A00 2260
CONTROL_STD_FUSE_OPP_DPLL_0	R	32	0x0000 0264	0x4A00 2264
CONTROL_STD_FUSE_OPP_DPLL_1	R	32	0x0000 0268	0x4A00 2268
CONTROL_STATUS	R	32	0x0000 02C4	0x4A00 22C4
CONTROL_SEC_ERR_STATUS_FUNC	RW	32	0x0000 02D0	0x4A00 22D0
CONTROL_SEC_ERR_STATUS_DEBUG	RW	32	0x0000 02D4	0x4A00 22D4
CONTROL_STD_FUSE_OPP_VDD_MPU_5	R	32	0x0000 02D8	0x4A00 22D8
CONTROL_STD_FUSE_OPP_VDD_IVA_5	R	32	0x0000 02DC	0x4A00 22DC
CONTROL_DEV_CONF	RW	32	0x0000 0300	0x4A00 2300
RESERVED	RW	32	0x0000 0304	0x4A00 2304
CONTROL_LDOVBB_IVA_VOLTAGE_CTRL	RW	32	0x0000 0314	0x4A00 2314
CONTROL_LDOVBB_MPU_VOLTAGE_CTRL	RW	32	0x0000 0318	0x4A00 2318
CONTROL_LDOSRAM_IVA_VOLTAGE_CTRL	RW	32	0x0000 0320	0x4A00 2320
CONTROL_LDOSRAM_MPU_VOLTAGE_CTRL	RW	32	0x0000 0324	0x4A00 2324
CONTROL_LDOSRAM_CORE_VOLTAGE_CTRL	RW	32	0x0000 0328	0x4A00 2328
CONTROL_TEMP_SENSOR	RW	32	0x0000 032C	0x4A00 232C
CONTROL_DPLL_NWELL_TRIM_0	RW	32	0x0000 0330	0x4A00 2330
CONTROL_DPLL_NWELL_TRIM_1	RW	32	0x0000 0334	0x4A00 2334
RESERVED	RW	32	0x0000 0338	0x4A00 2338
CONTROL_USBOTGHS_CONTROL	RW	32	0x0000 033C	0x4A00 233C
CONTROL_DSS_CONTROL	RW	32	0x0000 0340	0x4A00 2340
RESERVED	RO	32	0x0000 0344	0x4A00 2344
CONTROL_CORTEX_M3_MMUADDRTRANSLTR	RW	32	0x0000 0348	0x4A00 2348
CONTROL_CORTEX_M3_MMUADDRLOGICTR	RW	32	0x0000 034C	0x4A00 234C
CONTROL_HWOBS_CONTROL	RW	32	0x0000 0350	0x4A00 2350
RESERVED	RO	32	0x0000 03FC	0x4A00 23FC
CONTROL_DEBOBS_FINAL_MUX_SEL	RW	32	0x0000 0400	0x4A00 2400
RESERVED	RO	32	0x0000 0404	0x4A00 2404
CONTROL_DEBOBS_MMR_MPU	RW	32	0x0000 0408	0x4A00 2408
RESERVED	RO	32	0x0000 0410	0x4A00 2410
RESERVED	RO	32	0x0000 0414	0x4A00 2414
RESERVED	RO	32	0x0000 0418	0x4A00 2418
RESERVED	RO	32	0x0000 041C	0x4A00 241C
RESERVED	RO	32	0x0000 0420	0x4A00 2420
RESERVED	RO	32	0x0000 0424	0x4A00 2424
CONTROL_CONF_SDMA_REQ_SEL0	RW	32	0x0000 042C	0x4A00 242C
CONTROL_CONF_SDMA_REQ_SEL1	RW	32	0x0000 0430	0x4A00 2430
CONTROL_CONF_SDMA_REQ_SEL2	RW	32	0x0000 0434	0x4A00 2434
CONTROL_CONF_SDMA_REQ_SEL3	RW	32	0x0000 0438	0x4A00 2438
CONTROL_CONF_CLK_SEL0	RW	32	0x0000 0440	0x4A00 2440
CONTROL_CONF_CLK_SEL1	RW	32	0x0000 0444	0x4A00 2444
CONTROL_CONF_CLK_SEL2	RW	32	0x0000 0448	0x4A00 2448
CONTROL_CONF_DPLL_FREQLOCK_SEL	RW	32	0x0000 044C	0x4A00 244C
CONTROL_CONF_DPLL_TINITZ_SEL	RW	32	0x0000 0450	0x4A00 2450

**Table 18-113. SYSCTRL\_GENERAL\_CORE Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_GENERAL_CORE Physical Address
CONTROL_CONF_DPLL_PHASELOCK_SEL	RW	32	0x0000 0454	0x4A00 2454
RESERVED	RW	32	0x0000 0458	0x4A00 2458
CONTROL_CONF_DPLL_TENABLE_SEL	RW	32	0x0000 045C	0x4A00 245C
CONTROL_CONF_DPLL_BYPASSACK_SEL	RW	32	0x0000 0464	0x4A00 2464
CONTROL_CONF_DPLL_IDLE_SEL	RW	32	0x0000 0468	0x4A00 2468
CONTROL_CORE_CONF_DEBUG_SEL_TST_0	RW	32	0x0000 0480	0x4A00 2480
CONTROL_CORE_CONF_DEBUG_SEL_TST_1	RW	32	0x0000 0484	0x4A00 2484
CONTROL_CORE_CONF_DEBUG_SEL_TST_2	RW	32	0x0000 0488	0x4A00 2488
CONTROL_CORE_CONF_DEBUG_SEL_TST_3	RW	32	0x0000 048C	0x4A00 248C
CONTROL_CORE_CONF_DEBUG_SEL_TST_4	RW	32	0x0000 0490	0x4A00 2490
CONTROL_CORE_CONF_DEBUG_SEL_TST_5	RW	32	0x0000 0494	0x4A00 2494
CONTROL_CORE_CONF_DEBUG_SEL_TST_6	RW	32	0x0000 0498	0x4A00 2498
CONTROL_CORE_CONF_DEBUG_SEL_TST_7	RW	32	0x0000 049C	0x4A00 249C
CONTROL_CORE_CONF_DEBUG_SEL_TST_8	RW	32	0x0000 04A0	0x4A00 24A0
CONTROL_CORE_CONF_DEBUG_SEL_TST_9	RW	32	0x0000 04A4	0x4A00 24A4
CONTROL_CORE_CONF_DEBUG_SEL_TST_10	RW	32	0x0000 04A8	0x4A00 24A8
CONTROL_CORE_CONF_DEBUG_SEL_TST_11	RW	32	0x0000 04AC	0x4A00 24AC
CONTROL_CORE_CONF_DEBUG_SEL_TST_12	RW	32	0x0000 04B0	0x4A00 24B0
CONTROL_CORE_CONF_DEBUG_SEL_TST_13	RW	32	0x0000 04B4	0x4A00 24B4
CONTROL_CORE_CONF_DEBUG_SEL_TST_14	RW	32	0x0000 04B8	0x4A00 24B8
CONTROL_CORE_CONF_DEBUG_SEL_TST_15	RW	32	0x0000 04BC	0x4A00 24BC
CONTROL_CORE_CONF_DEBUG_SEL_TST_16	RW	32	0x0000 04C0	0x4A00 24C0
CONTROL_CORE_CONF_DEBUG_SEL_TST_17	RW	32	0x0000 04C4	0x4A00 24C4
CONTROL_CORE_CONF_DEBUG_SEL_TST_18	RW	32	0x0000 04C8	0x4A00 24C8
CONTROL_CORE_CONF_DEBUG_SEL_TST_19	RW	32	0x0000 04CC	0x4A00 24CC
CONTROL_CORE_CONF_DEBUG_SEL_TST_20	RW	32	0x0000 04D0	0x4A00 24D0
CONTROL_CORE_CONF_DEBUG_SEL_TST_21	RW	32	0x0000 04D4	0x4A00 24D4
CONTROL_CORE_CONF_DEBUG_SEL_TST_22	RW	32	0x0000 04D8	0x4A00 24D8
CONTROL_CORE_CONF_DEBUG_SEL_TST_23	RW	32	0x0000 04DC	0x4A00 24DC
CONTROL_CORE_CONF_DEBUG_SEL_TST_24	RW	32	0x0000 04E0	0x4A00 24E0

**Table 18-113. SYSCTRL\_GENERAL\_CORE Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_GENERAL_CORE Physical Address
CONTROL_CORE_CONF_DEBUG_SEL_TST_25	RW	32	0x0000 04E4	0x4A00 24E4
CONTROL_CORE_CONF_DEBUG_SEL_TST_26	RW	32	0x0000 04E8	0x4A00 24E8
CONTROL_CORE_CONF_DEBUG_SEL_TST_27	RW	32	0x0000 04EC	0x4A00 24EC
CONTROL_CORE_CONF_DEBUG_SEL_TST_28	RW	32	0x0000 04F0	0x4A00 24F0
CONTROL_CORE_CONF_DEBUG_SEL_TST_29	RW	32	0x0000 04F4	0x4A00 24F4
CONTROL_CORE_CONF_DEBUG_SEL_TST_30	RW	32	0x0000 04F8	0x4A00 24F8
CONTROL_CORE_CONF_DEBUG_SEL_TST_31	RW	32	0x0000 04FC	0x4A00 24FC

**18.6.3 SYSCTRL\_GENERAL\_CORE Register Description**

**Table 18-114. CONTROL\_GEN\_CORE\_REVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Control module instance revision identifier		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 18-115. Register Call Summary for Register CONTROL\_GEN\_CORE\_REVISION**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-116. CONTROL\_GEN\_CORE\_HWINFO**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Information about the IP module hardware configuration		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP_HWINFO																															

Bits	Field Name	Description	Type	Reset
31:0	IP_HWINFO	IP-module dependent	R	0x0000 0000

**Table 18-117. Register Call Summary for Register CONTROL\_GEN\_CORE\_HWINFO**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-118. CONTROL\_GEN\_CORE\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IP_SYSCONFIG_IDLEMODE			RESERVED												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:2	IP_SYSCONFIG_IDLEMODE	Select the local clock-gating strategy 0x0: Clock is manually gated 0x1: Clock is manually enabled 0x2,0x3: Clock is automatically gated when there is no access to the Control Module through L4-interconnect	RW	0x2
1:0	RESERVED		R	0x0

**Table 18-119. Register Call Summary for Register CONTROL\_GEN\_CORE\_SYSCONFIG**

Control Module Functional Description

- [General Core Control Module Instance: \[0\]](#)

Control Module Programming Guide

- [Control Module Global Initialization: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[2\]](#)

**Table 18-120. CONTROL\_STD\_FUSE\_DIE\_ID\_0**

<b>Address Offset</b>	0x0000 0200	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Die ID Register - Part 0. Access conditions. Read: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	R	Single identifier

**Table 18-121. Register Call Summary for Register CONTROL\_STD\_FUSE\_DIE\_ID\_0**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-122. CONTROL\_ID\_CODE**

<b>Address Offset</b>	0x0000 0204	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	ID_CODE Key Register Access conditions. Read: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION								HAWKEYE								TI_IDM								-							

Bits	Field Name	Description	Type	Reset
31:28	VERSION	Revision number	R	0x0
27:12	HAWKEYE	Hawkeye number	R	0xB852
11:1	TI_IDM	Manufacturer identity(TI)	R	0x17
0	-	Always set to 1	R	0x1

**Table 18-123. Register Call Summary for Register CONTROL\_ID\_CODE**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-124. CONTROL\_STD\_FUSE\_DIE\_ID\_1**

<b>Address Offset</b>	0x0000 0208	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Die ID Register - Part 1. Access conditions. Read: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	R	Single identifier

**Table 18-125. Register Call Summary for Register CONTROL\_STD\_FUSE\_DIE\_ID\_1**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-126. CONTROL\_STD\_FUSE\_DIE\_ID\_2**

<b>Address Offset</b>	0x0000 020C		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Die ID Register - Part 2. Access conditions. Read: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	R	Single identifier

**Table 18-127. Register Call Summary for Register CONTROL\_STD\_FUSE\_DIE\_ID\_2**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-128. CONTROL\_STD\_FUSE\_DIE\_ID\_3**

<b>Address Offset</b>	0x0000 0210		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Die ID Register - Part 3. Access conditions. Read: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	R	Single identifier

**Table 18-129. Register Call Summary for Register CONTROL\_STD\_FUSE\_DIE\_ID\_3**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-130. CONTROL\_STD\_FUSE\_PROD\_ID\_0**

<b>Address Offset</b>	0x0000 0214		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Prod ID Register - Part 0. This register shows the device type. Access conditions. Read: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DEVICE_TYPE															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x-
7:0	DEVICE_TYPE	Define the device type 0xF0 = GP device Other values = Reserved	R	0x-

**Table 18-131. Register Call Summary for Register CONTROL\_STD\_FUSE\_PROD\_ID\_0**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-132. CONTROL\_STD\_FUSE\_PROD\_ID\_1**

<b>Address Offset</b>	0x0000 0218	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Prod ID Register - Part 1. This register shows the device type. Access conditions. Read: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	R	0x-

**Table 18-133. Register Call Summary for Register CONTROL\_STD\_FUSE\_PROD\_ID\_1**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-134. CONTROL\_STD\_FUSE\_USB\_CONF**

<b>Address Offset</b>	0x0000 021C	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Standard Fuse conf [31:0]. Register shows part of the chip standard eFuse configuration. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USB_PROD_ID																USB_VENDOR_ID															

Bits	Field Name	Description	Type	Reset
31:16	USB_PROD_ID	USB Product Identification	R	0x0000
15:0	USB_VENDOR_ID	USB Vendor Identification	R	0x0000

**Table 18-135. Register Call Summary for Register CONTROL\_STD\_FUSE\_USB\_CONF**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-136. CONTROL\_STD\_FUSE\_OPP\_VDD\_WKUP**

<b>Address Offset</b>	0x0000 0228		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_WKUP [31:0]. Register shows part of the chip standard eFuse configuration. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_WKUP																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_WKUP		R	0x0000 0000

**Table 18-137. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_WKUP**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-138. CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_0**

<b>Address Offset</b>	0x0000 022C		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_IVA_0. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDD_IVA_OPP50_SENN_REC								RESERVED																							

Bits	Field Name	Description	Type	Reset
31:24	VDD_IVA_OPP50_SENN_REC	Voltage domain VDD_IVA OPP50 [7:0] (0.92V) from Standard Fuse.	R	0x00
23:0	RESERVED		R	0x000000

**Table 18-139. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_0**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)



**Table 18-140. CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_1**

<b>Address Offset</b>	0x0000 0230		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_IVA_1. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VDD_IVA_OPP50_SENP_GAIN		VDD_IVA_OPP50_SENN_GAIN		VDD_IVA_OPP50_SENP_REC											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	VDD_IVA_OPP50_SENP_GAIN	Voltage domain VDD_IVA OPP50 [23:20] (0.92V) from Standard Fuse.	R	0x0
11:8	VDD_IVA_OPP50_SENN_GAIN	Voltage domain VDD_IVA OPP50 [19:16] (0.92V) from Standard Fuse.	R	0x0
7:0	VDD_IVA_OPP50_SENP_REC	Voltage domain VDD_IVA OPP50 [15:8] (0.92V) from Standard Fuse.	R	0x00

**Table 18-141. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_1**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-142. CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_2**

<b>Address Offset</b>	0x0000 0234		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_IVA_2. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDD_IVA_OPP100_SENP_GAIN				VDD_IVA_OPP100_SENN_GAIN				VDD_IVA_OPP100_SENP_REC								VDD_IVA_OPP100_SENN_REC								RESERVED							

Bits	Field Name	Description	Type	Reset
31:28	VDD_IVA_OPP100_SENP_GAIN	Voltage domain VDD_IVA OPP100 [23:20] (1.1V) from Standard Fuse.	R	0x0
27:24	VDD_IVA_OPP100_SENN_GAIN	Voltage domain VDD_IVA OPP100 [19:16] (1.1V) from Standard Fuse.	R	0x0
23:16	VDD_IVA_OPP100_SENP_REC	Voltage domain VDD_IVA OPP100 [15:8] (1.1V) from Standard Fuse.	R	0x00
15:8	VDD_IVA_OPP100_SENN_REC	Voltage domain VDD_IVA OPP100 [7:0] (1.1V) from Standard Fuse.	R	0x00
7:0	RESERVED		R	0x00

**Table 18-143. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_2**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-144. CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_3**

<b>Address Offset</b>	0x0000 0238	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>		
<b>Description</b>	Standard Fuse OPP VDD_IVA_3. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				VDD_IVA_SR_SENN		VDD_IVA_SR_SENP		RESERVED																							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	VDD_IVA_SR_SENN	VDD_IVA Smart Reflex from Standard FUSE	R	0x0
25:24	VDD_IVA_SR_SENP	VDD_IVA Smart Reflex from Standard FUSE	R	0x0
23:0	RESERVED		R	0x000000

**Table 18-145. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_3**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-146. CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_4**

<b>Address Offset</b>	0x0000 023C
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a> <b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_IVA_4. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDD_IVA_OPP132_SENN_REC								RESERVED																							

Bits	Field Name	Description	Type	Reset
31:25	VDD_IVA_OPP132_SENN_REC	Voltage domain VDD_IVA OPP132 [6:0] (1.26V) from Standard Fuse.	R	0x00
24:0	RESERVED		R	0x0000000

**Table 18-147. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_4**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-148. CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_0**

<b>Address Offset</b>	0x0000 0240
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a> <b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_MPU_0. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDD_MPU_OPP50_SENN_REC								RESERVED																							

Bits	Field Name	Description	Type	Reset
31:24	VDD_MPU_OPP50_SENN_REC	Voltage domain VDD_MPU OPP50 [7:0] (0.92V) from Standard Fuse.	R	0x00
23:0	RESERVED		R	0x0000000

**Table 18-149. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_0**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-150. CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_1**

<b>Address Offset</b>	0x0000 0244	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>		
<b>Description</b>	Standard Fuse OPP VDD_MPU_1. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VDD_MPU_OPP50_SENP_GAIN				VDD_MPU_OPP50_SENN_GAIN				VDD_MPU_OPP50_SENP_REC															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	VDD_MPU_OPP50_SENP_GAIN	Voltage domain VDD_MPU OPP50 [23:20] (0.92V) from Standard Fuse.	R	0x0
11:8	VDD_MPU_OPP50_SENN_GAI N	Voltage domain VDD_MPU OPP50 [19:16] (0.92V) from Standard Fuse.	R	0x0
7:0	VDD_MPU_OPP50_SENP_REC	Voltage domain VDD_MPU OPP50 [15:8] (0.92V) from Standard Fuse.	R	0x00

**Table 18-151. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_1**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-152. CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_2**

<b>Address Offset</b>	0x0000 0248	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>		
<b>Description</b>	Standard Fuse OPP VDD_MPU_2. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDD_MPU_OPP100_SENP_GAIN				VDD_MPU_OPP100_SENN_GAIN				VDD_MPU_OPP100_SENP_REC				VDD_MPU_OPP100_SENN_REC				RESERVED															

Bits	Field Name	Description	Type	Reset
31:28	VDD_MPU_OPP100_SENP_GAIN	Voltage domain VDD_MPU OPP100 [23:20] (1.1V) from Standard Fuse.	R	0x0
27:24	VDD_MPU_OPP100_SENN_GAIN	Voltage domain VDD_MPU OPP100 [19:16] (1.1V) from Standard Fuse.	R	0x0
23:16	VDD_MPU_OPP100_SENP_REC	Voltage domain VDD_MPU OPP100 [15:8] (1.1V) from Standard Fuse.	R	0x00
15:8	VDD_MPU_OPP100_SENN_REC	Voltage domain VDD_MPU OPP100 [7:0] (1.1V) from Standard Fuse.	R	0x00
7:0	RESERVED		R	0x00

**Table 18-153. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_2**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-154. CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_3**

<b>Address Offset</b>	0x0000 024C	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>		
<b>Description</b>	Standard Fuse OPP VDD_MPU_3. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				VDD_MPU_SR_SENN		VDD_MPU_SR_SENP		RESERVED																							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	VDD_MPU_SR_SENN	VDD_MPU Smart Reflex from Standard FUSE	R	0x0
25:24	VDD_MPU_SR_SENP	VDD_MPU Smart Reflex from Standard FUSE	R	0x0
23:0	RESERVED		R	0x000000

**Table 18-155. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_3**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-156. CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_4**

<b>Address Offset</b>	0x0000 0250
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a> <b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_MPU_4. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDD_MPU_OPP132_SENN_REC																RESERVED															

Bits	Field Name	Description	Type	Reset
31:25	VDD_MPU_OPP132_SENN_REC	Voltage domain VDD_MPU OPP132 [6:0] (1.26V) from Standard Fuse.	R	0x00
24:0	RESERVED		R	0x000000

**Table 18-157. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_4**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-158. CONTROL\_STD\_FUSE\_OPP\_VDD\_CORE\_0**

<b>Address Offset</b>	0x0000 0254
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a> <b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_CORE_0. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDD_CORE_OPP50_SENN_REC								RESERVED																							

Bits	Field Name	Description	Type	Reset
31:24	VDD_CORE_OPP50_SENN_REC	Voltage domain VDD_CORE OPP50 [7:0] (0.92V) from Standard Fuse.	R	0x0
23:0	RESERVED		R	0x0

**Table 18-159. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_CORE\_0**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-160. CONTROL\_STD\_FUSE\_OPP\_VDD\_CORE\_1**

<b>Address Offset</b>	0x0000 0258	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>		
<b>Description</b>	Standard Fuse OPP VDD_CORE_1. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDD_CORE_OPP100_SENP_REC								VDD_CORE_OPP100_SENN_REC								VDD_CORE_OPP50_SENP_GAIN				VDD_CORE_OPP50_SENN_GAIN				VDD_CORE_OPP50_SENP_REC							

Bits	Field Name	Description	Type	Reset
31:24	VDD_CORE_OPP100_SENP_REC	Voltage domain VDD_CORE OPP100 [15:8] (1.1V) from Standard Fuse.	R	0x00
23:16	VDD_CORE_OPP100_SENN_REC	Voltage domain VDD_CORE OPP100 [7:0] (1.1V) from Standard Fuse.	R	0x00
15:12	VDD_CORE_OPP50_SENP_GAIN	Voltage domain VDD_CORE OPP50 [23:20] (0.92V) from Standard Fuse.	R	0x0
11:8	VDD_CORE_OPP50_SENN_GAIN	Voltage domain VDD_CORE OPP50 [19:16] (0.92V) from Standard Fuse.	R	0x0

Bits	Field Name	Description	Type	Reset
7:0	VDD_CORE_OPP50_SENP_RE C	Voltage domain VDD_CORE OPP50 [15:8] (0.92V) from Standard Fuse.	R	0x00

**Table 18-161. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_CORE\_1**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-162. CONTROL\_STD\_FUSE\_OPP\_VDD\_CORE\_2**

<b>Address Offset</b>	0x0000 025C		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_CORE_2. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							VDD_MPU_SR_SENN	VDD_MPU_SR_SENP	VDD_MPU_OPP100_SENP_GAIN	VDD_MPU_OPP100_SENN_GAIN					

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:10	VDD_CORE_SR_SENN	VDD_CORE Smart Reflex from Standard FUSE	R	0x0
9:8	VDD_CORE_SR_SENP	VDD_CORE Smart Reflex from Standard FUSE	R	0x0
7:4	VDD_CORE_OPP100_SENP_G AIN	Voltage domain VDD_CORE OPP100 [23:20] (1.1V) from Standard Fuse.	R	0x0
3:0	VDD_CORE_OPP100_SENN_G AIN	Voltage domain VDD_CORE OPP100 [19:16] (1.1V) from Standard Fuse.	R	0x0

**Table 18-163. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_CORE\_2**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)



**Table 18-164. CONTROL\_STD\_FUSE\_OPP\_BGAP**

<b>Address Offset</b>	0x0000 0260		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP BGAP. Register shows part of the chip standard eFuse configuration. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_BGAP																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_BGAP		R	0x0000 0000

**Table 18-165. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_BGAP**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-166. CONTROL\_STD\_FUSE\_OPP\_DPLL\_0**

<b>Address Offset</b>	0x0000 0264		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP DPLL. Register shows part of the chip standard eFuse configuration. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_DPLL_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_DPLL_0		R	0x0000 0000

**Table 18-167. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_DPLL\_0**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-168. CONTROL\_STD\_FUSE\_OPP\_DPLL\_1**

<b>Address Offset</b>	0x0000 0268		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP DPLL. Register shows part of the chip standard eFuse configuration. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_DPLL_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_DPLL_1		R	0x0000 0000

**Table 18-169. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_DPLL\_1**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-170. CONTROL\_STATUS**

<b>Address Offset</b>	0x0000 02C4	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>
<b>Description</b>	Control Module Status Register Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SYSCTRL_GENERAL_CONF		DEVICE_TYPE		SYS_BOOT											

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x000000
12:11	SYSCTRL_GENERAL_CONF	Sysctrl_General_IP configuration Read 0x1: SMP configuration	R	0x0
10:8	DEVICE_TYPE	Device type captured at reset time Device type value sampled at power-on reset. Read 0x3: General Purpose (GP)	R	0x0
7:0	SYS_BOOT	Sys.Boot pins state captured at reset time Sys.Boot pin values sampled at power-on reset	R	0x00

**Table 18-171. Register Call Summary for Register CONTROL\_STATUS**

Control Module Functional Description

- [Control Module Initialization: \[0\]](#)
- [General Core Control Module Instance: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[2\]](#)

**Table 18-172. CONTROL\_SEC\_ERR\_STATUS\_FUNC**

<b>Address Offset</b>	0x0000 02D0		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Firewall Error Status functional Register Access conditions. Read Only		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																												
RESERVED								C2C_INIT_FW_ERROR				L4_AUDIOBE_FW_ERROR				DEBUGSS_FW_ERROR				L4_CONFIG_FW_ERROR				L4_PERIPH_FW_ERROR				ISS_FW_ERROR				DSS_FW_ERROR				SGX_FW_ERROR				C2C_FW_ERROR				RESERVED				RESERVED				RESERVED				RESERVED				RESERVED				SL2_FW_ERROR				DUAL_CORTX_M3_FW_ERROR				IVAHD_FW_ERROR				EMIF_FW_ERROR				GPMC_FW_ERROR				L3RAM_FW_ERROR				RESERVED			

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20	C2C_INIT_FW_ERROR	C2C init firewall. MID config: unused (reserved) 0x0: No error from firewall 0x1: Error from firewall	R	0
19	L4_AUDIOBE_FW_ERROR	L4 AudioBE firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
18	DEBUGSS_FW_ERROR	DebugSS firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
17	L4_CONFIG_FW_ERROR	L4 Config firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
16	L4_PERIPH_FW_ERROR	L4 PER firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
15	ISS_FW_ERROR	ISS firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
14	DSS_FW_ERROR	DSS firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
13	SGX_FW_ERROR	SGX firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
12	C2C_FW_ERROR	C2C firewall. MID config: unused (reserved). 0x0: No error from firewall 0x1: Error from firewall	R	0
11	RESERVED	Reserved to non-GP devices only	R	0
10	RESERVED	Reserved to Non-GP devices only	R	0

Bits	Field Name	Description	Type	Reset
9	RESERVED		R	0
8	RESERVED	Reserved to Non-GP devices only	R	0
7	RESERVED	Reserved to Non-GP devices only	R	0
6	SL2_FW_ERROR	SL2 firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
5	DUAL_CORTEX_M3_FW_ERRO R	Dual Cortex M3 firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
4	IVAHD_FW_ERROR	IVAHD firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
3	EMIF_FW_ERROR	EMIF firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
2	GPMC_FW_ERROR	GPMC firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
1	L3RAM_FW_ERROR	L3RAM firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
0	RESERVED	Reserved to Non-GP devices only	R	0

**Table 18-173. Register Call Summary for Register CONTROL\_SEC\_ERR\_STATUS\_FUNC**

Control Module Functional Description

- [Protection Status Registers: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[16\]](#)

**Table 18-174. CONTROL\_SEC\_ERR\_STATUS\_DEBUG**

<b>Address Offset</b>	0x0000 02D4
<b>Physical Address</b>	See <a href="#">Table 18-113</a>
<b>Description</b>	Error Status debug Register. Read All / Write All. Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																								
RESERVED								RESERVED								RESERVED								RESERVED																																																																																															
								C2C_INIT_DBGFW_ERROR								L4_AUDIOBE_DBGFW_ERROR								DEBUGSS_DBGFW_ERROR								L4_CONFIG_DBGFW_ERROR								L4_PERIPH_DBGFW_ERROR								ISS_DBGFW_ERROR								DSS_DBGFW_ERROR								SGX_DBGFW_ERROR								C2C_DBGFW_ERROR								RESERVED								RESERVED								RESERVED								RESERVED								RESERVED							
								SL2_DBGFW_ERROR								DUAL_CORTEX_M3_DBGFW_ERROR								IVAHD_DBGFW_ERROR								EMIF_DBGFW_ERROR								GPMC_DBGFW_ERROR								L3RAM_DBGFW_ERROR								RESERVED																																																															

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20	C2C_INIT_DBGFW_ERROR	C2C init debug firewall. MID config: unused (reserved) 0x0: No error from firewall 0x1: Error from firewall	R	0
19	L4_AUDIOBE_DBGFW_ERROR	L4 AudioBE debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
18	DEBUGSS_DBGFW_ERROR	DebugSS debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
17	L4_CONFIG_DBGFW_ERROR	L4 Config debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
16	L4_PERIPH_DBGFW_ERROR	L4 PER debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
15	ISS_DBGFW_ERROR	ISS debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
14	DSS_DBGFW_ERROR	DSS debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
13	SGX_DBGFW_ERROR	SGX debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
12	C2C_DBGFW_ERROR	C2C debug firewall. MID config: unused (reserved). 0x0: No error from firewall 0x1: Error from firewall	R	0
11	RESERVED	Reserved to non-GP devices only	R	0
10	RESERVED	Reserved to Non-GP devices only	R	0
9	RESERVED		R	0
8	RESERVED	Reserved to Non-GP devices only	R	0
7	RESERVED	Reserved to Non-GP devices only	R	0
6	SL2_DBGFW_ERROR	SL2 debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
5	DUAL_CORTEX_M3_DBGFW_ERROR	Dual Cortex M3 debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
4	IVAHD_DBGFW_ERROR	IVAHD debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
3	EMIF_DBGFW_ERROR	EMIF debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
2	GPMC_DBGFW_ERROR	GPMC debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0

Bits	Field Name	Description	Type	Reset
1	L3RAM_DBGFW_ERROR	L3RAM debug firewall 0x0: No error from firewall 0x1: Error from firewall	R	0
0	RESERVED	Reserved to Non-GP devices only	R	0

**Table 18-175. Register Call Summary for Register CONTROL\_SEC\_ERR\_STATUS\_DEBUG**

Control Module Functional Description

- [Protection Status Registers: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[16\]](#)

**Table 18-176. CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_5**

<b>Address Offset</b>	0x0000 02D8		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_MPU_5. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VDD_MPU_OPP132_SENP_GAIN			VDD_MPU_OPP132_SENN_GAIN			VDD_MPU_OPP132_SENP_REC				VDD_MPU_OPP132_SENN_REC					

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:13	VDD_MPU_OPP132_SENP_GAIN	Voltage domain VDD_MPU OPP132 [23:20] (1.26V) from Standard Fuse.	R	0x0
12:9	VDD_MPU_OPP132_SENN_GAIN	Voltage domain VDD_MPU OPP132 [19:16] (1.26V) from Standard Fuse.	R	0x0
8:1	VDD_MPU_OPP132_SENP_REC	Voltage domain VDD_MPU OPP132 [15:8] (1.26V) from Standard Fuse.	R	0x00
0	VDD_MPU_OPP132_SENN_REC	Voltage domain VDD_MPU OPP132 [7] (1.26V) from Standard Fuse.	R	0x0

**Table 18-177. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_MPU\_5**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-178. CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_5**

<b>Address Offset</b>	0x0000 02DC		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Standard Fuse OPP VDD_IVA_5. Register shows part of the chip eFuse configuration on the L4 interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VDD_IVA_OPP132_SENP_GAIN			VDD_IVA_OPP132_SENN_GAIN			VDD_IVA_OPP132_SENP_REC							VDD_IVA_OPP132_SENN_REC		

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0000
16:13	VDD_IVA_OPP132_SENP_GAIN	Voltage domain VDD_IVA OPP132 [23:20] (1.26V) from Standard Fuse.	R	0x0
12:9	VDD_IVA_OPP132_SENN_GAIN	Voltage domain VDD_IVA OPP132 [19:16] (1.26V) from Standard Fuse.	R	0x0
8:1	VDD_IVA_OPP132_SENP_REC	Voltage domain VDD_IVA OPP132 [15:8] (1.26V) from Standard Fuse.	R	0x00
0	VDD_IVA_OPP132_SENN_REC	Voltage domain VDD_IVA OPP132 [7] (1.26V) from Standard Fuse.	R	0x0

**Table 18-179. Register Call Summary for Register CONTROL\_STD\_FUSE\_OPP\_VDD\_IVA\_5**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-180. CONTROL\_DEV\_CONF**

<b>Address Offset</b>	0x0000 0300		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Device configuration register. Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEV_CONF																															USBPHY_PD

Bits	Field Name	Description	Type	Reset
31:1	DEV_CONF	Spare bits for device configuration.	RW	0x0000 0000
0	USBPHY_PD	Power down entire USB phy (data, common module and UTMI). controls USB2PHYCORE.PD pin. 0x0: Normal operation 0x1: Power down the USB PHY	RW	0

**Table 18-181. Register Call Summary for Register CONTROL\_DEV\_CONF**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-182. CONTROL\_LDOVBB\_IVA\_VOLTAGE\_CTRL**

<b>Address Offset</b>	0x0000 0314	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	IVA Voltage Body Bias LDO control register Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				LDOVBBIVA_RBB_MUX_CTRL	LDOVBBIVA_RBB_VSET_IN				LDOVBBIVA_RBB_VSET_OUT				RESERVED				LDOVBBIVA_FBB_MUX_CTRL	LDOVBBIVA_FBB_VSET_IN				LDOVBBIVA_FBB_VSET_OUT									

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	LDOVBBIVA_RBB_MUX_CTRL	Override control of EFUSE Reverse Body Bias voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
25:21	LDOVBBIVA_RBB_VSET_IN	EFUSE Reverse Body Bias voltage value	R	0x00
20:16	LDOVBBIVA_RBB_VSET_OUT	Override value for Reverse Body Bias voltage	RW	0x00
15:11	RESERVED		R	0x00
10	LDOVBBIVA_FBB_MUX_CTRL	Override control of EFUSE Forward Body Bias voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
9:5	LDOVBBIVA_FBB_VSET_IN	EFUSE Forward Body Bias voltage value	R	0x00
4:0	LDOVBBIVA_FBB_VSET_OUT	Override value for Forward Body Bias voltage	RW	0x00

**Table 18-183. Register Call Summary for Register CONTROL\_LDOVBB\_IVA\_VOLTAGE\_CTRL**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)



**Table 18-184. CONTROL\_LDOVBB\_MPU\_VOLTAGE\_CTRL**

<b>Address Offset</b>	0x0000 0318	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	MPU Voltage Body Bias LDO control register Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						LDOVBBMPU_RBB_MUX_CTRL	LDOVBBMPU_RBB_VSET_IN				LDOVBBMPU_RBB_VSET_OUT				RESERVED						LDOVBBMPU_FBB_MUX_CTRL	LDOVBBMPU_FBB_VSET_IN				LDOVBBMPU_FBB_VSET_OUT					

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	LDOVBBMPU_RBB_MUX_CTRL	Override control of EFUSE Reverse Body Bias voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
25:21	LDOVBBMPU_RBB_VSET_IN	EFUSE Reverse Body Bias voltage value	R	0x00
20:16	LDOVBBMPU_RBB_VSET_OUT	Override value for Reverse Body Bias voltage	RW	0x00
15:11	RESERVED		R	0x00
10	LDOVBBMPU_FBB_MUX_CTRL	Override control of EFUSE Forward Body Bias voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
9:5	LDOVBBMPU_FBB_VSET_IN	EFUSE Forward Body Bias voltage value	R	0x00
4:0	LDOVBBMPU_FBB_VSET_OUT	Override value for Forward Body Bias voltage	RW	0x00

**Table 18-185. Register Call Summary for Register CONTROL\_LDOVBB\_MPU\_VOLTAGE\_CTRL**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-186. CONTROL\_LDOSRAM\_IVA\_VOLTAGE\_CTRL**

<b>Address Offset</b>	0x0000 0320	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	IVA SRAM LDO control register Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED								LDOSRAMIVA_RETMODE_MUX_CTRL				LDOSRAMIVA_RETMODE_VSET_IN				LDOSRAMIVA_RETMODE_VSET_OUT				RESERVED								LDOSRAMIVA_ACTMODE_MUX_CTRL				LDOSRAMIVA_ACTMODE_VSET_IN				LDOSRAMIVA_ACTMODE_VSET_OUT			

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	LDOSRAMIVA_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
25:21	LDOSRAMIVA_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x00
20:16	LDOSRAMIVA_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x00
15:11	RESERVED		R	0x00
10	LDOSRAMIVA_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
9:5	LDOSRAMIVA_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x00
4:0	LDOSRAMIVA_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x00

**Table 18-187. Register Call Summary for Register CONTROL\_LDOSRAM\_IVA\_VOLTAGE\_CTRL**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-188. CONTROL\_LDOSRAM\_MPU\_VOLTAGE\_CTRL**

<b>Address Offset</b>	0x0000 0324	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	MPU SRAM LDO control register Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LDSRAMMPU_RETMODE_MUX_CTRL	LDSRAMMPU_RETMODE_VSET_IN			LDSRAMMPU_RETMODE_VSET_OUT			RESERVED					LDSRAMMPU_ACTMODE_MUX_CTRL	LDSRAMMPU_ACTMODE_VSET_IN			LDSRAMMPU_ACTMODE_VSET_OUT										

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	LDSRAMMPU_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
25:21	LDSRAMMPU_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x00
20:16	LDSRAMMPU_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x00
15:11	RESERVED		R	0x00
10	LDSRAMMPU_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
9:5	LDSRAMMPU_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x00
4:0	LDSRAMMPU_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x00

**Table 18-189. Register Call Summary for Register CONTROL\_LDOSRAM\_MPU\_VOLTAGE\_CTRL**

Control Module Functional Description

- [General Core Control Module Instance: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-190. CONTROL\_LDOSRAM\_CORE\_VOLTAGE\_CTRL**

<b>Address Offset</b>	0x0000 0328	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Core SRAM LDO control register Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				LDOSRAMCORE_RETMODE_MUX_CTRL		LDOSRAMCORE_RETMODE_VSET_IN				LDOSRAMCORE_RETMODE_VSET_OUT				RESERVED				LDOSRAMCORE_ACTMODE_MUX_CTRL		LDOSRAMCORE_ACTMODE_VSET_IN				LDOSRAMCORE_ACTMODE_VSET_OUT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	LDOSRAMCORE_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
25:21	LDOSRAMCORE_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x00
20:16	LDOSRAMCORE_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x00
15:11	RESERVED		R	0x00
10	LDOSRAMCORE_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: efuse value is used 0x1: override value is used	RW	0
9:5	LDOSRAMCORE_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x00
4:0	LDOSRAMCORE_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x00

**Table 18-191. Register Call Summary for Register CONTROL\_LDOSRAM\_CORE\_VOLTAGE\_CTRL**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-192. CONTROL\_TEMP\_SENSOR**

<b>Address Offset</b>	0x0000 032C		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Control VBGAPTS temperature sensor and thermal comparator shutdown register Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BGAP_TEMPSOFF		BGAP_TSHUT		BGAP_TEMP_SENSOR_CONTCONV		BGAP_TEMP_SENSOR_SOC		BGAP_TEMP_SENSOR_EOCZ		BGAP_TEMP_SENSOR_DTEMP					

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	BGAP_TEMPSOFF	Temperature sensor and thermal shutdown mode. 0x0: Temperature sensor and thermal shutdown is active 0x1: Temperature sensor and thermal shutdown set in OFF mode	RW	0
11	BGAP_TSHUT	Thermal shutdown comparator output. It is low during normal operation and goes high during a thermal shutdown event.	R	0
10	BGAP_TEMP_SENSOR_CONT CONV	VDD level digital inputs. When high the ADC is in continuous conversion mode. 0x0: ADC Single Conversion Mode 0x1: ADC Continuous Conversion Mode	RW	0
9	BGAP_TEMP_SENSOR_SOC	ADC Start of Conversion. A transition to high starts a new ADC conversion cycle	RW	0
8	BGAP_TEMP_SENSOR_EOCZ	ADC End of Conversion. Active low, when CTRL_TEMP(7:0) is valid.	R	0
7:0	BGAP_TEMP_SENSOR_DTEMP	Temperature data from the ADC. Valid if EOCZ is low.	R	0x00

**Table 18-193. Register Call Summary for Register CONTROL\_TEMP\_SENSOR**

Control Module Functional Description

- [General Core Control Module Instance](#): [0]
- [Band Gap Voltage and Temperature Sensor](#): [1] [2]
- [Single Conversion Mode \(CONTCONV = 0\)](#): [3] [4] [5] [6]
- [Continuous Conversion Mode \(CONTCONV = 1\)](#): [7] [8] [9] [10]
- [ADC Codes Versus Temperature](#): [11]
- [Temperature Sensor Control Register](#): [12] [13]

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary](#): [14]

**Table 18-194. CONTROL\_DPLL\_NWELL\_TRIM\_0**

<b>Address Offset</b>	0x0000 0330		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Dpll trim (SW override) - Part 0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
RESERVED				DPLL_ABE_NWELL_TRIM_MUX_CTRL				DPLL_ABE_NWELL_TRIM				DPLL_PER_NWELL_TRIM_MUX_CTRL				DPLL_PER_NWELL_TRIM				DPLL_CORE_NWELL_TRIM_MUX_CTRL				DPLL_CORE_NWELL_TRIM				DPLL_IVA_NWELL_TRIM_MUX_CTRL				DPLL_IVA_NWELL_TRIM				DPLL_MPU_NWELL_TRIM_MUX_CTRL				DPLL_MPU_NWELL_TRIM			

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	DPLL_ABE_NWELL_TRIM_MUX_CTRL	Software override selection over efuse values. 0x0: efuse value is used 0x1: override value is used	RW	0
28:24	DPLL_ABE_NWELL_TRIM	Software override value.	RW	0x00
23	DPLL_PER_NWELL_TRIM_MUX_CTRL	Software override selection over efuse values. 0x0: efuse value is used 0x1: override value is used	RW	0
22:18	DPLL_PER_NWELL_TRIM	Software override value.	RW	0x00
17	DPLL_CORE_NWELL_TRIM_MUX_CTRL	Software override selection over efuse values. 0x0: efuse value is used 0x1: override value is used	RW	0
16:12	DPLL_CORE_NWELL_TRIM	Software override value.	RW	0x00
11	DPLL_IVA_NWELL_TRIM_MUX_CTRL	Software override selection over efuse values. 0x0: efuse value is used 0x1: override value is used	RW	0
10:6	DPLL_IVA_NWELL_TRIM	Software override value.	RW	0x00
5	DPLL_MPU_NWELL_TRIM_MUX_CTRL	Software override selection over efuse values. 0x0: efuse value is used 0x1: override value is used	RW	0
4:0	DPLL_MPU_NWELL_TRIM	Software override value.	RW	0x00

**Table 18-195. Register Call Summary for Register CONTROL\_DPLL\_NWELL\_TRIM\_0**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-196. CONTROL\_DPLL\_NWELL\_TRIM\_1**

<b>Address Offset</b>	0x0000 0334		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Dpll trim (SW override) - Part 1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED						DPLL_USB_NWELL_TRIM_MUX_CTRL	DPLL_USB_NWELL_TRIM				RESERVED				DPLL_DSI2_NWELL_TRIM_MUX_CTRL	DPLL_DSI2_NWELL_TRIM		DPLL_DSI1_NWELL_TRIM_MUX_CTRL	DPLL_DSI1_NWELL_TRIM										

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:24	RESERVED	Reserved	RW	0x00
23	DPLL_USB_NWELL_TRIM_MUX_CTRL	Software override selection over efuse values. 0x0: efuse value is used 0x1: override value is used	RW	0
22:18	DPLL_USB_NWELL_TRIM	Software override value.	RW	0x00
17:12	RESERVED	Reserved	RW	0
11	DPLL_DSI2_NWELL_TRIM_MUX_CTRL	Software override selection over efuse values. 0x0: efuse value is used 0x1: override value is used	RW	0
10:6	DPLL_DSI2_NWELL_TRIM	Software override value.	RW	0x00
5	DPLL_DSI1_NWELL_TRIM_MUX_CTRL	Software override selection over efuse values. 0x0: efuse value is used 0x1: override value is used	RW	0
4:0	DPLL_DSI1_NWELL_TRIM	Software override value.	RW	0x00

**Table 18-197. Register Call Summary for Register CONTROL\_DPLL\_NWELL\_TRIM\_1**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-198. CONTROL\_USBOTGHS\_CONTROL**

<b>Address Offset</b>	0x0000 033C		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	USBOTGHS software control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DISCHRGVBUS	CHRGVBUS	DRVVBUS	IDPULLUP	IDDIG	SESSEND	VBUSVALID	BVALID	AVALID							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0000000
8	DISCHRGVBUS	USBOTGHS value for signal dischrgvbus (OTG_PD_VBUS). controls discharging of VBUS for completing SRP.  Read 0x0: OTG transceiver does not discharge VBUS Read 0x1: OTG transceiver discharges VBUS	R	0
7	CHRGVBUS	USBOTGHS value for signal chrgvbus (OTG_PU_VBUS). controls charging of VBUS for initiating SRP.  Read 0x0: OTG transceiver does not charge VBUS Read 0x1: OTG transceiver charges VBUS	R	0
6	DRVVBUS	USBOTGHS value for signal drvvbus (OTG_DRV_VBUS). controls the driving of 5V power on VBUS.  Read 0x0: OTG transceiver does not drive VBUS Read 0x1: OTG transceiver drives VBUS	R	0
5	IDPULLUP	USBOTGHS value for signal Idpullup (OTG_PU_ID). Enables sampling of the ID pin of the USB connector.  Read 0x0: OTG transceiver does not apply a pullup to ID Read 0x1: OTG transceiver applies a pullup to ID	R	0
4	IDDIG	Sets the USBOTGHS signal iddig (ID). Indicates the value of the ID pin of the USB connector.  0x0: ID pin is grounded 0x1: ID pin is high-impedance	RW	1
3	SESSEND	Sets the USBOTGHS signal sessend (BSESSEND). Indicates if VBUS is below the B-Device session end threshold. The threshold Vth is between 0.2V and 0.8V.  0x0: VBUS voltage is below VB_SESS_END 0x1: VBUS voltage is above VB_SESS_END	RW	1
2	VBUSVALID	Sets the USBOTGHS signal vbvalid (VBUSVLD). Indicates if VBUS is above the threshold for normal operation. The threshold Vth is between 4.4V and 4.75V.	RW	0
1	BVALID	Sets the USBOTGHS signal bvalid (BSESSVLD). Signal is currently unconnected (reserved for future use).  0x0: VBUS voltage is below VB_SESS_VLD 0x1: VBUS voltage is above VB_SESS_VLD	RW	0



Bits	Field Name	Description	Type	Reset
0	AVALID	Sets the USBOTGHS signal avalid (ASESSVLD). Indicates if VBUS is above the A-Device session valid threshold. The threshold Vth is between 0.8V and 2.0V.  0x0: VBUS voltage is below VA_SESS_VLD 0x1: VBUS voltage is above VA_SESS_VLD	RW	0

**Table 18-199. Register Call Summary for Register CONTROL\_USBOTGHS\_CONTROL**

Control Module Functional Description

- [General Core Control Module Instance: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-200. CONTROL\_DSS\_CONTROL**

<b>Address Offset</b>	0x0000 0340		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	DSS software control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																																DSS_MUX6_SELECT

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	DSS_MUX6_SELECT	Mux6 select value for DSS.  0x0: enables DSS_CLK as functional clock input for the DSI2 PHY bypass clock 0x1: enables PLL1_CLK4 as functional clock input for the DSI2 PHY bypass clock	RW	0

**Table 18-201. Register Call Summary for Register CONTROL\_DSS\_CONTROL**

Control Module Functional Description

- [General Core Control Module Instance: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-202. CONTROL\_CORTEX\_M3\_MMUADDRTRANSLTR**

<b>Address Offset</b>	0x0000 0348		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	CORTEX_M3 reg Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CORTEX_M3_MMUADDRTRANSLTR																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	CORTEX_M3_MMUADDRTRANSLTR	Used to save the mmu address boot SLTR	RW	0x00000

**Table 18-203. Register Call Summary for Register CONTROL\_CORTEX\_M3\_MMUADDRTRANSLTR**

Control Module Functional Description

- [General Core Control Module Instance: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-204. CONTROL\_CORTEX\_M3\_MMUADDRLOGICTR**

<b>Address Offset</b>	0x0000 034C	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-113</a>		
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CORTEX_M3_MMUADDRLOGICTR																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:0	CORTEX_M3_MMUADDRLOGICTR		RW	0x00000

**Table 18-205. Register Call Summary for Register CONTROL\_CORTEX\_M3\_MMUADDRLOGICTR**

Control Module Functional Description

- [General Core Control Module Instance: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-206. CONTROL\_HWOBS\_CONTROL**

<b>Address Offset</b>	0x0000 0350	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Hardware observability control. This register enables or disables hardware observability outputs (to save power primarily) Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HWOBS_CLKOBS_GATE_ENABLE	HWOBS_CLKDIV_SEL				HWOBS_ALL_ZERO_MODE	HWOBS_ALL_ONE_MODE	HWOBS_MACRO_ENABLE								

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	HWOBS_CLKOBS_GATE_ENABLE	Used to gate clk observable signals at the output of the DPLLs in order to save power	RW	0x0
7:3	HWOBS_CLKDIV_SEL	Clock divider selection on po_hwoobs(0). 0x1: output is not divided 0x2: output is divided by 2 0x4: output is divided by 4 0x8: output is divided by 8 0x16: output is divided by 16 <sup>(1)</sup>	RW	0x00
2	HWOBS_ALL_ZERO_MODE	Used to gate observable signals. When set all outputs are set to zero (can be used to check the path from hardware observability to external pads). 0x0: Hardware observability ports are not gated 0x1: Hardware observability ports are all set to 0	RW	0
1	HWOBS_ALL_ONE_MODE	Used to gate observable signals. When set all outputs are set to one (can be used to check the path from hardware observability to external pads). 0x0: Hardware observability ports are not gated 0x1: Hardware observability ports are all set to 1	RW	0
0	HWOBS_MACRO_ENABLE	Used to gate observable signals coming from macros using the 32-bit HWOBS bus definition. When deasserted all outputs of the HWOBS busdef are set to zero. 0x0: Hardware observability ports from macros are gated and set to zero 0x1: Hardware observability ports from macros are not gated	RW	0

<sup>(1)</sup> The values different than 1, 2, 4, 8, and 16 are reserved.

**Table 18-207. Register Call Summary for Register CONTROL\_HWOBS\_CONTROL**

Control Module Functional Description

- [General Core Control Module Instance: \[0\]](#)
- [Observability Gating Capabilities: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[10\] \[11\] \[12\] \[13\] \[14\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[15\]](#)

**Table 18-208. CONTROL\_DEBOBS\_FINAL\_MUX\_SEL**

<b>Address Offset</b>	0x0000 0400		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Final mux select signal. It selects between core and wkup signal (controls external observability logic). Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELECT																															

Bits	Field Name	Description	Type	Reset
31:0	SELECT	The i-th (i=0..31) bit from this bit field controls the multiplexing between the core and the wake-up signals to be observed at i-th hw_dbg line. Ob0: Selects a signal from the WAKEUP domain Ob1: Select a signal from the CORE domain	RW	0x0000 0000

**Table 18-209. Register Call Summary for Register CONTROL\_DEBOBS\_FINAL\_MUX\_SEL**

Control Module Functional Description

- [General Core Control Module Instance: \[0\]](#)
- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[1\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[3\]](#)

**Table 18-210. CONTROL\_DEBOBS\_MMR\_MPU**

<b>Address Offset</b>	0x0000 0408		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Dual Cortex-A9 register to control hardware observability muxing inside dual Cortex-A9 (controls external observability logic).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SELECT															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved. Read returns reset value.	R	0x0000 0000
3:0	SELECT	Control external observability logic	RW	0x0

**Table 18-211. Register Call Summary for Register CONTROL\_DEBOBS\_MMR\_MPU**

Control Module Functional Description

- [Dual Cortex-A9 Subsystem Observable Signals: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[9\]](#)

**Table 18-212. CONTROL\_CONF\_SDMA\_REQ\_SEL0**

<b>Address Offset</b>	0x0000 042C		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	System DMA requests view channel 0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6:0	MULT	Select one of the following signals: 0x0: hwobs_sdma_dma_req_0 0x1: hwobs_sdma_dma_req_1 0x2: hwobs_sdma_dma_req_2 0x3: hwobs_sdma_dma_req_3 0x4: hwobs_sdma_dma_req_4 0x5: hwobs_sdma_dma_req_5 0x6: hwobs_sdma_dma_req_6 0x7: hwobs_sdma_dma_req_7 0x8: hwobs_sdma_dma_req_8 0x9: hwobs_sdma_dma_req_9 0xA: hwobs_sdma_dma_req_10 0xB: hwobs_sdma_dma_req_11 0xC: hwobs_sdma_dma_req_12 0xD: hwobs_sdma_dma_req_13 0xE: hwobs_sdma_dma_req_14 0xF: hwobs_sdma_dma_req_15 0x10: hwobs_sdma_dma_req_16 0x11: hwobs_sdma_dma_req_17 0x12: hwobs_sdma_dma_req_18 0x13: hwobs_sdma_dma_req_19 0x14: hwobs_sdma_dma_req_20 0x15: hwobs_sdma_dma_req_21 0x16: hwobs_sdma_dma_req_22 0x17: hwobs_sdma_dma_req_23 0x18: hwobs_sdma_dma_req_24 0x19: hwobs_sdma_dma_req_25 0x1A: hwobs_sdma_dma_req_26 0x1B: hwobs_sdma_dma_req_27 0x1C: hwobs_sdma_dma_req_28 0x1D: hwobs_sdma_dma_req_29 0x1E: hwobs_sdma_dma_req_30 0x1F: hwobs_sdma_dma_req_31 0x20: hwobs_sdma_dma_req_32 0x21: hwobs_sdma_dma_req_33 0x22: hwobs_sdma_dma_req_34 0x23: hwobs_sdma_dma_req_35 0x24: hwobs_sdma_dma_req_36	RW	0x00

Bits	Field Name	Description	Type	Reset
		0x25: hwobs_sdma_dma_req_37		
		0x26: hwobs_sdma_dma_req_38		
		0x27: hwobs_sdma_dma_req_39		
		0x28: hwobs_sdma_dma_req_40		
		0x29: hwobs_sdma_dma_req_41		
		0x2A: hwobs_sdma_dma_req_42		
		0x2B: hwobs_sdma_dma_req_43		
		0x2C: hwobs_sdma_dma_req_44		
		0x2D: hwobs_sdma_dma_req_45		
		0x2E: hwobs_sdma_dma_req_46		
		0x2F: hwobs_sdma_dma_req_47		
		0x30: hwobs_sdma_dma_req_48		
		0x31: hwobs_sdma_dma_req_49		
		0x32: hwobs_sdma_dma_req_50		
		0x33: hwobs_sdma_dma_req_51		
		0x34: hwobs_sdma_dma_req_52		
		0x35: hwobs_sdma_dma_req_53		
		0x36: hwobs_sdma_dma_req_54		
		0x37: hwobs_sdma_dma_req_55		
		0x38: hwobs_sdma_dma_req_56		
		0x39: hwobs_sdma_dma_req_57		
		0x3A: hwobs_sdma_dma_req_58		
		0x3B: hwobs_sdma_dma_req_59		
		0x3C: hwobs_sdma_dma_req_60		
		0x3D: hwobs_sdma_dma_req_61		
		0x3E: hwobs_sdma_dma_req_62		
		0x3F: hwobs_sdma_dma_req_63		
		0x40: hwobs_sdma_dma_req_64		
		0x41: hwobs_sdma_dma_req_65		
		0x42: hwobs_sdma_dma_req_66		
		0x43: hwobs_sdma_dma_req_67		
		0x44: hwobs_sdma_dma_req_68		
		0x45: hwobs_sdma_dma_req_69		
		0x46: hwobs_sdma_dma_req_70		
		0x47: hwobs_sdma_dma_req_71		
		0x48: hwobs_sdma_dma_req_72		
		0x49: hwobs_sdma_dma_req_73		
		0x4A: hwobs_sdma_dma_req_74		
		0x4B: hwobs_sdma_dma_req_75		
		0x4C: hwobs_sdma_dma_req_76		
		0x4D: hwobs_sdma_dma_req_77		
		0x4E: hwobs_sdma_dma_req_78		
		0x4F: hwobs_sdma_dma_req_79		
		0x50: hwobs_sdma_dma_req_80		
		0x51: hwobs_sdma_dma_req_81		
		0x52: hwobs_sdma_dma_req_82		
		0x53: hwobs_sdma_dma_req_83		
		0x54: hwobs_sdma_dma_req_84		
		0x55: hwobs_sdma_dma_req_85		

Bits	Field Name	Description	Type	Reset
		0x56: hwobs_sdma_dma_req_86		
		0x57: hwobs_sdma_dma_req_87		
		0x58: hwobs_sdma_dma_req_88		
		0x59: hwobs_sdma_dma_req_89		
		0x5A: hwobs_sdma_dma_req_90		
		0x5B: hwobs_sdma_dma_req_91		
		0x5C: hwobs_sdma_dma_req_92		
		0x5D: hwobs_sdma_dma_req_93		
		0x5E: hwobs_sdma_dma_req_94		
		0x5F: hwobs_sdma_dma_req_95		
		0x60: hwobs_sdma_dma_req_96		
		0x61: hwobs_sdma_dma_req_97		
		0x62: hwobs_sdma_dma_req_98		
		0x63: hwobs_sdma_dma_req_99		
		0x64: hwobs_sdma_dma_req_100		
		0x65: hwobs_sdma_dma_req_101		
		0x66: hwobs_sdma_dma_req_102		
		0x67: hwobs_sdma_dma_req_103		
		0x68: hwobs_sdma_dma_req_104		
		0x69: hwobs_sdma_dma_req_105		
		0x6A: hwobs_sdma_dma_req_106		
		0x6B: hwobs_sdma_dma_req_107		
		0x6C: hwobs_sdma_dma_req_108		
		0x6D: hwobs_sdma_dma_req_109		
		0x6E: hwobs_sdma_dma_req_110		
		0x6F: hwobs_sdma_dma_req_111		
		0x70: hwobs_sdma_dma_req_112		
		0x71: hwobs_sdma_dma_req_113		
		0x72: hwobs_sdma_dma_req_114		
		0x73: hwobs_sdma_dma_req_115		
		0x74: hwobs_sdma_dma_req_116		
		0x75: hwobs_sdma_dma_req_117		
		0x76: hwobs_sdma_dma_req_118		
		0x77: hwobs_sdma_dma_req_119		
		0x78: hwobs_sdma_dma_req_120		
		0x79: hwobs_sdma_dma_req_121		
		0x7A: hwobs_sdma_dma_req_122		
		0x7B: hwobs_sdma_dma_req_123		
		0x7C: hwobs_sdma_dma_req_124		
		0x7D: hwobs_sdma_dma_req_125		
		0x7E: hwobs_sdma_dma_req_126		

**Table 18-213. Register Call Summary for Register CONTROL\_CONF\_SDMA\_REQ\_SEL0**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[2\]](#)

**Table 18-214. CONTROL\_CONF\_SDMA\_REQ\_SEL1**

<b>Address Offset</b>	0x0000 0430		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	System DMA requests view channel 1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6:0	MULT	Select one of the following signals: 0x0: hwobs_sdma_dma_req_0 0x1: hwobs_sdma_dma_req_1 0x2: hwobs_sdma_dma_req_2 0x3: hwobs_sdma_dma_req_3 0x4: hwobs_sdma_dma_req_4 0x5: hwobs_sdma_dma_req_5 0x6: hwobs_sdma_dma_req_6 0x7: hwobs_sdma_dma_req_7 0x8: hwobs_sdma_dma_req_8 0x9: hwobs_sdma_dma_req_9 0xA: hwobs_sdma_dma_req_10 0xB: hwobs_sdma_dma_req_11 0xC: hwobs_sdma_dma_req_12 0xD: hwobs_sdma_dma_req_13 0xE: hwobs_sdma_dma_req_14 0xF: hwobs_sdma_dma_req_15 0x10: hwobs_sdma_dma_req_16 0x11: hwobs_sdma_dma_req_17 0x12: hwobs_sdma_dma_req_18 0x13: hwobs_sdma_dma_req_19 0x14: hwobs_sdma_dma_req_20 0x15: hwobs_sdma_dma_req_21 0x16: hwobs_sdma_dma_req_22 0x17: hwobs_sdma_dma_req_23 0x18: hwobs_sdma_dma_req_24 0x19: hwobs_sdma_dma_req_25 0x1A: hwobs_sdma_dma_req_26 0x1B: hwobs_sdma_dma_req_27 0x1C: hwobs_sdma_dma_req_28 0x1D: hwobs_sdma_dma_req_29 0x1E: hwobs_sdma_dma_req_30 0x1F: hwobs_sdma_dma_req_31 0x20: hwobs_sdma_dma_req_32 0x21: hwobs_sdma_dma_req_33 0x22: hwobs_sdma_dma_req_34 0x23: hwobs_sdma_dma_req_35 0x24: hwobs_sdma_dma_req_36	RW	0x01



<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
		0x25: hwoobs_sdma_dma_req_37		
		0x26: hwoobs_sdma_dma_req_38		
		0x27: hwoobs_sdma_dma_req_39		
		0x28: hwoobs_sdma_dma_req_40		
		0x29: hwoobs_sdma_dma_req_41		
		0x2A: hwoobs_sdma_dma_req_42		
		0x2B: hwoobs_sdma_dma_req_43		
		0x2C: hwoobs_sdma_dma_req_44		
		0x2D: hwoobs_sdma_dma_req_45		
		0x2E: hwoobs_sdma_dma_req_46		
		0x2F: hwoobs_sdma_dma_req_47		
		0x30: hwoobs_sdma_dma_req_48		
		0x31: hwoobs_sdma_dma_req_49		
		0x32: hwoobs_sdma_dma_req_50		
		0x33: hwoobs_sdma_dma_req_51		
		0x34: hwoobs_sdma_dma_req_52		
		0x35: hwoobs_sdma_dma_req_53		
		0x36: hwoobs_sdma_dma_req_54		
		0x37: hwoobs_sdma_dma_req_55		
		0x38: hwoobs_sdma_dma_req_56		
		0x39: hwoobs_sdma_dma_req_57		
		0x3A: hwoobs_sdma_dma_req_58		
		0x3B: hwoobs_sdma_dma_req_59		
		0x3C: hwoobs_sdma_dma_req_60		
		0x3D: hwoobs_sdma_dma_req_61		
		0x3E: hwoobs_sdma_dma_req_62		
		0x3F: hwoobs_sdma_dma_req_63		
		0x40: hwoobs_sdma_dma_req_64		
		0x41: hwoobs_sdma_dma_req_65		
		0x42: hwoobs_sdma_dma_req_66		
		0x43: hwoobs_sdma_dma_req_67		
		0x44: hwoobs_sdma_dma_req_68		
		0x45: hwoobs_sdma_dma_req_69		
		0x46: hwoobs_sdma_dma_req_70		
		0x47: hwoobs_sdma_dma_req_71		
		0x48: hwoobs_sdma_dma_req_72		
		0x49: hwoobs_sdma_dma_req_73		
		0x4A: hwoobs_sdma_dma_req_74		
		0x4B: hwoobs_sdma_dma_req_75		
		0x4C: hwoobs_sdma_dma_req_76		
		0x4D: hwoobs_sdma_dma_req_77		
		0x4E: hwoobs_sdma_dma_req_78		
		0x4F: hwoobs_sdma_dma_req_79		
		0x50: hwoobs_sdma_dma_req_80		
		0x51: hwoobs_sdma_dma_req_81		
		0x52: hwoobs_sdma_dma_req_82		
		0x53: hwoobs_sdma_dma_req_83		
		0x54: hwoobs_sdma_dma_req_84		
		0x55: hwoobs_sdma_dma_req_85		

Bits	Field Name	Description	Type	Reset
		0x56: hwobs_sdma_dma_req_86		
		0x57: hwobs_sdma_dma_req_87		
		0x58: hwobs_sdma_dma_req_88		
		0x59: hwobs_sdma_dma_req_89		
		0x5A: hwobs_sdma_dma_req_90		
		0x5B: hwobs_sdma_dma_req_91		
		0x5C: hwobs_sdma_dma_req_92		
		0x5D: hwobs_sdma_dma_req_93		
		0x5E: hwobs_sdma_dma_req_94		
		0x5F: hwobs_sdma_dma_req_95		
		0x60: hwobs_sdma_dma_req_96		
		0x61: hwobs_sdma_dma_req_97		
		0x62: hwobs_sdma_dma_req_98		
		0x63: hwobs_sdma_dma_req_99		
		0x64: hwobs_sdma_dma_req_100		
		0x65: hwobs_sdma_dma_req_101		
		0x66: hwobs_sdma_dma_req_102		
		0x67: hwobs_sdma_dma_req_103		
		0x68: hwobs_sdma_dma_req_104		
		0x69: hwobs_sdma_dma_req_105		
		0x6A: hwobs_sdma_dma_req_106		
		0x6B: hwobs_sdma_dma_req_107		
		0x6C: hwobs_sdma_dma_req_108		
		0x6D: hwobs_sdma_dma_req_109		
		0x6E: hwobs_sdma_dma_req_110		
		0x6F: hwobs_sdma_dma_req_111		
		0x70: hwobs_sdma_dma_req_112		
		0x71: hwobs_sdma_dma_req_113		
		0x72: hwobs_sdma_dma_req_114		
		0x73: hwobs_sdma_dma_req_115		
		0x74: hwobs_sdma_dma_req_116		
		0x75: hwobs_sdma_dma_req_117		
		0x76: hwobs_sdma_dma_req_118		
		0x77: hwobs_sdma_dma_req_119		
		0x78: hwobs_sdma_dma_req_120		
		0x79: hwobs_sdma_dma_req_121		
		0x7A: hwobs_sdma_dma_req_122		
		0x7B: hwobs_sdma_dma_req_123		
		0x7C: hwobs_sdma_dma_req_124		
		0x7D: hwobs_sdma_dma_req_125		
		0x7E: hwobs_sdma_dma_req_126		

**Table 18-215. Register Call Summary for Register CONTROL\_CONF\_SDMA\_REQ\_SEL1**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-216. CONTROL\_CONF\_SDMA\_REQ\_SEL2**

<b>Address Offset</b>	0x0000 0434		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	System DMA requests view channel 2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6:0	MULT	Select one of the following signals: 0x0: hwobs_sdma_dma_req_0 0x1: hwobs_sdma_dma_req_1 0x2: hwobs_sdma_dma_req_2 0x3: hwobs_sdma_dma_req_3 0x4: hwobs_sdma_dma_req_4 0x5: hwobs_sdma_dma_req_5 0x6: hwobs_sdma_dma_req_6 0x7: hwobs_sdma_dma_req_7 0x8: hwobs_sdma_dma_req_8 0x9: hwobs_sdma_dma_req_9 0xA: hwobs_sdma_dma_req_10 0xB: hwobs_sdma_dma_req_11 0xC: hwobs_sdma_dma_req_12 0xD: hwobs_sdma_dma_req_13 0xE: hwobs_sdma_dma_req_14 0xF: hwobs_sdma_dma_req_15 0x10: hwobs_sdma_dma_req_16 0x11: hwobs_sdma_dma_req_17 0x12: hwobs_sdma_dma_req_18 0x13: hwobs_sdma_dma_req_19 0x14: hwobs_sdma_dma_req_20 0x15: hwobs_sdma_dma_req_21 0x16: hwobs_sdma_dma_req_22 0x17: hwobs_sdma_dma_req_23 0x18: hwobs_sdma_dma_req_24 0x19: hwobs_sdma_dma_req_25 0x1A: hwobs_sdma_dma_req_26 0x1B: hwobs_sdma_dma_req_27 0x1C: hwobs_sdma_dma_req_28 0x1D: hwobs_sdma_dma_req_29 0x1E: hwobs_sdma_dma_req_30 0x1F: hwobs_sdma_dma_req_31 0x20: hwobs_sdma_dma_req_32 0x21: hwobs_sdma_dma_req_33 0x22: hwobs_sdma_dma_req_34 0x23: hwobs_sdma_dma_req_35 0x24: hwobs_sdma_dma_req_36	RW	0x02

Bits	Field Name	Description	Type	Reset
		0x25: hwobs_sdma_dma_req_37		
		0x26: hwobs_sdma_dma_req_38		
		0x27: hwobs_sdma_dma_req_39		
		0x28: hwobs_sdma_dma_req_40		
		0x29: hwobs_sdma_dma_req_41		
		0x2A: hwobs_sdma_dma_req_42		
		0x2B: hwobs_sdma_dma_req_43		
		0x2C: hwobs_sdma_dma_req_44		
		0x2D: hwobs_sdma_dma_req_45		
		0x2E: hwobs_sdma_dma_req_46		
		0x2F: hwobs_sdma_dma_req_47		
		0x30: hwobs_sdma_dma_req_48		
		0x31: hwobs_sdma_dma_req_49		
		0x32: hwobs_sdma_dma_req_50		
		0x33: hwobs_sdma_dma_req_51		
		0x34: hwobs_sdma_dma_req_52		
		0x35: hwobs_sdma_dma_req_53		
		0x36: hwobs_sdma_dma_req_54		
		0x37: hwobs_sdma_dma_req_55		
		0x38: hwobs_sdma_dma_req_56		
		0x39: hwobs_sdma_dma_req_57		
		0x3A: hwobs_sdma_dma_req_58		
		0x3B: hwobs_sdma_dma_req_59		
		0x3C: hwobs_sdma_dma_req_60		
		0x3D: hwobs_sdma_dma_req_61		
		0x3E: hwobs_sdma_dma_req_62		
		0x3F: hwobs_sdma_dma_req_63		
		0x40: hwobs_sdma_dma_req_64		
		0x41: hwobs_sdma_dma_req_65		
		0x42: hwobs_sdma_dma_req_66		
		0x43: hwobs_sdma_dma_req_67		
		0x44: hwobs_sdma_dma_req_68		
		0x45: hwobs_sdma_dma_req_69		
		0x46: hwobs_sdma_dma_req_70		
		0x47: hwobs_sdma_dma_req_71		
		0x48: hwobs_sdma_dma_req_72		
		0x49: hwobs_sdma_dma_req_73		
		0x4A: hwobs_sdma_dma_req_74		
		0x4B: hwobs_sdma_dma_req_75		
		0x4C: hwobs_sdma_dma_req_76		
		0x4D: hwobs_sdma_dma_req_77		
		0x4E: hwobs_sdma_dma_req_78		
		0x4F: hwobs_sdma_dma_req_79		
		0x50: hwobs_sdma_dma_req_80		
		0x51: hwobs_sdma_dma_req_81		
		0x52: hwobs_sdma_dma_req_82		
		0x53: hwobs_sdma_dma_req_83		
		0x54: hwobs_sdma_dma_req_84		
		0x55: hwobs_sdma_dma_req_85		

Bits	Field Name	Description	Type	Reset
		0x56: hwoobs_sdma_dma_req_86		
		0x57: hwoobs_sdma_dma_req_87		
		0x58: hwoobs_sdma_dma_req_88		
		0x59: hwoobs_sdma_dma_req_89		
		0x5A: hwoobs_sdma_dma_req_90		
		0x5B: hwoobs_sdma_dma_req_91		
		0x5C: hwoobs_sdma_dma_req_92		
		0x5D: hwoobs_sdma_dma_req_93		
		0x5E: hwoobs_sdma_dma_req_94		
		0x5F: hwoobs_sdma_dma_req_95		
		0x60: hwoobs_sdma_dma_req_96		
		0x61: hwoobs_sdma_dma_req_97		
		0x62: hwoobs_sdma_dma_req_98		
		0x63: hwoobs_sdma_dma_req_99		
		0x64: hwoobs_sdma_dma_req_100		
		0x65: hwoobs_sdma_dma_req_101		
		0x66: hwoobs_sdma_dma_req_102		
		0x67: hwoobs_sdma_dma_req_103		
		0x68: hwoobs_sdma_dma_req_104		
		0x69: hwoobs_sdma_dma_req_105		
		0x6A: hwoobs_sdma_dma_req_106		
		0x6B: hwoobs_sdma_dma_req_107		
		0x6C: hwoobs_sdma_dma_req_108		
		0x6D: hwoobs_sdma_dma_req_109		
		0x6E: hwoobs_sdma_dma_req_110		
		0x6F: hwoobs_sdma_dma_req_111		
		0x70: hwoobs_sdma_dma_req_112		
		0x71: hwoobs_sdma_dma_req_113		
		0x72: hwoobs_sdma_dma_req_114		
		0x73: hwoobs_sdma_dma_req_115		
		0x74: hwoobs_sdma_dma_req_116		
		0x75: hwoobs_sdma_dma_req_117		
		0x76: hwoobs_sdma_dma_req_118		
		0x77: hwoobs_sdma_dma_req_119		
		0x78: hwoobs_sdma_dma_req_120		
		0x79: hwoobs_sdma_dma_req_121		
		0x7A: hwoobs_sdma_dma_req_122		
		0x7B: hwoobs_sdma_dma_req_123		
		0x7C: hwoobs_sdma_dma_req_124		
		0x7D: hwoobs_sdma_dma_req_125		
		0x7E: hwoobs_sdma_dma_req_126		

**Table 18-217. Register Call Summary for Register CONTROL\_CONF\_SDMA\_REQ\_SEL2**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-218. CONTROL\_CONF\_SDMA\_REQ\_SEL3**

<b>Address Offset</b>	0x0000 0438		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	System DMA requests view channel 3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6:0	MULT	Select one of the following signals: 0x0: hwobs_sdma_dma_req_0 0x1: hwobs_sdma_dma_req_1 0x2: hwobs_sdma_dma_req_2 0x3: hwobs_sdma_dma_req_3 0x4: hwobs_sdma_dma_req_4 0x5: hwobs_sdma_dma_req_5 0x6: hwobs_sdma_dma_req_6 0x7: hwobs_sdma_dma_req_7 0x8: hwobs_sdma_dma_req_8 0x9: hwobs_sdma_dma_req_9 0xA: hwobs_sdma_dma_req_10 0xB: hwobs_sdma_dma_req_11 0xC: hwobs_sdma_dma_req_12 0xD: hwobs_sdma_dma_req_13 0xE: hwobs_sdma_dma_req_14 0xF: hwobs_sdma_dma_req_15 0x10: hwobs_sdma_dma_req_16 0x11: hwobs_sdma_dma_req_17 0x12: hwobs_sdma_dma_req_18 0x13: hwobs_sdma_dma_req_19 0x14: hwobs_sdma_dma_req_20 0x15: hwobs_sdma_dma_req_21 0x16: hwobs_sdma_dma_req_22 0x17: hwobs_sdma_dma_req_23 0x18: hwobs_sdma_dma_req_24 0x19: hwobs_sdma_dma_req_25 0x1A: hwobs_sdma_dma_req_26 0x1B: hwobs_sdma_dma_req_27 0x1C: hwobs_sdma_dma_req_28 0x1D: hwobs_sdma_dma_req_29 0x1E: hwobs_sdma_dma_req_30 0x1F: hwobs_sdma_dma_req_31 0x20: hwobs_sdma_dma_req_32 0x21: hwobs_sdma_dma_req_33 0x22: hwobs_sdma_dma_req_34 0x23: hwobs_sdma_dma_req_35 0x24: hwobs_sdma_dma_req_36	RW	0x03

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
		0x25: hwoobs_sdma_dma_req_37		
		0x26: hwoobs_sdma_dma_req_38		
		0x27: hwoobs_sdma_dma_req_39		
		0x28: hwoobs_sdma_dma_req_40		
		0x29: hwoobs_sdma_dma_req_41		
		0x2A: hwoobs_sdma_dma_req_42		
		0x2B: hwoobs_sdma_dma_req_43		
		0x2C: hwoobs_sdma_dma_req_44		
		0x2D: hwoobs_sdma_dma_req_45		
		0x2E: hwoobs_sdma_dma_req_46		
		0x2F: hwoobs_sdma_dma_req_47		
		0x30: hwoobs_sdma_dma_req_48		
		0x31: hwoobs_sdma_dma_req_49		
		0x32: hwoobs_sdma_dma_req_50		
		0x33: hwoobs_sdma_dma_req_51		
		0x34: hwoobs_sdma_dma_req_52		
		0x35: hwoobs_sdma_dma_req_53		
		0x36: hwoobs_sdma_dma_req_54		
		0x37: hwoobs_sdma_dma_req_55		
		0x38: hwoobs_sdma_dma_req_56		
		0x39: hwoobs_sdma_dma_req_57		
		0x3A: hwoobs_sdma_dma_req_58		
		0x3B: hwoobs_sdma_dma_req_59		
		0x3C: hwoobs_sdma_dma_req_60		
		0x3D: hwoobs_sdma_dma_req_61		
		0x3E: hwoobs_sdma_dma_req_62		
		0x3F: hwoobs_sdma_dma_req_63		
		0x40: hwoobs_sdma_dma_req_64		
		0x41: hwoobs_sdma_dma_req_65		
		0x42: hwoobs_sdma_dma_req_66		
		0x43: hwoobs_sdma_dma_req_67		
		0x44: hwoobs_sdma_dma_req_68		
		0x45: hwoobs_sdma_dma_req_69		
		0x46: hwoobs_sdma_dma_req_70		
		0x47: hwoobs_sdma_dma_req_71		
		0x48: hwoobs_sdma_dma_req_72		
		0x49: hwoobs_sdma_dma_req_73		
		0x4A: hwoobs_sdma_dma_req_74		
		0x4B: hwoobs_sdma_dma_req_75		
		0x4C: hwoobs_sdma_dma_req_76		
		0x4D: hwoobs_sdma_dma_req_77		
		0x4E: hwoobs_sdma_dma_req_78		
		0x4F: hwoobs_sdma_dma_req_79		
		0x50: hwoobs_sdma_dma_req_80		
		0x51: hwoobs_sdma_dma_req_81		
		0x52: hwoobs_sdma_dma_req_82		
		0x53: hwoobs_sdma_dma_req_83		
		0x54: hwoobs_sdma_dma_req_84		
		0x55: hwoobs_sdma_dma_req_85		

Bits	Field Name	Description	Type	Reset
		0x56: hwobs_sdma_dma_req_86		
		0x57: hwobs_sdma_dma_req_87		
		0x58: hwobs_sdma_dma_req_88		
		0x59: hwobs_sdma_dma_req_89		
		0x5A: hwobs_sdma_dma_req_90		
		0x5B: hwobs_sdma_dma_req_91		
		0x5C: hwobs_sdma_dma_req_92		
		0x5D: hwobs_sdma_dma_req_93		
		0x5E: hwobs_sdma_dma_req_94		
		0x5F: hwobs_sdma_dma_req_95		
		0x60: hwobs_sdma_dma_req_96		
		0x61: hwobs_sdma_dma_req_97		
		0x62: hwobs_sdma_dma_req_98		
		0x63: hwobs_sdma_dma_req_99		
		0x64: hwobs_sdma_dma_req_100		
		0x65: hwobs_sdma_dma_req_101		
		0x66: hwobs_sdma_dma_req_102		
		0x67: hwobs_sdma_dma_req_103		
		0x68: hwobs_sdma_dma_req_104		
		0x69: hwobs_sdma_dma_req_105		
		0x6A: hwobs_sdma_dma_req_106		
		0x6B: hwobs_sdma_dma_req_107		
		0x6C: hwobs_sdma_dma_req_108		
		0x6D: hwobs_sdma_dma_req_109		
		0x6E: hwobs_sdma_dma_req_110		
		0x6F: hwobs_sdma_dma_req_111		
		0x70: hwobs_sdma_dma_req_112		
		0x71: hwobs_sdma_dma_req_113		
		0x72: hwobs_sdma_dma_req_114		
		0x73: hwobs_sdma_dma_req_115		
		0x74: hwobs_sdma_dma_req_116		
		0x75: hwobs_sdma_dma_req_117		
		0x76: hwobs_sdma_dma_req_118		
		0x77: hwobs_sdma_dma_req_119		
		0x78: hwobs_sdma_dma_req_120		
		0x79: hwobs_sdma_dma_req_121		
		0x7A: hwobs_sdma_dma_req_122		
		0x7B: hwobs_sdma_dma_req_123		
		0x7C: hwobs_sdma_dma_req_124		
		0x7D: hwobs_sdma_dma_req_125		
		0x7E: hwobs_sdma_dma_req_126		

**Table 18-219. Register Call Summary for Register CONTROL\_CONF\_SDMA\_REQ\_SEL3**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[2\]](#)



**Table 18-220. CONTROL\_CONF\_CLK\_SEL0**

<b>Address Offset</b>	0x0000 0440	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	clk view channel 0 Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_clkout 0x1: hwobs_abedpll_clkout 0x2: hwobs_perdpll_clkout 0x3: hwobs_dsi1dpll_clkout 0x4: hwobs_dsi2dpll_clkout 0x5: reserved 0x6: hwobs_usbdpll_clkout 0x7: reserved	RW	0x0

**Table 18-221. Register Call Summary for Register CONTROL\_CONF\_CLK\_SEL0**

Control Module Programming Guide

- [Hardware Observability Settings: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-222. CONTROL\_CONF\_CLK\_SEL1**

<b>Address Offset</b>	0x0000 0444	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	clk view channel 1 Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_clkout 0x1: hwobs_abedpll_clkout 0x2: hwobs_perdpll_clkout 0x3: hwobs_dsi1dpll_clkout 0x4: hwobs_dsi2dpll_clkout 0x5: reserved 0x6: hwobs_usbdpll_clkout 0x7: reserved	RW	0x1

**Table 18-223. Register Call Summary for Register CONTROL\_CONF\_CLK\_SEL1**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

**Table 18-224. CONTROL\_CONF\_CLK\_SEL2**

<b>Address Offset</b>	0x0000 0448	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	clk view channel 2 Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_clkout 0x1: hwobs_abedpll_clkout 0x2: hwobs_perdppll_clkout 0x3: hwobs_dsi1dppll_clkout 0x4: hwobs_dsi2dppll_clkout 0x5: reserved 0x6: hwobs_usbdpll_clkout 0x7: reserved	RW	0x2

**Table 18-225. Register Call Summary for Register CONTROL\_CONF\_CLK\_SEL2**

Control Module Programming Guide

- [Hardware Observability Settings: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-226. CONTROL\_CONF\_DPLL\_FREQLOCK\_SEL**

<b>Address Offset</b>	0x0000 044C	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	dpll_freqlock view Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_freqlock 0x1: hwobs_abedpll_freqlock 0x2: hwobs_perdpll_freqlock 0x3: hwobs_dsi1dpll_freqlock 0x4: hwobs_dsi2dpll_freqlock 0x5: reserved 0x6: hwobs_usbdpll_freqlock 0x7: reserved	RW	0x0

**Table 18-227. Register Call Summary for Register CONTROL\_CONF\_DPLL\_FREQLOCK\_SEL**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)
- [Other CORE DPLL observable signals: \[1\]](#)
- [Other ABE DPLL observable signals: \[2\]](#)
- [Other PERIPH DPLL observable signals: \[3\]](#)
- [Other DSI1 DPLL observable signals: \[4\]](#)
- [Other DSI2 DPLL observable signals: \[5\]](#)
- [Other USB DPLL observable signals: \[6\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[7\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[8\]](#)

**Table 18-228. CONTROL\_CONF\_DPLL\_TINITZ\_SEL**

<b>Address Offset</b>	0x0000 0450	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	dpll_tinitz view Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_tinitz	RW	0x0

**Table 18-229. Register Call Summary for Register CONTROL\_CONF\_DPLL\_TINITZ\_SEL**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[2\]](#)

**Table 18-230. CONTROL\_CONF\_DPLL\_PHASELOCK\_SEL**

<b>Address Offset</b>	0x0000 0454		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	dpll_phaselock view Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_phaselock 0x1: hwobs_abedpll_phaselock 0x2: hwobs_perdpll_phaselock 0x3: hwobs_dsi1dpll_phaselock 0x4: hwobs_dsi2dpll_phaselock 0x5: reserved 0x6: hwobs_usbdpll_phaselock 0x7: reserved	RW	0x0

**Table 18-231. Register Call Summary for Register CONTROL\_CONF\_DPLL\_PHASELOCK\_SEL**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[2\]](#)

**Table 18-232. CONTROL\_CONF\_DPLL\_TENABLE\_SEL**

<b>Address Offset</b>	0x0000 045C		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	dpll_tenable view Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_tenable 0x1: hwobs_abedpll_tenable 0x2: hwobs_perdppll_tenable 0x3: hwobs_dsi1dppll_tenable 0x4: hwobs_dsi2dppll_tenable 0x5: reserved 0x6: hwobs_usbdpll_tenable 0x7: reserved	RW	0x0

**Table 18-233. Register Call Summary for Register CONTROL\_CONF\_DPLL\_TENABLE\_SEL**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[2\]](#)

**Table 18-234. CONTROL\_CONF\_DPLL\_TENABLEDIV\_SEL**

<b>Address Offset</b>	0x0000 0460	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	dpll_tenablediv view Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_tenablediv 0x1: hwobs_abedpll_tenablediv 0x2: hwobs_perdppll_tenablediv 0x3: hwobs_dsi1dppll_tenablediv 0x4: hwobs_dsi2dppll_tenablediv 0x5: reserved 0x6: hwobs_usbdpll_tenablediv 0x7: reserved	RW	0x0

**Table 18-235. Register Call Summary for Register CONTROL\_CONF\_DPLL\_TENABLEDIV\_SEL**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary:](#)

**Table 18-236. CONTROL\_CONF\_DPLL\_BYPASSACK\_SEL**

<b>Address Offset</b>	0x0000 0464		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	dpll_bypassack view Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_bypassack 0x1: hwobs_abedpll_bypassack 0x2: hwobs_perdpll_bypassack 0x3: hwobs_dsi1dpll_bypassack 0x4: hwobs_dsi2dpll_bypassack 0x5: reserved 0x6: hwobs_usbdpll_bypassack 0x7: reserved	RW	0x0

**Table 18-237. Register Call Summary for Register CONTROL\_CONF\_DPLL\_BYPASSACK\_SEL**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[2\]](#)

**Table 18-238. CONTROL\_CONF\_DPLL\_IDLE\_SEL**

<b>Address Offset</b>	0x0000 0468		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	dpll_idle view Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULT															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	MULT	Select one of the following signals: 0x0: hwobs_coredpll_idle 0x1: hwobs_abedpll_idle 0x2: hwobs_perdpll_idle 0x3: hwobs_dsi1dpll_idle 0x4: hwobs_dsi2dpll_idle 0x5: reserved 0x6: hwobs_usbdpll_idle	RW	0x0

Bits	Field Name	Description	Type	Reset
		0x7: reserved		

**Table 18-239. Register Call Summary for Register CONTROL\_CONF\_DPLL\_IDLE\_SEL**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)
- [Other CORE DPLL observable signals: \[1\]](#)
- [Other ABE DPLL observable signals: \[2\]](#)
- [Other PERIPH DPLL observable signals: \[3\]](#)
- [Other DSI1 DPLL observable signals: \[4\]](#)
- [Other DSI2 DPLL observable signals: \[5\]](#)
- [Other USB DPLL observable signals: \[6\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[7\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[8\]](#)

**Table 18-240. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_0**

<b>Address Offset</b>	0x0000 0480	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_0 0x1: clk_view_0 0x2: reserved 0x3: reserved 0x4: This information is not available in the public domain. 0x5: hwobs_int_cm2_0 0x6: hwobs_int_ivahd_0 0x7: reserved 0x8: hwobs_int_abe_0 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved 0xC: This information is not available in the public domain.	RW	0x0

**Table 18-241. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_0**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\] \[1\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[3\]](#)

**Table 18-242. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_1**

<b>Address Offset</b>	0x0000 0484		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_1 0x1: clk_view_1 0x2: reserved 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_1 0x6: hwobs_int_ivahd_1 0x7: reserved 0x8: hwobs_int_abe_1 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-243. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_1**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-244. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_2**

<b>Address Offset</b>	0x0000 0488		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_2 0x1: clk_view_2 0x2: reserved 0x3: reserved	RW	0x0



Bits	Field Name	Description	Type	Reset
		0x4: reserved		
		0x5: hwobs_int_cm2_2		
		0x6: hwobs_int_ivahd_2		
		0x7: reserved		
		0x8: hwobs_int_abe_2		
		0x9: This information is not available in the public domain.		
		0xA: reserved		
		0xB: reserved		

**Table 18-245. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_2**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-246. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_3**

<b>Address Offset</b>	0x0000 048C	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_3 0x1: dpll_freqlock 0x2: reserved 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_3 0x6: hwobs_int_ivahd_3 0x7: reserved 0x8: hwobs_int_abe_3 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-247. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_3**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\] \[1\] \[2\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[3\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[4\]](#)

**Table 18-248. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_4**

<b>Address Offset</b>	0x0000 0490		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_4 0x1: dpll_tinitz 0x2: reserved 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_4 0x6: hwobs_int_ivahd_4 0x7: reserved 0x8: hwobs_int_abe_4 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-249. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_4**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-250. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_5**

<b>Address Offset</b>	0x0000 0494		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_5 0x1: dpll_phaselock 0x2: reserved 0x3: reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
		0x4: reserved		
		0x5: hwobs_int_cm2_5		
		0x6: hwobs_int_ivahd_5		
		0x7: reserved		
		0x8: hwobs_int_abe_5		
		0x9: This information is not available in the public domain.		
		0xA: reserved		
		0xB: reserved		

**Table 18-251. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_5**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-252. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_6**

<b>Address Offset</b>	0x0000 0498	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_6 0x1: reserved 0x2: reserved 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_6 0x6: hwobs_int_ivahd_6 0x7: reserved 0x8: hwobs_int_abe_6 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-253. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_6**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-254. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_7**

<b>Address Offset</b>	0x0000 049C		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_7 0x1: dpll_tenable 0x2: sdma_req_view_0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_7 0x6: hwobs_int_ivahd_7 0x7: reserved 0x8: hwobs_int_abe_7 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-255. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_7**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-256. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_8**

<b>Address Offset</b>	0x0000 04A0		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_8 0x1: dpll_tenablediv 0x2: sdma_req_view_1 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_8 0x6: hwobs_int_ivahd_8 0x7: reserved 0x8: hwobs_int_abe_8 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-257. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_8**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-258. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_9**

<b>Address Offset</b>	0x0000 04A4	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_9 0x1: dpll_bypassack 0x2: sdma_req_view_2 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_9 0x6: hwobs_int_ivahd_9 0x7: reserved 0x8: hwobs_int_abe_9 0x9: This information is not available in the public domain. 0xA: reserved 0xB: hwobs_dssvencvtdetgp	RW	0x0

**Table 18-259. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_9**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-260. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_10**

<b>Address Offset</b>	0x0000 04A8	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_10 0x1: dpll_idle 0x2: sdma_req_view_3 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_10 0x6: hwobs_int_ivahd_10 0x7: reserved 0x8: hwobs_int_abe_10 0x9: This information is not available in the public domain. 0xA: reserved 0xB: hwobs_dssdacpwrndaczcvsbs_y	RW	0x0

**Table 18-261. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_10**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-262. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_11**

<b>Address Offset</b>	0x0000 04AC	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_11 0x1: hwobs_hsusbotg_sofpulse 0x2: sdma_req_view_all 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_11 0x6: hwobs_int_ivahd_11 0x7: reserved 0x8: hwobs_int_abe_11 0x9: This information is not available in the public domain. 0xA: reserved 0xB: hwobs_dssdacpwrndaczchroma	RW	0x0

**Table 18-263. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_11**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)
- [USBOTGHS Subsystem Observable Signals: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[2\]](#)

**Table 18-264. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_12**

<b>Address Offset</b>	0x0000 04B0	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_12 0x1: reserved 0x2: hwobs_coredivider_clkout3 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_12 0x6: hwobs_int_ivahd_12 0x7: reserved 0x8: hwobs_int_abe_12 0x9: This information is not available in the public domain. 0xA: reserved 0xB: hwobs_dssdacpwrndaccvbs	RW	0x0

**Table 18-265. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_12**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-266. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_13**

<b>Address Offset</b>	0x0000 04B4	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_13 0x1: reserved 0x2: hwobs_perdivider_clkout4 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_13 0x6: hwobs_int_ivahd_13 0x7: reserved 0x8: hwobs_int_abe_13 0x9: This information is not available in the public domain. 0xA: reserved 0xB: hwobs_dssdacpwrndbgz	RW	0x0

**Table 18-267. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_13**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-268. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_14**

<b>Address Offset</b>	0x0000 04B8	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_14 0x1: reserved 0x2: hwobs_pd_l4_per_per32k_gfclk 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_14 0x6: hwobs_int_ivahd_14 0x7: reserved 0x8: hwobs_int_abe_14 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-269. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_14**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-270. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_15**

<b>Address Offset</b>	0x0000 04BC	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_15 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_15 0x6: hwobs_int_ivahd_15 0x7: reserved 0x8: hwobs_int_abe_15 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-271. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_15**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-272. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_16**

<b>Address Offset</b>	0x0000 04C0	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_16 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_16 0x6: hwobs_int_ivahd_16 0x7: reserved 0x8: hwobs_int_abe_16 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-273. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_16**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-274. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_17**

<b>Address Offset</b>	0x0000 04C4	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_17 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_17 0x6: hwobs_int_ivahd_17 0x7: reserved 0x8: hwobs_int_abe_17 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-275. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_17**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-276. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_18**

<b>Address Offset</b>	0x0000 04C8	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_18 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_18 0x6: hwobs_int_ivahd_18 0x7: reserved 0x8: hwobs_int_abe_18 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-277. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_18**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-278. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_19**

<b>Address Offset</b>	0x0000 04CC	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_19 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_19 0x6: hwobs_int_ivahd_19 0x7: reserved 0x8: hwobs_int_abe_19 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-279. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_19**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-280. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_20**

<b>Address Offset</b>	0x0000 04D0	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_20 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_20 0x6: hwobs_int_ivahd_20 0x7: reserved 0x8: hwobs_int_abe_20 0x9: This information is not available in the public domain. 0xA: reserved 0xB: reserved	RW	0x0

**Table 18-281. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_20**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-282. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_21**

<b>Address Offset</b>	0x0000 04D4	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_21 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_21 0x6: hwobs_int_ivahd_21 0x7: reserved 0x8: hwobs_int_abe_21 0x9: This information is not available in the public domain. 0xA: reserved	RW	0x0

**Table 18-283. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_21**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-284. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_22**

<b>Address Offset</b>	0x0000 04D8	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_22 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_22 0x6: hwobs_int_ivahd_22 0x7: reserved 0x8: hwobs_int_abe_22 0x9: This information is not available in the public domain. 0xA: reserved	RW	0x0

**Table 18-285. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_22**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-286. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_23**

<b>Address Offset</b>	0x0000 04DC	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_23 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_23 0x6: hwobs_int_ivahd_23 0x7: reserved 0x8: hwobs_int_abe_23 0x9: This information is not available in the public domain. 0xA: reserved	RW	0x0

**Table 18-287. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_23**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-288. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_24**

<b>Address Offset</b>	0x0000 04E0	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Physical Address</b>	See <a href="#">Table 18-113</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_24 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_24 0x6: hwobs_int_ivahd_24 0x7: reserved 0x8: hwobs_int_abe_24 0x9: This information is not available in the public domain. 0xA: reserved	RW	0x0

**Table 18-289. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_24**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-290. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_25**

<b>Address Offset</b>	0x0000 04E4	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_25 0x1: reserved 0x2: 0 0x3: reserved 0x4: reserved 0x5: hwobs_int_cm2_25 0x6: hwobs_int_ivahd_25 0x7: reserved 0x8: hwobs_int_abe_25 0x9: This information is not available in the public domain. 0xA: reserved	RW	0x0

**Table 18-291. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_25**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-292. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_26**

<b>Address Offset</b>	0x0000 04E8	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_26 0x1: reserved 0x2: 0 0x3: reserved 0x4: 0 0x5: hwobs_int_cm2_26 0x6: hwobs_int_ivahd_26 0x7: reserved 0x8: hwobs_int_abe_26 0x9: This information is not available in the public domain. 0xA: reserved	RW	0x0

**Table 18-293. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_26**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-294. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_27**

<b>Address Offset</b>	0x0000 04EC		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_27 0x1: reserved 0x2: 0 0x3: reserved 0x4: 0 0x5: hwobs_int_cm2_27 0x6: hwobs_int_ivahd_27 0x7: reserved 0x8: hwobs_int_abe_27 0x9: This information is not available in the public domain. 0xA: reserved 0xB: 0 0xC: sdma_req_view_0	RW	0x0

**Table 18-295. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_27**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-296. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_28**

<b>Address Offset</b>	0x0000 04F0	
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_28 0x1: reserved 0x2: 0 0x3: reserved 0x4: 0 0x5: hwobs_int_cm2_28 0x6: hwobs_int_ivahd_28 0x7: reserved 0x8: hwobs_int_abe_28 0x9: This information is not available in the public domain. 0xA: reserved 0xB: 0 0xC: sdma_req_view_1	RW	0x0

**Table 18-297. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_28**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-298. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_29**

<b>Address Offset</b>	0x0000 04F4		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_29 0x1: reserved 0x2: 0 0x3: reserved 0x4: 0 0x5: hwobs_int_cm2_29 0x6: hwobs_int_ivahd_29 0x7: reserved 0x8: hwobs_int_abe_29 0x9: This information is not available in the public domain. 0xA: reserved 0xB: 0 0xC: sdma_req_view_2	RW	0x0

**Table 18-299. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_29**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-300. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_30**

<b>Address Offset</b>	0x0000 04F8		
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b>	SYSCTRL_GENERAL_CORE
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_30 0x1: reserved 0x2: 0 0x3: reserved 0x4: 0 0x5: hwobs_int_cm2_30 0x6: hwobs_int_ivahd_30 0x7: reserved 0x8: hwobs_int_abe_30 0x9: This information is not available in the public domain. 0xA: reserved 0xB: 0 0xC: sdma_req_view_3	RW	0x0

**Table 18-301. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_30**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[1\]](#)

**Table 18-302. CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_31**

<b>Address Offset</b>	0x0000 04FC																																																													
<b>Physical Address</b>	See <a href="#">Table 18-113</a>	<b>Instance</b> SYSCTRL_GENERAL_CORE																																																												
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> </thead> <tbody> <tr> <td colspan="16">RESERVED</td> <td colspan="12">MODE</td> </tr> </tbody> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																MODE											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
RESERVED																MODE																																														

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	MODE	Select one of the following signals: 0x0: hwobs_int_mpu_31 0x1: reserved 0x2: 0 0x3: reserved 0x4: 0 0x5: hwobs_int_cm2_31 0x6: hwobs_int_ivahd_31 0x7: reserved 0x8: hwobs_int_abe_31 0x9: This information is not available in the public domain. 0xA: reserved 0xB: 0 0xC: sdma_req_view_all	RW	0x0

**Table 18-303. Register Call Summary for Register CONTROL\_CORE\_CONF\_DEBUG\_SEL\_TST\_31**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_CORE Register Summary: \[0\]](#)

### 18.6.4 SYSCTRL\_GENERAL\_WKUP Register Summary

**Table 18-304. SYSCTRL\_GENERAL\_WKUP Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_GENERAL_WKUP Physical Address
<a href="#">CONTROL_GEN_WKUP_REVISION</a>	R	32	0x0000 0000	0x4A30 C000
<a href="#">CONTROL_GEN_WKUP_HWINFO</a>	R	32	0x0000 0004	0x4A30 C004
<a href="#">CONTROL_GEN_WKUP_SYSCONFIG</a>	RW	32	0x0000 0010	0x4A30 C010
RESERVED	RW	32	0x0000 0104	0x4A30 C104
RESERVED	RW	32	0x0000 0108	0x4A30 C108
<a href="#">CONTROL_OCPREG_SPARE</a>	RW	32	0x0000 010C	0x4A30 C10C
<a href="#">CONTROL_WKUP_CONF_DEBUG_SE L_TST_i<sup>(1)</sup></a>	RW	32	0x0000 0460 + (i*0x004)	0x4A30 C460 + (i*0x004)

<sup>(1)</sup> i = 0 to 31

### 18.6.5 SYSCTRL\_GENERAL\_WKUP Register Description

**Table 18-305. CONTROL\_GEN\_WKUP\_REVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	SYSCTRL_GENERAL_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-304</a>		
<b>Description</b>	Control module revision identifier Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 18-306. Register Call Summary for Register CONTROL\_GEN\_WKUP\_REVISION**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_WKUP Register Summary: \[0\]](#)

**Table 18-307. CONTROL\_GEN\_WKUP\_HWINFO**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 18-304</a>	<b>Instance</b>	SYSCTRL_GENERAL_WKUP
<b>Description</b>	Information about the IP module hardware configuration that is, typically the module HDL generics (if any). Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP_HWINFO																															

Bits	Field Name	Description	Type	Reset
31:0	IP_HWINFO	IP-module dependent	R	0x0000 0000

**Table 18-308. Register Call Summary for Register CONTROL\_GEN\_WKUP\_HWINFO**

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_WKUP Register Summary: \[0\]](#)

**Table 18-309. CONTROL\_GEN\_WKUP\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 18-304</a>	<b>Instance</b>	SYSCTRL_GENERAL_WKUP
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IP_SYSCONFIG_IDLEMODE		RESERVED													

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:2	IP_SYSCONFIG_IDLEMODE	Select the local clock-gating strategy 0x0: Clock is manually gated 0x1: Clock is manually enabled 0x2,0x3: Clock is automatically gated when there is no access to the Control Module through L4-interconnect	RW	0x2

Bits	Field Name	Description	Type	Reset
1:0	RESERVED		R	0x0

**Table 18-310. Register Call Summary for Register CONTROL\_GEN\_WKUP\_SYSCONFIG**

- Control Module Functional Description
- [General Wakeup Control Module Instance: \[0\]](#)
- Control Module Programming Guide
- [Control Module Global Initialization: \[1\]](#)
- Control Module Register Manual
- [SYSCTRL\\_GENERAL\\_WKUP Register Summary: \[2\]](#)

**Table 18-311. CONTROL\_OCPREG\_SPARE**

<b>Address Offset</b>	0x0000 010C
<b>Physical Address</b>	Please refer to <a href="#">Table 18-304</a>
<b>Description</b>	Spare Register Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPPREG_SPARE31	OCPPREG_SPARE30	OCPPREG_SPARE29	OCPPREG_SPARE28	OCPPREG_SPARE27	OCPPREG_SPARE26	OCPPREG_SPARE25	OCPPREG_SPARE24	OCPPREG_SPARE23	OCPPREG_SPARE22	OCPPREG_SPARE21	OCPPREG_SPARE20	OCPPREG_SPARE19	OCPPREG_SPARE18	OCPPREG_SPARE17	OCPPREG_SPARE16	OCPPREG_SPARE15	OCPPREG_SPARE14	OCPPREG_SPARE13	OCPPREG_SPARE12	OCPPREG_SPARE11	OCPPREG_SPARE10	OCPPREG_SPARE9	OCPPREG_SPARE8	OCPPREG_SPARE7	OCPPREG_SPARE6	OCPPREG_SPARE5	OCPPREG_SPARE4	OCPPREG_SPARE3	OCPPREG_SPARE2	OCPPREG_SPARE1	RESERVED

Bits	Field Name	Description	Type	Reset
31	OCPPREG_SPARE31	Spare register 31	RW	0
30	OCPPREG_SPARE30	Spare register 30	RW	0
29	OCPPREG_SPARE29	Spare register 29	RW	0
28	OCPPREG_SPARE28	Spare register 28	RW	0
27	OCPPREG_SPARE27	Spare register 27	RW	0
26	OCPPREG_SPARE26	Spare register 26	RW	0
25	OCPPREG_SPARE25	Spare register 25	RW	0
24	OCPPREG_SPARE24	Spare register 24	RW	0
23	OCPPREG_SPARE23	Spare register 23	RW	0
22	OCPPREG_SPARE22	Spare register 22	RW	0
21	OCPPREG_SPARE21	Spare register 21	RW	0
20	OCPPREG_SPARE20	Spare register 20	RW	0
19	OCPPREG_SPARE19	Spare register 19	RW	0
18	OCPPREG_SPARE18	Spare register 18	RW	0
17	OCPPREG_SPARE17	Spare register 17	RW	0
16	OCPPREG_SPARE16	Spare register 16	RW	0
15	OCPPREG_SPARE15	Spare register 15	RW	0
14	OCPPREG_SPARE14	Spare register 14	RW	0
13	OCPPREG_SPARE13	Spare register 13	RW	0
12	OCPPREG_SPARE12	Spare register 12	RW	0
11	OCPPREG_SPARE11	Spare register 11	RW	0
10	OCPPREG_SPARE10	Spare register 10	RW	0
9	OCPPREG_SPARE9	Spare register 9	RW	0

Bits	Field Name	Description	Type	Reset
8	OCPPREG_SPARE8	Spare register 8	RW	0
7	OCPPREG_SPARE7	Spare register 7	RW	0
6	OCPPREG_SPARE6	Spare register 6	RW	0
5	OCPPREG_SPARE5	Spare register 5	RW	0
4	OCPPREG_SPARE4	Spare register 4	RW	0
3	OCPPREG_SPARE3	Spare register 3	RW	0
2	OCPPREG_SPARE2	Spare register 2	RW	0
1	OCPPREG_SPARE1	Spare register 1	RW	0
0	RESERVED	Reserved	RW	0

**Table 18-312. Register Call Summary for Register CONTROL\_OCPREG\_SPARE**

Control Module Functional Description

- [General Wakeup Control Module Instance: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_WKUP Register Summary: \[1\]](#)

**Table 18-313. CONTROL\_WKUP\_CONF\_DEBUG\_SEL\_TST\_i**

<b>Address Offset</b>	0x0000 0460	<b>Instance</b>	SYSCTRL_GENERAL_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-304</a>		
<b>Description</b>	Select mode for debug port Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MODE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MODE	Select one of the following signals: 0x0: hwobs_int_prm_i <sup>(1)</sup> 0x1: hwobs_int_cm1_i	RW	0

<sup>(1)</sup> i = 0 to 31**Table 18-314. Register Call Summary for Register CONTROL\_WKUP\_CONF\_DEBUG\_SEL\_TST\_i**

Control Module Functional Description

- [Observability Signals Multiplexing at General Core/Wake-Up Control Level: \[0\] \[1\]](#)

Control Module Programming Guide

- [Hardware Observability Settings: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_GENERAL\\_WKUP Register Summary: \[3\]](#)

## 18.6.6 SYSCTRL\_PADCONF\_CORE Register Summary

**Table 18-315. SYSCTRL\_PADCONF\_CORE Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_PADCONF_C ORE Physical Address
<a href="#">CONTROL_PADCONF_CORE_REVISION</a>	R	32	0x00000000	0x4A10 0000



**Table 18-315. SYSCTRL\_PADCONF\_CORE Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_PADCONF_CORE Physical Address
CONTROL_PADCONF_CORE_HWINFO	R	32	0x00000004	0x4A10 0004
CONTROL_PADCONF_CORE_SYSCONFIG	RW	32	0x00000010	0x4A10 0010
CONTROL_CORE_PAD0_GPMC_AD0_PAD1_GPMC_AD1	RW	32	0x00000040	0x4A10 0040
CONTROL_CORE_PAD0_GPMC_AD2_PAD1_GPMC_AD3	RW	32	0x00000044	0x4A10 0044
CONTROL_CORE_PAD0_GPMC_AD4_PAD1_GPMC_AD5	RW	32	0x00000048	0x4A10 0048
CONTROL_CORE_PAD0_GPMC_AD6_PAD1_GPMC_AD7	RW	32	0x0000004C	0x4A10 004C
CONTROL_CORE_PAD0_GPMC_AD8_PAD1_GPMC_AD9	RW	32	0x00000050	0x4A10 0050
CONTROL_CORE_PAD0_GPMC_AD10_PAD1_GPMC_AD11	RW	32	0x00000054	0x4A10 0054
CONTROL_CORE_PAD0_GPMC_AD12_PAD1_GPMC_AD13	RW	32	0x00000058	0x4A10 0058
CONTROL_CORE_PAD0_GPMC_AD14_PAD1_GPMC_AD15	RW	32	0x0000005C	0x4A10 005C
CONTROL_CORE_PAD0_GPMC_A16_PAD1_GPMC_A17	RW	32	0x00000060	0x4A10 0060
CONTROL_CORE_PAD0_GPMC_A18_PAD1_GPMC_A19	RW	32	0x00000064	0x4A10 0064
CONTROL_CORE_PAD0_GPMC_A20_PAD1_GPMC_A21	RW	32	0x00000068	0x4A10 0068
CONTROL_CORE_PAD0_GPMC_A22_PAD1_GPMC_A23	RW	32	0x0000006C	0x4A10 006C
CONTROL_CORE_PAD0_GPMC_A24_PAD1_GPMC_A25	RW	32	0x00000070	0x4A10 0070
CONTROL_CORE_PAD0_GPMC_NCS0_PAD1_GPMC_NCS1	RW	32	0x00000074	0x4A10 0074
CONTROL_CORE_PAD0_GPMC_NCS2_PAD1_GPMC_NCS3	RW	32	0x00000078	0x4A10 0078
CONTROL_CORE_PAD0_GPMC_NWP_PAD1_GPMC_CLK	RW	32	0x0000007C	0x4A10 007C
CONTROL_CORE_PAD0_GPMC_NADV_ALE_PAD1_GPMC_NOE	RW	32	0x00000080	0x4A10 0080
CONTROL_CORE_PAD0_GPMC_NWE_PAD1_GPMC_NBE0_CLE	RW	32	0x00000084	0x4A10 0084
CONTROL_CORE_PAD0_GPMC_NBE1_PAD1_GPMC_WAIT0	RW	32	0x00000088	0x4A10 0088
CONTROL_CORE_PAD0_GPMC_WAIT1_PAD1_GPMC_WAIT2	RW	32	0x0000008C	0x4A10 008C
CONTROL_CORE_PAD0_GPMC_NCS4_PAD1_GPMC_NCS5	RW	32	0x00000090	0x4A10 0090
CONTROL_CORE_PAD0_GPMC_NCS6_PAD1_GPMC_NCS7	RW	32	0x00000094	0x4A10 0094
CONTROL_CORE_PAD0_GPIO63_PAD1_GPIO64	RW	32	0x00000098	0x4A10 0098
CONTROL_CORE_PAD0_GPIO65_PAD1_GPIO66	RW	32	0x0000009C	0x4A10 009C
CONTROL_CORE_PAD0_CSI21_DX0_PAD1_CSI21_DY0	RW	32	0x000000A0	0x4A10 00A0

**Table 18-315. SYSCTRL\_PADCONF\_CORE Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_PADCONF_CORE Physical Address
CONTROL_CORE_PAD0_CSI21_DX1_PAD1_CSI21_DY1	RW	32	0x000000A4	0x4A10 00A4
CONTROL_CORE_PAD0_CSI21_DX2_PAD1_CSI21_DY2	RW	32	0x000000A8	0x4A10 00A8
CONTROL_CORE_PAD0_CSI21_DX3_PAD1_CSI21_DY3	RW	32	0x000000AC	0x4A10 00AC
CONTROL_CORE_PAD0_CSI21_DX4_PAD1_CSI21_DY4	RW	32	0x000000B0	0x4A10 00B0
CONTROL_CORE_PAD0_CSI22_DX0_PAD1_CSI22_DY0	RW	32	0x000000B4	0x4A10 00B4
CONTROL_CORE_PAD0_CSI22_DX1_PAD1_CSI22_DY1	RW	32	0x000000B8	0x4A10 00B8
CONTROL_CORE_PAD0_CAM_SHUTTER_PAD1_CAM_STROBE	RW	32	0x000000BC	0x4A10 00BC
CONTROL_CORE_PAD0_CAM_GLOBALRESET_PAD1_USBB1_ULPITLL_CLK	RW	32	0x000000C0	0x4A10 00C0
CONTROL_CORE_PAD0_USBB1_ULPITLL_STP_PAD1_USBB1_ULPITLL_DIR	RW	32	0x000000C4	0x4A10 00C4
CONTROL_CORE_PAD0_USBB1_ULPITLL_NXT_PAD1_USBB1_ULPITLL_DATA0	RW	32	0x000000C8	0x4A10 00C8
CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT1_PAD1_USBB1_ULPITLL_DATA2	RW	32	0x000000CC	0x4A10 00CC
CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT3_PAD1_USBB1_ULPITLL_DATA4	RW	32	0x000000D0	0x4A10 00D0
CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT5_PAD1_USBB1_ULPITLL_DATA6	RW	32	0x000000D4	0x4A10 00D4
CONTROL_CORE_PAD0_USBB1_ULPITLL_DAT7_PAD1_USBB1_HSI_DATA	RW	32	0x000000D8	0x4A10 00D8
CONTROL_CORE_PAD0_USBB1_HSI_STROBE_PAD1_USBB1_ICUSB_DP	RW	32	0x000000DC	0x4A10 00DC
CONTROL_CORE_PAD0_USBB1_ICUSB_DM_PAD1_SDMMC1_CLK	RW	32	0x000000E0	0x4A10 00E0
CONTROL_CORE_PAD0_SDMMC1_CMD_PAD1_SDMMC1_DATA0	RW	32	0x000000E4	0x4A10 00E4
CONTROL_CORE_PAD0_SDMMC1_DATA1_PAD1_SDMMC1_DATA2	RW	32	0x000000E8	0x4A10 00E8
CONTROL_CORE_PAD0_SDMMC1_DATA3_PAD1_SDMMC1_DATA4	RW	32	0x000000EC	0x4A10 00EC
CONTROL_CORE_PAD0_SDMMC1_DATA5_PAD1_SDMMC1_DATA6	RW	32	0x000000F0	0x4A10 00F0
CONTROL_CORE_PAD0_SDMMC1_DATA7_PAD1_ABE_MCBSP2_CLKX	RW	32	0x000000F4	0x4A10 00F4
CONTROL_CORE_PAD0_ABE_MCBSP2_DR_PAD1_ABE_MCBSP2_DX	RW	32	0x000000F8	0x4A10 00F8
CONTROL_CORE_PAD0_ABE_MCBSP2_FSX_PAD1_ABE_MCBSP1_CLKX	RW	32	0x000000FC	0x4A10 00FC
CONTROL_CORE_PAD0_ABE_MCBSP1_DR_PAD1_ABE_MCBSP1_DX	RW	32	0x00000100	0x4A10 0100
CONTROL_CORE_PAD0_ABE_MCBSP1_FSX_PAD1_ABE_PDM_UL_DATA	RW	32	0x00000104	0x4A10 0104

**Table 18-315. SYSCTRL\_PADCONF\_CORE Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_PADCONF_CORE Physical Address
CONTROL_CORE_PAD0_ABE_PDM_DL_DATA_PAD1_ABE_PDM_FRAME	RW	32	0x00000108	0x4A10 0108
CONTROL_CORE_PAD0_ABE_PDM_LB_CLK_PAD1_ABE_CLKS	RW	32	0x0000010C	0x4A10 010C
CONTROL_CORE_PAD0_ABE_DMIC_CLK1_PAD1_ABE_DMIC_DIN1	RW	32	0x00000110	0x4A10 0110
CONTROL_CORE_PAD0_ABE_DMIC_DIN2_PAD1_ABE_DMIC_DIN3	RW	32	0x00000114	0x4A10 0114
CONTROL_CORE_PAD0_UART2_CTS_PAD1_UART2_RTS	RW	32	0x00000118	0x4A10 0118
CONTROL_CORE_PAD0_UART2_RX_PAD1_UART2_TX	RW	32	0x0000011C	0x4A10 011C
CONTROL_CORE_PAD0_HDQ_SIO_PAD1_I2C1_SCL	RW	32	0x00000120	0x4A10 0120
CONTROL_CORE_PAD0_I2C1_SDA_PAD1_I2C2_SCL	RW	32	0x00000124	0x4A10 0124
CONTROL_CORE_PAD0_I2C2_SDA_PAD1_I2C3_SCL	RW	32	0x00000128	0x4A10 0128
CONTROL_CORE_PAD0_I2C3_SDA_PAD1_I2C4_SCL	RW	32	0x0000012C	0x4A10 012C
CONTROL_CORE_PAD0_I2C4_SDA_PAD1_MCSP11_CLK	RW	32	0x00000130	0x4A10 0130
CONTROL_CORE_PAD0_MCSP11_SOMI_PAD1_MCSP11_SIMO	RW	32	0x00000134	0x4A10 0134
CONTROL_CORE_PAD0_MCSP11_CS0_PAD1_MCSP11_CS1	RW	32	0x00000138	0x4A10 0138
CONTROL_CORE_PAD0_MCSP11_CS2_PAD1_MCSP11_CS3	RW	32	0x0000013C	0x4A10 013C
CONTROL_CORE_PAD0_UART3_CTS_RCTX_PAD1_UART3_RTS_SD	RW	32	0x00000140	0x4A10 0140
CONTROL_CORE_PAD0_UART3_RX_IRRX_PAD1_UART3_TX_IRTX	RW	32	0x00000144	0x4A10 0144
CONTROL_CORE_PAD0_SDMMC5_CLK_PAD1_SDMMC5_CMD	RW	32	0x00000148	0x4A10 0148
CONTROL_CORE_PAD0_SDMMC5_DAT0_PAD1_SDMMC5_DAT1	RW	32	0x0000014C	0x4A10 014C
CONTROL_CORE_PAD0_SDMMC5_DAT2_PAD1_SDMMC5_DAT3	RW	32	0x00000150	0x4A10 0150
CONTROL_CORE_PAD0_MCSP14_CLK_PAD1_MCSP14_SIMO	RW	32	0x00000154	0x4A10 0154
CONTROL_CORE_PAD0_MCSP14_SOMI_PAD1_MCSP14_CS0	RW	32	0x00000158	0x4A10 0158
CONTROL_CORE_PAD0_UART4_RX_PAD1_UART4_TX	RW	32	0x0000015C	0x4A10 015C
CONTROL_CORE_PAD0_USBB2_ULPITLL_CLK_PAD1_USBB2_ULPITLL_STP	RW	32	0x00000160	0x4A10 0160
CONTROL_CORE_PAD0_USBB2_ULPITLL_DIR_PAD1_USBB2_ULPITLL_NXT	RW	32	0x00000164	0x4A10 0164
CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT0_PAD1_USBB2_ULPITLL_DAT1	RW	32	0x00000168	0x4A10 0168
CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT2_PAD1_USBB2_ULPITLL_DAT3	RW	32	0x0000016C	0x4A10 016C

**Table 18-315. SYSCTRL\_PADCONF\_CORE Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_PADCONF_CORE Physical Address
CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT4_PAD1_USBB2_ULPITLL_DAT5	RW	32	0x00000170	0x4A10 0170
CONTROL_CORE_PAD0_USBB2_ULPITLL_DAT6_PAD1_USBB2_ULPITLL_DAT7	RW	32	0x00000174	0x4A10 0174
CONTROL_CORE_PAD0_USBB2_HSI_C_DATA_PAD1_USBB2_HSI_C_STROBE	RW	32	0x00000178	0x4A10 0178
CONTROL_CORE_PAD0_KPD_COL3_PAD1_KPD_COL4	RW	32	0x0000017C	0x4A10 017C
CONTROL_CORE_PAD0_KPD_COL5_PAD1_KPD_COL0	RW	32	0x00000180	0x4A10 0180
CONTROL_CORE_PAD0_KPD_COL1_PAD1_KPD_COL2	RW	32	0x00000184	0x4A10 0184
CONTROL_CORE_PAD0_KPD_ROW3_PAD1_KPD_ROW4	RW	32	0x00000188	0x4A10 0188
CONTROL_CORE_PAD0_KPD_ROW5_PAD1_KPD_ROW0	RW	32	0x0000018C	0x4A10 018C
CONTROL_CORE_PAD0_KPD_ROW1_PAD1_KPD_ROW2	RW	32	0x00000190	0x4A10 0190
CONTROL_CORE_PAD0_USBA0_OTG_CE_PAD1_USBA0_OTG_DP	RW	32	0x00000194	0x4A10 0194
CONTROL_CORE_PAD0_USBA0_OTG_DM_PAD1_FREF_CLK1_OUT	RW	32	0x00000198	0x4A10 0198
CONTROL_CORE_PAD0_FREF_CLK2_OUT_PAD1_SYS_NIRQ1	RW	32	0x0000019C	0x4A10 019C
CONTROL_CORE_PAD0_SYS_NIRQ2_PAD1_SYS_BOOT0	RW	32	0x000001A0	0x4A10 01A0
CONTROL_CORE_PAD0_SYS_BOOT1_PAD1_SYS_BOOT2	RW	32	0x000001A4	0x4A10 01A4
CONTROL_CORE_PAD0_SYS_BOOT3_PAD1_SYS_BOOT4	RW	32	0x000001A8	0x4A10 01A8
CONTROL_CORE_PAD0_SYS_BOOT5_PAD1_DPM_EMU0	RW	32	0x000001AC	0x4A10 01AC
CONTROL_CORE_PAD0_DPM_EMU1_PAD1_DPM_EMU2	RW	32	0x000001B0	0x4A10 01B0
CONTROL_CORE_PAD0_DPM_EMU3_PAD1_DPM_EMU4	RW	32	0x000001B4	0x4A10 01B4
CONTROL_CORE_PAD0_DPM_EMU5_PAD1_DPM_EMU6	RW	32	0x000001B8	0x4A10 01B8
CONTROL_CORE_PAD0_DPM_EMU7_PAD1_DPM_EMU8	RW	32	0x000001BC	0x4A10 01BC
CONTROL_CORE_PAD0_DPM_EMU9_PAD1_DPM_EMU10	RW	32	0x000001C0	0x4A10 01C0
CONTROL_CORE_PAD0_DPM_EMU11_PAD1_DPM_EMU12	RW	32	0x000001C4	0x4A10 01C4
CONTROL_CORE_PAD0_DPM_EMU13_PAD1_DPM_EMU14	RW	32	0x000001C8	0x4A10 01C8
CONTROL_CORE_PAD0_DPM_EMU15_PAD1_DPM_EMU16	RW	32	0x000001CC	0x4A10 01CC
CONTROL_CORE_PAD0_DPM_EMU17_PAD1_DPM_EMU18	RW	32	0x000001D0	0x4A10 01D0
CONTROL_CORE_PAD0_DPM_EMU19	RW	32	0x000001D4	0x4A10 01D4

**Table 18-315. SYSCTRL\_PADCONF\_CORE Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_PADCONF_CORE Physical Address
CONTROL_PADCONF_WAKEUPEVENT_0	R	32	0x000001D8	0x4A10 01D8
CONTROL_PADCONF_WAKEUPEVENT_1	R	32	0x000001DC	0x4A10 01DC
CONTROL_PADCONF_WAKEUPEVENT_2	R	32	0x000001E0	0x4A10 01E0
CONTROL_PADCONF_WAKEUPEVENT_3	R	32	0x000001E4	0x4A10 01E4
CONTROL_PADCONF_WAKEUPEVENT_4	R	32	0x000001E8	0x4A10 01E8
CONTROL_PADCONF_WAKEUPEVENT_5	R	32	0x000001EC	0x4A10 01EC
CONTROL_PADCONF_WAKEUPEVENT_6	R	32	0x000001F0	0x4A10 01F0
CONTROL_PADCONF_GLOBAL	RW	32	0x000005A0	0x4A10 05A0
CONTROL_CORE_PADCONF_MODE	RW	32	0x000005A4	0x4A10 05A4
CONTROL_SMART1IO_PADCONF_0	RW	32	0x000005A8	0x4A10 05A8
CONTROL_SMART1IO_PADCONF_1	RW	32	0x000005AC	0x4A10 05AC
CONTROL_SMART2IO_PADCONF_0	RW	32	0x000005B0	0x4A10 05B0
CONTROL_SMART2IO_PADCONF_1	RW	32	0x000005B4	0x4A10 05B4
CONTROL_SMART3IO_PADCONF_0	RW	32	0x000005B8	0x4A10 05B8
CONTROL_SMART3IO_PADCONF_1	RW	32	0x000005BC	0x4A10 05BC
CONTROL_SMART3IO_PADCONF_2	RW	32	0x000005C0	0x4A10 05C0
CONTROL_USBB_HSIC	RW	32	0x000005C4	0x4A10 05C4
CONTROL_SMART3IO_PADCONF_3	RW	32	0x000005C8	0x4A10 05C8
CONTROL_SMART2IO_PADCONF_2	RW	32	0x000005CC	0x4A10 05CC
CONTROL_SMART1IO_PADCONF_2	RW	32	0x000005D0	0x4A10 05D0
CONTROL_SMART1IO_PADCONF_3	RW	32	0x000005D4	0x4A10 05D4
CONTROL_C2CIO_PADCONF_0	RW	32	0x000005D8	0x4A10 05D8
CONTROL_PBIASLITE	RW	32	0x00000600	0x4A10 0600
CONTROL_I2C_0	RW	32	0x00000604	0x4A10 0604
CONTROL_CAMERA_RX	RW	32	0x00000608	0x4A10 0608
CONTROL_AVDAC	RW	32	0x0000060C	0x4A10 060C
RESERVED	RW	32	0x00000610	0x4A10 0610
CONTROL_MMC2	RW	32	0x00000614	0x4A10 0614
CONTROL_DSIPHY	RW	32	0x00000618	0x4A10 0618
CONTROL_MCBSPLP	RW	32	0x0000061C	0x4A10 061C
CONTROL_USB2PHYCORE	RW	32	0x00000620	0x4A10 0620
RESERVED	RW	32	0x00000624	0x4A10 0624
CONTROL_MMC1	RW	32	0x00000628	0x4A10 0628
CONTROL_HSI	RW	32	0x0000062C	0x4A10 062C
CONTROL_USB	RW	32	0x00000630	0x4A10 0630
CONTROL_HDQ	RW	32	0x00000634	0x4A10 0634
CONTROL_LPDDR2IO1_0	RW	32	0x00000638	0x4A10 0638
CONTROL_LPDDR2IO1_1	RW	32	0x0000063C	0x4A10 063C
CONTROL_LPDDR2IO1_2	RW	32	0x00000640	0x4A10 0640
CONTROL_LPDDR2IO1_3	RW	32	0x00000644	0x4A10 0644
CONTROL_LPDDR2IO2_0	RW	32	0x00000648	0x4A10 0648
CONTROL_LPDDR2IO2_1	RW	32	0x0000064C	0x4A10 064C

**Table 18-315. SYSCTRL\_PADCONF\_CORE Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_PADCONF_CORE Physical Address
CONTROL_LPDDR2IO2_2	RW	32	0x00000650	0x4A10 0650
CONTROL_LPDDR2IO2_3	RW	32	0x00000654	0x4A10 0654
CONTROL_BUS_HOLD	RW	32	0x00000658	0x4A10 0658
CONTROL_C2C	RW	32	0x0000065C	0x4A10 065C
CONTROL_CORE_CONTROL_SPARE_RW	RW	32	0x00000660	0x4A10 0660
CONTROL_CORE_CONTROL_SPARE_R	R	32	0x00000664	0x4A10 0664
CONTROL_CORE_CONTROL_SPARE_R_C0	RW	32	0x00000668	0x4A10 0668
CONTROL_CORE_CONTROL_SPARE_RW1	RW	32	0x0000066C	0x4A10 066C
CONTROL_CORE_CONTROL_SPARE_RW2	RW	32	0x00000670	0x4A10 0670
CONTROL_CORE_CONTROL_SPARE_RW3	RW	32	0x00000674	0x4A10 0674
CONTROL_CORE_CONTROL_SPARE_RW4	RW	32	0x00000678	0x4A10 0678
CONTROL_CORE_CONTROL_SPARE_RW5	RW	32	0x0000067C	0x4A10 067C
CONTROL_CORE_CONTROL_SPARE_RW6	RW	32	0x00000680	0x4A10 0680
CONTROL_CORE_CONTROL_SPARE_RW7	RW	32	0x00000684	0x4A10 0684
CONTROL_CORE_CONTROL_SPARE_RW8	RW	32	0x00000688	0x4A10 0688
CONTROL_CORE_CONTROL_SPARE_RW9	RW	32	0x0000068C	0x4A10 068C
CONTROL_CORE_CONTROL_SPARE_R1	R	32	0x00000690	0x4A10 0690
CONTROL_EFUSE_1	RW	32	0x00000700	0x4A10 0700
CONTROL_EFUSE_2	RW	32	0x00000704	0x4A10 0704
CONTROL_EFUSE_3	R	32	0x00000708	0x4A10 0708
CONTROL_EFUSE_4	RW	32	0x0000070C	0x4A10 070C

**18.6.7 SYSCTRL\_PADCONF\_CORE Register Description****Table 18-316. CONTROL\_PADCONF\_CORE\_REVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Control module revision identifier Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 18-317. Register Call Summary for Register CONTROL\_PADCONF\_CORE\_REVISION**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-318. CONTROL\_PADCONF\_CORE\_HWINFO**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Information about the IP module hardware configuration that is, typically the module HDL generics (if any). Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP_HWINFO																															

Bits	Field Name	Description	Type	Reset
31:0	IP_HWINFO	IP-module dependent	R	0x0000 0000

**Table 18-319. Register Call Summary for Register CONTROL\_PADCONF\_CORE\_HWINFO**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-320. CONTROL\_PADCONF\_CORE\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IP_SYSCONFIG_IDLEMODE	RESERVED		

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:2	IP_SYSCONFIG_IDLEMODE	Select the local clock-gating strategy 0x0: Clock is manually gated 0x1: Clock is manually enabled 0x2,0x3: Clock is automatically gated when there is no access to the Control Module through L4-interconnect	RW	0x2
1:0	RESERVED		R	0x0

**Table 18-321. Register Call Summary for Register CONTROL\_PADCONF\_CORE\_SYSCONFIG**

Control Module Functional Description	<ul style="list-style-type: none"> <li>• <a href="#">Device Core Control Module Instance: [0]</a></li> </ul>
Control Module Programming Guide	<ul style="list-style-type: none"> <li>• <a href="#">Control Module Global Initialization: [1]</a></li> <li>• <a href="#">Off Mode Preliminary Settings: [2]</a></li> </ul>
Control Module Register Manual	<ul style="list-style-type: none"> <li>• <a href="#">SYSCTRL_PADCONF_CORE Register Summary: [3]</a></li> </ul>

**Table 18-322. CONTROL\_CORE\_PAD0\_GPMC\_AD0\_PAD1\_GPMC\_AD1**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ad0 and gpmc_ad1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_AD1_WAKEUPEVENT	GPMC_AD1_WAKEUPENABLE	GPMC_AD1_OFFMODEPULLTYPESELECT	GPMC_AD1_OFFMODEPULLUDENABLE	GPMC_AD1_OFFMODEOUTVALUE	GPMC_AD1_OFFMODEOUTENABLE	GPMC_AD1_OFFMODEENABLE	GPMC_AD1_INPUTENABLE	RESERVED			GPMC_AD1_PULLTYPESELECT	GPMC_AD1_PULLUDENABLE	GPMC_AD1_MUXMODE			GPMC_AD0_WAKEUPEVENT	GPMC_AD0_WAKEUPENABLE	GPMC_AD0_OFFMODEPULLTYPESELECT	GPMC_AD0_OFFMODEPULLUDENABLE	GPMC_AD0_OFFMODEOUTVALUE	GPMC_AD0_OFFMODEOUTENABLE	GPMC_AD0_OFFMODEENABLE	GPMC_AD0_INPUTENABLE	RESERVED			GPMC_AD0_PULLTYPESELECT	GPMC_AD0_PULLUDENABLE			GPMC_AD0_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPMC_AD1_WAKEUPEVENT	Pad_x wake-up event status latched in the I/O Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_AD1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_AD1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_AD1_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_AD1_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad1 0x0: Set value at 0 0x1: Set value at 1	RW	0



Bits	Field Name	Description	Type	Reset
26	GPMC_AD1_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_AD1_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_AD1_INPUTENABLE	Input enable value for pad gpmc_ad1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_AD1_PULLTYPESELECTION	pullup/down selection for pad gpmc_ad1 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_AD1_PULLUDENABLE	pullup/down enable for pad gpmc_ad1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_AD1_MUXMODE	Functional multiplexing selection for pad gpmc_ad1 0x0: Select gpmc_ad1 0x1: Select sdmmc2_dat1	RW	0x0
15	GPMC_AD0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_AD0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_AD0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_AD0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_AD0_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad0 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_AD0_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_AD0_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_AD0_INPUTENABLE	Input enable value for pad gpmc_ad0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	GPMC_AD0_PULLTYPESELECTION	pullup/down selection for pad gpmc_ad0 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_AD0_PULLUDENABLE	pullup/down enable for pad gpmc_ad0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_AD0_MUXMODE	Functional multiplexing selection for pad gpmc_ad0 0x0: Select gpmc_ad0 0x1: Select sdmmc2_dat0	RW	0x0

**Table 18-323. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_AD0\_PAD1\_GPMC\_AD1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-324. CONTROL\_CORE\_PAD0\_GPMC\_AD2\_PAD1\_GPMC\_AD3**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ad2 and gpmc_ad3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_AD3_WAKEUPEVENT	GPMC_AD3_WAKEUPENABLE	GPMC_AD3_OFFMODEPULLTYPESELECTION	GPMC_AD3_OFFMODEPULLUDENABLE	GPMC_AD3_OFFMODEOUTVALUE	GPMC_AD3_OFFMODEOUTENABLE	GPMC_AD3_OFFMODEENABLE	GPMC_AD3_INPUTENABLE	RESERVED			GPMC_AD3_PULLTYPESELECTION	GPMC_AD3_PULLUDENABLE	GPMC_AD3_MUXMODE			GPMC_AD2_WAKEUPEVENT	GPMC_AD2_WAKEUPENABLE	GPMC_AD2_OFFMODEPULLTYPESELECTION	GPMC_AD2_OFFMODEPULLUDENABLE	GPMC_AD2_OFFMODEOUTVALUE	GPMC_AD2_OFFMODEOUTENABLE	GPMC_AD2_OFFMODEENABLE	GPMC_AD2_INPUTENABLE	RESERVED			GPMC_AD2_PULLTYPESELECTION	GPMC_AD2_PULLUDENABLE	GPMC_AD2_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	GPMC_AD3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_AD3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_AD3_OFFMODEPULLTYPESELECTION	OffMode mode pullup/down selection for pad gpmc_ad3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
28	GPMC_AD3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_AD3_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad3 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_AD3_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_AD3_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_AD3_INPUTENABLE	Input enable value for pad gpmc_ad3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_AD3_PULLTYPESELECTION	pullup/down selection for pad gpmc_ad3 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_AD3_PULLUDENABLE	pullup/down enable for pad gpmc_ad3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_AD3_MUXMODE	Functional multiplexing selection for pad gpmc_ad3 0x0: Select gpmc_ad3 0x1: Select sdmmc2_dat3	RW	0x0
15	GPMC_AD2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_AD2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_AD2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_AD2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_AD2_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad2 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_AD2_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
9	GPMC_AD2_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_AD2_INPUTENABLE	Input enable value for pad gpmc_ad2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_AD2_PULLTYPESELECT	pullup/down selection for pad gpmc_ad2 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_AD2_PULLUDENABLE	pullup/down enable for pad gpmc_ad2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_AD2_MUXMODE	Functional multiplexing selection for pad gpmc_ad2 0x0: Select gpmc_ad2 0x1: Select sdmmc2_dat2	RW	0x0

**Table 18-325. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_AD2\_PAD1\_GPMC\_AD3**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-326. CONTROL\_CORE\_PAD0\_GPMC\_AD4\_PAD1\_GPMC\_AD5**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ad4 and gpmc_ad5 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
GPMC_AD5_WAKEUPEVENT	GPMC_AD5_WAKEUPENABLE	GPMC_AD5_OFFMODEPULLTYPESELECT	GPMC_AD5_OFFMODEPULLUDENABLE	GPMC_AD5_OFFMODEOUTVALUE	GPMC_AD5_OFFMODEOUTENABLE	GPMC_AD5_OFFMODEENABLE	GPMC_AD5_INPUTENABLE	RESERVED			GPMC_AD5_PULLTYPESELECT	GPMC_AD5_PULLUDENABLE			GPMC_AD5_MUXMODE			GPMC_AD4_WAKEUPEVENT	GPMC_AD4_WAKEUPENABLE	GPMC_AD4_OFFMODEPULLTYPESELECT	GPMC_AD4_OFFMODEPULLUDENABLE	GPMC_AD4_OFFMODEOUTVALUE	GPMC_AD4_OFFMODEOUTENABLE	GPMC_AD4_OFFMODEENABLE	GPMC_AD4_INPUTENABLE	RESERVED			GPMC_AD4_PULLTYPESELECT	GPMC_AD4_PULLUDENABLE			GPMC_AD4_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPMC_AD5_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_AD5_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_AD5_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad5 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_AD5_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad5 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_AD5_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad5 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_AD5_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad5. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_AD5_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad5 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_AD5_INPUTENABLE	Input enable value for pad gpmc_ad5 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_AD5_PULLTYPESELECT	pullup/down selection for pad gpmc_ad5 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_AD5_PULLUDENABLE	pullup/down enable for pad gpmc_ad5 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_AD5_MUXMODE	Functional multiplexing selection for pad gpmc_ad5 0x0: Select gpmc_ad5 0x1: Select sdmmc2_dat5 0x2: Select sdmmc2_dir_dat1	RW	0x0
15	GPMC_AD4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_AD4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_AD4_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad4 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	GPMC_AD4_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad4 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_AD4_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad4 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_AD4_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad4. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_AD4_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad4 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_AD4_INPUTENABLE	Input enable value for pad gpmc_ad4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_AD4_PULLTYPESELECTION	pullup/down selection for pad gpmc_ad4 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_AD4_PULLUDENABLE	pullup/down enable for pad gpmc_ad4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_AD4_MUXMODE	Functional multiplexing selection for pad gpmc_ad4 0x0: Select gpmc_ad4 0x1: Select sdmmc2_dat4 0x2: Select sdmmc2_dir_dat0	RW	0x0

**Table 18-327. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_AD4\_PAD1\_GPMC\_AD5**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-328. CONTROL\_CORE\_PAD0\_GPMC\_AD6\_PAD1\_GPMC\_AD7**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ad6 and gpmc_ad7 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_AD7_WAKEUPEVENT	GPMC_AD7_WAKEUPENABLE	GPMC_AD7_OFFMODEPULLTYPESELECT	GPMC_AD7_OFFMODEPULLUDENABLE	GPMC_AD7_OFFMODEOUTVALUE	GPMC_AD7_OFFMODEOUTENABLE	GPMC_AD7_OFFMODEENABLE	GPMC_AD7_INPUTENABLE	RESERVED			GPMC_AD7_PULLTYPESELECT	GPMC_AD7_PULLUDENABLE			GPMC_AD7_MUXMODE	GPMC_AD6_WAKEUPEVENT	GPMC_AD6_WAKEUPENABLE	GPMC_AD6_OFFMODEPULLTYPESELECT	GPMC_AD6_OFFMODEPULLUDENABLE	GPMC_AD6_OFFMODEOUTVALUE	GPMC_AD6_OFFMODEOUTENABLE	GPMC_AD6_OFFMODEENABLE	GPMC_AD6_INPUTENABLE	RESERVED			GPMC_AD6_PULLTYPESELECT	GPMC_AD6_PULLUDENABLE			GPMC_AD6_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPMC_AD7_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_AD7_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_AD7_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad7 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_AD7_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad7 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_AD7_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad7 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_AD7_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad7. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_AD7_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad7 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_AD7_INPUTENABLE	Input enable value for pad gpmc_ad7 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_AD7_PULLTYPESELECT	pullup/down selection for pad gpmc_ad7 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_AD7_PULLUDENABLE	pullup/down enable for pad gpmc_ad7 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
18:16	GPMC_AD7_MUXMODE	Functional multiplexing selection for pad gpmc_ad7 0x0: Select gpmc_ad7 0x1: Select sdmmc2_dat7 0x2: Select sdmmc2_clk_fdbk	RW	0x0
15	GPMC_AD6_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_AD6_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_AD6_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad6 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_AD6_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad6 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_AD6_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad6 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_AD6_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad6. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_AD6_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad6 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_AD6_INPUTENABLE	Input enable value for pad gpmc_ad6 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_AD6_PULLTYPESELECT	pullup/down selection for pad gpmc_ad6 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_AD6_PULLUDENABLE	pullup/down enable for pad gpmc_ad6 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_AD6_MUXMODE	Functional multiplexing selection for pad gpmc_ad6 0x0: Select gpmc_ad6 0x1: Select sdmmc2_dat6 0x2: Select sdmmc2_dir_cmd	RW	0x0

**Table 18-329. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_AD6\_PAD1\_GPMC\_AD7**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)



**Table 18-330. CONTROL\_CORE\_PAD0\_GPMC\_AD8\_PAD1\_GPMC\_AD9**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads gpmc_ad8 and gpmc_ad9 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_AD9_WAKEUPEVENT	GPMC_AD9_WAKEUPENABLE	GPMC_AD9_OFFMODEPULLTYPESELECT	GPMC_AD9_OFFMODEPULLUDENABLE	GPMC_AD9_OFFMODEOUTVALUE	GPMC_AD9_OFFMODEOUTENABLE	GPMC_AD9_OFFMODEENABLE	GPMC_AD9_INPUTENABLE	RESERVED			GPMC_AD9_PULLTYPESELECT	GPMC_AD9_PULLUDENABLE		GPMC_AD9_MUXMODE		GPMC_AD8_WAKEUPEVENT	GPMC_AD8_WAKEUPENABLE	GPMC_AD8_OFFMODEPULLTYPESELECT	GPMC_AD8_OFFMODEPULLUDENABLE	GPMC_AD8_OFFMODEOUTVALUE	GPMC_AD8_OFFMODEOUTENABLE	GPMC_AD8_OFFMODEENABLE	GPMC_AD8_INPUTENABLE	RESERVED			GPMC_AD8_PULLTYPESELECT	GPMC_AD8_PULLUDENABLE		GPMC_AD8_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	GPMC_AD9_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_AD9_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_AD9_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad9 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_AD9_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad9 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_AD9_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad9 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_AD9_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad9. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_AD9_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad9 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_AD9_INPUTENABLE	Input enable value for pad gpmc_ad9 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1

Bits	Field Name	Description	Type	Reset
23:21	RESERVED		R	0x0
20	GPMC_AD9_PULLTYPESELECTION	pullup/down selection for pad gpmc_ad9 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_AD9_PULLUDENABLE	pullup/down enable for pad gpmc_ad9 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_AD9_MUXMODE	Functional multiplexing selection for pad gpmc_ad9 0x0: Select gpmc_ad9 0x1: Select kpd_row1 0x2: Select c2c_data14 0x3: Select gpio_33 0x5: Select sdmmc1_dat1	RW	0x0
15	GPMC_AD8_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_AD8_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_AD8_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad8 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_AD8_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad8 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_AD8_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad8 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_AD8_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad8. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_AD8_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad8 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_AD8_INPUTENABLE	Input enable value for pad gpmc_ad8 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_AD8_PULLTYPESELECTION	pullup/down selection for pad gpmc_ad8 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_AD8_PULLUDENABLE	pullup/down enable for pad gpmc_ad8 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
2:0	GPMC_AD8_MUXMODE	Functional multiplexing selection for pad gpmc_ad8 0x0: Select gpmc_ad8 0x1: Select kpd_row0 0x2: Select c2c_data15 0x3: Select gpio_32 0x5: Select sdmmc1_dat0	RW	0x0

**Table 18-331. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_AD8\_PAD1\_GPMC\_AD9**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-332. CONTROL\_CORE\_PAD0\_GPMC\_AD10\_PAD1\_GPMC\_AD11**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ad10 and gpmc_ad11 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_AD11_WAKEUPEVENT	GPMC_AD11_WAKEUPENABLE	GPMC_AD11_OFFMODEPULLTYPESELECT	GPMC_AD11_OFFMODEPULLUDENENABLE	GPMC_AD11_OFFMODEOUTVALUE	GPMC_AD11_OFFMODEOUTENABLE	GPMC_AD11_OFFMODEENABLE	GPMC_AD11_INPUTENABLE	RESERVED			GPMC_AD11_PULLTYPESELECT	GPMC_AD11_PULLUDENENABLE	GPMC_AD11_MUXMODE			GPMC_AD10_WAKEUPEVENT	GPMC_AD10_WAKEUPENABLE	GPMC_AD10_OFFMODEPULLTYPESELECT	GPMC_AD10_OFFMODEPULLUDENENABLE	GPMC_AD10_OFFMODEOUTVALUE	GPMC_AD10_OFFMODEOUTENABLE	GPMC_AD10_OFFMODEENABLE	GPMC_AD10_INPUTENABLE	RESERVED			GPMC_AD10_PULLTYPESELECT	GPMC_AD10_PULLUDENENABLE	GPMC_AD10_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	GPMC_AD11_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_AD11_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_AD11_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad11 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
28	GPMC_AD11_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad11 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_AD11_OFFMODEOUTV ALUE	OffMode mode output value for pad gpmc_ad11 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_AD11_OFFMODEOUTE NABLE	OffMode mode output enable value for pad gpmc_ad11. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_AD11_OFFMODEENABL E	OffMode mode override control for pad gpmc_ad11 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_AD11_INPUTENABLE	Input enable value for pad gpmc_ad11 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_AD11_PULLTYPESELE CT	pullup/down selection for pad gpmc_ad11 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_AD11_PULLUDENABLE	pullup/down enable for pad gpmc_ad11 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_AD11_MUXMODE	Functional multiplexing selection for pad gpmc_ad11 0x0: Select gpmc_ad11 0x1: Select kpd_row3 0x2: Select c2c_data12 0x3: Select gpio_35 0x5: Select sdmmc1_dat3	RW	0x0
15	GPMC_AD10_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_AD10_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_AD10_OFFMODEPULLTYPESELE CT	OffMode mode pullup/down selection for pad gpmc_ad10 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_AD10_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad10 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_AD10_OFFMODEOUTV ALUE	OffMode mode output value for pad gpmc_ad10 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_AD10_OFFMODEOUTE NABLE	OffMode mode output enable value for pad gpmc_ad10. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
9	GPMC_AD10_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad10 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_AD10_INPUTENABLE	Input enable value for pad gpmc_ad10 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_AD10_PULLTYPESELECT	pullup/down selection for pad gpmc_ad10 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_AD10_PULLUDENABLE	pullup/down enable for pad gpmc_ad10 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_AD10_MUXMODE	Functional multiplexing selection for pad gpmc_ad10 0x0: Select gpmc_ad10 0x1: Select kpd_row2 0x2: Select c2c_data13 0x3: Select gpio_34 0x5: Select sdmmc1_dat2	RW	0x0

**Table 18-333. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_AD10\_PAD1\_GPMC\_AD11**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-334. CONTROL\_CORE\_PAD0\_GPMC\_AD12\_PAD1\_GPMC\_AD13**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ad12 and gpmc_ad13 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
GPMC_AD13_WAKEUPEVENT	GPMC_AD13_WAKEUPENABLE	GPMC_AD13_OFFMODEPULLTYPESELECT	GPMC_AD13_OFFMODEPULLUDENABLE	GPMC_AD13_OFFMODEOUTVALUE	GPMC_AD13_OFFMODEOUTENABLE	GPMC_AD13_OFFMODEENABLE	GPMC_AD13_INPUTENABLE	RESERVED	GPMC_AD13_PULLTYPESELECT	GPMC_AD13_PULLUDENABLE	GPMC_AD13_MUXMODE	GPMC_AD12_WAKEUPEVENT	GPMC_AD12_WAKEUPENABLE	GPMC_AD12_OFFMODEPULLTYPESELECT	GPMC_AD12_OFFMODEPULLUDENABLE	GPMC_AD12_OFFMODEOUTVALUE	GPMC_AD12_OFFMODEOUTENABLE	GPMC_AD12_OFFMODEENABLE	GPMC_AD12_INPUTENABLE	RESERVED	GPMC_AD12_PULLTYPESELECT	GPMC_AD12_PULLUDENABLE	GPMC_AD12_MUXMODE									

Bits	Field Name	Description	Type	Reset
31	GPMC_AD13_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_AD13_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_AD13_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad13 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_AD13_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad13 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_AD13_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad13 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_AD13_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad13. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_AD13_OFFMODEENABLE	OffMode mode override control for pad gpmc_ad13 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_AD13_INPUTENABLE	Input enable value for pad gpmc_ad13 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_AD13_PULLTYPESELECT	pullup/down selection for pad gpmc_ad13 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_AD13_PULLUDENABLE	pullup/down enable for pad gpmc_ad13 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_AD13_MUXMODE	Functional multiplexing selection for pad gpmc_ad13 0x0: Select gpmc_ad13 0x1: Select kpd_col1 0x2: Select c2c_data10 0x3: Select gpio_37 0x5: Select sdmmc1_dat5	RW	0x0
15	GPMC_AD12_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_AD12_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_AD12_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad12 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	GPMC_AD12_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad12 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_AD12_OFFMODEOUTV ALUE	OffMode mode output value for pad gpmc_ad12 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_AD12_OFFMODEOUTE NABLE	OffMode mode output enable value for pad gpmc_ad12. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_AD12_OFFMODEENABL E	OffMode mode override control for pad gpmc_ad12 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_AD12_INPUTENABLE	Input enable value for pad gpmc_ad12 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_AD12_PULLTYPESELE CT	pullup/down selection for pad gpmc_ad12 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_AD12_PULLUDENABLE	pullup/down enable for pad gpmc_ad12 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_AD12_MUXMODE	Functional multiplexing selection for pad gpmc_ad12 0x0: Select gpmc_ad12 0x1: Select kpd_col0 0x2: Select c2c_data11 0x3: Select gpio_36 0x5: Select sdmmc1_dat4	RW	0x0

**Table 18-335. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_AD12\_PAD1\_GPMC\_AD13**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-336. CONTROL\_CORE\_PAD0\_GPMC\_AD14\_PAD1\_GPMC\_AD15**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ad14 and gpmc_ad15 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_AD15_WAKEUPEVENT	GPMC_AD15_WAKEUPENABLE	GPMC_AD15_OFFMODEPULLTYPESELECT	GPMC_AD15_OFFMODEPULLUDENABLE	GPMC_AD15_OFFMODEOUTVALUE	GPMC_AD15_OFFMODEOUTENABLE	GPMC_AD15_OFFMODEEENABLE	GPMC_AD15_INPUTENABLE	RESERVED			GPMC_AD15_PULLTYPESELECT	GPMC_AD15_PULLUDENABLE	GPMC_AD15_MUXMODE			GPMC_AD14_WAKEUPEVENT	GPMC_AD14_WAKEUPENABLE	GPMC_AD14_OFFMODEPULLTYPESELECT	GPMC_AD14_OFFMODEPULLUDENABLE	GPMC_AD14_OFFMODEOUTVALUE	GPMC_AD14_OFFMODEOUTENABLE	GPMC_AD14_OFFMODEEENABLE	GPMC_AD14_INPUTENABLE	RESERVED			GPMC_AD14_PULLTYPESELECT	GPMC_AD14_PULLUDENABLE	GPMC_AD14_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	GPMC_AD15_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_AD15_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_AD15_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad15 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_AD15_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad15 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_AD15_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad15 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_AD15_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad15. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_AD15_OFFMODEEENABLE	OffMode mode override control for pad gpmc_ad15 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_AD15_INPUTENABLE	Input enable value for pad gpmc_ad15 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_AD15_PULLTYPESELECT	pullup/down selection for pad gpmc_ad15 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_AD15_PULLUDENABLE	pullup/down enable for pad gpmc_ad15 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1



Bits	Field Name	Description	Type	Reset
18:16	GPMC_AD15_MUXMODE	Functional multiplexing selection for pad gpmc_ad15 0x0: Select gpmc_ad15 0x1: Select kpd_col3 0x2: Select c2c_data8 0x3: Select gpio_39 0x5: Select sdmmc1_dat7	RW	0x0
15	GPMC_AD14_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_AD14_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_AD14_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ad14 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_AD14_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ad14 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_AD14_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ad14 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_AD14_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ad14. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_AD14_OFFMODEENABLER	OffMode mode override control for pad gpmc_ad14 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_AD14_INPUTENABLE	Input enable value for pad gpmc_ad14 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_AD14_PULLTYPESELECT	pullup/down selection for pad gpmc_ad14 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_AD14_PULLUDENABLE	pullup/down enable for pad gpmc_ad14 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_AD14_MUXMODE	Functional multiplexing selection for pad gpmc_ad14 0x0: Select gpmc_ad14 0x1: Select kpd_col2 0x2: Select c2c_data9 0x3: Select gpio_38 0x5: Select sdmmc1_dat6	RW	0x0

**Table 18-337. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_AD14\_PAD1\_GPMC\_AD15**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-338. CONTROL\_CORE\_PAD0\_GPMC\_A16\_PAD1\_GPMC\_A17**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_a16 and gpmc_a17 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_A17_WAKEUPEVENT	GPMC_A17_WAKEUPENABLE	GPMC_A17_OFFMODEPULLTYPESELECT	GPMC_A17_OFFMODEPULLUDENABLE	GPMC_A17_OFFMODEOUTVALUE	GPMC_A17_OFFMODEOUTENABLE	GPMC_A17_OFFMODEENABLE	GPMC_A17_INPUTENABLE	RESERVED			GPMC_A17_PULLTYPESELECT	GPMC_A17_PULLUDENABLE		GPMC_A17_MUXMODE		GPMC_A16_WAKEUPEVENT	GPMC_A16_WAKEUPENABLE	GPMC_A16_OFFMODEPULLTYPESELECT	GPMC_A16_OFFMODEPULLUDENABLE	GPMC_A16_OFFMODEOUTVALUE	GPMC_A16_OFFMODEOUTENABLE	GPMC_A16_OFFMODEENABLE	GPMC_A16_INPUTENABLE	RESERVED			GPMC_A16_PULLTYPESELECT	GPMC_A16_PULLUDENABLE		GPMC_A16_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	GPMC_A17_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_A17_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_A17_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a17 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_A17_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_a17 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_A17_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_a17 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_A17_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_a17. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
25	GPMC_A17_OFFMODEENABLE	OffMode mode override control for pad gpmc_a17 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_A17_INPUTENABLE	Input enable value for pad gpmc_a17 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_A17_PULLTYPESELECT	pullup/down selection for pad gpmc_a17 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_A17_PULLUDENABLE	pullup/down enable for pad gpmc_a17 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_A17_MUXMODE	Functional multiplexing selection for pad gpmc_a17 0x0: Select gpmc_a17 0x1: Select kpd_row5 0x2: Select c2c_datain1 0x3: Select gpio_41 0x4: Select venc_656_data1 0x7: Select safe_mode	RW	0x7
15	GPMC_A16_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_A16_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_A16_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a16 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_A16_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_a16 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_A16_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_a16 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_A16_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_a16. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_A16_OFFMODEENABLE	OffMode mode override control for pad gpmc_a16 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_A16_INPUTENABLE	Input enable value for pad gpmc_a16 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	GPMC_A16_PULLTYPESELECT	pullup/down selection for pad gpmc_a16 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_A16_PULLUDENABLE	pullup/down enable for pad gpmc_a16 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_A16_MUXMODE	Functional multiplexing selection for pad gpmc_a16 0x0: Select gpmc_a16 0x1: Select kpd_row4 0x2: Select c2c_datain0 0x3: Select gpio_40 0x4: Select venc_656_data0 0x7: Select safe_mode	RW	0x7

**Table 18-339. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_A16\_PAD1\_GPMC\_A17**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-340. CONTROL\_CORE\_PAD0\_GPMC\_A18\_PAD1\_GPMC\_A19**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_a18 and gpmc_a19 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_A19_WAKEUPEVENT	GPMC_A19_WAKEUPENABLE	GPMC_A19_OFFMODEPULLTYPESELECT	GPMC_A19_OFFMODEPULLUDENABLE	GPMC_A19_OFFMODEOUTVALUE	GPMC_A19_OFFMODEOUTENABLE	GPMC_A19_OFFMODEEENABLE	GPMC_A19_INPUTENABLE	RESERVED		GPMC_A19_PULLTYPESELECT	GPMC_A19_PULLUDENABLE		GPMC_A19_MUXMODE		GPMC_A18_WAKEUPEVENT	GPMC_A18_WAKEUPENABLE	GPMC_A18_OFFMODEPULLTYPESELECT	GPMC_A18_OFFMODEPULLUDENABLE	GPMC_A18_OFFMODEOUTVALUE	GPMC_A18_OFFMODEOUTENABLE	GPMC_A18_OFFMODEEENABLE	GPMC_A18_INPUTENABLE	RESERVED		GPMC_A18_PULLTYPESELECT	GPMC_A18_PULLUDENABLE		GPMC_A18_MUXMODE			

Bits	Field Name	Description	Type	Reset
31	GPMC_A19_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
30	GPMC_A19_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_A19_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a19 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_A19_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_a19 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_A19_OFFMODEOUTVAL UE	OffMode mode output value for pad gpmc_a19 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_A19_OFFMODEOUTEN ABLE	OffMode mode output enable value for pad gpmc_a19. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_A19_OFFMODEENABLE	OffMode mode override control for pad gpmc_a19 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_A19_INPUTENABLE	Input enable value for pad gpmc_a19 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_A19_PULLTYPESELECT	pullup/down selection for pad gpmc_a19 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_A19_PULLUDENABLE	pullup/down enable for pad gpmc_a19 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_A19_MUXMODE	Functional multiplexing selection for pad gpmc_a19 0x0: Select gpmc_a19 0x1: Select kpd_row7 0x2: Select c2c_datain3 0x3: Select gpio_43 0x4: Select venc_656_data3 0x7: Select safe_mode	RW	0x7
15	GPMC_A18_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_A18_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_A18_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a18 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	GPMC_A18_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_a18 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_A18_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_a18 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_A18_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_a18. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_A18_OFFMODEENABLE	OffMode mode override control for pad gpmc_a18 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_A18_INPUTENABLE	Input enable value for pad gpmc_a18 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_A18_PULLTYPESELECT	pullup/down selection for pad gpmc_a18 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_A18_PULLUDENABLE	pullup/down enable for pad gpmc_a18 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_A18_MUXMODE	Functional multiplexing selection for pad gpmc_a18 0x0: Select gpmc_a18 0x1: Select kpd_row6 0x2: Select c2c_datain2 0x3: Select gpio_42 0x4: Select venc_656_data2 0x7: Select safe_mode	RW	0x7

**Table 18-341. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_A18\_PAD1\_GPMC\_A19**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-342. CONTROL\_CORE\_PAD0\_GPMC\_A20\_PAD1\_GPMC\_A21**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads gpmc_a20 and gpmc_a21 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
GPMC_A21_WAKEUPEVENT	GPMC_A21_WAKEUPENABLE	GPMC_A21_OFFMODEPULLTYPESELECT	GPMC_A21_OFFMODEPULLUDENABE	GPMC_A21_OFFMODEOUTVALUE	GPMC_A21_OFFMODEOUTENABLE	GPMC_A21_OFFMODEENABLE	GPMC_A21_INPUTENABLE	RESERVED			GPMC_A21_PULLTYPESELECT	GPMC_A21_PULLUDENABE			GPMC_A21_MUXMODE			GPMC_A20_WAKEUPEVENT	GPMC_A20_WAKEUPENABLE	GPMC_A20_OFFMODEPULLTYPESELECT	GPMC_A20_OFFMODEPULLUDENABE	GPMC_A20_OFFMODEOUTVALUE	GPMC_A20_OFFMODEOUTENABLE	GPMC_A20_OFFMODEENABLE	GPMC_A20_INPUTENABLE	RESERVED			GPMC_A20_PULLTYPESELECT	GPMC_A20_PULLUDENABE			GPMC_A20_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPMC_A21_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_A21_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_A21_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a21 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_A21_OFFMODEPULLUDENABE	OffMode mode pullup/down enable for pad gpmc_a21 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_A21_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_a21 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_A21_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_a21. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_A21_OFFMODEENABLE	OffMode mode override control for pad gpmc_a21 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_A21_INPUTENABLE	Input enable value for pad gpmc_a21 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1

Bits	Field Name	Description	Type	Reset
23:21	RESERVED		R	0x0
20	GPMC_A21_PULLTYPESELECT	pullup/down selection for pad gpmc_a21 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_A21_PULLUDENABLE	pullup/down enable for pad gpmc_a21 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_A21_MUXMODE	Functional multiplexing selection for pad gpmc_a21 0x0: Select gpmc_a21 0x1: Select kpd_col5 0x2: Select c2c_datain5 0x3: Select gpio_45 0x4: Select venc_656_data5 0x7: Select safe_mode	RW	0x7
15	GPMC_A20_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_A20_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_A20_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a20 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_A20_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_a20 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_A20_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_a20 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_A20_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_a20. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_A20_OFFMODEENABLE	OffMode mode override control for pad gpmc_a20 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_A20_INPUTENABLE	Input enable value for pad gpmc_a20 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_A20_PULLTYPESELECT	pullup/down selection for pad gpmc_a20 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_A20_PULLUDENABLE	pullup/down enable for pad gpmc_a20 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1



Bits	Field Name	Description	Type	Reset
2:0	GPMC_A20_MUXMODE	Functional multiplexing selection for pad gpmc_a20 0x0: Select gpmc_a20 0x1: Select kpd_col4 0x2: Select c2c_datain4 0x3: Select gpio_44 0x4: Select venc_656_data4 0x7: Select safe_mode	RW	0x7

**Table 18-343. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_A20\_PAD1\_GPMC\_A21**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-344. CONTROL\_CORE\_PAD0\_GPMC\_A22\_PAD1\_GPMC\_A23**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_a22 and gpmc_a23 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_A23_WAKEUPEVENT	GPMC_A23_WAKEUPENABLE	GPMC_A23_OFFMODEPULLTYPESELECT	GPMC_A23_OFFMODEPULLUDENABLE	GPMC_A23_OFFMODEOUTVALUE	GPMC_A23_OFFMODEOUTENABLE	GPMC_A23_OFFMODEENABLE	GPMC_A23_INPUTENABLE	RESERVED			GPMC_A23_PULLTYPESELECT	GPMC_A23_PULLUDENABLE	GPMC_A23_MUXMODE			GPMC_A22_WAKEUPEVENT	GPMC_A22_WAKEUPENABLE	GPMC_A22_OFFMODEPULLTYPESELECT	GPMC_A22_OFFMODEPULLUDENABLE	GPMC_A22_OFFMODEOUTVALUE	GPMC_A22_OFFMODEOUTENABLE	GPMC_A22_OFFMODEENABLE	GPMC_A22_INPUTENABLE	RESERVED			GPMC_A22_PULLTYPESELECT	GPMC_A22_PULLUDENABLE			GPMC_A22_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPMC_A23_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_A23_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_A23_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a23 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
28	GPMC_A23_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_a23 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_A23_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_a23 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_A23_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_a23. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_A23_OFFMODEENABLE	OffMode mode override control for pad gpmc_a23 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_A23_INPUTENABLE	Input enable value for pad gpmc_a23 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_A23_PULLTYPESELECT	pullup/down selection for pad gpmc_a23 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_A23_PULLUDENABLE	pullup/down enable for pad gpmc_a23 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_A23_MUXMODE	Functional multiplexing selection for pad gpmc_a23 0x0: Select gpmc_a23 0x1: Select kpd_col7 0x2: Select c2c_datain7 0x3: Select gpio_47 0x4: Select venc_656_data7 0x7: Select safe_mode	RW	0x7
15	GPMC_A22_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_A22_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_A22_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a22 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_A22_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_a22 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_A22_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_a22 0x0: Set value at 0 0x1: Set value at 1	RW	0

Bits	Field Name	Description	Type	Reset
10	GPMC_A22_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_a22. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_A22_OFFMODEENABLE	OffMode mode override control for pad gpmc_a22 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_A22_INPUTENABLE	Input enable value for pad gpmc_a22 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_A22_PULLTYPESELECT	pullup/down selection for pad gpmc_a22 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_A22_PULLUDENABLE	pullup/down enable for pad gpmc_a22 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_A22_MUXMODE	Functional multiplexing selection for pad gpmc_a22 0x0: Select gpmc_a22 0x1: Select kpd_col6 0x2: Select c2c_datain6 0x3: Select gpio_46 0x4: Select venc_656_data6 0x7: Select safe_mode	RW	0x7

**Table 18-345. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_A22\_PAD1\_GPMC\_A23**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-346. CONTROL\_CORE\_PAD0\_GPMC\_A24\_PAD1\_GPMC\_A25**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_a24 and gpmc_a25 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_A25_WAKEUPEVENT	GPMC_A25_WAKEUPENABLE	GPMC_A25_OFFMODEPULLTYPESELECT	GPMC_A25_OFFMODEPULLUDENABE	GPMC_A25_OFFMODEOUTVALUE	GPMC_A25_OFFMODEOUTENABLE	GPMC_A25_OFFMODEENABLE	GPMC_A25_INPUTENABLE	RESERVED			GPMC_A25_PULLTYPESELECT	GPMC_A25_PULLUDENABE	GPMC_A25_MUXMODE			GPMC_A24_WAKEUPEVENT	GPMC_A24_WAKEUPENABLE	GPMC_A24_OFFMODEPULLTYPESELECT	GPMC_A24_OFFMODEPULLUDENABE	GPMC_A24_OFFMODEOUTVALUE	GPMC_A24_OFFMODEOUTENABLE	GPMC_A24_OFFMODEENABLE	GPMC_A24_INPUTENABLE	RESERVED			GPMC_A24_PULLTYPESELECT	GPMC_A24_PULLUDENABE	GPMC_A24_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	GPMC_A25_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_A25_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_A25_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a25 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_A25_OFFMODEPULLUDENABE	OffMode mode pullup/down enable for pad gpmc_a25 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_A25_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_a25 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_A25_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_a25. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_A25_OFFMODEENABLE	OffMode mode override control for pad gpmc_a25 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_A25_INPUTENABLE	Input enable value for pad gpmc_a25 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_A25_PULLTYPESELECT	pullup/down selection for pad gpmc_a25 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_A25_PULLUDENABE	pullup/down enable for pad gpmc_a25 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
18:16	GPMC_A25_MUXMODE	Functional multiplexing selection for pad gpmc_a25 0x0: Select gpmc_a25 0x2: Select c2c_clkout1 0x3: Select gpio_49 0x7: Select safe_mode	RW	0x7
15	GPMC_A24_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_A24_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_A24_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_a24 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_A24_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_a24 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_A24_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_a24 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_A24_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_a24. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_A24_OFFMODEENABLE	OffMode mode override control for pad gpmc_a24 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_A24_INPUTENABLE	Input enable value for pad gpmc_a24 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_A24_PULLTYPESELECT	pullup/down selection for pad gpmc_a24 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_A24_PULLUDENABLE	pullup/down enable for pad gpmc_a24 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_A24_MUXMODE	Functional multiplexing selection for pad gpmc_a24 0x0: Select gpmc_a24 0x1: Select kpd_col8 0x2: Select c2c_clkout0 0x3: Select gpio_48 0x7: Select safe_mode	RW	0x7

**Table 18-347. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_A24\_PAD1\_GPMC\_A25**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-348. CONTROL\_CORE\_PAD0\_GPMC\_NCS0\_PAD1\_GPMC\_NCS1**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ncs0 and gpmc_ncs1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NCS1_WAKEUPEVENT	GPMC_NCS1_WAKEUPENABLE	GPMC_NCS1_OFFMODEPULLTYPESELECT	GPMC_NCS1_OFFMODEPULLUDENABLE	GPMC_NCS1_OFFMODEOUTVALUE	GPMC_NCS1_OFFMODEOUTENABLE	GPMC_NCS1_OFFMODEENABLE	GPMC_NCS1_INPUENABLE	RESERVED			GPMC_NCS1_PULLTYPESELECT	GPMC_NCS1_PULLUDENABLE		GPMC_NCS1_MUXMODE		GPMC_NCS0_WAKEUPEVENT	GPMC_NCS0_WAKEUPENABLE	GPMC_NCS0_OFFMODEPULLTYPESELECT	GPMC_NCS0_OFFMODEPULLUDENABLE	GPMC_NCS0_OFFMODEOUTVALUE	GPMC_NCS0_OFFMODEOUTENABLE	GPMC_NCS0_OFFMODEENABLE	GPMC_NCS0_INPUENABLE	RESERVED			GPMC_NCS0_PULLTYPESELECT	GPMC_NCS0_PULLUDENABLE		GPMC_NCS0_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	GPMC_NCS1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_NCS1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_NCS1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ncs1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_NCS1_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ncs1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_NCS1_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ncs1 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_NCS1_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ncs1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
25	GPMC_NCS1_OFFMODEENABLE	OffMode mode override control for pad gpmc_ncs1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_NCS1_INPUTENABLE	Input enable value for pad gpmc_ncs1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_NCS1_PULLTYPESELECT	pullup/down selection for pad gpmc_ncs1 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_NCS1_PULLUDENABLE	pullup/down enable for pad gpmc_ncs1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_NCS1_MUXMODE	Functional multiplexing selection for pad gpmc_ncs1 0x0: Select gpmc_ncs1 0x2: Select c2c_dataout6 0x3: Select gpio_51 0x7: Select safe_mode	RW	0x7
15	GPMC_NCS0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_NCS0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_NCS0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ncs0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_NCS0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ncs0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_NCS0_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ncs0 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_NCS0_OFFMODEOUTPUTENABLE	OffMode mode output enable value for pad gpmc_ncs0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_NCS0_OFFMODEENABLE	OffMode mode override control for pad gpmc_ncs0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_NCS0_INPUTENABLE	Input enable value for pad gpmc_ncs0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	GPMC_NCS0_PULLTYPESELE CT	pullup/down selection for pad gpmc_ncs0 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_NCS0_PULLUDENABLE	pullup/down enable for pad gpmc_ncs0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_NCS0_MUXMODE	Functional multiplexing selection for pad gpmc_ncs0 0x0: Select gpmc_ncs0 0x3: Select gpio_50 0x4: Select sys_ndmareq0	RW	0x0

**Table 18-349. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_NCS0\_PAD1\_GPMC\_NCS1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-350. CONTROL\_CORE\_PAD0\_GPMC\_NCS2\_PAD1\_GPMC\_NCS3**

<b>Address Offset</b>	0x0000 0078	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ncs2 and gpmc_ncs3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
GPMC_NCS3_WAKEUPEVENT	GPMC_NCS3_WAKEUPENABLE	GPMC_NCS3_OFFMODEPULLTYPESELECT	GPMC_NCS3_OFFMODEPULLUDENABLE	GPMC_NCS3_OFFMODEOUTVALUE	GPMC_NCS3_OFFMODEOUTENABLE	GPMC_NCS3_OFFMODEENABLE	GPMC_NCS3_INPUTENABLE	RESERVED	GPMC_NCS3_PULLTYPESELECT	GPMC_NCS3_PULLUDENABLE	GPMC_NCS3_MUXMODE	GPMC_NCS2_WAKEUPEVENT	GPMC_NCS2_WAKEUPENABLE	GPMC_NCS2_OFFMODEPULLTYPESELECT	GPMC_NCS2_OFFMODEPULLUDENABLE	GPMC_NCS2_OFFMODEOUTVALUE	GPMC_NCS2_OFFMODEOUTENABLE	GPMC_NCS2_OFFMODEENABLE	GPMC_NCS2_INPUTENABLE	RESERVED	GPMC_NCS2_PULLTYPESELECT	GPMC_NCS2_PULLUDENABLE	GPMC_NCS2_MUXMODE									

Bits	Field Name	Description	Type	Reset
31	GPMC_NCS3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_NCS3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0



Bits	Field Name	Description	Type	Reset
29	GPMC_NCS3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ncs3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_NCS3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ncs3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_NCS3_OFFMODEOUTV ALUE	OffMode mode output value for pad gpmc_ncs3 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_NCS3_OFFMODEOUTE NABLE	OffMode mode output enable value for pad gpmc_ncs3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_NCS3_OFFMODEENAB LE	OffMode mode override control for pad gpmc_ncs3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_NCS3_INPUTENABLE	Input enable value for pad gpmc_ncs3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_NCS3_PULLTYPESELE CT	pullup/down selection for pad gpmc_ncs3 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_NCS3_PULLUDENABLE	pullup/down enable for pad gpmc_ncs3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_NCS3_MUXMODE	Functional multiplexing selection for pad gpmc_ncs3 0x0: Select gpmc_ncs3 0x1: Select gpmc_dir 0x2: Select c2c_dataout4 0x3: Select gpio_53 0x7: Select safe_mode	RW	0x7
15	GPMC_NCS2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_NCS2_WAKEUPENABL E	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_NCS2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ncs2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_NCS2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ncs2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_NCS2_OFFMODEOUTV ALUE	OffMode mode output value for pad gpmc_ncs2 0x0: Set value at 0 0x1: Set value at 1	RW	0

Bits	Field Name	Description	Type	Reset
10	GPMC_NCS2_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ncs2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_NCS2_OFFMODEENABLE	OffMode mode override control for pad gpmc_ncs2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_NCS2_INPUTENABLE	Input enable value for pad gpmc_ncs2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_NCS2_PULLTYPESELECTION	pullup/down selection for pad gpmc_ncs2 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_NCS2_PULLUDENABLE	pullup/down enable for pad gpmc_ncs2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_NCS2_MUXMODE	Functional multiplexing selection for pad gpmc_ncs2 0x0: Select gpmc_ncs2 0x1: Select kpd_row8 0x2: Select c2c_dataout7 0x3: Select gpio_52 0x7: Select safe_mode	RW	0x7

**Table 18-351. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_NCS2\_PAD1\_GPMC\_NCS3**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-352. CONTROL\_CORE\_PAD0\_GPMC\_NWP\_PAD1\_GPMC\_CLK**

<b>Address Offset</b>	0x0000 007C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_nwp and gpmc_clk Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
GPMC_CLK_WAKEUPEVENT	GPMC_CLK_WAKEUPENABLE	GPMC_CLK_OFFMODEPULLTYPESELECT	GPMC_CLK_OFFMODEPULLUDENABE	GPMC_CLK_OFFMODEOUTVALUE	GPMC_CLK_OFFMODEOUTENABLE	GPMC_CLK_OFFMODEENABLE	GPMC_CLK_INPUTENABLE	RESERVED			GPMC_CLK_PULLTYPESELECT	GPMC_CLK_PULLUDENABE			GPMC_CLK_MUXMODE			GPMC_NWP_WAKEUPEVENT	GPMC_NWP_WAKEUPENABLE	GPMC_NWP_OFFMODEPULLTYPESELECT	GPMC_NWP_OFFMODEPULLUDENABE	GPMC_NWP_OFFMODEOUTVALUE	GPMC_NWP_OFFMODEOUTENABLE	GPMC_NWP_OFFMODEENABLE	GPMC_NWP_INPUTENABLE	RESERVED			GPMC_NWP_PULLTYPESELECT	GPMC_NWP_PULLUDENABE			GPMC_NWP_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPMC_CLK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_CLK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_CLK_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_clk 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_CLK_OFFMODEPULLUDENABE	OffMode mode pullup/down enable for pad gpmc_clk 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_CLK_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_clk 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_CLK_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_clk. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_CLK_OFFMODEENABLE	OffMode mode override control for pad gpmc_clk 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_CLK_INPUTENABLE	Input enable value for pad gpmc_clk 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_CLK_PULLTYPESELECT	pullup/down selection for pad gpmc_clk 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_CLK_PULLUDENABE	pullup/down enable for pad gpmc_clk 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
18:16	GPMC_CLK_MUXMODE	Functional multiplexing selection for pad gpmc_clk 0x0: Select gpmc_clk 0x3: Select gpio_55 0x4: Select sys_ndmareq2 0x5: Select sdmmc1_cmd	RW	0x0
15	GPMC_NWP_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_NWP_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_NWP_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_nwp 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_NWP_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_nwp 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_NWP_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_nwp 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_NWP_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_nwp. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_NWP_OFFMODEENABLE	OffMode mode override control for pad gpmc_nwp 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_NWP_INPUTENABLE	Input enable value for pad gpmc_nwp 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_NWP_PULLTYPESELECT	pullup/down selection for pad gpmc_nwp 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_NWP_PULLUDENABLE	pullup/down enable for pad gpmc_nwp 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_NWP_MUXMODE	Functional multiplexing selection for pad gpmc_nwp 0x0: Select gpmc_nwp 0x1: Select dsi1_te0 0x3: Select gpio_54 0x4: Select sys_ndmareq1	RW	0x0

**Table 18-353. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_NWP\_PAD1\_GPMC\_CLK**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

**Table 18-353. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_NWP\_PAD1\_GPMC\_CLK (continued)**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-354. CONTROL\_CORE\_PAD0\_GPMC\_NADV\_ALE\_PAD1\_GPMC\_NOE**

<b>Address Offset</b>	0x0000 0080	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_nadv_ale and gpmc_noe Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NOE_WAKEUPEVENT	GPMC_NOE_WAKEUPENABLE	GPMC_NOE_OFFMODEPULLTYPESELECT	GPMC_NOE_OFFMODEPULLUDENABLE	GPMC_NOE_OFFMODEOUTVALUE	GPMC_NOE_OFFMODEOUTENABLE	GPMC_NOE_OFFMODEENABLE	GPMC_NOE_INPUTENABLE	RESERVED			GPMC_NOE_PULLTYPESELECT	GPMC_NOE_PULLUDENABLE	GPMC_NOE_MUXMODE			GPMC_NADV_ALE_WAKEUPEVENT	GPMC_NADV_ALE_WAKEUPENABLE	GPMC_NADV_ALE_OFFMODEPULLTYPESELECT	GPMC_NADV_ALE_OFFMODEPULLUDENABLE	GPMC_NADV_ALE_OFFMODEOUTVALUE	GPMC_NADV_ALE_OFFMODEOUTENABLE	GPMC_NADV_ALE_OFFMODEENABLE	GPMC_NADV_ALE_INPUTENABLE	RESERVED			GPMC_NADV_ALE_PULLTYPESELECT	GPMC_NADV_ALE_PULLUDENABLE	GPMC_NADV_ALE_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	GPMC_NOE_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_NOE_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_NOE_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_noe 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_NOE_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_noe 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_NOE_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_noe 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_NOE_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_noe. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
25	GPMC_NOE_OFFMODEENABLE	OffMode mode override control for pad gpmc_noe 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_NOE_INPUTENABLE	Input enable value for pad gpmc_noe 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_NOE_PULLTYPESELECT	pullup/down selection for pad gpmc_noe 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_NOE_PULLUDENABLE	pullup/down enable for pad gpmc_noe 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_NOE_MUXMODE	Functional multiplexing selection for pad gpmc_noe 0x0: Select gpmc_noe 0x1: Select sdmmc2_clk	RW	0x0
15	GPMC_NADV_ALE_WAKEUPVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_NADV_ALE_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_NADV_ALE_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_nadv_ale 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_NADV_ALE_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_nadv_ale 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_NADV_ALE_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_nadv_ale 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_NADV_ALE_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_nadv_ale. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_NADV_ALE_OFFMODEENABLE	OffMode mode override control for pad gpmc_nadv_ale 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_NADV_ALE_INPUTENABLE	Input enable value for pad gpmc_nadv_ale 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_NADV_ALE_PULLTYPESELECT	pullup/down selection for pad gpmc_nadv_ale 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	GPMC_NADV_ALE_PULLUDEN ABLE	pullup/down enable for pad gpmc_nadv_ale 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_NADV_ALE_MUXMODE	Functional multiplexing selection for pad gpmc_nadv_ale 0x0: Select gpmc_nadv_ale 0x1: Select dsi1_te1 0x3: Select gpio_56 0x4: Select sys_ndmareq3 0x5: Select sdmmc1_clk	RW	0x0

**Table 18-355. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_NADV\_ALE\_PAD1\_GPMC\_NOE**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-356. CONTROL\_CORE\_PAD0\_GPMC\_NWE\_PAD1\_GPMC\_NBE0\_CLE**

<b>Address Offset</b>	0x0000 0084	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_nwe and gpmc_nbe0_cle Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NBE0_CLE_WAKEUPEVENT	GPMC_NBE0_CLE_WAKEUPENABLE	GPMC_NBE0_CLE_OFFMODEPULLTYPESELECT	GPMC_NBE0_CLE_OFFMODEPULLUDENENABLE	GPMC_NBE0_CLE_OFFMODEOUTVALUE	GPMC_NBE0_CLE_OFFMODEOUTENABLE	GPMC_NBE0_CLE_OFFMODEENABLE	GPMC_NBE0_CLE_INPUTENABLE	RESERVED			GPMC_NBE0_CLE_PULLTYPESELECT	GPMC_NBE0_CLE_PULLUDENENABLE	GPMC_NBE0_CLE_MUXMODE			GPMC_NWE_WAKEUPEVENT	GPMC_NWE_WAKEUPENABLE	GPMC_NWE_OFFMODEPULLTYPESELECT	GPMC_NWE_OFFMODEPULLUDENENABLE	GPMC_NWE_OFFMODEOUTVALUE	GPMC_NWE_OFFMODEOUTENABLE	GPMC_NWE_OFFMODEENABLE	GPMC_NWE_INPUTENABLE	RESERVED			GPMC_NWE_PULLTYPESELECT	GPMC_NWE_PULLUDENENABLE	GPMC_NWE_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	GPMC_NBE0_CLE_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_NBE0_CLE_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
29	GPMC_NBE0_CLE_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_nbe0_cle 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_NBE0_CLE_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_nbe0_cle 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_NBE0_CLE_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_nbe0_cle 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_NBE0_CLE_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_nbe0_cle. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_NBE0_CLE_OFFMODEENABLE	OffMode mode override control for pad gpmc_nbe0_cle 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_NBE0_CLE_INPUTENABLE	Input enable value for pad gpmc_nbe0_cle 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_NBE0_CLE_PULLTYPESELECT	pullup/down selection for pad gpmc_nbe0_cle 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_NBE0_CLE_PULLUDENABLE	pullup/down enable for pad gpmc_nbe0_cle 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_NBE0_CLE_MUXMODE	Functional multiplexing selection for pad gpmc_nbe0_cle 0x0: Select gpmc_nbe0_cle 0x1: Select dsi2_te0 0x3: Select gpio_59	RW	0x0
15	GPMC_NWE_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_NWE_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_NWE_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_nwe 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_NWE_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_nwe 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_NWE_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_nwe 0x0: Set value at 0 0x1: Set value at 1	RW	0



Bits	Field Name	Description	Type	Reset
10	GPMC_NWE_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_nwe. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_NWE_OFFMODEENABLE	OffMode mode override control for pad gpmc_nwe 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_NWE_INPUTENABLE	Input enable value for pad gpmc_nwe 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_NWE_PULLTYPESELECTION	pullup/down selection for pad gpmc_nwe 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_NWE_PULLUDENABLE	pullup/down enable for pad gpmc_nwe 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_NWE_MUXMODE	Functional multiplexing selection for pad gpmc_nwe 0x0: Select gpmc_nwe 0x1: Select sdmmc2_cmd	RW	0x0

**Table 18-357. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_NWE\_PAD1\_GPMC\_NBE0\_CLE**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-358. CONTROL\_CORE\_PAD0\_GPMC\_NBE1\_PAD1\_GPMC\_WAIT0**

<b>Address Offset</b>	0x0000 0088		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads gpmc_nbe1 and gpmc_wait0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_WAIT0_WAKEUPEVENT	GPMC_WAIT0_WAKEUPENABLE	GPMC_WAIT0_OFFMODEPULLTYPESELECT	GPMC_WAIT0_OFFMODEPULLUDENENABLE	GPMC_WAIT0_OFFMODEOUTVALUE	GPMC_WAIT0_OFFMODEOUTENABLE	GPMC_WAIT0_OFFMODEEENABLE	GPMC_WAIT0_INPUTENABLE	RESERVED			GPMC_WAIT0_PULLTYPESELECT	GPMC_WAIT0_PULLUDENENABLE		GPMC_WAIT0_MUXMODE		GPMC_NBE1_WAKEUPEVENT	GPMC_NBE1_WAKEUPENABLE	GPMC_NBE1_OFFMODEPULLTYPESELECT	GPMC_NBE1_OFFMODEPULLUDENENABLE	GPMC_NBE1_OFFMODEOUTVALUE	GPMC_NBE1_OFFMODEOUTENABLE	GPMC_NBE1_OFFMODEEENABLE	GPMC_NBE1_INPUTENABLE	RESERVED			GPMC_NBE1_PULLTYPESELECT	GPMC_NBE1_PULLUDENENABLE		GPMC_NBE1_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	GPMC_WAIT0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_WAIT0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_WAIT0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_wait0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_WAIT0_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad gpmc_wait0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_WAIT0_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_wait0 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_WAIT0_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_wait0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_WAIT0_OFFMODEEENABLE	OffMode mode override control for pad gpmc_wait0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_WAIT0_INPUTENABLE	Input enable value for pad gpmc_wait0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_WAIT0_PULLTYPESELECT	pullup/down selection for pad gpmc_wait0 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPMC_WAIT0_PULLUDENENABLE	pullup/down enable for pad gpmc_wait0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
18:16	GPMC_WAIT0_MUXMODE	Functional multiplexing selection for pad gpmc_wait0 0x0: Select gpmc_wait0 0x1: Select dsi2_te1 0x3: Select gpio_61	RW	0x0
15	GPMC_NBE1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_NBE1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_NBE1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_nbe1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_NBE1_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_nbe1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_NBE1_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_nbe1 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_NBE1_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_nbe1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_NBE1_OFFMODEENABLE	OffMode mode override control for pad gpmc_nbe1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_NBE1_INPUTENABLE	Input enable value for pad gpmc_nbe1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_NBE1_PULLTYPESELECT	pullup/down selection for pad gpmc_nbe1 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_NBE1_PULLUDENABLE	pullup/down enable for pad gpmc_nbe1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_NBE1_MUXMODE	Functional multiplexing selection for pad gpmc_nbe1 0x0: Select gpmc_nbe1 0x2: Select c2c_dataout5 0x3: Select gpio_60 0x7: Select safe_mode	RW	0x7

**Table 18-359. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_NBE1\_PAD1\_GPMC\_WAIT0**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-360. CONTROL\_CORE\_PAD0\_GPMC\_WAIT1\_PAD1\_GPMC\_WAIT2**

<b>Address Offset</b>	0x0000 008C		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads gpmc_wait1 and gpmc_wait2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
GPMC_WAIT2_WAKEUPEVENT	GPMC_WAIT2_WAKEUPENABLE	GPMC_WAIT2_OFFMODEPULLTYPESELECT	GPMC_WAIT2_OFFMODEPULLUDENABE	GPMC_WAIT2_OFFMODEOUTVALUE	GPMC_WAIT2_OFFMODEOUTENABLE	GPMC_WAIT2_OFFMODEENABLE	GPMC_WAIT2_INPUTENABLE	RESERVED			GPMC_WAIT2_PULLTYPESELECT	GPMC_WAIT2_PULLUDENABE			GPMC_WAIT2_MUXMODE			GPMC_WAIT1_WAKEUPEVENT	GPMC_WAIT1_WAKEUPENABLE	GPMC_WAIT1_OFFMODEPULLTYPESELECT	GPMC_WAIT1_OFFMODEPULLUDENABE	GPMC_WAIT1_OFFMODEOUTVALUE	GPMC_WAIT1_OFFMODEOUTENABLE	GPMC_WAIT1_OFFMODEENABLE	GPMC_WAIT1_INPUTENABLE	RESERVED			GPMC_WAIT1_PULLTYPESELECT	GPMC_WAIT1_PULLUDENABE			GPMC_WAIT1_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPMC_WAIT2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_WAIT2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_WAIT2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_wait2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_WAIT2_OFFMODEPULLUDENABE	OffMode mode pullup/down enable for pad gpmc_wait2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_WAIT2_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_wait2 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_WAIT2_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_wait2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_WAIT2_OFFMODEENABLE	OffMode mode override control for pad gpmc_wait2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
24	GPMC_WAIT2_INPUTENABLE	Input enable value for pad gpmc_wait2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_WAIT2_PULLTYPESELE CT	pullup/down selection for pad gpmc_wait2 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_WAIT2_PULLUDENABL E	pullup/down enable for pad gpmc_wait2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_WAIT2_MUXMODE	Functional multiplexing selection for pad gpmc_wait2 0x0: Select gpmc_wait2 0x1: Select usbc1_icusb_txen 0x2: Select c2c_dataout3 0x3: Select gpio_100 0x4: Select sys_ndmareq0 0x7: Select safe_mode	RW	0x7
15	GPMC_WAIT1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_WAIT1_WAKEUPENABL E	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_WAIT1_ OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_wait1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_WAIT1_ OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_wait1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_WAIT1_OFFMODEOUTV ALUE	OffMode mode output value for pad gpmc_wait1 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_WAIT1_OFFMODEOUTE NABLE	OffMode mode output enable value for pad gpmc_wait1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_WAIT1_OFFMODEENAB LE	OffMode mode override control for pad gpmc_wait1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_WAIT1_INPUTENABLE	Input enable value for pad gpmc_wait1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	GPMC_WAIT1_PULLTYPESELECTION	pullup/down selection for pad gpmc_wait1 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPMC_WAIT1_PULLUDENABLER	pullup/down enable for pad gpmc_wait1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_WAIT1_MUXMODE	Functional multiplexing selection for pad gpmc_wait1 0x0: Select gpmc_wait1 0x2: Select c2c_dataout2 0x3: Select gpio_62 0x7: Select safe_mode	RW	0x7

**Table 18-361. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_WAIT1\_PAD1\_GPMC\_WAIT2**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-362. CONTROL\_CORE\_PAD0\_GPMC\_NCS4\_PAD1\_GPMC\_NCS5**

<b>Address Offset</b>	0x0000 0090	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ncs4 and gpmc_ncs5 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NCS5_WAKEUPEVENT	GPMC_NCS5_WAKEUPENABLE	GPMC_NCS5_OFFMODEPULLTYPESELECT	GPMC_NCS5_OFFMODEPULLUDENABLER	GPMC_NCS5_OFFMODEOUTVALUE	GPMC_NCS5_OFFMODEOUTENABLE	GPMC_NCS5_OFFMODEENABLE	GPMC_NCS5_INPUTENABLE	RESERVED	GPMC_NCS5_PULLTYPESELECT	GPMC_NCS5_PULLUDENABLER	GPMC_NCS5_MUXMODE	GPMC_NCS4_WAKEUPEVENT	GPMC_NCS4_WAKEUPENABLE	GPMC_NCS4_OFFMODEPULLTYPESELECT	GPMC_NCS4_OFFMODEPULLUDENABLER	GPMC_NCS4_OFFMODEOUTVALUE	GPMC_NCS4_OFFMODEOUTENABLE	GPMC_NCS4_OFFMODEENABLE	GPMC_NCS4_INPUTENABLE	RESERVED	GPMC_NCS4_PULLTYPESELECT	GPMC_NCS4_PULLUDENABLER	GPMC_NCS4_MUXMODE								

Bits	Field Name	Description	Type	Reset
31	GPMC_NCS5_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_NCS5_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
29	GPMC_NCS5_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ncs5 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_NCS5_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ncs5 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_NCS5_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ncs5 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_NCS5_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ncs5 . This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_NCS5_OFFMODEENABLE	OffMode mode override control for pad gpmc_ncs5 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPMC_NCS5_INPUTENABLE	Input enable value for pad gpmc_ncs5 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_NCS5_PULLTYPESELECT	pullup/down selection for pad gpmc_ncs5 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_NCS5_PULLUDENABLE	pullup/down enable for pad gpmc_ncs5 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_NCS5_MUXMODE	Functional multiplexing selection for pad gpmc_ncs5 0x0: Select gpmc_ncs5 0x1: Select dsi1_te1 0x2: Select c2c_clkin1 0x3: Select gpio_102 0x4: Select sys_ndmareq2 0x7: Select safe_mode	RW	0x7
15	GPMC_NCS4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_NCS4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
13	GPMC_NCS4_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ncs4 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_NCS4_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ncs4 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_NCS4_OFFMODEOUTV ALUE	OffMode mode output value for pad gpmc_ncs4 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_NCS4_OFFMODEOUTE NABLE	OffMode mode output enable value for pad gpmc_ncs4. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_NCS4_OFFMODEENAB LE	OffMode mode override control for pad gpmc_ncs4 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_NCS4_INPUTENABLE	Input enable value for pad gpmc_ncs4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPMC_NCS4_PULLTYPESELE CT	pullup/down selection for pad gpmc_ncs4 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_NCS4_PULLUDENABLE	pullup/down enable for pad gpmc_cs4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_NCS4_MUXMODE	Functional multiplexing selection for pad gpmc_ncs4 0x0: Select gpmc_ncs4 0x1: Select dsi1_te0 0x2: Select c2c_clkin0 0x3: Select gpio_101 0x4: Select sys_ndmareq1 0x7: Select safe_mode	RW	0x7

**Table 18-363. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_GPMC\_NCS4\_PAD1\_GPMC\_NCS5**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)



**Table 18-364. CONTROL\_CORE\_PAD0\_GPMC\_NCS6\_PAD1\_GPMC\_NCS7**

<b>Address Offset</b>	0x0000 0094	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpmc_ncs6 and gpmc_ncs7 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
GPMC_NCS7_WAKEUPEVENT	GPMC_NCS7_WAKEUPENABLE	GPMC_NCS7_OFFMODEPULLTYPESELECT	GPMC_NCS7_OFFMODEPULLUDENENABLE	GPMC_NCS7_OFFMODEOUTVALUE	GPMC_NCS7_OFFMODEOUTENABLE	GPMC_NCS7_OFFMODEENABLE	GPMC_NCS7_INPUTENABLE	RESERVED			GPMC_NCS7_PULLTYPESELECT	GPMC_NCS7_PULLUDENENABLE			GPMC_NCS7_MUXMODE			GPMC_NCS6_WAKEUPEVENT	GPMC_NCS6_WAKEUPENABLE	GPMC_NCS6_OFFMODEPULLTYPESELECT	GPMC_NCS6_OFFMODEPULLUDENENABLE	GPMC_NCS6_OFFMODEOUTVALUE	GPMC_NCS6_OFFMODEOUTENABLE	GPMC_NCS6_OFFMODEENABLE	GPMC_NCS6_INPUTENABLE	RESERVED			GPMC_NCS6_PULLTYPESELECT	GPMC_NCS6_PULLUDENENABLE			GPMC_NCS6_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPMC_NCS7_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_NCS7_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPMC_NCS7_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ncs7 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPMC_NCS7_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad gpmc_ncs7 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPMC_NCS7_OFFMODEOUTVALUE	OffMode mode output value for pad gpmc_ncs7 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPMC_NCS7_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpmc_ncs7. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPMC_NCS7_OFFMODEENABLE	OffMode mode override control for pad gpmc_ncs7 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
24	GPMC_NCS7_INPUTENABLE	Input enable value for pad gpmc_ncs7 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPMC_NCS7_PULLTYPESELE CT	pullup/down selection for pad gpmc_ncs7 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPMC_NCS7_PULLUDENABLE	pullup/down enable for pad gpmc_ncs7 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPMC_NCS7_MUXMODE	Functional multiplexing selection for pad gpmc_ncs7 0x0: Select gpmc_ncs7 0x1: Select dsi2_te1 0x2: Select c2c_dataout1 0x3: Select gpio_104 0x7: Select safe_mode	RW	0x7
15	GPMC_NCS6_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_NCS6_WAKEUPENABL E	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPMC_NCS6_ OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpmc_ncs6 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPMC_NCS6_ OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpmc_ncs6 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPMC_NCS6_OFFMODEOUTV ALUE	OffMode mode output value for pad gpmc_ncs6 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPMC_NCS6_OFFMODEOUTE NABLE	OffMode mode output enable value for pad gpmc_ncs6. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPMC_NCS6_OFFMODEENAB LE	OffMode mode override control for pad gpmc_ncs6 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPMC_NCS6_INPUTENABLE	Input enable value for pad gpmc_ncs6 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4	GPMC_NCS6_PULLTYPESELE CT	pullup/down selection for pad gpmc_ncs6  0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPMC_NCS6_PULLUDENABLE	pullup/down enable for pad gpmc_ncs6  0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPMC_NCS6_MUXMODE	Functional multiplexing selection for pad gpmc_ncs6  0x0: Select gpmc_ncs6 0x1: Select dsi2_te0 0x2: Select c2c_dataout0 0x3: Select gpio_103 0x4: Select sys_ndmareq3 0x7: Select safe_mode	RW	0x7

**Table 18-365. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPMC\_NCS6\_PAD1\_GPMC\_NCS7**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-366. CONTROL\_CORE\_PAD0\_GPIO63\_PAD1\_GPIO64**

<b>Address Offset</b>	0x0000 0098	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpio63 and gpio64 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
GPIO64_WAKEUPEVENT	GPIO64_WAKEUPENABLE	GPIO64_OFFMODEPULLTYPESELECT	GPIO64_OFFMODEPULLUDENABLE	GPIO64_OFFMODEOUTVALUE	GPIO64_OFFMODEOUTENABLE	GPIO64_OFFMODEENABLE	GPIO64_INPUTENABLE	RESERVED			GPIO64_PULLTYPESELECT	GPIO64_PULLUDENABLE			GPIO64_MUXMODE			GPIO63_WAKEUPEVENT	GPIO63_WAKEUPENABLE	GPIO63_OFFMODEPULLTYPESELECT	GPIO63_OFFMODEPULLUDENABLE	GPIO63_OFFMODEOUTVALUE	GPIO63_OFFMODEOUTENABLE	GPIO63_OFFMODEENABLE	GPIO63_INPUTENABLE	RESERVED			GPIO63_PULLTYPESELECT	GPIO63_PULLUDENABLE			GPIO63_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPIO64_WAKEUPEVENT	Pad_x wake-up event status latched in the IO  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
30	GPIO64_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPIO64_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpio64 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPIO64_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpio64 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPIO64_OFFMODEOUTVALUE	OffMode mode output value for pad gpio64 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPIO64_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpio64. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPIO64_OFFMODEENABLE	OffMode mode override control for pad gpio64 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPIO64_INPUTENABLE	Input enable value for pad gpio64 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPIO64_PULLTYPESELECT	pullup/down selection for pad gpio64 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPIO64_PULLUDENABLE	pullup/down enable for pad gpio64 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPIO64_MUXMODE	Functional multiplexing selection for pad gpio64 0x0: Reserved 0x3: Select gpio_64 0x7: Select safe_mode	RW	0x7
15	GPIO63_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPIO63_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPIO63_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpio63 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPIO63_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad gpio63 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0

Bits	Field Name	Description	Type	Reset
11	GPIO63_OFFMODEOUTVALUE	OffMode mode output value for pad gpio63 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPIO63_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpio63. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPIO63_OFFMODEENABLE	OffMode mode override control for pad gpio63 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPIO63_INPUTENABLE	Input enable value for pad gpio63 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPIO63_PULLTYPESELECT	pullup/down selection for pad gpio63 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPIO63_PULLUDENABLE	pullup/down enable for pad gpio63 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPIO63_MUXMODE	Functional multiplexing selection for pad gpio63 0x0: Reserved 0x3: Select gpio_63 0x7: Select safe_mode	RW	0x7

**Table 18-367. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPIO63\_PAD1\_GPIO64**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-368. CONTROL\_CORE\_PAD0\_GPIO65\_PAD1\_GPIO66**

<b>Address Offset</b>	0x0000 009C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads gpio65 and gpio66 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
GPIO66_WAKEUPEVENT	GPIO66_WAKEUPENABLE	GPIO66_OFFMODEPULLTYPESELECT	GPIO66_OFFMODEPULLUDENENABLE	GPIO66_OFFMODEOUTVALUE	GPIO66_OFFMODEOUTENABLE	GPIO66_OFFMODEEENABLE	GPIO66_INPUTENABLE	RESERVED			GPIO66_PULLTYPESELECT	GPIO66_PULLUDENENABLE			GPIO66_MUXMODE			GPIO65_WAKEUPEVENT	GPIO65_WAKEUPENABLE	GPIO65_OFFMODEPULLTYPESELECT	GPIO65_OFFMODEPULLUDENENABLE	GPIO65_OFFMODEOUTVALUE	GPIO65_OFFMODEOUTENABLE	GPIO65_OFFMODEEENABLE	GPIO65_INPUTENABLE	RESERVED			GPIO65_PULLTYPESELECT	GPIO65_PULLUDENENABLE			GPIO65_MUXMODE

Bits	Field Name	Description	Type	Reset
31	GPIO66_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPIO66_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	GPIO66_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad gpio66 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	GPIO66_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad gpio66 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	GPIO66_OFFMODEOUTVALUE	OffMode mode output value for pad gpio66 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	GPIO66_OFFMODEOUTENABLE	OffMode mode output enable value for pad gpio66. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	GPIO66_OFFMODEEENABLE	OffMode mode override control for pad gpio66 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	GPIO66_INPUTENABLE	Input enable value for pad gpio66 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPIO66_PULLTYPESELECT	pullup/down selection for pad gpio66 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPIO66_PULLUDENENABLE	pullup/down enable for pad gpio66 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
18:16	GPIO66_MUXMODE	Functional multiplexing selection for pad gpio66 0x0: Reserved 0x3: Select gpio_66 0x7: Select safe_mode	RW	0x7
15	GPIO65_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPIO65_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	GPIO65_OFFMODEPULLTYPES ELECT	OffMode mode pullup/down selection for pad gpio65 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	GPIO65_OFFMODEPULLUDEN ABLE	OffMode mode pullup/down enable for pad gpio65 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	GPIO65_OFFMODEOUTVALUE	OffMode mode output value for pad GPIO65 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	GPIO65_OFFMODEOUTENABL E	OffMode mode output enable value for pad gpio65. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	GPIO65_OFFMODEENABLE	OffMode mode override control for pad gpio65 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	GPIO65_INPUTENABLE	Input enable value for pad gpio65 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPIO65_PULLTYPESELECT	pullup/down selection for pad gpio65 0x0: pulldown selected 0x1: pullup selected	RW	1
3	GPIO65_PULLUDENABLE	pullup/down enable for pad gpio65 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPIO65_MUXMODE	Functional multiplexing selection for pad gpio65 0x0: Reserved 0x3: Select gpio_65 0x7: Select safe_mode	RW	0x7

**Table 18-369. Register Call Summary for Register CONTROL\_CORE\_PAD0\_GPIO65\_PAD1\_GPIO66**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-370. CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0**

<b>Address Offset</b>	0x0000 00A0		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads csi21_dx0 and csi21_dy0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSI21_DY0_WAKEUPEVENT	CSI21_DY0_WAKEUPENABLE	RESERVED					CSI21_DY0_INPUTENABLE	RESERVED	CSI21_DY0_PULLTYPESELECT	CSI21_DY0_PULLUDENABLE	CSI21_DY0_MUXMODE	CSI21_DX0_WAKEUPEVENT	CSI21_DX0_WAKEUPENABLE	RESERVED					CSI21_DX0_INPUTENABLE	RESERVED	CSI21_DX0_PULLTYPESELECT	CSI21_DX0_PULLUDENABLE	CSI21_DX0_MUXMODE								

Bits	Field Name	Description	Type	Reset
31	CSI21_DY0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	CSI21_DY0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	CSI21_DY0_INPUTENABLE	Input enable value for pad csi21_dy0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	CSI21_DY0_PULLTYPESELECT	pullup/down selection for pad csi21_dy0 0x0: pulldown selected 0x1: pullup selected	RW	0
19	CSI21_DY0_PULLUDENABLE	pullup/down enable for pad csi21_dy0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	CSI21_DY0_MUXMODE	Functional multiplexing selection for pad csi21_dy0 0x0: Select csi21_dy0 0x3: Select gpi_68 0x7: Select safe_mode	RW	0x7
15	CSI21_DX0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	CSI21_DX0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00



Bits	Field Name	Description	Type	Reset
8	CSI21_DX0_INPUTENABLE	Input enable value for pad csi21_dx0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	CSI21_DX0_PULLTYPESELECT	pullup/down selection for pad csi21_dx0 0x0: pulldown selected 0x1: pullup selected	RW	0
3	CSI21_DX0_PULLUDENABLE	pullup/down enable for pad csi21_dx0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	CSI21_DX0_MUXMODE	Functional multiplexing selection for pad csi21_dx0 0x0: Select csi21_dx0 0x3: Select gpi_67 0x7: Select safe_mode	RW	0x7

**Table 18-371. Register Call Summary for Register CONTROL\_CORE\_PAD0\_CSI21\_DX0\_PAD1\_CSI21\_DY0**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-372. CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1**

<b>Address Offset</b>	0x0000 00A4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads csi21_dx1 and csi21_dy1 Access conditions: Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
CSI21_DY1_WAKEUPEVENT	CSI21_DY1_WAKEUPENABLE	RESERVED						CSI21_DY1_INPUTENABLE	RESERVED						CSI21_DY1_PULLTYPESELECT	CSI21_DY1_PULLUDENABLE	CSI21_DY1_MUXMODE	CSI21_DX1_WAKEUPEVENT	CSI21_DX1_WAKEUPENABLE	RESERVED						CSI21_DX1_INPUTENABLE	RESERVED						CSI21_DX1_PULLTYPESELECT	CSI21_DX1_PULLUDENABLE	CSI21_DX1_MUXMODE

Bits	Field Name	Description	Type	Reset
31	CSI21_DY1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	CSI21_DY1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
24	CSI21_DY1_INPUTENABLE	Input enable value for pad csi21_dy1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	CSI21_DY1_PULLTYPESELECT	pullup/down selection for pad csi21_dy1 0x0: pulldown selected 0x1: pullup selected	RW	0
19	CSI21_DY1_PULLUDENABLE	pullup/down enable for pad csi21_dy1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	CSI21_DY1_MUXMODE	Functional multiplexing selection for pad csi21_dy1 0x0: Select csi21_dy1 0x3: Select gpi_70 0x7: Select safe_mode	RW	0x7
15	CSI21_DX1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	CSI21_DX1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	CSI21_DX1_INPUTENABLE	Input enable value for pad csi21_dx1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	CSI21_DX1_PULLTYPESELECT	pullup/down selection for pad csi21_dx1 0x0: pulldown selected 0x1: pullup selected	RW	0
3	CSI21_DX1_PULLUDENABLE	pullup/down enable for pad csi21_dx1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	CSI21_DX1_MUXMODE	Functional multiplexing selection for pad csi21_dx1 0x0: Select csi21_dx1 0x3: Select gpi_69 0x7: Select safe_mode	RW	0x7

**Table 18-373. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_CSI21\_DX1\_PAD1\_CSI21\_DY1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-374. CONTROL\_CORE\_PAD0\_CSI21\_DX2\_PAD1\_CSI21\_DY2**

<b>Address Offset</b>	0x0000 00A8		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads csi21_dx2 and csi21_dy2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSI21_DY2_WAKEUPEVENT	CSI21_DY2_WAKEUPENABLE	RESERVED					CSI21_DY2_INPUTENABLE	RESERVED	CSI21_DY2_PULLTYPESELECT	CSI21_DY2_PULLUDENABLE	CSI21_DY2_MUXMODE	CSI21_DX2_WAKEUPEVENT	CSI21_DX2_WAKEUPENABLE	RESERVED					CSI21_DX2_INPUTENABLE	RESERVED	CSI21_DX2_PULLTYPESELECT	CSI21_DX2_PULLUDENABLE	CSI21_DX2_MUXMODE								

Bits	Field Name	Description	Type	Reset
31	CSI21_DY2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	CSI21_DY2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	CSI21_DY2_INPUTENABLE	Input enable value for pad csi21_dy2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	CSI21_DY2_PULLTYPESELECT	pullup/down selection for pad csi21_dy2 0x0: pulldown selected 0x1: pullup selected	RW	0
19	CSI21_DY2_PULLUDENABLE	pullup/down enable for pad csi21_dy2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	CSI21_DY2_MUXMODE	Functional multiplexing selection for pad csi21_dy2 0x0: Select csi21_dy2 0x3: Select gpi_72 0x7: Select safe_mode	RW	0x7
15	CSI21_DX2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	CSI21_DX2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
8	CSI21_DX2_INPUTENABLE	Input enable value for pad csi21_dx2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	CSI21_DX2_PULLTYPESELECT	pullup/down selection for pad csi21_dx2 0x0: pulldown selected 0x1: pullup selected	RW	0
3	CSI21_DX2_PULLUDENABLE	pullup/down enable for pad csi21_dx2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	CSI21_DX2_MUXMODE	Functional multiplexing selection for pad csi21_dx2 0x0: Select csi21_dx2 0x3: Select gpi_71 0x7: Select safe_mode	RW	0x7

**Table 18-375. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_CSI21\_DX2\_PAD1\_CSI21\_DY2**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-376. CONTROL\_CORE\_PAD0\_CSI21\_DX3\_PAD1\_CSI21\_DY3**

<b>Address Offset</b>	0x0000 00AC	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads csi21_dx3 and csi21_dy3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSI21_DY3_WAKEUPEVENT	CSI21_DY3_WAKEUPENABLE	RESERVED						CSI21_DY3_INPUTENABLE	RESERVED		CSI21_DY3_PULLTYPESELECT	CSI21_DY3_PULLUDENABLE	CSI21_DY3_MUXMODE			CSI21_DX3_WAKEUPEVENT	CSI21_DX3_WAKEUPENABLE	RESERVED						CSI21_DX3_INPUTENABLE	RESERVED		CSI21_DX3_PULLTYPESELECT	CSI21_DX3_PULLUDENABLE	CSI21_DX3_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	CSI21_DY3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	CSI21_DY3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
24	CSI21_DY3_INPUTENABLE	Input enable value for pad csi21_dy3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	CSI21_DY3_PULLTYPESELECT	pullup/down selection for pad csi21_dy3 0x0: pulldown selected 0x1: pullup selected	RW	0
19	CSI21_DY3_PULLUDENABLE	pullup/down enable for pad csi21_dy3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	CSI21_DY3_MUXMODE	Functional multiplexing selection for pad csi21_dy3 0x0: Select csi21_dy3 0x3: Select gpi_74 0x7: Select safe_mode	RW	0x7
15	CSI21_DX3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	CSI21_DX3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	CSI21_DX3_INPUTENABLE	Input enable value for pad csi21_dx3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	CSI21_DX3_PULLTYPESELECT	pullup/down selection for pad csi21_dx3 0x0: pulldown selected 0x1: pullup selected	RW	0
3	CSI21_DX3_PULLUDENABLE	pullup/down enable for pad csi21_dx3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	CSI21_DX3_MUXMODE	Functional multiplexing selection for pad csi21_dx3 0x0: Select csi21_dx3 0x3: Select gpi_73 0x7: Select safe_mode	RW	0x7

**Table 18-377. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_CSI21\_DX3\_PAD1\_CSI21\_DY3**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-378. CONTROL\_CORE\_PAD0\_CSI21\_DX4\_PAD1\_CSI21\_DY4**

<b>Address Offset</b>	0x0000 00B0		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads csi21_dx4 and csi21_dy4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSI21_DY4_WAKEUPEVENT	CSI21_DY4_WAKEUPENABLE	RESERVED					CSI21_DY4_INPUTENABLE	RESERVED	CSI21_DY4_PULLTYPESELECT	CSI21_DY4_PULLUDENABLE	CSI21_DY4_MUXMODE	CSI21_DX4_WAKEUPEVENT	CSI21_DX4_WAKEUPENABLE	RESERVED					CSI21_DX4_INPUTENABLE	RESERVED	CSI21_DX4_PULLTYPESELECT	CSI21_DX4_PULLUDENABLE	CSI21_DX4_MUXMODE								

Bits	Field Name	Description	Type	Reset
31	CSI21_DY4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	CSI21_DY4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	CSI21_DY4_INPUTENABLE	Input enable value for pad csi21_dy4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	CSI21_DY4_PULLTYPESELECT	pullup/down selection for pad csi21_dy4 0x0: pulldown selected 0x1: pullup selected	RW	0
19	CSI21_DY4_PULLUDENABLE	pullup/down enable for pad csi21_dy4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	CSI21_DY4_MUXMODE	Functional multiplexing selection for pad csi21_dy4 0x0: Select csi21_dy4 0x3: Select gpi_76 0x7: Select safe_mode	RW	0x7
15	CSI21_DX4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	CSI21_DX4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
8	CSI21_DX4_INPUTENABLE	Input enable value for pad csi21_dx4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	CSI21_DX4_PULLTYPESELECT	pullup/down selection for pad csi21_dx4 0x0: pulldown selected 0x1: pullup selected	RW	0
3	CSI21_DX4_PULLUDENABLE	pullup/down enable for pad csi21_dx4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	CSI21_DX4_MUXMODE	Functional multiplexing selection for pad csi21_dx4 0x0: Select csi21_dx4 0x3: Select gpi_75 0x7: Select safe_mode	RW	0x7

**Table 18-379. Register Call Summary for Register CONTROL\_CORE\_PAD0\_CSI21\_DX4\_PAD1\_CSI21\_DY4**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-380. CONTROL\_CORE\_PAD0\_CSI22\_DX0\_PAD1\_CSI22\_DY0**

<b>Address Offset</b>	0x0000 00B4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads csi22_dx0 and csi22_dy0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSI22_DY0_WAKEUPEVENT	CSI22_DY0_WAKEUPENABLE	RESERVED					CSI22_DY0_INPUTENABLE	RESERVED			CSI22_DY0_PULLTYPESELECT	CSI22_DY0_PULLUDENABLE	CSI22_DY0_MUXMODE			CSI22_DX0_WAKEUPEVENT	CSI22_DX0_WAKEUPENABLE	RESERVED					CSI22_DX0_INPUTENABLE	RESERVED			CSI22_DX0_PULLTYPESELECT	CSI22_DX0_PULLUDENABLE	CSI22_DX0_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	CSI22_DY0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	CSI22_DY0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
24	CSI22_DY0_INPUTENABLE	Input enable value for pad csi22_dy0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	CSI22_DY0_PULLTYPESELECT	pullup/down selection for pad csi22_dy0 0x0: pulldown selected 0x1: pullup selected	RW	0
19	CSI22_DY0_PULLUDENABLE	pullup/down enable for pad csi22_dy0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	CSI22_DY0_MUXMODE	Functional multiplexing selection for pad csi22_dy0 0x0: Select csi22_dy0 0x3: Select gpi_78 0x7: Select safe_mode	RW	0x7
15	CSI22_DX0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	CSI22_DX0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	CSI22_DX0_INPUTENABLE	Input enable value for pad csi22_dx0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	CSI22_DX0_PULLTYPESELECT	pullup/down selection for pad csi22_dx0 0x0: pulldown selected 0x1: pullup selected	RW	0
3	CSI22_DX0_PULLUDENABLE	pullup/down enable for pad csi22_dx0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	CSI22_DX0_MUXMODE	Functional multiplexing selection for pad csi22_dx0 0x0: Select csi22_dx0 0x3: Select gpi_77 0x7: Select safe_mode	RW	0x7

**Table 18-381. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_CSI22\_DX0\_PAD1\_CSI22\_DY0**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)



**Table 18-382. CONTROL\_CORE\_PAD0\_CSI22\_DX1\_PAD1\_CSI22\_DY1**

<b>Address Offset</b>	0x0000 00B8		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads csi22_dx1 and csi22_dy1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSI22_DY1_WAKEUPEVENT	CSI22_DY1_WAKEUPENABLE	RESERVED					CSI22_DY1_INPUTENABLE	RESERVED	CSI22_DY1_PULLTYPESELECT	CSI22_DY1_PULLUDENABLE	CSI22_DY1_MUXMODE	CSI22_DX1_WAKEUPEVENT	CSI22_DX1_WAKEUPENABLE	RESERVED					CSI22_DX1_INPUTENABLE	RESERVED	CSI22_DX1_PULLTYPESELECT	CSI22_DX1_PULLUDENABLE	CSI22_DX1_MUXMODE								

Bits	Field Name	Description	Type	Reset
31	CSI22_DY1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	CSI22_DY1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	CSI22_DY1_INPUTENABLE	Input enable value for pad csi22_dy1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	CSI22_DY1_PULLTYPESELECT	pullup/down selection for pad csi22_dy1 0x0: pulldown selected 0x1: pullup selected	RW	0
19	CSI22_DY1_PULLUDENABLE	pullup/down enable for pad csi22_dy1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	CSI22_DY1_MUXMODE	Functional multiplexing selection for pad csi22_dy1 0x0: Select csi22_dy1 0x3: Select gpi_80 0x7: Select safe_mode	RW	0x7
15	CSI22_DX1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	CSI22_DX1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
8	CSI22_DX1_INPUTENABLE	Input enable value for pad csi22_dx1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	CSI22_DX1_PULLTYPESELECT	pullup/down selection for pad csi22_dx1 0x0: pulldown selected 0x1: pullup selected	RW	0
3	CSI22_DX1_PULLUDENABLE	pullup/down enable for pad csi22_dx1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	CSI22_DX1_MUXMODE	Functional multiplexing selection for pad csi22_dx1 0x0: Select csi22_dx1 0x3: Select gpi_79 0x7: Select safe_mode	RW	0x7

**Table 18-383. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_CSI22\_DX1\_PAD1\_CSI22\_DY1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-384. CONTROL\_CORE\_PAD0\_CAM\_SHUTTER\_PAD1\_CAM\_STROBE**

<b>Address Offset</b>	0x0000 00BC	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads cam_shutter and cam_strobe Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
CAM_STROBE_WAKEUPEVENT	CAM_STROBE_WAKEUPENABLE	CAM_STROBE_OFFMODEPULLTYPESELECT	CAM_STROBE_OFFMODEPULLUDENABLE	CAM_STROBE_OFFMODEOUTVALUE	CAM_STROBE_OFFMODEOUTENABLE	CAM_STROBE_OFFMODEENABLE	CAM_STROBE_INPUTENABLE	RESERVED	CAM_STROBE_PULLTYPESELECT	CAM_STROBE_PULLUDENABLE	CAM_STROBE_MUXMODE	CAM_SHUTTER_WAKEUPEVENT	CAM_SHUTTER_WAKEUPENABLE	CAM_SHUTTER_OFFMODEPULLTYPESELECT	CAM_SHUTTER_OFFMODEPULLUDENABLE	CAM_SHUTTER_OFFMODEOUTVALUE	CAM_SHUTTER_OFFMODEOUTENABLE	CAM_SHUTTER_OFFMODEENABLE	CAM_SHUTTER_INPUTENABLE	RESERVED	CAM_SHUTTER_PULLTYPESELECT	CAM_SHUTTER_PULLUDENABLE	CAM_SHUTTER_MUXMODE										

Bits	Field Name	Description	Type	Reset
31	CAM_STROBE_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
30	CAM_STROBE_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	CAM_STROBE_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad cam_strobe 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	CAM_STROBE_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad cam_strobe 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	CAM_STROBE_OFFMODEOUTVALUE	OffMode mode output value for pad cam_strobe 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	CAM_STROBE_OFFMODEOUTENABLE	OffMode mode output enable value for pad cam_strobe. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	CAM_STROBE_OFFMODEENABLE	OffMode mode override control for pad cam_strobe 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	CAM_STROBE_INPUTENABLE	Input enable value for pad cam_strobe 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	CAM_STROBE_PULLTYPESELECT	pullup/down selection for pad cam_strobe 0x0: pulldown selected 0x1: pullup selected	RW	0
19	CAM_STROBE_PULLUDENABLE	pullup/down enable for pad cam_strobe 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	CAM_STROBE_MUXMODE	Functional multiplexing selection for pad cam_strobe 0x0: Select cam_strobe 0x3: Select gpio_82 0x7: Select safe_mode	RW	0x7
15	CAM_SHUTTER_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	CAM_SHUTTER_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	CAM_SHUTTER_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad cam_shutter 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	CAM_SHUTTER_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad cam_shutter 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0

Bits	Field Name	Description	Type	Reset
11	CAM_SHUTTER_OFFMODEOUTVALUE	OffMode mode output value for pad cam_shutter 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	CAM_SHUTTER_OFFMODEOUTENABLE	OffMode mode output enable value for pad cam_shutter. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	CAM_SHUTTER_OFFMODEENABLE	OffMode mode override control for pad cam_shutter 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	CAM_SHUTTER_INPUTENABLE	Input enable value for pad cam_shutter 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	CAM_SHUTTER_PULLTYPESELECT	pullup/down selection for pad cam_shutter 0x0: pulldown selected 0x1: pullup selected	RW	0
3	CAM_SHUTTER_PULLUDENABLE	pullup/down enable for pad cam_shutter 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	CAM_SHUTTER_MUXMODE	Functional multiplexing selection for pad cam_shutter 0x0: Select cam_shutter 0x3: Select gpio_81 0x7: Select safe_mode	RW	0x7

**Table 18-385. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_CAM\_SHUTTER\_PAD1\_CAM\_STROBE**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-386. CONTROL\_CORE\_PAD0\_CAM\_GLOBALRESET\_PAD1\_USBB1\_ULPITLL\_CLK**

<b>Address Offset</b>	0x0000 00C0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads cam_globalreset and usbb1_ulpitll_clk Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB1_ULPITLL_CLK_WAKEUPEVENT	USBB1_ULPITLL_CLK_WAKEUPENABLE	USBB1_ULPITLL_CLK_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_CLK_OFFMODEPULLUDENENABLE	USBB1_ULPITLL_CLK_OFFMODEOUTVALUE	USBB1_ULPITLL_CLK_OFFMODEOUTENABLE	USBB1_ULPITLL_CLK_OFFMODEENABLE	USBB1_ULPITLL_CLK_INPUTENABLE	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CAM_GLOBALRESET_WAKEUPEVENT	CAM_GLOBALRESET_WAKEUPENABLE	CAM_GLOBALRESET_OFFMODEPULLTYPESELECT	CAM_GLOBALRESET_OFFMODEPULLUDENENABLE	CAM_GLOBALRESET_OFFMODEOUTVALUE	CAM_GLOBALRESET_OFFMODEOUTENABLE	CAM_GLOBALRESET_OFFMODEENABLE	CAM_GLOBALRESET_INPUTENABLE	RESERVED	RESERVED	RESERVED	CAM_GLOBALRESET_PULLTYPESELECT	CAM_GLOBALRESET_PULLUDENENABLE	CAM_GLOBALRESET_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	USBB1_ULPITLL_CLK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB1_ULPITLL_CLK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB1_ULPITLL_CLK_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_clk 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB1_ULPITLL_CLK_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_clk 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB1_ULPITLL_CLK_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_clk 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB1_ULPITLL_CLK_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_clk. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB1_ULPITLL_CLK_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_clk 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB1_ULPITLL_CLK_INPUTENABLE	Input enable value for pad usbb1_ulpitll_clk 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20	USBB1_ULPITLL_CLK_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_clk 0x0: pulldown selected 0x1: pullup selected	RW	0
19	USBB1_ULPITLL_CLK_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_clk 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB1_ULPITLL_CLK_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_clk 0x0: Select usbb1_ulpitll_clk 0x1: Select hsi1_cawake 0x3: Select gpio_84 0x4: Select usbb1_ulpiphyc_clk 0x6: Select hw_dbg20 0x7: Select safe_mode	RW	0x7
15	CAM_GLOBALRESET_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	CAM_GLOBALRESET_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	CAM_GLOBALRESET_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad cam_globalreset 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	CAM_GLOBALRESET_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad cam_globalreset 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	CAM_GLOBALRESET_OFFMODEOUTVALUE	OffMode mode output value for pad cam_globalreset 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	CAM_GLOBALRESET_OFFMODEOUTENABLE	OffMode mode output enable value for pad cam_globalreset. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	CAM_GLOBALRESET_OFFMODEENABLE	OffMode mode override control for pad cam_globalreset 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	CAM_GLOBALRESET_INPUTENABLE	Input enable value for pad cam_globalreset 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	CAM_GLOBALRESET_PULLTYPESELECT	pullup/down selection for pad cam_globalreset 0x0: pulldown selected 0x1: pullup selected	RW	0
3	CAM_GLOBALRESET_PULLUDENABLE	pullup/down enable for pad cam_globalreset 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
2:0	CAM_GLOBALRESET_MUXMODE	Functional multiplexing selection for pad cam_globalreset 0x0: Select cam_globalreset 0x3: Select gpio_83 0x7: Select safe_mode	RW	0x7

**Table 18-387. Register Call Summary for Register CONTROL\_CORE\_PAD0\_CAM\_GLOBALRESET\_PAD1\_USBB1\_ULPITLL\_CLK**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[3\]](#)

**Table 18-388. CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_STP\_PAD1\_USBB1\_ULPITLL\_DIR**

<b>Address Offset</b>	0x0000 00C4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb1_ulpitll_stp and usbb1_ulpitll_dir Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB1_ULPITLL_DIR_WAKEUPEVENT	USBB1_ULPITLL_DIR_WAKEUPENABLE	USBB1_ULPITLL_DIR_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_DIR_OFFMODEPULLUDENABLE	USBB1_ULPITLL_DIR_OFFMODEOUTVALUE	USBB1_ULPITLL_DIR_OFFMODEOUTENABLE	USBB1_ULPITLL_DIR_OFFMODEENABLE	USBB1_ULPITLL_DIR_INPUTENABLE	RESERVED	RESERVED	RESERVED	USBB1_ULPITLL_DIR_PULLTYPESELECT	USBB1_ULPITLL_DIR_PULLUDENABLE	USBB1_ULPITLL_DIR_MUXMODE	USBB1_ULPITLL_STP_WAKEUPEVENT	USBB1_ULPITLL_STP_WAKEUPENABLE	USBB1_ULPITLL_STP_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_STP_OFFMODEPULLUDENABLE	USBB1_ULPITLL_STP_OFFMODEOUTVALUE	USBB1_ULPITLL_STP_OFFMODEOUTENABLE	USBB1_ULPITLL_STP_OFFMODEENABLE	USBB1_ULPITLL_STP_INPUTENABLE	RESERVED	RESERVED	RESERVED	USBB1_ULPITLL_STP_PULLTYPESELECT	USBB1_ULPITLL_STP_PULLUDENABLE	USBB1_ULPITLL_STP_MUXMODE				

Bits	Field Name	Description	Type	Reset
31	USBB1_ULPITLL_DIR_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB1_ULPITLL_DIR_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB1_ULPITLL_DIR_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_dir 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
28	USBB1_ULPITLL_DIR_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_dir 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB1_ULPITLL_DIR_OFFMODEEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_dir 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB1_ULPITLL_DIR_OFFMODEEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_dir. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB1_ULPITLL_DIR_OFFMODEEENABLE	OffMode mode override control for pad usbb1_ulpitll_dir 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB1_ULPITLL_DIR_INPUTENABLE	Input enable value for pad usbb1_ulpitll_dir 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	USBB1_ULPITLL_DIR_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_dir 0x0: pulldown selected 0x1: pullup selected	RW	0
19	USBB1_ULPITLL_DIR_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_dir 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB1_ULPITLL_DIR_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_dir 0x0: Select usbb1_ulpitll_dir 0x1: Select hsi1_caflag 0x2: Select mcbasp4_fsr 0x3: Select gpio_86 0x4: Select usbb1_ulpiphy_dir 0x6: Select hw_dbg22 0x7: Select safe_mode	RW	0x7
15	USBB1_ULPITLL_STP_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB1_ULPITLL_STP_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB1_ULPITLL_STP_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_stp 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB1_ULPITLL_STP_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_stp 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0



Bits	Field Name	Description	Type	Reset
11	USBB1_ULPITLL_STP_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_stp 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB1_ULPITLL_STP_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_stp. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB1_ULPITLL_STP_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_stp 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	USBB1_ULPITLL_STP_INPUTENABLE	Input enable value for pad usbb1_ulpitll_stp 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB1_ULPITLL_STP_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_stp 0x0: pulldown selected 0x1: pullup selected	RW	1
3	USBB1_ULPITLL_STP_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_stp 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB1_ULPITLL_STP_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_stp 0x0: Select usbb1_ulpitll_stp 0x1: Select hsi1_cadata 0x2: Select mcbasp4_clkr 0x3: Select gpio_85 0x4: Select usbb1_ulpiphy_stp 0x5: Select usbb1_mm_rxdp 0x6: Select hw_dbg21 0x7: Select safe_mode	RW	0x7

**Table 18-389. Register Call Summary for Register CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_STP\_PAD1\_USBB1\_ULPITLL\_DIR**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-390. CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_NXT\_PAD1\_USBB1\_ULPITLL\_DAT0**

Address Offset	0x0000 00C8	Instance	SYSCTRL_PADCONF_CORE
Physical Address	See <a href="#">Table 18-315</a>		
Description	Register control for Pads usbb1_ulpitll_nxt and usbb1_ulpitll_dat0 Access conditions. Read: unrestricted, Write: unrestricted		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB1_ULPITLL_DAT0_WAKEUPEVENT	USBB1_ULPITLL_DAT0_WAKEUPENABLE	USBB1_ULPITLL_DAT0_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_DAT0_OFFMODEPULLUDENABLE	USBB1_ULPITLL_DAT0_OFFMODEOUTVALUE	USBB1_ULPITLL_DAT0_OFFMODEOUTENABLE	USBB1_ULPITLL_DAT0_OFFMODEENABLE	USBB1_ULPITLL_DAT0_INPUTENABLE	RESERVED			USBB1_ULPITLL_DAT0_PULLTYPESELECT	USBB1_ULPITLL_DAT0_PULLUDENABLE		USBB1_ULPITLL_DAT0_MUXMODE		USBB1_ULPITLL_NXT_WAKEUPEVENT	USBB1_ULPITLL_NXT_WAKEUPENABLE	USBB1_ULPITLL_NXT_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_NXT_OFFMODEPULLUDENABLE	USBB1_ULPITLL_NXT_OFFMODEOUTVALUE	USBB1_ULPITLL_NXT_OFFMODEOUTENABLE	USBB1_ULPITLL_NXT_OFFMODEENABLE	USBB1_ULPITLL_NXT_INPUTENABLE	RESERVED			USBB1_ULPITLL_NXT_PULLTYPESELECT	USBB1_ULPITLL_NXT_PULLUDENABLE		USBB1_ULPITLL_NXT_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	USBB1_ULPITLL_DAT0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB1_ULPITLL_DAT0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB1_ULPITLL_DAT0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_dat0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB1_ULPITLL_DAT0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_dat0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB1_ULPITLL_DAT0_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_dat0 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB1_ULPITLL_DAT0_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_dat0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB1_ULPITLL_DAT0_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_dat0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB1_ULPITLL_DAT0_INPUTENABLE	Input enable value for pad usbb1_ulpitll_dat0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20	USBB1_ULPITLL_DAT0_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_dat0 0x0: pulldown selected 0x1: pullup selected	RW	0
19	USBB1_ULPITLL_DAT0_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_dat0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB1_ULPITLL_DAT0_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_dat0 0x0: Select usbb1_ulpitll_dat0 0x1: Select hsi1_acwake 0x2: Select mcbsp4_clkx 0x3: Select gpio_88 0x4: Select usbb1_ulpiphy_dat0 0x5: Select usbb1_mm_txen 0x6: Select hw_dbg24 0x7: Select safe_mode	RW	0x7
15	USBB1_ULPITLL_NXT_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB1_ULPITLL_NXT_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB1_ULPITLL_NXT_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_nxt 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB1_ULPITLL_NXT_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_nxt 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB1_ULPITLL_NXT_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_nxt 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB1_ULPITLL_NXT_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_nxt. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB1_ULPITLL_NXT_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_nxt 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	USBB1_ULPITLL_NXT_INPUTENABLE	Input enable value for pad usbb1_ulpitll_nxt 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB1_ULPITLL_NXT_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_nxt 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	USBB1_ULPITLL_NXT_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_nxt 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB1_ULPITLL_NXT_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_nxt 0x0: Select usbb1_ulpitll_nxt 0x1: Select hsi1_acready 0x2: Select mcbsp4_fsx 0x3: Select gpio_87 0x4: Select usbb1_ulpiphy_nxt 0x5: Select usbb1_mm_rxdm 0x6: Select hw_dbg23 0x7: Select safe_mode	RW	0x7

**Table 18-391. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_NXT\_PAD1\_USBB1\_ULPITLL\_DAT0**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-392. CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_DAT1\_PAD1\_USBB1\_ULPITLL\_DAT2**

<b>Address Offset</b>	0x0000 00CC	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb1_ulpitll_dat1 and usbb1_ulpitll_dat2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB1_ULPITLL_DAT2_WAKEUPEVENT	USBB1_ULPITLL_DAT2_WAKEUPENABLE	USBB1_ULPITLL_DAT2_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_DAT2_OFFMODEPULLUDENABLE	USBB1_ULPITLL_DAT2_OFFMODEOUTVALUE	USBB1_ULPITLL_DAT2_OFFMODEOUTENABLE	USBB1_ULPITLL_DAT2_OFFMODEEENABLE	USBB1_ULPITLL_DAT2_INPUTENABLE	RESERVED			USBB1_ULPITLL_DAT2_PULLTYPESELECT	USBB1_ULPITLL_DAT2_PULLUDENABLE	USBB1_ULPITLL_DAT2_MUXMODE			USBB1_ULPITLL_DAT1_WAKEUPEVENT	USBB1_ULPITLL_DAT1_WAKEUPENABLE	USBB1_ULPITLL_DAT1_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_DAT1_OFFMODEPULLUDENABLE	USBB1_ULPITLL_DAT1_OFFMODEOUTVALUE	USBB1_ULPITLL_DAT1_OFFMODEOUTENABLE	USBB1_ULPITLL_DAT1_OFFMODEEENABLE	USBB1_ULPITLL_DAT1_INPUTENABLE	RESERVED			USBB1_ULPITLL_DAT1_PULLTYPESELECT	USBB1_ULPITLL_DAT1_PULLUDENABLE	USBB1_ULPITLL_DAT1_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	USBB1_ULPITLL_DAT2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
30	USBB1_ULPITLL_DAT2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB1_ULPITLL_DAT2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_dat2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB1_ULPITLL_DAT2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_dat2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB1_ULPITLL_DAT2_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_dat2 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB1_ULPITLL_DAT2_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_dat2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB1_ULPITLL_DAT2_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_dat2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB1_ULPITLL_DAT2_INPUTENABLE	Input enable value for pad usbb1_ulpitll_dat2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	USBB1_ULPITLL_DAT2_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_dat2 0x0: pulldown selected 0x1: pullup selected	RW	0
19	USBB1_ULPITLL_DAT2_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_dat2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB1_ULPITLL_DAT2_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_dat2 0x0: Select usbb1_ulpitll_dat2 0x1: Select hsi1_aclflag 0x2: Select mcbsp4_dr 0x3: Select gpio_90 0x4: Select usbb1_ulpiphy_dat2 0x5: Select usbb1_mm_txse0 0x6: Select hw_dbg26 0x7: Select safe_mode	RW	0x7
15	USBB1_ULPITLL_DAT1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB1_ULPITLL_DAT1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
13	USBB1_ULPITLL_DAT1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_dat1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB1_ULPITLL_DAT1_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_dat1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB1_ULPITLL_DAT1_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_dat1 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB1_ULPITLL_DAT1_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_dat1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB1_ULPITLL_DAT1_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_dat1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	USBB1_ULPITLL_DAT1_INPUTENABLE	Input enable value for pad usbb1_ulpitll_dat1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB1_ULPITLL_DAT1_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_dat1 0x0: pulldown selected 0x1: pullup selected	RW	0
3	USBB1_ULPITLL_DAT1_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_dat1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB1_ULPITLL_DAT1_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_dat1 0x0: Select usbb1_ulpitll_dat1 0x1: Select hsi1_acdata 0x2: Select mcbasp4_dx 0x3: Select gpio_89 0x4: Select usbb1_ulpiphy_dat1 0x5: Select usbb1_mm_txdat 0x6: Select hw_dbg25 0x7: Select safe_mode	RW	0x7

**Table 18-393. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_DAT1\_PAD1\_USBB1\_ULPITLL\_DAT2**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-394. CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_DAT3\_PAD1\_USBB1\_ULPITLL\_DAT4**

<b>Address Offset</b>	0x0000 00D0		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads usbb1_ulpitll_dat3 and usbb1_ulpitll_dat4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB1_ULPITLL_DAT4_WAKEUPEVENT	USBB1_ULPITLL_DAT4_WAKEUPENABLE	USBB1_ULPITLL_DAT4_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_DAT4_OFFMODEPULLUDENABE	USBB1_ULPITLL_DAT4_OFFMODEOUTVALUE	USBB1_ULPITLL_DAT4_OFFMODEOUTENABLE	USBB1_ULPITLL_DAT4_OFFMODEENABLE	USBB1_ULPITLL_DAT4_INPUTENABLE	RESERVED			USBB1_ULPITLL_DAT4_PULLTYPESELECT	USBB1_ULPITLL_DAT4_PULLUDENABE		USBB1_ULPITLL_DAT4_MUXMODE		USBB1_ULPITLL_DAT3_WAKEUPEVENT	USBB1_ULPITLL_DAT3_WAKEUPENABLE	USBB1_ULPITLL_DAT3_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_DAT3_OFFMODEPULLUDENABE	USBB1_ULPITLL_DAT3_OFFMODEOUTVALUE	USBB1_ULPITLL_DAT3_OFFMODEOUTENABLE	USBB1_ULPITLL_DAT3_OFFMODEENABLE	USBB1_ULPITLL_DAT3_INPUTENABLE	RESERVED			USBB1_ULPITLL_DAT3_PULLTYPESELECT	USBB1_ULPITLL_DAT3_PULLUDENABE		USBB1_ULPITLL_DAT3_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	USBB1_ULPITLL_DAT4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB1_ULPITLL_DAT4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB1_ULPITLL_DAT4_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_dat4 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB1_ULPITLL_DAT4_OFFMODEPULLUDENABE	OffMode mode pullup/down enable for pad usbb1_ulpitll_dat4 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB1_ULPITLL_DAT4_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_dat4 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB1_ULPITLL_DAT4_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_dat4. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
25	USBB1_ULPITLL_DAT4_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_dat4 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB1_ULPITLL_DAT4_INPUTENABLE	Input enable value for pad usbb1_ulpitll_dat4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	USBB1_ULPITLL_DAT4_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_dat4 0x0: pulldown selected 0x1: pullup selected	RW	0
19	USBB1_ULPITLL_DAT4_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_dat4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB1_ULPITLL_DAT4_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_dat4 0x0: Select usbb1_ulpitll_dat4 0x1: Select dmtimer8_pwm_evt 0x2: Select abe_mcbasp3_dr 0x3: Select gpio_92 0x4: Select usbb1_ulpipiphy_dat4 0x6: Select hw_dbg28 0x7: Select safe_mode	RW	0x7
15	USBB1_ULPITLL_DAT3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB1_ULPITLL_DAT3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB1_ULPITLL_DAT3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_dat3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB1_ULPITLL_DAT3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_dat3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB1_ULPITLL_DAT3_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_dat3 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB1_ULPITLL_DAT3_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_dat3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB1_ULPITLL_DAT3_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_dat3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0



Bits	Field Name	Description	Type	Reset
8	USBB1_ULPITLL_DAT3_INPUTENABLE	Input enable value for pad usbb1_ulpitll_dat3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB1_ULPITLL_DAT3_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_dat3 0x0: pulldown selected 0x1: pullup selected	RW	0
3	USBB1_ULPITLL_DAT3_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_dat3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB1_ULPITLL_DAT3_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_dat3 0x0: Select usbb1_ulpitll_dat3 0x1: Select hsi1_caready 0x3: Select gpio_91 0x4: Select usbb1_ulpiphy_dat3 0x5: Select usbb1_mm_rxcv 0x6: Select hw_dbg27 0x7: Select safe_mode	RW	0x7

**Table 18-395. Register Call Summary for Register CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_DAT3\_PAD1\_USBB1\_ULPITLL\_DAT4**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-396. CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_DAT5\_PAD1\_USBB1\_ULPITLL\_DAT6**

<b>Address Offset</b>	0x0000 00D4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb1_ulpitll_dat5 and usbb1_ulpitll_dat6 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB1_ULPITLL_DAT6_WAKEUPEVENT	USBB1_ULPITLL_DAT6_WAKEUPENABLE	USBB1_ULPITLL_DAT6_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_DAT6_OFFMODEPULLUDENABLE	USBB1_ULPITLL_DAT6_OFFMODEOUTVALUE	USBB1_ULPITLL_DAT6_OFFMODEOUTENABLE	USBB1_ULPITLL_DAT6_OFFMODEEENABLE	USBB1_ULPITLL_DAT6_INPUTENABLE	RESERVED			USBB1_ULPITLL_DAT6_PULLTYPESELECT	USBB1_ULPITLL_DAT6_PULLUDENABLE		USBB1_ULPITLL_DAT6_MUXMODE		USBB1_ULPITLL_DAT5_WAKEUPEVENT	USBB1_ULPITLL_DAT5_WAKEUPENABLE	USBB1_ULPITLL_DAT5_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_DAT5_OFFMODEPULLUDENABLE	USBB1_ULPITLL_DAT5_OFFMODEOUTVALUE	USBB1_ULPITLL_DAT5_OFFMODEOUTENABLE	USBB1_ULPITLL_DAT5_OFFMODEEENABLE	USBB1_ULPITLL_DAT5_INPUTENABLE	RESERVED			USBB1_ULPITLL_DAT5_PULLTYPESELECT	USBB1_ULPITLL_DAT5_PULLUDENABLE		USBB1_ULPITLL_DAT5_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	USBB1_ULPITLL_DAT6_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB1_ULPITLL_DAT6_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB1_ULPITLL_DAT6_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_dat6 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB1_ULPITLL_DAT6_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_dat6 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB1_ULPITLL_DAT6_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_dat6 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB1_ULPITLL_DAT6_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_dat6. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB1_ULPITLL_DAT6_OFFMODEEENABLE	OffMode mode override control for pad usbb1_ulpitll_dat6 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB1_ULPITLL_DAT6_INPUTENABLE	Input enable value for pad usbb1_ulpitll_dat6 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20	USBB1_ULPITLL_DAT6_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_dat6 0x0: pulldown selected 0x1: pullup selected	RW	0
19	USBB1_ULPITLL_DAT6_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_dat6 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB1_ULPITLL_DAT6_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_dat6 0x0: Select usbb1_ulpitll_dat6 0x1: Select dmtimer10_pwm_evt 0x2: Select abe_mcbasp3_clkx 0x3: Select gpio_94 0x4: Select usbb1_ulpipihy_dat6 0x5: Select abe_dmic_din3 0x6: Select hw_dbg30 0x7: Select safe_mode	RW	0x7
15	USBB1_ULPITLL_DAT5_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB1_ULPITLL_DAT5_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB1_ULPITLL_DAT5_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_dat5 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB1_ULPITLL_DAT5_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_dat5 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB1_ULPITLL_DAT5_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_dat5 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB1_ULPITLL_DAT5_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_dat5. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB1_ULPITLL_DAT5_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_dat5 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	USBB1_ULPITLL_DAT5_INPUTENABLE	Input enable value for pad usbb1_ulpitll_dat5 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB1_ULPITLL_DAT5_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_dat5 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	USBB1_ULPITLL_DAT5_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_dat5 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB1_ULPITLL_DAT5_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_dat5 0x0: Select usbb1_ulpitll_dat5 0x1: Select dmtimer9_pwm_evt 0x2: Select abe_mcbasp3_dx 0x3: Select gpio_93 0x4: Select usbb1_ulpiphy_dat5 0x6: Select hw_dbg29 0x7: Select safe_mode	RW	0x7

**Table 18-397. Register Call Summary for Register CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_DAT5\_PAD1\_USBB1\_ULPITLL\_DAT6**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-398. CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_DAT7\_PAD1\_USBB1\_HSIC\_DATA**

<b>Address Offset</b>	0x0000 00D8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb1_ulpitll_dat7 and usbb1_hsic_data Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB1_HSIC_DATA_WAKEUPEVENT	USBB1_HSIC_DATA_WAKEUPENABLE	RESERVED	USBB1_HSIC_DATA_OFFMODEOUTVALUE	USBB1_HSIC_DATA_OFFMODEOUTENABLE	USBB1_HSIC_DATA_OFFMODEEENABLE	RESERVED	USBB1_HSIC_DATA_MUXMODE	USBB1_ULPITLL_DAT7_WAKEUPEVENT	USBB1_ULPITLL_DAT7_WAKEUPENABLE	USBB1_ULPITLL_DAT7_OFFMODEPULLTYPESELECT	USBB1_ULPITLL_DAT7_OFFMODEPULLUDENABLE	USBB1_ULPITLL_DAT7_OFFMODEOUTVALUE	USBB1_ULPITLL_DAT7_OFFMODEOUTENABLE	USBB1_ULPITLL_DAT7_OFFMODEEENABLE	USBB1_ULPITLL_DAT7_INPUTENABLE	RESERVED	USBB1_ULPITLL_DAT7_PULLTYPESELECT	USBB1_ULPITLL_DAT7_PULLUDENABLE	USBB1_ULPITLL_DAT7_MUXMODE												

Bits	Field Name	Description	Type	Reset
31	USBB1_HSIC_DATA_WAKEUP EVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
30	USBB1_HSIC_DATA_WAKEUP_ENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:28	RESERVED		R	0x0
27	USBB1_HSIC_DATA_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_hsic_data 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB1_HSIC_DATA_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_hsic_data. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB1_HSIC_DATA_OFFMODEENABLE	OffMode mode override control for pad usbb1_hsic_data 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24:19	RESERVED		R	0x00
18:16	USBB1_HSIC_DATA_MUXMODE	Functional multiplexing selection for pad usbb1_hsic_data 0x0: Select usbb1_hsic_data 0x3: Select gpio_96 0x7: Select safe_mode	RW	0x7
15	USBB1_ULPITLL_DAT7_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB1_ULPITLL_DAT7_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB1_ULPITLL_DAT7_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb1_ulpitll_dat7 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB1_ULPITLL_DAT7_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb1_ulpitll_dat7 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB1_ULPITLL_DAT7_OFFMODEOUTVALUE	OffMode mode output value for pad usbb1_ulpitll_dat7 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB1_ULPITLL_DAT7_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb1_ulpitll_dat7. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB1_ULPITLL_DAT7_OFFMODEENABLE	OffMode mode override control for pad usbb1_ulpitll_dat7 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
8	USBB1_ULPITLL_DAT7_INPUTENABLE	Input enable value for pad usbb1_ulpitll_dat7 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB1_ULPITLL_DAT7_PULLTYPESELECT	pullup/down selection for pad usbb1_ulpitll_dat7 0x0: pulldown selected 0x1: pullup selected	RW	0
3	USBB1_ULPITLL_DAT7_PULLUDENABLE	pullup/down enable for pad usbb1_ulpitll_dat7 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB1_ULPITLL_DAT7_MUXMODE	Functional multiplexing selection for pad usbb1_ulpitll_dat7 0x0: Select usbb1_ulpitll_dat7 0x1: Select dmtimer11_pwm_evt 0x2: Select abe_mcbasp3_fsx 0x3: Select gpio_95 0x4: Select usbb1_ulpipihy_dat7 0x5: Select abe_dmic_clk3 0x6: Select hw_dbg31 0x7: Select safe_mode	RW	0x7

**Table 18-399. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBB1\_ULPITLL\_DAT7\_PAD1\_USBB1\_HSIC\_DATA**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[3\]](#)

**Table 18-400. CONTROL\_CORE\_PAD0\_USBB1\_HSIC\_STROBE\_PAD1\_USBC1\_ICUSB\_DP**

<b>Address Offset</b>	0x0000 00DC	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb1_hsic_strobe and usbc1_icusb_dp Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBC1_ICUSB_DP_WAKEUPEVENT	USBC1_ICUSB_DP_WAKEUPENABLE	USBC1_ICUSB_DP_OFFMODEPULLTYPESELECT	USBC1_ICUSB_DP_OFFMODEPULLUDENABE	USBC1_ICUSB_DP_OFFMODEOUTVALUE	USBC1_ICUSB_DP_OFFMODEOUTENABLE	USBC1_ICUSB_DP_OFFMODEENABLE	USBC1_ICUSB_DP_INPUTENABLE	RESERVED			USBC1_ICUSB_DP_PULLTYPESELECT	USBC1_ICUSB_DP_PULLUDENABE			USBC1_ICUSB_DP_MUXMODE			USBB1_HSIC_STROBE_WAKEUPEVENT	USBB1_HSIC_STROBE_WAKEUPENABLE	RESERVED			USBB1_HSIC_STROBE_OFFMODEOUTVALUE	USBB1_HSIC_STROBE_OFFMODEOUTENABLE	USBB1_HSIC_STROBE_OFFMODEENABLE	RESERVED					USBB1_HSIC_STROBE_MUXMODE

Bits	Field Name	Description	Type	Reset
31	USBC1_ICUSB_DP_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBC1_ICUSB_DP_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBC1_ICUSB_DP_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbc1_icusb_dp 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBC1_ICUSB_DP_OFFMODEPULLUDENABE	OffMode mode pullup/down enable for pad usbc1_icusb_dp 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBC1_ICUSB_DP_OFFMODEOUTVALUE	OffMode mode output value for pad usbc1_icusb_dp 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBC1_ICUSB_DP_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbc1_icusb_dp. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBC1_ICUSB_DP_OFFMODEENABLE	OffMode mode override control for pad usbc1_icusb_dp 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBC1_ICUSB_DP_INPUTENABLE	Input enable value for pad usbc1_icusb_dp 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	USBC1_ICUSB_DP_PULLTYPESELECT	pullup/down selection for pad usbc1_icusb_dp 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
19	USBC1_ICUSB_DP_PULLUDEN ABLE	pullup/down enable for pad usbc1_icusb_dp 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBC1_ICUSB_DP_MUXMODE	Functional multiplexing selection for pad usbc1_icusb_dp 0x0: Select usbc1_icusb_dp 0x3: Select gpio_98 0x7: Select safe_mode	RW	0x7
15	USBB1_HSIC_STROBE_WAKE UPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB1_HSIC_STROBE_WAKE UPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:12	RESERVED		R	0x0
11	USBB1_HSIC_ STROBE_OFFMODEOUTVALU E	OffMode mode output value for pad usbb1_hsic_strobe 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB1_HSIC_ STROBE_OFFMODEOUTENAB LE	OffMode mode output enable value for pad usbb1_hsic_strobe. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB1_HSIC_ STROBE_OFFMODEENABLE	OffMode mode override control for pad usbb1_hsic_strobe 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8:3	RESERVED		R	0x00
2:0	USBB1_HSIC_ STROBE_MUXMODE	Functional multiplexing selection for pad usbb1_hsic_strobe 0x0: Select usbb1_hsic_strobe 0x3: Select gpio_97 0x7: Select safe_mode	RW	0x7

**Table 18-401. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBB1\_HSIC\_STROBE\_PAD1\_USBC1\_ICUSB\_DP**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)



**Table 18-402. CONTROL\_CORE\_PAD0\_USBC1\_ICUSB\_DM\_PAD1\_SDMMC1\_CLK**

<b>Address Offset</b>	0x0000 00E0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbc1_icusb_dm and sdmmc1_clk Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDMMC1_CLK_WAKEUPEVENT	SDMMC1_CLK_WAKEUPENABLE	SDMMC1_CLK_OFFMODEPULLTYPESELECT	SDMMC1_CLK_OFFMODEPULLUDENAB	SDMMC1_CLK_OFFMODEOUTVALUE	SDMMC1_CLK_OFFMODEOUTENABLE	SDMMC1_CLK_OFFMODEEENABLE	SDMMC1_CLK_INPUTENABLE	RESERVED			SDMMC1_CLK_PULLTYPESELECT	SDMMC1_CLK_PULLUDENAB			SDMMC1_CLK_MUXMODE		USBC1_ICUSB_DM_WAKEUPEVENT	USBC1_ICUSB_DM_WAKEUPENABLE	USBC1_ICUSB_DM_OFFMODEPULLTYPESELECT	USBC1_ICUSB_DM_OFFMODEPULLUDENAB	USBC1_ICUSB_DM_OFFMODEOUTVALUE	USBC1_ICUSB_DM_OFFMODEOUTENABLE	USBC1_ICUSB_DM_OFFMODEEENABLE	USBC1_ICUSB_DM_INPUTENABLE	RESERVED			USBC1_ICUSB_DM_PULLTYPESELECT	USBC1_ICUSB_DM_PULLUDENAB			USBC1_ICUSB_DM_MUXMODE

Bits	Field Name	Description	Type	Reset
31	SDMMC1_CLK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SDMMC1_CLK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SDMMC1_CLK_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_clk 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	SDMMC1_CLK_OFFMODEPULLUDENAB	OffMode mode pullup/down enable for pad sdmmc1_clk 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SDMMC1_CLK_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_clk 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SDMMC1_CLK_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_clk. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SDMMC1_CLK_OFFMODEEENABLE	OffMode mode override control for pad sdmmc1_clk 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
24	SDMMC1_CLK_INPUTENABLE	Input enable value for pad sdmmc1_clk 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SDMMC1_CLK_PULLTYPESELECT	pullup/down selection for pad sdmmc1_clk 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SDMMC1_CLK_PULLUDENABLE	pullup/down enable for pad sdmmc1_clk 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SDMMC1_CLK_MUXMODE	Functional multiplexing selection for pad sdmmc1_clk 0x0: Select sdmmc1_clk 0x2: Select dpm_emu19 0x3: Select gpio_100 0x7: Select safe_mode	RW	0x7
15	USBC1_ICUSB_DM_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBC1_ICUSB_DM_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBC1_ICUSB_DM_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbc1_icusb_dm 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBC1_ICUSB_DM_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbc1_icusb_dm 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBC1_ICUSB_DM_OFFMODEOUTVALUE	OffMode mode output value for pad usbc1_icusb_dm 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBC1_ICUSB_DM_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbc1_icusb_dm. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBC1_ICUSB_DM_OFFMODEENABLE	OffMode mode override control for pad usbc1_icusb_dm 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	USBC1_ICUSB_DM_INPUTENABLE	Input enable value for pad usbc1_icusb_dm 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBC1_ICUSB_DM_PULLTYPESELECT	pullup/down selection for pad usbc1_icusb_dm 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	USBC1_ICUSB_DM_PULLUDENABLE	pullup/down enable for pad usbc1_icusb_dm 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBC1_ICUSB_DM_MUXMODE	Functional multiplexing selection for pad usbc1_icusb_dm 0x0: Select usbc1_icusb_dm 0x3: Select gpio_99 0x7: Select safe_mode	RW	0x7

**Table 18-403. Register Call Summary for Register CONTROL\_CORE\_PAD0\_USBC1\_ICUSB\_DM\_PAD1\_SDMMC1\_CLK**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-404. CONTROL\_CORE\_PAD0\_SDMMC1\_CMD\_PAD1\_SDMMC1\_DAT0**

<b>Address Offset</b>	0x0000 00E4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads sdmmc1_cmd and sdmmc1_dat0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMMC1_DAT0_WAKEUPEVENT	SDMMC1_DAT0_WAKEUPENABLE	SDMMC1_DAT0_OFFMODEPULLTYPESELECT	SDMMC1_DAT0_OFFMODEPULLUDENABLE	SDMMC1_DAT0_OFFMODEOUTVALUE	SDMMC1_DAT0_OFFMODEOUTENABLE	SDMMC1_DAT0_OFFMODEENABLE	SDMMC1_DAT0_INPUTENABLE	RESERVED			SDMMC1_DAT0_PULLTYPESELECT	SDMMC1_DAT0_PULLUDENABLE	SDMMC1_DAT0_MUXMODE			SDMMC1_CMD_WAKEUPEVENT	SDMMC1_CMD_WAKEUPENABLE	SDMMC1_CMD_OFFMODEPULLTYPESELECT	SDMMC1_CMD_OFFMODEPULLUDENABLE	SDMMC1_CMD_OFFMODEOUTVALUE	SDMMC1_CMD_OFFMODEOUTENABLE	SDMMC1_CMD_OFFMODEENABLE	SDMMC1_CMD_INPUTENABLE	RESERVED			SDMMC1_CMD_PULLTYPESELECT	SDMMC1_CMD_PULLUDENABLE	SDMMC1_CMD_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	SDMMC1_DAT0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SDMMC1_DAT0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SDMMC1_DAT0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_dat0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
28	SDMMC1_DAT0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc1_dat0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SDMMC1_DAT0_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_dat0 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SDMMC1_DAT0_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_dat0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SDMMC1_DAT0_OFFMODEENABLE	OffMode mode override control for pad sdmmc1_dat0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	SDMMC1_DAT0_INPUTENABLE	Input enable value for pad sdmmc1_dat0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SDMMC1_DAT0_PULLTYPESELECT	pullup/down selection for pad sdmmc1_dat0 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SDMMC1_DAT0_PULLUDENABLE	pullup/down enable for pad sdmmc1_dat0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SDMMC1_DAT0_MUXMODE	Functional multiplexing selection for pad sdmmc1_dat0 0x0: Select sdmmc1_dat0 0x2: Select dpm_emu18 0x3: Select gpio_102 0x7: Select safe_mode	RW	0x7
15	SDMMC1_CMD_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SDMMC1_CMD_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	SDMMC1_CMD_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_cmd 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SDMMC1_CMD_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc1_cmd 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SDMMC1_CMD_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_cmd 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SDMMC1_CMD_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_cmd. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
9	SDMMC1_CMD_OFFMODEENABLE	OffMode mode override control for pad sdmmc1_cmd 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SDMMC1_CMD_INPUTENABLE	Input enable value for pad sdmmc1_cmd 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SDMMC1_CMD_PULLTYPESELECT	pullup/down selection for pad sdmmc1_cmd 0x0: pulldown selected 0x1: pullup selected	RW	0
3	SDMMC1_CMD_PULLUDENABLE	pullup/down enable for pad sdmmc1_cmd 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SDMMC1_CMD_MUXMODE	Functional multiplexing selection for pad sdmmc1_cmd 0x0: Select sdmmc1_cmd 0x2: Select uart1_rx 0x3: Select gpio_101 0x7: Select safe_mode	RW	0x7

**Table 18-405. Register Call Summary for Register CONTROL\_CORE\_PAD0\_SDMMC1\_CMD\_PAD1\_SDMMC1\_DAT0**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-406. CONTROL\_CORE\_PAD0\_SDMMC1\_DAT1\_PAD1\_SDMMC1\_DAT2**

<b>Address Offset</b>	0x0000 00E8
<b>Physical Address</b>	See <a href="#">Table 18-315</a>
<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads sdmmc1_dat1 and sdmmc1_dat2 Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDMMC1_DAT2_WAKEUPEVENT	SDMMC1_DAT2_WAKEUPENABLE	SDMMC1_DAT2_OFFMODEPULLTYPESELECT	SDMMC1_DAT2_OFFMODEPULLUDENABLE	SDMMC1_DAT2_OFFMODEOUTVALUE	SDMMC1_DAT2_OFFMODEOUTENABLE	SDMMC1_DAT2_OFFMODEENABLE	SDMMC1_DAT2_INPUTENABLE	RESERVED			SDMMC1_DAT2_PULLTYPESELECT	SDMMC1_DAT2_PULLUDENABLE			SDMMC1_DAT2_MUXMODE																	
																SDMMC1_DAT1_WAKEUPEVENT	SDMMC1_DAT1_WAKEUPENABLE	SDMMC1_DAT1_OFFMODEPULLTYPESELECT	SDMMC1_DAT1_OFFMODEPULLUDENABLE	SDMMC1_DAT1_OFFMODEOUTVALUE	SDMMC1_DAT1_OFFMODEOUTENABLE	SDMMC1_DAT1_OFFMODEENABLE	SDMMC1_DAT1_INPUTENABLE	RESERVED			SDMMC1_DAT1_PULLTYPESELECT	SDMMC1_DAT1_PULLUDENABLE			SDMMC1_DAT1_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	SDMMC1_DAT2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SDMMC1_DAT2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SDMMC1_DAT2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_dat2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	SDMMC1_DAT2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc1_dat2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SDMMC1_DAT2_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_dat2 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SDMMC1_DAT2_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_dat2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SDMMC1_DAT2_OFFMODEENABLE	OffMode mode override control for pad sdmmc1_dat2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	SDMMC1_DAT2_INPUTENABLE	Input enable value for pad sdmmc1_dat2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SDMMC1_DAT2_PULLTYPESELECT	pullup/down selection for pad sdmmc1_dat2 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SDMMC1_DAT2_PULLUDENABLE	pullup/down enable for pad sdmmc1_dat2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SDMMC1_DAT2_MUXMODE	Functional multiplexing selection for pad sdmmc1_dat2 0x0: Select sdmmc1_dat2 0x2: Select dpm_emu16 0x3: Select gpio_104 0x4: Select jtag_tms_tmsc 0x7: Select safe_mode	RW	0x7
15	SDMMC1_DAT1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SDMMC1_DAT1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
13	SDMMC1_DAT1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_dat1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SDMMC1_DAT1_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc1_dat1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SDMMC1_DAT1_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_dat1 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SDMMC1_DAT1_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_dat1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	SDMMC1_DAT1_OFFMODEENABLE	OffMode mode override control for pad sdmmc1_dat1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SDMMC1_DAT1_INPUTENABLE	Input enable value for pad sdmmc1_dat1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SDMMC1_DAT1_PULLTYPESELECT	pullup/down selection for pad sdmmc1_dat1 0x0: pulldown selected 0x1: pullup selected	RW	0
3	SDMMC1_DAT1_PULLUDENABLE	pullup/down enable for pad sdmmc1_dat1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SDMMC1_DAT1_MUXMODE	Functional multiplexing selection for pad sdmmc1_dat1 0x0: Select sdmmc1_dat1 0x2: Select dpm_emu17 0x3: Select gpio_103 0x7: Select safe_mode	RW	0x7

**Table 18-407. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_SDMMC1\_DAT1\_PAD1\_SDMMC1\_DAT2**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-408. CONTROL\_CORE\_PAD0\_SDMMC1\_DAT3\_PAD1\_SDMMC1\_DAT4**

<b>Address Offset</b>	0x0000 00EC		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads sdmmc1_dat3 and sdmmc1_dat4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDMMC1_DAT4_WAKEUPEVENT	SDMMC1_DAT4_WAKEUPENABLE	SDMMC1_DAT4_OFFMODEPULLTYPESELECT	SDMMC1_DAT4_OFFMODEPULLUDENENABLE	SDMMC1_DAT4_OFFMODEOUTVALUE	SDMMC1_DAT4_OFFMODEOUTENABLE	SDMMC1_DAT4_OFFMODEENABLE	SDMMC1_DAT4_INPUTENABLE	RESERVED	RESERVED	RESERVED	SDMMC1_DAT4_PULLTYPESELECT	SDMMC1_DAT4_PULLUDENENABLE	SDMMC1_DAT4_MUXMODE	SDMMC1_DAT3_WAKEUPEVENT	SDMMC1_DAT3_WAKEUPENABLE	SDMMC1_DAT3_OFFMODEPULLTYPESELECT	SDMMC1_DAT3_OFFMODEPULLUDENENABLE	SDMMC1_DAT3_OFFMODEOUTVALUE	SDMMC1_DAT3_OFFMODEOUTENABLE	SDMMC1_DAT3_OFFMODEENABLE	SDMMC1_DAT3_INPUTENABLE	RESERVED	RESERVED	RESERVED	SDMMC1_DAT3_PULLTYPESELECT	SDMMC1_DAT3_PULLUDENENABLE	SDMMC1_DAT3_MUXMODE					

Bits	Field Name	Description	Type	Reset
31	SDMMC1_DAT4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SDMMC1_DAT4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SDMMC1_DAT4_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_dat4 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	SDMMC1_DAT4_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad sdmmc1_dat4 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SDMMC1_DAT4_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_dat4 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SDMMC1_DAT4_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_dat4. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SDMMC1_DAT4_OFFMODEENABLE	OffMode mode override control for pad sdmmc1_dat4 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0



Bits	Field Name	Description	Type	Reset
24	SDMMC1_DAT4_INPUTENABLE	Input enable value for pad sdmmc1_dat4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SDMMC1_DAT4_PULLTYPESELECT	pullup/down selection for pad sdmmc1_dat4 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SDMMC1_DAT4_PULLUDENABLE	pullup/down enable for pad sdmmc1_dat4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SDMMC1_DAT4_MUXMODE	Functional multiplexing selection for pad sdmmc1_dat4 0x0: Select sdmmc1_dat4 0x3: Select gpio_106 0x7: Select safe_mode	RW	0x7
15	SDMMC1_DAT3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SDMMC1_DAT3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	SDMMC1_DAT3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_dat3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SDMMC1_DAT3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc1_dat3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SDMMC1_DAT3_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_dat3 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SDMMC1_DAT3_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_dat3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	SDMMC1_DAT3_OFFMODEENABLE	OffMode mode override control for pad sdmmc1_dat3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SDMMC1_DAT3_INPUTENABLE	Input enable value for pad sdmmc1_dat3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SDMMC1_DAT3_PULLTYPESELECT	pullup/down selection for pad sdmmc1_dat3 0x0: pulldown selected 0x1: pullup selected	RW	0
3	SDMMC1_DAT3_PULLUDENABLE	pullup/down enable for pad sdmmc1_dat3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
2:0	SDMMC1_DAT3_MUXMODE	Functional multiplexing selection for pad sdmmc1_dat3 0x0: Select sdmmc1_dat3 0x2: Select dpm_emu15 0x3: Select gpio_105 0x4: Select jtag_tck 0x7: Select safe_mode	RW	0x7

**Table 18-409. Register Call Summary for Register CONTROL\_CORE\_PAD0\_SDMMC1\_DAT3\_PAD1\_SDMMC1\_DAT4**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-410. CONTROL\_CORE\_PAD0\_SDMMC1\_DAT5\_PAD1\_SDMMC1\_DAT6**

<b>Address Offset</b>	0x0000 00F0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads sdmmc1_dat5 and sdmmc1_dat6 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMMC1_DAT6_WAKEUPEVENT	SDMMC1_DAT6_WAKEUPENABLE	SDMMC1_DAT6_OFFMODEPULLTYPESELECT	SDMMC1_DAT6_OFFMODEPULLUDENENABLE	SDMMC1_DAT6_OFFMODEOUTVALUE	SDMMC1_DAT6_OFFMODEOUTENABLE	SDMMC1_DAT6_OFFMODEENABLE	SDMMC1_DAT6_INPUTENABLE	RESERVED			SDMMC1_DAT6_PULLTYPESELECT	SDMMC1_DAT6_PULLUDENENABLE	SDMMC1_DAT6_MUXMODE			SDMMC1_DAT5_WAKEUPEVENT	SDMMC1_DAT5_WAKEUPENABLE	SDMMC1_DAT5_OFFMODEPULLTYPESELECT	SDMMC1_DAT5_OFFMODEPULLUDENENABLE	SDMMC1_DAT5_OFFMODEOUTVALUE	SDMMC1_DAT5_OFFMODEOUTENABLE	SDMMC1_DAT5_OFFMODEENABLE	SDMMC1_DAT5_INPUTENABLE	RESERVED			SDMMC1_DAT5_PULLTYPESELECT	SDMMC1_DAT5_PULLUDENENABLE	SDMMC1_DAT5_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	SDMMC1_DAT6_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SDMMC1_DAT6_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SDMMC1_DAT6_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_dat6 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
28	SDMMC1_DAT6_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc1_dat6 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SDMMC1_DAT6_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_dat6 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SDMMC1_DAT6_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_dat6. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SDMMC1_DAT6_OFFMODEENABLE	OffMode mode override control for pad sdmmc1_dat6 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	SDMMC1_DAT6_INPUTENABLE	Input enable value for pad sdmmc1_dat6 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SDMMC1_DAT6_PULLTYPESELECT	pullup/down selection for pad sdmmc1_dat6 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SDMMC1_DAT6_PULLUDENABLE	pullup/down enable for pad sdmmc1_dat6 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SDMMC1_DAT6_MUXMODE	Functional multiplexing selection for pad sdmmc1_dat6 0x0: Select sdmmc1_dat6 0x3: Select gpio_108 0x7: Select safe_mode	RW	0x7
15	SDMMC1_DAT5_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SDMMC1_DAT5_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	SDMMC1_DAT5_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_dat5 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SDMMC1_DAT5_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc1_dat5 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SDMMC1_DAT5_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_dat5 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SDMMC1_DAT5_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_dat5. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
9	SDMMC1_DAT5_OFFMODEENABLE	OffMode mode override control for pad sdmmc1_dat5 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SDMMC1_DAT5_INPUTENABLE	Input enable value for pad sdmmc1_dat5 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SDMMC1_DAT5_PULLTYPESELECT	pullup/down selection for pad sdmmc1_dat5 0x0: pulldown selected 0x1: pullup selected	RW	0
3	SDMMC1_DAT5_PULLUDENABLE	pullup/down enable for pad sdmmc1_dat5 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SDMMC1_DAT5_MUXMODE	Functional multiplexing selection for pad sdmmc1_dat5 0x0: Select sdmmc1_dat5 0x3: Select gpio_107 0x7: Select safe_mode	RW	0x7

**Table 18-411. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_SDMMC1\_DAT5\_PAD1\_SDMMC1\_DAT6**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-412. CONTROL\_CORE\_PAD0\_SDMMC1\_DAT7\_PAD1\_ABE\_MCBSP2\_CLKX**

<b>Address Offset</b>	0x0000 00F4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads sdmmc1_dat7 and abe_mcbbsp2_clkx Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_MCBSP2_CLKX_WAKEUPEVENT	ABE_MCBSP2_CLKX_WAKEUPENABLE	ABE_MCBSP2_CLKX_OFFMODEPULLTYPESELECT	ABE_MCBSP2_CLKX_OFFMODEPULLUDENABLE	ABE_MCBSP2_CLKX_OFFMODEOUTVALUE	ABE_MCBSP2_CLKX_OFFMODEOUTENABLE	ABE_MCBSP2_CLKX_OFFMODEENABLE	ABE_MCBSP2_CLKX_INPUTENABLE	RESERVED			ABE_MCBSP2_CLKX_PULLTYPESELECT	ABE_MCBSP2_CLKX_PULLUDENABLE		ABE_MCBSP2_CLKX_MUXMODE		SDMMC1_DAT7_WAKEUPEVENT	SDMMC1_DAT7_WAKEUPENABLE	SDMMC1_DAT7_OFFMODEPULLTYPESELECT	SDMMC1_DAT7_OFFMODEPULLUDENABLE	SDMMC1_DAT7_OFFMODEOUTVALUE	SDMMC1_DAT7_OFFMODEOUTENABLE	SDMMC1_DAT7_OFFMODEENABLE	SDMMC1_DAT7_INPUTENABLE	RESERVED		SDMMC1_DAT7_PULLTYPESELECT	SDMMC1_DAT7_PULLUDENABLE		SDMMC1_DAT7_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	ABE_MCBSP2_CLKX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	ABE_MCBSP2_CLKX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	ABE_MCBSP2_CLKX_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_mcbbsp2_clkx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	ABE_MCBSP2_CLKX_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_mcbbsp2_clkx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	ABE_MCBSP2_CLKX_OFFMODEOUTVALUE	OffMode mode output value for pad abe_mcbbsp2_clkx 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	ABE_MCBSP2_CLKX_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_mcbbsp2_clkx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	ABE_MCBSP2_CLKX_OFFMODEENABLE	OffMode mode override control for pad abe_mcbbsp2_clkx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	ABE_MCBSP2_CLKX_INPUTENABLE	Input enable value for pad abe_mcbbsp2_clkx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	ABE_MCBSP2_CLKX_PULLTYPESELECT	pullup/down selection for pad abe_mcbbsp2_clkx 0x0: pulldown selected 0x1: pullup selected	RW	0
19	ABE_MCBSP2_CLKX_PULLUDENABLE	pullup/down enable for pad abe_mcbbsp2_clkx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	ABE_MCBSP2_CLKX_MUXMODE	Functional multiplexing selection for pad abe_mcbbsp2_clkx 0x0: Select abe_mcbbsp2_clkx 0x1: Select mcspi2_clk 0x2: Select abe_mcaspi_ahclkx 0x3: Select gpio_110 0x4: Select usbb2_mm_rxdm 0x7: Select safe_mode	RW	0x7
15	SDMMC1_DAT7_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SDMMC1_DAT7_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
13	SDMMC1_DAT7_ OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc1_dat7 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SDMMC1_DAT7_ OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc1_dat7 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SDMMC1_DAT7_ OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc1_dat7 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SDMMC1_DAT7_ OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc1_dat7. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	SDMMC1_DAT7_ OFFMODEENABLE	OffMode mode override control for pad sdmmc1_dat7 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SDMMC1_DAT7_ INPUTENABLE	Input enable value for pad sdmmc1_dat7 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SDMMC1_DAT7_ PULLTYPESELECT	pullup/down selection for pad sdmmc1_dat7 0x0: pulldown selected 0x1: pullup selected	RW	0
3	SDMMC1_DAT7_ PULLUDENABLE	pullup/down enable for pad sdmmc1_dat7 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SDMMC1_DAT7_ MUXMODE	Functional multiplexing selection for pad sdmmc1_dat7 0x0: Select sdmmc1_dat7 0x3: Select gpio_109 0x7: Select safe_mode	RW	0x7

**Table 18-413. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_SDMMC1\_DAT7\_PAD1\_ABE\_MCBSP2\_CLKX**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-414. CONTROL\_CORE\_PAD0\_ABE\_MCBSP2\_DR\_PAD1\_ABE\_MCBSP2\_DX**

<b>Address Offset</b>	0x0000 00F8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads abe_mcbasp2_dr and abe_mcbasp2_dx Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_MCBSP2_DX_WAKEUPEVENT	ABE_MCBSP2_DX_WAKEUPENABLE	ABE_MCBSP2_DX_OFFMODEPULLTYPESELECT	ABE_MCBSP2_DX_OFFMODEPULLUDENABLE	ABE_MCBSP2_DX_OFFMODEOUTVALUE	ABE_MCBSP2_DX_OFFMODEOUTENABLE	ABE_MCBSP2_DX_OFFMODEENABLE	ABE_MCBSP2_DX_INPUTENABLE	RESERVED			ABE_MCBSP2_DX_PULLTYPESELECT	ABE_MCBSP2_DX_PULLUDENABLE		ABE_MCBSP2_DX_MUXMODE		ABE_MCBSP2_DR_WAKEUPEVENT	ABE_MCBSP2_DR_WAKEUPENABLE	ABE_MCBSP2_DR_OFFMODEPULLTYPESELECT	ABE_MCBSP2_DR_OFFMODEPULLUDENABLE	ABE_MCBSP2_DR_OFFMODEOUTVALUE	ABE_MCBSP2_DR_OFFMODEOUTENABLE	ABE_MCBSP2_DR_OFFMODEENABLE	ABE_MCBSP2_DR_INPUTENABLE	RESERVED			ABE_MCBSP2_DR_PULLTYPESELECT	ABE_MCBSP2_DR_PULLUDENABLE		ABE_MCBSP2_DR_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	ABE_MCBSP2_DX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	ABE_MCBSP2_DX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	ABE_MCBSP2_DX_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_mcbasp2_dx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	ABE_MCBSP2_DX_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_mcbasp2_dx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	ABE_MCBSP2_DX_OFFMODEOUTVALUE	OffMode mode output value for pad abe_mcbasp2_dx 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	ABE_MCBSP2_DX_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_mcbasp2_dx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	ABE_MCBSP2_DX_OFFMODEENABLE	OffMode mode override control for pad abe_mcbasp2_dx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
24	ABE_MCBSP2_DX_INPUTENABLE	Input enable value for pad abe_mcbasp2_dx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	ABE_MCBSP2_DX_PULLTYPESELECT	pullup/down selection for pad abe_mcbasp2_dx 0x0: pulldown selected 0x1: pullup selected	RW	0
19	ABE_MCBSP2_DX_PULLUDENABLE	pullup/down enable for pad abe_mcbasp2_dx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	ABE_MCBSP2_DX_MUXMODE	Functional multiplexing selection for pad abe_mcbasp2_dx 0x0: Select abe_mcbasp2_dx 0x1: Select mcspi2_simo 0x2: Select abe_mcasp_amute 0x3: Select gpio_112 0x4: Select usbb2_mm_rxcv 0x7: Select safe_mode	RW	0x7
15	ABE_MCBSP2_DR_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	ABE_MCBSP2_DR_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	ABE_MCBSP2_DR_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_mcbasp2_dr 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	ABE_MCBSP2_DR_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_mcbasp2_dr 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	ABE_MCBSP2_DR_OFFMODEOUTVALUE	OffMode mode output value for pad abe_mcbasp2_dr 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	ABE_MCBSP2_DR_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_mcbasp2_dr. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	ABE_MCBSP2_DR_OFFMODEENABLE	OffMode mode override control for pad abe_mcbasp2_dr 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	ABE_MCBSP2_DR_INPUTENABLE	Input enable value for pad abe_mcbasp2_dr 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	ABE_MCBSP2_DR_PULLTYPESELECT	pullup/down selection for pad abe_mcbasp2_dr 0x0: pulldown selected 0x1: pullup selected	RW	0



Bits	Field Name	Description	Type	Reset
3	ABE_MCBSP2_DR_PULLUDENABLE	pullup/down enable for pad abe_mcbasp2_dr 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	ABE_MCBSP2_DR_MUXMODE	Functional multiplexing selection for pad abe_mcbasp2_dr 0x0: Select abe_mcbasp2_dr 0x1: Select mcspi2_somi 0x2: Select abe_mcaspx_axr 0x3: Select gpio_111 0x4: Select usbb2_mm_rxdp 0x7: Select safe_mode	RW	0x7

**Table 18-415. Register Call Summary for Register CONTROL\_CORE\_PAD0\_ABE\_MCBSP2\_DR\_PAD1\_ABE\_MCBSP2\_DX**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-416. CONTROL\_CORE\_PAD0\_ABE\_MCBSP2\_FSX\_PAD1\_ABE\_MCBSP1\_CLKX**

<b>Address Offset</b>	0x0000 00FC	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads abe_mcbasp2_fsx and abe_mcbasp1_clkx Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_MCBSP1_CLKX_WAKEUPEVENT	ABE_MCBSP1_CLKX_WAKEUPENABLE	ABE_MCBSP1_CLKX_OFFMODEPULLTYPESELECT	ABE_MCBSP1_CLKX_OFFMODEPULLUDENABLE	ABE_MCBSP1_CLKX_OFFMODEOUTVALUE	ABE_MCBSP1_CLKX_OFFMODEOUTENABLE	ABE_MCBSP1_CLKX_OFFMODEENABLE	ABE_MCBSP1_CLKX_INPUTENABLE	RESERVED			ABE_MCBSP1_CLKX_PULLTYPESELECT	ABE_MCBSP1_CLKX_PULLUDENABLE	ABE_MCBSP1_CLKX_MUXMODE			ABE_MCBSP2_FSX_WAKEUPEVENT	ABE_MCBSP2_FSX_WAKEUPENABLE	ABE_MCBSP2_FSX_OFFMODEPULLTYPESELECT	ABE_MCBSP2_FSX_OFFMODEPULLUDENABLE	ABE_MCBSP2_FSX_OFFMODEOUTVALUE	ABE_MCBSP2_FSX_OFFMODEOUTENABLE	ABE_MCBSP2_FSX_OFFMODEENABLE	ABE_MCBSP2_FSX_INPUTENABLE	RESERVED			ABE_MCBSP2_FSX_PULLTYPESELECT	ABE_MCBSP2_FSX_PULLUDENABLE	ABE_MCBSP2_FSX_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	ABE_MCBSP1_CLKX_WAKEUP EVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
30	ABE_MCBSP1_CLKX_WAKEUP_ENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	ABE_MCBSP1_CLKX_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_mcbbsp1_clkx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	ABE_MCBSP1_CLKX_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_mcbbsp1_clkx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	ABE_MCBSP1_CLKX_OFFMODEOUTVALUE	OffMode mode output value for pad abe_mcbbsp1_clkx 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	ABE_MCBSP1_CLKX_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_mcbbsp1_clkx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	ABE_MCBSP1_CLKX_OFFMODEENABLE	OffMode mode override control for pad abe_mcbbsp1_clkx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	ABE_MCBSP1_CLKX_INPUTENABLE	Input enable value for pad abe_mcbbsp1_clkx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	ABE_MCBSP1_CLKX_PULLTYPESELECT	pullup/down selection for pad abe_mcbbsp1_clkx 0x0: pulldown selected 0x1: pullup selected	RW	0
19	ABE_MCBSP1_CLKX_PULLUDENABLE	pullup/down enable for pad abe_mcbbsp1_clkx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	ABE_MCBSP1_CLKX_MUXMODE	Functional multiplexing selection for pad abe_mcbbsp1_clkx 0x0: Select abe_mcbbsp1_clkx 0x1: Select abe_slimbus1_clock 0x3: Select gpio_114 0x7: Select safe_mode	RW	0x7
15	ABE_MCBSP2_FSX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	ABE_MCBSP2_FSX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	ABE_MCBSP2_FSX_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_mcbbsp2_fsx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	ABE_MCBSP2_FSX_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_mcbbsp2_fsx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	ABE_MCBSP2_FSX_OFFMODEOUTVALUE	OffMode mode output value for pad abe_mcbbsp2_fsx 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	ABE_MCBSP2_FSX_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_mcbbsp2_fsx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	ABE_MCBSP2_FSX_OFFMODEENABLE	OffMode mode override control for pad abe_mcbbsp2_fsx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	ABE_MCBSP2_FSX_INPUTENABLE	Input enable value for pad abe_mcbbsp2_fsx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	ABE_MCBSP2_FSX_PULLTYPESELECT	pullup/down selection for pad abe_mcbbsp2_fsx 0x0: pulldown selected 0x1: pullup selected	RW	0
3	ABE_MCBSP2_FSX_PULLUDENABLE	pullup/down enable for pad abe_mcbbsp2_fsx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	ABE_MCBSP2_FSX_MUXMODE	Functional multiplexing selection for pad abe_mcbbsp2_fsx 0x0: Select abe_mcbbsp2_fsx 0x1: Select mcspi2_cs0 0x2: Select abe_mcaspi2_afsx 0x3: Select gpio_113 0x4: Select usbb2_mm_txen 0x7: Select safe_mode	RW	0x7

**Table 18-417. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_ABE\_MCBSP2\_FSX\_PAD1\_ABE\_MCBSP1\_CLKX**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-418. CONTROL\_CORE\_PAD0\_ABE\_MCBSP1\_DR\_PAD1\_ABE\_MCBSP1\_DX**

<b>Address Offset</b>	0x0000 0100		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads abe_mcbasp1_dr and abe_mcbasp1_dx Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_MCBSP1_DX_WAKEUPEVENT	ABE_MCBSP1_DX_WAKEUPENABLE	ABE_MCBSP1_DX_OFFMODEPULLTYPESELECT	ABE_MCBSP1_DX_OFFMODEPULLUDENENABLE	ABE_MCBSP1_DX_OFFMODEOUTVALUE	ABE_MCBSP1_DX_OFFMODEOUTENABLE	ABE_MCBSP1_DX_OFFMODEENABLE	ABE_MCBSP1_DX_INPUTENABLE	RESERVED			ABE_MCBSP1_DX_PULLTYPESELECT	ABE_MCBSP1_DX_PULLUDENENABLE		ABE_MCBSP1_DX_MUXMODE		ABE_MCBSP1_DR_WAKEUPEVENT	ABE_MCBSP1_DR_WAKEUPENABLE	ABE_MCBSP1_DR_OFFMODEPULLTYPESELECT	ABE_MCBSP1_DR_OFFMODEPULLUDENENABLE	ABE_MCBSP1_DR_OFFMODEOUTVALUE	ABE_MCBSP1_DR_OFFMODEOUTENABLE	ABE_MCBSP1_DR_OFFMODEENABLE	ABE_MCBSP1_DR_INPUTENABLE	RESERVED			ABE_MCBSP1_DR_PULLTYPESELECT	ABE_MCBSP1_DR_PULLUDENENABLE		ABE_MCBSP1_DR_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	ABE_MCBSP1_DX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	ABE_MCBSP1_DX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	ABE_MCBSP1_DX_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_mcbasp1_dx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	ABE_MCBSP1_DX_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad abe_mcbasp1_dx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	ABE_MCBSP1_DX_OFFMODEOUTVALUE	OffMode mode output value for pad abe_mcbasp1_dx 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	ABE_MCBSP1_DX_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_mcbasp1_dx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	ABE_MCBSP1_DX_OFFMODEENABLE	OffMode mode override control for pad abe_mcbasp1_dx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
24	ABE_MCBSP1_ DX_INPUTENABLE	Input enable value for pad abe_mcbbsp1_dx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	ABE_MCBSP1_ DX_PULLTYPESELECT	pullup/down selection for pad abe_mcbbsp1_dx 0x0: pulldown selected 0x1: pullup selected	RW	0
19	ABE_MCBSP1_ DX_PULLUDENABLE	pullup/down enable for pad abe_mcbbsp1_dx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	ABE_MCBSP1_ DX_MUXMODE	Functional multiplexing selection for pad abe_mcbbsp1_dx 0x0: Select abe_mcbbsp1_dx 0x1: Select sdmmc3_dat2 0x2: Select abe_mcaspc_aclxx 0x3: Select gpio_116 0x7: Select safe_mode	RW	0x7
15	ABE_MCBSP1_ DR_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	ABE_MCBSP1_ DR_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	ABE_MCBSP1_ DR_OFFMODEPULLTYPESELE CT	OffMode mode pullup/down selection for pad abe_mcbbsp1_dr 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	ABE_MCBSP1_ DR_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_mcbbsp1_dr 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	ABE_MCBSP1_ DR_OFFMODEOUTVALUE	OffMode mode output value for pad abe_mcbbsp1_dr 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	ABE_MCBSP1_ DR_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_mcbbsp1_dr. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	ABE_MCBSP1_ DR_OFFMODEENABLE	OffMode mode override control for pad abe_mcbbsp1_dr 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	ABE_MCBSP1_ DR_INPUTENABLE	Input enable value for pad abe_mcbbsp1_dr 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	ABE_MCBSP1_ DR_PULLTYPESELECT	pullup/down selection for pad abe_mcbbsp1_dr 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	ABE_MCBSP1_DR_PULLUDENABLE	pullup/down enable for pad abe_mcbasp1_dr 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	ABE_MCBSP1_DR_MUXMODE	Functional multiplexing selection for pad abe_mcbasp1_dr 0x0: Select abe_mcbasp1_dr 0x1: Select abe_slimbus1_data 0x3: Select gpio_115 0x7: Select safe_mode	RW	0x7

**Table 18-419. Register Call Summary for Register CONTROL\_CORE\_PAD0\_ABE\_MCBSP1\_DR\_PAD1\_ABE\_MCBSP1\_DX**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-420. CONTROL\_CORE\_PAD0\_ABE\_MCBSP1\_FSX\_PAD1\_ABE\_PDM\_UL\_DATA**

<b>Address Offset</b>	0x0000 0104	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads abe_mcbasp1_fsx and abe_pdm_ul_data Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ABE_PDM_UL_DATA_WAKEUPEVENT	ABE_PDM_UL_DATA_WAKEUPENABLE	RESERVED						ABE_PDM_UL_DATA_INPUTENABLE	RESERVED				ABE_PDM_UL_DATA_PULLTYPESELECT	ABE_PDM_UL_DATA_PULLUDENABLE	ABE_PDM_UL_DATA_MUXMODE			ABE_MCBSP1_FSX_WAKEUPEVENT	ABE_MCBSP1_FSX_WAKEUPENABLE	ABE_MCBSP1_FSX_OFFMODEPULLTYPESELECT	ABE_MCBSP1_FSX_OFFMODEPULLUDENABLE	ABE_MCBSP1_FSX_OFFMODEOUTVALUE	ABE_MCBSP1_FSX_OFFMODEOUTENABLE	ABE_MCBSP1_FSX_OFFMODEENABLE	ABE_MCBSP1_FSX_INPUTENABLE	RESERVED				ABE_MCBSP1_FSX_PULLTYPESELECT	ABE_MCBSP1_FSX_PULLUDENABLE	ABE_MCBSP1_FSX_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	ABE_PDM_UL_DATA_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	ABE_PDM_UL_DATA_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
24	ABE_PDM_UL_DATA_INPUTENABLE	Input enable value for pad abe_pdm_ul_data 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	ABE_PDM_UL_DATA_PULLTYPESELECT	pullup/down selection for pad abe_pdm_ul_data 0x0: pulldown selected 0x1: pullup selected	RW	0
19	ABE_PDM_UL_DATA_PULLUDENABLE	pullup/down enable for pad abe_pdm_ul_data 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	ABE_PDM_UL_DATA_MUXMODE	Functional multiplexing selection for pad abe_pdm_ul_data 0x0: Select abe_pdm_ul_data 0x1: Select abe_mcbasp3_dr 0x7: Select safe_mode	RW	0x7
15	ABE_MCBSP1_FSX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	ABE_MCBSP1_FSX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	ABE_MCBSP1_FSX_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_mcbasp1_fsx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	ABE_MCBSP1_FSX_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_mcbasp1_fsx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	ABE_MCBSP1_FSX_OFFMODEOUTVALUE	OffMode mode output value for pad abe_mcbasp1_fsx 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	ABE_MCBSP1_FSX_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_mcbasp1_fsx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	ABE_MCBSP1_FSX_OFFMODEENABLE	OffMode mode override control for pad abe_mcbasp1_fsx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	ABE_MCBSP1_FSX_INPUTENABLE	Input enable value for pad abe_mcbasp1_fsx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	ABE_MCBSP1_FSX_PULLTYPESELECT	pullup/down selection for pad abe_mcbasp1_fsx 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	ABE_MCBSP1_FSX_PULLUENABLE	pullup/down enable for pad abe_mcbasp1_fsx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	ABE_MCBSP1_FSX_MUXMODE	Functional multiplexing selection for pad abe_mcbasp1_fsx 0x0: Select abe_mcbasp1_fsx 0x1: Select sdmmc3_dat3 0x2: Select abe_mcasp_amutein 0x3: Select gpio_117 0x7: Select safe_mode	RW	0x7

**Table 18-421. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_ABE\_MCBSP1\_FSX\_PAD1\_ABE\_PDM\_UL\_DATA**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-422. CONTROL\_CORE\_PAD0\_ABE\_PDM\_DL\_DATA\_PAD1\_ABE\_PDM\_FRAME**

<b>Address Offset</b>	0x0000 0108	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads abe_pdm_dl_data and abe_pdm_frame Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_PDM_FRAME_WAKEUPEVENT	ABE_PDM_FRAME_WAKEUPENABLE	ABE_PDM_FRAME_OFFMODEPULLTYPESELECT	ABE_PDM_FRAME_OFFMODEPULLUENABLE	ABE_PDM_FRAME_OFFMODEOUTVALUE	ABE_PDM_FRAME_OFFMODEOUTENABLE	ABE_PDM_FRAME_OFFMODEENABLE	ABE_PDM_FRAME_INPUTENABLE	RESERVED			ABE_PDM_FRAME_PULLTYPESELECT	ABE_PDM_FRAME_PULLUENABLE	ABE_PDM_FRAME_MUXMODE			ABE_PDM_DL_DATA_WAKEUPEVENT	ABE_PDM_DL_DATA_WAKEUPENABLE	ABE_PDM_DL_DATA_OFFMODEPULLTYPESELECT	ABE_PDM_DL_DATA_OFFMODEPULLUENABLE	ABE_PDM_DL_DATA_OFFMODEOUTVALUE	ABE_PDM_DL_DATA_OFFMODEOUTENABLE	ABE_PDM_DL_DATA_OFFMODEENABLE	ABE_PDM_DL_DATA_INPUTENABLE	RESERVED			ABE_PDM_DL_DATA_PULLTYPESELECT	ABE_PDM_DL_DATA_PULLUENABLE	ABE_PDM_DL_DATA_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	ABE_PDM_FRAME_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0



Bits	Field Name	Description	Type	Reset
30	ABE_PDM_FRAME_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	ABE_PDM_FRAME_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad <code>abe_pdm_frame</code> 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	ABE_PDM_FRAME_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad <code>abe_pdm_frame</code> 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	ABE_PDM_FRAME_OFFMODEOUTVALUE	OffMode mode output value for pad <code>abe_pdm_frame</code> 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	ABE_PDM_FRAME_OFFMODEOUTENABLE	OffMode mode output enable value for pad <code>abe_pdm_frame</code> . This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	ABE_PDM_FRAME_OFFMODEENABLE	OffMode mode override control for pad <code>abe_pdm_frame</code> 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	ABE_PDM_FRAME_INPUTENABLE	Input enable value for pad <code>abe_pdm_frame</code> 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	ABE_PDM_FRAME_PULLTYPESELECT	pullup/down selection for pad <code>abe_pdm_frame</code> 0x0: pulldown selected 0x1: pullup selected	RW	0
19	ABE_PDM_FRAME_PULLUDENABLE	pullup/down enable for pad <code>abe_pdm_frame</code> 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	ABE_PDM_FRAME_MUXMODE	Functional multiplexing selection for pad <code>abe_pdm_frame</code> 0x0: Select <code>abe_pdm_frame</code> 0x1: Select <code>abe_mcbasp3_clkx</code> 0x7: Select <code>safe_mode</code>	RW	0x7
15	ABE_PDM_DL_DATA_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	ABE_PDM_DL_DATA_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	ABE_PDM_DL_DATA_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad <code>abe_pdm_dl_data</code> 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	ABE_PDM_DL_DATA_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_pdm_dl_data 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	ABE_PDM_DL_DATA_OFFMODEOUTVALUE	OffMode mode output value for pad abe_pdm_dl_data 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	ABE_PDM_DL_DATA_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_pdm_dl_data. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	ABE_PDM_DL_DATA_OFFMODEENABLE	OffMode mode override control for pad abe_pdm_dl_data 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	ABE_PDM_DL_DATA_INPUTENABLE	Input enable value for pad abe_pdm_dl_data 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	ABE_PDM_DL_DATA_PULLTYPESELECT	pullup/down selection for pad abe_pdm_dl_data 0x0: pulldown selected 0x1: pullup selected	RW	0
3	ABE_PDM_DL_DATA_PULLUDENABLE	pullup/down enable for pad abe_pdm_dl_data 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	ABE_PDM_DL_DATA_MUXMODE	Functional multiplexing selection for pad abe_pdm_dl_data 0x0: Select abe_pdm_dl_data 0x1: Select abe_mcbasp3_dx 0x7: Select safe_mode	RW	0x7

**Table 18-423. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_ABE\_PDM\_DL\_DATA\_PAD1\_ABE\_PDM\_FRAME**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-424. CONTROL\_CORE\_PAD0\_ABE\_PDM\_LB\_CLK\_PAD1\_ABE\_CLKS**

<b>Address Offset</b>	0x0000 010C		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads abe_pdm_lb_clk and abe_clks Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_CLKS_WAKEUPEVENT	ABE_CLKS_WAKEUPENABLE	ABE_CLKS_OFFMODEPULLTYPESELECT	ABE_CLKS_OFFMODEPULLUDENABLE	ABE_CLKS_OFFMODEOUTVALUE	ABE_CLKS_OFFMODEOUTENABLE	ABE_CLKS_OFFMODEENABLE	ABE_CLKS_INPUTENABLE	RESERVED			ABE_CLKS_PULLTYPESELECT	ABE_CLKS_PULLUDENABLE	ABE_CLKS_MUXMODE			ABE_PDM_LB_CLK_WAKEUPEVENT	ABE_PDM_LB_CLK_WAKEUPENABLE	ABE_PDM_LB_CLK_OFFMODEPULLTYPESELECT	ABE_PDM_LB_CLK_OFFMODEPULLUDENABLE	ABE_PDM_LB_CLK_OFFMODEOUTVALUE	ABE_PDM_LB_CLK_OFFMODEOUTENABLE	ABE_PDM_LB_CLK_OFFMODEENABLE	ABE_PDM_LB_CLK_INPUTENABLE	RESERVED			ABE_PDM_LB_CLK_PULLTYPESELECT	ABE_PDM_LB_CLK_PULLUDENABLE	ABE_PDM_LB_CLK_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	ABE_CLKS_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	ABE_CLKS_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	ABE_CLKS_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_clks 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	ABE_CLKS_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_clks 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	ABE_CLKS_OFFMODEOUTVALUE	OffMode mode output value for pad abe_clks 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	ABE_CLKS_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_clks. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	ABE_CLKS_OFFMODEENABLE	OffMode mode override control for pad abe_clks 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	ABE_CLKS_INPUTENABLE	Input enable value for pad abe_clks 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	ABE_CLKS_PULLTYPESELECT	pullup/down selection for pad abe_clks 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
19	ABE_CLKS_PULLUDENABLE	pullup/down enable for pad abe_clks 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	ABE_CLKS_MUXMODE	Functional multiplexing selection for pad abe_clks 0x0: Select abe_clks 0x3: Select gpio_118 0x7: Select safe_mode	RW	0x7
15	ABE_PDM_LB_CLK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	ABE_PDM_LB_CLK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	ABE_PDM_LB_CLK_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_pdm_lb_clk 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	ABE_PDM_LB_CLK_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_pdm_lb_clk 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	ABE_PDM_LB_CLK_OFFMODEOUTVALUE	OffMode mode output value for pad abe_pdm_lb_clk 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	ABE_PDM_LB_CLK_OFFMODEOUTENABLE	OffMode mode output enable value for pad abe_pdm_lb_clk. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	ABE_PDM_LB_CLK_OFFMODEENABLE	OffMode mode override control for pad abe_pdm_lb_clk 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	ABE_PDM_LB_CLK_INPUTENABLE	Input enable value for pad abe_pdm_lb_clk 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	ABE_PDM_LB_CLK_PULLTYPESELECT	pullup/down selection for pad abe_pdm_lb_clk 0x0: pulldown selected 0x1: pullup selected	RW	0
3	ABE_PDM_LB_CLK_PULLUDENABLE	pullup/down enable for pad abe_pdm_lb_clk 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	ABE_PDM_LB_CLK_MUXMODE	Functional multiplexing selection for pad abe_pdm_lb_clk 0x0: Select abe_pdm_lb_clk 0x1: Select abe_mcbasp3_fsx 0x7: Select safe_mode	RW	0x7

**Table 18-425. Register Call Summary for Register CONTROL\_CORE\_PAD0\_ABE\_PDM\_LB\_CLK\_PAD1\_ABE\_CLKS**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-426. CONTROL\_CORE\_PAD0\_ABE\_DMIC\_CLK1\_PAD1\_ABE\_DMIC\_DIN1**

<b>Address Offset</b>	0x0000 0110	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads abe_dmic_clk1 and abe_dmic_din1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_DMIC_DIN1_WAKEUPEVENT	ABE_DMIC_DIN1_WAKEUPENABLE	ABE_DMIC_DIN1_OFFMODEPULLTYPESELECT	ABE_DMIC_DIN1_OFFMODEPULLUDENENABLE	ABE_DMIC_DIN1_OFFMODEOUTVALUE	ABE_DMIC_DIN1_OFFMODEOUTENABLE	ABE_DMIC_DIN1_OFFMODEENABLE	ABE_DMIC_DIN1_INPUTENABLE	RESERVED			ABE_DMIC_DIN1_PULLTYPESELECT	ABE_DMIC_DIN1_PULLUDENENABLE	ABE_DMIC_DIN1_MUXMODE			ABE_DMIC_CLK1_WAKEUPEVENT	ABE_DMIC_CLK1_WAKEUPENABLE	ABE_DMIC_CLK1_OFFMODEPULLTYPESELECT	ABE_DMIC_CLK1_OFFMODEPULLUDENENABLE	ABE_DMIC_CLK1_OFFMODEOUTVALUE	ABE_DMIC_CLK1_OFFMODEOUTENABLE	ABE_DMIC_CLK1_OFFMODEENABLE	ABE_DMIC_CLK1_INPUTENABLE	RESERVED			ABE_DMIC_CLK1_PULLTYPESELECT	ABE_DMIC_CLK1_PULLUDENENABLE	ABE_DMIC_CLK1_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	ABE_DMIC_DIN1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	ABE_DMIC_DIN1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	ABE_DMIC_DIN1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_dmic_din1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	ABE_DMIC_DIN1_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad abe_dmic_din1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	ABE_DMIC_DIN1_OFFMODEOUTVALUE	OffMode mode output value for pad abe_dmic_din1 0x0: Set value at 0 0x1: Set value at 1	RW	0

Bits	Field Name	Description	Type	Reset
26	ABE_DMIC_DIN1_OFFMODEO UTENABLE	OffMode mode output enable value for pad abe_dmic_din1. This is an active low signal  0x0: Output enable 0x1: Output disable	RW	0
25	ABE_DMIC_DIN1_OFFMODEEEN ABLE	OffMode mode override control for pad abe_dmic_din1  0x0: I/O state keeps its previous state when OFF mode is active  0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	ABE_DMIC_DIN1_INPUTENABL E	Input enable value for pad abe_dmic_din1  0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	ABE_DMIC_DIN1_PULLTYPESE LECT	pullup/down selection for pad abe_dmic_din1  0x0: pulldown selected 0x1: pullup selected	RW	0
19	ABE_DMIC_DIN1_PULLUDENA BLE	pullup/down enable for pad abe_dmic_din1  0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	ABE_DMIC_DIN1_MUXMODE	Functional multiplexing selection for pad abe_dmic_din1  0x0: Select abe_dmic_din1 0x3: Select gpio_120 0x4: Select usbb2_mm_txdat 0x5: Select uart4_rts 0x7: Select safe_mode	RW	0x7
15	ABE_DMIC_CLK1_WAKEUPEV ENT	Pad_x wake-up event status latched in the IO  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	ABE_DMIC_CLK1_WAKEUPEN ABLE	Input pad wake-up enable (and OFF mode input enable value)  0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	ABE_DMIC_ CLK1_OFFMODEPULLTYPESE LECT	OffMode mode pullup/down selection for pad abe_dmic_clk1  0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	ABE_DMIC_ CLK1_OFFMODEPULLUDENAB LE	OffMode mode pullup/down enable for pad abe_dmic_clk1  0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	ABE_DMIC_CLK1_OFFMODEO UTVALUE	OffMode mode output value for pad abe_dmic_clk1  0x0: Set value at 0 0x1: Set value at 1	RW	0
10	ABE_DMIC_CLK1_OFFMODEO UTENABLE	OffMode mode output enable value for pad abe_dmic_clk1. This is an active low signal  0x0: Output enable 0x1: Output disable	RW	0
9	ABE_DMIC_CLK1_OFFMODEE NABLE	OffMode mode override control for pad abe_dmic_clk1  0x0: I/O state keeps its previous state when OFF mode is active  0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
8	ABE_DMIC_CLK1_INPUTENABLE	Input enable value for pad abe_dmic_clk1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	ABE_DMIC_CLK1_PULLTYPES ELECT	pullup/down selection for pad abe_dmic_clk1 0x0: pulldown selected 0x1: pullup selected	RW	0
3	ABE_DMIC_CLK1_PULLUDENABLE	pullup/down enable for pad abe_dmic_clk1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	ABE_DMIC_CLK1_MUXMODE	Functional multiplexing selection for pad abe_dmic_clk1 0x0: Select abe_dmic_clk1 0x3: Select gpio_119 0x4: Select usb2_mm_txse0 0x5: Select uart4_cts 0x7: Select safe_mode	RW	0x7

**Table 18-427. Register Call Summary for Register CONTROL\_CORE\_PAD0\_ABE\_DMIC\_CLK1\_PAD1\_ABE\_DMIC\_DIN1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-428. CONTROL\_CORE\_PAD0\_ABE\_DMIC\_DIN2\_PAD1\_ABE\_DMIC\_DIN3**

<b>Address Offset</b>	0x0000 0114	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads abe_dmic_din2 and abe_dmic_din3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ABE_DMIC_DIN3_WAKEUPEVENT	ABE_DMIC_DIN3_WAKEUPENABLE	ABE_DMIC_DIN3_OFFMODEPULLTYPESELECT	ABE_DMIC_DIN3_OFFMODEPULLUDENABLE	ABE_DMIC_DIN3_OFFMODEOUTVALUE	ABE_DMIC_DIN3_OFFMODEOUTENABLE	ABE_DMIC_DIN3_OFFMODEENABLE	ABE_DMIC_DIN3_INPUTENABLE	RESERVED			ABE_DMIC_DIN3_PULLTYPESELECT	ABE_DMIC_DIN3_PULLUDENABLE			ABE_DMIC_DIN3_MUXMODE			ABE_DMIC_DIN2_WAKEUPEVENT	ABE_DMIC_DIN2_WAKEUPENABLE	ABE_DMIC_DIN2_OFFMODEPULLTYPESELECT	ABE_DMIC_DIN2_OFFMODEPULLUDENABLE	ABE_DMIC_DIN2_OFFMODEOUTVALUE	ABE_DMIC_DIN2_OFFMODEOUTENABLE	ABE_DMIC_DIN2_OFFMODEENABLE	ABE_DMIC_DIN2_INPUTENABLE	RESERVED			ABE_DMIC_DIN2_PULLTYPESELECT	ABE_DMIC_DIN2_PULLUDENABLE			ABE_DMIC_DIN2_MUXMODE

Bits	Field Name	Description	Type	Reset
31	ABE_DMIC_DIN3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	ABE_DMIC_DIN3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	ABE_DMIC_DIN3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_dmic_din3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	ABE_DMIC_DIN3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_dmic_din3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	ABE_DMIC_DIN3_OFFMODEOUTPUTVALUE	OffMode mode output value for pad abe_dmic_din3 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	ABE_DMIC_DIN3_OFFMODEOUTPUTENABLE	OffMode mode output enable value for pad abe_dmic_din3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	ABE_DMIC_DIN3_OFFMODEOVERRIDEENABLE	OffMode mode override control for pad abe_dmic_din3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	ABE_DMIC_DIN3_INPUTENABLE	Input enable value for pad abe_dmic_din3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	ABE_DMIC_DIN3_PULLTYPESELECT	pullup/down selection for pad abe_dmic_din3 0x0: pulldown selected 0x1: pullup selected	RW	0
19	ABE_DMIC_DIN3_PULLUDENABLE	pullup/down enable for pad abe_dmic_din3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	ABE_DMIC_DIN3_MUXMODE	Functional multiplexing selection for pad abe_dmic_din3 0x0: Select abe_dmic_din3 0x1: Select slimbus2_data 0x2: Select abe_dmic_clk2 0x3: Select gpio_122 0x5: Select dmtimer9_pwm_evt 0x7: Select safe_mode	RW	0x7
15	ABE_DMIC_DIN2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	ABE_DMIC_DIN2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0



Bits	Field Name	Description	Type	Reset
13	ABE_DMIC_DIN2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad abe_dmic_din2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	ABE_DMIC_DIN2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad abe_dmic_din2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	ABE_DMIC_DIN2_OFFMODEOUTPUTVALUE	OffMode mode output value for pad abe_dmic_din2 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	ABE_DMIC_DIN2_OFFMODEOUTPUTENABLE	OffMode mode output enable value for pad abe_dmic_din2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	ABE_DMIC_DIN2_OFFMODEOVERRIDEENABLE	OffMode mode override control for pad abe_dmic_din2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	ABE_DMIC_DIN2_INPUTENABLE	Input enable value for pad abe_dmic_din2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	ABE_DMIC_DIN2_PULLTYPESELECT	pullup/down selection for pad abe_dmic_din2 0x0: pulldown selected 0x1: pullup selected	RW	0
3	ABE_DMIC_DIN2_PULLUDENABLE	pullup/down enable for pad abe_dmic_din2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	ABE_DMIC_DIN2_MUXMODE	Functional multiplexing selection for pad abe_dmic_din2 0x0: Select abe_dmic_din2 0x1: Select slimbus2_clock 0x2: Select abe_mcaspx_axr 0x3: Select gpio_121 0x5: Select dmtimer11_pwm_evt 0x7: Select safe_mode	RW	0x7

**Table 18-429. Register Call Summary for Register CONTROL\_CORE\_PAD0\_ABE\_DMIC\_DIN2\_PAD1\_ABE\_DMIC\_DIN3**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-430. CONTROL\_CORE\_PAD0\_UART2\_CTS\_PAD1\_UART2\_RTS**

<b>Address Offset</b>	0x0000 0118		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads uart2_cts and uart2_rts Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART2_RTS_WAKEUPEVENT	UART2_RTS_WAKEUPENABLE	UART2_RTS_OFFMODEPULLTYPESELECT	UART2_RTS_OFFMODEPULLUDENENABLE	UART2_RTS_OFFMODEOUTVALUE	UART2_RTS_OFFMODEOUTENABLE	UART2_RTS_OFFMODEENABLE	UART2_RTS_INPUTENABLE	RESERVED			UART2_RTS_PULLTYPESELECT	UART2_RTS_PULLUDENENABLE		UART2_RTS_MUXMODE		UART2_CTS_WAKEUPEVENT	UART2_CTS_WAKEUPENABLE	UART2_CTS_OFFMODEPULLTYPESELECT	UART2_CTS_OFFMODEPULLUDENENABLE	UART2_CTS_OFFMODEOUTVALUE	UART2_CTS_OFFMODEOUTENABLE	UART2_CTS_OFFMODEENABLE	UART2_CTS_INPUTENABLE	RESERVED			UART2_CTS_PULLTYPESELECT	UART2_CTS_PULLUDENENABLE		UART2_CTS_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	UART2_RTS_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	UART2_RTS_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	UART2_RTS_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad uart2_rts 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	UART2_RTS_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad uart2_rts 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	UART2_RTS_OFFMODEOUTVALUE	OffMode mode output value for pad uart2_rts 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	UART2_RTS_OFFMODEOUTENABLE	OffMode mode output enable value for pad uart2_rts. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	UART2_RTS_OFFMODEENABLE	OffMode mode override control for pad uart2_rts 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	UART2_RTS_INPUTENABLE	Input enable value for pad uart2_rts 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1

Bits	Field Name	Description	Type	Reset
23:21	RESERVED		R	0x0
20	UART2_RTS_PULLTYPESELECTION	pullup/down selection for pad uart2_rts 0x0: pulldown selected 0x1: pullup selected	RW	1
19	UART2_RTS_PULLUDENABLE	pullup/down enable for pad uart2_rts 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	UART2_RTS_MUXMODE	Functional multiplexing selection for pad uart2_rts 0x0: Select uart2_rts 0x1: Select sdmmc3_cmd 0x3: Select gpio_124 0x7: Select safe_mode	RW	0x7
15	UART2_CTS_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	UART2_CTS_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	UART2_CTS_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad uart2_cts 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	UART2_CTS_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad uart2_cts 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	UART2_CTS_OFFMODEOUTVALUE	OffMode mode output value for pad uart2_cts 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	UART2_CTS_OFFMODEOUTENABLE	OffMode mode output enable value for pad uart2_cts. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	UART2_CTS_OFFMODEENABLE	OffMode mode override control for pad uart2_cts 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	UART2_CTS_INPUTENABLE	Input enable value for pad uart2_cts 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	UART2_CTS_PULLTYPESELECTION	pullup/down selection for pad uart2_cts 0x0: pulldown selected 0x1: pullup selected	RW	1
3	UART2_CTS_PULLUDENABLE	pullup/down enable for pad uart2_cts 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
2:0	UART2_CTS_MUXMODE	Functional multiplexing selection for pad uart2_cts 0x0: Select uart2_cts 0x1: Select sdmmc3_clk 0x3: Select gpio_123 0x7: Select safe_mode	RW	0x7

**Table 18-431. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_UART2\_CTS\_PAD1\_UART2\_RTS**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Programming Guide

- [Pad Configuration Programming Points: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[10\]](#)

**Table 18-432. CONTROL\_CORE\_PAD0\_UART2\_RX\_PAD1\_UART2\_TX**

<b>Address Offset</b>	0x0000 011C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads uart2_rx and uart2_tx Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART2_TX_WAKEUPEVENT	UART2_TX_WAKEUPENABLE	UART2_TX_OFFMODEPULLTYPESELECT	UART2_TX_OFFMODEPULLUDENENABLE	UART2_TX_OFFMODEOUTVALUE	UART2_TX_OFFMODEOUTENABLE	UART2_TX_OFFMODEENABLE	UART2_TX_INPUTENABLE	RESERVED			UART2_TX_PULLTYPESELECT	UART2_TX_PULLUDENENABLE	UART2_TX_MUXMODE			UART2_RX_WAKEUPEVENT	UART2_RX_WAKEUPENABLE	UART2_RX_OFFMODEPULLTYPESELECT	UART2_RX_OFFMODEPULLUDENENABLE	UART2_RX_OFFMODEOUTVALUE	UART2_RX_OFFMODEOUTENABLE	UART2_RX_OFFMODEENABLE	UART2_RX_INPUTENABLE	RESERVED			UART2_RX_PULLTYPESELECT	UART2_RX_PULLUDENENABLE	UART2_RX_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	UART2_TX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	UART2_TX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	UART2_TX_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad uart2_tx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
28	UART2_TX_OFFMODEPULLUD ENABLE	OffMode mode pullup/down enable for pad uart2_tx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	UART2_TX_OFFMODEOUTVAL UE	OffMode mode output value for pad uart2_tx 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	UART2_TX_OFFMODEOUTENA BLE	OffMode mode output enable value for pad uart2_tx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	UART2_TX_OFFMODEENABLE	OffMode mode override control for pad uart2_tx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	UART2_TX_INPUTENABLE	Input enable value for pad uart2_tx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	UART2_TX_PULLTYPESELECT	pullup/down selection for pad uart2_tx 0x0: pulldown selected 0x1: pullup selected	RW	1
19	UART2_TX_PULLUDENABLE	pullup/down enable for pad uart2_tx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	UART2_TX_MUXMODE	Functional multiplexing selection for pad uart2_tx 0x0: Select uart2_tx 0x1: Select sdmmc3_dat1 0x3: Select gpio_126 0x7: Select safe_mode	RW	0x7
15	UART2_RX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	UART2_RX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	UART2_RX_ OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad uart2_rx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	UART2_RX_ OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad uart2_rx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	UART2_RX_OFFMODEOUTVAL UE	OffMode mode output value for pad uart2_rx 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	UART2_RX_OFFMODEOUTENA BLE	OffMode mode output enable value for pad uart2_rx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
9	UART2_RX_OFFMODEENABLE	OffMode mode override control for pad uart2_rx  0x0: I/O state keeps its previous state when OFF mode is active  0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	UART2_RX_INPUTENABLE	Input enable value for pad uart2_rx  0x0: Input buffere of I/O cell is disabled  0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	UART2_RX_PULLTYPESELECT	pullup/down selection for pad uart2_rx  0x0: pulldown selected  0x1: pullup selected	RW	1
3	UART2_RX_PULLUDENABLE	pullup/down enable for pad uart2_rx  0x0: pullup/down disabled  0x1: pullup/down enabled	RW	1
2:0	UART2_RX_MUXMODE	Functional multiplexing selection for pad uart2_rx  0x0: Select uart2_rx  0x1: Select sdmmc3_dat0  0x3: Select gpio_125  0x7: Select safe_mode	RW	0x7

**Table 18-433. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_UART2\_RX\_PAD1\_UART2\_TX**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Programming Guide

- [Pad Configuration Programming Points: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[10\]](#)

**Table 18-434. CONTROL\_CORE\_PAD0\_HDQ\_SIO\_PAD1\_I2C1\_SCL**

<b>Address Offset</b>	0x0000 0120	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads hdq_sio and i2c1_scl Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C1_SCL_WAKEUPEVENT	I2C1_SCL_WAKEUPEENABLE	RESERVED						I2C1_SCL_INPUTENABLE	RESERVED			I2C1_SCL_PULLTYPESELECT	I2C1_SCL_PULLUDENABLE	RESERVED			HDQ_SIO_WAKEUPEVENT	HDQ_SIO_WAKEUPEENABLE	HDQ_SIO_OFFMODEPULLTYPESELECT	HDQ_SIO_OFFMODEPULLUDENABLE	HDQ_SIO_OFFMODEOUTVALUE	HDQ_SIO_OFFMODEOUTENABLE	HDQ_SIO_OFFMODEENABLE	HDQ_SIO_INPUTENABLE	RESERVED			HDQ_SIO_PULLTYPESELECT	HDQ_SIO_PULLUDENABLE	HDQ_SIO_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	I2C1_SCL_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	I2C1_SCL_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	I2C1_SCL_INPUTENABLE	Input enable value for pad i2c1_scl 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	I2C1_SCL_PULLTYPESELECT	pullup/down selection for pad i2c1_scl 0x0: pulldown selected 0x1: pullup selected	RW	1
19	I2C1_SCL_PULLUDENABLE	pullup/down enable for pad i2c1_scl 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	RESERVED		R	0x0
15	HDQ_SIO_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	HDQ_SIO_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	HDQ_SIO_OFFMODEPULLTYPESELECT	OffMode mode Pull-Up/Down selection for pad hdq_sio 0x0: Offmode Pull-Down selected 0x1: Offmode Pull-Up selected	RW	0x0
12	HDQ_SIO_OFFMODEPULLUDENABLE	OffMode mode Pull-Up/Down enable for pad hdq_sio 0x0: Offmode Pull-Up/Down disabled 0x1: Offmode Pull-Up/Down enabled	RW	0x0
11	HDQ_SIO_OFFMODEOUTVALUE	OffMode mode output value for pad hdq_sio 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	HDQ_SIO_OFFMODEOUTENABLE	OffMode mode output enable value for pad hdq_sio. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	HDQ_SIO_OFFMODEENABLE	OffMode mode override control for pad hdq_sio 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	HDQ_SIO_INPUTENABLE	Input enable value for pad hdq_sio 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	HDQ_SIO_PULLTYPESELECT	Pull-Up/Down selection for pad hdq_sio 0x0: Pull-Down selected 0x1: Pull-Up selected	RW	0x0

Bits	Field Name	Description	Type	Reset
3		0x0: Pull-Up/Down disabled 0x1: Pull-Up/Down enabled		0x0
2:0	HDQ_SIO_MUXMODE	Functional multiplexing selection for pad hdq_sio 0x0: Select hdq_sio 0x1: Select i2c3_sccb 0x2: Select i2c2_sccb 0x3: Select gpio_127 0x7: Select safe_mode	RW	0x7

**Table 18-435. Register Call Summary for Register CONTROL\_CORE\_PAD0\_HDQ\_SIO\_PAD1\_I2C1\_SCL**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-436. CONTROL\_CORE\_PAD0\_I2C1\_SDA\_PAD1\_I2C2\_SCL**

<b>Address Offset</b>	0x0000 0124	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads i2c1_sda and i2c2_scl Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C2_SCL_WAKEUPEVENT	I2C2_SCL_WAKEUPENABLE	I2C2_SCL_OFFMODEPULLTYPESELECT	I2C2_SCL_OFFMODEPULLUDENABE	I2C2_SCL_OFFMODEOUTVALUE	I2C2_SCL_OFFMODEOUTENABLE	I2C2_SCL_OFFMODEENABLE	I2C2_SCL_INPUTENABLE	RESERVED			I2C2_SCL_PULLTYPESELECT	I2C2_SCL_PULLUDENABE	I2C2_SCL_MUXMODE			I2C1_SDA_WAKEUPEVENT	I2C1_SDA_WAKEUPENABLE	RESERVED					I2C1_SDA_INPUTENABLE	RESERVED			I2C1_SDA_PULLTYPESELECT	I2C1_SDA_PULLUDENABE	RESERVED		

Bits	Field Name	Description	Type	Reset
31	I2C2_SCL_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	I2C2_SCL_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	I2C2_SCL_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad i2c2_scl 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0



Bits	Field Name	Description	Type	Reset
28	I2C2_SCL_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad i2c2_scl 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	I2C2_SCL_OFFMODEOUTVALUE	OffMode mode output value for pad i2c2_scl 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	I2C2_SCL_OFFMODEOUTENABLE	OffMode mode output enable value for pad i2c2_scl. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	I2C2_SCL_OFFMODEENABLE	OffMode mode override control for pad i2c2_scl 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	I2C2_SCL_INPUTENABLE	Input enable value for pad i2c2_scl 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	I2C2_SCL_PULLTYPESELECT	pullup/down selection for pad i2c2_scl 0x0: pulldown selected 0x1: pullup selected	RW	1
19	I2C2_SCL_PULLUDENABLE	pullup/down enable for pad i2c2_scl 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	I2C2_SCL_MUXMODE	Functional multiplexing selection for pad i2c2_scl 0x0: Select i2c2_scl 0x1: Select uart1_rx 0x3: Select gpio_128 0x7: Select safe_mode	RW	0x7
15	I2C1_SDA_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	I2C1_SDA_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	I2C1_SDA_INPUTENABLE	Input enable value for pad i2c1_sda 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	I2C1_SDA_PULLTYPESELECT	pullup/down selection for pad i2c1_sda 0x0: pulldown selected 0x1: pullup selected	RW	1
3	I2C1_SDA_PULLUDENABLE	pullup/down enable for pad i2c1_sda 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	RESERVED		R	0x0

**Table 18-437. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_I2C1\_SDA\_PAD1\_I2C2\_SCL**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-438. CONTROL\_CORE\_PAD0\_I2C2\_SDA\_PAD1\_I2C3\_SCL**

<b>Address Offset</b>	0x0000 0128	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads i2c2_sda and i2c3_scl Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C3_SCL_WAKEUPEVENT	I2C3_SCL_WAKEUPENABLE	I2C3_SCL_OFFMODEPULLTYPESELECT	I2C3_SCL_OFFMODEPULLUDENENABLE	I2C3_SCL_OFFMODEOUTVALUE	I2C3_SCL_OFFMODEOUTENABLE	I2C3_SCL_OFFMODEENABLE	I2C3_SCL_INPUENABLE	RESERVED			I2C3_SCL_PULLTYPESELECT	I2C3_SCL_PULLUDENENABLE	I2C3_SCL_MUXMODE			I2C2_SDA_WAKEUPEVENT	I2C2_SDA_WAKEUPENABLE	I2C2_SDA_OFFMODEPULLTYPESELECT	I2C2_SDA_OFFMODEPULLUDENENABLE	I2C2_SDA_OFFMODEOUTVALUE	I2C2_SDA_OFFMODEOUTENABLE	I2C2_SDA_OFFMODEENABLE	I2C2_SDA_INPUENABLE	RESERVED			I2C2_SDA_PULLTYPESELECT	I2C2_SDA_PULLUDENENABLE	I2C2_SDA_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	I2C3_SCL_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	I2C3_SCL_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	I2C3_SCL_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad i2c3_scl 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	I2C3_SCL_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad i2c3_scl 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	I2C3_SCL_OFFMODEOUTVALUE	OffMode mode output value for pad i2c3_scl 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	I2C3_SCL_OFFMODEOUTENABLE	OffMode mode output enable value for pad i2c3_scl. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
25	I2C3_SCL_OFFMODEENABLE	OffMode mode override control for pad i2c3_scl 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	I2C3_SCL_INPUTENABLE	Input enable value for pad i2c3_scl 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	I2C3_SCL_PULLTYPESELECT	pullup/down selection for pad i2c3_scl 0x0: pulldown selected 0x1: pullup selected	RW	1
19	I2C3_SCL_PULLUDENABLE	pullup/down enable for pad i2c3_scl 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	I2C3_SCL_MUXMODE	Functional multiplexing selection for pad i2c3_scl 0x0: Select i2c3_scl 0x3: Select gpio_130 0x7: Select safe_mode	RW	0x7
15	I2C2_SDA_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	I2C2_SDA_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	I2C2_SDA_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad i2c2_sda 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	I2C2_SDA_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad i2c2_sda 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	I2C2_SDA_OFFMODEOUTVALUE	OffMode mode output value for pad i2c2_sda 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	I2C2_SDA_OFFMODEOUTENABLE	OffMode mode output enable value for pad i2c2_sda. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	I2C2_SDA_OFFMODEENABLE	OffMode mode override control for pad i2c2_sda 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	I2C2_SDA_INPUTENABLE	Input enable value for pad i2c2_sda 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	I2C2_SDA_PULLTYPESELECT	pullup/down selection for pad i2c2_sda 0x0: pulldown selected 0x1: pullup selected	RW	1

Bits	Field Name	Description	Type	Reset
3	I2C2_SDA_PULLUDENABLE	pullup/down enable for pad i2c2_sda 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	I2C2_SDA_MUXMODE	Functional multiplexing selection for pad i2c2_sda 0x0: Select i2c2_sda 0x1: Select uart1_tx 0x3: Select gpio_129 0x7: Select safe_mode	RW	0x7

**Table 18-439. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_I2C2\_SDA\_PAD1\_I2C3\_SCL**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-440. CONTROL\_CORE\_PAD0\_I2C3\_SDA\_PAD1\_I2C4\_SCL**

<b>Address Offset</b>	0x0000 012C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads i2c3_sda and i2c4_scl Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C4_SCL_WAKEUPEVENT	I2C4_SCL_WAKEUPENABLE	I2C4_SCL_OFFMODEPULLTYPESELECT	I2C4_SCL_OFFMODEPULLUDENABLE	I2C4_SCL_OFFMODEOUTVALUE	I2C4_SCL_OFFMODEOUTENABLE	I2C4_SCL_OFFMODEENABLE	I2C4_SCL_INPUTENABLE	RESERVED			I2C4_SCL_PULLTYPESELECT	I2C4_SCL_PULLUDENABLE	I2C4_SCL_MUXMODE			I2C3_SDA_WAKEUPEVENT	I2C3_SDA_WAKEUPENABLE	I2C3_SDA_OFFMODEPULLTYPESELECT	I2C3_SDA_OFFMODEPULLUDENABLE	I2C3_SDA_OFFMODEOUTVALUE	I2C3_SDA_OFFMODEOUTENABLE	I2C3_SDA_OFFMODEENABLE	I2C3_SDA_INPUTENABLE	RESERVED			I2C3_SDA_PULLTYPESELECT	I2C3_SDA_PULLUDENABLE	I2C3_SDA_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	I2C4_SCL_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	I2C4_SCL_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	I2C4_SCL_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad i2c4_scl 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
28	I2C4_SCL_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad i2c4_scl 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	I2C4_SCL_OFFMODEOUTVALUE	OffMode mode output value for pad i2c4_scl 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	I2C4_SCL_OFFMODEOUTENABLE	OffMode mode output enable value for pad i2c4_scl. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	I2C4_SCL_OFFMODEENABLE	OffMode mode override control for pad i2c4_scl 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	I2C4_SCL_INPUTENABLE	Input enable value for pad i2c4_scl 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	I2C4_SCL_PULLTYPESELECT	pullup/down selection for pad i2c4_scl 0x0: pulldown selected 0x1: pullup selected	RW	1
19	I2C4_SCL_PULLUDENABLE	pullup/down enable for pad i2c4_scl 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	I2C4_SCL_MUXMODE	Functional multiplexing selection for pad i2c4_scl 0x0: Select i2c4_scl 0x3: Select gpio_132 0x7: Select safe_mode	RW	0x7
15	I2C3_SDA_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	I2C3_SDA_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	I2C3_SDA_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad i2c3_sda 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	I2C3_SDA_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad i2c3_sda 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	I2C3_SDA_OFFMODEOUTVALUE	OffMode mode output value for pad i2c3_sda 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	I2C3_SDA_OFFMODEOUTENABLE	OffMode mode output enable value for pad i2c3_sda. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
9	I2C3_SDA_OFFMODEENABLE	OffMode mode override control for pad i2c3_sda 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	I2C3_SDA_INPUTENABLE	Input enable value for pad i2c3_sda 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	I2C3_SDA_PULLTYPESELECT	pullup/down selection for pad i2c3_sda 0x0: pulldown selected 0x1: pullup selected	RW	1
3	I2C3_SDA_PULLUDENABLE	pullup/down enable for pad i2c3_sda 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	I2C3_SDA_MUXMODE	Functional multiplexing selection for pad i2c3_sda 0x0: Select i2c3_sda 0x3: Select gpio_131 0x7: Select safe_mode	RW	0x7

**Table 18-441. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_I2C3\_SDA\_PAD1\_I2C4\_SCL**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-442. CONTROL\_CORE\_PAD0\_I2C4\_SDA\_PAD1\_MCSP1\_CLK**

<b>Address Offset</b>	0x0000 0130	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads i2c4_sda and mcspi1_clk Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSP11_CLK_WAKEUPEVENT	MCSP11_CLK_WAKEUPENABLE	MCSP11_CLK_OFFMODEPULLTYPESELECT	MCSP11_CLK_OFFMODEPULLUDENABLE	MCSP11_CLK_OFFMODEOUTVALUE	MCSP11_CLK_OFFMODEOUTENABLE	MCSP11_CLK_OFFMODEENABLE	MCSP11_CLK_INPUTENABLE	RESERVED			MCSP11_CLK_PULLTYPESELECT	MCSP11_CLK_PULLUDENABLE			MCSP11_CLK_MUXMODE																
																I2C4_SDA_WAKEUPEVENT	I2C4_SDA_WAKEUPENABLE	I2C4_SDA_OFFMODEPULLTYPESELECT	I2C4_SDA_OFFMODEPULLUDENABLE	I2C4_SDA_OFFMODEOUTVALUE	I2C4_SDA_OFFMODEOUTENABLE	I2C4_SDA_OFFMODEENABLE	I2C4_SDA_INPUTENABLE	RESERVED			I2C4_SDA_PULLTYPESELECT	I2C4_SDA_PULLUDENABLE			I2C4_SDA_MUXMODE

Bits	Field Name	Description	Type	Reset
31	MCSP11_CLK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	MCSP11_CLK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	MCSP11_CLK_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi1_clk 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	MCSP11_CLK_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad mcspi1_clk 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	MCSP11_CLK_OFFMODEOUTPUTVALUE	OffMode mode output value for pad mcspi1_clk 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	MCSP11_CLK_OFFMODEOUTPUTENABLE	OffMode mode output enable value for pad mcspi1_clk. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	MCSP11_CLK_OFFMODEENABLE	OffMode mode override control for pad mcspi1_clk 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	MCSP11_CLK_INPUTENABLE	Input enable value for pad mcspi1_clk 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	MCSP11_CLK_PULLTYPESELECT	pullup/down selection for pad mcspi1_clk 0x0: pulldown selected 0x1: pullup selected	RW	0
19	MCSP11_CLK_PULLUDENABLE	pullup/down enable for pad mcspi1_clk 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	MCSP11_CLK_MUXMODE	Functional multiplexing selection for pad mcspi1_clk 0x0: Select mcspi1_clk 0x3: Select gpio_134 0x7: Select safe_mode	RW	0x7
15	I2C4_SDA_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	I2C4_SDA_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	I2C4_SDA_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad i2c4_sda 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	I2C4_SDA_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad i2c4_sda 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	I2C4_SDA_OFFMODEOUTVALUE	OffMode mode output value for pad i2c4_sda 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	I2C4_SDA_OFFMODEOUTENABLE	OffMode mode output enable value for pad i2c4_sda. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	I2C4_SDA_OFFMODEENABLE	OffMode mode override control for pad i2c4_sda 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	I2C4_SDA_INPUTENABLE	Input enable value for pad i2c4_sda 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	I2C4_SDA_PULLTYPESELECT	pullup/down selection for pad i2c4_sda 0x0: pulldown selected 0x1: pullup selected	RW	1
3	I2C4_SDA_PULLUDENABLE	pullup/down enable for pad i2c4_sda 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	I2C4_SDA_MUXMODE	Functional multiplexing selection for pad i2c4_sda 0x0: Select i2c4_sda 0x3: Select gpio_133 0x7: Select safe_mode	RW	0x7

**Table 18-443. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_I2C4\_SDA\_PAD1\_MCSP1\_CLK**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-444. CONTROL\_CORE\_PAD0\_MCSP1\_SOMI\_PAD1\_MCSP1\_SIMO**

<b>Address Offset</b>	0x0000 0134	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads mcspi1_somi and mcspi1_simo Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSP11_SIMO_WAKEUPEVENT	MCSP11_SIMO_WAKEUPENABLE	MCSP11_SIMO_OFFMODEPULLTYPESELECT	MCSP11_SIMO_OFFMODEPULLUDENENABLE	MCSP11_SIMO_OFFMODEOUTVALUE	MCSP11_SIMO_OFFMODEOUTENABLE	MCSP11_SIMO_OFFMODEEENABLE	MCSP11_SIMO_INPUTENABLE	RESERVED			MCSP11_SIMO_PULLTYPESELECT	MCSP11_SIMO_PULLUDENENABLE		MCSP11_SIMO_MUXMODE		MCSP11_SOMI_WAKEUPEVENT	MCSP11_SOMI_WAKEUPENABLE	MCSP11_SOMI_OFFMODEPULLTYPESELECT	MCSP11_SOMI_OFFMODEPULLUDENENABLE	MCSP11_SOMI_OFFMODEOUTVALUE	MCSP11_SOMI_OFFMODEOUTENABLE	MCSP11_SOMI_OFFMODEEENABLE	MCSP11_SOMI_INPUTENABLE	RESERVED			MCSP11_SOMI_PULLTYPESELECT	MCSP11_SOMI_PULLUDENENABLE		MCSP11_SOMI_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	MCSP11_SIMO_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	MCSP11_SIMO_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	MCSP11_SIMO_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi1_simo 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	MCSP11_SIMO_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad mcspi1_simo 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	MCSP11_SIMO_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi1_simo 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	MCSP11_SIMO_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi1_simo. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	MCSP11_SIMO_OFFMODEEENABLE	OffMode mode override control for pad mcspi1_simo 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	MCSP11_SIMO_INPUTENABLE	Input enable value for pad mcspi1_simo 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	MCSP11_SIMO_PULLTYPESELECT	pullup/down selection for pad mcspi1_simo 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
19	MCSP11_SIMO_PULLUDENABLE	pullup/down enable for pad mcspi1_simo 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	MCSP11_SIMO_MUXMODE	Functional multiplexing selection for pad mcspi1_simo 0x0: Select mcspi1_simo 0x3: Select gpio_136 0x7: Select safe_mode	RW	0x7
15	MCSP11_SOMI_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	MCSP11_SOMI_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	MCSP11_SOMI_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi1_somi 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	MCSP11_SOMI_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad mcspi1_somi 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	MCSP11_SOMI_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi1_somi 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	MCSP11_SOMI_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi1_somi. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	MCSP11_SOMI_OFFMODEENABLE	OffMode mode override control for pad mcspi1_somi 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	MCSP11_SOMI_INPUTENABLE	Input enable value for pad mcspi1_somi 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	MCSP11_SOMI_PULLTYPESELECT	pullup/down selection for pad mcspi1_somi 0x0: pulldown selected 0x1: pullup selected	RW	0
3	MCSP11_SOMI_PULLUDENABLE	pullup/down enable for pad mcspi1_somi 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	MCSP11_SOMI_MUXMODE	Functional multiplexing selection for pad mcspi1_somi 0x0: Select mcspi1_somi 0x3: Select gpio_135 0x7: Select safe_mode	RW	0x7

**Table 18-445. Register Call Summary for Register CONTROL\_CORE\_PAD0\_MCSP11\_SOMI\_PAD1\_MCSP11\_SIMO**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-446. CONTROL\_CORE\_PAD0\_MCSP11\_CS0\_PAD1\_MCSP11\_CS1**

<b>Address Offset</b>	0x0000 0138	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads mcspi1_cs0 and mcspi1_cs1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSP11_CS1_WAKEUPEVENT	MCSP11_CS1_WAKEUPENABLE	MCSP11_CS1_OFFMODEPULLTYPESELECT	MCSP11_CS1_OFFMODEPULLUDENENABLE	MCSP11_CS1_OFFMODEOUTVALUE	MCSP11_CS1_OFFMODEOUTENABLE	MCSP11_CS1_OFFMODEENABLE	MCSP11_CS1_INPUTENABLE	RESERVED			MCSP11_CS1_PULLTYPESELECT	MCSP11_CS1_PULLUDENENABLE			MCSP11_CS1_MUXMODE	MCSP11_CS0_WAKEUPEVENT	MCSP11_CS0_WAKEUPENABLE	MCSP11_CS0_OFFMODEPULLTYPESELECT	MCSP11_CS0_OFFMODEPULLUDENENABLE	MCSP11_CS0_OFFMODEOUTVALUE	MCSP11_CS0_OFFMODEOUTENABLE	MCSP11_CS0_OFFMODEENABLE	MCSP11_CS0_INPUTENABLE	RESERVED			MCSP11_CS0_PULLTYPESELECT	MCSP11_CS0_PULLUDENENABLE			MCSP11_CS0_MUXMODE

Bits	Field Name	Description	Type	Reset
31	MCSP11_CS1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	MCSP11_CS1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	MCSP11_CS1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi1_cs1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	MCSP11_CS1_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad mcspi1_cs1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	MCSP11_CS1_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi1_cs1 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	MCSP11_CS1_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi1_cs1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
25	MCSP11_CS1_OFFMODEENABLE	OffMode mode override control for pad mcspi1_cs1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	MCSP11_CS1_INPUTENABLE	Input enable value for pad mcspi1_cs1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	MCSP11_CS1_PULLTYPESELECT	pullup/down selection for pad mcspi1_cs1 0x0: pulldown selected 0x1: pullup selected	RW	0
19	MCSP11_CS1_PULLUDENABLE	pullup/down enable for pad mcspi1_cs1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	MCSP11_CS1_MUXMODE	Functional multiplexing selection for pad mcspi1_cs1 0x0: Select mcspi1_cs1 0x1: Select uart1_rx 0x3: Select gpio_138 0x7: Select safe_mode	RW	0x7
15	MCSP11_CS0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	MCSP11_CS0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	MCSP11_CS0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi1_cs0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	MCSP11_CS0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad mcspi1_cs0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	MCSP11_CS0_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi1_cs0 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	MCSP11_CS0_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi1_cs0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	MCSP11_CS0_OFFMODEENABLE	OffMode mode override control for pad mcspi1_cs0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	MCSP11_CS0_INPUTENABLE	Input enable value for pad mcspi1_cs0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	MCSP11_CS0_PULLTYPESELECT	pullup/down selection for pad mcspi1_cs0 0x0: pulldown selected 0x1: pullup selected	RW	0
3	MCSP11_CS0_PULLUDENABLE	pullup/down enable for pad mcspi1_cs0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	MCSP11_CS0_MUXMODE	Functional multiplexing selection for pad mcspi1_cs0 0x0: Select mcspi1_cs0 0x3: Select gpio_137 0x7: Select safe_mode	RW	0x7

**Table 18-447. Register Call Summary for Register CONTROL\_CORE\_PAD0\_MCSP11\_CS0\_PAD1\_MCSP11\_CS1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-448. CONTROL\_CORE\_PAD0\_MCSP11\_CS2\_PAD1\_MCSP11\_CS3**

<b>Address Offset</b>	0x0000 013C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads mcspi1_cs2 and mcspi1_cs3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSP11_CS3_WAKEUPEVENT	MCSP11_CS3_WAKEUPENABLE	MCSP11_CS3_OFFMODEPULLTYPESELECT	MCSP11_CS3_OFFMODEPULLUDENABLE	MCSP11_CS3_OFFMODEOUTVALUE	MCSP11_CS3_OFFMODEOUTENABLE	MCSP11_CS3_OFFMODEENABLE	MCSP11_CS3_INPUTENABLE	RESERVED	MCSP11_CS3_PULLTYPESELECT	MCSP11_CS3_PULLUDENABLE	MCSP11_CS3_MUXMODE	MCSP11_CS2_WAKEUPEVENT	MCSP11_CS2_WAKEUPENABLE	MCSP11_CS2_OFFMODEPULLTYPESELECT	MCSP11_CS2_OFFMODEPULLUDENABLE	MCSP11_CS2_OFFMODEOUTVALUE	MCSP11_CS2_OFFMODEOUTENABLE	MCSP11_CS2_OFFMODEENABLE	MCSP11_CS2_INPUTENABLE	RESERVED	MCSP11_CS2_PULLTYPESELECT	MCSP11_CS2_PULLUDENABLE	MCSP11_CS2_MUXMODE								

Bits	Field Name	Description	Type	Reset
31	MCSP11_CS3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	MCSP11_CS3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
29	MCSP11_CS3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi1_cs3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	MCSP11_CS3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad mcspi1_cs3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	MCSP11_CS3_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi1_cs3 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	MCSP11_CS3_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi1_cs3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	MCSP11_CS3_OFFMODEENABLE	OffMode mode override control for pad mcspi1_cs3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	MCSP11_CS3_INPUTENABLE	Input enable value for pad mcspi1_cs3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	MCSP11_CS3_PULLTYPESELECT	pullup/down selection for pad mcspi1_cs3 0x0: pulldown selected 0x1: pullup selected	RW	1
19	MCSP11_CS3_PULLUDENABLE	pullup/down enable for pad mcspi1_cs3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	MCSP11_CS3_MUXMODE	Functional multiplexing selection for pad mcspi1_cs3 0x0: Select mcspi1_cs3 0x1: Select uart1_rts 0x2: Select slimbus2_data 0x3: Select gpio_140 0x7: Select safe_mode	RW	0x7
15	MCSP11_CS2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	MCSP11_CS2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	MCSP11_CS2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi1_cs2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	MCSP11_CS2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad mcspi1_cs2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	MCSP11_CS2_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi1_cs2 0x0: Set value at 0 0x1: Set value at 1	RW	0

Bits	Field Name	Description	Type	Reset
10	MCSP11_CS2_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi1_cs2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	MCSP11_CS2_OFFMODEENABLE	OffMode mode override control for pad mcspi1_cs2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	MCSP11_CS2_INPUTENABLE	Input enable value for pad mcspi1_cs2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	MCSP11_CS2_PULLTYPESELECT	pullup/down selection for pad mcspi1_cs2 0x0: pulldown selected 0x1: pullup selected	RW	1
3	MCSP11_CS2_PULLUDENABLE	pullup/down enable for pad mcspi1_cs2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	MCSP11_CS2_MUXMODE	Functional multiplexing selection for pad mcspi1_cs2 0x0: Select mcspi1_cs2 0x1: Select uart1_cts 0x2: Select slimbus2_clock 0x3: Select gpio_139 0x7: Select safe_mode	RW	0x7

**Table 18-449. Register Call Summary for Register CONTROL\_CORE\_PAD0\_MCSP11\_CS2\_PAD1\_MCSP11\_CS3**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-450. CONTROL\_CORE\_PAD0\_UART3\_CTS\_RCTX\_PAD1\_UART3\_RTS\_SD**

<b>Address Offset</b>	0x0000 0140	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads uart3_cts_rctx and uart3_rts_sd Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
UART3_RTS_SD_WAKEUPEVENT	UART3_RTS_SD_WAKEUPENABLE	UART3_RTS_SD_OFFMODEPULLTYPESELECT	UART3_RTS_SD_OFFMODEPULLUDENENABLE	UART3_RTS_SD_OFFMODEOUTVALUE	UART3_RTS_SD_OFFMODEOUTENABLE	UART3_RTS_SD_OFFMODEEENABLE	UART3_RTS_SD_INPUTENABLE	RESERVED			UART3_RTS_SD_PULLTYPESELECT	UART3_RTS_SD_PULLUDENENABLE			UART3_RTS_SD_MUXMODE		UART3_CTS_RCTX_WAKEUPEVENT	UART3_CTS_RCTX_WAKEUPENABLE	UART3_CTS_RCTX_OFFMODEPULLTYPESELECT	UART3_CTS_RCTX_OFFMODEPULLUDENENABLE	UART3_CTS_RCTX_OFFMODEOUTVALUE	UART3_CTS_RCTX_OFFMODEOUTENABLE	UART3_CTS_RCTX_OFFMODEEENABLE	UART3_CTS_RCTX_INPUTENABLE	RESERVED			UART3_CTS_RCTX_PULLTYPESELECT	UART3_CTS_RCTX_PULLUDENENABLE			UART3_CTS_RCTX_MUXMODE

Bits	Field Name	Description	Type	Reset
31	UART3_RTS_SD_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	UART3_RTS_SD_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	UART3_RTS_SD_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad uart3_rts_sd 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	UART3_RTS_SD_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad uart3_rts_sd 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	UART3_RTS_SD_OFFMODEOUTVALUE	OffMode mode output value for pad uart3_rts_sd 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	UART3_RTS_SD_OFFMODEOUTENABLE	OffMode mode output enable value for pad uart3_rts_sd. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	UART3_RTS_SD_OFFMODEEENABLE	OffMode mode override control for pad uart3_rts_sd 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	UART3_RTS_SD_INPUTENABLE	Input enable value for pad uart3_rts_sd 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	UART3_RTS_SD_PULLTYPESELECT	pullup/down selection for pad uart3_rts_sd 0x0: pulldown selected 0x1: pullup selected	RW	1



Bits	Field Name	Description	Type	Reset
19	UART3_RTS_SD_PULLUDENABLE	pullup/down enable for pad uart3_rts_sd 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	UART3_RTS_SD_MUXMODE	Functional multiplexing selection for pad uart3_rts_sd 0x0: Select uart3_rts_sd 0x3: Select gpio_142 0x7: Select safe_mode	RW	0x7
15	UART3_CTS_RCTX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	UART3_CTS_RCTX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	UART3_CTS_RCTX_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad uart3_cts_rctx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	UART3_CTS_RCTX_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad uart3_cts_rctx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	UART3_CTS_RCTX_OFFMODEOUTVALUE	OffMode mode output value for pad uart3_cts_rctx 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	UART3_CTS_RCTX_OFFMODEOUTENABLE	OffMode mode output enable value for pad uart3_cts_rctx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	UART3_CTS_RCTX_OFFMODEENABLE	OffMode mode override control for pad uart3_cts_rctx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	UART3_CTS_RCTX_INPUTENABLE	Input enable value for pad uart3_cts_rctx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	UART3_CTS_RCTX_PULLTYPESELECT	pullup/down selection for pad uart3_cts_rctx 0x0: pulldown selected 0x1: pullup selected	RW	1
3	UART3_CTS_RCTX_PULLUDENABLE	pullup/down enable for pad uart3_cts_rctx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	UART3_CTS_RCTX_MUXMODE	Functional multiplexing selection for pad uart3_cts_rctx 0x0: Select uart3_cts_rctx 0x1: Select uart1_tx 0x3: Select gpio_141 0x7: Select safe_mode	RW	0x7

**Table 18-451. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_UART3\_CTS\_RCTX\_PAD1\_UART3\_RTS\_SD**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-452. CONTROL\_CORE\_PAD0\_UART3\_RX\_IRRX\_PAD1\_UART3\_TX\_IRTX**

<b>Address Offset</b>	0x0000 0144	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads uart3_rx_irrx and uart3_tx_irtx Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART3_TX_IRTX_WAKEUPEVENT	UART3_TX_IRTX_WAKEUPENABLE	UART3_TX_IRTX_OFFMODEPULLTYPESELECT	UART3_TX_IRTX_OFFMODEPULLUDENENABLE	UART3_TX_IRTX_OFFMODEOUTVALUE	UART3_TX_IRTX_OFFMODEOUTENABLE	UART3_TX_IRTX_OFFMODEENABLE	UART3_TX_IRTX_INPUTENABLE	RESERVED			UART3_TX_IRTX_PULLTYPESELECT	UART3_TX_IRTX_PULLUDENENABLE		UART3_TX_IRTX_MUXMODE		UART3_RX_IRRX_WAKEUPEVENT	UART3_RX_IRRX_WAKEUPENABLE	UART3_RX_IRRX_OFFMODEPULLTYPESELECT	UART3_RX_IRRX_OFFMODEPULLUDENENABLE	UART3_RX_IRRX_OFFMODEOUTVALUE	UART3_RX_IRRX_OFFMODEOUTENABLE	UART3_RX_IRRX_OFFMODEENABLE	UART3_RX_IRRX_INPUTENABLE	RESERVED			UART3_RX_IRRX_PULLTYPESELECT	UART3_RX_IRRX_PULLUDENENABLE		UART3_RX_IRRX_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	UART3_TX_IRTX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	UART3_TX_IRTX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	UART3_TX_IRTX_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad uart3_tx_irtx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	UART3_TX_IRTX_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad uart3_tx_irtx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	UART3_TX_IRTX_OFFMODEOUTVALUE	OffMode mode output value for pad uart3_tx_irtx 0x0: Set value at 0 0x1: Set value at 1	RW	0

Bits	Field Name	Description	Type	Reset
26	UART3_TX_IRTX_OFFMODEO UTENABLE	OffMode mode output enable value for pad uart3_tx_irtx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	UART3_TX_IRTX_OFFMODEEEN ABLE	OffMode mode override control for pad uart3_tx_irtx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	UART3_TX_IRTX_INPUTENABL E	Input enable value for pad uart3_tx_irtx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	UART3_TX_IRTX_PULLTYPESE LECT	pullup/down selection for pad uart3_tx_irtx 0x0: pulldown selected 0x1: pullup selected	RW	1
19	UART3_TX_IRTX_PULLUDENA BLE	pullup/down enable for pad uart3_tx_irtx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	UART3_TX_IRTX_MUXMODE	Functional multiplexing selection for pad uart3_tx_irtx 0x0: Select uart3_tx_irtx 0x1: Select dmtimer9_pwm_evt 0x3: Select gpio_144 0x7: Select safe_mode	RW	0x7
15	UART3_RX_ IRRX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	UART3_RX_ IRRX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	UART3_RX_ IRRX_OFFMODEPULLTYPESEL ECT	OffMode mode pullup/down selection for pad uart3_rx_irrx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	UART3_RX_ IRRX_OFFMODEPULLUDENAB LE	OffMode mode pullup/down enable for pad uart3_rx_irrx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	UART3_RX_ IRRX_OFFMODEOUTVALUE	OffMode mode output value for pad uart3_rx_irrx 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	UART3_RX_ IRRX_OFFMODEOUTENABLE	OffMode mode output enable value for pad uart3_rx_irrx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	UART3_RX_IRRX_OFFMODEE NABLE	OffMode mode override control for pad uart3_rx_irrx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
8	UART3_RX_IRRX_INPUTENABLE	Input enable value for pad uart3_rx_irrx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	UART3_RX_IRRX_PULLTYPESELECT	pullup/down selection for pad uart3_rx_irrx 0x0: pulldown selected 0x1: pullup selected	RW	1
3	UART3_RX_IRRX_PULLUDENABLE	pullup/down enable for pad uart3_rx_irrx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	UART3_RX_IRRX_MUXMODE	Functional multiplexing selection for pad uart3_rx_irrx 0x0: Select uart3_rx_irrx 0x1: Select dmtimer8_pwm_evt 0x3: Select gpio_143 0x7: Select safe_mode	RW	0x7

**Table 18-453. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_UART3\_RX\_IRRX\_PAD1\_UART3\_TX\_IRTX**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-454. CONTROL\_CORE\_PAD0\_SDMMC5\_CLK\_PAD1\_SDMMC5\_CMD**

<b>Address Offset</b>	0x0000 0148	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads sdmmc5_clk and sdmmc5_cmd Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDMMC5_CMD_WAKEUPEVENT	SDMMC5_CMD_WAKEUPENABLE	SDMMC5_CMD_OFFMODEPULLTYPESELECT	SDMMC5_CMD_OFFMODEPULLUDENABLE	SDMMC5_CMD_OFFMODEOUTVALUE	SDMMC5_CMD_OFFMODEOUTENABLE	SDMMC5_CMD_OFFMODEENABLE	SDMMC5_CMD_INPUTENABLE	RESERVED	SDMMC5_CMD_PULLTYPESELECT	SDMMC5_CMD_PULLUDENABLE	SDMMC5_CMD_MUXMODE	SDMMC5_CLK_WAKEUPEVENT	SDMMC5_CLK_WAKEUPENABLE	SDMMC5_CLK_OFFMODEPULLTYPESELECT	SDMMC5_CLK_OFFMODEPULLUDENABLE	SDMMC5_CLK_OFFMODEOUTVALUE	SDMMC5_CLK_OFFMODEOUTENABLE	SDMMC5_CLK_OFFMODEENABLE	SDMMC5_CLK_INPUTENABLE	RESERVED	SDMMC5_CLK_PULLTYPESELECT	SDMMC5_CLK_PULLUDENABLE	SDMMC5_CLK_MUXMODE									

Bits	Field Name	Description	Type	Reset
31	SDMMC5_CMD_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SDMMC5_CMD_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SDMMC5_CMD_OFFMODEPUL LTYPESELECT	OffMode mode pullup/down selection for pad sdmmc5_cmd 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	SDMMC5_CMD_OFFMODEPUL LUDENABLE	OffMode mode pullup/down enable for pad sdmmc5_cmd 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SDMMC5_CMD_OFFMODEOUT VALUE	OffMode mode output value for pad sdmmc5_cmd 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SDMMC5_CMD_OFFMODEOUT ENABLE	OffMode mode output enable value for pad sdmmc5_cmd. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SDMMC5_CMD_OFFMODEENA BLE	OffMode mode override control for pad sdmmc5_cmd 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	SDMMC5_CMD_INPUTENABLE	Input enable value for pad sdmmc5_cmd 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SDMMC5_CMD_PULLTYPESELECT	pullup/down selection for pad sdmmc5_cmd 0x0: pulldown selected 0x1: pullup selected	RW	1
19	SDMMC5_CMD_PULLUDENABLE	pullup/down enable for pad sdmmc5_cmd 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SDMMC5_CMD_MUXMODE	Functional multiplexing selection for pad sdmmc5_cmd 0x0: Select sdmmc5_cmd 0x1: Select mcspi2_simo 0x2: Select usbc1_ibus_dm 0x3: Select gpio_146 0x5: Select sdmmc2_cmd 0x7: Select safe_mode	RW	0x7
15	SDMMC5_CLK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SDMMC5_CLK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
13	SDMMC5_CLK_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc5_clk 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SDMMC5_CLK_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc5_clk 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SDMMC5_CLK_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc5_clk 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SDMMC5_CLK_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc5_clk. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	SDMMC5_CLK_OFFMODEENABLE	OffMode mode override control for pad sdmmc5_clk 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SDMMC5_CLK_INPUTENABLE	Input enable value for pad sdmmc5_clk 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SDMMC5_CLK_PULLTYPESELECT	pullup/down selection for pad sdmmc5_clk 0x0: pulldown selected 0x1: pullup selected	RW	0
3	SDMMC5_CLK_PULLUDENABLE	pullup/down enable for pad sdmmc5_clk 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SDMMC5_CLK_MUXMODE	Functional multiplexing selection for pad sdmmc5_clk 0x0: Select sdmmc5_clk 0x1: Select mcspi2_clk 0x2: Select usbc1_ibusb_dp 0x3: Select gpio_145 0x5: Select sdmmc2_clk 0x7: Select safe_mode	RW	0x7

**Table 18-455. Register Call Summary for Register CONTROL\_CORE\_PAD0\_SDMMC5\_CLK\_PAD1\_SDMMC5\_CMD**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-456. CONTROL\_CORE\_PAD0\_SDMMC5\_DAT0\_PAD1\_SDMMC5\_DAT1**

<b>Address Offset</b>	0x0000 014C		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads sdmmc5_dat0 and sdmmc5_dat1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDMMC5_DAT1_WAKEUPEVENT	SDMMC5_DAT1_WAKEUPENABLE	SDMMC5_DAT1_OFFMODEPULLTYPESELECT	SDMMC5_DAT1_OFFMODEPULLUDENABLE	SDMMC5_DAT1_OFFMODEOUTVALUE	SDMMC5_DAT1_OFFMODEOUTENABLE	SDMMC5_DAT1_OFFMODEENABLE	SDMMC5_DAT1_INPUTENABLE	RESERVED			SDMMC5_DAT1_PULLTYPESELECT	SDMMC5_DAT1_PULLUDENABLE			SDMMC5_DAT1_MUXMODE		SDMMC5_DAT0_WAKEUPEVENT	SDMMC5_DAT0_WAKEUPENABLE	SDMMC5_DAT0_OFFMODEPULLTYPESELECT	SDMMC5_DAT0_OFFMODEPULLUDENABLE	SDMMC5_DAT0_OFFMODEOUTVALUE	SDMMC5_DAT0_OFFMODEOUTENABLE	SDMMC5_DAT0_OFFMODEENABLE	SDMMC5_DAT0_INPUTENABLE	RESERVED			SDMMC5_DAT0_PULLTYPESELECT	SDMMC5_DAT0_PULLUDENABLE			SDMMC5_DAT0_MUXMODE

Bits	Field Name	Description	Type	Reset
31	SDMMC5_DAT1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SDMMC5_DAT1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SDMMC5_DAT1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc5_dat1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	SDMMC5_DAT1_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc5_dat1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SDMMC5_DAT1_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc5_dat1 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SDMMC5_DAT1_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc5_dat1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SDMMC5_DAT1_OFFMODEENABLE	OffMode mode override control for pad sdmmc5_dat1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
24	SDMMC5_DAT1_INPUTENABLE	Input enable value for pad sdmmc5_dat1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SDMMC5_DAT1_PULLTYPESELECT	pullup/down selection for pad sdmmc5_dat1 0x0: pulldown selected 0x1: pullup selected	RW	1
19	SDMMC5_DAT1_PULLUDENABLE	pullup/down enable for pad sdmmc5_dat1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SDMMC5_DAT1_MUXMODE	Functional multiplexing selection for pad sdmmc5_dat1 0x0: Select sdmmc5_dat1 0x2: Select usbc1_icusb_txen 0x3: Select gpio_148 0x5: Select sdmmc2_dat1 0x7: Select safe_mode	RW	0x7
15	SDMMC5_DAT0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SDMMC5_DAT0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	SDMMC5_DAT0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc5_dat0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SDMMC5_DAT0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc5_dat0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SDMMC5_DAT0_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc5_dat0 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SDMMC5_DAT0_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc5_dat0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	SDMMC5_DAT0_OFFMODEENABLE	OffMode mode override control for pad sdmmc5_dat0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SDMMC5_DAT0_INPUTENABLE	Input enable value for pad sdmmc5_dat0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SDMMC5_DAT0_PULLTYPESELECT	pullup/down selection for pad sdmmc5_dat0 0x0: pulldown selected 0x1: pullup selected	RW	1



Bits	Field Name	Description	Type	Reset
3	SDMMC5_DAT0_PULLUDENABLE	pullup/down enable for pad sdmmc5_dat0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SDMMC5_DAT0_MUXMODE	Functional multiplexing selection for pad sdmmc5_dat0 0x0: Select sdmmc5_dat0 0x1: Select mcspi2_somi 0x2: Select usbc1_icusb_rcv 0x3: Select gpio_147 0x5: sdmmc2_dat0 0x7: Select safe_mode	RW	0x7

**Table 18-457. Register Call Summary for Register CONTROL\_CORE\_PAD0\_SDMMC5\_DAT0\_PAD1\_SDMMC5\_DAT1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-458. CONTROL\_CORE\_PAD0\_SDMMC5\_DAT2\_PAD1\_SDMMC5\_DAT3**

<b>Address Offset</b>	0x0000 0150	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads sdmmc5_dat2 and sdmmc5_dat3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SDMMC5_DAT3_WAKEUPEVENT	SDMMC5_DAT3_WAKEUPENABLE	SDMMC5_DAT3_OFFMODEPULLTYPESELECT	SDMMC5_DAT3_OFFMODEPULLUDENABLE	SDMMC5_DAT3_OFFMODEOUTVALUE	SDMMC5_DAT3_OFFMODEOUTENABLE	SDMMC5_DAT3_OFFMODEENABLE	SDMMC5_DAT3_INPUTENABLE	RESERVED	SDMMC5_DAT3_PULLTYPESELECT	SDMMC5_DAT3_PULLUDENABLE	SDMMC5_DAT3_MUXMODE	SDMMC5_DAT2_WAKEUPEVENT	SDMMC5_DAT2_WAKEUPENABLE	SDMMC5_DAT2_OFFMODEPULLTYPESELECT	SDMMC5_DAT2_OFFMODEPULLUDENABLE	SDMMC5_DAT2_OFFMODEOUTVALUE	SDMMC5_DAT2_OFFMODEOUTENABLE	SDMMC5_DAT2_OFFMODEENABLE	SDMMC5_DAT2_INPUTENABLE	RESERVED	SDMMC5_DAT2_PULLTYPESELECT	SDMMC5_DAT2_PULLUDENABLE	SDMMC5_DAT2_MUXMODE									

Bits	Field Name	Description	Type	Reset
31	SDMMC5_DAT3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SDMMC5_DAT3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
29	SDMMC5_DAT3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc5_dat3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	SDMMC5_DAT3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc5_dat3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SDMMC5_DAT3_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc5_dat3 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SDMMC5_DAT3_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc5_dat3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SDMMC5_DAT3_OFFMODEENABLE	OffMode mode override control for pad sdmmc5_dat3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	SDMMC5_DAT3_INPUTENABLE	Input enable value for pad sdmmc5_dat3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SDMMC5_DAT3_PULLTYPESELECT	pullup/down selection for pad sdmmc5_dat3 0x0: pulldown selected 0x1: pullup selected	RW	1
19	SDMMC5_DAT3_PULLUDENABLE	pullup/down enable for pad sdmmc5_dat3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SDMMC5_DAT3_MUXMODE	Functional multiplexing selection for pad sdmmc5_dat3 0x0: Select sdmmc5_dat3 0x1: Select mcspi2_cs0 0x3: Select gpio_150 0x5: Select sdmmc2_dat3 0x7: Select safe_mode	RW	0x7
15	SDMMC5_DAT2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SDMMC5_DAT2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	SDMMC5_DAT2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sdmmc5_dat2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SDMMC5_DAT2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sdmmc5_dat2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0

Bits	Field Name	Description	Type	Reset
11	SDMMC5_DAT2_OFFMODEOUTVALUE	OffMode mode output value for pad sdmmc5_dat2 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SDMMC5_DAT2_OFFMODEOUTENABLE	OffMode mode output enable value for pad sdmmc5_dat2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	SDMMC5_DAT2_OFFMODEENABLE	OffMode mode override control for pad sdmmc5_dat2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SDMMC5_DAT2_INPUTENABLE	Input enable value for pad sdmmc5_dat2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SDMMC5_DAT2_PULLTYPESELECT	pullup/down selection for pad sdmmc5_dat2 0x0: pulldown selected 0x1: pullup selected	RW	1
3	SDMMC5_DAT2_PULLUDENABLE	pullup/down enable for pad sdmmc5_dat2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SDMMC5_DAT2_MUXMODE	Functional multiplexing selection for pad sdmmc5_dat2 0x0: Select sdmmc5_dat2 0x1: Select mcspi2_cs1 0x3: Select gpio_149 0x5: Select sdmmc2_dat2 0x7: Select safe_mode	RW	0x7

**Table 18-459. Register Call Summary for Register CONTROL\_CORE\_PAD0\_SDMMC5\_DAT2\_PAD1\_SDMMC5\_DAT3**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-460. CONTROL\_CORE\_PAD0\_MCSPi4\_CLK\_PAD1\_MCSPi4\_SIMO**

<b>Address Offset</b>	0x0000 0154	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads mcspi4_clk and mcspi4_simo Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSP14_SIMO_WAKEUPEVENT	MCSP14_SIMO_WAKEUPENABLE	MCSP14_SIMO_OFFMODEPULLTYPESELECT	MCSP14_SIMO_OFFMODEPULLUDENABE	MCSP14_SIMO_OFFMODEOUTVALUE	MCSP14_SIMO_OFFMODEOUTENABLE	MCSP14_SIMO_OFFMODEEENABLE	MCSP14_SIMO_INPUTENABLE	RESERVED			MCSP14_SIMO_PULLTYPESELECT	MCSP14_SIMO_PULLUDENABE		MCSP14_SIMO_MUXMODE		MCSP14_CLK_WAKEUPEVENT	MCSP14_CLK_WAKEUPENABLE	MCSP14_CLK_OFFMODEPULLTYPESELECT	MCSP14_CLK_OFFMODEPULLUDENABE	MCSP14_CLK_OFFMODEOUTVALUE	MCSP14_CLK_OFFMODEOUTENABLE	MCSP14_CLK_OFFMODEEENABLE	MCSP14_CLK_INPUTENABLE	RESERVED			MCSP14_CLK_PULLTYPESELECT	MCSP14_CLK_PULLUDENABE		MCSP14_CLK_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	MCSP14_SIMO_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	MCSP14_SIMO_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	MCSP14_SIMO_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi4_simo 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	MCSP14_SIMO_OFFMODEPULLUDENABE	OffMode mode pullup/down enable for pad mcspi4_simo 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	MCSP14_SIMO_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi4_simo 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	MCSP14_SIMO_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi4_simo. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	MCSP14_SIMO_OFFMODEEENABLE	OffMode mode override control for pad mcspi4_simo 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	MCSP14_SIMO_INPUTENABLE	Input enable value for pad mcspi4_simo 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	MCSP14_SIMO_PULLTYPESELECT	pullup/down selection for pad mcspi4_simo 0x0: pulldown selected 0x1: pullup selected	RW	1

Bits	Field Name	Description	Type	Reset
19	MCSPi4_SIMO_PULLUDENABLE	pullup/down enable for pad mcspi4_simo 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	MCSPi4_SIMO_MUXMODE	Functional multiplexing selection for pad mcspi4_simo 0x0: Select mcspi4_simo 0x1: Select sdmmc4_cmd 0x2: Select kpd_col7 0x3: Select gpio_152 0x7: Select safe_mode	RW	0x7
15	MCSPi4_CLK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	MCSPi4_CLK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	MCSPi4_CLK_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi4_clk 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	MCSPi4_CLK_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad mcspi4_clk 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	MCSPi4_CLK_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi4_clk 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	MCSPi4_CLK_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi4_clk. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	MCSPi4_CLK_OFFMODEENABLE	OffMode mode override control for pad mcspi4_clk 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	MCSPi4_CLK_INPUTENABLE	Input enable value for pad mcspi4_clk 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	MCSPi4_CLK_PULLTYPESELECT	pullup/down selection for pad mcspi4_clk 0x0: pulldown selected 0x1: pullup selected	RW	0
3	MCSPi4_CLK_PULLUDENABLE	pullup/down enable for pad mcspi4_clk 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	MCSPi4_CLK_MUXMODE	Functional multiplexing selection for pad mcspi4_clk 0x0: Select mcspi4_clk 0x1: Select sdmmc4_clk 0x2: Select kpd_col6 0x3: Select gpio_151 0x7: Select safe_mode	RW	0x7

**Table 18-461. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_MCSPi4\_CLK\_PAD1\_MCSPi4\_SIMO**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-462. CONTROL\_CORE\_PAD0\_MCSPi4\_SOMI\_PAD1\_MCSPi4\_CS0**

<b>Address Offset</b>	0x0000 0158	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads mcspi4_somi and mcspi4_cs0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSPi4_CS0_WAKEUPEVENT	MCSPi4_CS0_WAKEUPENABLE	MCSPi4_CS0_OFFMODEPULLTYPESELECT	MCSPi4_CS0_OFFMODEPULLUDENABLE	MCSPi4_CS0_OFFMODEOUTVALUE	MCSPi4_CS0_OFFMODEOUTENABLE	MCSPi4_CS0_OFFMODEENABLE	MCSPi4_CS0_INPUTENABLE	RESERVED			MCSPi4_CS0_PULLTYPESELECT	MCSPi4_CS0_PULLUDENABLE		MCSPi4_CS0_MUXMODE		MCSPi4_SOMI_WAKEUPEVENT	MCSPi4_SOMI_WAKEUPENABLE	MCSPi4_SOMI_OFFMODEPULLTYPESELECT	MCSPi4_SOMI_OFFMODEPULLUDENABLE	MCSPi4_SOMI_OFFMODEOUTVALUE	MCSPi4_SOMI_OFFMODEOUTENABLE	MCSPi4_SOMI_OFFMODEENABLE	MCSPi4_SOMI_INPUTENABLE	RESERVED			MCSPi4_SOMI_PULLTYPESELECT	MCSPi4_SOMI_PULLUDENABLE		MCSPi4_SOMI_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	MCSPi4_CS0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	MCSPi4_CS0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	MCSPi4_CS0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi4_cs0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	MCSPi4_CS0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad mcspi4_cs0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	MCSPi4_CS0_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi4_cs0 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	MCSPi4_CS0_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi4_cs0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
25	MCSPi4_CS0_OFFMODEENABLE	OffMode mode override control for pad mcspi4_cs0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	MCSPi4_CS0_INPUTENABLE	Input enable value for pad mcspi4_cs0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	MCSPi4_CS0_PULLTYPESELECT	pullup/down selection for pad mcspi4_cs0 0x0: pulldown selected 0x1: pullup selected	RW	1
19	MCSPi4_CS0_PULLUDENABLE	pullup/down enable for pad mcspi4_cs0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	MCSPi4_CS0_MUXMODE	Functional multiplexing selection for pad mcspi4_cs0 0x0: Select mcspi4_cs0 0x1: Select sdmmc4_dat3 0x2: Select kpd_row7 0x3: Select gpio_154 0x7: Select safe_mode	RW	0x7
15	MCSPi4_SOMI_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	MCSPi4_SOMI_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	MCSPi4_SOMI_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad mcspi4_somi 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	MCSPi4_SOMI_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad mcspi4_somi 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	MCSPi4_SOMI_OFFMODEOUTVALUE	OffMode mode output value for pad mcspi4_somi 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	MCSPi4_SOMI_OFFMODEOUTENABLE	OffMode mode output enable value for pad mcspi4_somi. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	MCSPi4_SOMI_OFFMODEENABLE	OffMode mode override control for pad mcspi4_somi 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	MCSPi4_SOMI_INPUTENABLE	Input enable value for pad mcspi4_somi 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	MCSPi4_SOMI_PULLTYPESELECT	pullup/down selection for pad mcspi4_somi 0x0: pulldown selected 0x1: pullup selected	RW	1
3	MCSPi4_SOMI_PULLUDENABLE	pullup/down enable for pad mcspi4_somi 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	MCSPi4_SOMI_MUXMODE	Functional multiplexing selection for pad mcspi4_somi 0x0: Select mcspi4_somi 0x1: Select sdmmc4_dat0 0x2: Select kpd_row6 0x3: Select gpio_153 0x7: Select safe_mode	RW	0x7

**Table 18-463. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_MCSPi4\_SOMI\_PAD1\_MCSPi4\_CS0**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-464. CONTROL\_CORE\_PAD0\_UART4\_RX\_PAD1\_UART4\_TX**

<b>Address Offset</b>	0x0000 015C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads uart4_rx and uart4_tx Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART4_TX_WAKEUPEVENT	UART4_TX_WAKEUPENABLE	UART4_TX_OFFMODEPULLTYPESELECT	UART4_TX_OFFMODEPULLUDENABLE	UART4_TX_OFFMODEOUTVALUE	UART4_TX_OFFMODEOUTENABLE	UART4_TX_OFFMODEEENABLE	UART4_TX_INPUTENABLE	RESERVED			UART4_TX_PULLTYPESELECT	UART4_TX_PULLUDENABLE	UART4_TX_MUXMODE			UART4_RX_WAKEUPEVENT	UART4_RX_WAKEUPENABLE	UART4_RX_OFFMODEPULLTYPESELECT	UART4_RX_OFFMODEPULLUDENABLE	UART4_RX_OFFMODEOUTVALUE	UART4_RX_OFFMODEOUTENABLE	UART4_RX_OFFMODEEENABLE	UART4_RX_INPUTENABLE	RESERVED			UART4_RX_PULLTYPESELECT	UART4_RX_PULLUDENABLE	UART4_RX_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	UART4_TX_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	UART4_TX_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0



Bits	Field Name	Description	Type	Reset
29	UART4_TX_ OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad uart4_tx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	UART4_TX_ OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad uart4_tx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	UART4_TX_ OFFMODEOUTVALUE	OffMode mode output value for pad uart4_tx 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	UART4_TX_ OFFMODEOUTENABLE	OffMode mode output enable value for pad uart4_tx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	UART4_TX_ OFFMODEENABLE	OffMode mode override control for pad uart4_tx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	UART4_TX_ INPUTENABLE	Input enable value for pad uart4_tx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	UART4_TX_ PULLTYPESELECT	pullup/down selection for pad uart4_tx 0x0: pulldown selected 0x1: pullup selected	RW	1
19	UART4_TX_ PULLUDENABLE	pullup/down enable for pad uart4_tx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	UART4_TX_ MUXMODE	Functional multiplexing selection for pad uart4_tx 0x0: Select uart4_tx 0x1: Select sdmmc4_dat1 0x2: Select kpd_col8 0x3: Select gpio_156 0x7: Select safe_mode	RW	0x7
15	UART4_RX_ WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	UART4_RX_ WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	UART4_RX_ OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad uart4_rx 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	UART4_RX_ OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad uart4_rx 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	UART4_RX_ OFFMODEOUTVALUE	OffMode mode output value for pad uart4_rx 0x0: Set value at 0 0x1: Set value at 1	RW	0

Bits	Field Name	Description	Type	Reset
10	UART4_RX_OFFMODEOUTENABLE	OffMode mode output enable value for pad uart4_rx. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	UART4_RX_OFFMODEENABLE	OffMode mode override control for pad uart4_rx 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	UART4_RX_INPUTENABLE	Input enable value for pad uart4_rx 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	UART4_RX_PULLTYPESELECT	pullup/down selection for pad uart4_rx 0x0: pulldown selected 0x1: pullup selected	RW	1
3	UART4_RX_PULLUDENABLE	pullup/down enable for pad uart4_rx 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	UART4_RX_MUXMODE	Functional multiplexing selection for pad uart4_rx 0x0: Select uart4_rx 0x1: Select sdmmc4_dat2 0x2: Select kpd_row8 0x3: Select gpio_155 0x7: Select safe_mode	RW	0x7

**Table 18-465. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_UART4\_RX\_PAD1\_UART4\_TX**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-466. CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_CLK\_PAD1\_USBB2\_ULPITLL\_STP**

<b>Address Offset</b>	0x0000 0160	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb2_ulpitll_clk and usbb2_ulpitll_stp Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB2_ULPITLL_STP_WAKEUPEVENT	USBB2_ULPITLL_STP_WAKEUPENABLE	USBB2_ULPITLL_STP_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_STP_OFFMODEPULLUDENABLE	USBB2_ULPITLL_STP_OFFMODEOUTVALUE	USBB2_ULPITLL_STP_OFFMODEOUTENABLE	USBB2_ULPITLL_STP_OFFMODEENABLE	USBB2_ULPITLL_STP_INPUTENABLE	RESERVED			USBB2_ULPITLL_STP_PULLTYPESELECT	USBB2_ULPITLL_STP_PULLUDENABLE	USBB2_ULPITLL_STP_MUXMODE			USBB2_ULPITLL_CLK_WAKEUPEVENT	USBB2_ULPITLL_CLK_WAKEUPENABLE	USBB2_ULPITLL_CLK_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_CLK_OFFMODEPULLUDENABLE	USBB2_ULPITLL_CLK_OFFMODEOUTVALUE	USBB2_ULPITLL_CLK_OFFMODEOUTENABLE	USBB2_ULPITLL_CLK_OFFMODEENABLE	USBB2_ULPITLL_CLK_INPUTENABLE	RESERVED			USBB2_ULPITLL_CLK_PULLTYPESELECT	USBB2_ULPITLL_CLK_PULLUDENABLE	USBB2_ULPITLL_CLK_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	USBB2_ULPITLL_STP_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB2_ULPITLL_STP_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB2_ULPITLL_STP_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_stp 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB2_ULPITLL_STP_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_stp 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB2_ULPITLL_STP_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_stp 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB2_ULPITLL_STP_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_stp. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB2_ULPITLL_STP_OFFMODEENABLE	OffMode mode override control for pad usbb2_ulpitll_stp 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB2_ULPITLL_STP_INPUTENABLE	Input enable value for pad usbb2_ulpitll_stp 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20	USBB2_ULPITLL_STP_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_stp 0x0: pulldown selected 0x1: pullup selected	RW	1
19	USBB2_ULPITLL_STP_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_stp 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB2_ULPITLL_STP_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_stp 0x0: Select usbb2_ulpitll_stp 0x1: Select usbb2_ulpiphy_stp 0x2: Select sdmmc4_clk 0x3: Select gpio_158 0x4: Select hsi2_cadata 0x5: Select dispc2_data23 0x7: Select safe_mode	RW	0x7
15	USBB2_ULPITLL_CLK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB2_ULPITLL_CLK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB2_ULPITLL_CLK_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_clk 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB2_ULPITLL_CLK_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_clk 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB2_ULPITLL_CLK_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_clk 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB2_ULPITLL_CLK_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_clk. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB2_ULPITLL_CLK_OFFMODEENABLE	OffMode mode override control for pad usbb2_ulpitll_clk 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	USBB2_ULPITLL_CLK_INPUTENABLE	Input enable value for pad usbb2_ulpitll_clk 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB2_ULPITLL_CLK_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_clk 0x0: pulldown selected 0x1: pullup selected	RW	0
3	USBB2_ULPITLL_CLK_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_clk 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
2:0	USBB2_ULPITLL_CLK_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_clk 0x0: Select usbb2_ulpitll_clk 0x1: Select usbb2_ulpiphy_clk 0x2: Select sdmmc4_cmd 0x3: Select gpio_157 0x4: Select hsi2_cawake 0x7: Select safe_mode	RW	0x7

**Table 18-467. Register Call Summary for Register CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_CLK\_PAD1\_USBB2\_ULPITLL\_STP**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-468. CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DIR\_PAD1\_USBB2\_ULPITLL\_NXT**

<b>Address Offset</b>	0x0000 0164	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb2_ulpitll_dir and usbb2_ulpitll_nxt Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB2_ULPITLL_NXT_WAKEUPEVENT	USBB2_ULPITLL_NXT_WAKEUPENABLE	USBB2_ULPITLL_NXT_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_NXT_OFFMODEPULLUDENENABLE	USBB2_ULPITLL_NXT_OFFMODEOUTVALUE	USBB2_ULPITLL_NXT_OFFMODEOUTENABLE	USBB2_ULPITLL_NXT_OFFMODEEENABLE	USBB2_ULPITLL_NXT_INPUTENABLE	RESERVED			USBB2_ULPITLL_NXT_PULLTYPESELECT	USBB2_ULPITLL_NXT_PULLUDENENABLE	USBB2_ULPITLL_NXT_MUXMODE			USBB2_ULPITLL_DIR_WAKEUPEVENT	USBB2_ULPITLL_DIR_WAKEUPENABLE	USBB2_ULPITLL_DIR_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_DIR_OFFMODEPULLUDENENABLE	USBB2_ULPITLL_DIR_OFFMODEOUTVALUE	USBB2_ULPITLL_DIR_OFFMODEOUTENABLE	USBB2_ULPITLL_DIR_OFFMODEEENABLE	USBB2_ULPITLL_DIR_INPUTENABLE	RESERVED			USBB2_ULPITLL_DIR_PULLTYPESELECT	USBB2_ULPITLL_DIR_PULLUDENENABLE	USBB2_ULPITLL_DIR_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	USBB2_ULPITLL_NXT_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB2_ULPITLL_NXT_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
29	USBB2_ULPITLL_NXT_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_nxt 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB2_ULPITLL_NXT_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_nxt 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB2_ULPITLL_NXT_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_nxt 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB2_ULPITLL_NXT_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_nxt. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB2_ULPITLL_NXT_OFFMODEENABLE	OffMode mode override control for pad usbb2_ulpitll_nxt 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB2_ULPITLL_NXT_INPUTENABLE	Input enable value for pad usbb2_ulpitll_nxt 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	USBB2_ULPITLL_NXT_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_nxt 0x0: pulldown selected 0x1: pullup selected	RW	1
19	USBB2_ULPITLL_NXT_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_nxt 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB2_ULPITLL_NXT_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_nxt 0x0: Select usbb2_ulpitll_nxt 0x1: Select usbb2_ulpiphy_nxt 0x2: Select sdmmc4_dat1 0x3: Select gpio_160 0x4: Select hsi2_acready 0x5: Select disp2_data21 0x7: Select safe_mode	RW	0x7
15	USBB2_ULPITLL_DIR_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB2_ULPITLL_DIR_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB2_ULPITLL_DIR_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_dir 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	USBB2_ULPITLL_DIR_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_dir 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB2_ULPITLL_DIR_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_dir 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB2_ULPITLL_DIR_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_dir. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB2_ULPITLL_DIR_OFFMODEENABLE	OffMode mode override control for pad usbb2_ulpitll_dir 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	USBB2_ULPITLL_DIR_INPUTENABLE	Input enable value for pad usbb2_ulpitll_dir 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB2_ULPITLL_DIR_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_dir 0x0: pulldown selected 0x1: pullup selected	RW	1
3	USBB2_ULPITLL_DIR_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_dir 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB2_ULPITLL_DIR_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_dir 0x0: Select usbb2_ulpitll_dir 0x1: Select usbb2_ulpiphy_dir 0x2: Select sdmmc4_dat0 0x3: Select gpio_159 0x4: Select hsi2_cafmag 0x5: Select dispc2_data22 0x7: Select safe_mode	RW	0x7

**Table 18-469. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DIR\_PAD1\_USBB2\_ULPITLL\_NXT**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-470. CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DAT0\_PAD1\_USBB2\_ULPITLL\_DAT1**

<b>Address Offset</b>	0x0000 0168		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads usbb2_ulpitll_dat0 and usbb2_ulpitll_dat1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB2_ULPITLL_DAT1_WAKEUPEVENT	USBB2_ULPITLL_DAT1_WAKEUPENABLE	USBB2_ULPITLL_DAT1_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_DAT1_OFFMODEPULLUDENABLE	USBB2_ULPITLL_DAT1_OFFMODEOUTVALUE	USBB2_ULPITLL_DAT1_OFFMODEOUTENABLE	USBB2_ULPITLL_DAT1_OFFMODEENABLE	USBB2_ULPITLL_DAT1_INPUTENABLE	RESERVED	RESERVED	RESERVED	USBB2_ULPITLL_DAT1_PULLTYPESELECT	USBB2_ULPITLL_DAT1_PULLUDENABLE	USBB2_ULPITLL_DAT1_MUXMODE	USBB2_ULPITLL_DAT0_WAKEUPEVENT	USBB2_ULPITLL_DAT0_WAKEUPENABLE	USBB2_ULPITLL_DAT0_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_DAT0_OFFMODEPULLUDENABLE	USBB2_ULPITLL_DAT0_OFFMODEOUTVALUE	USBB2_ULPITLL_DAT0_OFFMODEOUTENABLE	USBB2_ULPITLL_DAT0_OFFMODEENABLE	USBB2_ULPITLL_DAT0_INPUTENABLE	RESERVED	RESERVED	RESERVED	USBB2_ULPITLL_DAT0_PULLTYPESELECT	USBB2_ULPITLL_DAT0_PULLUDENABLE	USBB2_ULPITLL_DAT0_MUXMODE				

Bits	Field Name	Description	Type	Reset
31	USBB2_ULPITLL_DAT1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB2_ULPITLL_DAT1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB2_ULPITLL_DAT1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_dat1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB2_ULPITLL_DAT1_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_dat1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB2_ULPITLL_DAT1_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_dat1 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB2_ULPITLL_DAT1_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_dat1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0



Bits	Field Name	Description	Type	Reset
25	USBB2_ULPITLL_DAT1_OFFM ODEENABLE	OffMode mode override control for pad usbb2_ulpitll_dat1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB2_ULPITLL_DAT1_INPUTENABLE	Input enable value for pad usbb2_ulpitll_dat1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	USBB2_ULPITLL_DAT1_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_dat1 0x0: pulldown selected 0x1: pullup selected	RW	1
19	USBB2_ULPITLL_DAT1_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_dat1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB2_ULPITLL_DAT1_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_dat1 0x0: Select usbb2_ulpitll_dat1 0x1: Select usbb2_ulpiphy_dat1 0x2: Select sdmmc4_dat3 0x3: Select gpio_162 0x4: Select hsi2_acdata 0x5: Select dispc2_data19 0x6: Select usbb2_mm_txdat 0x7: Select safe_mode	RW	0x7
15	USBB2_ULPITLL_DAT0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB2_ULPITLL_DAT0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB2_ULPITLL_DAT0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_dat0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB2_ULPITLL_DAT0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_dat0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB2_ULPITLL_DAT0_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_dat0 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB2_ULPITLL_DAT0_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_dat0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB2_ULPITLL_DAT0_OFFMODEENABLE	OffMode mode override control for pad usbb2_ulpitll_dat0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
8	USBB2_ULPITLL_DAT0_INPUTENABLE	Input enable value for pad usbb2_ulpitll_dat0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB2_ULPITLL_DAT0_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_dat0 0x0: pulldown selected 0x1: pullup selected	RW	1
3	USBB2_ULPITLL_DAT0_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_dat0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB2_ULPITLL_DAT0_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_dat0 0x0: Select usbb2_ulpitll_dat0 0x1: Select usbb2_ulpiPHY_dat0 0x2: Select sdmmc4_dat2 0x3: Select gpio_161 0x4: Select hsi2_acwake 0x5: Select dispC2_data20 0x6: Select usbb2_mm_txen 0x7: Select safe_mode	RW	0x7

**Table 18-471. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DAT0\_PAD1\_USBB2\_ULPITLL\_DAT1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-472. CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DAT2\_PAD1\_USBB2\_ULPITLL\_DAT3**

<b>Address Offset</b>	0x0000 016C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb2_ulpitll_dat2 and usbb2_ulpitll_dat3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB2_ULPITLL_DAT3_WAKEUPEVENT	USBB2_ULPITLL_DAT3_WAKEUPENABLE	USBB2_ULPITLL_DAT3_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_DAT3_OFFMODEPULLUDENABLE	USBB2_ULPITLL_DAT3_OFFMODEOUTVALUE	USBB2_ULPITLL_DAT3_OFFMODEOUTENABLE	USBB2_ULPITLL_DAT3_OFFMODEEENABLE	USBB2_ULPITLL_DAT3_INPUTENABLE	RESERVED			USBB2_ULPITLL_DAT3_PULLTYPESELECT	USBB2_ULPITLL_DAT3_PULLUDENABLE		USBB2_ULPITLL_DAT3_MUXMODE		USBB2_ULPITLL_DAT2_WAKEUPEVENT	USBB2_ULPITLL_DAT2_WAKEUPENABLE	USBB2_ULPITLL_DAT2_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_DAT2_OFFMODEPULLUDENABLE	USBB2_ULPITLL_DAT2_OFFMODEOUTVALUE	USBB2_ULPITLL_DAT2_OFFMODEOUTENABLE	USBB2_ULPITLL_DAT2_OFFMODEEENABLE	USBB2_ULPITLL_DAT2_INPUTENABLE	RESERVED			USBB2_ULPITLL_DAT2_PULLTYPESELECT	USBB2_ULPITLL_DAT2_PULLUDENABLE		USBB2_ULPITLL_DAT2_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	USBB2_ULPITLL_DAT3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB2_ULPITLL_DAT3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB2_ULPITLL_DAT3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_dat3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB2_ULPITLL_DAT3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_dat3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB2_ULPITLL_DAT3_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_dat3 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB2_ULPITLL_DAT3_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_dat3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB2_ULPITLL_DAT3_OFFMODEEENABLE	OffMode mode override control for pad usbb2_ulpitll_dat3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB2_ULPITLL_DAT3_INPUTENABLE	Input enable value for pad usbb2_ulpitll_dat3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20	USBB2_ULPITLL_DAT3_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_dat3 0x0: pulldown selected 0x1: pullup selected	RW	1
19	USBB2_ULPITLL_DAT3_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_dat3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB2_ULPITLL_DAT3_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_dat3 0x0: Select usbb2_ulpitll_dat3 0x1: Select usbb2_ulpiphy_dat3 0x2: Select sdmmc3_dat1 0x3: Select gpio_164 0x4: Select hsi2_caready 0x5: Select dispc2_data15 0x6: Select rfb_data15 0x7: Select safe_mode	RW	0x7
15	USBB2_ULPITLL_DAT2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB2_ULPITLL_DAT2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB2_ULPITLL_DAT2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_dat2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB2_ULPITLL_DAT2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_dat2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB2_ULPITLL_DAT2_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_dat2 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB2_ULPITLL_DAT2_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_dat2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB2_ULPITLL_DAT2_OFFMODEENABLE	OffMode mode override control for pad usbb2_ulpitll_dat2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	USBB2_ULPITLL_DAT2_INPUTENABLE	Input enable value for pad usbb2_ulpitll_dat2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB2_ULPITLL_DAT2_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_dat2 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	USBB2_ULPITLL_DAT2_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_dat2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB2_ULPITLL_DAT2_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_dat2 0x0: Select usbb2_ulpitll_dat2 0x1: Select usbb2_ulpiphy_dat2 0x2: Select sdmmc3_dat2 0x3: Select gpio_163 0x4: Select hsi2_acflag 0x5: Select dispc2_data18 0x6: Select usbb2_mm_txse0 0x7: Select safe_mode	RW	0x7

**Table 18-473. Register Call Summary for Register CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DAT2\_PAD1\_USBB2\_ULPITLL\_DAT3**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-474. CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DAT4\_PAD1\_USBB2\_ULPITLL\_DAT5**

<b>Address Offset</b>	0x0000 0170	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb2_ulpitll_dat4 and usbb2_ulpitll_dat5 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB2_ULPITLL_DAT5_WAKEUPEVENT	USBB2_ULPITLL_DAT5_WAKEUPENABLE	USBB2_ULPITLL_DAT5_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_DAT5_OFFMODEPULLUDENABLE	USBB2_ULPITLL_DAT5_OFFMODEOUTVALUE	USBB2_ULPITLL_DAT5_OFFMODEOUTENABLE	USBB2_ULPITLL_DAT5_OFFMODEENABLE	USBB2_ULPITLL_DAT5_INPUTENABLE	RESERVED			USBB2_ULPITLL_DAT5_PULLTYPESELECT	USBB2_ULPITLL_DAT5_PULLUDENABLE	USBB2_ULPITLL_DAT5_MUXMODE			USBB2_ULPITLL_DAT4_WAKEUPEVENT	USBB2_ULPITLL_DAT4_WAKEUPENABLE	USBB2_ULPITLL_DAT4_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_DAT4_OFFMODEPULLUDENABLE	USBB2_ULPITLL_DAT4_OFFMODEOUTVALUE	USBB2_ULPITLL_DAT4_OFFMODEOUTENABLE	USBB2_ULPITLL_DAT4_OFFMODEENABLE	USBB2_ULPITLL_DAT4_INPUTENABLE	RESERVED			USBB2_ULPITLL_DAT4_PULLTYPESELECT	USBB2_ULPITLL_DAT4_PULLUDENABLE	USBB2_ULPITLL_DAT4_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	USBB2_ULPITLL_DAT5_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB2_ULPITLL_DAT5_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB2_ULPITLL_DAT5_OFFM_ODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_dat5 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB2_ULPITLL_DAT5_OFFM_ODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_dat5 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB2_ULPITLL_DAT5_OFFM_ODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_dat5 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB2_ULPITLL_DAT5_OFFM_ODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_dat5. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB2_ULPITLL_DAT5_OFFM_ODEENABLE	OffMode mode override control for pad usbb2_ulpitll_dat5 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB2_ULPITLL_DAT5_INPUTENABLE	Input enable value for pad usbb2_ulpitll_dat5 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	USBB2_ULPITLL_DAT5_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_dat5 0x0: pulldown selected 0x1: pullup selected	RW	1
19	USBB2_ULPITLL_DAT5_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_dat5 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB2_ULPITLL_DAT5_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_dat5 0x0: Select usbb2_ulpitll_dat5 0x1: Select usbb2_ulpiphy_dat5 0x2: Select sdmmc3_dat3 0x3: Select gpio_166 0x4: Select mcspi3_cs0 0x5: Select disp2_data13 0x6: Select rfb_data13 0x7: Select safe_mode	RW	0x7
15	USBB2_ULPITLL_DAT4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
14	USBB2_ULPITLL_DAT4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB2_ULPITLL_DAT4_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_dat4 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB2_ULPITLL_DAT4_OFFMODEPULLUDENAB LE	OffMode mode pullup/down enable for pad usbb2_ulpitll_dat4 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB2_ULPITLL_DAT4_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_dat4 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB2_ULPITLL_DAT4_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_dat4. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB2_ULPITLL_DAT4_OFFMODEENABLE	OffMode mode override control for pad usbb2_ulpitll_dat4 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	USBB2_ULPITLL_DAT4_INPUTENABLE	Input enable value for pad usbb2_ulpitll_dat4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB2_ULPITLL_DAT4_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_dat4 0x0: pulldown selected 0x1: pullup selected	RW	1
3	USBB2_ULPITLL_DAT4_PULLUDENAB LE	pullup/down enable for pad usbb2_ulpitll_dat4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB2_ULPITLL_DAT4_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_dat4 0x0: Select usbb2_ulpitll_dat4 0x1: Select usbb2_ulpiphy_dat4 0x2: Select sdmmc3_dat0 0x3: Select gpio_165 0x4: Select mcspi3_somi 0x5: Select dispc2_data14 0x6: Select rfi_data14 0x7: Select safe_mode	RW	0x7

**Table 18-475. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DAT4\_PAD1\_USBB2\_ULPITLL\_DAT5**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-476. CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DAT6\_PAD1\_USBB2\_ULPITLL\_DAT7**

<b>Address Offset</b>	0x0000 0174		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads usbb2_ulpitll_dat6 and usbb2_ulpitll_dat7 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBB2_ULPITLL_DAT7_WAKEUPEVENT	USBB2_ULPITLL_DAT7_WAKEUPENABLE	USBB2_ULPITLL_DAT7_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_DAT7_OFFMODEPULLUDENENABLE	USBB2_ULPITLL_DAT7_OFFMODEOUTVALUE	USBB2_ULPITLL_DAT7_OFFMODEOUTENABLE	USBB2_ULPITLL_DAT7_OFFMODEENABLE	USBB2_ULPITLL_DAT7_INPUTENABLE	RESERVED	RESERVED	RESERVED	USBB2_ULPITLL_DAT7_PULLTYPESELECT	USBB2_ULPITLL_DAT7_PULLUDENENABLE	USBB2_ULPITLL_DAT7_MUXMODE	USBB2_ULPITLL_DAT6_WAKEUPEVENT	USBB2_ULPITLL_DAT6_WAKEUPENABLE	USBB2_ULPITLL_DAT6_OFFMODEPULLTYPESELECT	USBB2_ULPITLL_DAT6_OFFMODEPULLUDENENABLE	USBB2_ULPITLL_DAT6_OFFMODEOUTVALUE	USBB2_ULPITLL_DAT6_OFFMODEOUTENABLE	USBB2_ULPITLL_DAT6_OFFMODEENABLE	USBB2_ULPITLL_DAT6_INPUTENABLE	RESERVED	RESERVED	RESERVED	USBB2_ULPITLL_DAT6_PULLTYPESELECT	USBB2_ULPITLL_DAT6_PULLUDENENABLE	USBB2_ULPITLL_DAT6_MUXMODE				

Bits	Field Name	Description	Type	Reset
31	USBB2_ULPITLL_DAT7_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB2_ULPITLL_DAT7_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	USBB2_ULPITLL_DAT7_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_dat7 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	USBB2_ULPITLL_DAT7_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_dat7 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	USBB2_ULPITLL_DAT7_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_dat7 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB2_ULPITLL_DAT7_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_dat7. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0



Bits	Field Name	Description	Type	Reset
25	USBB2_ULPITLL_DAT7_OFFM ODEENABLE	OffMode mode override control for pad usbb2_ulpitll_dat7 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	USBB2_ULPITLL_DAT7_INPUTENABLE	Input enable value for pad usbb2_ulpitll_dat7 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	USBB2_ULPITLL_DAT7_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_dat7 0x0: pulldown selected 0x1: pullup selected	RW	1
19	USBB2_ULPITLL_DAT7_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_dat7 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBB2_ULPITLL_DAT7_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_dat7 0x0: Select usbb2_ulpitll_dat7 0x1: Select usbb2_ulpiphy_dat7 0x2: Select sdmmc3_clk 0x3: Select gpio_168 0x4: Select mcspi3_clk 0x5: Select dispc2_data11 0x6: Select rfb_data11 0x7: Select safe_mode	RW	0x7
15	USBB2_ULPITLL_DAT6_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB2_ULPITLL_DAT6_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	USBB2_ULPITLL_DAT6_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad usbb2_ulpitll_dat6 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	USBB2_ULPITLL_DAT6_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad usbb2_ulpitll_dat6 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	USBB2_ULPITLL_DAT6_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_ulpitll_dat6 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB2_ULPITLL_DAT6_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_ulpitll_dat6. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB2_ULPITLL_DAT6_OFFMODEENABLE	OffMode mode override control for pad usbb2_ulpitll_dat6 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
8	USBB2_ULPITLL_DAT6_INPUTENABLE	Input enable value for pad usbb2_ulpitll_dat6 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	USBB2_ULPITLL_DAT6_PULLTYPESELECT	pullup/down selection for pad usbb2_ulpitll_dat6 0x0: pulldown selected 0x1: pullup selected	RW	1
3	USBB2_ULPITLL_DAT6_PULLUDENABLE	pullup/down enable for pad usbb2_ulpitll_dat6 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBB2_ULPITLL_DAT6_MUXMODE	Functional multiplexing selection for pad usbb2_ulpitll_dat6 0x0: Select usbb2_ulpitll_dat6 0x1: Select usbb2_ulpiphy_dat6 0x2: Select sdmmc3_cmd 0x3: Select gpio_167 0x4: Select mcspi3_simo 0x5: Select disp2_data12 0x6: Select rfi_data12 0x7: Select safe_mode	RW	0x7

**Table 18-477. Register Call Summary for Register CONTROL\_CORE\_PAD0\_USBB2\_ULPITLL\_DAT6\_PAD1\_USBB2\_ULPITLL\_DAT7**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-478. CONTROL\_CORE\_PAD0\_USBB2\_HSIC\_DATA\_PAD1\_USBB2\_HSIC\_STROBE**

<b>Address Offset</b>	0x0000 0178	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads usbb2_hsic_data and usbb2_hsic_strobe Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
USBB2_HSIC_STROBE_WAKEUPEVENT USBB2_HSIC_STROBE_WAKEUPENABLE		RESERVED		USBB2_HSIC_STROBE_OFFMODEOUTVALUE USBB2_HSIC_STROBE_OFFMODEOUTENABLE		USBB2_HSIC_STROBE_OFFMODEENABLE		RESERVED								USBB2_HSIC_STROBE_MUXMODE		USBB2_HSIC_DATA_WAKEUPEVENT USBB2_HSIC_DATA_WAKEUPENABLE		RESERVED		USBB2_HSIC_DATA_OFFMODEOUTVALUE USBB2_HSIC_DATA_OFFMODEOUTENABLE		USBB2_HSIC_DATA_OFFMODEENABLE		RESERVED								USBB2_HSIC_DATA_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	USBB2_HSIC_STROBE_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	USBB2_HSIC_STROBE_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:28	RESERVED		R	0x0
27	USBB2_HSIC_STROBE_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_hsic_strobe 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	USBB2_HSIC_STROBE_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_hsic_strobe. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	USBB2_HSIC_STROBE_OFFMODEENABLE	OffMode mode override control for pad usbb2_hsic_strobe 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24:19	RESERVED		R	0x00
18:16	USBB2_HSIC_STROBE_MUXMODE	Functional multiplexing selection for pad usbb2_hsic_strobe 0x0: Select usbb2_hsic_strobe 0x3: Select gpio_170 0x7: Select safe_mode	RW	0x7
15	USBB2_HSIC_DATA_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB2_HSIC_DATA_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11	USBB2_HSIC_DATA_OFFMODEOUTVALUE	OffMode mode output value for pad usbb2_hsic_data 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	USBB2_HSIC_DATA_OFFMODEOUTENABLE	OffMode mode output enable value for pad usbb2_hsic_data. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	USBB2_HSIC_DATA_OFFMODEENABLE	OffMode mode override control for pad usbb2_hsic_data 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8:3	RESERVED		R	0x00
2:0	USBB2_HSIC_DATA_MUXMODE	Functional multiplexing selection for pad usbb2_hsic_data 0x0: Select usbb2_hsic_data 0x3: Select gpio_169 0x7: Select safe_mode	RW	0x7

**Table 18-479. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBB2\_HSIC\_DATA\_PAD1\_USBB2\_HSIC\_STROBE**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-480. CONTROL\_CORE\_PAD0\_KPD\_COL3\_PAD1\_KPD\_COL4**

<b>Address Offset</b>	0x0000 017C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads kpd_col3 and kpd_col4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
KPD_COL4_WAKEUPEVENT	KPD_COL4_WAKEUPENABLE	KPD_COL4_OFFMODEPULLTYPESELECT	KPD_COL4_OFFMODEPULLUDENENABLE	KPD_COL4_OFFMODEOUTVALUE	KPD_COL4_OFFMODEOUTENABLE	KPD_COL4_OFFMODEENABLE	KPD_COL4_INPUTENABLE	RESERVED			KPD_COL4_PULLTYPESELECT	KPD_COL4_PULLUDENENABLE			KPD_COL4_MUXMODE			KPD_COL3_WAKEUPEVENT	KPD_COL3_WAKEUPENABLE	KPD_COL3_OFFMODEPULLTYPESELECT	KPD_COL3_OFFMODEPULLUDENENABLE	KPD_COL3_OFFMODEOUTVALUE	KPD_COL3_OFFMODEOUTENABLE	KPD_COL3_OFFMODEENABLE	KPD_COL3_INPUTENABLE	RESERVED			KPD_COL3_PULLTYPESELECT	KPD_COL3_PULLUDENENABLE			KPD_COL3_MUXMODE

Bits	Field Name	Description	Type	Reset
31	KPD_COL4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	KPD_COL4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	KPD_COL4_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad kpd_col4 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	KPD_COL4_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad kpd_col4 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	KPD_COL4_OFFMODEOUTVALUE	OffMode mode output value for pad kpd_col4 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	KPD_COL4_OFFMODEOUTENABLE	OffMode mode output enable value for pad kpd_col4. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	KPD_COL4_OFFMODEENABLE	OffMode mode override control for pad kpd_col4 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	KPD_COL4_INPUTENABLE	Input enable value for pad kpd_col4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	KPD_COL4_PULLTYPESELECT	pullup/down selection for pad kpd_col4 0x0: pulldown selected 0x1: pullup selected	RW	0
19	KPD_COL4_PULLUDENABLE	pullup/down enable for pad kpd_col4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	KPD_COL4_MUXMODE	Functional multiplexing selection for pad kpd_col4 0x0: Select kpd_col4 0x1: Select kpd_col1 0x3: Select gpio_172 0x7: Select safe_mode	RW	0x7
15	KPD_COL3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
14	KPD_COL3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	KPD_COL3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad kpd_col3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	KPD_COL3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad kpd_col3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	KPD_COL3_OFFMODEOUTVALUE	OffMode mode output value for pad kpd_col3 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	KPD_COL3_OFFMODEOUTENABLE	OffMode mode output enable value for pad kpd_col3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	KPD_COL3_OFFMODEENABLE	OffMode mode override control for pad kpd_col3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	KPD_COL3_INPUTENABLE	Input enable value for pad kpd_col3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	KPD_COL3_PULLTYPESELECT	pullup/down selection for pad kpd_col3 0x0: pulldown selected 0x1: pullup selected	RW	0
3	KPD_COL3_PULLUDENABLE	pullup/down enable for pad kpd_col3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	KPD_COL3_MUXMODE	Functional multiplexing selection for pad kpd_col3 0x0: Select kpd_col3 0x1: Select kpd_col0 0x3: Select gpio_171 0x7: Select safe_mode	RW	0x7

**Table 18-481. Register Call Summary for Register CONTROL\_CORE\_PAD0\_KPD\_COL3\_PAD1\_KPD\_COL4**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-482. CONTROL\_CORE\_PAD0\_KPD\_COL5\_PAD1\_KPD\_COLO**

<b>Address Offset</b>	0x0000 0180		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads kpd_col5 and kpd_col0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
KPD_COLO_WAKEUPEVENT	KPD_COLO_WAKEUPENABLE	KPD_COLO_OFFMODEPULLTYPESELECT	KPD_COLO_OFFMODEPULLUDENABLE	KPD_COLO_OFFMODEOUTVALUE	KPD_COLO_OFFMODEOUTENABLE	KPD_COLO_OFFMODEENABLE	KPD_COLO_INPUTENABLE	RESERVED			KPD_COLO_PULLTYPESELECT	KPD_COLO_PULLUDENABLE			KPD_COLO_MUXMODE			KPD_COL5_WAKEUPEVENT	KPD_COL5_WAKEUPENABLE	KPD_COL5_OFFMODEPULLTYPESELECT	KPD_COL5_OFFMODEPULLUDENABLE	KPD_COL5_OFFMODEOUTVALUE	KPD_COL5_OFFMODEOUTENABLE	KPD_COL5_OFFMODEENABLE	KPD_COL5_INPUTENABLE	RESERVED			KPD_COL5_PULLTYPESELECT	KPD_COL5_PULLUDENABLE			KPD_COL5_MUXMODE

Bits	Field Name	Description	Type	Reset
31	KPD_COLO_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	KPD_COLO_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	KPD_COLO_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad kpd_col0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	KPD_COLO_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad kpd_col0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	KPD_COLO_OFFMODEOUTVALUE	OffMode mode output value for pad kpd_col0 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	KPD_COLO_OFFMODEOUTENABLE	OffMode mode output enable value for pad kpd_col0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	KPD_COLO_OFFMODEENABLE	OffMode mode override control for pad kpd_col0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
24	KPD_COL0_INPUTENABLE	Input enable value for pad kpd_col0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	KPD_COL0_PULLTYPESELECT	pullup/down selection for pad kpd_col0 0x0: pulldown selected 0x1: pullup selected	RW	0
19	KPD_COL0_PULLUDENABLE	pullup/down enable for pad kpd_col0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	KPD_COL0_MUXMODE	Functional multiplexing selection for pad kpd_col0 0x0: Select kpd_col0 0x1: Select kpd_col3 0x3: Select gpio_174 0x7: Select safe_mode	RW	0x7
15	KPD_COL5_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	KPD_COL5_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	KPD_COL5_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad kpd_col5 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	KPD_COL5_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad kpd_col5 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	KPD_COL5_OFFMODEOUTVALUE	OffMode mode output value for pad kpd_col5 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	KPD_COL5_OFFMODEOUTENABLE	OffMode mode output enable value for pad kpd_col5. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	KPD_COL5_OFFMODEENABLE	OffMode mode override control for pad kpd_col5 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	KPD_COL5_INPUTENABLE	Input enable value for pad kpd_col5 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0



Bits	Field Name	Description	Type	Reset
4	KPD_COL5_PULLTYPESELECT	pullup/down selection for pad kpd_col5 0x0: pulldown selected 0x1: pullup selected	RW	0
3	KPD_COL5_PULLUDENABLE	pullup/down enable for pad kpd_col5 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	KPD_COL5_MUXMODE	Functional multiplexing selection for pad kpd_col5 0x0: Select kpd_col5 0x1: Select kpd_col2 0x3: Select gpio_173 0x7: Select safe_mode	RW	0x7

**Table 18-483. Register Call Summary for Register CONTROL\_CORE\_PAD0\_KPD\_COL5\_PAD1\_KPD\_COLO**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-484. CONTROL\_CORE\_PAD0\_KPD\_COL1\_PAD1\_KPD\_COL2**

<b>Address Offset</b>	0x0000 0184	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads kpd_col1 and kpd_col2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPD_COL2_WAKEUPEVENT	KPD_COL2_WAKEUPENABLE	KPD_COL2_OFFMODEPULLTYPESELECT	KPD_COL2_OFFMODEPULLUDENABLE	KPD_COL2_OFFMODEOUTVALUE	KPD_COL2_OFFMODEOUTENABLE	KPD_COL2_OFFMODEENABLE	KPD_COL2_INPUTENABLE	RESERVED		KPD_COL2_PULLTYPESELECT	KPD_COL2_PULLUDENABLE		KPD_COL2_MUXMODE		KPD_COL1_WAKEUPEVENT	KPD_COL1_WAKEUPENABLE	KPD_COL1_OFFMODEPULLTYPESELECT	KPD_COL1_OFFMODEPULLUDENABLE	KPD_COL1_OFFMODEOUTVALUE	KPD_COL1_OFFMODEOUTENABLE	KPD_COL1_OFFMODEENABLE	KPD_COL1_INPUTENABLE	RESERVED		KPD_COL1_PULLTYPESELECT	KPD_COL1_PULLUDENABLE		KPD_COL1_MUXMODE			

Bits	Field Name	Description	Type	Reset
31	KPD_COL2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
30	KPD_COL2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	KPD_COL2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad kpd_col2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	KPD_COL2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad kpd_col2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	KPD_COL2_OFFMODEOUTVALUE	OffMode mode output value for pad kpd_col2 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	KPD_COL2_OFFMODEOUTENABLE	OffMode mode output enable value for pad kpd_col2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	KPD_COL2_OFFMODEENABLE	OffMode mode override control for pad kpd_col2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	KPD_COL2_INPUTENABLE	Input enable value for pad kpd_col2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	KPD_COL2_PULLTYPESELECT	pullup/down selection for pad kpd_col2 0x0: pulldown selected 0x1: pullup selected	RW	0
19	KPD_COL2_PULLUDENABLE	pullup/down enable for pad kpd_col2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	KPD_COL2_MUXMODE	Functional multiplexing selection for pad kpd_col2 0x0: Select kpd_col2 0x1: Select kpd_col5 0x3: Select gpio_1 0x7: Select safe_mode	RW	0x7
15	KPD_COL1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	KPD_COL1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
13	KPD_COL1_ OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad kpd_col1  0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	KPD_COL1_ OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad kpd_col1  0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	KPD_COL1_ OFFMODEOUTVALUE	OffMode mode output value for pad kpd_col1  0x0: Set value at 0 0x1: Set value at 1	RW	0
10	KPD_COL1_ OFFMODEOUTENABLE	OffMode mode output enable value for pad kpd_col1. This is an active low signal  0x0: Output enable 0x1: Output disable	RW	0
9	KPD_COL1_ OFFMODEENABLE	OffMode mode override control for pad kpd_col1  0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	KPD_COL1_ INPUTENABLE	Input enable value for pad kpd_col1  0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	KPD_COL1_ PULLTYPESELECT	pullup/down selection for pad kpd_col1  0x0: pulldown selected 0x1: pullup selected	RW	0
3	KPD_COL1_ PULLUDENABLE	pullup/down enable for pad kpd_col1  0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	KPD_COL1_ MUXMODE	Functional multiplexing selection for pad kpd_col1  0x0: Select kpd_col1 0x1: Select kpd_col4 0x3: Select gpio_0 0x7: Select safe_mode	RW	0x7

**Table 18-485. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_KPD\_COL1\_PAD1\_KPD\_COL2**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-486. CONTROL\_CORE\_PAD0\_KPD\_ROW3\_PAD1\_KPD\_ROW4**

<b>Address Offset</b>	0x0000 0188		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads kpd_row3 and kpd_row4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPD_ROW4_WAKEUPEVENT	KPD_ROW4_WAKEUPEVENT	RESERVED				KPD_ROW4_WAKEUPEVENT	RESERVED	KPD_ROW4_WAKEUPEVENT	KPD_ROW4_WAKEUPEVENT	KPD_ROW4_WAKEUPEVENT	KPD_ROW3_WAKEUPEVENT	KPD_ROW3_WAKEUPENABLE	RESERVED				KPD_ROW3_INPUENABLE	RESERVED	KPD_ROW3_PULTYPESELECT	KPD_ROW3_PULLUDENABE	KPD_ROW3_MUXMODE										

Bits	Field Name	Description	Type	Reset
31	KPD_ROW4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	KPD_ROW4_WAKEUPEVENT	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	KPD_ROW4_WAKEUPEVENT	Input enable value for pad kpd_row4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	KPD_ROW4_WAKEUPEVENT	pullup/down selection for pad kpd_row4 0x0: pulldown selected 0x1: pullup selected	RW	1
19	KPD_ROW4_WAKEUPEVENT	pullup/down enable for pad kpd_row4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	KPD_ROW4_WAKEUPEVENT	Functional multiplexing selection for pad kpd_row4 0x0: Select kpd_row4 0x1: Select kpd_row1 0x3: Select gpio_176 0x7: Select safe_mode	RW	0x7
15	KPD_ROW3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
14	KPD_ROW3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	KPD_ROW3_INPUTENABLE	Input enable value for pad kpd_row3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	KPD_ROW3_PULLTYPESELECTION	pullup/down selection for pad kpd_row3 0x0: pulldown selected 0x1: pullup selected	RW	1
3	KPD_ROW3_PULLUDENABLE	pullup/down enable for pad kpd_row3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	KPD_ROW3_MUXMODE	Functional multiplexing selection for pad kpd_row3 0x0: Select kpd_row3 0x1: Select kpd_row0 0x3: Select gpio_175 0x7: Select safe_mode	RW	0x7

**Table 18-487. Register Call Summary for Register CONTROL\_CORE\_PAD0\_KPD\_ROW3\_PAD1\_KPD\_ROW4**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-488. CONTROL\_CORE\_PAD0\_KPD\_ROW5\_PAD1\_KPD\_ROW0**

<b>Address Offset</b>	0x0000 018C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads kpd_row5 and kpd_row0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPD_ROW0_WAKEUPEVENT	KPD_ROW0_WAKEUPENABLE	RESERVED						KPD_ROW0_INPUTENABLE	RESERVED		KPD_ROW0_PULLTYPESELECT	KPD_ROW0_PULLUDENABLE	KPD_ROW0_MUXMODE			KPD_ROW5_WAKEUPEVENT	KPD_ROW5_WAKEUPENABLE	RESERVED						KPD_ROW5_INPUTENABLE	RESERVED		KPD_ROW5_PULLTYPESELECT	KPD_ROW5_PULLUDENABLE	KPD_ROW5_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	KPD_ROW0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	KPD_ROW0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	KPD_ROW0_INPUTENABLE	Input enable value for pad kpd_row0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	KPD_ROW0_PULLTYPESELECTION	pullup/down selection for pad kpd_row0 0x0: pulldown selected 0x1: pullup selected	RW	1
19	KPD_ROW0_PULLUDENABLE	pullup/down enable for pad kpd_row0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	KPD_ROW0_MUXMODE	Functional multiplexing selection for pad kpd_row0 0x0: Select kpd_row0 0x1: Select kpd_row3 0x3: Select gpio_178 0x7: Select safe_mode	RW	0x7
15	KPD_ROW5_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	KPD_ROW5_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	KPD_ROW5_INPUTENABLE	Input enable value for pad kpd_row5 0x0: Input buffer of I/O cell is disabled 0x1: Input buffer of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	KPD_ROW5_PULLTYPESELECTION	pullup/down selection for pad kpd_row5 0x0: pulldown selected 0x1: pullup selected	RW	1
3	KPD_ROW5_PULLUDENABLE	pullup/down enable for pad kpd_row5 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
2:0	KPD_ROW5_MUXMODE	Functional multiplexing selection for pad kpd_row5 0x0: Select kpd_row5 0x1: Select kpd_row2 0x3: Select gpio_177 0x7: Select safe_mode	RW	0x7

**Table 18-489. Register Call Summary for Register CONTROL\_CORE\_PAD0\_KPD\_ROW5\_PAD1\_KPD\_ROW0**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-490. CONTROL\_CORE\_PAD0\_KPD\_ROW1\_PAD1\_KPD\_ROW2**

<b>Address Offset</b>	0x0000 0190	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads kpd_row1 and kpd_row2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPD_ROW2_WAKEUPEVENT	KPD_ROW2_WAKEUPENABLE	RESERVED						KPD_ROW2_INPUTENABLE	RESERVED		KPD_ROW2_PULLTYPESELECT	KPD_ROW2_PULLUDENABE	KPD_ROW2_MUXMODE			KPD_ROW1_WAKEUPEVENT	KPD_ROW1_WAKEUPENABLE	RESERVED						KPD_ROW1_INPUTENABLE	RESERVED		KPD_ROW1_PULLTYPESELECT	KPD_ROW1_PULLUDENABE	KPD_ROW1_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	KPD_ROW2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	KPD_ROW2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	KPD_ROW2_INPUTENABLE	Input enable value for pad kpd_row2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	KPD_ROW2_PULLTYPESELECTION	pullup/down selection for pad kpd_row2 0x0: pulldown selected 0x1: pullup selected	RW	1

Bits	Field Name	Description	Type	Reset
19	KPD_ROW2_PULLUDENABLE	pullup/down enable for pad kpd_row2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	KPD_ROW2_MUXMODE	Functional multiplexing selection for pad kpd_row2 0x0: Select kpd_row2 0x1: Select kpd_row5 0x3: Select gpio_3 0x7: Select safe_mode	RW	0x7
15	KPD_ROW1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	KPD_ROW1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	KPD_ROW1_INPUTENABLE	Input enable value for pad kpd_row1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	KPD_ROW1_PULLTYPESELEC T	pullup/down selection for pad kpd_row1 0x0: pulldown selected 0x1: pullup selected	RW	1
3	KPD_ROW1_PULLUDENABLE	pullup/down enable for pad kpd_row1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	KPD_ROW1_MUXMODE	Functional multiplexing selection for pad kpd_row1 0x0: Select kpd_row1 0x1: Select kpd_row4 0x3: Select gpio_2 0x7: Select safe_mode	RW	0x7

**Table 18-491. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_KPD\_ROW1\_PAD1\_KPD\_ROW2**

## Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)



**Table 18-492. CONTROL\_CORE\_PAD0\_USBA0\_OTG\_CE\_PAD1\_USBA0\_OTG\_DP**

<b>Address Offset</b>	0x0000 0194		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads usba0_otg_ce and usba0_otg_dp Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA0_OTG_DP_WAKEUPEVENT	USBA0_OTG_DP_WAKEUPENABLE										USBA0_OTG_DP_PULLTYPESELECT	USBA0_OTG_DP_PULLUDENABE	USBA0_OTG_DP_MUXMODE	RESERVED											USBA0_OTG_CE_PULLTYPESELECT	USBA0_OTG_CE_PULLUDENABE	RESERVED				

Bits	Field Name	Description	Type	Reset
31	USBA0_OTG_DP_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x1: A wake-up event occurred Read 0x0: No wake-up event occurred	R	0x0
30	USBA0_OTG_DP_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection on low level 0x1: wake-up detection on high level	RW	0x0
29:21	RESERVED		R	0x000
20	USBA0_OTG_DP_PULLTYPESELECT	pullup/down selection for pad usba0_otg_dp 0x0: pulldown selected 0x1: pullup selected	RW	0
19	USBA0_OTG_DP_PULLUDENABE	pullup/down enable for pad usba0_otg_dp 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	USBA0_OTG_DP_MUXMODE	Functional multiplexing selection for pad usba0_otg_dp 0x0: Select usba0_otg_dp 0x1: Select uart3_rx_irrx 0x2: Select uart2_rx 0x7: Select safe_mode	RW	0x7
15:5	RESERVED		R	0x000
4	USBA0_OTG_CE_PULLTYPESELECT	pullup/down selection for pad usba0_otg_ce 0x0: pulldown selected 0x1: pullup selected	RW	0
3	USBA0_OTG_CE_PULLUDENABE	pullup/down enable for pad usba0_otg_ce 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	RESERVED		R	0x0

**Table 18-493. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBA0\_OTG\_CE\_PAD1\_USBA0\_OTG\_DP**

Control Module Functional Description
• <a href="#">Pad Multiplexing Register Fields: [0] [1]</a>
Control Module Register Manual
• <a href="#">SYSCTRL_PADCONF_CORE Register Summary: [2]</a>

**Table 18-494. CONTROL\_CORE\_PAD0\_USBA0\_OTG\_DM\_PAD1\_FREF\_CLK1\_OUT**

<b>Address Offset</b>	0x0000 0198
<b>Physical Address</b>	See <a href="#">Table 18-315</a>
<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads usba0_otg_dm and fref_clk1_out Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREF_CLK1_OUT_WAKEUPEVENT	FREF_CLK1_OUT_WAKEUPENABLE	FREF_CLK1_OUT_OFFMODEPULLTYPESELECT	FREF_CLK1_OUT_OFFMODEPULLUDENENABLE	FREF_CLK1_OUT_OFFMODEOUTVALUE	FREF_CLK1_OUT_OFFMODEOUTENENABLE	FREF_CLK1_OUT_OFFMODEENENABLE	FREF_CLK1_OUT_INPUTENABLE	RESERVED			FREF_CLK1_OUT_PULLTYPESELECT	FREF_CLK1_OUT_PULLUDENENABLE		FREF_CLK1_OUT_MUXMODE																	
																RESERVED											USBA0_OTG_DM_PULLTYPESELECT	USBA0_OTG_DM_PULLUDENENABLE			USBA0_OTG_DM_MUXMODE

Bits	Field Name	Description	Type	Reset
31	FREF_CLK1_OUT_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	FREF_CLK1_OUT_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	FREF_CLK1_OUT_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad fref_clk1_out 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	FREF_CLK1_OUT_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad fref_clk1_out 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	FREF_CLK1_OUT_OFFMODEOUTVALUE	OffMode mode output value for pad fref_clk1_out 0x0: Set value at 0 0x1: Set value at 1	RW	0

Bits	Field Name	Description	Type	Reset
26	FREF_CLK1_OUT_OFFMODEOUTENABLE	OffMode mode output enable value for pad fref_clk1_out. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	FREF_CLK1_OUT_OFFMODEENABLE	OffMode mode override control for pad fref_clk1_out 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	FREF_CLK1_OUT_INPUTENABLE	Input enable value for pad fref_clk1_out 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	FREF_CLK1_OUT_PULLTYPESSELECT	pullup/down selection for pad fref_clk1_out 0x0: pulldown selected 0x1: pullup selected	RW	0
19	FREF_CLK1_OUT_PULLUDENABLE	pullup/down enable for pad fref_clk1_out 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	FREF_CLK1_OUT_MUXMODE	Functional multiplexing selection for pad fref_clk1_out 0x0: Select fref_clk1_out 0x3: Select gpio_181 0x7: Select safe_mode	RW	0x7
15:5	RESERVED		R	0x000
4	USBA0_OTG_DM_PULLTYPESELECT	pullup/down selection for pad usba0_otg_dm 0x0: pulldown selected 0x1: pullup selected	RW	0
3	USBA0_OTG_DM_PULLUDENABLE	pullup/down enable for pad usba0_otg_dm 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	USBA0_OTG_DM_MUXMODE	Functional multiplexing selection for pad usba0_otg_dm 0x0: Select usba0_otg_dm 0x1: Select uart3_tx_irtx 0x2: Select uart2_tx 0x7: Select safe_mode	RW	0x7

**Table 18-495. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_USBA0\_OTG\_DM\_PAD1\_FREF\_CLK1\_OUT**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-496. CONTROL\_CORE\_PAD0\_FREF\_CLK2\_OUT\_PAD1\_SYS\_NIRQ1**

<b>Address Offset</b>	0x0000 019C		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads fref_clk2_out and sys_nirq1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_NIRQ1_WAKEUPEVENT	SYS_NIRQ1_WAKEUPENABLE	RESERVED					SYS_NIRQ1_INPUTENABLE	RESERVED	SYS_NIRQ1_PULLTYPESELECT	SYS_NIRQ1_PULLUDENABLE	SYS_NIRQ1_MUXMODE					FREF_CLK2_OUT_WAKEUPEVENT	FREF_CLK2_OUT_WAKEUPENABLE	FREF_CLK2_OUT_OFFMODEPULLTYPESELECT	FREF_CLK2_OUT_OFFMODEPULLUDENABLE	FREF_CLK2_OUT_OFFMODEOUTVALUE	FREF_CLK2_OUT_OFFMODEOUTENABLE	FREF_CLK2_OUT_OFFMODEENABLE	FREF_CLK2_OUT_INPUTENABLE	RESERVED	FREF_CLK2_OUT_PULLTYPESELECT	FREF_CLK2_OUT_PULLUDENABLE	FREF_CLK2_OUT_MUXMODE				

Bits	Field Name	Description	Type	Reset
31	SYS_NIRQ1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SYS_NIRQ1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	SYS_NIRQ1_INPUTENABLE	Input enable value for pad sys_nirq1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SYS_NIRQ1_PULLTYPESELECT	pullup/down selection for pad sys_nirq1 0x0: pulldown selected 0x1: pullup selected	RW	1
19	SYS_NIRQ1_PULLUDENABLE	pullup/down enable for pad sys_nirq1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SYS_NIRQ1_MUXMODE	Functional multiplexing selection for pad sys_nirq1 0x0: Select sys_nirq1 0x7: Select safe_mode	RW	0x7
15	FREF_CLK2_OUT_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
14	FREF_CLK2_OUT_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	FREF_CLK2_OUT_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad <code>fref_clk2_out</code> 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	FREF_CLK2_OUT_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad <code>fref_clk2_out</code> 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	FREF_CLK2_OUT_OFFMODEOUTVALUE	OffMode mode output value for pad <code>fref_clk2_out</code> 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	FREF_CLK2_OUT_OFFMODEOUTENABLE	OffMode mode output enable value for pad <code>fref_clk2_out</code> . This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	FREF_CLK2_OUT_OFFMODEENABLE	OffMode mode override control for pad <code>fref_clk2_out</code> 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	FREF_CLK2_OUT_INPUTENABLE	Input enable value for pad <code>fref_clk2_out</code> 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	FREF_CLK2_OUT_PULLTYPESELECT	pullup/down selection for pad <code>fref_clk2_out</code> 0x0: pulldown selected 0x1: pullup selected	RW	0
3	FREF_CLK2_OUT_PULLUDENABLE	pullup/down enable for pad <code>fref_clk2_out</code> 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	FREF_CLK2_OUT_MUXMODE	Functional multiplexing selection for pad <code>fref_clk2_out</code> 0x0: Select <code>fref_clk2_out</code> 0x3: Select <code>gpio_182</code> 0x7: Select <code>safe_mode</code>	RW	0x7

**Table 18-497. Register Call Summary for Register CONTROL\_CORE\_PAD0\_FREF\_CLK2\_OUT\_PAD1\_SYS\_NIRQ1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-498. CONTROL\_CORE\_PAD0\_SYS\_NIRQ2\_PAD1\_SYS\_BOOT0**

<b>Address Offset</b>	0x0000 01A0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads sys_nirq2 and sys_boot0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_BOOT0_WAKEUPEVENT	SYS_BOOT0_WAKEUPENABLE	SYS_BOOT0_OFFMODEPULLTYPESELECT	SYS_BOOT0_OFFMODEPULLUDENABLE	SYS_BOOT0_OFFMODEOUTVALUE	SYS_BOOT0_OFFMODEOUTENABLE	SYS_BOOT0_OFFMODEENABLE	SYS_BOOT0_INPUTENABLE	RESERVED			SYS_BOOT0_PULLTYPESELECT	SYS_BOOT0_PULLUDENABLE		SYS_BOOT0_MUXMODE		SYS_NIRQ2_WAKEUPEVENT	SYS_NIRQ2_WAKEUPENABLE	SYS_NIRQ2_OFFMODEPULLTYPESELECT	SYS_NIRQ2_OFFMODEPULLUDENABLE	SYS_NIRQ2_OFFMODEOUTVALUE	SYS_NIRQ2_OFFMODEOUTENABLE	SYS_NIRQ2_OFFMODEENABLE	SYS_NIRQ2_INPUTENABLE	RESERVED			SYS_NIRQ2_PULLTYPESELECT	SYS_NIRQ2_PULLUDENABLE		SYS_NIRQ2_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	SYS_BOOT0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SYS_BOOT0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SYS_BOOT0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sys_boot0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	SYS_BOOT0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sys_boot0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SYS_BOOT0_OFFMODEOUTVALUE	OffMode mode output value for pad sys_boot0 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SYS_BOOT0_OFFMODEOUTENABLE	OffMode mode output enable value for pad sys_boot0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SYS_BOOT0_OFFMODEENABLE	OffMode mode override control for pad sys_boot0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	SYS_BOOT0_INPUTENABLE	Input enable value for pad sys_boot0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1

Bits	Field Name	Description	Type	Reset
23:21	RESERVED		R	0x0
20	SYS_BOOT0_PULLTYPESELECTION	pullup/down selection for pad sys_boot0 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SYS_BOOT0_PULLUDENABLE	pullup/down enable for pad sys_boot0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SYS_BOOT0_MUXMODE	Functional multiplexing selection for pad sys_boot0 0x0: Select sys_boot0 0x3: Select gpio_184 0x7: Select safe_mode	RW	0x0
15	SYS_NIRQ2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SYS_NIRQ2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	SYS_NIRQ2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sys_nirq2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SYS_NIRQ2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sys_nirq2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SYS_NIRQ2_OFFMODEOUTVALUE	OffMode mode output value for pad sys_nirq2 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SYS_NIRQ2_OFFMODEOUTENABLE	OffMode mode output enable value for pad sys_nirq2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	SYS_NIRQ2_OFFMODEENABLE	OffMode mode override control for pad sys_nirq2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SYS_NIRQ2_INPUTENABLE	Input enable value for pad sys_nirq2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SYS_NIRQ2_PULLTYPESELECTION	pullup/down selection for pad sys_nirq2 0x0: pulldown selected 0x1: pullup selected	RW	1
3	SYS_NIRQ2_PULLUDENABLE	pullup/down enable for pad sys_nirq2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SYS_NIRQ2_MUXMODE	Functional multiplexing selection for pad sys_nirq2 0x0: Select sys_nirq2 0x3: Select gpio_183 0x7: Select safe_mode	RW	0x7

**Table 18-499. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_SYS\_NIRQ2\_PAD1\_SYS\_BOOT0**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-500. CONTROL\_CORE\_PAD0\_SYS\_BOOT1\_PAD1\_SYS\_BOOT2**

<b>Address Offset</b>	0x0000 01A4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads sys_boot1 and sys_boot2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_BOOT2_WAKEUPEVENT	SYS_BOOT2_WAKEUPENABLE	SYS_BOOT2_OFFMODEPULLTYPESELECT	SYS_BOOT2_OFFMODEPULLUDENABLE	SYS_BOOT2_OFFMODEOUTVALUE	SYS_BOOT2_OFFMODEOUTENABLE	SYS_BOOT2_OFFMODEENABLE	SYS_BOOT2_INPUENABLE	RESERVED			SYS_BOOT2_PULLTYPESELECT	SYS_BOOT2_PULLUDENABLE		SYS_BOOT2_MUXMODE		SYS_BOOT1_WAKEUPEVENT	SYS_BOOT1_WAKEUPENABLE	SYS_BOOT1_OFFMODEPULLTYPESELECT	SYS_BOOT1_OFFMODEPULLUDENABLE	SYS_BOOT1_OFFMODEOUTVALUE	SYS_BOOT1_OFFMODEOUTENABLE	SYS_BOOT1_OFFMODEENABLE	SYS_BOOT1_INPUENABLE	RESERVED			SYS_BOOT1_PULLTYPESELECT	SYS_BOOT1_PULLUDENABLE		SYS_BOOT1_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	SYS_BOOT2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SYS_BOOT2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SYS_BOOT2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sys_boot2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	SYS_BOOT2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sys_boot2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SYS_BOOT2_OFFMODEOUTVALUE	OffMode mode output value for pad sys_boot2 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SYS_BOOT2_OFFMODEOUTENABLE	OffMode mode output enable value for pad sys_boot2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0



Bits	Field Name	Description	Type	Reset
25	SYS_BOOT2_OFFMODEENABLE	OffMode mode override control for pad sys_boot2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	SYS_BOOT2_INPUTENABLE	Input enable value for pad sys_boot2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SYS_BOOT2_PULLTYPESELECTION	pullup/down selection for pad sys_boot2 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SYS_BOOT2_PULLUDENABLE	pullup/down enable for pad sys_boot2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SYS_BOOT2_MUXMODE	Functional multiplexing selection for pad sys_boot2 0x0: Select sys_boot2 0x3: Select gpio_186 0x7: Select safe_mode	RW	0x0
15	SYS_BOOT1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SYS_BOOT1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	SYS_BOOT1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sys_boot1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SYS_BOOT1_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sys_boot1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SYS_BOOT1_OFFMODEOUTVALUE	OffMode mode output value for pad sys_boot1 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SYS_BOOT1_OFFMODEOUTENABLE	OffMode mode output enable value for pad sys_boot1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	SYS_BOOT1_OFFMODEENABLE	OffMode mode override control for pad sys_boot1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SYS_BOOT1_INPUTENABLE	Input enable value for pad sys_boot1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SYS_BOOT1_PULLTYPESELECTION	pullup/down selection for pad sys_boot1 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	SYS_BOOT1_PULLUDENABLE	pullup/down enable for pad sys_boot1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SYS_BOOT1_MUXMODE	Functional multiplexing selection for pad sys_boot1 0x0: Select sys_boot1 0x3: Select gpio_185 0x7: Select safe_mode	RW	0x0

**Table 18-501. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_SYS\_BOOT1\_PAD1\_SYS\_BOOT2**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-502. CONTROL\_CORE\_PAD0\_SYS\_BOOT3\_PAD1\_SYS\_BOOT4**

<b>Address Offset</b>	0x0000 01A8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads sys_boot3 and sys_boot4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_BOOT4_WAKEUPEVENT	SYS_BOOT4_WAKEUPENABLE	SYS_BOOT4_OFFMODEPULLTYPESELECT	SYS_BOOT4_OFFMODEPULLUDENABLE	SYS_BOOT4_OFFMODEOUTVALUE	SYS_BOOT4_OFFMODEOUTENABLE	SYS_BOOT4_OFFMODEENABLE	SYS_BOOT4_INPUTENABLE	RESERVED			SYS_BOOT4_PULLTYPESELECT	SYS_BOOT4_PULLUDENABLE	SYS_BOOT4_MUXMODE			SYS_BOOT3_WAKEUPEVENT	SYS_BOOT3_WAKEUPENABLE	SYS_BOOT3_OFFMODEPULLTYPESELECT	SYS_BOOT3_OFFMODEPULLUDENABLE	SYS_BOOT3_OFFMODEOUTVALUE	SYS_BOOT3_OFFMODEOUTENABLE	SYS_BOOT3_OFFMODEENABLE	SYS_BOOT3_INPUTENABLE	RESERVED			SYS_BOOT3_PULLTYPESELECT	SYS_BOOT3_PULLUDENABLE	SYS_BOOT3_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	SYS_BOOT4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SYS_BOOT4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	SYS_BOOT4_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sys_boot4 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
28	SYS_BOOT4_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sys_boot4 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	SYS_BOOT4_OFFMODEOUTVALUE	OffMode mode output value for pad sys_boot4 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	SYS_BOOT4_OFFMODEOUTENABLE	OffMode mode output enable value for pad sys_boot4. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	SYS_BOOT4_OFFMODEENABLE	OffMode mode override control for pad sys_boot4 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	SYS_BOOT4_INPUTENABLE	Input enable value for pad sys_boot4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SYS_BOOT4_PULLTYPESELECT	pullup/down selection for pad sys_boot4 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SYS_BOOT4_PULLUDENABLE	pullup/down enable for pad sys_boot4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SYS_BOOT4_MUXMODE	Functional multiplexing selection for pad sys_boot4 0x0: Select sys_boot4 0x3: Select gpio_188 0x7: Select safe_mode	RW	0x0
15	SYS_BOOT3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SYS_BOOT3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	SYS_BOOT3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sys_boot3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	SYS_BOOT3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sys_boot3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SYS_BOOT3_OFFMODEOUTVALUE	OffMode mode output value for pad sys_boot3 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SYS_BOOT3_OFFMODEOUTENABLE	OffMode mode output enable value for pad sys_boot3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
9	SYS_BOOT3_OFFMODEENABLE	OffMode mode override control for pad sys_boot3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SYS_BOOT3_INPUTENABLE	Input enable value for pad sys_boot3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SYS_BOOT3_PULLTYPESELECT	pullup/down selection for pad sys_boot3 0x0: pulldown selected 0x1: pullup selected	RW	0
3	SYS_BOOT3_PULLUDENABLE	pullup/down enable for pad sys_boot3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SYS_BOOT3_MUXMODE	Functional multiplexing selection for pad sys_boot3 0x0: Select sys_boot3 0x3: Select gpio_187 0x7: Select safe_mode	RW	0x0

**Table 18-503. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_SYS\_BOOT3\_PAD1\_SYS\_BOOT4**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-504. CONTROL\_CORE\_PAD0\_SYS\_BOOT5\_PAD1\_DPM\_EMU0**

<b>Address Offset</b>	0x0000 01AC	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads sys_boot5 and dpm_emu0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU0_WAKEUPEVENT	DPM_EMU0_WAKEUPENABLE	DPM_EMU0_OFFMODEPULLTYPESELECT	DPM_EMU0_OFFMODEPULLUDENABLE	DPM_EMU0_OFFMODEOUTVALUE	DPM_EMU0_OFFMODEOUTENABLE	DPM_EMU0_OFFMODEENABLE	DPM_EMU0_INPUTENABLE	RESERVED			DPM_EMU0_PULLTYPESELECT	DPM_EMU0_PULLUDENABLE			DPM_EMU0_MUXMODE																
																SYS_BOOT5_WAKEUPEVENT	SYS_BOOT5_WAKEUPENABLE	SYS_BOOT5_OFFMODEPULLTYPESELECT	SYS_BOOT5_OFFMODEPULLUDENABLE	SYS_BOOT5_OFFMODEOUTVALUE	SYS_BOOT5_OFFMODEOUTENABLE	SYS_BOOT5_OFFMODEENABLE	SYS_BOOT5_INPUTENABLE	RESERVED			SYS_BOOT5_PULLTYPESELECT	SYS_BOOT5_PULLUDENABLE			SYS_BOOT5_MUXMODE

Bits	Field Name	Description	Type	Reset
31	DPM_EMU0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	DPM_EMU0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	DPM_EMU0_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu0 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU0_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu0 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU0_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu0 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU0_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu0. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	DPM_EMU0_OFFMODEENABLE	OffMode mode override control for pad dpm_emu0 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	DPM_EMU0_INPUTENABLE	Input enable value for pad dpm_emu0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	DPM_EMU0_PULLTYPESELECTION	pullup/down selection for pad dpm_emu0 0x0: pulldown selected 0x1: pullup selected	RW	1
19	DPM_EMU0_PULLUDENABLE	pullup/down enable for pad dpm_emu0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	DPM_EMU0_MUXMODE	Functional multiplexing selection for pad dpm_emu0 0x0: Select dpm_emu0 0x3: Select gpio_11 0x6: Select hw_dbg0 0x7: Select safe_mode	RW	0x0
15	SYS_BOOT5_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SYS_BOOT5_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	SYS_BOOT5_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad sys_boot5 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	SYS_BOOT5_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad sys_boot5 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	SYS_BOOT5_OFFMODEOUTVALUE	OffMode mode output value for pad sys_boot5 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	SYS_BOOT5_OFFMODEOUTENABLE	OffMode mode output enable value for pad sys_boot5. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	SYS_BOOT5_OFFMODEENABLE	OffMode mode override control for pad sys_boot5 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	SYS_BOOT5_INPUTENABLE	Input enable value for pad sys_boot5 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SYS_BOOT5_PULLTYPESELECTION	pullup/down selection for pad sys_boot5 0x0: pulldown selected 0x1: pullup selected	RW	0
3	SYS_BOOT5_PULLUDENABLE	pullup/down enable for pad sys_boot5 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	SYS_BOOT5_MUXMODE	Functional multiplexing selection for pad sys_boot5 0x0: Select sys_boot5 0x3: Select gpio_189 0x7: Select safe_mode	RW	0x0

**Table 18-505. Register Call Summary for Register CONTROL\_CORE\_PAD0\_SYS\_BOOT5\_PAD1\_DPM\_EMU0**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[3\]](#)

**Table 18-506. CONTROL\_CORE\_PAD0\_DPM\_EMU1\_PAD1\_DPM\_EMU2**

<b>Address Offset</b>	0x0000 01B0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads dpm_emu1 and dpm_emu2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU2_WAKEUPEVENT	DPM_EMU2_WAKEUPENABLE	DPM_EMU2_OFFMODEPULLTYPESELECT	DPM_EMU2_OFFMODEPULLUDENABLE	DPM_EMU2_OFFMODEOUTVALUE	DPM_EMU2_OFFMODEOUTENABLE	DPM_EMU2_OFFMODEEENABLE	DPM_EMU2_INPUTENABLE	RESERVED			DPM_EMU2_PULLTYPESELECT	DPM_EMU2_PULLUDENABLE		DPM_EMU2_MUXMODE		DPM_EMU1_WAKEUPEVENT	DPM_EMU1_WAKEUPENABLE	DPM_EMU1_OFFMODEPULLTYPESELECT	DPM_EMU1_OFFMODEPULLUDENABLE	DPM_EMU1_OFFMODEOUTVALUE	DPM_EMU1_OFFMODEOUTENABLE	DPM_EMU1_OFFMODEEENABLE	DPM_EMU1_INPUTENABLE	RESERVED			DPM_EMU1_PULLTYPESELECT	DPM_EMU1_PULLUDENABLE		DPM_EMU1_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	DPM_EMU2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	DPM_EMU2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	DPM_EMU2_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu2 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU2_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu2 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU2_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu2 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU2_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu2. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	DPM_EMU2_OFFMODEEENABLE	OffMode mode override control for pad dpm_emu2 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	DPM_EMU2_INPUTENABLE	Input enable value for pad dpm_emu2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	DPM_EMU2_PULLTYPESELECT	pullup/down selection for pad dpm_emu2 0x0: pulldown selected 0x1: pullup selected	RW	0
19	DPM_EMU2_PULLUDENABLE	pullup/down enable for pad dpm_emu2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
18:16	DPM_EMU2_MUXMODE	Functional multiplexing selection for pad dpm_emu2 0x0: Select dpm_emu2 0x1: Select usba0_ulpihy_clk 0x3: Select gpio_13 0x5: Select disp2_fid 0x6: Select hw_dbg2 0x7: Select safe_mode	RW	0x7
15	DPM_EMU1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	DPM_EMU1_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu1 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	DPM_EMU1_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu1 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU1_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu1 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU1_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu1. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU1_OFFMODEENABLE	OffMode mode override control for pad dpm_emu1 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	DPM_EMU1_INPUTENABLE	Input enable value for pad dpm_emu1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	DPM_EMU1_PULLTYPESELECT	pullup/down selection for pad dpm_emu1 0x0: pulldown selected 0x1: pullup selected	RW	1
3	DPM_EMU1_PULLUDENABLE	pullup/down enable for pad dpm_emu1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU1_MUXMODE	Functional multiplexing selection for pad dpm_emu1 0x0: Select dpm_emu1 0x3: Select gpio_12 0x6: Select hw_dbg1 0x7: Select safe_mode	RW	0x0



**Table 18-507. Register Call Summary for Register CONTROL\_CORE\_PAD0\_DPM\_EMU1\_PAD1\_DPM\_EMU2**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-508. CONTROL\_CORE\_PAD0\_DPM\_EMU3\_PAD1\_DPM\_EMU4**

<b>Address Offset</b>	0x0000 01B4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads dpm_emu3 and dpm_emu4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU4_WAKEUPEVENT	DPM_EMU4_WAKEUPENABLE	DPM_EMU4_OFFMODEPULLTYPESELECT	DPM_EMU4_OFFMODEPULLUDENABLE	DPM_EMU4_OFFMODEOUTVALUE	DPM_EMU4_OFFMODEOUTENABLE	DPM_EMU4_OFFMODEENABLE	DPM_EMU4_INPUTENABLE	RESERVED			DPM_EMU4_PULLTYPESELECT	DPM_EMU4_PULLUDENABLE		DPM_EMU4_MUXMODE		DPM_EMU3_WAKEUPEVENT	DPM_EMU3_WAKEUPENABLE	DPM_EMU3_OFFMODEPULLTYPESELECT	DPM_EMU3_OFFMODEPULLUDENABLE	DPM_EMU3_OFFMODEOUTVALUE	DPM_EMU3_OFFMODEOUTENABLE	DPM_EMU3_OFFMODEENABLE	DPM_EMU3_INPUTENABLE	RESERVED			DPM_EMU3_PULLTYPESELECT	DPM_EMU3_PULLUDENABLE		DPM_EMU3_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	DPM_EMU4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	DPM_EMU4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	DPM_EMU4_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu4 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU4_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu4 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU4_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu4 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU4_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu4. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0

Bits	Field Name	Description	Type	Reset
25	DPM_EMU4_OFFMODEENABLE	OffMode mode override control for pad dpm_emu4 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	DPM_EMU4_INPUTENABLE	Input enable value for pad dpm_emu4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	DPM_EMU4_PULLTYPESELECT	pullup/down selection for pad dpm_emu4 0x0: pulldown selected 0x1: pullup selected	RW	0
19	DPM_EMU4_PULLUDENABLE	pullup/down enable for pad dpm_emu4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	DPM_EMU4_MUXMODE	Functional multiplexing selection for pad dpm_emu4 0x0: Select dpm_emu4 0x1: Select usba0_ulpiphy_dir 0x3: Select gpio_15 0x4: Select rfb_data9 0x5: Select disp2_data9 0x6: Select hw_dbg4 0x7: Select safe_mode	RW	0x7
15	DPM_EMU3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	DPM_EMU3_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu3 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	DPM_EMU3_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu3 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU3_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu3 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU3_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu3. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU3_OFFMODEENABLE	OffMode mode override control for pad dpm_emu3 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	DPM_EMU3_INPUTENABLE	Input enable value for pad dpm_emu3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4	DPM_EMU3_PULLTYPESELECTION	pullup/down selection for pad dpm_emu3 0x0: pulldown selected 0x1: pullup selected	RW	0
3	DPM_EMU3_PULLUDENABLE	pullup/down enable for pad dpm_emu3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU3_MUXMODE	Functional multiplexing selection for pad dpm_emu3 0x0: Select dpm_emu3 0x1: Select usba0_ulpiphy_stp 0x3: Select gpio_14 0x4: Select rfb_data10 0x5: Select dispc2_data10 0x6: Select hw_dbg3 0x7: Select safe_mode	RW	0x7

**Table 18-509. Register Call Summary for Register CONTROL\_CORE\_PAD0\_DPM\_EMU3\_PAD1\_DPM\_EMU4**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-510. CONTROL\_CORE\_PAD0\_DPM\_EMU5\_PAD1\_DPM\_EMU6**

<b>Address Offset</b>	0x0000 01B8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads dpm_emu5 and dpm_emu6 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU6_WAKEUPEVENT	DPM_EMU6_WAKEUPENABLE	DPM_EMU6_OFFMODEPULLTYPESELECT	DPM_EMU6_OFFMODEPULLUDENABLE	DPM_EMU6_OFFMODEOUTVALUE	DPM_EMU6_OFFMODEOUTENABLE	DPM_EMU6_OFFMODEENABLE	DPM_EMU6_INPUTENABLE	RESERVED		DPM_EMU6_PULLTYPESELECT	DPM_EMU6_PULLUDENABLE		DPM_EMU6_MUXMODE		DPM_EMU5_WAKEUPEVENT	DPM_EMU5_WAKEUPENABLE	DPM_EMU5_OFFMODEPULLTYPESELECT	DPM_EMU5_OFFMODEPULLUDENABLE	DPM_EMU5_OFFMODEOUTVALUE	DPM_EMU5_OFFMODEOUTENABLE	DPM_EMU5_OFFMODEENABLE	DPM_EMU5_INPUTENABLE	RESERVED		DPM_EMU5_PULLTYPESELECT	DPM_EMU5_PULLUDENABLE		DPM_EMU5_MUXMODE			

Bits	Field Name	Description	Type	Reset
31	DPM_EMU6_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
30	DPM_EMU6_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	DPM_EMU6_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu6 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU6_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu6 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU6_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu6 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU6_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu6. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	DPM_EMU6_OFFMODEENABLE	OffMode mode override control for pad dpm_emu6 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	DPM_EMU6_INPUTENABLE	Input enable value for pad dpm_emu6 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	DPM_EMU6_PULLTYPESELECTION	pullup/down selection for pad dpm_emu6 0x0: pulldown selected 0x1: pullup selected	RW	0
19	DPM_EMU6_PULLUDENABLE	pullup/down enable for pad dpm_emu6 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	DPM_EMU6_MUXMODE	Functional multiplexing selection for pad dpm_emu6 0x0: Select dpm_emu6 0x1: Select usba0_ulpiphys_dat0 0x2: Select uart3_tx_irtx 0x3: Select gpio_17 0x4: Select rfb_i_hsync0 0x5: Select disp2_data17 0x6: Select hw_dbg6 0x7: Select safe_mode	RW	0x7
15	DPM_EMU5_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU5_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	DPM_EMU5_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu5 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	DPM_EMU5_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu5 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU5_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu5 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU5_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu5. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU5_OFFMODEENABLE	OffMode mode override control for pad dpm_emu5 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	DPM_EMU5_INPUTENABLE	Input enable value for pad dpm_emu5 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	DPM_EMU5_PULLTYPESELECTION	pullup/down selection for pad dpm_emu5 0x0: pulldown selected 0x1: pullup selected	RW	0
3	DPM_EMU5_PULLUDENABLE	pullup/down enable for pad dpm_emu5 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU5_MUXMODE	Functional multiplexing selection for pad dpm_emu5 0x0: Select dpm_emu5 0x1: Select usba0_ulpipiphy_nxt 0x3: Select gpio_16 0x4: Select rfb_i_te_vsync0 0x5: Select dispc2_data16 0x6: Select hw_dbg5 0x7: Select safe_mode	RW	0x7

**Table 18-511. Register Call Summary for Register CONTROL\_CORE\_PAD0\_DPM\_EMU5\_PAD1\_DPM\_EMU6**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-512. CONTROL\_CORE\_PAD0\_DPM\_EMU7\_PAD1\_DPM\_EMU8**

<b>Address Offset</b>	0x0000 01BC		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads dpm_emu7 and dpm_emu8 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU8_WAKEUPEVENT	DPM_EMU8_WAKEUPENABLE	DPM_EMU8_OFFMODEPULLTYPESELECT	DPM_EMU8_OFFMODEPULLUDENABLE	DPM_EMU8_OFFMODEOUTVALUE	DPM_EMU8_OFFMODEOUTENABLE	DPM_EMU8_OFFMODEENABLE	DPM_EMU8_INPUTENABLE	RESERVED			DPM_EMU8_PULLTYPESELECT	DPM_EMU8_PULLUDENABLE		DPM_EMU8_MUXMODE		DPM_EMU7_WAKEUPEVENT	DPM_EMU7_WAKEUPENABLE	DPM_EMU7_OFFMODEPULLTYPESELECT	DPM_EMU7_OFFMODEPULLUDENABLE	DPM_EMU7_OFFMODEOUTVALUE	DPM_EMU7_OFFMODEOUTENABLE	DPM_EMU7_OFFMODEENABLE	DPM_EMU7_INPUTENABLE	RESERVED			DPM_EMU7_PULLTYPESELECT	DPM_EMU7_PULLUDENABLE		DPM_EMU7_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	DPM_EMU8_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	DPM_EMU8_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	DPM_EMU8_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu8 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU8_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu8 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU8_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu8 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU8_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu8. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	DPM_EMU8_OFFMODEENABLE	OffMode mode override control for pad dpm_emu8 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	DPM_EMU8_INPUTENABLE	Input enable value for pad dpm_emu8 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1

Bits	Field Name	Description	Type	Reset
23:21	RESERVED		R	0x0
20	DPM_EMU8_PULLTYPESELECTION	pullup/down selection for pad dpm_emu8 0x0: pulldown selected 0x1: pullup selected	RW	0
19	DPM_EMU8_PULLUDENABLE	pullup/down enable for pad dpm_emu8 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	DPM_EMU8_MUXMODE	Functional multiplexing selection for pad dpm_emu8 0x0: Select dpm_emu8 0x1: Select usba0_ulpiphy_dat2 0x2: Select uart3_rts_sd 0x3: Select gpio_19 0x4: Select rfbire 0x5: Select disp2_pclk 0x6: Select hw_dbg8 0x7: Select safe_mode	RW	0x7
15	DPM_EMU7_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU7_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	DPM_EMU7_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu7 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	DPM_EMU7_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu7 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU7_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu7 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU7_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu7. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU7_OFFMODEENABLE	OffMode mode override control for pad dpm_emu7 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	DPM_EMU7_INPUTENABLE	Input enable value for pad dpm_emu7 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	DPM_EMU7_PULLTYPESELECTION	pullup/down selection for pad dpm_emu7 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	DPM_EMU7_PULLUDENABLE	pullup/down enable for pad dpm_emu7 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU7_MUXMODE	Functional multiplexing selection for pad dpm_emu7 0x0: Select dpm_emu7 0x1: Select usba0_ulpiphy_dat1 0x2: Select uart3_rx_irrx 0x3: Select gpio_18 0x4: Select rfb_i_cs0 0x5: Select disp2_hsync 0x6: Select hw_dbg7 0x7: Select safe_mode	RW	0x7

**Table 18-513. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_DPM\_EMU7\_PAD1\_DPM\_EMU8**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-514. CONTROL\_CORE\_PAD0\_DPM\_EMU9\_PAD1\_DPM\_EMU10**

<b>Address Offset</b>	0x0000 01C0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads dpm_emu9 and dpm_emu10 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
DPM_EMU10_WAKEUPEVENT	DPM_EMU10_WAKEUPENABLE	DPM_EMU10_OFFMODEPULLTYPESELECT	DPM_EMU10_OFFMODEPULLUDENABLE	DPM_EMU10_OFFMODEOUTVALUE	DPM_EMU10_OFFMODEOUTENABLE	DPM_EMU10_OFFMODEEENABLE	DPM_EMU10_INPUTENABLE	RESERVED			DPM_EMU10_PULLTYPESELECT	DPM_EMU10_PULLUDENABLE			DPM_EMU10_MUXMODE			DPM_EMU9_WAKEUPEVENT	DPM_EMU9_WAKEUPENABLE	DPM_EMU9_OFFMODEPULLTYPESELECT	DPM_EMU9_OFFMODEPULLUDENABLE	DPM_EMU9_OFFMODEOUTVALUE	DPM_EMU9_OFFMODEOUTENABLE	DPM_EMU9_OFFMODEEENABLE	DPM_EMU9_INPUTENABLE	RESERVED			DPM_EMU9_PULLTYPESELECT	DPM_EMU9_PULLUDENABLE			DPM_EMU9_MUXMODE

Bits	Field Name	Description	Type	Reset
31	DPM_EMU10_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0



Bits	Field Name	Description	Type	Reset
30	DPM_EMU10_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	DPM_EMU10_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu10 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU10_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu10 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU10_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu10 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU10_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu10. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	DPM_EMU10_OFFMODEENABLE	OffMode mode override control for pad dpm_emu10 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	DPM_EMU10_INPUTENABLE	Input enable value for pad dpm_emu10 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	DPM_EMU10_PULLTYPESELE CT	pullup/down selection for pad dpm_emu10 0x0: pulldown selected 0x1: pullup selected	RW	0
19	DPM_EMU10_PULLUDENABLE	pullup/down enable for pad dpm_emu10 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	DPM_EMU10_MUXMODE	Functional multiplexing selection for pad dpm_emu10 0x0: Select dpm_emu10 0x1: Select usba0_ulpiphy_dat4 0x3: Select gpio_21 0x4: Select rbi_a0 0x5: Select disp2_de 0x6: Select hw_dbg10 0x7: Select safe_mode	RW	0x7
15	DPM_EMU9_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU9_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	DPM_EMU9_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu9 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	DPM_EMU9_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu9 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU9_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu9 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU9_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu9. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU9_OFFMODEENABLE	OffMode mode override control for pad dpm_emu9 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	DPM_EMU9_INPUTENABLE	Input enable value for pad dpm_emu9 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	DPM_EMU9_PULLTYPESELECTION	pullup/down selection for pad dpm_emu9 0x0: pulldown selected 0x1: pullup selected	RW	0
3	DPM_EMU9_PULLUDENABLE	pullup/down enable for pad dpm_emu9 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU9_MUXMODE	Functional multiplexing selection for pad dpm_emu9 0x0: Select dpm_emu9 0x1: Select usba0_ulpipiphy_dat3 0x2: Select uart3_cts_rctx 0x3: Select gpio_20 0x4: Select rfb_i_we 0x5: Select dispc2_vsync 0x6: Select hw_dbg9 0x7: Select safe_mode	RW	0x7

**Table 18-515. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_DPM\_EMU9\_PAD1\_DPM\_EMU10**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-516. CONTROL\_CORE\_PAD0\_DPM\_EMU11\_PAD1\_DPM\_EMU12**

<b>Address Offset</b>	0x0000 01C4		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads dpm_emu11 and dpm_emu12 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU12_WAKEUPEVENT	DPM_EMU12_WAKEUPENABLE	DPM_EMU12_OFFMODEPULLTYPESELECT	DPM_EMU12_OFFMODEPULLUDENENABLE	DPM_EMU12_OFFMODEOUTVALUE	DPM_EMU12_OFFMODEOUTENABLE	DPM_EMU12_OFFMODEENABLE	DPM_EMU12_INPUTENABLE	RESERVED	RESERVED	RESERVED	DPM_EMU12_PULLTYPESELECT	DPM_EMU12_PULLUDENENABLE	DPM_EMU12_MUXMODE	DPM_EMU11_WAKEUPEVENT	DPM_EMU11_WAKEUPENABLE	DPM_EMU11_OFFMODEPULLTYPESELECT	DPM_EMU11_OFFMODEPULLUDENENABLE	DPM_EMU11_OFFMODEOUTVALUE	DPM_EMU11_OFFMODEOUTENABLE	DPM_EMU11_OFFMODEENABLE	DPM_EMU11_INPUTENABLE	RESERVED	RESERVED	RESERVED	RESERVED	DPM_EMU11_PULLTYPESELECT	DPM_EMU11_PULLUDENENABLE	DPM_EMU11_MUXMODE			

Bits	Field Name	Description	Type	Reset
31	DPM_EMU12_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	DPM_EMU12_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	DPM_EMU12_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu12 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU12_OFFMODEPULLUDENENABLE	OffMode mode pullup/down enable for pad dpm_emu12 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU12_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu12 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU12_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu12. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	DPM_EMU12_OFFMODEENABLE	OffMode mode override control for pad dpm_emu12 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
24	DPM_EMU12_INPUTENABLE	Input enable value for pad dpm_emu12 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	DPM_EMU12_PULLTYPESELE CT	pullup/down selection for pad dpm_emu12 0x0: pulldown selected 0x1: pullup selected	RW	0
19	DPM_EMU12_PULLUDENABLE	pullup/down enable for pad dpm_emu12 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	DPM_EMU12_MUXMODE	Functional multiplexing selection for pad dpm_emu12 0x0: Select dpm_emu12 0x1: Select usba0_ulpiphy_dat6 0x3: Select gpio_23 0x4: Select rfi_data7 0x5: Select disp2_data7 0x6: Select hw_dbg12 0x7: Select safe_mode	RW	0x7
15	DPM_EMU11_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU11_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	DPM_EMU11_ OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu11 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	DPM_EMU11_ OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu11 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU11_ OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu11 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU11_ OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu11. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU11_ OFFMODEENABLE	OffMode mode override control for pad dpm_emu11 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	DPM_EMU11_INPUTENABLE	Input enable value for pad dpm_emu11 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	DPM_EMU11_PULLTYPESELE CT	pullup/down selection for pad dpm_emu11 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	DPM_EMU11_PULLUDENABLE	pullup/down enable for pad dpm_emu11 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU11_MUXMODE	Functional multiplexing selection for pad dpm_emu11 0x0: Select dpm_emu11 0x1: Select usba0_ulpiphy_dat5 0x3: Select gpio_22 0x4: Select rfb_data8 0x5: Select disp2_data8 0x6: Select hw_dbg11 0x7: Select safe_mode	RW	0x7

**Table 18-517. Register Call Summary for Register CONTROL\_CORE\_PAD0\_DPM\_EMU11\_PAD1\_DPM\_EMU12**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-518. CONTROL\_CORE\_PAD0\_DPM\_EMU13\_PAD1\_DPM\_EMU14**

<b>Address Offset</b>	0x0000 01C8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads dpm_emu13 and dpm_emu14 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU14_WAKEUPEVENT	DPM_EMU14_WAKEUPENABLE	DPM_EMU14_OFFMODEPULLTYPESELECT	DPM_EMU14_OFFMODEPULLUDENABLE	DPM_EMU14_OFFMODEOUTVALUE	DPM_EMU14_OFFMODEOUTENABLE	DPM_EMU14_OFFMODEENABLE	DPM_EMU14_INPUTENABLE	RESERVED			DPM_EMU14_PULLTYPESELECT	DPM_EMU14_PULLUDENABLE	DPM_EMU14_MUXMODE			DPM_EMU13_WAKEUPEVENT	DPM_EMU13_WAKEUPENABLE	DPM_EMU13_OFFMODEPULLTYPESELECT	DPM_EMU13_OFFMODEPULLUDENABLE	DPM_EMU13_OFFMODEOUTVALUE	DPM_EMU13_OFFMODEOUTENABLE	DPM_EMU13_OFFMODEENABLE	DPM_EMU13_INPUTENABLE	RESERVED			DPM_EMU13_PULLTYPESELECT	DPM_EMU13_PULLUDENABLE	DPM_EMU13_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	DPM_EMU14_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	DPM_EMU14_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
29	DPM_EMU14_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu14 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU14_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu14 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU14_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu14 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU14_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu14. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	DPM_EMU14_OFFMODEENABLE	OffMode mode override control for pad dpm_emu14 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	DPM_EMU14_INPUTENABLE	Input enable value for pad dpm_emu14 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	DPM_EMU14_PULLTYPESELE CT	pullup/down selection for pad dpm_emu14 0x0: pulldown selected 0x1: pullup selected	RW	0
19	DPM_EMU14_PULLUDENABLE	pullup/down enable for pad dpm_emu14 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	DPM_EMU14_MUXMODE	Functional multiplexing selection for pad dpm_emu14 0x0: Select dpm_emu14 0x1: Reserved to Non-GP devices 0x2: Select uart1_rx 0x3: Select gpio_25 0x4: Select rfb_data5 0x5: Select dispc2_data5 0x6: Select hw_dbg14 0x7: Select safe_mode	RW	0x7
15	DPM_EMU13_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU13_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	DPM_EMU13_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu13 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
12	DPM_EMU13_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu13 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU13_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu13 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU13_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu13. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU13_OFFMODEENABLE	OffMode mode override control for pad dpm_emu13 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	DPM_EMU13_INPUTENABLE	Input enable value for pad dpm_emu13 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	DPM_EMU13_PULLTYPESELE CT	pullup/down selection for pad dpm_emu13 0x0: pulldown selected 0x1: pullup selected	RW	0
3	DPM_EMU13_PULLUDENABLE	pullup/down enable for pad dpm_emu13 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU13_MUXMODE	Functional multiplexing selection for pad dpm_emu13 0x0: Select dpm_emu13 0x1: Select usba0_ulpipiphy_dat7 0x3: Select gpio_24 0x4: Select rfb_data6 0x5: Select dispc2_data6 0x6: Select hw_dbg13 0x7: Select safe_mode	RW	0x7

**Table 18-519. Register Call Summary for Register CONTROL\_CORE\_PAD0\_DPM\_EMU13\_PAD1\_DPM\_EMU14**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-520. CONTROL\_CORE\_PAD0\_DPM\_EMU15\_PAD1\_DPM\_EMU16**

<b>Address Offset</b>	0x0000 01CC		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pads dpm_emu15 and dpm_emu16 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU16_WAKEUPEVENT	DPM_EMU16_WAKEUPENABLE	DPM_EMU16_OFFMODEPULLTYPESELECT	DPM_EMU16_OFFMODEPULLUDENABE	DPM_EMU16_OFFMODEOUTVALUE	DPM_EMU16_OFFMODEOUTENABLE	DPM_EMU16_OFFMODEENABLE	DPM_EMU16_INPUTENABLE	RESERVED			DPM_EMU16_PULLTYPESELECT	DPM_EMU16_PULLUDENABE			DPM_EMU16_MUXMODE	DPM_EMU15_WAKEUPEVENT	DPM_EMU15_WAKEUPENABLE	DPM_EMU15_OFFMODEPULLTYPESELECT	DPM_EMU15_OFFMODEPULLUDENABE	DPM_EMU15_OFFMODEOUTVALUE	DPM_EMU15_OFFMODEOUTENABLE	DPM_EMU15_OFFMODEENABLE	DPM_EMU15_INPUTENABLE	RESERVED			DPM_EMU15_PULLTYPESELECT	DPM_EMU15_PULLUDENABE			DPM_EMU15_MUXMODE

Bits	Field Name	Description	Type	Reset
31	DPM_EMU16_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	DPM_EMU16_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	DPM_EMU16_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu16 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU16_OFFMODEPULLUDENABE	OffMode mode pullup/down enable for pad dpm_emu16 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU16_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu16 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU16_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu16. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	DPM_EMU16_OFFMODEENABLE	OffMode mode override control for pad dpm_emu16 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0



Bits	Field Name	Description	Type	Reset
24	DPM_EMU16_INPUTENABLE	Input enable value for pad dpm_emu16 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	DPM_EMU16_PULLTYPESELE CT	pullup/down selection for pad dpm_emu16 0x0: pulldown selected 0x1: pullup selected	RW	0
19	DPM_EMU16_PULLUDENABLE	pullup/down enable for pad dpm_emu16 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	DPM_EMU16_MUXMODE	Functional multiplexing selection for pad dpm_emu16 0x0: Select dpm_emu16 0x1: Select dmtimer8_pwm_evt 0x2: Select dsi1_te0 0x3: Select gpio_27 0x4: Select rfb_data3 0x5: Select disp2_data3 0x6: Select hw_dbg16 0x7: Select safe_mode	RW	0x7
15	DPM_EMU15_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU15_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	DPM_EMU15_ OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu15 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	DPM_EMU15_ OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu15 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU15_ OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu15 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU15_ OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu15. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU15_ OFFMODEENABLE	OffMode mode override control for pad dpm_emu15 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	DPM_EMU15_INPUTENABLE	Input enable value for pad dpm_emu15 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	DPM_EMU15_PULLTYPESELE CT	pullup/down selection for pad dpm_emu15 0x0: pulldown selected 0x1: pullup selected	RW	0
3	DPM_EMU15_PULLUDENABLE	pullup/down enable for pad dpm_emu15 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU15_MUXMODE	Functional multiplexing selection for pad dpm_emu15 0x0: Select dpm_emu15 0x1: Reserved 0x3: Select gpio_26 0x4: Select rfb_data4 0x5: Select disp2_data4 0x6: Select hw_dbg15 0x7: Select safe_mode	RW	0x7

**Table 18-521. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_DPM\_EMU15\_PAD1\_DPM\_EMU16**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-522. CONTROL\_CORE\_PAD0\_DPM\_EMU17\_PAD1\_DPM\_EMU18**

<b>Address Offset</b>	0x0000 01D0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Register control for Pads dpm_emu17 and dpm_emu18 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU18_WAKEUPEVENT	DPM_EMU18_WAKEUPENABLE	DPM_EMU18_OFFMODEPULLTYPESELECT	DPM_EMU18_OFFMODEPULLUDENABLE	DPM_EMU18_OFFMODEOUTVALUE	DPM_EMU18_OFFMODEOUTENABLE	DPM_EMU18_OFFMODEENABLE	DPM_EMU18_INPUTENABLE	RESERVED			DPM_EMU18_PULLTYPESELECT	DPM_EMU18_PULLUDENABLE		DPM_EMU18_MUXMODE		DPM_EMU17_WAKEUPEVENT	DPM_EMU17_WAKEUPENABLE	DPM_EMU17_OFFMODEPULLTYPESELECT	DPM_EMU17_OFFMODEPULLUDENABLE	DPM_EMU17_OFFMODEOUTVALUE	DPM_EMU17_OFFMODEOUTENABLE	DPM_EMU17_OFFMODEENABLE	DPM_EMU17_INPUTENABLE	RESERVED			DPM_EMU17_PULLTYPESELECT	DPM_EMU17_PULLUDENABLE		DPM_EMU17_MUXMODE	

Bits	Field Name	Description	Type	Reset
31	DPM_EMU18_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
30	DPM_EMU18_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29	DPM_EMU18_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu18 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
28	DPM_EMU18_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu18 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
27	DPM_EMU18_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu18 0x0: Set value at 0 0x1: Set value at 1	RW	0
26	DPM_EMU18_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu18. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
25	DPM_EMU18_OFFMODEENABLE	OffMode mode override control for pad dpm_emu18 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
24	DPM_EMU18_INPUTENABLE	Input enable value for pad dpm_emu18 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	DPM_EMU18_PULLTYPESELE CT	pullup/down selection for pad dpm_emu18 0x0: pulldown selected 0x1: pullup selected	RW	0
19	DPM_EMU18_PULLUDENABLE	pullup/down enable for pad dpm_emu18 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	DPM_EMU18_MUXMODE	Functional multiplexing selection for pad dpm_emu18 0x0: Select dpm_emu18 0x1: Select dmtimer10_pwm_evt 0x2: Select dsi2_te0 0x3: Select gpio_190 0x4: Select rfb_data1 0x5: Select dispc2_data1 0x6: Select hw_dbg18 0x7: Select safe_mode	RW	0x7
15	DPM_EMU17_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU17_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
13	DPM_EMU17_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu17 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	DPM_EMU17_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu17 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU17_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu17 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU17_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu17. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU17_OFFMODEENABLE	OffMode mode override control for pad dpm_emu17 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0
8	DPM_EMU17_INPUTENABLE	Input enable value for pad dpm_emu17 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	DPM_EMU17_PULLTYPESELE CT	pullup/down selection for pad dpm_emu17 0x0: pulldown selected 0x1: pullup selected	RW	0
3	DPM_EMU17_PULLUDENABLE	pullup/down enable for pad dpm_emu17 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU17_MUXMODE	Functional multiplexing selection for pad dpm_emu17 0x0: Select dpm_emu17 0x1: Select dmtimer9_pwm_evt 0x2: Select dsi1_te1 0x3: Select gpio_28 0x4: Select rfb_data2 0x5: Select dispc2_data2 0x6: Select hw_dbg17 0x7: Select safe_mode	RW	0x7

**Table 18-523. Register Call Summary for Register  
CONTROL\_CORE\_PAD0\_DPM\_EMU17\_PAD1\_DPM\_EMU18**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-524. CONTROL\_CORE\_PAD0\_DPM\_EMU19**

<b>Address Offset</b>	0x0000 01D4		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Register control for Pad dpm_emu19 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPM_EMU19_WAKEUPEVENT	DPM_EMU19_WAKEUPENABLE	DPM_EMU19_OFFMODEPULLTYPESELECT	DPM_EMU19_OFFMODEPULLUDENABLE	DPM_EMU19_OFFMODEOUTVALUE	DPM_EMU19_OFFMODEOUTENABLE	DPM_EMU19_OFFMODEENABLE	DPM_EMU19_INPUTENABLE	RESERVED			DPM_EMU19_PULLTYPESELECT	DPM_EMU19_PULLUDENABLE	DPM_EMU19_MUXMODE		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	DPM_EMU19_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	DPM_EMU19_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13	DPM_EMU19_OFFMODEPULLTYPESELECT	OffMode mode pullup/down selection for pad dpm_emu19 0x0: Offmode pulldown selected 0x1: Offmode pullup selected	RW	0
12	DPM_EMU19_OFFMODEPULLUDENABLE	OffMode mode pullup/down enable for pad dpm_emu19 0x0: Offmode pullup/down disabled 0x1: Offmode pullup/down enabled	RW	0
11	DPM_EMU19_OFFMODEOUTVALUE	OffMode mode output value for pad dpm_emu19 0x0: Set value at 0 0x1: Set value at 1	RW	0
10	DPM_EMU19_OFFMODEOUTENABLE	OffMode mode output enable value for pad dpm_emu19. This is an active low signal 0x0: Output enable 0x1: Output disable	RW	0
9	DPM_EMU19_OFFMODEENABLE	OffMode mode override control for pad dpm_emu19 0x0: I/O state keeps its previous state when OFF mode is active 0x1: I/O state is forced to OFF mode value when OFF mode is active	RW	0

Bits	Field Name	Description	Type	Reset
8	DPM_EMU19_INPUTENABLE	Input enable value for pad dpm_emu19 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	DPM_EMU19_PULLTYPESELE CT	pullup/down selection for pad dpm_emu19 0x0: pulldown selected 0x1: pullup selected	RW	0
3	DPM_EMU19_PULLUDENABLE	pullup/down enable for pad dpm_emu19 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	DPM_EMU19_MUXMODE	Functional multiplexing selection for pad dpm_emu19 0x0: Select dpm_emu19 0x1: Select dmtimer11_pwm_evt 0x2: Select dsi2_te1 0x3: Select gpio_191 0x4: Select rfb_data0 0x5: Select disp2_data0 0x6: Select hw_dbg19 0x7: Select safe_mode	RW	0x7

**Table 18-525. Register Call Summary for Register CONTROL\_CORE\_PAD0\_DPM\_EMU19**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-526. CONTROL\_PADCONF\_WAKEUPEVENT\_0**

<b>Address Offset</b>	0x0000 01D8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_CLK_DUPLICATEWAKEUPEVENT	GPMC_NWP_DUPLICATEWAKEUPEVENT	GPMC_NCS3_DUPLICATEWAKEUPEVENT	GPMC_NCS2_DUPLICATEWAKEUPEVENT	GPMC_NCS1_DUPLICATEWAKEUPEVENT	GPMC_NCS0_DUPLICATEWAKEUPEVENT	GPMC_A25_DUPLICATEWAKEUPEVENT	GPMC_A24_DUPLICATEWAKEUPEVENT	GPMC_A23_DUPLICATEWAKEUPEVENT	GPMC_A22_DUPLICATEWAKEUPEVENT	GPMC_A21_DUPLICATEWAKEUPEVENT	GPMC_A20_DUPLICATEWAKEUPEVENT	GPMC_A19_DUPLICATEWAKEUPEVENT	GPMC_A18_DUPLICATEWAKEUPEVENT	GPMC_A17_DUPLICATEWAKEUPEVENT	GPMC_A16_DUPLICATEWAKEUPEVENT	GPMC_AD15_DUPLICATEWAKEUPEVENT	GPMC_AD14_DUPLICATEWAKEUPEVENT	GPMC_AD13_DUPLICATEWAKEUPEVENT	GPMC_AD12_DUPLICATEWAKEUPEVENT	GPMC_AD11_DUPLICATEWAKEUPEVENT	GPMC_AD10_DUPLICATEWAKEUPEVENT	GPMC_AD9_DUPLICATEWAKEUPEVENT	GPMC_AD8_DUPLICATEWAKEUPEVENT	GPMC_AD7_DUPLICATEWAKEUPEVENT	GPMC_AD6_DUPLICATEWAKEUPEVENT	GPMC_AD5_DUPLICATEWAKEUPEVENT	GPMC_AD4_DUPLICATEWAKEUPEVENT	GPMC_AD3_DUPLICATEWAKEUPEVENT	GPMC_AD2_DUPLICATEWAKEUPEVENT	GPMC_AD1_DUPLICATEWAKEUPEVENT	GPMC_AD0_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	GPMC_CLK_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_clk Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPMC_NWP_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_nwp Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
29	GPMC_NCS3_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ncs3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
28	GPMC_NCS2_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ncs2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
27	GPMC_NCS1_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ncs1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
26	GPMC_NCS0_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ncs0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
25	GPMC_A25_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a25 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
24	GPMC_A24_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a24 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
23	GPMC_A23_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a23 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
22	GPMC_A22_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a22 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
21	GPMC_A21_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a21 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
20	GPMC_A20_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a20 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
19	GPMC_A19_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a19 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
18	GPMC_A18_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a18 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
17	GPMC_A17_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a17 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
16	GPMC_A16_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_a16 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
15	GPMC_AD15_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad15 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPMC_AD14_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad14 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
13	GPMC_AD13_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad13 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
12	GPMC_AD12_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad12 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
11	GPMC_AD11_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad11 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
10	GPMC_AD10_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad10 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
9	GPMC_AD9_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad9 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
8	GPMC_AD8_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad8 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
7	GPMC_AD7_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad7 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
6	GPMC_AD6_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad6 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0



Bits	Field Name	Description	Type	Reset
5	GPMC_AD5_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad5 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
4	GPMC_AD4_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
3	GPMC_AD3_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
2	GPMC_AD2_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
1	GPMC_AD1_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
0	GPMC_AD0_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpmc_ad0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

**Table 18-527. Register Call Summary for Register CONTROL\_PADCONF\_WAKEUPEVENT\_0**

Control Module Functional Description

- [Wake-Up Event Detection: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[1\]](#)

**Table 18-528. CONTROL\_PADCONF\_WAKEUPEVENT\_1**

<b>Address Offset</b>	0x0000 01DC	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAM_STROBE_DUPLICATEWAKEUPEVENT	CAM_SHUTTER_DUPLICATEWAKEUPEVENT	CSI22_DY1_DUPLICATEWAKEUPEVENT	CSI22_DX1_DUPLICATEWAKEUPEVENT	CSI22_DY0_DUPLICATEWAKEUPEVENT	CSI22_DX0_DUPLICATEWAKEUPEVENT	CSI21_DY4_DUPLICATEWAKEUPEVENT	CSI21_DX4_DUPLICATEWAKEUPEVENT	CSI21_DY3_DUPLICATEWAKEUPEVENT	CSI21_DX3_DUPLICATEWAKEUPEVENT	CSI21_DY2_DUPLICATEWAKEUPEVENT	CSI21_DX2_DUPLICATEWAKEUPEVENT	CSI21_DY1_DUPLICATEWAKEUPEVENT	CSI21_DX1_DUPLICATEWAKEUPEVENT	CSI21_DY0_DUPLICATEWAKEUPEVENT	CSI21_DX0_DUPLICATEWAKEUPEVENT	GPIO66_DUPLICATEWAKEUPEVENT	GPIO65_DUPLICATEWAKEUPEVENT	GPIO64_DUPLICATEWAKEUPEVENT	GPIO63_DUPLICATEWAKEUPEVENT	GPMC_NCS7_DUPLICATEWAKEUPEVENT	GPMC_NCS6_DUPLICATEWAKEUPEVENT	GPMC_NCS5_DUPLICATEWAKEUPEVENT	GPMC_NCS4_DUPLICATEWAKEUPEVENT	GPMC_WAIT2_DUPLICATEWAKEUPEVENT	GPMC_WAIT1_DUPLICATEWAKEUPEVENT	GPMC_WAIT0_DUPLICATEWAKEUPEVENT	GPMC_NBE1_DUPLICATEWAKEUPEVENT	GPMC_NBE0_CLE_DUPLICATEWAKEUPEVENT	GPMC_NWE_DUPLICATEWAKEUPEVENT	GPMC_NOE_DUPLICATEWAKEUPEVENT	GPMC_NADV_ALE_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	CAM_STROBE_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad cam_strobe Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	CAM_SHUTTER_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad cam_shutter Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
29	CSI22_DY1_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad csi22_dy1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
28	CSI22_DX1_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad csi22_dx1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
27	CSI22_DY0_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad csi22_dy0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
26	CSI22_DX0_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad csi22_dx0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
25	CSI21_DY4_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad csi21_dy4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
24	CSI21_DX4_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad csi21_dx4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
23	CSI21_DY3_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad csi21_dy3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
22	CSI21_DX3_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad csi21_dx3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
21	CSI21_DY2_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad csi21_dy2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
20	CSI21_DX2_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad csi21_dx2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
19	CSI21_DY1_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad csi21_dy1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
18	CSI21_DX1_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad csi21_dx1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
17	CSI21_DY0_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad csi21_dy0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
16	CSI21_DX0_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad csi21_dx0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
15	GPIO66_DUPLICATEWAKEUPE VENT	Wake-up event status latched in the I/O for pad gpio66 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPIO65_DUPLICATEWAKEUPE VENT	Wake-up event status latched in the I/O for pad gpio65 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
13	GPIO64_DUPLICATEWAKEUPE VENT	Wake-up event status latched in the I/O for pad gpio64 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
12	GPIO63_DUPLICATEWAKEUPE VENT	Wake-up event status latched in the I/O for pad gpio63 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
11	GPMC_NCS7_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ncs7 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
10	GPMC_NCS6_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ncs6 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
9	GPMC_NCS5_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ncs5 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
8	GPMC_NCS4_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_ncs4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
7	GPMC_WAIT2_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_wait2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
6	GPMC_WAIT1_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_wait1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
5	GPMC_WAIT0_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_wait0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
4	GPMC_NBE1_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_nbe1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
3	GPMC_NBE0_CLE_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_nbe0_cle Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
2	GPMC_NWE_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_nwe Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
1	GPMC_NOE_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_noe Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
0	GPMC_NADV_ALE_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpmc_nadv_ale Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

**Table 18-529. Register Call Summary for Register CONTROL\_PADCONF\_WAKEUPEVENT\_1**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-530. CONTROL\_PADCONF\_WAKEUPEVENT\_2**

<b>Address Offset</b>	0x0000 01E0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_MCBSP1_CLKX_DUPLICATEWAKEUPEVENT	ABE_MCBSP2_FSX_DUPLICATEWAKEUPEVENT	ABE_MCBSP2_DX_DUPLICATEWAKEUPEVENT	ABE_MCBSP2_DR_DUPLICATEWAKEUPEVENT	ABE_MCBSP2_CLKX_DUPLICATEWAKEUPEVENT	SDMMC1_DAT7_DUPLICATEWAKEUPEVENT	SDMMC1_DAT6_DUPLICATEWAKEUPEVENT	SDMMC1_DAT5_DUPLICATEWAKEUPEVENT	SDMMC1_DAT4_DUPLICATEWAKEUPEVENT	SDMMC1_DAT3_DUPLICATEWAKEUPEVENT	SDMMC1_DAT2_DUPLICATEWAKEUPEVENT	SDMMC1_DAT1_DUPLICATEWAKEUPEVENT	SDMMC1_DAT0_DUPLICATEWAKEUPEVENT	SDMMC1_CMD_DUPLICATEWAKEUPEVENT	SDMMC1_CLK_DUPLICATEWAKEUPEVENT	USBC1_ICUSB_DM_DUPLICATEWAKEUPEVENT	USBC1_ICUSB_DP_DUPLICATEWAKEUPEVENT	USBB1_HSIC_STROBE_DUPLICATEWAKEUPEVENT	USBB1_HSIC_DATA_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_DAT7_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_DAT6_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_DAT5_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_DAT4_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_DAT3_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_DAT2_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_DAT1_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_DAT0_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_NXT_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_DIR_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_STP_DUPLICATEWAKEUPEVENT	USBB1_ULPITLL_CLK_DUPLICATEWAKEUPEVENT	CAM_GLOBALRESET_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	ABE_MCBSP1_CLKX_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad <code>abe_mcbbsp1_clkx</code> Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	ABE_MCBSP2_FSX_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad <code>abe_mcbbsp2_fsx</code> Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
29	ABE_MCBSP2_DX_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad <code>abe_mcbbsp2_dx</code> Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
28	ABE_MCBSP2_DR_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad <code>abe_mcbbsp2_dr</code> Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
27	ABE_MCBSP2_CLKX_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad <code>abe_mcbbsp2_clkx</code> Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
26	SDMMC1_DAT7_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad <code>sdmmc1_dat7</code> Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
25	SDMMC1_DAT6_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc1_dat6 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
24	SDMMC1_DAT5_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc1_dat5 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
23	SDMMC1_DAT4_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc1_dat4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
22	SDMMC1_DAT3_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc1_dat3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
21	SDMMC1_DAT2_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc1_dat2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
20	SDMMC1_DAT1_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc1_dat1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
19	SDMMC1_DAT0_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc1_dat0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
18	SDMMC1_CMD_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc1_cmd Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
17	SDMMC1_CLK_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc1_clk Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
16	USBC1_ICUSB_DM_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbc1_icusb_dm Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
15	USBC1_ICUSB_DP_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbc1_icusb_dp Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	USBB1_HSIC_STROBE_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_hsic_strobe Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
13	USBB1_HSIC_DATA_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_hsic_data Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
12	USBB1_ULPITLL_DAT7_DUPLI CATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_dat7  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
11	USBB1_ULPITLL_DAT6_DUPLI CATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_dat6  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
10	USBB1_ULPITLL_DAT5_DUPLI CATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_dat5  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
9	USBB1_ULPITLL_DAT4_DUPLI CATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_dat4  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
8	USBB1_ULPITLL_DAT3_DUPLI CATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_dat3  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
7	USBB1_ULPITLL_DAT2_DUPLI CATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_dat2  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
6	USBB1_ULPITLL_DAT1_DUPLI CATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_dat1  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
5	USBB1_ULPITLL_DAT0_DUPLI CATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_dat0  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
4	USBB1_ULPITLL_NXT_DUPLIC ATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_nxt  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
3	USBB1_ULPITLL_DIR_DUPLICA TEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_dir  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
2	USBB1_ULPITLL_STP_DUPLIC ATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_stp  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
1	USBB1_ULPITLL_CLK_DUPLIC ATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb1_ulpitll_clk  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
0	CAM_GLOBALRESET_DUPLIC ATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad cam_globalreset  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

**Table 18-531. Register Call Summary for Register CONTROL\_PADCONF\_WAKEUPEVENT\_2**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-532. CONTROL\_PADCONF\_WAKEUPEVENT\_3**

<b>Address Offset</b>	0x0000 01E4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCSP11_CS3_DUPLICATEWAKEUPEVENT	MCSP11_CS2_DUPLICATEWAKEUPEVENT	MCSP11_CS1_DUPLICATEWAKEUPEVENT	MCSP11_CS0_DUPLICATEWAKEUPEVENT	MCSP11_SIMO_DUPLICATEWAKEUPEVENT	MCSP11_SOMI_DUPLICATEWAKEUPEVENT	MCSP11_CLK_DUPLICATEWAKEUPEVENT	I2C4_SDA_DUPLICATEWAKEUPEVENT	I2C4_SCL_DUPLICATEWAKEUPEVENT	I2C3_SDA_DUPLICATEWAKEUPEVENT	I2C3_SCL_DUPLICATEWAKEUPEVENT	I2C2_SDA_DUPLICATEWAKEUPEVENT	I2C2_SCL_DUPLICATEWAKEUPEVENT	I2C1_SDA_DUPLICATEWAKEUPEVENT	I2C1_SCL_DUPLICATEWAKEUPEVENT	HDQ_SIO_DUPLICATEWAKEUPEVENT	UART2_TX_DUPLICATEWAKEUPEVENT	UART2_RX_DUPLICATEWAKEUPEVENT	UART2_RTS_DUPLICATEWAKEUPEVENT	UART2_CTS_DUPLICATEWAKEUPEVENT	ABE_DMIC_DIN3_DUPLICATEWAKEUPEVENT	ABE_DMIC_DIN2_DUPLICATEWAKEUPEVENT	ABE_DMIC_DIN1_DUPLICATEWAKEUPEVENT	ABE_DMIC_CLK1_DUPLICATEWAKEUPEVENT	ABE_CLKS_DUPLICATEWAKEUPEVENT	ABE_PDM_LB_CLK_DUPLICATEWAKEUPEVENT	ABE_PDM_FRAME_DUPLICATEWAKEUPEVENT	ABE_PDM_DL_DATA_DUPLICATEWAKEUPEVENT	ABE_PDM_UL_DATA_DUPLICATEWAKEUPEVENT	ABE_MCBSP1_FSX_DUPLICATEWAKEUPEVENT	ABE_MCBSP1_DX_DUPLICATEWAKEUPEVENT	ABE_MCBSP1_DR_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	MCSP11_CS3_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi1_cs3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	MCSP11_CS2_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi1_cs2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
29	MCSP11_CS1_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi1_cs1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
28	MCSP11_CS0_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi1_cs0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
27	MCSP11_SIMO_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi1_simo Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0



Bits	Field Name	Description	Type	Reset
26	MCSP11_SOMI_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi1_somi Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
25	MCSP11_CLK_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi1_clk Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
24	I2C4_SDA_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad i2c4_sda Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
23	I2C4_SCL_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad i2c4_scl Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
22	I2C3_SDA_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad i2c3_sda Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
21	I2C3_SCL_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad i2c3_scl Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
20	I2C2_SDA_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad i2c2_sda Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
19	I2C2_SCL_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad i2c2_scl Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
18	I2C1_SDA_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad i2c1_sda Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
17	I2C1_SCL_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad i2c1_scl Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
16	HDQ_SIO_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad hdq_sio Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
15	UART2_TX_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart2_tx Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	UART2_RX_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart2_rx Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
13	UART2_RTS_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart2_rts Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
12	UART2_CTS_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart2_cts Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
11	ABE_DMIC_DIN3_DUPLICATE WAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_dmic_din3  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
10	ABE_DMIC_DIN2_DUPLICATE WAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_dmic_din2  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
9	ABE_DMIC_DIN1_DUPLICATE WAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_dmic_din1  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
8	ABE_DMIC_CLK1_DUPLICATE WAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_dmic_clk1  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
7	ABE_CLKS_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad abe_clks  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
6	ABE_PDM_LB_CLK_DUPLICAT EWAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_pdm_lb_clk  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
5	ABE_PDM_FRAME_DUPLICAT EWAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_pdm_frame  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
4	ABE_PDM_DL_DATA_DUPLICA TEWAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_pdm_dl_data  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
3	ABE_PDM_UL_DATA_DUPLICA TEWAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_pdm_ul_data  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
2	ABE_MCBSP1_FSX_DUPLICAT EWAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_mcbbsp1_fsx  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
1	ABE_MCBSP1_DX_DUPLICATE WAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_mcbbsp1_dx  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
0	ABE_MCBSP1_DR_DUPLICATE WAKEUPEVENT	Wake-up event status latched in the I/O for pad abe_mcbbsp1_dr  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

**Table 18-533. Register Call Summary for Register CONTROL\_PADCONF\_WAKEUPEVENT\_3**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-534. CONTROL\_PADCONF\_WAKEUPEVENT\_4**

<b>Address Offset</b>	0x0000 01E8		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPD_COL4_DUPLICATEWAKEUPEVENT	KPD_COL3_DUPLICATEWAKEUPEVENT	USBB2_HSIC_STROBE_DUPLICATEWAKEUPEVENT	USBB2_HSIC_DATA_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_DAT7_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_DAT6_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_DAT5_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_DAT4_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_DAT3_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_DAT2_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_DAT1_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_DAT0_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_NXT_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_DIR_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_STP_DUPLICATEWAKEUPEVENT	USBB2_ULPITLL_CLK_DUPLICATEWAKEUPEVENT	UART4_TX_DUPLICATEWAKEUPEVENT	UART4_RX_DUPLICATEWAKEUPEVENT	MCSP14_CS0_DUPLICATEWAKEUPEVENT	MCSP14_SOMI_DUPLICATEWAKEUPEVENT	MCSP14_SIMO_DUPLICATEWAKEUPEVENT	MCSP14_CLK_DUPLICATEWAKEUPEVENT	SDMMC5_DAT3_DUPLICATEWAKEUPEVENT	SDMMC5_DAT2_DUPLICATEWAKEUPEVENT	SDMMC5_DAT1_DUPLICATEWAKEUPEVENT	SDMMC5_DAT0_DUPLICATEWAKEUPEVENT	SDMMC5_CMD_DUPLICATEWAKEUPEVENT	SDMMC5_CLK_DUPLICATEWAKEUPEVENT	UART3_TX_IRTX_DUPLICATEWAKEUPEVENT	UART3_RX_IRRX_DUPLICATEWAKEUPEVENT	UART3_RTS_SD_DUPLICATEWAKEUPEVENT	UART3_CTS_RCTX_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	KPD_COL4_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad kpd_col4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	KPD_COL3_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad kpd_col3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
29	USBB2_HSIC_STROBE_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_hsic_strobe Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
28	USBB2_HSIC_DATA_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_hsic_data Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
27	USBB2_ULPITLL_DAT7_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_dat7 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
26	USBB2_ULPITLL_DAT6_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_dat6 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
25	USBB2_ULPITLL_DAT5_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_dat5 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
24	USBB2_ULPITLL_DAT4_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_dat4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
23	USBB2_ULPITLL_DAT3_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_dat3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
22	USBB2_ULPITLL_DAT2_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_dat2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
21	USBB2_ULPITLL_DAT1_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_dat1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
20	USBB2_ULPITLL_DAT0_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_dat0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
19	USBB2_ULPITLL_NXT_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_nxt Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
18	USBB2_ULPITLL_DIR_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_dir Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
17	USBB2_ULPITLL_STP_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_stp Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
16	USBB2_ULPITLL_CLK_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad usbb2_ulpitll_clk Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
15	UART4_TX_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart4_tx Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	UART4_RX_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart4_rx Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
13	MCSPi4_CS0_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi4_cs0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
12	MCSPi4_SOMI_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi4_somi Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
11	MCSPi4_SIMO_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi4_simo Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
10	MCSPi4_CLK_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad mcspi4_clk Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
9	SDMMC5_DAT3_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc5_dat3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
8	SDMMC5_DAT2_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc5_dat2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
7	SDMMC5_DAT1_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc5_dat1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
6	SDMMC5_DAT0_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc5_dat0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
5	SDMMC5_CMD_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc5_cmd Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
4	SDMMC5_CLK_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sdmmc5_clk Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
3	UART3_TX_IRTX_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart3_tx_irtx Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
2	UART3_RX_IRRX_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart3_rx_irrx Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
1	UART3_RTS_SD_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart3_rts_sd Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
0	UART3_CTS_RCTX_ DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad uart3_cts_rctx Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

**Table 18-535. Register Call Summary for Register CONTROL\_PADCONF\_WAKEUPEVENT\_4**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-536. CONTROL\_PADCONF\_WAKEUPEVENT\_5**

<b>Address Offset</b>	0x0000 01EC		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPM_EMU10_DUPLICATEWAKEUPEVENT	DPM_EMU9_DUPLICATEWAKEUPEVENT	DPM_EMU8_DUPLICATEWAKEUPEVENT	DPM_EMU7_DUPLICATEWAKEUPEVENT	DPM_EMU6_DUPLICATEWAKEUPEVENT	DPM_EMU5_DUPLICATEWAKEUPEVENT	DPM_EMU4_DUPLICATEWAKEUPEVENT	DPM_EMU3_DUPLICATEWAKEUPEVENT	DPM_EMU2_DUPLICATEWAKEUPEVENT	DPM_EMU1_DUPLICATEWAKEUPEVENT	DPM_EMU0_DUPLICATEWAKEUPEVENT	SYS_BOOT5_DUPLICATEWAKEUPEVENT	SYS_BOOT4_DUPLICATEWAKEUPEVENT	SYS_BOOT3_DUPLICATEWAKEUPEVENT	SYS_BOOT2_DUPLICATEWAKEUPEVENT	SYS_BOOT1_DUPLICATEWAKEUPEVENT	SYS_BOOT0_DUPLICATEWAKEUPEVENT	SYS_NIR02_DUPLICATEWAKEUPEVENT	SYS_NIR01_DUPLICATEWAKEUPEVENT	FREF_CLK2_OUT_DUPLICATEWAKEUPEVENT	FREF_CLK1_OUT_DUPLICATEWAKEUPEVENT	USBA0_OTG_DP_DUPLICATEWAKEUPEVENT	KPD_ROW2_DUPLICATEWAKEUPEVENT	KPD_ROW1_DUPLICATEWAKEUPEVENT	KPD_ROW0_DUPLICATEWAKEUPEVENT	KPD_ROW5_DUPLICATEWAKEUPEVENT	KPD_ROW4_DUPLICATEWAKEUPEVENT	KPD_ROW3_DUPLICATEWAKEUPEVENT	KPD_COL2_DUPLICATEWAKEUPEVENT	KPD_COL1_DUPLICATEWAKEUPEVENT	KPD_COL0_DUPLICATEWAKEUPEVENT	KPD_COL5_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31	DPM_EMU10_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu10 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	DPM_EMU9_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu9 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
29	DPM_EMU8_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu8 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
28	DPM_EMU7_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu7 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
27	DPM_EMU6_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu6 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
26	DPM_EMU5_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu5 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
25	DPM_EMU4_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
24	DPM_EMU3_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad dpm_emu3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
23	DPM_EMU2_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad dpm_emu2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
22	DPM_EMU1_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad dpm_emu1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
21	DPM_EMU0_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad dpm_emu0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
20	SYS_BOOT5_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad sys_boot5 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
19	SYS_BOOT4_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad sys_boot4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
18	SYS_BOOT3_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad sys_boot3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
17	SYS_BOOT2_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad sys_boot2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
16	SYS_BOOT1_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad sys_boot1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
15	SYS_BOOT0_DUPLICATEWAK EUPEVENT	Wake-up event status latched in the I/O for pad sys_boot0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SYS_NIRQ2_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad sys_nirq2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
13	SYS_NIRQ1_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad sys_nirq1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
12	FREF_CLK2_OUT_DUPLICATE WAKEUPEVENT	Wake-up event status latched in the I/O for pad fref_clk2_out Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
11	FREF_CLK1_OUT_DUPLICATE WAKEUPEVENT	Wake-up event status latched in the I/O for pad fref_clk1_out  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
10	USBA0_OTG_DP_DUPLICATE WAKEUPEVENT	Wake-up event status latched in the I/O for pad usba0_otg_dp  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
9	KPD_ROW2_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_row2  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
8	KPD_ROW1_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_row1  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
7	KPD_ROW0_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_row0  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
6	KPD_ROW5_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_row5  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
5	KPD_ROW4_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_row4  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
4	KPD_ROW3_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_row3  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
3	KPD_COL2_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_col2  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
2	KPD_COL1_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_col1  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
1	KPD_COL0_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_col0  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
0	KPD_COL5_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad kpd_col5  Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

**Table 18-537. Register Call Summary for Register CONTROL\_PADCONF\_WAKEUPEVENT\_5**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)



**Table 18-538. CONTROL\_PADCONF\_WAKEUPEVENT\_6**

<b>Address Offset</b>	0x0000 01F0		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							DPM_EMU19_DUPLICATEWAKEUPEVENT	DPM_EMU18_DUPLICATEWAKEUPEVENT	DPM_EMU17_DUPLICATEWAKEUPEVENT	DPM_EMU16_DUPLICATEWAKEUPEVENT	DPM_EMU15_DUPLICATEWAKEUPEVENT	DPM_EMU14_DUPLICATEWAKEUPEVENT	DPM_EMU13_DUPLICATEWAKEUPEVENT	DPM_EMU12_DUPLICATEWAKEUPEVENT	DPM_EMU11_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31: 9	RESERVED		R	0x0000000
8	DPM_EMU19_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the IO for pad dpm_emu19 Read 0x1: A wakeup event occurred Read 0x0: No wakeup event occurred	R	0x0000000
7	DPM_EMU18_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu18 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
6	DPM_EMU17_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu17 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
5	DPM_EMU16_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu16 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
4	DPM_EMU15_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu15 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
3	DPM_EMU14_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu14 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
2	DPM_EMU13_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu13 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
1	DPM_EMU12_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu12 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
0	DPM_EMU11_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad dpm_emu11 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

**Table 18-539. Register Call Summary for Register CONTROL\_PADCONF\_WAKEUPEVENT\_6**

Control Module Functional Description

- [Wake-Up Event Detection: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[1\]](#)

**Table 18-540. CONTROL\_PADCONF\_GLOBAL**

<b>Address Offset</b>	0x0000 05A0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	Global PAD control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31	FORCE_OFFMODE_EN	Force offmode enable for PADS 0x0: Offmode is not forced active 0x1: Offmode is forced active	RW	0
30:0	RESERVED		R	0x0000 0000

**Table 18-541. Register Call Summary for Register CONTROL\_PADCONF\_GLOBAL**

Control Module Functional Description

- [Device Core Control Module Instance: \[0\]](#)
- [Off Mode PAD Control Overview: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-542. CONTROL\_CORE\_PADCONF\_MODE**

<b>Address Offset</b>	0x0000 05A4		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	PAD Voltage Mode control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDS_DV_BANK0	VDDS_DV_BANK1	VDDS_DV_BANK3	VDDS_DV_BANK4	VDDS_DV_BANK5	VDDS_DV_BANK6	VDDS_DV_C2C	VDDS_DV_CAM	VDDS_DV_GPMC0	VDDS_DV_SDMMC2	VDDS_DV_BANK7	VDDS_DV_GPMC1	VDDS_DV_BANK2_SHARED0																			

Bits	Field Name	Description	Type	Reset
31	VDDS_DV_BANK0	PAD Voltage level control for vdds_dv_bank0 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
30	VDDS_DV_BANK1	PAD Voltage level control for vdds_dv_bank1 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
29	VDDS_DV_BANK3	PAD Voltage level control for vdds_dv_bank3 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
28	VDDS_DV_BANK4	PAD Voltage level control for vdds_dv_bank4 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
27	VDDS_DV_BANK5	PAD Voltage level control for vdds_dv_bank5 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
26	VDDS_DV_BANK6	PAD Voltage level control for vdds_dv_bank6 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
25	VDDS_DV_C2C	PAD Voltage level control for vdds_dv_c2c 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
24	VDDS_DV_CAM	PAD Voltage level control for vdds_dv_cam 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
23	VDDS_DV_GPMC0	PAD Voltage level control for vdds_dv_gpmc0 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
22	VDDS_DV_SDMMC2	PAD Voltage level control for vdds_dv_sdmmc2 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0

Bits	Field Name	Description	Type	Reset
21	VDDS_DV_BANK7	PAD Voltage level control for vdds_dv_bank7 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
20	VDDS_DV_GPMC1	PAD Voltage level control for vdds_dv_gpmc1 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
19	VDDS_DV_BANK2_SHARED0	PAD Voltage level control for vdds_dv_bank2_shared0 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
18:0	RESERVED		R	0x000000

**Table 18-543. Register Call Summary for Register CONTROL\_CORE\_PADCONF\_MODE**

Control Module Functional Description

- [Dual Voltage-Supplied Peripheral Controls: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[1\]](#)

**Table 18-544. CONTROL\_SMART1IO\_PADCONF\_0**

<b>Address Offset</b>	0x0000 05A8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	SMART1 I/O control 0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_DR0_SC		CAM_DR0_SC		RESERVED				GPIO_DR8_SC	GPIO_DR9_SC		RESERVED	GPMC_DR3_SC		RESERVED	GPIO_63_64_DR0_SC	MCSP11_DR0_SC		UART1_DR0_SC		UART3_DR0_SC	UART3_DR1_SC		RESERVED								

Bits	Field Name	Description	Type	Reset
31:30	ABE_DR0_SC	Slew rate control for group abe_dr0. Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
29:28	CAM_DR0_SC	Slew rate control for group cam_dr0. Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
27:24	RESERVED	Reserved	RW	0x0
23:22	GPIO_DR8_SC	Slew rate control for group gpio_dr8. Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0

Bits	Field Name	Description	Type	Reset
21:20	GPIO_DR9_SC	Slew rate control for group gpio_dr9. Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
19:18	RESERVED	Reserved	RW	0x0
17:16	GPMC_DR3_SC	Slew rate control for group gpmc_dr3. Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x1
15:14	RESERVED	Reserved	RW	0x0
13:12	GPIO_63_64_DR0_SC	Slew rate control for group gpio_63_64_dr0. Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
11:10	MCSP11_DR0_SC	Slew rate control for group mcspi1_dr0 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
9:8	UART1_DR0_SC	Slew rate control for group uart1_dr0 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
7:6	UART3_DR0_SC	Slew rate control for group uart3_dr0 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
5:4	UART3_DR1_SC	Slew rate control for group uart3_dr1 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
3:0	RESERVED	Reserved	RW	0x0

**Table 18-545. Register Call Summary for Register CONTROL\_SMART1IO\_PADCONF\_0**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[11\]](#)

**Table 18-546. CONTROL\_SMART1IO\_PADCONF\_1**

<b>Address Offset</b>	0x0000 05AC	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	SMART1 I/O control 1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABE_DR0_LB		CAM_DR0_LB		RESERVED				GPIO_DR8_LB		GPIO_DR9_LB		RESERVED		GPMC_DR3_LB		RESERVED		GPIO_63_64_DR0_LB		MCSPI1_DR0_LB		UART1_DR0_LB		UART3_DR0_LB		UART3_DR1_LB		RESERVED			

Bits	Field Name	Description	Type	Reset
31:30	ABE_DR0_LB	Load control for group abe_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
29:28	CAM_DR0_LB	Load control for group cam_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
27:24	RESERVED	Reserved	RW	0x0
23:22	GPIO_DR8_LB	Load control for group gpio_dr8 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
21:20	GPIO_DR9_LB	Load control for group gpio_dr9 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
19:18	RESERVED	Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
17:16	GPMC_DR3_LB	Load control for group gpmc_dr3 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
15:14	RESERVED	Reserved	RW	0x0
13:12	GPIO_63_64_DR0_LB	Load control for group gpio_63_64_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
11:10	MCSP11_DR0_LB	Load control for group mcspi1_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
9:8	UART1_DR0_LB	Load control for group uart1_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
7:6	UART3_DR0_LB	Load control for group uart3_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
5:4	UART3_DR1_LB	Load control for group uart3_dr1 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
3:0	RESERVED	Reserved	RW	0x0

**Table 18-547. Register Call Summary for Register CONTROL\_SMART1IO\_PADCONF\_1**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[9\]](#)

**Table 18-548. CONTROL\_SMART2IO\_PADCONF\_0**

<b>Address Offset</b>	0x0000 05B0	
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b> SYSCTRL_PADCONF_CORE
<b>Description</b>	SMART2 I/O control 0 Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED		RW	0

**Table 18-549. Register Call Summary for Register CONTROL\_SMART2IO\_PADCONF\_0**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[1\]](#)

**Table 18-550. CONTROL\_SMART2IO\_PADCONF\_1**

<b>Address Offset</b>	0x0000 05B4	
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b> SYSCTRL_PADCONF_CORE
<b>Description</b>	SMART2 I/O control 1 Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED		R	0x0000 0000

**Table 18-551. Register Call Summary for Register CONTROL\_SMART2IO\_PADCONF\_1**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)



**Table 18-552. CONTROL\_SMART3IO\_PADCONF\_0**

<b>Address Offset</b>	0x0000 05B8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	SMART3 I/O control 0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMIC_DR0_MB		GPIO_DR3_MB		GPIO_DR4_MB		GPIO_DR5_MB		GPIO_DR6_MB		HSI_DR1_MB		HSI_DR2_MB		HSI_DR3_MB		MCBSP2_DR0_MB		MCSP14_DR0_MB		MCSP14_DR1_MB		SDMMC3_DR0_MB		RESERVED					SPI2_DR0_MB		

Bits	Field Name	Description	Type	Reset
31:30	DMIC_DR0_MB	50-Ω output buffer mode control for group dmic_dr0. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x2
29:28	GPIO_DR3_MB	25-Ω output buffer mode control for group gpio_dr3. Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x3
27:26	GPIO_DR4_MB	25-Ω output buffer mode control for group gpio_dr4. Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x3
25:24	GPIO_DR5_MB	50-Ω output buffer mode control for group gpio_dr5. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
23:22	GPIO_DR6_MB	50-Ω output buffer mode control for group gpio_dr6. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
21:20	HSI_DR1_MB	25-Ω output buffer mode control for group hsi_dr1. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x3
19:18	HSI_DR2_MB	25-Ω output buffer mode control for group hsi_dr2. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x3
17:16	HSI_DR3_MB	25-Ω output buffer mode control for group hsi_dr3. Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x3
15:14	MCBSP2_DR0_MB	50-Ω output buffer mode control for group mcbbsp2_dr0. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
13:12	MCSP14_DR0_MB	50-Ω output buffer mode control for group mcspi4_dr0. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
11:10	MCSP14_DR1_MB	50-Ω output buffer mode control for group mcspi4_dr1. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1

Bits	Field Name	Description	Type	Reset
9:8	SDMMC3_DR0_MB	50-Ω output buffer mode control for group sdmmc3_dr0. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
7:2	RESERVED		R	0x00
1:0	SPI2_DR0_MB	50-Ω output buffer mode control for group spi2_dr0. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1

**Table 18-553. Register Call Summary for Register CONTROL\_SMART3IO\_PADCONF\_0**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[14\]](#)

**Table 18-554. CONTROL\_SMART3IO\_PADCONF\_1**

<b>Address Offset</b>	0x0000 05BC	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	SMART3 I/O control 1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
SPI2_DR1_MB		SPI2_DR2_MB		UART2_DR0_MB		UART2_DR1_MB		UART4_DR0_MB		HSI_DR0_MB		RESERVED					GPMC_DR1_MB		RESERVED					PDM_DR0_MB		FREF_DR2_MB		FREF_DR3_MB		USB1_DR2_MB			

Bits	Field Name	Description	Type	Reset
31:30	SPI2_DR1_MB	50-Ω output buffer mode control for group spi2_dr1. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
29:28	SPI2_DR2_MB	50-Ω output buffer mode control for group spi2_dr1. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
27:26	UART2_DR0_MB	50-Ω output buffer mode control for group spi2_dr1. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
25:24	UART2_DR1_MB	50-Ω output buffer mode control for group spi2_dr1. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
23:22	UART4_DR0_MB	50-Ω output buffer mode control for group spi2_dr1. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
21:20	HSI_DR0_MB	25-Ω output buffer mode control for group hsi_dr0. Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x3

Bits	Field Name	Description	Type	Reset
19:14	RESERVED		R	0x00
13:12	GPMC_DR1_MB	50-Ω output buffer mode control for group gpmc_dr1. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x3
11:8	RESERVED		RW	0x0
7:6	PDM_DR0_MB	50-Ω output buffer mode control for group pdm_dr0. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
5:4	FREF_DR2_MB	50-Ω output buffer mode control for group fref_dr2. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
3:2	FREF_DR3_MB	50-Ω output buffer mode control for group fref_dr3. Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
1:0	USBB1_DR2_MB	25-Ω output buffer mode control for group usbb1_dr2. Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x3

**Table 18-555. Register Call Summary for Register CONTROL\_SMART3IO\_PADCONF\_1**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[12\]](#)

**Table 18-556. CONTROL\_SMART3IO\_PADCONF\_2**

<b>Address Offset</b>	0x0000 05C0	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	SMART3 I/O control 2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMIC_DR0_LB	GPIO_DR3_LB	GPIO_DR4_LB	GPIO_DR5_LB	GPIO_DR6_LB	HSI_DR1_LB	HSI_DR2_LB	HSI_DR3_LB	MCBSP2_DR0_LB	MCSPI4_DR0_LB	MCSPI4_DR1_LB	RESERVED	SLIMBUS2_DR0_LB	RESERVED	SPI2_DR0_LB	SPI2_DR1_LB	SPI2_DR2_LB	UART2_DR0_LB	UART2_DR1_LB	UART4_DR0_LB	HSI_DR0_LB	FREF_DR3_LB	GPMC_DR1_LB	PDM_DR0_LB	FREF_DR2_LB	SDMMC3_DR0_LB	USBB1_DR2_LB	RESERVED				

Bits	Field Name	Description	Type	Reset
31	DMIC_DR0_LB	50-Ω output buffer load control dmic_dr0_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
30	GPIO_DR3_LB	25-Ω output buffer load control gpio_dr3_lb Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	1

Bits	Field Name	Description	Type	Reset
29	GPIO_DR4_LB	25-Ω output buffer load control gpio_dr4_lb Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	1
28	GPIO_DR5_LB	50-Ω output buffer load control gpio_dr5_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
27	GPIO_DR6_LB	50-Ω output buffer load control gpio_dr6_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
26	HSI_DR1_LB	25-Ω output buffer load control hsi_dr1_lb Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	1
25	HSI_DR2_LB	25-Ω output buffer load control hsi_dr2_lb Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	1
24	HSI_DR3_LB	25-Ω output buffer load control hsi_dr3_lb Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	1
23	MCBSP2_DR0_LB	50-Ω output buffer load control mcbbsp2_dr0_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
22	MCSPi4_DR0_LB	50-Ω output buffer load control mcspi4_dr0_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
21	MCSPi4_DR1_LB	50-Ω output buffer load control mcspi4_dr1_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
20: 17	RESERVED		R	0x0
16	SPI2_DR0_LB	50-Ω output buffer load control spi2_dr0_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
15	SPI2_DR1_LB	50-Ω output buffer load control spi2_dr1_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
14	SPI2_DR2_LB	50-Ω output buffer load control spi2_dr2_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
13	UART2_DR0_LB	50-Ω output buffer load control uart2_dr0_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
12	UART2_DR1_LB	50-Ω output buffer load control uart2_dr1_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
11	UART4_DR0_LB	50-Ω output buffer load control uart4_dr0_lb Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0

Bits	Field Name	Description	Type	Reset
10	HSI_DR0_LB	25-Ω output buffer load control hsi_dr0 Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	1
9	FREF_DR3_LB	50-Ω output buffer load control fref_dr3 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
8	GPMC_DR1_LB	50-Ω output buffer load control gpmc_dr1 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	1
7	PDM_DR0_LB	50-Ω output buffer load control pdm_dr0 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
6	FREF_DR2_LB	50-Ω output buffer load control fref_dr2 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
5	SDMMC3_DR0_LB	50-Ω output buffer load control sdmmc3_dr0 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
4	USBB1_DR2_LB	25-Ω output buffer load control usbb1_dr2 Refer to I/Os with combined Mode and Load Settings for more details on mode settings.	RW	1
3:0	RESERVED		R	0x0

**Table 18-557. Register Call Summary for Register CONTROL\_SMART3IO\_PADCONF\_2**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[26\]](#)

**Table 18-558. CONTROL\_USB\_HSIC**

<b>Address Offset</b>	0x0000 05C4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	USBB HSIC control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
USBB2_DR1_SR		USBB2_DR1_I		USBB1_DR1_SR		USBB1_DR1_I		USBB1_HSIC_DATA_WD		USBB1_HSIC_STROBE_WD		USBB2_HSIC_DATA_WD		USBB2_HSIC_STROBE_WD		USBB1_HSIC_DATA_OFFMODE_WD		USBB1_HSIC_STROBE_OFFMODE_WD		USBB1_HSIC_DATA_OFFMODE_WD		USBB1_HSIC_STROBE_OFFMODE_WD		USBB2_HSIC_DATA_OFFMODE_WD		USBB2_HSIC_STROBE_OFFMODE_WD		USBB2_HSIC_DATA_OFFMODE_WD		USBB2_HSIC_STROBE_OFFMODE_WD		RESERVED	

Bits	Field Name	Description	Type	Reset
31:30	USBB2_DR1_SR	usbb2 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x0
29:27	USBB2_DR1_I	usbb2 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x6
26:25	USBB1_DR1_SR	usbb1 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x0
24:22	USBB1_DR1_I	usbb1 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x6

Bits	Field Name	Description	Type	Reset
21:20	USBB1_HSIC_DATA_WD	usbb1_hsic_data wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
19:18	USBB1_HSIC_STROBE_WD	usbb1_hsic_strobe wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
17:16	USBB2_HSIC_DATA_WD	usbb2_hsic_data wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
15:14	USBB2_HSIC_STROBE_WD	usbb2_hsic_strobe wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
13	USBB1_HSIC_DATA_OFFMODE_WD_ENABLE	usbb1_hsic_data_offmode_wd enable 0x0: offmode_wd1 0x1: offmode_wd1	RW	0
12:11	USBB1_HSIC_DATA_OFFMODE_WD	usbb1_hsic_data_offmode_wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x0
10	USBB1_HSIC_STROBE_OFFMODE_WD_ENABLE	usbb1_hsic_strobe_offmode_wd enable 0x0: offmode_wd1 0x1: offmode_wd1	RW	0
9:8	USBB1_HSIC_STROBE_OFFMODE_WD	usbb1_hsic_strobe_offmode_wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x0
7	USBB2_HSIC_DATA_OFFMODE_WD_ENABLE	usbb2_hsic_data_offmode_wd enable 0x0: offmode_wd1 0x1: offmode_wd1	RW	0
6:5	USBB2_HSIC_DATA_OFFMODE_WD	usbb2_hsic_data_offmode_wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x0
4	USBB2_HSIC_STROBE_OFFMODE_WD_ENABLE	usbb2_hsic_strobe_offmode_wd enable 0x0: offmode_wd1 0x1: offmode_wd1	RW	0

Bits	Field Name	Description	Type	Reset
3:2	USBB2_HSIC_STROBE_OFFMODE_WD	usbb2_hsic_strobe_offmode_wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x0
1:0	RESERVED		R	0x0

**Table 18-559. Register Call Summary for Register CONTROL\_USBB\_HSIC**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[1\]](#)

**Table 18-560. CONTROL\_SMART3IO\_PADCONF\_3**

<b>Address Offset</b>	0x0000 05C8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>		
<b>Description</b>	SMART3 IO control 3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLIMBUS1_DR0_MB	SLIMBUS1_DR1_MB	SLIMBUS2_DR0_MB	SLIMBUS2_DR1_MB	SLIMBUS2_DR2_MB	SLIMBUS2_DR3_MB	SLIMBUS1_DR0_LB	SLIMBUS2_DR1_LB	SLIMBUS2_DR2_LB	SLIMBUS2_DR3_LB	SLIMBUS1_DR1_LB	SLIMBUS2_DR0_LB	RESERVED																			

Bits	Field Name	Description	Type	Reset
31:30	SLIMBUS1_DR0_MB	50-Ω output buffer mode control for group slimbus1_dr0 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
29:28	SLIMBUS1_DR1_MB	50-Ω output buffer mode control for group slimbus1_dr1 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
27:26	SLIMBUS2_DR0_MB	50-Ω output buffer mode control for group slimbus2_dr0 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
25:24	SLIMBUS2_DR1_MB	50-Ω output buffer mode control for group slimbus2_dr1 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
23:22	SLIMBUS2_DR2_MB	50-Ω output buffer mode control for group slimbus2_dr2 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
21:20	SLIMBUS2_DR3_MB	50-Ω output buffer mode control for group slimbus2_dr3 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1



Bits	Field Name	Description	Type	Reset
19	SLIMBUS1_DR0_LB	50-Ω output buffer load control for group slimbus1_dr0 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
18	SLIMBUS2_DR1_LB	50-Ω output buffer load control for group slimbus2_dr1 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
17	SLIMBUS2_DR2_LB	50-Ω output buffer load control for group slimbus2_dr2 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
16	SLIMBUS2_DR3_LB	50-Ω output buffer load control for group slimbus2_dr3 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
15	SLIMBUS1_DR1_LB	50-Ω output buffer load control for group slimbus1_dr1 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
14	SLIMBUS2_DR0_LB	50-Ω output buffer load control for group slimbus2_dr0 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
13:0	RESERVED		R	0x0000

**Table 18-561. Register Call Summary for Register CONTROL\_SMART3IO\_PADCONF\_3**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [50-Ω Output Buffer I/Os with Combined Mode and Load Settings: \[1\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[14\]](#)

**Table 18-562. CONTROL\_SMART2IO\_PADCONF\_2**

<b>Address Offset</b>	0x0000 05CC
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a> <b>Instance</b> SYSCTRL_PADCONF_CORE
<b>Description</b>	SMART2 IO control 2 Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
DPM_DR1_DS	DPM_DR2_DS	DPM_DR3_DS	GPIO_DR10_DS	HSI2_DR0_DS	HSI2_DR1_DS	HSI2_DR2_DS	SDMMC3_DR0_DS	SDMMC4_DR0_DS	SDMMC4_DR1_DS	SPI3_DR0_DS	SPI3_DR1_DS	UART3_DR2_DS	UART3_DR3_DS	UART3_DR4_DS	UART3_DR5_DS	USB_A0_DR0_DS	USB_A0_DR1_DS	USB_A_DR2_DS	USB_B2_DR0_DS	USB_B1_DR0_DS	RESERVED															

Bits	Field Name	Description	Type	Reset
31	DPM_DR1_DS	Drive strength control bit for group dpm_dr1 0x0: 50 Ω 0x1: 25 Ω	RW	0

Bits	Field Name	Description	Type	Reset
30	DPM_DR2_DS	Drive strength control bit for group dpm_dr2 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
29	DPM_DR3_DS	Drive strength control bit for group dpm_dr3 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
28	GPIO_DR10_DS	Drive strength control bit for group gpio_dr10 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
27	HSI2_DR0_DS	Drive strength control bit for group hsi2_dr0 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
26	HSI2_DR1_DS	Drive strength control bit for group hsi2_dr1 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
25	HSI2_DR2_DS	Drive strength control bit for group hsi2_dr2 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
24	SDMMC3_DR0_DS	Drive strength control bit for group sdmmc3_dr0 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
23	SDMMC4_DR0_DS	Drive strength control bit for group sdmmc4_dr0 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
22	SDMMC4_DR1_DS	Drive strength control bit for group sdmmc4_dr1 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
21	SPI3_DR0_DS	Drive strength control bit for group spi3_dr0 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
20	SPI3_DR1_DS	Drive strength control bit for group spi3_dr1 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
19	UART3_DR2_DS	Drive strength control bit for group uart3_dr2 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
18	UART3_DR3_DS	Drive strength control bit for group uart3_dr3 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
17	UART3_DR4_DS	Drive strength control bit for group uart3_dr4 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
16	UART3_DR5_DS	Drive strength control bit for group uart3_dr5 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0
15	USBA0_DR0_DS	Drive strength control bit for group usba0_dr0 0x0: 50 $\Omega$ 0x1: 25 $\Omega$	RW	0

Bits	Field Name	Description	Type	Reset
14	USBA0_DR1_DS	Drive strength control bit for group usba0_dr1 0x0: 50 Ω 0x1: 25 Ω	RW	0
13	USBA_DR2_DS	Drive strength control bit for group usba_dr2 0x0: 50 Ω 0x1: 25 Ω	RW	0
12	USBB2_DR0_DS	Drive strength control bit for group usbb2_dr0 0x0: 50 Ω 0x1: 25 Ω	RW	0
11	USBB1_DR0_DS	Drive strength control bit for group usbb1_dr0 0x0: 50 Ω 0x1: 25 Ω	RW	0
10:0	RESERVED		R	0x000

**Table 18-563. Register Call Summary for Register CONTROL\_SMART2IO\_PADCONF\_2**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [I/O cells with Configurable Output Driver Impedance: \[1\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[23\]](#)

**Table 18-564. CONTROL\_SMART1IO\_PADCONF\_2**

<b>Address Offset</b>	0x0000 05D0
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a> <b>Instance</b> SYSCTRL_PADCONF_CORE
<b>Description</b>	SMART1 IO control 2 Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
KPD_DR0_SC	KPD_DR1_SC	KPD_DR2_SC	KPD_DR3_SC	HDQ_DR0_SC	RESERVED																																	

Bits	Field Name	Description	Type	Reset
31:30	KPD_DR0_SC	Slew rate control for group kpd_dr0 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
29:28	KPD_DR1_SC	Slew rate control for group kpd_dr1 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
27:26	KPD_DR2_SC	Slew rate control for group kpd_dr2 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0

Bits	Field Name	Description	Type	Reset
25:24	KPD_DR3_SC	Slew rate control for group kpd_dr3 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
23:22	HDQ_DR0_SC	Slew rate control for group hdq_dr0 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
21:0	RESERVED		R	0x000000

**Table 18-565. Register Call Summary for Register CONTROL\_SMART1IO\_PADCONF\_2**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[6\]](#)

**Table 18-566. CONTROL\_SMART1IO\_PADCONF\_3**

<b>Address Offset</b>	0x0000 05D4	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>		
<b>Description</b>	SMART1 IO control 3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
KPD_DR0_LB	KPD_DR1_LB	KPD_DR2_LB	KPD_DR3_LB	HDQ_DR0_LB	RESERVED																																

Bits	Field Name	Description	Type	Reset
31:30	KPD_DR0_LB	Load control for group kpd_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
29:28	KPD_DR1_LB	Load control for group kpd_dr1 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0

Bits	Field Name	Description	Type	Reset
27:26	KPD_DR2_LB	Load control for group kpd_dr2 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
25:24	KPD_DR3_LB	Load control for group kpd_dr3 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
23:22	HDQ_DR0_LB	Load control for group hdq_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
21:0	RESERVED		R	0x000000

**Table 18-567. Register Call Summary for Register CONTROL\_SMART110\_PADCONF\_3**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[6\]](#)

**Table 18-568. CONTROL\_C2CIO\_PADCONF\_0**

<b>Address Offset</b>	0x0000 05D8	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>		
<b>Description</b>	C2C IO control 0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2C_DR0_LB0	C2C_DR1_LB0	C2C_DR2_LB0	GPIO_DR0_LB0	GPIO_DR1_LB0	GPIO_DR2_LB0	GPMC_DR0_LB0	GPMC_DR11_LB0	GPMC_DR10_LB0	GPMC_DR2_LB0	GPMC_DR4_LB0	GPMC_DR5_LB0	GPMC_DR6_LB0	GPMC_DR7_LB0	GPMC_DR8_LB0	GPMC_DR9_LB0	KPD_DR4_LB0	KPD_DR5_LB0	SDMMC2_DR0_LB0	CMOSEN_C2C_0_FRM_CTRL	CMOSEN_C2C_1_FRM_CTRL	C2C_VREF_CCAP	C2C_INT_VREF_EN	C2C_INT_VREF_AUTO_EN	RESERVED							

Bits	Field Name	Description	Type	Reset
31	C2C_DR0_LB0	Mode selection bit for group c2c_dr0_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
30	C2C_DR1_LB0	Mode selection bit for group c2c_dr1_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
29	C2C_DR2_LB0	Mode selection bit for group c2c_dr2_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
28	GPIO_DR0_LB0	Mode selection bit for group gpio_dr0_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
27	GPIO_DR1_LB0	Mode selection bit for group gpio_dr1_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
26	GPIO_DR2_LB0	Mode selection bit for group gpio_dr2_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
25	GPMC_DR0_LB0	Mode selection bit for group gpmc_dr0_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
24	GPMC_DR11_LB0	Mode selection bit for group gpmc_dr11_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
23	GPMC_DR10_LB0	Mode selection bit for group gpmc_dr10_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
22	GPMC_DR2_LB0	Mode selection bit for group gpmc_dr2_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
21	GPMC_DR4_LB0	Mode selection bit for group gpmc_dr4_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
20	GPMC_DR5_LB0	Mode selection bit for group gpmc_dr5_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
19	GPMC_DR6_LB0	Mode selection bit for group gpmc_dr6_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
18	GPMC_DR7_LB0	Mode selection bit for group gpmc_dr7_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
17	GPMC_DR8_LB0	Mode selection bit for group gpmc_dr8_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
16	GPMC_DR9_LB0	Mode selection bit for group gpmc_dr9_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0

Bits	Field Name	Description	Type	Reset
15	KPD_DR4_LB0	Mode selection bit for group kpd_dr4_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
14	KPD_DR5_LB0	Mode selection bit for group kpd_dr5_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
13	SDMMC2_DR0_LB0	Mode selection bit for group sdmmc2_dr0_LB0 0x0: sdmmc2 1.2V and 1.8V / gpmc 1.2V 0x1: c2c 1.2V and 1.8V / gpmc 1.8V	RW	0
12	CMOSEN_C2C_0_FRM_CTRL	lvcmos receiver enable in 1.2V mode for group c2c_0 0x0: lvcmos rx disable 0x1: lvcmos rx enable	RW	1
11	CMOSEN_C2C_1_FRM_CTRL	lvcmos receiver enable in 1.2V mode for group c2c_1 0x0: lvcmos rx disable 0x1: lvcmos rx enable	RW	1
10:9	C2C_VREF_CCAP	Selection for coupling cap connection on ivref pad 0x0: No capacitor connected 0x1: Capacitor between BIAS2 and VSS 0x3: Capacitor between BIAS2 and VDDS as well as BIAS2 and VSS 0x2: Capacitor between BIAS2 and VDDS	RW	0x1
8	C2C_INT_VREF_EN	Internal VREF enable for C2C pads in manual mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1
7	C2C_INT_VREF_AUTO_EN	Internal VREF enable for C2C pads in auto mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1
6:0	RESERVED		R	0x00

**Table 18-569. Register Call Summary for Register CONTROL\_C2CIO\_PADCONF\_0**

## Control Module Functional Description

- [Device Core Control Module Instance: \[0\]](#)
- [Signal Integrity Parameter Controls Overview: \[1\]](#)
- [C2CI/O cells with load setting adjustable according to interface requirement / voltage mode: \[2\] \[3\] \[4\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[24\]](#)

**Table 18-570. CONTROL\_PBIASLITE**

<b>Address Offset</b>	0x0000 0600		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	PBIASLITE control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBIASLITE1_HIZ_MODE	PBIASLITE1_SUPPLY_HI_OUT	PBIASLITE1_VMODE_ERROR	PBIASLITE1_PWRDNZ	PBIASLITE1_VMODE	MMC1_PWRDNZ	MMC1_PBIASLITE_HIZ_MODE	MMC1_PBIASLITE_SUPPLY_HI_OUT	MMC1_PBIASLITE_VMODE_ERROR	MMC1_PBIASLITE_PWRDNZ	MMC1_PBIASLITE_VMODE	USBC1_ICUSB_PWRDNZ	RESERVED																			

Bits	Field Name	Description	Type	Reset
31	PBIASLITE1_HIZ_MODE	HIZ MODE of the PBIASLITE1 0x0: PBIAS1 is in normal operation mode. 0x1: PBIAS1 output is in high-impedance state.	RW	0
30	PBIASLITE1_SUPPLY_HI_OUT	SUPPLY_HI_OUT from PBIASLITE1 Read 0x0: SIM_VDDS = 1.8 V Read 0x1: SIM_VDDS = 3 V	R	0
29	PBIASLITE1_VMODE_ERROR	VMODE ERROR from PBIASLITE1 Read 0x0: VMODE level is the same as SUPPLY_HI_OUT. Read 0x1: VMODE level is not the same as SUPPLY_HI_OUT.	R	0
28	PBIASLITE1_PWRDNZ	PWRDNZ control to PBIASLITE1 0x0: Software must keep this bit enabled whenever SIM_VDDS ramps up/down. 0x1: Software can keep this bit disabled when SIM_VDDS is stable.	RW	0
27	PBIASLITE1_VMODE	VMODE control to PBIASLITE1 0x0: SIM_VDDS = 1.8 V 0x1: SIM_VDDS = 3 V	RW	1
26	MMC1_PWRDNZ	PWRDNZ control to MMC1 IO 0x0: This signal is used to protect MMC IOs when MMC1_VDDS is not stable 0x1: SW keep this bit to 1'b1 after MMC1_VDDS stabilizing	RW	0
25	MMC1_PBIASLITE_HIZ_MODE	HIZ_MODE from MMC1 PBIASLITE 0x0: PBIAS in normal operation mode 0x1: PBIAS output is in high impedance state	RW	0
24	MMC1_PBIASLITE_SUPPLY_HI_OUT	SUPPLY_HI_OUT from MMC1 PBIASLITE Read 0x0: MMC1_VDDS = 1.8V Read 0x1: MMC1_VDDS = 3V	R	0



Bits	Field Name	Description	Type	Reset
23	MMC1_PBIASLITE_VMODE_ERROR	VMODE ERROR from MMC1 PBIASLITE Read 0x0: VMODE level is same as SUPPLY_HI_OUT Read 0x1: VMODE level is not same as SUPPLY_HI_OUT	R	0
22	MMC1_PBIASLITE_PWRDNZ	PWRDNZ control to MMC1 PBIASLITE 0x0: SW has to keep this bit enabled whenever MMC1_VDDS is ramping up/down 0x1: SW can keep this bit disabled when MMC1_VDDS is stable	RW	0
21	MMC1_PBIASLITE_VMODE	VMODE control to MMC1 PBIASLITE 0x0: MMC1_VDDS = 1.8V 0x1: MMC1_VDDS = 3V	RW	1
20	USBC1_ICUSB_PWRDNZ	PWRDNZ control to USBC1 IO 0x0: This signal is used to protect MMC IOs when MMC1_VDDS is not stable 0x1: SW keep this bit to 1'b1 after MMC1_VDDS stabilizing	RW	0
19:0	RESERVED		R	0x00000

**Table 18-571. Register Call Summary for Register CONTROL\_PBIASLITE**

Control Module Functional Description

- [Extended-Drain I/O and PBIAS Cell: \[0\] \[1\]](#)
- [PBIAS Cell: \[2\] \[3\]](#)
- [Extended-Drain I/O: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [PBIAS Error Generation: \[19\] \[20\] \[21\]](#)
- [PBIASLITE Control Register: \[22\] \[23\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[24\]](#)

Control Module Programming Guide

- [Extended-Drain I/Os and PBIAS Cells Programming Guide: \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\] \[44\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[45\]](#)

**Table 18-572. CONTROL\_I2C\_0**

<b>Address Offset</b>	0x0000 0604	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	I2C pads control 0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C4_SDA_GLFENB	I2C4_SDA_LOAD_BITS	I2C4_SDA_PULLUPRESX	I2C3_SDA_GLFENB	I2C3_SDA_LOAD_BITS	I2C3_SDA_PULLUPRESX	I2C2_SDA_GLFENB	I2C2_SDA_LOAD_BITS	I2C2_SDA_PULLUPRESX	I2C1_SDA_GLFENB	I2C1_SDA_LOAD_BITS	I2C1_SDA_PULLUPRESX	I2C4_SCL_GLFENB	I2C4_SCL_LOAD_BITS	I2C4_SCL_PULLUPRESX	I2C3_SCL_GLFENB	I2C3_SCL_LOAD_BITS	I2C3_SCL_PULLUPRESX	I2C2_SCL_GLFENB	I2C2_SCL_LOAD_BITS	I2C2_SCL_PULLUPRESX	I2C1_SCL_GLFENB	I2C1_SCL_LOAD_BITS	I2C1_SCL_PULLUPRESX								

Bits	Field Name	Description	Type	Reset
31	I2C4_SDA_GLFENB	Active_high glitch free operation enable pin for i2c4 receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0
30:29	I2C4_SDA_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for i2c4 0x0: 4.5 k $\Omega$ (5-15 pF) / 1.66 k $\Omega$ (5-12 pF) 0x1: 2.1 k $\Omega$ (15-50 pF) / 920 $\Omega$ (12-25 pF) 0x2: 860 $\Omega$ (50-150 pF) / 500 $\Omega$ (25-50 pF) 0x3: N.A / 300 $\Omega$ (50-80 pF)	RW	0x1
28	I2C4_SDA_PULLUPRESX	Active_low internal pull_up resistor enabled for i2c4 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resistor	RW	0
27	I2C3_SDA_GLFENB	Active_high glitch free operation enable pin for i2c3 receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0
26:25	I2C3_SDA_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for i2c3 0x0: 4.5 k $\Omega$ (5-15 pF) / 1.66 k $\Omega$ (5-12 pF) 0x1: 2.1 k $\Omega$ (15-50 pF) / 920 $\Omega$ (12-25 pF) 0x2: 860 $\Omega$ (50-150 pF) / 500 $\Omega$ (25-50 pF) 0x3: N.A / 300 $\Omega$ (50-80 pF)	RW	0x1
24	I2C3_SDA_PULLUPRESX	Active_low internal pull_up resistor enabled for i2c3 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resistor	RW	0
23	I2C2_SDA_GLFENB	Active_high glitch free operation enable pin for i2c2 receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0
22:21	I2C2_SDA_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for i2c2 0x0: 4.5 k $\Omega$ (5-15 pF) / 1.66 k $\Omega$ (5-12 pF) 0x1: 2.1 k $\Omega$ (15-50 pF) / 920 $\Omega$ (12-25 pF) 0x2: 860 $\Omega$ (50-150 pF) / 500 $\Omega$ (25-50 pF) 0x3: N.A / 300 $\Omega$ (50-80 pF)	RW	0x1
20	I2C2_SDA_PULLUPRESX	Active_low internal pull_up resistor enabled for i2c2 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resistor	RW	0
19	I2C1_SDA_GLFENB	Active_high glitch free operation enable pin for i2c1 receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0
18:17	I2C1_SDA_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for i2c1 0x0: 4.5 k $\Omega$ (5-15 pF) / 1.66 k $\Omega$ (5-12 pF) 0x1: 2.1 k $\Omega$ (15-50 pF) / 920 $\Omega$ (12-25 pF) 0x2: 860 $\Omega$ (50-150 pF) / 500 $\Omega$ (25-50 pF) 0x3: N.A / 300 $\Omega$ (50-80 pF)	RW	0x1
16	I2C1_SDA_PULLUPRESX	Active_low internal pull_up resistor enabled for i2c1 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resistor	RW	0

Bits	Field Name	Description	Type	Reset
15	I2C4_SCL_GLFENB	Active_high glitch free operation enable pin for i2c4 receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0
14:13	I2C4_SCL_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for i2c4 0x0: 4.5 kΩ (5-15 pF) / 1.66 kΩ (5-12 pF) 0x1: 2.1 kΩ (15-50 pF) / 920 Ω (12-25 pF) 0x2: 860 Ω (50-150 pF) / 500 Ω (25-50 pF) 0x3: N.A / 300 Ω (50-80 pF)	RW	0x1
12	I2C4_SCL_PULLUPRESX	Active_low internal pull_up resistor enabled for i2c4 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resistor	RW	0
11	I2C3_SCL_GLFENB	Active_high glitch free operation enable pin for i2c3 receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0
10:9	I2C3_SCL_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for i2c3 0x0: 4.5 kΩ (5-15 pF) / 1.66 kΩ (5-12 pF) 0x1: 2.1 kΩ (15-50 pF) / 920 Ω (12-25 pF) 0x2: 860 Ω (50-150 pF) / 500 Ω (25-50 pF) 0x3: N.A / 300 Ω (50-80 pF)	RW	0x1
8	I2C3_SCL_PULLUPRESX	Active_low internal pull_up resistor enabled for i2c3 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resistor	RW	0
7	I2C2_SCL_GLFENB	Active_high glitch free operation enable pin for i2c2 receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0
6:5	I2C2_SCL_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for i2c2 0x0: 4.5 kΩ (5-15 pF) / 1.66 kΩ (5-12 pF) 0x1: 2.1 kΩ (15-50 pF) / 920 Ω (12-25 pF) 0x2: 860 Ω (50-150 pF) / 500 Ω (25-50 pF) 0x3: N.A / 300 Ω (50-80 pF)	RW	0x1
4	I2C2_SCL_PULLUPRESX	Active_low internal pull_up resistor enabled for i2c2 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resistor	RW	0
3	I2C1_SCL_GLFENB	Active_high glitch free operation enable pin for i2c1 receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0
2:1	I2C1_SCL_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for i2c1 0x0: 4.5 kΩ (5-15 pF) / 1.66 kΩ (5-12 pF) 0x1: 2.1 kΩ (15-50 pF) / 920 Ω (12-25 pF) 0x2: 860 Ω (50-150 pF) / 500 Ω (25-50 pF) 0x3: N.A / 300 Ω (50-80 pF)	RW	0x1
0	I2C1_SCL_PULLUPRESX	Active_low internal pull_up resistor enabled for i2c1 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resistor	RW	0

**Table 18-573. Register Call Summary for Register CONTROL\_I2C\_0**

## Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[25\]](#)

**Table 18-574. CONTROL\_CAMERA\_RX**

<b>Address Offset</b>	0x0000 0608	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	CAMERA RX control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED	CAMERARX_CSI22_LANEENABLE		CAMERARX_CSI21_LANEENABLE					RESERVED	CAMERARX_CSI22_CTRLCLKEN	CAMERARX_CSI22_CAMMODE	CAMERARX_CSI21_CTRLCLKEN	CAMERARX_CSI21_CAMMODE																				

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30:29	CAMERARX_CSI22_LANEENABLE	CSI22 CAMERARX Lane Enable 0x0: Lane module disabled 0x1: Lane module enabled	RW	0x0
28:24	CAMERARX_CSI21_LANEENABLE	CSI21 CAMERARX Lane Enable 0x0: Lane module disabled 0x1: Lane module enabled	RW	0x00
23:22	RESERVED		R	0x0
21	CAMERARX_CSI22_CTRLCLKEN	CSI22 CAMERARX clock enable control 0x0: Disable for CTRLCLK 0x1: Active high enable for CTRLCLK	RW	0
20:19	CAMERARX_CSI22_CAMMODE	CSI22 CAMERARX CAMMODE control 0x0: DPHY mode 0x1: Data/Strobe Transmission Format 0x2: Data/Clock Transmission Format 0x3: GPI mode	RW	0x3
18	CAMERARX_CSI21_CTRLCLKEN	CSI21 CAMERARX clock enable control 0x0: Disable for CTRLCLK 0x1: Active high enable for CTRLCLK	RW	0

Bits	Field Name	Description	Type	Reset
17:16	CAMERARX_CSI21_CAMMODE	CSI21 CAMERARX CAMMODE control 0x0: DPHY mode 0x1: Data/Strobe Transmission Format 0x2: Data/Clock Transmission Format 0x3: GPI mode	RW	0x3
15:0	RESERVED		R	0x0000

**Table 18-575. Register Call Summary for Register CONTROL\_CAMERA\_RX**

Control Module Functional Description

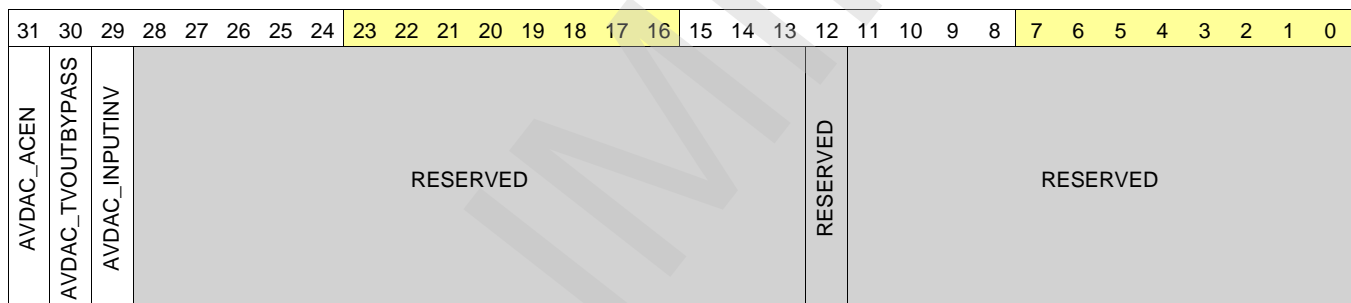
- [CSI Receiver Control Register: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-576. CONTROL\_AVDAC**

<b>Address Offset</b>	0x0000 060C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	AVDAC control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31	AVDAC_ACEN	AC coupling enable 0x0: AC coupling disable 0x1: AC coupling enable	RW	0
30	AVDAC_TVOUTBYPASS	TV OUT bypass signal 0x0: Video buffer is not bypassed 0x1: Video buffer is bypassed	RW	0
29	AVDAC_INPUTINV	Inversion control for DAC input data – din[9:0] 0x0: Data inversion is disabled. 0x1: Data inversion is enabled.	RW	0
28:13	RESERVED		RW	0x0000
12	RESERVED		RW (Write 1 to Clear)	0
11:0	RESERVED		R	0x000

**Table 18-577. Register Call Summary for Register CONTROL\_AVDAC**

Control Module Programming Guide

- [Video Driver Configuration: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[2\]](#)

**Table 18-578. CONTROL\_MMC2**

<b>Address Offset</b>	0x0000 0614	
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b> SYSCTRL_PADCONF_CORE
<b>Description</b>	MMC2 control Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31	MMC2_FEEDBACK_CLK_SEL	Feed_back clock select 0x0: Clock feed_back at PAD level 0x1: Clock feed_back at board level	RW	0
30:0	RESERVED		R	0x0000 0000

**Table 18-579. Register Call Summary for Register CONTROL\_MMC2**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-580. CONTROL\_DSIPHY**

<b>Address Offset</b>	0x0000 0618	
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b> SYSCTRL_PADCONF_CORE
<b>Description</b>	DSIPHY control Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSI2_LANEENABLE		DSI1_LANEENABLE		DSI1_PIPD				DSI2_PIPD				RESERVED																			

Bits	Field Name	Description	Type	Reset
31:29	DSI2_LANEENABLE	DSI2 Lane Enable 0x0: Lane module disabled 0x1: Lane module enabled	RW	0x0
28:24	DSI1_LANEENABLE	DSI1 Lane Enable 0x0: Lane module disabled 0x1: Lane module enabled	RW	0x00
23:19	DSI1_PIPD	DSI2 PD Enable 0x0: Pull down enabled 0x1: Pull down disabled	RW	0x00
18:14	DSI2_PIPD	DSI1 PD Enable 0x0: Pull down enabled 0x1: Pull down disabled	RW	0x00
13:0	RESERVED		R	0x0000

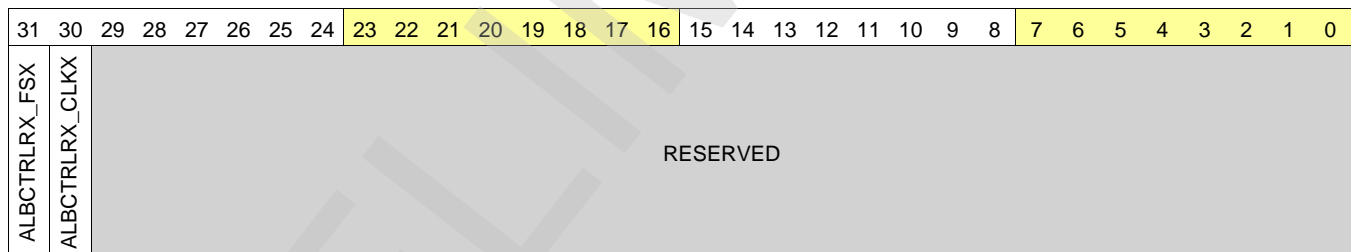
**Table 18-581. Register Call Summary for Register CONTROL\_DSIPHY**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-582. CONTROL\_MCBSPLP**

<b>Address Offset</b>	0x0000 061C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	McBSPLP control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31	ALBCTRLRX_FSX	Analog loop_back control for FSX 0x0: PIFSR is used 0x1: PIFSX is used instead of PIFSR 4pin mode	RW	0
30	ALBCTRLRX_CLKX	Analog loop_back control for CLKX 0x0: PICLKR is used 0x1: PICLKX is used instead of PICLKR 4pin mode	RW	0
29:0	RESERVED		R	0x0000 0000

**Table 18-583. Register Call Summary for Register CONTROL\_MCBSPLP**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-584. CONTROL\_USB2PHYCORE**

<b>Address Offset</b>	0x0000 0620		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	USB2PHYCORE control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
USB2PHY_AUTORESUME_EN	USB2PHY_DISCHGDET	USB2PHY_GPIOMODE	USB2PHY_CHG_DET_EXT_CTL	USB2PHY_RDM_PD_CHGDET_EN	USB2PHY_RDP_PU_CHGDET_EN	USB2PHY_CHG_VSRC_EN	USB2PHY_CHG_ISINK_EN	USB2PHY_CHG_DET_STATUS			USB2PHY_CHG_DET_DM_COMP	USB2PHY_CHG_DET_DP_COMP	USB2PHY_DATADET	USB2PHY_SINKONDP	USB2PHY_SRCONDM	USB2PHY_RESTARTCHGDET	USB2PHY_CHGDETDONE	USB2PHY_CHGDETECTED	USB2PHY_MCPCPUEN	USB2PHY_MCPCMODEEN	RESERVED	USB2PHY_UTMIRESETDONE	USB2PHY_TXBITSTUFFENABLE	USB2PHY_DATAPOLARITYN	USB2PHY_FREQLOCK	USB2PHY_RESETDONETCLK	RESERVED									

Bits	Field Name	Description	Type	Reset
31	USB2PHY_AUTORESUME_EN	Auto resume enable 0x0: Disable autoresume 0x1: Enable autoresume	RW	0
30	USB2PHY_DISCHGDET	Disable charger detect 0x0: Charger detect function enabled 0x1: Charger detect function disabled	RW	0x0
29	USB2PHY_GPIOMODE	GPIO mode 0x0: USB mode enabled 0x1: GPIO mode enabled	RW	0
28	USB2PHY_CHG_DET_EXT_CTL	Charge detect external control 0x0: Charger detect internal state-machine used 0x1: Charge detect state-machine is bypassed	RW	0
27	USB2PHY_RDM_PD_CHGDET_EN	DM Pull down control 0x0: PD disabled 0x1: PD enabled	RW	0
26	USB2PHY_RDP_PU_CHGDET_EN	DP Pull up control 0x0: PU disabled 0x1: PU enabled	RW	0
25	USB2PHY_CHG_VSRC_EN	VSRC enable on DP line-Host charger case 0x0: Disable VSRC drive on DP 0x1: Drives VSRC 600 mV on DP line	RW	0
24	USB2PHY_CHG_ISINK_EN	ISINK enable on DM line-Host charger case 0x0: Disable the ISINK on DM 0x1: Enables the ISINK (100 A) on DM line	RW	0



Bits	Field Name	Description	Type	Reset
23:21	USB2PHY_CHG_DET_STATUS	Status of charger detection Read 0x0: Wait state Read 0x1: No contact Read 0x2: PS/2 Read 0x3: Unknown error Read 0x4: Dedicated charger Read 0x5: HOST charger Read 0x6: PC Read 0x7: Interrupt	R	0x1
20	USB2PHY_CHG_DET_DM_COMP	Output of the comparator on DM during the resistor host detect protocol Read 0x0: DM line is below 0.75V to 0.95V Read 0x1: DM line is above 0.75V to 0.95V	R	0
19	USB2PHY_CHG_DET_DP_COMP	Output of the comparator on DP during the resistor host detect protocol Read 0x0: DP line is below 0.75V to 0.95V Read 0x1: DP line is above 0.75V to 0.95V	R	0X1
18	USB2PHY_DATADET	Output of the charger detect comparator Read 0x0: DM line is below 0.25V to 0.4V Read 0x1: DM line is above 0.25V to 0.4V	R	0
17	USB2PHY_SINKONDP	When 1 current sink is connected to DP instead of DM 0x0: Default value 0x1: Enables the ISINK on DP instead of DM	RW	0
16	USB2PHY_SRCONDM	When 1 voltage source is connected to DP instead of DM 0x0: Default value 0x1: Enable the VSRC on DM instead of DP	RW	0
15	USB2PHY_RESTARTCHGDET	restartchgdet = 1 for 1 msec cause the CD_START to reset 0x0: Default value 0x1: A high pulse of 1 msec causes the charger detect to restart on negative edge of restartchgdet	RW	0
14	USB2PHY_CHGDETDONE	Status indicates that charger detection protocol is over Read 0x0: Charger detection protocol is not over Read 0x1: Charger detection protocol is over	R	0
13	USB2PHY_CHGDETECTED	Output of the charger detection protocol Read 0x0: Charger not detected Read 0x1: Charger detected	R	0
12	USB2PHY_MCPCPUEN	MCPC Pull up enable 0x0: Disable the MCPC pull up 0x1: Enable the 4.7K to10K pull up on receive line DP when datapolarity is 0 and DM when datapolarity is 1	RW	0
11	USB2PHY_MCPCMODEEN	MCPC Mode enable 0x0: Disable MCPC mode 0x1: Enable MCPC mode	RW	0
10	RESERVED	Read returns reset value	R	0
9	USB2PHY_UTMIRESETDONE	UTMI FSM reset status Read 0x0: UTMI FSMs are in reset Read 0x1: UTMI FSMs are out of reset	R	0
8	USB2PHY_TXBITSTUFFENABLE	TX data bit stuff enable 0x0: DATAOUT [7] 0x1: DATAOUT [7]	RW	0

Bits	Field Name	Description	Type	Reset
7	USB2PHY_DATAPOLARITYN	Data polarity 0x0: DP functionality is on DP and DM functionality is on DM 0x1: DP functionality is on DM and DM functionality is on DP	RW	0
6	USBDPLL_FREQLOCK	Status from USB DPLL	R	0
5	USB2PHY_RESETDONETCLK	resetdonetclk status from USB2PHY	R	0
4:0	RESERVED		R	0x00

**Table 18-585. Register Call Summary for Register CONTROL\_USB2PHYCORE**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-586. CONTROL\_MMC1**

<b>Address Offset</b>	0x0000 0628	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	MMC1 control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMMC1_PUSTRENGTH_GRP0	SDMMC1_PUSTRENGTH_GRP1	SDMMC1_PUSTRENGTH_GRP2	SDMMC1_PUSTRENGTH_GRP3	SDMMC1_DR0_SPEEDCTRL	SDMMC1_DR1_SPEEDCTRL	SDMMC1_DR2_SPEEDCTRL	USBC1_DR0_SPEEDCTRL	USB_FD_CDEN	USBC1_ICUSB_DP_PDDIS	USBC1_ICUSB_DM_PDDIS	RESERVED																				

Bits	Field Name	Description	Type	Reset
31	SDMMC1_PUSTRENGTH_GRP0	Pullstrength control for sdmmc1_pustrength_grp0 0x0: Pull up with 50 to 100 kΩ on MMC1 data pins 0x1: Pull up with 10 to 50 kΩ on MMC1 data pins	RW	1
30	SDMMC1_PUSTRENGTH_GRP1	Pullstrength control for sdmmc1_pustrength_grp1 0x0: Pull up with 50 to 100 kΩ on MMC1 data pins 0x1: Pull up with 10 to 50 kΩ on MMC1 data pins	RW	0
29	SDMMC1_PUSTRENGTH_GRP2	Pullstrength control for sdmmc1_pustrength_grp2 0x0: Pull up with 50 to 100 kΩ on MMC1 data pins 0x1: Pull up with 10 to 50 kΩ on MMC1 data pins	RW	1
28	SDMMC1_PUSTRENGTH_GRP3	Pullstrength control for sdmmc1_pustrength_grp3 0x0: Pull up with 50 to 100 kΩ on MMC1 data pins 0x1: Pull up with 10 to 50 kΩ on MMC1 data pins	RW	1
27	SDMMC1_DR0_SPEEDCTRL	Speed control for group sdmmc1_dr0 0x0: FMAX = 26Mhz at 30pF 0x1: FMAX = 65Mhz at 30 pF	RW	0

Bits	Field Name	Description	Type	Reset
26	SDMMC1_DR1_SPEEDCTRL	Speed control for group sdmmc1_dr1 0x0: FMAX = 26Mhz at 30pF 0x1: FMAX = 65Mhz at 30 pF	RW	0
25	SDMMC1_DR2_SPEEDCTRL	Speed control for group sdmmc1_dr2 0x0: FMAX = 26Mhz at 30pF 0x1: FMAX = 65Mhz at 30 pF	RW	0
24	USBC1_DR0_SPEEDCTRL	Speed control for group usbc1_dr0 0x0: FMAX = 26Mhz at 30pF 0x1: FMAX = 65Mhz at 30 pF	RW	0
23	USB_FD_CDEN	USB FD pull down select 0x0: select pull down from USB FD logic 0x1: select pull down from device control modules registers	RW	0
22	USBC1_ICUSB_DP_PDDIS	usbc1_icusb_dp pull down disable 0x0: usbc1_icusb_dp pull down disabled 0x1: usbc1_icusb_dp pull down enabled	RW	0
21	USBC1_ICUSB_DM_PDDIS	usbc1_icusb_dm pull down disable 0x0: usbc1_icusb_dm pull down disabled 0x1: usbc1_icusb_dm pull down enabled	RW	0
20:0	RESERVED		R	0x000000

**Table 18-587. Register Call Summary for Register CONTROL\_MMC1**

Control Module Functional Description

- [Extended-Drain I/O and PBIAS Cell: \[0\] \[1\] \[2\] \[3\]](#)
- [Extended-Drain I/O: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [Signal Integrity Parameter Controls Overview: \[13\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]](#)

Control Module Programming Guide

- [Extended-Drain I/Os and PBIAS Cells Programming Guide: \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[33\]](#)

**Table 18-588. CONTROL\_HSI**

<b>Address Offset</b>	0x0000 062C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	HSI control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSI1_CALLOOP_SEL	HSI1_CALMUX_SEL	HSI2_CALLOOP_SEL	HSI2_CALMUX_SEL	RESERVED																											

Bits	Field Name	Description	Type	Reset
31	HSI1_CALLOOP_SEL	hsi1 calibration loop select 0x0: Calibration mode 0x1: Enable Calibration mode	RW	0
30	HSI1_CALMUX_SEL	hsi1 calibration mux select 0x0: Calibration mode 0x1: Enable Calibration mode	RW	0
29	HSI2_CALLOOP_SEL	hsi2 calibration loop select 0x0: Calibration mode 0x1: Enable Calibration mode	RW	0
28	HSI2_CALMUX_SEL	hsi2 calibration mux select 0x0: Calibration mode 0x1: Enable Calibration mode	RW	0
27:0	RESERVED		R	0x0000000

**Table 18-589. Register Call Summary for Register CONTROL\_HSI**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-590. CONTROL\_USB**

<b>Address Offset</b>	0x0000 0630	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	USB control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARKIT_USBA0_ULPIPHY_DAT0_AUTO_EN	CARKIT_USBA0_ULPIPHY_DAT1_AUTO_EN	RESERVED																													

Bits	Field Name	Description	Type	Reset
31	CARKIT_USBA0_ULPIPHY_DAT0_AUTO_EN	carkit usba0 ulpiphy dat0 auto enable 0x0: Carkit Auto mode disabled 0x1: Carkit Auto mode enabled	RW	0
30	CARKIT_USBA0_ULPIPHY_DAT1_AUTO_EN	carkit usba0 ulpiphy dat1 auto enable 0x0: Carkit Auto mode disabled 0x1: Carkit Auto mode enabled	RW	0
29:0	RESERVED		R	0x0000 0000

**Table 18-591. Register Call Summary for Register CONTROL\_USB**

Control Module Functional Description

- [Device Interfaces Signal Group Controls Mapping: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[16\]](#)

**Table 18-592. CONTROL\_HDQ**

<b>Address Offset</b>	0x0000 0634	
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b> SYSCTRL_PADCONF_CORE
<b>Description</b>	HDQ control Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED		R	0x0000 0000

**Table 18-593. Register Call Summary for Register CONTROL\_HDQ**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-594. CONTROL\_LPDDR2IO1\_0**

<b>Address Offset</b>	0x0000 0638	
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b> SYSCTRL_PADCONF_CORE
<b>Description</b>	LPDDR2 1 I/O control 0 Access conditions. Read: unrestricted, Write: unrestricted	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPDDR2IO1_GR4_SR	LPDDR2IO1_GR4_I	LPDDR2IO1_GR4_WD	RESERVED	LPDDR2IO1_GR3_SR	LPDDR2IO1_GR3_I	LPDDR2IO1_GR3_WD	RESERVED	LPDDR2IO1_GR2_SR	LPDDR2IO1_GR2_I	LPDDR2IO1_GR2_WD	RESERVED	LPDDR2IO1_GR1_SR	LPDDR2IO1_GR1_I	LPDDR2IO1_GR1_WD	RESERVED																

Bits	Field Name	Description	Type	Reset
31:30	LPDDR2IO1_GR4_SR	Group 4 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2

Bits	Field Name	Description	Type	Reset
29:27	LPDDR2IO1_GR4_I	Group 4 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
26:25	LPDDR2IO1_GR4_WD	Group 4 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
24	RESERVED		R	0
23:22	LPDDR2IO1_GR3_SR	Group 3 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
21:19	LPDDR2IO1_GR3_I	Group 3 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
18:17	LPDDR2IO1_GR3_WD	Group 3 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
16	RESERVED		R	0
15:14	LPDDR2IO1_GR2_SR	Group 2 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
13:11	LPDDR2IO1_GR2_I	Group 2 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3

Bits	Field Name	Description	Type	Reset
10:9	LPDDR2IO1_GR2_WD	Group 2 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
8	RESERVED		R	0
7:6	LPDDR2IO1_GR1_SR	Group 1 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
5:3	LPDDR2IO1_GR1_I	Group 1 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
2:1	LPDDR2IO1_GR1_WD	Group 1 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
0	RESERVED		R	0

**Table 18-595. Register Call Summary for Register CONTROL\_LPDDR2IO1\_0**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[13\]](#)

**Table 18-596. CONTROL\_LPDDR2IO1\_1**

<b>Address Offset</b>	0x0000 063C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	LPDDR2 1 I/O control 1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPDDR2IO1_GR8_SR		LPDDR2IO1_GR8_I		LPDDR2IO1_GR8_WD	RESERVED	LPDDR2IO1_GR7_SR		LPDDR2IO1_GR7_I		LPDDR2IO1_GR7_WD	RESERVED	LPDDR2IO1_GR6_SR		LPDDR2IO1_GR6_I		LPDDR2IO1_GR6_WD	RESERVED	LPDDR2IO1_GR5_SR		LPDDR2IO1_GR5_I		LPDDR2IO1_GR5_WD	RESERVED								

Bits	Field Name	Description	Type	Reset
31:30	LPDDR2IO1_GR8_SR	Group 8 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
29:27	LPDDR2IO1_GR8_I	Group 8 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
26:25	LPDDR2IO1_GR8_WD	Group 8 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
24	RESERVED		R	0
23:22	LPDDR2IO1_GR7_SR	Group 7 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
21:19	LPDDR2IO1_GR7_I	Group 7 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
18:17	LPDDR2IO1_GR7_WD	Group 7 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
16	RESERVED		R	0
15:14	LPDDR2IO1_GR6_SR	Group 6 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2



Bits	Field Name	Description	Type	Reset
13:11	LPDDR2IO1_GR6_I	Group 6 Impedance control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
10:9	LPDDR2IO1_GR6_WD	Group 6 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
8	RESERVED		R	0
7:6	LPDDR2IO1_GR5_SR	Group 5 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
5:3	LPDDR2IO1_GR5_I	Group 5 Impedance control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
2:1	LPDDR2IO1_GR5_WD	Group 5 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
0	RESERVED		R	0

**Table 18-597. Register Call Summary for Register CONTROL\_LPDDR2IO1\_1**

## Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[13\]](#)

**Table 18-598. CONTROL\_LPDDR2IO1\_2**

<b>Address Offset</b>	0x0000 0640		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	LPDDR2 1 I/O control 2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPDDR2IO1_GR11_SR		LPDDR2IO1_GR11_I		LPDDR2IO1_GR11_WD		RESERVED		LPDDR2IO1_GR10_SR		LPDDR2IO1_GR10_I		LPDDR2IO1_GR10_WD		RESERVED		LPDDR2IO1_GR9_SR		LPDDR2IO1_GR9_I		LPDDR2IO1_GR9_WD		RESERVED									

Bits	Field Name	Description	Type	Reset
31:30	LPDDR2IO1_GR11_SR	Group 11 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
29:27	LPDDR2IO1_GR11_I	Group 11 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
26:25	LPDDR2IO1_GR11_WD	Group 11 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
24	RESERVED		R	0
23:22	LPDDR2IO1_GR10_SR	Group 10 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2

Bits	Field Name	Description	Type	Reset
21:19	LPDDR2IO1_GR10_I	Group 10 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
18:17	LPDDR2IO1_GR10_WD	Group 10 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
16	RESERVED		R	0
15:14	LPDDR2IO1_GR9_SR	Group 9 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
13:11	LPDDR2IO1_GR9_I	Group 9 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
10:9	LPDDR2IO1_GR9_WD	Group 9 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
8:0	RESERVED		R	0x000

**Table 18-599. Register Call Summary for Register CONTROL\_LPDDR2IO1\_2**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[10\]](#)

**Table 18-600. CONTROL\_LPDDR2IO1\_3**

<b>Address Offset</b>	0x0000 0644		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	LPDDR2 1 I/O control 3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPDDR21_VREF_CA_CCAP0	LPDDR21_VREF_CA_CCAP1	LPDDR21_VREF_CA_INT_CCAP0	LPDDR21_VREF_CA_INT_CCAP1	LPDDR21_VREF_CA_INT_TAP0	LPDDR21_VREF_CA_INT_TAP1	LPDDR21_VREF_CA_TAP0	LPDDR21_VREF_CA_TAP1	LPDDR21_VREF_DQ_INT0_CCAP0	LPDDR21_VREF_DQ_INT0_CCAP1	LPDDR21_VREF_DQ_INT0_TAP0	LPDDR21_VREF_DQ_INT0_TAP1	LPDDR21_VREF_DQ_INT1_CCAP0	LPDDR21_VREF_DQ_INT1_CCAP1	LPDDR21_VREF_DQ_INT1_TAP0	LPDDR21_VREF_DQ_INT1_TAP1	LPDDR21_VREF_DQ_CCAP0	LPDDR21_VREF_DQ_CCAP1	LPDDR21_VREF_DQ_TAP0	LPDDR21_VREF_DQ_TAP1	LPDDR21_VREF_DQ_INT2_CCAP0	LPDDR21_VREF_DQ_INT2_CCAP1	LPDDR21_VREF_DQ_INT2_TAP0	LPDDR21_VREF_DQ_INT2_TAP1	LPDDR21_VREF_DQ_INT3_CCAP0	LPDDR21_VREF_DQ_INT3_CCAP1	LPDDR21_VREF_DQ_INT3_TAP0	LPDDR21_VREF_DQ_INT3_TAP1	LPDDR21_VREF_EN_CA	LPDDR21_VREF_EN_DQ	LPDDR21_VREF_AUTO_EN_CA	LPDDR21_VREF_AUTO_EN_DQ

Bits	Field Name	Description	Type	Reset
31	LPDDR21_VREF_CA_CCAP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
30	LPDDR21_VREF_CA_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
29	LPDDR21_VREF_CA_INT_CCA P0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
28	LPDDR21_VREF_CA_INT_CCA P1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
27	LPDDR21_VREF_CA_INT_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
26	LPDDR21_VREF_CA_INT_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
25	LPDDR21_VREF_CA_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
24	LPDDR21_VREF_CA_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
23	LPDDR21_VREF_DQ_INT0_CC AP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1

Bits	Field Name	Description	Type	Reset
22	LPDDR21_VREF_DQ_INT0_CC AP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
21	LPDDR21_VREF_DQ_INT0_TA P0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
20	LPDDR21_VREF_DQ_INT0_TA P1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
19	LPDDR21_VREF_DQ_INT1_CC AP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
18	LPDDR21_VREF_DQ_INT1_CC AP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
17	LPDDR21_VREF_DQ_INT1_TA P0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
16	LPDDR21_VREF_DQ_INT1_TA P1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
15	LPDDR21_VREF_DQ_CCAPO	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
14	LPDDR21_VREF_DQ_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
13	LPDDR21_VREF_DQ_TAPO	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
12	LPDDR21_VREF_DQ_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
11	LPDDR21_VREF_DQ_INT2_CC AP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
10	LPDDR21_VREF_DQ_INT3_CC AP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
9	LPDDR21_VREF_DQ_INT2_CC AP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
8	LPDDR21_VREF_DQ_INT3_CC AP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
7	LPDDR21_VREF_DQ_INT2_TA P0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1

Bits	Field Name	Description	Type	Reset
6	LPDDR21_VREF_DQ_INT3_TA P0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
5	LPDDR21_VREF_DQ_INT2_TA P1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
4	LPDDR21_VREF_DQ_INT3_TA P1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
3	LPDDR21_INT_VREF_EN_CA	Internal vref enable for CA in manual mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1
2	LPDDR21_INT_VREF_EN_DQ	Internal vref enable for DQ in manual mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1
1	LPDDR21_INT_VREF_AUTO_E N_CA	Internal vref enable for CA in auto mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1
0	LPDDR21_INT_VREF_AUTO_E N_DQ	Internal vref enable for DQ in auto mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1

**Table 18-601. Register Call Summary for Register CONTROL\_LPDDR2IO1\_3**

## Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [High-speed I/O Buffers with Impedance, Slew Rate and Weak Driver Settings: \[1\] \[2\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[3\]](#)

**Table 18-602. CONTROL\_LPDDR2IO2\_0**

<b>Address Offset</b>	0x0000 0648	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	LPDDR2 2 I/O control 0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPDDR2IO2_GR4_SR		LPDDR2IO2_GR4_I		LPDDR2IO2_GR4_WD		RESERVED		LPDDR2IO2_GR3_SR		LPDDR2IO2_GR3_I		LPDDR2IO2_GR3_WD		RESERVED		LPDDR2IO2_GR2_SR		LPDDR2IO2_GR2_I		LPDDR2IO2_GR2_WD		RESERVED		LPDDR2IO2_GR1_SR		LPDDR2IO2_GR1_I		LPDDR2IO2_GR1_WD		RESERVED	

Bits	Field Name	Description	Type	Reset
31:30	LPDDR2IO2_GR4_SR	Group 4 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
29:27	LPDDR2IO2_GR4_I	Group 4 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
26:25	LPDDR2IO2_GR4_WD	Group 4 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
24	RESERVED		R	0
23:22	LPDDR2IO2_GR3_SR	Group 3 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
21:19	LPDDR2IO2_GR3_I	Group 3 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
18:17	LPDDR2IO2_GR3_WD	Group 3 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
16	RESERVED		R	0
15:14	LPDDR2IO2_GR2_SR	Group 2 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2

Bits	Field Name	Description	Type	Reset
13:11	LPDDR2IO2_GR2_I	Group 2 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
10:9	LPDDR2IO2_GR2_WD	Group 2 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
8	RESERVED		R	0
7:6	LPDDR2IO2_GR1_SR	Group 1 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
5:3	LPDDR2IO2_GR1_I	Group 1 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
2:1	LPDDR2IO2_GR1_WD	Group 1 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
0	RESERVED		R	0

**Table 18-603. Register Call Summary for Register CONTROL\_LPDDR2IO2\_0**

## Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[13\]](#)



**Table 18-604. CONTROL\_LPDDR2IO2\_1**

<b>Address Offset</b>	0x0000 064C		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	LPDDR2 2 I/O control 1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPDDR2IO2_GR8_SR	LPDDR2IO2_GR8_I	LPDDR2IO2_GR8_WD	RESERVED	LPDDR2IO2_GR7_SR	LPDDR2IO2_GR7_I	LPDDR2IO2_GR7_WD	RESERVED	LPDDR2IO2_GR6_SR	LPDDR2IO2_GR6_I	LPDDR2IO2_GR6_WD	RESERVED	LPDDR2IO2_GR5_SR	LPDDR2IO2_GR5_I	LPDDR2IO2_GR5_WD	RESERVED																

Bits	Field Name	Description	Type	Reset
31:30	LPDDR2IO2_GR8_SR	Group 8 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
29:27	LPDDR2IO2_GR8_I	Group 8 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
26:25	LPDDR2IO2_GR8_WD	Group 8 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
24	RESERVED		R	0
23:22	LPDDR2IO2_GR7_SR	Group 7 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2

Bits	Field Name	Description	Type	Reset
21:19	LPDDR2IO2_GR7_I	Group 7 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
18:17	LPDDR2IO2_GR7_WD	Group 7 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
16	RESERVED		R	0
15:14	LPDDR2IO2_GR6_SR	Group 6 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
13:11	LPDDR2IO2_GR6_I	Group 6 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
10:9	LPDDR2IO2_GR6_WD	Group 6 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
8	RESERVED		R	0
7:6	LPDDR2IO2_GR5_SR	Group 5 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
5:3	LPDDR2IO2_GR5_I	Group 5 Impedence control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3

Bits	Field Name	Description	Type	Reset
2:1	LPDDR2IO2_GR5_WD	Group 5 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
0	RESERVED		R	0

**Table 18-605. Register Call Summary for Register CONTROL\_LPDDR2IO2\_1**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[13\]](#)

**Table 18-606. CONTROL\_LPDDR2IO2\_2**

<b>Address Offset</b>	0x0000 0650	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	LPDDR2 2 I/O control 2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
LPDDR2IO2_GR11_SR		LPDDR2IO2_GR11_I		LPDDR2IO2_GR11_WD		RESERVED		LPDDR2IO2_GR10_SR		LPDDR2IO2_GR10_I		LPDDR2IO2_GR10_WD		RESERVED		LPDDR2IO2_GR9_SR		LPDDR2IO2_GR9_I		LPDDR2IO2_GR9_WD		RESERVED															

Bits	Field Name	Description	Type	Reset
31:30	LPDDR2IO2_GR11_SR	Group 11 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
29:27	LPDDR2IO2_GR11_I	Group 11 Impedance control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3

Bits	Field Name	Description	Type	Reset
26:25	LPDDR2IO2_GR11_WD	Group 11 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
24	RESERVED		R	0
23:22	LPDDR2IO2_GR10_SR	Group 10 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
21:19	LPDDR2IO2_GR10_I	Group 10 Impedance control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
18:17	LPDDR2IO2_GR10_WD	Group 10 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
16	RESERVED		R	0
15:14	LPDDR2IO2_GR9_SR	Group 9 Slew Rate control sr1:sr0 0x0: Turn_on time 125ps for Drv8 0x1: Turn_on time 500ps for Drv8 0x2: Turn_on time 325ps for Drv8 0x3: Turn_on time 750ps for Drv8	RW	0x2
13:11	LPDDR2IO2_GR9_I	Group 9 Impedance control i2:i0 0x0: for Drv5 0x1: for Drv6 0x2: for Drv7 0x3: for Drv8 0x4: for Drv9 0x5: for Drv10 0x6: for Drv11 0x7: Drv12	RW	0x3
10:9	LPDDR2IO2_GR9_WD	Group 9 Weak driver control wd1:wd0 0x0: Pull logic disabled 0x1: pull up 0x2: pull down 0x3: Maintain the previous output value	RW	0x2
8:0	RESERVED		R	0x000

**Table 18-607. Register Call Summary for Register CONTROL\_LPDDR2IO2\_2**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[10\]](#)

**Table 18-608. CONTROL\_LPDDR2IO2\_3**

<b>Address Offset</b>	0x0000 0654	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	LPDDR2 2 I/O control 3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPDDR22_VREF_CA_CCAP0	LPDDR22_VREF_CA_CCAP1	LPDDR22_VREF_CA_INT_CCAP0	LPDDR22_VREF_CA_INT_CCAP1	LPDDR22_VREF_CA_INT_TAP0	LPDDR22_VREF_CA_INT_TAP1	LPDDR22_VREF_CA_TAP0	LPDDR22_VREF_CA_TAP1	LPDDR22_VREF_DQ_INT0_CCAP0	LPDDR22_VREF_DQ_INT0_CCAP1	LPDDR22_VREF_DQ_INT0_TAP0	LPDDR22_VREF_DQ_INT0_TAP1	LPDDR22_VREF_DQ_INT1_CCAP0	LPDDR22_VREF_DQ_INT1_CCAP1	LPDDR22_VREF_DQ_INT1_TAP0	LPDDR22_VREF_DQ_INT1_TAP1	LPDDR22_VREF_DQ_CCAP0	LPDDR22_VREF_DQ_CCAP1	LPDDR22_VREF_DQ_TAP0	LPDDR22_VREF_DQ_TAP1	LPDDR22_VREF_DQ_INT2_CCAP0	LPDDR22_VREF_DQ_INT3_CCAP0	LPDDR22_VREF_DQ_INT2_CCAP1	LPDDR22_VREF_DQ_INT3_CCAP1	LPDDR22_VREF_DQ_INT2_TAP0	LPDDR22_VREF_DQ_INT3_TAP0	LPDDR22_VREF_DQ_INT2_TAP1	LPDDR22_VREF_DQ_INT3_TAP1	LPDDR22_INT_VREF_EN_CA	LPDDR22_INT_VREF_EN_DQ	LPDDR22_INT_VREF_AUTO_EN_CA	LPDDR22_INT_VREF_AUTO_EN_DQ

Bits	Field Name	Description	Type	Reset
31	LPDDR22_VREF_CA_CCAP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
30	LPDDR22_VREF_CA_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
29	LPDDR22_VREF_CA_INT_CCA P0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
28	LPDDR22_VREF_CA_INT_CCA P1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
27	LPDDR22_VREF_CA_INT_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
26	LPDDR22_VREF_CA_INT_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
25	LPDDR22_VREF_CA_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1

Bits	Field Name	Description	Type	Reset
24	LPDDR22_VREF_CA_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
23	LPDDR22_VREF_DQ_INT0_CC AP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
22	LPDDR22_VREF_DQ_INT0_CC AP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
21	LPDDR22_VREF_DQ_INT0_TA P0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
20	LPDDR22_VREF_DQ_INT0_TA P1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
19	LPDDR22_VREF_DQ_INT1_CC AP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
18	LPDDR22_VREF_DQ_INT1_CC AP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
17	LPDDR22_VREF_DQ_INT1_TA P0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
16	LPDDR22_VREF_DQ_INT1_TA P1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
15	LPDDR22_VREF_DQ_CCAP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
14	LPDDR22_VREF_DQ_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
13	LPDDR22_VREF_DQ_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
12	LPDDR22_VREF_DQ_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
11	LPDDR22_VREF_DQ_INT2_CC AP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
10	LPDDR22_VREF_DQ_INT3_CC AP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	1
9	LPDDR22_VREF_DQ_INT2_CC AP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
8	LPDDR22_VREF_DQ_INT3_CC AP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0
7	LPDDR22_VREF_DQ_INT2_TA P0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
6	LPDDR22_VREF_DQ_INT3_TA P0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
5	LPDDR22_VREF_DQ_INT2_TA P1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
4	LPDDR22_VREF_DQ_INT3_TA P1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	1
3	LPDDR22_INT_VREF_EN_CA	Internal vref enable for CA in manual mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1
2	LPDDR22_INT_VREF_EN_DQ	Internal vref enable for DQ in manual mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1
1	LPDDR22_INT_VREF_AUTO_E N_CA	Internal vref enable for CA in auto mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1
0	LPDDR22_INT_VREF_AUTO_E N_DQ	Internal vref enable for DQ in auto mode (if both manual and auto mode is enabled then manual mode takes the priority) 0x0: Disabled 0x1: Enabled	RW	1

**Table 18-609. Register Call Summary for Register CONTROL\_LPDDR2IO2\_3**

## Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [High-speed I/O Buffers with Impedance, Slew Rate and Weak Driver Settings: \[1\] \[2\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[3\]](#)

**Table 18-610. CONTROL\_BUS\_HOLD**

<b>Address Offset</b>	0x0000 0658																																																																	
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b> SYSCTRL_PADCONF_CORE																																																																
<b>Description</b>	BUS HOLD I/O controls Access conditions. Read: unrestricted, Write: unrestricted																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">RESERVED</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
RESERVED																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	RESERVED		R	0x0000 0000																																																														

**Table 18-611. Register Call Summary for Register CONTROL\_BUS\_HOLD**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-612. CONTROL\_C2C**

<b>Address Offset</b>	0x0000 065C																																																																	
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b> SYSCTRL_PADCONF_CORE																																																																
<b>Description</b>	C2C controls Access conditions. Read: unrestricted, Write: unrestricted																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">C2C_SPARE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	C2C_SPARE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
C2C_SPARE																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	C2C_SPARE	C2C spare register bits	RW	0x0000 0000																																																														

**Table 18-613. Register Call Summary for Register CONTROL\_C2C**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-614. CONTROL\_CORE\_CONTROL\_SPARE\_RW**

<b>Address Offset</b>	0x0000 0660																																																																	
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b> SYSCTRL_PADCONF_CORE																																																																
<b>Description</b>	CORE control spare RW Access conditions. Read: unrestricted, Write: unrestricted																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">CORE_CONTROL_SPARE_RW</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CORE_CONTROL_SPARE_RW																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
CORE_CONTROL_SPARE_RW																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	CORE_CONTROL_SPARE_RW	Core control spare register bits RW	RW	0x0000 0000																																																														



**Table 18-615. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-616. CONTROL\_CORE\_CONTROL\_SPARE\_R**

<b>Address Offset</b>	0x0000 0664		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	CORE control spare R Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_R																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_R	Core control spare register bits R	R	0x0000 0000

**Table 18-617. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_R**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-618. CONTROL\_CORE\_CONTROL\_SPARE\_R\_C0**

<b>Address Offset</b>	0x0000 0668		
<b>Physical Address</b>	See <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	CORE control spare RC Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_R_C0	CORE_CONTROL_SPARE_R_C1	CORE_CONTROL_SPARE_R_C2	CORE_CONTROL_SPARE_R_C3	CORE_CONTROL_SPARE_R_C4	CORE_CONTROL_SPARE_R_C5	CORE_CONTROL_SPARE_R_C6	CORE_CONTROL_SPARE_R_C7	RESERVED																							

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_R_C	Core control spare register bits RC	RW (Write 1 to Clear)	0
30:1	CORE_CONTROL_SPARE_R_C	Core control spare register bits RC	RW (Write 1 to Clear)	0
29:2	CORE_CONTROL_SPARE_R_C	Core control spare register bits RC	RW (Write 1 to Clear)	0

Bits	Field Name	Description	Type	Reset
28	CORE_CONTROL_SPARE_R_C 3	Core control spare register bits RC	RW (Write 1 to Clear)	0
27	CORE_CONTROL_SPARE_R_C 4	Core control spare register bits RC	RW (Write 1 to Clear)	0
26	CORE_CONTROL_SPARE_R_C 5	Core control spare register bits RC	RW (Write 1 to Clear)	0
25	CORE_CONTROL_SPARE_R_C 6	Core control spare register bits RC	RW (Write 1 to Clear)	0
24	CORE_CONTROL_SPARE_R_C 7	Core control spare register bits RC	RW (Write 1 to Clear)	0
23:0	RESERVED		R	0x000000

**Table 18-619. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_R\_C0**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-620. CONTROL\_CORE\_CONTROL\_SPARE\_RW1**

<b>Address Offset</b>	0x0000 066C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>		
<b>Description</b>	CORE control spare RW1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_RW1																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_RW 1	Core control spare register bits RW1	RW	0x0000 0000

**Table 18-621. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW1**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-622. CONTROL\_CORE\_CONTROL\_SPARE\_RW2**

<b>Address Offset</b>	0x0000 0670	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>		
<b>Description</b>	CORE control spare RW2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_RW2																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_RW 2	Core control spare register bits RW2	RW	0x0000 0000

**Table 18-623. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW2**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-624. CONTROL\_CORE\_CONTROL\_SPARE\_RW3**

<b>Address Offset</b>	0x0000 0674	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>		
<b>Description</b>	CORE control spare RW3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_RW3																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_RW3	Core control spare register bits RW3	RW	0x0000 0000

**Table 18-625. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW3**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-626. CONTROL\_CORE\_CONTROL\_SPARE\_RW4**

<b>Address Offset</b>	0x0000 0678	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>		
<b>Description</b>	CORE control spare RW4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_RW4																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_RW4	Core control spare register bits RW4	RW	0x0000 0000

**Table 18-627. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW4**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-628. CONTROL\_CORE\_CONTROL\_SPARE\_RW5**

<b>Address Offset</b>	0x0000 067C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>		
<b>Description</b>	CORE control spare RW5 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_RW5																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_RW5	Core control spare register bits RW5	RW	0x0000 0000

**Table 18-629. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW5**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-630. CONTROL\_CORE\_CONTROL\_SPARE\_RW6**

<b>Address Offset</b>	0x0000 0680		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	CORE control spare RW6 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_RW6																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_RW6	Core control spare register bits RW6	RW	0x0000 0000

**Table 18-631. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW6**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-632. CONTROL\_CORE\_CONTROL\_SPARE\_RW7**

<b>Address Offset</b>	0x0000 0684		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	CORE control spare RW7 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_RW7																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_RW7	Core control spare register bits RW7	RW	0x0000 0000

**Table 18-633. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW7**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-634. CONTROL\_CORE\_CONTROL\_SPARE\_RW8**

<b>Address Offset</b>	0x0000 0688		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	CORE control spare RW8 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_RW8																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_RW8	Core control spare register bits RW8	RW	0x0000 0000

**Table 18-635. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW8**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-636. CONTROL\_CORE\_CONTROL\_SPARE\_RW9**

<b>Address Offset</b>	0x0000 068C		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	CORE control spare RW9 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_RW9																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_RW9	Core control spare register bits RW9	RW	0xFFFF FFFF

**Table 18-637. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_RW9**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-638. CONTROL\_CORE\_CONTROL\_SPARE\_R1**

<b>Address Offset</b>	0x0000 0690		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-315</a>	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Description</b>	CORE control spare R1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CONTROL_SPARE_R1																															

Bits	Field Name	Description	Type	Reset
31:0	CORE_CONTROL_SPARE_R1	Core control spare register bits R1	R	0x0000 0000

**Table 18-639. Register Call Summary for Register CONTROL\_CORE\_CONTROL\_SPARE\_R1**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-640. CONTROL\_EFUSE\_1**

<b>Address Offset</b>	0x0000 0700	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	EFUSE control 1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	AVDAC_TRIM_BYTE3							AVDAC_TRIM_BYTE2							AVDAC_TRIM_BYTE1							AVDAC_TRIM_BYTE0									

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30:24	AVDAC_TRIM_BYTE3	AVDAC trim byte3 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_avdac_trim_byte_3	RW	0x00
23:16	AVDAC_TRIM_BYTE2	AVDAC trim byte2 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_avdac_trim_byte_2	RW	0x00
15:8	AVDAC_TRIM_BYTE1	AVDAC trim byte1 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_avdac_trim_byte_1	RW	0x00
7:0	AVDAC_TRIM_BYTE0	AVDAC trim byte0 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_avdac_trim_byte_0	RW	0x00

**Table 18-641. Register Call Summary for Register CONTROL\_EFUSE\_1**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-642. CONTROL\_EFUSE\_2**

<b>Address Offset</b>	0x0000 0704	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	EFUSE control 2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EFUSE_SMART2TEST_P0	EFUSE_SMART2TEST_P1	EFUSE_SMART2TEST_P2	EFUSE_SMART2TEST_P3	EFUSE_SMART2TEST_N0	EFUSE_SMART2TEST_N1	EFUSE_SMART2TEST_N2	EFUSE_SMART2TEST_N3	LPDDR2_PTV_N1	LPDDR2_PTV_N2	LPDDR2_PTV_N3	LPDDR2_PTV_N4	LPDDR2_PTV_N5	LPDDR2_PTV_P1	LPDDR2_PTV_P2	LPDDR2_PTV_P3	LPDDR2_PTV_P4	LPDDR2_PTV_P5	RESERVED													

Bits	Field Name	Description	Type	Reset
31	EFUSE_SMART2TEST_P0	Smart2/3/C2C I/O PMOS process compensation bit 0 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_sr_p0	RW	0
30	EFUSE_SMART2TEST_P1	Smart2/3/C2C I/O PMOS process compensation bit 1 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_sr_p1	RW	0
29	EFUSE_SMART2TEST_P2	Smart2/3/C2C I/O PMOS process compensation bit 2 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_sr_p2	RW	0
28	EFUSE_SMART2TEST_P3	Smart2/3/C2C I/O PMOS process compensation bit 3 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_sr_p3	RW	0
27	EFUSE_SMART2TEST_N0	Smart2/3/C2C I/O NMOS process compensation bit 0 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_sr_n0	RW	0
26	EFUSE_SMART2TEST_N1	Smart2/3/C2C I/O NMOS process compensation bit 1 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_sr_n1	RW	0
25	EFUSE_SMART2TEST_N2	Smart2/3/C2C I/O NMOS process compensation bit 2 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_sr_n2	RW	0
24	EFUSE_SMART2TEST_N3	Smart2/3/C2C I/O NMOS process compensation bit 3 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_sr_n3	RW	0
23	LPDDR2_PTV_N1	LPDDR2IO NMOS PTV code bit 1 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_n1	RW	0
22	LPDDR2_PTV_N2	LPDDR2IO NMOS PTV code bit 2 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_n2	RW	0
21	LPDDR2_PTV_N3	LPDDR2IO NMOS PTV code bit 3 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_n3	RW	0
20	LPDDR2_PTV_N4	LPDDR2IO NMOS PTV code bit 4 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_n4	RW	0
19	LPDDR2_PTV_N5	LPDDR2IO NMOS PTV code bit 5 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_n5	RW	0
18	LPDDR2_PTV_P1	LPDDR2IO PMOS PTV code bit 1 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_p1	RW	0
17	LPDDR2_PTV_P2	LPDDR2IO PMOS PTV code bit 2 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_p2	RW	0
16	LPDDR2_PTV_P3	LPDDR2IO PMOS PTV code bit 3 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_p3	RW	0
15	LPDDR2_PTV_P4	LPDDR2IO PMOS PTV code bit 4 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_p4	RW	0
14	LPDDR2_PTV_P5	LPDDR2IO PMOS PTV code bit 5 (Reset value exported from efuse) Note that reset is exported. Its value is = pi_lpddr2_p5	RW	0
13:0	RESERVED		R	0x0000

**Table 18-643. Register Call Summary for Register CONTROL\_EFUSE\_2**

## Control Module Functional Description

- [High-speed I/O Buffers with Impedance, Slew Rate and Weak Driver Settings: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [I/Os with Combined Mode and Load Settings: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [50-Ω Output Buffer I/Os with Combined Mode and Load Settings: \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [I/O cells with Configurable Output Driver Impedance: \[18\] \[19\] \[20\] \[21\] \[22\]](#)
- [C2C/I/O cells with load setting adjustable according to interface requirement / voltage mode: \[23\] \[24\] \[25\] \[26\] \[27\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[28\]](#)

**Table 18-644. CONTROL\_EFUSE\_3**

<b>Address Offset</b>	0x0000 0708	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	EFUSE control 3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_SPARE_1								STD_FUSE_SPARE_2								STD_FUSE_SPARE_3								STD_FUSE_SPARE_4							

Bits	Field Name	Description	Type	Reset
31:24	STD_FUSE_SPARE_1	std efuse spare bits (Read Only) Note that reset is exported. Its value is = pi_std_fuse_spare1	R	0x00
23:16	STD_FUSE_SPARE_2	std efuse spare bits (Read Only) Note that reset is exported. Its value is = pi_std_fuse_spare2	R	0x00
15:8	STD_FUSE_SPARE_3	std efuse spare bits (Read Only) Note that reset is exported. Its value is = pi_std_fuse_spare3	R	0x00
7:0	STD_FUSE_SPARE_4	std efuse spare bits (Read Only) Note that reset is exported. Its value is = pi_std_fuse_spare4	R	0x00

**Table 18-645. Register Call Summary for Register CONTROL\_EFUSE\_3**

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

**Table 18-646. CONTROL\_EFUSE\_4**

<b>Address Offset</b>	0x0000 070C	<b>Instance</b>	SYSCTRL_PADCONF_CORE
<b>Physical Address</b>	See <a href="#">Table 18-315</a>		
<b>Description</b>	EFUSE control 4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_SPARE_5								STD_FUSE_SPARE_6								STD_FUSE_SPARE_7								STD_FUSE_SPARE_8							



Bits	Field Name	Description	Type	Reset
31:24	STD_FUSE_SPARE_5	std efuse spare bits (RW) Note that reset is exported. Its value is = pi_std_fuse_spare5	RW	0x00
23:16	STD_FUSE_SPARE_6	std efuse spare bits (RW) Note that reset is exported. Its value is = pi_std_fuse_spare6	RW	0x00
15:8	STD_FUSE_SPARE_7	std efuse spare bits (RW) Note that reset is exported. Its value is = pi_std_fuse_spare7	RW	0x00
7:0	STD_FUSE_SPARE_8	std efuse spare bits (RW) Note that reset is exported. Its value is = pi_std_fuse_spare8	RW	0x00

**Table 18-647. Register Call Summary for Register CONTROL\_EFUSE\_4**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_CORE Register Summary: \[0\]](#)

### 18.6.8 SYSCTRL\_PADCONF\_WKUP Register Summary

**Table 18-648. SYSCTRL\_PADCONF\_WKUP Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_PADCONF_WKUP Physical Address
<a href="#">CONTROL_PADCONF_WKUP_REVISION</a>	R	32	0x0000 0000	0x4A31 E000
<a href="#">CONTROL_PADCONF_WKUP_HWINFO</a>	R	32	0x0000 0004	0x4A31 E004
<a href="#">CONTROL_PADCONF_WKUP_SYS_CONFIG</a>	RW	32	0x0000 0010	0x4A31 E010
<a href="#">CONTROL_WKUP_PAD0_GPIO_WK0_PAD1_GPIO_WK1</a>	RW	32	0x0000 0040	0x4A31 E040
<a href="#">CONTROL_WKUP_PAD0_GPIO_WK2_PAD1_GPIO_WK3</a>	RW	32	0x0000 0044	0x4A31 E044
<a href="#">CONTROL_WKUP_PAD0_GPIO_WK4_PAD1_SR_SCL</a>	RW	32	0x0000 0048	0x4A31 E048
<a href="#">CONTROL_WKUP_PAD0_SR_SDA_PAD1_FREF_XTAL_IN</a>	RW	32	0x0000 004C	0x4A31 E04C
<a href="#">CONTROL_WKUP_PAD0_FREF_SLICER_IN_PAD1_FREF_CLK_IOREQ</a>	RW	32	0x0000 0050	0x4A31 E050
<a href="#">CONTROL_WKUP_PAD0_FREF_CLK0_OUT_PAD1_FREF_CLK3_REQ</a>	RW	32	0x0000 0054	0x4A31 E054
<a href="#">CONTROL_WKUP_PAD0_FREF_CLK3_OUT_PAD1_FREF_CLK4_REQ</a>	RW	32	0x0000 0058	0x4A31 E058
<a href="#">CONTROL_WKUP_PAD0_FREF_CLK4_OUT_PAD1_SYS_32K</a>	RW	32	0x0000 005C	0x4A31 E05C
<a href="#">CONTROL_WKUP_PAD0_SYS_NRESPWRON_PAD1_SYS_NRESWARM</a>	RW	32	0x0000 0060	0x4A31 E060
<a href="#">CONTROL_WKUP_PAD0_SYS_PWR_REQ_PAD1_SYS_PWRON_RESET_OUT</a>	RW	32	0x0000 0064	0x4A31 E064
<a href="#">CONTROL_WKUP_PAD0_SYS_BOOT6_PAD1_SYS_BOOT7</a>	RW	32	0x0000 0068	0x4A31 E068
<a href="#">CONTROL_WKUP_PAD0_JTAG_NTRST_PAD1_JTAG_TCK</a>	RW	32	0x0000 006C	0x4A31 E06C
<a href="#">CONTROL_WKUP_PAD0_JTAG_RTCK_PAD1_JTAG_TMS_TMSC</a>	RW	32	0x0000 0070	0x4A31 E070

**Table 18-648. SYSCTRL\_PADCONF\_WKUP Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	SYSCTRL_PADCONF_WKUP Physical Address
CONTROL_WKUP_PAD0_JTAG_TDI_PAD1_JTAG_TDO	RW	32	0x0000 0074	0x4A31 E074
CONTROL_WKUP_PADCONF_WAK_EUPEVENT_0	R	32	0x0000 007C	0x4A31 E07C
CONTROL_SMART1NOPMIO_PADCONF_0	RW	32	0x0000 05A0	0x4A31 E5A0
CONTROL_SMART1NOPMIO_PADCONF_1	RW	32	0x0000 05A4	0x4A31 E5A4
CONTROL_WKUP_PADCONF_MODE	RW	32	0x0000 05A8	0x4A31 E5A8
CONTROL_XTAL_OSCILLATOR	RW	32	0x0000 05AC	0x4A31 E5AC
CONTROL_SMART3NOPMIO_PADCONF_0	RW	32	0x0000 05B0	0x4A31 E5B0
CONTROL_SMART3NOPMIO_PADCONF_1	RW	32	0x0000 05B4	0x4A31 E5B4
CONTROL_GPIOWK	RW	32	0x0000 0600	0x4A31 E600
CONTROL_I2C_2	RW	32	0x0000 0604	0x4A31 E604
CONTROL_JTAG	RW	32	0x0000 0608	0x4A31 E608
CONTROL_SYS	RW	32	0x0000 060C	0x4A31 E60C
CONTROL_WKUP_CONTROL_SPARE_RW	RW	32	0x0000 0614	0x4A31 E614
CONTROL_WKUP_CONTROL_SPARE_R	R	32	0x0000 0618	0x4A31 E618
CONTROL_WKUP_CONTROL_SPARE_R_C0	RW	32	0x0000 061C	0x4A31 E61C
CONTROL_WKUP_CONTROL_SPARE_RW1	RW	32	0x0000 0620	0x4A31 E620
CONTROL_WKUP_CONTROL_SPARE_RW2	RW	32	0x0000 0624	0x4A31 E624
CONTROL_WKUP_CONTROL_SPARE_RW3	RW	32	0x0000 0628	0x4A31 E628
CONTROL_WKUP_CONTROL_SPARE_RW4	RW	32	0x0000 062C	0x4A31 E62C
CONTROL_WKUP_CONTROL_SPARE_RW5	RW	32	0x0000 0630	0x4A31 E630
CONTROL_WKUP_CONTROL_SPARE_RW6	RW	32	0x0000 0634	0x4A31 E634
CONTROL_WKUP_CONTROL_SPARE_RW7	RW	32	0x0000 0638	0x4A31 E638
CONTROL_WKUP_CONTROL_SPARE_RW8	RW	32	0x0000 063C	0x4A31 E63C
CONTROL_WKUP_CONTROL_SPARE_RW9	RW	32	0x0000 0640	0x4A31 E640
CONTROL_WKUP_CONTROL_SPARE_R1	R	32	0x0000 064C	0x4A31 E64C

**18.6.9 SYSCTRL\_PADCONF\_WKUP Register Description**

**Table 18-649. CONTROL\_PADCONF\_WKUP\_REVISION**

<b>Address Offset</b>	0x0000 0000																																																																																																
<b>Physical Address</b>	See <a href="#">Table 18-648</a>																<b>Instance</b>																SYSCTRL_PADCONF_WKUP																																																																
<b>Description</b>	Control module revision identifier Access conditions. Read: unrestricted, Write: unrestricted																																																																																																
<b>Type</b>	R																																																																																																
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="33">REVISION</td> </tr> </table>																																	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	REVISION																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																		
REVISION																																																																																																	

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 18-650. Register Call Summary for Register CONTROL\_PADCONF\_WKUP\_REVISION**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-651. CONTROL\_PADCONF\_WKUP\_HWINFO**

<b>Address Offset</b>	0x0000 0004																																																																																																
<b>Physical Address</b>	See <a href="#">Table 18-648</a>																<b>Instance</b>																SYSCTRL_PADCONF_WKUP																																																																
<b>Description</b>	Information about the IP module hardware configuration that is, typically the module HDL generics (if any). Access conditions. Read: unrestricted, Write: unrestricted																																																																																																
<b>Type</b>	R																																																																																																
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="33">IP_HWINFO</td> </tr> </table>																																	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	IP_HWINFO																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																		
IP_HWINFO																																																																																																	

Bits	Field Name	Description	Type	Reset
31:0	IP_HWINFO	IP-module dependent	R	0x0000 0000

**Table 18-652. Register Call Summary for Register CONTROL\_PADCONF\_WKUP\_HWINFO**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-653. CONTROL\_PADCONF\_WKUP\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010																																
<b>Physical Address</b>	See <a href="#">Table 18-648</a>																<b>Instance</b>																SYSCTRL_PADCONF_WKUP
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted																																
<b>Type</b>	RW																																

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IP_SYSCONFIG_IDLEMODE			RESERVED												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:2	IP_SYSCONFIG_IDLEMODE	Select the local clock-gating strategy 0x0: Clock is manually gated 0x1: Clock is manually enabled 0x2,0x3: Clock is automatically gated when there is no access to the Control Module through L4-interconnect	RW	0x2
1:0	RESERVED		R	0x0

**Table 18-654. Register Call Summary for Register CONTROL\_PADCONF\_WKUP\_SYSCONFIG**

Control Module Functional Description

- [Device Wakeup Control Module Instance: \[0\]](#)

Control Module Programming Guide

- [Control Module Global Initialization: \[1\]](#)
- [Off Mode Preliminary Settings: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[3\]](#)

**Table 18-655. CONTROL\_WKUP\_PAD0\_GPIO\_WK0\_PAD1\_GPIO\_WK1**

<b>Address Offset</b>	0x0000 0040
<b>Physical Address</b>	See <a href="#">Table 18-648</a>
<b>Description</b>	Register control for Pads gpio_wk0 and gpio_wk1 Access conditions. Read: unrestricted, Write: unrestricted
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO_WK1_WAKEUPEVENT	GPIO_WK1_WAKEUPENABLE	RESERVED				GPIO_WK1_INPUTENABLE	RESERVED	GPIO_WK1_PULLTYPESELECT	GPIO_WK1_PULLUDENABLE	GPIO_WK1_MUXMODE	GPIO_WK0_WAKEUPEVENT	GPIO_WK0_WAKEUPENABLE	RESERVED				GPIO_WK0_INPUTENABLE	RESERVED	GPIO_WK0_PULLTYPESELECT	GPIO_WK0_PULLUDENABLE	GPIO_WK0_MUXMODE										

Bits	Field Name	Description	Type	Reset
31	GPIO_WK1_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPIO_WK1_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	GPIO_WK1_INPUTENABLE	Input enable value for pad gpio_wk1 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPIO_WK1_PULLTYPESELECT	pullup/down selection for pad gpio_wk1 0x0: pulldown selected 0x1: pullup selected	RW	0
19	GPIO_WK1_PULLUDENABLE	pullup/down enable for pad gpio_wk1 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPIO_WK1_MUXMODE	Functional multiplexing selection for pad gpio_wk1 0x0: Reserved 0x3: Select gpio_wk1 0x6: Select hw_dbg2 0x7: Select safe_mode	RW	0x7
15	GPIO_WK0_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPIO_WK0_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	GPIO_WK0_INPUTENABLE	Input enable value for pad gpio_wk0 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPIO_WK0_PULLTYPESELECT	pullup/down selection for pad gpio_wk0 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPIO_WK0_PULLUDENABLE	pullup/down enable for pad gpio_wk0 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPIO_WK0_MUXMODE	Functional multiplexing selection for pad gpio_wk0 0x0: Reserved 0x3: Select gpio_wk0 0x6: Select hw_dbg1 0x7: Select safe_mode	RW	0x7

**Table 18-656. Register Call Summary for Register  
CONTROL\_WKUP\_PAD0\_GPIO\_WK0\_PAD1\_GPIO\_WK1**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[3\]](#)

**Table 18-657. CONTROL\_WKUP\_PAD0\_GPIO\_WK2\_PAD1\_GPIO\_WK3**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Register control for Pads gpio_wk2 and gpio_wk3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO_WK3_WAKEUPEVENT	GPIO_WK3_WAKEUPENABLE	RESERVED					GPIO_WK3_INPUTENABLE	RESERVED	GPIO_WK3_PULLTYPESELECT	GPIO_WK3_PULLUDENABLE	GPIO_WK3_MUXMODE			GPIO_WK2_WAKEUPEVENT	GPIO_WK2_WAKEUPENABLE	RESERVED					GPIO_WK2_INPUTENABLE	RESERVED	GPIO_WK2_PULLTYPESELECT	GPIO_WK2_PULLUDENABLE	GPIO_WK2_MUXMODE						

Bits	Field Name	Description	Type	Reset
31	GPIO_WK3_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	GPIO_WK3_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	GPIO_WK3_INPUTENABLE	Input enable value for pad gpio_wk3 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	GPIO_WK3_PULLTYPESELECT	pullup/down selection for pad gpio_wk3 0x0: pulldown selected 0x1: pullup selected	RW	1
19	GPIO_WK3_PULLUDENABLE	pullup/down enable for pad gpio_wk3 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	GPIO_WK3_MUXMODE	Functional multiplexing selection for pad gpio_wk3 0x0: Reserved 0x3: Select gpio_wk3 0x6: Select hw_dbg4 0x7: Select safe_mode	RW	0x7

Bits	Field Name	Description	Type	Reset
15	GPIO_WK2_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPIO_WK2_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	GPIO_WK2_INPUTENABLE	Input enable value for pad gpio_wk2 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPIO_WK2_PULLTYPESELECT	pullup/down selection for pad gpio_wk2 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPIO_WK2_PULLUDENABLE	pullup/down enable for pad gpio_wk2 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPIO_WK2_MUXMODE	Functional multiplexing selection for pad gpio_wk2 0x0: Reserved 0x3: Select gpio_wk2 0x6: Select hw_dbg3 0x7: Select safe_mode	RW	0x7

**Table 18-658. Register Call Summary for Register CONTROL\_WKUP\_PAD0\_GPIO\_WK2\_PAD1\_GPIO\_WK3**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[3\]](#)

**Table 18-659. CONTROL\_WKUP\_PAD0\_GPIO\_WK4\_PAD1\_SR\_SCL**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Register control for Pads gpio_wk4 and sr_scl Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SR_SCL_WAKEUPEVENT	SR_SCL_WAKEUPENABLE	RESERVED						SR_SCL_INPUTENABLE	RESERVED		SR_SCL_PULLTYPESELECT	SR_SCL_PULLUDENABLE	RESERVED			GPIO_WK4_WAKEUPEVENT	GPIO_WK4_WAKEUPENABLE	RESERVED						GPIO_WK4_INPUTENABLE	RESERVED		GPIO_WK4_PULLTYPESELECT	GPIO_WK4_PULLUDENABLE	GPIO_WK4_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	SR_SCL_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SR_SCL_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	SR_SCL_INPUTENABLE	Input enable value for pad sr_scl 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SR_SCL_PULLTYPESELECT	pullup/down selection for pad sr_scl 0x0: pulldown selected 0x1: pullup selected	RW	1
19	SR_SCL_PULLUDENABLE	pullup/down enable for pad sr_scl 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	RESERVED		R	0x0
15	GPIO_WK4_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	GPIO_WK4_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	GPIO_WK4_INPUTENABLE	Input enable value for pad gpio_wk4 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	GPIO_WK4_PULLTYPESELECT	pullup/down selection for pad gpio_wk4 0x0: pulldown selected 0x1: pullup selected	RW	0
3	GPIO_WK4_PULLUDENABLE	pullup/down enable for pad gpio_wk4 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	GPIO_WK4_MUXMODE	Functional multiplexing selection for pad gpio_wk4 0x0: Reserved 0x3: Select gpio_wk4 0x6: Select hw_dbg5 0x7: Select safe_mode	RW	0x7

**Table 18-660. Register Call Summary for Register  
CONTROL\_WKUP\_PAD0\_GPIO\_WK4\_PAD1\_SR\_SCL**

## Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[3\]](#)



**Table 18-661. CONTROL\_WKUP\_PAD0\_SR\_SDA\_PAD1\_FREF\_XTAL\_IN**

<b>Address Offset</b>	0x0000 004C		
<b>Physical Address</b>	See <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	Register control for Pads sr_sda and fref_xtal_in Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FREF_XTAL_IN_MUXMODE				SR_SDA_WAKEUPEVENT		SR_SDA_WAKEUPENABLE		RESERVED				SR_SDA_INPUTENABLE		RESERVED		SR_SDA_PULLTYPESELECT		SR_SDA_PULLUDENABLE		RESERVED			

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18:16	FREF_XTAL_IN_MUXMODE	Functional multiplexing selection for pad fref_xtal_in 0x0: Select fref_xtal_in 0x4: Select c2c_wakereqin	RW	0x0
15	SR_SDA_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SR_SDA_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	SR_SDA_INPUTENABLE	Input enable value for pad sr_sda 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SR_SDA_PULLTYPESELECT	pullup/down selection for pad sr_sda 0x0: pulldown selected 0x1: pullup selected	RW	1
3	SR_SDA_PULLUDENABLE	pullup/down enable for pad sr_sda 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	RESERVED		R	0x0

**Table 18-662. Register Call Summary for Register CONTROL\_WKUP\_PAD0\_SR\_SDA\_PAD1\_FREF\_XTAL\_IN**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[2\]](#)

**Table 18-663. CONTROL\_WKUP\_PAD0\_FREF\_SLICER\_IN\_PAD1\_FREF\_CLK\_IOREQ**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	Register control for Pads <code>fref_slicer_in</code> and <code>fref_clk_ioreq</code> Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREF_CLK_IOREQ_WAKEUPEVENT	FREF_CLK_IOREQ_WAKEUPENABLE	RESERVED					FREF_CLK_IOREQ_INPUTENABLE	RESERVED			FREF_CLK_IOREQ_PULLTYPESELECT	FREF_CLK_IOREQ_PULLUDENABLE	RESERVED											FREF_SLICER_IN_MUXMODE							

Bits	Field Name	Description	Type	Reset
31	FREF_CLK_IOREQ_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	FREF_CLK_IOREQ_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	FREF_CLK_IOREQ_INPUTENABLE	Input enable value for pad <code>fref_clk_ioreq</code> 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	FREF_CLK_IOREQ_PULLTYPESELECT	pullup/down selection for pad <code>fref_clk_ioreq</code> 0x0: pulldown selected 0x1: pullup selected	RW	0
19	FREF_CLK_IOREQ_PULLUDENABLE	pullup/down enable for pad <code>fref_clk_ioreq</code> 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:3	RESERVED		R	0x0000
2:0	FREF_SLICER_IN_MUXMODE	Functional multiplexing selection for pad <code>fref_slicer_in</code> 0x0: Select <code>fref_slicer_in</code> 0x3: Select <code>gpi_wk5</code> 0x4: Select <code>c2c_wakereqin</code> 0x7: Select <code>safe_mode</code>	RW	0x0

**Table 18-664. Register Call Summary for Register CONTROL\_WKUP\_PAD0\_FREF\_SLICER\_IN\_PAD1\_FREF\_CLK\_IOREQ**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[2\]](#)

**Table 18-665. CONTROL\_WKUP\_PAD0\_FREF\_CLK0\_OUT\_PAD1\_FREF\_CLK3\_REQ**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Register control for Pads fref_clk0_out and fref_clk3_req Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
FREF_CLK3_REQ_WAKEUPEVENT	FREF_CLK3_REQ_WAKEUPENABLE	RESERVED				FREF_CLK3_REQ_INPUTENABLE	RESERVED		FREF_CLK3_REQ_PULLTYPESELECT	FREF_CLK3_REQ_PULLUDENABLE	FREF_CLK3_REQ_MUXMODE		FREF_CLK0_OUT_WAKEUPEVENT	FREF_CLK0_OUT_WAKEUPENABLE	RESERVED				FREF_CLK0_OUT_INPUTENABLE	RESERVED		FREF_CLK0_OUT_PULLTYPESELECT	FREF_CLK0_OUT_PULLUDENABLE	FREF_CLK0_OUT_MUXMODE									

Bits	Field Name	Description	Type	Reset
31	FREF_CLK3_REQ_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	FREF_CLK3_REQ_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	FREF_CLK3_REQ_INPUTENABLE	Input enable value for pad fref_clk3_req 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	FREF_CLK3_REQ_PULLTYPESELECT	pullup/down selection for pad fref_clk3_req 0x0: pulldown selected 0x1: pullup selected	RW	0
19	FREF_CLK3_REQ_PULLUDENABLE	pullup/down enable for pad fref_clk3_req 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
18:16	FREF_CLK3_REQ_MUXMODE	Functional multiplexing selection for pad <code>fref_clk3_req</code> 0x0: Select <code>fref_clk3_req</code> 0x1: Select <code>fref_clk1_req</code> 0x2: Reserved to Non-GP devices 0x3: Select <code>gpio_wk30</code> 0x4: Select <code>c2c_wakereqin</code> 0x5: Select <code>sdmmc2_dat4</code> 0x6: Select <code>hw_dbg7</code> 0x7: Select <code>safe_mode</code>	RW	0x7
15	FREF_CLK0_OUT_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	FREF_CLK0_OUT_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	FREF_CLK0_OUT_INPUTENABLE	Input enable value for pad <code>fref_clk0_out</code> 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	FREF_CLK0_OUT_PULLTYPESELECT	pullup/down selection for pad <code>fref_clk0_out</code> 0x0: pulldown selected 0x1: pullup selected	RW	0
3	FREF_CLK0_OUT_PULLUDENABLE	pullup/down enable for pad <code>fref_clk0_out</code> 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	FREF_CLK0_OUT_MUXMODE	Functional multiplexing selection for pad <code>fref_clk0_out</code> 0x0: Select <code>fref_clk0_out</code> 0x1: Select <code>fref_clk1_req</code> 0x2: Reserved to Non-GP devices 0x3: Select <code>gpio_wk6</code> 0x5: Select <code>sdmmc2_dat7</code> 0x6: Select <code>hw_dbg6</code> 0x7: Select <code>safe_mode</code>	RW	0x7

**Table 18-666. Register Call Summary for Register  
CONTROL\_WKUP\_PAD0\_FREF\_CLK0\_OUT\_PAD1\_FREF\_CLK3\_REQ**

## Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[3\]](#)

**Table 18-667. CONTROL\_WKUP\_PAD0\_FREQ\_CLK3\_OUT\_PAD1\_FREQ\_CLK4\_REQ**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	Register control for Pads freq_clk3_out and freq_clk4_req Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREF_CLK4_REQ_WAKEUPEVENT	FREF_CLK4_REQ_WAKEUPENABLE	RESERVED					FREF_CLK4_REQ_INPUTENABLE	RESERVED		FREF_CLK4_REQ_PULLTYPESELECT	FREF_CLK4_REQ_PULLUDENABLE	FREF_CLK4_REQ_MUXMODE			FREF_CLK3_OUT_WAKEUPEVENT	FREF_CLK3_OUT_WAKEUPENABLE	RESERVED					FREF_CLK3_OUT_INPUTENABLE	RESERVED		FREF_CLK3_OUT_PULLTYPESELECT	FREF_CLK3_OUT_PULLUDENABLE					

Bits	Field Name	Description	Type	Reset
31	FREF_CLK4_REQ_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	FREF_CLK4_REQ_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	FREF_CLK4_REQ_INPUTENABLE	Input enable value for pad freq_clk4_req 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	FREF_CLK4_REQ_PULLTYPESELECT	pullup/down selection for pad freq_clk4_req 0x0: pulldown selected 0x1: pullup selected	RW	0
19	FREF_CLK4_REQ_PULLUDENABLE	pullup/down enable for pad freq_clk4_req 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	FREF_CLK4_REQ_MUXMODE	Functional multiplexing selection for pad freq_clk4_req 0x0: Select freq_clk4_req 0x1: Select freq_clk5_out 0x3: Select gpio_wk7 0x5: Select sdmmc2_dat6 0x6: Select hw_dbg9	RW	0x0
15	FREF_CLK3_OUT_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
14	FREF_CLK3_OUT_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	FREF_CLK3_OUT_INPUTENABLE	Input enable value for pad fref_clk3_out 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	FREF_CLK3_OUT_PULLTYPESELECT	pullup/down selection for pad fref_clk3_out 0x0: pulldown selected 0x1: pullup selected	RW	0
3	FREF_CLK3_OUT_PULLUDENABLE	pullup/down enable for pad fref_clk3_out 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	FREF_CLK3_OUT_MUXMODE	Functional multiplexing selection for pad fref_clk3_out 0x0: Select fref_clk3_out 0x1: Select fref_clk2_req 0x2: Reserved to non-GP devices 0x3: Select gpio_wk31 0x4: Select c2c_wakereqout 0x5: Select sdmmc2_dat5 0x6: Select hw_dbg8 0x7: Select safe_mode	RW	0x7

**Table 18-668. Register Call Summary for Register CONTROL\_WKUP\_PAD0\_FREF\_CLK3\_OUT\_PAD1\_FREF\_CLK4\_REQ**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[3\]](#)

**Table 18-669. CONTROL\_WKUP\_PAD0\_FREF\_CLK4\_OUT\_PAD1\_SYS\_32K**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Register control for Pads fref_clk4_out and sys_32k Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_32K_WAKEUPEVENT	SYS_32K_WAKEUPENABLE	RESERVED				SYS_32K_INPUTENABLE	RESERVED	SYS_32K_PULLTYPESELECT	SYS_32K_PULLUDENABLE	RESERVED	FREF_CLK4_OUT_WAKEUPEVENT	FREF_CLK4_OUT_WAKEUPENABLE	RESERVED				FREF_CLK4_OUT_INPUTENABLE	RESERVED	FREF_CLK4_OUT_PULLTYPESELECT	FREF_CLK4_OUT_PULLUDENABLE	RESERVED		FREF_CLK4_OUT_MUXMODE								

Bits	Field Name	Description	Type	Reset
31	SYS_32K_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SYS_32K_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	SYS_32K_INPUTENABLE	Input enable value for pad sys_32k 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SYS_32K_PULLTYPESELECT	pullup/down selection for pad sys_32k 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SYS_32K_PULLUDENABLE	pullup/down enable for pad sys_32k 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	0
18:16	RESERVED		R	0x0
15	FREF_CLK4_OUT_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	FREF_CLK4_OUT_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	FREF_CLK4_OUT_INPUTENABLE	Input enable value for pad fref_clk4_out 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	FREF_CLK4_OUT_PULLTYPESELECT	pullup/down selection for pad fref_clk4_out 0x0: pulldown selected 0x1: pullup selected	RW	0

Bits	Field Name	Description	Type	Reset
3	FREF_CLK4_OUT_PULLUDENABLE	pullup/down enable for pad fref_clk4_out 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	FREF_CLK4_OUT_MUXMODE	Functional multiplexing selection for pad fref_clk4_out 0x0: Select fref_clk4_out 0x3: Select gpio_wk8 0x6: Select hw_dbg10	RW	0x0

**Table 18-670. Register Call Summary for Register  
CONTROL\_WKUP\_PAD0\_FREF\_CLK4\_OUT\_PAD1\_SYS\_32K**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[3\]](#)

**Table 18-671. CONTROL\_WKUP\_PAD0\_SYS\_NRESPWRON\_PAD1\_SYS\_NRESWARM**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Register control for Pads sys_nrespwron and sys_nreswarm Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_NRESWARM_WAKEUPEVENT	SYS_NRESWARM_WAKEUPENABLE	RESERVED														SYS_NRESPWRON_PULLTYPESELECT	SYS_NRESPWRON_PULLUDENABLE	RESERVED													

Bits	Field Name	Description	Type	Reset
31	SYS_NRESWARM_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SYS_NRESWARM_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:5	RESERVED		R	0x0000000
4	SYS_NRESPWRON_PULLTYPESELECT	pullup/down selection for pad sys_nrespwron 0x0: pulldown selected 0x1: pullup selected	RW	0



Bits	Field Name	Description	Type	Reset
3	SYS_NRESPWRON_PULLUENABLE	pullup/down enable for pad sys_nrespwron 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	0
2:0	RESERVED		R	0x0

**Table 18-672. Register Call Summary for Register CONTROL\_WKUP\_PAD0\_SYS\_NRESPWRON\_PAD1\_SYS\_NRESWARM**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[2\]](#)

**Table 18-673. CONTROL\_WKUP\_PAD0\_SYS\_PWR\_REQ\_PAD1\_SYS\_PWRON\_RESET\_OUT**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Register control for Pads sys_pwr_req and sys_pwron_reset_out Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
SYS_PWRON_RESET_OUT_WAKEUPEVENT	SYS_PWRON_RESET_OUT_WAKEUPENABLE	RESERVED						SYS_PWRON_RESET_OUT_INPUTENABLE	RESERVED			SYS_PWRON_RESET_OUT_PULLYPESELECT	SYS_PWRON_RESET_OUT_PULLUENABLE	SYS_PWRON_RESET_OUT_MUXMODE			SYS_PWR_REQ_WAKEUPEVENT	SYS_PWR_REQ_WAKEUPENABLE	RESERVED						SYS_PWR_REQ_INPUTENABLE	RESERVED			SYS_PWR_REQ_PULLYPESELECT	SYS_PWR_REQ_PULLUENABLE	RESERVED		

Bits	Field Name	Description	Type	Reset
31	SYS_PWRON_RESET_OUT_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SYS_PWRON_RESET_OUT_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	SYS_PWRON_RESET_OUT_INPUTENABLE	Input enable value for pad sys_pwron_reset_out 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
20	SYS_PWRON_RESET_OUT_PULLTYPESELECT	pullup/down selection for pad sys_pwrn_reset_out 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SYS_PWRON_RESET_OUT_PULLUDENABLE	pullup/down enable for pad sys_pwrn_reset_out 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	SYS_PWRON_RESET_OUT_MUXMODE	Functional multiplexing selection for pad sys_pwrn_reset_out 0x0: Select sys_pwrn_reset_out 0x3: Select gpio_wk29 0x6: Select hw_dbg11	RW	0x0
15	SYS_PWR_REQ_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SYS_PWR_REQ_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	SYS_PWR_REQ_INPUTENABLE	Input enable value for pad sys_pwr_req 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	SYS_PWR_REQ_PULLTYPESELECT	pullup/down selection for pad sys_pwr_req 0x0: pulldown selected 0x1: pullup selected	RW	1
3	SYS_PWR_REQ_PULLUDENABLE	pullup/down enable for pad sys_pwr_req 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	RESERVED		R	0x0

**Table 18-674. Register Call Summary for Register  
CONTROL\_WKUP\_PAD0\_SYS\_PWR\_REQ\_PAD1\_SYS\_PWRON\_RESET\_OUT**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[3\]](#)

**Table 18-675. CONTROL\_WKUP\_PAD0\_SYS\_BOOT6\_PAD1\_SYS\_BOOT7**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Register control for Pads sys_boot6 and sys_boot7 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
SYS_BOOT7_WAKEUPEVENT	SYS_BOOT7_WAKEUPENABLE	RESERVED						SYS_BOOT7_INPUTENABLE	RESERVED						SYS_BOOT7_PULLTYPESELECT	SYS_BOOT7_PULLUDENABLE	SYS_BOOT7_MUXMODE			SYS_BOOT6_WAKEUPEVENT	SYS_BOOT6_WAKEUPENABLE	RESERVED						SYS_BOOT6_INPUTENABLE	RESERVED						SYS_BOOT6_PULLTYPESELECT	SYS_BOOT6_PULLUDENABLE	SYS_BOOT6_MUXMODE		

Bits	Field Name	Description	Type	Reset
31	SYS_BOOT7_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	SYS_BOOT7_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	SYS_BOOT7_INPUTENABLE	Input enable value for pad sys_boot7 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	SYS_BOOT7_PULLTYPESELECT	pullup/down selection for pad sys_boot7 0x0: pulldown selected 0x1: pullup selected	RW	0
19	SYS_BOOT7_PULLUDENABLE	pullup/down enable for pad sys_boot7 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	0
18:16	SYS_BOOT7_MUXMODE	Functional multiplexing selection for pad sys_boot7 0x0: Select sys_boot7 0x1: Select dpm_emu19 0x3: Select gpio_wk10 0x6: Select hw_dbg13 0x7: Select safe_mode	RW	0x0
15	SYS_BOOT6_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SYS_BOOT6_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	SYS_BOOT6_INPUTENABLE	Input enable value for pad sys_boot6 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	SYS_BOOT6_PULLTYPESELECT	pullup/down selection for pad sys_boot6 0x0: pulldown selected 0x1: pullup selected	RW	0
3	SYS_BOOT6_PULLUDENABLE	pullup/down enable for pad sys_boot6 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	0
2:0	SYS_BOOT6_MUXMODE	Functional multiplexing selection for pad sys_boot6 0x0: Select sys_boot6 0x1: Select dpm_emu18 0x3: Select gpio_wk9 0x4: Select c2c_wakereqout 0x6: Select hw_dbg12 0x7: Select safe_mode	RW	0x0

**Table 18-676. Register Call Summary for Register  
CONTROL\_WKUP\_PAD0\_SYS\_BOOT6\_PAD1\_SYS\_BOOT7**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)
- [Observability Signals Multiplexing at Device Core Control Level: \[2\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[3\]](#)

**Table 18-677. CONTROL\_WKUP\_PAD0\_JTAG\_NTRST\_PAD1\_JTAG\_TCK**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Register control for Pads jtag_ntrst and jtag_tck Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
JTAG_TCK_WAKEUPEVENT	JTAG_TCK_WAKEUPENABLE	RESERVED						JTAG_TCK_INPUTENABLE	RESERVED		JTAG_TCK_PULLTYPESELECT	JTAG_TCK_PULLUDENABLE	JTAG_TCK_MUXMODE				JTAG_NTRST_WAKEUPEVENT	JTAG_NTRST_WAKEUPENABLE	RESERVED						JTAG_NTRST_INPUTENABLE	RESERVED		JTAG_NTRST_PULLTYPESELECT	JTAG_NTRST_PULLUDENABLE	RESERVED			

Bits	Field Name	Description	Type	Reset
31	JTAG_TCK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	JTAG_TCK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0

Bits	Field Name	Description	Type	Reset
29:25	RESERVED		R	0x00
24	JTAG_TCK_INPUTENABLE	Input enable value for pad jtag_tck 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:21	RESERVED		R	0x0
20	JTAG_TCK_PULLTYPESELECT	pullup/down selection for pad jtag_tck 0x0: pulldown selected 0x1: pullup selected	RW	0
19	JTAG_TCK_PULLUDENABLE	pullup/down enable for pad jtag_tck 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	JTAG_TCK_MUXMODE	Functional multiplexing selection for pad jtag_tck 0x0: Select jtag_tck 0x7: Select safe_mode	RW	0x0
15	JTAG_NTRST_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	JTAG_NTRST_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	JTAG_NTRST_INPUTENABLE	Input enable value for pad jtag_nrst 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	JTAG_NTRST_PULLTYPESELECT	pullup/down selection for pad jtag_nrst 0x0: pulldown selected 0x1: pullup selected	RW	0
3	JTAG_NTRST_PULLUDENABLE	pullup/down enable for pad jtag_nrst 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	RESERVED		R	0x0

**Table 18-678. Register Call Summary for Register CONTROL\_WKUP\_PAD0\_JTAG\_NTRST\_PAD1\_JTAG\_TCK**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[2\]](#)

**Table 18-679. CONTROL\_WKUP\_PAD0\_JTAG\_RTCK\_PAD1\_JTAG\_TMS\_TMSC**

<b>Address Offset</b>	0x0000 0070		
<b>Physical Address</b>	See <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	Register control for Pads jtag_rtck and jtag_tms_tmsc Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAG_TMS_TMSC_WAKEUPEVENT	JTAG_TMS_TMSC_WAKEUPENABLE	RESERVED					JTAG_TMS_TMSC_INPUTENABLE	RESERVED					JTAG_TMS_TMSC_MUXMODE	JTAG_RTCK_WAKEUPEVENT	JTAG_RTCK_WAKEUPENABLE	RESERVED							JTAG_RTCK_PULLTYPESELECT	JTAG_RTCK_PULLUDENABLE	RESERVED						

Bits	Field Name	Description	Type	Reset
31	JTAG_TMS_TMSC_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	JTAG_TMS_TMSC_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x00
24	JTAG_TMS_TMSC_INPUTENABLE	Input enable value for pad jtag_tms_tmsc 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
23:19	RESERVED		R	0x00
18:16	JTAG_TMS_TMSC_MUXMODE	Functional multiplexing selection for pad jtag_tms_tmsc 0x0: Select jtag_tms_tmsc 0x7: Select safe_mode	RW	0x0
15	JTAG_RTCK_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	JTAG_RTCK_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:5	RESERVED		R	0x000
4	JTAG_RTCK_PULLTYPESELECT	pullup/down selection for pad jtag_rtck 0x0: pulldown selected 0x1: pullup selected	RW	0
3	JTAG_RTCK_PULLUDENABLE	pullup/down enable for pad jtag_rtck 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1

Bits	Field Name	Description	Type	Reset
2:0	RESERVED		R	0x0

**Table 18-680. Register Call Summary for Register CONTROL\_WKUP\_PAD0\_JTAG\_RTCK\_PAD1\_JTAG\_TMS\_TMSC**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[2\]](#)

**Table 18-681. CONTROL\_WKUP\_PAD0\_JTAG\_TDI\_PAD1\_JTAG\_TDO**

<b>Address Offset</b>	0x0000 0074	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Register control for Pads jtag_tdi and jtag_tdo Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAG_TDO_WAKEUPEVENT	JTAG_TDO_WAKEUPENABLE	RESERVED					JTAG_TDO_INPUTENABLE	RESERVED	JTAG_TDO_PULLTYPESELECT	JTAG_TDO_PULLUDENABLE	RESERVED	JTAG_TDI_WAKEUPEVENT	JTAG_TDI_WAKEUPENABLE	RESERVED					JTAG_TDI_INPUTENABLE	RESERVED	JTAG_TDI_PULLTYPESELECT	JTAG_TDI_PULLUDENABLE	RESERVED								

Bits	Field Name	Description	Type	Reset
31	JTAG_TDO_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
30	JTAG_TDO_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
29:25	RESERVED		R	0x000
24	JTAG_TDO_INPUTENABLE	Input enable value for pad jtag_tdo 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	0x1
23:21	RESERVED		R	0x000
20	JTAG_TDO_PULLTYPESELECT	pullup/down selection for pad jtag_tdo 0x0: pulldown selected 0x1: pullup selected	RW	1
19	JTAG_TDO_PULLUDENABLE	pullup/down enable for pad jtag_tdo 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
18:16	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
15	JTAG_TDI_WAKEUPEVENT	Pad_x wake-up event status latched in the IO Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	JTAG_TDI_WAKEUPENABLE	Input pad wake-up enable (and OFF mode input enable value) 0x0: wake-up detection is disabled 0x1: wake-up detection is enabled	RW	0
13:9	RESERVED		R	0x00
8	JTAG_TDI_INPUTENABLE	Input enable value for pad jtag_tdi 0x0: Input buffere of I/O cell is disabled 0x1: Input buffere of I/O cell is enabled	RW	1
7:5	RESERVED		R	0x0
4	JTAG_TDI_PULLTYPESELECT	pullup/down selection for pad jtag_tdi 0x0: pulldown selected 0x1: pullup selected	RW	1
3	JTAG_TDI_PULLUDENABLE	pullup/down enable for pad jtag_tdi 0x0: pullup/down disabled 0x1: pullup/down enabled	RW	1
2:0	RESERVED		R	0x0

**Table 18-682. Register Call Summary for Register  
CONTROL\_WKUP\_PAD0\_JTAG\_TDI\_PAD1\_JTAG\_TDO**

Control Module Functional Description

- [Pad Multiplexing Register Fields: \[0\] \[1\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[2\]](#)

**Table 18-683. CONTROL\_WKUP\_PADCONF\_WAKEUPEVENT\_0**

<b>Address Offset</b>	0x0000 007C	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED								JTAG_TDO_DUPLICATEWAKEUPEVENT	JTAG_TDI_DUPLICATEWAKEUPEVENT	JTAG_TMS_TMSC_DUPLICATEWAKEUPEVENT	JTAG_RTCK_DUPLICATEWAKEUPEVENT	JTAG_TCK_DUPLICATEWAKEUPEVENT	JTAG_NTRST_DUPLICATEWAKEUPEVENT	SYS_BOOT7_DUPLICATEWAKEUPEVENT	SYS_BOOT6_DUPLICATEWAKEUPEVENT	SYS_PWRON_RESET_OUT_DUPLICATEWAKEUPEVENT	SYS_PWR_REQ_DUPLICATEWAKEUPEVENT	SYS_NRESWARM_DUPLICATEWAKEUPEVENT	SYS_32K_DUPLICATEWAKEUPEVENT	FREF_CLK4_OUT_DUPLICATEWAKEUPEVENT	FREF_CLK4_REQ_DUPLICATEWAKEUPEVENT	FREF_CLK3_OUT_DUPLICATEWAKEUPEVENT	FREF_CLK3_REQ_DUPLICATEWAKEUPEVENT	FREF_CLK0_OUT_DUPLICATEWAKEUPEVENT	FREF_CLK0_REQ_DUPLICATEWAKEUPEVENT	SR_SDA_DUPLICATEWAKEUPEVENT	SR_SCL_DUPLICATEWAKEUPEVENT	GPIO_WK4_DUPLICATEWAKEUPEVENT	GPIO_WK3_DUPLICATEWAKEUPEVENT	GPIO_WK2_DUPLICATEWAKEUPEVENT	GPIO_WK1_DUPLICATEWAKEUPEVENT	GPIO_WK0_DUPLICATEWAKEUPEVENT

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	JTAG_TDO_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad jtag_tdo Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
23	JTAG_TDI_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad jtag_tdi Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
22	JTAG_TMS_TMSC_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad jtag_tms_tmsc Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
21	JTAG_RTCK_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad jtag_rtck Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
20	JTAG_TCK_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad jtag_tck Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
19	JTAG_NTRST_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad jtag_ntrst Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
18	SYS_BOOT7_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sys_boot7 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
17	SYS_BOOT6_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sys_boot6 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
16	SYS_PWRON_RESET_OUT_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sys_pwrn_reset_out Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
15	SYS_PWR_REQ_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sys_pwr_req Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
14	SYS_NRESWARM_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sys_nreswarm Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
13	SYS_32K_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sys_32k Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
12	FREF_CLK4_OUT_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad fref_clk4_out Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
11	FREF_CLK4_REQ_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad fref_clk4_req Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
10	FREF_CLK3_OUT_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad fref_clk3_out Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
9	FREF_CLK3_REQ_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad fref_clk3_req Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
8	FREF_CLK0_OUT_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad fref_clk0_out Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
7	FREF_CLK_IOREQ_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad fref_clk_ioreq Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
6	SR_SDA_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sr_sda Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
5	SR_SCL_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad sr_scl Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
4	GPIO_WK4_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpio_wk4 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
3	GPIO_WK3_DUPLICATEWAKEUPEVENT	Wake-up event status latched in the I/O for pad gpio_wk3 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

Bits	Field Name	Description	Type	Reset
2	GPIO_WK2_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpio_wk2 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
1	GPIO_WK1_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpio_wk1 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0
0	GPIO_WK0_DUPLICATEWAKE UPEVENT	Wake-up event status latched in the I/O for pad gpio_wk0 Read 0x0: No wake-up event occurred Read 0x1: A wake-up event occurred	R	0

**Table 18-684. Register Call Summary for Register CONTROL\_WKUP\_PADCONF\_WAKEUPEVENT\_0**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-685. CONTROL\_SMART1NOPMIO\_PADCONF\_0**

<b>Address Offset</b>	0x0000 05A0	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	SMART1 NOPM I/O control 0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREF_DR0_SC		RESERVED						GPIO_DR7_SC		RESERVED		DPM_DR0_SC		GPIOWK4_DR0_SC		RESERVED															

Bits	Field Name	Description	Type	Reset
31:30	FREF_DR0_SC	Slew rate control for group fref_dr0 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
29:20	RESERVED		R	0x000
19:18	GPIO_DR7_SC	Slew rate control for group gpio_dr7 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
17:16	RESERVED		R	0x0
15:14	DPM_DR0_SC	Slew rate control for group dpm_dr0 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
13:12	GPIOWK4_DR0_SC	Slew rate control for group gpiowk4_dr0 Refer to <a href="#">Section 18.4.12.6.3</a> , section Low Speed I/Os Combined Slew Rate vs TL Length and Load Settings for more details on SR settings.	RW	0x0
11:0	RESERVED	Reserved	R	0x000

**Table 18-686. Register Call Summary for Register CONTROL\_SMART1NOPMIO\_PADCONF\_0**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[5\]](#)

**Table 18-687. CONTROL\_SMART1NOPMIO\_PADCONF\_1**

<b>Address Offset</b>	0x0000 05A4	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	SMART1 NOPM I/O control 1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREF_DR0_LB		RESERVED										GPIO_DR7_LB		RESERVED		DPM_DR0_LB		GPIOWK4_DR0_LB		RESERVED											

Bits	Field Name	Description	Type	Reset
31:30	FREF_DR0_LB	Load control for group fref_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
29:20	RESERVED		R	0x000
19:18	GPIO_DR7_LB	Load control for group gpio_dr7 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
17:16	RESERVED		R	0x0
15:14	DPM_DR0_LB	Load control for group dpm_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0

Bits	Field Name	Description	Type	Reset
13:12	GPIOWK4_DR0_LB	Load control for group gpiowk4_dr0 0x0: TL Length= [2cm-20cm] / Farend cap per TL= [1pF-10pF] 0x1: Dual TL Length= [20cm-40cm] / Farend cap per TL= [1pF-10pF] 0x2: TL Length = [2cm-20cm] / Farend cap per TL= [10pF-20pF] 0x3: Dual TL Length = [20cm-40cm] / Farend cap per TL= [10pF-20pF]	RW	0x0
11:0	RESERVED		R	0x000

**Table 18-688. Register Call Summary for Register CONTROL\_SMART1NOPMIO\_PADCONF\_1**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[5\]](#)

**Table 18-689. CONTROL\_WKUP\_PADCONF\_MODE**

<b>Address Offset</b>	0x0000 05A8	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	PAD Voltage Mode control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDS_DV_FREF	VDDS_DV_BANK2_SHARED1	RESERVED																													

Bits	Field Name	Description	Type	Reset
31	VDDS_DV_FREF	PAD Voltage level control for vdds_dv_fref 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
30	VDDS_DV_BANK2_SHARED1	PAD Voltage level control for vdds_dv_bank2_shared1 0x0: VDDS = 1.8V 0x1: VDDS = 1.2V	RW	0
29:0	RESERVED		R	0x0000 0000

**Table 18-690. Register Call Summary for Register CONTROL\_WKUP\_PADCONF\_MODE**

Control Module Functional Description

- [Dual Voltage-Supplied Peripheral Controls: \[0\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[1\]](#)

**Table 18-691. CONTROL\_XTAL\_OSCILLATOR**

<b>Address Offset</b>	0x0000 05AC		
<b>Physical Address</b>	See <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	XTAL OSCILLATOR control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSCILLATOR_BOOST	OSCILLATOR_OS_OUT	RESERVED																													

Bits	Field Name	Description	Type	Reset
31	OSCILLATOR_BOOST	Fast startup control 0x0: Fast startup is disabled 0x1: Fast startup is enabled	RW	1
30	OSCILLATOR_OS_OUT	Oscillator output Read 0x0: low to high transition in BOOST mode Read 0x1: BOOST is disabled	R	0
29:0	RESERVED		R	0x0000 0000

**Table 18-692. Register Call Summary for Register CONTROL\_XTAL\_OSCILLATOR**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-693. CONTROL\_SMART3NOPMIO\_PADCONF\_0**

<b>Address Offset</b>	0x0000 05B0		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	SMART3 NOPM IO control 0 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREF_DR1_MB	FREF_DR5_MB	FREF_DR6_MB	FREF_DR7_MB	FREF_DR4_MB	RESERVED																										

Bits	Field Name	Description	Type	Reset
31:30	FREF_DR1_MB	50-Ω output buffer mode control for group fref_dr1 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1

Bits	Field Name	Description	Type	Reset
29:28	FREF_DR5_MB	50-Ω output buffer mode control for group fref_dr5 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
27:26	FREF_DR6_MB	50-Ω output buffer mode control for group fref_dr6 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
25:24	FREF_DR7_MB	50-Ω output buffer mode control for group Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
23:22	FREF_DR4_MB	50-Ω output buffer mode control for group fref_dr4 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0x1
21:0	RESERVED		R	0x000000

**Table 18-694. Register Call Summary for Register CONTROL\_SMART3NOPMIO\_PADCONF\_0**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [50-Ω Output Buffer I/Os with Combined Mode and Load Settings: \[1\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[2\] \[3\] \[4\] \[5\] \[6\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[7\]](#)

**Table 18-695. CONTROL\_SMART3NOPMIO\_PADCONF\_1**

<b>Address Offset</b>	0x0000 05B4	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>		
<b>Description</b>	SMART3 NOPM IO control 1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREF_DR1_LB0	FREF_DR5_LB0	FREF_DR6_LB0	FREF_DR7_LB0	FREF_DR4_LB0	RESERVED																										

Bits	Field Name	Description	Type	Reset
31	FREF_DR1_LB0	50-Ω output buffer load control for group fref_dr1 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
30	FREF_DR5_LB0	50-Ω output buffer load control for group fref_dr5 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
29	FREF_DR6_LB0	50-Ω output buffer load control for group fref_dr6 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
28	FREF_DR7_LB0	50-Ω output buffer load control for group fref_dr7 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0

Bits	Field Name	Description	Type	Reset
27	FREF_DR4_LB0	50-Ω output buffer load control for group fref_dr4 Refer to 50-Ω output buffer I/Os with combined Mode and Load Settings for more details on mode settings.	RW	0
26:0	RESERVED		R	0x0000000

**Table 18-696. Register Call Summary for Register CONTROL\_SMART3NOPMIO\_PADCONF\_1**

## Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [50-Ω Output Buffer I/Os with Combined Mode and Load Settings: \[1\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[2\] \[3\] \[4\] \[5\] \[6\]](#)

## Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[7\]](#)

**Table 18-697. CONTROL\_GPIOWK**

<b>Address Offset</b>	0x600	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>		
<b>Description</b>	GPIOWK pads I/O control Access conditions. Read: unrestricted; Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAD_GPIO_WK1_LOW	Reserved	PAD_GPIO_WK2_LOW	GPIOWK_IO_PWRDNZ	Reserved																											

Bits	Field Name	Description	Type	Reset
31	PAD_GPIO_WK1_LOW	LOW enable for gpio_wk1 PAD 0: Strong pulldown is disabled 1: Strong Pull-down after reset release	RW	1
30	Reserved		RO Rreturns 0s	0
29	PAD_GPIO_WK2_LOW	LOW enable for gpio_wk2 PAD 0: Strong Pull-down is disabled 1: Strong Pull-down after reset release	RW	1
28	GPIOWK_IO_PWRDNZ	PWRDNZ control to gpiowk0-gpiowk2 IOs This signal is used to protect gpiowk0-through gpiowk2 associated IO cell 0: SW keeps this bit in 0b0 when SIM_VDD5 is not stable 1: SW sets this bit to 1'b1 after SIM_VDD5 is stabilized	RW	0



Bits	Field Name	Description	Type	Reset
27:0	Reserved		RO Rreturns 0s	0x0000000

**Table 18-698. Register Call Summary for Register CONTROL\_GPIOWK**

Control Module Functional Description

- [Extended-Drain I/O and PBIAS Cell: \[0\] \[1\]](#)
- [Extended-Drain I/O: \[2\] \[3\] \[4\] \[5\]](#)

Control Module Programming Guide

- [Extended-Drain I/Os and PBIAS Cells Programming Guide: \[6\] \[7\] \[8\] \[9\] \[10\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[11\]](#)

**Table 18-699. CONTROL\_I2C\_2**

<b>Address Offset</b>	0x0000 0604	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	I2C pads control 2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SR_SDA_GLFENB	SR_SDA_LOAD_BITS	SR_SDA_PULLUPRESX	SR_SCL_GLFENB	SR_SCL_LOAD_BITS	SR_SCL_PULLUPRESX	RESERVED																									

Bits	Field Name	Description	Type	Reset
31	SR_SDA_GLFENB	Active_high glitch free operation enable pin for sr i2c receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0
30:29	SR_SDA_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for sr i2c 0x0: 4.5 kΩ (5-15 pF) / 1.66 kΩ (5-12 pF) 0x1: 2.1 kΩ (15-50 pF) / 920 Ω (12-25 pF) 0x2: 860 Ω (50-150 pF) / 500 Ω (25-50 pF) 0x3: N.A / 300 Ω (50-80 pF)	RW	0x0
28	SR_SDA_PULLUPRESX	Active_low internal pull_up resistor enabled for sr i2c 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resisitor	RW	0
27	SR_SCL_GLFENB	Active_high glitch free operation enable pin for sr i2c receiver 0x0: Disable i2c4 glitch free operation 0x1: Enable i2c4 glitch free operation	RW	0

Bits	Field Name	Description	Type	Reset
26:25	SR_SCL_LOAD_BITS	Internal Pull up resistor in Fast/High_speed mode for sr i2c 0x0: 4.5 kΩ (5-15 pF) / 1.66 kΩ (5-12 pF) 0x1: 2.1 kΩ (15-50 pF) / 920 Ω (12-25 pF) 0x2: 860 Ω (50-150 pF) / 500 Ω (25-50 pF) 0x3: N.A / 300 Ω (50-80 pF)	RW	0x0
24	SR_SCL_PULLUPRESX	Active_low internal pull_up resistor enabled for sr i2c 0x0: Enable Internal Pull up resistor 0x1: Disable Internal Pull up resisitor	RW	0
23:0	RESERVED		R	0x000000

**Table 18-700. Register Call Summary for Register CONTROL\_I2C\_2**

Control Module Functional Description

- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- [Device Interfaces Signal Group Controls Mapping: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[7\]](#)

**Table 18-701. CONTROL\_JTAG**

<b>Address Offset</b>	0x0000 0608	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	JTAG pads control Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTAG_NTRST_EN	JTAG_TCK_EN	JTAG_RTCK_EN	JTAG_TDI_EN	JTAG_TDO_EN	RESERVED																										

Bits	Field Name	Description	Type	Reset
31	JTAG_NTRST_EN	bus holder enable for jtag_nrst PAD 0x0: Disable bus holder functionality 0x1: Enable bus holder functionality	RW	0
30	JTAG_TCK_EN	bus holder enable for jtag_tck PAD 0x0: Disable bus holder functionality 0x1: Enable bus holder functionality	RW	0
29	JTAG_RTCK_EN	bus holder enable for jtag_rtck PAD 0x0: Disable bus holder functionality 0x1: Enable bus holder functionality	RW	0
28	JTAG_TDI_EN	bus holder enable for jtag_tdi PAD 0x0: Disable bus holder functionality 0x1: Enable bus holder functionality	RW	0
27	JTAG_TDO_EN	bus holder enable for jtag_tdo PAD 0x0: Disable bus holder functionality 0x1: Enable bus holder functionality	RW	0
26:0	RESERVED		R	0x0000000

**Table 18-702. Register Call Summary for Register CONTROL\_JTAG**

- Control Module Functional Description
- [Signal Integrity Parameter Controls Overview: \[0\]](#)
- Control Module Register Manual
- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[1\]](#)

**Table 18-703. CONTROL\_SYS**

<b>Address Offset</b>	0x0000 060C	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	SYS controls Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31	SYS_NRESWARM_PIPU	Pull up enable for sys_nreswarm PAD 0x0: Enable pull up 0x1: Disable pull up	RW	0
30:0	RESERVED		R	0x0000 0000

**Table 18-704. Register Call Summary for Register CONTROL\_SYS**

- Control Module Register Manual
- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-705. CONTROL\_WKUP\_CONTROL\_SPARE\_RW**

<b>Address Offset</b>	0x0000 0614	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Physical Address</b>	See <a href="#">Table 18-648</a>		
<b>Description</b>	Wake-up control spare RW Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW																															

Bits	Field Name	Description	Type	Reset
31:0	WKUP_CONTROL_SPARE_RW	Wake-up control spare register bits RW	RW	0x0000 0000

**Table 18-706. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW**

- Control Module Register Manual
- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-707. CONTROL\_WKUP\_CONTROL\_SPARE\_R**

<b>Address Offset</b>	0x0000 0618		
<b>Physical Address</b>	See <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	Wake-up control spare R Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_R																															

Bits	Field Name	Description	Type	Reset
31:0	WKUP_CONTROL_SPARE_R	Wake-up control spare register bits R	R	0x0000 0000

**Table 18-708. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_R**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-709. CONTROL\_WKUP\_CONTROL\_SPARE\_R\_C0**

<b>Address Offset</b>	0x0000 061C		
<b>Physical Address</b>	See <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	Wake-up control spare RC Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_R_C0	WKUP_CONTROL_SPARE_R_C1	WKUP_CONTROL_SPARE_R_C2	WKUP_CONTROL_SPARE_R_C3	WKUP_CONTROL_SPARE_R_C4	WKUP_CONTROL_SPARE_R_C5	WKUP_CONTROL_SPARE_R_C6	WKUP_CONTROL_SPARE_R_C7	RESERVED																							

Bits	Field Name	Description	Type	Reset
31 0	WKUP_CONTROL_SPARE_R_C	Wake-up control spare register bits RC	RW (Write 1 to Clear)	0
30 1	WKUP_CONTROL_SPARE_R_C	Wake-up control spare register bits RC	RW (Write 1 to Clear)	0
29 2	WKUP_CONTROL_SPARE_R_C	Wake-up control spare register bits RC	RW (Write 1 to Clear)	0
28 3	WKUP_CONTROL_SPARE_R_C	Wake-up control spare register bits RC	RW (Write 1 to Clear)	0
27 4	WKUP_CONTROL_SPARE_R_C	Wake-up control spare register bits RC	RW (Write 1 to Clear)	0

Bits	Field Name	Description	Type	Reset
26 5	WKUP_CONTROL_SPARE_R_C	Wake-up control spare register bits RC	RW (Write 1 to Clear)	0
25 6	WKUP_CONTROL_SPARE_R_C	Wake-up control spare register bits RC	RW (Write 1 to Clear)	0
24 7	WKUP_CONTROL_SPARE_R_C	Wake-up control spare register bits RC	RW (Write 1 to Clear)	0
23:0	RESERVED		R	0x000000

**Table 18-710. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_R\_C0**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-711. CONTROL\_WKUP\_CONTROL\_SPARE\_RW1**

<b>Address Offset</b>	0x0000 0620		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare RW1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW1																															

Bits	Field Name	Description	Type	Reset
31:0 1	WKUP_CONTROL_SPARE_RW	wkup control spare register bits RW1	RW	0x0000 0000

**Table 18-712. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW1**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-713. CONTROL\_WKUP\_CONTROL\_SPARE\_RW2**

<b>Address Offset</b>	0x0000 0624		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare RW2 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW2																															

Bits	Field Name	Description	Type	Reset
31:0 2	WKUP_CONTROL_SPARE_RW	wkup control spare register bits RW2	RW	0x0000 0000

**Table 18-714. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW2**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-715. CONTROL\_WKUP\_CONTROL\_SPARE\_RW3**

<b>Address Offset</b>	0x0000 0628		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare RW3 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW3																															

Bits	Field Name	Description	Type	Reset
31:0 3	WKUP_CONTROL_SPARE_RW	wkup control spare register bits RW3	RW	0x0000 0000

**Table 18-716. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW3**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-717. CONTROL\_WKUP\_CONTROL\_SPARE\_RW4**

<b>Address Offset</b>	0x0000 062C		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare RW4 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW4																															

Bits	Field Name	Description	Type	Reset
31:0 4	WKUP_CONTROL_SPARE_RW	wkup control spare register bits RW4	RW	0x0000 0000

**Table 18-718. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW4**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-719. CONTROL\_WKUP\_CONTROL\_SPARE\_RW5**

<b>Address Offset</b>	0x0000 0630		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare RW5 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW5																															

Bits	Field Name	Description	Type	Reset
31:0 5	WKUP_CONTROL_SPARE_RW	wkup control spare register bits RW5	RW	0x0000 0000

**Table 18-720. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW5**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-721. CONTROL\_WKUP\_CONTROL\_SPARE\_RW6**

<b>Address Offset</b>	0x0000 0634		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare RW6 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW6																															

Bits	Field Name	Description	Type	Reset
31:0	WKUP_CONTROL_SPARE_RW6	wkup control spare register bits RW6	RW	0x0000 0000

**Table 18-722. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW6**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-723. CONTROL\_WKUP\_CONTROL\_SPARE\_RW7**

<b>Address Offset</b>	0x0000 0638		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare RW7 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW7																															

Bits	Field Name	Description	Type	Reset
31:0	WKUP_CONTROL_SPARE_RW7	wkup control spare register bits RW7	RW	0x0000 0000

**Table 18-724. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW7**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-725. CONTROL\_WKUP\_CONTROL\_SPARE\_RW8**

<b>Address Offset</b>	0x0000 063C		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare RW8 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW8																															

Bits	Field Name	Description	Type	Reset
31:0	WKUP_CONTROL_SPARE_RW8	wkup control spare register bits RW8	RW	0x0000 0000

**Table 18-726. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW8**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-727. CONTROL\_WKUP\_CONTROL\_SPARE\_RW9**

<b>Address Offset</b>	0x0000 0640		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare RW9 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_RW9																															

Bits	Field Name	Description	Type	Reset
31:0	WKUP_CONTROL_SPARE_RW9	wkup control spare register bits RW9	RW	0xFFFF FFFF

**Table 18-728. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_RW9**

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)

**Table 18-729. CONTROL\_WKUP\_CONTROL\_SPARE\_R1**

<b>Address Offset</b>	0x0000 064C		
<b>Physical Address</b>	Please refer to <a href="#">Table 18-648</a>	<b>Instance</b>	SYSCTRL_PADCONF_WKUP
<b>Description</b>	WKUP control spare R1 Access conditions. Read: unrestricted, Write: unrestricted		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_CONTROL_SPARE_R1																															

Bits	Field Name	Description	Type	Reset
31:0	WKUP_CONTROL_SPARE_R1	wkup control spare register bits R1	R	0x0000 0000



---

**Table 18-730. Register Call Summary for Register CONTROL\_WKUP\_CONTROL\_SPARE\_R1**

---

Control Module Register Manual

- [SYSCTRL\\_PADCONF\\_WKUP Register Summary: \[0\]](#)
- 

PRELIMINARY

PRELIMINARY

---

---

---

## Mailbox

---

---

---

This chapter describes the Mailbox module in the device.

Topic	Page
<b>19.1 Mailbox Overview .....</b>	<b>3876</b>
<b>19.2 Mailbox Integration .....</b>	<b>3878</b>
<b>19.3 Mailbox Functional Description .....</b>	<b>3882</b>
<b>19.4 Mailbox Programming Guide .....</b>	<b>3888</b>
<b>19.5 Mailbox Register Manual .....</b>	<b>3891</b>

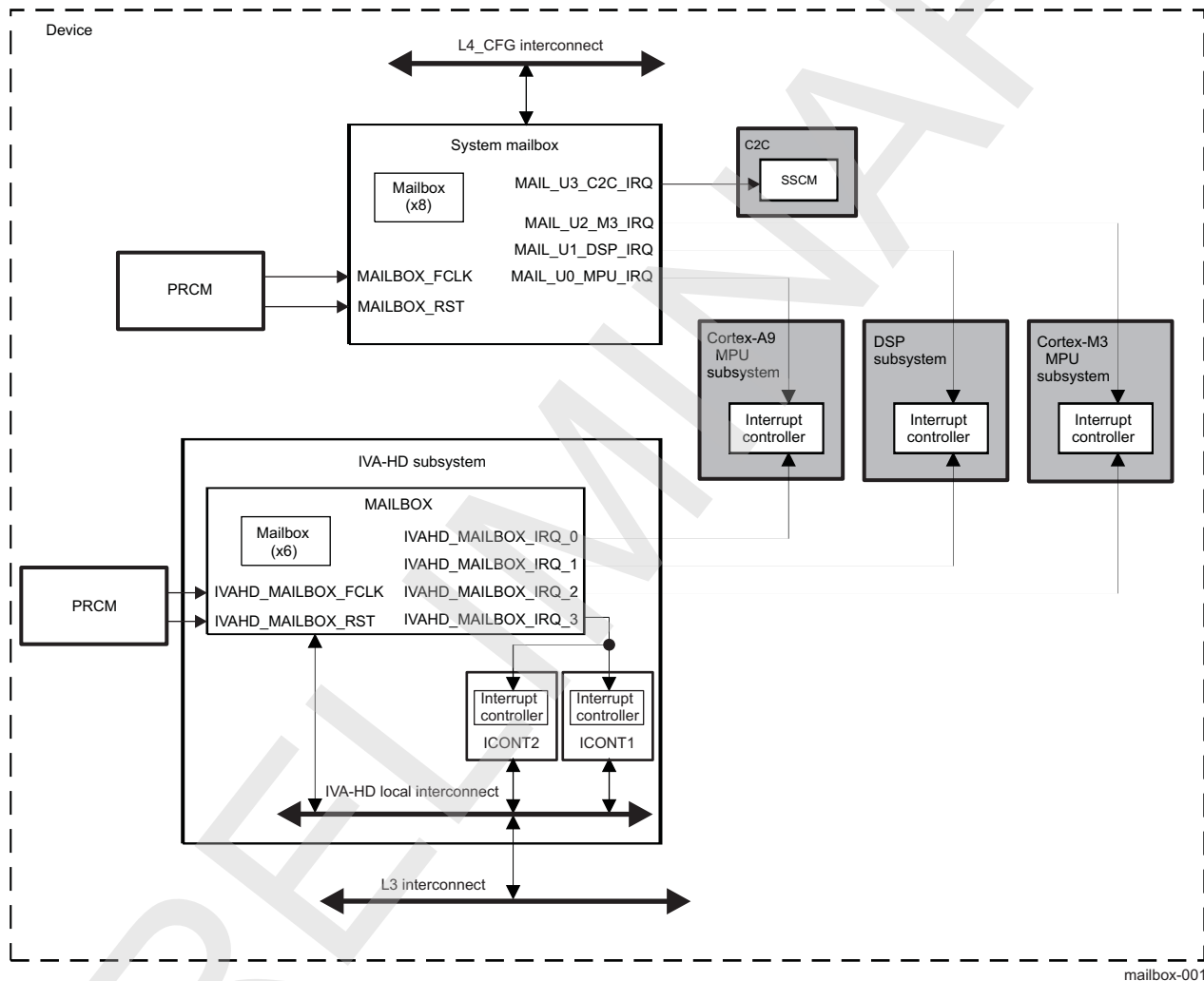
## 19.1 Mailbox Overview

Communication between the on-chip processors of the device uses a queued mailbox-interrupt mechanism.

The queued mailbox-interrupt mechanism allows the software to establish a communication channel between two processors through a set of registers and associated interrupt signals by sending and receiving messages (mailboxes).

Figure 19-1 shows an overview of the mailbox module.

**Figure 19-1. Mailbox Overview**



mailbox-001

There are two mailbox module instances in the device:

- System mailbox - used for Cortex-A9 microprocessor unit (Cortex-A9 MPU), digital signal processor (DSP) and Cortex-M3 microprocessor unit (Cortex-M3 MPU) communications.
- IVA-HD mailbox - used for communication between one internal to the IVA-HD subsystem user (imaging controller 1 - ICONT1, or imaging controller 2 - ICONT2) and three external to the IVA-HD subsystem users (Cortex-A9 MPU, DSP and Cortex-M3 MPU). This communication is insured through three pairs of mailboxes.

The mailbox module includes the following features:

- Three users for the system mailbox instance (Cortex-A9 MPU, DSP and Cortex-M3 MPU)/four users for the IVA-HD mailbox instance (ICONT1/ICONT2, Cortex-A9 MPU, DSP and Cortex-M3 MPU)

- Eight mailbox message queues for the system mailbox instance/six mailbox message queues for the IVA-HD mailbox instance
- Flexible assignment of receiver and sender for each mailbox through interrupt configuration
- Three interrupts (one per user) for the system mailbox instance/four interrupts (one per user) for the IVA-HD mailbox instance
- 32-bit message width
- Four-message FIFO depth for each message queue
- Message reception and queue-not-full notification using interrupts
- Support of 16-/32-bit addressing scheme
- Power management support

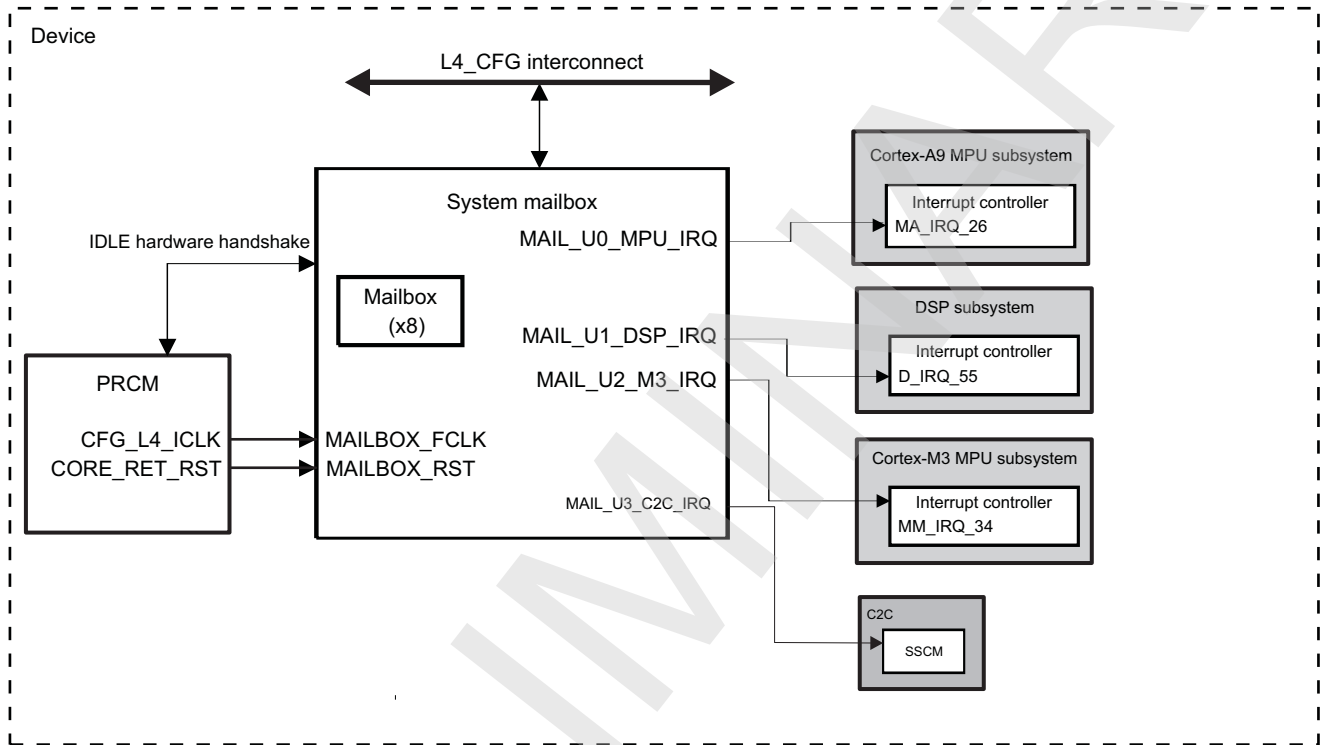
## 19.2 Mailbox Integration

This section describes the mailbox integration in the device, including information about clocks, resets, and hardware requests.

### 19.2.1 System Mailbox Integration

Figure 19-2 shows the system mailbox integration.

**Figure 19-2. System Mailbox Integration**



mailbox-002

**NOTE:** For more information about the IDLE hardware handshake, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

[Table 19-1](#) through [Table 19-3](#) summarize the system mailbox integration in the device.

**Table 19-1. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
SYSTEM_MAILBOX	PD_CORE	NA	L4_CFG

**Table 19-2. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SYSTEM_MAILBOX	MAILBOX_FCLK	CFG_L4_ICLK	PRCM	System mailbox functional/interface clock. For information about PRCM clock gating and management, see <a href="#">Section 3.6.14, CD_L4_CFG Clock Management</a> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SYSTEM_MAILBOX	MAILBOX_RST	CORE_RET_RST	PRCM	System mailbox hardware reset. For information about PRCM reset sources and distribution, see <a href="#">Section 3.5.4, Reset Domains</a> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .

**Table 19-3. Hardware Requests**

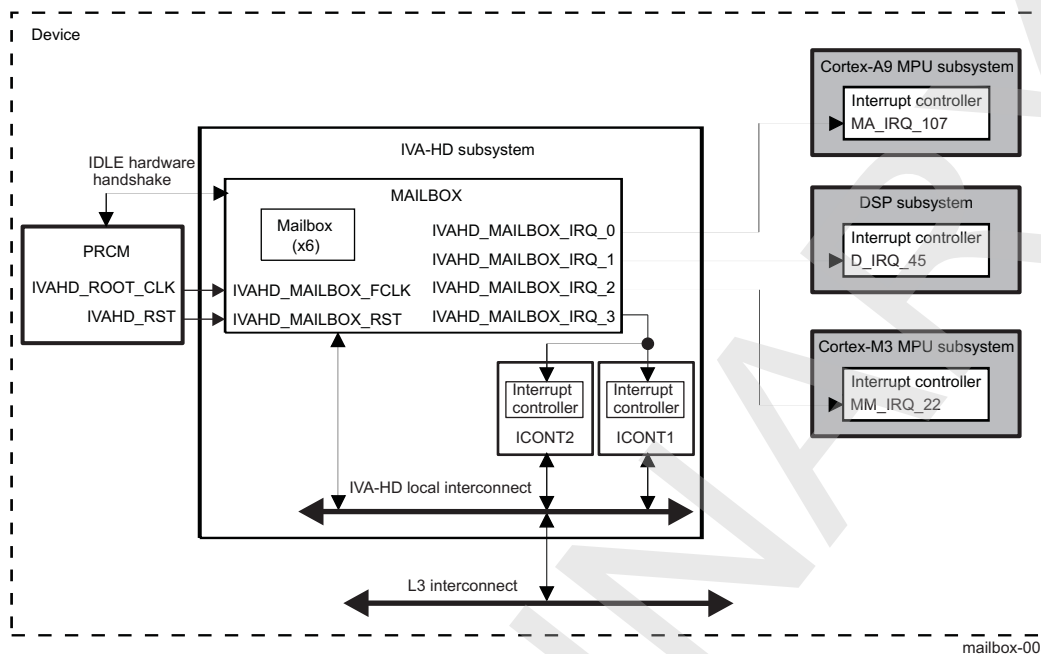
Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
SYSTEM_MAILBOX	MAIL_U0_MPU_IRQ	MA_IRQ_26	Cortex-A9	System mailbox user 0 interrupt request. For information about Cortex-A9 MPU interrupt mapping, see <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-A9 MPU Interrupt Controller</a> , in <a href="#">Chapter 17, Interrupt Controllers</a> .
	MAIL_U1_DSP_IRQ	D_IRQ_55	DSP	System mailbox user 1 interrupt request. For information about DSP interrupt mapping, see <a href="#">Section 5, DSP Subsystem</a> , in <a href="#">Chapter 5, DSP Subsystem</a> .
	MAIL_U2_M3_IRQ	MM_IRQ_34	Cortex-M3	System mailbox user 2 interrupt request. For information about Cortex-M3 MPU interrupt mapping, see <a href="#">Section 17.3.2, Interrupt Requests to the Cortex-M3 MPU Interrupt Controller</a> , in <a href="#">Chapter 17, Interrupt Controllers</a> .
	MAIL_U3_C2C_IRQ	C2C_SSCM_GPI_1	C2C INTC	Interrupt to the SSCM of C2C
No DMA Requests				

**NOTE:** For information about interrupt source description, see [Section 19.3.4, Interrupt Requests](#).

## 19.2.2 IVA-HD Mailbox Integration

[Figure 19-3](#) shows the IVA-HD mailbox integration.

Figure 19-3. IVA-HD Mailbox Integration



**NOTE:** For more information about the IDLE hardware handshake, see [Section 3.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**NOTE:** The two imaging controllers - ICONT1 and ICONT2, are mapped on a shared interrupt line. The choice between them is done by masking mailbox interrupt on ICONT1 or ICONT2. For more information, see [IVA-HD Imaging Controller](#) in [Chapter 6, IVA-HD Subsystem](#).

Table 19-4 through Table 19-6 summarize the IVA-HD mailbox integration in the device.

Table 19-4. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
IVAHD_MAILBOX	PD_IVAHD	NA	IVA-HD local interconnect

Table 19-5. Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
IVAHD_MAILBOX	IVAHD_MAILBOX_FCLK	IVAHD_CLK	PRCM	IVA-HD mailbox functional/interface clock.
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
IVAHD_MAILBOX	IVAHD_MAILBOX_RST	IVAHD_RST	PRCM	IVA-HD mailbox hardware reset.

Table 19-6. Hardware Requests

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
IVAHD_MAILBOX	IVAHD_MAILBOX_IRQ_0	MA_IRQ_107	Cortex-A9	IVA-HD mailbox user 0 interrupt request



**Table 19-6. Hardware Requests (continued)**

IVAHD_MAILBOX_IRQ_1	D_IRQ_45	DSP	IVA-HD mailbox user 1 interrupt request
IVAHD_MAILBOX_IRQ_2	MM_IRQ_22	Cortex-M3	IVA-HD mailbox user 2 interrupt request
IVAHD_MAILBOX_IRQ_3	IRQ#11	ICONT1/ICONT2	IVA-HD mailbox user 3 interrupt request. For information about ICONT1/ICONT2 interrupt mapping, see , <i>ICONT</i> .
<b>No DMA Requests</b>			

**NOTE:** For information about interrupt source description, see [Section 19.3.4, Interrupt Requests](#).

## 19.3 Mailbox Functional Description

**NOTE:** In the mailbox functional description,  $u$  is the user number and  $m$  is the mailbox number as follows:

- for the system mailbox module instance,  $u=0$  to 2 and  $m=0$  to 7
- for the IVA-HD mailbox module instance,  $u=0$  to 3 and  $m=0$  to 5

The mailbox module provides a means of communication through message queues among the users (depending on the mailbox module instance). The individual mailbox modules (8 for the system mailbox instance, 6 for the IVA-HD mailbox instance), or FIFOs, can associate (or de-associate) with any of the processors using the [MAILBOX\\_IRQENABLE\\_SET\\_u](#) (or [MAILBOX\\_IRQENABLE\\_CLR\\_u](#)) register.

### CAUTION

For the IVA-HD mailbox instance, communication is possible only if one of the users is ICONT1 or ICONT2.

The mailbox module includes the following user subsystems:

- User 0: Cortex-A9 MPU subsystem ( $u = 0$ )
- User 1: DSP subsystem ( $u = 1$ )
- User 2: Cortex-M3 MPU subsystem ( $u = 2$ )
- User 3:
  - IVA-HD subsystem ( $u = 3$ ) - only available to the IVA-HD mailbox instance
  - Mailbox subsystem - only available to SSCM module of C2C interface

Each user has a dedicated interrupt signal from the corresponding mailbox module instance and dedicated interrupt enabling and status registers.

Each [MAILBOX\\_IRQSTATUS\\_RAW\\_u](#)/[MAILBOX\\_IRQSTATUS\\_CLR\\_u](#) interrupt status register corresponds to a particular user.

For the system mailbox instance, a user can query its interrupt status register through the L4\_CFG interconnect.

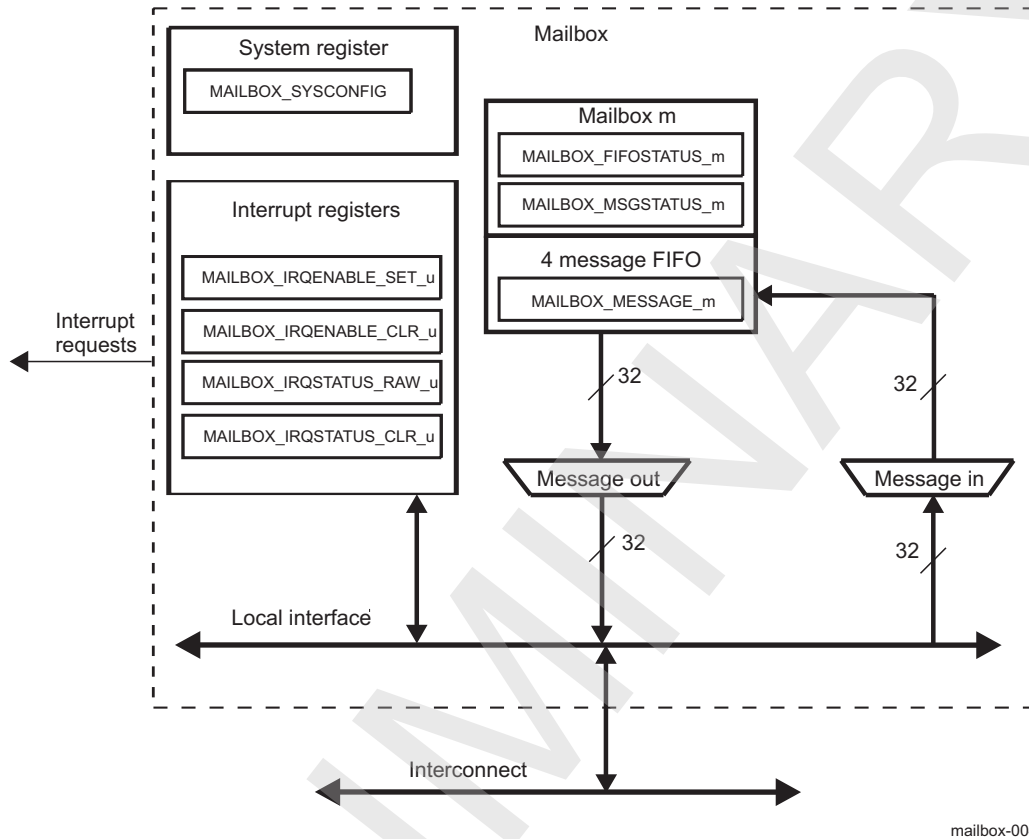
For the IVA-HD mailbox instance, a user can query its interrupt status register as follows:

- Cortex-A9 MPU and Cortex-M3 MPU - through the L3 interconnect
- ICONT1/ICONT2 and DSP - private access (directly through the IVA-HD local interconnect)

### 19.3.1 Mailbox Block Diagram

Figure 19-4 shows the mailbox block diagram.

Figure 19-4. Mailbox Block Diagram



**NOTE:** The interrupt requests and the interconnect depend on the mailbox module instance. For more information, see [Section 19.2, Mailbox Integration](#).

### 19.3.2 Mailbox Software Reset

The mailbox module supports a software reset through the [MAILBOX\\_SYSCONFIG\[0\] SOFTRESET](#) bit. Setting this bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Reading the [MAILBOX\\_SYSCONFIG\[0\] SOFTRESET](#) bit gives the status of the software reset:

- Read 1: the software reset is on-going.
- Read 0: the software reset is complete.

The software must ensure that the software reset completes before doing mailbox operations.

### 19.3.3 Mailbox Power Management

[Table 19-7](#) describes power-management features available for the mailbox module.

**NOTE:**

- For information about source clock gating and sleep/wake-up transitions description, see [Section 3.1.1.1.4, Clock Domain Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).
- For descriptions of EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**Table 19-7. Local Power Management Features**

Feature	Registers	Description
Clock autogating	NA	Feature not available
Slave idle modes	<a href="#">MAILBOX_SYSCONFIG[3:2]</a> SIDLEMODE bit field	Force-idle, no-idle and smart-idle modes are available
Clock activity	NA	Feature not available
Master standby modes	NA	Feature not available
Global wake-up enable	NA	Feature not available
Wake-up sources enable	NA	Feature not available

The mailbox module can be configured using the [MAILBOX\\_SYSCONFIG\[3:2\]](#) SIDLEMODE bit field to one of the following acknowledgment modes:

- Force-idle mode (SIDLEMODE = 0x0): The mailbox module immediately enters the idle state on receiving a low-power-mode request from the PRCM module. In this mode, the software must ensure that there are no asserted output interrupts before requesting this mode to go into the idle state.
- No-idle mode (SIDLEMODE = 0x1): The mailbox module never enters the idle state.
- Smart-idle mode (SIDLEMODE = 0x2): After receiving a low-power-mode request from the PRCM module, the mailbox module enters the idle state only after all asserted output interrupts are acknowledged.

### 19.3.4 Mailbox Interrupt Requests

An interrupt request allows the user of the mailbox to be notified when a message is received or when the message queue is not full. There is one interrupt per user.

**NOTE:** For more information about interrupt requests, see [Section 19.2, Mailbox Integration](#).

[Table 19-8](#) lists the event flags, and their mask, that can cause module interrupts.

**Table 19-8. Interrupt Events**

Non-Maskable Event Flag <sup>(1)</sup>	Maskable Event Flag	Event Mask Bit	Event Unmask Bit	Description
<a href="#">MAILBOX_IRQSTATUS_RAW_u[0+m*2]</a> NEWMSGSTATUSUUM Bm	<a href="#">MAILBOX_IRQSTATUS_CLR_u[0+m*2]</a> NEWMSGSTATUSUUM Bm	<a href="#">MAILBOX_IRQENABLE_CLR_u[0+m*2]</a> NEWMSGSTATUSUUM Bm	<a href="#">MAILBOX_IRQENABLE_SET_u[0+m*2]</a> NEWMSGSTATUSUUM Bm	Mailbox <i>m</i> receives a new message.
<a href="#">MAILBOX_IRQSTATUS_RAW_u[1+m*2]</a> NOTFULLSTATUSUUM Bm	<a href="#">MAILBOX_IRQSTATUS_CLR_u[1+m*2]</a> NOTFULLSTATUSUUM Bm	<a href="#">MAILBOX_IRQENABLE_CLR_u[1+m*2]</a> NOTFULLSTATUSUUM Bm	<a href="#">MAILBOX_IRQENABLE_SET_u[1+m*2]</a> NOTFULLSTATUSUUM Bm	Mailbox <i>m</i> message queue is not full.

<sup>(1)</sup> MAILBOX.MAILBOX\_IRQSTATUS\_RAW\_u register is mostly used for debug purposes.

**CAUTION**

Once an event generating the interrupt request has been processed by the software, it must be cleared by writing a logical 1 in the corresponding bit of the [MAILBOX\\_IRQSTATUS\\_CLR\\_u](#) register.

Writing a logical 1 in a bit of the [MAILBOX\\_IRQSTATUS\\_CLR\\_u](#) register will also clear to 0 the corresponding bit in the appropriate [MAILBOX\\_IRQSTATUS\\_RAW\\_u](#) register.

An event can generate an interrupt request when a logical 1 is written to the corresponding unmask bit in the [MAILBOX\\_IRQENABLE\\_SET\\_u](#) register. Events are reported in the appropriate [MAILBOX\\_IRQSTATUS\\_CLR\\_u](#) and [MAILBOX\\_IRQSTATUS\\_RAW\\_u](#) registers.

An event stops generating interrupt requests when a logical 1 is written to the corresponding mask bit in the [MAILBOX\\_IRQENABLE\\_CLR\\_u](#) register. Events are only reported in the appropriate [MAILBOX\\_IRQSTATUS\\_RAW\\_u](#) register.

In case of the [MAILBOX\\_IRQSTATUS\\_RAW\\_u](#) register, the event is reported in the corresponding bit even if the interrupt request generation is disabled for this event.

### 19.3.5 Mailbox Assignment

#### 19.3.5.1 Description

To assign a receiver to a mailbox, set the new message interrupt enable bit corresponding to the desired mailbox in the [MAILBOX\\_IRQENABLE\\_SET\\_u](#) register. The receiver reads the [MAILBOX\\_MESSAGE\\_m](#) register to retrieve a message from the mailbox.

An alternate method for the receiver that does not use the interrupts is to poll the [MAILBOX\\_FIFOSTATUS\\_m](#) and/or [MAILBOX\\_MSGSTATUS\\_m](#) registers to know when to send or retrieve a message to or from the mailbox. This method does not require assigning a receiver to a mailbox. Because this method does not include the explicit assignment of the mailbox, the software must avoid having multiple receivers use the same mailbox, which can result in incoherency.

To assign a sender to a mailbox, set the queue-not-full interrupt enable bit of the desired mailbox in the [MAILBOX\\_IRQENABLE\\_SET\\_u](#) register, where *u* is the number of the sending user. However, direct allocation of a mailbox to a sender is not recommended because it can cause the sending processor to be constantly interrupted.

It is recommended that register polling be used to:

- Check the status of either the [MAILBOX\\_FIFOSTATUS\\_m](#) or [MAILBOX\\_MSGSTATUS\\_m](#) registers
- Write the message to the corresponding [MAILBOX\\_MESSAGE\\_m](#) register, if space is available.

The sender might use the queue-not-full interrupt when the initial mailbox status check indicates the mailbox is full. In this case, the sender can enable the queue-not-full interrupt for its mailbox in the appropriate [MAILBOX\\_IRQENABLE\\_SET\\_u](#) register. This allows the sender to be notified by interrupt only when a FIFO queue has at least one available entry.

Reading the [MAILBOX\\_IRQSTATUS\\_CLR\\_u](#) register determines the status of the new message and the queue-not-full interrupts for a particular user. Writing 1 to the corresponding bit in the [MAILBOX\\_IRQSTATUS\\_CLR\\_u](#) register acknowledges, and subsequently clears, an interrupt.

**CAUTION**

Assigning multiple senders or multiple receivers to the same mailbox is not recommended.

## 19.3.6 Sending and Receiving Messages

### 19.3.6.1 Description

When a 32-bit message is written to the [MAILBOX\\_MESSAGE\\_m](#) register, the message is appended into the FIFO queue. This queue holds four messages. If the queue is full, the message is discarded.

Queue overflow can be avoided by first reading the [MAILBOX\\_FIFOSTATUS\\_m](#) register to check that the mailbox message queue is not full before writing a new message to it.

Reading the [MAILBOX\\_MESSAGE\\_m](#) register returns the message at the beginning of the FIFO queue and removes it from the queue. If the FIFO queue is empty when the [MAILBOX\\_MESSAGE\\_m](#) register is read, the value 0 is returned.

The new message interrupt is asserted when at least one message is in the mailbox message FIFO queue. To determine the number of messages in the mailbox message FIFO queue, read the [MAILBOX\\_MSGSTATUS\\_m](#) register.

## 19.3.7 16-Bit Register Access

### 19.3.7.1 Description

So that 16-bit processors can access the mailbox module, the module allows 16-bit register read and write access, with restrictions for the [MAILBOX\\_MESSAGE\\_m](#) registers. The 16-bit half-words are organized in little endian fashion; that is, the least-significant 16 bits are at the low address and the most-significant 16 bits are at the high address (low address + 0x02).

All mailbox module registers can be read or written to directly using individual 16-bit accesses with no restriction on interleaving, except the [MAILBOX\\_MESSAGE\\_m](#) registers, which must always be accessed by either single 32-bit accesses or two consecutive 16-bit accesses.

#### CAUTION

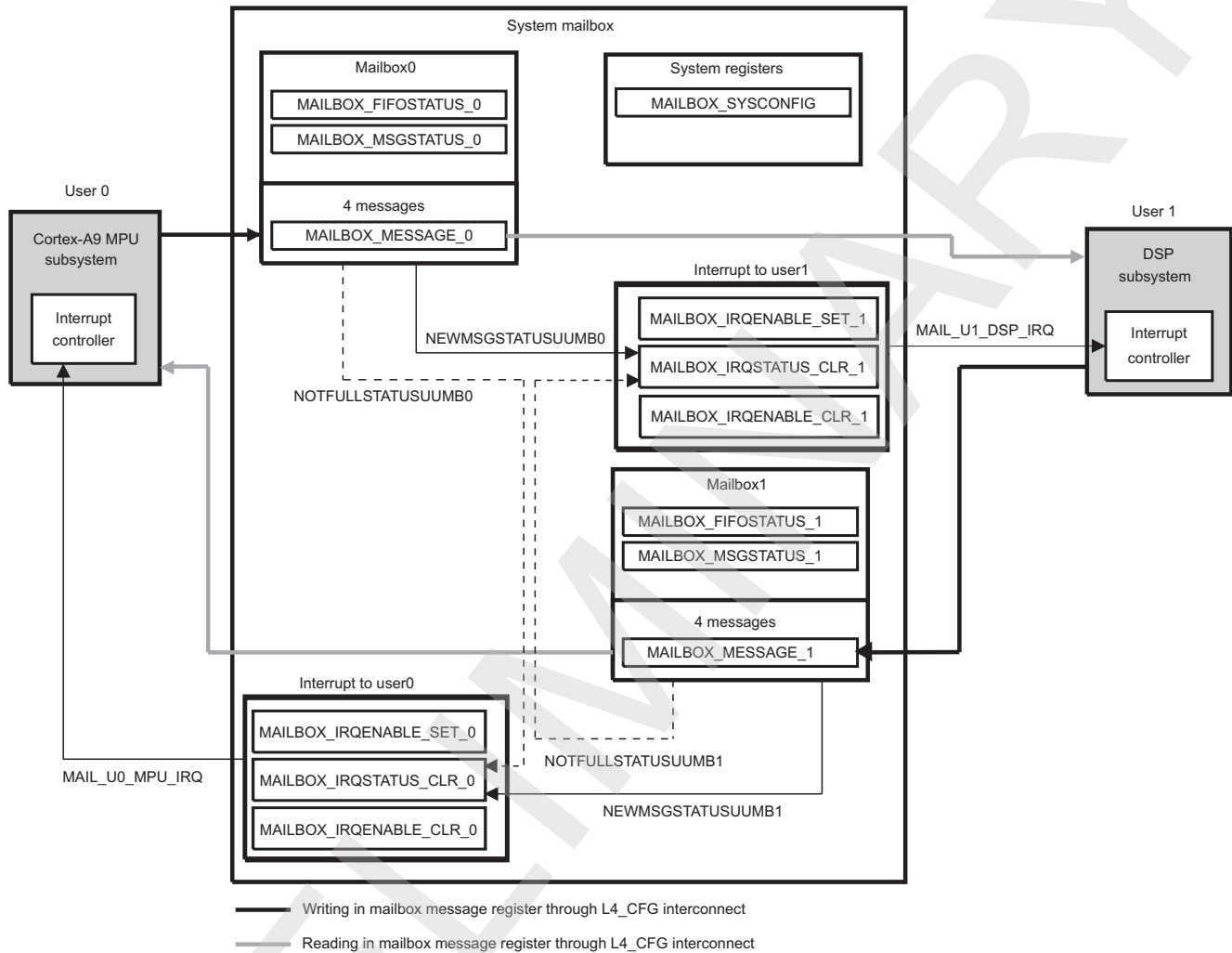
When using 16-bit accesses, it is critical to ensure that the mailbox used has only one assigned receiver and only one assigned sender.

When using 16-bit accesses to the [MAILBOX\\_MESSAGE\\_m](#) registers, the order of access must be the least-significant half-word first (low address) and the most-significant half-word last (high address). This requirement is because of the update operation by the message FIFO of the [MAILBOX\\_MSGSTATUS\\_m](#) registers. The update of the FIFO queue contents and the associated status registers and possible interrupt generation occurs only when the most-significant 16 bits of a [MAILBOX\\_MESSAGE\\_m](#) are accessed.

### 19.3.8 Example of Communication

Figure 19-5 shows an example of communication between Cortex-A9 MPU and DSP subsystems.

Figure 19-5. Example of Communication



mailbox-005

## 19.4 Mailbox Programming Guide

### 19.4.1 Mailbox Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the mailbox module.

#### 19.4.1.1 Global Initialization

##### 19.4.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the mailbox module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the mailbox.

See [Section 19.2, Mailbox Integration](#), for further information.

**Table 19-9. Global Initialization of Surrounding Modules for System Mailbox**

Surrounding Modules	Comments
PRCM	Mailbox functional/interface clock must be enabled. For more information, see <a href="#">Section 3.6.14</a> , <i>CD_L4_CFG Clock Domain</i> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Interrupt Controllers	Cortex-A9 MPU, or Cortex-M3 MPU, or DSP interrupt controller must be configured to enable the interrupt request generation to the Cortex-A9 MPU, or Cortex-M3 MPU, or DSP subsystem. For information about enabling interrupts in Cortex-A9 MPU INTC, see the <i>ARM Cortex™-A9 MPCore Technical Reference Manual</i> . For information about enabling interrupts in Cortex-M3 MPU INTC, see the <i>ARM Cortex™-M3 Technical Reference Manual</i> . For information about enabling interrupts in DSP INTC, see <a href="#">Chapter 5, DSP Subsystem</a> .
Interconnect	For information about L4-CFG interconnect configuration, see <a href="#">Section 13.3, L4 Interconnects</a> , in <a href="#">Chapter 13, Interconnect</a> .

**Table 19-10. Global Initialization of Surrounding Modules for IVA-HD Mailbox**

Surrounding Modules	Comments
PRCM	Mailbox functional/interface clock must be enabled. For more information, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Interrupt Controllers	Cortex-A9 MPU, or Cortex-M3 MPU, or DSP, or ICONT1/ICONT2 interrupt controller must be configured to enable the interrupt request generation to the Cortex-A9 MPU, or Cortex-M3 MPU, or DSP, or IVA-HD subsystem. For information about enabling interrupts in Cortex-A9 MPU INTC, or Cortex-M3 MPU INTC, or DSP INTC, see <a href="#">Table 19-9, Global Initialization of Surrounding Modules for System Mailbox</a> . For information about enabling interrupts in ICONT1/ICONT2, see <i>ICONT</i> in <a href="#">Chapter 6, IVA-HD Subsystem</a> .
Interconnect	For information about L3 interconnect configuration, see <a href="#">Section 13.2, L3 Interconnects</a> , in <a href="#">Chapter 13, Interconnect</a> .

##### 19.4.1.1.2 Mailbox Global Initialization

###### 19.4.1.1.2.1 Main Sequence - Mailbox Global Initialization

This procedure initializes the mailbox module after a power-on or software reset.

**Table 19-11. Mailbox Global Initialization**

Step	Register/ Bit Field / Programming Model	Value
Perform a software reset	<a href="#">MAILBOX_SYSCONFIG</a> [0] SOFTRESET	0x1
Wait until reset is complete	<a href="#">MAILBOX_SYSCONFIG</a> [0] SOFTRESET	= 0x0
Set idle mode configuration	<a href="#">MAILBOX_SYSCONFIG</a> [3:2] SIDLEMODE	0x-



## 19.4.1.2 Mailbox Operational Modes Configuration

### 19.4.1.2.1 Mailbox Processing modes

#### 19.4.1.2.1.1 Main Sequence - Sending a Message (Polling Method)

**Table 19-12. Sending a Message (Polling Method)**

Step	Register/ Bit Field / Programming Model	Value
<b>IF</b> : Is FIFO full ?	MAILBOX_FIFOSTATUS_m[0] FIFOFULLMB	= 0x1
Wait until at least one message slot is available	MAILBOX_FIFOSTATUS_m[0] FIFOFULLMB	= 0x0
<b>ELSE</b>		
Write message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
<b>ENDIF</b>		

#### 19.4.1.2.1.2 Main Sequence - Sending a Message (Interrupt Method)

**Table 19-13. Sending a Message (Interrupt Method)**

Step	Register/ Bit Field / Programming Model	Value
<b>IF</b> : Is FIFO full ?	MAILBOX_FIFOSTATUS_m[0] FIFOFULLMB	= 0x1
Enable interrupt event	MAILBOX_IRQENABLE_SET_u[1+ m*2]	0x1
User (processor) can perform another task until interrupt occurs See <a href="#">Section 19.4.1.3.1</a> for interrupt handling in sending mode		
<b>ELSE</b>		
Write message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
<b>ENDIF</b>		

#### 19.4.1.2.1.3 Main Sequence - Receiving a Message (Polling Method)

**Table 19-14. Receiving a Message (Polling Method)**

Step	Register/ Bit Field / Programming Model	Value
<b>IF</b> : Number of messages is not equal to 0	MAILBOX_MSGSTATUS_m[2:0] NBOFMSGMB	!= 0x0
Read message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
<b>ENDIF</b>		

#### 19.4.1.2.1.4 Main Sequence - Receiving a Message (Interrupt Method)

**Table 19-15. Receiving a Message (Interrupt Method)**

Step	Register/ Bit Field / Programming Model	Value
Enable interrupt event	MAILBOX_IRQENABLE_SET_u[0 + m*2]	0x1
User (processor) can perform another task until interrupt occurs See <a href="#">Section 19.4.1.3.2</a> for interrupt handling in receiving mode		

### 19.4.1.3 Mailbox Events Servicing

#### 19.4.1.3.1 Events Servicing in Sending Mode

Table 19-16 describes the events servicing in sending mode.

**Table 19-16. Events Servicing in Sending Mode**

Step	Register/ Bit Field / Programming Model	Value
Read interrupt status bit	MAILBOX_IRQSTATUS_CLR_u[1 + m*2]	0x1
Write message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
Write 1 to acknowledge interrupt	MAILBOX_IRQSTATUS_CLR_u[1 + m*2]	0x1

#### 19.4.1.3.2 Events Servicing in Receiving Mode

Table 19-17 describes the events servicing in receiving mode.

**Table 19-17. Events Servicing in Receiving Mode**

Step	Register/ Bit Field / Programming Model	Value
Read interrupt status bit	MAILBOX_IRQSTATUS_CLR_u[0 + m*2]	0x1
<b>IF</b> : Number of messages is not equal to 0 ?	MAILBOX_MSGSTATUS_m[2:0] NBOFMSGMB	!= 0x0
Read message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
<b>ELSE</b>		
Write 1 to acknowledge interrupt	MAILBOX_IRQSTATUS_CLR_u[0 + m*2]	0x1
<b>ENDIF</b>		

## 19.5 Mailbox Register Manual

### 19.5.1 Mailbox Instance Summary

**Table 19-18. System Mailbox Instance Summary**

Module Name	Base Address L4_CFG Interconnect	Size
System Mailbox	0x4A0F 4000	4KB

**Table 19-19. IVAHD Mailbox Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address DSP Subsystem Private Access	Base Address Private Access	Size
IVAHD Mailbox	0x5A05 A800	0x01E5 A800	0x008D A800	4KB

**NOTE:** Private access is an access that does not use the L3/L4 interconnects.

### 19.5.2 Mailbox Registers

#### 19.5.2.1 Mailbox Register Summary

**Table 19-20. System Mailbox Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L4_CFG Interconnect
<a href="#">MAILBOX_REVISION</a>	R	32	0x0000 0000	0x4A0F 4000
<a href="#">MAILBOX_SYSCONFIG</a>	RW	32	0x0000 0010	0x4A0F 4010
<a href="#">MAILBOX_MESSAGE_m<sup>(1)</sup></a>	RW	32	0x0000 0040 + (0x4 * m)	0x4A0F 4040 + (0x4 * m)
<a href="#">MAILBOX_FIFOSTATUS_m<sup>(1)</sup></a>	R	32	0x0000 0080 + (0x4 * m)	0x4A0F 4080 + (0x4 * m)
<a href="#">MAILBOX_MSGSTATUS_m<sup>(1)</sup></a>	R	32	0x0000 00C0 + (0x4 * m)	0x4A0F 40C0 + (0x4 * m)
<a href="#">MAILBOX_IRQSTATUS_RAW_u<sup>(2)</sup></a>	RW	32	0x0000 0100 + (0x10 * u)	0x4A0F 4100 + (0x10 * u)
<a href="#">MAILBOX_IRQSTATUS_CLR_u<sup>(2)</sup></a>	RW	32	0x0000 0104 + (0x10 * u)	0x4A0F 4104 + (0x10 * u)
<a href="#">MAILBOX_IRQENABLE_SET_u<sup>(2)</sup></a>	RW	32	0x0000 0108 + (0x10 * u)	0x4A0F 4108 + (0x10 * u)
<a href="#">MAILBOX_IRQENABLE_CLR_u<sup>(2)</sup></a>	RW	32	0x0000 010C + (0x10 * u)	0x4A0F 410C + (0x10 * u)
RESERVED			0x0000 0140	0x4A0F 4140

<sup>(1)</sup> m = 0 to 7

<sup>(2)</sup> u = 0 to 2

**Table 19-21. IVAHD Mailbox Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address DSP Private Access	Physical Address ICONT Private Access
<a href="#">MAILBOX_REVISION</a>	R	32	0x0000 0000	0x5A05 A800	0x01E5 A800	0x008D A800
<a href="#">MAILBOX_SYSCONFIG</a>	RW	32	0x0000 0010	0x5A05 A810	0x01E5 A810	0x008D A810
<a href="#">MAILBOX_MESSAGE_m<sup>(1)</sup></a>	RW	32	0x0000 0040 + (0x4 * m)	0x5A05 A840 + (0x4 * m)	0x01E5 A840 + (0x4 * m)	0x008D A840 + (0x4 * m)
<a href="#">MAILBOX_FIFOSTATUS_m<sup>(1)</sup></a>	R	32	0x0000 0080 + (0x4 * m)	0x5A05 A880 + (0x4 * m)	0x01E5 A880 + (0x4 * m)	0x008D A880 + (0x4 * m)
<a href="#">MAILBOX_MSGSTATUS_m<sup>(1)</sup></a>	R	32	0x0000 00C0 + (0x4 * m)	0x5A05 A8C0 + (0x4 * m)	0x01E5 A8C0 + (0x4 * m)	0x008D A8C0 + (0x4 * m)

<sup>(1)</sup> m = 0 to 5

**Table 19-21. IVAHD Mailbox Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address DSP Private Access	Physical Address ICONT Private Access
<a href="#">MAILBOX_IRQSTATUS_R AW_u<sup>(2)</sup></a>	RW	32	0x0000 0100 + (0x10 * u)	0x5A05 A900 + (0x10 * u)	0x01E5 A900 + (0x10 * u)	0x008D A900 + (0x10 * u)
<a href="#">MAILBOX_IRQSTATUS_C LR_u<sup>(2)</sup></a>	RW	32	0x0000 0104 + (0x10 * u)	0x5A05 A904 + (0x10 * u)	0x01E5 A904 + (0x10 * u)	0x008D A904 + (0x10 * u)
<a href="#">MAILBOX_IRQENABLE_S ET_u<sup>(2)</sup></a>	RW	32	0x0000 0108 + (0x10 * u)	0x5A05 A908 + (0x10 * u)	0x01E5 A908 + (0x10 * u)	0x008D A908 + (0x10 * u)
<a href="#">MAILBOX_IRQENABLE_C LR_u<sup>(2)</sup></a>	RW	32	0x0000 010C + (0x10 * u)	0x5A05 A90C + (0x10 * u)	0x01E5 A90C + (0x10 * u)	0x008D A90C + (0x10 * u)
RESERVED			0x0000 0140	0x5A05 A940	0x01E5 A940	0x008D A940

<sup>(2)</sup> u = 0 to 3**19.5.2.2 Mailbox Register Description****Table 19-22. MAILBOX\_REVISION**

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See <a href="#">Table 19-20</a>
	<b>Instance</b> System Mailbox
	<b>Instance</b> IVA-HD Mailbox
<b>Description</b>	This register contains the IP revision code
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

**Table 19-23. Register Call Summary for Register MAILBOX\_REVISION**

Mailbox Register Manual

- [Mailbox Register Summary: \[0\] \[1\]](#)

**Table 19-24. MAILBOX\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010
<b>Physical Address</b>	See <a href="#">Table 19-20</a>
	<b>Instance</b> System Mailbox
	<b>Instance</b> IVA-HD Mailbox
<b>Description</b>	This register controls the various parameters of the communication interface
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIDLEMODE	RESERVED	SOFTRESET	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	RW	0x0000000
3:2	SIDLEMODE	Idle Mode 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module based on the internal activity of the module 0x3: reserved do not use	RW	0x2
1	RESERVED	Reserved	RW	0
0	SOFTRESET	Softreset Read 0x0: Soft/Hard reset done Write 0x0: No action Read 0x1: Reset is ongoing Write 0x1: Start the soft reset sequence	RW	0

**Table 19-25. Register Call Summary for Register MAILBOX\_SYSCONFIG**

Mailbox Functional Description

- [Mailbox Software Reset: \[0\] \[1\]](#)
- [Mailbox Power Management: \[2\] \[3\]](#)

Mailbox Programming Guide

- [Main Sequence - Mailbox Global Initialization: \[4\] \[5\] \[6\]](#)

Mailbox Register Manual

- [Mailbox Register Summary: \[7\] \[8\]](#)

**Table 19-26. MAILBOX\_MESSAGE\_m**

<b>Address Offset</b>	0x0000 0040 + (0x4 * m)	<b>index:</b>	m = 0 to 7 (System Mailbox) or m = 0 to 5 (IVAHD Mailbox)
<b>Physical Address</b>	See <a href="#">Table 19-20</a>	<b>Instance</b>	System Mailbox
	See <a href="#">Table 19-21</a>	<b>Instance</b>	IVA-HD Mailbox
<b>Description</b>	The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MESSAGEVALUEMBM																															

Bits	Field Name	Description	Type	Reset
31:0	MESSAGEVALUEMBM	Message in Mailbox	RW	0x0000 0000

**Table 19-27. Register Call Summary for Register MAILBOX\_MESSAGE\_m**

Mailbox Functional Description

- [Description: \[0\] \[1\]](#)
- [Description: \[2\] \[3\] \[4\]](#)
- [Description: \[5\] \[6\] \[7\] \[8\]](#)

Mailbox Programming Guide

- [Main Sequence - Sending a Message \(Polling Method\): \[9\]](#)
- [Main Sequence - Sending a Message \(Interrupt Method\): \[10\]](#)
- [Main Sequence - Receiving a Message \(Polling Method\): \[11\]](#)
- [Events Servicing in Sending Mode: \[12\]](#)
- [Events Servicing in Receiving Mode: \[13\]](#)

**Table 19-27. Register Call Summary for Register MAILBOX\_MESSAGE\_m (continued)**

Mailbox Register Manual

- [Mailbox Register Summary: \[14\] \[15\]](#)

**Table 19-28. MAILBOX\_FIFOSTATUS\_m**

<b>Address Offset</b>	0x0000 0080 + (0x4 * m)	<b>index:</b>	m = 0 to 7 (System Mailbox) or m = 0 to 5 (IVAHD Mailbox)
<b>Physical Address</b>	See <a href="#">Table 19-20</a>	<b>Instance</b>	System Mailbox
	See <a href="#">Table 19-21</a>	<b>Instance</b>	IVA-HD Mailbox
<b>Description</b>	The FIFO status register has the status related to the mailbox internal FIFO		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FIFOFULLMBM

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	R	0x0000 0000
0	FIFOFULLMBM	Full flag for Mailbox Read 0x0: Mailbox FIFO is not full Read 0x1: Mailbox FIFO is full	R	0

**Table 19-29. Register Call Summary for Register MAILBOX\_FIFOSTATUS\_m**

Mailbox Functional Description

- [Description: \[0\] \[1\]](#)
- [Description: \[2\]](#)

Mailbox Programming Guide

- [Main Sequence - Sending a Message \(Polling Method\): \[3\] \[4\]](#)
- [Main Sequence - Sending a Message \(Interrupt Method\): \[5\]](#)

Mailbox Register Manual

- [Mailbox Register Summary: \[6\] \[7\]](#)

**Table 19-30. MAILBOX\_MSGSTATUS\_m**

<b>Address Offset</b>	0x0000 00C0 + (0x4 * m)	<b>index:</b>	m = 0 to 7 (System Mailbox) or m = 0 to 5 (IVAHD Mailbox)
<b>Physical Address</b>	See <a href="#">Table 19-20</a>	<b>Instance</b>	System Mailbox
	See <a href="#">Table 19-21</a>	<b>Instance</b>	IVA-HD Mailbox
<b>Description</b>	The message status register has the status of the messages in the mailbox.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															NBOFMSGMBM

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved. Read returns 0	R	0x0000 0000
2:0	NBOFMSGMBM	Number of unread messages in Mailbox Note: Limited to four messages per mailbox.	R	0x00

**Table 19-31. Register Call Summary for Register MAILBOX\_MSGSTATUS\_m**

Mailbox Functional Description

- [Description: \[0\] \[1\]](#)
- [Description: \[2\]](#)
- [Description: \[3\]](#)

Mailbox Programming Guide

- [Main Sequence - Receiving a Message \(Polling Method\): \[4\]](#)
- [Events Servicing in Receiving Mode: \[5\]](#)

Mailbox Register Manual

- [Mailbox Register Summary: \[6\] \[7\]](#)

**Table 19-32. MAILBOX\_IRQSTATUS\_RAW\_u**

<b>Address Offset</b>	0x0000 0100 + (0x10 * u)	<b>index:</b>	u = 0 to 2 (System Mailbox) or u = 0 to 3 (IVA-HD Mailbox)
<b>Physical Address</b>	See <a href="#">Table 19-20</a>	<b>Instance</b>	System Mailbox
	See <a href="#">Table 19-21</a>	<b>Instance</b>	IVA-HD Mailbox
<b>Description</b>	The interrupt status register has the raw status for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit sets this bit. This register is mainly used for debug purpose.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NOTFULLSTATUSUUMB7	NEWMSGSTATUSUUMB7	NOTFULLSTATUSUUMB6	NEWMSGSTATUSUUMB6	NOTFULLSTATUSUUMB5	NEWMSGSTATUSUUMB5	NOTFULLSTATUSUUMB4	NEWMSGSTATUSUUMB4	NOTFULLSTATUSUUMB3	NEWMSGSTATUSUUMB3	NOTFULLSTATUSUUMB2	NEWMSGSTATUSUUMB2	NOTFULLSTATUSUUMB1	NEWMSGSTATUSUUMB1	NOTFULLSTATUSUUMB0	NEWMSGSTATUSUUMB0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0
15	NOTFULLSTATUSUUMB7	NotFull Status bit for User u, Mailbox 7 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
14	NEWMSGSTATUSUUMB7	NewMessage Status bit for User u, Mailbox 7 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0

Bits	Field Name	Description	Type	Reset
13	NOTFULLSTATUSUUMB6	NotFull Status bit for User u, Mailbox 6 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
12	NEWMSGSTATUSUUMB6	NewMessage Status bit for User u, Mailbox 6 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
11	NOTFULLSTATUSUUMB5	NotFull Status bit for User u, Mailbox 5 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
10	NEWMSGSTATUSUUMB5	NewMessage Status bit for User u, Mailbox 5 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
9	NOTFULLSTATUSUUMB4	NotFull Status bit for User u, Mailbox 4 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
8	NEWMSGSTATUSUUMB4	NewMessage Status bit for User u, Mailbox 4 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
7	NOTFULLSTATUSUUMB3	NotFull Status bit for User u, Mailbox 3 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
6	NEWMSGSTATUSUUMB3	NewMessage Status bit for User u, Mailbox 3 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
5	NOTFULLSTATUSUUMB2	NotFull Status bit for User u, Mailbox 2 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1



Bits	Field Name	Description	Type	Reset
4	NEWMSGSTATUSUUMB2	NewMessage Status bit for User u, Mailbox 2 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
3	NOTFULLSTATUSUUMB1	NotFull Status bit for User u, Mailbox 1 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
2	NEWMSGSTATUSUUMB1	NewMessage Status bit for User u, Mailbox 1 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
1	NOTFULLSTATUSUUMB0	NotFull Status bit for User u, Mailbox 0 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
0	NEWMSGSTATUSUUMB0	NewMessage Status bit for User u, Mailbox 0 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0

**Table 19-33. Register Call Summary for Register MAILBOX\_IRQSTATUS\_RAW\_u**

Mailbox Functional Description

- [Mailbox Functional Description: \[0\]](#)
- [Mailbox Interrupt Requests: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

Mailbox Register Manual

- [Mailbox Register Summary: \[7\] \[8\]](#)
- [Mailbox Register Description: \[9\]](#)

**Table 19-34. MAILBOX\_IRQSTATUS\_CLR\_u**

<b>Address Offset</b>	0x0000 0104 + (0x10 * u)	<b>index:</b>	u = 0 to 2 (System Mailbox) or u = 0 to 3 (IVA-HD Mailbox)
<b>Physical Address</b>	See <a href="#">Table 19-20</a>	<b>Instance</b>	System Mailbox
	See <a href="#">Table 19-21</a>	<b>Instance</b>	IVA-HD Mailbox
<b>Description</b>	The interrupt status register has the status combined with irq-enable for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NOTFULLSTATUSUUMB7	NEWMSGSTATUSUUMB7	NOTFULLSTATUSUUMB6	NEWMSGSTATUSUUMB6	NOTFULLSTATUSUUMB5	NEWMSGSTATUSUUMB5	NOTFULLSTATUSUUMB4	NEWMSGSTATUSUUMB4	NOTFULLSTATUSUUMB3	NEWMSGSTATUSUUMB3	NOTFULLSTATUSUUMB2	NEWMSGSTATUSUUMB2	NOTFULLSTATUSUUMB1	NEWMSGSTATUSUUMB1	NOTFULLSTATUSUUMB0	NEWMSGSTATUSUUMB0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0
15	NOTFULLSTATUSUUMB7	NotFull Status bit for User u, Mailbox 7 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
14	NEWMSGSTATUSUUMB7	NewMessage Status bit for User u, Mailbox 7 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
13	NOTFULLSTATUSUUMB6	NotFull Status bit for User u, Mailbox 6 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
12	NEWMSGSTATUSUUMB6	NewMessage Status bit for User u, Mailbox 6 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
11	NOTFULLSTATUSUUMB5	NotFull Status bit for User u, Mailbox 5 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
10	NEWMSGSTATUSUUMB5	NewMessage Status bit for User u, Mailbox 5 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
9	NOTFULLSTATUSUUMB4	NotFull Status bit for User u, Mailbox 4 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
8	NEWMSGSTATUSENUUMB4	NewMessage Status bit for User u, Mailbox 4 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
7	NOTFULLSTATUSENUUMB3	NotFull Status bit for User u, Mailbox 3 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
6	NEWMSGSTATUSENUUMB3	NewMessage Status bit for User u, Mailbox 3 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
5	NOTFULLSTATUSENUUMB2	NotFull Status bit for User u, Mailbox 2 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
4	NEWMSGSTATUSENUUMB2	NewMessage Status bit for User u, Mailbox 2 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
3	NOTFULLSTATUSENUUMB1	NotFull Status bit for User u, Mailbox 1 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
2	NEWMSGSTATUSENUUMB1	NewMessage Status bit for User u, Mailbox 1 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
1	NOTFULLSTATUSENUUMB0	NotFull Status bit for User u, Mailbox 0 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
0	NEWMSGSTATUSENUUMB0	NewMessage Status bit for User u, Mailbox 0 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0

**Table 19-35. Register Call Summary for Register MAILBOX\_IRQSTATUS\_CLR\_u**

## Mailbox Functional Description

- [Mailbox Functional Description: \[0\]](#)
- [Mailbox Interrupt Requests: \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Description: \[6\] \[7\]](#)

## Mailbox Programming Guide

- [Events Servicing in Sending Mode: \[8\] \[9\]](#)
- [Events Servicing in Receiving Mode: \[10\] \[11\]](#)

## Mailbox Register Manual

- [Mailbox Register Summary: \[12\] \[13\]](#)
- [Mailbox Register Description: \[14\]](#)

**Table 19-36. MAILBOX\_IRQENABLE\_SET\_u**

<b>Address Offset</b>	0x0000 0108 + (0x10 * u)	<b>index:</b>	u = 0 to 2 (System Mailbox) or u = 0 to 3 (IVA-HD Mailbox)
<b>Physical Address</b>	See <a href="#">Table 19-20</a>	<b>Instance</b>	System Mailbox
	See <a href="#">Table 19-21</a>	<b>Instance</b>	IVA-HD Mailbox
<b>Description</b>	The interrupt enable register enables to unmask the module internal source of interrupt to the corresponding user. This register is write 1 to set.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	
RESERVED																NOTFULLENABLEUUMB7	NEWMMSGENABLEUUMB7	NOTFULLENABLEUUMB6	NEWMMSGENABLEUUMB6	NOTFULLENABLEUUMB5	NEWMMSGENABLEUUMB5	NOTFULLENABLEUUMB4	NEWMMSGENABLEUUMB4	NOTFULLENABLEUUMB3	NEWMMSGENABLEUUMB3	NOTFULLENABLEUUMB2	NEWMMSGENABLEUUMB2	NOTFULLENABLEUUMB1	NEWMMSGENABLEUUMB1	NOTFULLENABLEUUMB0	NEWMMSGENABLEUUMB0																	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0
15	NOTFULLENABLEUUMB7	NotFull Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
14	NEWMMSGENABLEUUMB7	NewMessage Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
13	NOTFULLENABLEUUMB6	NotFull Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
12	NEWMSGENABLEUUMB6	NewMessage Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
11	NOTFULLENABLEUUMB5	NotFull Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
10	NEWMSGENABLEUUMB5	NewMessage Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
9	NOTFULLENABLEUUMB4	NotFull Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
8	NEWMSGENABLEUUMB4	NewMessage Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
7	NOTFULLENABLEUUMB3	NotFull Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
6	NEWMSGENABLEUUMB3	NewMessage Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
5	NOTFULLENABLEUUMB2	NotFull Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
4	NEWMSGENABLEUUMB2	NewMessage Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
3	NOTFULLENABLEUUMB1	NotFull Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
2	NEWMSGENABLEUUMB1	NewMessage Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
1	NOTFULLENABLEUUMB0	NotFull Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
0	NEWMSGENABLEUUMB0	NewMessage Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

**Table 19-37. Register Call Summary for Register MAILBOX\_IRQENABLE\_SET\_u**

Mailbox Functional Description

- [Mailbox Functional Description: \[0\]](#)
- [Mailbox Interrupt Requests: \[1\] \[2\] \[3\]](#)
- [Description: \[4\] \[5\] \[6\]](#)

Mailbox Programming Guide

- [Main Sequence - Sending a Message \(Interrupt Method\): \[7\]](#)
- [Main Sequence - Receiving a Message \(Interrupt Method\): \[8\]](#)

Mailbox Register Manual

- [Mailbox Register Summary: \[9\] \[10\]](#)
- [Mailbox Register Description: \[11\]](#)

**Table 19-38. MAILBOX\_IRQENABLE\_CLR\_u**

<b>Address Offset</b>	0x0000 010C + (0x10 * u)	<b>index:</b>	u = 0 to 2 (System Mailbox) or u = 0 to 3 (IVA-HD Mailbox)
<b>Physical Address</b>	See <a href="#">Table 19-20</a>	<b>Instance</b>	System Mailbox
	See <a href="#">Table 19-21</a>	<b>Instance</b>	IVA-HD Mailbox
<b>Description</b>	The interrupt enable register enables to mask the module internal source of interrupt to the corresponding user. This register is write 1 to clear.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NOTFULLENABLEUUMB7	NEWMSGENABLEUUMB7	NOTFULLENABLEUUMB6	NEWMSGENABLEUUMB6	NOTFULLENABLEUUMB5	NEWMSGENABLEUUMB5	NOTFULLENABLEUUMB4	NEWMSGENABLEUUMB4	NOTFULLENABLEUUMB3	NEWMSGENABLEUUMB3	NOTFULLENABLEUUMB2	NEWMSGENABLEUUMB2	NOTFULLENABLEUUMB1	NEWMSGENABLEUUMB1	NOTFULLENABLEUUMB0	NEWMSGENABLEUUMB0

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Read returns 0.	RW	0
15	NOTFULLENABLEUUMB7	NotFull Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
14	NEWMSGENABLEUUMB7	NewMessage Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
13	NOTFULLENABLEUUMB6	NotFull Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
12	NEWMSGENABLEUUMB6	NewMessage Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
11	NOTFULLENABLEUUMB5	NotFull Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
10	NEWMSGENABLEUUMB5	NewMessage Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
9	NOTFULLENABLEUUMB4	NotFull Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
8	NEWMSGENABLEUUMB4	NewMessage Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
7	NOTFULLENABLEUUMB3	NotFull Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
6	NEWMSGENABLEUUMB3	NewMessage Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
5	NOTFULLENABLEUUMB2	NotFull Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
4	NEWMSGENABLEUUMB2	NewMessage Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
3	NOTFULLENABLEUUMB1	NotFull Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
2	NEWMSGENABLEUUMB1	NewMessage Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
1	NOTFULLENABLEUUMB0	NotFull Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
0	NEWMSGENABLEUUMB0	NewMessage Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0



---

**Table 19-39. Register Call Summary for Register MAILBOX\_IRQENABLE\_CLR\_u**


---

## Mailbox Functional Description

- [Mailbox Functional Description](#): [0]
  - [Mailbox Interrupt Requests](#): [1] [2] [3]
- 

## Mailbox Register Manual

- [Mailbox Register Summary](#): [4] [5]
  - [Mailbox Register Description](#): [6]
- 

---

**NOTE:** For each interrupt status and enable register ([MAILBOX\\_IRQSTATUS\\_RAW\\_u](#), [MAILBOX\\_IRQSTATUS\\_CLR\\_u](#), [MAILBOX\\_IRQENABLE\\_SET\\_u](#) and [MAILBOX\\_IRQENABLE\\_CLR\\_u](#)), bits [15:12] have the given meaning only for the System Mailbox instance. For the IVA-HD Mailbox instance, these bits are considered as Reserved.

---

PRELIMINARY

## Memory Management Units

This chapter describes the memory management units (MMUs).

Topic	Page
<b>20.1 MMU Overview .....</b>	<b>3908</b>
<b>20.2 MMU Integration .....</b>	<b>3909</b>
<b>20.3 MMU Functional Description .....</b>	<b>3912</b>
<b>20.4 MMU Low-level Programming Models .....</b>	<b>3925</b>
<b>20.5 MMU Register Manual .....</b>	<b>3928</b>

## 20.1 MMU Overview

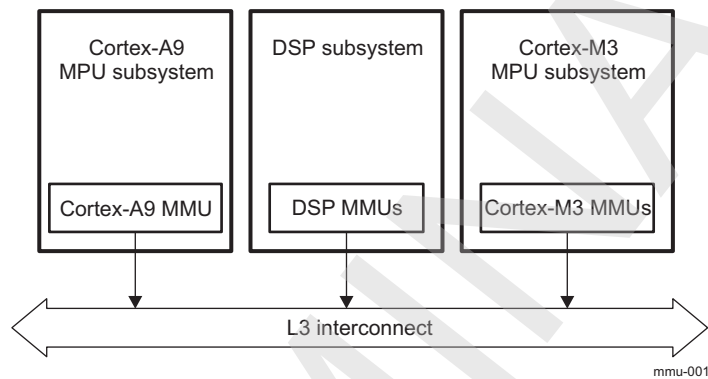
A memory management unit (MMU) is a hardware component responsible for handling accesses to memory requested by a processing unit. MMU functions include translation of virtual addresses to physical addresses (that is, virtual memory management) and cache control.

The device contains the following MMUs:

- In DSP - L1 Shared cache MMU and attribute MMU.
- In Dual Cortex-M3 MPU Subsystem - L1 Shared cache MMU and L2 MMU
- Cortex-A9 MMU

Figure 20-1 shows the MMU instances in the device.

**Figure 20-1. Device MMU Instances**

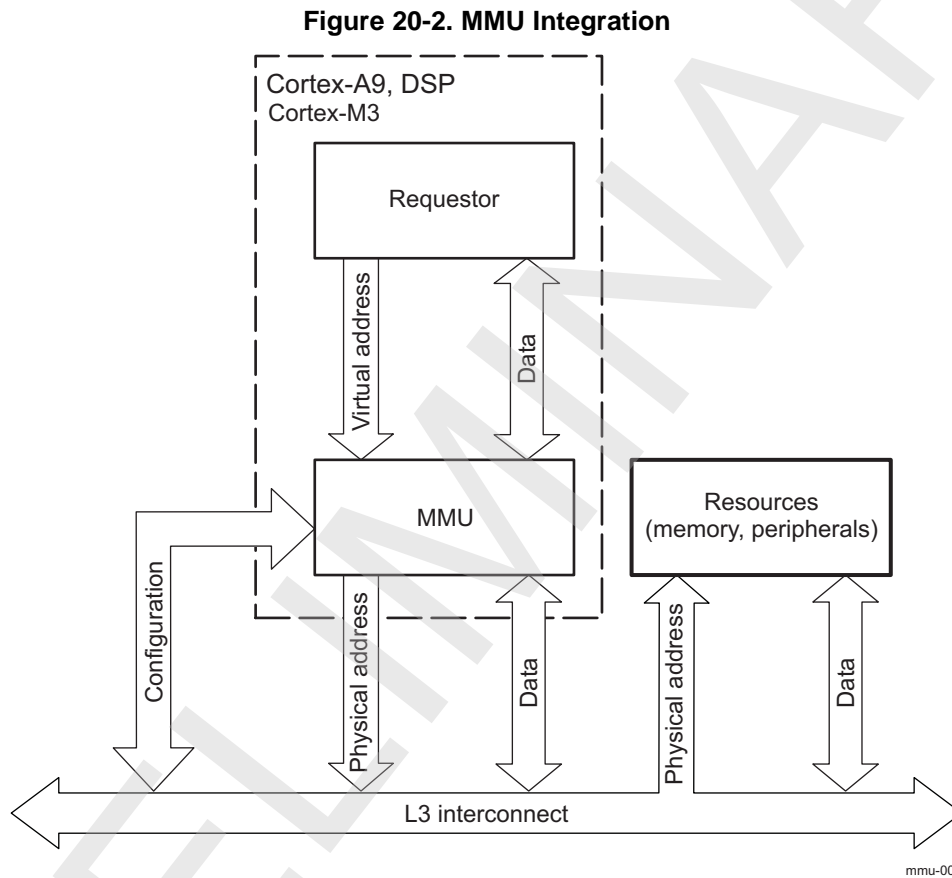


## 20.2 MMU Integration

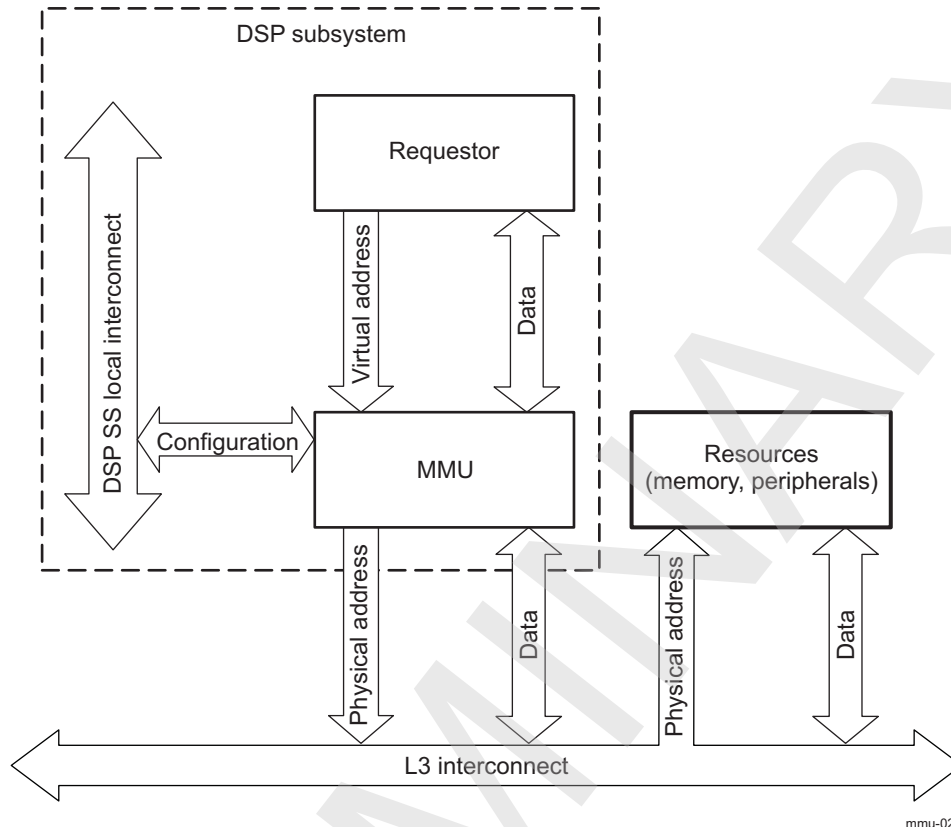
This section describes module integration in the device, including information about clocks, resets, and hardware requests.

The MMU communicates accesses from the requestor (DSP, Cortex-M3 microprocessor unit (MPU), or Cortex-A9 MPU) to the L3 main interconnect, performing virtual to physical address translation. Although all MMUs are programmed (configured) through the L3 interconnect, the DSP MMU can be programmed both through the L3 interconnect or through the DSP local interconnect. DSP MMU and Cortex-M3 MMU error conditions are signaled as interrupts to the system main processor Cortex-A9 MPU.

Figure 20-2 shows typical MMU integration.



**NOTE:** All MMUs are programmed through the L3 interconnect.

**Figure 20-3. DSP MMU Integration**

**NOTE:** The DSP MMU can be programmed from both the DSP local interconnect and from the L3 interconnect.

Table 20-1 through Table 20-3 summarize the MMU integration in the various modules of the device.

**Table 20-1. Integration Attributes**

Module Instance	Power Domain
CORTEXM3_L2MMU and CORTEXM3_SCACHE_MMU	PD_CORE
CORTEXA9_MMU	PD_MPU

**Table 20-2. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
CORTEXA9_MMU	ARM_FCLK	MPU_DPLL_CLK	MPU DPLL	Internal clock in the Cortex-A9 MPU subsystem; supplies the MMU in Cortex-A9.
CORTEXM3_L2MMU and CORTEXM3_SCACHE_MMU	MPU_M3_CLK	MPU_M3_CLK	PRCM	External clock from the power, reset, and clock management (PRCM) module; supplies the shared cache MMU and L2 MMU in Cortex-M3.
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
CORTEXM3_L2MMU	CORTEXM3_PWRON_RST	CORTEXM3_PWRON_RST	PRCM	Power-on reset signal for the Cortex-M3 MPU subsystem
	CORTEXM3_RST3	CORTEXM3_RST3	PRCM	Reset signal to the L2 MMU

**Table 20-2. Clocks and Resets (continued)**

CORTEXM3_SCACHE_MMU	CORTEXM3_RST3	CORTEXM3_RST3	PRCM	Reset signal for the shared cache MMU
	CORTEXM3_RET_RST	CORTEXM3_RET_RST	PRCM	Retention reset signal for the Cortex-M3 MPU subsystem
CORTEXA9_MMU	MPU_PWRON_RST	MPU_PWRON_RST	PRCM	Power-on reset signal to the MMU and the rest of the Cortex-A9 MPU subsystem
	MPU_RST	MPU_RST	PRCM	Reset signal to the MMU and the rest of the Cortex-A9 MPU subsystem

**Table 20-3. Hardware Requests**

Module Instance	Source Signal Name	Interrupt Requests		Description
		Destination Signal Name	Destination	
CORTEXM3_L2MMU	CORTEXM3_MMU_IRQ	MA_IRQ_100	MPU_INT C	Cortex-M3 MPU L2MMU interrupt to the main MPU subsystem
	XLATE_MMU_FAULT	MM_IRQ_0	CortexM3_INT C	Internal interrupt from the L2MMU to the Cortex-M3 interrupt controller (INTC)
CORTEXM3_SCACHE_MMU	SHARED_CACHE_MMU_C PU_INT	MM_IRQ_1	CortexM3_INT C	Internal interrupt from the shared cache MMU to the CortexM3 INTC

**NOTE:**

- For a description of the interrupt source, see [Section 20.3.5, Interrupt Requests](#).

There are 5 MMU modules in the device:

- Shared cache MMU – there are two Share cache MMUs, one in the Cortex-M3 MPU and other in DSP subsystem. Refer to the corresponding chapters for more information.
- L2 MMUs – there are two L2 MMUs, one in the Cortex-M3 MPU and other in DSP subsystem. The L2 MMUs are directly operating with the L3 interconnect (shared device memory space). These L2 MMUs are described in this chapter.
- Cortex-A9 MMU – this MMU is located in the ARM Cortex-A9 processor. For information about this MMU, see the ARM® Cortex™-A9 MPCore Technical Reference Manual.

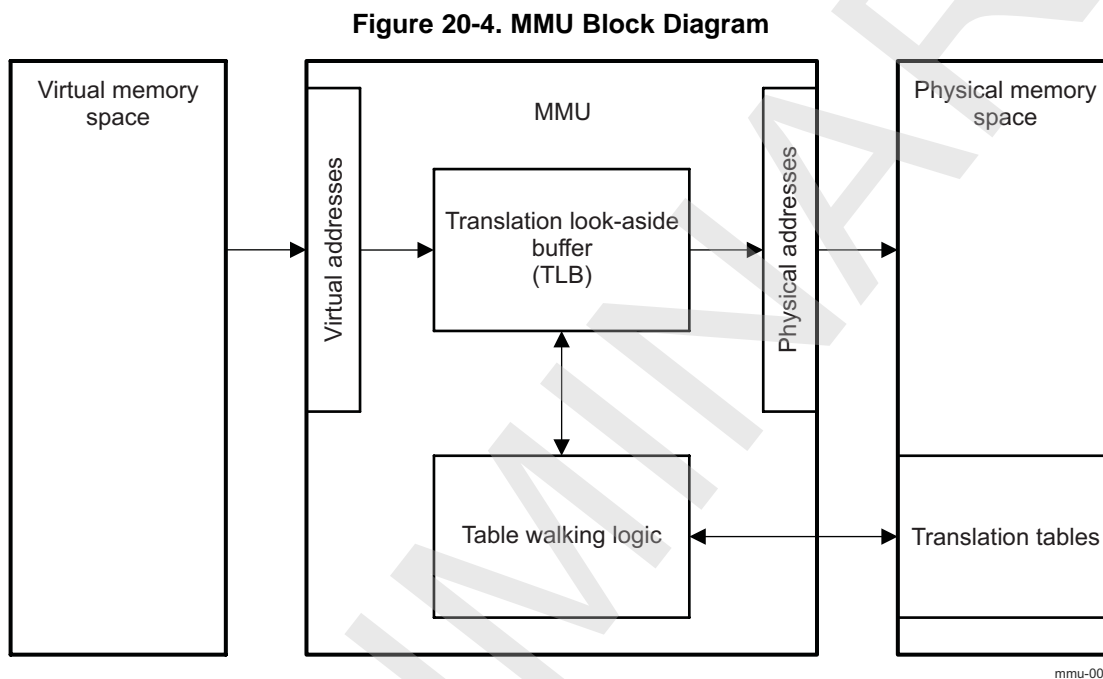
**NOTE:** There is PAT module in the Dynamic Memory Manager, which has similar to the MMU functionality. Refer to the DMM chapter, for more information.

## 20.3 MMU Functional Description

### 20.3.1 MMU Block Diagram

The MMU manages the virtual to physical address translation for external addresses, as well as endianness conversion. The MMU can be programmed through the L3 interconnect. MMU programming is expected to be accomplished by the Cortex-A9 MPU, except for the DSP MMU, of which the configuration can be accomplished through the DSP local interconnect.

Figure 20-4 is the MMU block diagram.



Each table entry describes the translation of one contiguous memory region. For a description of the structure of these tables, see [Section 20.3.1.2, Translation Tables](#).

Two major functional units exist in the MMU to provide address translation automatically based on the table entries:

- The table walker automatically retrieves the correct translation table entry for a requested translation. If two-level translation is used (for the translation of small memory pages), the table walker also automatically reads the required second-level translation table entry. The two-level translation is described later in the chapter.
- The translation look-aside buffer (TLB) stores recently used translation entries, acting like a cache of the translation table.

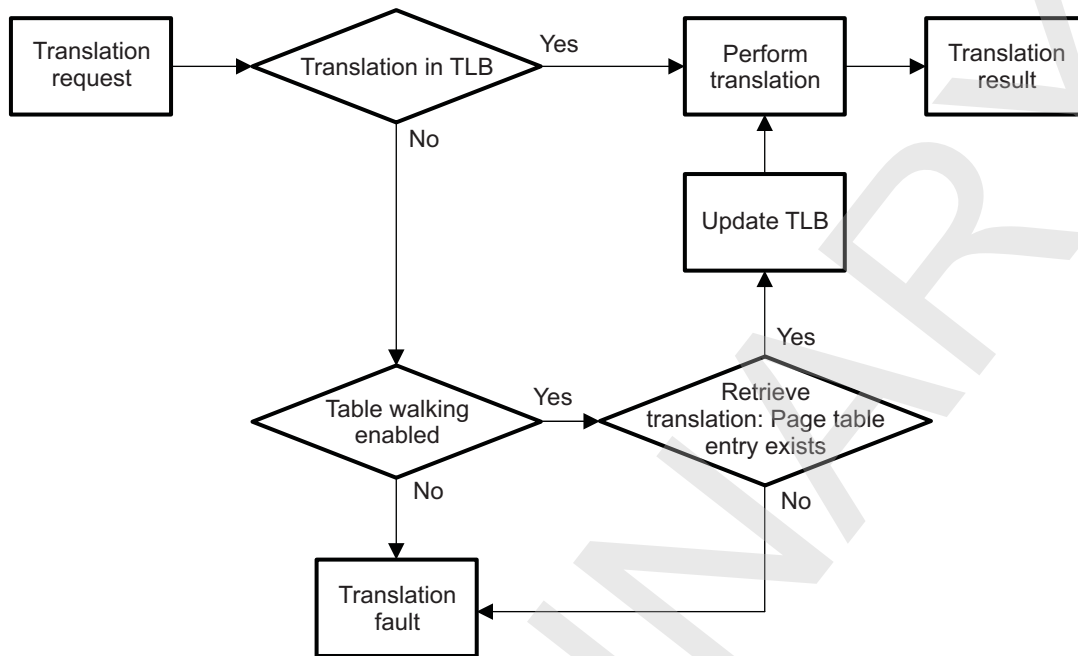
#### 20.3.1.1 MMU Address Translation Process

Whenever an address translation is requested (that is, for every access with the MMU enabled), the MMU first checks whether the translation is contained in the TLB, which acts like a cache storing recent translations. The TLB can also be programmed manually to ensure that time-critical data can be translated without delay.

If the requested translation is not in the TLB, the table-walking logic retrieves this translation from the translation table(s), and then updates the TLB. The address translation is then performed. [Figure 20-5](#) summarizes the process.



Figure 20-5. Translation Process



mmu-007

### 20.3.1.2 Translation Tables

The translation of virtual to physical addresses is based on entries in translation tables that define the following properties:

- Address translation, that is, the correspondence between virtual and physical addresses
- Size of the memory region the entry translates
- Endianness, data access size, and the mixed property of this memory region

The virtual addresses index the translation tables. Each virtual address corresponds to exactly one entry in the translation table.

#### 20.3.1.2.1 Translation Table Hierarchy

When developing a table-based address translation scheme, one of the most important design parameters is the memory page size described by each translation table entry. MMU instances support 4KB and 64KB pages, a 1MB section, and a 16MB supersection. Using bigger page sizes means a smaller translation table.

Using a smaller page size greatly increases the efficiency of dynamic memory allocation and defragmentation. That is why many operating systems (OSs) can operate on memory blocks as small as 4KB; however, the smaller size implies a more complex table structure.

A quick calculation shows that using 4KB memory pages with one translation table would require one million entries to span the entire 4GB address range. The table itself would be 32MB, a size that is not feasible.

However, using bigger pages greatly reduces the functionality of the OS memory management. Implementing a two-level hierarchy reconciles these two requirements. Within this hierarchy, one first-level translation table describes the translation properties based on 1MB memory regions.

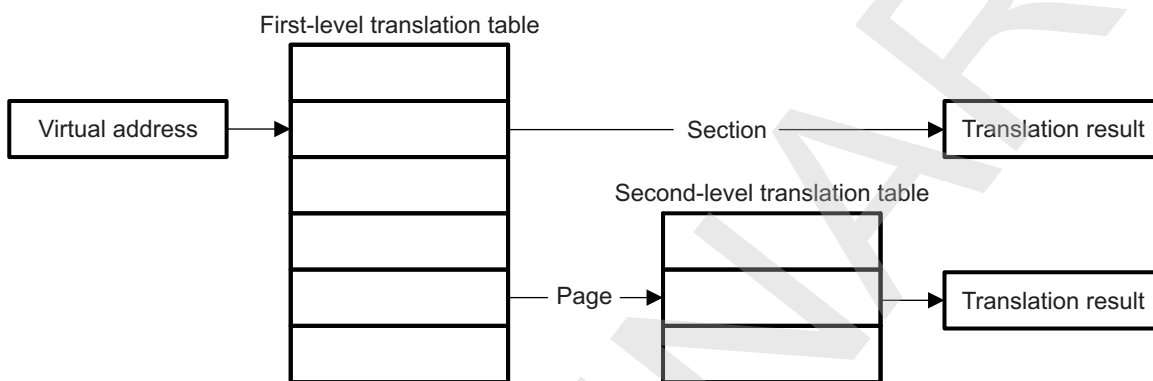
Each of the entries in this first-level translation table can specify the following:

- The translation properties for a big memory section. This memory section can be either 1MB (section) or 16MB (supersection). In this case, all translation parameters are specified in the first-level translation table entry.

- A pointer to a second-level translation table that specifies individual translation properties based on smaller pages within the 1MB page of memory. These pages can be either 64KB (large page) or 4KB (small page). In this case, the actual translation parameters are specified in the second-level translation table entry. The first-level translation table entry specifies only the base address of the second-level translation table.

This hierarchical approach means that additional translation information for smaller pages must be provided only when the pages are actually used. [Figure 20-6](#) shows the hierarchy.

**Figure 20-6. Translation Hierarchy**



mmu-008

The structure of the first and second-level translation tables and their entries are described in more detail in [Section 20.3.1.2.2, First-Level Translation Table](#), and [Section 20.3.1.2.3, Two-Level Translation](#).

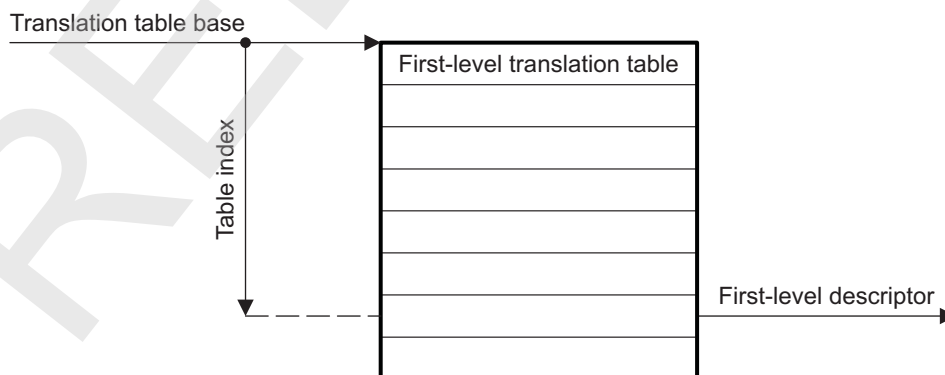
### 20.3.1.2.2 First-Level Translation Table

The first-level translation table describes the translation properties for 1MB sections. To describe a 4GB address range requires 4096 32-bit entries (so-called first-level descriptors).

The first-level translation table start address must be aligned on a multiple of the table size with a 128-byte minimum. Consequently, an alignment of at least 16K bytes is required for a complete 4096-entry table; that is, at least the last fourteen address bits must be zero.

The start address of the first-level translation table is specified by the so-called translation table base. The table is indexed by the upper 12-bits of the virtual address. [Figure 20-7](#) shows this mechanism.

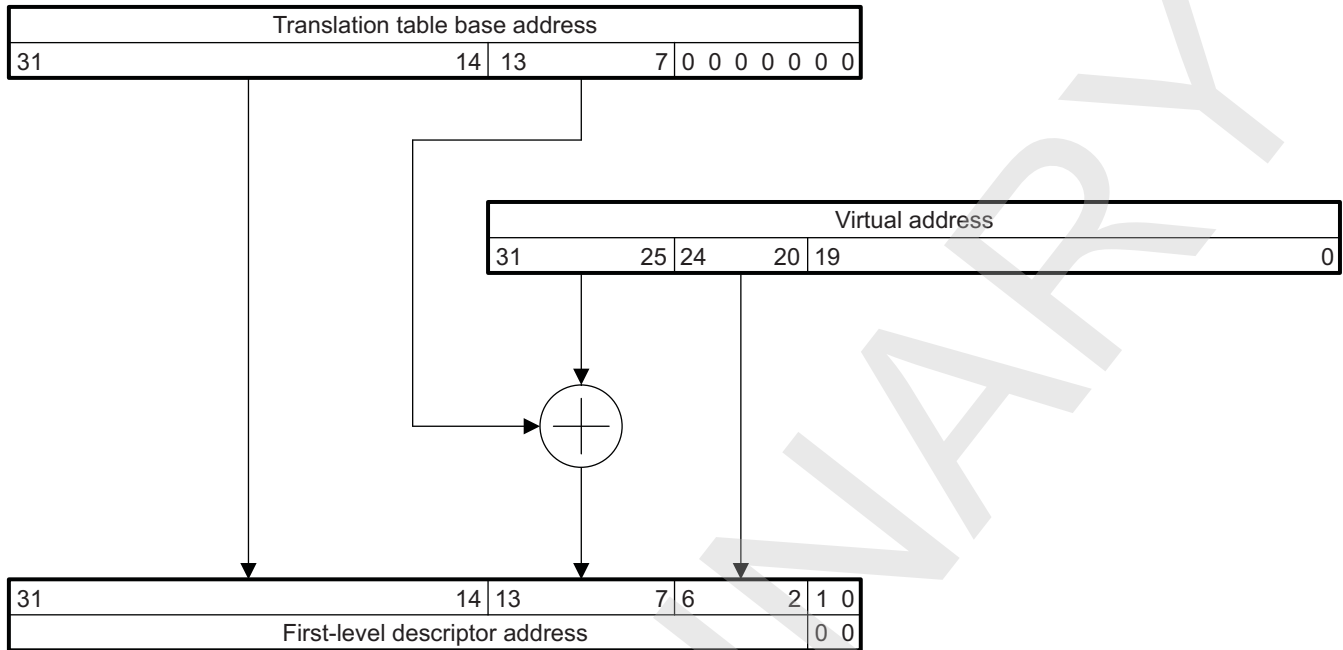
**Figure 20-7. First-level Descriptor Address Calculation**



mmu-009

To summarize, the translation table base and the translation table index together define the first-level descriptor address. [Figure 20-8](#) outlines the precise mechanism used to calculate this address.

Figure 20-8. Detailed First-level Descriptor Address Calculation



mmu-010

As an example of this mechanism, consider a translation table base address of 0x8000:0000 and a virtual address of 0x1234:5678. In this case, the first-level descriptor address is 0x8000:0000 + (0x123 <<2) = 0x8000:048C.

### 20.3.1.2.2.1 First-Level Descriptor Format

Each first-level descriptor provides either the complete address translation for 1MB or 16MB sections or provides a pointer to a second-level translation table for 4KB or 64KB pages. Table 20-4 shows the first-level descriptor format.

Table 20-4. First-Level Descriptor Format

First-Level Descriptor Format												
31:24	23:20	19	18	17	16	15	14:12	11:10	9:2	1	0	
X										0	0	Fault
Second-Level Translation Table Base Address									X	0	1	Page
Section Base Address		X	0	M	X	E <sup>(1)</sup>	X	ES	X	1	0	Section
Supersection Base Address		X	1	M	X	E	X	ES	X	1	0	Supersection
X										1	1	Fault

<sup>(1)</sup> See Table 20-42 for endianness limitations.

M = Mixed region: 0 = Page-based endianness, 1 = Access-based endianness

E = Endianness: 0 = Little endian, 1 = Big endian (endianness is locked on little endian)

ES = Element Size: 00 = 8-bit, 01 = 16-bit, 10 = 32-bit, 11 = No endianness conversion

X = Don't care

### 20.3.1.2.2.2 First-Level Page Descriptor Format

If a translation granularity smaller than 1MB is required, a two-level translation process is used. In this case, the first-level block descriptor specifies only the start address of a second-level translation table. The second-level translation table entries specify the actual translation properties.

### 20.3.1.2.2.3 First-Level Section Descriptor Format

Each section descriptor in the first-level translation table specifies the complete translation properties for a 1MB section or a 16MB supersection.

**NOTE:** Supersection descriptors must be repeated 16 times, because each descriptor in the first-level translation table describes 1MB of memory. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way.

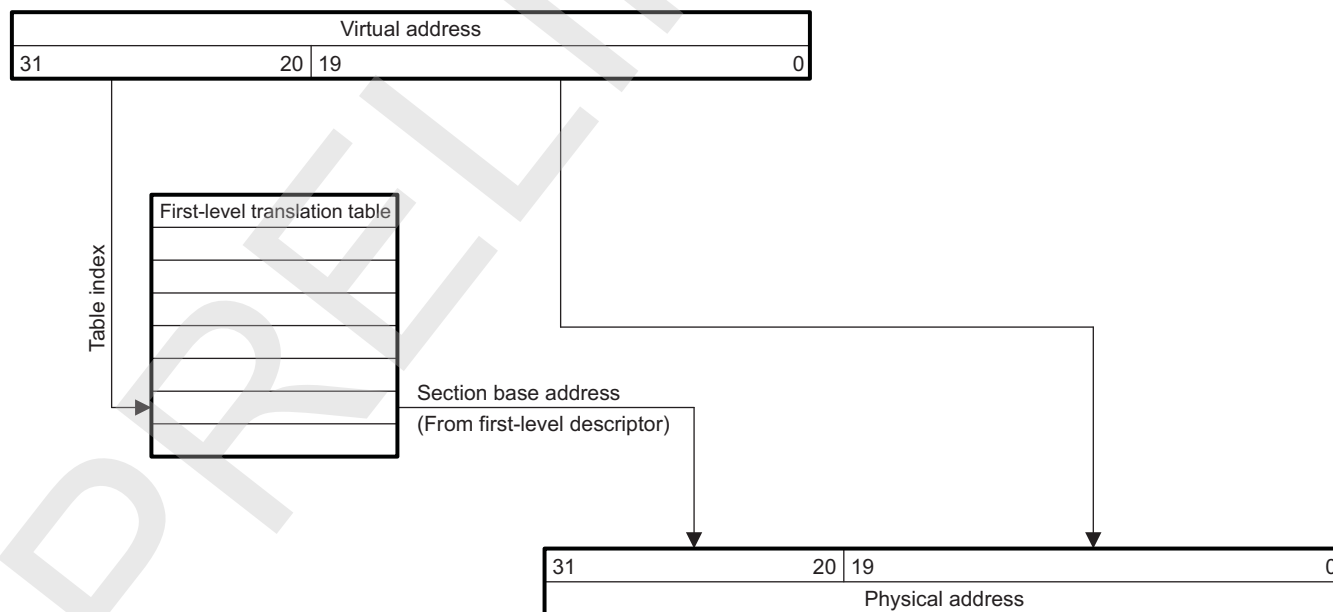
In addition to the address translation itself, three parameters are specified in the section descriptors:

- **Endianness**  
The *endianness* parameter specifies whether the memory section uses a big- or little-endian data format. This parameter is locked to little endian. See [Table 20-42](#) for more information.
- **Element size**  
The *element size* parameter can optionally specify the data access size (8, 16, or 32 bits) for all data items in the defined section.
- **Mixed region**  
The *mixed region* parameter specifies whether the information about the data access size is detected from the access itself (access-based detection) or if the specified element size parameter is used (page-based detection). For example, the specified element size parameter can be used when several smaller sized accesses are packed into a bigger sized access, such as two 16-bit accesses packed into one 32-bit access. In this case, with no specified data access size, 32 bits would be the access size detected, leading to an incorrect result. To avoid this problem, specify the data access size for the memory section.

### 20.3.1.2.2.4 Section Translation Summary

Sections and supersections can be translated based solely on the information in the first-level translation table. [Figure 20-9](#) summarizes the address translation process for a section.

**Figure 20-9. Section Translation Summary**



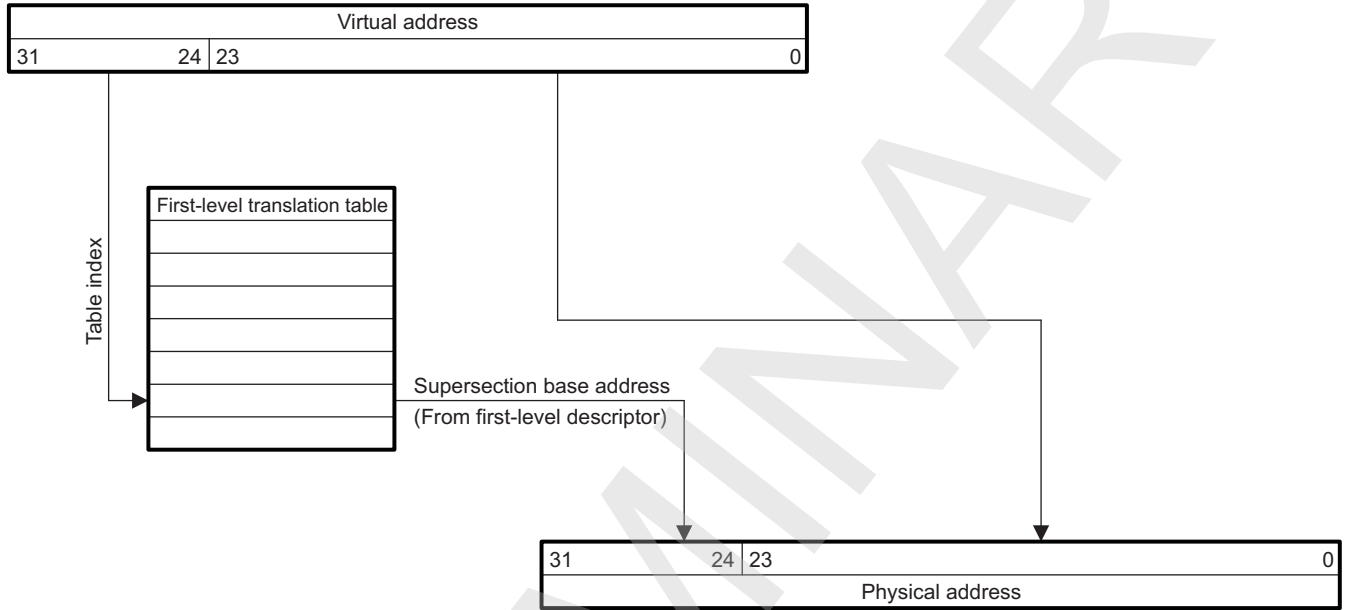
mmu-011

**20.3.1.2.2.5 Supersection Translation Summary**

The translation of a supersection is similar to the translation of a section. The difference is that for a supersection only bits 31 to 24 index into the first-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a supersection.

Figure 20-10 shows the translation mechanism for a supersection.

**Figure 20-10. Supersection Translation Summary**

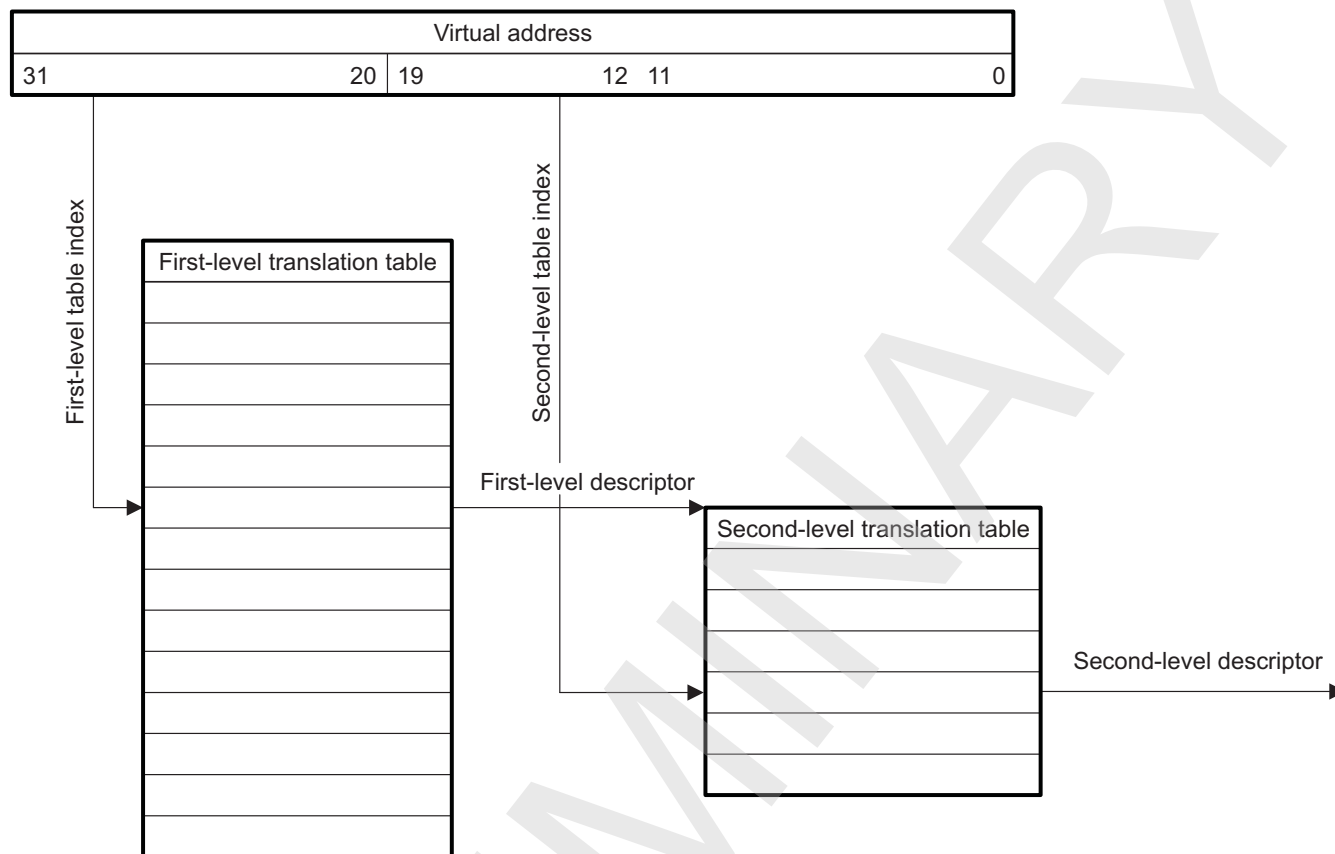


mmu-012

**20.3.1.2.3 Two-Level Translation**

Two-level translation is used when fine-grain granularity is required, that is, when memory sections smaller than 1MB are needed. In this case, the first-level descriptor provides a pointer to the base address of a second-level translation table. This second-level table is indexed by bits 19 to 12 of the virtual address. Figure 20-11 shows this indexing mechanism.

**Figure 20-11. Two-Level Translation**



mmu-013

Each second-level translation table describes the translation of 1MB of address space in pages of 64KB (large page) or 4KB (small page). It consists of 256 second-level descriptors describing 4KB each.

**NOTE:** In the case of a large page, the same descriptor must be repeated 16 times. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way.

**20.3.1.2.3.1 Second-Level Descriptor Format**

Similar to first-level section descriptors, second-level descriptors provide all of the necessary information for the translation of a large or small page. Table 20-5 shows the format of second-level descriptors. The translation parameters (endianness, element size, and mixed region) have the same meaning as those for sections.

**Table 20-5. Second-Level Descriptor Format**

Second-Level Descriptor Format											
31:16	15:12	11	10	9	8:6	5:4	3:2	1	0		
X									0	0	Fault
Large Page Base Address	X	M	X	E <sup>(1)</sup>	X	ES	X	0	1	Large Page	
Small Page Base Address		M	X	E	X	ES	X	1	X	Small Page	

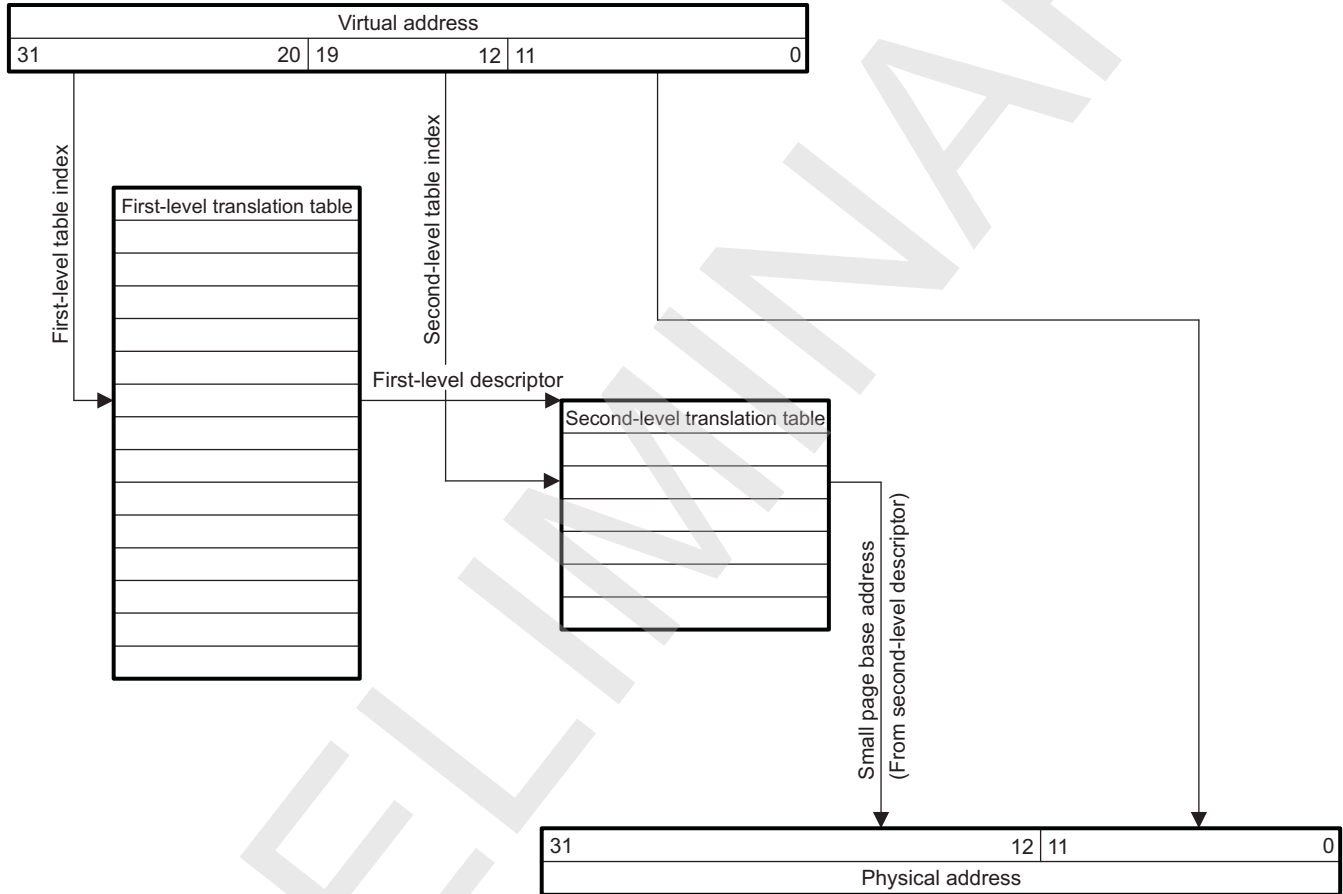
<sup>(1)</sup> See Table 20-42 for endianness limitations.

- M = Mixed region: 0 = Page-based endianness, 1 = Access-based endianness
- E = Endianness: 0 = Little-endian, 1 = Big-endian (endianness is locked on little endian)
- ES = Element Size: 00 = 8-bit, 01 = 16-bit, 10 = 32-bit, 11 = No endianness conversion
- X = Don't care

**20.3.1.2.3.2 Small Page Translation Summary**

Figure 20-12 summarizes the translation process for small pages.

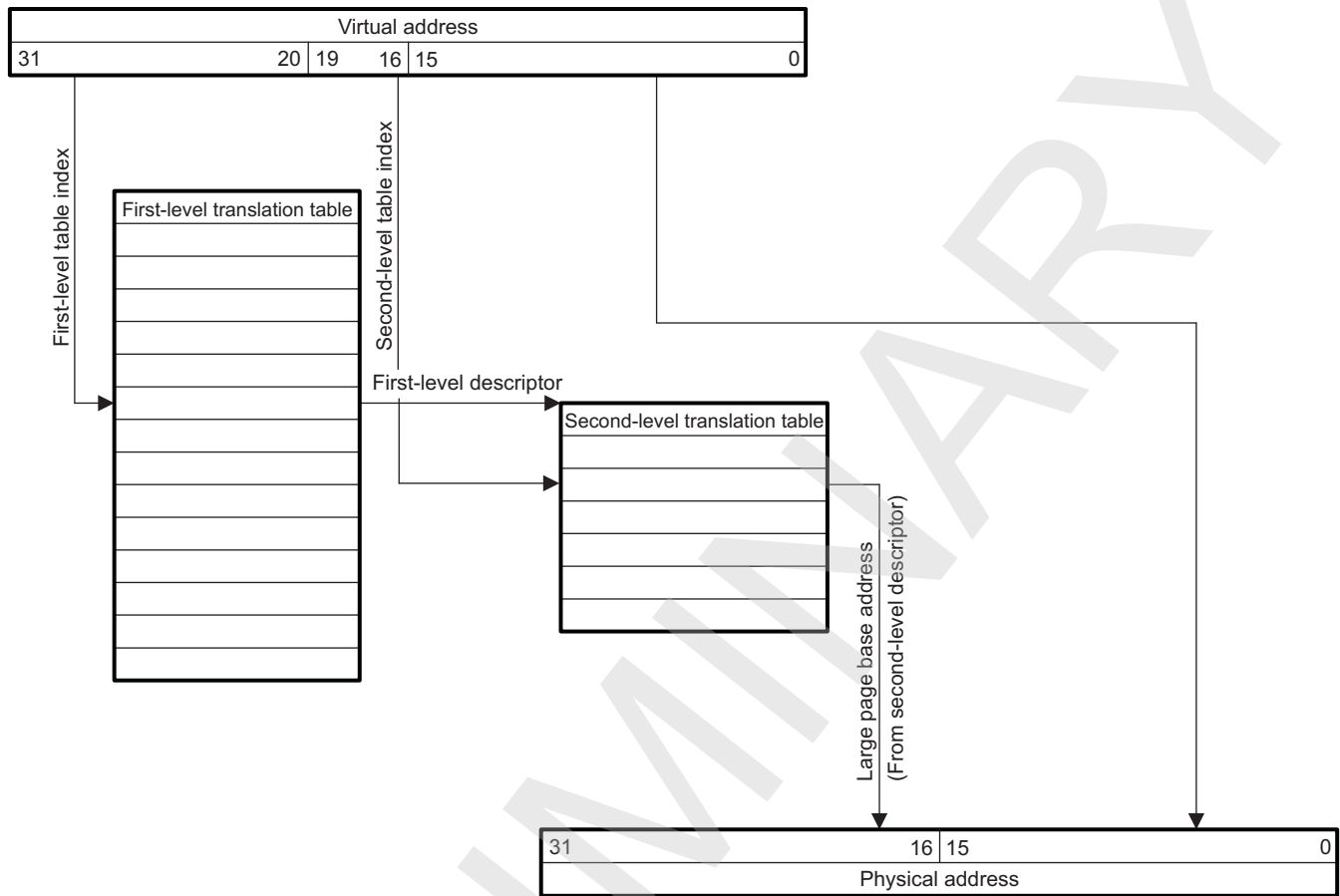
**Figure 20-12. Small Page Translation Summary**



mmu-014

**20.3.1.2.3.3 Large Page Translation Summary**

The translation of a large page is similar to the translation of a small page. The difference is that, for a large page, only bits 19 to 16 index into the second-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a large page. This is shown in Figure 20-13.

**Figure 20-13. Large Page Translation Summary**

mmu-015

### 20.3.1.3 Translation Lookaside Buffer

Translating virtual addresses to physical addresses is required for each memory access in systems using an MMU. To accelerate this translation process, a cache, or TLB, holds the result of recent translations.

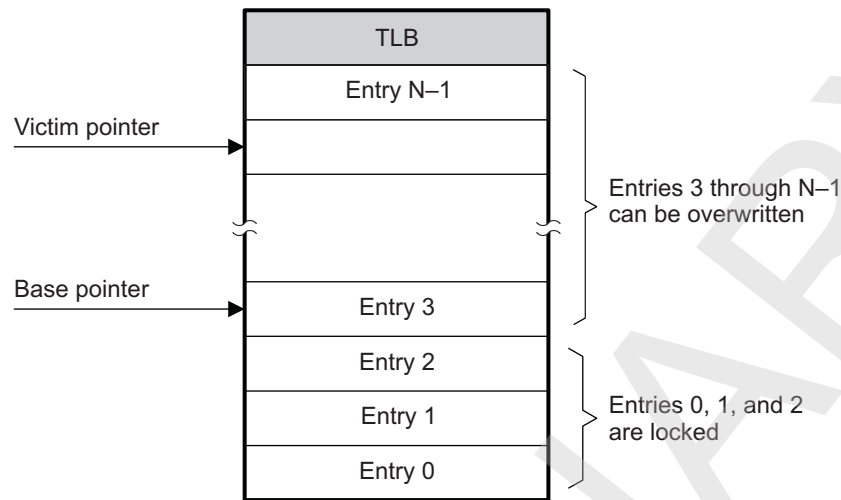
For every translation, the MMU internal logic first checks whether the requested translation is already cached in the TLB. If the translation is cached, this translation is used; otherwise the translation is retrieved from the translation tables and the TLB is updated. If the TLB is full, one of its entries must be replaced. This entry is selected on a random basis.

The first  $n$  TLB entries, where  $n < \text{Total Number } N \text{ of TLB Entries}$ , can be protected (locked) against being overwritten by setting the TLB base pointer to  $n$ . When this mechanism is used, only unprotected entries can be overwritten. The victim pointer indicates the next TLB entry to be written. Figure 20-14 shows an example of the TLB with  $N$  TLB entries (ranging from 0 to  $N-1$ ). The base pointer contains the value "3" protecting Entry 0, Entry 1, and Entry 2 and the victim pointer points to the next TLB entry to be updated.

**NOTE:** The last TLB entry (Entry  $N-1$ ) always remains unprotected.



Figure 20-14. TLB Entry Lock Mechanism



The table walking logic automatically writes the TLB entries. The entries can also be manually written, which is done typically to ensure that the translation of time-critical data accesses is already present in the TLB so that they execute as fast as possible. The entries must be locked to prevent them from being overwritten.

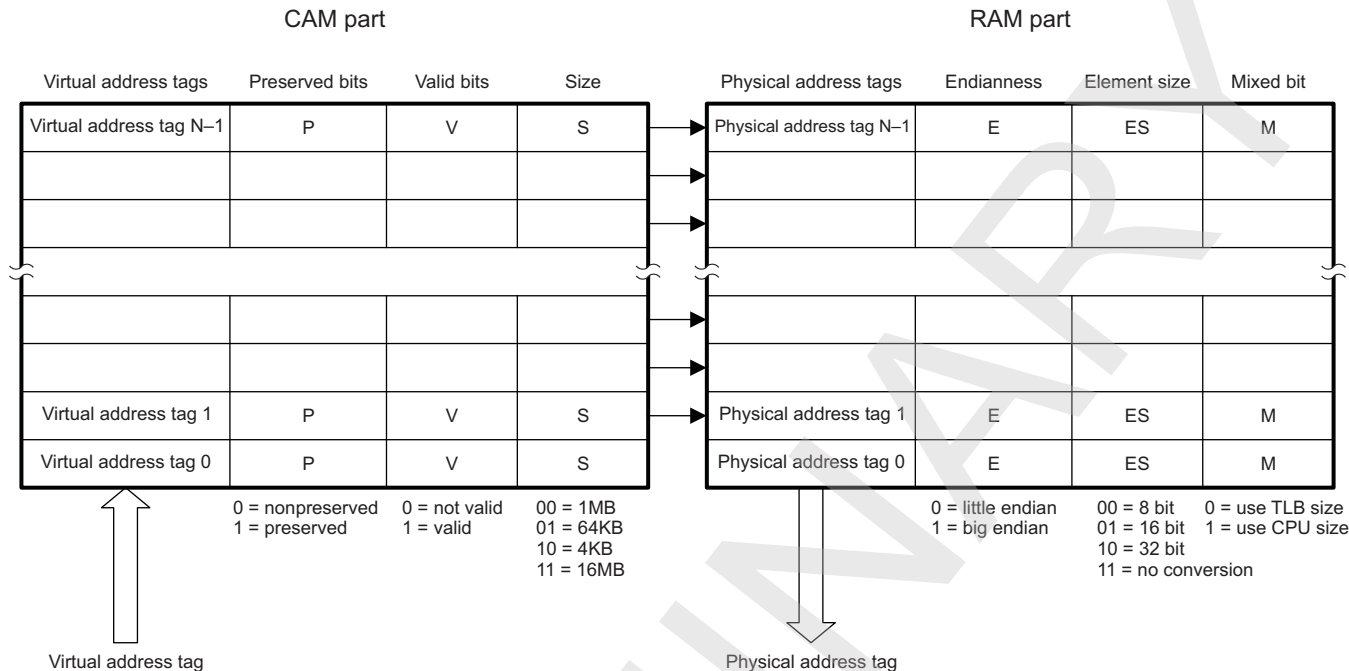
### 20.3.1.3.1 TLB Entry Format

TLB entries consist of two parts:

- The CAM part contains the virtual address tag used to determine if a virtual address translation is in the TLB. The TLB acts like a fully associative cache addressed by the virtual address tag. The CAM part also contains the section/page size, as well as the preserved and the valid parameters. See the [MMU\\_CAM](#) register table for more details.
- The RAM part contains the address translation that belongs to the virtual address tag as well as the endianness, element size, and mixed parameters described in [Section 20.3.1.2.2, First-Level Translation Table](#). See the [MMU\\_RAM](#) register table for more details.

The valid parameter specifies whether an entry is valid or not. The preserved parameter determines the behavior of an entry in the event of a TLB flush. If an entry is set as preserved, it is not deleted when a TLB is flushed, that is, when [MMU\\_GFLUSH\[0\] GLOBALFLUSH](#) is set to 1. Preserved entries must be deleted manually. [Section 20.3.1.2.2](#) describes the procedure to delete TLB entries.

[Figure 20-16](#) shows the TLB entry structure.

**Figure 20-15. TLB Entry Structure**

mmu-017

### 20.3.2 MMU Clock Configuration

There are two clock domains: The functional clock domain for the MMU, which is synchronous to the clock for the interconnect slave and master access ports; and the clock domain for the interconnect slave configuration port. As these clocks are matched, there is a single input clock with enables for each of the clock domains. If a clock domain should run at the same frequency as the input clock, that enable can be tied high.

Two clock enable signals exist, one to enable the interconnect data master and slave ports, and the other to enable the clock on the configuration L3 interconnect port. The clock signals are configured through the [MMU\\_SYSCONFIG](#) register. This is a system configuration register that controls the various parameters of the L3 interface.

### 20.3.3 MMU Software Reset

This section describes the software reset feature of the module. The MMU instances are reset together with their respective reset domains. See [Table 20-2](#) for information about the reset domains of the different MMU instances.

To perform a software reset, write 1 in the [MMU\\_SYSCONFIG\[1\]](#) SOFTRESET bit. The [MMU\\_SYSSTATUS\[0\]](#) RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the [MMU\\_SYSCONFIG\[1\]](#) SOFTRESET bit is automatically reset. The software must ensure that the software reset completes before doing MMU operations. When an MMU instance is released from reset, its TLB is empty and the MMU is disabled.

### 20.3.4 MMU Power Management

As part of the device system-wide power management scheme, each MMU instance supports a communication protocol with the PRCM module that allows the PRCM module to request an MMU instance to enter a low-power state. When the MMU instance acknowledges a low-power mode request from the PRCM module, the clock to the instance is gated off at the PRCM clock generator. Because the clock is disabled at the source, the low-power mode offers lower power consumption than the internal clock gating method in the local power management.

[Table 20-6](#) describes the power-management features available for the MMU modules.

**NOTE:**

- For information about source clock gating and sleep/wake-up transitions description, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).
- For descriptions of EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see [Section 3.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**Table 20-6. MMU Local Power Management Features**

Feature	Register
Idle modes	MMU_SYSCONFIG[4:3] IDLEMODE
Clock activity	MMU_SYSCONFIG[9:8] CLOCKACTIVITY
Clock autogating	MMU_SYSCONFIG[0] AUTOIDLE

**NOTE:** The MMU\_SYSCONFIG[9:8] CLOCKACTIVITY bits are read only.

**20.3.5 MMU Interrupt Requests**

Table 20-7 lists the event flags and their masks that can cause module interrupts.

**Table 20-7. Events**

Event Flag	Event Mask	Synchrono- us	Sensitivit- y	Map to	Description
MMU_IRQSTATUS[4] MULTIHITFAULT	MMU_IRQENABLE[4] MULTIHITFAULT	Yes	Level	M3_IRQ_0	Error in the L2 MMU due to multiple matches in the TLB
MMU_IRQSTATUS[3] TABLEWALKFAULT	MMU_IRQENABLE[3] TABLEWALKFAULT	Yes	Level	M3_IRQ_0	Error in the L2 MMU due to error response received during a Table Walk
MMU_IRQSTATUS[2] EMUMISS	MMU_IRQENABLE[2] EMUMISS				For more information about emulation and debug features, see <a href="#">Chapter 28, On-Chip Debug Support</a> .
MMU_IRQSTATUS[1] TRANSLATIONFAULT	MMU_IRQENABLE[1] TRANSLATIONFAULT	Yes	Level	M3_IRQ_0	Error in the L2 MMU due to invalid descriptor in the translation tables (translation fault)
MMU_IRQSTATUS[0] TLBMISS	MMU_IRQENABLE[0] TLBMISS	Yes	Level	M3_IRQ_0	Error in L2 MMU due to unrecoverable TLB miss (hardware TWL disabled)

**20.3.6 MMU Error Handling**

Table 20-8 summarizes the intended operation for real and potential error conditions.

**Table 20-8. Error Handling**

Item	Condition	Action
1	Table-walk read has an error response.	Treat generally the same as a translation fault, but set the TableWalkFault interrupt status bit to aid in diagnosis
2	MMU is disabled during table-walk.	Not permitted; can result in loss of the current transaction but must not deadlock the MMU.  Avoid this condition by first disabling the table-walk logic and then polling the TWLRunning bit to ensure that no table walk is pending
3	MMU is disabled during an address translation.	Not permitted; can result in access to an unintended location, but must not deadlock MMU.  This condition should be avoided by ensuring that no accesses are pending.
4	TLB is accessed during an address translation or a table walk.	Reading permitted; write should be done with care to ensure that the TLB is self-consistent at all times that a translation can occur.

**Table 20-8. Error Handling (continued)**

Item	Condition	Action
5	TLB is flushed during address translation or a table walk.	Permitted; the flush is processed first, followed by the TWL update.
6	MMU is disabled while an interrupt is pending.	Not permitted; all pending interrupts should be processed before disabling the MMU.

L3 Interconnect configuration port: Accesses to undecoded register addresses must not give an error response.

To protect against changes to the address translation between a READEX and the corresponding write or during a burst the following configuration operations are protected against writes during these processes:

- TLB update
- Global flush
- Flush entry
- MMU disable

The protection is implemented by stalling the configuration interconnect transaction until the write can proceed safely.

## 20.4 MMU Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

### 20.4.1 Global Initialization

#### 20.4.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the MMU module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the MMU. For more information, see Section [Section 20.2, MMU Module Integration](#).

**Table 20-9. Global Initialization of Surrounding Modules**

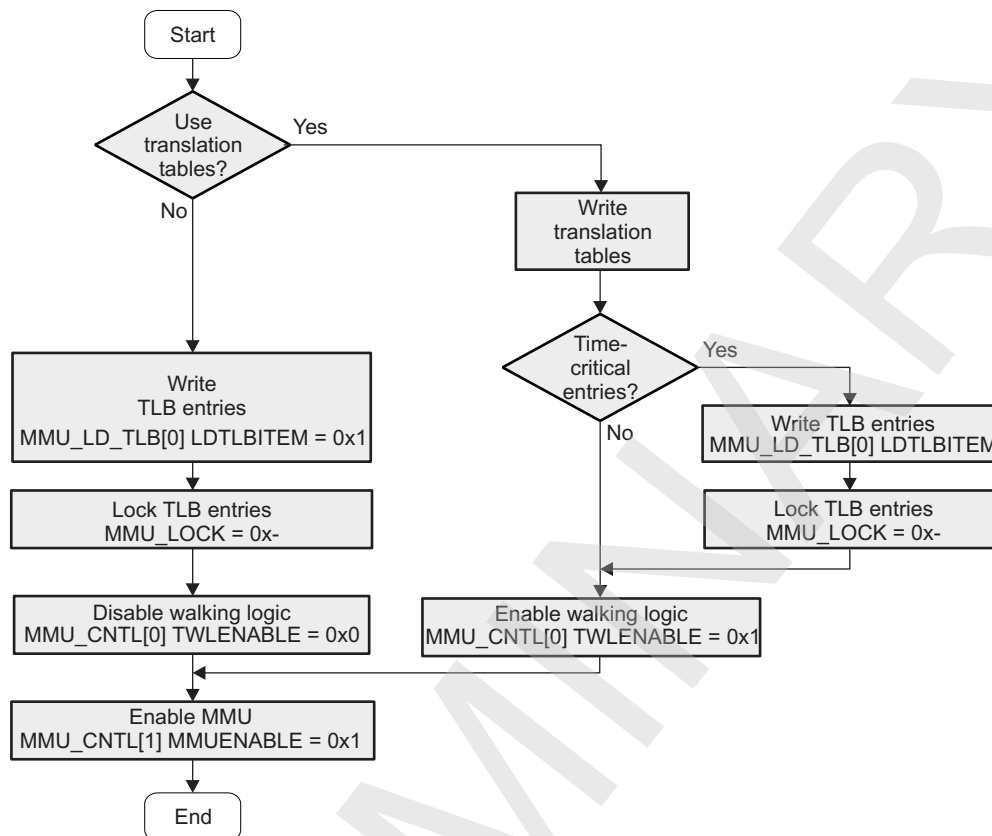
Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. For more information, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
(optional) MPU INTC or DSP INTC or Cortex-M3 INTC	MPU, DSP, and Cortex-M3 interrupt controller configuration must be done to enable the interrupts from module.
Interconnect	For more information about the L3 interconnect configuration, see <a href="#">Section 13.2, L3 Interconnect</a> .

**NOTE:** If the interrupt-based communication mode is used, the MPU, DSP, and Cortex-M3 interrupt controllers configurations are required.

#### 20.4.1.2 MMU Global Initialization

##### 20.4.1.2.1 Main Sequence—MMU Global Initialization

[Figure 20-16](#) shows the procedure to initialize the MMU after a power-on or software reset.

**Figure 20-16. MMU Global Initialization**

mmu-018

**Table 20-10. Register Call Summary for Main Sequence—MMU Global Initialization**

Register Name	Register Name	Register Name
MMU_LD_TLB	MMU_LOCK	MMU_CNTL

#### 20.4.1.2.2 Subsequence—Configure a TLB entry

**Table 20-11. Configure a TLB Entry**

Step	Register / Bit Field / Programming Model	Value
Load the Virtual Address Tag	MMU_CAM[31:12] VATAG	0x-
Protect the TLB entry against flush	MMU_CAM[3] P	0x1
Validate the TLB entry	MMU_CAM[2] V	0x1
Define the page size	MMU_CAM[1:0] PAGESIZE	0x-

### 20.4.1.3 Operational Modes Configuration

#### 20.4.1.3.1 Main Sequence—Writing TLB Entries Statically

Writing TLB entries statically avoids the need to write translation tables in memory and is commonly used for relatively small address spaces. This method ensures that the translation of time-critical data accesses execute as fast as possible with entries already present in the TLB. These entries must be locked to prevent them from being overwritten.

**Table 20-12. MMU Writing TLB Entries Statically**

Step	Register/ Bit Field / Programming Model	Value
Execute software reset	MMU_SYSCONFIG[1] SOFTRESET	0x1
Wait for reset to complete	MMU_SYSSTATUS[0] RESETDONE	=0x1
Enable power saving via automatic interface clock gating	MMU_SYSCONFIG[0] AUTOIDLE	0x1
Configure TLB entries	See Table 20-11	
Load the physical Address of the page	MMU_RAM[31:12] PHYSICALADDRESS	0x-
Define the endianness of the page (little endian or big endian)	MMU_RAM[9] ENDIANNES	0x-
Select the element size	MMU_RAM[8:7] ELEMENTSIZE	0x-
Define mixed page attribute	MMU_RAM[6] MIXED	0x-
Specify the TLB entry you want to write	MMU_LOCK[8:4] CURRENTVICTIM	0x-
Load the specified entry in the TLB	MMU_LD_TLB[0] LDTLBITEM	0x1
Enable multihit fault and TLB miss	MMU_IRQENABLE[4] MULTIHITFAULT	0x1
	MMU_IRQENABLE[0] TLBMISS	0x1
Enable memory translations	MMU_CNTL[1] MMUENABLE	0x1

#### 20.4.1.3.2 Main Sequence—Protecting TLB Entries

The first  $n$  TLB entries (with  $n <$  total number of TLB entries) can be protected from being overwritten with new translations. This is useful to ensure that certain commonly used or time-critical translations are always in the TLB and do not require retrieval using the table walking process.

**Table 20-13. Protecting TLB Entries**

Step	Register/Bit Field/Programming Model	Value
Locks the TLB entries	MMU_LOCK[14:10] BASEVALUE	0x-

#### 20.4.1.3.3 Main Sequence—Deleting TLB Entries

Two mechanisms exist to delete TLB entries. All unpreserved TLB entries, that is, TLB entries written with the preserved bit set to zero, can be deleted by invoking a TLB flush. The preserved bit should only be used on protected TLB entries, as it does not prevent replacement by the table walking logic.

**Table 20-14. Deleting TLB Entries**

Step	Register / Bit Field / Programming Model	Value
Flush all nonprotected TLB entries	MMU_GFLUSH[0] GLOBALFLUSH	0x1
Flush all TLB entries specified by the CAM register	MMU_FLUSH_ENTRY[0] FLUSHENTRY	0x1

#### 20.4.1.3.4 Main Sequence—Read TLB Entries

TLB entries can be read by the programmer to determine the TLB content at runtime.

**Table 20-15. Read TLB Entries**

Step	Register / Bit Field / Programming Model	Value
Set the current victim pointer	MMU_LOCK[8:4] CURRENTVICTIM	0x-
Read RAM parts of the TLB entry	MMU_READ_RAM	
Read CAM parts of the TLB entry	MMU_READ_CAM	

## 20.5 MMU Register Manual

### 20.5.1 MMU Instance Summary

Table 20-16. MMU Instance Summary

Module Name	Base Address	Size
CORTEXM3_L2MMU	0x5508 2000	4KB
DSP MMU	0x4A06 6000	4KB

### 20.5.2 MMU Registers

#### 20.5.2.1 CORTEXM3\_L2MMU and DSP MMU Register Summary

Table 20-17. CORTEXM3\_L2MMU and DSP MMU Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CORTEXM3_L2M MU Base Address	DSP MMU Base Address
MMU_REVISION	R	32	0x0000 0000	0x5508 2000	0x4A06 6000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x5508 2010	0x4A06 6010
MMU_SYSSTATUS	R	32	0x0000 0014	0x5508 2014	0x4A06 6014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x5508 2018	0x4A06 6018
MMU_IRQENABLE	RW	32	0x0000 001C	0x5508 201C	0x4A06 601C
MMU_WALKING_ST	R	32	0x0000 0040	0x5508 2040	0x4A06 6040
MMU_CNTL	RW	32	0x0000 0044	0x5508 2044	0x4A06 6044
MMU_FAULT_AD	R	32	0x0000 0048	0x5508 2048	0x4A06 6048
MMU_TTB	RW	32	0x0000 004C	0x5508 204C	0x4A06 604C
MMU_LOCK	RW	32	0x0000 0050	0x5508 2050	0x4A06 6050
MMU_LD_TLB	RW	32	0x0000 0054	0x5508 2054	0x4A06 6054
MMU_CAM	RW	32	0x0000 0058	0x5508 2058	0x4A06 6058
MMU_RAM	RW	32	0x0000 005C	0x5508 205C	0x4A06 605C
MMU_GFLUSH	RW	32	0x0000 0060	0x5508 2060	0x4A06 6060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x5508 2064	0x4A06 6064
MMU_READ_CAM	R	32	0x0000 0068	0x5508 2068	0x4A06 6068
MMU_READ_RAM	R	32	0x0000 006C	0x5508 206C	0x4A06 606C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x5508 2070	0x4A06 6070
MMU_FAULT_PC	R	32	0x0000 0080	0x5508 2080	0x4A06 6080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x5508 2084	0x4A06 6084
MMU_GP_REG	RW	32	0x0000 0088	0x5508 2088	N/A

**NOTE:** [MMU\\_IRQENABLE](#), [MMU\\_CNTL](#) and [MMU\\_TTB](#) registers have retention capabilities. For more information about the device retention state management, see the PRCM chapter.

#### 20.5.2.2 MMU Register Description



**Table 20-18. MMU\_REVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register contains the IP revision code		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 20-19. Register Call Summary for Register MMU\_REVISION**

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[0\]](#)

**Table 20-20. MMU\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register controls the various parameters of the L3 interconnect interface		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLOCKACTIVITY	RESERVED	IDLEMODE	RESERVED	SOFTRESET	AUTOIDLE										

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0's for future compatibility Reads returns 0	R	0x000000
9:8	CLOCKACTIVITY	Clock activity during wake-up mode 00 Functional and Interconnect clocks can be switched off	R	0x0
7:5	RESERVED	Write 0's for future compatibility Reads returns 0	R	0x0
4:3	IDLEMODE	IdleMode 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module 0x3: reserved do not use	RW	0x0
2	RESERVED	Write 0's for future compatibility Reads returns 0	R	0
1	SOFTRESET	Software reset. This bit is automatically reset by the hardware. During reads, it always return 0 Read 0x0: always return 0 Write 0x0: no functional effect Write 0x1: The module is reset Read 0x1: never happens	RW	0

Bits	Field Name	Description	Type	Reset
0	AUTOIDLE	Internal interconnect clock gating strategy 0x0: interconnect clock is free-running 0x1: Automatic interconnect clock gating strategy is applied, based on the interconnect interface activity	RW	0

**Table 20-21. Register Call Summary for Register MMU\_SYSCONFIG**

## MMU Functional Description

- [MMU Clock Configuration: \[0\]](#)
- [MMU Software Reset: \[1\] \[2\]](#)
- [MMU Power Management: \[3\] \[4\] \[5\] \[6\]](#)

## MMU Low-level Programming Models

- [Main Sequence—Writing TLB Entries Statically: \[7\] \[8\]](#)

## MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[9\]](#)

**Table 20-22. MMU\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register provides status information about the module, excluding the interrupt status information		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RESETDONE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	R	0x000000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset in on-going Read 0x1: Reset completed	R	-

**Table 20-23. Register Call Summary for Register MMU\_SYSSTATUS**

## MMU Functional Description

- [MMU Software Reset: \[0\]](#)

## MMU Low-level Programming Models

- [Main Sequence—Writing TLB Entries Statically: \[1\]](#)

## MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[2\]](#)

**Table 20-24. MMU\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This interrupt status register regroups all the status of the module internal events that can generate an interrupt.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULTIHITFAULT		TABLEWALKFAULT	EMUMISS	TRANSLATIONFAULT		TLBMISS									

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0's for future compatibility read returns 0	R	0x00000000
4	MULTIHITFAULT	Error due to multiple matches in the TLB Read 0x0: MultiHitFault false Write 0x0: MultiHitFault status bit unchanged Write 0x1: MultiHitFault status bit is reset Read 0x1: MultiHitFault is true ("pending")	RW W1toClr	0
3	TABLEWALKFAULT	Error response received during a Table Walk Read 0x0: TableWalkFault false Write 0x0: TableWalkFault status bit unchanged Write 0x1: TableWalkFault status bit is reset Read 0x1: TableWalkFault is true ("pending")	RW W1toClr	0
2	EMUMISS	Unrecoverable TLB miss during debug (hardware TWL disabled) Read 0x0: EMUMiss false Write 0x0: EMUMiss status bit unchanged Write 0x1: EMUMiss status bit is reset Read 0x1: EMUMiss is true ("pending")	RW W1toClr	0
1	TRANSLATIONFAULT	Invalid descriptor in translation tables (translation fault) Read 0x0: TranslationFault false Write 0x0: TranslationFault status bit unchanged Write 0x1: TranslationFault status bit is reset Read 0x1: TranslationFault is true ("pending")	RW W1toClr	0
0	TLBMISS	Unrecoverable TLB miss (hardware TWL disabled) Read 0x0: TLBMiss false Write 0x0: TLBMiss status bit unchanged Write 0x1: TLBMiss status bit is reset Read 0x1: TLBMiss is true ("pending")	RW W1toClr	0

**Table 20-25. Register Call Summary for Register MMU\_IRQSTATUS**

MMU Functional Description

- [MMU Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[5\]](#)

**Table 20-26. MMU\_IRQENABLE**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MULTIHITFAULT		TABLEWALKFAULT		EMUMISS		TRANSLATIONFAULT		TLBMISS							

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0's for future compatibility Read returns 0	R	0x0000000
4	MULTIHITFAULT	Error due to multiple matches in the TLB 0x0: MultiHitFault is masked 0x1: MultiHitFault event generates an interrupt if occurs	RW	0
3	TABLEWALKFAULT	Error response received during a Table Walk 0x0: TableWalkFault is masked 0x1: TableWalkFault event generates an interrupt if occurs	RW	0
2	EMUMISS	Unrecoverable TLB miss during debug (hardware TWL disabled) 0x0: EMUMiss interrupt is masked 0x1: EMUMiss event generates an interrupt when it occurs	RW	0
1	TRANSLATIONFAULT	Invalid descriptor in translation tables (translation fault) 0x0: TranslationFault is masked 0x1: TranslationFault event generates an interrupt if occurs	RW	0
0	TLBMISS	Unrecoverable TLB miss (hardware TWL disabled) 0x0: TLBMiss interrupt is masked 0x1: TLBMiss event generates an interrupt when if occurs	RW	0

**Table 20-27. Register Call Summary for Register MMU\_IRQENABLE**

## MMU Functional Description

- [MMU Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

## MMU Low-level Programming Models

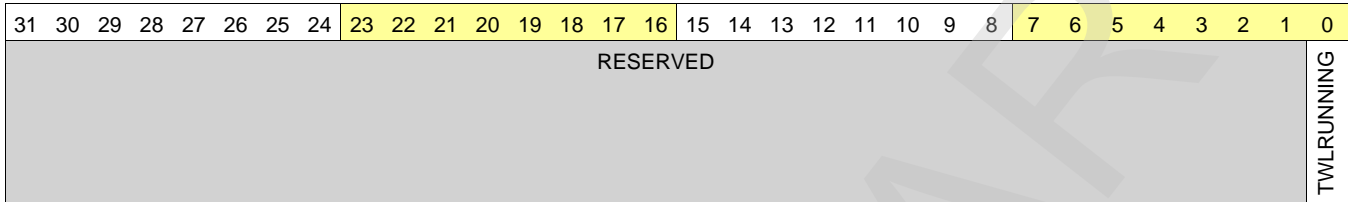
- [Main Sequence—Writing TLB Entries Statically: \[5\] \[6\]](#)

## MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[7\] \[8\]](#)

**Table 20-28. MMU\_WALKING\_ST**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register provides status information about the table walking logic		
<b>Type</b>	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0	R	0x0000 0000
0	TWLRUNNING	Table Walking Logic is running Read 0x0: TWL Completed Read 0x1: TWL Running	R	0

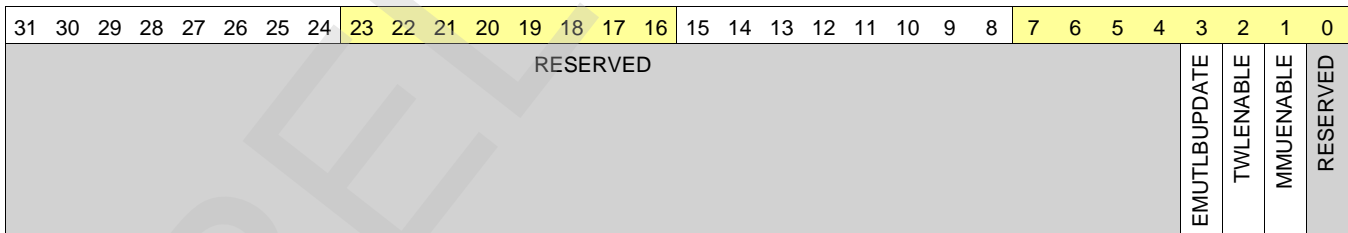
**Table 20-29. Register Call Summary for Register MMU\_WALKING\_ST**

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[0\]](#)

**Table 20-30. MMU\_CNTL**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register programs the MMU features		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Write 0's for future compatibility Reads return 0	R	0x00000000
3	EMUTLBUPDATE	Enable TLB update on emulator table walk 0x0: Emulator TLB update disabled 0x1: Emulator TLB update enabled	RW	0
2	TWLENABLE	Table Walking Logic enable 0x0: TWL disabled 0x1: TWL enabled	RW	0
1	MMUENABLE	MMU enable 0x0: MMU disabled 0x1: MMU enabled	RW	0
0	RESERVED	Write 0's for future compatibility Reads return 0	R	0

**Table 20-31. Register Call Summary for Register MMU\_CNTL**

MMU Low-level Programming Models

- [Main Sequence—MMU Global Initialization: \[0\]](#)
- [Main Sequence—Writing TLB Entries Statically: \[1\]](#)

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[2\] \[3\]](#)

**Table 20-32. MMU\_FAULT\_AD**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register contains the virtual address that generated the interrupt		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAULTADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	FAULTADDRESS	Virtual address of the access that generated a fault	R	0x0000 0000

**Table 20-33. Register Call Summary for Register MMU\_FAULT\_AD**

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[0\]](#)

**Table 20-34. MMU\_TTB**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register contains the Translation Table Base address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTBADDRESS																RESERVED															

Bits	Field Name	Description	Type	Reset
31:7	TTBADDRESS	Translation Table Base Address	RW	0x00000000
6:0	RESERVED	Write 0's for future compatibility Reads return 0	R	0x00

**Table 20-35. Register Call Summary for Register MMU\_TTB**

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[0\] \[1\]](#)

**Table 20-36. MMU\_LOCK**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register locks some of the TLB entries		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASEVALUE				RESERVED	CURRENTVICTIM				RESERVED						

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Write 0's for future compatibility Reads return 0	R	0x00000
14:10	BASEVALUE	Locked entries base value	RW	0x00
9	RESERVED	Write 0's for future compatibility Read returns 0	R	0
8:4	CURRENTVICTIM	Current entry to be updated either by the TWL or by the SW Write value : TLB entry to be updated by software, Read value : TLB entry that will be updated by table walk logic	RW	0x00
3:0	RESERVED	Write 0's for future compatibility Reads return 0	R	0x0

**Table 20-37. Register Call Summary for Register MMU\_LOCK**

MMU Low-level Programming Models

- [Main Sequence—MMU Global Initialization: \[0\]](#)
- [Main Sequence—Writing TLB Entries Statically: \[1\]](#)
- [Main Sequence—Protecting TLB Entries: \[2\]](#)
- [Main Sequence—Read TLB Entries: \[3\]](#)

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[4\]](#)

**Table 20-38. MMU\_LD\_TLB**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register loads a TLB entry (CAM+RAM)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	LDTLBITEM														

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility Reads return 0	RW	0x0000 0000
0	LDTLBITEM	Write (load) data in the TLB Read 0x0: always return 0 Write 0x0: no functional effect Write 0x1: load TLB data Read 0x1: never happens	RW	0

**Table 20-39. Register Call Summary for Register MMU\_LD\_TLB**

MMU Low-level Programming Models

- [Main Sequence—MMU Global Initialization: \[0\]](#)
- [Main Sequence—Writing TLB Entries Statically: \[1\]](#)

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[2\]](#)

**Table 20-40. MMU\_CAM**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register holds a CAM entry		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VATAG																RESERVED								P	V	PAGESIZE					

Bits	Field Name	Description	Type	Reset
31:12	VATAG	Virtual address tag	RW	0x00000
11:4	RESERVED	Write 0's for future compatibility Reads return 0	R	0x00
3	P	Preserved bit 0x0: TLB entry may be flushed 0x1: TLB entry is protected against flush	RW	0
2	V	Valid bit 0x0: TLB entry is invalid 0x1: TLB entry is valid	RW	0
1:0	PAGESIZE	Page size 0x0: Section (1MB) 0x1: Large page (64KB) 0x2: Small page (4KB) 0x3: Supersection (16MB)	RW	0x0

**Table 20-41. Register Call Summary for Register MMU\_CAM**

MMU Functional Description

- [TLB Entry Format: \[0\]](#)

MMU Low-level Programming Models

- [Subsequence—Configure a TLB entry: \[1\] \[2\] \[3\] \[4\]](#)

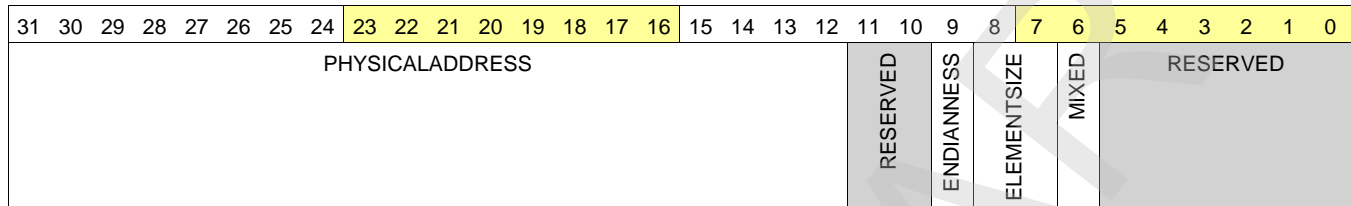
MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[5\]](#)
- [MMU Register Description: \[6\]](#)



**Table 20-42. MMU\_RAM**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register holds a RAM entry		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:12	PHYSICALADDRESS	Physical address of the page	RW	0x00000
11:10	RESERVED	Write 0's for future compatibility Reads return 0	R	0x0
9	ENDIANNESS	Endianness of the page 0x0: Little Endian 0x1: Big endian	RW	0
8:7	ELEMENTSIZE	Element size of the page (8, 16, 32, no translation) 0x0: 8-bits 0x1: 16-bits 0x2: 32-bits 0x3: No translation	RW	0x0
6	MIXED	Mixed page attribute (use CPU element size) 0x0: Use TLB element size 0x1: Use CPU element size	RW	0
5:0	RESERVED	Write 0's for future compatibility Reads return 0	R	0x00

**Table 20-43. Register Call Summary for Register MMU\_RAM**

MMU Functional Description	<ul style="list-style-type: none"> <li>• <a href="#">TLB Entry Format: [0]</a></li> </ul>
MMU Low-level Programming Models	<ul style="list-style-type: none"> <li>• <a href="#">Main Sequence—Writing TLB Entries Statically: [1] [2] [3] [4]</a></li> </ul>
MMU Register Manual	<ul style="list-style-type: none"> <li>• <a href="#">CORTEXM3_L2MMU and DSP MMU Register Summary: [5]</a></li> </ul>

**Table 20-44. MMU\_GFLUSH**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register flushes all the non-protected TLB entries		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												GLOBALFLUSH			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility Reads return 0	RW	0x0000 0000
0	GLOBALFLUSH	Flush all the non-protected TLB entries when set Read 0x0: always return 0 Write 0x0: no functional effect Write 0x1: flush all the non-protected TLB entries Read 0x1: never happens	RW	0

**Table 20-45. Register Call Summary for Register MMU\_GFLUSH**

## MMU Functional Description

- [TLB Entry Format: \[0\]](#)

## MMU Low-level Programming Models

- [Main Sequence—Deleting TLB Entries: \[1\]](#)

## MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[2\]](#)

**Table 20-46. MMU\_FLUSH\_ENTRY**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register flushes the entry pointed to by the CAM virtual address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												FLUSHENTRY			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility Reads return 0	RW	0x0000 0000
0	FLUSHENTRY	Flush the TLB entry pointed by the virtual address (VATag) in <a href="#">MMU_CAM</a> register, even if this entry is set protected Read 0x0: always return 0 Write 0x0: no functional effect Write 0x1: flush all the TLB entries specified by the CAM register Read 0x1: never happens	RW	0

**Table 20-47. Register Call Summary for Register MMU\_FLUSH\_ENTRY**

MMU Low-level Programming Models

- [Main Sequence—Deleting TLB Entries: \[0\]](#)

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[1\]](#)

**Table 20-48. MMU\_READ\_CAM**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register reads CAM data from a CAM entry		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VATAG																RESERVED							P	V	PAGESIZE						

Bits	Field Name	Description	Type	Reset
31:12	VATAG	Virtual address tag	R	0x00000
11:4	RESERVED	Reads return 0	R	0x00
3	P	Preserved bit Read 0x0: TLB entry may be flushed Read 0x1: TLB entry is protected against flush	R	0
2	V	Valid bit Read 0x0: TLB entry is invalid Read 0x1: TLB entry is valid	R	0
1:0	PAGESIZE	Page size Read 0x0: Section (1MB) Read 0x1: Large page (64KB) Read 0x2: Small page (4KB) Read 0x3: Supersection (16MB)	R	0x0

**Table 20-49. Register Call Summary for Register MMU\_READ\_CAM**

MMU Low-level Programming Models

- [Main Sequence—Read TLB Entries: \[0\]](#)

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[1\]](#)

**Table 20-50. MMU\_READ\_RAM**

<b>Address Offset</b>	0x0000 006C	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register reads RAM data from a RAM entry		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYSICALADDRESS																RESERVED	ENDIANNESS	ELEMENTSIZE	MIXED	RESERVED											

Bits	Field Name	Description	Type	Reset
31:12	PHYSICALADDRESS	Physical address of the page	R	0x00000
11:10	RESERVED	Reads return 0	R	0x0
9	ENDIANNESS	Endianness of the page Read 0x0: Little Endian Read 0x1: Big endian	R	0
8:7	ELEMENTSIZE	Element size of the page (8, 16, 32, no translation) Read 0x0: 8-bits Read 0x1: 16-bits Read 0x2: 32-bits Read 0x3: No translation	R	0x0
6	MIXED	Mixed page attribute (use CPU element size) Read 0x0: Use TLB element size Read 0x1: Use CPU element size	R	0
5:0	RESERVED	Reads return 0	R	0x00

**Table 20-51. Register Call Summary for Register MMU\_READ\_RAM**

MMU Low-level Programming Models

- [Main Sequence—Read TLB Entries: \[0\]](#)

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[1\]](#)

**Table 20-52. MMU\_EMU\_FAULT\_AD**

<b>Address Offset</b>	0x0000 0070	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	This register contains the last virtual address of a fault caused by the debugger		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMUFAULTADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	EMUFAULTADDRESS	Virtual address of the last emulator access that generated a fault	R	0x0000 0000

**Table 20-53. Register Call Summary for Register MMU\_EMU\_FAULT\_AD**

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[0\]](#)
- [MMU Register Description: \[1\]](#)

**Table 20-54. MMU\_FAULT\_PC**

<b>Address Offset</b>	0x0000 0080	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	Capture first fault PC value, controlled by <a href="#">MMU_FAULT_STATUS[0]</a> FAULTINDICATION. Notes: The address value is captured at <a href="#">MMU_EMU_FAULT_AD[31:0]</a> EMUFAULTADDRESS. Data-Read-access : corresponding PC. Data-write-access : not perfect accuracy due to Posted-write. All this description is valid only for the DSP MMU. The Cortex-M3 L2 MMU always reads zero from this register.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC																															

Bits	Field Name	Description	Type	Reset
31:0	PC	CPU program counter value where cause MMU fault	R	0x0000 0000

**Table 20-55. Register Call Summary for Register MMU\_FAULT\_PC**

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[0\]](#)

**Table 20-56. MMU\_FAULT\_STATUS**

<b>Address Offset</b>	0x0000 0084	<b>Instance</b>	CORTEXM3_L2MMU DSP MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MMU_FAULT_TRANS_ID		RD_WR	MMU_FAULT_TYPE		FAULTINDICATION										

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:4	MMU_FAULT_TRANS_ID	<p>Master ID who cause a fault</p> <p>Read 0x0: posted writes out of shared cache for Cortex-M3 MMU / EDMA read port 1 for DSP MMU</p> <p>Read 0x1: Cortex-M3 video and display control processor I/D bus access for Cortex-M3 MMU / EDMA read port 2 for DSP MMU</p> <p>Read 0x2: Cortex-M3 video and display control processor S bus access for Cortex-M3 MMU / EDMA write port 1 for DSP MMU</p> <p>Read 0x3: Cortex-M3 SIMCOP control processor I/D bus access for Cortex-M3 MMU / EDMA write port 2 for DSP MMU</p> <p>Read 0x4: Cortex-M3 SIMCOP control processor S bus access for Cortex-M3 MMU / shared cache Evictions/stores/Non-cacheable for DSP MMU</p> <p>Read 0x5: reserved for Cortex-M3 MMU / shared cache request for Program/Data for DSP MMU</p> <p>Read 0x6: reserved for Cortex-M3 MMU / shared cache request for DMA for DSP MMU</p> <p>Read 0x7: reserved (for both Cortex-M3 MMU and DSP MMU)</p> <p>Read 0x8: MMU hardware table walk (for both Cortex-M3 MMU and DSP MMU)</p> <p>Read 0x9 to 0xF: reserved (for both Cortex-M3 MMU and DSP MMU)</p>	R	0x0
3	RD_WR	indicates read or write	R	0
2:1	MMU_FAULT_TYPE	<p>MReq Type[1:0]</p> <p>Read 0x2: reserved for Cortex-M3 MMU / DMA address for DSP MMU</p> <p>Read 0x1: Fetch address</p> <p>Read 0x0: Data Load/Store</p>	R	0x0
0	FAULTINDICATION	indicates a MMU fault	RW W1toClr	0

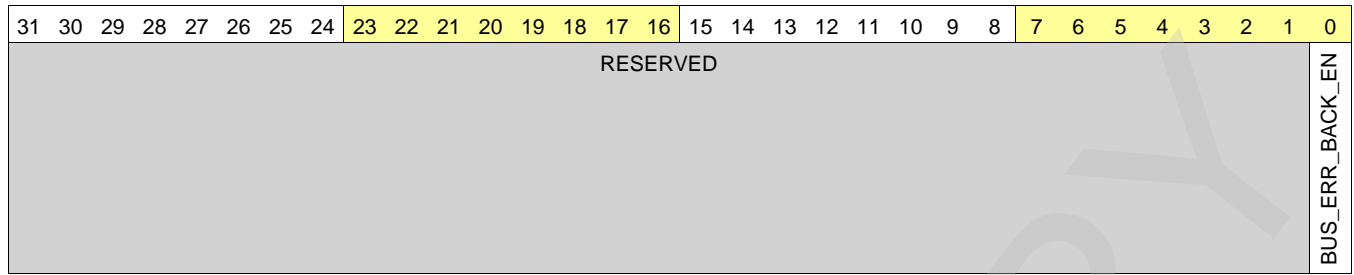
**Table 20-57. Register Call Summary for Register MMU\_FAULT\_STATUS**

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[0\]](#)
- [MMU Register Description: \[1\]](#)

**Table 20-58. MMU\_GP\_REG**

<b>Address Offset</b>	0x0000 0088	<b>Instance</b>	CORTEXM3_L2MMU
<b>Physical Address</b>	See <a href="#">Table 20-17</a>		
<b>Description</b>	Bus-error back response enable register. For more information about the register usage, see section L2 MMU, part of the Dual Cortex-M3 MPU chapter.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	BUS_ERR_BACK_EN	Bus-error back response enable bit 0x0: Default behaviour for MMU page Faults 0x1: All MMU faults return bus-error back	RW	0x0

**Table 20-59. Register Call Summary for Register MMU\_GP\_REG**

MMU Register Manual

- [CORTEXM3\\_L2MMU and DSP MMU Register Summary: \[0\]](#)

PRELIMINARY



# Spinlock

This chapter describes the Spinlock module of the device.

Topic	Page
21.1 Spinlock Overview .....	3946
21.2 Spinlock Integration .....	3947
21.3 Spinlock Functional Description .....	3948
21.4 Spinlock Programming Guide .....	3951
21.5 Spinlock Register Manual .....	3954

## 21.1 Spinlock Overview

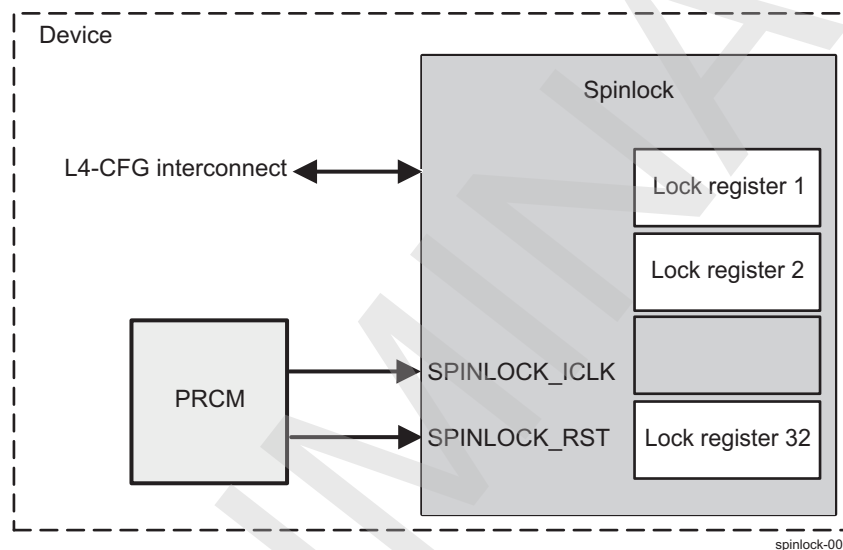
The Spinlock module provides hardware assistance for synchronizing the processes running on multiple processors in the device:

- Cortex-A9 microprocessor unit (MPU) subsystem
- Digital signal processor (DSP) subsystem
- Cortex-M3 MPU subsystem

The Spinlock module implements 32 spinlocks (or hardware semaphores), which provide an efficient way to perform a lock operation of a device resource using a single read-access, avoiding the need of a read-modify-write bus transfer that the programmable cores are not capable of.

Figure 21-1 shows the Spinlock module.

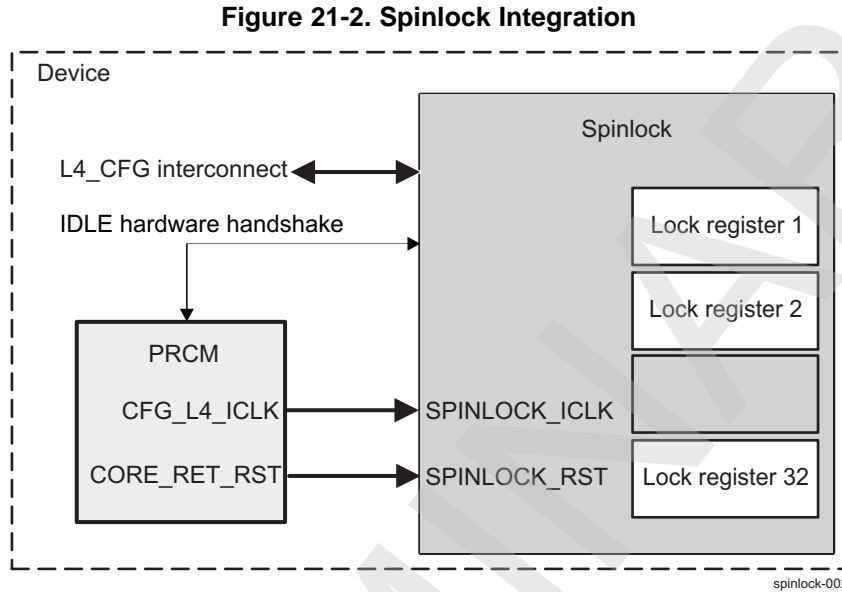
**Figure 21-1. Spinlock Module**



## 21.2 Spinlock Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 21-2 shows the Spinlock integration.



**NOTE:** For more information about the IDLE hardware handshake and the wake-up request, see [Section 3.1.1.1.2, Module Level Clock Management, Chapter 3, Power, Reset, and Clock Management](#).

Table 21-1 and Table 21-2 summarize the integration of the module in the device.

**Table 21-1. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
SPINLOCK	PD_CORE	L4_CFG

**Table 21-2. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SPINLOCK	SPINLOCK_ICLK	CFG_L4_ICLK	PRCM	Spinlock interface clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SPINLOCK	SPINLOCK_RST	CORE_RET_RST	PRCM	Spinlock hardware reset. This reset is asynchronously applied to the Spinlock internal registers.

The Spinlock module does not support any interrupt and DMA requests.

## 21.3 Spinlock Functional Description

### 21.3.1 Spinlock Software Reset

The Spinlock module can be reset by software through the `SPINLOCK_SYSCONFIG[1]` `SOFTRESET` bit. Setting this bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. The `SPINLOCK_SYSTATUS[0]` `RESETDONE` bit can be polled to check the reset status (reading 1 indicates that reset sequence is done; reading 0 indicates that reset sequence is in progress). The software must ensure that the software reset completes before doing Spinlock operations.

### 21.3.2 Spinlock Power Management

Table 21-3 describes power-management features available to the Spinlock module.

**NOTE:**

- For information about source clock gating and sleep/wake-up transitions description, see [Section 3.1.1.1.4, Clock Domain Level Clock Management, Chapter 3, Power, Reset, and Clock Management](#).
- For descriptions of `EnaWakeUp`, `IdleMode`, `ClockActivity`, and `StandbyMode` features, see [Section 3.1.1.1.2, Module Level Clock Management, Chapter 3, Power, Reset, and Clock Management](#).

**Table 21-3. Local Power Management Features**

Feature	Registers	Description
Clock auto gating	<code>SPINLOCK_SYSCONFIG[0]</code> <code>AUTOGATING</code> bit	This bit indicates that the module uses an automatic internal interface clock gating strategy, based on interface activity.
Slave idle modes	<code>SPINLOCK_SYSCONFIG[4:3]</code> <code>SIDLEMODE</code> bit field	This bit field indicates that the module uses smart-idle mode.
Clock activity	<code>SPINLOCK_SYSCONFIG[8]</code> <code>CLOCKACTIVITY</code> bit	This bit indicates that the interface clock is not required by the module during idle mode and may be switched off.
Global wake-up enable	<code>SPINLOCK_SYSCONFIG[2]</code> <code>ENAWAKEUP</code> bit	This bit indicates that the wake-up generation feature (at module level) is disabled.

**NOTE:** All the local power management features are non-configurable (that is, their respective bits or bit fields are read-only).

**CAUTION**

The PRCM module has no hardware means of reading `CLOCKACTIVITY` settings. Thus, software must ensure consistent programming between the `CLOCKACTIVITY` bit and Spinlock clock PRCM control bits.

The Spinlock module is normally idle, except when processing a request from its slave interface port. The smart-idle mode acknowledges idle requests from the PRCM only when the module is prepared to go idle. The Spinlock module is always ready to go idle if it does not have any request that it is processing.

The Spinlock module uses retention flops to retain state including the Taken state of each lock register. This means that the module can be placed in retention at any time when it is not processing a request and it is known that the system will not need to access the module.

Software must ensure to only power off the Spinlock module when no locks would be lost. In general, the steps to powering down the Spinlock module are:

- Check that all masters which might use the Spinlock module are either:
  - (a) Already powered off, or
  - (b) Notified that Spinlock is not available and the notification is acknowledged.

- If desired, check that no locks are currently held in the Spinlock module. The status of each bank of 32 locks can be read from the [SPINLOCK\\_SYSTATUS](#) register. If any locks are held, they are orphaned because they are not held by any master that is still active. Alternatively, you may decide to wait a timeout period to allow any active master to clean up its locks before powering down.
- The Spinlock module can now be powered off by writing the appropriate status to the PRCM.

In the case of powering off the whole system, these steps are unnecessary.

### 21.3.3 About Spinlocks

Spinlocks are present to solve the need for synchronization and mutual exclusion between heterogeneous processors and those not operating under a single, shared operating system. There is no alternative mechanism to accomplish these operations between processors in separate subsystems.

Spinlocks are not the best way to synchronize between tasks or threads on one CPU. Instead, spinlocks are for use in synchronization between different subsystems in the device that don't have any other means of hardware-based synchronization.

Spinlocks do not solve all system synchronization issues. They have limited applicability and should be used with care to implement higher level synchronization protocols.

A spinlock is appropriate for mutual exclusion for access to a shared data structure. It should be used only when:

1. The time to hold the lock is predictable and small (for example, a maximum hold time of less than 200 CPU cycles may be acceptable).
2. The locking task cannot be preempted, suspended, or interrupted while holding the lock (this would make the hold time large and unpredictable).
3. The lock is lightly contended, that is the chance of any other process (or processor) trying to acquire the lock while it is held is small.

If these conditions are met, then the locking code can retry a failed attempt to acquire the lock until success.

If the conditions are not met, then a spinlock is not a good candidate. One alternative is to use a spinlock for critical section control (engineered to meet the conditions) to implement a higher level semaphore that can support preemption, notification, timeout or other higher level properties.

### 21.3.4 Spinlock Functional Operation

The Spinlock module supports 32 spinlocks. It accepts only a single command at a time and processes the command fully before accepting the next command. A lock is requested by reading the [SPINLOCK\\_LOCK\\_REG\\_i\[0\]](#) TAKEN bit. There are two states: Taken ([SPINLOCK\\_LOCK\\_REG\\_i\[0\]](#) TAKEN = 1) or Not Taken ([SPINLOCK\\_LOCK\\_REG\\_i\[0\]](#) TAKEN = 0).

When the status of lock *i* (where *i* = 0 to 31) is Not Taken (free), a read from the [SPINLOCK\\_LOCK\\_REG\\_i](#) register returns 0 and sets the lock to Taken (locked). When the status of lock *i* is Taken, a read returns 1 and does not change the state of the lock.

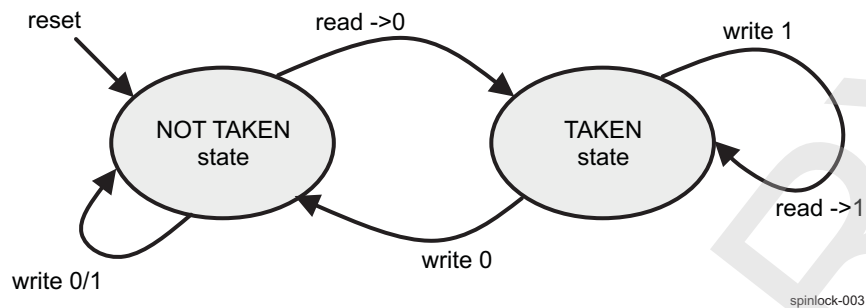
A write to the [SPINLOCK\\_LOCK\\_REG\\_i](#) register does not change the state of lock, unless when writing 0 when the lock is in Taken state. By doing this, the requester frees the lock.

#### CAUTION

Only 32-bit reads and writes are supported.

Figure 21-3 shows the [SPINLOCK\\_LOCK\\_REG\\_i](#) register state diagram.

Figure 21-3. Lock Register State Diagram

**NOTE:**

- There is no support to ensure that a lock register is locked and unlocked by the same process. This must be ensured in software.
- There is no support to check that the same initiator that acquired the lock is the one that is freeing the lock.

## 21.4 Spinlock Programming Guide

### 21.4.1 Spinlock Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

#### 21.4.1.1 Surrounding Modules Global Initialization

This procedure initializes the surrounding modules when the Spinlock module is used for the first time after a device reset.

**Table 21-4. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	Spinlock interface clock must be enabled. For more information, see <a href="#">Section 3.6.14</a> , <a href="#">CD_L4_CFG Clock Domain</a> , <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Interconnect	For more information about the L4-CFG interconnect configuration, see <a href="#">Section 13.3, L4 Interconnects</a> .

#### 21.4.1.2 Basic Spinlock Operations

The main spinlock operations are:

- Clear all the Taken spinlocks (only after a system bug recovery)
- Take a spinlock
- Release spinlock

##### 21.4.1.2.1 Spinlocks Clearing After a System Bug Recovery

Module initialization (after reset) is not needed, except after system bug recovery. The following table presents the Spinlock initialization after a system bug recovery. Software should store 0 into each of the [SPINLOCK\\_LOCK\\_REG\\_i](#) registers at system startup to insure that all locks are initialized to Not Taken.

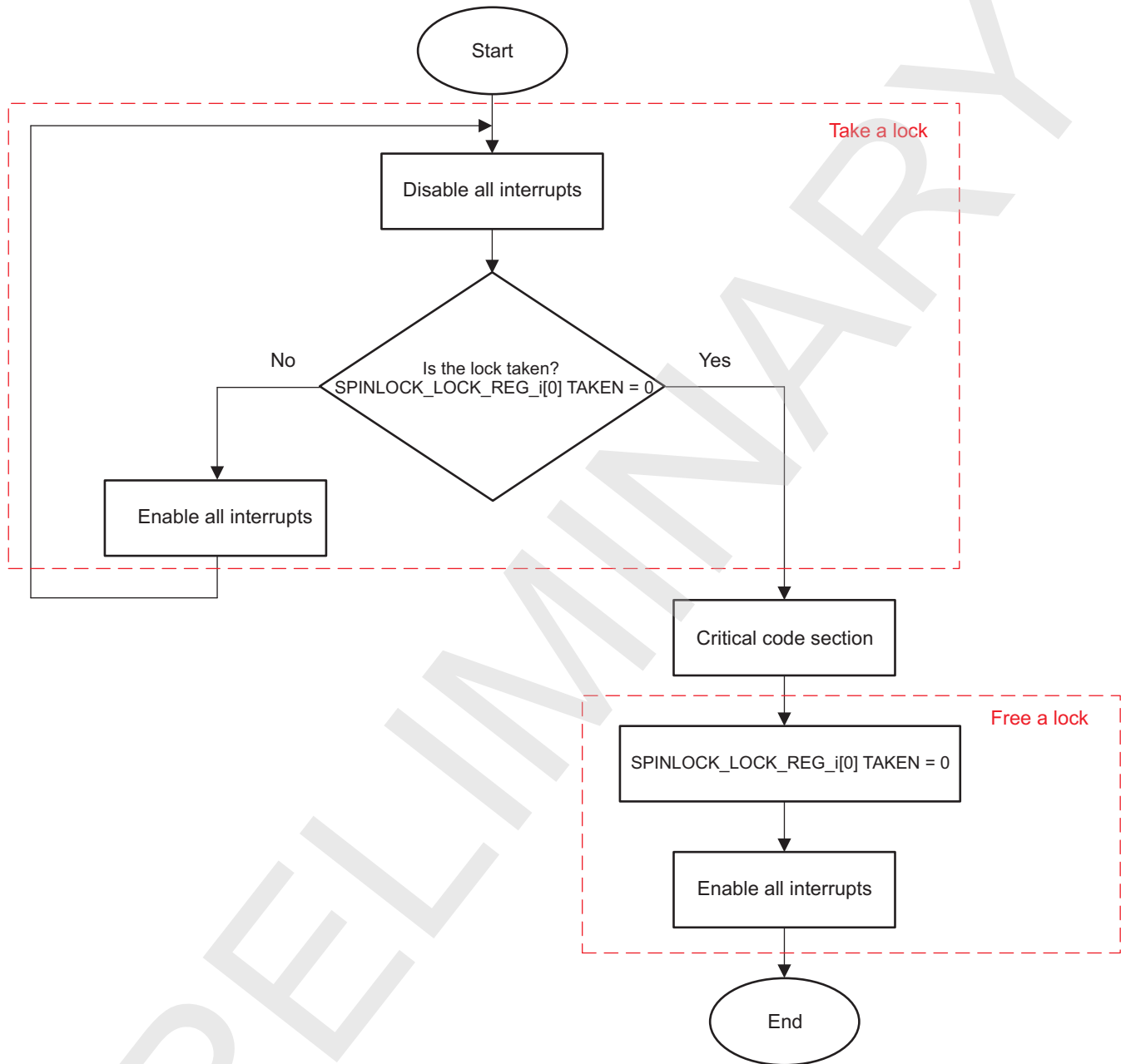
**Table 21-5. Spinlock System Bug Recovery**

Step	Register	Value
IF: <a href="#">SPINLOCK_SYSTATUS[0] IU0</a> == 1?	<a href="#">SPINLOCK_SYSTATUS[0] IU0</a>	
Free the 32 locks	<a href="#">SPINLOCK_LOCK_REG_i[0] TAKEN</a> (i=0 to 31)	0x0
END		

##### 21.4.1.2.2 Take and Release Spinlock

This procedure configures the take and release (free) operations for the Spinlock module. A spinlock should only be held with interrupts disabled. So, before attempting to obtain the spinlock, software should disable interrupts. Then it should read the [SPINLOCK\\_LOCK\\_REG\\_i\[0\] TAKEN](#) bit to attempt to obtain the lock. If it succeeds, it should proceed directly through the critical section then unlock and re-enable interrupts. If the acquisition attempt fails, the acquisition should be reattempted. To prevent unknown interrupt disabled time, interrupts should be re-enabled and then disabled before reattempting to acquire the lock. [Figure 21-4](#) shows the described above procedure.

**Figure 21-4. Take and Release Spinlock**



spinlock-004

**Table 21-6. Register Call Summary**

Register Name
SPINLOCK_LOCK_REG_i[0] TAKEN

**Table 21-7. Subprocess Call Summary**

Subprocess Name	Cross Reference
Disable Interrupts	For information about disabling interrupts in Cortex-A9 MPU INTC, see the <i>ARMCortex-A9 MPCore Technical Reference Manual</i> . For information about disabling interrupts in Cortex-M3 MPU INTC, see the <i>ARMCortex-M3 Technical Reference Manual</i> . For information about disabling interrupts in DSP INTC, see <a href="#">Chapter 5, DSP Subsystem</a>



**Table 21-7. Subprocess Call Summary (continued)**

<b>Subprocess Name</b>	<b>Cross Reference</b>
Enable Interrupts	For information about enabling interrupts in Cortex-A9 MPU INTC, see the <i>ARM Cortex-A9 MPCore Technical Reference Manual</i> . For information about enabling interrupts in Cortex-M3 MPU INTC, see the <i>ARM Cortex-M3 Technical Reference Manual</i> . For information about enabling interrupts in DSP INTC, see <a href="#">Chapter 5, DSP Subsystem</a>

## 21.5 Spinlock Register Manual

### 21.5.1 Spinlock Instance Summary

**Table 21-8. Spinlock Instance Summary**

Module Name	Base Address	Size
Spinlock	0x4A0F 6000	4K bytes

### 21.5.2 Spinlock Registers

#### 21.5.2.1 Spinlock Register Summary

**Table 21-9. Spinlock Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Spinlock L4-CFG Base Address
<a href="#">SPINLOCK_REVISION</a>	R	32	0x0000 0000	0x4A0F 6000
<a href="#">SPINLOCK_SYSCONFIG</a>	RW	32	0x0000 0010	0x4A0F 6010
<a href="#">SPINLOCK_SYSTATUS</a>	R	32	0x0000 0014	0x4A0F 6014
<a href="#">SPINLOCK_LOCK_REG_i<sup>(1)</sup></a>	RW	32	0x0000 0800 + (0x4 * i)	0x4A0F 6800 + (0x4 * i)

<sup>(1)</sup> i = 0 to 31

#### 21.5.2.2 Spinlock Register Description

**Table 21-10. SPINLOCK\_REVISION**

<b>Address Offset</b>	0x0000 0000																																																																	
<b>Physical Address</b>	See <a href="#">Table 21-9</a>	<b>Instance</b> SPINLOCK																																																																
<b>Description</b>	This register contains the IP revision code																																																																	
<b>Type</b>	R																																																																	
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> </thead> <tbody> <tr> <td colspan="16"></td> <td colspan="16">REVISION</td> </tr> </tbody> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																	REVISION															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
																REVISION																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	REVISION	IP revision	R	TI internal data																																																														

**Table 21-11. Register Call Summary for Register SPINLOCK\_REVISION**

Spinlock Register Manual

- [Spinlock Register Summary: \[0\]](#)

**Table 21-12. SPINLOCK\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	SPINLOCK
<b>Physical Address</b>	See <a href="#">Table 21-9</a>		
<b>Description</b>	This register controls the various parameters of the L4-CFG interface. Note that most fields are read-only.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLOCKACTIVITY	RESERVED		SIDLEMODE	ENWAKEUP	SOFTRESET	AUTOGATING									

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved. Reads return 0.	R	0x000000
8	CLOCKACTIVITY	Indicates whether the module requires the interface clock when in IDLE mode.  Read 0x0: Interface clock is not required by the module during IDLE mode and may be switched off. Read 0x1: Interface clock is required by the module, even during idle mode.	R	0
7:5	RESERVED	Reserved. Reads return 0.	R	0x0
4:3	SIDLEMODE	Slave interface power management (IDLE request/acknowledgement control).  Read 0x0: Force-idle. IDLE request is acknowledged unconditionally and immediately. Read 0x1: No-idle. IDLE request is never acknowledged. Read 0x2: Smart-idle. IDLE request acknowledgement is based on the internal module activity. Read 0x3: Reserved. Do not use.	R	0x2
2	ENWAKEUP	Asynchronous wakeup generation.  Read 0x0: Wakeup generation is disabled. Read 0x1: Wakeup generation is enabled.	R	0
1	SOFTRESET	Module software reset.  Write 0x0: No action Write 0x1: Start soft reset sequence	W	0
0	AUTOGATING	Internal interface clock gating strategy.  Read 0x0: Interface clock is not gated when the L4-CFG interface is idle. Read 0x1: Automatic internal interface clock gating strategy is applied, based on the L4-CFG interface activity.	R	1

**Table 21-13. Register Call Summary for Register SPINLOCK\_SYSCONFIG**

- Spinlock Functional Description
- [Spinlock Software Reset: \[0\]](#)
  - [Spinlock Power Management: \[1\] \[2\] \[3\] \[4\]](#)
- Spinlock Register Manual
- [Spinlock Register Summary: \[5\]](#)

**Table 21-14. SPINLOCK\_SYSTATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	SPINLOCK
<b>Physical Address</b>	See <a href="#">Table 21-9</a>		
<b>Description</b>	This register provides status information about this instance of the Spinlock module.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
NUMLOCKS								RESERVED								IU7	IU6	IU5	IU4	IU3	IU2	IU1	IU0	RESERVED								RESETDONE

Bits	Field Name	Description	Type	Reset
31:24	NUMLOCKS	Number of lock registers implemented. Read 0x1: This instance has 32 lock registers. Read 0x2: This instance has 64 lock registers. Read 0x4: This instance has 128 lock registers. Read 0x8: This instance has 256 lock registers.	R	0x01
23:16	RESERVED	Reserved. Reads return 0.	R	0x00
15	IU7	In-Use flag 7. Reads always return 0.	R	0
14	IU6	In-Use flag 6. Reads always return 0.	R	0
13	IU5	In-Use flag 5. Reads always return 0.	R	0
12	IU4	In-Use flag 4. Reads always return 0.	R	0
11	IU3	In-Use flag 3. Reads always return 0.	R	0
10	IU2	In-Use flag 2. Reads always return 0.	R	0
9	IU1	In-Use flag 1. Reads always return 0.	R	0
8	IU0	In-Use flag 0, covering lock registers 0 - 31. Read 0x0: All lock registers 0 31 are in the Not Taken state. Read 0x1: At least one of the lock registers 0 31 is in the Taken state.	R	0
7:1	RESERVED	Reserved. Reads return 0.	R	0x00
0	RESETDONE	Reset done status. Read 0x0: Reset in progress. Read 0x1: Reset is completed.	R	1

**Table 21-15. Register Call Summary for Register SPINLOCK\_SYSTATUS**

## Spinlock Functional Description

- [Spinlock Software Reset: \[0\]](#)
- [Spinlock Power Management: \[1\]](#)

## Spinlock Programming Guide

- [Spinlocks Clearing After a System Bug Recovery: \[2\] \[3\]](#)

## Spinlock Register Manual

- [Spinlock Register Summary: \[4\]](#)

**Table 21-16. SPINLOCK\_LOCK\_REG\_i**

<b>Address Offset</b>	0x0000 0800	<b>index</b>	i = 0 to 31
<b>Physical Address</b>	See <a href="#">Table 21-9</a>	<b>Instance</b>	SPINLOCK
<b>Description</b>	This register contains the state of one lock.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	TAKEN														

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved. Reads return 0. Writes are ignored.	R	0x0000 0000
0	TAKEN	<p>Lock State</p> <p>Read 0x0: Lock was previously Not Taken (free). The requester is granted the lock.</p> <p>Write 0x0: Set the lock to Not Taken (free).</p> <p>Read 0x1: Lock was previously Taken. The requester is not granted the lock and must retry.</p> <p>Write 0x1: No update to the lock value.</p>	RW	0

**Table 21-17. Register Call Summary for Register SPINLOCK\_LOCK\_REG\_i**

Spinlock Functional Description

- [Spinlock Functional Operation: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)

Spinlock Programming Guide

- [Spinlocks Clearing After a System Bug Recovery: \[6\] \[7\]](#)
- [Take and Release Spinlock: \[8\] \[9\]](#)

Spinlock Register Manual

- [Spinlock Register Summary: \[10\]](#)

PRELIMINARY

# Timers

---

---

---

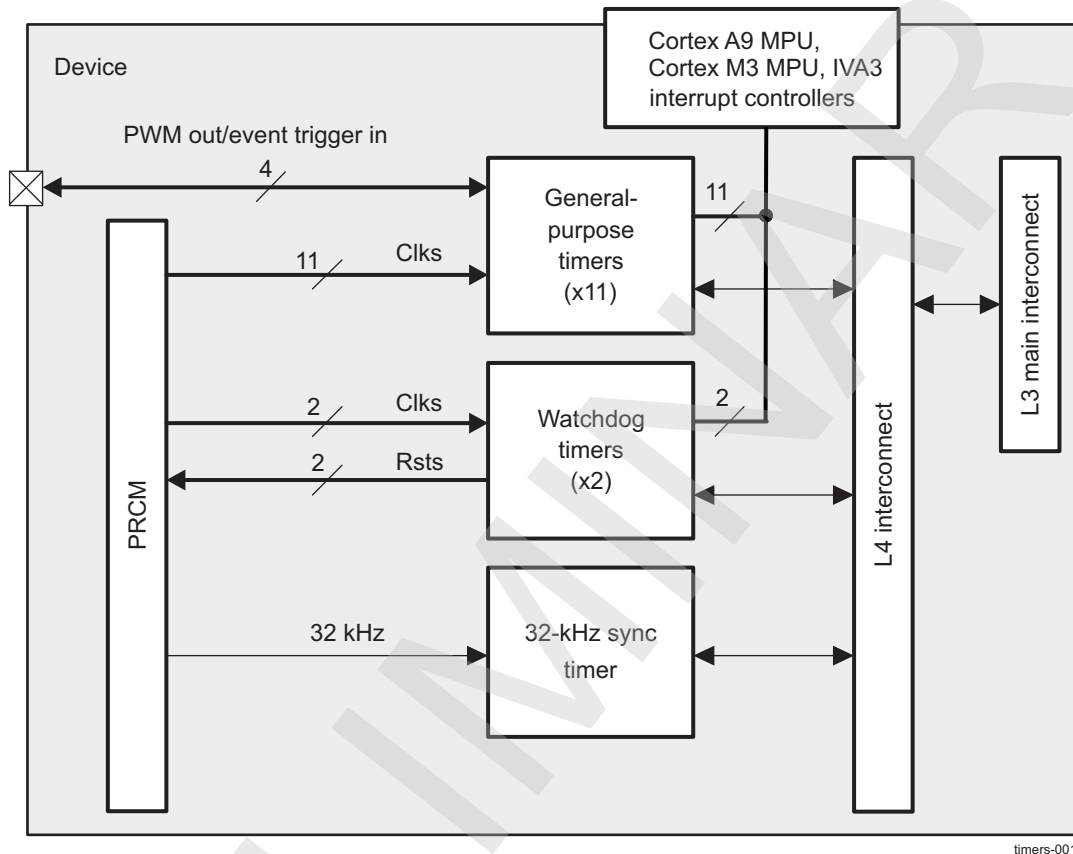
This chapter describes the timer modules for the device.

Topic	Page
22.1 Timers Overview .....	3960
22.2 General-Purpose Timers .....	3961
22.3 Watchdog Timers .....	4011
22.4 32-kHz Synchronized Timer .....	4037

## 22.1 Timers Overview

The device includes several types of timers used by the system software, including 11 general-purpose (GP) timers, two watchdog timers, and a 32-kHz synchronized timer (sync timer). Figure 22-1 is a high-level block diagram of the device timers.

**Figure 22-1. Timers**



The two watchdog timers are clocked with 32-kHz clocks. The 32-kHz sync timer, which is reset only at power up, provides the operating system (OS) with a stable timing source that stores the relative time since the last power cycle of the product. Finally, 11 GP timers, which are useful simply as basic timers, are included to generate time-stamp-based interrupts to the system software or to use as a source of pulse-width modulation (PWM) signals.



## 22.2 General-Purpose Timers

### 22.2.1 General-Purpose Timers Overview

The device has 11 general-purpose (GP) timers: GPTIMER1 through GPTIMER11.

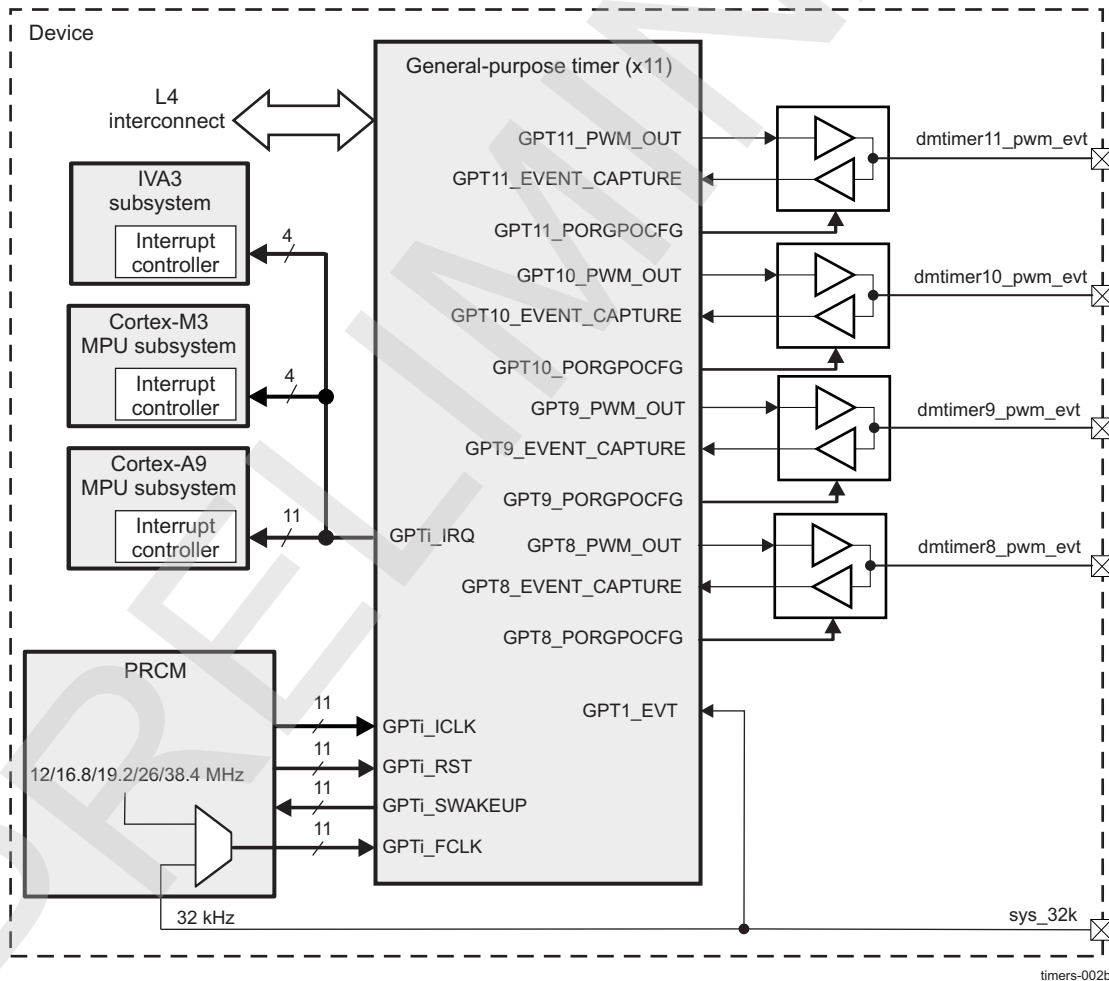
Each timer can be clocked from either the system clock (12, 16.8, 19.2, 26, or 38.4 MHz) or the 32-kHz clock. The selection of clock source is made at the power, reset, and clock management (PRCM) module level. For more information, see [Section 3.6.3.1, PRM Clock Source](#).

GPTIMER1 has its GPT1\_EVENT\_CAPTURE pin tied to the 32-kHz clock and can be used to gauge the system clock input; it detects its frequency among 12, 16.8, 19.2, 26, or 38.4 MHz.

Each timer can provide an interrupt to the microprocessor unit (MPU) subsystem. In addition, GPTIMER5 through GPTIMER8 also have interrupts connected to the IVA3 subsystem.

GPTIMER1, GPTIMER2, and GPTIMER10 include specific functions to generate accurate tick interrupts to the operating system. GPTIMER8 through GPTIMER11 are connected to external pins by their PWM output or their event capture input pin (for external timer triggering). [Figure 22-2](#) is an overview of the GP timers.

Figure 22-2. GP Timers Overview



#### 22.2.1.1 GP Timer Features

The following are the main features of the GP timer controllers:

- L4 slave interface support:

- 32-bit data bus width
- 32-/16-bit access supported
- 8-bit access not supported
- 10-bit address bus width
- Burst mode not supported
- Write nonposted transaction mode supported
- Interrupts generated on overflow, compare, and capture
- Free-running 32-bit upward counter
- Compare and capture modes
- Autoreload mode
- Start/stop mode
- Programmable divider clock source ( $2^n$  where  $n = [0:8]$ )
- Dedicated input trigger for capture mode and dedicated output trigger/PWM signal
- Dedicated GP output signal for using the GPTi\_GPOCFG signal
- On-the-fly read/write register (while counting)
- 1-ms tick with 32,768-Hz functional clock generated (only GPTIMER1, GPTIMER2, and GPTIMER10)

## 22.2.2 GP Timer Environment

### 22.2.2.1 GP Timer External System Interface

Four of the 11 GP timers can send or receive stimulus to/from the external (off-chip) system. In the device; however, only GPTIMER8 through GPTIMER11 are configured to output a pulse-width modulation (PWM) pulse or receive an external event signal used as a trigger to capture the current timer count. GPTIMER1 is also configured to receive an event trigger input (GPT1\_EVENT\_CAPTURE) tied to the internal 32-kHz clock. This event signal gauges the system clock input; detecting its frequency among 12, 16.8, 19.2, 26, or 38.4 MHz.

Figure 22-3 shows the external system interface for the GP timers, and Table 22-1 describes the GP timer inputs and outputs.

**NOTE:** Software control must ensure that MUX mode is configured to select the dmtimerx\_pwm\_evt (where x = 8 to 11) signal on only one pad. Other pads on which the same signal is multiplexed must be configured in safe mode or non-dmtimer mode to avoid two different pads driving the same signal.

For more information about the configuration of the dmtimer8\_pwm\_evt through dmtimer11\_pwm\_evt I/O pads, see Section 18.4.8, *PAD Functional Multiplexing and Configuration*.

Figure 22-3. GP Timers External System Interface

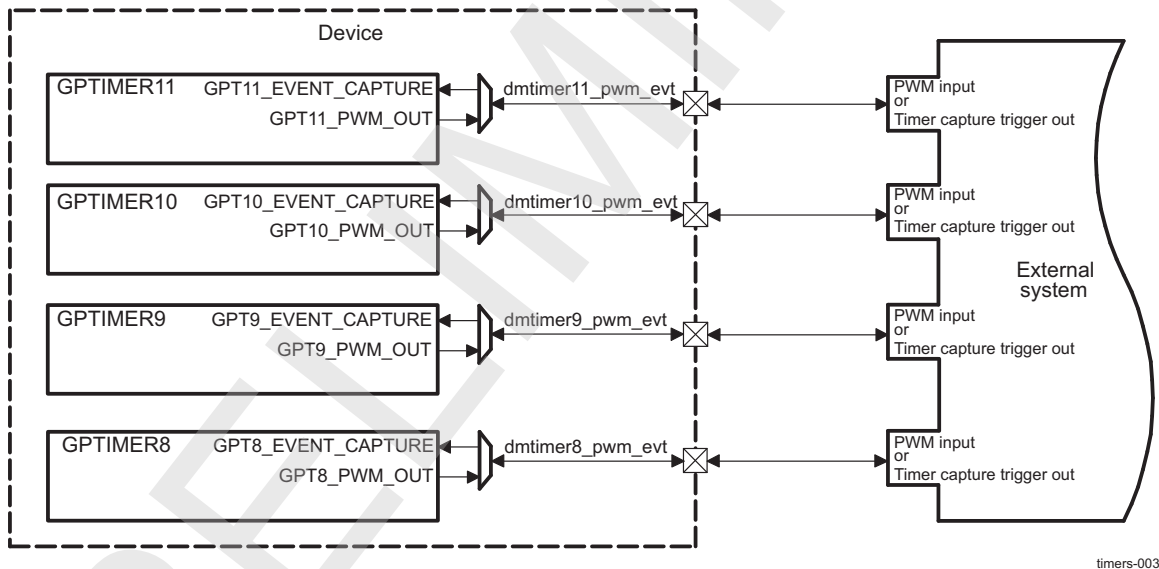


Table 22-1. Input/Output Description

Pin Name	Type <sup>(1)</sup>	Reset Value	Signal Name	Description
dmtimer8_pwm_evt	I/O	0	GPT8_EVENT_CAPTURE GPT8_PWM_OUT	GPTIMER8 trigger input/ PWM output
dmtimer9_pwm_evt	I/O	0	GPT9_EVENT_CAPTURE GPT9_PWM_OUT	GPTIMER9 trigger input/ PWM output
dmtimer10_pwm_evt	I/O	0	GPT10_EVENT_CAPTURE GPT10_PWM_OUT	GPTIMER10 trigger input/ PWM output
dmtimer11_pwm_evt	I/O	0	GPT11_EVENT_CAPTURE GPT11_PWM_OUT	GPTIMER11 trigger input/ PWM output

<sup>(1)</sup> When configured for that function; I = input, O = output

---

**NOTE:** The event trigger input (GPTi\_EVENT\_CAPTURE) for GPTIMER2 through GPTIMER7 is internally tied low, and the PWM output (GPTi\_PWM\_OUT) is not connected.

---

**NOTE:** For GPTIMER8 through GPTIMER11, the GPTi\_PORGPOCFG signal selects to connect the GPT pwm output or the GPT capture input to the dmtimerx\_pwm\_evt (where x = 8 to 11) pad at the top level. When the TCLR[14] GPO\_CFG bit = 0b1, the dmtimerx\_pwm\_evt functions as a capture input. When the TCLR[14] GPO\_CFG bit = 0b0, the dmtimerx\_pwm\_evt functions as a pwm output (GPO\_CFG = 0b0).

---

### 22.2.3 GP Timer Integration

Figure 22-4 shows the integration of the GP timer in the device.

Figure 22-4. GP Timer Integration

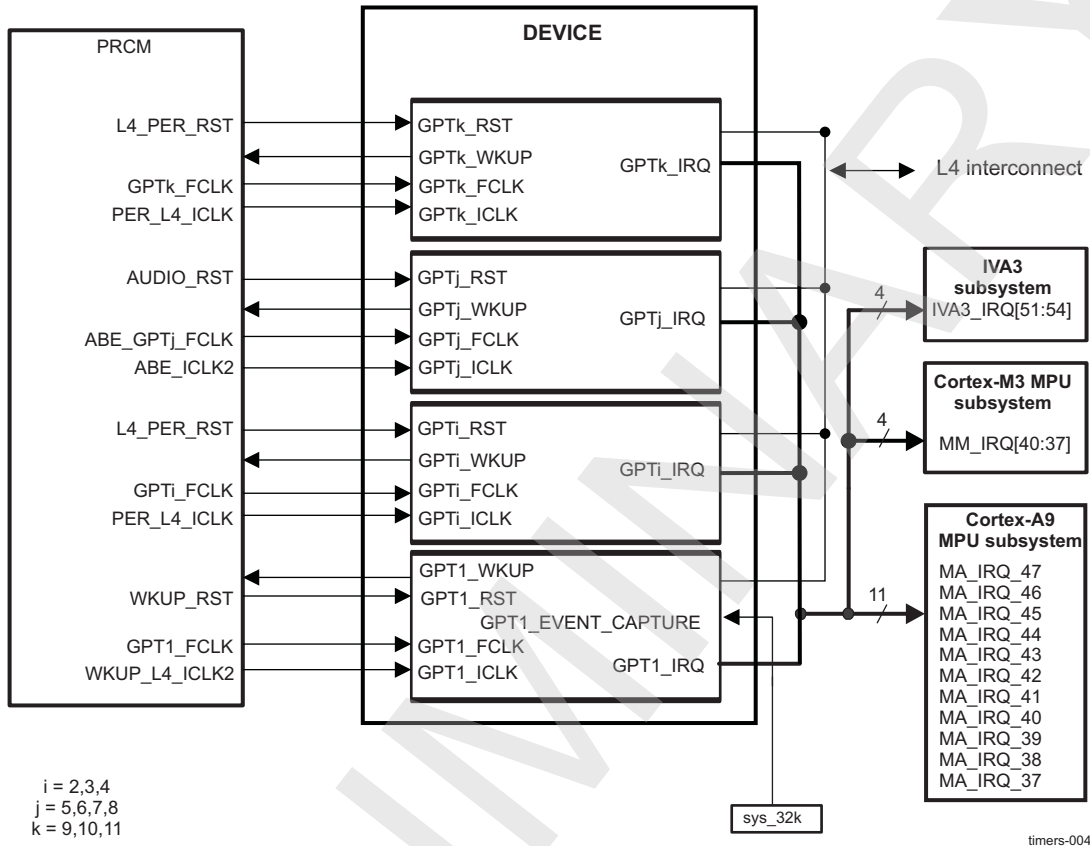


Table 22-2 through Table 22-4 summarize the integration of the module in the device.

Table 22-2. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
GPTIMER1	PD_WKUP	Yes	L4_WKUP
GPTIMER2	PD_L4_PER	Yes	L4_PER
GPTIMER3	PD_L4_PER	Yes	L4_PER
GPTIMER4	PD_L4_PER	Yes	L4_PER
GPTIMER5	PD_AUDIO	Yes	L4_ABE
GPTIMER6	PD_AUDIO	Yes	L4_ABE
GPTIMER7	PD_AUDIO	Yes	L4_ABE
GPTIMER8	PD_AUDIO	Yes	L4_ABE
GPTIMER9	PD_L4_PER	Yes	L4_PER
GPTIMER10	PD_L4_PER	Yes	L4_PER
GPTIMER11	PD_L4_PER	Yes	L4_PER

Table 22-3. Clocks and Resets

Module Instance	Clocks			
	Destination Signal Name	Source Signal Name	Source	Description

**Table 22-3. Clocks and Resets (continued)**

GPTIMER1	GPT1_FCLK	GPT1_FCLK	PRCM	GPTIMER1 functional clock
GPTIMER1	GPT1_ICLK	WKUP_L4_ICLK2	PRCM	GPTIMER1 interface Clock
GPTIMER2	GPT2_FCLK	GPT2_FCLK	PRCM	GPTIMER2 functional clock
GPTIMER2	GPT2_ICLK	PER_L4_ICLK	PRCM	GPTIMER2 interface Clock
GPTIMER3	GPT3_FCLK	GPT3_FCLK	PRCM	GPTIMER3 functional clock
GPTIMER3	GPT3_ICLK	PER_L4_ICLK	PRCM	GPTIMER3 interface Clock
GPTIMER4	GPT4_FCLK	GPT4_FCLK	PRCM	GPTIMER4 functional clock
GPTIMER4	GPT4_ICLK	PER_L4_ICLK	PRCM	GPTIMER4 interface Clock
GPTIMER5	GPT5_FCLK	ABE_GPT5_FCLK	PRCM	GPTIMER5 functional clock
GPTIMER5	GPT5_ICLK	ABE_ICLK2	PRCM	GPTIMER5 interface Clock
GPTIMER6	GPT6_FCLK	ABE_GPT6_FCLK	PRCM	GPTIMER6 functional clock
GPTIMER6	GPT6_ICLK	ABE_ICLK2	PRCM	GPTIMER6 interface Clock
GPTIMER7	GPT7_FCLK	ABE_GPT7_FCLK	PRCM	GPTIMER7 functional clock
GPTIMER7	GPT7_ICLK	ABE_ICLK2	PRCM	GPTIMER7 interface Clock
GPTIMER8	GPT8_FCLK	ABE_GPT8_FCLK	PRCM	GPTIMER8 functional clock
GPTIMER8	GPT8_ICLK	ABE_ICLK2	PRCM	GPTIMER8 interface Clock
GPTIMER9	GPT9_FCLK	GPT9_FCLK	PRCM	GPTIMER9 functional clock
GPTIMER9	GPT9_ICLK	PER_L4_ICLK	PRCM	GPTIMER9 interface Clock
GPTIMER10	GPT10_FCLK	GPT10_FCLK	PRCM	GPTIMER10 functional clock
GPTIMER10	GPT10_ICLK	PER_L4_ICLK	PRCM	GPTIMER10 interface Clock
GPTIMER11	GPT11_FCLK	GPT11_FCLK	PRCM	GPTIMER11 functional clock
GPTIMER11	GPT11_ICLK	PER_L4_ICLK	PRCM	GPTIMER11 interface Clock
<b>Resets</b>				
GPTIMER1	GPT1_RST	WKUP_RST	PRM	Reset to GPTIMER1
GPTIMER2	GPT2_RST	L4_PER_RST	PRM	Reset to GPTIMER2
GPTIMER3	GPT3_RST	L4_PER_RST	PRM	Reset to GPTIMER3
GPTIMER4	GPT4_RST	L4_PER_RST	PRM	Reset to GPTIMER4
GPTIMER5	GPT5_RST	AUDIO_RST	PRM	Reset to GPTIMER5
GPTIMER6	GPT6_RST	AUDIO_RST	PRM	Reset to GPTIMER6
GPTIMER7	GPT7_RST	AUDIO_RST	PRM	Reset to GPTIMER7
GPTIMER8	GPT8_RST	AUDIO_RST	PRM	Reset to GPTIMER8
GPTIMER9	GPT9_RST	L4_PER_RST	PRM	Reset to GPTIMER9
GPTIMER10	GPT10_RST	L4_PER_RST	PRM	Reset to GPTIMER10
GPTIMER11	GPT11_RST	L4_PER_RST	PRM	Reset to GPTIMER11

**Table 22-4. Hardware Requests**

Module Instance	Source Signal Name	Destination Signal Name	Interrupt Requests	
			Destination	Description
GPTIMER1	GPT1_IRQ	MA_IRQ_37	Cortex™-A9 MPU	GPTIMER1 interrupt to Cortex-A9 MPU
GPTIMER2	GPT2_IRQ	MA_IRQ_38	Cortex-A9™ MPU	GPTIMER2 interrupt to Cortex-A9 MPU
GPTIMER3	GPT3_IRQ	MA_IRQ_39	Cortex-A9 MPU	GPTIMER3 interrupt to Cortex-A9 MPU
		MM_IRQ_37	Cortex-M3™ MPU	GPTIMER3 interrupt to Cortex-M3 MPU
GPTIMER4	GPT4_IRQ	MA_IRQ_40	Cortex-A9 MPU	GPTIMER3 interrupt to Cortex-A9 MPU
		MM_IRQ_38	Cortex-M3 MPU	GPTIMER4 interrupt to Cortex-M3 MPU
GPTIMER5	GPT5_IRQ	D_IRQ_51	DSP	GPTIMER5 interrupt to DSP
		MA_IRQ_41	Cortex-A9 MPU	GPTIMER5 interrupt to Cortex-A9 MPU
GPTIMER6	GPT6_IRQ	D_IRQ_52	DSP	GPTIMER6 interrupt to DSP

**Table 22-4. Hardware Requests (continued)**

	GPT6_IRQ	MA_IRQ_42	Cortex-A9 MPU	GPTIMER6 interrupt to Cortex-A9 MPU
GPTIMER7	GPT7_IRQ	D_IRQ_53	DSP	GPTIMER7 interrupt to DSP
	GPT7_IRQ	MA_IRQ_43	Cortex-A9 MPU	GPTIMER7 interrupt to Cortex-A9 MPU
GPTIMER8	GPT8_IRQ	D_IRQ_54	DSP	GPTIMER8 interrupt to DSP
	GPT8_IRQ	MA_IRQ_44	Cortex-A9 MPU	GPTIMER8 interrupt to Cortex-A9 MPU
GPTIMER9	GPT9_IRQ	MA_IRQ_45	Cortex-A9 MPU	GPTIMER3 interrupt to Cortex-A9 MPU
	GPT9_IRQ	MM_IRQ_39	Cortex-M3 MPU	GPTIMER9 interrupt to Cortex-M3 MPU
GPTIMER10	GPT10_IRQ	MA_IRQ_46	Cortex-A9 MPU	GPTIMER10 interrupt to Cortex-A9 MPU
GPTIMER11	GPT11_IRQ	MA_IRQ_47	Cortex-A9 MPU	GPTIMER3 interrupt to Cortex-A9 MPU
	GPT11_IRQ	MM_IRQ_40	Cortex-M3 MPU	GPTIMER11 interrupt to Cortex-M3 MPU
<b>No DMA Requests</b>				

**NOTE:** For the description of the interrupt source description, see [Section 22.2.4.5, GP Timer Interrupts](#).

## 22.2.4 GP Timer Functional Description

Each GP timer contains a free-running upward counter with autoreload capability on overflow. The timer counter can be read and written on-the-fly (while counting). Each GP timer includes compare logic to allow an interrupt event on a programmable counter matching value. A dedicated output signal can be pulsed or toggled on either an overflow or a match event. This offers time-stamp trigger signaling or PWM signal sources. A dedicated input signal can be used to trigger an automatic timer counter capture or an interrupt event on a programmable input signal transition. A programmable clock divider (prescaler) allows reduction of the timer input clock frequency. All internal timer interrupt sources are merged into one module interrupt line and one wake-up line.

Each internal interrupt source can be independently enabled/disabled with a dedicated bit of the [GPT\\_TIER](#) register for the interrupt features, and a dedicated bit of the [GPT\\_TWER](#) register for the wake-up of GPTIMER1, GPTIMER2, and GPTIMER10. In addition, these timers have a mechanism implemented to generate an accurate tick interrupt.

For the other GP Timers, each internal interrupt source can be independently enabled/disabled through the [GPT\\_IRQENABLE\\_SET](#) and [GPT\\_IRQENABLE\\_CLR](#) register bit fields.

For each GP timer implemented in the device, there are two possible clock sources:

- 32-kHz clock
- System clock

Selection of the input clock source is done in the registers in the PRCM configuration (see [Section 22.2.1, GP Timer Overview](#)).

Each GP timer supports three functional modes:

- Timer mode
- Capture mode
- Compare mode

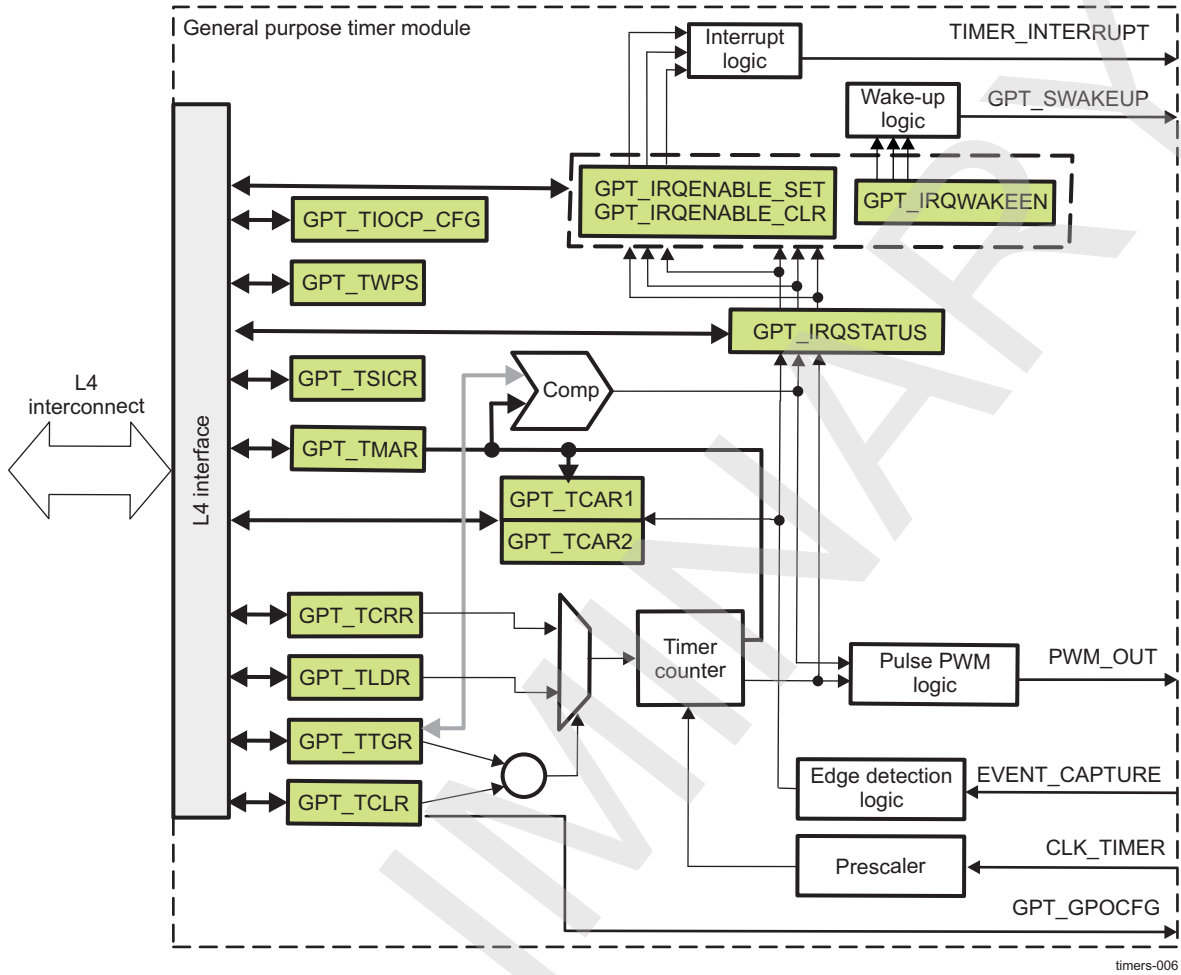
The capture and compare modes are disabled by default after core reset.

### 22.2.4.1 GP Timer Block Diagram

[Figure 22-5](#) is a block diagram of the common GP timers, and [Figure 22-6](#) is a block diagram of the GP timers with 1-ms tick generation module.

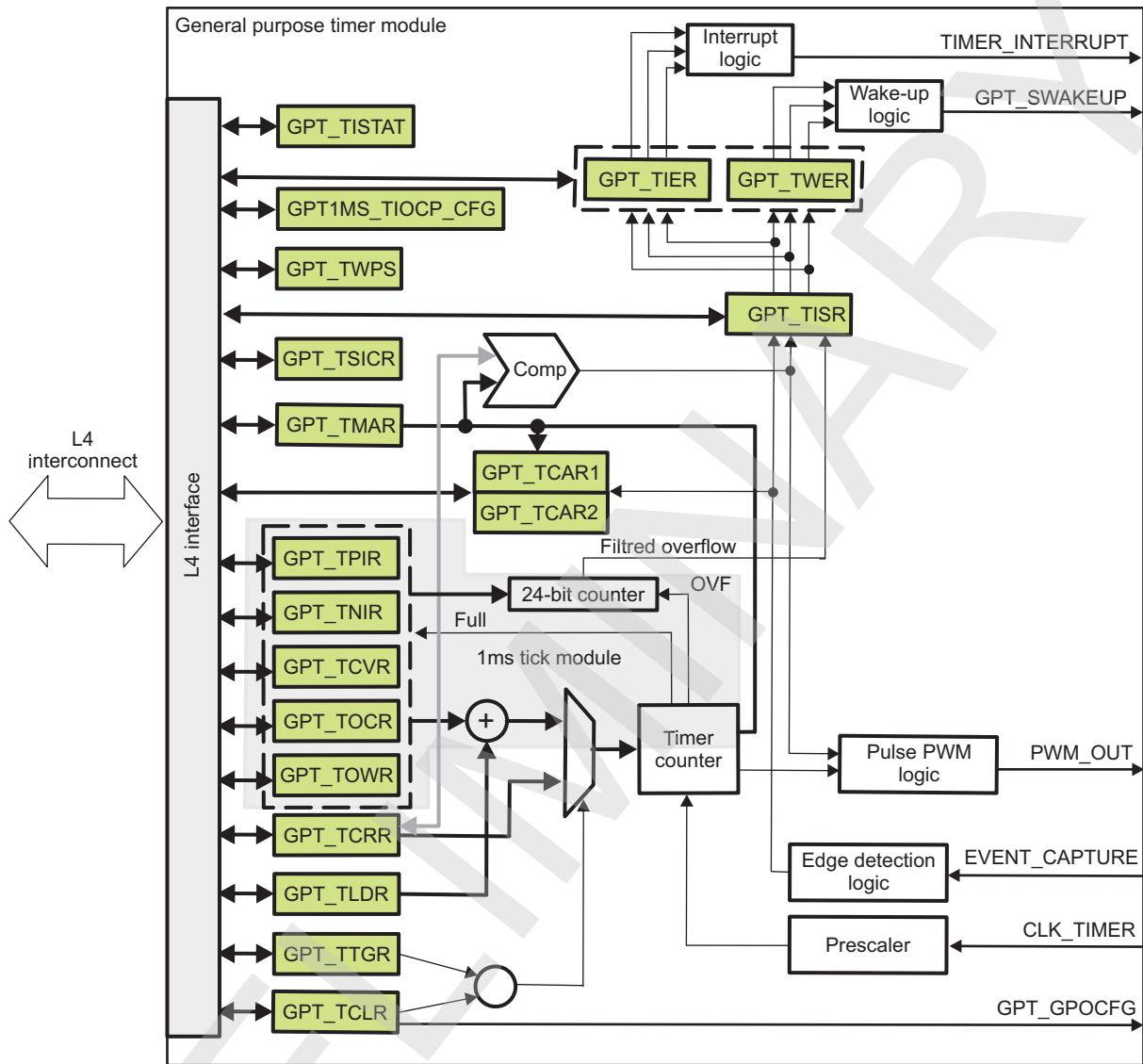


Figure 22-5. Block Diagram of GPTIMER3 Through GPTIMER9, and GPTIMER11



timers-006

Figure 22-6. Block Diagram of GPTIMER1, GPTIMER2, and GPTIMER10



timers-007

### 22.2.4.2 GPTIMER1, GPTIMER2, and GPTIMER10 Power Management

The GP timer modules have an internal bit, `GPT1MS_TIOCP_CFG[0]` AUTOIDLE, that is used to apply an internal interface clock gating strategy.

At the PRCM module level, when all conditions to shut off the PRCM module functional or interface output clocks are met (see [Section 3.1.1.1.4, Clock Domain Level Clock Management](#)), the PRCM module automatically launches a hardware handshake protocol to ensure the GP timer is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the GP timer.

Although this handshake is a hardware function and is out of software control, the way the GP timer acknowledges the PRCM IDLE request is configurable through the `GPT1MS_TIOCP_CFG[4:3]` IDLEMODE bit field.

[Table 22-7](#) lists the IDLEMODE settings and the related acknowledgment modes.

**Table 22-5. IDLEMODE Settings**

IDLEMODE Value	Selected Mode	Description
00	Force-idle	The GP timer unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully, because it does not prevent the loss of data when the clock is switched off.
01	No-idle	The GP timer never acknowledges an IDLE request from the PRCM module. This mode is safe from a module point of view, because it ensures that the clocks remain active. It is not efficient from a power-saving perspective, because it does not allow the PRCM output clock to be shut off and thus the power domain to be set to a lower power state.
10	Smart-idle	The GP timer acknowledges the IDLE request, basing its decision on its internal activity. The acknowledge signal is asserted only when all pending transactions and IRQ requests are treated. This is the best approach to efficient system power management.
11	Reserved	

When configured in smart-idle mode, the GP timer also offers an additional granularity on `GPTi_FCLK` and `GPTi_ICLK` gating. The `GPT1MS_TIOCP_CFG[9:8]` CLOCKACTIVITY bit field is used to determine which clock is shut down (`GPTi_FCLK`, `GPTi_ICLK`, neither, or both clocks).

The CLOCKACTIVITY setting is used internally to the GP timer to determine the part of the module on which the conditions to acknowledge the PRCM IDLE request are tested. For example, if `GPTi_FCLK` is not to be shut down on a PRCM IDLE request, the GP timer considers only `GPTi_ICLK` and associated pending activities before acknowledging the request.

Some GP timer features are associated with `GPTi_ICLK` and others are associated with `GPTi_FCLK`. Using the CLOCKACTIVITY setting along with the smart-idle mode ensures that the features associated with the clock that remains active are always enabled, even if the GP timer acknowledges an IDLE request.

[Table 22-6](#) lists the CLOCKACTIVITY settings and associated features.

**Table 22-6. CLOCKACTIVITY Settings**

CLOCKACTIVITY Value	GPTi_ICLK Effects	GPTi_FCLK Effects	Description	Associated Features
00	OFF	OFF	<code>GPTi_ICLK</code> and <code>GPTi_FCLK</code> are considered for generating the acknowledge. This setting also means <code>GPTi_FCLK</code> and <code>GPTi_ICLK</code> are likely to be shut down on PRCM IDLE request.	The idle acknowledge signal is asserted when there are no pending activities on the functional clock domain (improved latency in assertion of idle acknowledge). The wake-up capability of the GP timer is disabled.
01	ON	OFF	<code>GPTi_ICLK</code> is not shut down on PRCM IDLE request; only <code>GPTi_FCLK</code> is affected.	

**Table 22-6. CLOCKACTIVITY Settings (continued)**

CLOCKACTIVITY Value	GPTi_ICLK Effects	GPTi_FCLK Effects	Description	Associated Features
10	OFF	ON	GPTi_FCLK is not shut down on PRCM IDLE request; only GPTi_ICLK is affected.	The idle acknowledge signal is asserted when there are no pending activities on the interface clock domain, without evaluating the pending activities on the functional clock domain (the GP timer enters into sleep mode, and if a pending interrupt event is completed during idle mode, the wake-up signal is asserted). The wake-up signal is enabled.
11	ON	ON	None of the clocks are shut down. Therefore, the GP timer can potentially acknowledge the IDLE request without checking the internal functionalities linked to its clocks.	

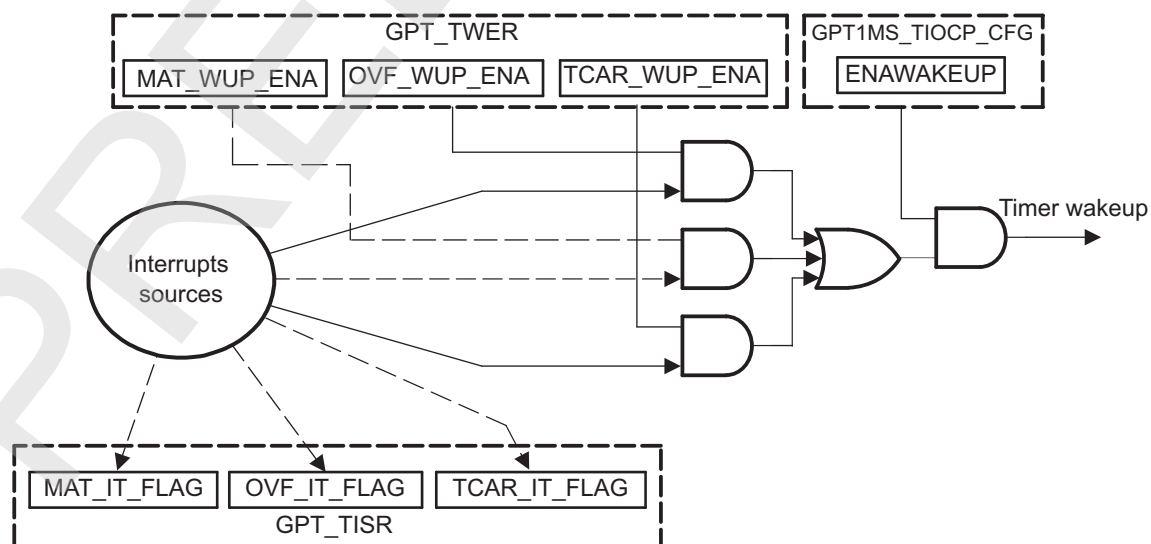
**CAUTION**

The PRCM module does not have any hardware means to read the CLOCKACTIVITY settings. The software must ensure consistent programming between the GP timer CLOCKACTIVITY and the PRCM module functional clock and interface clock control bits. If the GP timer is disabled in the CM\_FCLKEN and CM\_ICLKEN PRCM registers while CLOCKACTIVITY is set to 11, nothing prevents the PRCM module from asserting its IDLE request, which is acknowledged regardless of the features associated with the GP timer clocks. This can lead to unpredictable behavior.

**22.2.4.2.1 Wake-Up Capability**

If the `GPT1MS_TIOCP_CFG[4:3]` IDLEMODE bit field sets the smart-idle mode, the timer evaluates its internal capability to have the interface clock switched off. When there is no further internal activity (no pending interrupt sources: match, overflow, or timer capture events), the idle acknowledge signal is asserted and the timer enters sleep mode, ready to issue a wake-up request. This wake-up request is sent only if the `GPT1MS_TIOCP_CFG[2]` ENAWAKEUP bit enables the timer wake-up capability.

Figure 22-7 shows the wake-up request generation. For more information on the GP timer clock control, see Section 3.6, *Clock Management Functional Description* in Chapter 3, *Power, Reset and Clock Management*.

**Figure 22-7. Wake-Up Request Generation**

timers-005

For GPTIMER1, GPTIMER2, and GPTIMER10, the timer wake-up enable-register (**GPT\_TWVER**) allows masking the expected source of the wake-up event that generates a wake-up request. The **GPT\_TWVER** register is synchronously programmed with the interface clock before the PRCM module sends an idle mode request. The expected source of the wake-up event is an overflow (**GPT\_TCRR**), a timer match (the compare result of **GPT\_TCRR** and **GPT\_TMAR** matches the counter value), and a timer capture (detection of an external pulse transition of the correct polarity on the **GPT\_EVENT\_CAPTURE**).

When the wake-up event is issued, the associated interrupt status bit is set in the timer status register (**GPT\_TISR**). The pending wake-up event is reset when the set status bit is overwritten with 1.

**NOTE:** The status bit must be reset to re-enter idle mode.

### 22.2.4.3 Power Management of Other GP Timers

At the PRCM module level, when all conditions to shut off the PRCM module functional or interface output clocks are met (see [Section 3.1.1.1.4, Clock Domain Level Clock Management](#)), the PRCM module automatically launches a hardware handshake protocol to ensure the GP timer is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the GP timer.

Although this handshake is a hardware function and is out of software control, the way the GP timer acknowledges the PRCM IDLE request is configurable through the **GPT\_TIOCP\_CFG[3:2]** IDLEMODE bit field.

[Table 22-7](#) lists the IDLEMODE settings and the related acknowledgment modes.

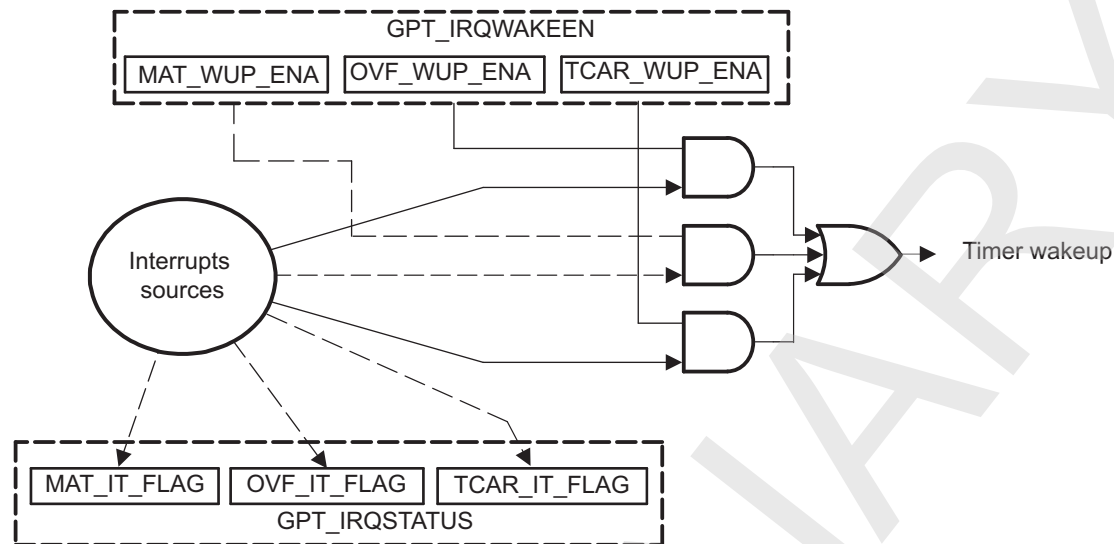
**Table 22-7. IDLEMODE Settings**

IDLEMODE Value	Selected Mode	Description
00	Force-idle	The GP timer unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully, because it does not prevent the loss of data when the clock is switched off.
01	No-idle	The GP timer never acknowledges an IDLE request from the PRCM module. This mode is safe from a module point of view, because it ensures that the clocks remain active. It is not efficient from a power-saving perspective, however, because it does not allow the PRCM output clock to be shut off and thus the power domain to be set to a lower power state.
10	Smart-idle	The GP timer acknowledges the IDLE request, basing its decision on its internal activity. The acknowledge signal is asserted only when all pending transactions and IRQ requests are treated. This is the best approach to efficient system power management.
11	Reserved	

#### 22.2.4.3.1 Wake-Up Capability

If the **GPT\_TIOCP\_CFG[3:2]** IDLEMODE bit field sets the smart-idle mode, the timer evaluates its internal capability to have the interface clock switched off. When there is no further internal activity (no pending interrupt sources: match, overflow, or timer capture events), the idle acknowledge signal is asserted and the timer enters sleep mode, ready to issue a wake-up request. This wake-up request is sent only if the **GPT\_IRQWAKEEN[2:0]** bit field enables the timer wake-up capability.

[Figure 22-8](#) shows the wake-up request generation. For more information on the GP timer clock control, see [Section 3.6, Clock Management Functional Description](#) in [Chapter 3, Power, Reset and Clock Management](#).

**Figure 22-8. Wake-Up Request Generation**

timers-014

When the wake-up event is issued, the associated interrupt status bit is set in the timer status register ([GPT\\_IRQSTATUS](#)). The pending wake-up event is reset when the set status bit is overwritten with 1.

---

**NOTE:** The status bit must be reset to re-enter idle mode.

---

#### 22.2.4.4 Software Reset

Two bit fields can generate a software reset of the GP timer:

- [GPT1MS\\_TIOCP\\_CFG\[1\]](#) SOFTRESET for GPTIMER1, GPTIMER2, and GPTIMER10, and [GPT\\_TIOCP\\_CFG\[0\]](#) SOFTRESET for other timers
- [GPT\\_TSICR\[1\]](#) SFT for all timers

For both bits, all read accesses return 0.

The [GPT1MS\\_TIOCP\\_CFG\[1\]](#) SOFTRESET/[GPT\\_TIOCP\\_CFG\[0\]](#) SOFTRESET bit allows resetting the functional and interface domains. The [GPT\\_TSICR\[1\]](#) SFT bit allows resetting the functional part of the GP timer.

Before accessing or using the GP timer, the local host must ensure that both internal resets are released by reading the [GPT\\_TISTAT\[0\]](#) RESETDONE (for GPTIMER1, GPTIMER2, and GPTIMER10), or the [GPT\\_TIOCP\\_CFG\[0\]](#) SOFTRESET bit (for other timers). This bit monitors the internal reset status.

#### 22.2.4.5 GP Timer Interrupts

The timer can issue an overflow interrupt, a timer match interrupt, and a timer capture interrupt. Each internal interrupt source can be independently enabled/disabled in the interrupt-enable register ([GPT\\_TIER](#) for GPTIMER1, GPTIMER2, and GPTIMER10), or, for other timers, enable in the interrupt-enable register ([GPT\\_IRQENABLE\\_SET](#)) and disable in the interrupt disable register ([GPT\\_IRQENABLE\\_CLR](#)). When the interrupt event is issued, the associated interrupt status bit is set in the timer status register ([GPT\\_TISR](#) for GPTIMER1, GPTIMER2, and GPTIMER10, and [GPT\\_IRQSTATUS](#) for other timers).

#### 22.2.4.6 Timer Mode Functionality

The timer is an upward counter that can be started and stopped at any time through the timer control register ([GPT\\_TCLR\[0\]](#) ST bit). The timer counter register ([GPT\\_TCRR](#)) can be loaded when stopped or on-the-fly (while counting). [GPT\\_TCRR](#) can be loaded directly by a [GPT\\_TCRR](#) write access with a new

timer value. The `GPT_TCR` register can also be loaded with the value held in the timer-load register `GPT_TLDR` by a trigger register (`GPT_TTGR`) write access. Loading of `GPT_TCR` is done regardless of the `GPT_TTGR` written value. The timer counter register `GPT_TCR` value can be read when stopped, or captured on-the-fly by a `GPT_TCR` read access. The timer is stopped and the counter value is set to 0 when the module reset is asserted. The timer is maintained at stop after the reset is released.

In one-shot mode (the `GPT_TCLR[1]` AR bit set to 0), the counter is stopped after counting overflow occurs (the counter value remains at 0).

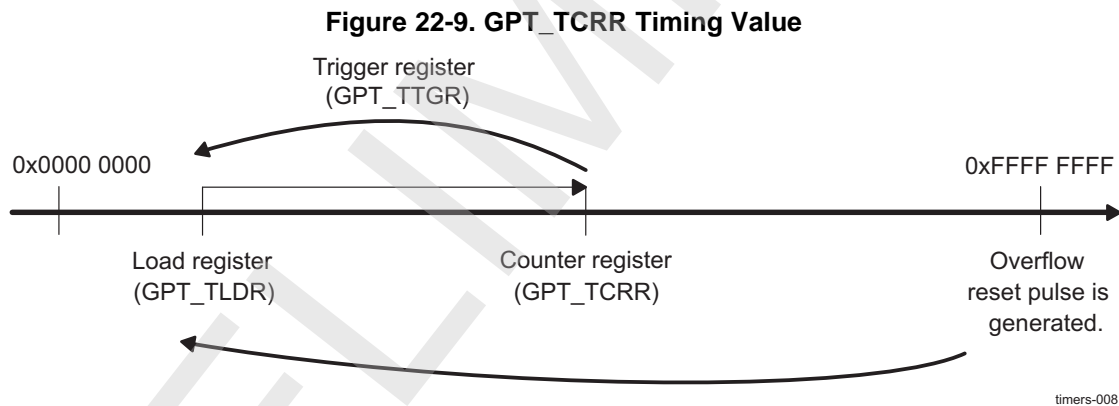
When the autoreload mode is enabled (the `GPT_TCLR[1]` AR bit set to 1), the `GPT_TCR` register is reloaded with the timer-load register (`GPT_TLDR`) value after a counting overflow occurs.

**CAUTION**

Do not put the overflow value (0xFFFF FFFF) in the `GPT_TLDR` register because it can lead to undesirable results.

An interrupt can be issued on overflow if the overflow interrupt-enable bit is set in the timer interrupt-enable register (`GPT_TIER[1]` OVF\_IT\_ENA bit set to 1 for GPTIMER1, GPTIMER2, and GPTIMER10, or `GPT_IRQENABLE_SET[1]` OVF\_IT\_FLAG bit set to 1 for other timers). A dedicated output pin (timer PWM) can be programmed in `GPT_TCLR[12]` through `GPT_TCLR[11:10]` (PT and TRG bits) to generate one positive pulse (prescaler duration) or to invert the current value (toggle mode) when an overflow occurs. The `GPT_TCLR[12]` PT bit selects pulse/toggle modulation (`GPT_TCLR[11:10]` TRG bit field selects trigger mode).

Figure 22-9 shows the `GPT_TCR` timing value.



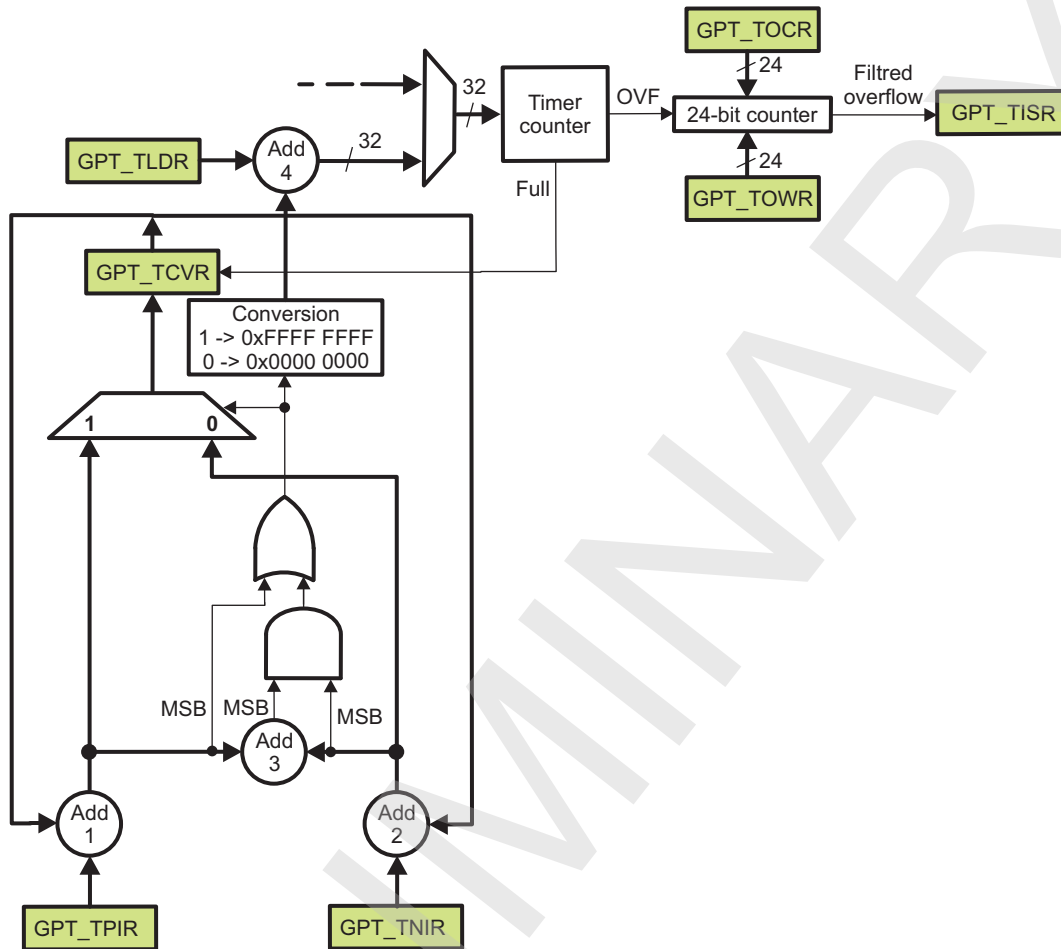
#### 22.2.4.6.1 1-ms Tick Generation (Only GPTIMER1, GPTIMER2, and GPTIMER10)

The interrupt period is not exactly 1 ms because the timer input clock is 32,768 Hz. If the clock counts up to 32, it obtains a 0.977-ms period; if it counts up to 33, it obtains a 1.007-ms period. For large granularity, the error is cumulative and can generate important deviations from the standard value.

To minimize the error between a true 1-ms tick and the tick generated by the 32,768 Hz timer, the sequencing of periods less than 1 ms and periods greater than 1 ms must be shuffled. An additional 1-ms block is used to correct this error. See Figure 22-10.

In this implementation, the increment sequencing is automatically managed by the timer to minimize the error. The user must define only the value of the timer positive increment register (`GPT_TPIR[31:0]` POSITIVE\_INC\_VALUE bit field) and the timer negative increment register (`GPT_TNIR[31:0]` NEGATIVE\_INC\_VALUE bit field). An automatic adaptation mechanism is used to simplify the programming model.

Figure 22-10. Block Diagram of the 1-ms Tick Module



timers-009

The **GPT\_TPIR**, **GPT\_TNIR**, and **GPT\_TCRV** registers and adders Add1, Add2, and Add3 are used to define whether the next value loaded in the timer counter register (**GPT\_TCRV[31:0]** **TIMER\_COUNTER** bit field) is the value of the **GPT\_TLDR[31:0]** **LOAD\_VALUE** bit field (period less than 1 ms) or the value of **GPT\_TLDR[31:0]** **LOAD\_VALUE** –1 (period greater than 1 ms).

Table 22-8 lists the value loaded in the **GPT\_TCRV** register according to the sign of the result of Add1, Add2, and Add3.

MSB = 0: Positive value, MSB = 1: Negative value

Table 22-8. Value Loaded in **GPT\_TCRV** to Generate 1-ms Tick

Add1 MSB	Add2 MSB	Add3 MSB	Value of <b>GPT_TCRV</b> Register
0	0	0	<b>GPT_TLDR[31:0]</b> <b>LOAD_VALUE</b> bit field
0	0	1	<b>GPT_TLDR[31:0]</b> <b>LOAD_VALUE</b> bit field
0	1	0	<b>GPT_TLDR[31:0]</b> <b>LOAD_VALUE</b> bit field
0	1	1	<b>GPT_TLDR[31:0]</b> <b>LOAD_VALUE</b> –1
1	0	0	N/A
1	0	1	N/A
1	1	0	<b>GPT_TLDR[31:0]</b> <b>LOAD_VALUE</b> –1
1	1	1	<b>GPT_TLDR[31:0]</b> <b>LOAD_VALUE</b> –1



The values of the **GPT\_TPIR** and **GPT\_TNIR** registers are calculated using the following formula:

- Positive increment value =  $(\text{INTEGER}[F_{\text{clk}} * T_{\text{tick}}] + 1) * 1\text{e}6) - (F_{\text{clk}} * T_{\text{tick}} * 1\text{e}6)$
- Negative increment value =  $(\text{INTEGER}[F_{\text{clk}} * T_{\text{tick}}] * 1\text{e}6) - (F_{\text{clk}} * T_{\text{tick}} * 1\text{e}6)$

---

**NOTE:** Fclk clock frequency (kHz)

Ttick tick period (ms)

---

The timer overflow counter register (**GPT\_TOCR**) and the timer overflow wrapping register **GPT\_TOWR** are used to filter interrupts. When the timer overflows, it increments the 24-bit **GPT\_TOCR** register. When the 24-bit **GPT\_TOCR** register values match the value in the 24-bit **GPT\_TOWR** register and the timer overflow is asserted, the **GPT\_TOCR** register is reset and an interrupt is generated to the **GPT\_TISR** register.

With the conversion block in reset state (the positive increment register, negative increment register, and counter value register are zeroed), the programming model and the behavior of GPTIMER1, GPTIMER2, and GPTIMER10 remain unchanged.

For 1-ms tick with a 32,768-Hz clock:

- **GPT\_TPIR**[31:0] POSITIVE\_INC\_VALUE = 232000
- **GPT\_TNIR**[31:0] NEGATIVE\_INC\_VALUE = -768000
- **GPT\_TLDR**[31:0] LOAD\_VALUE = 0xFFFF FFE0

---

**NOTE:** Any value of the tick period can be generated with the appropriate value of the **GPT\_TPIR**, **GPT\_TNIR**, and **GPT\_TLDR** registers.

By default, the **GPT\_TPIR**, **GPT\_TNIR**, **GPT\_TCVR**, **GPT\_TOCR**, and **GPT\_TOWR** registers and the associated logic are in reset mode (all 0s) and have no action on the programming model.

---

#### 22.2.4.7 Capture Mode Functionality

When a transition is detected on the module input pin (**EVENT\_CAPTURE**), the timer value in the **GPT\_TCRR** register can be captured and saved in the **GPT\_TCAR1** or **TCAR2** register function of the mode selected in the **GPT\_TCLR**[13] **CAPT\_MODE** bit. The edge detection circuitry monitors transitions on the input pin (**EVENT\_CAPTURE**).

The rising edge, falling edge, or both, can be selected in the **GPT\_TCLR**[9:8] **TCM** bit field to trigger the timer counter capture. The module sets the **GPT\_TISR**[2] **TCAR\_IT\_FLAG** bit for GPTIMER1, GPTIMER2, or GPTIMER10, and the **GPT\_IRQSTATUS**[2] **TCAR\_IT\_FLAG** bit for other timers, when an active edge is detected, and at the same time, the counter value **GPT\_TCRR** is stored in timer capture register **GPT\_TCAR1** or **GPT\_TCAR2**, as follows:

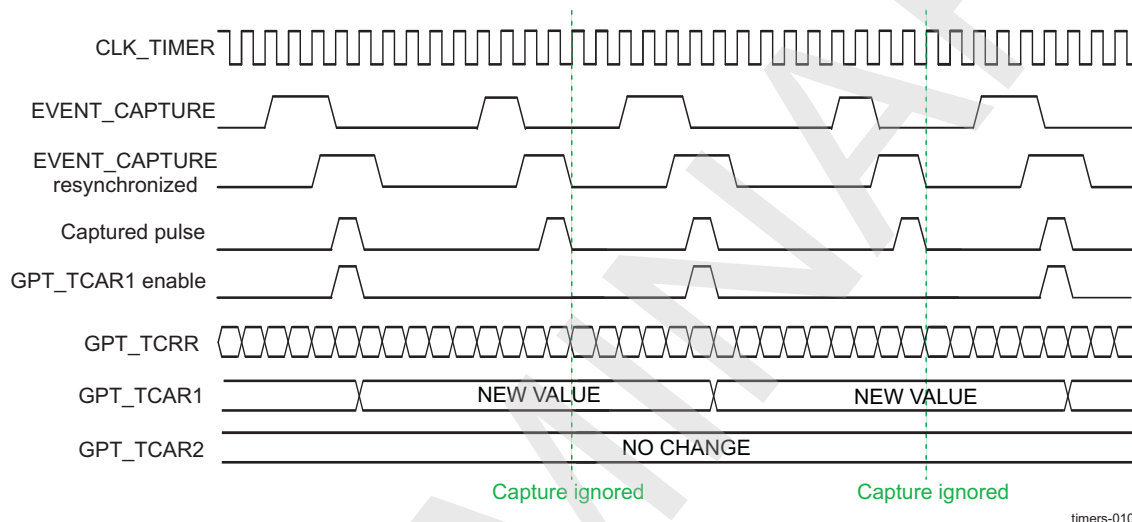
- If the **GPT\_TCLR**[13] **CAPT\_MODE** bit is 0, on the first enabled capture event, the value of the counter register is saved in the **GPT\_TCAR1** register, and all the next events are ignored (no update on the **GPT\_TCAR1** register and no interrupt triggering) until the detection logic is reset or the **GPT\_TISR**[2] **TCAR\_IT\_FLAG** bit (or **GPT\_IRQSTATUS**[2] **TCAR\_IT\_FLAG**, depending on the timer) is cleared by writing 1 to it.
- If the **GPT\_TCLR**[13] **CAPT\_MODE** bit is 1, on the first enabled capture event, the value of the counter register is saved in the **GPT\_TCAR1** register, and on the second enabled capture event, the value of the counter register is saved in the **GPT\_TCAR2** register. If a capture interrupt is enabled, the interrupt triggers on the second event capture. All other events are ignored (no update on **GPT\_TCAR1**/**GPT\_TCAR2** and no interrupt triggering) until the detection logic is reset or **GPT\_TISR**[2] **TCAR\_IT\_FLAG** bit (or **GPT\_IRQSTATUS**[2] **TCAR\_IT\_FLAG**, depending on the timer) is cleared by writing 1 to it. This mechanism is useful for period calculation of a clock, if that clock is connected to the **EVENT\_CAPTURE** input pin.

The edge detection logic is reset (a new capture is enabled) when the active capture interrupt is served. The **GPT\_TISR**[2] **TCAR\_IT\_FLAG** bit or the **GPT\_IRQSTATUS**[2] **TCAR\_IT\_FLAG** (previously 1) is cleared by writing 1 to it or when the edge detection mode bits (the **GPT\_TCLR**[9:8] **TCM** bit field) are

changed from no-capture mode detection to any other mode. The timer functional clock (input to prescaler) is used to sample the input pin (EVENT\_CAPTURE). A negative or positive pulse input can be detected when the pulse time is greater than the functional clock period. An interrupt is issued on edge detection if the capture interrupt-enable bit is set in the `GPT_TIER[2]` `TCAR_IT_ENA` bit (for GPTIMER1, GPTIMER2, or GPTIMER10), or in the `GPT_IRQENABLE_SET[2]` `TCAR_EN_FLAG` bit (for other timers). See the examples in [Figure 22-11](#) and [Figure 22-12](#).

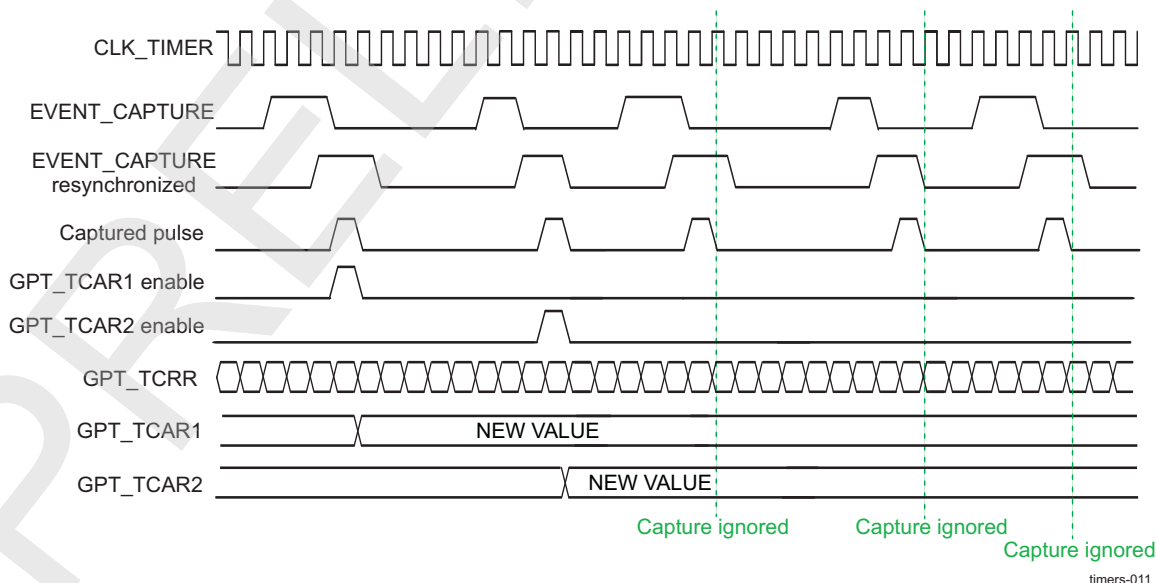
In [Figure 22-11](#), the `GPT_TCLR[9:8]` TCM bit field value is 0b01, and `GPT_TCLR[13]` `CAPT_MODE` bit is 0. Only the rising edge of `EVENT_CAPTURE` triggers a capture in the `GPT_TCAR1` and `GPT_TCAR2` registers, and only the `GPT_TCAR1` register updates.

**Figure 22-11. Capture Wave Example for `GPT_TCLR[13]` `CAPT_MODE` = 0**



In [Figure 22-12](#), the value of the `GPT_TCLR[9:8]` TCM bit field is 0b01, and the `GPT_TCLR[13]` `CAPT_MODE` bit is 1. Only the rising edge of `EVENT_CAPTURE` triggers a capture in the `GPT_TCAR1` register on the first enabled event, and the `GPT_TCAR2` register updates on the second enabled event.

**Figure 22-12. Capture Wave Example for `GPT_TCLR[13]` `CAPT_MODE` = 1**



#### 22.2.4.8 Compare Mode Functionality

When compare enable-register `GPT_TCLR[6]` `CE` bit is set to 1, the timer value (`GPT_TCR[31:0]`)

TIMER\_COUNTER bit field) is continuously compared to the value held in the timer match register (GPT\_TMAR). The GPT\_TMAR[31:0] COMPARE\_VALUE bit field value can be loaded at any time (timer counting or stopped). When the GPT\_TCRR and the GPT\_TMAR values match, an interrupt is issued, if the GPT\_TIER[0] MAT\_IT\_ENA (for GPTIMER1, GPTIMER2, and GPTIMER10), or GPT\_IRQSTATUS\_SET[0] MAT\_EN\_FLAG bit (for other timers) is set.

The right programming way is to write a compare value to the GPT\_TMAR register before setting the GPT\_TCLR[6] CE bit to prevent any unwanted interrupts due to a reset value matching effect.

The dedicated output pin (timer PWM) can be programmed in the GPT\_TCLR[12] PT bit through the GPT\_TCLR[11:10] TRG bit field to generate one positive pulse (timer clock duration) or to invert the current value (toggle mode) when an overflow or a match occurs.

### 22.2.4.9 Prescaler Functionality

A prescaler can be used to divide the timer counter input clock frequency. The prescaler is enabled when the GPT\_TCLR[5] PRE bit is set. The GPT\_TCLR[4:2] PTV bit field sets the second prescaler ratio. The prescaler counter is reset when the timer counter is stopped or reloaded on-the-fly.

Table 22-9 lists the prescaler/timer reload values versus contexts.

**Table 22-9. Prescaler/Timer Reload Values Versus Contexts**

Context	Prescaler	Timer Counter
Overflow (when autoreload is on)	Reset	GPT_TLDR[31:0]
GPT_TCRR write	Reset	GPT_TCRR[31:0]
GPT_TTGR write	Reset	GPT_TLDR[31:0]
Stop	Reset	Frozen

### 22.2.4.10 Pulse-Width Modulation

The timer can be configured to provide a programmable PWM output. The timer PWM output pin can be configured to toggle on an event. The GPT\_TCLR[11:10] TRG bit field determines on which register value the PWM pin toggles. Either overflow alone or both overflow and match can be selected to toggle the timer PWM pin when a compare condition occurs.

**CAUTION**

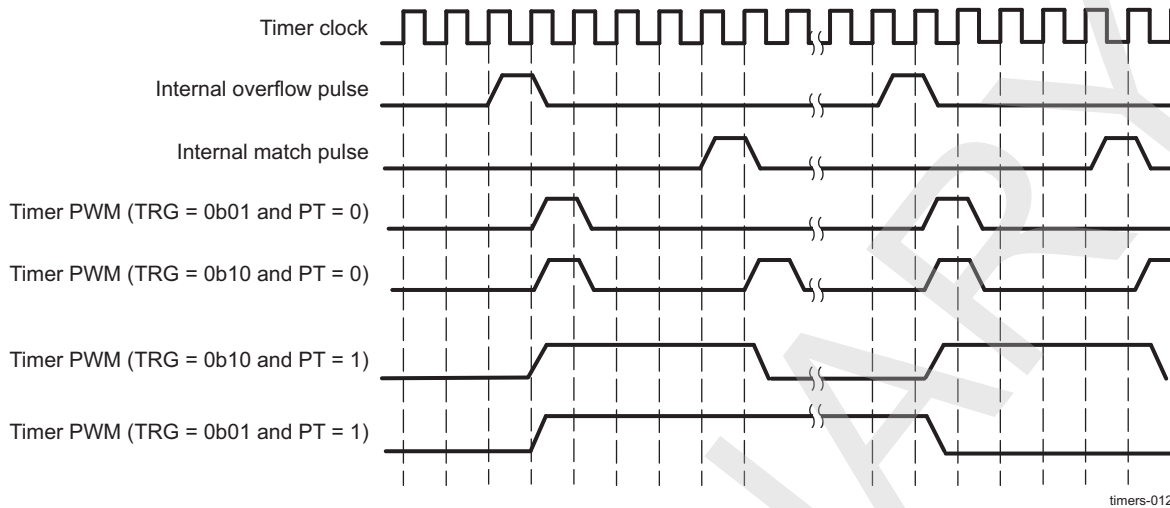
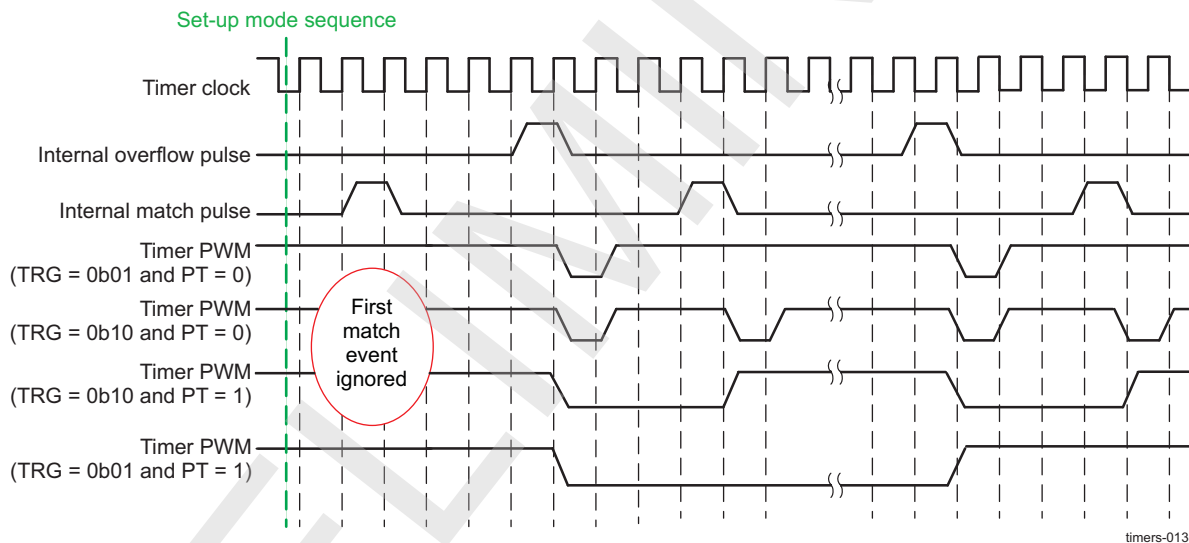
In toggle mode, when GPT\_TCLR[11:10] TRG = 0x2 (overflow and match), the first event that toggles the PWM line is an overflow event. If a match event occurs first; it does not toggle the PWM line (see Figure 22-14).

The GPT\_TCLR[7] SCPWM bit can be programmed to set or clear the timer PWM output signal only while the counter is stopped or the trigger is off. This allows setting the output pin to a known state before modulation starts. Modulation synchronously stops when the GPT\_TCLR[11:10] TRG bit field is cleared and overflow occurs. This allows fixing a deterministic state of the output pin when modulation stops.

In Figure 22-13, the internal overflow pulse is set each time (0xFFFF FFFF - GPT\_TLDR[31:0] LOAD\_VALUE + 1) the value is reached, and the internal match pulse is set when the counter reaches the GPT\_TMAR register value. Depending on the value of the GPT\_TCLR[12] PT bit and GPT\_TCLR[11:10] TRG bit field, the timer provides pulse or PWM event on the output pin (timer PWM).

The GPT\_TLDR and GPT\_TMAR registers must keep values below the overflow value (0xFFFF FFFF) by at least two units. In case the PWM trigger events are both overflow and match, the difference between the values kept in the GPT\_TMAR register and the value in the GPT\_TLDR register must be at least two units. When match event is used, the compare mode GPT\_TCLR[6] CE bit must be set.

In Figure 22-13, the GPT\_TCLR[7] SCPWM bit is set to 0. In Figure 22-14, the GPT\_TCLR[7] SCPWM bit is set to 1. To obtain the desired wave form, start the counter at 0xFFFF FFFE value (to ensure an overflow first) or adjust the line polarity (GPT\_TCLR[7] SCPWM bit).

**Figure 22-13. Timing Diagram of PWM With GPT\_TCLR[7] SCPWM Bit = 0****Figure 22-14. Timing Diagram of PWM With GPT\_TCLR[7] SCPWM Bit = 1**

### 22.2.4.11 Timer Counting Rate

The timer rate is defined by the following values:

- Value of the prescaler fields (the [GPT\\_TCLR\[5\] PRE](#) bit and the [GPT\\_TCLR\[4:2\] PTV](#) bit field)
- Value loaded into the timer-load register ([GPT\\_TLDR](#))

[Table 22-10](#) lists the prescaler clock ratio values.

**Table 22-10. Prescaler Clock Ratio Values**

<a href="#">GPT_TCLR[5] PRE</a>	<a href="#">GPT_TCLR[4:2] PTV</a>	Divisor (PS)
0	X	1
1	0	2
1	1	4
1	2	8
1	3	16
1	4	32

**Table 22-10. Prescaler Clock Ratio Values (continued)**

GPT_TCLR[5] PRE	GPT_TCLR[4:2] PTV	Divisor (PS)
1	5	64
1	6	128
1	7	256

Thus, the timer overflow rate is expressed as:

$$OVF\_Rate = (0xFFFF FFFF - GPT\_TLDR + 1) * (\text{timer-functional clock period}) * PS$$

With (timer-functional clock period) = 1/(timer-functional clock frequency) and PS = 2<sup>(PTV + 1)</sup> if prescaler is enabled, or PS = 1 if prescaler is disabled.

**CAUTION**

Internal resynchronization causes any write to the GPT\_TCLR[1] ST bit to have some latency before the register is updated:

2.5 \* functional clock cycles write\_GPT\_TCLR\_latency 3.5 \* functional clock cycles

Remember to consider this latency whenever the timer must be started or stopped by a software change to the GPT\_TCLR[1] ST bit.

**CAUTION**

- In the non-PWM mode, GPT\_TLDR must be maintained at less than or equal to 0xFFFF FFFE.
- In the PWM mode, GPT\_TLDR must be maintained at less than or equal to 0xFFFF FFFD.

For example, with a timer clock input of 32 kHz and the GPT\_TCLR[5] PRE bit set to 0, the timer output period is as listed in Table 22-11.

**Table 22-11. Value and Corresponding Interrupt Period**

GPT_TLDR[31:0] LOAD_VALUE	Interrupt Period
0x0000 0000	39 h
0xFFFF 0000	2.1 s
0xFFFF FFF0	524 μs
0xFFFF FFFE	65.5 μs

**22.2.4.12 Timer Under Emulation**

During emulation mode, the timer continues to run according to the value of the GPT1MS\_TIOCP\_CFG[5]/GPT\_TIOCP\_CFG[1] EMUFREE bit.

If the GPT1MS\_TIOCP\_CFG[5]/GPT\_TIOCP\_CFG[1] EMUFREE bit is set to 1, timer execution is not stopped in emulation mode and the interrupt is still generated when overflow or match is reached.

If the GPT1MS\_TIOCP\_CFG[5]/GPT\_TIOCP\_CFG[1] EMUFREE bit is set to 0, the prescaler and timer are frozen and both resume on exit from emulation mode. The asynchronous external input pin (dmtimer\_i\_pwm\_evt, with i=[8:11]) is internally synchronized on two timer-clock rising edges.

**22.2.4.13 Accessing GP Timer Registers**

All accesses are nonposted until software reconfiguration. All registers are 32 bits wide, accessible through the L4 interface with 16-bit or 32-bit access (read/write).

Any 16-bit write access must be least-significant bit (LSB) first, and the second write access must be most-significant bit (MSB) first. Write operations to the following GP timer registers can skip the MSB access if it is not necessary to update the 16 MSBs of the register:

- GPT\_TIDR
- GPT\_TIOCP\_CFG
- GPT1MS\_TIOCP\_CFG
- GPT\_IRQSTATUS\_RAW
- GPT\_IRQSTATUS
- GPT\_IRQENABLE\_SET
- GPT\_IRQENABLE\_CLR
- GPT\_IRQWAKEEN
- GPT\_TISR
- GPT\_TIER
- GPT\_TWER
- GPT\_TSICR

Write operations to the following functional registers must be complete (the MSB must be written even if the MSB data is not used):

- GPT\_TCLR
- GPT\_TCRR
- GPT\_TLDR
- GPT\_TTGR
- GPT\_TMAR
- GPT\_TPIR
- GPT\_TNIR
- GPT\_TCVR
- GPT\_TOCR
- GPT\_TOWR

The following L4 synchronous registers are not affected by the posted/nonposted mode selection; the write/read operation is effective and acknowledged (command accepted) after one L4 clock cycle from command assertion:

- GPT\_TIDR
- GPT\_TIOCP\_CFG
- GPT1MS\_TIOCP\_CFG
- GPT\_TISTAT
- GPT\_IRQSTATUS
- GPT\_IRQSTATUS\_RAW
- GPT\_IRQENABLE\_SET
- GPT\_IRQENABLE\_CLR
- GPT\_IRQWAKEEN
- GPT\_TWPS
- GPT\_TSICR

#### 22.2.4.13.1 Writing to Timer Registers

The host uses the L4 interface to write to the following registers synchronously with the timer interface clock:

- GPT\_TLDR
- GPT\_TCRR
- GPT\_TCLR
- GPT\_TIOCP\_CFG

- [GPT1MS\\_TIOCP\\_CFG](#)
- [GPT\\_IRQSTATUS](#)
- [GPT\\_IRQENABLE\\_SET](#)
- [GPT\\_IRQENABLE\\_CLR](#)
- [GPT\\_IRQWAKEEN](#)
- [GPT\\_TTGR](#)
- [GPT\\_TSICR](#)
- [GPT\\_TMAR](#)

GPTIMER1, GPTIMER2, and GPTIMER10 also have the following registers:

- [GPT\\_TPIR](#)
- [GPT\\_TNIR](#)
- [GPT\\_TCVR](#)
- [GPT\\_TOCR](#)
- [GPT\\_TOWR](#)
- [GPT\\_TIER](#)
- [GPT\\_TISR](#)
- [GPT\\_TWER](#)

In 16-bit access mode, the 16 LSBs must be written before writing to the 16 MSBs.

#### **22.2.4.13.1.1 Write Posting Synchronization Mode**

This mode is used if the [GPT\\_TSICR](#)[2] POSTED bit is set to 1.

This mode uses a posted write scheme to update any internal register ([GPT\\_TCLR](#), [GPT\\_TCRR](#), [GPT\\_TLDR](#), [GPT\\_TTGR](#), [GPT\\_TMAR](#), and [GPT\\_TPIR](#), [GPT\\_TNIR](#), [GPT\\_TCVR](#), [GPT\\_TOCR](#), and [GPT\\_TOWR](#) for GPTIMER1, GPTIMER2, and GPTIMER10). Therefore, the write transaction is immediately acknowledged on the L4 interface, although the effective write operation occurs later due to a resynchronization in the timer clock domain. The advantage is that neither the interconnect nor the device that requested the write transaction are stalled.

For each register, a status bit is provided in the timer write-posted status register [GPT\\_TWPS](#). In this mode, it is mandatory that the software checks this status bit prior to any write access. In case a write is attempted to a register with a previous access pending, the previous access is discarded without notice.

The timer module updates the timer counter register value synchronously with the L4 clock. Consequently, any read access to the timer counter register [GPT\\_TCRR](#) does not add any resynchronization latency; the current value is always available.

---

**NOTE:** Since the overflow IRQ is generated, when the TCRR value reaches 0xFFFF FFFF, and not when it changes its value to the value after overflow, one must wait a delay of (1 \* PS \* timer-functional clock period) before any read access to the timer counter register [GPT\\_TCRR](#) to ensure a correct reading of its content.

---

If a write access is pending for a register, reading from this register does not yield a correct result. Software synchronization must be used to avoid incorrect results.

The drawback of this automatic update mechanism is that it assumes a given relationship between the timer interface frequency and the timer clock frequency.

Functional frequency range:  $\text{freq}(\text{timer clock}) < \text{freq}(\text{L4 interface clock})/4$ .

### 22.2.4.13.1.2 Write Nonposting Synchronization Mode

This mode is used if the `GPT_TSICR[2]` POSTED bit is set to 0. It uses a nonposted write scheme to update any internal register. Therefore, the write transaction is not acknowledged on the L4 interface until the effective write operation occurs after the resynchronization in the timer functional clock domain. The drawback is that both the interconnect and the device that requested the write transaction are stalled during this period.

The same full resynchronization scheme is used for a read transaction, and the same stall period applies. A register read following a write to the same register is always coherent.

This mode is functional regardless of the ratio between the L4 interface frequency and the timer clock frequency.

### 22.2.4.13.2 Reading From Timer Counter Registers

In 16-bit access mode, reading the 16 LSBs from the timer counter registers (`GPT_TCRR`, `GPT_TCAR1`, and `GPT_TCAR2`) captures the current timer counter value. This must be followed by reading the 16 MSBs.

IVA3 subsystem 16-bit accesses can be interleaved with MPU subsystem 32-bit accesses.

---

**NOTE:** LSB/MSB accesses cannot be interleaved (that is, the sequence LSB register 1, LSB register 2, MSB register 1, MSB register 2 is not supported).

---

## 22.2.5 GP Timer Low-Level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

### 22.2.5.1 Global Initialization

#### 22.2.5.1.1 Global Initialization of Surrounding Modules

This section identifies the requirements for initializing the surrounding modules when the GP timer module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the GP timer. For more information, see [Section 22.2.3 GP Timers Integration](#) and [Section 22.2.2 GP Timers Environment](#).

**Table 22-12. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. For more information about the module configuration, see <a href="#">Section 3.1.1.1.2, Module Level Clock Management</a> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Control Module	The module specific pad muxing must be set in the control module. For more information about the module configuration, see <a href="#">Section 18.4.8 PAD Functional Multiplexing and Configuration</a> , in <a href="#">Chapter 18, Control Module</a> .
MPU INTC	The MPU INTC configuration must be done to enable the interrupts from the GP timer module. See <a href="#">Interrupt Controller Registers</a> , in <a href="#">Chapter 4, Cortex-A9 MPU Subsystem</a> .
DSP INTC	The DSP INTC configuration must be done to enable the interrupts from GP timer module. See <a href="#">INTC</a> , in <a href="#">Chapter 5, DSP Subsystem</a> .

#### 22.2.5.1.2 GP Timer Module Global Initialization

##### 22.2.5.1.2.1 Main Sequence – GP Timer Module Global Initialization

This section identifies the main steps for initializing the GP timer module when the module is to be used for the first time.



**Table 22-13. GP Timer Module Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Execute software reset.	GPT_TIOCP_CFG[0]SOFTRESET <sup>(1)</sup> or GPT1MS_TIOCP_CFG[0] SOFTRESET <sup>(2)</sup>	0x1
Wait until reset release?	GPT_TIOCP_CFG[0] SOFTRESET <sup>(1)</sup> or GPT1MS_TIOCP_CFG[0] SOFTRESET <sup>(2)</sup>	0x0
Configure Idle mode.	GPT_TIOCP_CFG[3:2] IDLEMODE <sup>(1)</sup> or GPT1MS_TIOCP_CFG[4:3] IDLEMODE <sup>(2)</sup>	xx
Configure clock activity.	GPT1MS_TIOCP_CFG[9:8] CLOCKACTIVITY <sup>(2)</sup>	xx
Configure wake-up feature.	GPT1MS_TIOCP_CFG[2] ENAWAKEUP <sup>(2)</sup>	x
Enable wake-up interrupt events.	GPT_IRQWAKEEN[2:0] <sup>(1)</sup> or GPT_TWERR[2:0] <sup>(2)</sup>	x
Select posted mode.	GPT_TSICR[2] POSTED	x

<sup>(1)</sup> Applies only to GPTIMER3 through GPTIMER9, and GPTIMER11

<sup>(2)</sup> Applies only to GPTIMER1, GPTIMER2, and GPTIMER10

## 22.2.5.2 Operational Mode Configuration

### 22.2.5.2.1 GP Timer Mode

#### 22.2.5.2.1.1 Main Sequence – GP Timer Mode Configuration

**Table 22-14. GP Timer Mode Configuration**

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode	GPT_TCLR[1] AR	x
Set prescale timer value.	GPT_TCLR[4:2] PTV	xxx
Enable prescaler.	GPT_TCLR[5] PRE	0x1
Enable overflow interrupt.	GPT_TIER[1] OVF_IT_ENA <sup>(1)</sup> or GPT_IRQENABLE_SET[1] OVF_EN_FLAG <sup>(2)</sup>	0x1
Load timer counter value.	GPT_TCRR	xxx
Load timer load value.	GPT_TLDR	xxx
Start the timer.	GPT_TCLR[0] ST	0x1

<sup>(1)</sup> Applies only to GPTIMER1, GPTIMER2, and GPTIMER10.

<sup>(2)</sup> Applies only to GPTIMER3 through GPTIMER 9, and GPTIMER11.

### 22.2.5.2.2 GP Timer Compare Mode

#### 22.2.5.2.2.1 Main Sequence – GP Timer Compare Mode Configuration

**Table 22-15. GP Timer Compare Mode Configuration**

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	GPT_TCLR[1] AR	x
Set prescale timer value.	GPT_TCLR[4:2] PTV	xxx
Enable prescaler.	GPT_TCLR[5] PRE	0x1
Enable match interrupt.	GPT_TIER[0] MAT_IT_ENA <sup>(1)</sup> or GPT_IRQENABLE_SET[0] MAT_EN_FLAG <sup>(2)</sup>	0x1
Load timer counter value.	GPT_TCRR	xxx
Load timer compare value.	GPT_TMAR	xxx
Enable Compare mode.	GPT_TCLR[6] CE	0x1
Start the timer.	GPT_TCLR[0] ST	0x1

<sup>(1)</sup> Applies only to GPTIMER1, GPTIMER2, and GPTIMER10.

<sup>(2)</sup> Applies only to GPTIMER3 through GPTIMER9, and GPTIMER11.

### 22.2.5.2.3 GP Timer Capture Mode

#### 22.2.5.2.3.1 Main Sequence – GP Timer Capture Mode Configuration

**Table 22-16. GP Timer Capture Mode Configuration**

Step	Register/Bit Field/Programming Model	Value
Initialize capture mode.	See <a href="#">Section 22.2.5.2.3.2</a> .	
Enable capture interrupt.	<a href="#">GPT_TIER[2]</a> TCAR_IT_ENA <sup>(1)</sup> or <a href="#">GPT_IRQENABLE_SET[2]</a> TCAR_EN_FLAG <sup>(2)</sup>	0x1
Start the timer.	<a href="#">GPT_TCLR[0]</a> ST	0x1
Detect event.	See <a href="#">Section 22.2.5.2.3.3</a> .	

<sup>(1)</sup> Applies only to GPTIMER1, GPTIMER2, and GPTIMER10.

<sup>(2)</sup> Applies only to GPTIMER3 through GPTIMER9, and GPTIMER11.

#### 22.2.5.2.3.2 Subsequence – Initialize Capture Mode

**Table 22-17. Initialize Capture Mode**

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	<a href="#">GPT_TCLR[1]</a> AR	x
Set prescale timer value.	<a href="#">GPT_TCLR[4:2]</a> PTV	xxx
Enable prescaler.	<a href="#">GPT_TCLR[5]</a> PRE	0x1
Select GPTx (where x = 8 to 11). Capture input at device pin dmtimerx_pwm_evt.	<a href="#">GPT_TCLR[14]</a> GPO_CFG <sup>(1)</sup>	0x1
Select single or second event capture.	<a href="#">GPT_TCLR[13]</a> CAPT_MODE	x
Select transition capture mode.	<a href="#">GPT_TCLR[9:8]</a> TCM	xx

<sup>(1)</sup> Applies only to GPTIMER8 through GPTIMER11

#### 22.2.5.2.3.3 Subsequence – Detect Event

**Table 22-18. Detect Event**

Step	Register/Bit Field/Programming Model	Value
Wait until event detected?	<a href="#">GPT_TISR[2]</a> TCAR_IT_FLAG <sup>(1)</sup> or <a href="#">GPT_IRQSTATUS[2]</a> TCAR_IT_FLAG <sup>(2)</sup>	= 0x1
Read Timer capture value.	<a href="#">GPT_TCAR1</a> and/or <a href="#">GPT_TCAR2</a>	
Clear capture interrupt request.	<a href="#">GPT_TISR[2]</a> TCAR_IT_FLAG <sup>(1)</sup> or <a href="#">GPT_IRQSTATUS[2]</a> TCAR_IT_FLAG <sup>(2)</sup>	0x1

<sup>(1)</sup> Applies only to GPTIMER1, GPTIMER2, and GPTIMER10.

<sup>(2)</sup> Applies only to GPTIMER3 through GPTIMER9, and GPTIMER11.

### 22.2.5.2.4 GP Timer PWM Mode

#### 22.2.5.2.4.1 Main Sequence – GP Timer PWM Mode Configuration

**Table 22-19. GP Timer PWM Mode Configuration**

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	<a href="#">GPT_TCLR[1]</a> AR	x
Set prescale timer value.	<a href="#">GPT_TCLR[4:2]</a> PTV	xxx
Enable prescaler.	<a href="#">GPT_TCLR[5]</a> PRE	0x1
Select trigger output mode.	<a href="#">GPT_TCLR[11:10]</a> TRG	xx

**Table 22-19. GP Timer PWM Mode Configuration (continued)**

<b>Step</b>	<b>Register/Bit Field/Programming Model</b>	<b>Value</b>
Select pulse or toggle modulation PWM mode.	<a href="#">GPT_TCLR</a> [12] PT	x
Select GPTx (where x = 8 to 11) pwm output at device pin dmtimerx_pwm_evt.	<a href="#">GPT_TCLR</a> [14] GPO_CFG <sup>(1)</sup>	0x0
Configure PWM output pin default value.	<a href="#">GPT_TCLR</a> [7] SCPWM	x
Load timer load value.	<a href="#">GPT_TLDR</a>	xxx
Load timer compare value.	<a href="#">GPT_TMAR</a>	xxx
Enable compare.	<a href="#">GPT_TCLR</a> [6] CE	0x1
Start the timer.	<a href="#">GPT_TCLR</a> [0] ST	0x1

<sup>(1)</sup> Applies only to GPTIMER8 through GPTIMER11

## 22.2.6 GP Timer Register Manual

### 22.2.6.1 GP Timer Instance Summary

Table 22-20 lists the base address and block size for the GP timer module instances.

**Table 22-20. GP Timer Instance Summary**

Module Name	Base Address L4 Interconnect	Base Address L3 Interconnect	Base Address Cortex-A9 Private Access	Base Address DSP Private Access	Size
GPTIMER2	0x4803 2000	-	-	-	4 KB
GPTIMER3	0x4803 4000	-	-	-	4 KB
GPTIMER4	0x4803 6000	-	-	-	4 KB
GPTIMER9	0x4803 E000	-	-	-	4 KB
GPTIMER10	0x4808 6000	-	-	-	4 KB
GPTIMER11	0x4808 8000	-	-	-	4 KB
GPTIMER5	-	0x4903 8000	0x4013 8000	0x01D3 8000	4 KB
GPTIMER6	-	0x4903 A000	0x4013 A000	0x01D3 A000	4 KB
GPTIMER7	-	0x4903 C000	0x4013 C000	0x01D3 C000	4 KB
GPTIMER8	-	0x4903 E000	0x4013 E000	0x01D3 E000	4 KB
GPTIMER1	0x4A31 8000	-	-	-	4 KB

**NOTE:** Private Access is an access which is not using the L3/L4 interconnects.

### 22.2.6.2 GP Timer Registers

#### 22.2.6.2.1 GP Timer Register Summary

#### CAUTION

The GP timer registers are limited to 32-bit and 16-bit data accesses; 8-bit access is not allowed and can corrupt the register content.

Table 22-21 through Table 22-27 provide the register summary and associated offset addresses for the 11 GP timer internal registers.

**Table 22-21. GPTIMER1, GPTIMER2, and GPTIMER10 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPTIMER1 Physical Address L4 Interconnect	GPTIMER2 Physical Address L4 Interconnect	GPTIMER10 Physical Address L4 Interconnect
GPT_TIDR	R	32	0x0000 0000	0x4A31 8000	0x4803 2000	0x4808 6000
GPT1MS_TIOCP_CFG	RW	32	0x0000 0010	0x4A31 8010	0x4803 2010	0x4808 6010
GPT_TISTAT	R	32	0x0000 0014	0x4A31 8014	0x4803 2014	0x4808 6014
GPT_TISR	RW	32	0x0000 0018	0x4A31 8018	0x4803 2018	0x4808 6018
GPT_TIER	RW	32	0x0000 001C	0x4A31 801C	0x4803 201C	0x4808 601C
GPT_TWER	RW	32	0x0000 0020	0x4A31 8020	0x4803 2020	0x4808 6020
GPT_TCLR	RW	32	0x0000 0024	0x4A31 8024	0x4803 2024	0x4808 6024
GPT_TCRR	RW	32	0x0000 0028	0x4A31 8028	0x4803 2028	0x4808 6028
GPT_TLDR	RW	32	0x0000 002C	0x4A31 802C	0x4803 202C	0x4808 602C
GPT_TTGR	RW	32	0x0000 0030	0x4A31 8030	0x4803 2030	0x4808 6030
GPT_TWPS	R	32	0x0000 0034	0x4A31 8034	0x4803 2034	0x4808 6034
GPT_TMAR	RW	32	0x0000 0038	0x4A31 8038	0x4803 2038	0x4808 6038
GPT_TCAR1	R	32	0x0000 003C	0x4A31 803C	0x4803 203C	0x4808 603C
GPT_TSICR	RW	32	0x0000 0040	0x4A31 8040	0x4803 2040	0x4808 6040
GPT_TCAR2	R	32	0x0000 0044	0x4A31 8044	0x4803 2044	0x4808 6044
GPT_TPIR	RW	32	0x0000 0048	0x4A31 8048	0x4803 2048	0x4808 6048
GPT_TNIR	RW	32	0x0000 004C	0x4A31 804C	0x4803 204C	0x4808 604C
GPT_TCVR	RW	32	0x0000 0050	0x4A31 8050	0x4803 2050	0x4808 6050
GPT_TOCR	RW	32	0x0000 0054	0x4A31 8054	0x4803 2054	0x4808 6054
GPT_TOWR	RW	32	0x0000 0058	0x4A31 8058	0x4803 2058	0x4808 6058

**Table 22-22. GPTIMER3 and GPTIMER4 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPTIMER3 Physical Address L4 Interconnect	GPTIMER4 Physical Address L4 Interconnect
GPT_TIDR	R	32	0x0000 0000	0x4803 4000	0x4803 6000
GPT_TIOCP_CFG	RW	32	0x0000 0010	0x4803 4010	0x4803 6010
RESERVED			0x0000 0020	0x4803 4020	0x4803 6020
GPT_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4803 4024	0x4803 6024
GPT_IRQSTATUS	RW	32	0x0000 0028	0x4803 4028	0x4803 6028
GPT_IRQENABLE_SET	RW	32	0x0000 002C	0x4803 402C	0x4803 602C
GPT_IRQENABLE_CLR	RW	32	0x0000 0030	0x4803 4030	0x4803 6030
GPT_IRQWAKEEN	RW	32	0x0000 0034	0x4803 4034	0x4803 6034
GPT_TCLR	RW	32	0x0000 0038	0x4803 4038	0x4803 6038
GPT_TCRR	RW	32	0x0000 003C	0x4803 403C	0x4803 603C
GPT_TLDR	RW	32	0x0000 0040	0x4803 4040	0x4803 6040
GPT_TTGR	RW	32	0x0000 0044	0x4803 4044	0x4803 6044
GPT_TWPS	R	32	0x0000 0048	0x4803 4048	0x4803 6048
GPT_TMAR	RW	32	0x0000 004C	0x4803 404C	0x4803 604C
GPT_TCAR1	R	32	0x0000 0050	0x4803 4050	0x4803 6050
GPT_TSICR	RW	32	0x0000 0054	0x4803 4054	0x4803 6054
GPT_TCAR2	R	32	0x0000 0058	0x4803 4058	0x4803 6058

**Table 22-23. GPTIMER5 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPTIMER5 Physical Address	GPTIMER5 Physical Address	GPTIMER5 Physical Address
				L3 Interconnect	Cortex-A9 Private Access	DSP Private Access
GPT_TIDR	R	32	0x0000 0000	0x4903 8000	0x4013 8000	0x01D3 8000
GPT_TIOCP_CFG	RW	32	0x0000 0010	0x4903 8010	0x4013 8010	0x01D3 8010
RESERVED			0x0000 0020	0x4903 8020	0x4013 8020	0x01D3 8020
GPT_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4903 8024	0x4013 8024	0x01D3 8024
GPT_IRQSTATUS	RW	32	0x0000 0028	0x4903 8028	0x4013 8028	0x01D3 8028
GPT_IRQENABLE_SET	RW	32	0x0000 002C	0x4903 802C	0x4013 802C	0x01D3 802C
GPT_IRQENABLE_CLR	RW	32	0x0000 0030	0x4903 8030	0x4013 8030	0x01D3 8030
GPT_IRQWAKEEN	RW	32	0x0000 0034	0x4903 8034	0x4013 8034	0x01D3 8034
GPT_TCLR	RW	32	0x0000 0038	0x4903 8038	0x4013 8038	0x01D3 8038
GPT_TCRR	RW	32	0x0000 003C	0x4903 803C	0x4013 803C	0x01D3 803C
GPT_TLDR	RW	32	0x0000 0040	0x4903 8040	0x4013 8040	0x01D3 8040
GPT_TTGR	RW	32	0x0000 0044	0x4903 8044	0x4013 8044	0x01D3 8044
GPT_TWPS	R	32	0x0000 0048	0x4903 8048	0x4013 8048	0x01D3 8048
GPT_TMAR	RW	32	0x0000 004C	0x4903 804C	0x4013 804C	0x01D3 804C
GPT_TCAR1	R	32	0x0000 0050	0x4903 8050	0x4013 8050	0x01D3 8050
GPT_TSICR	RW	32	0x0000 0054	0x4903 8054	0x4013 8054	0x01D3 8054
GPT_TCAR2	R	32	0x0000 0058	0x4903 8058	0x4013 8058	0x01D3 8058

**Table 22-24. GPTIMER6 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPTIMER6 Physical Address	GPTIMER6 Physical Address	GPTIMER6 Physical Address
				L3 Interconnect	Cortex-A9 Private Access	DSP Private Access
GPT_TIDR	R	32	0x0000 0000	0x4903 A000	0x4013 A000	0x01D3 A000
GPT_TIOCP_CFG	RW	32	0x0000 0010	0x4903 A010	0x4013 A010	0x01D3 A010
RESERVED			0x0000 0020	0x4903 A020	0x4013 A020	0x01D3 A020
GPT_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4903 A024	0x4013 A024	0x01D3 A024
GPT_IRQSTATUS	RW	32	0x0000 0028	0x4903 A028	0x4013 A028	0x01D3 A028
GPT_IRQENABLE_SET	RW	32	0x0000 002C	0x4903 A02C	0x4013 A02C	0x01D3 A02C
GPT_IRQENABLE_CLR	RW	32	0x0000 0030	0x4903 A030	0x4013 A030	0x01D3 A030
GPT_IRQWAKEEN	RW	32	0x0000 0034	0x4903 A034	0x4013 A034	0x01D3 A034
GPT_TCLR	RW	32	0x0000 0038	0x4903 A038	0x4013 A038	0x01D3 A038
GPT_TCRR	RW	32	0x0000 003C	0x4903 A03C	0x4013 A03C	0x01D3 A03C
GPT_TLDR	RW	32	0x0000 0040	0x4903 A040	0x4013 A040	0x01D3 A040
GPT_TTGR	RW	32	0x0000 0044	0x4903 A044	0x4013 A044	0x01D3 A044
GPT_TWPS	R	32	0x0000 0048	0x4903 A048	0x4013 A048	0x01D3 A048
GPT_TMAR	RW	32	0x0000 004C	0x4903 A04C	0x4013 A04C	0x01D3 A04C
GPT_TCAR1	R	32	0x0000 0050	0x4903 A050	0x4013 A050	0x01D3 A050
GPT_TSICR	RW	32	0x0000 0054	0x4903 A054	0x4013 A054	0x01D3 A054
GPT_TCAR2	R	32	0x0000 0058	0x4903 A058	0x4013 A058	0x01D3 A058

**Table 22-25. GPTIMER7 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPTIMER7 Physical Address L3 Interconnect	GPTIMER7 Physical Address Cortex-A9 Private Access	GPTIMER7 Physical Address DSP Private Access
GPT_TIDR	R	32	0x0000 0000	0x4903 C000	0x4013 C000	0x01D3 C000
GPT_TIOCP_CFG	RW	32	0x0000 0010	0x4903 C010	0x4013 C010	0x01D3 C010
RESERVED			0x0000 0020	0x4903 C020	0x4013 C020	0x01D3 C020
GPT_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4903 C024	0x4013 C024	0x01D3 C024
GPT_IRQSTATUS	RW	32	0x0000 0028	0x4903 C028	0x4013 C028	0x01D3 C028
GPT_IRQENABLE_SET	RW	32	0x0000 002C	0x4903 C02C	0x4013 C02C	0x01D3 C02C
GPT_IRQENABLE_CLR	RW	32	0x0000 0030	0x4903 C030	0x4013 C030	0x01D3 C030
GPT_IRQWAKEEN	RW	32	0x0000 0034	0x4903 C034	0x4013 C034	0x01D3 C034
GPT_TCLR	RW	32	0x0000 0038	0x4903 C038	0x4013 C038	0x01D3 C038
GPT_TCRR	RW	32	0x0000 003C	0x4903 C03C	0x4013 C03C	0x01D3 C03C
GPT_TLDR	RW	32	0x0000 0040	0x4903 C040	0x4013 C040	0x01D3 C040
GPT_TTGR	RW	32	0x0000 0044	0x4903 C044	0x4013 C044	0x01D3 C044
GPT_TWPS	R	32	0x0000 0048	0x4903 C048	0x4013 C048	0x01D3 C048
GPT_TMAR	RW	32	0x0000 004C	0x4903 C04C	0x4013 C04C	0x01D3 C04C
GPT_TCAR1	R	32	0x0000 0050	0x4903 C050	0x4013 C050	0x01D3 C050
GPT_TSICR	RW	32	0x0000 0054	0x4903 C054	0x4013 C054	0x01D3 C054
GPT_TCAR2	R	32	0x0000 0058	0x4903 C058	0x4013 C058	0x01D3 C058

**Table 22-26. GPTIMER8 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPTIMER8 Physical Address L3 Interconnect	GPTIMER8 Physical Address Cortex-A9 Private Access	GPTIMER8 Physical Address DSP Private Access
GPT_TIDR	R	32	0x0000 0000	0x4903 E000	0x4013 E000	0x01D3 E000
GPT_TIOCP_CFG	RW	32	0x0000 0010	0x4903 E010	0x4013 E010	0x01D3 E010
RESERVED			0x0000 0020	0x4903 E020	0x4013 E020	0x01D3 E020
GPT_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4903 E024	0x4013 E024	0x01D3 E024
GPT_IRQSTATUS	RW	32	0x0000 0028	0x4903 E028	0x4013 E028	0x01D3 E028
GPT_IRQENABLE_SET	RW	32	0x0000 002C	0x4903 E02C	0x4013 E02C	0x01D3 E02C
GPT_IRQENABLE_CLR	RW	32	0x0000 0030	0x4903 E030	0x4013 E030	0x01D3 E030
GPT_IRQWAKEEN	RW	32	0x0000 0034	0x4903 E034	0x4013 E034	0x01D3 E034
GPT_TCLR	RW	32	0x0000 0038	0x4903 E038	0x4013 E038	0x01D3 E038
GPT_TCRR	RW	32	0x0000 003C	0x4903 E03C	0x4013 E03C	0x01D3 E03C
GPT_TLDR	RW	32	0x0000 0040	0x4903 E040	0x4013 E040	0x01D3 E040
GPT_TTGR	RW	32	0x0000 0044	0x4903 E044	0x4013 E044	0x01D3 E044
GPT_TWPS	R	32	0x0000 0048	0x4903 E048	0x4013 E048	0x01D3 E048
GPT_TMAR	RW	32	0x0000 004C	0x4903 E04C	0x4013 E04C	0x01D3 E04C
GPT_TCAR1	R	32	0x0000 0050	0x4903 E050	0x4013 E050	0x01D3 E050
GPT_TSICR	RW	32	0x0000 0054	0x4903 E054	0x4013 E054	0x01D3 E054
GPT_TCAR2	R	32	0x0000 0058	0x4903 E058	0x4013 E058	0x01D3 E058

**Table 22-27. GPTIMER9 and GPTIMER11 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPTIMER9 Physical Address L4 Interconnect	GPTIMER11 Physical Address L4 Interconnect
GPT_TIDR	R	32	0x0000 0000	0x4803 E000	0x4808 8000
GPT_TIOCP_CFG	RW	32	0x0000 0010	0x4803 E010	0x4808 8010
RESERVED			0x0000 0020	0x4803 E020	0x4808 8020
GPT_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4803 E024	0x4808 8024
GPT_IRQSTATUS	RW	32	0x0000 0028	0x4803 E028	0x4808 8028
GPT_IRQENABLE_SET	RW	32	0x0000 002C	0x4803 E02C	0x4808 802C
GPT_IRQENABLE_CLR	RW	32	0x0000 0030	0x4803 E030	0x4808 8030
GPT_IRQWAKEEN	RW	32	0x0000 0034	0x4803 E034	0x4808 8034
GPT_TCLR	RW	32	0x0000 0038	0x4803 E038	0x4808 8038
GPT_TCRR	RW	32	0x0000 003C	0x4803 E03C	0x4808 803C
GPT_TLDR	RW	32	0x0000 0040	0x4803 E040	0x4808 8040
GPT_TTGR	RW	32	0x0000 0044	0x4803 E044	0x4808 8044
GPT_TWPS	R	32	0x0000 0048	0x4803 E048	0x4808 8048
GPT_TMAR	RW	32	0x0000 004C	0x4803 E04C	0x4808 804C
GPT_TCAR1	R	32	0x0000 0050	0x4803 E050	0x4808 8050
GPT_TSICR	RW	32	0x0000 0054	0x4803 E054	0x4808 8054
GPT_TCAR2	R	32	0x0000 0058	0x4803 E058	0x4808 8058

**22.2.6.2.2 GP Timer Register Description****Table 22-28. GPT\_TIDR**

<b>Address Offset</b>	0x0000 0000																																																																	
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .	<b>Instance</b> GP TIMER																																																																
<b>Description</b>	This read-only register contains the revision number of the module. A write to this register has no effect. This register is used by software to track features, bugs, and compatibility.																																																																	
<b>Type</b>	R																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">REVISION</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	REVISION																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
REVISION																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>																																																																
31: 0	REVISION	IP Revision																																																																
		<b>Type</b> R																																																																
		<b>Reset</b> 0x– <sup>(1)</sup>																																																																

<sup>(1)</sup> TI internal data**Table 22-29. Register Call Summary for Register GPT\_TIDR**

General-Purpose Timers

- [Accessing GP Timer Registers: \[0\] \[1\]](#)
- [GP Timer Register Summary: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)



**Table 22-30. GPT\_TIOCP\_CFG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	This register controls the various parameters of the L4 interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE	EMUFREE	SOFTRESET	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3:2	IDLEMODE	Power management, req/ack control  0x0: Force-idle mode: local target idle state follows (acknowledges) the system idle requests unconditionally, that is, regardless of the IP module internal requirements. Back-up mode, for debug only.  0x1: No-idle mode: local target never enters idle state. Back-up mode, for debug only.  0x2: Smart-idle mode: local target idle state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module should not generate (IRQ- or DMA-request-related) wake-up events.  0x3: Smart-idle wake-up-capable mode: local target idle state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module may generate (IRQ- or DMA-request-related) wake-up events when in idle state. Mode is only relevant if the appropriate IP module <i>wake-up</i> output(s) is (are) implemented.	RW	0x0
1	EMUFREE	Emulation mode  0x0: The timer is frozen in emulation mode (PINSUSPENDN signal active).  0x1: The timer runs free, regardless of PINSUSPENDN value.	RW	0
0	SOFTRESET	Software reset  0x0: Read 0: reset done, no pending action Write 0: No action  0x1: Read 1: initiate software reset Write 1: Reset ongoing	RW	0

**Table 22-31. Register Call Summary for Register GPT\_TIOCP\_CFG**

General-Purpose Timers

- [Power Management of Other GP Timers: \[0\]](#)
- [Wake-Up Capability: \[1\]](#)
- [Software Reset: \[2\] \[3\] \[4\]](#)
- [Timer Under Emulation: \[5\] \[6\] \[7\]](#)
- [Accessing GP Timer Registers: \[8\] \[9\]](#)
- [Writing to Timer Registers: \[10\]](#)
- [GP Timer Module Global Initialization: \[11\] \[12\] \[13\]](#)
- [GP Timer Register Summary: \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)

**Table 22-32. GPT1MS\_TIOCP\_CFG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .	<b>Instance</b>	DMTIMER_DMC1MS
<b>Description</b>	This register controls the various parameters of the OCP interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLOCKACTIVITY		RESERVED	EMUFREE	IDLEMODE	ENAWAKEUP	SOFTRESET	AUTOIDLE								

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0000000
9:8	CLOCKACTIVITY	<p>Clock activity</p> <p>0x0: Functional clock can be switched-off ; L4 clock can be switched-off.</p> <p>0x1: Functional clock can be switched-off ; L4 clock is maintained during wake-up period.</p> <p>0x2: Functional clock is maintained during wake-up period; L4 clock can be switched-off.</p> <p>0x3: Functional clock is maintained during wake-up period; L4 clock is maintained during wake-up period.</p>	RW	0x0
7:6	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0
5	EMUFREE	<p>Emulation mode</p> <p>0x0: Timer counter frozen in emulation</p> <p>0x1: Timer counter free-running in emulation</p>	RW	0
4:3	IDLEMODE	<p>Power management, req/ack control</p> <p>0x0: Force-idle. An idle request is acknowledged unconditionally.</p> <p>0x1: No-idle. An idle request is never acknowledged.</p> <p>0x2: Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module.</p> <p>0x3: Reserved, do not use.</p>	RW	0x0
2	ENAWAKEUP	<p>Wake-up feature global control</p> <p>0x0: No wake-up line assertion in idle mode</p> <p>0x1: Wake-up line assertion enabled in smart-idle mode.</p>	RW	0
1	SOFTRESET	<p>Software reset. This bit is automatically reset by the hardware. During reads, it always returns 0.</p> <p>0x0: Normal mode</p> <p>0x1: The module is reset.</p>	RW	0
0	AUTOIDLE	<p>Internal L4 clock gating strategy</p> <p>0x0: L4 clock is free-running.</p> <p>0x1: Automatic L4 clock gating strategy is applied, based on the L4 interface activity.</p>	RW	0

**Table 22-33. Register Call Summary for Register GPT1MS\_TIOCP\_CFG**

General-Purpose Timers

- [GPTIMER1, GPTIMER2, and GPTIMER10 Power Management: \[0\] \[1\] \[2\]](#)
- [Wake-Up Capability: \[3\] \[4\]](#)
- [Software Reset: \[5\] \[6\]](#)
- [Timer Under Emulation: \[7\] \[8\] \[9\]](#)
- [Accessing GP Timer Registers: \[10\] \[11\]](#)
- [Writing to Timer Registers: \[12\]](#)
- [GP Timer Module Global Initialization: \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [GP Timer Register Summary: \[18\]](#)

**Table 22-34. GPT\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	Component interrupt-request status. Check the corresponding secondary status register. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG													

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_IT_FLAG	IRQ status for Capture Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Trigger IRQ event by software.	RW	0
1	OVF_IT_FLAG	IRQ status for Overflow Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Trigger IRQ event by software.	RW	0
0	MAT_IT_FLAG	IRQ status for Match Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Trigger IRQ event by software	RW	0

**Table 22-35. Register Call Summary for Register GPT\_IRQSTATUS\_RAW**

General-Purpose Timers

- [Accessing GP Timer Registers: \[0\] \[1\]](#)
- [GP Timer Register Summary: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

**Table 22-36. GPT\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	Component interrupt-request status. Check the corresponding secondary status register. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_IT_FLAG	IRQ status for capture Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Clear any pending event.	RW	0
1	OVF_IT_FLAG	IRQ status for Overflow Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Clear any pending event.	RW	0
0	MAT_IT_FLAG	IRQ status for Match Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Clear any pending event.	RW	0

**Table 22-37. Register Call Summary for Register GPT\_IRQSTATUS**

## General-Purpose Timers

- [Wake-Up Capability: \[0\]](#)
- [GP Timer Interrupts: \[1\]](#)
- [Capture Mode Functionality: \[2\] \[3\] \[4\] \[5\]](#)
- [Accessing GP Timer Registers: \[6\] \[7\]](#)
- [Writing to Timer Registers: \[8\]](#)
- [GP Timer Capture Mode: \[9\] \[10\]](#)
- [GP Timer Register Summary: \[11\] \[12\] \[13\] \[14\] \[15\] \[16\]](#)

**Table 22-38. GPT\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	Component interrupt-request enable. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for Compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled Write 1: Set IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for Overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0
0	MAT_EN_FLAG	IRQ enable for Match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0

**Table 22-39. Register Call Summary for Register GPT\_IRQENABLE\_SET**

General-Purpose Timers

- [GP Timer Functional Description: \[0\]](#)
- [GP Timer Interrupts: \[1\]](#)
- [Timer Mode Functionality: \[2\]](#)
- [Capture Mode Functionality: \[3\]](#)
- [Accessing GP Timer Registers: \[4\] \[5\]](#)
- [Writing to Timer Registers: \[6\]](#)
- [GP Timer Mode: \[7\]](#)
- [GP Timer Compare Mode: \[8\]](#)
- [GP Timer Capture Mode: \[9\]](#)
- [GP Timer Register Summary: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

**Table 22-40. GPT\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	Component interrupt-request enable. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for Compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for Overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
0	MAT_EN_FLAG	IRQ enable for Match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0

**Table 22-41. Register Call Summary for Register GPT\_IRQENABLE\_CLR**

## General-Purpose Timers

- [GP Timer Functional Description: \[0\]](#)
- [GP Timer Interrupts: \[1\]](#)
- [Accessing GP Timer Registers: \[2\] \[3\]](#)
- [Writing to Timer Registers: \[4\]](#)
- [GP Timer Register Summary: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

**Table 22-42. GPT\_IRQWAKEEN**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	Wake-up-enabled events taking place when module is idle should generate an asynchronous wake-up.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TCAR_WUP_ENA	OVF_WUP_ENA	MAT_WUP_ENA					

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_WUP_ENA	Wake-up generation for Compare 0x0: Wake-up disabled 0x1: Wake-up enabled	RW	0
1	OVF_WUP_ENA	Wake-up generation for Overflow 0x0: Wake-up disabled 0x1: Wake-up enabled	RW	0
0	MAT_WUP_ENA	Wake-up generation for Match 0x0: Wake-up disabled 0x1: Wake-up enabled	RW	0

**Table 22-43. Register Call Summary for Register GPT\_IRQWAKEEN**

General-Purpose Timers

- [Wake-Up Capability: \[0\]](#)
- [Accessing GP Timer Registers: \[1\] \[2\]](#)
- [Writing to Timer Registers: \[3\]](#)
- [GP Timer Module Global Initialization: \[4\]](#)
- [GP Timer Register Summary: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

**Table 22-44. GPT\_TCLR**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	This register controls optional features specific to the timer functionality.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																GPO_CFG	CAPT_MODE	PT	TRG	TCM	SCPWM	CE	PRE	PTV	AR	ST					

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	GPO_CFG	General-purpose output - this register directly drives the GPTi_PORGPOCFG output pin. For specific use of the GPO_CFG bit, see <a href="#">Section 22.2.2.1, GP Timer External System Interface</a> . 0x0: GPTi_PORGPOCFG drives 0. 0x1: GPTi_PORGPOCFG drives 1.	RW	0
13	CAPT_MODE	Capture mode select bit (first/second) 0x0: Single capture: Capture the first enabled capture event in TCAR1. 0x1: Capture on second event: Capture the second enabled capture event in TCAR2.	RW	0
12	PT	Pulse or toggle mode on GPTi_PWM_out output pin 0x0: Pulse modulation 0x1: Toggle modulation	RW	0
11:10	TRG	Trigger output mode on GPTi_PWM_out output pin 0x0: No trigger 0x1: Trigger on overflow. 0x2: Trigger on overflow and match. 0x3: Reserved	RW	0x0
9:8	TCM	Transition capture mode on GPTi_EVENT_CAPTURE input pin (When the TCM field passed from (00) to any other combination, the TCAR_IT_FLAG and the edge detection logic are cleared.) 0x0: No capture 0x1: Capture on rising edges of GPTi_EVENT_CAPTURE pin 0x2: Capture on falling edges of GPTi_EVENT_CAPTURE pin 0x3: Capture on both edges of GPTi_EVENT_CAPTURE pin	RW	0x0

## General-Purpose Timers

www.ti.com

Bits	Field Name	Description	Type	Reset
7	SCPWM	Pulse width modulation output pin default setting This bit must be set or clear while the timer is stopped or the trigger is off.  0x0: Clear the GPTi_PWM_out output pin and select positive pulse for pulse mode.  0x1: Set the GPTi_PWM_out output pin and select negative pulse for pulse mode.	RW	0
6	CE	Compare enable  0x0: Compare mode is disable.  0x1: Compare mode is enable.	RW	0
5	PRE	Prescaler enable  0x0: The TIMER clock input pin clocks the counter.  0x1: The divided input pin clocks the counter.	RW	0
4:2	PTV	Prescale clock timer value The timer counter is prescaled with the value $2^{(PTV+1)}$ . Example: PTV = 3, counter increases value (if started) after 16 functional clock periods.	RW	0x0
1	AR	Autoreload mode  0x0: One shot timer  0x1: Autoreload timer	RW	0
0	ST	Start/stop timer control  0x0: Stop timer: Only the counter is frozen. If one-shot mode selected (AR =0), this bit is automatically reset by internal logic when the counter is overflowed.  0x1: Start timer	RW	0

**Table 22-45. Register Call Summary for Register GPT\_TCLR**

## General-Purpose Timers

- [Timer Mode Functionality: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Capture Mode Functionality: \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [Compare Mode Functionality: \[16\] \[17\] \[18\] \[19\]](#)
- [Prescaler Functionality: \[20\] \[21\]](#)
- [Pulse-Width Modulation: \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [Timer Counting Rate: \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\]](#)
- [Accessing GP Timer Registers: \[39\]](#)
- [Writing to Timer Registers: \[40\] \[41\]](#)
- [GP Timer Mode: \[42\] \[43\] \[44\] \[45\]](#)
- [GP Timer Compare Mode: \[46\] \[47\] \[48\] \[49\] \[50\]](#)
- [GP Timer Capture Mode: \[51\] \[52\] \[53\] \[54\] \[55\] \[56\] \[57\]](#)
- [GP Timer PWM Mode: \[58\] \[59\] \[60\] \[61\] \[62\] \[63\] \[64\] \[65\] \[66\]](#)
- [GP Timer Register Summary: \[67\] \[68\] \[69\] \[70\] \[71\] \[72\] \[73\]](#)
- [GP Timer Register Description: \[74\] \[75\]](#)

**Table 22-46. GPT\_TCRR**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	GP TIMER																																																																
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .																																																																		
<b>Description</b>	This register holds the value of the internal counter.																																																																		
<b>Type</b>	RW																																																																		
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">TIMER_COUNTER</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	TIMER_COUNTER																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
TIMER_COUNTER																																																																			



Bits	Field Name	Description	Type	Reset
31:0	TIMER_COUNTER	Value of TIMER counter	RW	0x0000 0000

**Table 22-47. Register Call Summary for Register GPT\_TCRR**

General-Purpose Timers

- Wake-Up Capability: [0] [1]
- Timer Mode Functionality: [2] [3] [4] [5] [6] [7] [8] [9] [10]
- 1-ms Tick Generation (Only GPTIMER1, GPTIMER2, and GPTIMER10): [11] [12] [13]
- Capture Mode Functionality: [14] [15]
- Compare Mode Functionality: [16] [17]
- Prescaler Functionality: [18] [19]
- Accessing GP Timer Registers: [20]
- Writing to Timer Registers: [21] [22] [23] [24]
- Reading From Timer Counter Registers: [25]
- GP Timer Mode: [26]
- GP Timer Compare Mode: [27]
- GP Timer Register Summary: [28] [29] [30] [31] [32] [33] [34]
- GP Timer Register Description: [35] [36]

**Table 22-48. GPT\_TLDR**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See Table 22-40.		
<b>Description</b>	This register holds the timer load value.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	LOAD_VALUE	Timer counter value loaded on overflow in autoreload mode or on GPT_TTGR write access. LOAD_VALUE must be different than the timer overflow value (0xFFFF FFFF).	RW	0x0000 0000

**Table 22-49. Register Call Summary for Register GPT\_TLDR**

General-Purpose Timers

- Timer Mode Functionality: [0] [1] [2]
- 1-ms Tick Generation (Only GPTIMER1, GPTIMER2, and GPTIMER10): [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]
- Prescaler Functionality: [13] [14]
- Pulse-Width Modulation: [15] [16] [17]
- Timer Counting Rate: [18] [19] [20] [21] [22]
- Accessing GP Timer Registers: [23]
- Writing to Timer Registers: [24] [25]
- GP Timer Mode: [26]
- GP Timer PWM Mode: [27]
- GP Timer Register Summary: [28] [29] [30] [31] [32] [33] [34]
- GP Timer Register Description: [35] [36]

**Table 22-50. GPT\_TTGR**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	The read value of this register is always 0xFFFF FFFF.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTGR_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	TTGR_VALUE	Writing to the TTGR register causes the <a href="#">GPT_TCRR</a> to be loaded from <a href="#">GPT_TLDR</a> and the prescaler counter to be cleared. Reload is done regardless of the AR field value of the <a href="#">GPT_TCLR</a> register.	RW Rreturns 1s	0xFFFF FFFF

**Table 22-51. Register Call Summary for Register GPT\_TTGR**

## General-Purpose Timers

- [Timer Mode Functionality: \[0\] \[1\]](#)
- [Prescaler Functionality: \[2\]](#)
- [Accessing GP Timer Registers: \[3\]](#)
- [Writing to Timer Registers: \[4\] \[5\]](#)
- [GP Timer Register Summary: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [GP Timer Register Description: \[13\] \[14\]](#)

**Table 22-52. GPT\_TWPS**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	This register contains the write posting bits for all writable functional registers.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																												W_PEND_TMAR	W_PEND_TTGR	W_PEND_TLDR	W_PEND_TCRR	W_PEND_TCLR

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4	W_PEND_TMAR	When equal to 1, a write is pending to the <a href="#">GPT_TMAR</a> register.	R	0
3	W_PEND_TTGR	When equal to 1, a write is pending to the <a href="#">GPT_TTGR</a> register.	R	0
2	W_PEND_TLDR	When equal to 1, a write is pending to the <a href="#">GPT_TLDR</a> register.	R	0
1	W_PEND_TCRR	When equal to 1, a write is pending to the <a href="#">GPT_TCRR</a> register.	R	0
0	W_PEND_TCLR	When equal to 1, a write is pending to the <a href="#">GPT_TCLR</a> register.	R	0

**Table 22-53. Register Call Summary for Register GPT\_TWPS**

- General-Purpose Timers
- [Accessing GP Timer Registers: \[0\]](#)
  - [Writing to Timer Registers: \[1\]](#)
  - [GP Timer Register Summary: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)

**Table 22-54. GPT\_TMAR**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	The compare logic consists of a 32-bit-wide, read/write data TMAR register and logic to compare counter.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMPARE_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	COMPARE_VALUE	Value to be compared to the timer counter	RW	0x0000 0000

**Table 22-55. Register Call Summary for Register GPT\_TMAR**

- General-Purpose Timers
- [Wake-Up Capability: \[0\]](#)
  - [Compare Mode Functionality: \[1\] \[2\] \[3\] \[4\]](#)
  - [Pulse-Width Modulation: \[5\] \[6\] \[7\]](#)
  - [Accessing GP Timer Registers: \[8\]](#)
  - [Writing to Timer Registers: \[9\] \[10\]](#)
  - [GP Timer Compare Mode: \[11\]](#)
  - [GP Timer PWM Mode: \[12\]](#)
  - [GP Timer Register Summary: \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)
  - [GP Timer Register Description: \[20\]](#)

**Table 22-56. GPT\_TCAR1**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	This register holds the first captured value of the counter register.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_VALUE1																															

Bits	Field Name	Description	Type	Reset
31:0	CAPTURE_VALUE1	First timer counter value captured on an external event trigger	R	0x0000 0000

**Table 22-57. Register Call Summary for Register GPT\_TCAR1**

- General-Purpose Timers
- [Capture Mode Functionality: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
  - [Reading From Timer Counter Registers: \[9\]](#)
  - [GP Timer Capture Mode: \[10\]](#)
  - [GP Timer Register Summary: \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)

**Table 22-58. GPT\_TSICR**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	Timer synchronous interface control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																POSTED		SFT		RESERVED											

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	POSTED	Posted mode selection 0x0: Posted mode inactive: Delay the command accept output signal. 0x1: Posted mode active	RW	-
1	SFT	This bit resets all the functional part of the module. 0x0: Software reset is disabled. 0x1: Software reset is enabled.	RW	0
0	RESERVED	Reserved	R	0

**Table 22-59. Register Call Summary for Register GPT\_TSICR**

## General-Purpose Timers

- [Software Reset: \[0\] \[1\]](#)
- [Accessing GP Timer Registers: \[2\] \[3\]](#)
- [Writing to Timer Registers: \[4\] \[5\] \[6\]](#)
- [GP Timer Module Global Initialization: \[7\]](#)
- [GP Timer Register Summary: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

**Table 22-60. GPT\_TCAR2**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	This register holds the second captured value of the counter register.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_VALUE2																															

Bits	Field Name	Description	Type	Reset
31:0	CAPTURE_VALUE2	Second timer counter value captured on an external event trigger	R	0x0000 0000

**Table 22-61. Register Call Summary for Register GPT\_TCAR2**

## General-Purpose Timers

- [Capture Mode Functionality: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Reading From Timer Counter Registers: \[5\]](#)
- [GP Timer Capture Mode: \[6\]](#)
- [GP Timer Register Summary: \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)

22.2.6.2.3 GPTIMER1, GPTIMER2 , and GPTIMER10 Register Description

Table 22-62. GPT\_TISTAT

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	This register provides status information about the module, excluding interrupt status information.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RESETDONE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0.	R	0x0000 0000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset in ongoing. Read 0x1: Reset complete	R	-

Table 22-63. Register Call Summary for Register GPT\_TISTAT

General-Purpose Timers

- [Software Reset: \[0\]](#)
- [Accessing GP Timer Registers: \[1\]](#)
- [GP Timer Register Summary: \[2\]](#)

Table 22-64. GPT\_TISR

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	The timer status register is used to determine which of the timer events requested an interrupt.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads return 0.	R	0x0000 0000
2	TCAR_IT_FLAG	Indicates when an external pulse transition of the correct polarity is detected on external pin GPTi_EVENT_CAPTURE  0x0: No capture interrupt request 0x1: Capture interrupt request	RW	0
1	OVF_IT_FLAG	TCRR overflow  0x0: No overflow interrupt request 0x1: Overflow interrupt pending	RW	0

Bits	Field Name	Description	Type	Reset
0	MAT_IT_FLAG	The compare result of TCRR and TMAR 0x0: No compare interrupt request 0x1: Compare interrupt pending	RW	0

**Table 22-65. Register Call Summary for Register GPT\_TISR**

## General-Purpose Timers

- [Wake-Up Capability: \[0\]](#)
- [GP Timer Interrupts: \[1\]](#)
- [1-ms Tick Generation \(Only GPTIMER1, GPTIMER2, and GPTIMER10\): \[2\]](#)
- [Capture Mode Functionality: \[3\] \[4\] \[5\] \[6\]](#)
- [Accessing GP Timer Registers: \[7\]](#)
- [Writing to Timer Registers: \[8\]](#)
- [GP Timer Capture Mode: \[9\] \[10\]](#)
- [GP Timer Register Summary: \[11\]](#)

**Table 22-66. GPT\_TIER**

<b>Address Offset</b>	0x0000 001C	
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .	<b>Instance</b> GP TIMER
<b>Description</b>	This register controls (enable/disable) the interrupt events.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TCAR_IT_ENA		OVF_IT_ENA		MAT_IT_ENA											

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads return 0.	R	0x0000 0000
2	TCAR_IT_ENA	Capture interrupt enable 0x0: Disable capture interrupt. 0x1: Enable capture interrupt.	RW	0
1	OVF_IT_ENA	Overflow interrupt enable 0x0: Disable overflow interrupt. 0x1: Enable overflow interrupt.	RW	0
0	MAT_IT_ENA	Match interrupt enable 0x0: Disable match interrupt. 0x1: Enable match interrupt.	RW	0

**Table 22-67. Register Call Summary for Register GPT\_TIER**

- General-Purpose Timers
- [GP Timer Functional Description: \[0\]](#)
  - [GP Timer Interrupts: \[1\]](#)
  - [Timer Mode Functionality: \[2\]](#)
  - [Capture Mode Functionality: \[3\]](#)
  - [Compare Mode Functionality: \[4\]](#)
  - [Accessing GP Timer Registers: \[5\]](#)
  - [Writing to Timer Registers: \[6\]](#)
  - [GP Timer Mode: \[7\]](#)
  - [GP Timer Compare Mode: \[8\]](#)
  - [GP Timer Capture Mode: \[9\]](#)
  - [GP Timer Register Summary: \[10\]](#)

**Table 22-68. GPT\_TWER**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	GP TIMER
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .		
<b>Description</b>	This register controls (enable/disable) the wake-up feature on specific interrupt events.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TCAR_WUP_ENA		OVF_WUP_ENA		MAT_WUP_ENA											

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads return 0	R	0x0000 0000
2	TCAR_WUP_ENA	Capture wake-up enable 0x0: Disable capture wake-up. 0x1: Enable capture wake-up.	RW	0
1	OVF_WUP_ENA	Overflow wake-up enable 0x0: Disable overflow wake-up. 0x1: Enable overflow wake-up.	RW	0
0	MAT_WUP_ENA	Match wake-up enable 0x0: Disable match wake-up. 0x1: Enable match wake-up.	RW	0

**Table 22-69. Register Call Summary for Register GPT\_TWER**

- General-Purpose Timers
- [GP Timer Functional Description: \[0\]](#)
  - [Wake-Up Capability: \[1\] \[2\]](#)
  - [Accessing GP Timer Registers: \[3\]](#)
  - [Writing to Timer Registers: \[4\]](#)
  - [GP Timer Module Global Initialization: \[5\]](#)
  - [GP Timer Register Summary: \[6\]](#)

**Table 22-70. GPT\_TPIR**

<b>Address Offset</b>	0x0000 0048
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .
<b>Instance</b>	DMTIMER_DMC1MS
<b>Description</b>	This register is used for 1-ms tick generation. The TPIR register holds the value of the positive increment. The value of this register is added to the value of TCVR to determine whether next value loaded in TCRR is the subperiod value or the overperiod value.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSITIVE_INC_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	POSITIVE_INC_VALUE	Value of the positive increment	RW	0x0000 0000

**Table 22-71. Register Call Summary for Register GPT\_TPIR**

## General-Purpose Timers

- [1-ms Tick Generation \(Only GPTIMER1, GPTIMER2, and GPTIMER10\): \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Accessing GP Timer Registers: \[6\]](#)
- [Writing to Timer Registers: \[7\] \[8\]](#)
- [GP Timer Register Summary: \[9\]](#)

**Table 22-72. GPT\_TNIR**

<b>Address Offset</b>	0x0000 004C
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .
<b>Instance</b>	DMTIMER_DMC1MS
<b>Description</b>	This register is used for 1-ms tick generation. The TNIR register holds the value of the negative increment. The value of this register is added to the value of the TCVR to determine whether next value loaded in TCRR is the subperiod value or the overperiod value.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEGATIVE_INV_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	NEGATIVE_INV_VALUE	Value of the negative increment	RW	0x0000 0000

**Table 22-73. Register Call Summary for Register GPT\_TNIR**

## General-Purpose Timers

- [1-ms Tick Generation \(Only GPTIMER1, GPTIMER2, and GPTIMER10\): \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Accessing GP Timer Registers: \[6\]](#)
- [Writing to Timer Registers: \[7\] \[8\]](#)
- [GP Timer Register Summary: \[9\]](#)



**Table 22-74. GPT\_TCVR**

<b>Address Offset</b>	0x0000 0050
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .
<b>Instance</b>	DMTIMER_DMC1MS
<b>Description</b>	This register is used for 1-ms tick generation. The TCVR register determines whether next value loaded in TCRR is the subperiod value or the overperiod value.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER_VALUE	Value of CVR counter	RW	0x0000 0000

**Table 22-75. Register Call Summary for Register GPT\_TCVR**

General-Purpose Timers

- [1-ms Tick Generation \(Only GPTIMER1, GPTIMER2, and GPTIMER10\): \[0\] \[1\]](#)
- [Accessing GP Timer Registers: \[2\]](#)
- [Writing to Timer Registers: \[3\] \[4\]](#)
- [GP Timer Register Summary: \[5\]](#)

**Table 22-76. GPT\_TOCR**

<b>Address Offset</b>	0x0000 0054
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .
<b>Instance</b>	DMTIMER_DMC1MS
<b>Description</b>	This register is used to mask the tick interrupt for a selected number of ticks.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OVF_COUNTER_VALUE																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads return 0.	R	0x00
23:0	OVF_COUNTER_VALUE	Number of overflow events	RW	0x000000

**Table 22-77. Register Call Summary for Register GPT\_TOCR**

General-Purpose Timers

- [1-ms Tick Generation \(Only GPTIMER1, GPTIMER2, and GPTIMER10\): \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Accessing GP Timer Registers: \[5\]](#)
- [Writing to Timer Registers: \[6\] \[7\]](#)
- [GP Timer Register Summary: \[8\]](#)

**Table 22-78. GPT\_TOWR**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Table 22-40</a> .	<b>Instance</b>	DMTIMER_DMC1MS
<b>Description</b>	This register holds the number of masked overflow interrupts.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OVF_WRAPPING_VALUE																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads return 0.	R	0x00
23:0	OVF_WRAPPING_VALUE	Number of masked interrupts	RW	0x000000

**Table 22-79. Register Call Summary for Register GPT\_TOWR**

## General-Purpose Timers

- [1-ms Tick Generation \(Only GPTIMER1, GPTIMER2, and GPTIMER10\): \[0\] \[1\] \[2\]](#)
- [Accessing GP Timer Registers: \[3\]](#)
- [Writing to Timer Registers: \[4\] \[5\]](#)
- [GP Timer Register Summary: \[6\]](#)

## 22.3 Watchdog Timers

### 22.3.1 Watchdog Timers Overview

The device includes two instances of the 32-bit watchdog timer: WDT2 and WDT3. Figure 22-15 shows how each timer is connected in the device.

**NOTE:** WDT<sub>i</sub> (where *i* is the watchdog timer instance: *i* = 2 or 3) stands for the following:

- WDT2: Watchdog timer 2, also called MPU watchdog timer
- WDT3: Watchdog timer 3, also called IVA3 watchdog timer

Each watchdog timer is an upward counter capable of generating a pulse on the reset pin and an interrupt to the device system modules following an overflow condition. The MPU watchdog timer serves resets to the PRCM module (its interrupt outputs are unused), and the IVA3 watchdog timer serves watchdog interrupts to the MPU (its reset outputs are unused).

The watchdog timers can be accessed, loaded, and cleared by registers through the L4 interface. The MPU and IVA3 watchdog timers have the 32-kHz clock for their timer clock input.

The MPU watchdog timer directly generates a warm reset condition on overflow. The IVA3 watchdog timer generates an MPU interrupt condition on overflow, which can be used by the application software through the PRCM module to indirectly trigger a reset condition (that is, to the IVA3 subsystem).

The MPU watchdog timer connects to a single target agent port on the L4 interconnect.

**Figure 22-15. Watchdog Timers Block Diagram**

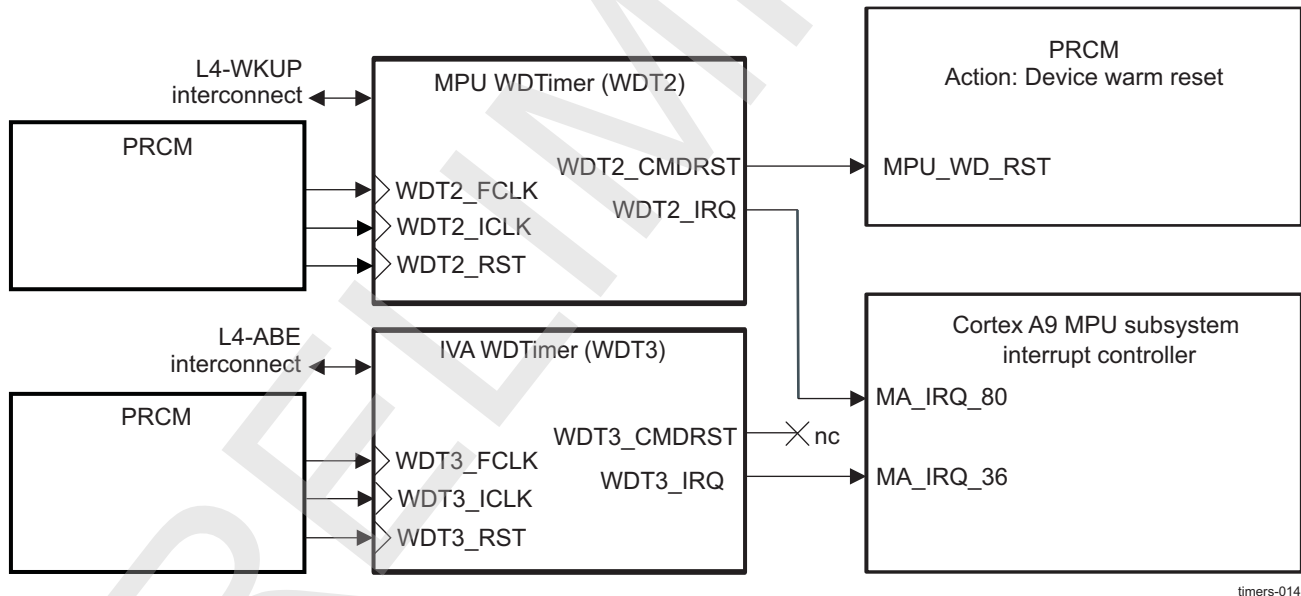


Table 22-80 lists the default state of the watchdog timers in the device.

**Table 22-80. Watchdog Timers Default State**

Timer	Default State	
MPU WDT2	Enabled	Not running
IVA3 WDT3	Enabled	Not running

---

**NOTE:** The default state of the watchdog timers described in [Table 22-80](#) is considered to be their state immediately after ROM code execution. For more information, see [Chapter 27, Initialization](#).

---

### 22.3.1.1 Watchdog Timers Features

The main features of the watchdog timer controllers are:

- L4 slave interface support:
  - 32-bit data bus width
  - 32-/16-bit access supported
  - 8-bit access not supported
  - 11-bit address bus width
  - Burst mode not supported
  - Write nonposted transaction mode only
- Free-running 32-bit upward counter
- Programmable divider clock source ( $2^n$  where  $n = [0:7]$ )
- On-the-fly read/write register (while counting)
- Subset programming model of the GP timer
- The watchdog timers are reset either on power-on or after a warm reset before they start counting.
- Reset or interrupt actions when a timer overflow condition occurs
- The watchdog timer generates a reset or an interrupt in its hardware integration (WDT2 or WDT3).

### 22.3.2 Watchdog Timer Environment

The watchdog timers are accessible through the L4 interface.

### 22.3.3 Watchdog Timer Integration

Figure 22-16 shows the integration of the watchdog timers in the device.

Figure 22-16. Watchdog Timers Integration

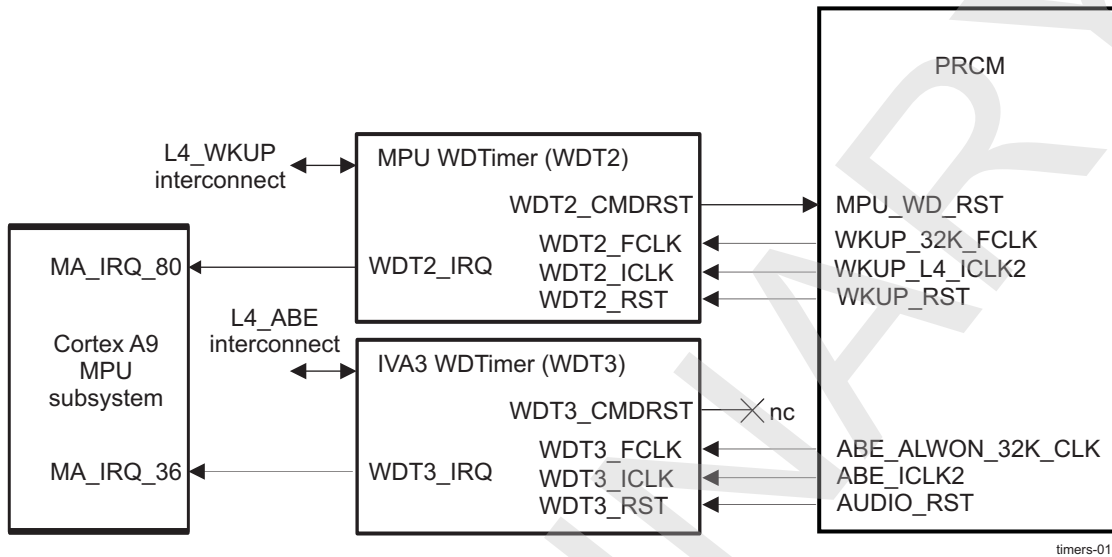


Table 22-81 through Table 22-83 summarize the integration of the module in the device.

Table 22-81. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
WDT2	PD_WKUP	Yes	L4_WKUP
WDT3	PD_AUDIO	Yes	L4_ABE

Table 22-82. Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
WDT2	WDT2_FCLK	WKUP_32K_FCLK	PRCM	WDT2 functional clock
WDT2	WDT2_ICLK	WKUP_L4_ICLK2	PRCM	WDT2 interface clock
WDT3	WDT3_FCLK	ABE_ALWON_32K_CLK	PRCM	WDT3 functional clock
WDT3	WDT3_ICLK	ABE_ICLK2	PRCM	WDT3 interface clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
WDT2	WDT2_RST	WKUP_RST	PRM	Reset to WDT2
WDT2	MPU_WD_RST	WDT2_CMDRST	WDT2	Reset to PRCM
WDT3	WDT3_RST	AUDIO_RST	PRM	Reset to WDT3

**NOTE:** WDT2 is reset on power on or after a warm reset before it starts counting.

WDT3 is reset on power on or after a warm reset, and then it does not start counting.

**Table 22-83. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
WDT2	WDT2_IRQ	MA_IRQ_80	Cortex-A9 MPU	WDT2 interrupt to Cortex-A9 MPU
WDT3	WDT3_IRQ	MA_IRQ_36	Cortex-A9 MPU	IVA3 watchdog overflow
No DMA Requests				

**NOTE:**

- See [Section 22.3.4.2, Interrupts](#), for interrupt source description.

## 22.3.4 Watchdog Timers Functional Description

### 22.3.4.1 Power Management

There are two clock domains in the watchdog timers:

- Functional clock domain: WDTi\_FCLK is the watchdog timer functional clock. It is used to clock the watchdog timer internal logic.
- Interface clock domain: WDTi\_ICLK is the watchdog timer interface clock. It is used to synchronize the watchdog timer L4 port to the L4 interconnect. All accesses from the interconnect are synchronous to WDTi\_ICLK.

Table 22-82 lists the source clocks for each watchdog timer in the device. For more information about clock control and domains, see Section 3.1.1.1, *Clock Management* in Chapter 3, *Power, Reset, and Clock Management*.

From a global system power-management perspective, when one or both of the watchdog timer clocks is no longer required, the watchdog timers can be deactivated at the PRCM module level in the corresponding registers.

At the PRCM module level, when the conditions to shut off the PRCM module functional or interface output clocks are met (for more information, see the Section 3.1.1.1.4, *Clock Domain Level Clock Management*), the PRCM module automatically launches a hardware handshake protocol to ensure the watchdog timer is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the watchdog timer.

Although this handshake is a hardware function and out of software control, the way the watchdog timer acknowledges the PRCM IDLE request is configurable through the WDT\_WDSC[4:3] IDLEMODE bit field. Table 22-84 lists the settings and related acknowledgment modes of the IDLEMODE bit field.

**Table 22-84. IDLEMODE Settings**

IDLEMODE Value	Selected Mode	Description
00	Force-idle	The watchdog timer unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully, because it does not prevent loss of data when the clock is switched off.
01	No-idle	The watchdog timer never acknowledges an IDLE request from the PRCM module. This mode is safe from a module point of view, because it ensures that the clocks remain active. It is not efficient from a power-saving perspective, however, because it does not allow the PRCM module output clock to be shut off and thus the power domain to be set to a lower power state.
10	Smart-idle	The watchdog timer acknowledges the IDLE request, basing its decision on its internal activity. The acknowledge signal is asserted only when all pending transactions and IRQ requests are treated. This is the best approach for efficient system power management.
11	Smart-idle wakeup-capable mode	The watchdog timer acknowledges the IDLE request, basing its decision on its internal activity. The timer generates (IRQ-request-related) wake-up events when in IDLE state if the WDT_WIRQWAKEEN[1:0] bits are set to 1.

#### 22.3.4.1.1 Wake-Up Capability

If the WDT\_WDSC[4:3] IDLEMODE bit field sets smart-idle wakeup-capable mode (= 0 × 3), the timer evaluates its internal capability to have the interface clock switched off. When there is no more internal activity (no pending interrupt sources: match, overflow, or timer capture events), the idle acknowledge signal is asserted and the timer enters into sleep mode, ready to issue a wake-up request. This wake-up request is sent only if the WDT\_WIRQWAKEEN[0] OVF\_WK\_ENA and/or the WDT\_WIRQWAKEEN[1] DLY\_WK\_ENA bits enable the overflow and/or the delay wake-up capability.

#### 22.3.4.2 Interrupts

Table 22-85 list the event flags, and their masks, that cause module interrupts.

**Table 22-85. Watchdog Timer Events**

Event Flag	Event Mask	Mapping	Comments
WDT_WIRQSTAT[0] EVENT_OVF	WDT_WIRQENSET/WDT_WIRQENCLR[0] OVF_IT_ENA	WDTi_IRQ	Watchdog timer overflow
WDT_WIRQSTAT[1] EVENT_DLY	WDT_WIRQENSET/WDT_WIRQENCLR[1] DLY_IT_ENA	WDTi_IRQ	Watchdog delay value reached

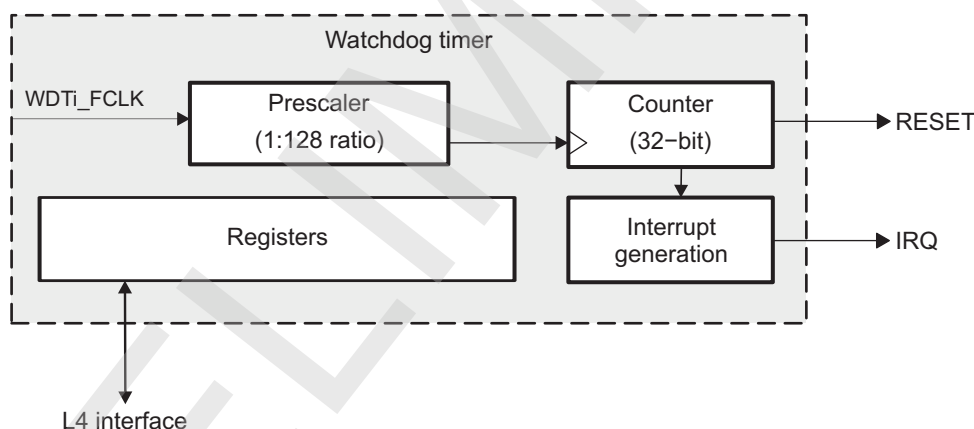
### 22.3.4.3 General Watchdog Timer Operation

The watchdog timers are based on an upward 32-bit counter coupled with a prescaler. The counter overflow is signaled through two independent signals: a simple reset signal and an interrupt signal, both active low. The use of these signals depends on whether they are connected or not. For this information, see [Figure 22-15](#). The interrupt generation mechanism is controlled through the [WDT\\_WIRQENSET/WDT\\_WIRQENCLR](#) and [WDT\\_WIRQSTAT](#) registers.

The prescaler ratio can be set from 1 to 128 by accessing the [WDT\\_WCLR\[4:2\]](#) PTV bit field and the [WDT\\_WCLR\[5\]](#) PRE bit of the watchdog control register ([WDT\\_WCLR](#)).

The current timer value can be accessed on-the-fly by reading the watchdog timer counter register ([WDT\\_WCRR](#)), modified by accessing the watchdog timer load register ([WDT\\_WLDR](#)) (no on-the-fly update), or reloaded by following a specific reload sequence on the watchdog timer trigger register ([WDT\\_WTGR](#)). A start/stop sequence applied to the watchdog timer start/stop register ([WDT\\_WSPR](#)) can start and stop the watchdog timers.

[Figure 22-17](#) is a functional block diagram of the watchdog timer.

**Figure 22-17. 32-Bit Watchdog Timer Functional Block Diagram**

timers-016

### 22.3.4.4 Reset Context

The watchdog timers are enabled after reset. [Table 22-86](#) lists the default reset values of the two watchdog timer load registers (the [WDT\\_WLDR](#)) and prescaler ratios (the [WDT\\_WCLR\[4:2\]](#) PTV bit field). To get these values, software must read the corresponding [WDT\\_WCLR\[4:2\]](#) PTV bit field and the 32-bit register to retrieve the static configuration of the module.

**Table 22-86. Count and Prescaler Default Reset Values**

Timer	WDT_WLDR Reset Value	PTV Reset Value
MPU WDT (WDT2)	0xFFFB 0000	0
IVA3 WDT (WDT3)	0xFFFB 0000	0

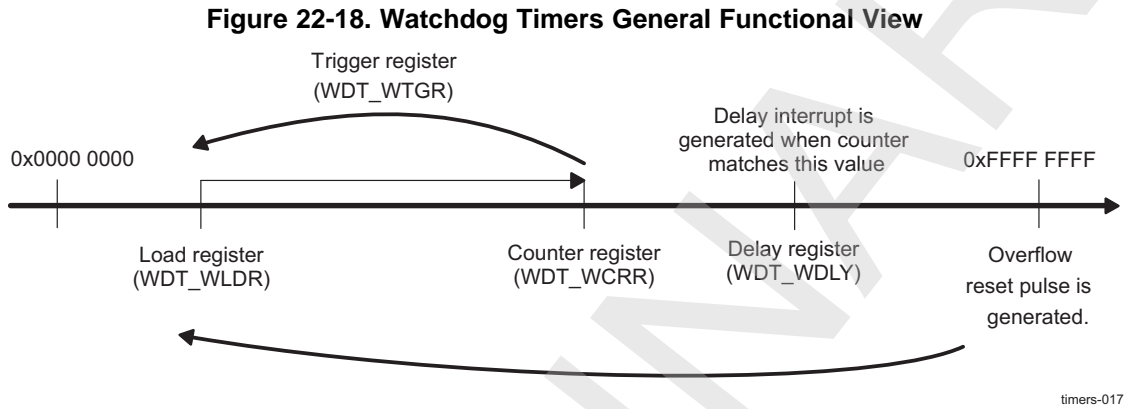


### 22.3.4.5 Overflow/Reset Generation

When the watchdog timer counter register ([WDT\\_WCRR](#)) overflows, an active-low reset pulse is generated to the PRCM module. This pulse is one prescaled timer clock cycle wide and occurs at the same time as the timer counter overflow.

After reset generation, the counter is automatically reloaded with the value stored in the watchdog load register ([WDT\\_WLDR](#)) and the prescaler is reset (the prescaler ratio remains unchanged). When the reset pulse output is generated, the timer counter begins incrementing again.

Figure 22-18 shows a general functional view of the watchdog timers.



### 22.3.4.6 Prescaler Value/Timer Reset Frequency

Each watchdog timer is composed of a prescaler stage and a timer counter.

The timer rate is defined by the following values:

- Value of the prescaler fields (the [WDT\\_WCLR\[5\]](#) PRE bit and the [WDT\\_WCLR\[4:2\]](#) PTV bit field)
- Value loaded into the timer load register ([WDT\\_WLDR](#))

The prescaler stage is clocked with the timer clock and acts as a clock divider for the timer counter stage. The ratio is managed by accessing the ratio definition field (the [WDT\\_WCLR\[4:2\]](#) PTV bit field) and is enabled with the [WDT\\_WCLR\[5\]](#) PRE bit.

Table 22-87 lists the prescaler clock ratio values.

**Table 22-87. Prescaler Clock Ratio Values**

<a href="#">WDT_WCLR[5]</a> PRE	<a href="#">WDT_WCLR[4:2]</a> PTV	Clock Divider (PS)
0	X	1
1	0	1
1	1	2
1	2	4
1	3	8
1	4	16
1	5	32
1	6	64
1	7	128

Thus the watchdog timer overflow rate is expressed as:

$$\text{OVF\_Rate} = (0\text{x}\text{FFFF FFFF} - \text{WDT\_WLDR} + 1) \times (\text{wd-functional clock period}) \times \text{PS}$$

where wd-functional clock period =  $1/(\text{wd-functional clock frequency})$  and  $\text{PS} = 2^{(\text{PTV})}$

#### CAUTION

Internal resynchronization causes some latency in any software write to [WDT\\_WSPR](#) before [WDT\\_WSPR](#) is updated with the programmed value:

$1.5 \times \text{functional clock cycles} \leq \text{write\_WDT\_WSPR\_latency} \leq 2.5 \times \text{functional clock cycles}$

Remember to consider this latency whenever the watchdog timer must be started or stopped.

For example, for a timer clock input of 32 kHz with a prescaler ratio value of 0x1 (clock divided by 2) and [WDT\\_WCLR](#)[5] PRE = 1 (clock divider enabled), the reset period is as listed in [Table 22-88](#).

**Table 22-88. Reset Period Examples**

<a href="#">WDT_WLDR</a> Value	Reset Period
0x0000 0000	74 h 56 min
0xFFFF 0000	4 s
0xFFFF FFF0	1 ms
0xFFFF FFFF	62.5 us

#### CAUTION

- Ensure that the reloaded value allows the correct operation of the application. When a watchdog timer is enabled, software must periodically trigger a reload before the counter overflows. Hence, the value of the [WDT\\_WLDR](#)[31:0] bit field must be chosen according to the ongoing activity preceding the watchdog reload.
- Due to design reasons, [WDT\\_WLDR](#)[31:0] = 0xFFFF FFFF is a special case, although such a value of [WDT\\_WLDR](#) is meaningless. When [WDT\\_WLDR](#) is programmed with the overflow value, a triggering event generates a reset/interrupt one functional clock cycle later, even if the watchdog timer is stopped.

[Table 22-89](#) lists the default reset periods for the watchdog timers.

**Table 22-89. Default Watchdog Timer Reset Periods**

Watchdog Timers	Clock Source	Default Reset Period
MPU/IVA3 WDTs	32 kHz	10 s

### 22.3.4.7 Triggering a Timer Reload

To reload the timer counter and reset the prescaler before reaching overflow, a reload command is executed by accessing the watchdog timer trigger register ([WDT\\_WTGR](#)) using a specific reload sequence.

The specific reload sequence is performed whenever the written value on the [WDT\\_WTGR](#) register differs from its previous value. In this case, reload is executed in the same way as an overflow autoreload, but without the generation of a reset pulse.

The timer counter is loaded with the value of the watchdog timer load register (the [WDT\\_WLDR\[31:0\] TIMER\\_LOAD](#) bit field), and the prescaler is reset.

#### 22.3.4.8 Start/Stop Sequence for Watchdog Timers (Using the WDT\_WSPR Register)

To start and stop a watchdog timer, access must be made through the start/stop register ([WDT\\_WSPR](#)) using a specific sequence.

To disable the timer, follow this sequence:

1. Write 0xXXXX AAAA in [WDT\\_WSPR](#).
2. Write 0xXXXX 5555 in [WDT\\_WSPR](#).

To enable the timer, follow this sequence:

1. Write 0xXXXX BBBB in [WDT\\_WSPR](#).
2. Write 0xXXXX 4444 in [WDT\\_WSPR](#).

All other write sequences on the [WDT\\_WSPR](#) register have no effect on the start/stop feature of the module.

#### 22.3.4.9 Modifying Timer Count/Load Values and Prescaler Setting

To modify the timer counter value (the [WDT\\_WCRR](#) register), prescaler ratio (the [WDT\\_WCLR\[4:2\] PTV](#) bit field), delay configuration value (the [WDT\\_WDLY\[31:0\] DLY\\_VALUE](#) bit field), or the load value (the [WDT\\_WLDR\[31:0\] TIMER\\_LOAD](#) bit field), the watchdog timer must be disabled by using the start/stop sequence (the [WDT\\_WSPR](#) register).

After a write access, the load register value and prescaler ratio registers are updated immediately, but new values are considered only after the next consecutive counter overflow or after a new trigger command (the [WDT\\_WTGR](#) register).

#### 22.3.4.10 Watchdog Counter Register Access Restriction (WDT\_WCRR Register)

A 32-bit shadow register is implemented to read a coherent value of the [WDT\\_WCRR](#) register because the [WDT\\_WCRR](#) register is directly related to the timer counter value and is updated on the timer clock ([WDT\\_FCLK](#)). The shadow register is updated by a 16-bit LSB read command.

---

**NOTE:** Although the L4 clock ([WDT\\_ICLK](#)) is completely asynchronous with the timer clock ([WDT\\_FCLK](#)), some synchronization is performed to ensure that the value of the [WDT\\_WCRR](#) register is not read while it is being incremented.

---

When 32-bit read access is performed, the shadow register is not updated. Read access is performed directly from the accessed register.

To ensure that a coherent value is read inside [WDT\\_WCRR](#), the first read access is to the lower 16 bits (offset = 0x08), followed by read access to the upper 16 bits (offset = 0x0A).

#### 22.3.4.11 Watchdog Timer Interrupt Generation

When an interrupt source occurs, the interrupt status bit (the [WDT\\_WIRQSTAT\[0\] EVENT\\_OVF](#) or [WDT\\_WIRQSTAT\[1\] EVENT\\_DLY](#) bit) is set to 1. The output interrupt line ([WDTi\\_IRQ](#)) is asserted (active low) when status (the [EVENT\\_XXX](#) bit) and enable (the [XXX\\_IT\\_ENA](#) bit) flags are set to 1; the order is not relevant. Writing 1 to the enable bit (the status is already set at 1) also triggers the interrupt in the normal order (enable first, status next). The pending interrupt event is cleared when the set status bit is overwritten by a value of 1 by a write command in the [WDT\\_WIRQSTAT](#) register. Reading the [WDT\\_WIRQSTAT](#) register and writing the value back allows a fast interrupt acknowledge process.

The watchdog timer issues an overflow interrupt if this interrupt is enabled in the watchdog interrupt enable register ([WDT\\_WIRQENSET\[0\] OVF\\_IT\\_ENA](#) = 1). When the overflow occurs, the interrupt status bit (the [WDT\\_WIRQSTAT\[0\] EVENT\\_OVF](#) bit) is set to 1. The output interrupt line ([WDT\\_IRQ](#)) is asserted (active low) when status ([EVENT\\_OVF](#)) and enable ([OVF\\_IT\\_ENA](#)) flags are set to 1; the order is not relevant. This interrupt can be disabled by setting the [WDT\\_WIRQENCLR\[0\] OVF\\_IT\\_ENA](#) bit to 1.

The watchdog can issue the delay interrupt if this interrupt is enabled in the interrupt enable register ([WDT\\_WIRQENSET\[1\] DLY\\_IT\\_ENA = 1](#)). When the counter is running and the counter value matches the value stored in the delay configuration register ([WDT\\_WDLY](#)), the corresponding interrupt status bit is set in the watchdog status register ([WDT\\_WIRQSTAT](#)) and the output interrupt line is asserted (active low) when the flag ([EVENT\\_DLY](#)) and enable ([DLY\\_IT\\_ENA](#)) bits are 1 in the [WDT\\_WIRQSTAT](#) and [WDT\\_WIRQENSET](#) registers, respectively; the order (normally enable, then flag), is not relevant. This interrupt can be disabled by setting the [WDT\\_WIRQENCLR\[1\] DLY\\_IT\\_ENA](#) bit to 1.

---

**NOTE:** Writing 0 to the [WDT\\_WIRQSTAT\[0\] EVENT\\_OVF](#) bit or the [WDT\\_WIRQSTAT\[1\] EVENT\\_DLY](#) bit has no effect.

---

The two clock domains are resynchronized because the interrupt event is generated on the functional clock domain ([WDTi\\_FCLK](#)) during the updating of the interrupt status register ([WDT\\_WIRQSTAT](#)).

The [WDT\\_WDLY](#) register is used to specify the value of the delay configuration register. The delay time to interrupt is the difference between the reload value stored in the counter load register ([WDT\\_WLDR](#)) and the programmed value in this register ([WDT\\_WDLY](#)).

Use the following formula to estimate the delay time:

Delay time period = ([WDT\\_WDLY](#) – [WDT\\_WLDR](#) + 1) × Timer clock period × Clock divider

Where:

- Timer clock period = 1/(Timer clock frequency)
- Clock divider = 2PTV

If the counter value ([WDT\\_WCRR](#)) reaches the programmed value ([WDT\\_WDLY](#)), the status bit ([EVENT\\_DLY](#)) gets set in the interrupt status register ([WDT\\_WIRQSTAT](#)), and an interrupt occurs if the corresponding enable bit is set in the interrupt enable register ([WDT\\_WIRQENSET](#)).

#### CAUTION

If the reload event occurs (after a triggering sequence or after a reset sequence) before reaching the programmed value ([WDT\\_WDLY\[31:0\] WDLY\\_VALUE](#)), no interrupt is generated.

Also, no interrupt is generated if the value programmed in the delay configuration register ([WDT\\_WDLY](#)) is less than the value stored in the counter load register ([WDT\\_WLDR](#)).

#### 22.3.4.12 Watchdog Timers Under Emulation

During emulation mode, the watchdog timer can/cannot continue running, according to the value of the [WDT\\_WDSC\[5\] EMUFREE](#) bit of the system configuration register ([WDT\\_WDSC](#)).

- When [EMUFREE](#) is 1, watchdog timer execution is not stopped and a reset pulse is still generated when overflow is reached.
- When [EMUFREE](#) is 0, the counters (prescaler/timer) are frozen and incrementation restarts after exiting from emulation mode.

#### 22.3.4.13 Accessing Watchdog Timer Registers

Posted/nonposted selection applies only to functional registers that require synchronization on/from the timer functional clock domain ([WDTi\\_FCLK](#)). For write/read operation, the following registers are affected:

- [WDT\\_WCLR](#)
- [WDT\\_WCRR](#)
- [WDT\\_WLDR](#)
- [WDT\\_WTGR](#)
- [WDT\\_WDLY](#)
- [WDT\\_WSPR](#)

The timer interface clock domain synchronous registers are not affected by the posted/nonposted selection; the write/read operation is effective and acknowledged (command accepted) after one WDT\_ICLK cycle from the command assertion. The timer interface clock domain synchronous registers are:

- [WDT\\_WIDR](#)
- [WDT\\_WDSC](#)
- [WDT\\_WDST](#)
- [WDT\\_WIRQSTATRAW](#)
- [WDT\\_WIRQSTAT](#)
- [WDT\\_WIRQENSET](#)
- [WDT\\_WIRQENCLR](#)
- [WDT\\_WIRQWAKEEN](#)
- [WDT\\_WWPS](#)

---

**NOTE:** Accesses to WDT2 and WDT3 are posted.

---

### 22.3.5 Watchdog Timer Low-Level Programming Model

This section covers the low-level hardware programming sequences for configuration and use of the module.

#### 22.3.5.1 Global Initialization

##### 22.3.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the watchdog timer is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the watchdog timer (see [Table 22-90](#)). For more information, see [Section 22.3.2, Watchdog Timer Environment](#), and [Section 22.3.3, Watchdog Timer Integration](#).

**Table 22-90. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. for more information about the module configuration. For more information about the module configuration, see <a href="#">Section 3.1.1.1.2, Module Level Clock Management</a> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Control module	Module-specific pad muxing must be set in the control module. For more information about the module configuration, see <a href="#">Section 18.4.8, PAD Functional Multiplexing and Configuration</a> in <a href="#">Chapter 18, Control Module</a> .
MPU INTC	The MPU INTC configuration must be performed to enable the interrupts from the watchdog timer. See <a href="#">, Interrupt Controller Registers</a> in <a href="#">Chapter 4, Cortex-A9 MPU Subsystem</a> .

### 22.3.5.1.2 Watchdog Timer Module Global Initialization

#### 22.3.5.1.2.1 Main Sequence – Watchdog Timer Module Global Initialization

Table 22-91 lists the steps for initializing the watchdog timer module when the module is to be used for the first time.

**Table 22-91. Watchdog Timer Module Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Execute software reset.	WDT_WDSC[1] SOFTRESET	0x1
Wait until reset release?	WDT_WDSC[1] SOFTRESET	0x0
Configure idle mode.	WDT_WDSC[4:3] IDLEMODE	xx
Enable delay wakeup.	WDT_WIRQWAKEEN[1] DLY_WK_ENA	0x1
Enable overflow wakeup.	WDT_WIRQWAKEEN[0] OVF_WK_ENA	0x1
Enable delay interrupt.	WDT_WIRQENSET[1] ENABLE_DLY	0x1
Enable overflow interrupt.	WDT_WIRQENSET[0] ENABLE_OVF	0x1

### 22.3.5.2 Operational Mode Configuration

#### 22.3.5.2.1 Watchdog Timer Basic Configuration

##### 22.3.5.2.1.1 Main Sequence – Watchdog Timer Basic Configuration

Table 22-92 lists the steps for the basic configuration of the watchdog timer.

**Table 22-92. Watchdog Timer Basic Configuration**

Step	Register/Bit Field/Programming Model	Value
Disable the watchdog timer.	See Section 22.3.5.2.1.2.	
Set prescaler value.	WDT_WCLR[4:2] PTV	xxx
Enable prescaler.	WDT_WCLR[5] PRE	0x1
Load delay configuration value.	WDT_WDLY	xxx
Load timer counter value.	WDT_WCRR	xxx
Enable the watchdog timer.	See Section 22.3.5.2.1.3.	

##### 22.3.5.2.1.2 Subsequence – Disable the Watchdog Timer

Table 22-93 lists the steps to disable the watchdog timer.

**Table 22-93. Disable the Watchdog Timer**

Step	Register/Bit Field/Programming Model	Value
Write disable sequence Data1.	WDT_WSPR	0xXXXX AAAA
Write disable sequence Data2.	WDT_WSPR	0xXXXX 5555

##### 22.3.5.2.1.3 Subsequence – Enable the Watchdog Timer

Table 22-94 lists the steps to enable the watchdog timer.

**Table 22-94. Enable the Watchdog Timer**

Step	Register/Bit Field/Programming Model	Value
Write enable sequence Data1.	WDT_WSPR	0xXXXX BBBB
Write enable sequence Data2.	WDT_WSPR	0xXXXX 4444

## 22.3.6 Watchdog Timer Register Manual

### 22.3.6.1 Watchdog Timer Instance Summary

Table 22-95 lists the base address and address space for the watchdog timer module instances.

**Table 22-95. Watchdog Timer Instance Summary**

Module Name	Base Address L4 Interconnect	Base Address L3 Interconnect	Base Address Cortex-A9 Private Access	Base Address DSP Private Access	Size
WDTIMER2	0x4A31 4000	-	-	-	4KB
WDTIMER3	-	0x4903 0000	0x4013 0000	0x01D3 0000	4KB

**NOTE:** Private access is an access which is not using the L3/L4 interconnects.

### 22.3.6.2 Watchdog Timer Registers

#### 22.3.6.2.1 Watchdog Timer Register Summary

**CAUTION**

The watchdog timers registers are limited to 32-bit and 16-bit data accesses; 8-bit access is not allowed and can corrupt register content.

Table 22-96 lists the WDTIMER2 registers.

**Table 22-96. WDTIMER2 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L4 Interconnect
WDT_WIDR	R	32	0x0000 0000	0x4A31 4000
WDT_WDSC	RW	32	0x0000 0010	0x4A31 4010
WDT_WDST	R	32	0x0000 0014	0x4A31 4014
WDT_WISR	RW	32	0x0000 0018	0x4A31 4018
WDT_WIER	RW	32	0x0000 001C	0x4A31 401C
WDT_WWER	RW	32	0x0000 0020	0x4A31 4020
WDT_WCLR	RW	32	0x0000 0024	0x4A31 4024
WDT_WCRR	RW	32	0x0000 0028	0x4A31 4028
WDT_WLDR	RW	32	0x0000 002C	0x4A31 402C
WDT_WTGR	RW	32	0x0000 0030	0x4A31 4030
WDT_WWPS	R	32	0x0000 0034	0x4A31 4034
WDT_WDLY	RW	32	0x0000 0044	0x4A31 4044
WDT_WSPR	RW	32	0x0000 0048	0x4A31 4048
RESERVED			0x0000 0050	0x4A31 4050
WDT_WIRQSTATRAW	RW	32	0x0000 0054	0x4A31 4054
WDT_WIRQSTAT	RW	32	0x0000 0058	0x4A31 4058
WDT_WIRQENSET	RW	32	0x0000 005C	0x4A31 405C
WDT_WIRQENCLR	RW	32	0x0000 0060	0x4A31 4060
WDT_WIRQWAKEEN	RW	32	0x0000 0064	0x4A31 4064

Table 22-97 lists the WDTIMER3 registers.

**Table 22-97. WDTIMER3 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
WDT_WIDR	R	32	0x0000 0000	0x4903 0000	0x4013 0000	0x01D3 0000
WDT_WDSC	RW	32	0x0000 0010	0x4903 0010	0x4013 0010	0x01D3 0010
WDT_WDST	R	32	0x0000 0014	0x4903 0014	0x4013 0014	0x01D3 0014
WDT_WISR	RW	32	0x0000 0018	0x4903 0018	0x4013 0018	0x01D3 0018
WDT_WIER	RW	32	0x0000 001C	0x4903 001C	0x4013 001C	0x01D3 001C
WDT_WWER	RW	32	0x0000 0020	0x4903 0020	0x4013 0020	0x01D3 0020
WDT_WCLR	RW	32	0x0000 0024	0x4903 0024	0x4013 0024	0x01D3 0024
WDT_WCRR	RW	32	0x0000 0028	0x4903 0028	0x4013 0028	0x01D3 0028
WDT_WLDR	RW	32	0x0000 002C	0x4903 002C	0x4013 002C	0x01D3 002C
WDT_WTGR	RW	32	0x0000 0030	0x4903 0030	0x4013 0030	0x01D3 0030
WDT_WWPS	R	32	0x0000 0034	0x4903 0034	0x4013 0034	0x01D3 0034
WDT_WDLY	RW	32	0x0000 0044	0x4903 0044	0x4013 0044	0x01D3 0044
WDT_WSPR	RW	32	0x0000 0048	0x4903 0048	0x4013 0048	0x01D3 0048
RESERVED			0x0000 0050	0x4903 0050	0x4013 0050	0x01D3 0050
WDT_WIRQSTATRAW	RW	32	0x0000 0054	0x4903 0054	0x4013 0054	0x01D3 0054
WDT_WIRQSTAT	RW	32	0x0000 0058	0x4903 0058	0x4013 0058	0x01D3 0058
WDT_WIRQENSET	RW	32	0x0000 005C	0x4903 005C	0x4013 005C	0x01D3 005C
WDT_WIRQENCLR	RW	32	0x0000 0060	0x4903 0060	0x4013 0060	0x01D3 0060
WDT_WIRQWAKEEN	RW	32	0x0000 0064	0x4903 0064	0x4013 0064	0x01D3 0064

**NOTE:**

- The [WDT\\_WISR](#) and [WDT\\_WIRQSTATRAW](#) registers have the same functionality. The [WDT\\_WISR](#) register is used for software backward compatibility.
- The [WDT\\_WIER](#) and [WDT\\_WIRQENSET/WDT\\_WIRQENCLR](#) registers have the same functionality. The [WDT\\_WIER](#) register is used for software backward compatibility.
- The [WDT\\_WWER](#) and [WDT\\_WIRQWAKEEN](#) registers have the same functionality. The [WDT\\_WWER](#) is used for software backward compatibility.
- The [WDT\\_WIRQSTATRAW](#) and [WDT\\_WIRQSTAT](#) registers give the same information when read. The [WDT\\_WIRQSTATRAW](#) register is used for debug.



22.3.6.2.2 Watchdog Timer Register Description

Table 22-98 through Table 22-132 describe the watchdog timer registers.

Table 22-98. WDT\_WIDR

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .	<b>Instance</b> WDTIMER2 and WDTIMER3
<b>Description</b>	IP revision identifier	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x- <sup>(1)</sup>

<sup>(1)</sup> TI internal data

Table 22-99. Register Call Summary for Register WDT\_WIDR

- Watchdog Timers Functional Description
- [Accessing Watchdog Timer Registers: \[0\]](#)
- Watchdog Timer Register Manual
- [Watchdog Timer Register Summary: \[1\] \[2\]](#)

Table 22-100. WDT\_WDSC

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .	<b>Instance</b> WDTIMER2 and WDTIMER3
<b>Description</b>	This register controls the various parameters of the L4 interface.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EMUFREE	IDLEMODE	RESERVED	SOFTRESET	RESERVED													

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Write os for future compatibility. Reads return 0.	R	0x0000000
5	EMUFREE	Emulation mode 0x0: Timer counter frozen in emulation 0x1: Timer counter free-running in emulation	RW	0

Bits	Field Name	Description	Type	Reset
4:3	IDLEMODE	<p>Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.</p> <p>0x0: Force-idle mode: local target IDLE state follows (acknowledges) the system idle requests unconditionally, that is, regardless of the IP module internal requirements. Backup mode, for debug only.</p> <p>0x1: No-idle mode: local target never enters IDLE state. Backup mode, for debug only.</p> <p>0x2: Smart-idle mode: local target IDLE state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module should not generate (IRQ- or DMA-request-related) wake-up events.</p> <p>0x3: Smart-idle wake-up-capable mode: local target IDLE state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. Mode is relevant only if the appropriate IP module <i>wake-up</i> output(s) is (are) implemented.</p>	RW	0x2
2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0
1	SOFTRESET	<p>Software reset. (Optional)</p> <p>Read 0x0: Reset done, no pending action</p> <p>Write 0x0: No action</p> <p>Write 0x1: Initiate software reset.</p> <p>Read 0x1: Reset (software or other) ongoing</p>	RW	0
0	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0

**Table 22-101. Register Call Summary for Register WDT\_WDSC**

## Watchdog Timers Functional Description

- [Power Management: \[0\]](#)
- [Wake Up Capability: \[1\]](#)
- [Watchdog Timers Under Emulation: \[2\] \[3\]](#)
- [Accessing Watchdog Timer Registers: \[4\]](#)

## Watchdog Timer Low-level Programming Model

- [Main Sequence - WD Timer Module Global Initialization: \[5\] \[6\] \[7\]](#)

## Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[8\] \[9\]](#)

**Table 22-102. WDT\_WDST**

<b>Address Offset</b>	0x0000 0014																														
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .																<b>Instance</b>	WDTIMER2 and WDTIMER3													
<b>Description</b>	This register provides status information about the module.																														
<b>Type</b>	R																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RESETDONE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0.	R	0x0000 0000
0	RESETDONE	Internal module reset monitoring Read 0x0: Internal module reset is ongoing. Read 0x1: Reset completed	R	1

**Table 22-103. Register Call Summary for Register WDT\_WDST**

Watchdog Timers Functional Description

- [Accessing Watchdog Timer Registers: \[0\]](#)

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[1\] \[2\]](#)

**Table 22-104. WDT\_WISR**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .		
<b>Description</b>	This register shows which interrupt events are pending inside the module.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DLY_IT_FLAG		OVF_IT_FLAG													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads return 0.	R	0x0000 0000
1	DLY_IT_FLAG	Pending delay interrupt status. Read 0x0: No delay interrupt pending Write 0x0: Status unchanged Write 0x1: Status bit cleared Read 0x1: Delay interrupt pending	RW W1toClr	0
0	OVF_IT_FLAG	Pending overflow interrupt status. Read 0x0: No overflow interrupt pending Write 0x0: Status unchanged Write 0x1: Status bit cleared Read 0x1: Overflow interrupt pending	RW W1toClr	0

**Table 22-105. Register Call Summary for Register WDT\_WISR**

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[0\] \[1\] \[2\] \[3\]](#)

**Table 22-106. WDT\_WIER**

<b>Address Offset</b>	0x0000 001C		
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Description</b>	This register controls (enable/disable) the interrupt events.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DLY_IT_ENA		OVF_IT_ENA													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads return 0.	R	0x0000 0000
1	DLY_IT_ENA	Delay interrupt enable/disable 0x0: Disable delay interrupt. 0x1: Enable delay interrupt.	RW	0
0	OVF_IT_ENA	Overflow interrupt enable/disable 0x0: Disable overflow interrupt. 0x1: Enable overflow interrupt.	RW	0

**Table 22-107. Register Call Summary for Register WDT\_WIER**

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[0\] \[1\] \[2\] \[3\]](#)

**Table 22-108. WDT\_WWER**

<b>Address Offset</b>	0x0000 0020		
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Description</b>	This register controls (enable/disable) the wake-up events.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DLY_WK_ENA		OVF_WK_ENA													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	DLY_WK_ENA	Delay wake-up enable 0x0: Disable delay wakeup. 0x1: Enable delay wakeup.	RW	0
0	OVF_WK_ENA	Overflow wake-up enable 0x0: Disable overflow wakeup. 0x1: Enable overflow wakeup.	RW	0

**Table 22-109. Register Call Summary for Register WDT\_WWER**

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[0\] \[1\] \[2\] \[3\]](#)

**Table 22-110. WDT\_WCLR**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .		
<b>Description</b>	This register controls the prescaler stage of the counter.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PRE		PTV			RESERVED										

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reads return 0.	R	0x0000000
5	PRE	Prescaler enable/disable configuration 0x0: Prescaler disabled 0x1: Prescaler enabled	RW	1
4:2	PTV	Prescaler value The timer counter is prescaled with the value: 2**PTV. Example: PTV = 3 -> counter increases value if started after 8 functional clock periods. On reset, it is loaded from PI_PTV_RESET_VALUE input port.	RW	0x0
1:0	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0

**Table 22-111. Register Call Summary for Register WDT\_WCLR**

Watchdog Timers Functional Description

- [General Watchdog Timers Operation: \[0\] \[1\] \[2\]](#)
- [Reset Context: \[3\] \[4\]](#)
- [Prescaler Value/Timer Reset Frequency: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [Modifying Timer Count/Load Values and Prescaler Setting: \[12\]](#)
- [Accessing Watchdog Timer Registers: \[13\]](#)

Watchdog Timer Low-level Programming Model

- [Main Sequence - WD Timer Basic Configuration: \[14\] \[15\]](#)

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[16\] \[17\]](#)

**Table 22-112. WDT\_WCRR**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .		
<b>Description</b>	This register holds the value of the internal counter.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER_COUNTER																															

Bits	Field Name	Description	Type	Reset
31:0	TIMER_COUNTER	Value of the timer counter register	RW	0x0000 0000

**Table 22-113. Register Call Summary for Register WDT\_WCRR**

## Watchdog Timers Functional Description

- [General Watchdog Timers Operation](#): [0]
- [Overflow/Reset Generation](#): [1]
- [Modifying Timer Count/Load Values and Prescaler Setting](#): [2]
- [Watchdog Counter Register Access Restriction \(WDT\\_WCRR Register\)](#): [3] [4] [5] [6]
- [Watchdog Timer Interrupt Generation](#): [7]
- [Accessing Watchdog Timer Registers](#): [8]

## Watchdog Timer Low-level Programming Model

- [Main Sequence - WD Timer Basic Configuration](#): [9]

## Watchdog Timer Register Manual

- [Watchdog Timer Register Summary](#): [10] [11]

**Table 22-114. WDT\_WLDR**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .		
<b>Description</b>	This register holds the timer load value.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER_LOAD																															

Bits	Field Name	Description	Type	Reset
31:0	TIMER_LOAD	Value of the timer load register	RW	0x0000 0000

**Table 22-115. Register Call Summary for Register WDT\_WLDR**

## Watchdog Timers Functional Description

- [General Watchdog Timers Operation](#): [0]
- [Reset Context](#): [1] [2]
- [Overflow/Reset Generation](#): [3]
- [Prescaler Value/Timer Reset Frequency](#): [4] [5] [6] [7] [8] [9] [10]
- [Triggering a Timer Reload](#): [11]
- [Modifying Timer Count/Load Values and Prescaler Setting](#): [12]
- [Watchdog Timer Interrupt Generation](#): [13] [14] [15]
- [Accessing Watchdog Timer Registers](#): [16]

## Watchdog Timer Register Manual

- [Watchdog Timer Register Summary](#): [17] [18]

**Table 22-116. WDT\_WTGR**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .		
<b>Description</b>	Writing a different value than the one already written in this register does a watchdog counter reload.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTGR_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	TTGR_VALUE	Value of the trigger register	RW	0x0000 0000

**Table 22-117. Register Call Summary for Register WDT\_WTGR**

Watchdog Timers Functional Description

- [General Watchdog Timers Operation: \[0\]](#)
- [Triggering a Timer Reload: \[1\] \[2\]](#)
- [Modifying Timer Count/Load Values and Prescaler Setting: \[3\]](#)
- [Accessing Watchdog Timer Registers: \[4\]](#)

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[5\] \[6\]](#)

**Table 22-118. WDT\_WWPS**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .		
<b>Description</b>	This register contains the write posting bits for all writeable functional registers.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							W_PEND_WDLY	W_PEND_WSPR	W_PEND_WTGR	W_PEND_WLDR	W_PEND_WCRR	W_PEND_WCLR			

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x00000000
5	W_PEND_WDLY	Write pending for register WDLY Read 0x0: No register write pending Read 0x1: Register write pending	R	0
4	W_PEND_WSPR	Write pending for register WSPR Read 0x0: No register write pending Read 0x1: Register write pending	R	0
3	W_PEND_WTGR	Write pending for register WTGR Read 0x0: No register write pending Read 0x1: Register write pending	R	0
2	W_PEND_WLDR	Write pending for register WLDR Read 0x0: No register write pending Read 0x1: Register write pending	R	0
1	W_PEND_WCRR	Write pending for register WCRR Read 0x0: No register write pending Read 0x1: Register write pending	R	0
0	W_PEND_WCLR	Write pending for register WCLR Read 0x0: No register write pending Read 0x1: Register write pending	R	0

**Table 22-119. Register Call Summary for Register WDT\_WWPS**

Watchdog Timers Functional Description

- [Accessing Watchdog Timer Registers: \[0\]](#)

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[1\] \[2\]](#)

**Table 22-120. WDT\_WDLY**

<b>Address Offset</b>	0x0000 0044																																																																	
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .	<b>Instance</b> WDTIMER2 and WDTIMER3																																																																
<b>Description</b>	This register holds the delay value that controls the internal pre-overflow event detection.																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">WDLY_VALUE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	WDLY_VALUE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
WDLY_VALUE																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	WDLY_VALUE	Value of the delay register	RW	0x0000 0000																																																														

**Table 22-121. Register Call Summary for Register WDT\_WDLY**

Watchdog Timers Functional Description

- [Modifying Timer Count/Load Values and Prescaler Setting: \[0\]](#)
- [Watchdog Timer Interrupt Generation: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [Accessing Watchdog Timer Registers: \[8\]](#)

Watchdog Timer Low-level Programming Model

- [Main Sequence - WD Timer Basic Configuration: \[9\]](#)

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[10\] \[11\]](#)

**Table 22-122. WDT\_WSPR**

<b>Address Offset</b>	0x0000 0048																																																																	
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .	<b>Instance</b> WDTIMER2 and WDTIMER3																																																																
<b>Description</b>	This register holds the start-stop value that controls the internal start-stop FSM.																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">WSPR_VALUE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	WSPR_VALUE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
WSPR_VALUE																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	WSPR_VALUE	Value of the start-stop register	RW	0x0000 0000																																																														

**Table 22-123. Register Call Summary for Register WDT\_WSPR**

Watchdog Timers Functional Description

- [General Watchdog Timers Operation: \[0\]](#)
- [Prescaler Value/Timer Reset Frequency: \[1\] \[2\]](#)
- [Start/Stop Sequence for Watchdog Timers \(Using WDT\\_WSPR Register\): \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [Modifying Timer Count/Load Values and Prescaler Setting: \[9\]](#)
- [Accessing Watchdog Timer Registers: \[10\]](#)

Watchdog Timer Low-level Programming Model

- [Subsequence - Disable the WD Timer: \[11\] \[12\]](#)
- [Sub-sequence - Enable the WD Timer: \[13\] \[14\]](#)



**Table 22-123. Register Call Summary for Register WDT\_WSPR (continued)**

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[15\] \[16\]](#)

**Table 22-124. WDT\_WIRQSTATRAW**

<b>Address Offset</b>	0x0000 0054	
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .	<b>Instance</b> WDTIMER2 and WDTIMER3
<b>Description</b>	IRQ unmasked status, status set per-event raw interrupt status vector, line 0. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT_DLY		EVENT_OVF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	EVENT_DLY	Settable raw status for delay event Read 0x0: No event pending Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending	RW W1toSet	0
0	EVENT_OVF	Settable raw status for overflow event Read 0x0: No event pending Write 0x0: No action Write 0x1: Set event (debug) Read 0x1: Event pending	RW W1toSet	0

**Table 22-125. Register Call Summary for Register WDT\_WIRQSTATRAW**

Watchdog Timers Functional Description

- [Accessing Watchdog Timer Registers: \[0\]](#)

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[1\] \[2\] \[3\] \[4\] \[5\]](#)

**Table 22-126. WDT\_WIRQSTAT**

<b>Address Offset</b>	0x0000 0058	
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .	<b>Instance</b> WDTIMER2 and WDTIMER3
<b>Description</b>	IRQ masked status, status clear per-event enabled interrupt status vector, line 0. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT_DLY		EVENT_OVF													

## Watchdog Timers

www.ti.com

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	EVENT_DLY	Clearable, enabled status for delay event Read 0x0: No (enabled) event pending Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending	RW W1toClr	0
0	EVENT_OVF	Clearable, enabled status for overflow event Read 0x0: No (enabled) event pending Write 0x0: No action Write 0x1: Clear (raw) event Read 0x1: Event pending	RW W1toClr	0

**Table 22-127. Register Call Summary for Register WDT\_WIRQSTAT**

## Watchdog Timers Functional Description

- [Interrupts: \[0\] \[1\]](#)
- [General Watchdog Timers Operation: \[2\]](#)
- [Watchdog Timer Interrupt Generation: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [Accessing Watchdog Timer Registers: \[14\]](#)

## Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[15\] \[16\]](#)

**Table 22-128. WDT\_WIRQENSET**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .		
<b>Description</b>	IRQ enable set per-event interrupt enable bit vector, line 0. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE_DLY		ENABLE_OVF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	ENABLE_DLY	Enable for delay event Read 0x0: Interrupt disabled (masked) Write 0x0: No action Write 0x1: Enable interrupt. Read 0x1: Interrupt enabled	RW W1toSet	0
0	ENABLE_OVF	Enable for overflow event Read 0x0: Interrupt disabled (masked) Write 0x0: No action Write 0x1: Enable interrupt. Read 0x1: Interrupt enabled	RW W1toSet	0

**Table 22-129. Register Call Summary for Register WDT\_WIRQENSET**

Watchdog Timers Functional Description

- [Interrupts: \[0\] \[1\]](#)
- [General Watchdog Timers Operation: \[2\]](#)
- [Watchdog Timer Interrupt Generation: \[3\] \[4\] \[5\] \[6\]](#)
- [Accessing Watchdog Timer Registers: \[7\]](#)

Watchdog Timer Low-level Programming Model

- [Main Sequence - WD Timer Module Global Initialization: \[8\] \[9\]](#)

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[10\] \[11\] \[12\]](#)

**Table 22-130. WDT\_WIRQENCLR**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .		
<b>Description</b>	IRQ enable clear per-event interrupt enable bit vector, line 0. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE_DLY		ENABLE_OVF													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	ENABLE_DLY	Enable for delay event Read 0x0: Interrupt disabled (masked) Write 0x0: No action Write 0x1: Disable interrupt. Read 0x1: Interrupt enabled	RW W1toClr	0
0	ENABLE_OVF	Enable for overflow event Read 0x0: Interrupt disabled (masked) Write 0x0: No action Write 0x1: Disable interrupt. Read 0x1: Interrupt enabled	RW W1toClr	0

**Table 22-131. Register Call Summary for Register WDT\_WIRQENCLR**

Watchdog Timers Functional Description

- [Interrupts: \[0\] \[1\]](#)
- [General Watchdog Timers Operation: \[2\]](#)
- [Watchdog Timer Interrupt Generation: \[3\] \[4\]](#)
- [Accessing Watchdog Timer Registers: \[5\]](#)

Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[6\] \[7\] \[8\]](#)

**Table 22-132. WDT\_WIRQWAKEEN**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See <a href="#">Table 22-96</a> .	<b>Instance</b>	WDTIMER2 and WDTIMER3
<b>Description</b>	This register controls (enable/disable) the wake-up events.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DLY_WK_ENA		OVF_WK_ENA													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0s for future compatibility. Reads return 0.	R	0x0000 0000
1	DLY_WK_ENA	Enable delay wake-up 0x0: Disable delay wakeup 0x1: Enable delay wakeup	RW	0
0	OVF_WK_ENA	Enable overflow wakeup 0x0: Disable overflow wakeup 0x1: Enable overflow wakeup	RW	0

**Table 22-133. Register Call Summary for Register WDT\_WIRQWAKEEN**

## Watchdog Timers Functional Description

- [Power Management: \[0\]](#)
- [Wake Up Capability: \[1\] \[2\]](#)
- [Accessing Watchdog Timer Registers: \[3\]](#)

## Watchdog Timer Low-level Programming Model

- [Main Sequence - WD Timer Module Global Initialization: \[4\] \[5\]](#)

## Watchdog Timer Register Manual

- [Watchdog Timer Register Summary: \[6\] \[7\] \[8\]](#)

## 22.4 32-kHz Synchronized Timer

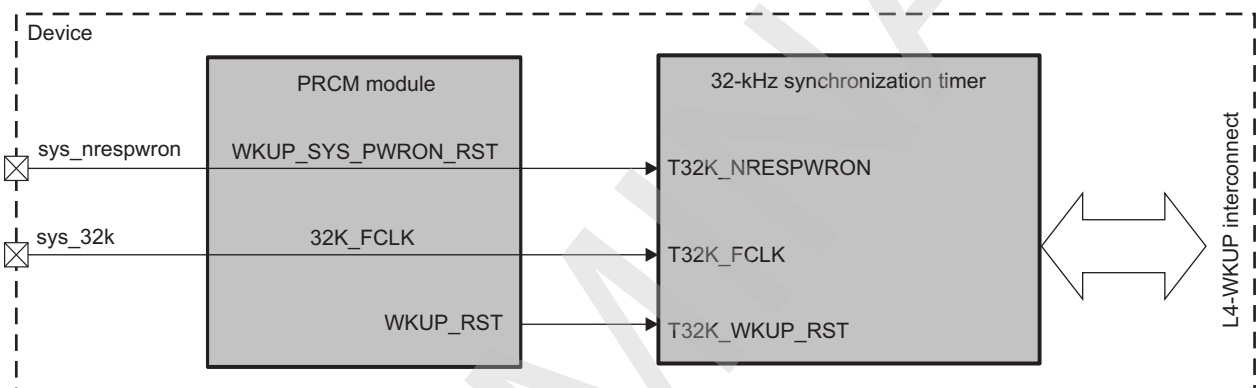
### 22.4.1 32-kHz Synchronized Timer Overview

The 32-kHz synchronized timer is a 32-bit counter clocked by the falling edge of the 32-kHz system clock. It is reset while the external asynchronous power-up reset (`sys_nrespwron`) primary input/output (I/O) is active (main device reset). When `sys_nrespwron` is released (on the rising edge of `sys_nrespwron`), after three 32-kHz clock periods, the counter starts counting up from the reset value of the counter register on the falling edge of the 32-kHz system clock. After reaching its highest value, the counter wraps back to 0 and starts counting again.

Figure 22-19 is the block diagram of the 32-kHz synchronized timer.

**NOTE:** `sys_nrespwron` is an active-low I/O.

Figure 22-19. 32-kHz Synchronized Timer Block Diagram



timers-018

#### 22.4.1.1 32-kHz Synchronized Timer Features

The main features of the 32-kHz synchronized timer controller are:

- L4 slave interface support:
  - 32-bit data bus width
  - 32-/16-bit access supported
  - 8-bit access not supported
  - 5-bit address bus width
  - Burst mode not supported
  - Write nonposted transaction mode supported
- Only read operations are supported on the module registers; no write operation is supported (no error/no action on write).
- Free-running 32-bit upward counter
- Start and keep counting after power-on reset
- Automatic roll over to 0; highest value reached: 0xFFFF FFFF
- On-the-fly read (while counting)

#### 22.4.2 32-kHz Synchronized Timer Environment

The synchronized timer is accessible only through the L4 interface.

#### 22.4.3 32-kHz Synchronized Timer Integration

Table 22-134 through Table 22-136 summarize the integration of the module in the device.

**Table 22-134. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
32KTIMER	PD_WKUP	No	L4_WKUP

**Table 22-135. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
32KTIMER	T32K_FCLK	32K_FCLK	PRCM	32KTIMER functional clock
32KTIMER	T32K_ICLK	WKUP_L4_ICLK2	PRCM	32KTIMER interface clock
Resets				
32KTIMER	T32K_NRESPWRON	WKUP_SYS_PWRON_RST	PRM	Reset to 32KTIMER
32KTIMER	T32K_WKUP_RST	WKUP_RST	PRM	Reset to 32KTIMER

**Table 22-136. Hardware Requests**

No Interrupt Requests
No DMA Requests

#### 22.4.4 32-kHz Synchronized Timer Functional Description

The synchronized timer is a counter that starts on the rising edge of an external asynchronous signal (`sys_nrespwron`). When `sys_nrespwron` is released (on the rising edge of `sys_nrespwron`), the counter starts counting up from the reset value of the counter register on the falling edge of the 32-kHz system clock after three inverted 32-kHz clock periods. After reaching its highest value, the counter wraps back to 0 and starts counting again with no additional delay. When `sys_nrespwron` is released (on the rising edge of `sys_nrespwron`), after three inverted 32-kHz clock periods, the counter starts counting up from the reset value of the counter register on the falling edge of the 32-kHz system clock.

Figure 22-20 shows the reset synchronization timing diagram. After reaching its highest value, the counter wraps back to 0 and starts counting again without any extra delay.

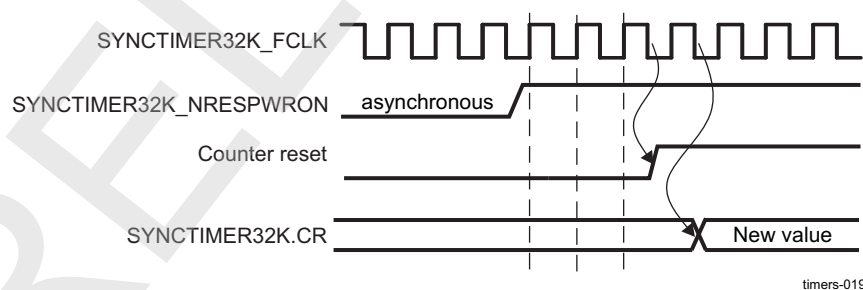
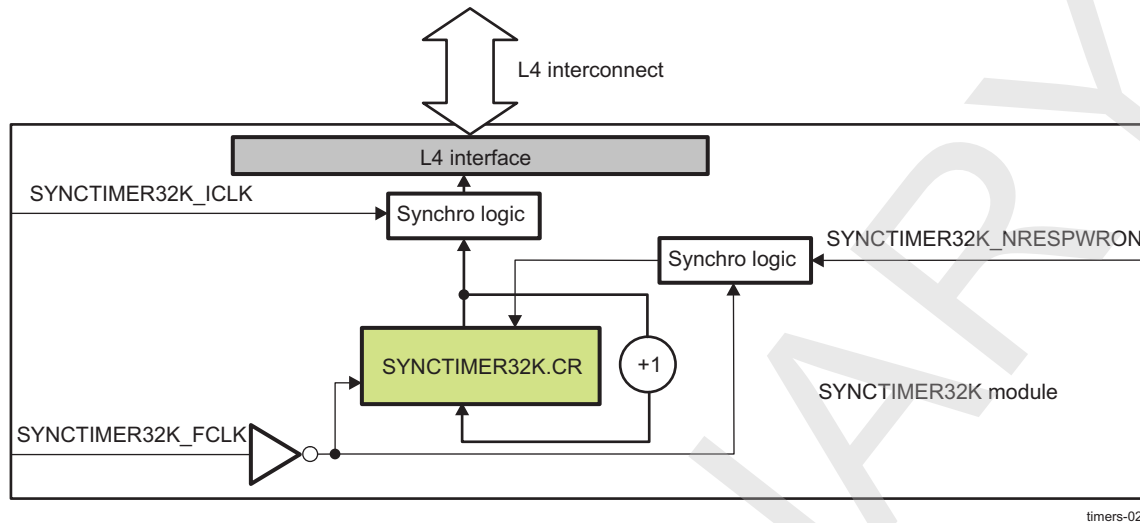
**Figure 22-20. Reset Resynchronization Timing Diagram**

Figure 22-21 is the block diagram of the synchronized timer.

**NOTE:** `sys_nrespwron` is an active-low input.

Figure 22-21. SYNCTIMER32K Block Diagram



The sync logic ensures the read transaction correctness by synchronizing the counter register read access on 32KSYNCNT\_ICLK because the 32KSYNCNT\_ICLK clock signal is completely asynchronous with S32KSYNCNT\_FCLK. The sys\_nrespwron input resets the counter register (32KSYNCNT\_CR). The inverted 32KSYNCNT\_FCLK clocks the sys\_nrespwron reset signal.

#### 22.4.4.1 Reading the 32-kHz Synchronized Timer

The counter register (32KSYNCNT\_CR) is 32 bits wide. For correct count capture, it must be accessed as 16-bit LSB access first and 16-bit MSB access next. Internal synchronization logic allows reading of the counter value with 32KSYNCNT\_ICLK while the counter is running. The time latency to read the synchronized counter register is one 32KSYNCNT\_ICLK clock period.

## 22.4.5 32KTimer Register Manual

Table 22-137 lists the base address and block size for the 32-kHz synchronized timer. It is memory mapped to the L4 peripheral bus memory space.

**Table 22-137. 32KTimer Instance Summary**

Module Name	Base Address L4 Interconnect	Size
32KTimer	0x4A30 4000	4KB

### 22.4.5.1 32KTimer Register Mapping Summary

#### CAUTION

The 32-kHz synchronized timer registers are limited to 32-bit and 16-bit data accesses; 8-bit access is not allowed and can corrupt the register content.

Table 22-138 lists the 32-kHz synchronized timer registers. Table 22-139 through Table 22-143 describe the register bits.

**Table 22-138. 32KTimer Register Summary**

Register Name	Type	Register Width (Bits)	Offset Address	Physical Address L4 Interconnect
<a href="#">32KSYNCNT_REV</a>	R	32	0x0000	0x4A30 4000
<a href="#">32KSYNCNT_SYSCONFIG</a>	R/W	32	0x0004	0x4A30 4004
<a href="#">32KSYNCNT_CR</a>	R	32	0x0010	0x4A30 4010

### 22.4.5.2 32KTimer Register Description

**Table 22-139. 32KSYNCNT\_REV**

<b>Address Offset</b>	0x0000
<b>Physical Address</b>	0x4A30 4000
<b>Description</b>	This register contains the sync counter IP revision code.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x- <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 22-140. Register Call Summary for Register 32KSYNCNT\_REV**

32-kHz Synchronized Timer Register Manual

- [32-kHz Synchronized Timer Register Mapping Summary: \[0\]](#)



**Table 22-141. 32KSYNCNT\_SYSCONFIG**

<b>Address Offset</b>	0x0004
<b>Physical Address</b>	0x4A30 4004
<b>Description</b>	This register is used for IDLE modes only.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																							IDLEMODE			Reserved					

Bits	Field Name	Description	Type	Reset
31:5	Reserved	Reads return 0.	R	0x0
4:3	IDLEMODE	Power management REQ/ACK control 0x0: Force idle. An idle request is acknowledged unconditionally. 0x1: No-idle. An idle request is never acknowledged. 0x2: Reserved 0x3: Reserved	RW	0x0
2:0	Reserved	Reads return 0.	R	0x0

**Table 22-142. Register Call Summary for Register 32KSYNCNT\_SYSCONFIG**

- 32-kHz Synchronized Timer Register Manual
- [32-kHz Synchronized Timer Register Mapping Summary: \[0\]](#)

**Table 22-143. 32KSYNCNT\_CR**

<b>Address Offset</b>	0x0010
<b>Physical Address</b>	0x4A30 4010
<b>Description</b>	This register contains the 32-kHz sync counter value.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER_VALUE	Counter register value	R	0x00000003

**Table 22-144. Register Call Summary for Register 32KSYNCNT\_CR**

- 32-kHz Synchronized Timer Functional Description
- [32-kHz Synchronized Timer Functional Description: \[0\]](#)
  - [Reading the 32-kHz Synchronized Timer: \[1\]](#)
- 32-kHz Synchronized Timer Register Manual
- [32-kHz Synchronized Timer Register Mapping Summary: \[2\]](#)

PRELIMINARY

## Serial Communication Interface

This chapter describes the features and operation of the device serial communication interface (SCI).

Topic	Page
23.1 Multimaster High-Speed I <sup>2</sup> C Controller .....	4044
23.2 HDQ/1-Wire .....	4135
23.3 UART/IrDA/CIR .....	4155
23.4 Multichannel Serial Port Interface (MCSPi) .....	4264
23.5 Multichannel Buffered Serial Port (McBSP) .....	4327
23.6 Multichannel PDM Controller .....	4433
23.7 Digital Microphone Module .....	4477
23.8 Multichannel Audio Serial Port .....	4510
23.9 Serial Low-Power Inter-Chip Media Bus Controller .....	4567
23.10 MIPI-HSI .....	4690
23.11 High-Speed Multiport USB Host Subsystem .....	4765
23.12 High-Speed USB OTG Controller .....	4888
23.13 Full-Speed USB Host Controller .....	4930

## 23.1 Multimaster High-Speed I<sup>2</sup>C Controller

### 23.1.1 HS I<sup>2</sup>C Overview

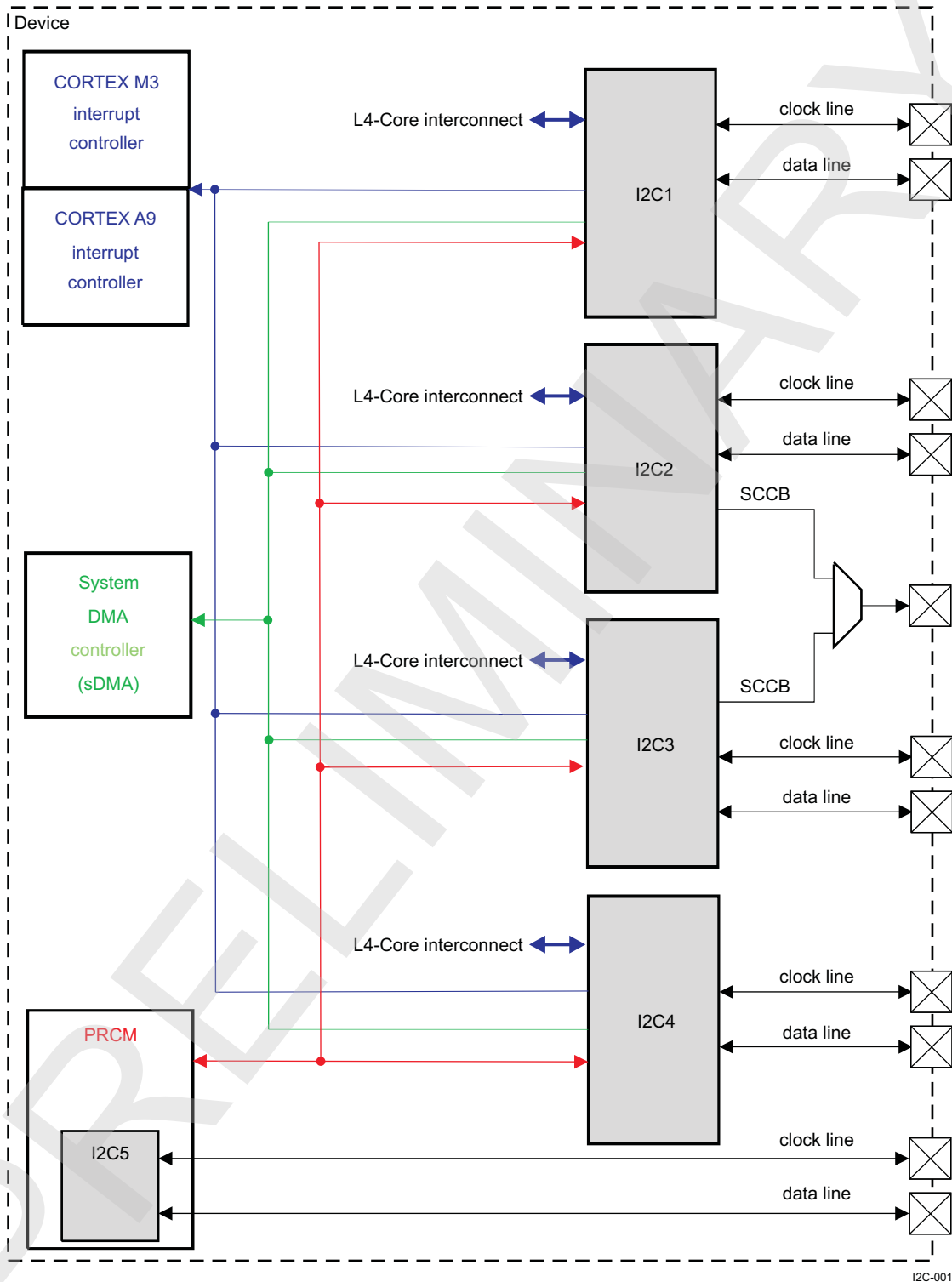
The device contains four multimaster high-speed (HS) inter-integrated circuit (I<sup>2</sup>C) controllers (I2Ci modules, where i = 1, 2, 3, 4), each of which provides an interface between a local host (LH), such as a digital signal processor (DSP), and any I<sup>2</sup>C-bus-compatible device that connects through the I<sup>2</sup>C serial bus. External components attached to the I<sup>2</sup>C bus can serially transmit and receive up to 8 bits of data to and from the LH device through the 2-wire I<sup>2</sup>C interface.

Each multimaster HS I<sup>2</sup>C controller can be configured to act like a slave or master I<sup>2</sup>C-compatible device. Moreover, each multimaster HS I<sup>2</sup>C controller can be configured in serial camera control bus (SCCB) mode (the SCCB is a serial bus developed by Omnivision Technologies, Inc.) to act as a master on a 2-wire SCCB bus. Only multimaster HS I<sup>2</sup>C controllers I2C2 and I2C3 can be configured in SCCB mode to act as a master device on a 3-wire SCCB bus.

The device contains an additional fifth master transmitter HS I<sup>2</sup>C interface (I2C5) in the power, reset, and clock management (PRCM) module to perform dynamic voltage control and power sequencing with an external power integrated circuit (IC).

Figure 23-1 shows the I<sup>2</sup>C.

Figure 23-1. HS I<sup>2</sup>C Controllers



The four multimaster HS I<sup>2</sup>C controllers have the following features:

- Compliant with Philips I2C specification version 3.0
- Supports a standard mode (up to 100 Kbps), fast mode (up to 400 Kbps) and fast mode+ (up to 1 Mbps)
- Supports High-speed (HS) mode for transfer up to 3.4 Mbps

- Support for 3-wire/2-wire SCCB master mode for I2C2 and I2C3 modules, 2-wire SCCB master mode for I2C1 and I2C4 modules, up to 100 Kbps
- 7-bit and 10-bit device addressing modes
- General call
- Start/Restart/Stop
- Multimaster transmitter/slave receiver mode
- Multimaster receiver/slave transmitter mode
- Combined master transmit/receive and receive/transmit mode
- Built-in configurable FIFOs (8, 16, 32, 64 bytes) for buffered read or write
- Module enable/disable capability
- Programmable multislave channel (responds to four separate addresses)
- Programmable clock generation
- 8-bit wide data access
- Open-core protocol (OCP) interface with LH application (OCP-IP 2.0 compliant)
- Designed for low power consumption
- Implement Auto Idle mechanism
- Implement Idle Request/Idle Acknowledge handshake mechanism
- Support for asynchronous wake-up mechanism
- Two direct memory access (DMA) channels
- Wide interrupt capability
- Supports OmniVision SCCB protocol
- Compliant with Highlander 0.8

The master transmitter HS I<sup>2</sup>C controller I2C5 has the following features:

- Support of HS and fast modes
- 7-bit addressing mode only
- Master transmitter mode only
- Start/restart/stop

---

**NOTE:** Before using HS I<sup>2</sup>C mode, determine that the target device supports this mode.

---

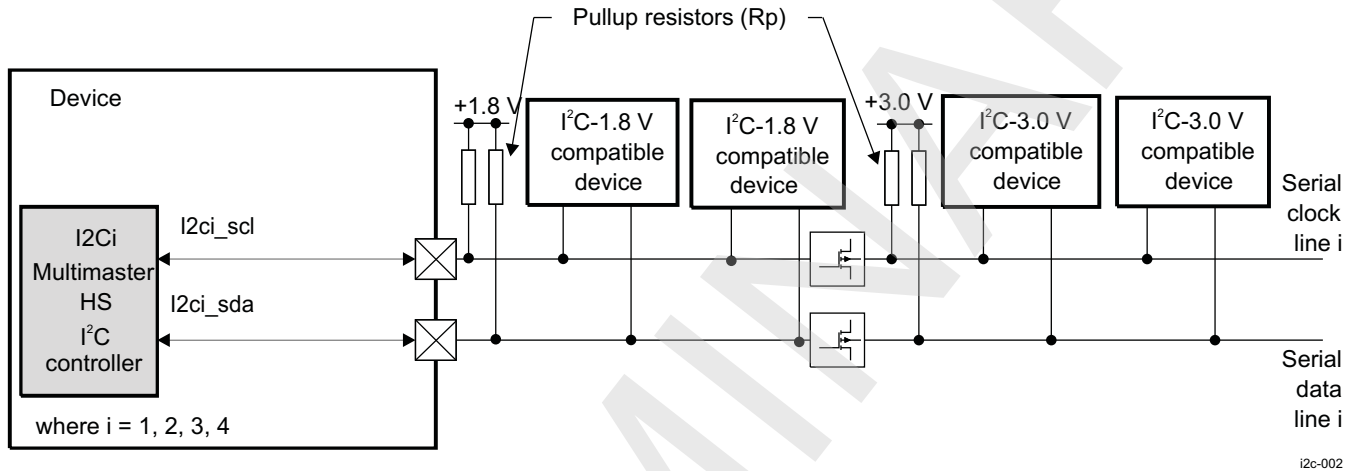
### 23.1.2 HS I<sup>2</sup>C Environment

This section describes the HS I<sup>2</sup>C application fields from an environment point of view (external connections). It describes HS I<sup>2</sup>C connectivity options, lists all possible interfaces, and describes the protocol and data format used in each case.

#### 23.1.2.1 HS I<sup>2</sup>C in I<sup>2</sup>C Mode

Figure 23-2 shows the multimaster HS I<sup>2</sup>C controllers and their related connections with I<sup>2</sup>C-compliant devices in I<sup>2</sup>C mode.

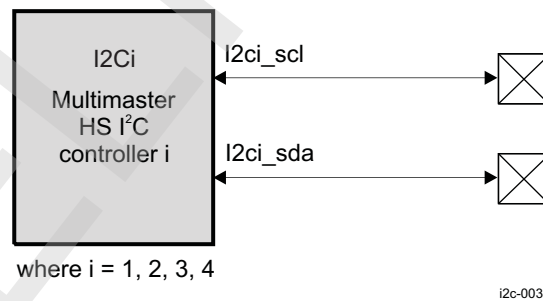
Figure 23-2. HS I<sup>2</sup>C and Typical Connections to I<sup>2</sup>C Devices



#### 23.1.2.1.1 HS I<sup>2</sup>C Pins for Typical Connections in I<sup>2</sup>C Mode

Figure 23-3 shows the multimaster HS I<sup>2</sup>C controller pins used for typical connections with I<sup>2</sup>C devices.

Figure 23-3. HS I<sup>2</sup>C Interface Signals in I<sup>2</sup>C Mode



#### 23.1.2.1.2 HS I<sup>2</sup>C Interface Typical Connections

Table 23-1 lists the pins associated with the I<sup>2</sup>C interface.

Table 23-1. HS I<sup>2</sup>C Input/Output

Signal	I/O <sup>(1)</sup>	Description	Reset Value
i2ci_scl	I/O	I <sup>2</sup> C serial clock line <sup>(2)</sup> . Open-drain output buffer. Requires external pullup resistor (Rp).	1
i2ci_sda	I/O	I <sup>2</sup> C serial data line. Open-drain output buffer. Requires external Rp.	1

<sup>(1)</sup> I = Input; O = Output

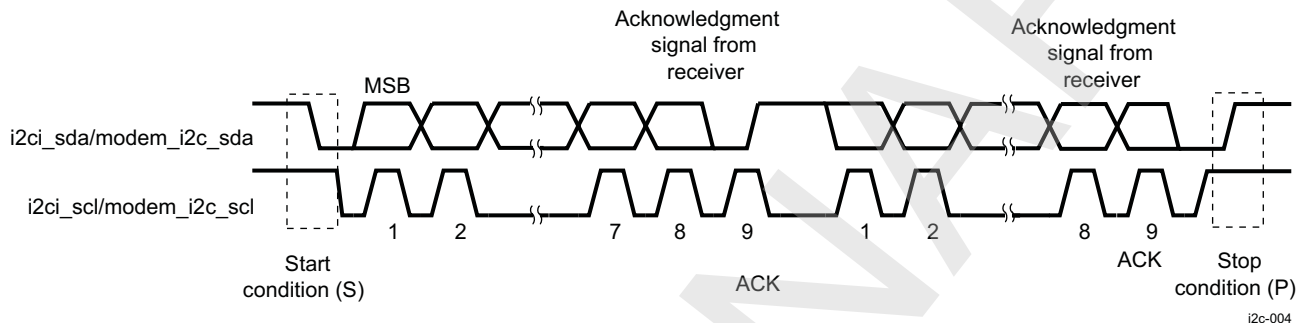
<sup>(2)</sup> This output signal is also used as retiming input.

### 23.1.2.1.3 HS I<sup>2</sup>C Typical Connection Protocol and Data Format

#### 23.1.2.1.3.1 HS I<sup>2</sup>C Serial Data Format

The I<sup>2</sup>C controller operates in 8-bit word data format (byte write access supported for the last access). Each byte transmitted or received on the serial data line is 8 bits long. The number of bytes that can be transmitted or received is not restricted. The data is transferred with the most-significant bit (MSB) first. In receiver mode, each byte is followed by an acknowledge bit from the I<sup>2</sup>C. Figure 23-4 shows a typical I<sup>2</sup>C communication format.

**Figure 23-4. HS I<sup>2</sup>C Data Transfer**

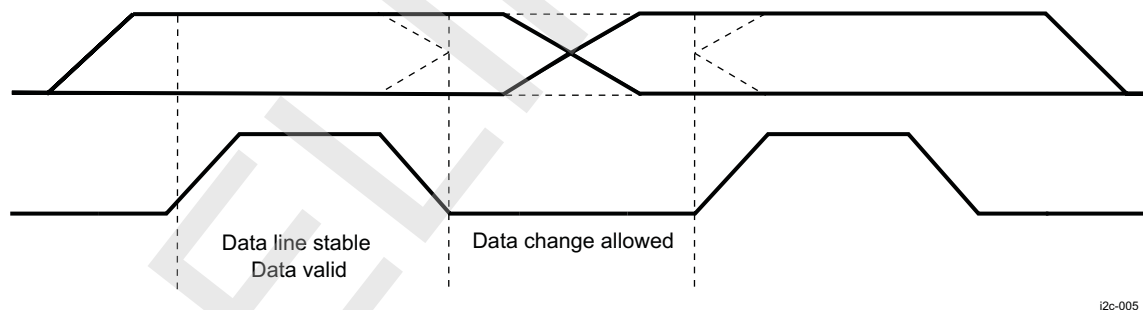


#### 23.1.2.1.3.2 HS I<sup>2</sup>C Data Validity

The data on the serial data line (SDA) must be stable during the high period of the serial clock line. The high and low states of the data line can change only when the clock signal on the serial clock line (SCK) is low.

Figure 23-5 is an example of data validity requirements.

**Figure 23-5. HS I<sup>2</sup>C Bit Transfer on the I<sup>2</sup>C Bus**



#### 23.1.2.1.3.3 HS I<sup>2</sup>C Start and Stop Conditions

The I<sup>2</sup>C module generates start (S) and stop (P) conditions when it is configured as a master.

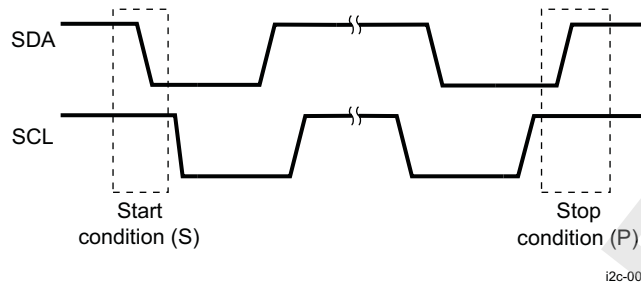
- An S condition is a high-to-low transition on the serial data line while serial clock line is high.
- A P condition is a low-to-high transition on the serial data line while serial clock line is high.

The bus is considered busy after the S condition (the `I2Ci.I2C_STAT[12]` BB bit is 1 to indicate that the bus is busy) and free after the P condition (the `I2Ci.I2C_STAT[12]` BB bit is 0 to indicate that the bus is free).

Figure 23-6 shows the waveforms that occur during an S and a P condition.



Figure 23-6. HS I<sup>2</sup>C S and P Condition Events



23.1.2.1.3.4 HS I<sup>2</sup>C Addressing

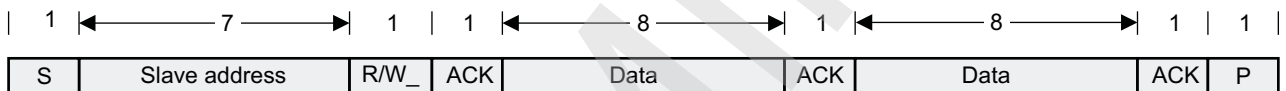
The I<sup>2</sup>C module supports two data formats in fast/standard (F/S) and HS modes:

- 7-bit/10-bit addressing format
- 7-bit/10-bit addressing format with repeated start (Sr) condition

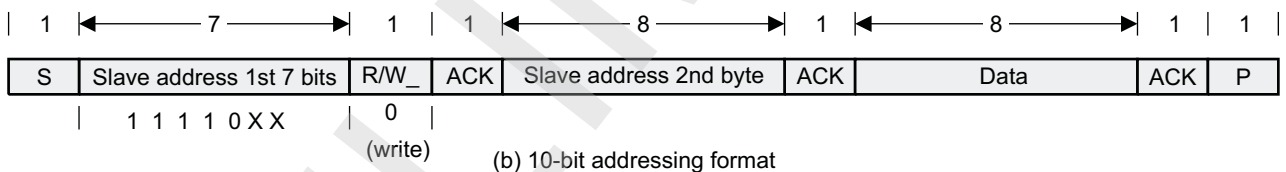
23.1.2.1.3.4.1 Data Transfer Formats in F/S Mode

Figure 23-7 shows the I<sup>2</sup>C data transfer formats in F/S mode.

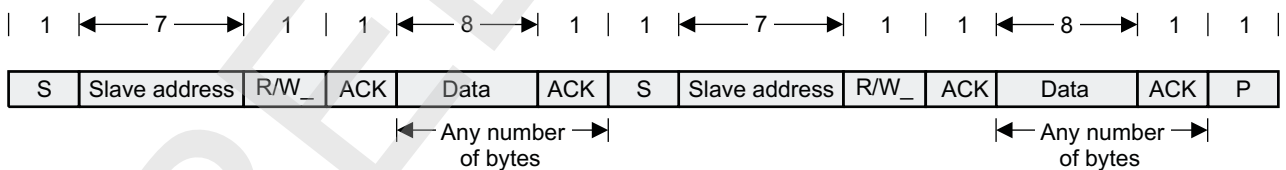
Figure 23-7. HS I<sup>2</sup>C Data Transfer Formats in F/S Mode



(a) 7-bit addressing format



(b) 10-bit addressing format



(c) Addressing format with repeated start condition

i2c-007

The first word after an S condition consists of 8 bits. In acknowledge mode, an extra dedicated acknowledgment bit is inserted after each byte.

In addressing formats with 7-bit addresses, the first byte is composed of 7 MSB slave address bits and 1 least-significant bit (LSB) R/W<sub>-</sub> bit.

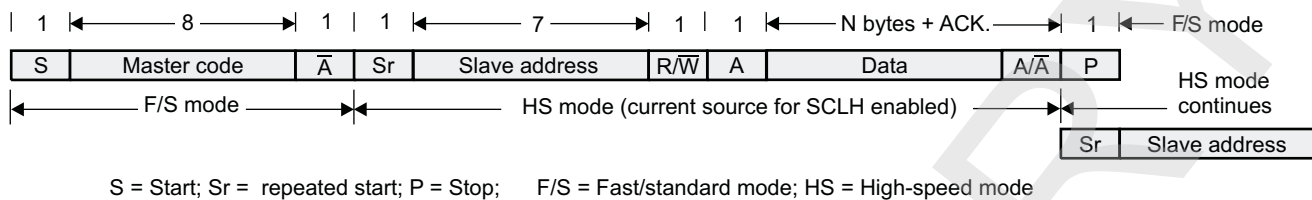
The LSB R/W<sub>-</sub> bit of the address byte indicates the transmission direction of the data bytes that follow it. If R/W<sub>-</sub> is 0, the master writes data to the selected slave; if it is 1, the master reads data from the slave.

In addressing formats with 10-bit addresses, the structure of the first byte is 11110XXY, where XX is the 2 MSBs of the 10-bit addresses, and Y is the R/W<sub>-</sub> bit. If the R/W<sub>-</sub> bit is 0, the next byte contains the last 8 bits of the slave address. If the R/W<sub>-</sub> bit is 1, the next byte contains data transmitted from the slave to the master.

### 23.1.2.1.3.4.2 Data Transfer Format in HS Mode

Figure 23-8 shows the I<sup>2</sup>C data transfer format in HS mode.

**Figure 23-8. HS I<sup>2</sup>C Data Transfer in HS Mode**



i2c-008

Each multimaster HS I<sup>2</sup>C controller can also operate in HS mode. In this case, after the S condition, the module, which is in F/S mode, writes the master code address (000001XXX, where XXX is the variable portion of the master code) on the bus. No device connected on the same bus acknowledges this address. The module switches the clock to the HS clock and after an Sr condition, and sends the slave address and the data, as shown in Figure 23-8.

### 23.1.2.1.3.5 HS I<sup>2</sup>C Master Transmitter

In master transmitter mode, data assembled in one of the previously described data formats is shifted out on the serial data line SDA in sync with the self-generated clock pulses on the serial clock line SCL. The clock pulses are inhibited and SCL is held low when the intervention of the processor is required (XUDF) after a byte is transmitted.

### 23.1.2.1.3.6 HS I<sup>2</sup>C Master Receiver

Master receiver mode can be entered only from master transmitter mode. With any of the address formats (a), (b), or (c) (see Figure 23-7), if R/W<sub>0</sub> is high, the module enters master receiver mode after the slave address byte and bit R/W<sub>0</sub> are transmitted. Serial data bits received on bus line SDA are shifted in synchronization with the self-generated clock pulses on SCL.

### 23.1.2.1.3.7 HS I<sup>2</sup>C Slave Transmitter

Slave transmitter mode can be entered only from slave receiver mode. With any of the address formats (a), (b), or (c) (see Figure 23-7), the slave transmitter is entered if the slave address byte is the same as its own address and bit R/W<sub>0</sub> is transmitted, if R/W<sub>0</sub> is high. The slave transmitter shifts the serial data out on the data line SDA in sync with the clock pulses that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the LH is required (XUDF).

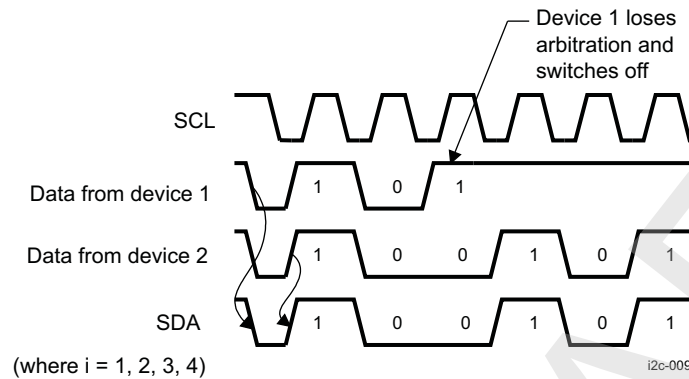
### 23.1.2.1.3.8 HS I<sup>2</sup>C Slave Receiver

In this mode, serial data bits received on the bus line SDA are shifted-in in sync with the clock pulses on SCL that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the LH is required (ROVR) after a byte is received.

### 23.1.2.1.3.9 HS I<sup>2</sup>C Bus Arbitration

If two or more master transmitters start a transmission on the same bus almost simultaneously, an arbitration procedure is invoked. The arbitration procedure uses the data presented on the serial bus by the competing transmitters. When a transmitter senses that a high signal it has presented on the bus has been overruled by a low signal, it switches to the slave receiver mode, sets the arbitration lost (AL) flag, and generates the arbitration lost interrupt. Figure 23-9 shows the arbitration procedure between two devices. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. If two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

Figure 23-9. HS I<sup>2</sup>C Arbitration Between Master Transmitters

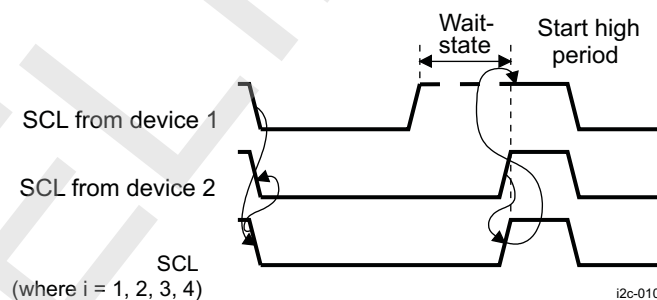


### 23.1.2.1.3.10 HS I<sup>2</sup>C Clock Generation and Synchronization

Under normal conditions, only one master device generates the clock signal, SCL. During the arbitration procedure, however, there are two or more master devices and the clock must be synchronized so that the data output can be compared. The wired-AND property of the clock line means that a device that first generates a low period of the clock line overrules the other devices. At this high/low transition, the clock generators of the other devices are forced to start generation of their own low period. The clock line is then held low by the device with the longest low period, while the other devices that finish their low periods must wait for the clock line to be released before starting their high periods. A synchronized signal on the clock line is thus obtained, where the slowest device determines the length of the low period and the fastest device determines the length of the high period. If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the WAIT-state. In this way a slave can slow down a fast master and the slow device can create enough time to store a received byte or prepare a byte to be transmitted.

Figure 23-10 shows clock synchronization.

Figure 23-10. HS I<sup>2</sup>C Clock Generators Synchronization



### 23.1.2.1.3.11 HS I<sup>2</sup>C External Clock Configuration

Each multimaster HS I<sup>2</sup>C controller is clocked with an independent functional clock (I2Ci\_FCLK) and an interface clock (I2Ci\_ICLK) for interfacing with the L4-PER interconnect. These clocks are provided by the APE PRCM module for I2Ci (where i = 1, 2, 3, 4).

The SYS\_CLK clock provided by the clock generator of the PRCM module is connected to the functional and interface clocks of the HS I<sup>2</sup>C controller I2C5. For detailed information about the module clocking, see [Chapter 3, Power, Reset, and Clock Management](#).

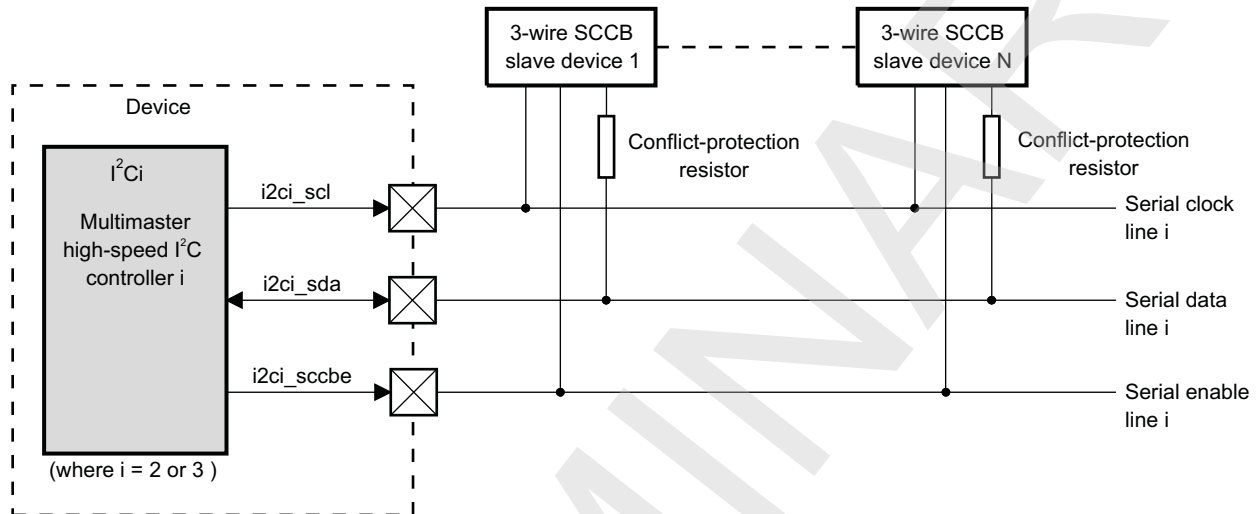
The functional clock is processed by a prescaler block to produce the internal sampling clock. This clock is generated by the I<sup>2</sup>C prescaler block. The prescaler block consists of the I2Ci.I2C\_PSC[7:0] PSC bit field (where i = 1, 2, 3, 4) that is used to divide down the functional clock to obtain an internal sampling clock with a frequency value of I2Ci\_FCLK/(I2Ci.I2C\_PSC[7:0] PSC bit field value + 1, where i = 1, 2, 3, 4).

**NOTE:** The I2Ci.I2C\_PSC[7:0] PSC bit field (where i = 5) of I2C5 is not accessible by software.

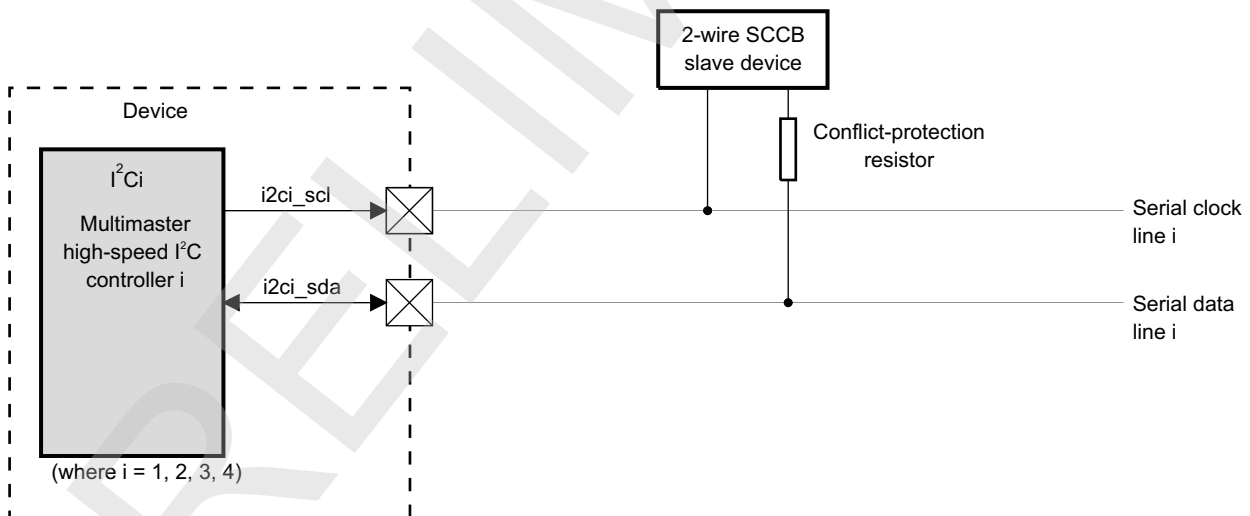
### 23.1.2.2 HS I<sup>2</sup>C in SCCB Mode

The multimaster HS I<sup>2</sup>C controllers support the 2-wire SCCB protocol in master mode as listed in Figure 23-11. Only I2C2 and I2C3 support the 3-wire SCCB protocol in master mode.

**Figure 23-11. HS I<sup>2</sup>C and Typical Connections to SCCB Devices**



(a) Typical connection with 3-wire SCCB devices



(b) Typical connection with 2-wire SCCB devices

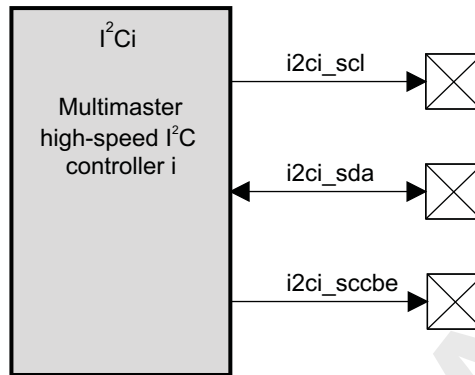
I2C-011

**NOTE:** Only one 2-wire SCCB slave device can be connected to the 2-wire SCCB bus.

#### 23.1.2.2.1 HS I<sup>2</sup>C Pins for Typical Connections in SCCB Mode

Figure 23-12 shows the multimaster HS I<sup>2</sup>C controller pins used for typical connections with 3-wire or 2-wire SCCB devices.

Figure 23-12. HS I<sup>2</sup>C Interface Signals in SCCB Mode



(where i = 1, 2, 3, 4 and i2ci\_sccb\_e is not available for I2C1 and I2C4)

I2C-012

23.1.2.2.2 HS I<sup>2</sup>C SCCB Interface Typical Connections

Table 23-2 lists the pins associated with the SCCB interface.

Table 23-2. HS I<sup>2</sup>C Input/Output

Signal	I/O <sup>(1)</sup>	Description	Reset Value <sup>(2)</sup>	I2Ci.I2C_CON[15] I2C_EN bit = 0
i2ci_scl	O	SCCB serial clock line <sup>(3)</sup> . Standard CMOS output buffer.	Hi-Z	High
i2ci_sda	I/O(OD)	SCCB serial data line. Standard CMOS 3-state output buffer. Requires external conflict-protection resistor for each slave device connected to the bus.	Hi-Z	Hi-Z
i2ci_sccb_e <sup>(4)</sup>	O	SCCB enable line. Standard CMOS output buffer.	High	High

<sup>(1)</sup> I = Input; O = Output; OD = Open Drain

<sup>(2)</sup> Hi-Z = High Impedance

<sup>(3)</sup> This output signal is also used as retiming input.

<sup>(4)</sup> This signal is used for the 3-wire SCCB protocol only.

**NOTE:** Because they share the same ball, the i2c2\_sccb\_e and i2c3\_sccb\_e signals are not available at the same time. For detailed information about pin configuration, see Chapter 18, Control Module.

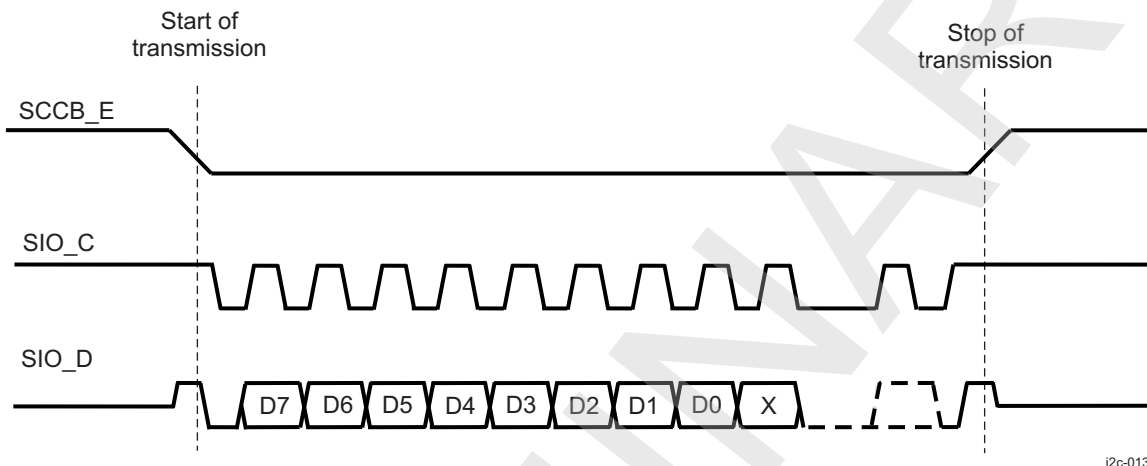
I2C1 and I2C4 do not provide any i2c1\_sccb\_e signal at the chip boundary of the device; thus, these modules do not support the 3-wire SCCB protocol

### 23.1.2.2.3 HS I<sup>2</sup>C SCCB Typical Connection Protocol and Data Format

#### 23.1.2.2.3.1 HS I<sup>2</sup>C SCCB Data Format

If the module is configured in SCCB mode of operation, the data format of transmissions appears as shown in Figure 23-13. For SCCB protocol compatibility I<sup>2</sup>C provides one more signal (SCCB\_E) which acts like a transaction enable signal.

**Figure 23-13. HS I<sup>2</sup>C 3-wire SCCB Transmission Timing Diagram**

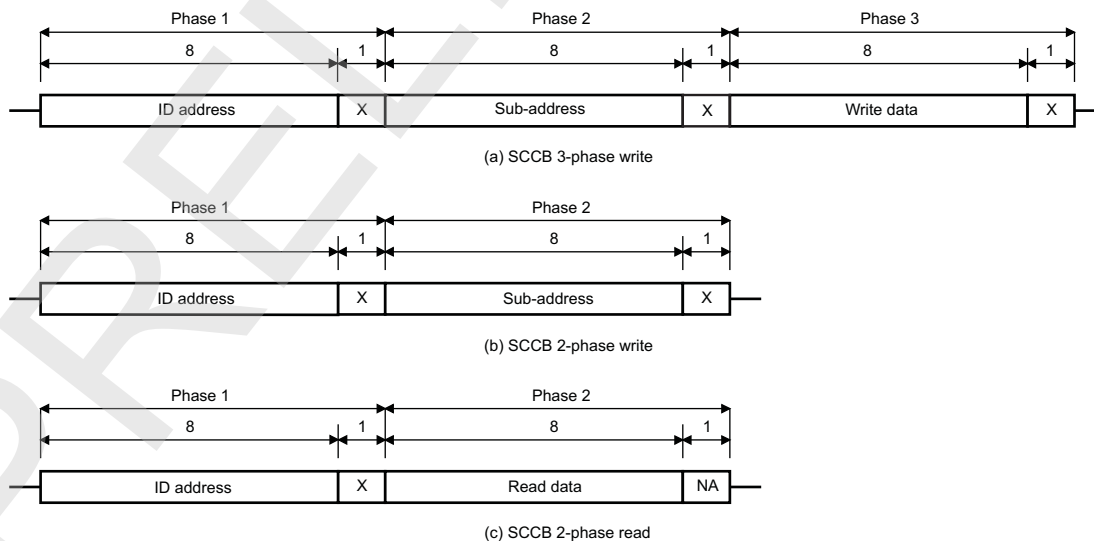


**NOTE:** When operating in 2-wire SCCB mode, the i2ci\_sccb\_e signal is not used by the 2-wire SCCB-compliant slave device attached to the 2-wire SCCB bus.

#### 23.1.2.2.3.2 HS I<sup>2</sup>C SCCB Transmission Data Formats

Figure 23-14 describes the data format of the three kinds of transmission.

**Figure 23-14. HS I<sup>2</sup>C SCCB Transmission Data Formats**



R/W: 0 = Write, 1 = Read

X: don't care

NA: ninth bit of a read phase. This bit must be set to 1 by the master device.

I2C-014

The basic element of a data transmission is a phase. A phase contains 9 bits, which consist of an 8-bit sequential data transmission followed by a ninth bit. The ninth bit is a don't-care bit (X) or an NA bit, depending on whether the data transmission is a write or a read. The maximum number of phases that can be included in a transmission is three. The MSB is always asserted first for each phase.

A data transmission is one of three types:

- 3-phase write transmission:

The 3-phase write transmission cycle (see (a) of [Figure 23-14](#)) is a full write operation in which the master can write 1 byte of data to a specific slave(s). The 7-bit slave address in the ID value identifies the specific slave that the master intends to access. The subaddress identifies the register location of the specified slave. The write data contains 8-bit data that the master intends to write over the content of this specific address. The ninth bit of each of the three phases is a don't-care bit (X bit).

- 2-phase write transmission:

The 2-phase write transmission cycle is followed by a 2-phase read transmission cycle (see below). The purpose of issuing a 2-phase write transmission cycle (see (b) of [Figure 23-14](#)) is to identify the subaddress of some specific slave from which the master intends to read data for the following 2-phase read transmission cycle. The ninth bit of each phase is a don't-care bit (X bit).

- 2-phase read transmission:

Either a 3-phase or a 2-phase write transmission cycle must be asserted ahead of a 2-phase read transmission cycle. The 2-phase read transmission cycle (see (c) of [Figure 23-14](#)) has no ability to identify the subaddress. The 2-phase write transmission cycle contains read data of 8 bits and a ninth don't-care bit or NA bit. The master must drive the NA bit at logical 1.

In each transmission type, phase 1 (the 7-bit slave address of the ID value) is asserted by the master to identify the selected slave to which the data is read or written. Each slave has a unique 7-bit slave address. The 7-bit slave address of the ID value comprises 7 bits, from bit 7 to bit 1, that can identify up to 128 slaves. The eighth bit, bit 0, is the read/write selector bit that specifies the transmission direction of the current cycle. A logical 0 represents a write cycle and a logical 1 represents a read cycle. The ninth bit of phase 1 is a don't-care bit (X bit).

Phase 2 (subaddress/read data) is asserted by the master (subaddress) or the slave(s) (read data). A phase 2 transmission asserted by the master identifies the subaddress of the slave(s) the master intends to access. A phase 2 transmission asserted by the slave(s) indicates the read data that the master will receive. The slave(s) recognize the subaddress of this read data according to the previous 3-phase or 2-phase write transmission cycle. The ninth bit is defined as a don't-care bit (X bit) when the master asserts phase 2. The ninth bit is defined as an NA bit when the slave(s) asserts the phase 2 transmission. The master is responsible for a logical 1 during the period of the NA bit.

Phase 3 (write data) is asserted only by the master. This phase contains the data the master intends to write to the slave(s). Because the master asserts the transmission, the ninth bit of the phase 3 transmission is defined as a don't-care bit (X bit).

---

**NOTE:** A multimaster HS I<sup>2</sup>C controller configured in SCCB mode can perform two operations:

- Writing a single byte to an SCCB slave device by using the 3-phase write transmission cycle
  - Reading a single byte from an SCCB slave device by using the 2-phase write transmission cycle followed by the 2-phase read transmission cycle
- 

### 23.1.2.2.3.3 HS I<sup>2</sup>C Data Validity

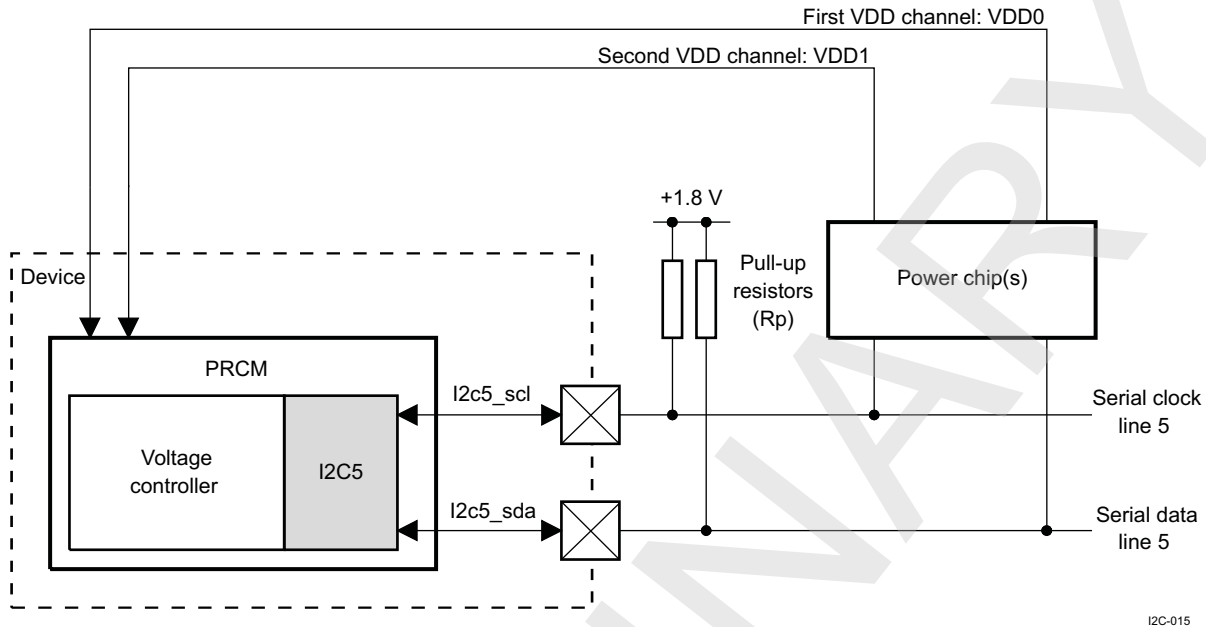
The data validity is the same as described in [Section 23.1.2.1.3.2, I<sup>2</sup>C Data Validity](#).

### 23.1.2.2.3.4 HS I<sup>2</sup>C S and P Conditions

The S and P conditions are the same as described in [Section 23.1.2.1.3.3, I<sup>2</sup>C Start and Stop Conditions](#).

## 23.1.2.3 HS I<sup>2</sup>C for Communication With Power Chip(s)

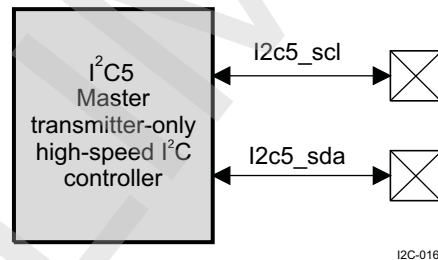
[Figure 23-15](#) shows a typical connection between the master transmitter HS I<sup>2</sup>C controller I2C5 of the device and external power chip(s).

**Figure 23-15. HS I<sup>2</sup>C and Typical Connection Between the HS I<sup>2</sup>C and Power Chip(s)**


I2C-015

### 23.1.2.3.1 HS I<sup>2</sup>C I2C5 Pins for Typical Connections

Figure 23-16 shows the HS I<sup>2</sup>C controller I2C5 pins used for typical connections with power chips.

**Figure 23-16. HS I<sup>2</sup>C I2C5 Interface Signals**


I2C-016

### 23.1.2.3.2 HS I<sup>2</sup>C I2C5 Interface Typical Connections

The master transmitter HS I<sup>2</sup>C controller I2C5 interfaces between the and external power chip(s) for voltage control. This module is always configured as an I<sup>2</sup>C master transmitter; it does not support the SCCB protocol.

Table 23-3 lists the pins associated with the I<sup>2</sup>C interface of the HS I<sup>2</sup>C controller I2C5 of the device.

**Table 23-3. HS I<sup>2</sup>C Input/Output Description**

Signal	I/O <sup>(1)</sup>	Description	Reset Value
i2c5_scl	I/O(OD)	I <sup>2</sup> C serial clock line <sup>(2)</sup> . Open-drain output buffer. Requires external Rp.	1
i2c5_sda	I/O(OD)	I <sup>2</sup> C serial data line. Open-drain output buffer. Requires external Rp.	1

<sup>(1)</sup> I = Input; O = Output; OD = Open Drain

<sup>(2)</sup> This signal is also used as re-timing input.



### **23.1.2.3.3 HS I<sup>2</sup>C Typical Connections Protocol and Data Format for I2C5**

#### **23.1.2.3.3.1 HS I<sup>2</sup>C Serial Data Format for I2C5**

The serial data format is the same as described in [Section 23.1.2.1.3.1](#), *I<sup>2</sup>C Serial Data Format*.

#### **23.1.2.3.3.2 HS I<sup>2</sup>C Data Validity for I2C5**

The data validity is the same as described in [Section 23.1.2.1.3.2](#), *I<sup>2</sup>C Data Validity*.

#### **23.1.2.3.3.3 HS I<sup>2</sup>C S and P Conditions for I2C5**

The S and P conditions are the same as described in [Section 23.1.2.1.3.3](#), *I<sup>2</sup>C Start and Stop Conditions*.

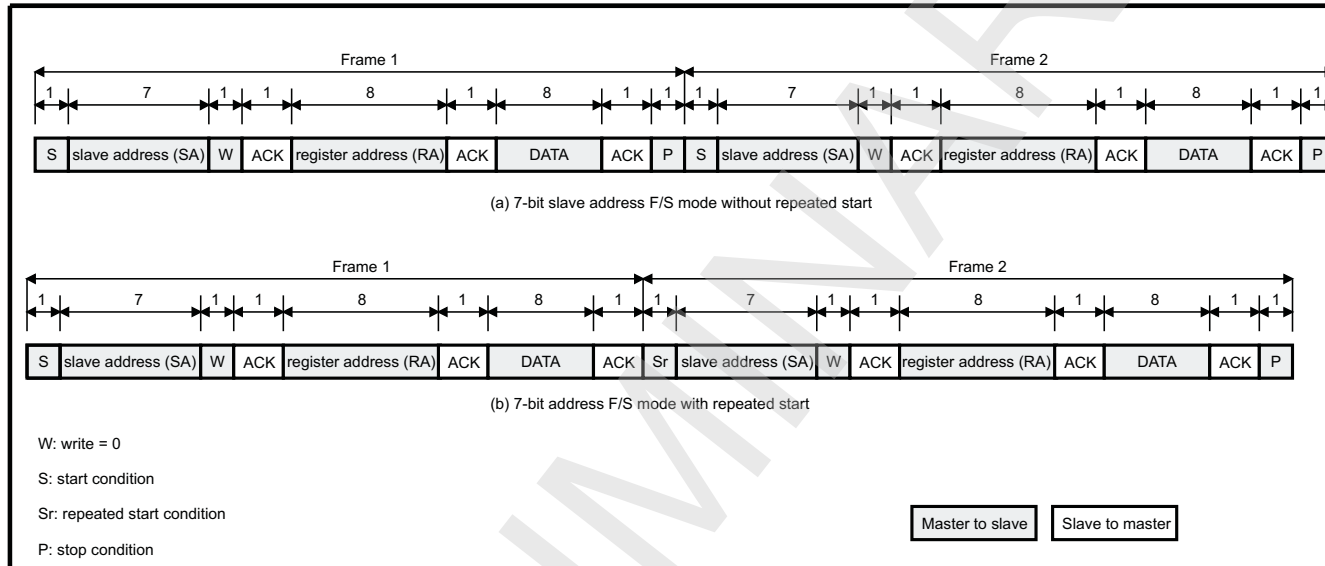
#### **23.1.2.3.3.4 HS I<sup>2</sup>C Addressing for I2C5**

The master transmitter HS I<sup>2</sup>C controller I2C5 supports only the 7-bit addressing mode. For each frame, the master writes the 8-bit value (DATA) in the register specified by the 8-bit register address (RA) of the slave addressed by the slave address (SA).

23.1.2.3.3.4.1 HS I<sup>2</sup>C Data Transfer Format in F/S Mode

Figure 23-17 shows the I<sup>2</sup>C data transfer format in F/S mode for I2C5.

Figure 23-17. HS I<sup>2</sup>C Data Transfer Format in F/S Mode for I2C5

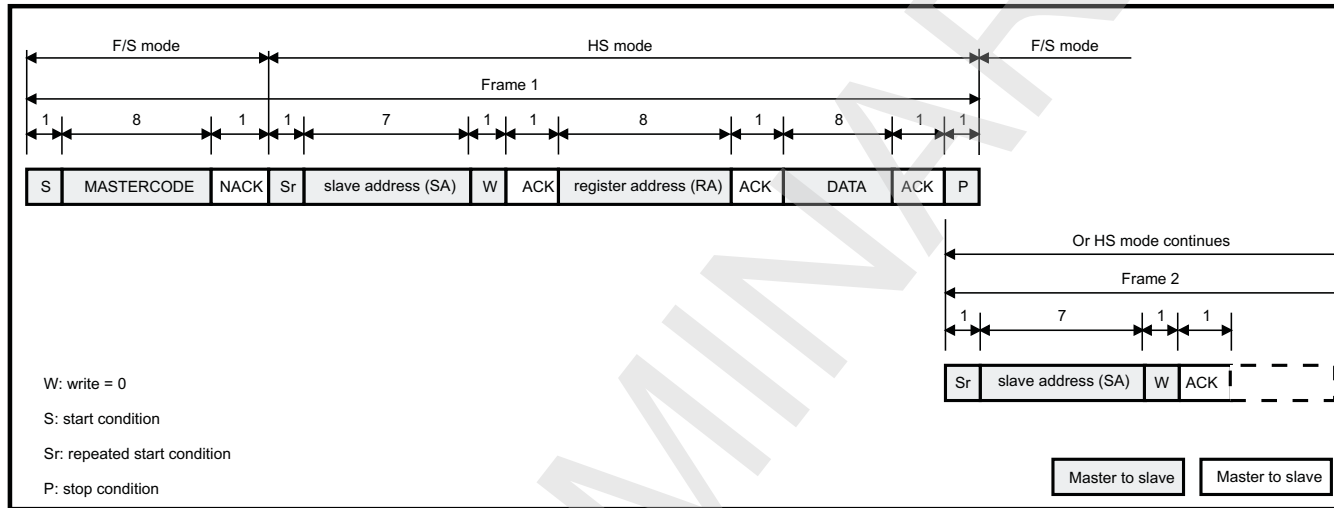


I2C-017

23.1.2.3.3.4.2 Data Transfer Format in HS Mode

Figure 23-18 shows the I<sup>2</sup>C data transfer format in HS mode for I2C5.

Figure 23-18. HS I<sup>2</sup>C Data Transfer Format in HS Mode for the I2C5 Module



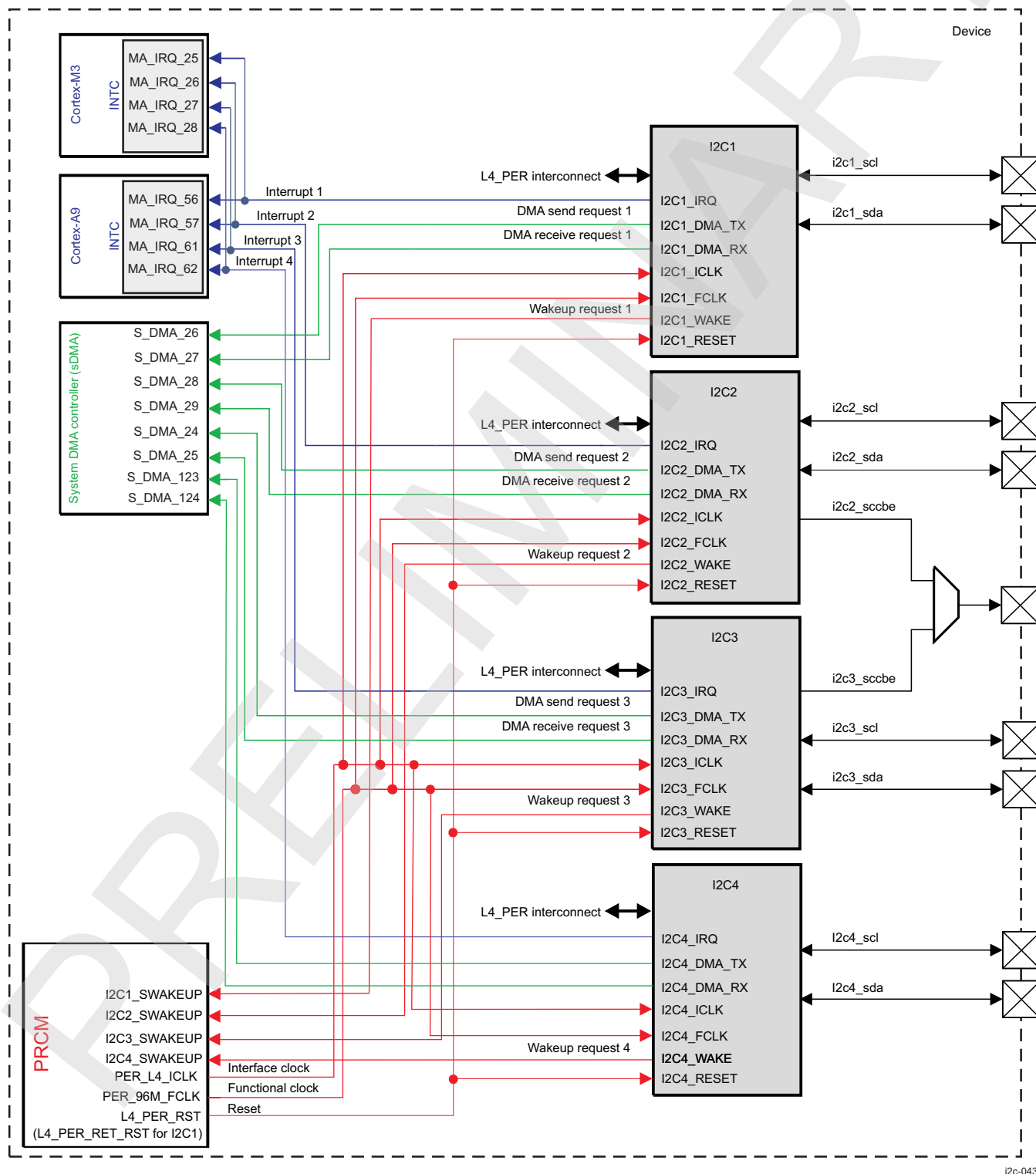
I2C-018

### 23.1.3 HS I<sup>2</sup>C Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 23-19 shows the integration of the four HS I<sup>2</sup>C controllers in the device.

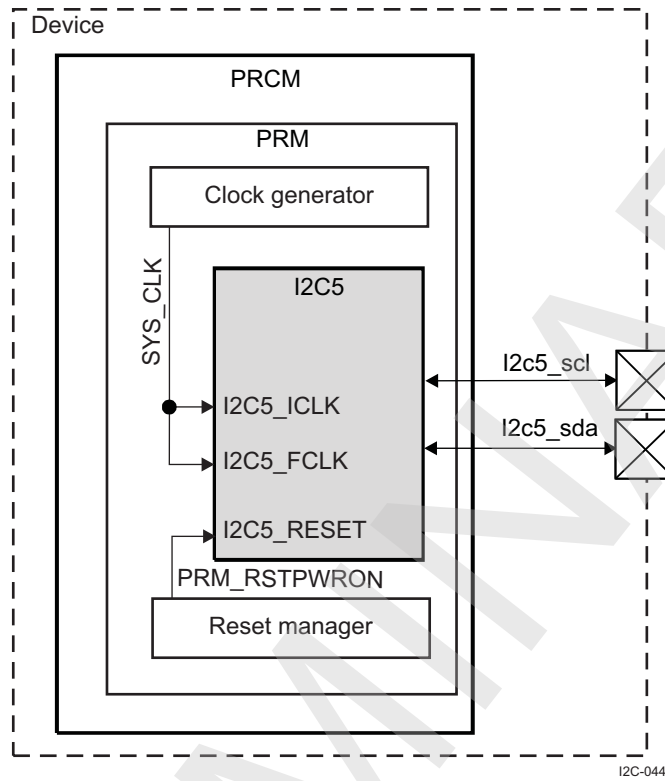
Figure 23-19. HS I<sup>2</sup>C Integration



i2c-043

Figure 23-20 shows HS I<sup>2</sup>C integration of I2C5 inside the PRCM module.

Figure 23-20. HS I<sup>2</sup>C I2C5 Integration



**NOTE:** For more information about the IDLE hardware handshake and the wake-up request, see [Section 3.7, Power Management Functional Description](#), in [Chapter 3, Power, Reset, and Clock Management](#).

Table 23-4 through Table 23-6 summarize the integration of the module in the device.

Table 23-4. HS I<sup>2</sup>C Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
I2C1	PD_L4_PER	L4_PER
I2C2	PD_L4_PER	L4_PER
I2C3	PD_L4_PER	L4_PER
I2C4	PD_L4_PER	L4_PER

Table 23-5. HS I<sup>2</sup>C Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Clocks	
			Source	Description
I2C1	I2C1_ICLK	PER_L4_ICLK	PRCM	I2C1 interface clock
	I2C1_FCLK	PER_96M_FCLK	PRCM	I2C1 functional clock
I2C2	I2C2_ICLK	PER_L4_ICLK	PRCM	I2C2 interface clock
	I2C2_FCLK	PER_96M_FCLK	PRCM	I2C2 functional clock
I2C3	I2C3_ICLK	PER_L4_ICLK	PRCM	I2C3 interface clock
	I2C3_FCLK	PER_96M_FCLK	PRCM	I2C3 functional clock

**Table 23-5. HS I<sup>2</sup>C Clocks and Resets (continued)**

I2C4	I2C4_ICLK	PER_L4_ICLK	PRCM	I2C4 interface clock
	I2C4_FCLK	PER_96M_FCLK	PRCM	I2C4 functional clock
<b>Resets</b>				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
I2C1	I2C1_RESET	L4_PER_RET_RST	PRCM	I2C1 reset
I2C2	I2C2_RESET	L4_PER_RST	PRCM	I2C2 reset
I2C3	I2C3_RESET	L4_PER_RST	PRCM	I2C3 reset
I2C4	I2C4_RESET	L4_PER_RST	PRCM	I2C4 reset

**Table 23-6. HS I<sup>2</sup>C Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
I2C1	I2C1_IRQ	MA_IRQ_56	Cortex-A9 INTC	I2C1 interrupt request
	I2C1_IRQ	MM_IRQ_25	Cortex-M3 INTC	I2C1 interrupt request
I2C2	I2C2_IRQ	MA_IRQ_57	Cortex-A9 INTC	I2C2 interrupt request
	I2C2_IRQ	MM_IRQ_26	Cortex-M3 INTC	I2C2 interrupt request
I2C3	I2C3_IRQ	MA_IRQ_61	Cortex-A9 INTC	I2C3 interrupt request
	I2C3_IRQ	MM_IRQ_27	Cortex-M3 INTC	I2C3 interrupt request
I2C4	I2C4_IRQ	MA_IRQ_62	Cortex-A9 INTC	I2C4 interrupt request
	I2C4_IRQ	MM_IRQ_28	Cortex-M3 INTC	I2C4 interrupt request
DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
I2C1	I2C1_DMA_TX	S_DMA_26	sDMA	I2C1 DMA transmit request
	I2C1_DMA_RX	S_DMA_27	sDMA	I2C1 DMA receive request
I2C2	I2C2_DMA_TX	S_DMA_28	sDMA	I2C2 DMA transmit request
	I2C2_DMA_RX	S_DMA_29	sDMA	I2C2 DMA receive request
I2C3	I2C3_DMA_TX	S_DMA_24	sDMA	I2C3 DMA transmit request
	I2C3_DMA_RX	S_DMA_25	sDMA	I2C3 DMA receive request
I2C4	I2C4_DMA_TX	S_DMA_123	sDMA	I2C4 DMA transmit request
	I2C4_DMA_RX	S_DMA_124	sDMA	I2C4 DMA receive request

**NOTE:**

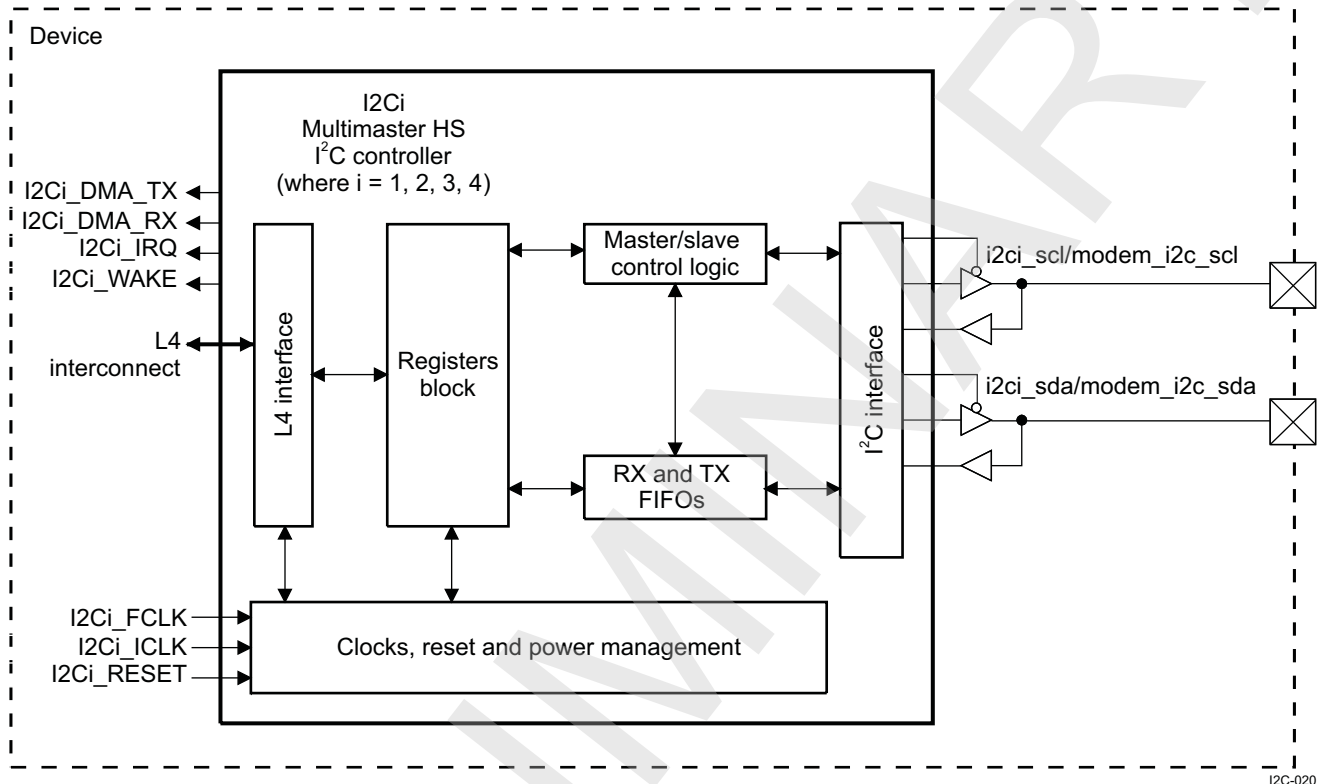
- For a description of interrupt source, see [Table 23-12, Interrupt Requests](#).
- For a description of DMA source, see [Table 23-13, DMA Requests](#).

### 23.1.4 HS I<sup>2</sup>C Functional Description

#### 23.1.4.1 HS I<sup>2</sup>C Block Diagram

Figure 23-21 is the multimaster I<sup>2</sup>C HS controller block diagram.

Figure 23-21. HS I<sup>2</sup>C Block Diagram



The five multimaster HS I<sup>2</sup>C controllers can be configured in F/S I<sup>2</sup>C mode, HS I<sup>2</sup>C mode, or SCCB mode. The operation mode is selected by configuring the I2Ci.I2C\_CON[13:12] OPMODE bit field. Table 23-7 lists the available operation modes.

**NOTE:** The fifth master transmitter HS I<sup>2</sup>C interface (I2C5) in the power, reset, and clock management (PRCM) could also be configured in High-speed mode.

Table 23-7. HS I<sup>2</sup>C Operation Mode Selection

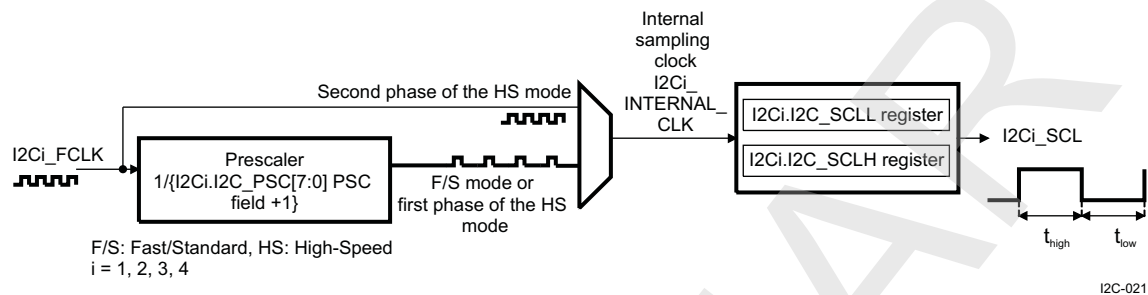
Operation Mode	Value of I2Ci.I2C_CON[13:12] OPMODE Bit Field
F/S I <sup>2</sup> C	0x0
HS I <sup>2</sup> C	0x1
SCCB	0x2
Reserved (not used)	0x3

### 23.1.4.2 HS I<sup>2</sup>C Clocks

#### 23.1.4.2.1 HS I<sup>2</sup>C Clocking

Figure 23-22 shows the I<sup>2</sup>C clock generation of the HS I<sup>2</sup>C controllers.

**Figure 23-22. HS I<sup>2</sup>C Clock Generation**



Each multimaster HS I<sup>2</sup>C controller uses the I2Ci\_FCLK functional clock in the PRCM module. The internal sampling clock I2Ci\_INTERNAL\_CLK is generated by dividing the functional clock by (the I2Ci.I2C\_PSC[7:0] PSC bit field value + 1) in F/S mode, in SCCB mode, or in the first phase of HS mode; or by directly using the functional clock in the second phase of HS mode (prescaler is bypassed).

The low time of the I2Ci\_SCL signal is determined by the I2Ci.I2C\_SCLL[7:0] SCLL bit field in F/S mode, in SCCB mode, or in the first phase of HS mode; or by the I2Ci.I2C\_SCLL[15:8] HSSCLL bit field in the second phase of HS mode.

The high time of the I2Ci\_SCL signal is determined by the I2Ci.I2C\_SCLH[7:0] SCLH bit field in F/S mode, in SCCB mode, or in the first phase of HS mode; or by the I2Ci.I2C\_SCLH[15:8] HSSCLH bit field in the second phase of HS mode.

Table 23-8 lists the  $t_{LOW}$  and  $t_{high}$  values in master mode only (in slave mode, the I<sup>2</sup>C controller does not generate the I<sup>2</sup>C clock).

**Table 23-8. HS I<sup>2</sup>C  $t_{LOW}$  and  $t_{high}$  Values of the I<sup>2</sup>C Clock**

Mode	I2Ci_INTERNAL_CLK	$t_{LOW}$	$t_{high}$
F/S, SCCB, or HS first phase	I2Ci_FCLK / (I2Ci.I2C_PSC[7:0] PSC bit field + 1)	(I2Ci.I2C_SCLL[7:0] SCLL bit field value + 7) x I2Ci_INTERNAL_CLK period	(I2Ci.I2C_SCLH[7:0] SCLH bit field value + 5) x I2Ci_INTERNAL_CLK period
HS second phase	I2Ci_FCLK	(I2Ci.I2C_SCLL[15:8] HSSCLL bit field value + 7) x I2Ci_INTERNAL_CLK period	(I2Ci.I2C_SCLH[15:8] HSSCLH bit field value + 5) x I2Ci_INTERNAL_CLK period

**NOTE:** For HS mode, the I2Ci.I2C\_SCLL[15:8] HSSCLL and I2Ci.I2C\_SCLL[7:0] SCLL bit fields must be programmed (the first phase of an HS transaction is performed at F/S speed).

For HS mode, the I2Ci.I2C\_SCLH[15:8] HSSCLH and I2Ci.I2C\_SCLH[7:0] SCLH bit fields must be programmed (the first phase of an HS transaction is performed at F/S speed).

**NOTE:** The equations in Table 23-8 give the SCL timing values for SCLL/SCLH/HSSCLL/HSSCLH at HS I<sup>2</sup>C controller outputs. Actual  $t_{low}$  and  $t_{high}$  periods may vary depending on the board (the load capacitance on the SCL signal). If necessary, any adjustments to the SCLL/SCLH/HSSCLL/HSSCLH values must be determined by measurements of actual SCL signal on the board



### CAUTION

During active mode (I2Ci.I2C\_CON[15] I2C\_EN bit is set to 1), make no changes to the I2Ci.I2C\_SCLL and I2Ci.I2C\_SCLH registers. Changes may result in unpredictable behavior.

**NOTE:** Each multimaster HS I<sup>2</sup>C controller can be used with an internal secondary pullup. This pullup is mandatory when the I<sup>2</sup>C controller is configured in HS mode for a bit rate of 3.4 Mbps, and the bus line capacitance exceeds 45 pF. Pullups can be programmed through:

- CCONTROL\_I2C\_0[0] I2C1\_SCL\_PULLUPRESX bit or CCONTROL\_I2C\_0[16] I2C1\_SDA\_PULLUPRESX bit for I2C1
- CONTROL\_I2C\_0[4] I2C2\_SCL\_PULLUPRESX bit or CONTROL\_I2C\_0[20] I2C2\_SDA\_PULLUPRESX bit for I2C2
- CONTROL\_I2C\_0[8] I2C3\_SCL\_PULLUPRESX bit or CONTROL\_I2C\_0[24] I2C3\_SDA\_PULLUPRESX bit for I2C3
- CONTROL\_I2C\_0[12] I2C4\_SCL\_PULLUPRESX bit or CONTROL\_I2C\_0[28] I2C4\_SDA\_PULLUPRESX bit for I2C4

The maximum bit rate specified by the SCCB specifications is 100 Kbps.

For more information see [Section 18.4.12.6.9](#), *I2Cx I/Os Group Pullupresx Controls and Load Range Settings*.

Table 23-9 lists the register values for obtaining the maximum I<sup>2</sup>C bit rates and the maximum period of the filtered spikes in F/S mode and HS mode.

**Table 23-9. HS I<sup>2</sup>C Register Values for Maximum I<sup>2</sup>C Bit Rates in I<sup>2</sup>C F/S, I<sup>2</sup>C HS, and SCCB Modes<sup>(1)</sup>**

	I <sup>2</sup> C Mode for I2Ci, where i = 1, 2, 3, 4				Description	SCCB Mode for I2Ci, where i = 2, 3		I <sup>2</sup> C Mode for I2C5  Fast and High-Speed Mode	
	Standard Mode	Fast Mode	Fast Mode+	High-Speed Mode					
I2Ci_FCLK frequency (MHz)	96					96		12, 16.8, 19.2, 26, or 38.4	SYS_CLK clock frequency
<b>I2Ci.I2C_PSC[7:0] PSC field value</b>	<b>23</b>	<b>9</b>	<b>5</b>	<b>1</b>	Prescaler value for F/S and HS modes	<b>4</b>	Prescaler value for SCCB mode	Not accessible by software	
I2Ci_INTERNAL_CLK frequency (MHz)	4	9.6	19.2	96	19.2				
<b>I2Ci.I2C_SCLL[7:0] SCLL field value</b>	<b>13</b>	<b>5</b>	<b>3</b>	<b>113</b>	Value for F/S mode and first phase of HS mode	<b>89</b>	Value for SCCB mode	<b>PRM_VC_CFG_I2C_CLK[15:8] SCLL</b> (see Chapter 3, Power, Reset, and Clock Management)	Value for F/S mode and first phase of HS mode
<b>I2Ci.I2C_SCLH[7:0] SCLH field value</b>	<b>15</b>	<b>7</b>	<b>4</b>	<b>115</b>	Value for F/S mode and first phase of HS mode	<b>91</b>	Value for SCCB mode	<b>PRM_VC_CFG_I2C_CLK[7:0] SCLH</b> (see Chapter 3, Power, Reset, and Clock Management)	Value for F/S mode and first phase of HS mode
Maximum bit rate (Mbps)	0.1	0.4	1.0	0.4	F/S mode and first phase in HS mode maximum bit rate	0.1	SCCB mode maximum bit rate	0.4	F/S mode and first phase in HS mode maximum bit rate
Maximum filter period (ns)	250	104	52	10		50		50	
<b>I2Ci.I2C_SCLL[15:8] HSSCLL field value</b>				<b>7</b>	Values for second phase of HS mode			<b>PRM_VC_CFG_I2C_CLK[31:24] HSSCLL</b> (see Chapter 3, Power, Reset, and Clock Management)	Values for second phase of HS mode
<b>I2Ci.I2C_SCLH[15:8] HSSCLH field value</b>				<b>10</b>	Values for second phase of HS mode			<b>PRM_VC_CFG_I2C_CLK[23:16] HSSCLH</b> (see Chapter 3, Power, Reset, and Clock Management)	Values for second phase of HS mode
HS mode maximum bit rate (Mbps)				3.31	HS mode maximum bit rate			0.8, 1.28, 1.73, 2.56, 3.31	HS mode maximum bit rate according to the corresponding SYS_CLK clock frequency
Maximum filter period (ns)				10				10	

<sup>(1)</sup> Programmable fields are in bold.

---

**NOTE:** This table presents informative values only for the configuration parameters and the I<sup>2</sup>C bus performance obtained according to these values. The delays added by the analog pads are not considered in these figures.

---



---

**NOTE:** I<sup>2</sup>Ci\_INTERNAL\_CLK freq = SYS\_CLK / PSC  
 F/S filter period = 1 / 2Ci\_INTERNAL\_CLK  
 HS filter period = 1 / SYS\_CLK freq  
 HS bit rate = SYS\_CLK freq / (HSSCLL + 7 + HSSCLH + 5)  
 F/S bit rate = 2Ci\_INTERNAL\_CLK / (SCLL + 7 + SCLH + 5)

---

PRELIMINARY

### 23.1.4.2.2 HS I<sup>2</sup>C Automatic Blocking of the I<sup>2</sup>C Clock Feature (I<sup>2</sup>C Mode Only)

This feature offers the possibility for the LH to command the blocking of the I<sup>2</sup>C clock after the slave addressing phase, when the I<sup>2</sup>C controller is addressed by an external master device using a certain Own Address.

The release of the I<sup>2</sup>C clock can be performed independently for each Own Address (I2Ci.I2C\_OA<sub>i</sub>, where i = 1, 2, 3, 4 and I2Ci.I2C\_OAx registers, where x = 1, 2, 3) by deasserting the corresponding bit in the I2Ci.I2C\_SBLOCK register.

### 23.1.4.3 HS I<sup>2</sup>C Software Reset

Each multimaster HS I<sup>2</sup>C controller supports the software reset by accessing the I2Ci.I2C\_SYSC[1] SRST bit (1: reset; 0: normal mode).

The software reset status can be checked by accessing the I2Ci.I2C\_SYSS[0] RDONE bit (1: reset is done; 0: reset is ongoing).

To perform a software reset:

1. Ensure that the module is disabled (clear the I2Ci.I2C\_CON[15] I2C\_EN bit to 0).
2. Set the I2Ci.I2C\_SYSC[1] SRST bit to 1.
3. Enable the module by setting I2Ci.I2C\_CON[15] I2C\_EN bit to 1.
4. Check the I2Ci.I2C\_SYSS[0] RDONE bit until it is set to 1 to indicate the software reset is complete.

---

**NOTE:** The I2Ci.I2C\_CON[15] I2C\_EN bit can hold the functional clock domain of the multimaster HS I<sup>2</sup>C controller in reset after the device reset has been released. When the system bus reset is removed, this bit remains cleared. The functional part of the I<sup>2</sup>C controller is held in reset state while this bit is 0, and all configuration registers can be accessed.

The I2Ci.I2C\_CON[15] I2C\_EN bit must be set to 1 to enable the functional part of the I<sup>2</sup>C controller.

The I2Ci.I2C\_SYSS[0] RDONE bit is asserted only after the module is enabled by setting the I2Ci.I2C\_CON[15] I2C\_EN bit to 1.

---

### 23.1.4.4 HS I<sup>2</sup>C Power Management

Table 23-10 describes power-management features available for the multimaster HS I<sup>2</sup>C controllers.

**NOTE:**

- For information about source clock gating and sleep/wake-up transitions description, see Section 3.1.1.1.2, *Module Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.
  - For descriptions of EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see Section 3.1.1.1.2, *Module Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.
- 

**Table 23-10. HS I<sup>2</sup>C Local Power-Management Features**

Feature	Registers	Description
Clock auto gating	I2Ci.I2C_SYSC[0] AUTOIDLE	This bit allows a local power optimization inside the module
Slave idle modes	I2Ci.I2C_SYSC[4:3] IDLEMODE	Force-idle, no-idle, smart-idle, and smart-idle wake-up-capable modes are available.
Clock activity	I2Ci.I2C_SYSC[9:8] CLOCKACTIVITY	For configuration details, see Table 23-11.
Global wake-up enable	I2Ci.I2C_SYSC[2] ENAWAKEUP	This bit enables the wake-up feature at module level.

**NOTE:** The voltage controllers, in which the HS I<sup>2</sup>C controller I2C5 are implemented, have no idle request/acknowledge mechanism. The idle modes for the voltage controllers are directly managed by the PRCM module.

**Table 23-11. HS I<sup>2</sup>C Clock Activity Settings**

I2Ci.I2C_SYSC[9:8] CLOCKACTIVITY	Clock State When Module is in IDLE State		Features Available/Unavailable When Module is in IDLE State
	I2Ci_ICLK	I2C_FCLK	
00	OFF	OFF	Both clocks are disabled
10	OFF	ON	Interface clock is disabled; functional clock is enabled
01	ON	OFF	Functional clock is disabled; interface clock is enabled
11	ON	ON	Both clocks are enabled

**CAUTION**

The PRCM module has no hardware means of reading the settings of CLOCKACTIVITY. Thus, software must ensure consistent programming between the I2C CLOCKACTIVITY and I2C clock PRCM control bits. For a description of the ClockActivity feature, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

**23.1.4.5 HS I<sup>2</sup>C Interrupt Requests**

[Table 23-12](#) lists the event flags, and their mask, that can cause module interrupts.

**Table 23-12. HS I<sup>2</sup>C Events**

Event Flag	Event Mask	Event Unmask	Map to	Description
I2Ci.I2C_IRQSTATUS[0] AL	I2Ci.I2C_IRQENABLE_SET[0] AL_IE	I2Ci.I2C_IRQENABLE_C LR[0] AL_IE	I2Ci_IRQ	Arbitration lost IRQ enabled status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to the MPU subsystem. During reads, it always returns 0.
I2Ci.I2C_IRQSTATUS[1] NACK	I2Ci.I2C_IRQENABLE_SET[1] NACK_IE	I2Ci.I2C_IRQENABLE_C LR[1] NACK_IE	I2Ci_IRQ	No acknowledgment IRQ enabled status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPU subsystem. Write 1 to clear this bit.
I2Ci.I2C_IRQSTATUS[2] ARDY	I2Ci.I2C_IRQENABLE_SET[2] ARDY_IE	I2Ci.I2C_IRQENABLE_C LR[2] ARDY_IE	I2Ci_IRQ	Register access ready IRQ enabled status. When set to 1 it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to MPU subsystem. Write 1 to clear.
I2Ci.I2C_IRQSTATUS[3] RRDY	I2Ci.I2C_IRQENABLE_SET[3] RRDY_IE	I2Ci.I2C_IRQENABLE_C LR[3] RRDY_IE	I2Ci_IRQ	Receive data ready IRQ enabled status. Set to 1 by core when receiver mode, a new data is able to be read. When set to 1 by core, an interrupt is signaled to MPU subsystem. Write 1 to clear.

**Table 23-12. HS I<sup>2</sup>C Events (continued)**

Event Flag	Event Mask	Event Unmask	Map to	Description
I2Ci.I2C_IRQSTATUS[4] XRDY	I2Ci.I2C_IRQENABLE_SET[4] XRDY_IE	I2Ci.I2C_IRQENABLE_C LR[4] XRDY_IE	I2Ci_IRQ	Transmit data ready IRQ enabled status. Set to 1 by core when transmitter and when new data is requested. When set to 1 by core, an interrupt is signaled to MPU subsystem. Write 1 to clear.
I2Ci.I2C_IRQSTATUS[5] GC	I2Ci.I2C_IRQENABLE_SET[5] GC_IE	I2Ci.I2C_IRQENABLE_C LR[5] GC_IE	I2Ci_IRQ	General call IRQ enabled status. Set to 1 by core when General call address detected and interrupt signaled to MPU subsystem. Write 1 to clear.
I2Ci.I2C_IRQSTATUS[6] STC	I2Ci.I2C_IRQENABLE_SET[6] STC_IE	I2Ci.I2C_IRQENABLE_C LR[6] STC_IE	I2Ci_IRQ	Start Condition IRQ enabled status.
I2Ci.I2C_IRQSTATUS[7] AERR	I2Ci.I2C_IRQENABLE_SET[7] AERR_IE	I2Ci.I2C_IRQENABLE_C LR[7] AERR_IE	I2Ci_IRQ	Access Error IRQ enabled status.
I2Ci.I2C_IRQSTATUS[8] BF	I2Ci.I2C_IRQENABLE_SET[8] BF_IE	I2Ci.I2C_IRQENABLE_C LR[8] BF_IE	I2Ci_IRQ	Access Error IRQ enabled status.
I2Ci.I2C_IRQSTATUS[9] AAS	I2Ci.I2C_IRQENABLE_SET[9] AAS_IE	I2Ci.I2C_IRQENABLE_C LR[9] AAS_IE	I2Ci_IRQ	Address recognized as slave IRQ enabled status.
I2Ci.I2C_IRQSTATUS[10] XUDF	I2Ci.I2C_IRQENABLE_SET[1 0] XUDF_IE	I2Ci.I2C_IRQENABLE_C LR[10] XUDF_IE	I2Ci_IRQ	Transmit underflow enabled status. Writing into this bit has no effect.
I2Ci.I2C_IRQSTATUS[11] ROVR	I2Ci.I2C_IRQENABLE_SET[1 1] ROVR_IE	I2Ci.I2C_IRQENABLE_C LR[11] ROVR_IE	I2Ci_IRQ	Receive overrun enabled status. Writing into this bit has no effect.
I2Ci.I2C_IRQSTATUS[12] BB	I2Ci.I2C_IRQENABLE_SET[1 2] BB_IE	I2Ci.I2C_IRQENABLE_C LR[12] BB_IE	I2Ci_IRQ	Bus busy enabled status. Writing into this bit has no effect.
I2Ci.I2C_IRQSTATUS[13] RDR	I2Ci.I2C_IRQENABLE_SET[1 3] RDR_IE	I2Ci.I2C_IRQENABLE_C LR[13] RDR_IE	I2Ci_IRQ	Receive draining IRQ enabled status.
I2Ci.I2C_IRQSTATUS[14] XDR	I2Ci.I2C_IRQENABLE_SET[1 4] XDR_IE	I2Ci.I2C_IRQENABLE_C LR[14] XDR_IE	I2Ci_IRQ	Transmit draining IRQ enabled status.

#### 23.1.4.6 HS I<sup>2</sup>C DMA Requests

Each multimaster HS I<sup>2</sup>C controller can generate two DMA requests to the system DMA (sDMA) controller. [Table 23-13](#) lists the DMA requests with mapping on the sDMA controller. For information about DMA generation, see [Section 23.1.4.8.3, HS I<sup>2</sup>C FIFO DMA Mode \(I<sup>2</sup>C Mode Only\)](#).

**Table 23-13. HS I<sup>2</sup>C DMA Requests**

Name	Source	Destination	Description
I2C1_DMA_TX	I2C1	S_DMA_26	I2C1 DMA write request to inform the sDMA to write new data in the I2C1.I2C_DATA[7:0] register
I2C1_DMA_RX	I2C1	S_DMA_27	I2C1 DMA read request to inform the sDMA to read the data in the I2C1.I2C_DATA[7:0] register
I2C2_DMA_TX	I2C2	S_DMA_28	I2C2 DMA write request to inform the sDMA to write new data in the I2C2.I2C_DATA[7:0] register
I2C2_DMA_RX	I2C2	S_DMA_29	I2C2 DMA read request to inform the sDMA to read the data in the I2C2.I2C_DATA[7:0] register
I2C3_DMA_TX	I2C3	S_DMA_24	I2C3 DMA write request to inform the sDMA to write new data in the I2C3.I2C_DATA[7:0] register
I2C3_DMA_RX	I2C3	S_DMA_25	I2C3 DMA read request to inform the sDMA to read the data in the I2C3.I2C_DATA[7:0] register

**Table 23-13. HS I<sup>2</sup>C DMA Requests (continued)**

Name	Source	Destination	Description
I2C4_DMA_TX	I2C4	S_DMA_123	I2C4 DMA write request to inform the sDMA to write new data in the I2C4.I2C_DATA[7:0] register
I2C4_DMA_RX	I2C4	S_DMA_124	I2C4 DMA read request to inform the sDMA to read the data in the I2C4.I2C_DATA[7:0] register

**NOTE:** The HS I<sup>2</sup>C controller I2C5 does not generate any DMA requests.

**23.1.4.7 HS I<sup>2</sup>C Programmable Multislave Channel Feature (I<sup>2</sup>C Mode Only)**

This feature allows each multimaster HS I<sup>2</sup>C controller to be addressed using four separate Own Addresses configured in the I2Ci.I2C\_OA and I2Ci.I2C\_OAx registers (where x = 1, 2, 3). An additional register (I2Ci.I2C\_ACTOA) is used to indicate to the LH which address is used by the external master to communicate with the I<sup>2</sup>C controller.

Each Own Address can be independently configured in 7-bit or 10-bit mode by setting the corresponding bit (I2Ci.I2C\_CON[7] XOA, I2Ci.I2C\_CON[6] XOA1, I2Ci.I2C\_CON[5] XOA2, or I2Ci.I2C\_CON[4] XOA3).

**23.1.4.8 HS I<sup>2</sup>C FIFO Management**

Each multimaster HS I<sup>2</sup>C controller implements two internal 8-bit FIFOs, the RX and TX FIFOs.

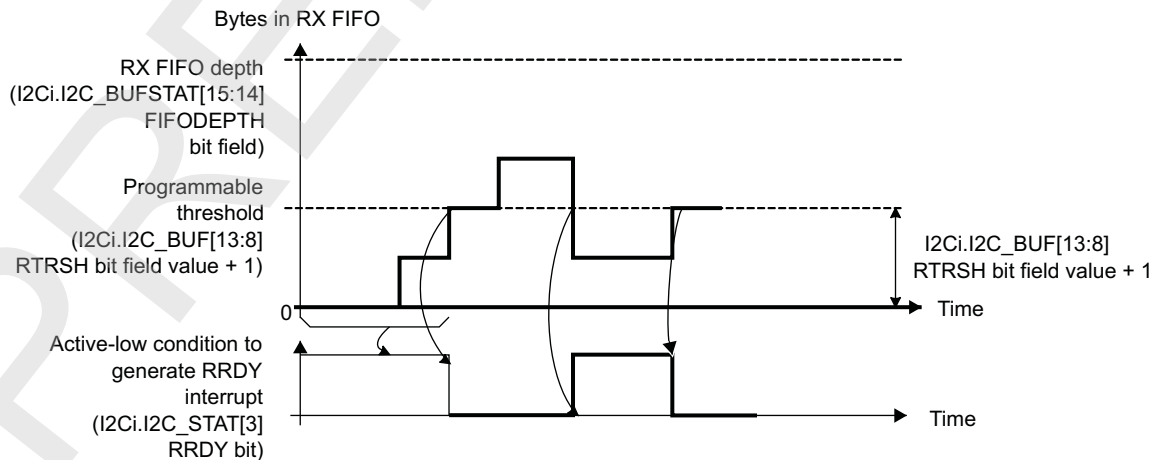
The depth of the RX and TX FIFOs can be checked by reading the I2Ci.I2C\_BUFSTAT[15:14] FIFODEPTH bit field (0x0: 8 bytes, 0x1: 16 bytes, 0x2: 32 bytes, and 0x3: 64 bytes).

**23.1.4.8.1 HS I<sup>2</sup>C FIFO Interrupt Mode**

In FIFO interrupt mode (relevant interrupts enabled by the I2Ci.I2C\_IE register), an interrupt signal informs the processor of the receiver and transmitter status. These interrupts are raised when the RX/TX FIFO thresholds (defined by the I2Ci.I2C\_BUF[13:8] RTRSH bit field value + 1 for the RX FIFO or the I2Ci.I2C\_BUF[5:0] TXTRSH bit field value + 1 for the TX FIFO) are reached; the interrupt signals instruct the LH to transfer data to the destination (from the I<sup>2</sup>C controller in receive mode and/or from any source to the I<sup>2</sup>C controller FIFO in transmit mode).

Figure 23-23 and Figure 23-24 show receive and transmit operations, respectively, from a FIFO management point of view.

**Figure 23-23. HS I<sup>2</sup>C Receive FIFO Interrupt Request Generation**

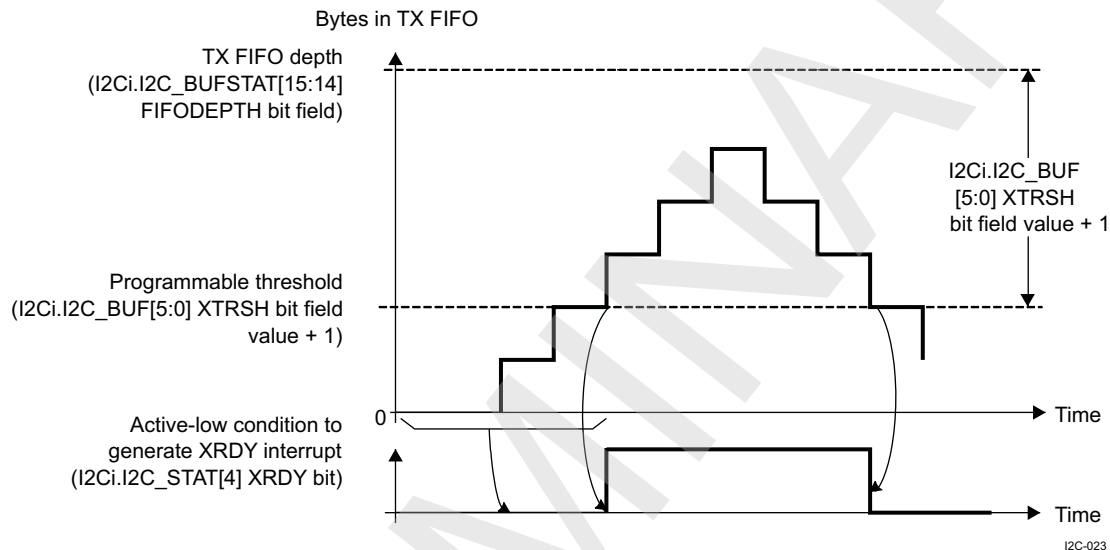


I2C-022

In [Figure 23-23](#), the RRDY interrupt condition shows that the condition for generating an RRDY interrupt is achieved. The interrupt request is generated when this signal is active, and it can be cleared only by the LH by writing 1 in the I2Ci.I2C\_STAT[3] RRDY bit. If the condition is still present after clearing the previous interrupt, another interrupt request is generated.

In receive mode, an RRDY interrupt is generated as soon as the FIFO reaches its receive threshold (I2Ci.I2C\_BUF[13:8] RTRSH bit field value + 1). The interrupt can be deasserted only when the LH has handled enough bytes to make the number of bytes in the RX FIFO lower than the programmed threshold. For each interrupt, the LH can be configured to read a number of bytes equal to the value of the RX FIFO threshold.

**Figure 23-24. HS I<sup>2</sup>C Transmit FIFO Interrupt Request Generation**



In [Figure 23-24](#), the XRDY interrupt condition shows that the condition for generating an XRDY interrupt is achieved. The interrupt request is generated when TX FIFO is empty or when the TX FIFO threshold is not reached, and the LH can clear the XRDY status bit by setting the I2Ci.I2C\_STAT[4] XRDY bit to 1 after transmitting the configured number of bytes. If the condition is still present after clearing the previous interrupt, another interrupt request is generated.

In interrupt mode, the module offers two options for the LH application to handle the interrupts:

- When detecting an interrupt request (XRDY or RRDY type), the LH can write/read 1 data byte to/from the TX/RX FIFO and then clear the interrupt. The module reasserts the interrupt until the interrupt condition is not met.
- When detecting an interrupt request (XRDY or RRDY type), the LH can be programmed to write/read the amount of data bytes specified by the corresponding FIFO threshold (I2C\_BUF[5:0] TXTRSH + 1 or I2C\_BUF[5:0] RXTRSH + 1). In this case, the interrupt condition is cleared and the next interrupt is asserted again when the XRDY or RRDY condition is met again.

If the second-interrupt-serving approach is used, an additional mechanism (draining feature) is implemented for cases where the transfer length is not a multiple of the FIFO threshold value (see [Section 23.1.4.8.4, Draining Feature \[I<sup>2</sup>C Mode Only\]](#)).

**NOTE:** In slave transmit mode (the I2Ci.I2C\_CON[10] MST bit is cleared and the I2Ci.I2C\_CON[9] TRX bit is set to 1), the draining feature must not be used, because the transfer length is not known at configuration time, and the external master can end the transfer at any point by not acknowledging 1 data byte. If the draining feature is used in slave transmit mode, data can remain in the TX FIFO without being transmitted over the I<sup>2</sup>C bus. In this case, the TX FIFO must be cleared by setting the I2Ci.I2C\_BUF[6] TXFIFO\_CLR bit.



**NOTE:** In SCCB mode, the RX and TX threshold values must be set to 1 by setting the I2Ci.I2C\_BUF[13:8]RTRSH and I2Ci.I2C\_BUF[5:0] TXTRSH fields to 0x0.

### 23.1.4.8.2 HS I<sup>2</sup>C FIFO Polling Mode

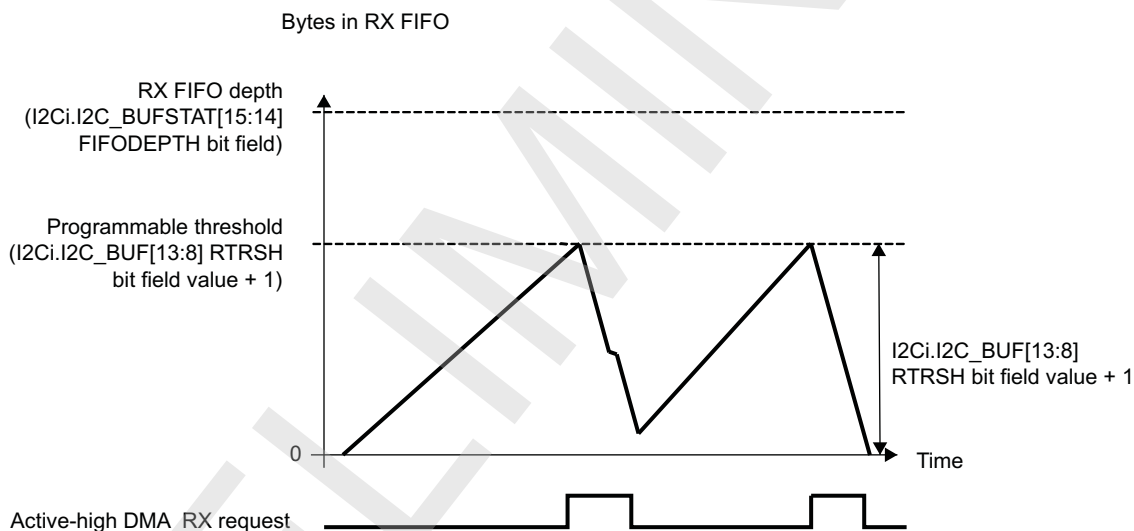
In FIFO polling mode (the I2Ci.I2C\_IE[4] XRDY\_IE and I2Ci.I2C\_IE[3] RRDY\_IE bits are disabled, and the I2Ci.I2C\_BUF[15] RDMA\_EN and I2Ci.I2C\_BUF[7] XDMA\_EN bits are disabled), the status of the module (receiver or transmitter) can be checked by polling the I2Ci.I2C\_STAT[4] XRDY and the I2Ci.I2C\_STAT[3] RRDY bits (the I2Ci.I2C\_STAT[13] RDR and I2Ci.I2C\_STAT[14] XDR bits can also be polled if the draining feature is enabled). The I2Ci.I2C\_STAT[4] XRDY and I2Ci.I2C\_STAT[3] RRDY bits accurately reflect the interrupt conditions described in the discussion of FIFO interrupt mode.

### 23.1.4.8.3 HS I<sup>2</sup>C FIFO DMA Mode (I<sup>2</sup>C Mode Only)

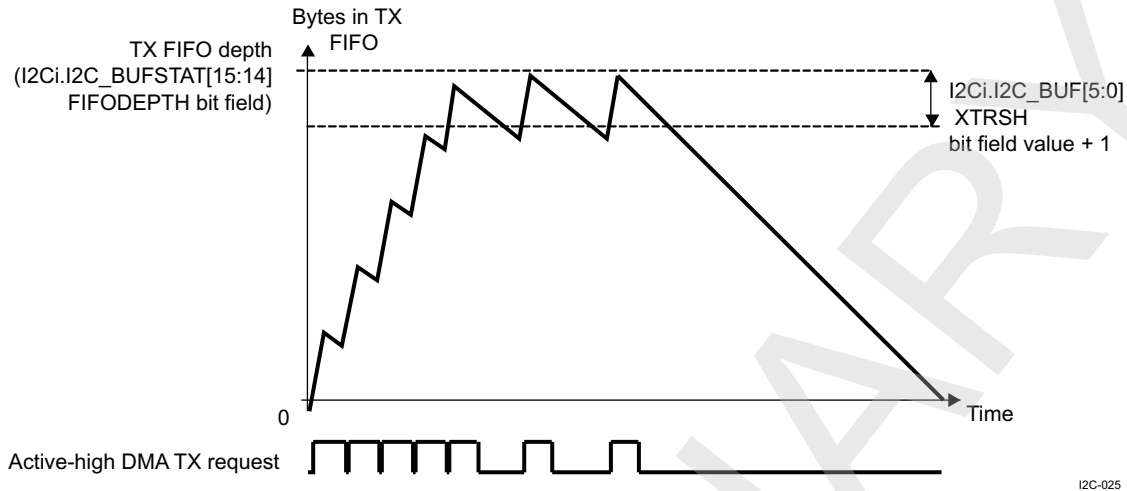
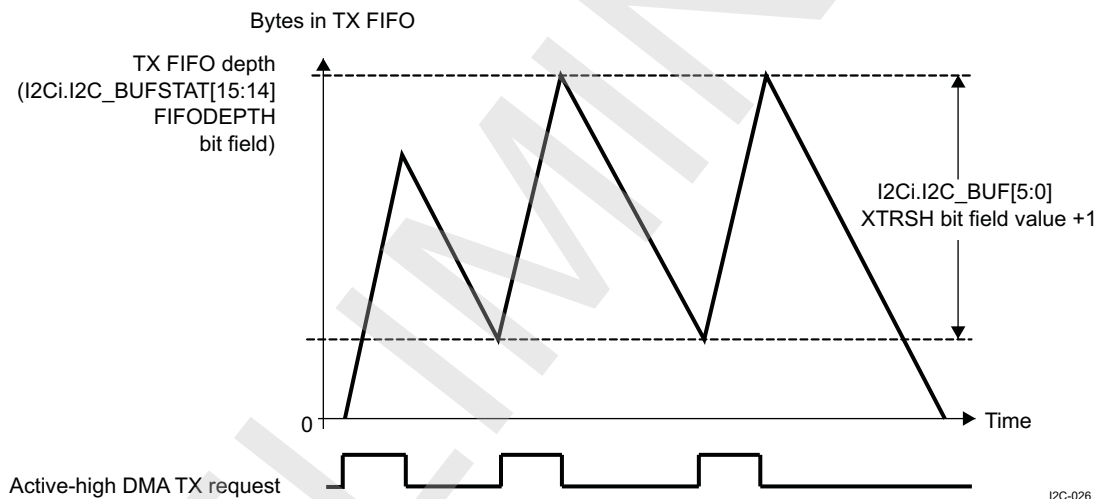
In receive mode, a DMA request is generated by the I2Ci\_DMA\_RX signal as soon as the RX FIFO exceeds its threshold level (the I2Ci.I2C\_BUF[13:8] RTRSH bit field value + 1). This request is deasserted when the number of bytes defined by the threshold level is read by the DMA controller.

Figure 23-25 shows the DMA request generation in receive mode.

**Figure 23-25. HS I<sup>2</sup>C Receive FIFO DMA Request Generation**



In transmit mode, a DMA request is automatically asserted by the I2Ci\_DMA\_TX signal when the TX FIFO is empty. This request is deasserted when the number of bytes (the I2Ci.I2C\_BUF[5:0] TXTRSH bit field value + 1) is written in the FIFO by the DMA controller. If an insufficient number of bytes is written, the sDMA request remains active. Figure 23-26 and Figure 23-27 show the DMA TX transfers with different values for the I2Ci.I2C\_BUF[5:0] TXTRSH bit field.

**Figure 23-26. HS I<sup>2</sup>C Transmit FIFO Request Generation (High Threshold)****Figure 23-27. HS I<sup>2</sup>C Transmit FIFO Request Generation (Low Threshold)**

**NOTE:** In SCCB mode, the RX and TX threshold values must be set to 1 by setting the I2Ci.I2C\_BUF[13:8]RTRSH and I2Ci.I2C\_BUF[5:0]XTRSH bit fields to 0x0.

#### 23.1.4.8.4 HS I<sup>2</sup>C Draining Feature (I<sup>2</sup>C Mode Only)

The draining feature is implemented to handle the end of a transfer whose length is not a multiple of the FIFO threshold values (the I2Ci.I2C\_BUF[13:8] RTRSH bit field value + 1 for the RX threshold and I2Ci.I2C\_BUF[5:0] TXTRSH field value + 1 for the TX threshold). It can also transfer the remaining number of bytes (because the threshold is not reached).

This feature prevents the LH or the DMA controller from trying more FIFO accesses than necessary (for example, to generate at the end of a transfer a DMA RX request having fewer bytes in the FIFO than the configured DMA transfer length). Otherwise, an AERR interrupt is generated by the I2Ci.I2C\_STAT[7] AERR bit.

The draining mechanism generates an interrupt using the I2Ci.I2C\_STAT[13] RDR or I2Ci.I2C\_STAT[14] XDR bit at the end of the transfer, informing the LH that it must check the amount of data left to be transferred (the I2Ci.I2C\_BUFSTAT[13:8] RXSTAT or I2Ci.I2C\_BUFSTAT[5:0] TXSTAT bit fields) and enable the draining feature of the DMA controller by reconfiguring the DMA transfer length according to this value (when the DMA mode is enabled) or perform only the required number of data accesses (when the DMA mode is disabled).

In receive mode (master or slave), if the RX FIFO threshold (the I2Ci.I2C\_BUF[13:8] RTRSH bit field value + 1) is not reached, but the transfer ends on the I<sup>2</sup>C bus and data remains in the RX FIFO (less than the threshold), the receive draining interrupt (the I2Ci.I2C\_STAT[13] RDR bit) is asserted to inform the LH that it can read the amount of data in the RX FIFO (the I2Ci.I2C\_BUFSTAT[13:8] RXSTAT bit field). The LH performs a number of data read accesses equal to the I2Ci.I2C\_BUFSTAT[13:8] RXSTAT bit field (interrupt or polling mode), or reconfigures the sDMA controller with the required value to drain the FIFO.

In master transmit mode, if the TX FIFO threshold (I2Ci.I2C\_BUF[5:0] TXTRSH bit field value + 1) is not reached, but the amount of data remaining to be written in the TX FIFO is less than the threshold, the transmit draining interrupt (the I2Ci.I2C\_STAT[14] XDR bit) is asserted to inform the LH that it can read the amount of data remaining to be written in the TX FIFO (the I2Ci.I2C\_BUFSTAT[5:0] TXSTAT bit field). The LH must write the required number of data bytes specified by the I2Ci.I2C\_BUFSTAT[5:0] TXSTAT bit field value or reconfigure the sDMA controller with the value required to transfer the last bytes to the FIFO.

In master mode, the LH can alternately not check the values of the I2Ci.I2C\_BUFSTAT[5:0] TXSTAT and I2Ci.I2C\_BUFSTAT[13:8] RXSTAT bit fields, the LH can obtain this information internally (by computing the I2Ci.I2C\_CNT[15:0] DATACOUNT bit field value modulo I2Ci.I2C\_BUF[13:8] RTRSH or I2Ci.I2C\_BUF[5:0] TXTRSH).

By default, the draining feature is disabled; it can be enabled using the I2Ci.I2C\_IE[14] XDR\_IE or I2Ci.I2C\_IE[13] RDR\_IE bits (default disabled) only for transfers with lengths not equal to the threshold values (I2Ci.I2C\_BUF[5:0] TXTRSH bit field value + 1 for the TX threshold or I2Ci.I2C\_BUF[13:8] RTRSH bit field value + 1 for the RX threshold).

#### 23.1.4.9 HS I<sup>2</sup>C Noise Filter

The noise filter is used to suppress any noise that is 50 ns or less in case of F/S operation modes and SCCB modes, and any noise that is 10 ns or less in case of HS mode operation. The noise filter is always one period of the I2Ci\_INTERNAL\_CLK clock. This way, for HS mode operation (prescaler bypassed), the filter suppresses spikes of less than 10.4 ns.

For SCCB modes (for example, I2Ci.I2C\_PSC[7:0] PSC bit field = 4), the maximum width of suppressed spikes is 52 ns.

To ensure correct filtering, the prescaler must be programmed accordingly by the 2Ci.I2C\_PSC[7:0] PSC bit field.

#### 23.1.4.10 HS I<sup>2</sup>C Write and Read Operations in SCCB Mode

In SCCB mode, the multimaster HS I<sup>2</sup>C controller can write or read a single byte to or from the external SCCB device.

To write a single byte to the external SCCB device, the multimaster HS I<sup>2</sup>C controller must be configured in multimaster transmitter mode by setting the I2Ci.I2C\_CON[10] MST and I2Ci.I2C\_CON[9] TRX bits to 1. The external device slave address (7-bit address of the ID value) is set in the I2Ci.I2C\_SA register; the register address (8-bit subaddress in the external SCCB device) is set in the I2Ci.I2C\_OA0 register. The 8-bit data to be transmitted is written by the LH in the I2Ci.I2C\_DATA register.

To read a single byte from the external SCCB device, the multimaster HS I<sup>2</sup>C controller must be configured in multimaster receiver mode by setting the I2Ci.I2C\_CON[10] MST bit to 1 and by clearing the I2Ci.I2C\_CON[9] TRX bit to 0. The external device slave address (7-bit address of the ID value) is set in the I2Ci.I2C\_SA register; the register address (8-bit subaddress in the external SCCB device) is set in the I2Ci.I2C\_OA0 register. The 8-bit data received from the external SCCB device is read by the LH from the I2Ci.I2C\_DATA register.

**NOTE:** In SCCB mode, the RX and TX thresholds must be set to 1 by configuring the I2Ci.I2C\_BUF[13:8] RTRSH and I2Ci.I2C\_BUF[5:0]XTRSH fields to 0x0.

### 23.1.4.11 HS I<sup>2</sup>C System Test Mode

A system test mode is available for multimaster HS I<sup>2</sup>C controller module testing. This mode is enabled by setting the I2Ci.I2C\_SYSTEST[15] ST\_EN bit to 1. When this bit is cleared to 0, the I<sup>2</sup>C controller is configured in normal operation mode.

In system test mode, the I2Ci\_SYSTEST[13:12] TMODE bit field selects the type of test. Table 23-14 lists the tests available for the multimaster HS I<sup>2</sup>C controllers.

**Table 23-14. HS I<sup>2</sup>C List of Tests**

I2Ci.I2C_SYSTEST[13:12] TMODE	Test	Description
00	Functional mode	Normal operation mode
01	Reserved (not used)	
10	Test of i2ci_scl serial clock line	The i2ci_scl line is driven with a permanent clock as if mastered with the parameters set in the I2Ci.I2C_PSC, I2Ci.I2C_SCLL, and I2Ci.I2C_SCLH registers.
11	Loop-back mode + i2ci_scl/ i2ci_sda input/output	In master transmit mode only, data transmitted out of the I2Ci.I2C_DATA register (write action) is received in the same I2Ci.I2C_DATA register through an internal path through the FIFO buffers. The DMA and interrupt requests are normally generated if they are enabled. Moreover, the i2ci_scl, i2ci_sda, and i2ci_sccbe lines are controlled with the I2Ci.I2C_SYSTEST[4:0] bits..

**NOTE:** When the I2Ci.I2C\_SYSTEST[13:12] TMODE bit field is set to 11, the I<sup>2</sup>C controller must be configured in I<sup>2</sup>C F/S (the I2Ci.I2C\_CON[13:12] OPMODE set to 00) or I<sup>2</sup>C HS mode (the I2Ci.I2C\_CON[13:12] OPMODE set to 01). The loop-back mode is not available in SCCB mode (the I2Ci.I2C\_CON[13:12] OPMODE = b10).

**NOTE:** In normal operation mode (I2Ci.I2C\_SYSTEST[15] ST\_EN clear to 0), the I2Ci.I2C\_SYSTEST[4:0] bits that control the i2ci\_scl, i2ci\_sda, and i2ci\_sccbe lines in system test mode are read-only bits.

In system test mode (I2Ci.I2C\_SYSTEST[15] ST\_EN set to 1), the I2Ci.I2C\_STAT[5:0] status bits can be set to 1 when the I2Ci.I2C\_SYSTEST[11] SSB bit is set to 1. Clearing the I2Ci.I2C\_SYSTEST[11] SSB bit to 0 does not clear the I2Ci.I2C\_STAT[5:0] bits to 0. The I2Ci.I2C\_STAT[5:0] bit field can be cleared to 0 only by writing 1 in the corresponding bits.

## 23.1.5 HS I<sup>2</sup>C Programming Guide

### 23.1.5.1 HS I<sup>2</sup>C Low-Level Programming Models

#### 23.1.5.1.1 HS I<sup>2</sup>C Programming Model in I<sup>2</sup>C Mode

This section describes the programming model of the multimaster HS I<sup>2</sup>C controllers configured in I<sup>2</sup>C mode.

##### 23.1.5.1.1.1 Main Program

###### 23.1.5.1.1.1.1 Configure the Module Before Enabling the I<sup>2</sup>C Controller

Before enabling the I<sup>2</sup>C controller, perform the following:

1. Enable the functional and interface clocks.
2. Program the prescaler to obtain an approximately 12-MHz internal sampling clock by programming the corresponding value in the I2Ci.I2C\_PSC[7:0] PSC field. This value depends on the frequency of the functional clock (I2Ci\_FCLK).
3. Program the I2Ci.I2C\_SCLL[7:0] SCLL and I2Ci.I2C\_SCLH[7:0] SCLH bit fields to obtain a bit rate of 100 Kbps, 400 Kbps or 1Mbps. These values depend on the internal sampling clock frequency (see [Table 23-8](#)).
4. (Optional) Program the I2Ci.I2C\_SCLL[15:8] HSSCLL and I2Ci.I2C\_SCLH[15:8] HSSCLH bit fields to obtain a bit rate of 400 Kbps or 3.4 Mbps (for the second phase of HS mode). These values depend on the internal sampling clock frequency (see [Table 23-8](#)).
5. (Optional) If a bit rate of 3.4 Mbps is used and the bus line capacitance exceeds 45 pF, see [Section 18.4.8, PAD Functional Multiplexing and Configuration](#).
6. Configure the Own Address of the I2C controller by storing it in the I2Ci.I2C\_OA register. Up to four Own Addresses can be programmed in the I2Ci.I2C\_OAx registers (where x = 0, 1, 2, 3) for each I<sup>2</sup>C controller.

---

**NOTE:** For a 10-bit address, set the corresponding expand Own Address bit in the I2Ci.I2C\_CON register.

---

7. Set the TX threshold (in transmitter mode) and the RX threshold (in receiver mode) by setting the I2Ci.I2C\_BUF[5:0] TXTRSH bit field to (TX threshold to 1) and the I2Ci.I2C\_BUF[13:8] RTRSH bit field to (RX threshold to 1), where the TX and RX thresholds are greater than or equal to 1.
8. Take the I<sup>2</sup>C controller out of reset by setting the I2Ci.I2C\_CON[15] I2C\_EN bit to 1.

###### 23.1.5.1.1.1.2 Initialize the I<sup>2</sup>C Controller

To initialize the I<sup>2</sup>C controller, perform the following steps:

1. Configure the I2Ci.I2C\_CON register:
  - For master or slave mode, set the I2Ci.I2C\_CON[10] MST bit (0: slave; 1: master).
  - For transmitter or receiver mode, set the I2Ci.I2C\_CON[9] TRX bit (0: receiver; 1: transmitter).
2. If using an interrupt to transmit and receive data, set the corresponding bit in the I2Ci.I2C\_IE register to 1 (the I2Ci.I2C\_IE[4] XRDY\_IE bit for the transmit interrupt, the I2Ci.I2C\_IE[3] RRDY bit for the receive interrupt).
3. If using DMA to receive and transmit data, set the corresponding bit in the I2Ci.I2C\_BUF register to 1 (the I2Ci.I2C\_BUF[15] RDMA\_EN bit for the receive DMA channel, the I2Ci.I2C\_BUF[7] XDMA\_EN bit for the transmit DMA channel).

### 23.1.5.1.1.1.3 Configure Slave Address and the Data Control Register

In master mode, configure the slave address register by programming the I2Ci.I2C\_SA[9:0] SA bit field and the number of data bytes (I<sup>2</sup>C data payload) associated with the transfer by programming the I2Ci.I2C\_CNT[15:0] DCOUNT bit field.

---

**NOTE:** For a 10-bit address, set the I2Ci.I2C\_CON[8] XSA bit to 1.

---

### 23.1.5.1.1.1.4 Initiate a Transfer

Poll the I2Ci.I2C\_STAT[12] BB bit. If it is cleared to 0 (bus not busy), configure the I2Ci.I2C\_CON[0] STT and I2Ci.I2C\_CON[1] STP bits. To initiate a transfer, the I2Ci.I2C\_CON[0] STT bit must be set to 1, and it is not mandatory to set the I2Ci.I2C\_CON[1] STP bit to 1.

### 23.1.5.1.1.1.5 Receive Data

Poll the I2Ci.I2C\_STAT[3] RRDY bit, or use the RRDY interrupt (the I2Ci.I2C\_IE[3] RRDY\_IE bit must be set to 1) or the DMA RX channel (the I2Ci.I2C\_BUF[15] RDMA\_EN bit must be set to 1) to read the receive data in the I2Ci.I2C\_DATA register.

If the transfer length does not equal the RX FIFO threshold (I2Ci.I2C\_BUF[13:8] RTRSH field + 1), use the draining feature (enable the RDR interrupt by setting the I2Ci.I2C\_IE[13] RDR\_IE bit to 1).

---

**NOTE:** In receive mode only, the I2Ci.I2C\_STAT[11] ROVR (receive overrun) bit indicates whether the receiver has experienced overrun. An overrun condition occurs when the shift register and the RX FIFO are full. An overrun condition does not result in data loss; the I<sup>2</sup>C controller simply holds i2ci\_scl to low to prevent other bytes from being received.

The I2Ci.I2C\_STAT[7] AERR bit is set to 1 when a read access is performed in the I2Ci.I2C\_DATA register while the RX FIFO is empty. The corresponding interrupt can be enabled by setting the I2Ci.I2C\_IE[7] AERR\_IE bit to 1.

---

### 23.1.5.1.1.1.6 Transmit Data

Poll the I2Ci.I2C\_STAT[4] XRDY bit, or use the XRDY interrupt (the I2Ci.I2C\_IE[4] XRDY\_IE bit must be set to 1) or the DMA TX channel (the I2Ci.I2C\_BUF[7] XDMA\_EN bit must be set to 1) to write data to the I2Ci.I2C\_DATA register.

If the transfer length does not equal the TX FIFO threshold (I2Ci.I2C\_BUF[5:0] TXTRSH field + 1), use the draining feature (enable the XDR interrupt by setting the I2Ci.I2C\_IE[14] XDR\_IE bit to 1).

---

**NOTE:** In transmit mode only, the I2Ci.I2C\_STAT[10] XUDF bit indicates whether the transmitter has experienced underflow.

In master transmit mode, underflow occurs when the shift register and the TX FIFO are empty and there are still some bytes to transmit (the value of the I2Ci.I2C\_CNT[15:0] DCOUNT bit field is not 0).

In slave transmit mode, underflow occurs when the shift register and the TX FIFO are empty and the external I<sup>2</sup>C master device still requests data bytes to be read.

The I2Ci.I2C\_STAT[7] AERR bit is set to 1 when a write access is performed in the I2Ci.I2C\_DATA register while the TX FIFO is full. The corresponding interrupt can be enabled by setting the I2Ci.I2C\_IE[7] AERR\_IE bit to 1.

---

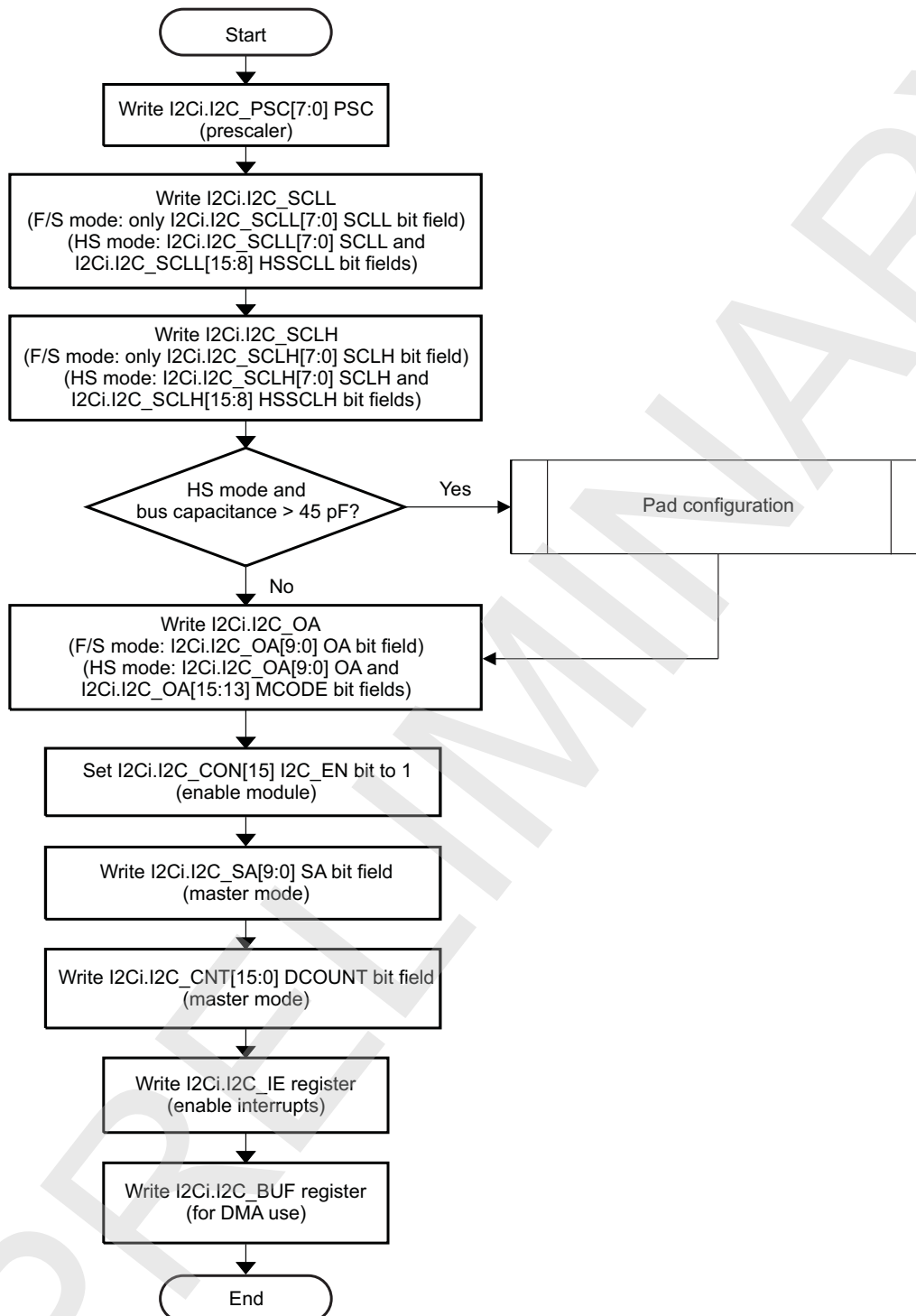
### 23.1.5.1.1.2 Interrupt Subroutine Sequence

1. Test for arbitration lost (the I2Ci.I2C\_STAT[0] AL bit) and resolve accordingly.
2. Test for no acknowledgment (the I2Ci.I2C\_STAT[1] NACK bit) and resolve accordingly.
3. Test for register access ready (the I2Ci.I2C\_STAT[2] ARDY bit) and resolve accordingly.

4. Test for receive data ready (the I2Ci.I2C\_STAT[3] RRDY bit) and resolve accordingly.
5. Test for transmit data ready (the I2Ci.I2C\_STAT[4] XRDY bit) and resolve accordingly.
6. Test for general call (the I2Ci.I2C\_STAT[5] GC bit) and resolve accordingly.
7. Test for start (S) condition (the I2Ci.I2C\_STAT[6] STC bit) and resolve accordingly. For this test, the functional clock must be inactive.
8. Test for access error (the I2Ci.I2C\_STAT[7] AERR bit) and resolve accordingly.
9. Test for bus free (the I2Ci.I2C\_STAT[8] BF bit) and resolve accordingly.

#### **23.1.5.1.1.3 Programming Flow Diagrams**

Figure 23-28 through Figure 23-36 are procedure flow charts for programming the F/S and HS I<sup>2</sup>C modes.

**Figure 23-28. HS I<sup>2</sup>C Setup Procedure**

I2C-027

**Table 23-15. HS I<sup>2</sup>C Subprocess call summary for Sequence - Setup Procedure**

Register Name	Register Name
Pad configuration	Refer to <a href="#">Section 18.4.8 PAD Functional Multiplexing and Configuration</a> .

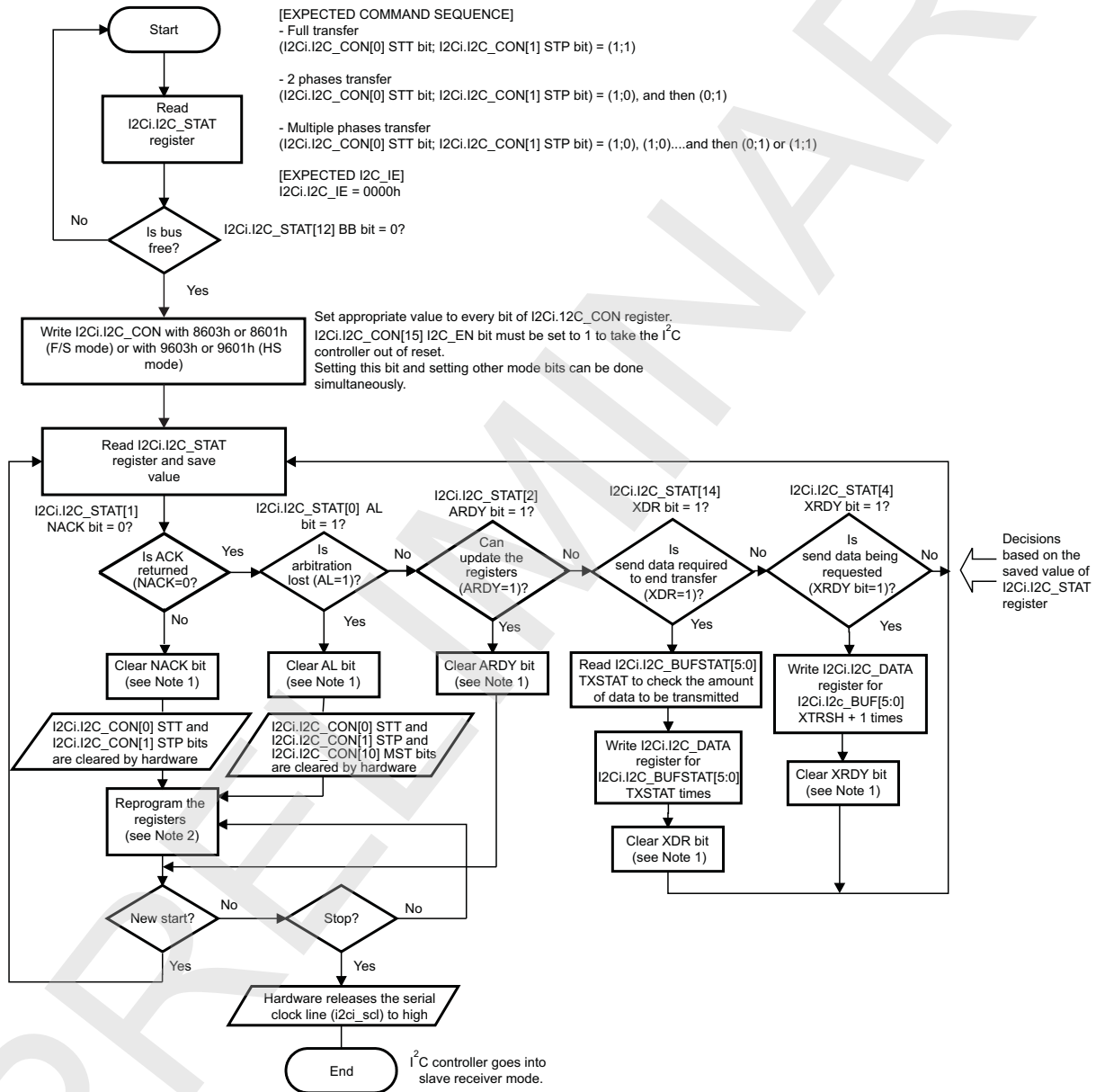


**Table 23-16. HS I<sup>2</sup>C Register call summary for Sequence - Setup Procedure**

Register Name	Register Name	Register Name
I2Ci.I2C_PSC <sup>(1)</sup>	I2Ci.I2C_CON <sup>(1)</sup>	I2Ci.I2C_BUF <sup>(1)</sup>
I2Ci.I2C_SCLL <sup>(1)</sup>	I2Ci.I2C_SA <sup>(1)</sup>	I2Ci.I2C_IE <sup>(1)</sup>
I2Ci.I2C_SCLH <sup>(1)</sup>	I2Ci.I2C_CNT <sup>(1)</sup>	I2Ci.I2C_OA <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

**Figure 23-29. HS I<sup>2</sup>C Master Transmitter Mode, Polling Method, in F/S and HS Modes**



- (1) The NACK, AL, ARDY, XDR, and XRDY bits are cleared by writing 1 to each corresponding bit in the I2Ci.I2C\_STAT register.
- (2) Reprogram the registers means: I2Ci.I2C\_CON[11] STB and/or I2Ci.I2C\_CON[10] MST bit and/or I2Ci.I2C\_SA[9:0] SA register and/or I2Ci.I2C\_CNT[15:0] DCOUNT register and/or I2Ci.I2C\_CON[0] STT bit and/or I2Ci.I2C\_CON[1] STP bit.

I2C-028

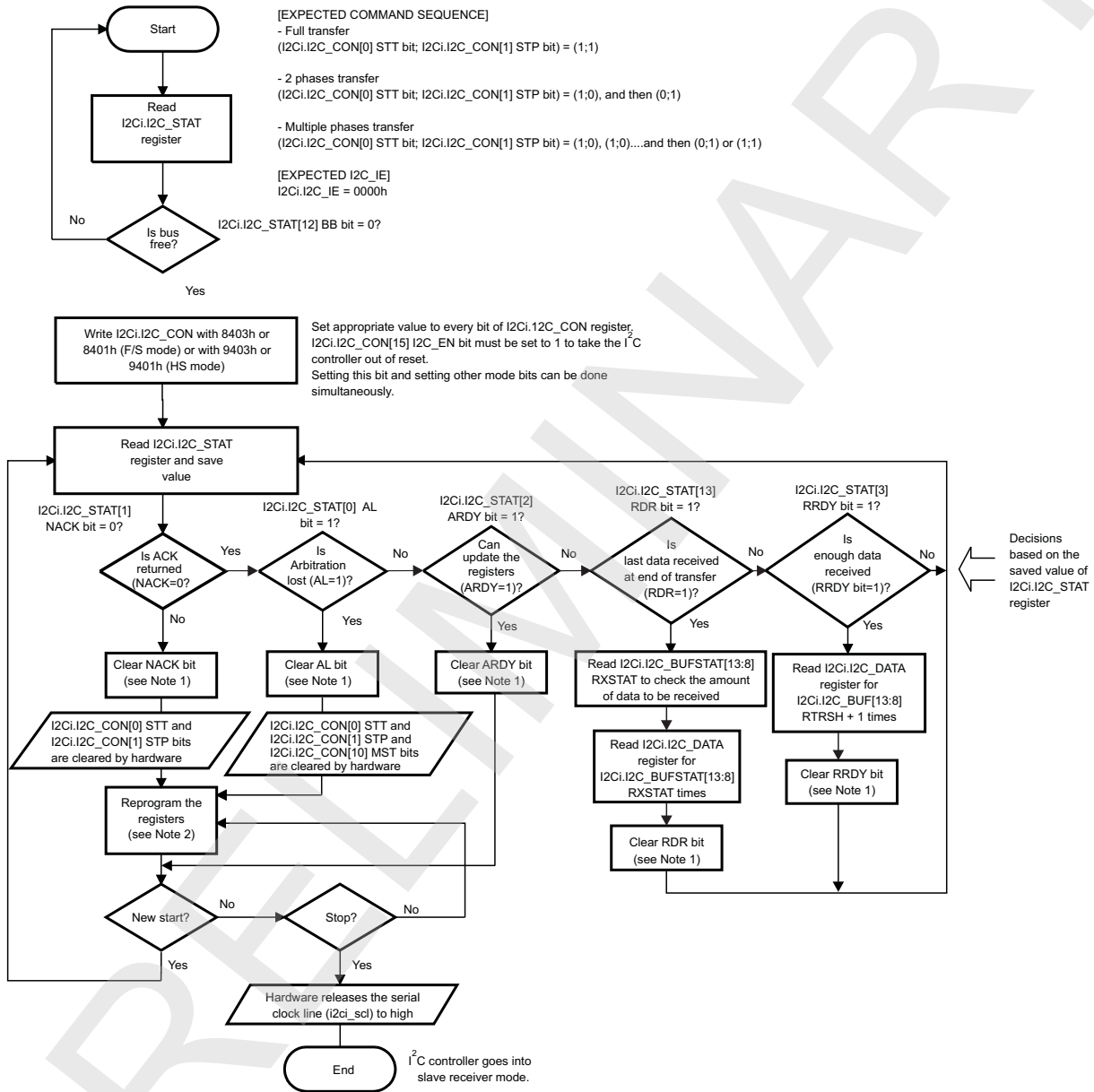
**Table 23-17. HS I<sup>2</sup>C Register call summary for Sequence - Master Transmitter Mode, Polling Method, in F/S and HS Modes**

Register Name	Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>	I2Ci.I2C_CON <sup>(1)</sup>
I2Ci.I2C_BUFSTAT <sup>(1)</sup>	I2Ci.I2C_CON <sup>(1)</sup>	I2Ci.I2C_BUF <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

**NOTE:** In HS mode, the Sr condition and clock frequency switching are automatically generated by the multimaster HS I<sup>2</sup>C controller.

**Figure 23-30. HS I<sup>2</sup>C Master Receiver Mode, Polling Method, in F/S and HS Modes**



- (1) The NACK, AL, ARDY, RDR, and RRDY bits are cleared by writing 1 to each corresponding bit in the I2Ci.I2C\_STAT register.
- (2) Reprogram registers means: I2Ci.I2C\_CON[11] STB and/or I2Ci.I2C\_CON[10] MST bit and/or I2Ci.I2C\_SA[9:0] SA register and/or I2Ci.I2C\_CNT[15:0] DCOUNT register and/or I2Ci.I2C\_CON[0] STT bit and/or I2Ci.I2C\_CON[1] STP bit.

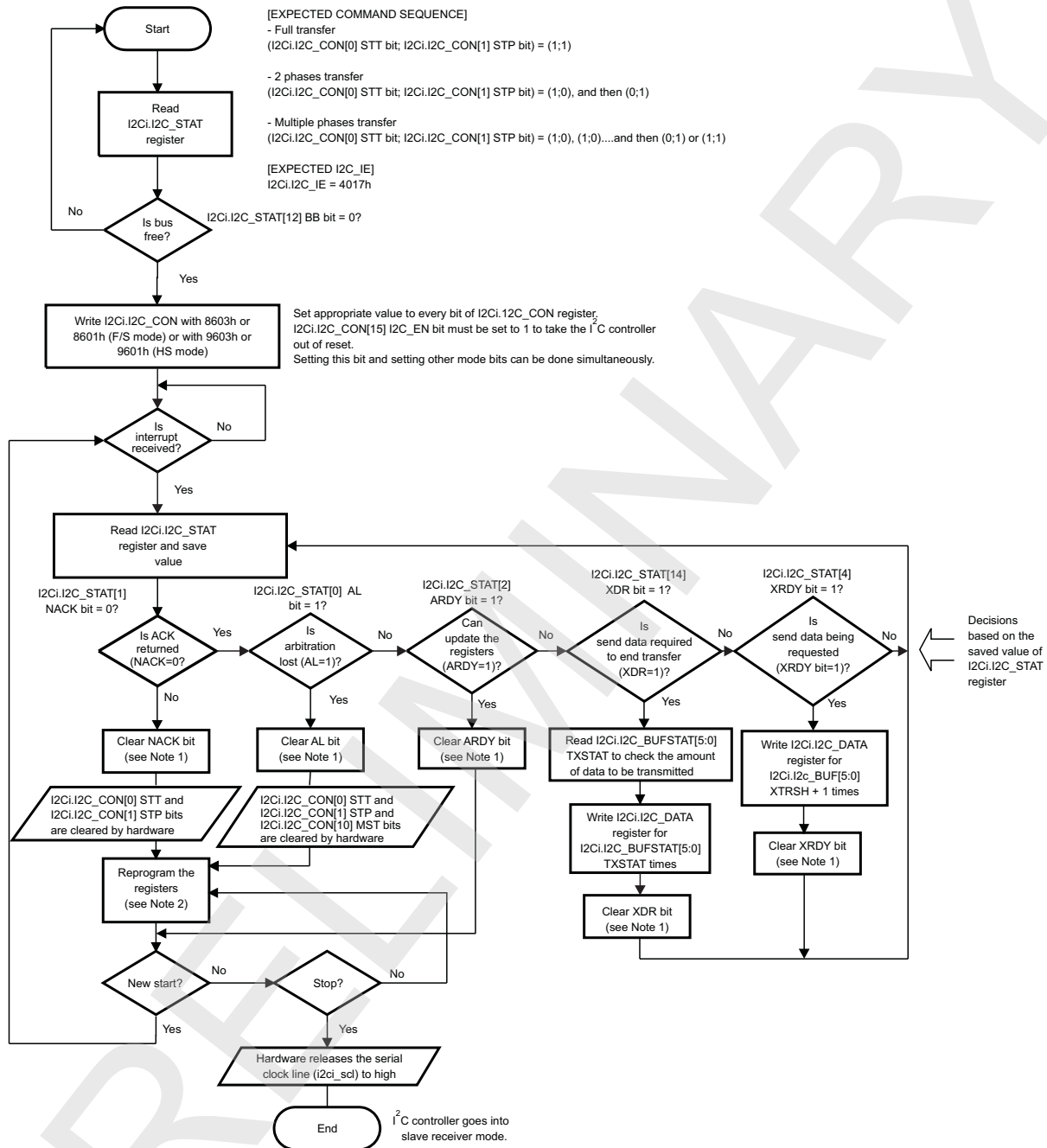
I2C-029

**Table 23-18. HS I<sup>2</sup>C Register call summary for Sequence - Master Receiver Mode, Polling Method, in F/S and HS Modes**

Register Name	Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_BUFSTAT <sup>(1)</sup>	I2Ci.I2C_BUF <sup>(1)</sup>
I2Ci.I2C_CON <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>	

<sup>(1)</sup> i = 1 to 4

Figure 23-31. HS I<sup>2</sup>C Master Transmitter Mode, Interrupt Method, in F/S and HS Modes



I2C-030

- (1) The NACK, AL, ARDY, XDR, and XRDY bits are cleared by writing 1 to each corresponding bit in the I2Ci.I2C\_STAT register.
- (2) Reprogram registers means: I2Ci.I2C\_CON[11] STB and/or I2Ci.I2C\_CON[10] MST bit and/or I2Ci.I2C\_SA[9:0] SA register and/or I2Ci.I2C\_CNT[15:0] DCOUNT register and/or I2Ci.I2C\_CON[0] STT bit and/or I2Ci.I2C\_CON[1] STP bit.

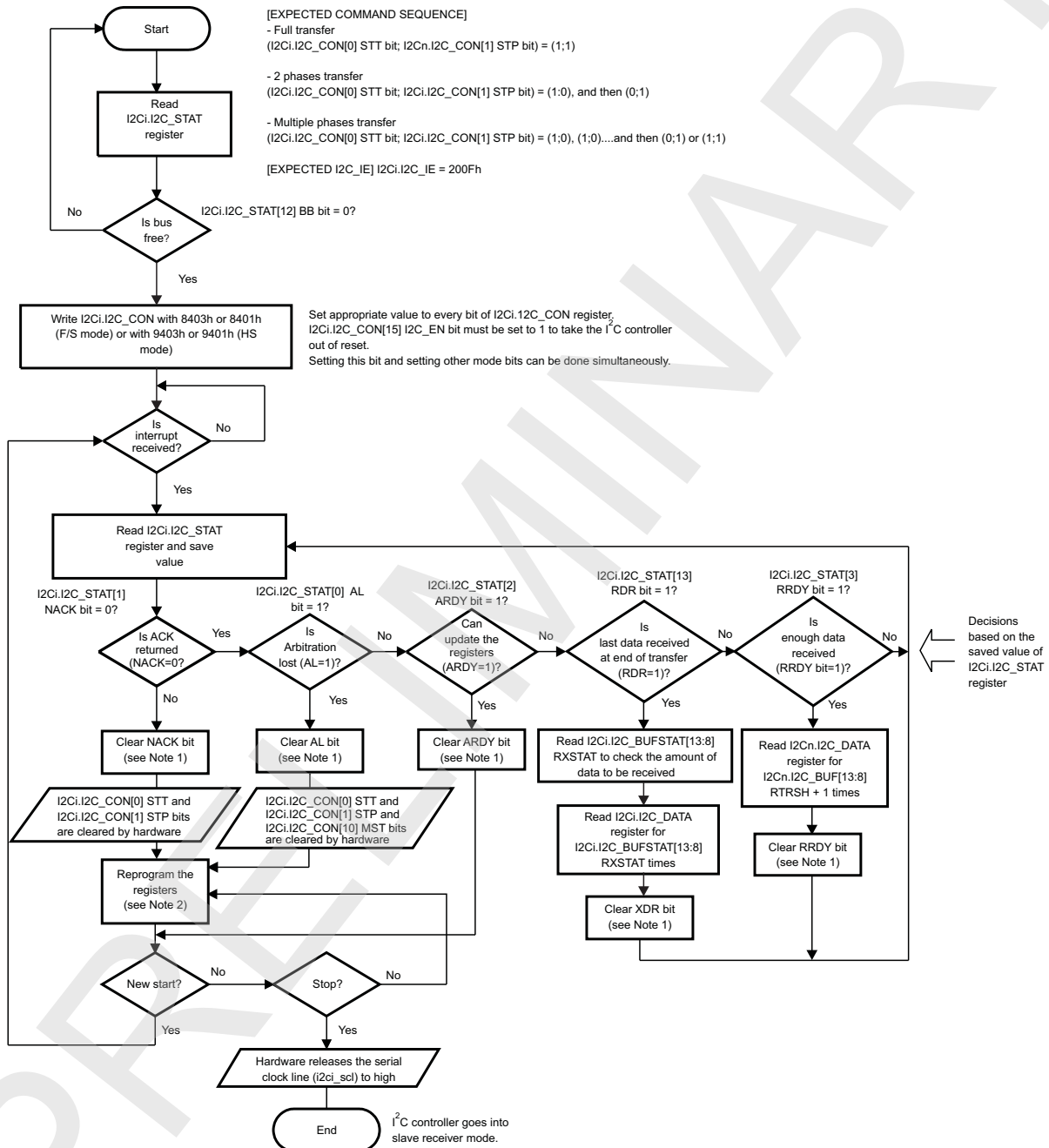
**Table 23-19. HS I<sup>2</sup>C Register call summary for Sequence - Master Transmitter Mode, Interrupt Method, in F/S and HS Modes**

Register Name	Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_BUFSTAT <sup>(1)</sup>	I2Ci.I2C_BUF <sup>(1)</sup>
I2Ci.I2C_CON <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>	

<sup>(1)</sup> i = 1 to 4

**NOTE:** In HS mode, the Sr condition and clock frequency switching are automatically generated by the multimaster HS I<sup>2</sup>C controller.

**Figure 23-32. HS I<sup>2</sup>C Master Receiver Mode, Interrupt Method, in F/S and HS Modes**



- (1) The NACK, AL, ARDY, RDR, and RRDY bits are cleared by writing 1 to each corresponding bit in the I2Ci.I2C\_STAT register.
- (2) Reprogram registers means: I2Ci.I2C\_CON[11] STB and/or I2Ci.I2C\_CON[10] MST bit and/or I2Ci.I2C\_SA[9:0] SA register and/or I2Ci.I2C\_CNT[15:0] DCOUNT register and/or I2Ci.I2C\_CON[0] STT bit and/or I2Ci.I2C\_CON[1] STP bit.

I2C-031

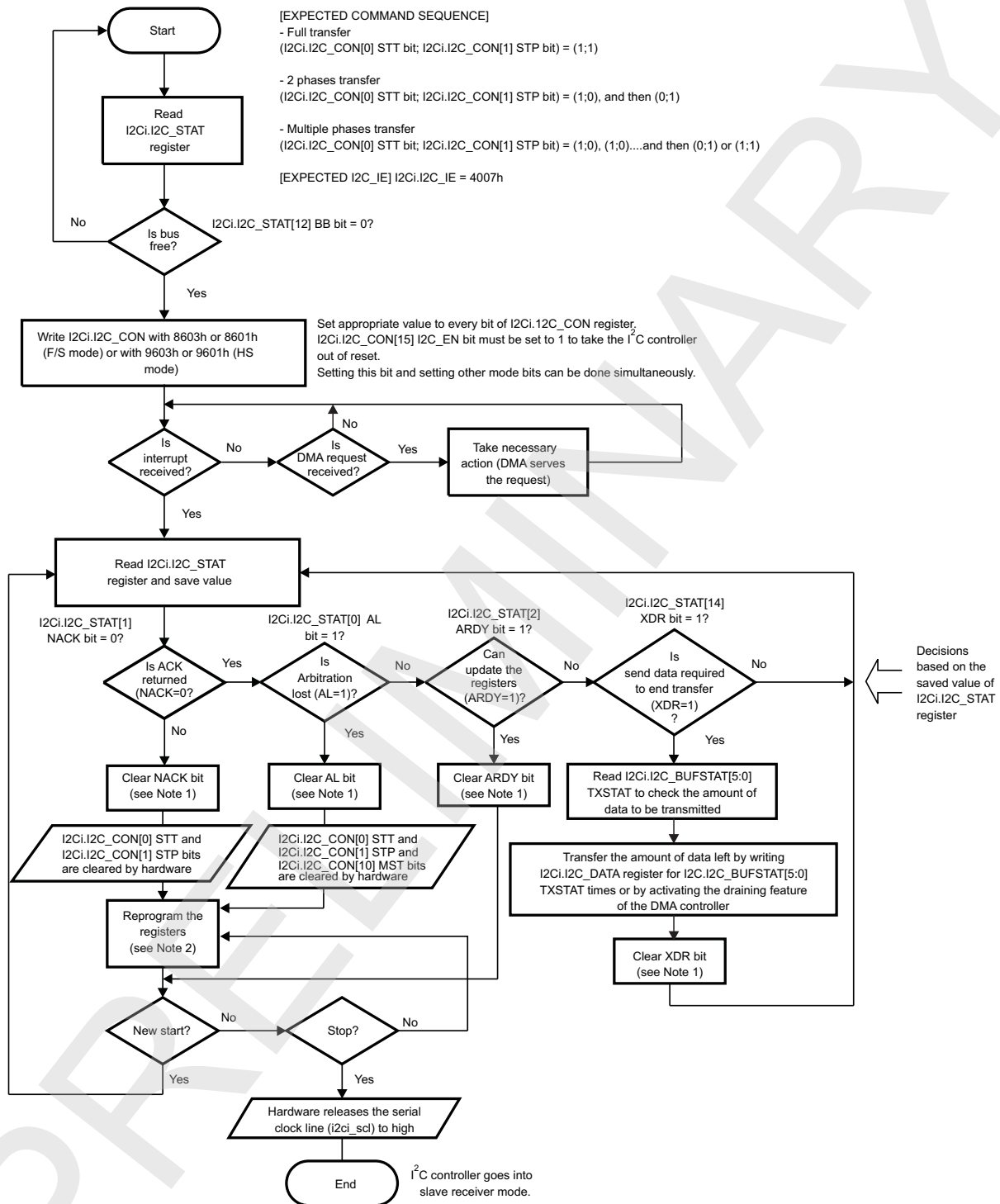
**Table 23-20. HS I<sup>2</sup>C Register call summary for Sequence - Master Receiver Mode, Interrupt Method, in F/S and HS Modes**

Register Name	Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_BUFSTAT <sup>(1)</sup>	I2Ci.I2C_BUF <sup>(1)</sup>
I2Ci.I2C_CON <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>	

<sup>(1)</sup> i = 1 to 4



Figure 23-33. HS I<sup>2</sup>C Master Transmitter Mode, DMA Method in F/S and HS Modes



I2C-032

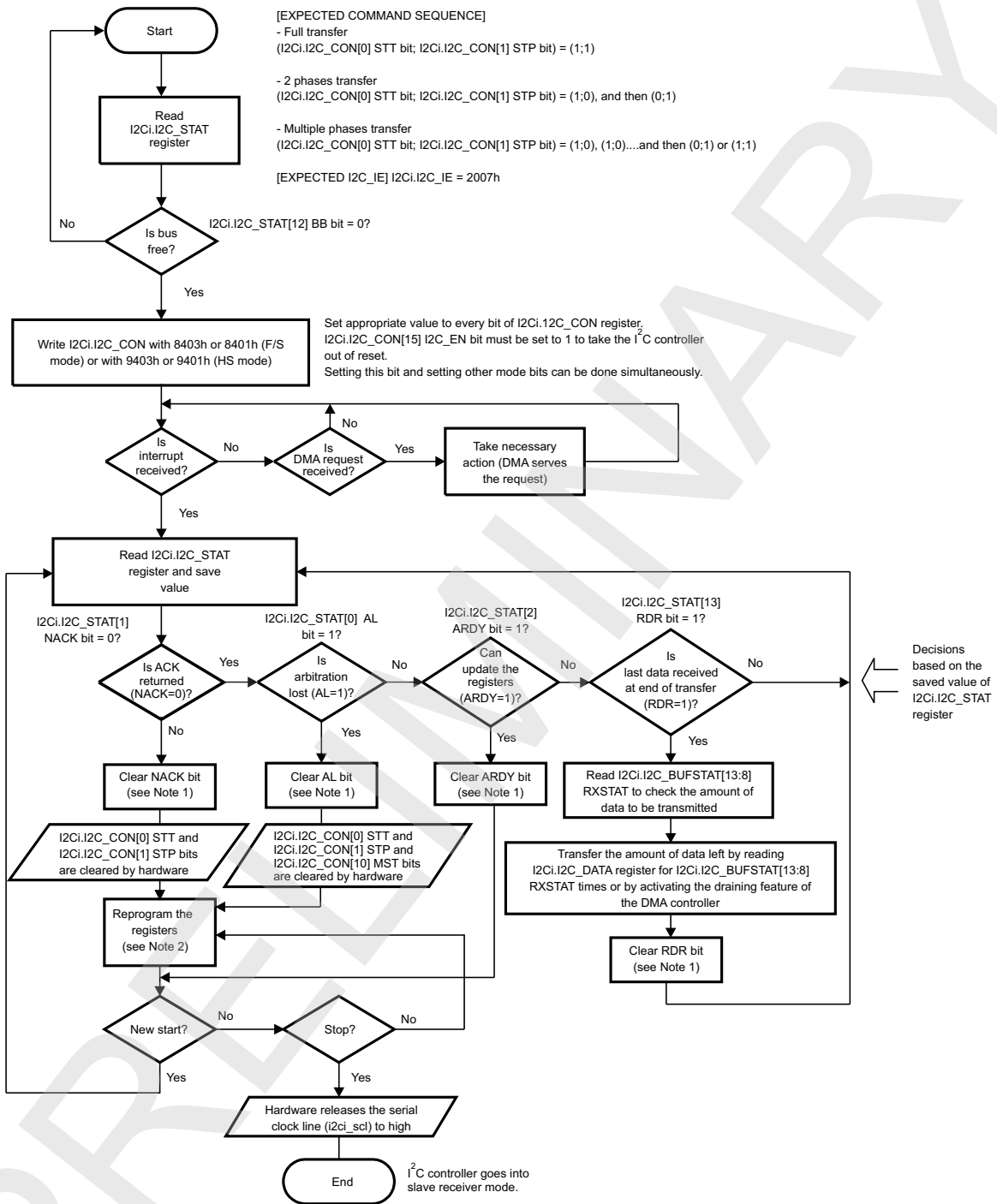
- (1) The NACK, AL, ARDY, and XDR bits are cleared by writing 1 to each corresponding bit in the I2Ci.I2C\_STAT register.
- (2) Reprogram registers means: I2Ci.I2C\_CON[11] STB and/or I2Ci.I2C\_CON[10] MST bit and/or I2Ci.I2C\_SA[9:0] SA register and/or I2Ci.I2C\_CNT[15:0] DCOUNT register and/or I2Ci.I2C\_CON[0] STT bit and/or I2Ci.I2C\_CON[1] STP bit.

**Table 23-21. HS I<sup>2</sup>C Register call summary for Sequence - Master Transmitter Mode, DMA Method in F/S and HS Modes**

Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_BUFSTAT <sup>(1)</sup>
I2Ci.I2C_CON <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

Figure 23-34. HS I<sup>2</sup>C Master Receiver Mode, DMA Method in F/S and HS Modes



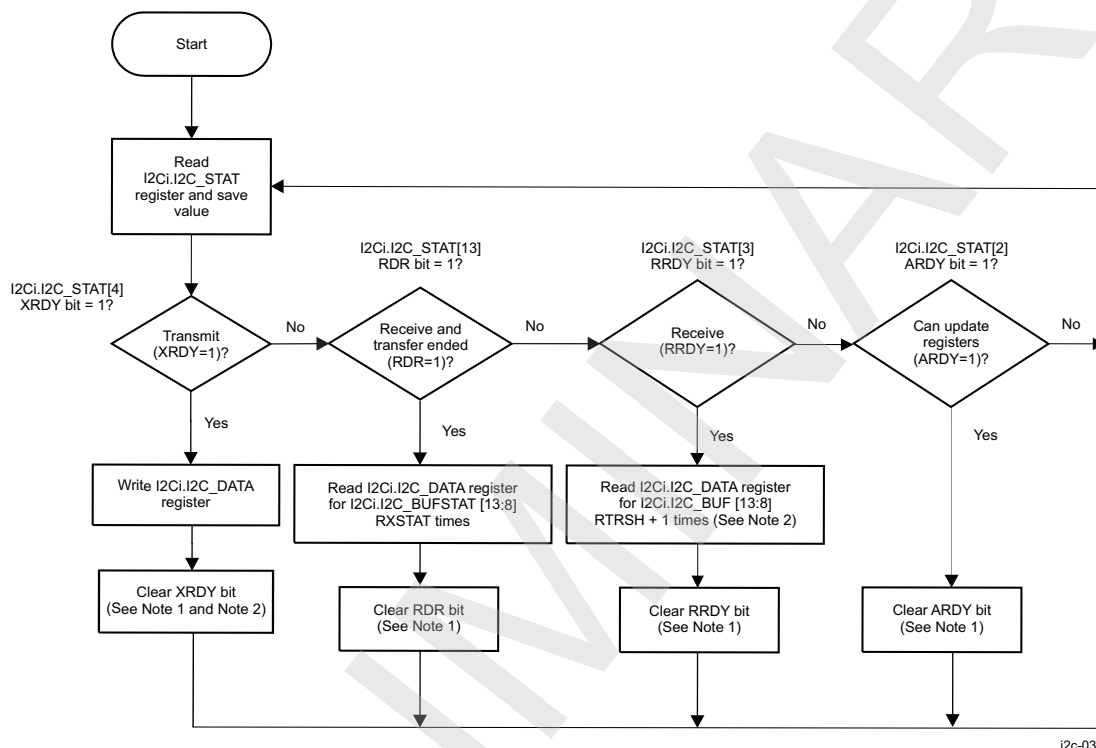
I2C-033

- (1) In HS mode, the Sr condition and clock frequency switching are automatically generated by the multimaster HS I<sup>2</sup>C controller.
- (2) The NACK, AL, ARDY, and RDR bits are cleared by writing 1 to each corresponding bit in the I2Ci.I2C\_STAT register.
- (3) Reprogram registers means: I2Ci.I2C\_CON[11] STB and/or I2Ci.I2C\_CON[10] MST bit and/or I2Ci.I2C\_SA[9:0] SA register and/or I2Ci.I2C\_CNT[15:0] DCOUNT register and/or I2Ci.I2C\_CON[0] STT bit and/or I2Ci.I2C\_CON[1] STP bit.

**Table 23-22. HS I<sup>2</sup>C Register call summary for Sequence - Master Receiver Mode, DMA Method in F/S and HS Modes**

Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_BUFSTAT <sup>(1)</sup>
I2Ci.I2C_CON <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

**Figure 23-35. HS I<sup>2</sup>C Slave Transmitter/Receiver Mode, Polling**


i2c-034

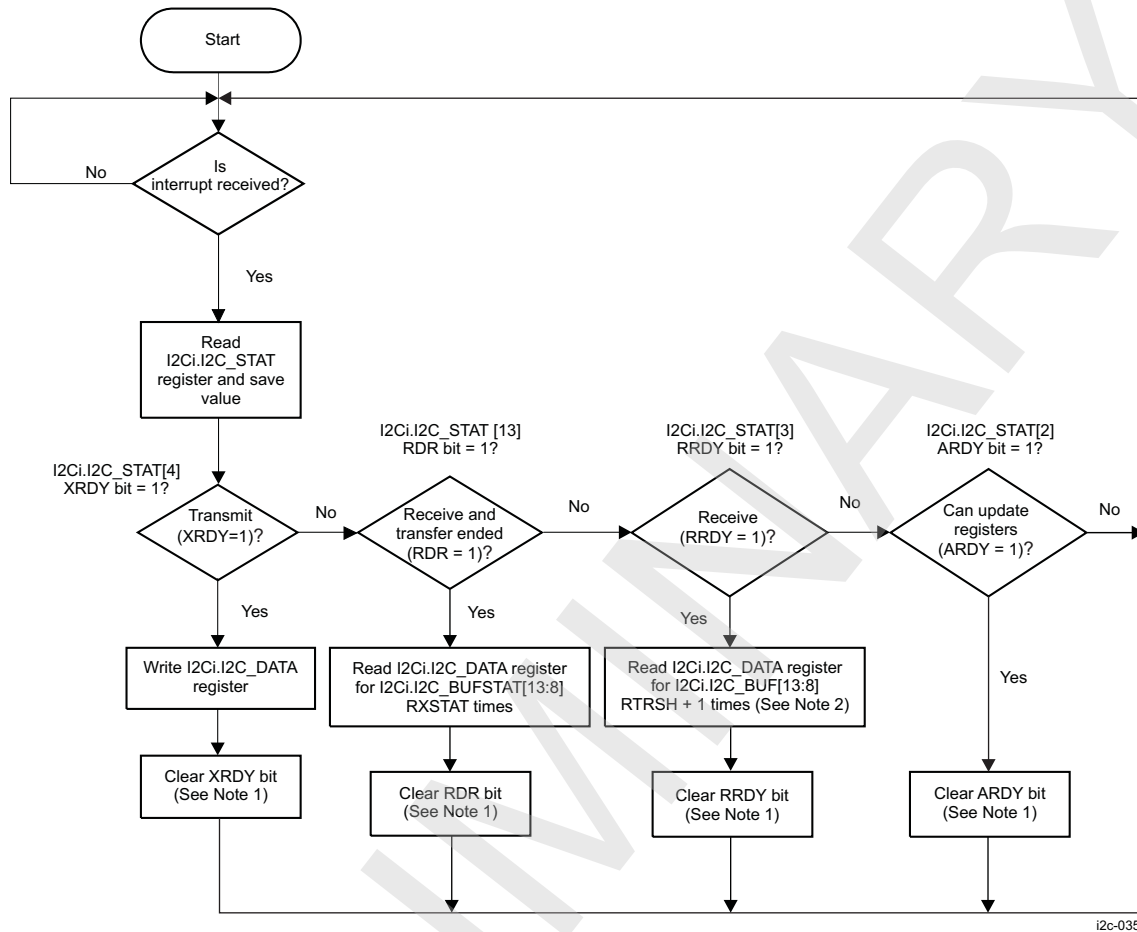
- (1) The XRDY, RDR, RRDY, and ARDY bits are cleared by writing 1 to each corresponding bit in the I2Ci.I2C\_STAT register.
- (2) In slave transmitter mode, the amount of data requested by the external master I<sup>2</sup>C device is unknown; thus, the I2Ci.I2C\_BUF[5:0] TXTRSH bit field must be configured to 0x0 (TX threshold = 1).

**Table 23-23. HS I<sup>2</sup>C Register call summary for Sequence - Slave Transmitter/Receiver Mode, Polling**

Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_BUFSTAT <sup>(1)</sup>
I2Ci.I2C_DATA <sup>(1)</sup>	I2Ci.I2C_BUF <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

Figure 23-36. HS I<sup>2</sup>C Slave Transmitter/Receiver Mode, Interrupt



- (1) The XRDY, RDR, RRDY, and ARDY bits are cleared by writing 1 to each corresponding bit in the I2Ci.I2C\_STAT register.
- (2) In slave transmitter mode, the amount of data requested by the external master I<sup>2</sup>C device is unknown; thus, the I2Ci.I2C\_BUF[5:0] TXTRSH bit field must be configured to 0x0 (TX threshold = 1).

Table 23-24. HS I<sup>2</sup>C Register call summary for Sequence - Slave Transmitter/Receiver Mode, Interrupt

Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_BUFSTAT <sup>(1)</sup>
I2Ci.I2C_DATA <sup>(1)</sup>	I2Ci.I2C_BUF <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

### 23.1.5.1.2 HS I<sup>2</sup>C (I2C5) Programming Model for Communication With Power Chips

This section describes the programming model of the master transmitter HS I<sup>2</sup>C controller (I2C5) for communication with one or more power chips. The HS I<sup>2</sup>C controller (I2C5) allows the two voltage Finite State Machines (FSMs) in the PRCM module of the device to be interfaced to external power chips through the I<sup>2</sup>C bus for dynamic voltage control of two power supplies and power sequencing. The primary programming tasks are to set up the configuration registers according to the external power supply chip(s) being used.

#### 23.1.5.1.2.1 Configure the Voltage Controller Registers

To use the voltage control function, the LH must configure the following registers in the voltage controller of the APE and 3G modem SS PRCM modules:

- The voltage configuration register address for each VDD channel in the PRCM.PRM\_VC\_VAL\_SMPS\_RA\_VOL register
- The ON/ON-low-power-Retention/OFF command configuration register address for each VDD channel in the PRCM.PRM\_VC\_VAL\_SMPS\_RA\_CMD register
- The set of ON/ON-low-power/Retention/OFF voltage/mode values in the APE PRCM.PRM\_VC\_VAL\_CMD\_VDD\_CORE\_L, APE PRCM.PRM\_VC\_VAL\_CMD\_VDD\_IVA\_L, and APE PRCM.PRM\_VC\_VAL\_CMD\_VDD\_MPU\_L registers, and 3G modem SS PRCM.PRM\_VC\_VAL\_CMD\_VDD\_WGM\_L register.
- The VDD channels configuration selection in the PRCM.PRM\_VC\_CFG\_CHANNEL register

For more information about the voltage controller register configuration, see [Chapter 3, Power, Reset, and Clock Management](#).

### 23.1.5.1.2.2 Configure the HS I<sup>2</sup>C Controller I2C5

At power-on reset (POR), the I2C5 is in HS mode (PRCM.PRM\_VC\_CFG\_I2C\_MODE[3] HSMODEEN bit). In HS mode, the LH must configure the master code value for the preamble I<sup>2</sup>C high-speed transmission by configuring the PRCM.PRM\_VC\_CFG\_I2C\_MODE[2:0] HSMCODE bit field. If the external power chips do not support the I<sup>2</sup>C HS mode, HS mode can be disabled and F/S mode enabled by clearing the PRCM.PRM\_VC\_CFG\_I2C\_MODE[3] HSMODEEN bit to 0.

By default, the Sr operation is enabled (PRCM.PRM\_VC\_CFG\_I2C\_MODE[4] SRMODEEN bit set to 1 at POR). In F/S mode, the Sr operation can be disabled by clearing the PRCM.PRM\_VC\_CFG\_I2C\_MODE[4] SRMODEEN bit to 0.

### 23.1.5.1.3 HS I<sup>2</sup>C Basic Programming Model in SCCB Mode

This section describes the programming model of the multimaster HS I<sup>2</sup>C controllers configured in SCCB mode.

#### 23.1.5.1.3.1 Main Program

##### 23.1.5.1.3.1.1 Configure the Module Before Enabling the I<sup>2</sup>C Controller

Before enabling the I<sup>2</sup>C controller, perform the following steps:

1. Enable the functional and interface clocks (see [Table 23-7](#)).
2. Program the prescaler to obtain an approximately 12-MHz internal sampling clock (I2Ci\_INTERNAL\_CLK) by programming the corresponding value in the I2Ci.I2C\_PSC[7:0] PSC field. This value depends on the frequency of the functional clock (I2Ci\_FCLK). Because this frequency is 96MHz, the I2Ci.I2C\_PSC[7:0] PSC field value is 0x7.
3. Program the I2Ci.I2C\_SCLL[7:0] SCLL and I2Ci.I2C\_SCLH[7:0] SCLH fields to obtain a bit rate of 100K bps (maximum authorized bit rate in SCCB mode). This value depends on the internal sampling clock frequency (see [Table 23-8](#)).
4. Configure the 7-bit slave address (ID value) by storing it in the I2Ci.I2C\_SA register.
5. Configure the 8-bit register address (subaddress) by storing it in the I2Ci.I2C\_OA0 register.
6. Configure the I2Ci.I2C\_BUF[13:8]RTRSH field to 0x0 (RX threshold to 1) and the I2Ci.I2C\_BUF[5:0]XTRSH field to 0x0 (TX threshold to 1).
7. Take the I<sup>2</sup>C controller out of reset by setting the I2Ci.I2C\_CON[15] I2C\_EN bit to 1.

##### 23.1.5.1.3.1.2 Initialize the I<sup>2</sup>C Controller

To initialize the I<sup>2</sup>C controller, perform the following steps:

1. Configure the I2Ci.I2C\_CON register:
  - In SCCB mode, only the master mode is supported; set the I2Ci.I2C\_CON[10] MST bit to 1.
  - For transmitter mode (write to the external SCCB device register) or receiver mode (read from the external SCCB device register), set the I2Ci.I2C\_CON[9] TRX bit (0: receiver, 1: transmitter).
2. If using an interrupt to transmit/receive data, set to 1 the corresponding bit in the I2Ci.I2C\_IE register (the I2Ci.I2C\_IE[4] XRDY\_IE bit for the transmit interrupt, the I2Ci.I2C\_IE[3] RRDY bit for the receive

interrupt).

#### 23.1.5.1.3.1.3 Initiate a Transfer

Poll the I2Ci.I2C\_STAT[12] BB bit. If it is cleared to 0 (bus not busy), set the I2Ci.I2C\_CON[0] STT bit to 1. Because a transfer allows the LH to write or read only a single byte to or from the external SCCB device, the transmission automatically stops at the end of the transfer. When the transfer is complete, the I2Ci.I2C\_STAT[2] ARDY bit is set to 1. In SCCB mode, the I2Ci.I2C\_CON[1] STP bit is not used.

#### 23.1.5.1.3.1.4 Receive Data

Poll the I2Ci.I2C\_STAT[3] RRDY bit, or use the RRDY interrupt (the I2Ci.I2C\_IE[3] RRDY\_IE bit must be set to 1) to read the receive data in the I2Ci.I2C\_DATA register.

---

**NOTE:** In SCCB mode, the I2Ci.I2C\_BUF[13:8] RTRSH field (RX threshold) must be set to a value of 0x0 (RX threshold = 1).

---

#### 23.1.5.1.3.1.5 Transmit Data

Poll the I2Ci.I2C\_STAT[4] XRDY bit, or use the XRDY interrupt (the I2Ci.I2C\_IE[4] XRDY\_IE bit must be set to 1) to write the data to the I2Ci.I2C\_DATA register.

---

**NOTE:** In SCCB mode, the I2Ci.I2C\_BUF[5:0] TXTRSH field (TX threshold) must be set to a value of 0x0 (TX threshold = 1).

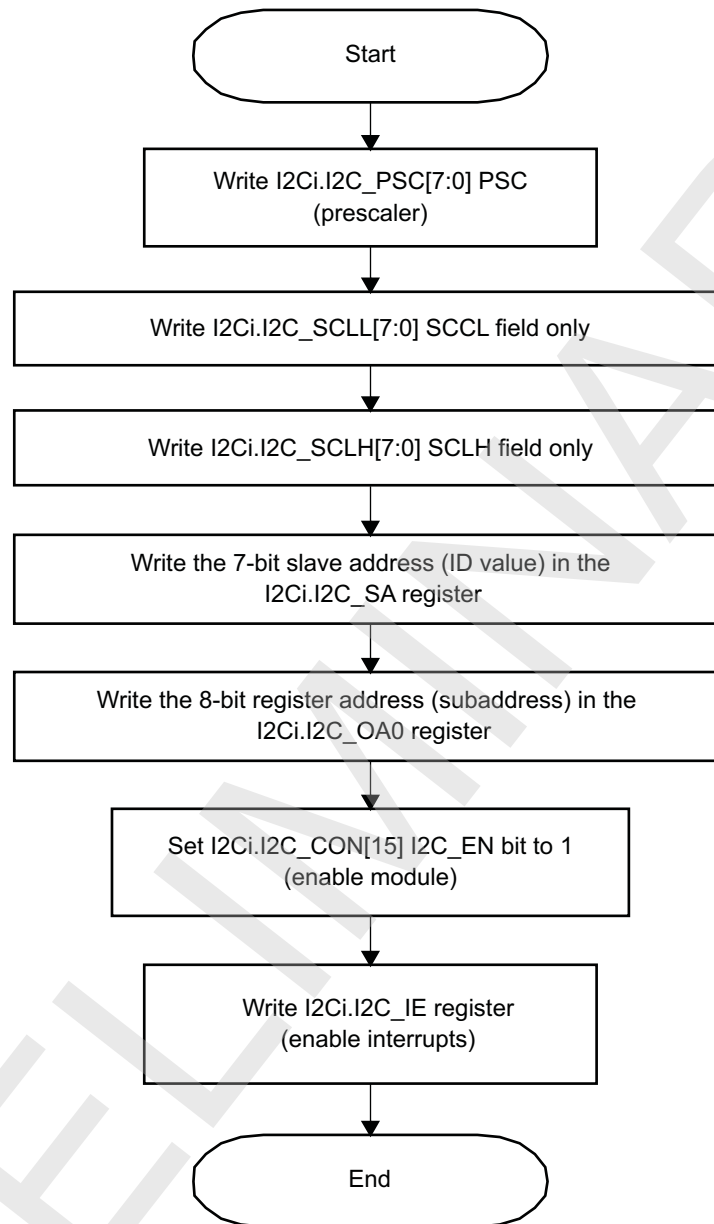
---

#### 23.1.5.1.3.2 Interrupt Subroutine Sequence

1. Test for register access ready (I2Ci.I2C\_STAT[2] ARDY status bit) and resolve accordingly.
2. Test for receive data ready (I2Ci.I2C\_STAT[3] RRDY status bit) and resolve accordingly.
3. Test for transmit data ready (I2Ci.I2C\_STAT[4] XRDY status bit) and resolve accordingly.

#### 23.1.5.1.3.3 Programming Flow Diagrams

Figure 23-37 through Figure 23-41 are procedure flowcharts for programming the SCCB mode.

**Figure 23-37. HS I<sup>2</sup>C SCCB Setup Procedure**

i2c-037

**Table 23-25. HS I<sup>2</sup>C Register call summary for Sequence - SCCB Setup Procedure**

Register Name	Register Name	Register Name
I2Ci.I2C_PSC <sup>(1)</sup>	I2Ci.I2C_SCLL <sup>(1)</sup>	I2Ci.I2C_SCLH <sup>(1)</sup>
I2Ci.I2C_SA <sup>(1)</sup>	I2Ci.I2C_OA0 <sup>(1)</sup>	I2Ci.I2C_CON <sup>(1)</sup>
I2Ci.I2C_IE <sup>(1)</sup>		

<sup>(1)</sup> i = 1 to 4



Figure 23-38. HS I<sup>2</sup>C SCCB Master Transmitter Mode, Polling

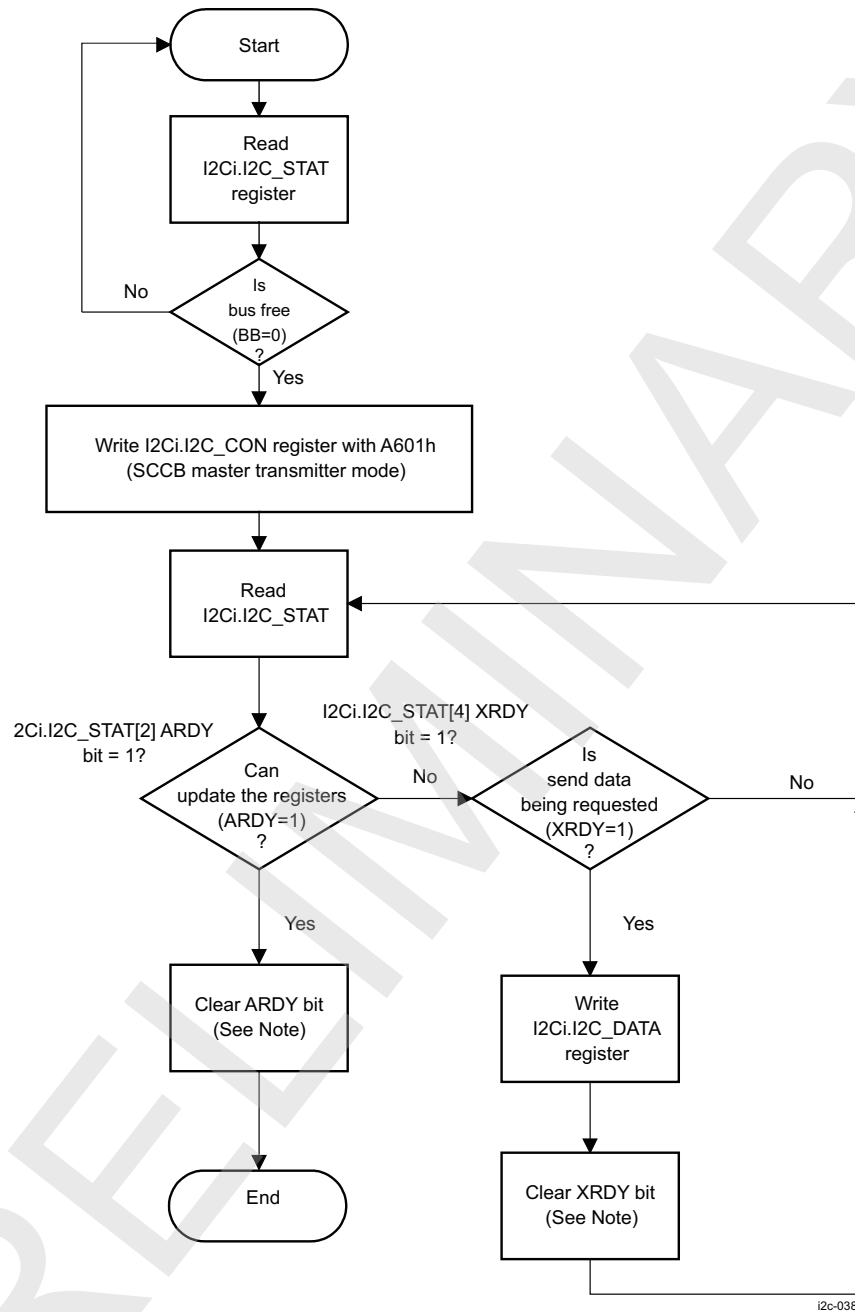


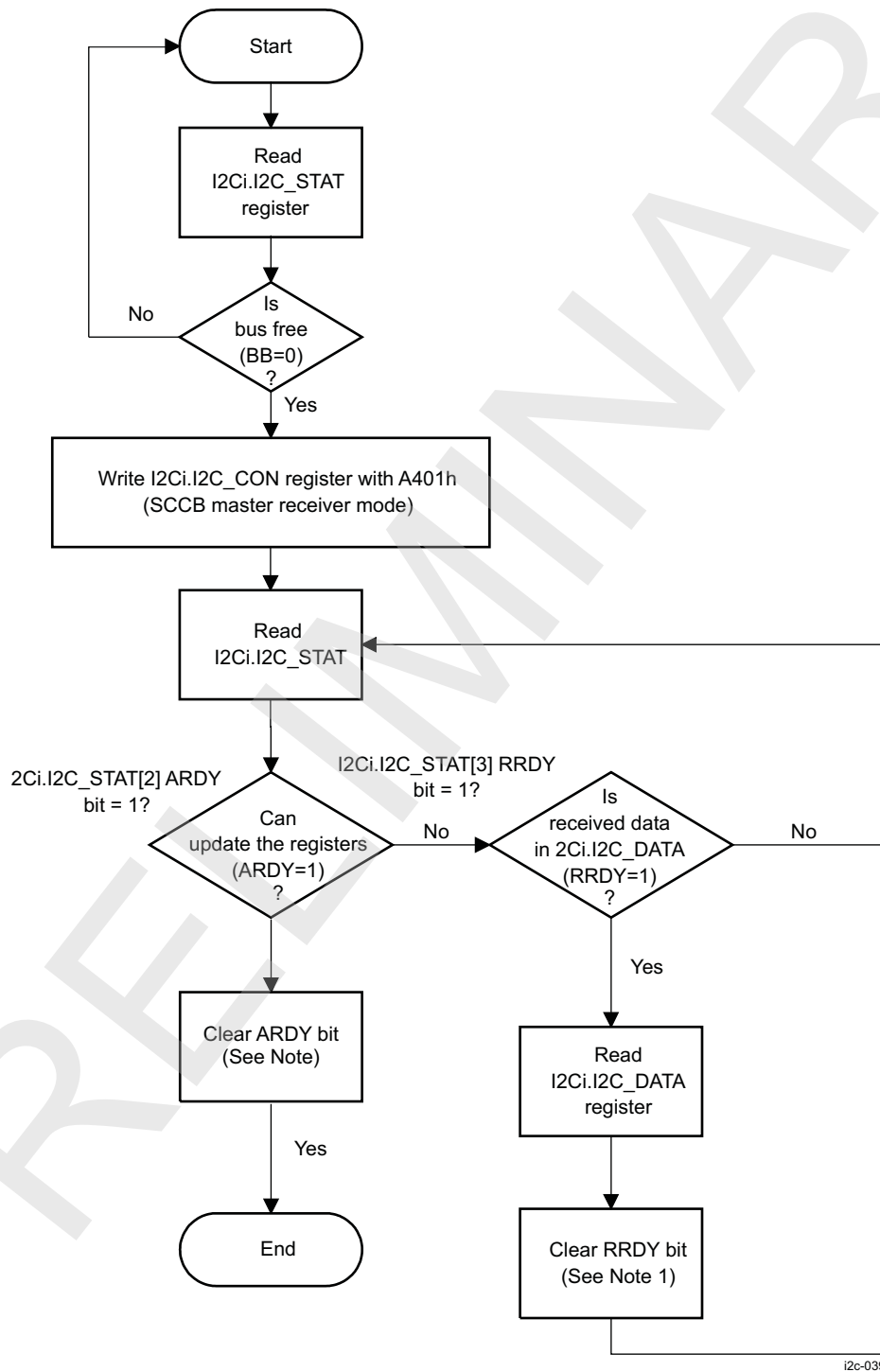
Table 23-26. HS I<sup>2</sup>C Register call summary for Sequence - SCCB Master Transmitter Mode, Polling

Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_SCLL <sup>(1)</sup>
I2Ci.I2C_CON <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

**NOTE:** The XRDY and ARDY bits are cleared by writing 1 to the corresponding bit in the I2Ci.I2C\_STAT register.

**Figure 23-39. HS I<sup>2</sup>C SCCB Master Receiver Mode, Polling**



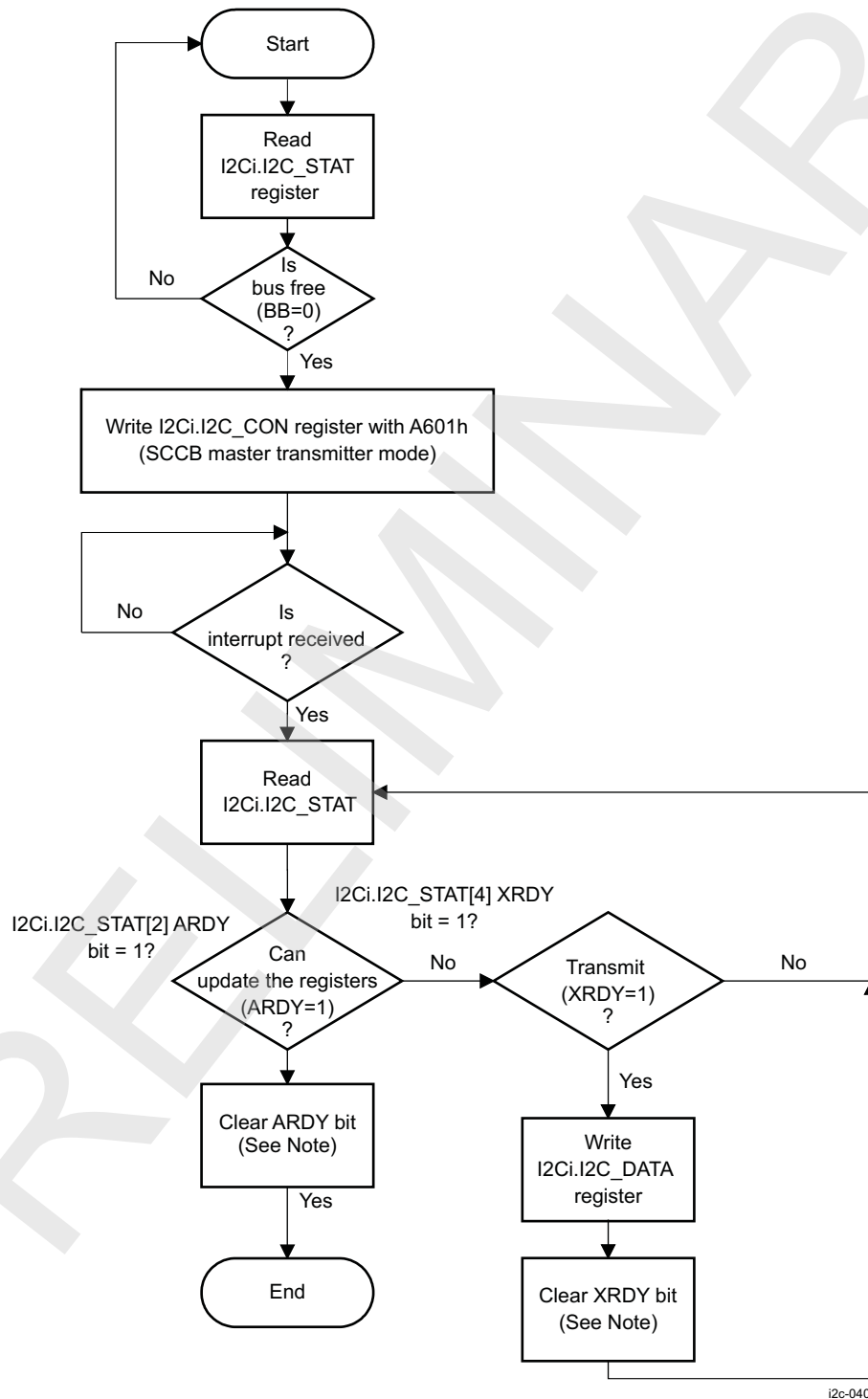
**Table 23-27. HS I<sup>2</sup>C Register call summary for Sequence - SCCB Master Receiver Mode, Polling**

Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_SCLL <sup>(1)</sup>
I2Ci.I2C_CON <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

**NOTE:** The RRDY and ARDY bits are cleared by writing 1 in the corresponding bit in the I2Ci.I2C\_STAT register.

**Figure 23-40. HS I<sup>2</sup>C SCCB Master Transmitter Mode, Interrupt**



**Table 23-28. HS I<sup>2</sup>C Register call summary for Sequence - SCCB Master Transmitter Mode, Interrupt**

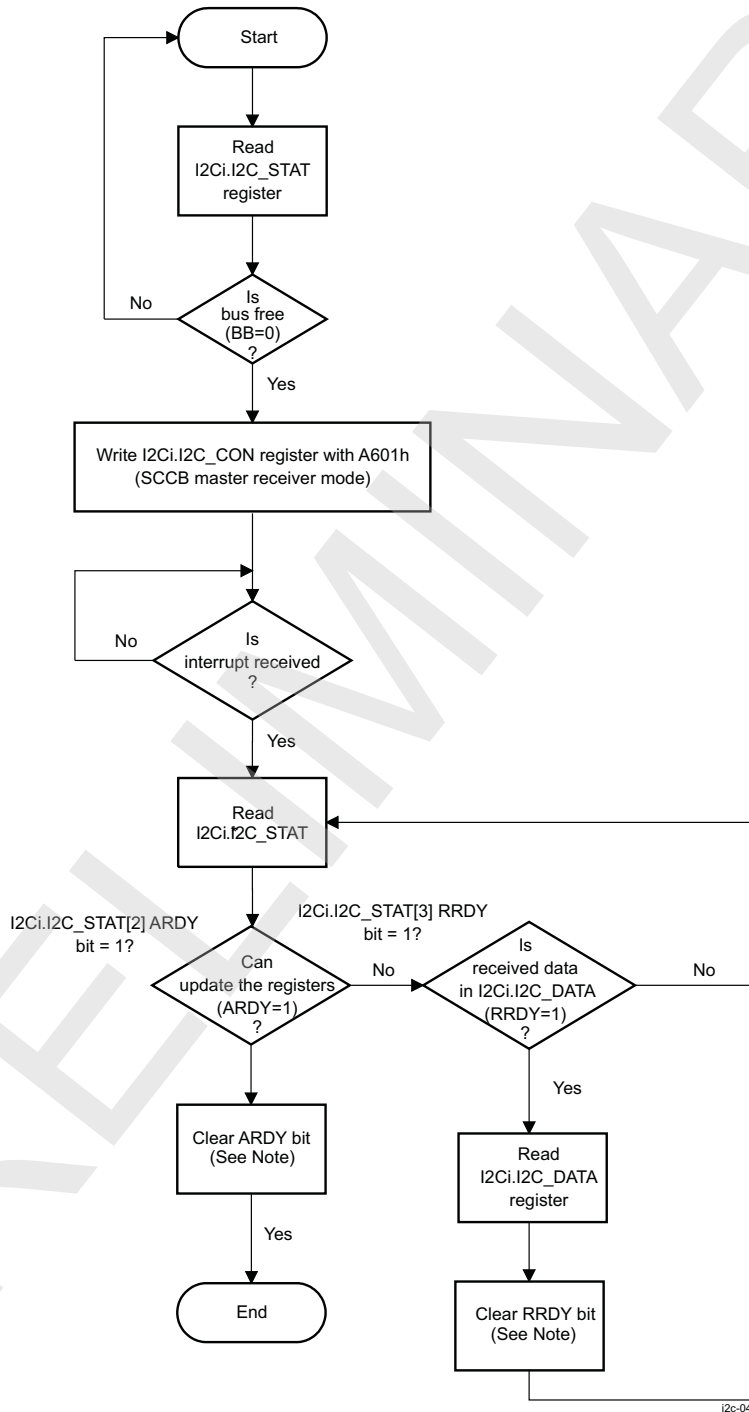
Register Name	Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>	I2Ci.I2C_CON <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

PRELIMINARY

**NOTE:** The XRDY and ARDY bits are cleared by writing 1 in the corresponding bit in the I2Ci.I2C\_STAT register.

**Figure 23-41. HS I<sup>2</sup>C SCCB Master Receiver Mode, Interrupt**



**Table 23-29. HS I<sup>2</sup>C Register call summary for Sequence - SCCB Master Receiver Mode, Interrupt**

Register Name	Register Name	Register Name
I2Ci.I2C_STAT <sup>(1)</sup>	I2Ci.I2C_DATA <sup>(1)</sup>	I2Ci.I2C_CON <sup>(1)</sup>

<sup>(1)</sup> i = 1 to 4

**NOTE:** The RRDY and ARDY bits are cleared by writing 1 in the corresponding bit in the I2Ci.I2C\_STAT register.

### 23.1.6 HS I<sup>2</sup>C Register Manual

#### 23.1.6.1 HS I<sup>2</sup>C Instance Summary

**Table 23-30. HS I<sup>2</sup>C Instance Summary**

Module Name	Module Base Address	Size
I2C3	0x4806 0000	256 bytes
I2C1	0x4807 0000	256 bytes
I2C2	0x4807 2000	256 bytes
I2C4	0x4835 0000	256 bytes

#### 23.1.6.2 HS I<sup>2</sup>C Registers

**CAUTION**

The HS I2Ci registers are limited to 16 bit and 8 bit data accesses; 32-bit data access is not allowed and can corrupt register content.

**NOTE:** For register details about the master transmitter HS I2C controller I2C5, see [Chapter 3, Power, Reset, and Clock Management](#).

#### 23.1.6.2.1 HS I<sup>2</sup>C Register Summary

**Table 23-31. HS I<sup>2</sup>C Registers Mapping Summary 1**

Register Name	Type	Register Width (Bits)	Address Offset	I2C3 Physical Address	I2C1 Physical Address
I2C_REVNB_LO	R	16	0x0000 0000	0x4806 0000	0x4807 0000
I2C_REVNB_HI	R	16	0x0000 0004	0x4806 0004	0x4807 0004
I2C_SYSC	RW	16	0x0000 0010	0x4806 0010	0x4807 0010
RESERVED	W	16	0x0000 0020	0x4806 0020	0x4807 0020
I2C_IRQSTATUS_RAW	RW	16	0x0000 0024	0x4806 0024	0x4807 0024
I2C_IRQSTATUS	RW	16	0x0000 0028	0x4806 0028	0x4807 0028
I2C_IRQENABLE_SET	RW	16	0x0000 002C	0x4806 002C	0x4807 002C
I2C_IRQENABLE_CLR	RW	16	0x0000 0030	0x4806 0030	0x4807 0030
I2C_WE	RW	16	0x0000 0034	0x4806 0034	0x4807 0034
I2C_DMARXENABLE_SET	RW	16	0x0000 0038	0x4806 0038	0x4807 0038
I2C_DMATXENABLE_SET	RW	16	0x0000 003C	0x4806 003C	0x4807 003C
I2C_DMARXENABLE_CLR	RW	16	0x0000 0040	0x4806 0040	0x4807 0040
I2C_DMATXENABLE_CLR	RW	16	0x0000 0044	0x4806 0044	0x4807 0044

**Table 23-31. HS I<sup>2</sup>C Registers Mapping Summary 1 (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	I2C3 Physical Address	I2C1 Physical Address
I2C_DMARXWAKE_EN	RW	16	0x0000 0048	0x4806 0048	0x4807 0048
I2C_DMATXWAKE_EN	RW	16	0x0000 004C	0x4806 004C	0x4807 004C
I2C_IE	RW	16	0x0000 0084	0x4806 0084	0x4807 0084
I2C_STAT	RW	16	0x0000 0088	0x4806 0088	0x4807 0088
I2C_SYSS	RW	16	0x0000 0090	0x4806 0090	0x4807 0090
I2C_BUF	RW	16	0x0000 0094	0x4806 0094	0x4807 0094
I2C_CNT	RW	16	0x0000 0098	0x4806 0098	0x4807 0098
I2C_DATA	RW	16	0x0000 009C	0x4806 009C	0x4807 009C
I2C_CON	RW	16	0x0000 00A4	0x4806 00A4	0x4807 00A4
I2C_OA	RW	16	0x0000 00A8	0x4806 00A8	0x4807 00A8
I2C_SA	RW	16	0x0000 00AC	0x4806 00AC	0x4807 00AC
I2C_PSC	RW	16	0x0000 00B0	0x4806 00B0	0x4807 00B0
I2C_SCLL	RW	16	0x0000 00B4	0x4806 00B4	0x4807 00B4
I2C_SCLH	RW	16	0x0000 00B8	0x4806 00B8	0x4807 00B8
I2C_SYSTEST	RW	16	0x0000 00BC	0x4806 00BC	0x4807 00BC
I2C_BUFSTAT	R	16	0x0000 00C0	0x4806 00C0	0x4807 00C0
I2C_OA1	RW	16	0x0000 00C4	0x4806 00C4	0x4807 00C4
I2C_OA2	RW	16	0x0000 00C8	0x4806 00C8	0x4807 00C8
I2C_OA3	RW	16	0x0000 00CC	0x4806 00CC	0x4807 00CC
I2C_ACTOA	R	16	0x0000 00D0	0x4806 00D0	0x4807 00D0
I2C_SBLOCK	RW	16	0x0000 00D4	0x4806 00D4	0x4807 00D4

**Table 23-32. HS I<sup>2</sup>C Registers Mapping Summary 2**

Register Name	Type	Register Width (Bits)	Address Offset	I2C2 Physical Address	I2C4 Physical Address
I2C_REVNB_LO	R	16	0x0000 0000	0x4807 2000	0x4835 0000
I2C_REVNB_HI	R	16	0x0000 0004	0x4807 2004	0x4835 0004
I2C_SYSC	RW	16	0x0000 0010	0x4807 2010	0x4835 0010
RESERVED	W	16	0x0000 0020	0x4807 2020	0x4835 0020
I2C_IRQSTATUS_RAW	RW	16	0x0000 0024	0x4807 2024	0x4835 0024
I2C_IRQSTATUS	RW	16	0x0000 0028	0x4807 2028	0x4835 0028
I2C_IRQENABLE_SET	RW	16	0x0000 002C	0x4807 202C	0x4835 002C
I2C_IRQENABLE_CLR	RW	16	0x0000 0030	0x4807 2030	0x4835 0030
I2C_WE	RW	16	0x0000 0034	0x4807 2034	0x4835 0034
I2C_DMARXENABLE_SET	RW	16	0x0000 0038	0x4807 2038	0x4835 0038
I2C_DMATXENABLE_SET	RW	16	0x0000 003C	0x4807 203C	0x4835 003C
I2C_DMARXENABLE_CLR	RW	16	0x0000 0040	0x4807 2040	0x4835 0040
I2C_DMATXENABLE_CLR	RW	16	0x0000 0044	0x4807 2044	0x4835 0044
I2C_DMARXWAKE_EN	RW	16	0x0000 0048	0x4807 2048	0x4835 0048
I2C_DMATXWAKE_EN	RW	16	0x0000 004C	0x4807 204C	0x4835 004C
I2C_IE	RW	16	0x0000 0084	0x4807 2084	0x4835 0084
I2C_STAT	RW	16	0x0000 0088	0x4807 2088	0x4835 0088
I2C_SYSS	RW	16	0x0000 0090	0x4807 2090	0x4835 0090
I2C_BUF	RW	16	0x0000 0094	0x4807 2094	0x4835 0094
I2C_CNT	RW	16	0x0000 0098	0x4807 2098	0x4835 0098



**Table 23-32. HS I<sup>2</sup>C Registers Mapping Summary 2 (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	I2C2 Physical Address	I2C4 Physical Address
I2C_DATA	RW	16	0x0000 009C	0x4807 209C	0x4835 009C
I2C_CON	RW	16	0x0000 00A4	0x4807 20A4	0x4835 00A4
I2C_OA	RW	16	0x0000 00A8	0x4807 20A8	0x4835 00A8
I2C_SA	RW	16	0x0000 00AC	0x4807 20AC	0x4835 00AC
I2C_PSC	RW	16	0x0000 00B0	0x4807 20B0	0x4835 00B0
I2C_SCLL	RW	16	0x0000 00B4	0x4807 20B4	0x4835 00B4
I2C_SCLH	RW	16	0x0000 00B8	0x4807 20B8	0x4835 00B8
I2C_SYSTEST	RW	16	0x0000 00BC	0x4807 20BC	0x4835 00BC
I2C_BUFSTAT	R	16	0x0000 00C0	0x4807 20C0	0x4835 00C0
I2C_OA1	RW	16	0x0000 00C4	0x4807 20C4	0x4835 00C4
I2C_OA2	RW	16	0x0000 00C8	0x4807 20C8	0x4835 00C8
I2C_OA3	RW	16	0x0000 00CC	0x4807 20CC	0x4835 00CC
I2C_ACTOA	R	16	0x0000 00D0	0x4807 20D0	0x4835 00D0
I2C_SBLOCK	RW	16	0x0000 00D4	0x4807 20D4	0x4835 00D4

**23.1.6.2.2 HS I<sup>2</sup>C Register Description**

**Table 23-33. I2C\_REVNB\_LO**

<b>Address Offset</b>	0x0000 0000
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

**Table 23-34. Register Call Summary for Register I2C\_REVNB\_LO**

- HS IC Register Manual
- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-35. I2C\_REVNB\_HI**

<b>Address Offset</b>	0x0000 0004
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

**Table 23-36. Register Call Summary for Register I2C\_REVNB\_HI**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-37. I2C\_SYSC**

<b>Address Offset</b>	0x0000 0010
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	System Configuration register
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						CLKACTIVITY		RESERVED			IDLEMODE	ENAWAKEUP	SRST	AUTOIDLE	

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved.	R	0x00
9:8	CLKACTIVITY	Clock Activity selection bits 0x0: Both clocks can be cut off 0x1: Only OCP clock must be kept active; system clock can be cut off 0x3: Both clocks must be kept active 0x2: Only system clock must be kept active; OCP clock can be cut off	RW	0x0
7:5	RESERVED	Reads return 0.	R	0x0
4:3	IDLEMODE	Idle Mode selection bits 0x0: Force Idle mode 0x1: No Idle mode 0x3: smartidle_wakeup 0x2: Smart Idle mode	RW	0x0
2	ENAWAKEUP	Enable Wakeup control bit 0x0: Wakeup mechanism is disabled 0x1: Wakeup mechanism is enabled	RW	0
1	SRST	SoftReset bit 0x0: Normal mode 0x1: The module is reset	RW	0
0	AUTOIDLE	Autoidle bit 0x0: Auto Idle mechanism is disabled 0x1: Auto Idle mechanism is enabled	RW	1

**Table 23-38. Register Call Summary for Register I2C\_SYSC**

HS IC Functional Description

- [HS I2C Software Reset: \[0\] \[1\]](#)
- [HS I2C Power Management: \[2\] \[3\] \[4\] \[5\] \[6\]](#)

HS IC Register Manual

- [HS I2C Register Summary: \[7\] \[8\]](#)

**Table 23-39. I2C\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0024
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Per-event raw interrupt status vector
<b>Type</b>	RW

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		XDR	RDR	BB	ROVR	XUDF	AAS	BF	AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
14	XDR	Transmit draining IRQ status. 0x0: Transmit draining inactive. 0x1: Transmit draining enabled.	RW	0
13	RDR	Receive draining IRQ status. 0x0: Receive draining inactive. 0x1: Receive draining enabled.	RW	0
12	BB	Bus busy status. Writing into this bit has no effect. Read 0x1: Bus is occupied. Read 0x0: Bus is free.	R	0
11	ROVR	Receive overrun status. Writing into this bit has no effect. Read 0x1: Receiver overrun. Read 0x0: Normal operation.	RW	0
10	XUDF	Transmit underflow status. Writing into this bit has no effect. Read 0x1: Transmit underflow. Read 0x0: Normal operation.	RW	0
9	AAS	Address recognized as slave IRQ status. 0x0: No action. 0x1: Address recognized.	RW	0
8	BF	Bus Free IRQ status. 0x0: No action. 0x1: Bus Free.	RW	0
7	AERR	Access Error IRQ status. 0x0: No action. 0x1: Access Error.	RW	0
6	STC	Start Condition IRQ status. 0x0: No action. 0x1: Start Condition detected.	RW	0
5	GC	General call IRQ status. Set to '1' by core when General call address detected and interrupt signaled to MPUSS. Write '1' to clear. 0x0: No general call detected. 0x1: General call address detected.	RW	0

Bits	Field Name	Description	Type	Reset
4	XRDY	Transmit data ready IRQ status. Set to '1' by core when transmitter and when new data is requested. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. 0x0: Transmission ongoing. 0x1: Transmit data ready.	RW	0
3	RRDY	Receive data ready IRQ status. Set to '1' by core when receiver mode, a new data is able to be read. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. 0x0: No data available. 0x1: Receive data available.	RW	0
2	ARDY	Register access ready IRQ status. When set to '1' it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to MPUSS. Write '1' to clear. 0x0: Module busy. 0x1: Access ready.	RW	0
1	NACK	No acknowledgement IRQ status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPUSS. Write '1' to clear this bit. 0x0: Normal operation. 0x1: Not Acknowledge detected.	RW	0
0	AL	Arbitration lost IRQ status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to MPUSS. During reads, it always returns 0. 0x0: Normal operation. 0x1: Arbitration lost detected.	RW	0

**Table 23-40. Register Call Summary for Register I2C\_IRQSTATUS\_RAW**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-41. I2C\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0028															
<b>Physical Address</b>	See <a href="#">Table 23-31</a>															
<b>Description</b>	Per-event enabled interrupt status vector															
<b>Type</b>	RW															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>RESERVED</b>		XDR	RDR	BB	ROVR	XUDF	AAS	BF	AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL
<b>Bits</b>	<b>Field Name</b>															
15	RESERVED															
14	XDR															
	<b>Description</b>															
	Write 0s for future compatibility. Read returns 0.															
	<b>Type</b>															
	RW															
	<b>Reset</b>															
	0															
	<b>Description</b>															
	Transmit draining IRQ enabled status.															
	0x0: Transmit draining inactive.															
	0x1: Transmit draining enabled.															

Bits	Field Name	Description	Type	Reset
13	RDR	Receive draining IRQ enabled status. 0x0: Receive draining inactive. 0x1: Receive draining enabled.	RW W1toClr	0
12	BB	Bus busy enabled status. Writing into this bit has no effect. Read 0x1: Bus is occupied. Read 0x0: Bus is free.	R	0
11	ROVR	Receive overrun enabled status. Writing into this bit has no effect. Read 0x1: Receiver overrun. Read 0x0: Normal operation.	RW W1toClr	0
10	XUDF	Transmit underflow enabled status. Writing into this bit has no effect. Read 0x1: Transmit underflow. Read 0x0: Normal operation.	RW W1toClr	0
9	AAS	Address recognized as slave IRQ enabled status. 0x0: No action. 0x1: Address recognized.	RW W1toClr	0
8	BF	Bus Free IRQ enabled status. 0x0: No action. 0x1: Bus Free.	RW W1toClr	0
7	AERR	Access Error IRQ enabled status. 0x0: No action. 0x1: Access Error.	RW W1toClr	0
6	STC	Start Condition IRQ enabled status. 0x0: No action. 0x1: Start Condition detected.	RW W1toClr	0
5	GC	General call IRQ enabled status. Set to '1' by core when General call address detected and interrupt signaled to MPUSS. Write '1' to clear. 0x0: No general call detected. 0x1: General call address detected.	RW W1toClr	0
4	XRDY	Transmit data ready IRQ enabled status. Set to '1' by core when transmitter and when new data is requested. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. 0x0: Transmission ongoing. 0x1: Transmit data ready.	RW W1toClr	0
3	RRDY	Receive data ready IRQ enabled status. Set to '1' by core when receiver mode, a new data is able to be read. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. 0x0: No data available. 0x1: Receive data available.	RW W1toClr	0
2	ARDY	Register access ready IRQ enabled status. When set to '1' it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to MPUSS. Write '1' to clear. 0x0: Module busy. 0x1: Access ready.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
1	NACK	No acknowledgement IRQ enabled status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPUSS. Write '1' to clear this bit.  0x0: Normal operation. 0x1: Not Acknowledge detected.	RW W1toClr	0
0	AL	Arbitration lost IRQ enabled status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to MPUSS. During reads, it always returns 0.  0x0: Normal operation. 0x1: Arbitration lost detected.	RW W1toClr	0

**Table 23-42. Register Call Summary for Register I2C\_IRQSTATUS**

HS IC Functional Description

- [HS I2C Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

HS IC Register Manual

- [HS I2C Register Summary: \[15\] \[16\]](#)

**Table 23-43. I2C\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 002C
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Per-event interrupt enable bit vector.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR_IE	RDR_IE	RESERVED	ROVR	XUDF	ASS_IE	BF_IE	AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
14	XDR_IE	Transmit Draining interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[XDR]</a> .  0x0: Transmit Draining interrupt disabled 0x1: Transmit Draining interrupt enabled	RW	0
13	RDR_IE	Receive Draining interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[RDR]</a> .  0x0: Receive Draining interrupt disabled 0x1: Receive Draining interrupt enabled	RW	0
12	RESERVED	reserved	R	0
11	ROVR	Receive overrun enable set.  0x0: Receive overrun interrupt disabled 0x1: Receive Draining interrupt enabled	RW	0
10	XUDF	Transmit underflow enable set.  0x0: Transmit underflow interrupt disabled 0x1: Transmit underflow interrupt enabled	RW	0
9	ASS_IE	Addressed as Slave interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[AAS]</a> .  0x0: Addressed as Slave interrupt disabled 0x1: Addressed as Slave interrupt enabled	RW	0

Bits	Field Name	Description	Type	Reset
8	BF_IE	Bus Free interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[BF]</a> . 0x0: Bus Free interrupt disabled 0x1: Bus Free interrupt enabled	RW	0
7	AERR_IE	Access Error interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[AERR]</a> . 0x0: Access Error interrupt disabled 0x1: Access Error interrupt enabled	RW	0
6	STC_IE	Start Condition interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[STC]</a> . 0x0: Start Condition interrupt disabled 0x1: Start Condition interrupt enabled	RW	0
5	GC_IE	General call Interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[GC]</a> . 0x0: General call interrupt disabled 0x1: General call interrupt enabled	RW	0
4	XRDY_IE	Transmit data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[XRDY]</a> . 0x0: Transmit data ready interrupt disabled 0x1: Transmit data ready interrupt enabled	RW	0
3	RRDY_IE	Receive data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[RRDY]</a> . 0x0: Receive data ready interrupt disabled 0x1: Receive data ready interrupt enabled	RW	0
2	ARDY_IE	Register access ready interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[ARDY]</a> . 0x0: Register access ready interrupt disabled 0x1: Register access ready interrupt enabled	RW	0
1	NACK_IE	No acknowledgement interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[NACK]</a> . 0x0: Not Acknowledge interrupt disabled 0x1: Not Acknowledge interrupt enabled	RW	0
0	AL_IE	Arbitration lost interrupt enable set. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[AL]</a> . 0x0: Arbitration lost interrupt disabled 0x1: Arbitration lost interrupt enabled	RW	0

**Table 23-44. Register Call Summary for Register I2C\_IRQENABLE\_SET**

## HS IC Functional Description

- [HS I2C Interrupt Requests](#): [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14]

## HS IC Register Manual

- [HS I2C Register Summary](#): [15] [16]

**Table 23-45. I2C\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 0030
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Per-event interrupt clear bit vector.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR_IE	RDR_IE	RESERVED	ROVR	XUDF	ASS_IE	BF_IE	AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
14	XDR_IE	Transmit Draining interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[XDR]</a> . 0x0: Transmit Draining interrupt disabled 0x1: Transmit Draining interrupt enabled	RW	0
13	RDR_IE	Receive Draining interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[RDR]</a> . 0x0: Receive Draining interrupt disabled 0x1: Receive Draining interrupt enabled	RW	0
12	RESERVED	reserved	R	0
11	ROVR	Receive overrun enable clear. 0x0: Receive overrun interrupt disabled 0x1: Receive Draining interrupt enabled	RW	0
10	XUDF	Transmit underflow enable clear. 0x0: Transmit underflow interrupt disabled 0x1: Transmit underflow interrupt enabled	RW	0
9	ASS_IE	Addressed as Slave interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[AAS]</a> . 0x0: Addressed as Slave interrupt disabled 0x1: Addressed as Slave interrupt enabled	RW	0
8	BF_IE	Bus Free interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[BF]</a> . 0x0: Bus Free interrupt disabled 0x1: Bus Free interrupt enabled	RW	0
7	AERR_IE	Access Error interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[AERR]</a> . 0x0: Access Error interrupt disabled 0x1: Access Error interrupt enabled	RW	0
6	STC_IE	Start Condition interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[STC]</a> . 0x0: Start Condition interrupt disabled 0x1: Start Condition interrupt enabled	RW	0
5	GC_IE	General call Interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[GC]</a> . 0x0: General call interrupt disabled 0x1: General call interrupt enabled	RW	0



Bits	Field Name	Description	Type	Reset
4	XRDY_IE	Transmit data ready interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[XRDY]</a> 0x0: Transmit data ready interrupt disabled 0x1: Transmit data ready interrupt enabled	RW	0
3	RRDY_IE	Receive data ready interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[RRDY]</a> 0x0: Receive data ready interrupt disabled 0x1: Receive data ready interrupt enabled	RW	0
2	ARDY_IE	Register access ready interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[ARDY]</a> 0x0: Register access ready interrupt disabled 0x1: Register access ready interrupt enabled	RW	0
1	NACK_IE	No acknowledgement interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[NACK]</a> 0x0: Not Acknowledge interrupt disabled 0x1: Not Acknowledge interrupt enabled	RW	0
0	AL_IE	Arbitration lost interrupt enable clear. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[AL]</a> 0x0: Arbitration lost interrupt disabled 0x1: Arbitration lost interrupt enabled	RW	0

**Table 23-46. Register Call Summary for Register I2C\_IRQENABLE\_CLR**

HS IC Functional Description

- [HS I2C Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

HS IC Register Manual

- [HS I2C Register Summary: \[15\] \[16\]](#)

**Table 23-47. I2C\_WE**

<b>Address Offset</b>	0x0000 0034
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	I2C wakeup enable vector.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF	RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved.	R	0
14	XDR	Transmit Draining wakeup set. 0x0: Transmit draining wakeup disabled 0x1: Transmit draining wakeup enabled	RW	0
13	RDR	Receive Draining wakeup set. 0x0: Receive draining wakeup disabled 0x1: Receive draining wakeup enabled	RW	0
12	RESERVED	Reserved.	R	0

Bits	Field Name	Description	Type	Reset
11	ROVR	Receive overrun wakeup set. 0x0: Receive overrun wakeup disabled 0x1: Receive overrun wakeup enabled	RW	0
10	XUDF	Transmit underflow wakeup set. 0x0: Transmit underflow wakeup disabled 0x1: Transmit underflow wakeup enabled	RW	0
9	AAS	Address as slave IRQ wakeup set. 0x0: Addressed as slave wakeup disabled 0x1: Addressed as slave wakeup enabled	RW	0
8	BF	Bus Free IRQ wakeup set. 0x0: Bus Free wakeup disabled 0x1: Bus Free wakeup enabled	RW	0
7	RESERVED	Reserved.	R	0
6	STC	Start Condition IRQ wakeup set. 0x0: Start condition wakeup disabled 0x1: Start condition wakeup enabled	RW	0
5	GC	General call IRQ wakeup set. 0x0: General call wakeup disabled 0x1: General call wakeup enabled	RW	0
4	RESERVED	Reserved	R	0
3	DRDY	Receive/Transmit data ready IRQ wakeup set. 0x0: Transmit/receive data ready wakeup disabled 0x1: Transmit/receive data ready wakeup enabled	RW	0
2	ARDY	Register access ready IRQ wakeup set. 0x0: Register access ready wakeup disabled 0x1: Register access ready wakeup enabled	RW	0
1	NACK	No acknowledgment IRQ wakeup set. 0x0: Not Acknowledge wakeup disabled 0x1: Not Acknowledge wakeup enabled	RW	0
0	AL	Arbitration lost IRQ wakeup set. 0x0: Arbitration lost wakeup disabled 0x1: Arbitration lost wakeup enabled	RW	0

**Table 23-48. Register Call Summary for Register I2C\_WE**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-49. I2C\_DMARXENABLE\_SET**

<b>Address Offset</b>	0x0000 0038
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Per-event DMA RX enable set.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMARX_ENABLE_SET

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved.	R	0x0000
0	DMARX_ENABLE_SET	Receive DMA channel enable set.	RW	0

**Table 23-50. Register Call Summary for Register I2C\_DMARXENABLE\_SET**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-51. I2C\_DMATXENABLE\_SET**

<b>Address Offset</b>	0x0000 003C
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Per-event DMA TX enable set.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMATX_ENABLE_SET

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved.	R	0x0000
0	DMATX_ENABLE_SET	Transmit DMA channel enable set.	RW	0

**Table 23-52. Register Call Summary for Register I2C\_DMATXENABLE\_SET**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-53. I2C\_DMARXENABLE\_CLR**

<b>Address Offset</b>	0x0000 0040
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Per-event DMA RX enable clear.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMARX_ENABLE_CLEAR

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved.	R	0x0000
0	DMARX_ENABLE_CLEAR	Receive DMA channel enable clear.	RW	0

**Table 23-54. Register Call Summary for Register I2C\_DMARXENABLE\_CLR**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-55. I2C\_DMATXENABLE\_CLR**

<b>Address Offset</b>	0x0000 0044
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Per-event DMA TX enable clear.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMATX_ENABLE_CLEAR

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved.	R	0x0000
0	DMATX_ENABLE_CLEAR	Transmit DMA channel enable clear.	RW	0

**Table 23-56. Register Call Summary for Register I2C\_DMATXENABLE\_CLR**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-57. I2C\_DMARXWAKE\_EN**

<b>Address Offset</b>	0x0000 0048
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Per-event DMA RX wakeup enable.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF	RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved.	R	0
14	XDR	Transmit Draining wakeup set. 0x0: Transmit draining wakeup disabled 0x1: Transmit draining wakeup enabled	RW	0
13	RDR	Receive Draining wakeup set. 0x0: Receive draining wakeup disabled 0x1: Receive draining wakeup enabled	RW	0
12	RESERVED	Reserved.	R	0
11	ROVR	Receive overrun wakeup set. 0x0: Receive overrun wakeup disabled 0x1: Receive overrun wakeup enabled	RW	0
10	XUDF	Transmit underflow wakeup set. 0x0: Transmit underflow wakeup disabled 0x1: Transmit underflow wakeup enabled	RW	0
9	AAS	Address as slave IRQ wakeup set. 0x0: Addressed as slave wakeup disabled 0x1: Addressed as slave wakeup enabled	RW	0
8	BF	Bus Free IRQ wakeup set. 0x0: Bus Free wakeup disabled 0x1: Bus Free wakeup enabled	RW	0
7	RESERVED	Reserved.	R	0
6	STC	Start Condition IRQ wakeup set. 0x0: Start condition wakeup disabled 0x1: Start condition wakeup enabled	RW	0
5	GC	General call IRQ wakeup set. 0x0: General call wakeup disabled 0x1: General call wakeup enabled	RW	0
4	RESERVED	Reserved	R	0
3	DRDY	Receive/Transmit data ready IRQ wakeup set. 0x0: Transmit/receive data ready wakeup disabled 0x1: Transmit/receive data ready wakeup enabled	RW	0
2	ARDY	Register access ready IRQ wakeup set. 0x0: Register access ready wakeup disabled 0x1: Register access ready wakeup enabled	RW	0

Bits	Field Name	Description	Type	Reset
1	NACK	No acknowledgment IRQ wakeup set. 0x0: Not Acknowledge wakeup disabled 0x1: Not Acknowledge wakeup enabled	RW	0
0	AL	Arbitration lost IRQ wakeup set. 0x0: Arbitration lost wakeup disabled 0x1: Arbitration lost wakeup enabled	RW	0

**Table 23-58. Register Call Summary for Register I2C\_DMARXWAKE\_EN**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-59. I2C\_DMATXWAKE\_EN**

<b>Address Offset</b>	0x0000 004C
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Per-event DMA TX wakeup enable.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF	RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved.	R	0
14	XDR	Transmit Draining wakeup set. 0x0: Transmit draining wakeup disabled 0x1: Transmit draining wakeup enabled	RW	0
13	RDR	Receive Draining wakeup set. 0x0: Receive draining wakeup disabled 0x1: Receive draining wakeup enabled	RW	0
12	RESERVED	Reserved.	R	0
11	ROVR	Receive overrun wakeup set. 0x0: Receive overrun wakeup disabled 0x1: Receive overrun wakeup enabled	RW	0
10	XUDF	Transmit underflow wakeup set. 0x0: Transmit underflow wakeup disabled 0x1: Transmit underflow wakeup enabled	RW	0
9	AAS	Address as slave IRQ wakeup set. 0x0: Addressed as slave wakeup disabled 0x1: Addressed as slave wakeup enabled	RW	0
8	BF	Bus Free IRQ wakeup set. 0x0: Bus Free wakeup disabled 0x1: Bus Free wakeup enabled	RW	0
7	RESERVED	Reserved.	R	0
6	STC	Start Condition IRQ wakeup set. 0x0: Start condition wakeup disabled 0x1: Start condition wakeup enabled	RW	0

Bits	Field Name	Description	Type	Reset
5	GC	General call IRQ wakeup set. 0x0: General call wakeup disabled 0x1: General call wakeup enabled	RW	0
4	RESERVED	Reserved	R	0
3	DRDY	Receive/Transmit data ready IRQ wakeup set. 0x0: Transmit/receive data ready wakeup disabled 0x1: Transmit/receive data ready wakeup enabled	RW	0
2	ARDY	Register access ready IRQ wakeup set. 0x0: Register access ready wakeup disabled 0x1: Register access ready wakeup enabled	RW	0
1	NACK	No acknowledgment IRQ wakeup set. 0x0: Not Acknowledge wakeup disabled 0x1: Not Acknowledge wakeup enabled	RW	0
0	AL	Arbitration lost IRQ wakeup set. 0x0: Arbitration lost wakeup disabled 0x1: Arbitration lost wakeup enabled	RW	0

**Table 23-60. Register Call Summary for Register I2C\_DMATXWAKE\_EN**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-61. I2C\_IE**

<b>Address Offset</b>	0x0000 0084
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	I2C interrupt enable vector (legacy).
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR_IE	RDR_IE	RESERVED	ROVR	XUDF	ASS_IE	BF_IE	AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
14	XDR_IE	Transmit Draining interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[XDR]</a> . 0x0: Transmit Draining interrupt disabled 0x1: Transmit Draining interrupt enabled	RW	0
13	RDR_IE	Receive Draining interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[RDR]</a> . 0x0: Receive Draining interrupt disabled 0x1: Receive Draining interrupt enabled	RW	0
12	RESERVED	reserved	R	0
11	ROVR	Receive overrun enable set. 0x0: Receive overrun interrupt disabled 0x1: Receive Draining interrupt enabled	RW	0

Bits	Field Name	Description	Type	Reset
10	XUDF	Transmit underflow enable set. 0x0: Transmit underflow interrupt disabled 0x1: Transmit underflow interrupt enabled	RW	0
9	ASS_IE	Addressed as Slave interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[AAS]</a> . 0x0: Addressed as Slave interrupt disabled 0x1: Addressed as Slave interrupt enabled	RW	0
8	BF_IE	Bus Free interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[BF]</a> . 0x0: Bus Free interrupt disabled 0x1: Bus Free interrupt enabled	RW	0
7	AERR_IE	Access Error interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[AERR]</a> . 0x0: Access Error interrupt disabled 0x1: Access Error interrupt enabled	RW	0
6	STC_IE	Start Condition interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[STC]</a> . 0x0: Start Condition interrupt disabled 0x1: Start Condition interrupt enabled	RW	0
5	GC_IE	General call Interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[GC]</a> . 0x0: General call interrupt disabled 0x1: General call interrupt enabled	RW	0
4	XRDY_IE	Transmit data ready interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[XRDY]</a> . 0x0: Transmit data ready interrupt disabled 0x1: Transmit data ready interrupt enabled	RW	0
3	RRDY_IE	Receive data ready interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[RRDY]</a> . 0x0: Receive data ready interrupt disabled 0x1: Receive data ready interrupt enabled	RW	0
2	ARDY_IE	Register access ready interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[ARDY]</a> . 0x0: Register access ready interrupt disabled 0x1: Register access ready interrupt enabled	RW	0
1	NACK_IE	No acknowledgement interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[NACK]</a> . 0x0: Not Acknowledge interrupt disabled 0x1: Not Acknowledge interrupt enabled	RW	0
0	AL_IE	Arbitration lost interrupt enable. Mask or unmask the interrupt signaled by bit in <a href="#">I2C_STAT[AL]</a> . 0x0: Arbitration lost interrupt disabled 0x1: Arbitration lost interrupt enabled	RW	0

**Table 23-62. Register Call Summary for Register I2C\_IE**

## HS IC Functional Description

- [HS I2C FIFO Interrupt Mode: \[0\]](#)
- [HS I2C FIFO Polling Mode: \[1\] \[2\]](#)
- [HS I2C Draining Feature \(I2C Mode Only\): \[3\] \[4\]](#)



**Table 23-62. Register Call Summary for Register I2C\_IE (continued)**

HS IC Programming Guide
<ul style="list-style-type: none"> <li>• Main Program: [5] [6] [7] [8] [9] [10] [11] [12] [13]</li> <li>• Programming Flow Diagrams: [14]</li> <li>• Main Program: [15] [16] [17] [18] [19]</li> <li>• Programming Flow Diagrams: [20]</li> </ul>
HS IC Register Manual
<ul style="list-style-type: none"> <li>• HS I2C Register Summary: [21] [22]</li> </ul>

**Table 23-63. I2C\_STAT**

<b>Address Offset</b>	0x0000 0088
<b>Physical Address</b>	See Table 23-31
<b>Description</b>	I2C interrupt status vector (legacy).
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	BB	ROVR	XUDF	AAS	BF	AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
14	XDR	Transmit draining IRQ status. 0x0: Transmit draining inactive. 0x1: Transmit draining enabled.	RW W1toClr	0
13	RDR	Receive draining IRQ status. 0x0: Receive draining inactive. 0x1: Receive draining enabled.	RW W1toClr	0
12	BB	Bus busy status. Writing into this bit has no effect. Read 0x1: Bus is occupied. Read 0x0: Bus is free.	R	0
11	ROVR	Receive overrun status. Writing into this bit has no effect. Read 0x1: Receiver overrun. Read 0x0: Normal operation.	RW W1toClr	0
10	XUDF	Transmit underflow status. Writing into this bit has no effect. Read 0x1: Transmit underflow. Read 0x0: Normal operation.	RW W1toClr	0
9	AAS	Address recognized as slave IRQ status. 0x0: No action. 0x1: Address recognized.	RW W1toClr	0
8	BF	Bus Free IRQ status. 0x0: No action. 0x1: Bus Free.	RW W1toClr	0
7	AERR	Access Error IRQ status. 0x0: No action. 0x1: Access Error.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
6	STC	Start Condition IRQ status. 0x0: No action. 0x1: Start Condition detected.	RW W1toClr	0
5	GC	General call IRQ status. Set to '1' by core when General call address detected and interrupt signaled to MPUSS. Write '1' to clear. 0x0: No general call detected. 0x1: General call address detected.	RW W1toClr	0
4	XRDY	Transmit data ready IRQ status. Set to '1' by core when transmitter and when new data is requested. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. 0x0: Transmission ongoing. 0x1: Transmit data ready.	RW W1toClr	0
3	RRDY	Receive data ready IRQ status. Set to '1' by core when receiver mode, a new data is able to be read. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. 0x0: No data available. 0x1: Receive data available.	RW W1toClr	0
2	ARDY	Register access ready IRQ status. When set to '1' it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to MPUSS. Write '1' to clear. 0x0: Module busy. 0x1: Access ready.	RW W1toClr	0
1	NACK	No acknowledgement IRQ status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPUSS. Write '1' to clear this bit. 0x0: Normal operation. 0x1: Not Acknowledge detected.	RW W1toClr	0
0	AL	Arbitration lost IRQ status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to MPUSS. During reads, it always returns 0. 0x0: Normal operation. 0x1: Arbitration lost detected.	RW W1toClr	0

**Table 23-64. Register Call Summary for Register I2C\_STAT**

## HS IC Environment

- [HS I2C Start and Stop Conditions: \[0\] \[1\]](#)

## HS IC Functional Description

- [HS I2C FIFO Interrupt Mode: \[2\] \[3\]](#)
- [HS I2C FIFO Polling Mode: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [HS I2C Draining Feature \(I2C Mode Only\): \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [HS I2C System Test Mode: \[15\] \[16\] \[17\]](#)

## HS IC Programming Guide

- [Main Program: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]](#)
- [Interrupt Subroutine Sequence: \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\]](#)
- [Programming Flow Diagrams: \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\]](#)
- [Main Program: \[42\] \[43\] \[44\] \[45\]](#)
- [Interrupt Subroutine Sequence: \[46\] \[47\] \[48\]](#)
- [Programming Flow Diagrams: \[49\] \[50\] \[51\] \[52\] \[53\] \[54\] \[55\] \[56\]](#)

**Table 23-64. Register Call Summary for Register I2C\_STAT (continued)**

HS IC Register Manual

- [HS I2C Register Summary: \[57\] \[58\]](#)
- [HS I2C Register Description: \[59\] \[60\] \[61\] \[62\] \[63\] \[64\] \[65\] \[66\] \[67\] \[68\] \[69\] \[70\] \[71\] \[72\] \[73\] \[74\] \[75\] \[76\] \[77\] \[78\] \[79\] \[80\] \[81\] \[82\] \[83\] \[84\] \[85\] \[86\] \[87\] \[88\] \[89\] \[90\] \[91\] \[92\] \[93\] \[94\]](#)

**Table 23-65. I2C\_SYSS**

<b>Address Offset</b>	0x0000 0090
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	System Status register
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															RDONE

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	RDONE	Reset done bit Read 0x1: Reset completed Read 0x0: Internal module reset in on-going	RW	1

**Table 23-66. Register Call Summary for Register I2C\_SYSS**

HS IC Functional Description

- [HS I2C Software Reset: \[0\] \[1\] \[2\]](#)

HS IC Register Manual

- [HS I2C Register Summary: \[3\] \[4\]](#)

**Table 23-67. I2C\_BUF**

<b>Address Offset</b>	0x0000 0094
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Buffer Configuration register
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDMA_EN	RXFIFO_CLR	RXTRSH						XDMA_EN	TXFIFO_CLR	TXTRSH					

Bits	Field Name	Description	Type	Reset
15	RDMA_EN	Receive DMA channel enable 0x0: Receive DMA channel disabled 0x1: Receive DMA channel enabled	RW	0
14	RXFIFO_CLR	Receive FIFO clear 0x0: Normal mode 0x1: Rx FIFO is reset	RW	0
13:8	RXTRSH	Threshold value for FIFO buffer in RX mode	RW	0x00

Bits	Field Name	Description	Type	Reset
7	XDMA_EN	Transmit DMA channel enable 0x0: Transmit DMA channel disabled 0x1: Transmit DMA channel enabled	RW	0
6	TXFIFO_CLR	Transmit FIFO clear 0x0: Normal mode 0x1: Tx FIFO is reset	RW	0
5:0	TXTRSH	Threshold value for FIFO buffer in TX mode	RW	0x00

**Table 23-68. Register Call Summary for Register I2C\_BUF**

## HS IC Functional Description

- [HS I2C FIFO Interrupt Mode: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [HS I2C FIFO Polling Mode: \[8\] \[9\]](#)
- [HS I2C FIFO DMA Mode \(I2C Mode Only\): \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [HS I2C Draining Feature \(I2C Mode Only\): \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\]](#)
- [HS I2C Write and Read Operations in SCCB Mode: \[23\] \[24\]](#)

## HS IC Programming Guide

- [Main Program: \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\]](#)
- [Programming Flow Diagrams: \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\]](#)
- [Main Program: \[41\] \[42\] \[43\] \[44\]](#)

## HS IC Register Manual

- [HS I2C Register Summary: \[45\] \[46\]](#)

**Table 23-69. I2C\_CNT**

<b>Address Offset</b>	0x0000 0098
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Data counter register
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCOUNT															

Bits	Field Name	Description	Type	Reset
15:0	DCOUNT	Data count	RW	0x0000

**Table 23-70. Register Call Summary for Register I2C\_CNT**

## HS IC Functional Description

- [HS I2C Draining Feature \(I2C Mode Only\): \[0\]](#)

## HS IC Programming Guide

- [Main Program: \[1\] \[2\]](#)
- [Programming Flow Diagrams: \[3\]](#)

## HS IC Register Manual

- [HS I2C Register Summary: \[4\] \[5\]](#)

**Table 23-71. I2C\_DATA**

<b>Address Offset</b>	0x0000 009C															
<b>Physical Address</b>	See <a href="#">Table 23-31</a>															
<b>Description</b>	Data access register															
<b>Type</b>	RW															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED								DATA							
<b>Bits</b>	<b>Field Name</b>		<b>Description</b>										<b>Type</b>	<b>Reset</b>		
15:8	RESERVED		Reserved										R	0x00		
7:0	DATA		Transmit/Receive data FIFO endpoint										RW	0x--		

**Table 23-72. Register Call Summary for Register I2C\_DATA**

HS IC Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">HS I2C DMA Requests: [0] [1] [2] [3] [4] [5] [6] [7]</a></li> <li>• <a href="#">HS I2C Write and Read Operations in SCCB Mode: [8] [9]</a></li> <li>• <a href="#">HS I2C System Test Mode: [10] [11]</a></li> </ul>
HS IC Programming Guide
<ul style="list-style-type: none"> <li>• <a href="#">Main Program: [12] [13] [14] [15]</a></li> <li>• <a href="#">Programming Flow Diagrams: [16] [17] [18] [19] [20] [21] [22] [23]</a></li> <li>• <a href="#">Main Program: [24] [25]</a></li> <li>• <a href="#">Programming Flow Diagrams: [26] [27] [28] [29]</a></li> </ul>
HS IC Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">HS I2C Register Summary: [30] [31]</a></li> </ul>

**Table 23-73. I2C\_CON**

<b>Address Offset</b>	0x0000 00A4															
<b>Physical Address</b>	See <a href="#">Table 23-31</a>															
<b>Description</b>	I2C configuration register.															
<b>Type</b>	RW															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I2C_EN	RESERVED	OPMODE	STB	MST	TRX	XSA	XOA0	XOA1	XOA2	XOA3	RESERVED	STP	STT		
<b>Bits</b>	<b>Field Name</b>		<b>Description</b>										<b>Type</b>	<b>Reset</b>		
15	I2C_EN		I2C module enable. 0x0: Controller in reset. FIFO are cleared and status bits are set to their default value 0x1: Module enabled										RW	0		
14	RESERVED		Reserved										R	0		
13:12	OPMODE		Operation mode selection. 0x0: I2C Fast/Standard mode. 0x1: I2C High Speed mode. 0x3: Reserved. 0x2: SCCB mode.										RW	0x0		

Bits	Field Name	Description	Type	Reset
11	STB	Start byte mode (master mode only). 0x0: Normal mode 0x1: Start byte mode	RW	0
10	MST	Master/slave mode. 0x0: Slave mode 0x1: Master mode	RW	0
9	TRX	Transmitter/Receiver mode (master mode only). 0x0: Receiver mode 0x1: Transmitter mode	RW	0
8	XSA	Expand Slave address. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
7	XOA0	Expand Own address 0. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
6	XOA1	Expand Own address 1. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
5	XOA2	Expand Own address 2. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
4	XOA3	Expand Own address 3. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
3:2	RESERVED	Reserved.	R	0x0
1	STP	Stop condition (master mode only). 0x0: No action or stop condition detected 0x1: Stop condition queried	RW	0
0	STT	Start condition (master mode only). 0x0: No action or start condition detected 0x1: Start condition queried	RW	0

**Table 23-74. Register Call Summary for Register I2C\_CON**

## HS IC Environment

- [HS I2C SCCB Interface Typical Connections: \[0\]](#)

## HS IC Functional Description

- [HS I2C Block Diagram: \[1\] \[2\]](#)
- [HS I2C Clocking: \[3\]](#)
- [HS I2C Software Reset: \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [HS I2C Programmable Multislave Channel Feature \(I2C Mode Only\): \[9\] \[10\] \[11\] \[12\]](#)
- [HS I2C FIFO Interrupt Mode: \[13\] \[14\]](#)
- [HS I2C Write and Read Operations in SCCB Mode: \[15\] \[16\] \[17\] \[18\]](#)
- [HS I2C System Test Mode: \[19\] \[20\] \[21\]](#)

## HS IC Programming Guide

- [Main Program: \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [Programming Flow Diagrams: \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\]](#)
- [Main Program: \[40\] \[41\] \[42\] \[43\] \[44\] \[45\]](#)
- [Programming Flow Diagrams: \[46\] \[47\] \[48\] \[49\] \[50\]](#)

## HS IC Register Manual

- [HS I2C Register Summary: \[51\] \[52\]](#)

**Table 23-75. I2C\_OA**

<b>Address Offset</b>	0x0000 00A8
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Own address register
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCODE				RESERVED				OA							

Bits	Field Name	Description	Type	Reset
15:13	MCODE	Master Code	RW	0x0
12:10	RESERVED	Reserved	R	0x0
9:0	OA	Own address	RW	0x000

**Table 23-76. Register Call Summary for Register I2C\_OA**

HS IC Functional Description

- [HS I2C Automatic Blocking of the I2C Clock Feature \(I2C Mode Only\): \[0\]](#)
- [HS I2C Programmable Multislave Channel Feature \(I2C Mode Only\): \[1\]](#)
- [HS I2C Write and Read Operations in SCCB Mode: \[2\] \[3\]](#)

HS IC Programming Guide

- [Main Program: \[4\]](#)
- [Programming Flow Diagrams: \[5\]](#)
- [Main Program: \[6\]](#)
- [Programming Flow Diagrams: \[7\]](#)

HS IC Register Manual

- [HS I2C Register Summary: \[8\] \[9\]](#)

**Table 23-77. I2C\_SA**

<b>Address Offset</b>	0x0000 00AC
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	Slave address register
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							SA								

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	SA	Slave address	RW	0x3FF

**Table 23-78. Register Call Summary for Register I2C\_SA**

HS IC Functional Description

- [HS I2C Write and Read Operations in SCCB Mode: \[0\] \[1\]](#)

HS IC Programming Guide

- [Main Program: \[2\]](#)
- [Programming Flow Diagrams: \[3\]](#)
- [Main Program: \[4\]](#)
- [Programming Flow Diagrams: \[5\]](#)

HS IC Register Manual

- [HS I2C Register Summary: \[6\] \[7\]](#)

**Table 23-79. I2C\_PSC**

<b>Address Offset</b>	0x0000 00B0															
<b>Physical Address</b>	See <a href="#">Table 23-31</a>															
<b>Description</b>	I2C Clock Prescaler Register															
<b>Type</b>	RW															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED								PSC							
<b>Bits</b>	<b>Field Name</b>		<b>Description</b>										<b>Type</b>	<b>Reset</b>		
15:8	RESERVED		Reserved										R	0x00		
7:0	PSC		Fast/Standard mode prescale sampling clock divider value 0x0: Divide by 1 0x1: Divide by 2 ..... 0xFF: Divide by 256										RW	0x00		

**Table 23-80. Register Call Summary for Register I2C\_PSC**

## HS IC Environment

- [HS I2C External Clock Configuration: \[0\] \[1\] \[2\]](#)

## HS IC Functional Description

- [HS I2C Clocking: \[3\] \[4\] \[5\]](#)
- [HS I2C Noise Filter: \[6\] \[7\]](#)
- [HS I2C System Test Mode: \[8\]](#)

## HS IC Programming Guide

- [Main Program: \[9\]](#)
- [Programming Flow Diagrams: \[10\]](#)
- [Main Program: \[11\] \[12\]](#)
- [Programming Flow Diagrams: \[13\]](#)

## HS IC Register Manual

- [HS I2C Register Summary: \[14\] \[15\]](#)

**Table 23-81. I2C\_SCLL**

<b>Address Offset</b>	0x0000 00B4															
<b>Physical Address</b>	See <a href="#">Table 23-31</a>															
<b>Description</b>	I2C SCL Low Time Register.															
<b>Type</b>	RW															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HSSCLL								SCLL							
<b>Bits</b>	<b>Field Name</b>		<b>Description</b>										<b>Type</b>	<b>Reset</b>		
15:8	HSSCLL		High Speed mode SCL low time										RW	0x00		
7:0	SCLL		Fast/Standard mode SCL low time.										RW	0x00		

**Table 23-82. Register Call Summary for Register I2C\_SCLL**

## HS IC Functional Description

- [HS I2C Clocking: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [HS I2C System Test Mode: \[9\]](#)



**Table 23-82. Register Call Summary for Register I2C\_SCLL (continued)**

- HS IC Programming Guide
- [Main Program: \[10\] \[11\]](#)
  - [Programming Flow Diagrams: \[12\]](#)
  - [Main Program: \[13\]](#)
  - [Programming Flow Diagrams: \[14\] \[15\] \[16\]](#)
- HS IC Register Manual
- [HS I2C Register Summary: \[17\] \[18\]](#)

**Table 23-83. I2C\_SCLH**

<b>Address Offset</b>	0x0000 00B8
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	I2C SCL High Time Register.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSSCLH								SCLH							

Bits	Field Name	Description	Type	Reset
15:8	HSSCLH	High Speed mode SCL high time	RW	0x00
7:0	SCLH	Fast/Standard mode SCL high time.	RW	0x00

**Table 23-84. Register Call Summary for Register I2C\_SCLH**

- HS IC Functional Description
- [HS I2C Clocking: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
  - [HS I2C System Test Mode: \[9\]](#)
- HS IC Programming Guide
- [Main Program: \[10\] \[11\]](#)
  - [Programming Flow Diagrams: \[12\]](#)
  - [Main Program: \[13\]](#)
  - [Programming Flow Diagrams: \[14\]](#)
- HS IC Register Manual
- [HS I2C Register Summary: \[15\] \[16\]](#)

**Table 23-85. I2C\_SYSTEST**

<b>Address Offset</b>	0x0000 00BC
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	I2C System Test Register.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_EN	FREE	TMODE	SSB	RESERVED	SCL_I_FUNC	SCL_O_FUNC	SDA_I_FUNC	SDA_O_FUNC	SCCB_E_O	SCL_I	SCL_O	SDA_I	SDA_O		

Bits	Field Name	Description	Type	Reset
15	ST_EN	System test enable. 0x0: Normal mode. All others bits in register are read only 0x1: System test enabled. Permit other system test registers bits to be set	RW	0
14	FREE	Free running mode (on breakpoint) 0x0: Stop mode (on breakpoint condition). If Master mode, it stops after completion of the on-going bit transfer. In slave mode, it stops during the phase transfer when 1 byte is completely transmitted/received. 0x1: Free running mode	RW	0
13:12	TMODE	Test mode select. 0x0: Functional mode (default) 0x1: Reserved 0x3: Loop back mode select + SDA/SCL IO mode select 0x2: Test of SCL counters (SCLL, SCLH, PSC). SCL provides a permanent clock with master mode.	RW	0x0
11	SSB	Set status bits from 0 to 14. 0x0: No action 0x1: Set interrupt status bits to 1	RW	0
10:9	RESERVED	Reserved	R	0x0
8	SCL_I_FUNC	SCL line input value (functional mode). Read 0x1: Read 1 from SCL line Read 0x0: Read 0 from SCL line	R	1
7	SCL_O_FUNC	SCL line output value (functional mode). Read 0x1: Driven 1 on SCL line Read 0x0: Driven 0 on SCL line	R	1
6	SDA_I_FUNC	SDA line input value (functional mode). Read 0x1: Read 1 from SDA line Read 0x0: Read 0 from SDA line	R	1
5	SDA_O_FUNC	SDA line output value (functional mode). Read 0x1: Driven 1 to SDA line Read 0x0: Driven 0 to SDA line	R	1
4	SCCB_E_O	SCCB_E line sense output value. 0x0: Write 0 to SCCBE line 0x1: Write 1 to SCCBE line	RW	0
3	SCL_I	SCL line sense input value Read 0x1: Read 1 from SCL line Read 0x0: Read 0 from SCL line	R	0
2	SCL_O	SCL line drive output value. 0x0: Write 0 to SCL line 0x1: Write 1 to SCL line	RW	0
1	SDA_I	SDA line sense input value. Read 0x1: Read 1 from SDA line Read 0x0: Read 0 from SDA line	R	0
0	SDA_O	SDA line drive output value. 0x0: Write 0 to SDA line 0x1: Write 1 to SDA line	RW	0

**Table 23-86. Register Call Summary for Register I2C\_SYSTEST**

HS IC Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">HS I2C System Test Mode: [0] [1] [2] [3] [4] [5] [6] [7] [8]</a></li> </ul>
HS IC Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">HS I2C Register Summary: [9] [10]</a></li> </ul>

**Table 23-87. I2C\_BUFSTAT**

<b>Address Offset</b>	0x0000 00C0
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	I2C Buffer Status Register.
<b>Type</b>	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFODEPTH		RXSTAT						RESERVED		TXSTAT					

Bits	Field Name	Description	Type	Reset
15:14	FIFODEPTH	Internal FIFO buffers depth.	R	0x3
13:8	RXSTAT	RX Buffer Status	R	0x00
7:6	RESERVED	Reserved	R	0x0
5:0	TXSTAT	TX Buffer Status.	R	0x00

**Table 23-88. Register Call Summary for Register I2C\_BUFSTAT**

HS IC Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">HS I2C FIFO Management: [0]</a></li> <li>• <a href="#">HS I2C Draining Feature (I2C Mode Only): [1] [2] [3] [4] [5] [6] [7] [8]</a></li> </ul>
HS IC Programming Guide
<ul style="list-style-type: none"> <li>• <a href="#">Programming Flow Diagrams: [9] [10] [11] [12] [13] [14] [15] [16]</a></li> </ul>
HS IC Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">HS I2C Register Summary: [17] [18]</a></li> </ul>

**Table 23-89. I2C\_OA1**

<b>Address Offset</b>	0x0000 00C4
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	I2C Own Address 1 Register
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OA1							

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	OA1	Own address 1	RW	0x000

**Table 23-90. Register Call Summary for Register I2C\_OA1**

HS IC Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">HS I2C Register Summary: [0] [1]</a></li> </ul>

**Table 23-91. I2C\_OA2**

<b>Address Offset</b>	0x0000 00C8															
<b>Physical Address</b>	See <a href="#">Table 23-31</a>															
<b>Description</b>	I2C Own Address 2 Register															
<b>Type</b>	RW															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED								OA2							
<b>Bits</b>	<b>Field Name</b>		<b>Description</b>										<b>Type</b>	<b>Reset</b>		
15:10	RESERVED		Reserved										R	0x00		
9:0	OA2		Own address 2										RW	0x000		

**Table 23-92. Register Call Summary for Register I2C\_OA2**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-93. I2C\_OA3**

<b>Address Offset</b>	0x0000 00CC															
<b>Physical Address</b>	See <a href="#">Table 23-31</a>															
<b>Description</b>	I2C Own Address 3 Register															
<b>Type</b>	RW															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED								OA3							
<b>Bits</b>	<b>Field Name</b>		<b>Description</b>										<b>Type</b>	<b>Reset</b>		
15:10	RESERVED		Reserved										R	0x00		
9:0	OA3		Own address 3										RW	0x000		

**Table 23-94. Register Call Summary for Register I2C\_OA3**

HS IC Register Manual

- [HS I2C Register Summary: \[0\] \[1\]](#)

**Table 23-95. I2C\_ACTOA**

<b>Address Offset</b>	0x0000 00D0															
<b>Physical Address</b>	See <a href="#">Table 23-31</a>															
<b>Description</b>	I2C Active Own Address Register.															
<b>Type</b>	R															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED												OA3_ACT	OA2_ACT	OA1_ACT	OA0_ACT

Bits	Field Name	Description	Type	Reset
15:4	RESERVED	Reserved	R	0x000
3	OA3_ACT	Own Address 3 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0
2	OA2_ACT	Own Address 2 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0
1	OA1_ACT	Own Address 1 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0
0	OA0_ACT	Own Address 0 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0

**Table 23-96. Register Call Summary for Register I2C\_ACTOA**

HS IC Functional Description

- [HS I2C Programmable Multislave Channel Feature \(I2C Mode Only\): \[0\]](#)

HS IC Register Manual

- [HS I2C Register Summary: \[1\] \[2\]](#)

**Table 23-97. I2C\_SBLOCK**

<b>Address Offset</b>	0x0000 00D4
<b>Physical Address</b>	See <a href="#">Table 23-31</a>
<b>Description</b>	I2C Clock Blocking Enable Register.
<b>Type</b>	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												OA3_EN	OA2_EN	OA1_EN	OA0_EN

Bits	Field Name	Description	Type	Reset
15:4	RESERVED	Reserved	R	0x000
3	OA3_EN	Enable I2C Clock Blocking for Own Address 3. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0
2	OA2_EN	Enable I2C Clock Blocking for Own Address 2. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0
1	OA1_EN	Enable I2C Clock Blocking for Own Address 1. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0
0	OA0_EN	Enable I2C Clock Blocking for Own Address 0. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0

**Table 23-98. Register Call Summary for Register I2C\_SBLOCK**

---

HS IC Functional Description

- [HS I2C Automatic Blocking of the I2C Clock Feature \(I2C Mode Only\): \[0\]](#)

---

HS IC Register Manual

- [HS I2C Register Summary: \[1\] \[2\]](#)
-

## 23.2 HDQ/1-Wire

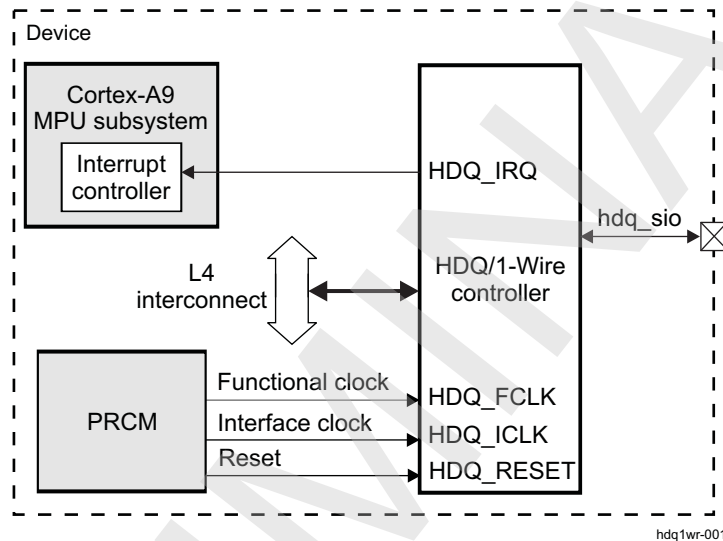
This section describes the HDQ/1-Wire interface for the device.

### 23.2.1 HDQ/1-Wire Overview

The HDQ/1-Wire module implements the hardware protocol of the master functions of the TI/Benchmark HDQ and the Dallas Semiconductor 1-Wire® protocols. These protocols use a single wire for communication between the master (HDQ/1-Wire controller) and the slaves (HDQ/1-Wire external compliant devices).

Figure 23-42 shows the HDQ/1-Wire.

Figure 23-42. HDQ/1-Wire Overview



The HDQ/1-Wire has a generic L4 interface and is intended to be used in an interrupt-driven fashion. The one-pin interface is implemented as an open-drain output at the device level.

The main features supported by the HDQ/1-Wire are the following:

- Benchmark HDQ protocol
- Dallas Semiconductor 1-Wire® protocol
- Power-down mode

The HDQ/1-Wire provides a communication rate of 5 Kbps over an address space of 128 bytes.

A typical application of the HDQ/1-Wire is the communication with battery monitor (gas gauge) integrated circuits.

## 23.2.2 HDQ/1-Wire Environment

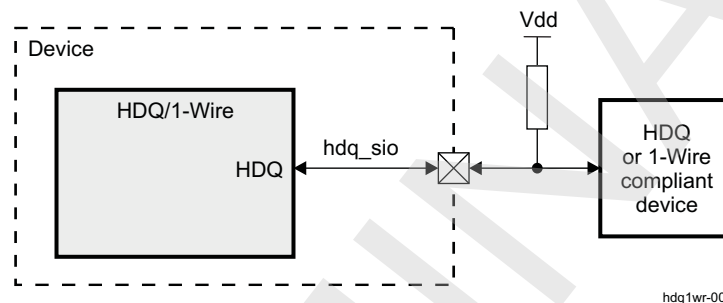
### 23.2.2.1 HDQ/1-Wire Functional Modes

The HDQ/1-Wire has two main modes: HDQ and 1-Wire. Each of these modes includes idle, active, and power-down submodes. [Table 23-99](#) lists the HDQ/1-Wire functional modes, and [Figure 23-43](#) shows an overview of a typical application.

**Table 23-99. Functional Modes**

Functions	Description
HDQ	Benchmark HDQ protocol
1-Wire	Dallas Semiconductor 1-Wire protocol

**Figure 23-43. HDQ/1-Wire Typical Application System Overview**



An external pullup is required, because the two protocols use a return-to-1 mechanism (that is, after any command by any of the connected devices, the line is pulled to a logical high level).

The HDQ/1-Wire operates according to a command structure that is programmed into transmit command registers (as described in [Section 23.2.5.1.2, HDQ/1-Wire Low-level Programming Model](#)).

The 1-Wire mode runs at slower speeds than the capabilities of the mode.

[Table 23-100](#) describes the external signal of the HDQ/1-Wire compliant module.

**Table 23-100. I/O Description**

Signal	I/O <sup>(1)</sup>	Description	Value at Reset
hdq_sio	I/O	Serial data input/output. Output is open drain type.	HiZ (pulled to 1 by pullup)

<sup>(1)</sup> I = Input; O = Output

### 23.2.2.2 HDQ and 1-Wire (SDQ) Protocols

#### 23.2.2.2.1 HDQ Protocol Initialization (Default)

In HDQ mode, the firmware does not require the host to create an initialization pulse to the slave. However, the slave can be reset by using an initialization pulse (also referred to as a break pulse). The initialization pulse is generated by setting the [HDQ\\_CTRL\\_STATUS\[2\]](#) INITIALIZATION bit and then setting the [HDQ\\_CTRL\\_STATUS\[4\]](#) GO bit. The slave does not respond with a presence pulse as it does in the 1-Wire protocol.

The HDQ is a command-based protocol in which the host sends a command byte to the slave. The command directs the slave either to store the next eight bits of data received to a register specified by the command byte (write operation) or to output the eight bits of data from a register specified by the command byte (read operation). The master implementation is a simple byte engine. Sending of the ID, command/address, and data is controlled by firmware. The master engine provides only a single HDQ.[HDQ\\_TX\\_DATA](#) register.

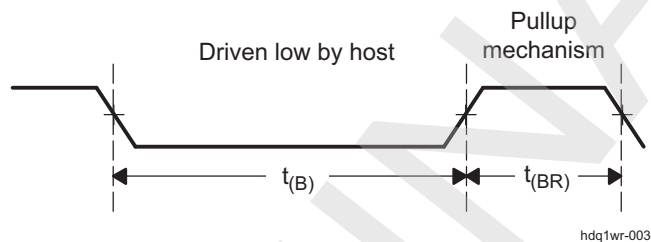


The command and data bytes consist of a stream of eight bits with a maximum transmission rate of 5 Kbps. The least-significant bit (LSB) of a command or data byte is transmitted first. If a communication time-out occurs between the host and the slave (for example, if the host waits longer than the specified time for the slave to respond, or if this is the first access command), then the host must send an initialization pulse (BREAK) before sending the command again.

The slave detects a break when the HDQ pin is driven to a logic-low state for a specified break time  $t_{(B)}$  or greater. The HDQ pin then returns to its normal ready-high logic state for a specified break-recovery time  $t_{(BR)}$ . The slave is then ready for a command from the host processor. Figure 23-44 shows this behavior.

An interrupt condition indicates a TX-complete, an RX-complete, or a time-out condition. Reading the interrupt status register clears all interrupt conditions. Only one interrupt signal is sent to the microcontroller, and only one overall mask bit can enable or disable the interrupt. The interrupt conditions cannot be individually masked.

Figure 23-44. HDQ Break-Pulse Timing Diagram

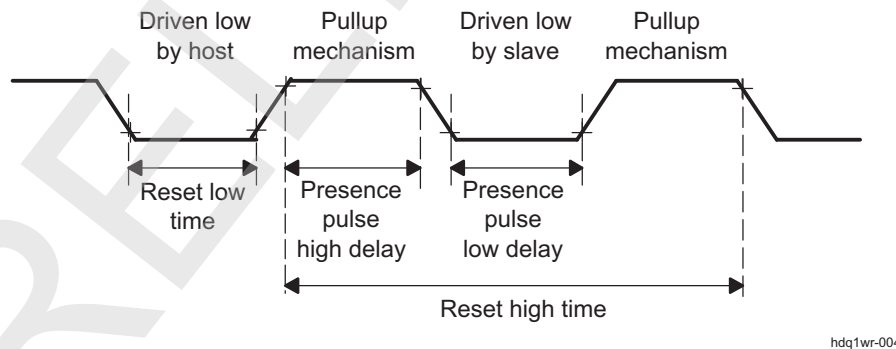


23.2.2.2.2 1-Wire (SDQ) Protocol Initialization

In 1-Wire (SDQ) protocol, the host first sends an initialization pulse (by pulling the line to a logic-low state) and then waits for the slave to respond with a presence pulse before enabling any communication sequence.

As for the initialization pulse, the presence pulse is a low-going edge on the line initiated by the slave. The timing diagram in Figure 23-45 shows the 1-Wire (SDQ) reset sequence.

Figure 23-45. 1-Wire (SDQ) Reset Timing Diagram



The host drives the line to a logic-low state for a minimum of reset low time. Once the slave detects this pulse, it must drive the line to a logic-low state within the presence pulse high delay for a minimum period of presence pulse low time.

If the slave does not respond within this interval of time, a time-out event occurs and no transaction can be initiated. The host must initiate the reset sequence again before sending any command to the slave.

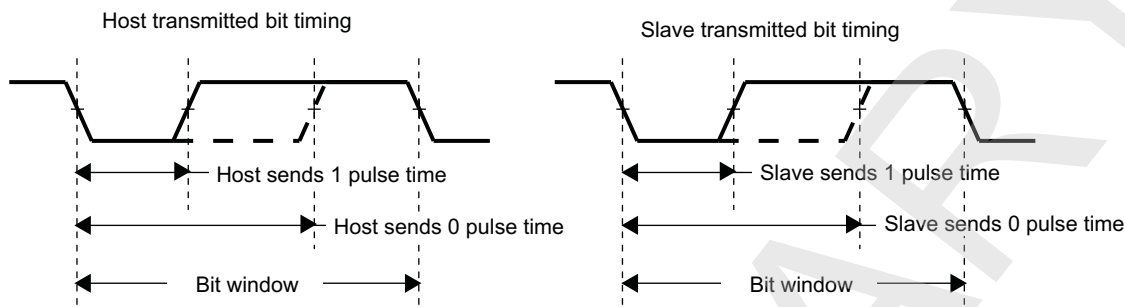
On the other hand, if the slave sends back its presence pulse within the specified time interval, the communication can be enabled after the reset high time.

23.2.2.2.3 Communication Sequence (HDQ and 1-Wire Protocols)

The description in this section applies to both protocols.

After a successful break pulse (HDQ mode) or initialization sequence (1-Wire protocol), the host and slave are ready for bit transmission. Each bit to transmit (either from the host to the slave or from the slave to the host) is preceded by a low-going edge on the line, as shown in [Figure 23-46](#).

**Figure 23-46. HDQ/1-Wire Transmitted Bit Timing**



hdq1wr-005

The return-to-1 data-bit frame consists of three distinct sections. The first section starts the transmission when either the slave or the host takes the line to a logic-low state. The next section is the actual data transmission in which the data must be valid during a specified period of time after the negative edge that starts the communication. The final section stops the transmission by returning the HDQ/1-Wire line to a logic-high state. Communication with an HDQ/1-Wire slave always occurs with the LSB being transmitted first.

The command byte of the HDQ/1-Wire protocols consists of eight contiguous valid command bits. The command byte contains two fields: R/W command and address. The R/W bit of the command byte determines whether the command is a read or a write, and the address field containing bits AD6-AD0 indicates the address to be read or written. [Table 23-101](#) lists the command byte values.

**Table 23-101. HDQ/1-Wire Command Byte**

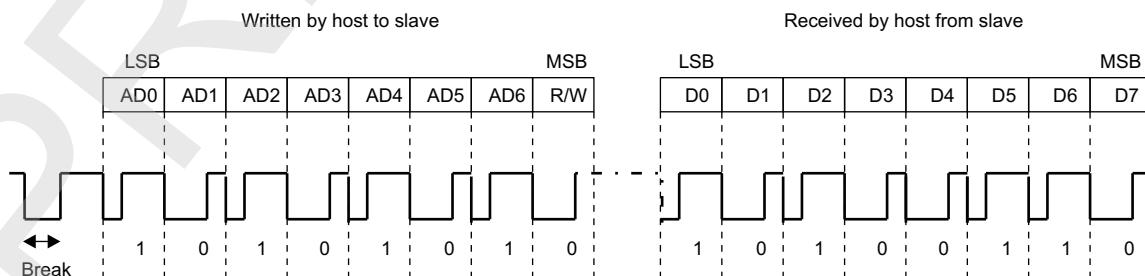
7	6	5	4	3	2	1	0
R/W	AD6	AD5	AD4	AD3	AD2	AD1	AD0

**R/W** Indicates whether the command byte is a read or a write. A 1 indicates a write command; the following eight bits must be written to the register specified by the address field of the command byte. A 0 indicates that the command is a read. On a read command, the slave outputs the requested register contents.

**AD6-AD0** Represent the seven bits labeled AD6-AD0 containing the address portion of the register to be accessed.

The communication sequence example in [Figure 23-47](#) shows a read command at address 0x55; the received data is 0x65.

**Figure 23-47. HDQ/1-Wire Communication Sequence**



hdq1wr-006

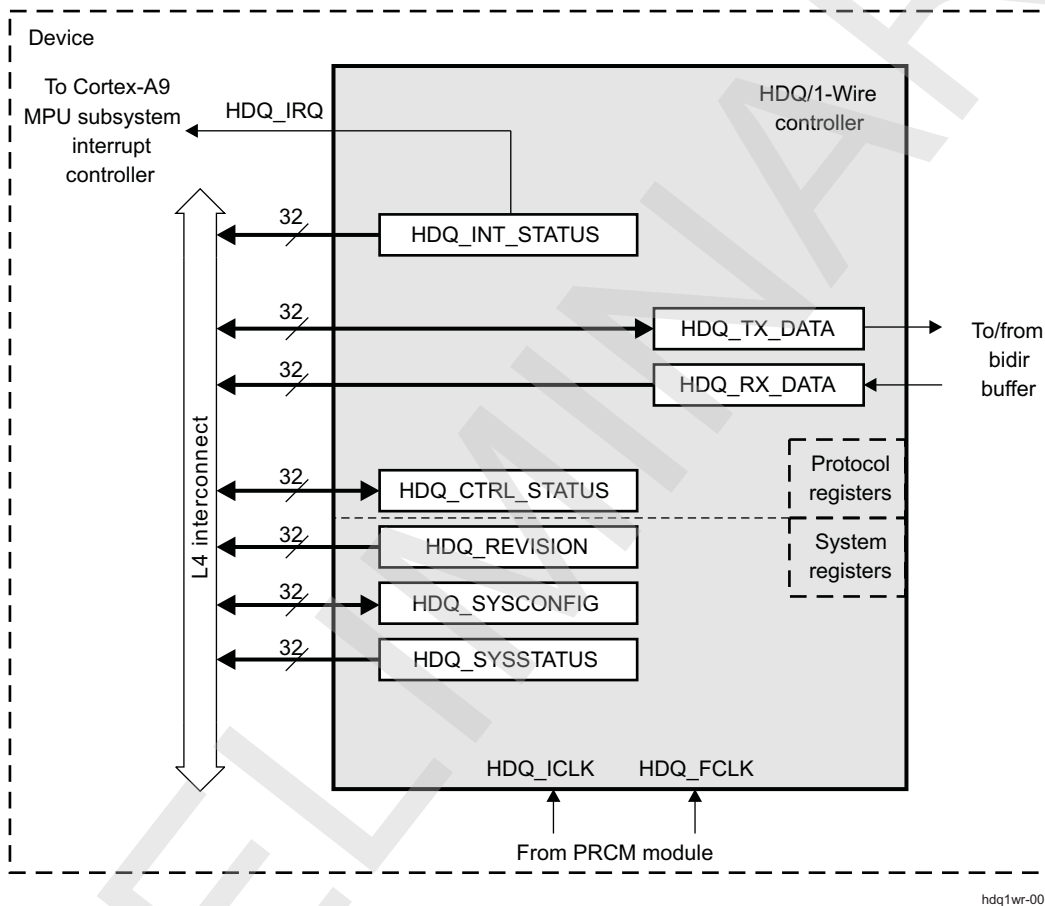
### 23.2.3 HDQ/1-Wire Functional Description

The HDQ/1-Wire works with HDQ and 1-Wire protocols. The protocols use a single wire to establish communication between the master and the slave. Both protocols use a return-to-1 mechanism; that is, after any command is driven, the line is pulled to a high level. This mechanism requires an external pullup.

#### 23.2.3.1 HDQ/1-Wire Block Diagram

Figure 23-48 is the HDQ/1-Wire block diagram.

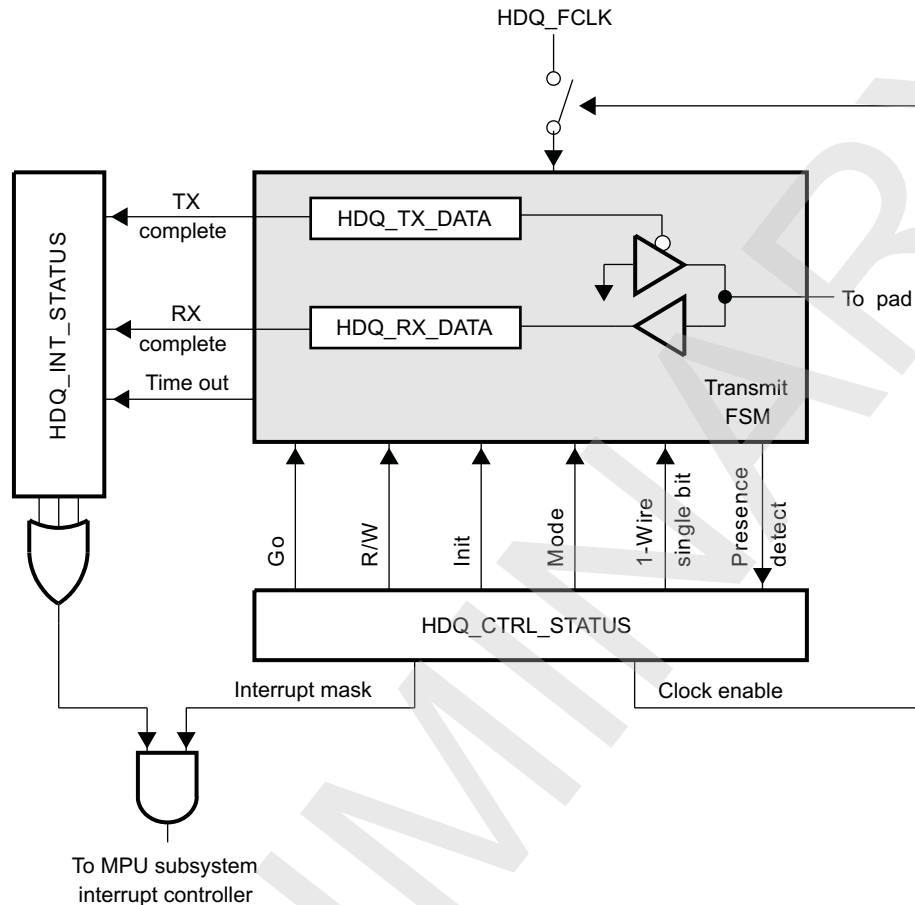
Figure 23-48. HDQ/1-Wire Block Diagram



The HDQ.HDQ\_CTRL\_STATUS[0] MODE bit allows selection between the HDQ and 1-Wire protocols. This bit is assumed static for design purposes. The configuration is in HDQ mode by default.

Figure 23-49 shows the protocol-dedicated register scheme.

Figure 23-49. Protocol Registers Description



hdq1wr-009

The receive and transmit operations of the HDQ/1-Wire module are performed with respect to the timing of the slower HDQ protocol. When the 1-Wire protocol is used, it runs at lower speed than its full capabilities, but is still able to meet the timing requirements and practical considerations.

### 23.2.3.2 HDQ/1-Wire Clocking Configuration

#### 23.2.3.2.1 HDQ/1-Wire Clocks

The HDQ/1-Wire subsystem operates from two clocks: a functional clock (HDQ\_FCLK) and an interface clock (HDQ\_ICLK).

- The HDQ\_FCLK functional clock is a fixed clock provided by the PRCM module. It is used to clock the internal module logic.

For more information about the clock, see [Section 3.6.8, CD\\_L4\\_PER Clock Domain](#).

When the HDQ/1-Wire no longer requires the HDQ\_FCLK, the software can disable it at the PRCM level. The clock is effectively cut, provided the other modules that receive it do not require it either.

For details about the PRCM register settings and clock configuration, see [Section 3.6.8, CD\\_L4\\_PER Clock Domain](#).

- The HDQ\_ICLK interface clock runs at L4 interconnect clock speed and is used to trigger access to the HDQ/1-Wire L4 interface.

When the HDQ/1-Wire no longer requires the HDQ\_ICLK (no transfer is in progress), the software can disable it at the PRCM level. The clock is effectively cut, provided the other modules that receive it do not require it either.

For more information, see [Section 3.6.8, CD\\_L4\\_PER Clock Domain](#).

### 23.2.3.3 HDQ/1-Wire Hardware and Software Reset

Global reset of the module is done at the power, reset, and clock management (PRCM) module level (for more information, see [Section 3.6.8, CD\\_L4\\_PER Clock Domain](#)) or by setting the HDQ.HDQ\_SYSCONFIG[1] SOFTRESET bit to 1. Setting this bit enables an active software reset functionality equivalent to a hardware reset. The HDQ\_FCLK functional clock must be enabled from the PRCM level and locally through the HDQ\_CTRL\_STATUS[5] CLOCKENABLE bit (set to 1) for the software reset to complete.

### 23.2.3.4 HDQ/1-Wire Power Management

[Table 23-102](#) describes power-management features available to the HDQ/1-Wire.

**Table 23-102. Local Power-Management Features**

Feature	Registers	Description
Clock auto gating	HDQ_SYSCONFIG [0] AUTOIDLE bit	Auto-idle mode
Slave idle modes	N/A	N/A
Clock enable	HDQ.HDQ_CTRL_STATUS[5] CLOCKENABLE bit	Power-down mode
Master standby modes	N/A	N/A
Global wake-up enable	N/A	N/A
Wake-up sources enable	N/A	N/A

#### 23.2.3.4.1 Auto-Idle Mode

The HDQ/1-Wire provides an auto-idle function in its interconnect clock domain.

The interconnect clock auto-idle power-saving mode is enabled or disabled through the HDQ.HDQ\_SYSCONFIG[0] AUTOIDLE bit. When this mode is enabled and there is no activity on the interconnect interface, the interconnect clock (HDQ\_ICLK) is disabled inside the module, thereby reducing power consumption. When there is new activity on the interconnect interface, the interconnect clock is restarted with no latency penalty. This mode is disabled by default after a reset.

The auto-idle mode can be enabled in order to reduce power consumption.

#### 23.2.3.4.2 Power-Down Mode

The HDQ/1-Wire also provides a power-saving function in its functional clock domain.

Setting the CLOCKENABLE bit in the control and status register (HDQ.HDQ\_CTRL\_STATUS[5] CLOCKENABLE bit) to 0 shuts off the functional clock (HDQ\_FCLK) to the state-machine. The state-machine is reset when the functional clock is disabled; if any transaction is ongoing, it is aborted into the reset state.

Before shutting off the functional clock, the software must wait for transaction-complete interrupt. In write operation the software must check whether the interrupt was generated after address/command byte was sent or after data byte was sent. The functional clock must not be shut off after address/command byte is sent; otherwise, the data is not written to the slave.

The register values are not affected by disabling the functional clock.

#### CAUTION

There is no hardware mechanism to prevent cutting off the HDQ/1-Wire clocks while the module is performing a transfer. This would result in loss of data being transferred.

### 23.2.3.5 HDQ Interrupt Requests

The HDQ/1-Wire module is intended to be used in an interrupt-driven way. If interrupts are masked, the [HDQ\\_CTRL\\_STATUS\[4\] GO](#) bit does not self-clear after the operation completes. Software then must clear the GO bit to avoid continuous execution.

The HDQ/1-Wire can generate one interrupt:

- [HDQ\\_IRQ](#): This is an interrupt to the MPU subsystem INTC. It is mapped on [MA\\_IRQ\\_58](#). [Table 23-103](#) lists the events that can generate this interrupt.

**Table 23-103. Events**

Event Flag	Event Mask	Sync	Sensitivity	Map to	Description
<a href="#">HDQ_INT_STATUS[2] TXCOMPLETE</a>	<a href="#">HDQ_CTRL_STATUS[31] INTERRUPTMASK</a>	Yes	Level	<a href="#">HDQ_IRQ</a>	A write operation of one byte was completed.
<a href="#">HDQ_INT_STATUS[1] RXCOMPLETE</a>	<a href="#">HDQ_CTRL_STATUS[31] INTERRUPTMASK</a>	Yes	Level	<a href="#">HDQ_IRQ</a>	A byte has been successfully read.
<a href="#">HDQ_INT_STATUS[0] TIMEOUT</a>	<a href="#">HDQ_CTRL_STATUS[31] INTERRUPTMASK</a>	Yes	Level	<a href="#">HDQ_IRQ</a>	After a read command initiated by the host, the slave did not pull the line low within the specified time.

### 23.2.3.6 HDQ Mode (Default)

#### 23.2.3.6.1 HDQ Mode Features

The HDQ mode supports the following:

- Benchmark HDQ protocol
- Power-down mode

#### 23.2.3.6.2 Description

In the HDQ mode, there is no need for the host to create an initialization pulse to the slave. However, the host can reset the slave by using an initialization pulse (also known as a break pulse). Setting the [HDQ.HDQ\\_CTRL\\_STATUS\[2\] INITIALIZATION](#) bit and then setting the [HDQ\\_CTRL\\_STATUS\[4\] GO](#) bit creates this pulse by pulling the line down for a defined duration. When the slave receives the pulse, it is ready for communication but does not respond with a presence pulse.

In a typical write operation, two bytes are sent to the slave. The first byte corresponds to the command/address byte, and the second byte corresponds to the data to be written.

In a typical read operation, the host sends a command/address byte and the slave returns a byte of data.

The master is implemented to send and receive bytes. Sending the command/address and data is controlled by the firmware. The master provides only a single data TX register.

The HDQ protocol is a return-to-1 protocol. Consequently, after a byte is sent to the slave (either command/address + data for a write, or just command/address for a read), the host pulls the line up. The line is set to the high-impedance state in the device and an external pullup brings it to a logical high level.

In the case of a read operation, the slave also drives the line to a logic-low state before sending the requested data.

If the host initiates a read and does not receive data within a specified interval of time (that is, the slave does not drive the line low within this interval), the [HDQ.HDQ\\_INT\\_STATUS\[0\] TIMEOUT](#) bit is set, thereby indicating a read failure. The TIMEOUT bit remains set until the host reads the interrupt status register ([HDQ.HDQ\\_INT\\_STATUS](#)).

An interrupt condition indicates either a TX-complete, an RX-complete, or a time-out on a transaction. The corresponding bit is set in the interrupt status register ([HDQ.HDQ\\_INT\\_STATUS](#)). This register is cleared as soon as it is read.

Only one interrupt signal is sent to the MPU, and only an overall mask can enable or disable the interrupts. These interrupts cannot be individually masked.

### 23.2.3.6.3 Single-Bit Mode

In HDQ mode, the single-bit mode (HDQ.HDQ\_CTRL\_STATUS[7] ONE\_WIRE\_SINGLE\_BIT bit set to 1) has no effect because the HDQ protocol supports only byte transfers.

### 23.2.3.6.4 Interrupt Conditions

The HDQ/1-Wire provides the following interrupt status:

- **Transmission complete:**  
A write operation of one byte was completed. Successful or failed completion is not indicated, because there is no acknowledgment from the slave in HDQ protocol. This interrupt condition is cleared by reading the interrupt status register (HDQ.HDQ\_INT\_STATUS).
- **Read complete:**  
In HDQ mode, the interrupt status indicates that a byte has been successfully read. This interrupt condition is cleared by reading the interrupt status register (HDQ.HDQ\_INT\_STATUS).
- **Presence detect/time-out:**  
In HDQ mode, the interrupt status indicates that after a read command initiated by the host, the slave did not pull the line low within the specified time. This interrupt condition is cleared by reading the interrupt status register (HDQ.HDQ\_INT\_STATUS).  
In HDQ mode, a time-out condition is also used to indicate the successful completion of a break pulse. That is, if the master has sent the break pulse, it is indicated with a time-out instead of a TX-complete.

Only one interrupt is generated to the MPU based on any of these interrupt conditions. A read operation on the interrupt status register clears all the interrupt status bits that were previously set.

### 23.2.3.7 1-Wire Mode

#### 23.2.3.7.1 1-Wire Mode Features

The 1-Wire mode supports the following:

- Dallas Semiconductor 1-Wire protocol
- Power-down mode
- Single-bit mode

#### 23.2.3.7.2 Description

The 1-Wire mode requires an initialization pulse to be sent to the slave(s) connected on the interface. If a slave is present, it responds with a presence pulse.

The initialization pulse is sent when the HDQ.HDQ\_CTRL\_STATUS[2] INITIALIZATION bit is set and the HDQ\_CTRL\_STATUS[4] GO bit is set afterwards.

When the slave receives the initialization pulse, it sends back its presence pulse by pulling down the line for a defined duration. The module detects this low-going edge and sets the HDQ.HDQ\_CTRL\_STATUS[3] PRESENCEDETECT bit.

In a similar way, if a presence pulse is not received from the slave after an initialization pulse is sent, the PRESENCEDETECT bit remains cleared.

Whether or not a presence pulse is detected after an initialization pulse is sent, the HDQ.HDQ\_INT\_STATUS[0] TIMEOUT bit is set and an interrupt condition is generated.

In 1-Wire mode, the generated interrupt condition means the maximum time allowed for receiving the response has elapsed and the software must check the PRESENCEDETECT bit to determine whether or not there was a presence pulse.

The INITIALIZATION bit is cleared at the end of the initialization pulse at the same time as the TIMEOUT bit is set. The TIMEOUT bit is cleared when the interrupt status register (HDQ.HDQ\_INT\_STATUS) is read.

For read operations, 1-Wire is a bit-by-bit protocol, which means the slave must be clocked by the host for each bit of the byte to read.

The line is pulled up at the end of the command/address byte. On the first read, the host creates a low-going edge to initiate a bit read. The line is then pulled up (pulled to the high-impedance state by the host and set to a high logical level by the external pullup) and the slave either drives the line low to transmit a 0, or does not drive the line to transmit a 1. This sequence is repeated for each bit to read.

The first bit the host receives is the LSB, and the last bit is the most-significant bit (MSB) in the receive data register (HDQ.HDQ\_RX\_DATA).

An interrupt condition indicates either a TX-complete, an RX-complete, or a time-out condition (that is, the time allowed for the slave to indicate its presence has elapsed). A read operation on the interrupt status register clears the interrupt conditions previously set. As in the HDQ mode, only one interrupt signal is sent to the MPU. Only an overall mask bit can enable or disable the interrupt (the interrupt conditions cannot be masked individually).

#### 23.2.3.7.3 1-Wire Single-Bit Mode Operation

A single-bit mode can be entered by setting the appropriate bit in the control and status register (ONE\_WIRE\_SINGLE\_BIT bit HDQ.HDQ\_CTRL\_STATUS[7]). In this mode, only one bit of data at a time is transferred between the master and the slave. After the bit is transferred, an interrupt is generated (that is, there is an RX-complete for a read operation and a TX-complete for a write operation). The ONE\_WIRE\_SINGLE\_BIT bit is cleared by hardware after every single bit is received. Software must set this bit to re-enable reception in single-bit mode. Bit 0 of the RX register (HDQ.HDQ\_RX\_DATA) is updated each time a bit is received from the slave; bit 0 of the TX register (HDQ.HDQ\_TX\_DATA) contains the bit to be sent.



### 23.2.3.7.4 Interrupt Conditions

The HDQ/1-Wire provides the following interrupt status:

- **Transmission complete:**  
A write operation of one byte was completed. Successful or failed completion is not indicated, because there is no acknowledgment from the slave in 1-Wire protocol. This interrupt condition is cleared by reading the interrupt status register (HDQ.HDQ\_INT\_STATUS).
- **Read complete:**  
In 1-Wire mode, the interrupt status indicates that a byte has been successfully read. This interrupt condition is cleared by reading the interrupt status register (HDQ.HDQ\_INT\_STATUS).
- **Presence detect/time-out:**  
In 1-Wire mode, the interrupt status indicates that it is now valid to check the PRESENCEDETECT bit. This interrupt condition is cleared by reading the interrupt status register (HDQ.HDQ\_INT\_STATUS).

Only one interrupt is generated to the MPU based on any of these interrupt conditions. A read operation on the interrupt status register clears all interrupt status bits that were previously set.

### 23.2.3.7.5 Status Flags

The presence-condition-detected status flag is contained in the HDQ.HDQ\_CTRL\_STATUS[3] PRESENCEDETECT bit. This is valid only in 1-Wire mode. The flag is updated when the HDQ.HDQ\_INT\_STATUS[0] TIMEOUT bit is set. Therefore, its correct value shows only after the interrupt is generated. The firmware must wait for the time-out condition; otherwise, the flag keeps its previous value and is undefined.

After the first successful presence detection, before a new 1-Wire initialization is performed, the user must execute a software reset on the HDQ/1-Wire module. This ensures a proper new initialization sequence in 1-Wire mode.

## 23.2.4 HDQ/1-Wire Integration

Figure 23-50 shows HDQ/1-Wire integration in the device.

Figure 23-50. HDQ/1-Wire Integration

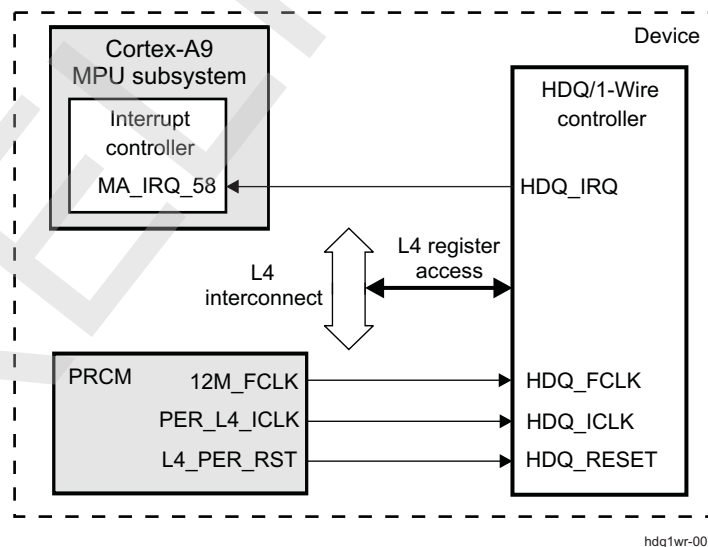


Table 23-104 through Table 23-106 summarize the integration of the module in the device.

**Table 23-104. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
HDQ/1-Wire	DPLL_PER	No	L4_PER

**Table 23-105. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HDQ/1-Wire	HDQ_ICLK	PER_L4_ICLK	PRCM	Interface clock
HDQ/1-Wire	HDQ_FCLK	12M_FCLK	PRCM	Functional clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HDQ/1-Wire	HDQ_RESET	L4_PER_RST	PRCM	HDQ/1-wire reset signal

**Table 23-106. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
HDQ/1-Wire	HDQ_IRQ	MA_IRQ_58	Cortex-A9	Interrupt to the Cortex A9 MPU subsystem interrupt controller (INTC).

**NOTE:** For the description of the interrupt source, see [Section 17.4.1, Cortex-A9 MPU INTC Functional Description](#), in [Chapter 17, Interrupt Controllers](#).

### 23.2.5 HDQ/1-Wire Programming Models

This section describes the low-level hardware programming sequences for configuration and usage of the module. The basic protocol functions, such as slave initialization (reset), read-byte, and write-byte operations, are described. For a description of the functions, see the HDQ/1-Wire protocol documentation.

#### 23.2.5.1 Global Initialization

##### 23.2.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the HDQ/1-Wire module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the HDQ/1-Wire. Refer to the HDQ/1-Wire Module Integration and Environment Sections for further information.

**Table 23-107. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. For more information about the module configuration, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Control Module	Module specific pad muxing and pullup must be set in the control module. For more information about the module configuration, see <a href="#">Chapter 18, Control Module</a> .
MPU INTC	MPU INTC configuration must be done to enable the interrupts from HDQ/1-Wire module. See <a href="#">Section 17.4.1, Cortex-A9 MPU INTC Functional Description</a> , in <a href="#">Chapter 17, Interrupt Controllers</a> .

##### 23.2.5.1.2 HDQ/1-Wire Module Global Initialization

**Table 23-108. HDQ/1-Wire Module Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Initiate software reset.	<a href="#">HDQ_SYSCONFIG[1]</a> SOFTRESET	0x1
Disable power-down mode.	<a href="#">HDQ_CTRL_STATUS[5]</a> CLOCKENABLE	0x1
Wait until reset complete?	<a href="#">HDQ_SYSSTATUS[0]</a> RESETDONE	= 0x1

**Table 23-108. HDQ/1-Wire Module Global Initialization (continued)**

Step	Register/Bit Field/Programming Model	Value
Disable power-down mode.	HDQ_CTRL_STATUS[5] CLOCKENABLE	0x1
Configure auto-idle mode.	HDQ_SYSCONFIG[0] AUTOIDLE	x

### 23.2.5.2 HDQ Operational Modes Configuration

#### 23.2.5.2.1 Main Sequence – HDQ Write Operation Mode

**Table 23-109. HDQ Mode Selection**

Step	Register/Bit Field/Programming Model	Value
Select HDQ mode.	HDQ_CTRL_STATUS[0] MODE	0x0
Enable interrupt generation.	HDQ_CTRL_STATUS[6] INTERRUPTMASK	0x1
Initialize HDQ slave.	See Section 23.2.5.2.2.1	

**Table 23-110. HDQ Write Operation Mode**

Step	Register/Bit Field/Programming Model	Value
Write command/address or data value.	HDQ_TX_DATA[7:0]	0x–
Select write operation.	HDQ_CTRL_STATUS[1] DIR	0x0
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
Reading HDQ_INT_STATUS clears interrupt conditions.	HDQ_INT_STATUS[2] TXCOMPLETE	0x1

#### 23.2.5.2.2 Main Sequence - HDQ Read Operation Mode

##### 23.2.5.2.2.1 Sub-sequence - Initialize HDQ Slave

**Table 23-111. HDQ Read Operation Mode**

Step	Register/Bit Field/ Programming Model	Value
Select read operation.	HDQ_CTRL_STATUS[1] DIR	0x1
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
Read and store HDQ_INT_STATUS. Reading HDQ_INT_STATUS clears interrupt conditions.	HDQ_INT_STATUS	0x-
IF: Read operation successful?	HDQ_INT_STATUS[1] RXCOMPLETE	= 0x1
	HDQ_INT_STATUS[0] TIMEOUT	= 0x0
Get received data.	HDQ_RX_DATA[7:0]	0x-
ENDIF		

**Table 23-112. Initialize HDQ Slave**

Step	Register/Bit Field/Programming Model	Value
Send Initialization Pulse	HDQ_CTRL_STATUS[2] INITIALIZATION	0x1
Send Command	HDQ_CTRL_STATUS[4] GO	0x1

### 23.2.5.3 1-Wire Operational Modes Configuration

#### 23.2.5.3.1 Main Sequence - 1-Wire Write Operation Mode

**Table 23-113. 1-Wire Mode Selection**

Step	Register/Bit Field/Programming Model	Value
Reset HDQ/1-Wire module.	See <a href="#">Section 23.2.5.1.2</a> .	
Select 1-Wire mode.	HDQ_CTRL_STATUS[0] MODE	0x1
Enable interrupt generation.	HDQ_CTRL_STATUS[6] INTERRUPTMASK	0x1
Initialize 1-Wire slave, check for slave presence.	See <a href="#">Section 23.2.5.3.3</a> .	

**Table 23-114. 1-Wire Write Operation Mode**

Step	Register/Bit Field/ Programming Model	Value
Write ID/command or data value.	HDQ_TX_DATA[7:0]	0x–
Select write operation.	HDQ_CTRL_STATUS[1] DIR	0x0
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
Reading <a href="#">HDQ_INT_STATUS</a> clears interrupt conditions.	HDQ_INT_STATUS[2] TXCOMPLETE	0x1

### 23.2.5.3.2 Main Sequence - 1-Wire Read Operation Mode

**Table 23-115. 1-Wire Read Operation Mode**

Step	Register/Bit Field/Programming Model	Value
Select read operation.	HDQ_CTRL_STATUS[1] DIR	0x1
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
Read and store <a href="#">HDQ_INT_STATUS</a> . Reading <a href="#">HDQ_INT_STATUS</a> clears interrupt conditions.	HDQ_INT_STATUS	0x–
IF: Read operation successful?	HDQ_INT_STATUS[1] RXCOMPLETE	= 0x1
Get received data.	HDQ_RX_DATA[7:0]	0x–
ENDIF		

### 23.2.5.3.3 Sub-sequence - Initialize 1-Wire Slave

**Table 23-116. Initialize 1-Wire Slave**

Step	Register/Bit Field/Programming Model	Value
Select sending initialization pulse operation.	HDQ_CTRL_STATUS[2] INITIALIZATION	0x1
Start operation.	HDQ_CTRL_STATUS[4] GO	0x1
Wait for interrupt.		
IF: Presence pulse detected?	HDQ_INT_STATUS[0] TIMEOUT	= 0x1
	HDQ_CTRL_STATUS[3] PRESENCEDETECT	= 0x1
Slave is present and initialized.		
ELSE		
Repeat initialization subsequence.		
ENDIF		

## 23.2.6 HDQ/1-Wire Register Manual

### 23.2.6.1 HDQ/1-Wire Instance Summary

**Table 23-117. HDQ/1-Wire Instance Summary**

Module Name	Base Address	Size
HDQ/1-Wire	0x480B 2000	4KB

**23.2.6.2 HDQ/1-Wire Registers****CAUTION**

The following rules must be observed when accessing the module registers:

- A read from the [HDQ\\_INT\\_STATUS](#) register or the [HDQ\\_RX\\_DATA](#) register is not allowed unless the processor has been interrupted by the module.
- After the release of the GO bit in the [HDQ\\_CTRL\\_STATUS](#) register, no access to the [HDQ\\_TX\\_DATA](#) or [HDQ\\_CTRL\\_STATUS](#) register is allowed until the processor has been interrupted by the module.
- Polling of the [HDQ\\_INT\\_STATUS](#) register by software to determine whether an interrupt was generated is not allowed.

**CAUTION**

The HDQ/1-Wire registers are limited to 32-bit data accesses; 16-bit and 8-bit data accesses are not allowed and can corrupt register content.

**23.2.6.2.1 HDQ/1-Wire Register Summary****Table 23-118. HDQ/1-Wire Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	HDQ/1-Wire L4 Base Address
<a href="#">HDQ_REVISION</a>	R	32	0x0000 0000	0x480B 2000
<a href="#">HDQ_TX_DATA</a>	RW	32	0x0000 0004	0x480B 2004
<a href="#">HDQ_RX_DATA</a>	R	32	0x0000 0008	0x480B 2008
<a href="#">HDQ_CTRL_STATUS</a>	RW	32	0x0000 000C	0x480B 200C
<a href="#">HDQ_INT_STATUS</a>	R	32	0x0000 0010	0x480B 2010
<a href="#">HDQ_SYSCONFIG</a>	RW	32	0x0000 0014	0x480B 2014
<a href="#">HDQ_SYSSTATUS</a>	R	32	0x0000 0018	0x480B 2018

**23.2.6.2.2 HDQ/1-Wire Register Description****Table 23-119. HDQ\_REVISION**

<b>Address Offset</b>	0x0000 0000																																																																																														
<b>Physical Address</b>	See <a href="#">Table 23-118</a>																<b>Instance</b>																HDQ/1-Wire																																																														
<b>Description</b>	This register contains the IP revision code																																																																																														
<b>Type</b>	R																																																																																														
<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 2.5%;">31</th><th style="width: 2.5%;">30</th><th style="width: 2.5%;">29</th><th style="width: 2.5%;">28</th><th style="width: 2.5%;">27</th><th style="width: 2.5%;">26</th><th style="width: 2.5%;">25</th><th style="width: 2.5%;">24</th><th style="width: 2.5%;">23</th><th style="width: 2.5%;">22</th><th style="width: 2.5%;">21</th><th style="width: 2.5%;">20</th><th style="width: 2.5%;">19</th><th style="width: 2.5%;">18</th><th style="width: 2.5%;">17</th><th style="width: 2.5%;">16</th><th style="width: 2.5%;">15</th><th style="width: 2.5%;">14</th><th style="width: 2.5%;">13</th><th style="width: 2.5%;">12</th><th style="width: 2.5%;">11</th><th style="width: 2.5%;">10</th><th style="width: 2.5%;">9</th><th style="width: 2.5%;">8</th><th style="width: 2.5%;">7</th><th style="width: 2.5%;">6</th><th style="width: 2.5%;">5</th><th style="width: 2.5%;">4</th><th style="width: 2.5%;">3</th><th style="width: 2.5%;">2</th><th style="width: 2.5%;">1</th><th style="width: 2.5%;">0</th> </tr> </thead> <tbody> <tr> <td colspan="16" style="text-align: center;">RESERVED</td> <td colspan="16" style="text-align: center;">REV</td> </tr> </tbody> </table>																																31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																REV															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																
RESERVED																REV																																																																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

**Table 23-120. Register Call Summary for Register HDQ\_REVISION**

HDQ/1-Wire

- [HDQ/1-Wire Registers: \[0\]](#)

**Table 23-121. HDQ\_TX\_DATA**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 23-118</a>	<b>Instance</b>	HDQ/1-Wire
<b>Description</b>	This register contains the data to be transmitted.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TX_DATA															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reads returns 0	R	0x000000
7:0	TX_DATA	Transmit data (used in both HDQ and 1-Wire modes)	RW	0x00

**Table 23-122. Register Call Summary for Register HDQ\_TX\_DATA**

HDQ/1-Wire

- [HDQ and 1-Wire \(SDQ\) Protocols: \[0\]](#)
- [1-Wire Mode: \[1\]](#)
- [HDQ Operational Modes Configuration: \[2\]](#)
- [1-Wire Operational Modes Configuration: \[3\]](#)
- [HDQ/1-Wire Registers: \[4\] \[5\]](#)

**Table 23-123. HDQ\_RX\_DATA**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 23-118</a>	<b>Instance</b>	HDQ/1-Wire
<b>Description</b>	This register contains the data to be received.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_DATA															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reads returns 0	R	0x000000
7:0	RX_DATA	Receive data (used in both HDQ and 1-Wire modes)	R	0x00

**Table 23-124. Register Call Summary for Register HDQ\_RX\_DATA**

HDQ/1-Wire

- [1-Wire Mode: \[0\] \[1\]](#)
- [HDQ Operational Modes Configuration: \[2\]](#)
- [1-Wire Operational Modes Configuration: \[3\]](#)
- [HDQ/1-Wire Registers: \[4\] \[5\]](#)

**Table 23-125. HDQ\_CTRL\_STATUS**

<b>Address Offset</b>	0x0000 000C		
<b>Physical Address</b>	See <a href="#">Table 23-118</a>	<b>Instance</b>	HDQ/1-Wire
<b>Description</b>	This register provides status information about the module.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ONE_WIRE_SINGLE_BIT	INTERRUPTMASK	CLOCKENABLE	GO	PRESENCEDETECT	INITIALIZATION	DIR	MODE								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reads returns 0	RW	0x0000000
7	ONE_WIRE_SINGLE_BIT	Single-bit mode for 1-Wire 0x0: Disabled 0x1: Enabled	RW	0
6	INTERRUPTMASK	Interrupt masking bit 0x0: Interrupts disable 0x1: Interrupts enable	RW	0
5	CLOCKENABLE	Power-down mode bit 0x0: Clock disable (power down) 0x1: Clock enable	RW	0
4	GO	Go bit. Write 1 to start the appropriate operation. Bit returns to 0 after the operation is complete if INTERRUPTMASK = 0x1.	RW	0
3	PRESENCEDETECT	Slave presence indicator. Actual only just after initialization time-out. Used in 1-Wire mode. Read-only flag. 0x0: No slave detected 0x1: Slave detected	R	0
2	INITIALIZATION	Write 1 to send initialization pulse. Bit returns to 0 after pulse is sent.	RW	0
1	DIR	DIR bit, determines if next command is read or write 0x0: Write 0x1: Read	RW	0
0	MODE	Mode selection bit 0x0: HDQ mode 0x1: 1-Wire mode	RW	0



**Table 23-126. Register Call Summary for Register HDQ\_CTRL\_STATUS**

HDQ/1-Wire

- HDQ and 1-Wire (SDQ) Protocols: [0] [1]
- HDQ/1-Wire Block Diagram: [2]
- HDQ/1-Wire Hardware and Software Reset: [3]
- HDQ/1-Wire Power Management: [4] [5]
- HDQ Interrupt Requests: [6] [7] [8] [9]
- HDQ Mode (Default): [10] [11] [12]
- 1-Wire Mode: [13] [14] [15] [16] [17]
- Global Initialization: [18] [19]
- HDQ Operational Modes Configuration: [20] [21] [22] [23] [24] [25] [26] [27]
- 1-Wire Operational Modes Configuration: [28] [29] [30] [31] [32] [33] [34] [35] [36]
- HDQ/1-Wire Registers: [37] [38] [39]

**Table 23-127. HDQ\_INT\_STATUS**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	HDQ/1-Wire
<b>Physical Address</b>	See Table 23-118		
<b>Description</b>	This register controls interrupts status		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TXCOMPLETE		RXCOMPLETE		TIMEOUT											

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads returns 0	R	0x0000 0000
2	TXCOMPLETE	TX-complete interrupt flag. Set to 1 if cause of interrupt. Set to 0 when register read.	R	0
1	RXCOMPLETE	Read-complete interrupt flag. Set to 1 if cause of interrupt. Set to 0 when register read.	R	0
0	TIMEOUT	Presence detect/timeout interrupt flag. In 1-Wire mode, set to 1 if slave's presence detected. In HDQ mode, set to 1 if timeout on read occurs. Set to 0 when register read.	R	0

**Table 23-128. Register Call Summary for Register HDQ\_INT\_STATUS**

HDQ/1-Wire

- HDQ Interrupt Requests: [0] [1] [2]
- HDQ Mode (Default): [3] [4] [5] [6] [7] [8]
- 1-Wire Mode: [9] [10] [11] [12] [13] [14]
- HDQ Operational Modes Configuration: [15] [16] [17] [18] [19] [20] [21]
- 1-Wire Operational Modes Configuration: [22] [23] [24] [25] [26] [27] [28]
- HDQ/1-Wire Registers: [29] [30] [31]

**Table 23-129. HDQ\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 23-118</a>	<b>Instance</b>	HDQ/1-Wire
<b>Description</b>	This register controls various bits		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											SOFTRESET	AUTOIDLE			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reads returns 0	RW	0x0000 0000
1	SOFTRESET	Start soft reset sequence. 0x0: Disabled 0x1: Enabled	RW	0
0	AUTOIDLE	Interconnect idle. 0x0: Module clock is free-running. 0x1: Module is in power saving mode: Clock is running only when module is accessed or inside logic is in function to process events.	RW	0

**Table 23-130. Register Call Summary for Register HDQ\_SYSCONFIG**

HDQ/1-Wire

- [HDQ/1-Wire Hardware and Software Reset: \[0\]](#)
- [HDQ/1-Wire Power Management: \[1\] \[2\]](#)
- [Global Initialization: \[3\] \[4\]](#)
- [HDQ/1-Wire Registers: \[5\]](#)

**Table 23-131. HDQ\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0018		
<b>Physical Address</b>	See <a href="#">Table 23-118</a>	<b>Instance</b>	HDQ/1-Wire
<b>Description</b>	This register monitors the reset sequence.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											RESETDONE				

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	R	0x0000 0000
0	RESETDONE	Reset monitoring. 0x0: The module is currently performing its reset. When the module is in power-down mode, set to 0 to indicate this fact. 0x1: The module has finished its reset.	R	1

**Table 23-132. Register Call Summary for Register HDQ\_SYSSTATUS**

HDQ/1-Wire

- Global Initialization: [0]
- HDQ/1-Wire Registers: [1]

### 23.3 UART/IrDA/CIR

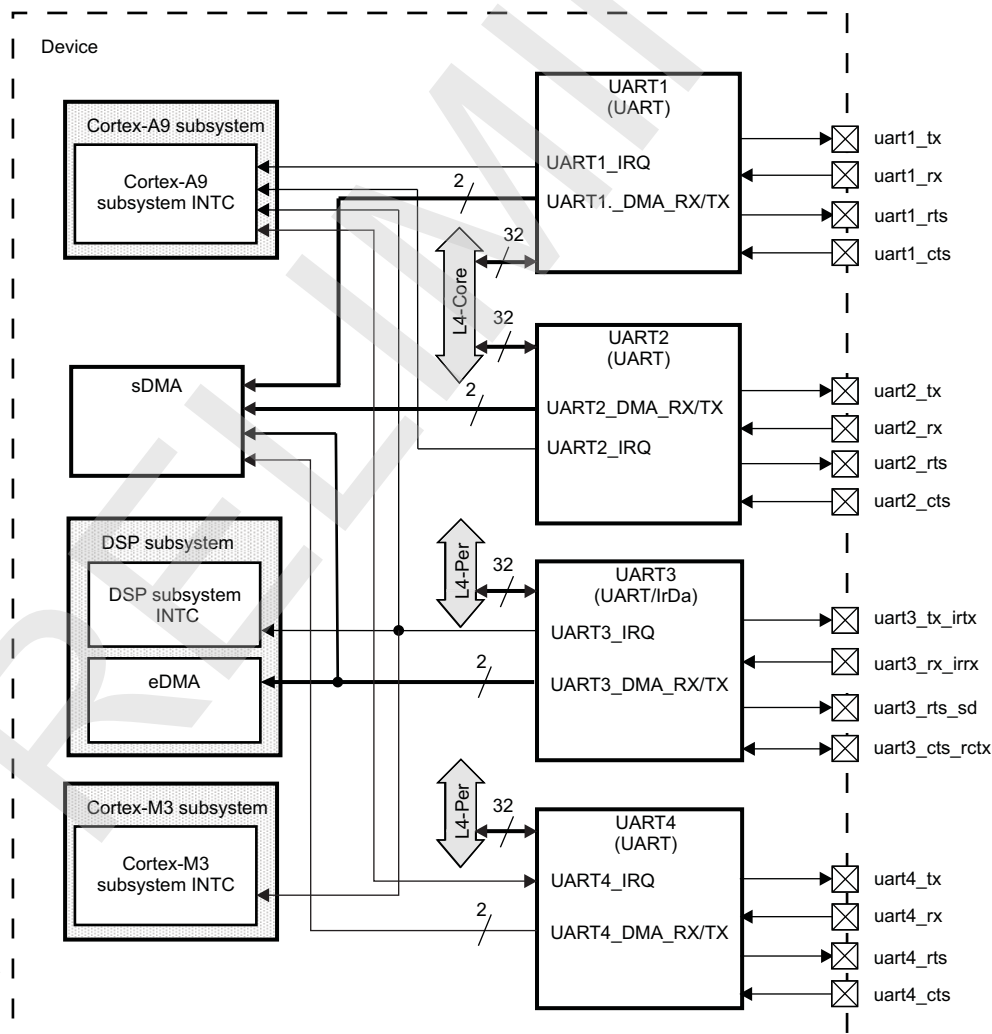
This chapter describes the function, operation, and configuration of the universal asynchronous receiver/transmitter (UART)/infrared data association (IrDA)/consumer infrared (CIR) module in the device.

#### 23.3.1 UART/IrDA/CIR Overview

The device contains four universal asynchronous receiver/transmitter (UART) devices controlled by the microprocessor unit (MPU) (see [Figure 23-51](#)):

- Three UART-only modules, UART1, UART2, and UART4, are pinned out for use as UART devices only.
- UART3, which adds infrared communication support, is pinned out for use as a UART, infrared data association (IrDA), or consumer infrared (CIR) device, and can be programmed to any available operating mode.

**Figure 23-51. UART Module**



uart-001

### 23.3.1.1 UART Features

The UARTs (UART1, UART2, UART3 when in UART mode, and UART4) include the following key features:

- 16C750 compatibility
- 64-byte first in first out (FIFO) buffer for receiver and 64-byte FIFO for transmitter
- Programmable interrupt trigger levels for FIFOs
- Baud generation based on programmable divisors N (N = 1...16,384) operating from a fixed functional clock of 48 MHz

Oversampling is programmed by software as 16 or 13; thus, the baud rate computation is one of two options:

- Baud rate = (functional clock/16)/N
- Baud rate = (functional clock/13)/N

This software programming mode enables higher baud rates with the same error amount without changing the clock source:

- Break character detection and generation
- Configurable data format:
  - Data bit: 5, 6, 7, or 8 bits
  - Parity bit: Even, odd, none
  - Stop-bit: 1, 1.5, 2 bit(s)
- Flow control: Hardware (RTS/CTS) or software (XON/XOFF)

The UART clocks are connected to produce a baud rate of up to 3.6 mbps. [Table 23-133](#) lists the supported baud rates, the requested divisor, and the corresponding error versus the standard baud rate.

**Table 23-133. UART Mode Baud Rates, Divisor Values, and Error Rates**

Baud Rate	Oversampling	Divisor	Error (%)
300	16	10,000	0
600	16	5000	0
1200	16	2500	0
2400	16	1250	0
4800	16	625	0
9600	16	312	0.16
14,400	16	208	0.16
19,200	16	156	0.16
28,800	16	704	0.16
38,400	16	78	0.16
57,600	16	52	0.16
115,200	16	26	0.16
230,400	16	13	0.16
460,800	13	8	0.16
921,600	13	4	0.16
1,843,200	13	2	0.16
3,000,000	16	1	0
3,686,400	13	1	0.16

### 23.3.1.2 IrDA Features

The IrDA (UART3 only) includes the following key features:

- Support of IrDA 1.4 slow infrared (SIR), medium infrared (MIR), and fast infrared (FIR) communications:
  - Frame formatting: Addition of variable beginning-of-frame (xBOF) characters and end-of-frame

- (EOF) characters
- Uplink/downlink cyclic redundancy check (CRC) generation/detection
  - Asynchronous transparency (automatic insertion of break character)
  - Eight-entry status FIFO (with selectable trigger levels) to monitor frame length and frame errors
  - Framing error, CRC error, illegal symbol (FIR), and abort pattern (SIR, MIR) detection

Table 23-134 lists the supported baud rates, the requested divisor, and the corresponding error versus the standard baud rate.

**Table 23-134. UART IrDA Mode Baud Rates, Divisor Values, and Error Rates**

Baud Rate	IR Mode	Encoding	Divisor	Error (%)
2400	SIR	3/16	1250	0
9600	SIR	3/16	312	0.16
19,200	SIR	3/16	156	0.16
38,400	SIR	3/16	78	0.16
57,600	SIR	3/16	52	0.16
115,200	SIR	3/16	26	0.16
576,000	MIR	1/4	2	0
1,152,000	MIR	1/4	1	0
4,000,000	FIR	4 PPM <sup>(1)</sup>	1	0

<sup>(1)</sup> PPM = pulse-position-modulation

### 23.3.1.3 CIR Features

CIR mode uses a variable pulse-width modulation (PWM) technique (based on multiples of a programmable  $t$  period) to encompass the various formats of infrared encoding for remote-control applications. The CIR logic transmits data packets based on a user-definable frame structure and packet content.

The CIR (UART3 only) includes the following key features to provide CIR support for remote-control applications:

- Transmit mode only (receive mode is not supported)
- Free data format (supports any remote-control private standards)
- Selectable bit rate
- Configurable carrier frequency
- 1/2, 5/12, 1/3, or 1/4 carrier duty cycle

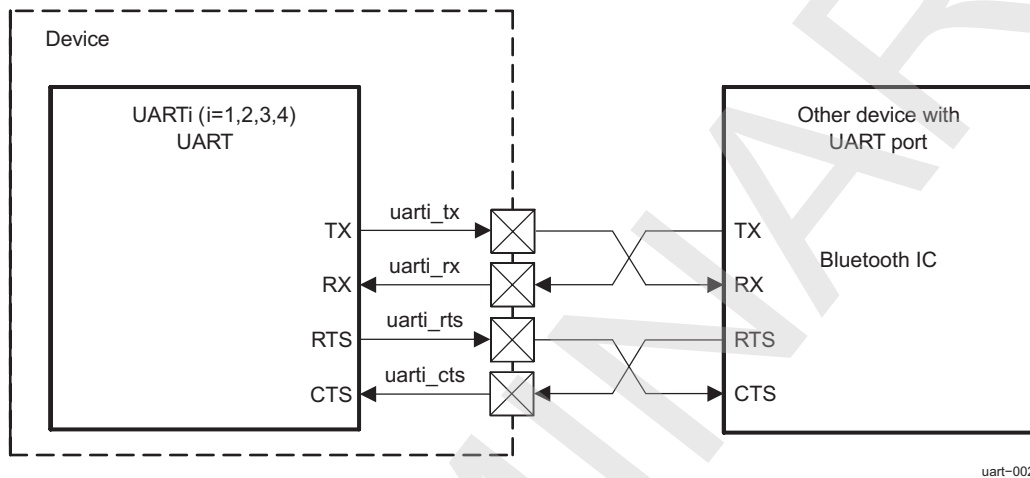
### 23.3.2 UART/IrDA/CIR Environment

This section describes the UART/IrDA/CIR connection with an external device.

#### 23.3.2.1 System Using UART Communication With Hardware Handshake

Each UART instance can be connected easily to the UART port of an external integrated circuit (IC) (see [Figure 23-52](#)).

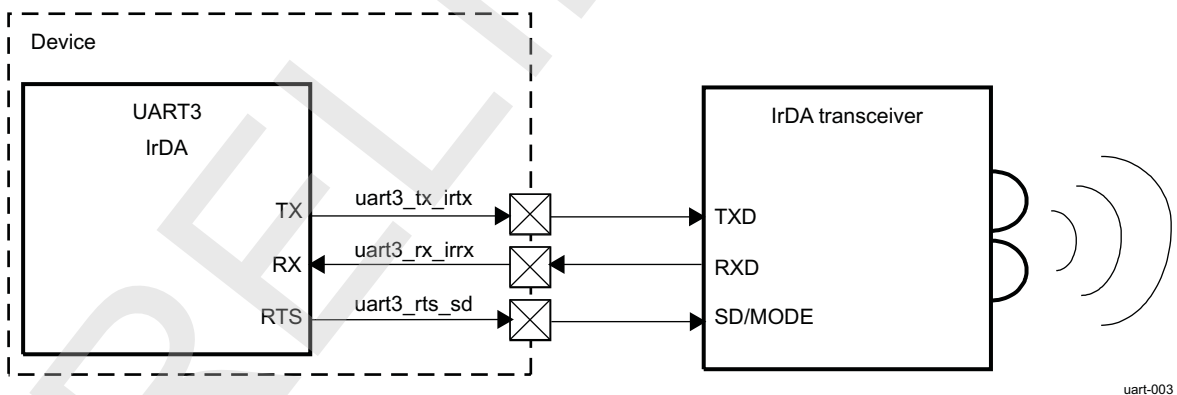
**Figure 23-52. UART Mode Bus System Overview**



#### 23.3.2.2 System Using IrDA Communication Protocol

As [Figure 23-53](#) shows, UART3 can be connected to an external infrared transceiver in the IrDA modes (FIR, SIR, and MIR).

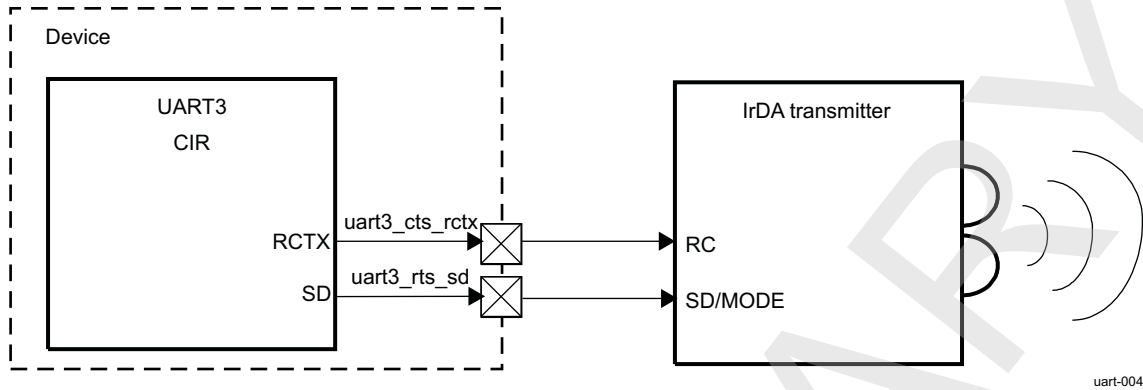
**Figure 23-53. IrDA System Overview**



#### 23.3.2.3 System Using CIR Communication Protocol With Remote Control

UART3 can be connected to an external Infrared transceiver in CIR mode (see [Figure 23-54](#)).

Figure 23-54. CIR System Overview



### 23.3.2.4 UART Interface

#### 23.3.2.4.1 Description

Table 23-135 lists the UART interface input/output (I/O) signals.

Table 23-135. UART I/O Pins

Signal	I/O <sup>(1)</sup>	Description	Reset
<b>UART Modem Signals</b>			
uarti_rx	I	Serial data input	Unknown
uarti_tx	O	Serial data output	1
		Because this pin is active high in IrDA mode and the output is muxed, this pin is set to low on reset (when the UARTi.UART_MDR1[2:0] bit field is set to 0x7) and takes the defined inactive level of that signal corresponding to when and how the UARTi.UART_MDR1 register is programmed; that is, the output is 1 (inactive for UART modem modes) and 0 (inactive for IrDA modes).	
uarti_cts	I	Clear to send	Unknown
		Active-low modem status signal. Reading the UARTi.UART_MSR[4] NCTS_STS bit checks the condition of uarti_cts. Reading the UARTi.UART_MSR[0] CTS_STS bit checks a change of state of uarti_cts since the last read of the modem status register. The auto-nCTS mode uses uarti_cts to control the transmitter.	
uarti_rts	O	Request to send	1
		When active (low), the module is ready to receive data. Setting the UARTi.UART_MCR[1] RTS bit activates uarti_rts, which becomes inactive as the result of a module reset, loopback mode, or clearing the UARTi.UART_MCR[1] RTS bit. In auto-RTS mode, uarti_rts becomes inactive as a result of the receiver threshold logic.	

<sup>(1)</sup> I = Input; O = Output

#### 23.3.2.4.2 UART Protocol and Data Format

The UART device operates in three modes:

- UART 16x (<= 230.4 kbps)
- UART 16x with autobauding (>= 1200 bps and >= 115.2 kbps)
- UART 13x (>= 460.8 kbps)

**CAUTION**

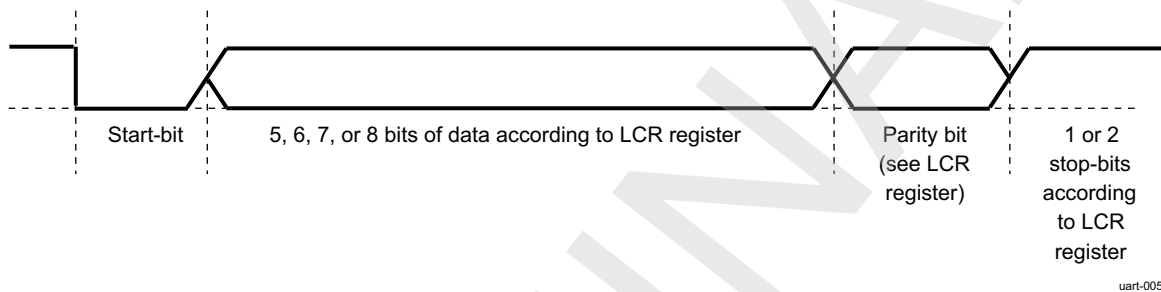
To be used as a UART, the operating mode must be programmed appropriately in the UARTi.UART\_MDR1[2:0] MODE\_SELECT bit field to select UART, IrDA, or CIR mode.

The UART uses a wired interface for serial communication with a remote device.

The UART is functionally compatible with the TL16C750 UART and earlier designs such as the TL16C550.

Figure 23-55 shows the UART frame data format.

**Figure 23-55. UART Frame Data Format**



### 23.3.2.5 IrDA Functional Interfaces

#### 23.3.2.5.1 UART3 Interface Description

Table 23-136 lists the UART3 interface I/O signals.

**Table 23-136. UART3 I/O Signals**

Signal	I/O <sup>(1)</sup>	Description	Reset
<b>IrDA Signals</b>			
uart3_rx_irrx	I	Serial data input	Unknown
uart3_tx_irtx	O	Serial data output in IrDA modes (SIR, MIR, and FIR). In other modes, this pin is set to the reset value (inactive state).	0
uart3_rts_sd	O	SD mode is used to configure the transceivers. The SD pinout is an inverted value of the UART3.UART_ACREG[6] SD_MOD bit.	1

<sup>(1)</sup> I = Input; O = Output



### 23.3.2.5.2 IrDA Protocol and Data Format

#### 23.3.2.5.2.1 SIR Mode

In SIR mode, data is transferred between the MPU and peripheral devices at speeds of up to 115,200 baud. A SIR transmit frame begins with start flags (a single 0xC0, a multiple 0xC0, or a single 0xC0 preceded by a number of 0xFF flags), is followed by frame data and a CRC-16, and ends with a stop flag (0xC1).

The bit format for a single word uses 1 start-bit, 8 data bits, and 1 stop-bit, and is unaffected by the use and settings of the UART3.UART\_LCR register.

The UART3.UART\_BLR[6] XBOF\_TYPE bit selects whether the 0xC0 or 0xFF start patterns are used when multiple start flags are required.

The SIR transmit state-machine attaches start flags, CRC-16, and stop flags, and checks the outgoing data to establish whether data transparency is required.

SIR transparency is carried out if the outgoing data between the start and stop flags contains 0xC0, 0xC1, or 0x7D. If one of these start flags is about to be transmitted, the SIR state-machine first sends an escape character (0x7D), then inverts the fifth bit of the real data to be sent, and sends this data immediately after the 0x7D character.

The SIR receive state-machine recovers the receive clock, removes the start flags and any transparency from the incoming data, and determines the frame boundary with reception of the stop flag. The SIR state-machine also checks for errors such as a frame abort (0x7D character followed immediately by a 0xC1 stop flag without transparency), a CRC error, or a frame-length error. At the end of a frame reception, the MPU reads the line status register (UART3.UART\_LSR) to find possible errors of the received frame.

---

**NOTE:** The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware. See the description of the UART3.UART\_ACREG[5] DIS\_IR\_RX bit. This applies to all three modes: SIR, MIR, and FIR.

---

Infrared output in SIR mode can be 1.6- $\mu$ s or 3/16 encoding, selected by the UART3.UART\_ACREG[7] PULSE\_TYPE bit. In 1.6- $\mu$ s encoding, the infrared pulse width is 1.6  $\mu$ s; and in 3/16th encoding, the infrared pulse width is 3/16th of a bit duration (1/ baud rate).

For back-to-back frames, the transmitting device must send at least two start flags at the start of each frame.

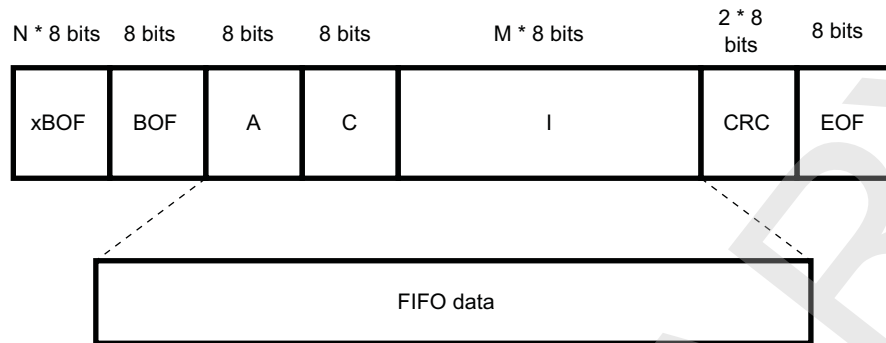
---

**NOTE:** Reception supports variable-length stop-bits.

---

#### 23.3.2.5.2.1.1 Frame Format

Figure 23-56 shows the IrDA SIR frame format.

**Figure 23-56. IrDA SIR Frame Format**

uart-006

The CRC is applied on the address (A), control (C), and information (I) bytes.

---

**NOTE:** The two words of CRC are written to the FIFO in reception.

---

#### 23.3.2.5.2.1.2 Asynchronous Transparency

Before transmitting a byte, the UART IrDA controller examines each byte of the payload and the CRC field (between BOF and EOF). For each byte equal to 0xC0 (BOF), 0xC1 (EOF), or 0x7D (control escape), the controller performs certain tasks:

- In transmission:

- Inserts a control escape (CE) byte preceding the byte
- Complements bit 5 of the byte (that is, exclusive ORs the byte with 0x20)

The byte sent for the CRC computation is the initial byte written in the TX FIFO (before the XOR with 0x20).

- In reception:

For the A, C, I, CRC field:

- Compares the byte with the CE byte; if they are not equal, sends the byte to the CRC detector and stores it in the RX FIFO
- If the byte is equal to the CE byte, discards the CE byte
- Complements bit 5 of the byte following the CE
- Sends the complemented byte to the CRC detector and stores it in the RX FIFO

#### 23.3.2.5.2.1.3 Abort Sequence

The transmitter can prematurely close a frame (abort) by sending the sequence 0x7DC1. The abort pattern closes the frame without a CRC field or an ending flag.

When a 0x7D character followed immediately by a 0xC1 character is received without transparency, the receiver treats the frame as an aborted frame.

#### 23.3.2.5.2.1.4 Pulse Shaping

SIR mode supports the 3/16 and the 1.6-us pulse duration methods. The UART3.UART\_ACREG[7] PULSE\_TYPE bit selects the pulse-width method in transmit mode.

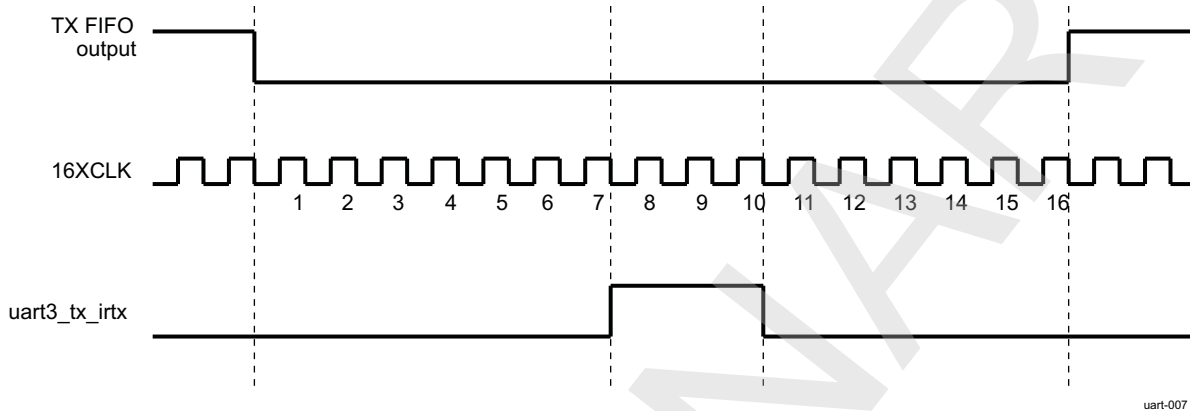
#### 23.3.2.5.2.1.5 Encoder

Serial data from the transmit state-machine are encoded to transmit data to the optoelectronics. While the TX FIFO output is high, the `uart3_tx_irtx` line is always low, and the counter used to form a pulse on `uart3_tx_irtx` is cleared continuously.

After the TX FIFO output resets to 0, `uart3_tx_irtx` rises on the falling edge of the seventh 16XCLK. On the falling edge of the tenth 16XCLK pulse, `uart3_tx_irtx` falls, creating a three-clock-wide pulse. While the TX FIFO output stays low, a pulse is transmitted during the seventh to the tenth clock of each 16-clock bit cycle.

Figure 23-57 shows the IrDA SIR encoding mechanism.

Figure 23-57. IrDA SIR Encoding Mechanism

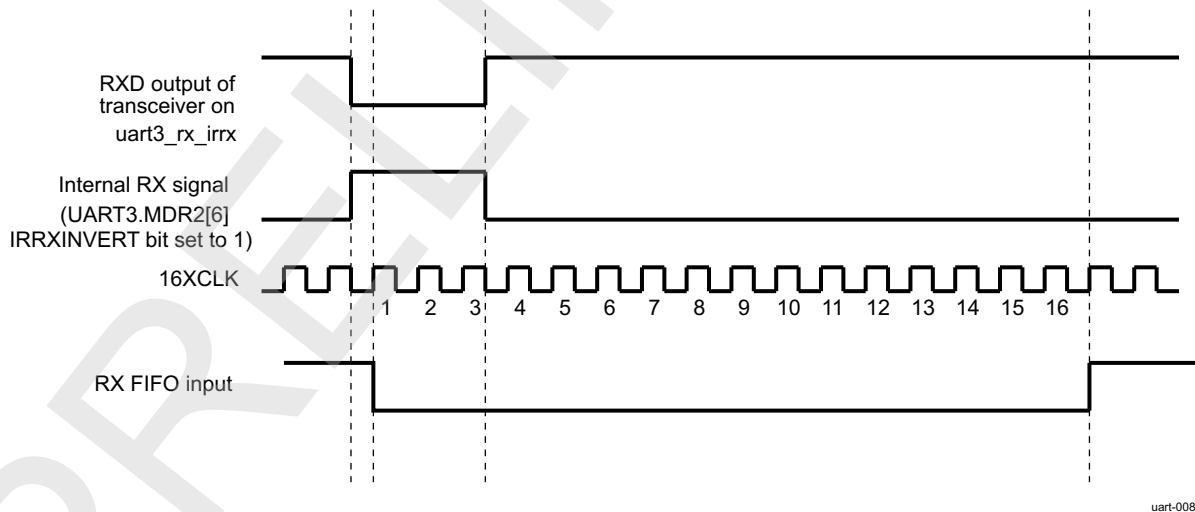


23.3.2.5.2.1.6 Decoder

After reset, the RX FIFO input is high and the 4-bit counter is cleared. When a rising edge is detected on RX, the RX FIFO input falls on the next rising edge of 16XCLK with sufficient setup time. The RX FIFO input stays low for 16 cycles (16XCLK) and then returns to high as required by the IrDA specification. As long as no pulses (rising edges) are detected on the RX, the RX FIFO input remains high.

Figure 23-58 shows the IrDA SIR decoding mechanism.

Figure 23-58. IrDA SIR Decoding Mechanism



The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware. The operation of the `uart3_rx_irrx` input can be disabled using the `UART3.UART_ACREG[5] DIS_IR_RX` bit. The `UART3.UART_MDR2[6] IRRXINVERT` bit can invert the signal from the transceiver (RXD) pin to the IRRX logic in the UART. This inversion is performed by default.

### 23.3.2.5.2.1.7 IR Address Checking

In all IR modes, when address checking is enabled by setting the `UART_EFR[1:0]` bit field (see [Table 23-137](#)), only frames intended for the device are written to the RX FIFO. This is to avoid receiving frames not meant for this device in a multipoint infrared environment. To program two frame addresses that the UART3 receives in IrDA mode, use the `UART3.UART_XON1_ADDR1[7:0]` bit field and the `UART3.UART_XON2_ADDR2[7:0]` bit field.

**Table 23-137. UART\_EFR[1:0] IR Address Checking Options**

UART_EFR[1]	UART_EFR[0]	IR Address Checking
0	0	All address-checking operations disabled
0	1	Only address 1 checking enabled
1	0	Only address 2 checking enabled
1	1	All address-checking operations enabled

### 23.3.2.5.2.2 SIR Free-Format Mode

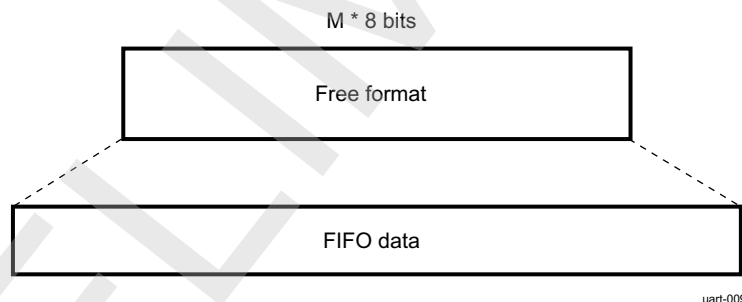
To allow complete software flexibility when transmitting and receiving infrared data packets, SIR free-format (FF) mode is a subfunction of the existing SIR mode. In FF mode, all frames going to and from the FIFO buffers are untouched with respect to appending and removing control characters and CRC values.

FF mode corresponds to a UART mode with a pulse modulation of 3/16 of baud rate pulse width.

For example, a normal SIR packet has BOF control and CRC error-checking data appended (transmitting) or removed (receiving) from the data going to and from the FIFOs.

[Figure 23-59](#) shows SIR FF mode.

**Figure 23-59. SIR FF Mode**

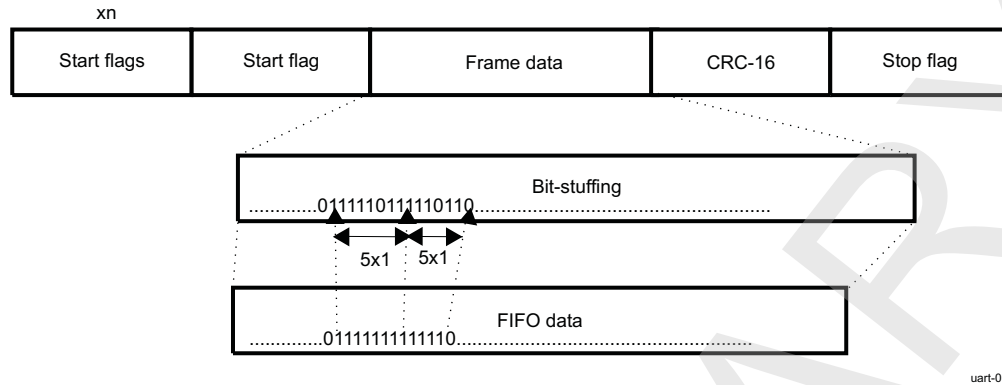


In SIR FF mode, the MPU software must construct (that is, encode and decode) the entire FIFO data packet.

### 23.3.2.5.2.3 MIR Mode

In MIR mode, data is transferred between the MPU and the peripheral devices at 0.576 or 1.152 mbps speed. A MIR transmit frame starts with at least two start flags, followed by a frame data, CRC-16, and ends with a stop flag (see [Figure 23-60](#)).

Figure 23-60. MIR Transmit Frame Format



On transmit, the MIR state-machine attaches start flags, CRC-16, and stop flags, as in SIR mode. All fields are transmitted least-significant bit (LSB) of each byte first.

In MIR mode:

- The state-machine looks for consecutive 1s in the frame data and automatically inserts 0 after five consecutive 1s (this is called bit-stuffing).
- 0x7E is used for start and stop flags (unambiguously, not data, because of bit-stuffing).
- An abort sequence requires a minimum of seven consecutive 1s (unambiguously, not data, because of bit-stuffing).
- Back-to-back frames are allowed with three or more stop flags between them. If two consecutive frames are not back to back, the gap between the last stop flag of the first frame and the start flag of the second frame must be separated by at least seven bit durations.

On receive, the MIR receive state-machine recovers the receive clock, removes the start flags, destuffs the incoming data, and determines the frame boundary with reception of the stop flag. The state-machine also checks for errors such as frame abort, CRC error, and frame-length error. At the end of a frame reception, the MPU reads the line status register (UART3.UART\_LSR) to detect errors of the received frame.

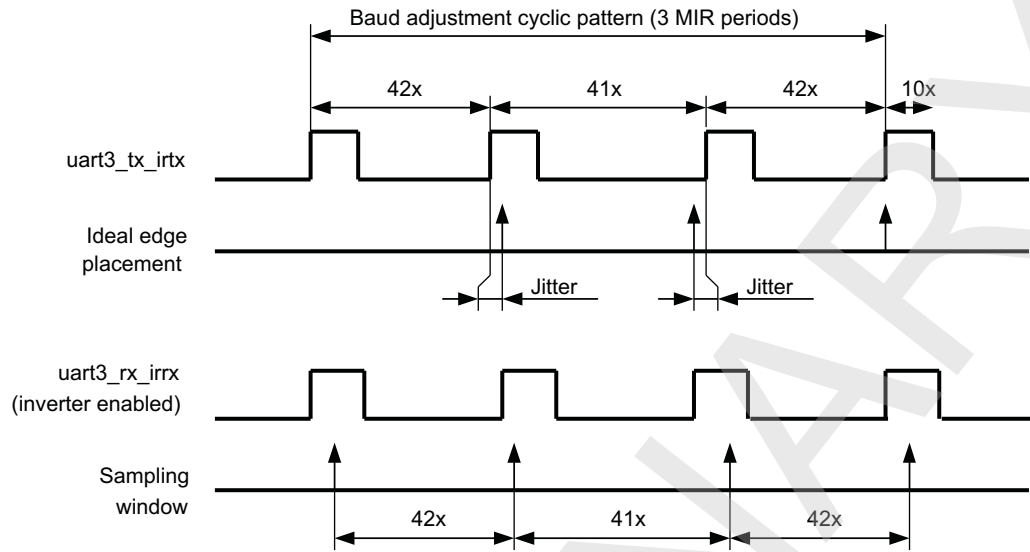
The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware.

### 23.3.2.5.2.3.1 MIR Encoder/Decoder

To meet the MIR baud rate tolerance of 0.1 percent with a 48-MHz clock input, a 42-41-42 encoding/decoding adjustment is performed. The reference start point is the first start flag, and the 42-41-42 cyclic pattern is repeated until the stop flag is sent or detected.

The jitter created this way is within MIR tolerances. The pulse width is not exactly 1/4, but it is within the tolerances defined by IrDA specifications.

Figure 23-61 shows the MIR baud rate adjustment mechanism.

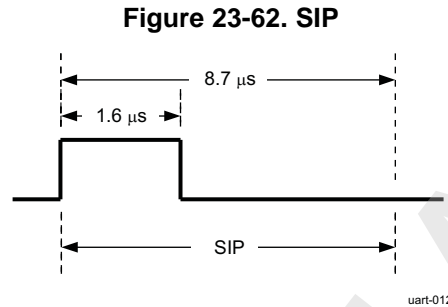
**Figure 23-61. MIR Baud Rate Adjustment Mechanism**

uart-011

**23.3.2.5.2.3.2 SIP Generation**

In the MIR and FIR operation modes, the transmitter must send a serial infrared interaction pulse (SIP) at least once every 500 ms. The SIP informs slow devices (operating in SIR mode) that the medium is occupied.

Figure 23-62 shows the SIP.



**23.3.2.5.2.4 FIR Mode**

In FIR mode, data is transferred between the MPU and the peripheral devices at 4 Mbps. A FIR transmit frame starts with a preamble that is followed by a start flag, frame data, CRC-32, and ends with a stop flag.

Figure 23-63 shows the FIR transmit frame format.

**Figure 23-63. FIR Transmit Frame Format**

Preamble (16x)	Start flag	Frame data	CRC-32	Stop flag
----------------	------------	------------	--------	-----------

On transmit, the FIR transmit state-machine attaches the preamble, start flag, CRC-32, and stop flag. An abort sequence requires at least two transmissions of 0000. Back-to-back frames are allowed, but each frame must be complete.

The state-machine also encodes the transmit data into 4-PPM format (see Table 23-138) and generates the SIP (see Section 23.3.2.5.2.3.2, SIP Generation).

**Table 23-138. 4-PPM Format**

Data Bit Pair (Bin)	4-PPM Data Symbol (Bin)
00	1000
01	0100
10	0010
11	0001

The four symbols described in Table 23-138 are the legal encoded data symbols. All other combinations are illegal for encoding data. Some of these illegal symbols are used in the definition of the preamble, start flag, and stop flag because they are unambiguously not data (see Table 23-139).

**Table 23-139. FIR Preamble, Start Flag, and Stop Flag**

Frame Part	Transmitted Frame (Bin)
Preamble	1000 0000 1010 1000 (16 repeated transmissions)
Start flag	0000 1100 0000 1100 0110 0000 0110 0000
Stop flag	0000 1100 0000 1100 0000 0110 0000 0110

All fields are transmitted the LSBs of each byte first (see [Table 23-140](#)).

**Table 23-140. FIR Data Byte Transmission Order Example**

Data Byte (Hex)	Data Byte Pair (Bin)	4-PPM Data Symbol (Bin)	Transmission Order
0x0B	00	1000	4
	00	1000	3
	10	0010	2
	11	0001	1

On receive, the FIR receive state-machine recovers the receive clock, removes the preamble and the start flag, decodes the 4-PPM incoming data, and determines the frame boundary with reception of the stop flag. The state-machine also checks for errors such as illegal symbol, CRC error, and frame-length error. At the end of a frame reception, the MPU reads the line status register (UART3.UART\_LSR) to detect errors of the received frame.

The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware.

### 23.3.2.6 CIR Functional Interfaces

#### 23.3.2.6.1 CIR Interface Description

[Table 23-141](#) lists the CIR interface I/O signals.

**Table 23-141. CIR I/O Signals**

Signal	I/O <sup>(1)</sup>	Description	Reset
<b>CIR Signals</b>			
uart3_rx_irrx	I	Serial data input	Unknown
uart3_cts_rctx	O	Serial data output in CIR mode. In other modes, this pin is set to the reset value (inactive state).	0
uart3_rts_sd	O	SD mode is used to configure the transceivers. The SD pinout is an inverted value of the UART3.UART_ACREG[6] SD_MOD bit.	1

<sup>(1)</sup> I = Input; O = Output

#### 23.3.2.6.2 CIR Protocol and Data Format

In CIR mode, the infrared operation functions as a programmable (universal) remote control.

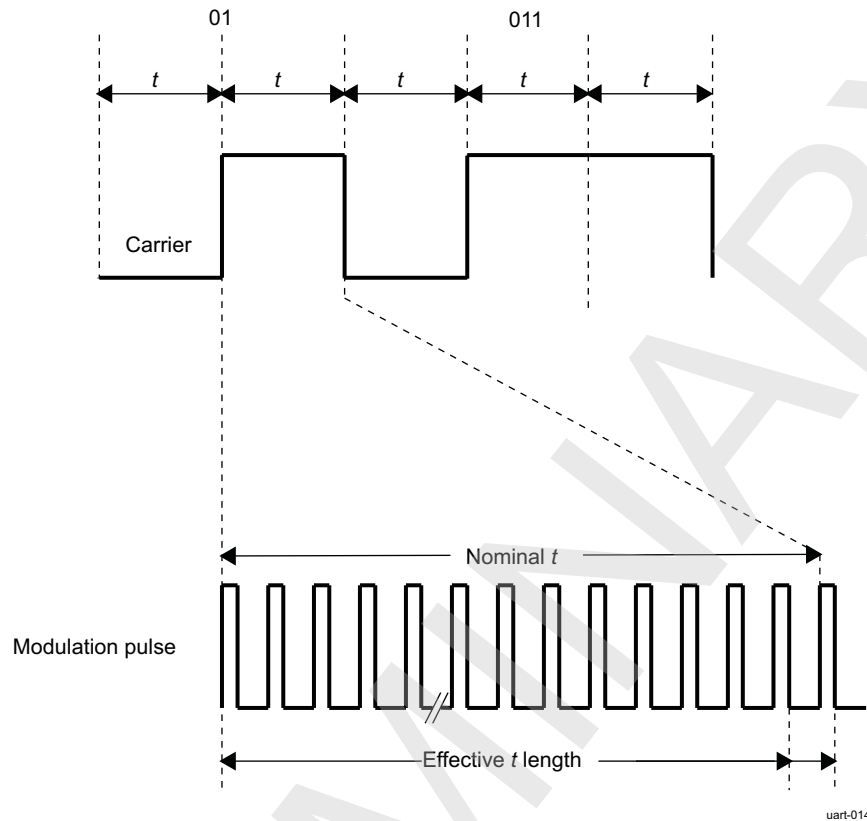
CIR mode uses a variable PWM technique (based on multiples of a programmable  $t$  period) to encompass the various formats of infrared encoding for remote-control applications. The CIR logic transmits data packets based on user-defined frame structure and packet content.

##### 23.3.2.6.2.1 Carrier Modulation

Each modulated pulse that constitutes a digit is a train of on/off pulses (see [Figure 23-64](#)).



Figure 23-64. CIR Pulse Modulation

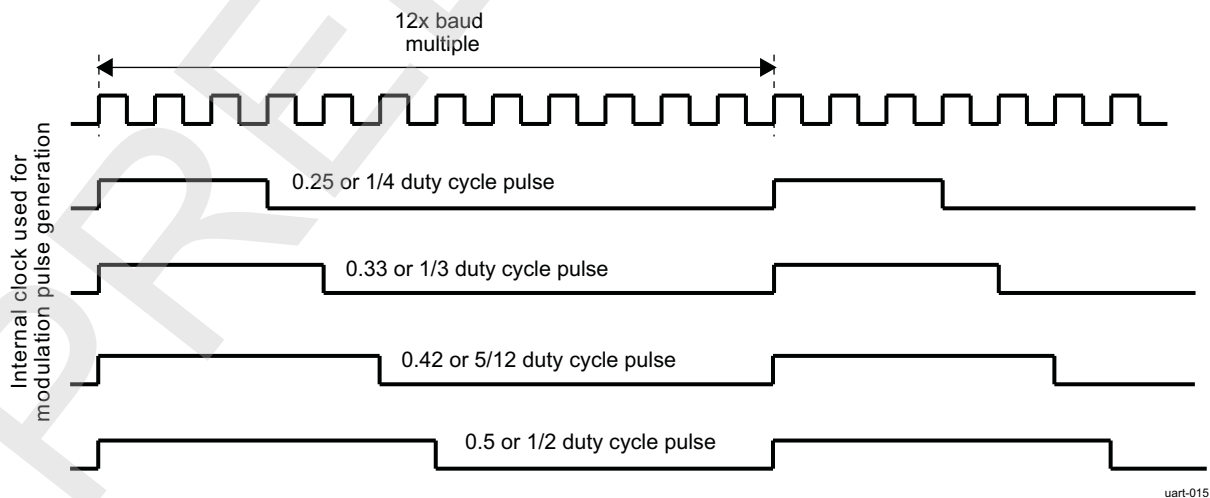


### 23.3.2.6.2.2 Pulse Duty Cycle

The programmer can choose one of four duty cycles for modulation pulses by setting the appropriate value in the UART3.UART\_MDR2[5:4] CIR\_PULSE\_MODE bit field (1/4, 1/3, 5/12, or 1/2).

Figure 23-65 shows the CIR modulation duty cycles.

Figure 23-65. CIR Modulation Duty Cycle



The transmission logic ensures that all pulses are transmitted completely (no cutoff during transmission). While transmitting continuous bytes back-to-back, no delay is inserted between 2 transmitted bytes. Thus, software must handle the delay between consecutively transmitted bytes if the receiving end requires it.

### 23.3.2.6.2.3 Consumer IR Encoding/Decoding

There are two methods of encoding for remote-control applications:

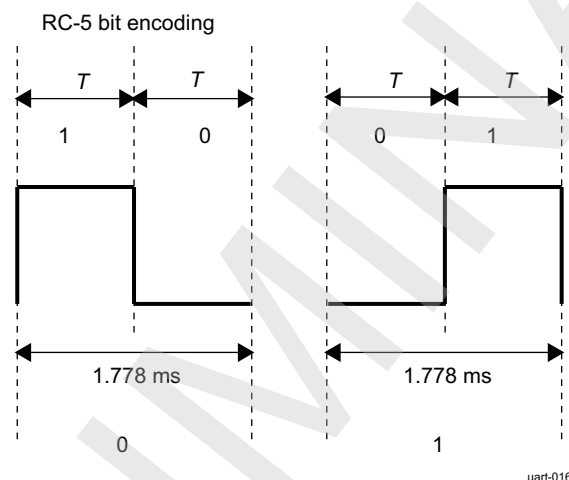
- Pulse duration encoding (time-extended bit forms): A variable pulse distance, or duration, in which the difference between logic 1 and logic 0 is the length of the pulse width
- Biphase encoding: The encoding of logic 0 and logic 1 is in the change of signal level from 1 to 0 or 0 to 1, respectively.

Japanese manufacturers favor pulse duration encoding; European manufacturers favor biphase encoding.

CIR mode uses a completely flexible free-format encoding in which 1 is transmitted from the TX FIFO as a modulated pulse with duration  $T$ .

Similarly, 0 is transmitted as a blank duration  $T$ . The MPU constructs and deciphers the protocol of the data. For example, the RC-5 protocol using Manchester encoding can be emulated as using a 01 pair for 1 and a 10 pair for 0 (see [Figure 23-66](#)).

**Figure 23-66. RC-5 Bit Encoding**

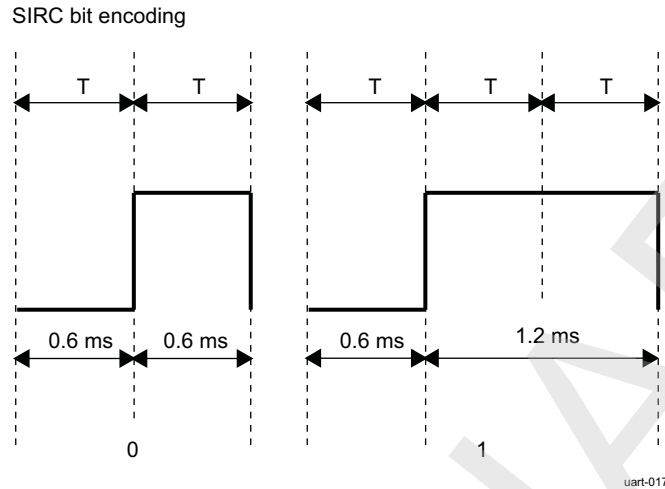


Because CIR mode logic does not impose a fixed format for infrared packets of data, the MPU software can define the format using simple data structures that are then modulated into an industry standard, such as RC-5 or SIRC. To send a sequence of 0101 in RC-5, the MPU software must write an 8-bit binary character of 10011001 to the data FIFO of the UART.

For SIRC, the modulation length (multiples of  $T$ ) is used to distinguish between 1 and 0. The subsequent SIRC digits show the difference in encoding between this and, for example, RC-5. The pulse width is extended for one digit.

[Figure 23-67](#) shows SIRC bit encoding.

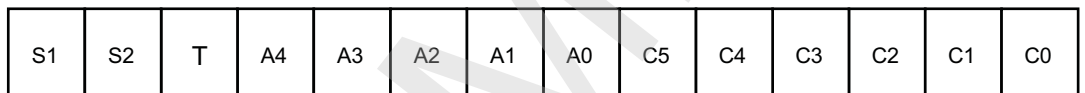
**Figure 23-67. SIRC Bit Encoding**



To construct comprehensive packets constituting remote-control commands, the MPU software must combine a number of 8-bit data characters in a sequence that follows one of the universally accepted formats.

Figure 23-68 shows a standard RC-5 frame as detected by UART3 in CIR mode (the SIRC format follows this). Each field in RC-5 can be considered as two T pulses (digital bits) from the TX FIFO.

**Figure 23-68. RC-5 Standard Packet Format**



uart-018

Where:

- S1, S2: Start-bits (always 1)
- T: Toggle bit
- A4..A0: Address (or system) bits
- C5..C0: Command bits

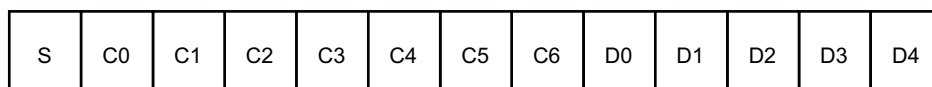
The toggle bit T changes when a new command is transmitted to detect when the same key is pressed twice (effectively receiving the same data from the host consecutively). A brief delay in the transmission of the same command is detected by the use of the toggle bit because a code is sent while the MPU transmits characters to the UART for transmission. The address bits define the machine or device for which the infrared transmission is intended, and the command defines the operation.

To accommodate an extended RC-5 format, the S2 bit is replaced by an additional command bit (C6) that lets the command range increase to 7 bits. This format is known as the extended RC-5 format.

SIRC encoding uses the duration of modulation for mark and space; therefore, the duration of data bits in the standard frame length varies.

Figure 23-69 shows the packet format and bit encoding. As Figure 23-70 shows, 1 start-bit of 2.4 ms and control codes are followed by data that constitute the entire frame.

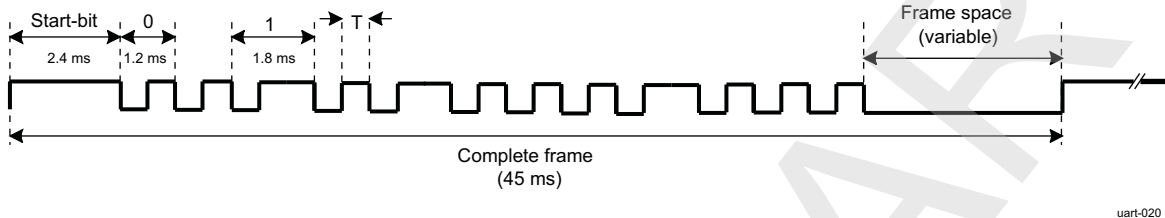
**Figure 23-69. SIRC Packet Format**



uart-019

**NOTE:** The encoding must take a standard duration, but the contents of the data can vary. This implies that the control software for sending and receiving data packets must exercise a scheme of interpacket delay, where successive packets can be sent only after a real-time delay expires.

**Figure 23-70. SIRC Bit Transmission Example**



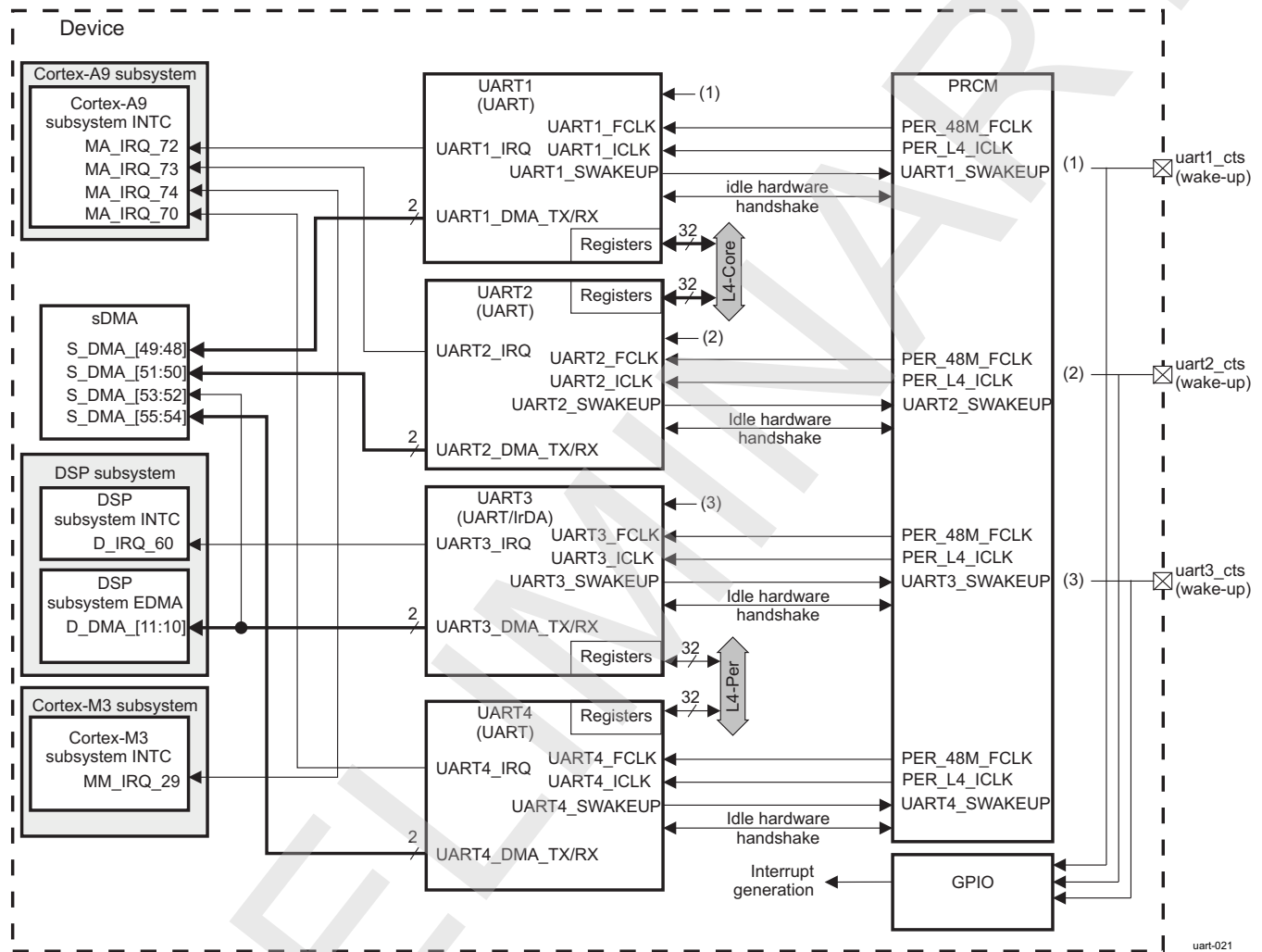
This document does not describe all encoding methods and techniques; the preceding information shows the considerations required to employ different encoding methods for different industry-standard protocols. See industry-standard documentation for specific methods of encoding and protocol use.

### 23.3.3 UART/IrDA/CIR Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 23-71 shows the device internal connections with related modules for UART functions.

Figure 23-71. UART/IrDA/CIR Integration



**NOTE:** For more information about the idle and standby hardware handshakes and the wake-up request, see Section 3.1.1.1.4, *Clock Domain Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 23-142 through Table 23-144 summarize the integration of the module in the device.

Table 23-142. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
UART1	PD_L4_PER	Yes	L4_PER
UART2	PD_L4_PER	Yes	L4_PER
UART3	PD_L4_PER	Yes	L4_PER
UART4	PD_L4_PER	Yes	L4_PER

**Table 23-143. Clocks and Resets**

Clocks				
Module Instance	Destination Signal	Source Signal	Source	Description
UART1	UART1_ICLK	PER_L4_ICLK	PRCM	UART1 interface clock
	UART1_FCLK	PER_48M_FCLK	PRCM	UART1 functional clock
UART2	UART2_ICLK	PER_L4_ICLK	PRCM	UART1 interface clock
	UART2_FCLK	PER_48M_FCLK	PRCM	UART1 functional clock
UART3	UART3_ICLK	PER_L4_ICLK	PRCM	UART3 interface clock
	UART3_FCLK	PER_48M_FCLK	PRCM	UART3 functional clock
UART4	UART4_ICLK	PER_L4_ICLK	PRCM	UART4 interface clock
	UART4_FCLK	PER_48M_FCLK	PRCM	UART4 functional clock
Resets				
Module Instance	Destination Signal	Source Signal	Source	Description
UART1	UART1_RST	L4_PER_RST	PRCM	UART1 reset signal
UART2	UART2_RST	L4_PER_RST	PRCM	UART2 reset signal
UART3	UART3_RST	L4_PER_RST	PRCM	UART3 interface reset
UART4	UART4_RST	L4_PER_RST	PRCM	UART4 reset

**Table 23-144. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal	Destination Signal	Destination	Description
UART1	UART1_IRQ	MA_IRQ_72	Cortex™-A9	UART module 1
UART2	UART2_IRQ	MA_IRQ_73	Cortex-A9	UART module 2
UART3	UART3_IRQ	D_IRQ_60	DSP	UART module 3 (also infrared)
	UART3_IRQ	MA_IRQ_74	Cortex-A9	UART module 3 (also infrared)
	UART3_IRQ	MM_IRQ_29	Cortex™-M3	UART module 3 (also infrared)
UART4	UART4_IRQ	MA_IRQ_70	Cortex-A9	UART module 4
Direct Memory Access (DMA) Requests				
Module Instance	Source Signal	Destination Signal	Destination	Description
UART1	UART1_DMA_TX	S_DMA_48	sDMA	UART module 1 – transmit request
	UART1_DMA_RX	S_DMA_49	sDMA	UART module 1 – receive request
UART2	UART2_DMA_TX	S_DMA_50	sDMA	UART module 2 – transmit request
	UART2_DMA_RX	S_DMA_51	sDMA	UART module 2 – receive request
UART3	UART3_DMA_TX	S_DMA_52	sDMA	UART module 3 – transmit request (also infrared)
	UART3_DMA_RX	S_DMA_53	sDMA	UART module 3 – receive request (also infrared)
	UART3_DMA_TX	D_DMA_10	DDMA	UART module 3 – transmit request [0]
	UART3_DMA_RX	D_DMA_11	DDMA	UART module 3 – receive request [0]
UART4	UART4_DMA_TX	S_DMA_54	sDMA	UART module 4 – transmit request
	UART4_DMA_RX	S_DMA_55	sDMA	UART module 4 – receive request

## 23.3.4 UART/IrDA/CIR Functional Description

### 23.3.4.1 Block Diagram

The UART/IrDA/CIR module can be divided into three main blocks:

- FIFO management
- Mode selection
- Protocol formatting

FIFO management is common to all functions and enables the transmission and reception of data from the host processor point of view.

There are two modes:

- Function mode: Routes the data to the chosen function (UART, IrDA, or CIR) and enables the mechanism corresponding to the chosen function
- Register mode: Enables conditional access to registers

For more information about mode configuration, see [Section 23.3.4.7, Mode Selection](#).

Protocol formatting has three subcategories:

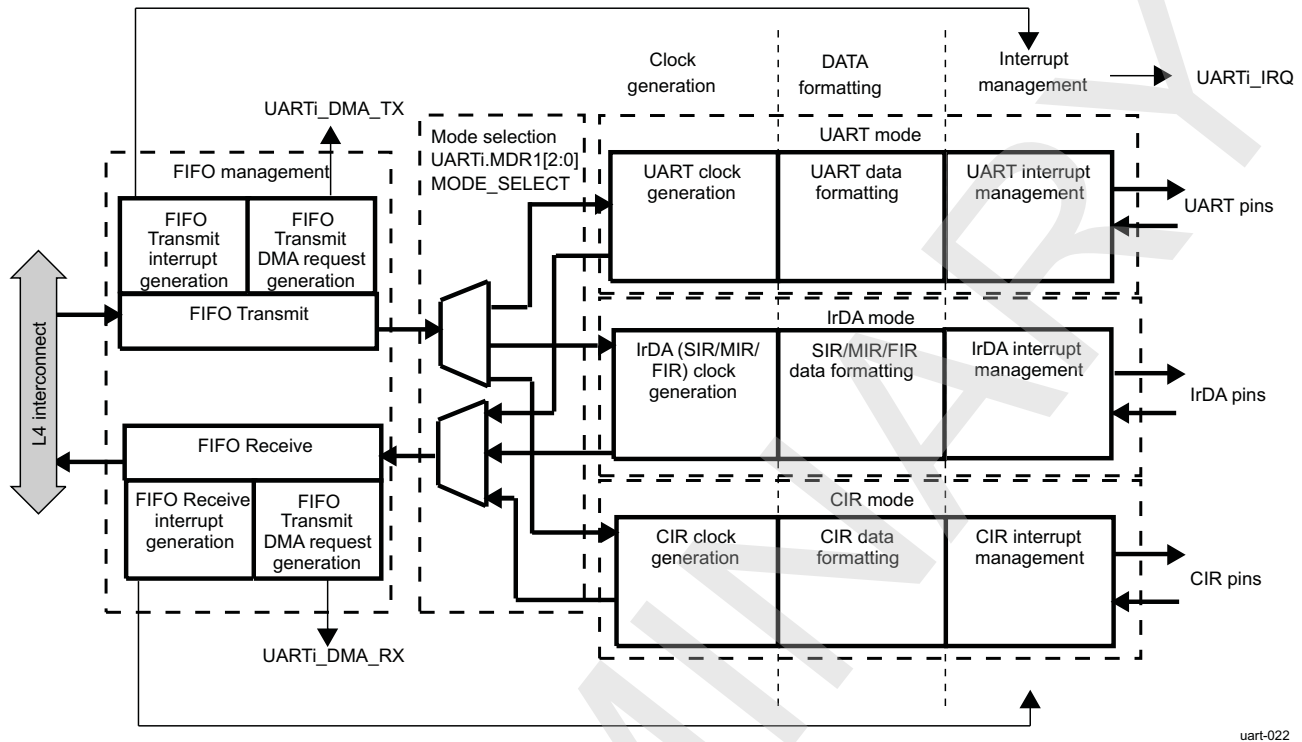
- Clock generation: The 48-MHz input clock generates all necessary clocks.
- Data formatting: Each function uses its own state-machine that is responsible for the transition between FIFO data and frame data associated with it.
- Interrupt management: Different interrupt types are generated depending on the chosen function:
  - UART mode interrupts: Seven interrupts prioritized in six different levels
  - IrDA mode interrupts: Eight interrupts. The interrupt line is activated when any interrupt is generated (there is no priority).
  - CIR mode interrupts: A subset of existing IrDA mode interrupts is used.

In each mode, when an interrupt is generated, the [UART\\_IIR](#) register indicates the interrupt type.

In parallel with these functional blocks, a power-saving strategy exists for each function.

[Figure 23-72](#) is the UART/IrDA/CIR block diagram.

Figure 23-72. UART/IrDA/CIR Functional Specification Block Diagram



uart-022

### 23.3.4.2 Clock Configuration

Each UART uses a 48-MHz functional clock for its logic and to generate external interface signals. Each UART uses an interface clock for register accesses. The power, reset, and clock management (PRCM) module generates and controls all these clocks (for more information, see [Section 3.6.4.4, Clock Domain Module Attributes](#), in [Chapter 3, Power, Reset, and Clock Management](#)).

The idle and wake-up processes use a handshake protocol between the PRCM and the UART (for a description of the protocol, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#)). The UARTi.UART\_SYSC[4:3] IDLEMODE bit field controls UART idle mode.

### 23.3.4.3 Software Reset

The UARTi.UART\_SYSC[1] SOFTRESET bit controls the software reset; setting this bit to 1 triggers a software-reset functionally equivalent to hardware reset.

### 23.3.4.4 Power Management

#### 23.3.4.4.1 UART Mode Power Management

##### 23.3.4.4.1.1 Module Power Saving

In UART modes, sleep mode is enabled by setting the UARTi.UART\_IER[4] SLEEP\_MODE bit to 1 (when the UARTi.UART\_EFR[4] ENHANCED\_EN bit is set to 1).

Sleep mode is entered when all the following conditions exist:

- The serial data input line, uarti\_rx, is idle.
- The TX FIFO and TX shift register are empty.
- The RX FIFO is empty.
- The only pending interrupts are THR interrupts.



Sleep mode is a good way to lower power consumption of the UART, but this state can be achieved only when the UART is set in modem mode. Therefore, even if the UART has no functional key role, it must be initialized in a functional mode to take advantage of sleep mode.

In sleep mode, the module clock and baud rate clock are stopped internally. Because most registers are clocked by these clocks, this greatly reduces power consumption. The module wakes up when a change is detected on the `uarti_rx` line, when data is written to the TX FIFO, and when there is a change in the state of the modem input pins.

An interrupt can be generated on a wake-up event by setting the `UARTi.UART_SCR[4] RX_CTS_WU_EN` bit to 1. To understand how to manage the interrupt, see [Section 23.3.4.5.1.1, Wake-Up Interrupt](#).

---

**NOTE:** There must be no writing to the divisor latches, `UARTi.UART_DLL` and `UARTi.UART_DLH`, to set the baud clock, BCLK, while in sleep mode. It is advisable to disable sleep mode using the `UARTi.UART_IER[4] SLEEP_MODE` bit before writing to the `UARTi.UART_DLL` register or the `UARTi.UART_DLH` register.

---

#### **23.3.4.4.1.2 System Power Saving**

Sleep and auto-idle modes are embedded power-saving features. Power-reduction techniques can be applied at the system level by shutting down certain internal clock and power domains of the device.

The UART supports an idle req/idle ack handshaking protocol used at the system level to shut down the UART clocks in a clean and controlled manner and to switch the UART from interrupt-generation mode to wake-up generation mode for unmasked events (see the `UARTi.UART_SYSC[2] ENAWAKEUP` bit and the `UARTi.UART_WER` register).

For more information, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

#### **23.3.4.4.2 IrDA Mode Power Management (UART3 Only)**

##### **23.3.4.4.2.1 Module Power Saving**

In IrDA modes, sleep mode is enabled by setting the `UART3.MDR[3] IR_SLEEP` bit to 1.

Sleep mode is entered when all the following conditions exist:

- The serial data input line, `uart3.rx_irrx`, is idle.
- The TX FIFO and TX shift register are empty.
- The RX FIFO is empty.
- No interrupts are pending except THR interrupts.

The module wakes up when a change is detected on the `uart3_rx_irrx` line or when data is written to the TX FIFO.

##### **23.3.4.4.2.2 System Power Saving**

System power saving for IrDA mode has the same function as that for UART mode (see [Section 23.3.4.4.1.2, System Power Saving](#)).

#### **23.3.4.4.3 CIR Mode Power Management (UART3 Only)**

##### **23.3.4.4.3.1 Module Power Saving**

Module power saving for CIR mode has the same function as that for IrDA mode (see [Section 23.3.4.4.2.1, Module Power Saving](#)).

### 23.3.4.4.3.2 System Power Saving

System power saving for CIR mode has the same function as that for UART mode (see [Section 23.3.4.4.1.2, System Power Saving](#)).

### 23.3.4.4.4 Local Power Management

[Table 23-145](#) describes power-management features available for the UART.

**NOTE:**

- For information about source clock gating and sleep/wake-up transitions description, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**Table 23-145. Local Power Management features**

Feature	Registers	Description
Clock autogating	<a href="#">UART_SYSC</a> [0] AUTOIDLE	This bit allows local power optimization in the module by gating the UARTi_ICLK clock on interface activity or gating the UARTi_FCLK clock on internal activity.
Slave idle modes	<a href="#">UART_SYSC</a> [4:3] IDLEMODE	Force-idle, no-idle, smart-idle, and smart-idle wakeup-capable modes are available
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	<a href="#">UART_SYSC</a> [2] ENAWAKEUP	This bit enables the wake-up feature at module level.
Wake-Up sources enable	N/A	Feature not available

### 23.3.4.5 Interrupt Requests

#### 23.3.4.5.1 UART Mode Interrupt Management

UART mode includes seven possible interrupts prioritized to six levels.

When an interrupt is generated, the interrupt identification register (UARTi.UART\_IIR) sets the UARTi.UART\_IIR[0] IT\_PENDING bit to 0 to indicate that an interrupt is pending, and indicates the type of interrupt through the UARTi.UART\_IIR[5:1] bit field. [Table 23-146](#) summarizes the interrupt control functions.

**Table 23-146. UART Mode Interrupts**

UART_IIR[5:0]	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Method
000001	None	None	None	None
000110	1	Receiver line status	OE, FE, PE, or BI errors occur in characters in the RX FIFO.	FE, PE, BI: Read the <a href="#">UART_RHR</a> register. OE: Read the <a href="#">UART_LSR</a> register.
001100	2	RX time-out	Stale data in RX FIFO	Read the <a href="#">UART_RHR</a> register.
000100	2	RHR interrupt	DRDY (data ready) (FIFO disable) RX FIFO above trigger level (FIFO enable)	Read the <a href="#">UART_RHR</a> register until the interrupt condition disappears.
000010	3	THR interrupt	TFE ( <a href="#">UART_THR</a> empty) (FIFO disable) TX FIFO below trigger level (FIFO enable)	Write to the <a href="#">UART_THR</a> until the interrupt condition disappears.
000000	4	Modem status	See the <a href="#">UART_MSR</a> register.	Read the <a href="#">UART_MSR</a> register.

**Table 23-146. UART Mode Interrupts (continued)**

<b>UART_IIR[5:0]</b>	<b>Priority Level</b>	<b>Interrupt Type</b>	<b>Interrupt Source</b>	<b>Interrupt Reset Method</b>
010000	5	XOFF interrupt/special character interrupt	Receive XOFF characters/special character	Receive XON character(s), if XOFF interrupt/read of the <a href="#">UART_IIR</a> register, if special character interrupt.
100000	6	CTS, RTS	RTS pin or CTS pin change state from active (low) to inactive (high).	Read the <a href="#">UART_IIR</a> register.

For the receiver-line status interrupt, the RX\_FIFO\_STS bit (UARTi.UART\_LSR[7]) generates the interrupt.

For the XOFF interrupt, if an XOFF flow character detection caused the interrupt, the interrupt is cleared by an XON flow character detection. If special character detection caused the interrupt, the interrupt is cleared by a read of the UARTi.UART\_IIR register.

#### 23.3.4.5.1.1 Wake-Up Interrupt

Wake-up interrupt is a special interrupt that works differently from other interrupts. This interrupt is enabled when the UARTi.UART\_SCR[4] RX\_CTS\_WU\_EN bit is set to 1. The UARTi.UART\_IIR register is not modified when this occurs; the UART3.UART\_SSR[1] RX\_CTS\_WU\_STS bit must be checked to detect a wake-up event.

When a wake-up interrupt occurs, it can be cleared only by resetting the UARTi.UART\_SCR[4] RX\_CTS\_WU\_EN bit. This bit must be reenabled (set to 1) after the current wake-up interrupt event is processed to detect the next incoming wake-up event.

#### 23.3.4.5.2 IrDA Mode Interrupt Management

##### 23.3.4.5.2.1 IrDA Interrupts

The IrDA function generates interrupts. All interrupts can be enabled and disabled by writing to the appropriate bit in the interrupt enable register (UART3.UART\_IER). The interrupt status of the device can be checked by reading the interrupt identification register (UART3.UART\_IIR).

UART, IrDA, and CIR modes have different interrupts in the UART/IrDA/CIR module and, therefore, different UART3.UART\_IER and UART3.UART\_IIR mappings, depending on the selected mode.

IrDA modes have eight possible interrupts (see [Table 23-147](#)). The interrupt line is activated when any interrupt is generated (there is no priority).

**Table 23-147. IrDA Mode Interrupts**

<b>UART_IIR Bit</b>	<b>Interrupt Type</b>	<b>Interrupt Source</b>	<b>Interrupt Reset Method</b>
0	RHR interrupt	DRDY (data ready) (FIFO disable) RX FIFO above trigger level (FIFO enable)	Read the <a href="#">UART_RHR</a> register until the interrupt condition disappears.
1	THR interrupt	TFE ( <a href="#">UART_THR</a> empty) (FIFO disable) TX FIFO below trigger level (FIFO enable)	Write to the <a href="#">UART_THR</a> until the interrupt condition disappears.
2	Last byte in RX FIFO	Last byte of frame in RX FIFO is available to be read at the RHR port.	Read the <a href="#">UART_RHR</a> register.
3	RX overrun	Write to the <a href="#">UART_RHR</a> register when the RX FIFO is full.	Read <a href="#">UART_RESUME</a> register.
4	Status FIFO interrupt	Status FIFO triggers level reached.	Read STATUS FIFO.

**Table 23-147. IrDA Mode Interrupts (continued)**

UART_IIR Bit	Interrupt Type	Interrupt Source	Interrupt Reset Method
5	TX status	1. <a href="#">UART_THR</a> empty before EOF sent. Last bit of transmission of the IrDA frame occurred, but with an underrun error. OR 2. Transmission of the last bit of the IrDA frame completed successfully.	1. Read the <a href="#">UART_RESUME</a> register. OR 2. Read the <a href="#">UART_IIR</a> register.
6	Receiver line status interrupt	CRC, ABORT, or frame-length error is written into the STATUS FIFO.	Read the STATUS FIFO (read until empty - maximum of eight reads required).
7	Received EOF	Received end-of-frame	Read the <a href="#">UART_IIR</a> register.

### 23.3.4.5.2.2 Wake-Up Interrupts

The wake-up interrupt for IrDA mode has the same function as that for UART mode (see [Section 23.3.4.5.1.1, Wake-Up Interrupt](#)).

#### CAUTION

Wake-Up interface implementation in this mode is based on the `UARTi_SIDLEACK` low-to-high transition instead of the `UARTi_SIDLEACK` state.

This does not ensure wake-up event generation as expected when configured in smart-idle mode, and the system wakes up for a short period.

### 23.3.4.5.3 CIR Mode Interrupt Management

#### 23.3.4.5.3.1 CIR Interrupts

The CIR function generates interrupts that can be enabled and disabled by writing to the appropriate bit in the interrupt enable register (`UART3.UART_IER`). The interrupt status of the device can be checked by reading the interrupt identification register (`UART3.UART_IIR`).

UART, IrDA, and CIR modes have different interrupts in the UART/IrDA/CIR module and, therefore, different `UART3.UART_IER` and `UART3.UART_IIR` mappings, depending on the selected mode.

[Table 23-148](#) lists the interrupt modes to be maintained. In CIR mode, the sole purpose of the `UART3.UART_IIR[5]` bit is to indicate that the last bit of infrared data was passed to the `uart3_cts_rctx` pin.

**Table 23-148. CIR Mode Interrupts**

UART_IIR Bit Number	Interrupt Type	Interrupt Source	Interrupt Reset Method
0	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
1	THR interrupt	TFE ( <a href="#">UART_THR</a> empty) (FIFO disable) TX FIFO below trigger level (FIFO enable)	Write to the <a href="#">UART_THR</a> register until the interrupt condition disappears.
2	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
3	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
4	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
5	TX status	Transmission of the last bit of the frame is complete successfully.	Read the <a href="#">UART_IIR</a> register.
6	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode

**Table 23-148. CIR Mode Interrupts (continued)**

UART_IIR Bit Number	Interrupt Type	Interrupt Source	Interrupt Reset Method
7	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode

**23.3.4.5.3.2 Wake-Up Interrupts**

The wake-up interrupt for CIR mode has the same function as that for UART mode (see [Section 23.3.4.5.1.1, Wake-Up Interrupt](#)).

**23.3.4.6 FIFO Management**

The FIFO is accessed by reading and writing the UARTi.UART\_RHR and UARTi.UART\_THR registers. Parameters are controlled using the FIFO control register (UARTi.UART\_FCR) and supplementary control register (UARTi.UART\_SCR). Reading the UARTi.UART\_SSR[0] TX\_FIFO\_FULL bit at 1 means the FIFO is full.

The UARTi.UART\_TLR register controls the FIFO trigger level, which enables DMA and interrupt generation. After reset, transmit (TX) and receive (RX) FIFOs are disabled; thus, the trigger level is the default value of 1 byte. [Figure 23-73](#) shows the FIFO management registers.

---

**NOTE:** Data in the UARTi.UART\_RHR register is not overwritten when an overflow occurs.

---

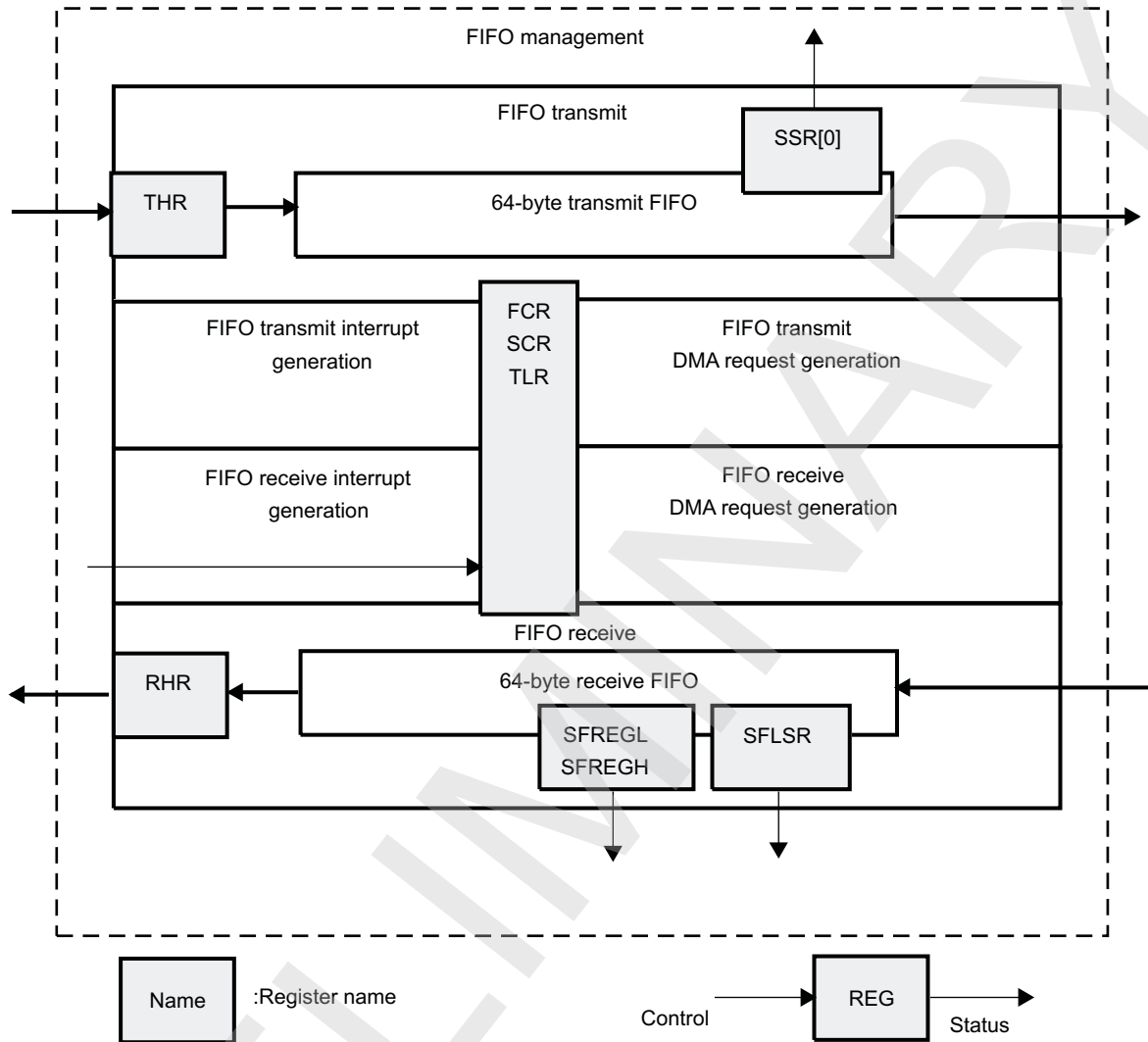


---

**NOTE:** The UARTi.UART\_SFLSR, UARTi.UART\_SFREGL, and UARTi.UART\_SFREGH status registers are used in IrDA mode only. For use, see [Section 23.3.4.8.2.3, IrDA Data Formatting](#).

---

**Figure 23-73. FIFO Management Registers**



uart-023

### 23.3.4.6.1 FIFO Trigger

#### 23.3.4.6.1.1 Transmit FIFO Trigger

Table 23-149 lists the TX FIFO trigger level settings.

**Table 23-149. TX FIFO Trigger Level Setting Summary**

UART_SCR[6]	UART_TLR[3:0]	TX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit field (8,16, 32, or 56 spaces)
0	!= 0x0	Defined by the UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit field (from 4 to 60 spaces with a granularity of 4 spaces)
1	Value	Defined by the concatenated value of TX_FIFO_TRIG_DMA and TX_FIFO_TRIG (from 1 to 63 spaces with a granularity of 1 space) <b>Note:</b> The combination of TX_FIFO_TRIG_DMA = 0x0 and TX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of one space required). All zeros result in unpredictable behavior.

#### 23.3.4.6.1.2 Receive FIFO Trigger

Table 23-150 lists the RX FIFO trigger level settings.

**Table 23-150. RX FIFO Trigger Level Setting Summary**

UART_SCR[7]	UART_TLR[7:4]	RX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[7:6] RX_FIFO_TRIG bit field (8,16, 56, or 60 characters)
0	!= 0x0	Defined by the UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA bit field (from 4 to 60 characters with a granularity of 4 characters)
1	Value	Defined by the concatenated value of RX_FIFO_TRIG_DMA and RX_FIFO_TRIG (from 1 to 63 characters with a granularity of 1 character) <b>Note:</b> The combination of RX_FIFO_TRIG_DMA = 0x0 and RX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of one character required). All zeros result in unpredictable behavior.

The receive threshold is programmed using the UARTi.UART\_TCR[7:4] RX\_FIFO\_TRIG\_START and UARTi.UART\_TCR[3:0] RX\_FIFO\_TRIG\_HALT bit fields:

- Trigger levels from 0 to 60 bytes are available with a granularity of 4. (trigger level = 4 x [4-bit register value]).
- To ensure correct device operation, ensure that RX\_FIFO\_TRIG\_HALT > RX\_FIFO\_TRIG when auto-RTS is enabled.

$$\text{Delay} = [4 + 16 \times (1 + \text{CHAR\_LENGTH} + \text{Parity} + \text{Stop } 0.5)] \times \text{Baud\_rate} + 4 \times \text{FCLK}$$

**NOTE:** The RTS signal is deasserted after the UART module receives the data over RX\_FIFO\_TRIG\_HALT. Delay means how long the UART module takes to deassert the RTS signal after reaching RX\_FIFO\_TRIG\_HALT.

- In FIFO interrupt mode with flow control, ensure that the trigger level to HALT transmission is greater than or equal to the RX FIFO trigger level (the UARTi.UART\_TCR[7:4] RX\_FIFO\_TRIG\_START bit field or the UARTi.UART\_FCR[7:6] RX\_FIFO\_TRIG bit field); otherwise, FIFO operation stalls. In FIFO DMA mode with flow control, this concept does not exist, because a DMA request is sent when a byte is received.

#### 23.3.4.6.2 FIFO Interrupt Mode

In FIFO interrupt mode (the FIFO control register UARTi.UART\_FCR[0] FIFO\_EN bit is set to 1 and

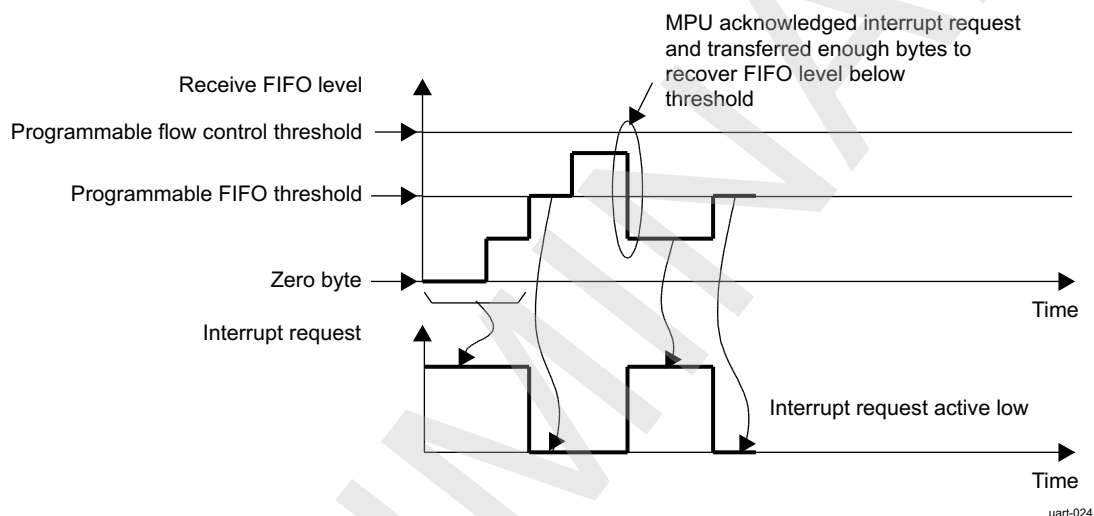
relevant interrupts are enabled by the UARTi.UART\_JER register), an interrupt signal informs the processor of the status of the receiver and transmitter. These interrupts are raised when the receive/TX FIFO threshold (the UARTi.UART\_TLR[7:4] RX\_FIFO\_TRIG\_DMA and UARTi.UART\_TLR[3:0] TX\_FIFO\_TRIG\_DMA bit fields or the UARTi.UART\_FCR[7:6] RX\_FIFO\_TRIG and UARTi.UART\_FCR[5:4] TX\_FIFO\_TRIG bit fields, respectively) is reached.

The interrupt signals instruct the MPU to transfer data to the destination (from the UART in receive mode and/or from any source to the UART FIFO in transmit mode).

When UART flow control is enabled with interrupt capabilities, the UART flow control FIFO threshold (the UARTi.UART\_TCR[3:0] RX\_FIFO\_TRIG\_HALT bit field) must be greater than or equal to the RX FIFO threshold.

Figure 23-74 shows the generation of the RX FIFO interrupt request.

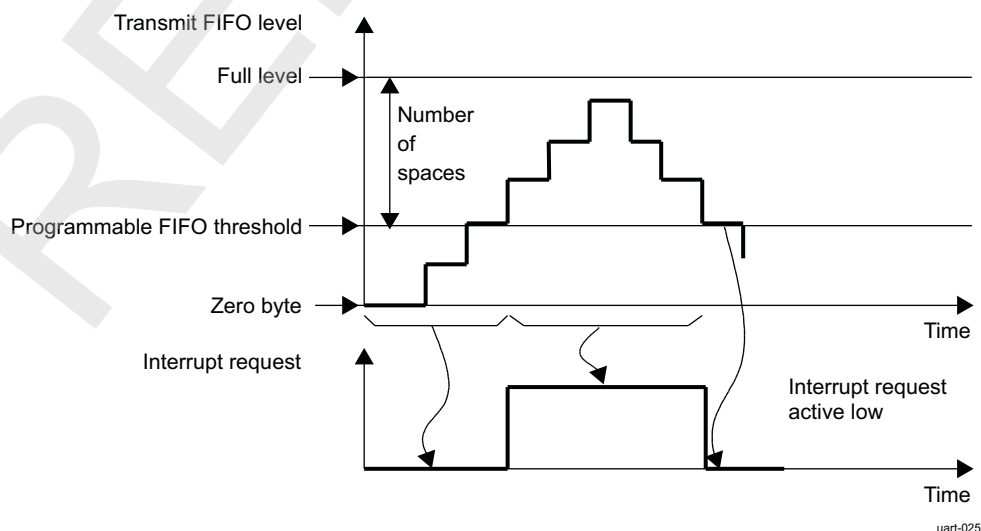
**Figure 23-74. Receive FIFO Interrupt Request Generation**



In receive mode, no interrupt is generated until the RX FIFO reaches its threshold. Once low, the interrupt can be deasserted only when the MPU has handled enough bytes to put the FIFO level below threshold. The flow control threshold is set at a higher value than the FIFO threshold.

Figure 23-75 shows the generation of the TX FIFO interrupt request.

**Figure 23-75. Transmit FIFO Interrupt Request Generation**





In transmit mode, an interrupt request is automatically asserted when the TX FIFO is empty. This request is deasserted when the TX FIFO crosses the threshold level. The interrupt line is deasserted until a sufficient number of elements is transmitted to go below the TX FIFO threshold.

#### **23.3.4.6.3 FIFO Polled Mode Operation**

In FIFO polled mode (the UARTi.UART\_FCR[0] FIFO\_EN bit is set to 0 and the relevant interrupts are disabled by the UARTi.UART\_IER register), the status of the receiver and transmitter can be checked by polling the line status register (UARTi.UART\_LSR).

This mode is an alternative to the FIFO interrupt mode of operation in which the status of the receiver and transmitter is automatically determined by sending interrupts to the MPU.

#### **23.3.4.6.4 FIFO DMA Mode Operation**

Although DMA operation includes four modes (DMA modes 0/1/2/3), the information in [Table 23-144](#) assumes that mode 1 is used. (Mode 2 and mode 3 are legacy modes that use only one DMA request for each module.)

In mode 2, the remaining DMA request is used for RX. In mode 3, the remaining DMA request is used for TX.

DMA requests in mode 2 and mode 3 use the following signals:

- S\_DMA\_48
- S\_DMA\_50
- S\_DMA\_52/D\_DMA\_10
- S\_DMA\_54

The following signals are not used by the module in mode 2 and mode 3:

- S\_DMA\_49
- S\_DMA\_51
- S\_DMA\_53/D\_DMA\_11
- S\_DMA\_55

These signals can be selected as follows:

- When the UARTi.UART\_SCR[0] DMA\_MODE\_CTL bit is set to 0, setting the UARTi.UART\_FCR[3] DMA\_MODE\_CTL bit to 0 enables DMA mode 0. Setting the DMA\_MODE bit to 1 enables DMA mode 1.
- When the DMA\_MODE\_CTL bit is set to 1, the UARTi.UART\_FCR[2:1] DMA\_MODE\_2 bit field determines DMA mode 0 to mode 3 based on the supplementary control register (UART\_SCR) description.

For example:

- If no DMA operation is desired, set the DMA\_MODE\_CTL bit to 1 and the DMA\_MODE\_2 bit field to 0x0. (The DMA\_MODE bit is discarded.)
- If DMA mode 1 is desired, set the DMA\_MODE\_CTL bit to 0 and the DMA\_MODE bit to 1, or set the DMA\_MODE\_CTL bit to 1 and the DMA\_MODE\_2 bit field to 01. (The DMA\_MODE bit is discarded.)

If the FIFOs are disabled (the UARTi.UART\_FCR[0] FIFO\_EN bit = 0), the DMA occurs in single-character transfers.

When DMA mode 0 is programmed, the signals associated with DMA operation are not active.

Depending on [UART\\_MDR3\[2\] SET\\_DMA\\_TX\\_THRESHOLD](#) the threshold can be programmed different ways:

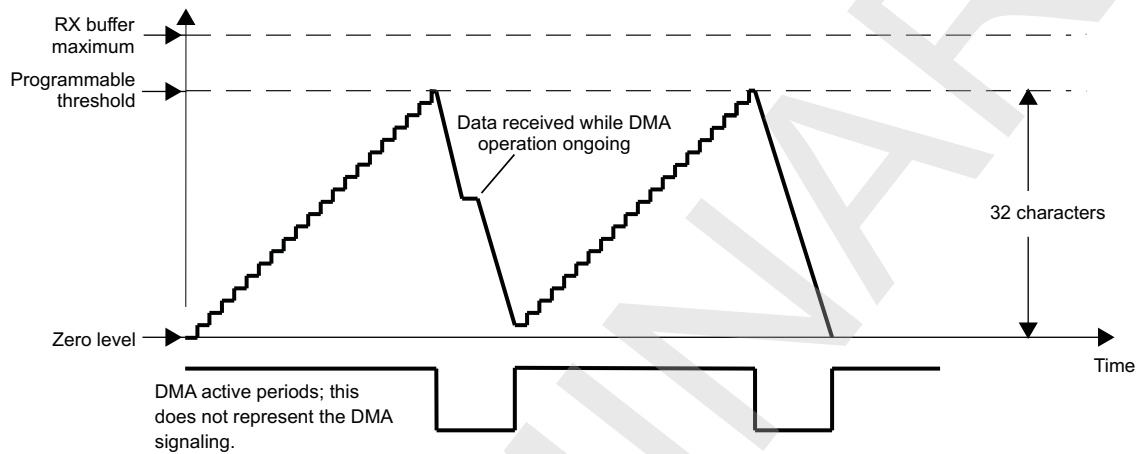
- SET\_TX\_DMA\_THRESHOLD = 1:  
The threshold value will be the value of the [UART\\_TX\\_DMA\\_THRESHOLD](#) register. If TX\_DMA\_THRESHOLD + TX trigger spaces > 64, then the default method of threshold will be used: threshold value = TX FIFO size.
- SET\_TX\_DMA\_THRESHOLD = 0:

The threshold value = TX FIFO size TX trigger space. The TX DMA line will be asserted if the TX FIFO level is lower than the threshold. It will remain asserted till TX trigger spaces number of bytes are written into the FIFO. Then the DMA line will deassert and the FIFO level will be compared with the threshold value.

#### 23.3.4.6.4.1 DMA Transfers (DMA Mode 1, 2, or 3)

Figure 23-76 through Figure 23-79 show the supported DMA operations.

**Figure 23-76. Receive FIFO DMA Request Generation (32 Characters)**

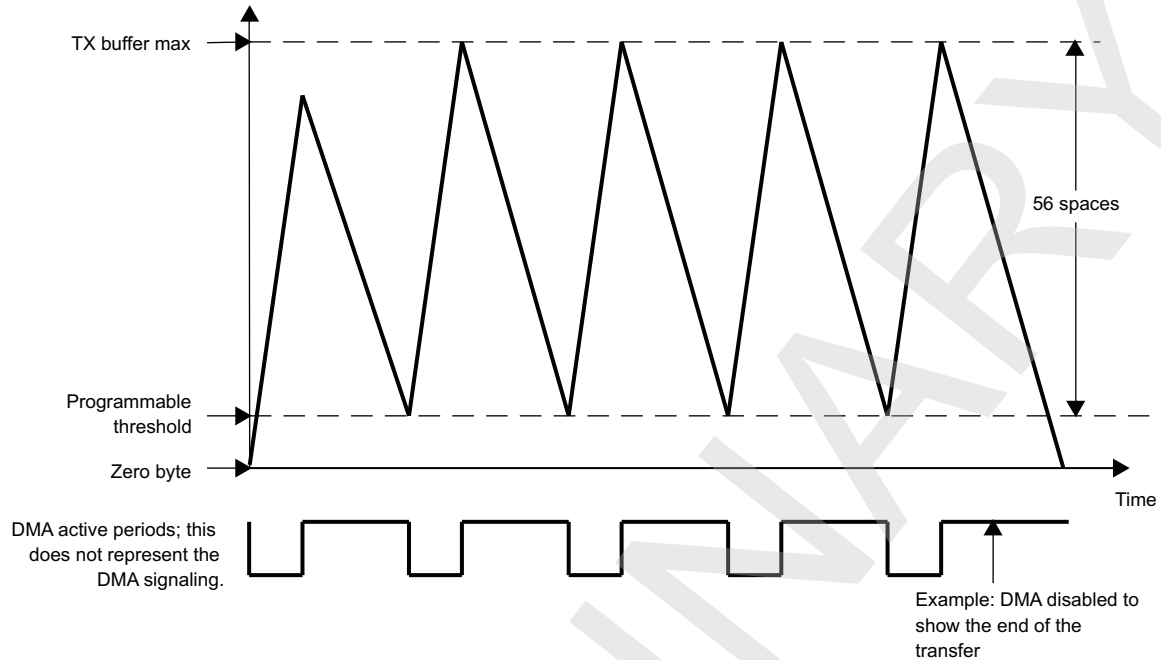


uart-026

In receive mode, a DMA request is generated when the RX FIFO reaches its threshold level defined in the trigger level register (UARTi.UART\_TLR). This request is deasserted when the number of bytes defined by the threshold level is read by the system DMA (sDMA).

In transmit mode, a DMA request is automatically asserted when the TX FIFO is empty. This request is deasserted when the number of bytes defined by the number of spaces in the trigger level register (UARTi.UART\_TLR) is written by the sDMA. If an insufficient number of characters is written, the DMA request stays active.

Figure 23-77. Transmit FIFO DMA Request Generation (56 Spaces)



uart-027

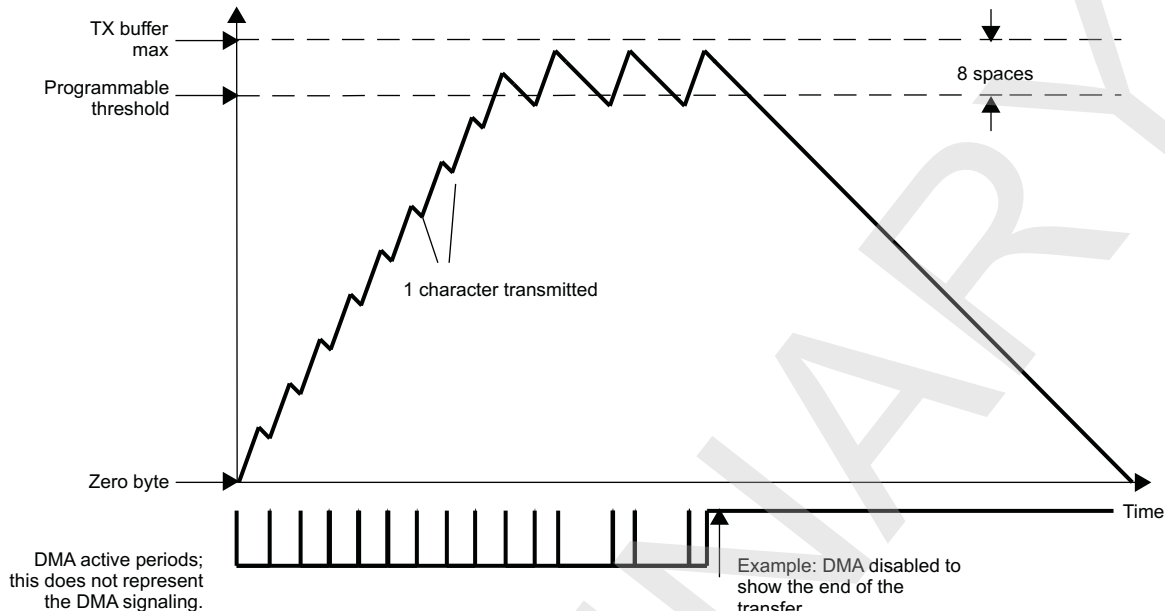
The DMA request is again asserted if the FIFO can receive the number of bytes defined by the `UARTi.UART_TLR` register.

The threshold can be programmed in a number of ways. Figure 23-77 shows a DMA transfer operating with a space setting of 56 that can arise from using the auto settings in the `UARTi.UART_FCR[5:4]` `TX_FIFO_TRIG` bit field or the `UARTi.UART_TLR[3:0]` `TX_FIFO_TRIG_DMA` bit field concatenated with the `TX_FIFO_TRIG` bit field.

The setting of 56 spaces in the UART/IrDA/CIR module must correlate with the settings of the sDMA so that the buffer does not overflow (program the DMA request size of the local host controller to equal the number of spaces in the UART/IrDA/CIR module).

Figure 23-78 shows an example with eight spaces to show the buffer level crossing the space threshold. The local host DMA controller settings must correspond to those of the UART/IrDA/CIR module.

**Figure 23-78. Transmit FIFO DMA Request Generation (8 Spaces)**



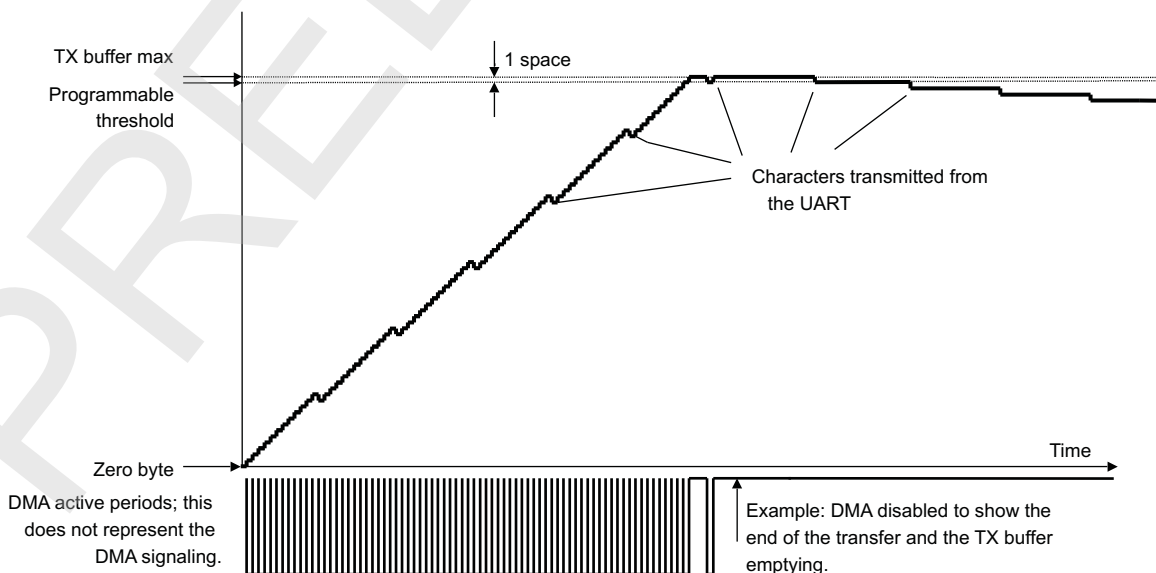
uart-028

The final example shows the setting of one space that uses the DMA for each transfer of one character to the transmit buffer (see [Figure 23-79](#)). The buffer is filled faster than the baud rate at which data is transmitted to the TX pin. Eventually, the buffer is completely full and the DMA operations stop transferring data to the transmit buffer.

On two occasions, the buffer holds the maximum amount of data words; shortly after this, the DMA is disabled to show the slower transmission of the data words to the TX pin. Eventually, the buffer is emptied at the rate specified by the baud rate settings of the UARTi.UART\_DLL and the UARTi.UART\_DLH registers.

The DMA settings must correspond to the system local host DMA controller settings to ensure correct operation of this logic.

**Figure 23-79. Transmit FIFO DMA Request Generation (1 Space)**

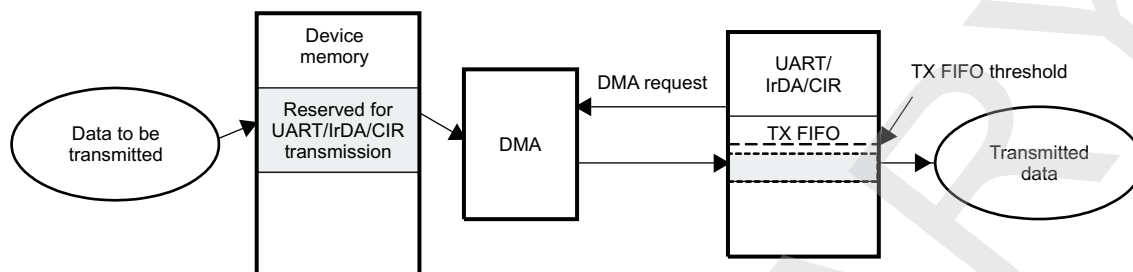


uart-029

### 23.3.4.6.4.2 DMA Transmission

Figure 23-80 shows DMA transmission.

Figure 23-80. DMA Transmission



uart-030

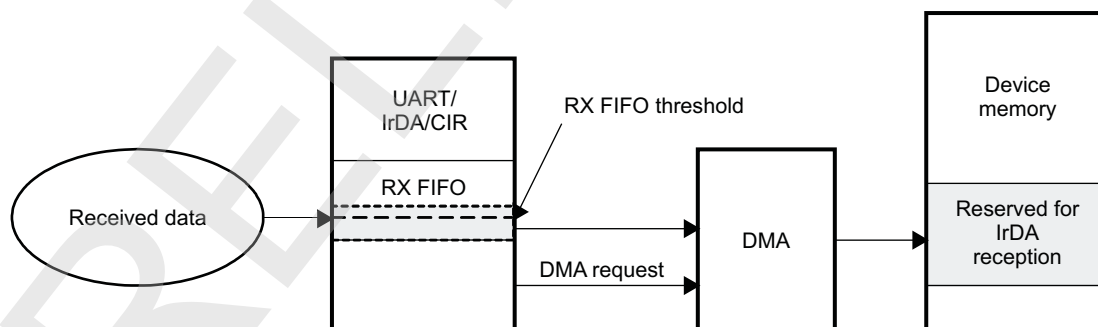
1. Data to be transmitted are put in the device memory reserved for UART/IrDA/CIR transmission by the DMA:
  - (a) Until the TX FIFO trigger level is not reached, a DMA request is generated.
  - (b) An element (1 byte) is transferred from the SDRAM to the TX FIFO at each DMA request (DMA element synchronization).
2. Data in the TX FIFO are automatically transmitted.
3. The end of the transmission is signaled by the UARTi.UART\_THR empty (TX FIFO empty).

**NOTE:** In IrDA mode, the transmission does not end immediately after the TX FIFO empties, at which point there are still the last data byte, the CRC field, and the stop flag to be transmitted; thus, the end of transmission is a few milliseconds after the UARTi.UART\_THR register empties.

### 23.3.4.6.4.3 DMA Reception

Figure 23-81 shows DMA reception.

Figure 23-81. DMA Reception



uart-031

1. Enable the reception.
2. Received data are put in the RX FIFO.
3. Data are transferred from the RX FIFO to the device memory by the DMA:
  - (a) At each received byte, the RX FIFO trigger level (one character) is reached and a DMA request is generated.
  - (b) An element (1 byte) is transferred from the RX FIFO to the SDRAM at each DMA request (DMA element synchronization).
4. The end of the reception is signaled by the EOF interrupt.

### 23.3.4.7 Mode Selection

#### 23.3.4.7.1 Register Access Modes

##### 23.3.4.7.1.1 Operational Mode and Configuration Modes

Register access depends on the register access mode, although register access modes are not correlated to functional mode selection. Three different modes are available:

- Operational mode
- Configuration mode A
- Configuration mode B

Operational mode is the selected mode when the function is active; serial data transfer can be performed in this mode.

Configuration mode A and configuration mode B are used during module initialization steps. These modes enable access to configuration registers, which are hidden in the operational mode. The modes are used when the module is inactive (no serial data transfer processed) and only for initialization or reconfiguration of the module.

The value of the UARTi.UART\_LCR register determines the register access mode (see [Table 23-151](#)).

**Table 23-151. UART/IrDA/CIR Register Access Mode Programming (Using UART\_LCR)**

Mode	Condition
Configuration mode A	UART_LCR[7] = 0x1 and UART_LCR[7:0] != 0xBF
Configuration mode B	UART_LCR[7] = 0x1 and UART_LCR[7:0] = 0xBF
Operational mode	UART_LCR[7] = 0x0

##### 23.3.4.7.1.2 Register Access Submode

In each access register mode (operational mode or configuration mode A/B), some register accesses are conditional on the programming of a submode (MSR\_SPR, TCR\_TLR, and XOFF). These registers are identified in [Section 23.3.6](#), *UART/IrDA/CIR Register Manual*.

[Table 23-152](#) through [Table 23-154](#) summarize the register access submodes.

**Table 23-152. Subconfiguration Mode A Summary**

Mode	Condition
MSR_SPR	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

**Table 23-153. Subconfiguration Mode B Summary**

Mode	Condition
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1
XOFF	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)

**Table 23-154. Suboperational Mode Summary**

Mode	Condition
MSR_SPR	UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

##### 23.3.4.7.1.3 Registers Available for the Register Access Modes

[Table 23-155](#) lists the names of the register hits in each access register mode. Gray shading indicates that the register does not depend on the register access mode (available in all modes).

**Table 23-155. UART/IrDA/CIR Register Access Mode Overview**

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER	UART_IER
0x008	UART_IIR	UART_FCR	UART_EFR	UART_EFR	UART_IIR	UART_FCR
0x00C	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR
0x010	UART_MCR	UART_MCR	UART_XON1_ADD R1	UART_XON1_AD DR1	UART_MCR	UART_MCR
0x014	UART_LSR	–	UART_XON2_ADD R2	UART_XON2_AD DR2	UART_LSR	–
0x018	UART_MSR ( <sup>1</sup> )/UART_TCR ( <sup>2</sup> )	UART_TCR ( <sup>2</sup> )	UART_TCR ( <sup>2</sup> )/UART_XOFF1 ( <sup>3</sup> )	UART_TCR ( <sup>2</sup> )/UART_XOFF1 ( <sup>3</sup> )	UART_MSR ( <sup>1</sup> )/UART_TCR ( <sup>2</sup> )	UART_TCR ( <sup>2</sup> )
0x01C	UART_SPR ( <sup>1</sup> )/UART_TLR ( <sup>2</sup> )	UART_SPR ( <sup>1</sup> )/UART_TLR ( <sup>2</sup> )	UART_TLR ( <sup>2</sup> )/UART_XOFF2 ( <sup>3</sup> )	UART_TLR ( <sup>2</sup> )/UART_XOFF2 ( <sup>3</sup> )	UART_SPR ( <sup>1</sup> )/UART_TLR ( <sup>2</sup> )	UART_SPR ( <sup>1</sup> )/UART_TLR ( <sup>2</sup> )
0x020	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL
0x02C	UART_RESUM E	UART_TXFLH	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH
0x030	UART_SFREG L	UART_RXFLL	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL
0x034	UART_SFREG H	UART_RXFLH	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH
0x038	UART_UASR	–	UART_UASR	–	UART_BLR	UART_BLR
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER
0x060	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS
0x064	UART_RXFIFO _LVL	UART_RXFIFO_ LVL	UART_RXFIFO_LV LVL	UART_RXFIFO_L VL	UART_RXFIFO_LV L	UART_RXFIFO_ LVL
0x068	UART_TXFIFO _LVL	UART_TXFIFO_ LVL	UART_TXFIFO_LV LVL	UART_TXFIFO_L VL	UART_TXFIFO_LV L	UART_TXFIFO_ LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_ SEL	UART_FREQ_S EL	UART_FREQ_SEL	UART_FREQ_SE L	UART_FREQ_SEL	UART_FREQ_ SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DM A_THRESHOLD	UART_TX_DMA_ _THRESHOLD	UART_TX_DMA_TH RESHOLD	UART_TX_DMA_ THRESHOLD	UART_TX_DMA_T HRESHOLD	UART_TX_DM A_THRESHOLD

(<sup>1</sup>) MSR\_SPR mode is active (see Section 23.3.4.7.1.2, Register Access Submode)

(<sup>2</sup>) TCR\_TLR mode is active (see Section 23.3.4.7.1.2, Register Access Submode)

(<sup>3</sup>) XOFF mode is active (see Section 23.3.4.7.1.2, Register Access Submode)

### 23.3.4.7.2 UART/IrDA (SIR, MIR, FIR)/CIR Mode Selection

To select a mode, set the UARTi.UART\_MDR1[2:0] MODE\_SELECT bit field (see [Table 23-156](#)).

**Table 23-156. UART Mode Selection**

Value	Mode
0x0:	UART 16x mode
0x1:	SIR mode (UART3 only)
0x2:	UART 16x auto-baud
0x3:	UART 13x mode
0x4:	MIR mode (UART3 only)
0x5:	FIR mode (UART3 only)
0x6:	CIR mode (UART3 only)

MODE\_SELECT is effective when the module is in operational mode (see [Section 23.3.4.7.1](#), *Register Access Modes*).

#### 23.3.4.7.2.1 Registers Available for the UART Function

Only the registers listed in [Table 23-157](#) are used for the UART function.

**Table 23-157. UART Mode Register Overview<sup>(1) (2)</sup>**

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER(UART)	UART_IER(UA RT)
0x008	UART_IIR	UART_FCR	UART_EFR[4]	UART_EFR[4]	UART_IIR(UART)	UART_FCR(UA RT)
0x00C	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR
0x010	UART_MCR	UART_MCR	UART_XON1_ADD R1	UART_XON1_AD DR1	UART_MCR	UART_MCR
0x014	UART_LSR(UA RT)	–	UART_XON2_ADD R2	UART_XON2_AD DR2	UART_LSR(UART)	–
0x018	UART_MSR/U ART_TCR	UART_TCR	UART_XOFF1/UAR T_TCR	UART_XOFF1/U ART_TCR	UART_MSR/UART _TCR	UART_TCR
0x01C	UART_TLR/U ART_SPR	UART_TLR/U ART_SPR	UART_TLR/UART_ XOFF2	UART_TLR/UAR T_XOFF2	UART_TLR/UART_ SPR	UART_TLR/U ART_SPR
0x020	UART_MDR1	UART_MDR1[2: 0]	UART_MDR1[2:0]	UART_MDR1[2:0]	UART_MDR1[2:0]	UART_MDR1[2 :0]
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	–	–	–	–	–	–
0x02C	–	–	–	–	–	–
0x030	–	–	–	–	–	–
0x034	–	–	–	–	–	–
0x038	UART_UASR	–	UART_UASR	–	–	–
0x03C	–	–	–	–	–	–
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	–	–

<sup>(1)</sup> REGISTER\_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions (described separately in [Section 23.3.6](#), *UART/IrDA/CIR Register Manual*).

<sup>(2)</sup> REGISTER\_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.



**Table 23-157. UART Mode Register Overview<sup>(1) (2)</sup> (continued)**

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER
0x060	–	–	–	–	–	–
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

#### 23.3.4.7.2.2 Registers Available for the IrDA Function (UART3 Only)

Only the registers listed in [Table 23-158](#) are used for the IrDA function.

**Table 23-158. IrDA Mode Register Overview<sup>(1) (2)</sup>**

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER(IrDA )	UART_IER(IrDA )
0x008	UART_IIR	UART_FCR	UART_EFR[4]	UART_EFR[4]	UART_IIR(IrDA )	UART_FCR(IrDA )
0x00C	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]
0x010	–	–	UART_XON1_ADD R1	UART_XON1_ADD R1	–	–
0x014	UART_LSR(IrDA )	–	UART_XON2_ADD R2	UART_XON2_ADD R2	UART_LSR(IrDA )	–
0x018	UART_MSR/UART_TCR	UART_TCR	UART_TCR	UART_TCR	UART_MSR/UART_TCR	UART_TCR
0x01C	UART_TLR/UART_SPR	UART_TLR/UART_SPR	UART_TLR	UART_TLR	UART_TLR/UART_SPR	UART_TLR/UART_SPR
0x020	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL
0x02C	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH

<sup>(1)</sup> REGISTER\_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions (described separately in [Section 23.3.6, UART/IrDA/CIR Register Manual](#)).

<sup>(2)</sup> REGISTER\_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.

**Table 23-158. IrDA Mode Register Overview<sup>(1) (2)</sup> (continued)**

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x030	UART_SFREG L	UART_RXFLL	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL
0x034	UART_SFREG H	UART_RXFLH	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH
0x038	–	–	–	–	UART_BLR	UART_BLR
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER[6: 4]	UART_WER[6:4 ]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6: 4]
0x060	–	–	–	–	–	–
0x064	UART_RXFIFO_ LVL	UART_RXFIFO_ LVL	UART_RXFIFO_LV L	UART_RXFIFO_L VL	UART_RXFIFO_LV L	UART_RXFIFO_ LVL
0x068	UART_TXFIFO_ LVL	UART_TXFIFO_ LVL	UART_TXFIFO_LV L	UART_TXFIFO_L VL	UART_TXFIFO_LV L	UART_TXFIFO_ LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_ SEL	UART_FREQ_S EL	UART_FREQ_SEL	UART_FREQ_SE L	UART_FREQ_SEL	UART_FREQ_ SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DM A_THRESHOLD	UART_TX_DMA_ _THRESHOLD	UART_TX_DMA_TH RESHOLD	UART_TX_DMA_ THRESHOLD	UART_TX_DMA_T HRESHOLD	UART_TX_DM A_THRESHOLD

### 23.3.4.7.2.3 Registers Available for the CIR Function (UART3 Only)

Only the registers listed in [Table 23-159](#) are used for the CIR function.

**Table 23-159. CIR Mode Register Overview<sup>(1) (2)</sup>**

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	–	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER(CIR)	UART_IER(CI R)
0x008	UART_IIR	UART_FCR	UART_EFR	UART_EFR	UART_IIR(CIR)	UART_FCR(CI R)
0x00C	UART_LCR	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]
0x010	–	–	–	–	–	–
0x014	UART_LSR(IrD A)	–	–	–	UART_LSR(IrDA)	–

<sup>(1)</sup> REGISTER\_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions (described separately in [Section 23.3.6, UART/IrDA/CIR Register Manual](#)).

<sup>(2)</sup> REGISTER\_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.

Table 23-159. CIR Mode Register Overview<sup>(1) (2)</sup> (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x018	UART_MSR/UART_TCR	UART_TCR	UART_TCR	UART_TCR	UART_MSR/UART_TCR	UART_TCR
0x01C	UART_TLR/UART_SPR	UART_TLR/UART_SPR	UART_TLR	UART_TLR	UART_TLR/UART_SPR	UART_TLR/UART_SPR
0x020	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	–	–	–	–	–	–
0x02C	UART_RESUME	–	UART_RESUME	–	UART_RESUME	–
0x030	–	–	–	–	–	–
0x034	–	–	–	–	–	–
0x038	–	–	–	–	–	–
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]
0x060	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

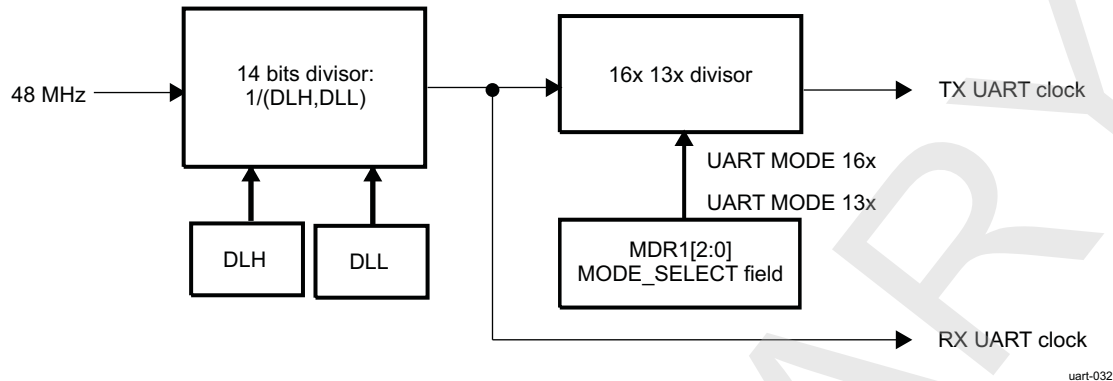
### 23.3.4.8 Protocol Formatting

#### 23.3.4.8.1 UART Mode

##### 23.3.4.8.1.1 UART Clock Generation: Baud Rate Generation

The UART function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate.

Figure 23-82 shows the baud rate generator and associated controls.

**Figure 23-82. Baud Rate Generation**

uart-032

**CAUTION**

Before initializing or modifying clock parameter controls (UARTi.UART\_DLH, UARTi.UART\_DLL), MODE\_SELECT = DISABLE (UARTi.UART\_MDR1[2:0]) must be set to 0x7. Failure to observe this rule can result in unpredictable module behavior.

**23.3.4.8.1.2 Choosing the Appropriate Divisor Value**

Two divisor values are:

- UART 16x mode: Divisor value = Operating frequency/(16 x baud rate)
- UART 13x mode: Divisor value = Operating frequency/(13 x baud rate)

Table 23-160 describes the UART baud rate settings.

**Table 23-160. UART Baud Rate Settings (48-MHz Clock)**

Baud Rate	Baud Multiple	DLH,DLL (Decimal)	DLH,DLL (Hex)	Actual Baud Rate	Error (%)
0.3 kbps	16x	10000	0x27, 0x10	0.3 kbps	0
0.6 kbps	16x	5000	0x13, 0x88	0.6 kbps	0
1.2 kbps	16x	2500	0x09, 0xC4	1.2 kbps	0
2.4 kbps	16x	1250	0x04, 0xE2	2.4 kbps	0
4.8 kbps	16x	625	0x02, 0x71	4.8 kbps	0
9.6 kbps	16x	312	0x01, 0x38	9.6153 kbps	+0.16
14.4 kbps	16x	208	0x00, 0xD0	14.423 kbps	+0.16
19.2 kbps	16x	156	0x00, 0x9C	19.231 kbps	+0.16
28.8 kbps	16x	104	0x00, 0x68	28.846 kbps	+0.16
38.4 kbps	16x	78	0x00, 0x4E	38.462 kbps	+0.16
57.6 kbps	16x	52	0x00, 0x34	57.692 kbps	+0.16
115.2 kbps	16x	26	0x00, 0x1A	115.38 kbps	+0.16
230.4 kbps	16x	13	0x00, 0x0D	230.77 kbps	+0.16
460.8 kbps	13x	8	0x00, 0x08	461.54 kbps	+0.16
921.6 kbps	13x	4	0x00, 0x04	923.08 kbps	+0.16
1.843 Mbps	13x	2	0x00, 0x02	1.846 Mbps	+0.16
3.6884 Mbps	13x	1	0x00, 0x01	3.6923 Mbps	+0.16

**23.3.4.8.1.3 UART Data Formatting**

The UART can use hardware flow control to manage transmission and reception. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the RTS output and CTS input signals.

The UART is enhanced with the autobauding function. In control mode, autobauding lets the speed, the number of bits per character, and the parity selected be set automatically.

**23.3.4.8.1.3.1 Frame Formatting**

When autobauding is not used, frame format attributes must be defined in the UARTi.UART\_LCR register.

Character length is specified using the UARTi.UART\_LCR[1:0] CHAR\_LENGTH bit field.

The number of stop-bits is specified using the UARTi.UART\_LCR[2] NB\_STOP bit.

The parity bit is programmed using the UARTi.UART\_LCR[5:3] PARITY\_EN, PARITY\_TYPE\_1, and PARITY\_TYPE\_2 bit fields (see Table 23-161).

**Table 23-161. UART Parity Bit Encoding**

PARITY_EN	PARITY_TYPE_1	PARITY_TYPE_2	Parity
0	N/A	N/A	No parity
1	0	0	Odd parity
1	1	0	Even parity
1	0	1	Forced 1
1	1	1	Forced 0

**23.3.4.8.1.3.2 Hardware Flow Control**

Hardware flow control is composed of auto-CTS and auto-RTS. Auto-CTS and auto-RTS can be enabled and disabled independently by programming the UARTi.UART\_EFR[7:6] AUTO\_CTS\_EN and AUTO\_RTS\_EN bit fields, respectively.

With auto-CTS, uarti\_cts must be active before the module can transmit data.

Auto-RTS activates the uarti\_rts output only when there is enough room in the RX FIFO to receive data. It deactivates the uarti\_rts output when the RX FIFO is sufficiently full. The HALT and RESTORE trigger levels in the UARTi.UART\_TCR register determine the levels at which uarti\_rts is activated and deactivated.

If auto-CTS and auto-RTS are enabled, data transmission does not occur unless the RX FIFO has empty space. Thus, overrun errors are eliminated during hardware flow control. If auto-CTS and auto-RTS are not enabled, overrun errors occur if the transmit data rate exceeds the RX FIFO latency.

- Auto-RTS:

Auto-RTS data flow control originates in the receiver block. The RX FIFO trigger levels used in auto-RTS are stored in the UARTi.UART\_TCR register. uarti\_rts is active if the RX FIFO level is below the HALT trigger level in the UARTi.UART\_TCR[3:0] RX\_FIFO\_TRIG\_HALT bit field. When the RX FIFO HALT trigger level is reached, uarti\_rts is deasserted. The sending device (for example, another UART) can send an additional byte after the trigger level is reached because it may not recognize the deassertion of RTS until it begins sending the additional byte.

uarti\_rts is automatically reasserted when the RX FIFO reaches the RESUME trigger level programmed by the UARTi.UART\_TCR[7:4] RX\_FIFO\_TRIG\_START bit field. This reassertion requests the sending device to resume transmission.

In this case, uarti\_rts is an active-low signal.

- Auto-CTS:

The transmitter circuitry checks uarti\_cts before sending the next data byte. When uarti\_cts is active, the transmitter sends the next byte. To stop the transmitter from sending the next byte, uarti\_cts must be deasserted before the middle of the last stop-bit currently sent.

The auto-CTS function reduces interrupts to the host system. When auto-CTS flow control is enabled,

the `uarti_cts` state changes do not have to trigger host interrupts because the device automatically controls its own transmitter. Without auto-CTS, the transmitter sends any data present in the transmit FIFO, and a receiver overrun error can result.

In this case, `uarti_cts` is an active-low signal.

### 23.3.4.8.1.3.3 Software Flow Control

Software flow control is enabled through the enhanced feature register (UARTi.UART\_EFR) and the modem control register (UARTi.UART\_MCR). Different combinations of software flow control can be enabled by setting different combinations of the UARTi.UART\_EFR[3:0] bit field (see Table 23-162).

Two other enhanced features relate to software flow control:

- XON any function (UARTi.UART\_MCR[5]): Operation resumes after receiving any character after the XOFF character is recognized. If special character detect is enabled and special character is received after XOFF1, it does not resume transmission. The special character is stored in the RX FIFO.

---

**NOTE:** The XON-any character is written into the RX FIFO even if it is a software flow character.

---

- Special character (UARTi.UART\_EFR[5]): Incoming data is compared to XOFF2. When the special character is detected, the XOFF interrupt (UARTi.UART\_IIR[4]) is set, but it does not halt transmission. The XOFF interrupt is cleared by a read of UARTi.UART\_IIR. The special character is transferred to the RX FIFO. Special character does not work with XON2, XOFF2, or sequential XOFFs.

**Table 23-162. UART\_EFR[0:3] Software Flow Control Options**

Bit 3	Bit 2	Bit 1	Bit 0	TX, RX Software Flow Controls
0	0	X	X	No transmit flow control
1	0	X	X	Transmit XON1, XOFF1
0	1	X	X	Transmit XON2, XOFF2
1	1	X	X	Transmit XON1, XON2: XOFF1, XOFF2 <sup>(1)</sup>
X	X	0	0	No receive flow control
X	X	1	0	Receiver compares XON1, XOFF1
X	X	0	1	Receiver compares XON2, XOFF2
X	X	1	1	Receiver compares XON1, XON2: XOFF1, XOFF2 <sup>(1)</sup>

<sup>(1)</sup> In these cases, the XON1 and XON2 characters or the XOFF1 and XOFF2 characters must be transmitted/received sequentially with XON1/XOFF1 followed by XON2/XOFF2. XON1 is defined in the UARTi.UART\_XON1\_ADDR1[7:0] XON\_WORD1 bit field. XON2 is defined in the UARTi.UART\_XON2\_ADDR2[7:0] XON\_WORD2 bit field. XOFF1 is defined in the UARTi.UART\_XOFF1[7:0] XOFF\_WORD1 bit field. XOFF2 is defined in the UARTi.UART\_XOFF2[7:0] XOFF\_WORD2 bit field.

### 23.3.4.8.1.3.3.1 Receive (RX)

When software flow control operation is enabled, the UART compares incoming data with XOFF1/2 programmed characters (in certain cases, XOFF1 and XOFF2 must be received sequentially). When the correct XOFF characters are received, transmission stops after transmission of the current character completes. XOFF detection also sets UARTi.UART\_IIR[4] (if enabled by UARTi.UART\_IER[5]) and causes the interrupt line to go low.

To resume transmission, an XON1/2 character must be received (in certain cases, XON1 and XON2 must be received sequentially). When the correct XON characters are received, UARTi.UART\_IIR[4] is cleared and the XOFF interrupt disappears.

---

**NOTE:** When a parity, framing, or break error occurs while receiving a software flow control character, this character is treated as normal data and is written to the RX FIFO.

---

When XON-any and special character detect are disabled and software flow control is enabled, no valid XON or XOFF characters are written to the RX FIFO. For example, when UARTi.UART\_EFR[1:0] = 0x2, if XON1 and XOFF1 characters are received, they are not written to the RX FIFO.

When pairs of software flow characters are programmed to be received sequentially (UARTi.UART\_EFR[1:0] = 0x3), the software flow characters are not written to the RX FIFO if they are received sequentially. However, received XON1/XOFF1 characters must be written to the RX FIFO if the subsequent character is not XON2/XOFF2.

#### 23.3.4.8.1.3.3.2 Transmit (TX)

Two XOFF1 characters are transmitted when the RX FIFO passes the trigger level programmed by UARTi.UART\_TCR[3:0]. As soon as the RX FIFO reaches the trigger level programmed by UARTi.UART\_TCR[7:4], two XON1 characters are sent, so the data transfer recovers.

---

**NOTE:** If software flow control is disabled after an XOFF character is sent, the module transmits XON characters automatically to enable normal transmission.

---

The transmission of XOFF(s)/XON(s) follows the same protocol as transmission of an ordinary byte from the TX FIFO. This means that even if the word length is 5, 6, or 7 characters, the 5, 6, or 7 LSBs of XOFF1/2 and XON1/2 are transmitted. The 5, 6, or 7 bits of a character are seldom transmitted, but this function is included to maintain compatibility with earlier designs.

It is assumed that software flow control and hardware flow control are never enabled simultaneously.

#### 23.3.4.8.1.3.4 Autobauding Modes

In autobauding mode, the UART can extract transfer characteristics (speed, length, and parity) from an "at" (AT) command (ASCII code). These characteristics are used to receive data after an AT and to send data.

The following AT commands are valid:

AT	DATA	<CR>
at	DATA	<CR>
A/		
a/		

A line break during the acquisition of the sequence AT is not recognized, and an echo function is not implemented in hardware.

A/ and a/ are not used to extract characteristics, but they must be recognized because of their special meaning. A/ or a/ is used to instruct the software to repeat the last received AT command; therefore, an a/ always follows an AT, and transfer characteristics are not expected to change between an AT and an a/.

When a valid AT is received, AT and all subsequent data, including the final <CR> (0x0D), are saved to the RX FIFO. The autobaud state-machine waits for the next valid AT command. If an a/ (A/) is received, the a/ (A/) is saved in the RX FIFO and the state-machine waits for the next valid AT command.

On the first successful detection of the baud rate, the UART activates an interrupt to signify that the AT (upper or lower case) sequence is detected. The UARTi.UART\_UASR register reflects the correct settings for the baud rate detected. Interrupt activity can continue in this fashion when a subsequent character is received. Therefore, it is recommended that the software enable the RHR interrupt when using the autobaud mode.

The following settings are detected in autobaud mode with a module clock of 48 MHz:

- Speed:
  - 115.2K baud
  - 57.6K baud
  - 38.4K baud
  - 28.8K baud
  - 19.2K baud
  - 14.4K baud

- 9.6K baud
- 4.8K baud
- 2.4K baud
- 1.2K baud
- Length: 7 or 8 bits
- Parity: Odd, even, or space

---

**NOTE:** The combination of 7-bit character plus space parity is not supported.

---

Autobauding mode is selected when the UARTi.UART\_MDR1[2:0] MODE\_SELECT bit field is set to 0x2. In UART autobauding mode, UARTi.UART\_DLL, UARTi.UART\_DLH, and UARTi.UART\_LCR[5:0] bit field settings are not used; instead, UART\_UASR is updated with the configuration detected by the autobauding logic.

#### UART\_UASR Autobauding Status Register Use

This register is used to set up transmission according to the characteristics of the previous reception instead of the UARTi.UART\_LCR, UARTi.UART\_DLL, and UARTi.UART\_DLH registers when the UART is in autobauding mode.

To reset the autobauding hardware (to start a new AT detection) or to set the UART in standard mode (no autobaud), the UARTi.UART\_MDR1[2:0] MODE\_SELECT bit field must be set to reset state (0x7) and then to the UART in autobauding mode (0x2) or to the UART in standard mode (0x0).

Use limitation:

- Only 7- and 8-bit characters (5- and 6-bit not supported)
- 7-bit character with space parity not supported
- Baud rate between 1200 and 115,200 bps (10 possibilities)

#### 23.3.4.8.1.3.5 Error Detection

When the UARTi.UART\_LSR register is read, the UARTi.UART\_LSR[4:2] bit field reflects the error bits (BI: break condition, FE: framing error, PE: parity error) of the character at the top of the RX FIFO (the next character to be read). Therefore, reading the UARTi.UART\_LSR register and then reading the UARTi.UART\_RHR register identifies errors in a character.

Reading the UARTi.UART\_RHR register updates the BI, FE, and PE bits (see Table 23-146 for the UART mode interrupts).

The UARTi.UART\_LSR[7] RX\_FIFO\_STS bit is set when there is an error in the RX FIFO and is cleared only when no errors remain in the RX FIFO.

---

**NOTE:** Reading UARTi.UART\_LSR does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the UARTi.UART\_RHR register.

---

Reading the UARTi.UART\_LSR register clears the OE bit if it is set (see Table 23-146 for the UART mode interrupts).

#### 23.3.4.8.1.3.6 Overrun During Receive

Overrun during receive occurs if the RX state-machine tries to write data into the RX FIFO when it is already full. When overrun occurs, the device interrupts the MPU with UARTi.UART\_IIR[5:1] IT\_TYPE = 0x3 (receiver line status error) and discards the remaining portion of the frame.

Overrun also causes an internal flag to be set, which disables further reception. Before the next frame can be received, the MPU must:

- Reset the RX FIFO.
- Read the UARTi.UART\_RESUME register, which clears the internal flag.



### **23.3.4.8.1.3.7 Time-Out and Break Conditions**

#### **23.3.4.8.1.3.7.1 Time-Out Counter**

An RX idle condition is detected when the receiver line (uarti\_rx) is high for a time that equals 4x the programmed word length + 12 bits. uarti\_rx is sampled midway through each bit.

For sleep mode, the counter is reset when there is activity on uarti\_rx.

For the time-out interrupt, the counter counts only when there is data in the RX FIFO, and the count is reset when there is activity on uarti\_rx or when the UARTi.UART\_RHR register is read.

#### **23.3.4.8.1.3.7.2 Break Condition**

When a break condition occurs, uarti\_tx is pulled low. A break condition is activated by setting the UARTi.UART\_LCR[6] BREAK\_EN bit. The break condition is not aligned on word stream (a break condition can occur in the middle of a character). The only way to send a break condition on a full character is:

1. Reset the TX FIFO (if enabled).
2. Wait for the transmit shift register to empty (UARTi.UART\_LSR[6] TX\_SR\_E = 1).
3. Take a guard time according to stop-bit definition.
4. Set the BREAK\_EN bit to 1.

The break condition is asserted while the BREAK\_EN bit is set to 1.

The time-out counter and break condition apply only to UART modem operation and not to IrDA/CIR mode operation.

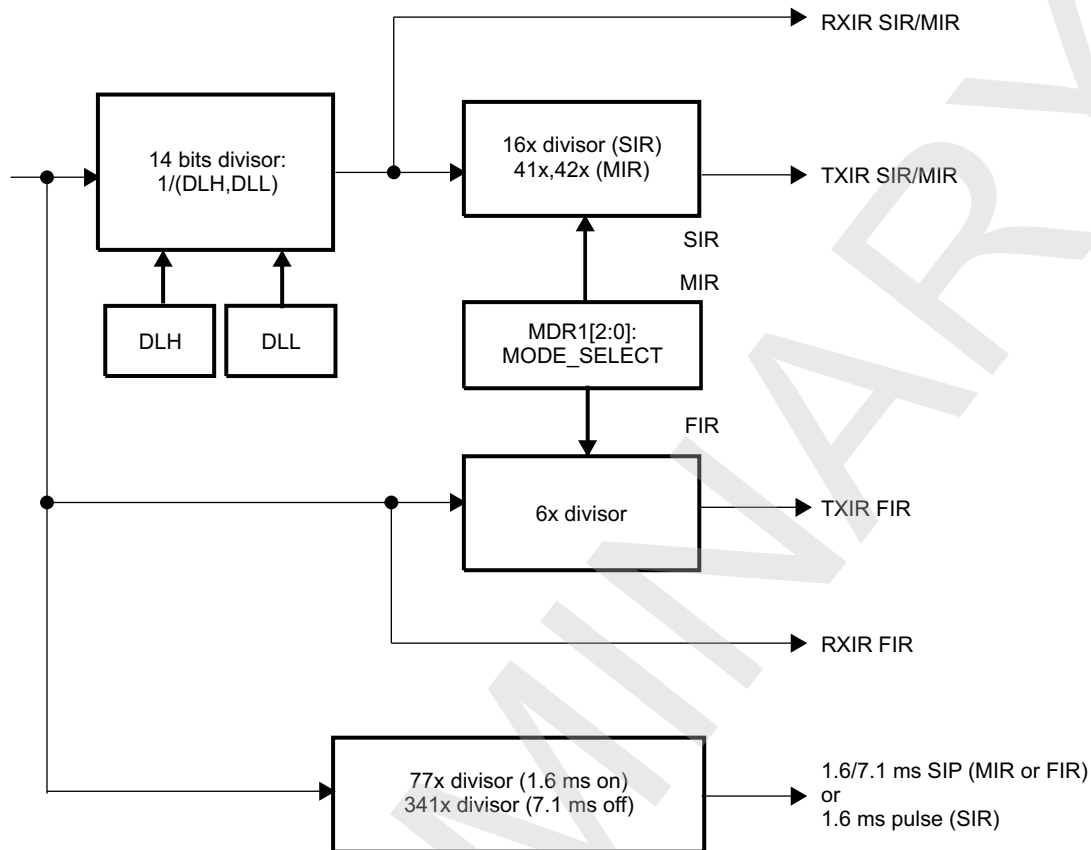
### **23.3.4.8.2 IrDA Mode (UART3 Only)**

#### **23.3.4.8.2.1 IrDA Clock Generation: Baud Generator**

The IrDA function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate.

Figure 23-83 shows the baud rate generator and associated controls.

Figure 23-83. Baud Rate Generator



uart-033

**CAUTION**

Before trying to initialize or modify clock parameter controls (UARTi.UART\_DLH, UARTi.UART\_DLL), MODE\_SELECT=DISABLE (UARTi.UART\_MDR1[2:0] must be set to 0x7.

Failure to observe this rule can result in unpredictable module behavior.

**23.3.4.8.2.2 Choosing the Appropriate Divisor Value**

Three divisor values are:

- SIR mode: Divisor value = Operating frequency/(16x baud rate)
- MIR mode: Divisor value = Operating frequency/(41x/42x baud rate)
- FIR mode: Divisor value = None

Table 23-163 lists the IrDA baud rate settings.

Table 23-163. IrDA Baud Rate Settings

Baud Rate	IR Mode	Baud Multiple	Encoding	DLH, DLL (Decimal)	Actual Baud Rate	Error (%)\$	Source Jitter (%)	Pulse Duration
2.4 kbps	SIR	16x	3/16	1250	2.4 kbps	0	0	78.1 $\mu$ s
9.6 kbps	SIR	16x	3/16	312	9.6153 kbps	+0.16	0	19.5 $\mu$ s
19.2 kbps	SIR	16x	3/16	156	19.231 kbps	+0.16	0	9.75 $\mu$ s
38.4 kbps	SIR	16x	3/16	78	38.462 kbps	+0.16	0	4.87 $\mu$ s
57.6 kbps	SIR	16x	3/16	52	57.692 kbps	+0.16	0	3.25 $\mu$ s

**Table 23-163. IrDA Baud Rate Settings (continued)**

Baud Rate	IR Mode	Baud Multiple	Encoding	DLH, DLL (Decimal)	Actual Baud Rate	Error (%)\$	Source Jitter (%)	Pulse Duration
115.2 kbps	SIR	16x	3/16	26	115.38 kbps	+0.16	0	1.62 $\mu$ s
0.576 Mbps	MIR	41x/42x	1/4	2	0.5756 Mbps <sup>(1)</sup>	0	+1.63/-0.80	416 ns
1.152 Mbps	MIR	41x/42x	1/4	1	1.1511 Mbps <sup>(1)</sup>	0	+1.63/-0.80	208 ns
4 Mbps	FIR	6x	4 PPM	–	4 Mbps	0	0	125 ns

<sup>(1)</sup> Average value

**NOTE:** Baud rate error and source jitter table values do not include 48-MHz reference clock error and jitter.

### 23.3.4.8.2.3 IrDA Data Formatting

The methods described in this section apply to all IrDA modes (SIR, MIR, and FIR).

#### 23.3.4.8.2.3.1 IRRX Polarity Control

The UART3.UART\_MDR2[6] IRRXINVERT bit provides the flexibility to invert the uart3\_rx\_irrx pin in the UART to ensure that the protocol at the output of the transceiver has the same polarity at module level. By default, the uart3\_rx\_irrx pin is inverted because most transceivers invert the IR receive pin.

#### 23.3.4.8.2.3.2 IrDA Reception Control

The module can transmit and receive data, but when the device is transmitting, the IRRX circuitry is automatically disabled by hardware.

Operation of the uart3\_rx\_irrx input can be disabled by the UART3.UART\_ACREG[5] DIS\_IR\_RX bit.

#### 23.3.4.8.2.3.3 IR Address Checking

In all IR modes, when address checking is enabled, only frames intended for the device are written to the RX FIFO. This restriction avoids receiving frames not meant for this device in a multipoint infrared environment. It is possible to program two frame addresses that the UART IrDA receives, with the UART3.UART\_XON1\_ADDR1[7:0] XON\_WORD1 and UART3.UART\_XON2\_ADDR2[7:0] XON\_WORD2 bit fields.

Setting the UART\_EFR[0] bit to 1 selects address1 checking. Setting the UART\_EFR[1] bit to 1 selects address2 checking. Setting the UART\_EFR[1:0] bit field to 0 disables all address checking operations. If both bits are set, the incoming frame is checked for private and public addresses.

If address checking is disabled, all received frames write to the RX FIFO.

#### 23.3.4.8.2.3.4 Frame Closing

A transmission frame can be terminated in two ways:

- Frame-length method: Set the UART3.UART\_MDR1[7] FRAME\_END\_MODE bit to 0. The MPU writes the value of the frame length to the UART3.UART\_TXFLH and UART3.UART\_TXFLL registers. The device automatically attaches end flags to the frame when the number of bytes transmitted equals the value of the frame length.
- Set-EOT bit method: Set the FRAME\_END\_MODE bit to 1. The MPU writes 1 to the UART3.UART\_ACREG[0] EOT bit just before it writes the last byte to the TX FIFO. When the MPU writes the last byte to the TX FIFO, the device internally sets the tag bit for that character in the TX FIFO. As the TX state-machine reads data from the TX FIFO, it uses this tag-bit information to attach end flags and correctly terminate the frame.

### 23.3.4.8.2.3.5 Store and Controlled Transmission

In store and controlled transmission (SCT) mode, the MPU starts writing data to the TX FIFO. Then, after writing a part of a frame (for a bigger frame) or an entire frame (a small frame; that is, a supervisory frame), the MPU writes 1 to the UART3.UART\_ACREG[2] SCTX\_EN bit (deferred TX start) to start transmission.

SCT mode is enabled by setting the UART3.UART\_MDR1[5] SCT bit to 1. This transmission method differs from normal mode, in which data transmission starts immediately after data is written to the TX FIFO. SCT mode is useful for sending short frames without TX underrun.

### 23.3.4.8.2.3.6 Error Detection

When the UART3.UART\_LSR register is read, the UART3.UART\_LSR[4:2] bit field reflects the error bits [FL, CRC, ABORT] of the frame at the top of the STATUS FIFO (the next frame status to be read).

The error is triggered by an interrupt (for IrDA mode interrupts, see [Table 23-147](#)). The STATUS FIFO must be read until empty (a maximum of eight reads is required).

### 23.3.4.8.2.3.7 Underrun During Transmission

Underrun during transmission occurs when the TX FIFO is empty before the end of the frame is transmitted. When underrun occurs, the device closes the frame with end flags but attaches an incorrect CRC value. The receiving device detects a CRC error and discards the frame; it can then ask for a retransmission.

Underrun also causes an internal flag to be set, which disables additional transmissions. Before the next frame can be transmitted, the MPU must:

- Reset the TX FIFO.
- Read the UART3.UART\_RESUME register, which clears the internal flag.

This function can be disabled by the UART3.UART\_ACREG[4] DIS\_TX\_UNDERRUN bit, compensated by the extension of the stop-bit in transmission if the TX FIFO is empty.

### 23.3.4.8.2.3.8 Overrun During Receive

Overrun during receive for the IrDA mode has the same function as that for the UART mode (see [Section 23.3.4.8.1.3.6, Overrun During Receive](#)).

### 23.3.4.8.2.3.9 Status FIFO

In IrDA modes, a status FIFO records the received frame status. When a complete frame is received, the length of the frame and the error bits associated with the frame are written to the status FIFO.

Reading the UART3.UART\_SFREGH[3:0] most-significant bit (MSB) and UART3.UART\_SFREGL[3:0] (LSB) bit fields obtains the frame length. The frame error status is read in the UART3.UART\_SFLSR register. Reading the UART3.UART\_SFLSR register increments the status FIFO read pointer. The status FIFO is eight entries deep and, therefore, can hold the status of eight frames.

The MPU uses the frame-length information to locate the frame boundary in the received frame data. The MPU can screen bad frames using the error status information and can later request the sender to resend only the bad frames.

This status FIFO can be used effectively in DMA mode because the MPU must be interrupted only when the programmed status FIFO trigger level is reached, not each time a frame is received.

### 23.3.4.8.2.4 SIR Mode Data Formatting

This section provides specific instructions for SIR mode programming.

#### 23.3.4.8.2.4.1 Abort Sequence

When the transmitter prematurely closes a frame, it sends the following sequence to abort: 0x7DC1. The abort pattern closes the frame without a CRC field or an ending flag.

A transmission frame can be aborted by setting the UART3.UART\_ACREG[1] ABORT\_EN bit to 1.

When this bit is set to 1, 0x7D and 0xC1 are transmitted and the frame is not terminated with CRC or stop flags.

The receiver treats a frame as an aborted frame when a 0x7D character followed immediately by a 0xC1 character is received without transparency.

#### CAUTION

When the TX FIFO is not empty and the UART3.UART\_MDR1[5] SCT bit is set to 1, the UART IrDA starts a new transfer with data of a previous frame when the abort frame is sent. Therefore, the TX FIFO must be reset before sending an abort frame.

#### 23.3.4.8.2.4.2 Pulse Shaping

SIR mode supports the 3/16 or the 1.6- $\mu$ s pulse duration methods. The UART3.UART\_ACREG[7] PULSE\_TYPE bit selects the pulse width method in the transmit mode.

#### 23.3.4.8.2.4.3 SIR Free Format Programming

The SIR FF mode is selected by setting the module in the UART mode (UART3.UART\_MDR1[2:0] MODE\_SELECT = 0x0) and the UART3.UART\_MDR2[3] UART\_PULSE bit to 1 to allow pulse shaping.

Because the bit format stays the same, some UART mode configuration registers must be set at specific values:

- UART3.UART\_LCR[1:0] CHAR\_LENGTH bit field = 0x3 (8 data bits)
- UART3.UART\_LCR[2] NB\_STOP bit = 0x0 (1 stop-bit)
- UART3.UART\_LCR[3] PARITY\_EN bit = 0x0 (no parity)

The UART mode interrupts are used for the SIR FF mode, but many are not relevant (XOFF, RTS, CTS, modem status register, etc.).

#### 23.3.4.8.2.5 MIR and FIR Mode Data Formatting

This section describes common instructions for FIR and MIR mode programming.

At the end of a frame reception, the MPU reads the line status register (UART3.UART\_LSR) to detect errors in the received frame.

When the UART3.UART\_MDR1[6] SIP\_MODE bit is set to 1, the TX state-machine always sends one SIP at the end of a transmission frame. However, when the SIP\_MODE bit is set to 0, SIP transmission depends on the UART3.UART\_ACREG[3] SEND\_SIP bit.

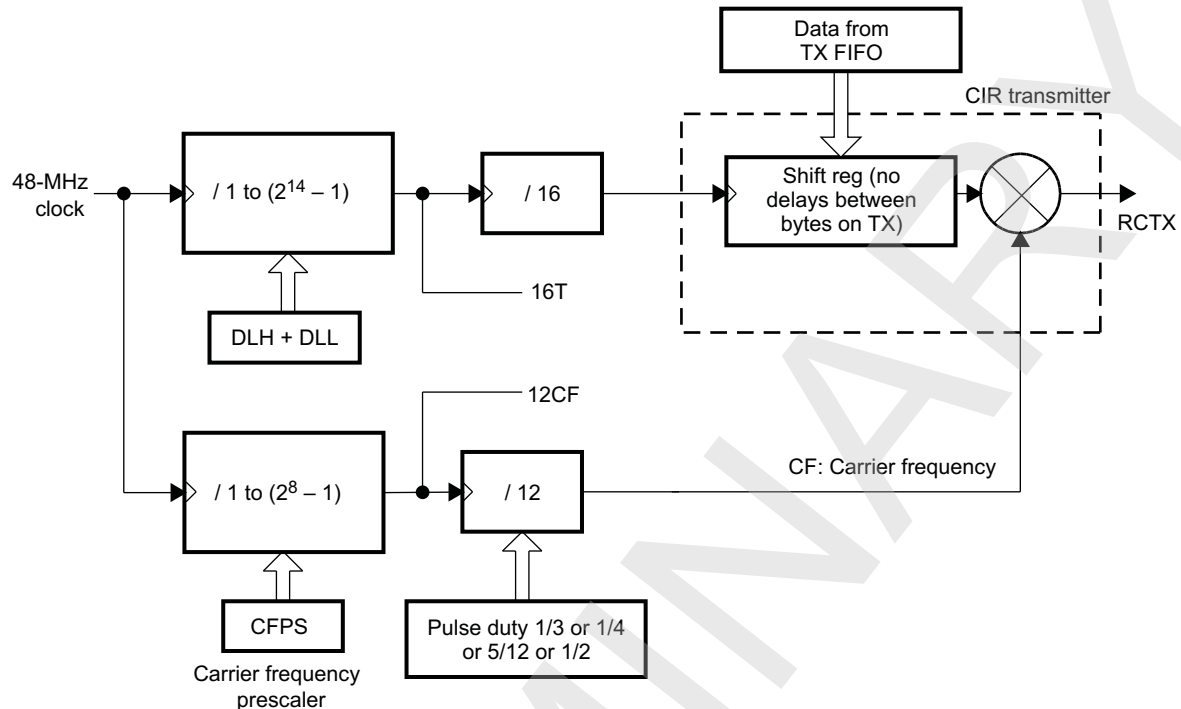
The MPU can set the SEND\_SIP bit at least once every 500 ms. The advantage of this approach over the default approach is that the TX state-machine does not have to send the SIP at the end of each frame, thus reducing the overhead required.

#### 23.3.4.8.3 CIR Mode (UART3 Only)

##### 23.3.4.8.3.1 CIR Mode Clock Generation

Depending on the encoding method (variable pulse distance/biphase), the MPU must develop a data structure that combines 1 and 0 with a  $t$  period to encode the complete frame to transmit. This can then be transmitted to the infrared output with a modulation method, as shown in [Figure 23-84](#).

Figure 23-84. CIR Mode Block Components



uart-034

Based on the requested modulation frequency, the `UART3.UART_CFPS` register must be set with the correct dividing value to provide an accurate pulse frequency:

$$\text{Dividing value} = (\text{FCLK}/12)/\text{MODfreq}$$

Where:

FCLK = System clock frequency (48 MHz)

12 = Real value of baud multiple

MODfreq = Effective frequency of the modulation (MHz)

Example: For a targeted modulation frequency of 36 kHz, the value of `UART3.UART_CFPS` must be set to 0x7 (decimal), which provides a modulation frequency of 36.04 kHz.

---

**NOTE:** The `UART3.UART_CFPS` register starts with a reset value of 105 (decimal), which translates to a frequency of 38.1 kHz.

---

The duty cycle of these pulses is user-defined by the pulse duty register bits in the `UART3.UART_MDR2` register. [Table 23-164](#) shows the duty cycle.

**Table 23-164. Duty Cycle**

UART_MDR2[5:4]	Duty Cycle (High-Level)
00	1/4
01	1/3
10	5/12
11	1/2

### 23.3.4.8.3.2 CIR Data Formatting

The methods described in this section apply to all CIR modes.

#### 23.3.4.8.3.2.1 IRRX Polarity Control

The IRRX polarity control for CIR mode has the same function as that for IrDA mode (see [Section 23.3.4.8.2.3.1, IRRX Polarity Control](#)).

#### 23.3.4.8.3.2.2 CIR Transmission

In transmission, the MPU software must exercise an element of real-time control to transmit data packets, each of which must be emitted at a constant delay from the start-bits of each individual packet. Thus, when sending a series of packets, the packet-to-packet delay must respect a specific delay. Two methods can be used to control this delay:

- Filling the TX FIFO with a number of zero bits that are transmitted with a  $t$  period
- Using an external system timer to control the delay between each start-of-frame or between the end of a frame and the start of the next one. This can be performed by:
  - Controlling the start of the frame using the UART3.UART\_MDR1[5] SCT bit and the UART3.UART\_ACREG[2] SCTX\_EN bit, depending on the timer status
  - Using the UART3.UART\_IIR[5] TX\_STATUS\_IT interrupt bit to preload the next frame in the TX FIFO and to control the start of the timer (in case of control delay between the end of a frame and the start of the next frame).

## 23.3.5 UART/IrDA/CIR Basic Programming Model

### 23.3.5.1 UART Programming Model

#### 23.3.5.1.1 Quick Start

This section describes the procedure for operating the UART with FIFO and DMA or interrupts. This 3-part procedure ensures the quick start of the UART. It does not cover every UART feature.

The first programming model covers software reset of the UART. The second programming model describes FIFO and DMA configuration. The last programming model describes protocol, baud rate, and interrupt configuration.

---

**NOTE:** Each programming model can be used independently of the other two; for instance, reconfiguring the FIFOs and DMA settings only.

Each programming model can be executed starting from any UART register access mode (register modes, submodes, and other register dependencies). However, if the UART register access mode is known before executing the programming model, some steps that enable or restore register access are optional. For more information, see [Section 23.3.4.7.1, Register Access Modes](#).

---

#### 23.3.5.1.1.1 Software Reset

To clear the UART registers, perform the following steps:

- Step 1. Initiate a software reset:  
Set the UARTi.UART\_SYSC[1] SOFTRESET bit to 1.
- Step 2. Wait for the end of the reset operation:  
Poll the UARTi.UART\_SYSS[0] RESETDONE bit until it equals 1.

### 23.3.5.1.1.2 FIFOs and DMA Settings

To enable and configure the receive and transmit FIFOs and program the DMA mode, perform the following steps:

- Step 1. Switch to register configuration mode B to access the UARTi.UART\_EFR register:
  - (a) Save the current UARTi.UART\_LCR register value.
  - (b) Set the UARTi.UART\_LCR register value to 0x00BF.
- Step 2. Enable register submode TCR\_TLR to access the UARTi.UART\_TLR register (part 1 of 2):
  - (a) Save the UARTi.UART\_EFR[4] ENHANCED\_EN value.
  - (b) Set the UARTi.UART\_EFR[4] ENHANCED\_EN bit to 1.
- Step 3. Switch to register configuration mode A to access the UARTi.UART\_MCR register:  
Set the UARTi.UART\_LCR register value to 0x0080.
- Step 4. Enable register submode TCR\_TLR to access the UARTi.UART\_TLR register (part 2 of 2):
  - (a) Save the UARTi.UART\_MCR[6] TCR\_TLR value.
  - (b) Set the UARTi.UART\_MCR[6] TCR\_TLR bit to 1.
- Step 5. Enable the FIFO; load the new FIFO triggers (part 1 of 3) and the new DMA mode (part 1 of 2):  
Set the following bits to the desired values:
  - UARTi.UART\_FCR[7:6] RX\_FIFO\_TRIG
  - UARTi.UART\_FCR[5:4] TX\_FIFO\_TRIG
  - UARTi.UART\_FCR[3] DMA\_MODE
  - UARTi.UART\_FCR[0] FIFO\_ENABLE (0: Disable the FIFO/1; Enable the FIFO)

---

**NOTE:** The UARTi.UART\_FCR register is not readable.

---

- Step 6. Switch to register configuration mode B to access the UARTi.UART\_EFR register:  
Set the UARTi.UART\_LCR register value to 0x00BF.
- Step 7. Load the new FIFO triggers (part 2 of 3):  
Set the following bits to the desired values:
  - UARTi.UART\_TLR[7:4] RX\_FIFO\_TRIG\_DMA
  - UARTi.UART\_TLR[3:0] TX\_FIFO\_TRIG\_DMA
- Step 8. Load the new FIFO triggers (part 3 of 3) and the new DMA mode (part 2 of 2):  
Set the following bits to the desired values:
  - UARTi.UART\_SCR[7] RX\_TRIG\_GRANU1
  - UARTi.UART\_SCR[6] TX\_TRIG\_GRANU1
  - UARTi.UART\_SCR[2:1] DMA\_MODE\_2
  - UARTi.UART\_SCR[0] DMA\_MODE\_CTL
- Step 9. Restore the UARTi.UART\_EFR[4] ENHANCED\_EN value saved in Step 2a.
- Step 10. Switch to register configuration mode A to access the UARTi.UART\_MCR register:  
Set the UARTi.UART\_LCR register value to 0x0080.
- Step 11. Restore the UARTi.UART\_MCR[6] TCR\_TLR value saved in Step 4a.
- Step 12. Restore the UARTi.UART\_LCR value saved in Step 1a.

Triggers are used to generate interrupt and DMA requests. See [Section 23.3.4.6.1.1, Transmit FIFO Trigger](#), to choose the following values:

- UARTi.UART\_FCR[5:4] TX\_FIFO\_TRIG



- UARTi.UART\_TLR[3:0] TX\_FIFO\_TRIG\_DMA
- UARTi.UART\_SCR[6] TX\_TRIG\_GRANU1

Triggers are used to generate interrupt and DMA requests. See [Section 23.3.4.6.1.2, Receive FIFO Trigger](#), to choose the following values:

- UARTi.UART\_FCR[7:6] RX\_FIFO\_TRIG
- UARTi.UART\_TLR[7:4] RX\_FIFO\_TRIG\_DMA
- UARTi.UART\_SCR[7] RX\_TRIG\_GRANU1

DMA mode enables DMA requests. See [Section 23.3.4.6.4, FIFO DMA Mode Operation](#), to choose the following values:

- UARTi.UART\_FCR[3] DMA\_MODE
- UARTi.UART\_SCR[2:1] DMA\_MODE\_2
- UARTi.UART\_SCR[0] DMA\_MODE\_CTL

### 23.3.5.1.1.3 Protocol, Baud Rate, and Interrupt Settings

To program the protocol, baud rate, and interrupt settings, perform the following steps:

Step 1. Disable UART to access the UARTi.UART\_DLL and UARTi.UART\_DLH registers:

Set the UARTi.UART\_MDR1[2:0] MODE\_SELECT bit field to 0x7.

Step 2. Switch to register configuration mode B to access the UARTi.UART\_EFR register:

Set the UARTi.UART\_LCR register value to 0x00BF.

Step 3. Enable access to the UARTi.UART\_IER[7:4] bit field:

(a) Save the UARTi.UART\_EFR[4] ENHANCED\_EN value.

(b) Set the UARTi.UART\_EFR[4] ENHANCED\_EN bit to 1.

Step 4. Switch to register operational mode to access the UARTi.UART\_IER register:

Set the UARTi.UART\_LCR register value to 0x0000.

Step 5. Clear the UARTi.UART\_IER register (set the UARTi.UART\_IER[4] SLEEP\_MODE bit to 0 to change the UARTi.UART\_DLL and UARTi.UART\_DLH registers). Set the UARTi.UART\_IER register value to 0x0000.

Step 6. Switch to register configuration mode B to access the UARTi.UART\_DLL and UARTi.UART\_DLH registers:

Set the UARTi.UART\_LCR register value to 0x00BF.

Step 7. Load the new divisor value:

Set the UARTi.UART\_DLL[7:0] CLOCK\_LSB and UARTi.UART\_DLH[5:0] CLOCK\_MSB bit fields to the desired values.

Step 8. Switch to register operational mode to access the UARTi.UART\_IER register:

Set the UARTi.UART\_LCR register value to 0x0000.

Step 9. Load the new interrupt configuration (0: Disable the interrupt; 1: Enable the interrupt):

Set the following bits to the desired values:

- UARTi.UART\_IER[7] CTS\_IT
- UARTi.UART\_IER[6] RTS\_IT
- UARTi.UART\_IER[5] XOFF\_IT
- UARTi.UART\_IER[4] SLEEP\_MODE
- UARTi.UART\_IER[3] MODEM\_STS\_IT
- UARTi.UART\_IER[2] LINE\_STS\_IT
- UARTi.UART\_IER[1] THR\_IT
- UARTi.UART\_IER[0] RHR\_IT

Step 10. Switch to register configuration mode B to access the UARTi.UART\_EFR register:

Set the UARTi.UART\_LCR register value to 0x00BF.

Step 11. Restore the UARTi.UART\_EFR[4] ENHANCED\_EN value saved in Step 3a.

Step 12. Load the new protocol formatting (parity, stop-bit, character length) and switch to register

operational mode:

Set the UARTi.UART\_LCR[7] DIV\_EN bit to 0.

Set the UARTi.UART\_LCR[6] BREAK\_EN bit to 0.

Set the following bits to the desired values:

- UARTi.UART\_LCR[5] PARITY\_TYPE\_2
- UARTi.UART\_LCR[4] PARITY\_TYPE\_1
- UARTi.UART\_LCR[3] PARITY\_EN
- UARTi.UART\_LCR[2] NB\_STOP
- UARTi.UART\_LCR[1:0] CHAR\_LENGTH

Step 13. Load the new UART mode:

Set the UARTi.UART\_MDR1[2:0] MODE\_SELECT bit field to the desired value.

See [Section 23.3.4.8.1.2, Choosing the Appropriate Divisor Value](#), to choose the following values:

- UARTi.UART\_DLL[7:0] CLOCK\_LSB
- UARTi.UART\_DLH[5:0] CLOCK\_MSB
- UARTi.UART\_MDR1[2:0] MODE\_SELECT

See [Section 23.3.4.8.1.3.1, Frame Formatting](#), to choose the following values:

- UARTi.UART\_LCR[5] PARITY\_TYPE\_2
- UARTi.UART\_LCR[4] PARITY\_TYPE\_1
- UARTi.UART\_LCR[3] PARITY\_EN
- UARTi.UART\_LCR[2] NB\_STOP
- UARTi.UART\_LCR[1:0] CHAR\_LENGTH

### 23.3.5.1.2 Hardware and Software Flow Control Configuration

This section describes the programming steps to enable and configure hardware and software flow control. Hardware and software flow control cannot be used at the same time.

---

**NOTE:** Each programming model can be executed starting from any UART register access mode (register modes, submodes, and other register dependencies). However, if the UART register access mode is known before executing the programming model, some steps that enable or restore register access are optional. For more information, see [Section 23.3.4.7.1, Register Access Modes](#).

---

#### 23.3.5.1.2.1 Hardware Flow Control Configuration

To enable and configure hardware flow control, perform the following steps:

Step 1. Switch to register configuration mode A to access the UARTi.UART\_MCR register:

- (a) Save the current UARTi.UART\_LCR register value.
- (b) Set the UARTi.UART\_LCR register value to 0x0080.

Step 2. Enable register submode TCR\_TLR to access the UARTi.UART\_TCR register (part 1 of 2):

- (a) Save the UARTi.UART\_MCR[6] TCR\_TLR value.
- (b) Set the UARTi.UART\_MCR[6] TCR\_TLR bit to 1.

Step 3. Switch to register configuration mode B to access the UARTi.UART\_EFR register:

Set the UARTi.UART\_LCR register value to 0x00BF.

Step 4. Enable register submode TCR\_TLR to access the UARTi.UART\_TCR register (part 2 of 2):

- (a) Save the UARTi.UART\_EFR[4] ENHANCED\_EN value.
- (b) Set the UARTi.UART\_EFR[4] ENHANCED\_EN bit to 1.

Step 5. Load the new start and halt trigger values for hardware flow control:

Set the following bits to the desired values:

- UARTi.UART\_TCR[7:4] AUTO\_RTS\_START

- UARTi.UART\_TCR[3:0] AUTO\_RTS\_HALT
- Step 6. Enable or disable receive and transmit hardware flow control mode and restore the UARTi.UART\_EFR[4] ENHANCED\_EN value saved in Step 4a.

Set the following bits to the desired values:

- UARTi.UART\_EFR[7] AUTO\_CTS\_EN (0: Disable; 1: Enable)
- UARTi.UART\_EFR[6] AUTO\_RTS\_EN (0: Disable; 1: Enable)

Restore the UARTi.UART\_EFR[4] ENHANCED\_EN bit to the saved value.

Step 7. Switch to register configuration mode A to access the UARTi.UART\_MCR register:

Set the UARTi.UART\_LCR register value to 0x0080.

Step 8. Restore the UARTi.UART\_MCR[6] TCR\_TLR value saved in Step 2a.

Step 9. Restore the UARTi.UART\_LCR value saved in Step 1a.

See [Section 23.3.4.8.1.3.2, Hardware Flow Control](#), to choose the following values:

- UARTi.UART\_EFR[7] AUTO\_CTS\_EN
- UARTi.UART\_EFR[6] AUTO\_RTS\_EN
- UARTi.UART\_TCR[7:4] AUTO\_RTS\_START
- UARTi.UART\_TCR[3:0] AUTO\_RTS\_HALT

### 23.3.5.1.2.2 Software Flow Control Configuration

To enable and configure software flow control, perform the following steps:

Step 1. Switch to register configuration mode B to access the UARTi.UART\_EFR register.

- (a) Save the current UARTi.UART\_LCR register value.
- (b) Set the UARTi.UART\_LCR register value to 0x00BF.

Step 2. Enable register submode XOFF to access the UARTi.UART\_XOFF1 and UARTi.UART\_XOFF2 registers:

- (a) Save the UARTi.UART\_EFR[4] ENHANCED\_EN value.
- (b) Set the UARTi.UART\_EFR[4] ENHANCED\_EN bit to 0.

Step 3. Load the new software flow control characters:

Set the following bits to the desired values:

- UARTi.UART\_XON1\_ADDR1[7:0] XON\_WORD1
- UARTi.UART\_XON2\_ADDR2[7:0] XON\_WORD2
- UARTi.UART\_XOFF1[7:0] XOFF\_WORD1
- UARTi.UART\_XOFF2[7:0] XOFF\_WORD2

Step 4. Enable access to the UARTi.UART\_MCR[7:5] bit field and enable register submode TCR\_TLR to access the UARTi.UART\_TCR register (part 1 of 2):

Set the UARTi.UART\_EFR[4] ENHANCED\_EN bit to 1.

Step 5. Switch to register configuration mode A to access the UARTi.UART\_MCR register:

Set the UARTi.UART\_LCR register value to 0x0080.

Step 6. Enable register submode TCR\_TLR to access the UARTi.UART\_TCR register (part 2 of 2) and enable or disable XON any function:

- (a) Save the UARTi.UART\_MCR[6] TCR\_TLR value.
- (b) Set the UARTi.UART\_MCR[6] TCR\_TLR bit to 1.  
Set the UARTi.UART\_MCR[5] XON\_EN bit to the desired value (0: Disable; 1: Enable).

Step 7. Switch to register configuration mode B to access the UARTi.UART\_EFR register:

Set the UARTi.UART\_LCR register value to 0x00BF.

Step 8. Load the new start and halt trigger values for software flow control:

Set the following bits to the desired values:

- UARTi.UART\_TCR[7:4] AUTO\_RTS\_START
- UARTi.UART\_TCR[3:0] AUTO\_RTS\_HALT

Step 9. Enable or disable special character function and load the new software flow control mode and

restore the UARTi.UART\_EFR[4] ENHANCED\_EN value saved in Step 2a:

Set the following bits to the desired values:

- UARTi.UART\_EFR[5] SPEC\_CHAR (0: Disable; 1: Enable)
- UARTi.UART\_EFR[3:0] SW\_FLOW\_CONTROL

Restore the UARTi.UART\_EFR[4] ENHANCED\_EN bit to the saved value.

Step 10. Switch to register configuration mode A to access the UARTi.UART\_MCR register:

Set the UARTi.UART\_LCR register value to 0x0080.

Step 11. Restore the UARTi.UART\_MCR[6] TCR\_TLR bit value saved in Step 6a.

Step 12. Restore the UARTi.UART\_LCR value saved in Step 1a.

See [Section 23.3.4.8.1.3.3, Software Flow Control](#), to choose the following values:

- UARTi.UART\_EFR[5] SPEC\_CHAR
- UARTi.UART\_EFR[3:0] SW\_FLOW\_CONTROL
- UARTi.UART\_TCR[7:4] AUTO\_RTS\_START
- UARTi.UART\_TCR[3:0] AUTO\_RTS\_HALT
- UARTi.UART\_XON1\_ADDR1[7:0] XON\_WORD1
- UARTi.UART\_XON2\_ADDR2[7:0] XON\_WORD2
- UARTi.UART\_XOFF1[7:0] XOFF\_WORD1
- UARTi.UART\_XOFF2[7:0] XOFF\_WORD2

### 23.3.5.2 IrDA Programming Model (UART3 Only)

#### 23.3.5.2.1 SIR Mode

##### 23.3.5.2.1.1 Receive

The following programming model explains how to program the module to receive an IrDA frame with parity forced to 1, baud rate = 112.5KB, FIFOs disabled, 2 stop-bits, and 8-bit word length:

1. Disable the UART before accessing the UARTi.[UART\\_DLL](#) and UARTi.[UART\\_DLH](#) registers:  
Set the UARTi.[UART\\_MDR1](#)[2:0] MODE\_SELECT bit field to 0x7.
2. Grant access to the [UART\\_DLL](#) and [UART\\_DLH](#) registers (the [UART\\_LCR](#)[7] DIV\_EN bit = 1):  
UART3.[UART\\_LCR](#) = 0x80 (Note: Data format is unaffected by the use and settings of the UART3.[UART\\_LCR](#) register in IrDA mode.)
3. Load the new baud rate (115.2 kbps):  
UART3.[UART\\_DLL](#) = 0x1A  
UART3.[UART\\_DLH](#) = 0x00
4. Set SIR mode:  
UART3.[UART\\_MDR1](#)[2:0] MODE\_SELECT = 0x1
5. Disable access to the [UART\\_DLL](#) and [UART\\_DLH](#) registers and switch to register operational mode:  
UART3.[UART\\_LCR](#) = 0x00.
6. Optional: Enable the RHR interrupt:  
UART3.[UART\\_IER](#)[0] RHR\_IT = 0x1

##### 23.3.5.2.1.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 6-byte frame with no parity, baud rate = 112.5 kbps, FIFOs disabled, 3/16 encoding, 2 stop-bits, and 7-bit word length:

1. Disable the UART before accessing the UARTi.[UART\\_DLL](#) and UARTi.[UART\\_DLH](#) registers:  
Set the [UART\\_MDR1](#)[2:0] MODE\_SELECT bit field to 0x7.
2. Grant access to the [UART\\_EFR](#) register:  
UART3.[UART\\_LCR](#) = 0xBF
3. Enable the enhanced features (the [UART\\_EFR](#)[4] ENHANCED\_EN bit = 1):  
Set the UART3.[UART\\_EFR](#) register value to 0x10.
4. Grant access to the [UART\\_DLL](#) and [UART\\_DLH](#) registers (the [UART\\_LCR](#)[7] DIV\_EN bit = 1):  
UART3.[UART\\_LCR](#) = 0x80 (Note: Data format is unaffected by the use and settings of the UART3.[UART\\_LCR](#) register in IrDA mode.)
5. Load the new baud rate (115.2 kbps):  
UART3.[UART\\_DLL](#) = 0x1A  
UART3.[UART\\_DLH](#) = 0x00
6. Set SIR mode (the [UART\\_MDR1](#)[2:0] MODE\_SELECT bit field = 0x1):  
UART3.[UART\\_MDR1](#) = 0x01
7. Disable access to the [UART\\_DLL](#) and [UART\\_DLH](#) registers and switch to register operational mode:  
UART3.[UART\\_LCR](#) = 0x00.
8. Force DTR output to active:  
UART3.[UART\\_MCR](#)[0] DTR = 1
9. Optional: Enable the THR interrupt:  
UART3.[UART\\_IER](#)[1] THR\_IT = 1
10. Set transmit frame length to 6 bytes:  
UART3.[UART\\_TXFLR](#) = 0x06
11. Set 7 starts of frame transmission:  
UART3.[UART\\_EBLR](#) = 0x08
12. Optional: Set SIR pulse width to be 1.6 us:  
UART3.[UART\\_ACREG](#)[7] PULSE\_TYPE = 1
13. Load [UART\\_THR](#) with the desired data to be transmitted.

### 23.3.5.2.2 MIR Mode

#### 23.3.5.2.2.1 Receive

The following programming model explains how to program the module to receive an IrDA frame with no parity, baud rate = 1.152 mpbs, and FIFOs disabled.

1. Disable the UART before accessing the UARTi.UART\_DLL and UARTi.UART\_DLH registers:  
Set the UARTi.UART\_MDR1[2:0] MODE\_SELECT bit field to 0x7.
2. Grant access to the UART\_DLL and UART\_DLH registers (UART\_LCR[7] DIV\_EN bit = 1):  
UART3.UART\_LCR = 0x80 (Note: Data format is unaffected by the use and settings of the UART3.UART\_LCR register in IrDA mode.)
3. Load the new baud rate (1.152 mpbs):  
UART3.UART\_DLL = 0x01  
UART3.UART\_DLH = 0x00
4. Set MIR mode:  
UART3.UART\_MDR1[2:0] MODE\_SELECT = 0x4
5. Disable access to the UART\_DLL and UART\_DLH registers and switch to register operational mode:  
UART3.UART\_LCR = 0x00
6. Force DTR output to active (UART\_MCR[0] DTR = 1):  
Force RTS output to active (UART\_MCR[1] RTS = 1).  
UART3.UART\_MCR = 0x3
7. Optional: Enable the RHR interrupt:  
UART3.UART\_IER[0] RHR\_IT = 1

#### 23.3.5.2.2.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 60-byte frame with no parity, baud rate = 1.152 Mbps, and FIFOs disabled:

1. Disable the UART before accessing the UARTi.UART\_DLL and UARTi.UART\_DLH registers:  
Set the UARTi.UART\_MDR1[2:0] MODE\_SELECT bit field to 0x7.
2. Grant access to the UART\_DLL and UART\_DLH registers (UART\_LCR[7] DIV\_EN bit = 1):  
UART3.UART\_LCR = 0x80 (**Note:** Data format is unaffected by the use and settings of the UART3.UART\_LCR register in IrDA mode.)
3. Load the new baud rate (1.152 mpbs):  
UART3.UART\_DLL = 0x01  
UART3.UART\_DLH = 0x00
4. Set MIR mode:  
UART3.UART\_MDR1[2:0] MODE\_SELECT = 0x4.
5. Disable access to the UART\_DLL and UART\_DLH registers and switch to register operational mode:  
UART3.UART\_LCR = 0x00
6. Force DTR output to active:  
UART3.UART\_MCR[0] DTR = 1
7. Optional: Enable the THR interrupt:  
UART3.UART\_IER[1] THR\_IT = 1
8. Set the frame length to 60 bytes:  
UART3.UART\_TXFLL = 0x3C
9. Optional: Transmit 8 additional starts of frame (MIR mode requires two starts):  
UART3.UART\_EBLR = 0x08
10. SIP is sent at the end of transmission:  
UART3.UART\_ACREG[3] = 1
11. Load the UART\_THR register with the data to be transmitted.

### 23.3.5.2.3 FIR Mode

#### 23.3.5.2.3.1 Receive

The following programming model explains how to program the module to receive IrDA frame with no parity, baud rate = 4M bits/s, FIFOs enable, 8-bit word length.

1. Disable the UART before accessing the UART3.UART\_DLL and UART3.UART\_DLH registers:  
Set the UART3.UART\_MDR1[2:0] MODE\_SELECT bit field to 0x7.
2. Grant access to the UART\_DLL and UART\_DLH registers (UART\_LCR[7] DIV\_EN bit = 1):  
UART3.UART\_LCR = 0x80 (**Note:** Data format is unaffected by the use and settings of the UART3.UART\_LCR register in IrDA mode.)
3. FIFO clear and enable:  
UART3.UART\_FCR = 0x7 (Tx/Rx FIFO trigger : UART\_FCR[7:6] and UART\_FCR[5:4])  
UART3.UART\_LCR[7]=0
4. Set FIR mode:  
UART3.UART\_MDR1[2:0] MODE\_SELECT = 0x5
5. Set frame length:  
UART3.UART\_RXFLL = 0xA (Data + CRC + STOP)
6. Disable access to UART3.UART\_DLL and UART3.UART\_DLH and switch to register operational mode:  
UART3.UART\_LCR[7] DIV\_EN = 0x0
7. Optional: Enable the RHR interrupt:  
UART3.UART\_IER[0] RHR\_IT = 1

#### 23.3.5.2.3.2 Transmit

The following programming model explains how to program the module to transmit IrDA 4 bytes frame with no parity, baud rate = 4 Mbps, FIFOs enable, 8-bit word length.

1. Disable the UART before accessing the UART3.UART\_DLL and UART3.UART\_DLH registers:  
Set the UART3.UART\_MDR1[2:0] MODE\_SELECT bit field to 0x7.
2. Grant access to EFR\_REG:  
UART3.UART\_LCR = 0xBF
3. Enable the enhanced features (EFR\_REG[4] ENAHNCED\_EN = 0x1):  
UART3.UART\_EFR = 0x10
4. FIFO clear and enable:  
UART3.UART\_FCR = 0x7 (Tx/Rx FIFO trigger: UART\_FCR[7:6] and UART\_FCR[5:4]).  
UART3.UART\_LCR[7] = 0
5. Set FIR mode and enable auto-SIP mode:  
UART3.UART\_MDR1 = 0x45
6. Set Frame Length:  
UART3.UART\_TXFLL = 0x4  
UART3.UART\_TXFLH = 0x0  
UART3.UART\_RXFLL = 0xA (Data + CRC + STOP)  
UART3.UART\_RXFLH = 0x0
7. Force DTR output to active:  
UART3.UART\_MCR[0] DTR = 0x1
8. Optional: Enable THR interrupt:  
UART3.UART\_IER[1] THR\_IT = 0x1
9. Optional: Transmit 8 additional starts of frame (MIR mode requires two starts):  
UART3.UART\_EBLR = 0x08
10. SIP is sent at the end of transmission:  
UART3.UART\_ACREG[3] = 1
11. Load the UART\_THR register with the data to be transmitted.

## 23.3.6 UART/IrDA/CIR Register Manual

### 23.3.6.1 UART/IrDA/CIR Instance Summary

Table 23-165 shows the base address and address space for the UART/IrDA/CIR module instances.

**Table 23-165. UART/IrDA/CIR Instance Summary**

Module Name	Base Address	Size
UART1 <sup>(1)</sup>	0x4806 A000	1KB
UART2 <sup>(1)</sup>	0x4806 C000	1KB
UART3 <sup>(2)</sup>	0x4802 0000	1KB
UART4 <sup>(1)</sup>	0x4806 E000	1KB

<sup>(1)</sup> UART mode only

<sup>(2)</sup> UART, IrDA, or CIR mode

### 23.3.6.2 UART/IrDA/CIR Registers

#### 23.3.6.2.1 UART/IrDA/CIR Register Summary

**Table 23-166. UART/IrDA/CIR Register Mapping Summary (UART1 and UART2)**

Register Name	Type	Register Width (Bits)	Address Offset	UART1 Base Address	UART2 Base Address
UART_THR	W	32	0x0000 0000	0x4806 A000	0x4806 C000
UART_RHR	R	32	0x0000 0000	0x4806 A000	0x4806 C000
UART_DLL	RW	32	0x0000 0000	0x4806 A000	0x4806 C000
UART_IER	RW	32	0x0000 0004	0x4806 A004	0x4806 C004
UART_IIR	R	32	0x0000 0008	0x4806 A008	0x4806 C008
UART_DLH	RW	32	0x0000 0004	0x4806 A004	0x4806 C004
UART_FCR	W	32	0x0000 0008	0x4806 A008	0x4806 C008
UART_EFR	RW	32	0x0000 0008	0x4806 A008	0x4806 C008
UART_LCR	RW	32	0x0000 000C	0x4806 A00C	0x4806 C00C
UART_XON1_ADDR1	RW	32	0x0000 0010	0x4806 A010	0x4806 C010
UART_MCR	RW	32	0x0000 0010	0x4806 A010	0x4806 C010
UART_LSR	R	32	0x0000 0014	0x4806 A014	0x4806 C014
UART_XON2_ADDR2	RW	32	0x0000 0014	0x4806 A014	0x4806 C014
UART_TCR	RW	32	0x0000 0018	0x4806 A018	0x4806 C018
UART_XOFF1	RW	32	0x0000 0018	0x4806 A018	0x4806 C018
UART_MSR	R	32	0x0000 0018	0x4806 A018	0x4806 C018
UART_SPR	RW	32	0x0000 001C	0x4806 A01C	0x4806 C01C
UART_TLR	RW	32	0x0000 001C	0x4806 A01C	0x4806 C01C
UART_XOFF2	RW	32	0x0000 001C	0x4806 A01C	0x4806 C01C
UART_MDR1	RW	32	0x0000 0020	0x4806 A020	0x4806 C020
UART_MDR2	RW	32	0x0000 0024	0x4806 A024	0x4806 C024
UART_SFLSR	R	32	0x0000 0028	0x4806 A028	0x4806 C028
UART_TXFLL	W	32	0x0000 0028	0x4806 A028	0x4806 C028
UART_RESUME	R	32	0x0000 002C	0x4806 A02C	0x4806 C02C
UART_TXFLH	W	32	0x0000 002C	0x4806 A02C	0x4806 C02C
UART_SFREGL	R	32	0x0000 0030	0x4806 A030	0x4806 C030
UART_RXFLL	W	32	0x0000 0030	0x4806 A030	0x4806 C030
UART_SFREGH	R	32	0x0000 0034	0x4806 A034	0x4806 C034
UART_RXFLH	W	32	0x0000 0034	0x4806 A034	0x4806 C034



**Table 23-166. UART/IrDA/CIR Register Mapping Summary (UART1 and UART2) (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	UART1 Base Address	UART2 Base Address
UART_BLR	RW	32	0x0000 0038	0x4806 A038	0x4806 C038
UART_UASR	R	32	0x0000 0038	0x4806 A038	0x4806 C038
UART_ACREG	RW	32	0x0000 003C	0x4806 A03C	0x4806 C03C
UART_SCR	RW	32	0x0000 0040	0x4806 A040	0x4806 C040
UART_SSR	R	32	0x0000 0044	0x4806 A044	0x4806 C044
UART_EBLR	RW	32	0x0000 0048	0x4806 A048	0x4806 C048
UART_MVR	R	32	0x0000 0050	0x4806 A050	0x4806 C050
UART_SYSC	RW	32	0x0000 0054	0x4806 A054	0x4806 C054
UART_SYSS	R	32	0x0000 0058	0x4806 A058	0x4806 C058
UART_WER	RW	32	0x0000 005C	0x4806 A05C	0x4806 C05C
UART_CFPS	RW	32	0x0000 0060	0x4806 A060	0x4806 C060
UART_RXFIFO_LVL	R	32	0x0000 0064	0x4806 A064	0x4806 C064
UART_TXFIFO_LVL	R	32	0x0000 0068	0x4806 A068	0x4806 C068
UART_IER2	RW	32	0x0000 006C	0x4806 A06C	0x4806 C06C
UART_ISR2	RW	32	0x0000 0070	0x4806 A070	0x4806 C070
UART_FREQ_SEL	RW	32	0x0000 0074	0x4806 A074	0x4806 C074
UART_MDR3	RW	32	0x0000 0080	0x4806 A080	0x4806 C080
UART_TX_DMA_THRESHOLD	RW	32	0x0000 0084	0x4806 A084	0x4806 C084

**Table 23-167. UART/IrDA/CIR Register Mapping Summary (UART3 and UART4)**

Register Name	Type	Register Width (Bits)	Address Offset	UART3 Base Address	UART4 Base Address
UART_THR	W	32	0x0000 0000	0x4802 0000	0x4806 E000
UART_RHR	R	32	0x0000 0000	0x4802 0000	0x4806 E000
UART_DLL	RW	32	0x0000 0000	0x4802 0000	0x4806 E000
UART_IER	RW	32	0x0000 0004	0x4802 0004	0x4806 E004
UART_IIR	R	32	0x0000 0008	0x4802 0008	0x4806 E008
UART_DLH	RW	32	0x0000 0004	0x4802 0004	0x4806 E004
UART_FCR	W	32	0x0000 0008	0x4802 0008	0x4806 E008
UART_EFR	RW	32	0x0000 0008	0x4802 0008	0x4806 E008
UART_LCR	RW	32	0x0000 000C	0x4802 000C	0x4806 E00C
UART_XON1_ADDR1	RW	32	0x0000 0010	0x4802 0010	0x4806 E010
UART_MCR	RW	32	0x0000 0010	0x4802 0010	0x4806 E010
UART_LSR	R	32	0x0000 0014	0x4802 0014	0x4806 E014
UART_XON2_ADDR2	RW	32	0x0000 0014	0x4802 0014	0x4806 E014
UART_TCR	RW	32	0x0000 0018	0x4802 0018	0x4806 E018
UART_XOFF1	RW	32	0x0000 0018	0x4802 0018	0x4806 E018
UART_MSR	R	32	0x0000 0018	0x4802 0018	0x4806 E018
UART_SPR	RW	32	0x0000 001C	0x4802 001C	0x4806 E01C
UART_TLR	RW	32	0x0000 001C	0x4802 001C	0x4806 E01C
UART_XOFF2	RW	32	0x0000 001C	0x4802 001C	0x4806 E01C
UART_MDR1	RW	32	0x0000 0020	0x4802 0020	0x4806 E020
UART_MDR2	RW	32	0x0000 0024	0x4802 0024	0x4806 E024
UART_SFLSR	R	32	0x0000 0028	0x4802 0028	0x4806 E028
UART_TXFLL	W	32	0x0000 0028	0x4802 0028	0x4806 E028
UART_RESUME	R	32	0x0000 002C	0x4802 002C	0x4806 E02C

**Table 23-167. UART/IrDA/CIR Register Mapping Summary (UART3 and UART4) (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	UART3 Base Address	UART4 Base Address
UART_TXFLH	W	32	0x0000 002C	0x4802 002C	0x4806 E02C
UART_SFREGL	R	32	0x0000 0030	0x4802 0030	0x4806 E030
UART_RXFLL	W	32	0x0000 0030	0x4802 0030	0x4806 E030
UART_SFREGH	R	32	0x0000 0034	0x4802 0034	0x4806 E034
UART_RXFLH	W	32	0x0000 0034	0x4802 0034	0x4806 E034
UART_BLR	RW	32	0x0000 0038	0x4802 0038	0x4806 E038
UART_UASR	R	32	0x0000 0038	0x4802 0038	0x4806 E038
UART_ACREG	RW	32	0x0000 003C	0x4802 003C	0x4806 E03C
UART_SCR	RW	32	0x0000 0040	0x4802 0040	0x4806 E040
UART_SSR	R	32	0x0000 0044	0x4802 0044	0x4806 E044
UART_EBLR	RW	32	0x0000 0048	0x4802 0048	0x4806 E048
UART_MVR	R	32	0x0000 0050	0x4802 0050	0x4806 E050
UART_SYSC	RW	32	0x0000 0054	0x4802 0054	0x4806 E054
UART_SYSS	R	32	0x0000 0058	0x4802 0058	0x4806 E058
UART_WER	RW	32	0x0000 005C	0x4802 005C	0x4806 E05C
UART_CFPS	RW	32	0x0000 0060	0x4802 0060	0x4806 E060
UART_RXFIFO_LVL	R	32	0x0000 0064	0x4802 0064	0x4806 E064
UART_TXFIFO_LVL	R	32	0x0000 0068	0x4802 0068	0x4806 E068
UART_IER2	RW	32	0x0000 006C	0x4802 006C	0x4806 E06C
UART_ISR2	RW	32	0x0000 0070	0x4802 0070	0x4806 E070
UART_FREQ_SEL	RW	32	0x0000 0074	0x4802 0074	0x4806 E074
UART_MDR3	RW	32	0x0000 0080	0x4802 0080	0x4806 E080
UART_TX_DMA_THRESHOL D	RW	32	0x0000 0084	0x4806 A084	0x4806 C084

**23.3.6.2.2 UART/IrDA/CIR Registers****Table 23-168. UART\_THR**

<b>Address Offset</b>	0x0000 0000																																																																
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .																																																																
<b>Description</b>	The transmitter section consists of the transmit holding register (THR) and the transmit shift register. The transmit holding register is a 64-byte FIFO. The local host (LH) writes data to the THR. The data is placed in the transmit shift register where it is shifted out serially on the TX output. If the FIFO is disabled, location 0 of the FIFO stores the data.																																																																
<b>Type</b>	W																																																																
<table border="1" style="width:100%; border-collapse: collapse;"> <tr> <td style="width:2.5%;">31</td><td style="width:2.5%;">30</td><td style="width:2.5%;">29</td><td style="width:2.5%;">28</td><td style="width:2.5%;">27</td><td style="width:2.5%;">26</td><td style="width:2.5%;">25</td><td style="width:2.5%;">24</td> <td style="width:2.5%; background-color: #ffffcc;">23</td><td style="width:2.5%; background-color: #ffffcc;">22</td><td style="width:2.5%; background-color: #ffffcc;">21</td><td style="width:2.5%; background-color: #ffffcc;">20</td><td style="width:2.5%; background-color: #ffffcc;">19</td><td style="width:2.5%; background-color: #ffffcc;">18</td><td style="width:2.5%; background-color: #ffffcc;">17</td><td style="width:2.5%; background-color: #ffffcc;">16</td> <td style="width:2.5%;">15</td><td style="width:2.5%;">14</td><td style="width:2.5%;">13</td><td style="width:2.5%;">12</td><td style="width:2.5%;">11</td><td style="width:2.5%;">10</td><td style="width:2.5%;">9</td><td style="width:2.5%;">8</td> <td style="width:2.5%; background-color: #ffffcc;">7</td><td style="width:2.5%; background-color: #ffffcc;">6</td><td style="width:2.5%; background-color: #ffffcc;">5</td><td style="width:2.5%; background-color: #ffffcc;">4</td><td style="width:2.5%; background-color: #ffffcc;">3</td><td style="width:2.5%; background-color: #ffffcc;">2</td><td style="width:2.5%; background-color: #ffffcc;">1</td><td style="width:2.5%; background-color: #ffffcc;">0</td> </tr> <tr> <td colspan="16" style="text-align: center;">RESERVED</td> <td colspan="2" style="text-align: center;">THR</td> </tr> </table>																31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																THR	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																		
RESERVED																THR																																																	
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>														<b>Type</b>	<b>Reset</b>																																																
31:8	RESERVED	Write has no effect.														W	0x000000																																																
7:0	THR	Transmit holding register														W	0x-																																																

**Table 23-169. Register Call Summary for Register UART\_THR**

- UART/IrDA/CIR
- [UART Mode Interrupt Management: \[0\] \[1\]](#)
  - [IrDA Interrupts: \[2\] \[3\] \[4\]](#)
  - [CIR Interrupts: \[5\] \[6\]](#)
  - [FIFO Management: \[7\]](#)
  - [DMA Transmission: \[8\] \[9\]](#)
  - [Registers Available for the Register Access Modes: \[10\]](#)
  - [Registers Available for the UART Function: \[11\]](#)
  - [Registers Available for the IrDA Function \(UART3 Only\): \[12\]](#)
  - [Registers Available for the CIR Function \(UART3 Only\): \[13\]](#)
  - [Transmit: \[14\]](#)
  - [Transmit: \[15\]](#)
  - [Transmit: \[16\]](#)
  - [UART/IrDA/CIR Register Summary: \[17\] \[18\]](#)

**Table 23-170. UART\_RHR**

<b>Address Offset</b>	0x0000 0000																																																													
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b> UART																																																												
<b>Description</b>	The receiver section consists of the receiver holding register (RHR) and the receiver shift register. The RHR is a 64-byte FIFO. The receiver shift register receives serial data from RX input. The data is converted to parallel data and moved to the RHR. If the FIFO is disabled, location 0 of the FIFO stores the single data character. <b>Note:</b> If an overflow occurs, the data in the RHR is not overwritten.																																																													
<b>Type</b>	R																																																													
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="12">RHR</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																RHR											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
RESERVED																RHR																																														
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																										
31:8	RESERVED	Read returns 0	R	0x000000																																																										
7:0	RHR	Receive holding register	R	0x-																																																										

**Table 23-171. Register Call Summary for Register UART\_RHR**

- UART/IrDA/CIR
- [UART Mode Interrupt Management: \[0\] \[1\] \[2\]](#)
  - [IrDA Interrupts: \[3\] \[4\] \[5\]](#)
  - [FIFO Management: \[6\] \[7\]](#)
  - [Registers Available for the Register Access Modes: \[8\]](#)
  - [Registers Available for the UART Function: \[9\]](#)
  - [Registers Available for the IrDA Function \(UART3 Only\): \[10\]](#)
  - [UART Data Formatting: \[11\] \[12\] \[13\] \[14\]](#)
  - [UART/IrDA/CIR Register Summary: \[15\] \[16\]](#)

**Table 23-172. UART\_DLL**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	This register, with <a href="#">UART_DLH</a> , stores the 14-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor. DLL stores the least-significant part of the divisor.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														CLOCK_LSB																	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	CLOCK_LSB	Stores the 8-bit LSB divisor value	RW	0x00

**Table 23-173. Register Call Summary for Register UART\_DLL**

## UART/IrDA/CIR

- [Module Power Saving: \[0\] \[1\]](#)
- [DMA Transfers \(DMA Mode 1, 2, or 3\): \[2\]](#)
- [Registers Available for the Register Access Modes: \[3\] \[4\] \[5\] \[6\]](#)
- [Registers Available for the UART Function: \[7\] \[8\] \[9\] \[10\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[11\] \[12\] \[13\] \[14\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[15\] \[16\] \[17\] \[18\]](#)
- [UART Clock Generation: Baud Rate Generation: \[19\]](#)
- [UART Data Formatting: \[20\] \[21\]](#)
- [IrDA Clock Generation: Baud Generator: \[22\]](#)
- [Protocol, Baud Rate, and Interrupt Settings: \[23\] \[24\] \[25\] \[26\] \[27\]](#)
- [Receive: \[28\] \[29\] \[30\] \[31\]](#)
- [Transmit: \[32\] \[33\] \[34\] \[35\]](#)
- [Receive: \[36\] \[37\] \[38\] \[39\]](#)
- [Transmit: \[40\] \[41\] \[42\] \[43\]](#)
- [Receive: \[44\] \[45\] \[46\]](#)
- [Transmit: \[47\]](#)
- [UART/IrDA/CIR Register Summary: \[48\] \[49\]](#)
- [UART/IrDA/CIR Registers: \[50\] \[51\]](#)

**Table 23-174. UART\_IER**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Interrupt enable register		
<b>Type</b>	RW		

**UART Bit Field Details**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														CTS_IT	RTS_IT	XOFF_IT	SLEEP_MODE	MODEM_STS_IT	LINE_STS_IT	THR_IT	RHR_IT										

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	CTS_IT	0x0: Disables the CTS* interrupt 0x1: Enables the CTS* interrupt	RW	0
6	RTS_IT	0x0: Disables the RTS* interrupt 0x1: Enables the RTS* interrupt	RW	0
5	XOFF_IT	0x0: Disables the XOFF interrupt 0x1: Enables the XOFF interrupt	RW	0
4	SLEEP_MODE	0x0: Disables sleep mode 0x1: Enables sleep mode (stop baud rate clock when the module is inactive)	RW	0
3	MODEM_STS_IT	0x0: Disables the modem status register interrupt 0x1: Enables the modem status register interrupt	RW	0
2	LINE_STS_IT	0x0: Disables the receiver line status interrupt 0x1: Enables the receiver line status interrupt	RW	0
1	THR_IT	0x0: Disables the THR interrupt 0x1: Enables the THR interrupt	RW	0
0	RHR_IT	0x0: Disables the RHR interrupt and time-out interrupt 0x1: Enables the RHR interrupt and time-out interrupt	RW	0

**Table 23-175. Register Call Summary for Register UART\_IER**

UART/IrDA/CIR

- [Module Power Saving: \[0\] \[1\]](#)
- [IrDA Interrupts: \[2\] \[3\]](#)
- [CIR Interrupts: \[4\] \[5\]](#)
- [FIFO Interrupt Mode: \[6\]](#)
- [FIFO Polled Mode Operation: \[7\]](#)
- [Registers Available for the Register Access Modes: \[8\] \[9\]](#)
- [Registers Available for the UART Function: \[10\] \[11\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[12\] \[13\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[14\] \[15\]](#)
- [UART Data Formatting: \[16\]](#)
- [Protocol, Baud Rate, and Interrupt Settings: \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\]](#)
- [Receive: \[31\]](#)
- [Transmit: \[32\]](#)
- [Receive: \[33\]](#)
- [Transmit: \[34\]](#)
- [Receive: \[35\]](#)
- [Transmit: \[36\]](#)
- [UART/IrDA/CIR Register Summary: \[37\] \[38\]](#)

**IrDA Bit Field Details**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EOF_IT	LINE_STS_IT	TX_STATUS_IT	STS_FIFO_TRIG_IT	RX_OVERRUN_IT	LAST_RX_BYTE_IT	THR_IT	RHR_IT

## UART/IrDA/CIR

www.ti.com

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	EOF_IT	0x0: Disables the received EOF interrupt 0x1: Enables the received EOF interrupt	RW	0
6	LINE_STS_IT	0x0: Disables the receiver line status interrupt 0x1: Enables the receiver line status interrupt	RW	0
5	TX_STATUS_IT	0x0: Disables the TX status interrupt 0x1: Enables the TX status interrupt	RW	0
4	STS_FIFO_TRIG_IT	0x0: Disables status FIFO trigger level interrupt 0x1: Enables status FIFO trigger level interrupt.	RW	0
3	RX_OVERRUN_IT	0x0: Disables the RX overrun interrupt 0x1: Enables the RX overrun interrupt	RW	0
2	LAST_RX_BYTE_IT	0x0: Disables the last byte of frame in RX FIFO interrupt 0x1: Enables the last byte of frame in RX FIFO interrupt	RW	0
1	THR_IT	0x0: Disables the THR interrupt 0x1: Enables the THR interrupt	RW	0
0	RHR_IT	0x0: Disables the RHR interrupt and time-out interrupt 0x1: Enables the RHR interrupt and time-out interrupt	RW	0

## CIR Bit Field Details

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	TX_STATUS_IT	RESERVED	RX_OVERRUN_IT	RX_STOP_IT	THR_IT	RHR_IT									

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:6	RESERVED	Not used in CIR mode	RW	0x0
5	TX_STATUS_IT	0x0: Disables the TX status interrupt 0x1: Enables the TX status interrupt	RW	0
4	RESERVED	Not used in CIR mode	RW	0
3	RX_OVERRUN_IT	0x0: Disables the RX overrun interrupt 0x1: Enables the RX overrun interrupt	RW	0
2	RX_STOP_IT	0x0: Disables the receive stop interrupt 0x1: Enables the receive stop interrupt	RW	0
1	THR_IT	0x0: Disables the THR interrupt 0x1: Enables the THR interrupt	RW	0
0	RHR_IT	0x0: Disables the RHR interrupt 0x1: Enables the RHR interrupt	RW	0

**Table 23-176. UART\_IIR**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Interrupt Identification register. The IIR is a read-only register that provides the source of the interrupt in a prioritized manner.		
<b>Type</b>	R		

**UART Bit Field Details**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FCR_MIRROR		IT_TYPE					IT_PENDING								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	R	0x000000
7:6	FCR_MIRROR	Mirror the contents of <a href="#">UART_FCR[0]</a> on both bits.	R	0x0
5:1	IT_TYPE	Read 0x0: Modem interrupt. Priority = 4 Read 0x1: THR interrupt. Priority = 3 Read 0x2: RHR interrupt. Priority = 2 Read 0x3: Receiver line status error. Priority = 3 Read 0x6: Rx time-out. Priority = 2 Read 0x8: XOFF/special character. Priority = 5 Read 0x10: CTS, RTS, DSR change state from active (low) to inactive (high) Priority = 6	R	0x00
0	IT_PENDING	Read 0x0: An interrupt is pending. Read 0x1: No interrupt is pending.	R	1

**Table 23-177. Register Call Summary for Register UART\_IIR**

UART/IrDA/CIR

- [Block Diagram: \[0\]](#)
- [UART Mode Interrupt Management: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [Wake-Up Interrupt: \[8\]](#)
- [IrDA Interrupts: \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [CIR Interrupts: \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [Registers Available for the Register Access Modes: \[19\] \[20\]](#)
- [Registers Available for the UART Function: \[21\] \[22\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[23\] \[24\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[25\] \[26\]](#)
- [UART Data Formatting: \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [CIR Data Formatting: \[32\]](#)
- [UART/IrDA/CIR Register Summary: \[33\] \[34\]](#)
- [UART/IrDA/CIR Registers: \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\]](#)

**IrDA Bit Field Details**

## UART/IrDA/CIR

www.ti.com

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EOF_IT		LINE_STS_IT	TX_STATUS_IT	STS_FIFO_IT	RX_OE_IT	RX_FIFO_LAST_BYTE_IT	THR_IT	RHR_IT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	R	0x000000
7	EOF_IT	Read 0x0: Receive EOF interrupt inactive Read 0x1: Received EOF interrupt active	R	0
6	LINE_STS_IT	Read 0x0: Receiver line status interrupt inactive Read 0x1: Receiver line status interrupt active	R	0
5	TX_STATUS_IT	Read 0x0: TX status interrupt inactive Read 0x1: TX status interrupt active	R	0
4	STS_FIFO_IT	Read 0x0: Status FIFO trigger level interrupt inactive Read 0x1: Status FIFO trigger level interrupt active	R	0
3	RX_OE_IT	Read 0x0: RX overrun interrupt inactive Read 0x1: RX overrun interrupt active	R	0
2	RX_FIFO_LAST_BYTE_IT	Read 0x0: Last byte of frame in RX FIFO interrupt inactive Read 0x1: Last byte of frame in RX FIFO interrupt active	R	0
1	THR_IT	Read 0x0: THR interrupt inactive Read 0x1: THR interrupt active	R	0
0	RHR_IT	Read 0x0: RHR interrupt inactive Read 0x1: RHR interrupt active	R	1

## CIR Bit Field Details

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		TX_STATUS_IT	RESERVED	RX_OE_IT	RX_STOP_IT	THR_IT	RHR_IT								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	R	0x000000
7:6	RESERVED	Not used in CIR mode	R	0x0
5	TX_STATUS_IT	Read 0x0: TX status interrupt inactive Read 0x1: TX status interrupt active	R	0
4	RESERVED	Not used in CIR mode	R	0
3	RX_OE_IT	Read 0x0: RX overrun interrupt inactive Read 0x1: RX overrun interrupt active	R	0
2	RX_STOP_IT	Read 0x0: Receive stop interrupt is inactive Read 0x1: Receive stop interrupt is active	R	0



Bits	Field Name	Description	Type	Reset
1	THR_IT	Read 0x0: THR interrupt inactive Read 0x1: THR interrupt active	R	0
0	RHR_IT	Read 0x0: RHR interrupt inactive Read 0x1: RHR interrupt active	R	0

**Table 23-178. UART\_DLH**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	UART
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .		
<b>Description</b>	This register, with <a href="#">UART_DLL</a> , stores the 14-bit divisor for generating the baud clock in the baud rate generator. DLH, stores the most-significant part of the divisor. DLL stores the least-significant part of the divisor.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		CLOCK_MSB													

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:6	RESERVED	Read returns 0. Write has no effect.	RW	0x0
5:0	CLOCK_MSB	Stores the 6-bit MSB divisor value	RW	0x00

**Table 23-179. Register Call Summary for Register UART\_DLH**

UART/IrDA/CIR

- [Module Power Saving](#): [0] [1]
- [DMA Transfers \(DMA Mode 1, 2, or 3\)](#): [2]
- [Registers Available for the Register Access Modes](#): [3] [4] [5] [6]
- [Registers Available for the UART Function](#): [7] [8] [9] [10]
- [Registers Available for the IrDA Function \(UART3 Only\)](#): [11] [12] [13] [14]
- [Registers Available for the CIR Function \(UART3 Only\)](#): [15] [16] [17] [18]
- [UART Clock Generation: Baud Rate Generation](#): [19]
- [UART Data Formatting](#): [20] [21]
- [IrDA Clock Generation: Baud Generator](#): [22]
- [Protocol, Baud Rate, and Interrupt Settings](#): [23] [24] [25] [26] [27]
- [Receive](#): [28] [29] [30] [31]
- [Transmit](#): [32] [33] [34] [35]
- [Receive](#): [36] [37] [38] [39]
- [Transmit](#): [40] [41] [42] [43]
- [Receive](#): [44] [45] [46]
- [Transmit](#): [47]
- [UART/IrDA/CIR Register Summary](#): [48] [49]
- [UART/IrDA/CIR Registers](#): [50] [51]

**Table 23-180. UART\_FCR**

<b>Address Offset</b>	0x0000 0008		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	FIFO control register <b>Notes:</b> Bits 4 and 5 can only be written to when <a href="#">UART_EFR[4]</a> = 1. Bits 0 and 3 can be changed only when the baud clock is not running (DLL and DLH set to 0) See <a href="#">Table 23-149</a> for <a href="#">UART_FCR[5:4]</a> setting restriction when <a href="#">UART_SCR[6]</a> = 1 See <a href="#">Table 23-150</a> for <a href="#">UART_FCR[7:6]</a> setting restriction when <a href="#">UART_SCR[7]</a> = 1.		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FIFO_TRIG		TX_FIFO_TRIG		DMA_MODE	TX_FIFO_CLEAR	RX_FIFO_CLEAR	FIFO_EN								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:6	RX_FIFO_TRIG	Sets the trigger level for the RX FIFO: If <a href="#">UART_SCR[7]</a> = 0 and <a href="#">UART_TLR[7:4]</a> = 0000: 00: 8 characters 01: 16 characters 10: 56 characters 11: 60 characters If <a href="#">UART_SCR[7]</a> = 0 and <a href="#">UART_TLR[7:4]</a> != 0000, RX_FIFO_TRIG is not considered. If <a href="#">UART_SCR[7]</a> = 1, RX_FIFO_TRIG is 2 LSB of the trigger level (1-63 on 6 bits) with the granularity 1.	W	0x0
5:4	TX_FIFO_TRIG	Sets the trigger level for the TX FIFO: If <a href="#">UART_SCR[6]</a> = 0 and <a href="#">UART_TLR[3:0]</a> = 0000: 00: 8 spaces 01: 16 spaces 10: 32 spaces 11: 56 spaces If <a href="#">UART_SCR[6]</a> = 0 and <a href="#">UART_TLR[3:0]</a> != 0000, TX_FIFO_TRIG is not considered. If <a href="#">UART_SCR[6]</a> = 1, TX_FIFO_TRIG is 2 LSB of the trigger level (1-63 on 6 bits) with the granularity 1	W	0x0
3	DMA_MODE	This register is considered if <a href="#">UART_SCR[0]</a> = 0. Write 0x0: DMA_MODE 0 (No DMA) Write 0x1: DMA_MODE 1 (UART_nDMA_REQ[0] in TX, UART_nDMA_REQ[1] in RX)	W	0
2	TX_FIFO_CLEAR	Write 0x0: No change Write 0x1: Clears the TX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.	W	0
1	RX_FIFO_CLEAR	Write 0x0: No change Write 0x1: Clears the RX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.	W	0
0	FIFO_EN	Write 0x0: Disables the transmit and RX FIFOs. The transmit and receive holding registers are one-byte FIFOs. Write 0x1: Enables the transmit and RX FIFOs. The transmit and receive holding registers are 64-byte FIFOs.	W	0

**Table 23-181. Register Call Summary for Register UART\_FCR**

- UART/IrDA/CIR
- FIFO Management: [0]
  - Transmit FIFO Trigger: [1]
  - Receive FIFO Trigger: [2] [3]
  - FIFO Interrupt Mode: [4] [5] [6]
  - FIFO Polled Mode Operation: [7]
  - FIFO DMA Mode Operation: [8] [9] [10]
  - DMA Transfers (DMA Mode 1, 2, or 3): [11]
  - Registers Available for the Register Access Modes: [12] [13]
  - Registers Available for the UART Function: [14] [15]
  - Registers Available for the IrDA Function (UART3 Only): [16] [17]
  - Registers Available for the CIR Function (UART3 Only): [18] [19]
  - FIFOs and DMA Settings: [20] [21] [22] [23] [24] [25] [26] [27]
  - Receive: [28] [29] [30]
  - Transmit: [31] [32] [33]
  - UART/IrDA/CIR Register Summary: [34] [35]
  - UART/IrDA/CIR Registers: [36] [37] [38] [39] [40] [41] [42] [43]

**Table 23-182. UART\_EFR**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	UART
<b>Physical Address</b>	See Table 23-166 to Table 23-167.		
<b>Description</b>	Enhanced feature register This register enables or disables enhanced features. Most of the enhanced functions apply only to UART modes, but <a href="#">UART_EFR[4]</a> enables write accesses to <a href="#">UART_FCR[5:4]</a> , the TX trigger level, which is also used in IrDA modes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AUTO_CTS_EN		AUTO_RTS_EN		SPECIAL_CHAR_DETECT		ENHANCED_EN		SW_FLOW_CONTROL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	AUTO_CTS_EN	Auto-CTS enable bit 0x0: Normal operation 0x1: Auto-CTS flow control is enabled. Transmission is halted when the CTS* pin is high (inactive).	RW	0
6	AUTO_RTS_EN	Auto-RTS enable bit 0x0: Normal operation 0x1: Auto-RTS flow control is enabled. RTS* pin goes high (inactive) when the RX FIFO HALT trigger level, <a href="#">UART_TCR[3:0]</a> , is reached, and goes low (active) when the RX FIFO RESTORE transmission trigger level is reached.	RW	0

UART/IrDA/CIR

www.ti.com

Bits	Field Name	Description	Type	Reset
5	SPECIAL_CHAR_DETECT	0x0: Normal operation  0x1: Special character detect enable. Received data is compared with XOFF2 data. If a match occurs, the received data is transferred to the RX FIFO and the <a href="#">UART_IIR[4]</a> bit is set to 1 to indicate that a special character was detected.	RW	0
4	ENHANCED_EN	Enhanced functions write enable bit  0x0: Disables writing to IER bits 4-7, <a href="#">UART_FCR</a> bits 4-5, and MCR bits 5-7.  0x1: Enables writing to IER bits 4-7, <a href="#">UART_FCR</a> bits 4-5, and MCR bits 5-7.	RW	0
3:0	SW_FLOW_CONTROL	Combinations of software flow control can be selected by programming bit 3 - bit 0. See <a href="#">Table 23-162</a> .	RW	0x0

**Table 23-183. Register Call Summary for Register UART\_EFR**

UART/IrDA/CIR

- [SIR Mode: \[0\] \[1\] \[2\]](#)
- [Module Power Saving: \[3\]](#)
- [Register Access Submode: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [Registers Available for the Register Access Modes: \[10\] \[11\]](#)
- [Registers Available for the UART Function: \[12\] \[13\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[14\] \[15\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[16\] \[17\]](#)
- [UART Data Formatting: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [IrDA Data Formatting: \[24\] \[25\] \[26\]](#)
- [FIFOs and DMA Settings: \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [Protocol, Baud Rate, and Interrupt Settings: \[32\] \[33\] \[34\] \[35\] \[36\]](#)
- [Hardware Flow Control Configuration: \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\] \[44\] \[45\]](#)
- [Software Flow Control Configuration: \[46\] \[47\] \[48\] \[49\] \[50\] \[51\] \[52\] \[53\] \[54\] \[55\] \[56\]](#)
- [Transmit: \[57\] \[58\] \[59\]](#)
- [Transmit: \[60\]](#)
- [UART/IrDA/CIR Register Summary: \[61\] \[62\]](#)
- [UART/IrDA/CIR Registers: \[63\] \[64\]](#)

**Table 23-184. UART\_LCR**

<b>Address Offset</b>	0x0000 000C																					
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b> UART																				
<b>Description</b>	Line control register  LCR[6:0] define transmission and reception parameters. <b>Note:</b> When LCR[6] is set to 1, the TX line is forced to 0 and remains in this state as long as LCR[6] = 1.																					
<b>Type</b>	RW																					
	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																					
	RESERVED															DIV_EN	BREAK_EN	PARITY_TYPE2	PARITY_TYPE1	PARITY_EN	NB_STOP	CHAR_LENGTH

Bits	Field Name	Description	Type	Reset																								
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000																								
7	DIV_EN	0x0: Normal operating condition 0x1: Divisor latch enable. Allows access to DLL, DLH, and other registers (see <a href="#">Table 23-166</a> and <a href="#">Table 23-167</a> )	RW	0																								
6	BREAK_EN	Break control bit 0x0: Normal operating condition 0x1: Forces the transmitter output to go low to alert the communication terminal	RW	0																								
5	PARITY_TYPE2	Selects the forced parity format (if <a href="#">UART_LCR[3]</a> = 1). If <a href="#">UART_LCR[5]</a> = 1 and <a href="#">UART_LCR[4]</a> = 0, the parity bit is forced to 1 in the transmitted and received data. If <a href="#">UART_LCR[5]</a> = 1 and <a href="#">UART_LCR[4]</a> = 1, the parity bit is forced to 0 in the transmitted and received data.	RW	0																								
		<table border="1"> <thead> <tr> <th><a href="#">UART_LCR[3]</a></th> <th><a href="#">UART_LCR[4]</a></th> <th><a href="#">UART_LCR[5]</a></th> <th>Parity</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>N/A</td> <td>N/A</td> <td>No parity</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Odd parity</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Even parity</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Forced 1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Forced 0</td> </tr> </tbody> </table>	<a href="#">UART_LCR[3]</a>	<a href="#">UART_LCR[4]</a>	<a href="#">UART_LCR[5]</a>	Parity	0	N/A	N/A	No parity	1	0	0	Odd parity	1	1	0	Even parity	1	0	1	Forced 1	1	1	1	Forced 0		
<a href="#">UART_LCR[3]</a>	<a href="#">UART_LCR[4]</a>	<a href="#">UART_LCR[5]</a>	Parity																									
0	N/A	N/A	No parity																									
1	0	0	Odd parity																									
1	1	0	Even parity																									
1	0	1	Forced 1																									
1	1	1	Forced 0																									
4	PARITY_TYPE1	0x0: Odd parity is generated (if <a href="#">UART_LCR[3]</a> = 1). 0x1: Even parity is generated (if <a href="#">UART_LCR[3]</a> = 1).	RW	0																								
3	PARITY_EN	0x0: No parity 0x1: A parity bit is generated during transmission and the receiver checks for received parity.	RW	0																								
2	NB_STOP	Specifies the number of stop-bits 0x0: 1 stop-bit (word length = 5, 6, 7, 8) 0x1: 1.5 stop-bits (word length = 5) 2 stop-bits (word length = 6, 7, 8)	RW	0																								
1:0	CHAR_LENGTH	Specifies the word length to be transmitted or received 0x0: 5 bits 0x1: 6 bits 0x2: 7 bits 0x3: 8 bits	RW	0x0																								

**Table 23-185. Register Call Summary for Register UART\_LCR**

## UART/IrDA/CIR

- [SIR Mode: \[0\]](#)
- [Operational Mode and Configuration Modes: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Registers Available for the Register Access Modes: \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [Registers Available for the UART Function: \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[25\] \[26\] \[27\] \[28\] \[29\] \[30\]](#)
- [UART Data Formatting: \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\]](#)
- [SIR Mode Data Formatting: \[38\] \[39\] \[40\]](#)
- [FIFOs and DMA Settings: \[41\] \[42\] \[43\] \[44\] \[45\] \[46\]](#)
- [Protocol, Baud Rate, and Interrupt Settings: \[47\] \[48\] \[49\] \[50\] \[51\] \[52\] \[53\] \[54\] \[55\] \[56\] \[57\] \[58\] \[59\] \[60\] \[61\] \[62\] \[63\]](#)
- [Hardware Flow Control Configuration: \[64\] \[65\] \[66\] \[67\] \[68\]](#)
- [Software Flow Control Configuration: \[69\] \[70\] \[71\] \[72\] \[73\] \[74\]](#)
- [Receive: \[75\] \[76\] \[77\] \[78\]](#)
- [Transmit: \[79\] \[80\] \[81\] \[82\] \[83\]](#)
- [Receive: \[84\] \[85\] \[86\] \[87\]](#)
- [Transmit: \[88\] \[89\] \[90\] \[91\]](#)
- [Receive: \[92\] \[93\] \[94\] \[95\] \[96\]](#)
- [Transmit: \[97\] \[98\]](#)
- [UART/IrDA/CIR Register Summary: \[99\] \[100\]](#)
- [UART/IrDA/CIR Registers: \[101\] \[102\] \[103\] \[104\] \[105\] \[106\] \[107\] \[108\] \[109\] \[110\] \[111\]](#)

**Table 23-186. UART\_XON1\_ADDR1**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	UART mode: XON1 character, IrDA mode: ADDR1 address		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XON_WORD1															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XON_WORD1	Stores the 8-bit XON1 character in UART modes and ADDR1 address 1 for IrDA modes	RW	0x00

**Table 23-187. Register Call Summary for Register UART\_XON1\_ADDR1**

## UART/IrDA/CIR

- [SIR Mode: \[0\]](#)
- [Registers Available for the Register Access Modes: \[1\] \[2\]](#)
- [Registers Available for the UART Function: \[3\] \[4\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[5\] \[6\]](#)
- [IrDA Data Formatting: \[7\]](#)
- [Software Flow Control Configuration: \[8\] \[9\]](#)
- [UART/IrDA/CIR Register Summary: \[10\] \[11\]](#)

**Table 23-188. UART\_MCR**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Modem control register MCR[3:0] controls the interface with the modem, data set, or peripheral device that emulates the modem.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RESERVED	TCR_TLR	XON_EN	LOOPBACK_EN	CD_STS_CH	RI_STS_CH	RTS	DTR	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	RESERVED	Read returns 0. Write has no effect.	RW	0
6	TCR_TLR	0x0: No action 0x1: Enables access to the <a href="#">UART_TCR</a> and <a href="#">UART_TLR</a> registers	RW	0
5	XON_EN	0x0: Disable XON any function. 0x1: Enable XON any function.	RW	0
4	LOOPBACK_EN	0x0: Normal operating mode 0x1: Enable local loopback mode (internal). In this mode, the MCR[3:0] signals are looped back into the <a href="#">UART_MSR[7:4]</a> bit field. The transmit output is looped back to the receive input internally	RW	0
3	CD_STS_CH	0x0: In loopback, forces DCD* input high and IRQ outputs to inactive state 0x1: In loopback, forces DCD* input low and IRQ outputs to inactive state	RW	0
2	RI_STS_CH	0x0: In loopback, forces RI* input high 0x1: In loopback, forces RI* input low	RW	0
1	RTS	In loopback, controls the <a href="#">UART_MSR[4]</a> bit. If auto-RTS is enabled, the RTS* output is controlled by hardware flow control. 0x0: Force RTS* output to inactive (high). 0x1: Force RTS* output to active (low).	RW	0
0	DTR	0x0: Force DTR* output to inactive (high). 0x1: Force DTR* output to active (low).	RW	0

**Table 23-189. Register Call Summary for Register UART\_MCR**

## UART/IrDA/CIR

- [Description: \[0\] \[1\]](#)
- [Register Access Submode: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [Registers Available for the Register Access Modes: \[8\] \[9\] \[10\] \[11\]](#)
- [Registers Available for the UART Function: \[12\] \[13\] \[14\] \[15\]](#)
- [UART Data Formatting: \[16\] \[17\]](#)
- [FIFOs and DMA Settings: \[18\] \[19\] \[20\] \[21\] \[22\]](#)
- [Hardware Flow Control Configuration: \[23\] \[24\] \[25\] \[26\] \[27\]](#)
- [Software Flow Control Configuration: \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\]](#)
- [Transmit: \[35\]](#)
- [Receive: \[36\] \[37\] \[38\]](#)
- [Transmit: \[39\]](#)
- [Transmit: \[40\]](#)
- [UART/IrDA/CIR Register Summary: \[41\] \[42\]](#)
- [UART/IrDA/CIR Registers: \[43\] \[44\] \[45\] \[46\] \[47\] \[48\] \[49\] \[50\]](#)

**Table 23-190. UART\_LSR**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	UART
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .		
<b>Description</b>	Line status register		
<b>Type</b>	R		

**UART Bit Field Details**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FIFO_STS	TX_SR_E	TX_FIFO_E	RX_BI	RX_FE	RX_PE	RX_OE	RX_FIFO_E								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	RX_FIFO_STS	Read 0x0: Normal operation Read 0x1: At least one parity error, framing error, or break indication in the RX FIFO. Bit 7 is cleared when no more errors are present in the RX FIFO.	R	0
6	TX_SR_E	Read 0x0: Transmitter hold (TX FIFO) and shift registers are not empty. Read 0x1: Transmitter hold (TX FIFO) and shift registers are empty.	R	1
5	TX_FIFO_E	Read 0x0: Transmit hold register (TX FIFO) is not empty. Read 0x1: Transmit hold register (TX FIFO) is empty. The transmission is not necessarily complete.	R	1
4	RX_BI	Read 0x0: No break condition Read 0x1: A break was detected while the data from the RX FIFO was received (for example, RX input was low for one character + 1 bit time frame).	R	0
3	RX_FE	Read 0x0: No framing error in data RX FIFO Read 0x1: Framing error occurred in data from RX FIFO (received data did not have a valid stop-bit).	R	0
2	RX_PE	Read 0x0: No parity error in data from RX FIFO Read 0x1: Parity error in data from RX FIFO	R	0



Bits	Field Name	Description	Type	Reset
1	RX_OE	Read 0x0: No overrun error Read 0x1: Overrun error occurred. Set when the character in the receive shift register is not transferred to the RX FIFO. This occurs only when the RX FIFO is full.	R	0
0	RX_FIFO_E	Read 0x0: No data in the RX FIFO Read 0x1: At least one data character in the RX FIFO	R	0

**Table 23-191. Register Call Summary for Register UART\_LSR**

UART/IrDA/CIR

- SIR Mode: [0]
- MIR Mode: [1]
- FIR Mode: [2]
- UART Mode Interrupt Management: [3] [4]
- FIFO Polled Mode Operation: [5]
- Registers Available for the Register Access Modes: [6] [7]
- Registers Available for the UART Function: [8] [9]
- Registers Available for the IrDA Function (UART3 Only): [10] [11]
- Registers Available for the CIR Function (UART3 Only): [12] [13]
- UART Data Formatting: [14] [15] [16] [17] [18] [19] [20]
- IrDA Data Formatting: [21] [22]
- MIR and FIR Mode Data Formatting: [23]
- UART/IrDA/CIR Register Summary: [24] [25]
- UART/IrDA/CIR Registers: [26]

**IrDA Bit Field Details**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																THR_EMPTY	STS_FIFO_FULL	RX_LAST_BYTE	FRAME_TOO_LONG	ABORT	CRC	STS_FIFO_E	RX_FIFO_E								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	THR_EMPTY	Read 0x0: Transmit holding register (TX FIFO) is not empty. Read 0x1: Transmit hold register (TX FIFO) is empty. The transmission is not necessarily complete.	R	1
6	STS_FIFO_FULL	Read 0x0: Status FIFO not full Read 0x1: Status FIFO full	R	0
5	RX_LAST_BYTE	Read 0x0: The RX FIFO (RHR) does not contain the last byte of the frame to be read. Read 0x1: The RX FIFO (RHR) contains the last byte of the frame to be read. This bit is set only when the last byte of a frame is available to be read. It determines the frame boundary. It is cleared on a single read of the LSR register. See the note below.	R	0
4	FRAME_TOO_LONG	Read 0x0: No frame-too-long error in frame	R	0

Bits	Field Name	Description	Type	Reset
		Read 0x1: Frame-too-long error in the frame at the top of the STATUS FIFO, (next character to be read). This bit is set to 1 when a frame exceeding the maximum length (set by RXFLH and RXFLL registers) is received. When this error is detected, current frame reception is terminated. Reception is stopped until the next START flag is detected.		
3	ABORT	Read 0x0: No abort pattern error in frame Read 0x1: Abort pattern is received. SIR and MIR: Abort pattern FIR: Illegal symbol	R	0
2	CRC	Read 0x0: No CRC error in frame Read 0x1: CRC error in the frame at the top of the STATUS FIFO (next character to be read)	R	0
1	STS_FIFO_E	Read 0x0: Status FIFO not empty Read 0x1: Status FIFO empty	R	1
0	RX_FIFO_E	Read 0x0: No data in the RX FIFO Read 0x1: At least one data character in the RX FIFO	R	1

### CIR Bit Field Details

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																THR_EMPTY	RESERVED	RX_STOP	RESERVED				RX_FIFO_E								

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	THR_EMPTY	Read 0x0: Transmit holding register (TX FIFO) is not empty. Read 0x1: Transmit hold register (TX FIFO) is empty. The transmission is not necessarily complete.	R	1
6	RESERVED	Not used in CIR mode	R	0
5	RX_STOP	The RX_STOP is generated based on the value set in the BOF Length register ( <a href="#">UART_EBLR</a> ). It is cleared on a single read of the <a href="#">UART_LSR</a> register. Read 0x0: Reception is ongoing or waiting for a new frame. Read 0x1: Reception is complete.	R	0
4:1	RESERVED	Not used in CIR mode	R	0x0
0	RX_FIFO_E	Read 0x0: At least one data character in the RX FIFO Read 0x1: No data in the RX FIFO	R	1

**Table 23-192. UART\_XON2\_ADDR2**

<b>Address Offset</b>	0x0000 0014	
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b> UART
<b>Description</b>	RW	
<b>Type</b>	RW	
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED		XON_WORD2
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>
31:8	RESERVED	Read returns 0. Write has no effect.
7:0	XON_WORD2	Stores the 8-bit XON2 character in UART modes and ADDR2 address 2 for IrDA modes
		<b>Type</b> RW
		<b>Reset</b> 0x000000
		0x00

**Table 23-193. Register Call Summary for Register UART\_XON2\_ADDR2**

UART/IrDA/CIR

- [SIR Mode: \[0\]](#)
- [Registers Available for the Register Access Modes: \[1\] \[2\]](#)
- [Registers Available for the UART Function: \[3\] \[4\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[5\] \[6\]](#)
- [IrDA Data Formatting: \[7\]](#)
- [Software Flow Control Configuration: \[8\] \[9\]](#)
- [UART/IrDA/CIR Register Summary: \[10\] \[11\]](#)

**Table 23-194. UART\_TCR**

<b>Address Offset</b>	0x0000 0018	
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b> UART
<b>Description</b>	Transmission control register This register stores the RX FIFO threshold levels to start/stop transmission during hardware/software flow control. Notes: Trigger levels from 0 - 60 bytes are available with a granularity of four. (Trigger level = 4 x [4-bit register value]) The programmer must ensure that <code>UART_TCR[3:0] &gt; UART_TCR[7:4]</code> when auto-RTS or software flow control is enabled to avoid a mis-operation of the device. In FIFO interrupt mode with flow control, the programmer must ensure that the trigger level to halt transmission is greater than or equal to the RX FIFO trigger level ( <code>UART_TLR[7:4]</code> or <code>UART_FCR[7:6]</code> ); otherwise, FIFO operation stalls. In FIFO DMA mode with flow control, this concept does not exist because a DMA request is sent each time a byte is received.	
<b>Type</b>	RW	
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED		RX_FIFO_TRIG_START RX_FIFO_TRIG_HALT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:4	RX_FIFO_TRIG_START	RX FIFO trigger level to RESTORE transmission (0 - 60)	RW	0x0
3:0	RX_FIFO_TRIG_HALT	RX FIFO trigger level to HALT transmission (0 - 60)	RW	0xF

**Table 23-195. Register Call Summary for Register UART\_TCR**

UART/IrDA/CIR

- [Receive FIFO Trigger: \[0\] \[1\] \[2\]](#)
- [FIFO Interrupt Mode: \[3\]](#)
- [Registers Available for the Register Access Modes: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [Registers Available for the UART Function: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[16\] \[17\] \[18\] \[19\] \[20\] \[21\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[22\] \[23\] \[24\] \[25\] \[26\] \[27\]](#)
- [UART Data Formatting: \[28\] \[29\] \[30\] \[31\] \[32\] \[33\]](#)
- [Hardware Flow Control Configuration: \[34\] \[35\] \[36\] \[37\] \[38\] \[39\]](#)
- [Software Flow Control Configuration: \[40\] \[41\] \[42\] \[43\] \[44\] \[45\]](#)
- [UART/IrDA/CIR Register Summary: \[46\] \[47\]](#)
- [UART/IrDA/CIR Registers: \[48\] \[49\] \[50\] \[51\]](#)

**Table 23-196. UART\_XOFF1**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	UART
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .		
<b>Description</b>	UART mode XOFF1 character		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XOFF_WORD1															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XOFF_WORD1	Stores the 8-bit XOFF1 character used in UART modes	RW	0x00

**Table 23-197. Register Call Summary for Register UART\_XOFF1**

UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\]](#)
- [Registers Available for the UART Function: \[2\] \[3\]](#)
- [Software Flow Control Configuration: \[4\] \[5\] \[6\]](#)
- [UART/IrDA/CIR Register Summary: \[7\] \[8\]](#)

**Table 23-198. UART\_MSR**

<b>Address Offset</b>	0x0000 0018		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Modem status register. UART mode only. This register provides information about the current state of the control lines from the modem, data set, or peripheral device to the LH. It also indicates when a control input from the modem changes state.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								NCD_STS	NRI_STS	NDSR_STS	NCTS_STS	DCD_STS	RI_STS	DSR_STS	CTS_STS

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	NCD_STS	This bit is the complement of the DCD* input. In loopback mode, it is equivalent to <a href="#">UART_MCR[3]</a> .	R	-
6	NRI_STS	This bit is the complement of the RI* input. In loopback mode, it is equivalent to <a href="#">UART_MCR[2]</a> .	R	-
5	NDSR_STS	This bit is the complement of the DSR* input. In loop-back mode, it is equivalent to <a href="#">UART_MCR[0]</a> .	R	-
4	NCTS_STS	This bit is the complement of the CTS* input. In loopback mode, it is equivalent to <a href="#">UART_MCR[1]</a> .	R	-
3	DCD_STS	Indicates that DCD* input (or <a href="#">UART_MCR[3]</a> in loopback) changed. Cleared on a read.	R	0
2	RI_STS	Indicates that RI* input (or <a href="#">UART_MCR[2]</a> in loopback) changed state from low to high. Cleared on a read.	R	0
1	DSR_STS	Read 0x1: Indicates that DSR* input (or <a href="#">UART_MCR[0]</a> in loopback) changed state. Cleared on a read.	R	0
0	CTS_STS	Read 0x1: Indicates that CTS* input (or <a href="#">UART_MCR[1]</a> in loopback) changed state. Cleared on a read.	R	0

**Table 23-199. Register Call Summary for Register UART\_MSR**

UART/IrDA/CIR

- [Description](#): [0] [1]
- [UART Mode Interrupt Management](#): [2] [3]
- [Registers Available for the Register Access Modes](#): [4] [5]
- [Registers Available for the UART Function](#): [6] [7]
- [Registers Available for the IrDA Function \(UART3 Only\)](#): [8] [9]
- [Registers Available for the CIR Function \(UART3 Only\)](#): [10] [11]
- [UART/IrDA/CIR Register Summary](#): [12] [13]
- [UART/IrDA/CIR Registers](#): [14] [15]

**Table 23-200. UART\_SPR**

<b>Address Offset</b>	0x0000 001C		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Scratchpad register This read/write register does not control the module. It is a scratchpad register to be used by the programmer to hold temporary data.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SPR_WORD															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	SPR_WORD	Scratchpad register	RW	0x00

**Table 23-201. Register Call Summary for Register UART\_SPR**

UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\] \[2\] \[3\]](#)
- [Registers Available for the UART Function: \[4\] \[5\] \[6\] \[7\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[8\] \[9\] \[10\] \[11\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[12\] \[13\] \[14\] \[15\]](#)
- [UART/IrDA/CIR Register Summary: \[16\] \[17\]](#)

**Table 23-202. UART\_TLR**

<b>Address Offset</b>	0x0000 001C		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Trigger level register This register stores the programmable transmit and RX FIFO trigger levels for DMA and IRQ generation.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FIFO_TRIG_DMA				TX_FIFO_TRIG_DMA											

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:4	RX_FIFO_TRIG_DMA	Receive FIFO trigger level	RW	0x0
3:0	TX_FIFO_TRIG_DMA	Transmit FIFO trigger level	RW	0x0

**Table 23-203. Register Call Summary for Register UART\_TLR**

UART/IrDA/CIR

- FIFO Management: [0]
- Transmit FIFO Trigger: [1] [2]
- Receive FIFO Trigger: [3] [4]
- FIFO Interrupt Mode: [5] [6]
- DMA Transfers (DMA Mode 1, 2, or 3): [7] [8] [9] [10]
- Registers Available for the Register Access Modes: [11] [12] [13] [14] [15] [16]
- Registers Available for the UART Function: [17] [18] [19] [20] [21] [22]
- Registers Available for the IrDA Function (UART3 Only): [23] [24] [25] [26] [27] [28]
- Registers Available for the CIR Function (UART3 Only): [29] [30] [31] [32] [33] [34]
- FIFOs and DMA Settings: [35] [36] [37] [38] [39] [40]
- UART/IrDA/CIR Register Summary: [41] [42]
- UART/IrDA/CIR Registers: [43] [44] [45] [46] [47] [48]

**Table 23-204. UART\_XOFF2**

<b>Address Offset</b>	0x0000 001C																																																													
<b>Physical Address</b>	See Table 23-166 to Table 23-167.	<b>Instance</b> UART																																																												
<b>Description</b>	UART mode XOFF2 character																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td style="background-color:yellow;">23</td><td style="background-color:yellow;">22</td><td style="background-color:yellow;">21</td><td style="background-color:yellow;">20</td><td style="background-color:yellow;">19</td><td style="background-color:yellow;">18</td><td style="background-color:yellow;">17</td><td style="background-color:yellow;">16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td style="background-color:yellow;">7</td><td style="background-color:yellow;">6</td><td style="background-color:yellow;">5</td><td style="background-color:yellow;">4</td><td style="background-color:yellow;">3</td><td style="background-color:yellow;">2</td><td style="background-color:yellow;">1</td><td style="background-color:yellow;">0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="12">XOFF_WORD2</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																XOFF_WORD2											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
RESERVED																XOFF_WORD2																																														
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																										
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000																																																										
7:0	XOFF_WORD2	Stores the 8-bit XOFF2 character used in UART modes.	RW	0x00																																																										

**Table 23-205. Register Call Summary for Register UART\_XOFF2**

UART/IrDA/CIR

- Registers Available for the Register Access Modes: [0] [1]
- Registers Available for the UART Function: [2] [3]
- Software Flow Control Configuration: [4] [5] [6]
- UART/IrDA/CIR Register Summary: [7] [8]

**Table 23-206. UART\_MDR1**

<b>Address Offset</b>	0x0000 0020	
<b>Physical Address</b>	See Table 23-166 to Table 23-167.	<b>Instance</b> UART
<b>Description</b>	<p>Mode definition register 1</p> <p>The mode of operation can be programmed by writing to MDR1[2:0] and therefore the MDR1 must be programmed on startup after configuration of the configuration registers (UART_DLL, UART_DLH, and UART_LCR). The value of MDR1[2:0] must not be changed again during normal operation. <b>Note:</b> If the module is disabled by setting the MODE_SELECT field to 111, interrupt requests can still be generated unless disabled through the interrupt enable register (IER). In this case, UART mode interrupts are visible. Reading the interrupt identification register (IIR) shows UART mode interrupt flags.</p>	
<b>Type</b>	RW	

## UART/IrDA/CIR

www.ti.com

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FRAME_END_MODE	SIP_MODE	SCT	SET_TXIR	IR_SLEEP	MODE_SELECT										

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	FRAME_END_MODE	IrDA mode only 0x0: Frame-length method 0x1: Set EOT bit method	RW	0
6	SIP_MODE	MIR/FIR modes only 0x0: Manual SIP mode: SIP is generated with the control of ACREG[3]. 0x1: Automatic SIP mode: SIP is generated after each transmission.	RW	0
5	SCT	Store and control the transmission. 0x0: Starts the infrared transmission when a value is written to THR 0x1: Starts the infrared transmission with the control of ACREG[2]. <b>Note:</b> Before starting any transmission, there must be no reception ongoing.	RW	0
4	SET_TXIR	Used to configure the infrared transceiver 0x0: a) No action if MDR2[7] = 0 b) TXIR pin output is forced low if MDR2[7] = 1. 0x1: IRTX pin output is forced high (not dependent on MDR2[7] value).	RW	0
3	IR_SLEEP	0x0: IrDA/CIR sleep mode disabled 0x1: IrDA/CIR sleep mode enabled	RW	0
2:0	MODE_SELECT	0x0: UART 16x mode 0x1: SIR mode 0x2: UART 16x auto-baud 0x3: UART 13x mode 0x4: MIR mode 0x5: FIR mode 0x6: CIR mode 0x7: Disable (default state)	RW	0x7



**Table 23-207. Register Call Summary for Register UART\_MDR1**

UART/IrDA/CIR

- Description: [0] [1]
- UART Protocol and Data Format: [2]
- Registers Available for the Register Access Modes: [3] [4] [5] [6] [7] [8]
- UART/IrDA (SIR, MIR, FIR)/CIR Mode Selection: [9]
- Registers Available for the UART Function: [10] [11] [12] [13] [14] [15]
- Registers Available for the IrDA Function (UART3 Only): [16] [17] [18] [19] [20] [21]
- Registers Available for the CIR Function (UART3 Only): [22] [23] [24] [25] [26] [27]
- UART Clock Generation: Baud Rate Generation: [28]
- UART Data Formatting: [29] [30]
- IrDA Clock Generation: Baud Generator: [31]
- IrDA Data Formatting: [32] [33]
- SIR Mode Data Formatting: [34] [35]
- MIR and FIR Mode Data Formatting: [36]
- CIR Data Formatting: [37]
- Protocol, Baud Rate, and Interrupt Settings: [38] [39] [40]
- Receive: [41] [42]
- Transmit: [43] [44] [45]
- Receive: [46] [47]
- Transmit: [48] [49]
- Receive: [50] [51]
- Transmit: [52] [53]
- UART/IrDA/CIR Register Summary: [54] [55]
- UART/IrDA/CIR Registers: [56]

**Table 23-208. UART\_MDR2**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	UART
<b>Physical Address</b>	See Table 23-166 to Table 23-167.		
<b>Description</b>	<p>Mode definition register 2</p> <p>IR-IrDA and IR-CIR modes only. <b>UART_MDR2</b>[0] describes the status of the interrupt in <b>UART_IIR</b>[5]. The IRTX_UNDERRUN bit should be read after an <b>UART_IIR</b>[5] TX_STATUS_IT interrupt. The bits [2:1] of this register set the trigger level for the frame status FIFO (8 entries) and must be programmed before the mode is programmed in <b>UART_MDR1</b>[2:0].</p> <p><b>Note:</b> The <b>UART_MDR2</b>[6] gives the flexibility to invert the RX pin in the UART to ensure that the protocol at the input of the transceiver module has the same polarity at module level. By default, the RX pin is inverted because most transceivers invert the IR receive pin.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SET_TXIR_ALT	IRRXINVERT	CIR_PULSE_MODE	UART_PULSE	STS_FIFO_TRIG	IRTX_UNDERRUN										

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	SET_TXIR_ALT	Provide alternate function for MDR1[4] (SET_TXIR). 0x0: Normal mode 0x1: Alternate mode for SET_TXIR	R	0

Bits	Field Name	Description	Type	Reset
6	IRRXINVERT	IR mode only (IRDA and CIR). Invert RX pin in the module before the voting or sampling system logic of the infrared block. This does not affect the RX path in UART modem modes.  0x0: Inversion is performed. 0x1: No inversion is performed.	RW	0
5:4	CIR_PULSE_MODE	CIR pulse modulation definition. Defines high level of the pulse width associated with a digit: 0x0: Pulse width of 3 from 12 cycles 0x1: Pulse width of 4 from 12 cycles 0x2: Pulse width of 5 from 12 cycles 0x3: Pulse width of 6 from 12 cycles	RW	0x0
3	UART_PULSE	UART mode only. Allows pulse shaping in UART mode. 0x0: Normal UART mode 0x1: UART mode with a pulse shaping	RW	0
2:1	STS_FIFO_TRIG	IR-IRDA mode only. Frame status FIFO threshold select: 0x0: 1 entry 0x1: 4 entries 0x2: 7 entries 0x3: 8 entries	RW	0x0
0	IRTX_UNDERRUN	IRDA transmission status interrupt. When the <a href="#">UART_IIR[5]</a> interrupt occurs, the meaning of the interrupt is :  Read 0x0: The last bit of the frame transmitted successfully without error.  Read 0x1: An underrun occurred. The last bit of the frame was transmitted but with an underrun error. The bit is reset to 0 when the <a href="#">UART_RESUME</a> register is read.	R	0

**Table 23-209. Register Call Summary for Register UART\_MDR2**

## UART/IrDA/CIR

- [SIR Mode: \[0\]](#)
- [Pulse Duty Cycle: \[1\]](#)
- [Registers Available for the Register Access Modes: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [Registers Available for the UART Function: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[20\] \[21\] \[22\] \[23\] \[24\] \[25\]](#)
- [IrDA Data Formatting: \[26\]](#)
- [SIR Mode Data Formatting: \[27\]](#)
- [CIR Mode Clock Generation: \[28\] \[29\]](#)
- [UART/IrDA/CIR Register Summary: \[30\] \[31\]](#)
- [UART/IrDA/CIR Registers: \[32\] \[33\]](#)

**Table 23-210. UART\_SFLSR**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a>	<b>Instance</b>	UART
<b>Description</b>	Status FIFO line status register IrDA modes only. Reading this register effectively reads frame status information from the status FIFO (this register does not physically exist). Reading this register increments the status FIFO read pointer (SFREGL and SFREGH must be read first).		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED		OE_ERROR	FRAME_TOO_LONG_ERROR	ABORT_DETECT	CRC_ERROR	RESERVED									

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:5	RESERVED	Read returns 0.	R	0x0
4	OE_ERROR	Read 0x1: Overrun error in RX FIFO when frame at top of RX FIFO was received <b>Note:</b> Top of RX FIFO = Next frame to be read from RX FIFO	R	-
3	FRAME_TOO_LONG_ERROR	Read 0x1: Frame-length too long error in frame at top of RX FIFO	R	-
2	ABORT_DETECT	Read 0x1: Abort pattern detected in frame at top of RX FIFO	R	-
1	CRC_ERROR	Read 0x1: CRC error in frame at top of RX FIFO	R	-
0	RESERVED		R	0

**Table 23-211. Register Call Summary for Register UART\_SFLSR**

UART/IrDA/CIR

- [FIFO Management](#): [0]
- [Registers Available for the Register Access Modes](#): [1] [2] [3]
- [Registers Available for the IrDA Function \(UART3 Only\)](#): [4] [5] [6]
- [IrDA Data Formatting](#): [7] [8]
- [UART/IrDA/CIR Register Summary](#): [9] [10]
- [UART/IrDA/CIR Registers](#): [11] [12]

**Table 23-212. UART\_TXFLL**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Transmit frame length register low IrDA modes only. The <a href="#">UART_TXFLL</a> and <a href="#">UART_TXFLH</a> registers hold the 13-bit transmit frame length (expressed in bytes). <a href="#">UART_TXFLL</a> holds the LSBs and <a href="#">UART_TXFLH</a> holds the MSBs. The frame length value is used if the frame length method of frame closing is used.		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TXFLL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:0	TXFLL	LSB register used to specify the frame length	W	0x00

**Table 23-213. Register Call Summary for Register UART\_TXFLL**

## UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\] \[2\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[3\] \[4\] \[5\]](#)
- [IrDA Data Formatting: \[6\]](#)
- [Transmit: \[7\]](#)
- [Transmit: \[8\]](#)
- [Transmit: \[9\]](#)
- [UART/IrDA/CIR Register Summary: \[10\] \[11\]](#)
- [UART/IrDA/CIR Registers: \[12\] \[13\] \[14\] \[15\]](#)

**Table 23-214. UART\_RESUME**

<b>Address Offset</b>	0x0000 002C		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	IR-IrDA and IR-CIR modes only. This register is used to clear internal flags, which halt transmission/reception when an underrun/overflow error occurs. Reading this register resumes the halted operation. This register does not physically exist and reads always as 0x00.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESUME															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	RESUME	Dummy read to restart the TX or RX	R	0x00

**Table 23-215. Register Call Summary for Register UART\_RESUME**

UART/IrDA/CIR

- IrDA Interrupts: [0] [1]
- Registers Available for the Register Access Modes: [2] [3] [4]
- Registers Available for the IrDA Function (UART3 Only): [5] [6] [7]
- Registers Available for the CIR Function (UART3 Only): [8] [9] [10]
- UART Data Formatting: [11]
- IrDA Data Formatting: [12]
- UART/IrDA/CIR Register Summary: [13] [14]
- UART/IrDA/CIR Registers: [15]

**Table 23-216. UART\_TXFLH**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	UART
<b>Physical Address</b>	See Table 23-166 to Table 23-167.		
<b>Description</b>	Transmit frame length register high IrDA modes only. The <code>UART_TXFLH</code> and <code>UART_TXFLH</code> registers hold the 13-bit transmit frame length (expressed in bytes). <code>UART_TXFLH</code> holds the LSBs and <code>UART_TXFLH</code> holds the MSBs. The frame length value is used if the frame length method of frame closing is used.		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED			TXFLH												

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:5	RESERVED	Write has no effect.	W	0x0
4:0	TXFLH	MSB register used to specify the frame length	W	0x00

**Table 23-217. Register Call Summary for Register UART\_TXFLH**

UART/IrDA/CIR

- Registers Available for the Register Access Modes: [0] [1] [2]
- Registers Available for the IrDA Function (UART3 Only): [3] [4] [5]
- IrDA Data Formatting: [6]
- Transmit: [7]
- UART/IrDA/CIR Register Summary: [8] [9]
- UART/IrDA/CIR Registers: [10] [11] [12] [13]

**Table 23-218. UART\_SFREGL**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Status FIFO register low IrDA modes only. The frame lengths of received frames are written into the status FIFO. This information can be read by reading the <a href="#">UART_SFREGL</a> and <a href="#">UART_SFREGH</a> registers (these registers do not physically exist). The LsBs are read from <a href="#">UART_SFREGL</a> and the MSBs are read from <a href="#">UART_SFREGH</a> . Reading these registers does not alter the status FIFO read pointer. These registers should be read before the pointer is incremented by reading the <a href="#">UART_SFLSR</a> register.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SFREGL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	SFREGL	LSB part of the frame length	R	0x-

**Table 23-219. Register Call Summary for Register UART\_SFREGL**

UART/IrDA/CIR

- [FIFO Management: \[0\]](#)
- [Registers Available for the Register Access Modes: \[1\] \[2\] \[3\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[4\] \[5\] \[6\]](#)
- [IrDA Data Formatting: \[7\]](#)
- [UART/IrDA/CIR Register Summary: \[8\] \[9\]](#)
- [UART/IrDA/CIR Registers: \[10\] \[11\] \[12\] \[13\]](#)

**Table 23-220. UART\_RXFLL**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Received frame length register low IrDA modes only. The <a href="#">UART_RXFLL</a> and <a href="#">UART_RXFLH</a> registers hold the 12-bit receive maximum frame length. <a href="#">UART_RXFLL</a> holds the LSBs and <a href="#">UART_RXFLH</a> holds the MSBs. If the intended maximum receive frame length is n bytes, program the <a href="#">UART_RXFLL</a> and <a href="#">UART_RXFLH</a> registers to be n + 3 in SIR or MIR modes and n + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; two bytes are associated with the FIR stop flag).		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RXFLL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:0	RXFLL	LSB register used to specify the frame length in reception	W	0x00

**Table 23-221. Register Call Summary for Register UART\_RXFLL**

UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\] \[2\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[3\] \[4\] \[5\]](#)
- [Receive: \[6\]](#)
- [Transmit: \[7\]](#)
- [UART/IrDA/CIR Register Summary: \[8\] \[9\]](#)
- [UART/IrDA/CIR Registers: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

**Table 23-222. UART\_SFREGH**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Status FIFO register high IrDA modes only. The frame lengths of received frames are written into the status FIFO. This information can be read by reading the <a href="#">UART_SFREGL</a> and <a href="#">UART_SFREGH</a> registers (these registers do not physically exist). The LSBs are read from <a href="#">UART_SFREGL</a> and the MSBs are read from <a href="#">UART_SFREGH</a> . Reading these registers does not alter the status FIFO read pointer. These registers should be read before the pointer is incremented by reading the <a href="#">UART_SFLSR</a> register.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				SFREGH											

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:4	RESERVED	Read returns 0.	R	0x0
3:0	SFREGH	MSB part of the frame length	R	0x-

**Table 23-223. Register Call Summary for Register UART\_SFREGH**

UART/IrDA/CIR

- [FIFO Management: \[0\]](#)
- [Registers Available for the Register Access Modes: \[1\] \[2\] \[3\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[4\] \[5\] \[6\]](#)
- [IrDA Data Formatting: \[7\]](#)
- [UART/IrDA/CIR Register Summary: \[8\] \[9\]](#)
- [UART/IrDA/CIR Registers: \[10\] \[11\] \[12\] \[13\]](#)

**Table 23-224. UART\_RXFLH**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	<p>Received frame length register high</p> <p>IrDA modes only. The <a href="#">UART_RXFLH</a> and <a href="#">UART_RXFLH</a> registers hold the 12-bit receive maximum frame length. <a href="#">UART_RXFLH</a> holds the LSBs and <a href="#">UART_RXFLH</a> holds the MSBs. If the intended maximum receive frame length is n bytes, program the <a href="#">UART_RXFLH</a> and <a href="#">UART_RXFLH</a> to be n + 3 in SIR or MIR modes and n + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; two bytes are associated with the FIR stop flag).</p>		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				RXFLH											

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:4	RESERVED	Write has no effect.	W	0x0
3:0	RXFLH	MSB register used to specify the frame length in reception	W	0x0

**Table 23-225. Register Call Summary for Register UART\_RXFLH**

UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\] \[2\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[3\] \[4\] \[5\]](#)
- [Transmit: \[6\]](#)
- [UART/IrDA/CIR Register Summary: \[7\] \[8\]](#)
- [UART/IrDA/CIR Registers: \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

**Table 23-226. UART\_BLR**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	<p>BOF control register</p> <p>IrDA modes only. The <a href="#">UART_BLR</a>[6] bit selects whether 0xC0 or 0xFF start patterns are to be used, when multiple start flags are required in SIR mode. If only one start flag is required, this is always be 0xC0. If n start flags are required, (-1) 0xC0 or (-1) 0xFF flags are sent, followed by a single 0xC0 flag (immediately preceding the first data byte).</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STS_FIFO_RESET		XBOF_TYPE		RESERVED											



Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	STS_FIFO_RESET	Status FIFO reset. This bit is self-clearing.	RW	0
6	XBOF_TYPE	SIR xBOF select 0x0: 0xFF 0x1: 0xC0	RW	1
5:0	RESERVED		R	0x00

**Table 23-227. Register Call Summary for Register UART\_BLR**

UART/IrDA/CIR

- [SIR Mode: \[0\]](#)
- [Registers Available for the Register Access Modes: \[1\] \[2\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[3\] \[4\]](#)
- [UART/IrDA/CIR Register Summary: \[5\] \[6\]](#)
- [UART/IrDA/CIR Registers: \[7\]](#)

**Table 23-228. UART\_UASR**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	UART
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .		
<b>Description</b>	<p>UART autobauding status register</p> <p>UART autobauding mode only. This status register returns the speed, the number of bits by characters, and the type of the parity in UART autobauding mode. In autobauding mode, the input frequency of the UART modem must be fixed to 48 MHz. Any other module clock frequency results in incorrect baud rate recognition.</p> <p><b>Note:</b> When the UART is in autobauding mode, this register, instead of the LCR, DLL, and DLH registers, is used to set up transmission according to the characteristics of the previous reception. To reset the autobauding hardware (to start a new AT detection), set MDR1[2:0] to 111 (reset value), then set MDR1[2:1] to 010 (UART in autobaud mode). To set the UART to standard mode (no autobaud), set MDR1[2:1] to 000.</p>		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PARITY_TYPE		BIT_BY_CHAR		SPEED											

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:6	PARITY_TYPE	Read 0x0: No parity identified Read 0x1: Parity space Read 0x2: Even parity Read 0x3: Odd parity	R	0x0
5	BIT_BY_CHAR	Read 0x0: 7-bit character identified Read 0x1: 8-bit character identified	R	0
4:0	SPEED	Used to report the speed identified Read 0x0: No speed identified Read 0x1: 115 200 baud Read 0x2: 57 600 baud Read 0x3: 38 400 baud Read 0x4: 28 800 baud	R	0x00

Bits	Field Name	Description	Type	Reset
		Read 0x5: 19 200 baud		
		Read 0x6: 14 400 baud		
		Read 0x7: 9 600 baud		
		Read 0x8: 4 800 baud		
		Read 0x9: 2 400 baud		
		Read 0xA: 1 200 baud		

**Table 23-229. Register Call Summary for Register UART\_UASR**

UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\]](#)
- [Registers Available for the UART Function: \[2\] \[3\]](#)
- [UART Data Formatting: \[4\] \[5\] \[6\]](#)
- [UART/IrDA/CIR Register Summary: \[7\] \[8\]](#)

**Table 23-230. UART\_ACREG**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	UART
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .		
<b>Description</b>	Auxiliary control register. IR-IrDA and IR-CIR modes only		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RESERVED																											PULSE_TYPE	SD_MOD	DIS_IR_RX	DIS_TX_UNDERRUN	SEND_SIP	SCTX_EN	ABORT_EN	EOT_EN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	PULSE_TYPE	SIR pulse width select 0x0: 3/16 of baud-rate pulse width 0x1: 1.6 us	RW	0
6	SD_MOD	Primary output used to configure transceivers. Connected to the SD/MODE input pin of IrDA transceivers. 0x0: SD pin is set to high. 0x1: SD pin is set to low.	RW	0
5	DIS_IR_RX	0x0: Normal operation (RX input automatically disabled during transmit but enabled outside of transmit operation) 0x1: Disables RX input (permanent state - independent of transmit)	RW	0
4	DIS_TX_UNDERRUN	It is recommended to disable TX FIFO underrun capability by masking corresponding underrun interrupt. When disabling underrun by setting ACREG[4] = 1, garbage data is sent over TX line. 0x0: Long stop-bits cannot be transmitted; TX underrun is enabled. 0x1: Long stop-bits can be transmitted; TX underrun is disabled.	RW	0

Bits	Field Name	Description	Type	Reset
3	SEND_SIP	MIR/FIR modes only. Send serial infrared interaction pulse (SIP). If this bit is set during an MIR/FIR transmission, the SIP is sent at the end of it. This bit is cleared automatically at the end of the SIP transmission. 0x0: No action 0x1: Send SIP pulse.	RW	0
2	SCTX_EN	Store and controlled TX start. When MDR1[5] = 1 and the LH writes 1 to this bit, the TX state-machine starts frame transmission. This bit is self-clearing.	RW	0
1	ABORT_EN	Frame abort. The LH can intentionally abort transmission of a frame by writing 1 to this bit. Neither the end flag nor the CRC bits are appended to the frame. If TX FIFO is not empty and MDR1[5] = 1, UART IrDA starts a new transfer with data of the previous frame when the abort frame is sent. Therefore, TX FIFO must be reset before sending an abort frame.	RW	0
0	EOT_EN	EOT (end of transmission) bit. The LH writes 1 to this bit just before it writes the last byte to the TX FIFO in set-EOT bit frame closing method. This bit is cleared automatically when the LH writes to the THR (TX FIFO).	RW	0

**Table 23-231. Register Call Summary for Register UART\_ACREG**

UART/IrDA/CIR

- [UART3 Interface Description: \[0\]](#)
- [SIR Mode: \[1\] \[2\] \[3\] \[4\]](#)
- [CIR Interface Description: \[5\]](#)
- [Registers Available for the Register Access Modes: \[6\] \[7\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[8\] \[9\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[10\] \[11\]](#)
- [IrDA Data Formatting: \[12\] \[13\] \[14\] \[15\]](#)
- [SIR Mode Data Formatting: \[16\] \[17\]](#)
- [MIR and FIR Mode Data Formatting: \[18\]](#)
- [CIR Data Formatting: \[19\]](#)
- [Transmit: \[20\]](#)
- [Transmit: \[21\]](#)
- [Transmit: \[22\]](#)
- [UART/IrDA/CIR Register Summary: \[23\] \[24\]](#)

**Table 23-232. UART\_SCR**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	UART
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .		
<b>Description</b>	Supplementary control register <b>Note:</b> Bit 4 enables the wake-up interrupt, but this interrupt is not mapped into the <a href="#">UART_IIR</a> register. Therefore, when an interrupt occurs and there is no interrupt pending in the <a href="#">UART_IIR</a> register, the SSR[1] bit must be checked. To clear the wake-up interrupt, bit <a href="#">UART_SCR[4]</a> must be reset to 0.		
<b>Type</b>	RW		

UART/IrDA/CIR

www.ti.com

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_TRIG_GRANU1	TX_TRIG_GRANU1	DSR_IT	RX_CTS_DSR_WAKE_UP_ENABLE	TX_EMPTY_CTL_IT	DMA_MODE_2	DMA_MODE_CTL	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	RX_TRIG_GRANU1	0x0: Disables the granularity of 1 for trigger RX level 0x1: Enables the granularity of 1 for trigger RX level	RW	0
6	TX_TRIG_GRANU1	0x0: Disables the granularity of 1 for trigger TX level 0x1: Enables the granularity of 1 for trigger TX level	RW	0
5	DSR_IT	0x0: Disables DSR* interrupt 0x1: Enables DSR* interrupt	RW	0
4	RX_CTS_DSR_WAKE_UP_ENABLE	0x0: Disables the wake-up interrupt and clears SSR[1]  0x1: Waits for a falling edge of pins RX, CTS*, or DSR* to generate an interrupt	RW	0
3	TX_EMPTY_CTL_IT	0x0: Normal mode for THR interrupt (see UART mode interrupts table) 0x1: THE THR interrupt is generated when TX FIFO and TX shift register are empty.	RW	0
2:1	DMA_MODE_2	Used to specify the DMA mode valid if the <a href="#">UART_SCR</a> [0] bit = 1  0x0: DMA mode 0 (no DMA) 0x1: DMA mode 1 (UART_nDMA_REQ[0] in TX, UART_nDMA_REQ[1] in RX) 0x2: DMA mode 2 (UART_nDMA_REQ[0] in RX) 0x3: DMA mode 3 (UART_nDMA_REQ[0] in TX)	RW	0x0
0	DMA_MODE_CTL	0x0: The DMA_MODE is set with <a href="#">UART_FCR</a> [3]. 0x1: The DMA_MODE is set with <a href="#">UART_SCR</a> [2:1].	RW	0

**Table 23-233. Register Call Summary for Register UART\_SCR**

UART/IrDA/CIR

- [Module Power Saving: \[0\]](#)
- [Wake-Up Interrupt: \[1\] \[2\]](#)
- [FIFO Management: \[3\]](#)
- [Transmit FIFO Trigger: \[4\]](#)
- [Receive FIFO Trigger: \[5\]](#)
- [FIFO DMA Mode Operation: \[6\] \[7\]](#)
- [Registers Available for the Register Access Modes: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [Registers Available for the UART Function: \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[20\] \[21\] \[22\] \[23\] \[24\] \[25\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [FIFOs and DMA Settings: \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\]](#)
- [UART/IrDA/CIR Register Summary: \[40\] \[41\]](#)
- [UART/IrDA/CIR Registers: \[42\] \[43\] \[44\] \[45\] \[46\] \[47\] \[48\] \[49\] \[50\] \[51\] \[52\] \[53\] \[54\]](#)

**Table 23-234. UART\_SSR**

<b>Address Offset</b>	0x0000 0044		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Supplementary status register <b>Note:</b> Bit 1 is reset only when <a href="#">UART_SCR[4]</a> is reset to 0.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											DMA_COUNTER_RST	RX_CTS_DSR_WAKE_UP_STS	TX_FIFO_FULL		

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:3	RESERVED	Read returns 0.	R	0x00
2	DMA_COUNTER_RST	0x0: The DMA counter will not be reset if the corresponding FIFO is reset (through FCR[1] or FCR[2]). 0x1: The DMA counter will be reset if corresponding FIFO is reset (through FCR[1] or FCR[2]).	RW	1
1	RX_CTS_DSR_WAKE_UP_STS	Read 0x0: No falling edge event on RX, CTS*, and DSR* Read 0x1: A falling edge occurred on RX, CTS*, or DSR*.	R	0
0	TX_FIFO_FULL	Read 0x0: TX FIFO is not full. Read 0x1: TX FIFO is full.	R	0

**Table 23-235. Register Call Summary for Register UART\_SSR**

UART/IrDA/CIR

- [Wake-Up Interrupt](#): [0]
- [FIFO Management](#): [1]
- [Registers Available for the Register Access Modes](#): [2] [3] [4]
- [Registers Available for the UART Function](#): [5] [6] [7]
- [Registers Available for the IrDA Function \(UART3 Only\)](#): [8] [9] [10]
- [Registers Available for the CIR Function \(UART3 Only\)](#): [11] [12] [13]
- [UART/IrDA/CIR Register Summary](#): [14] [15]

**Table 23-236. UART\_EBLR**

<b>Address Offset</b>	0x0000 0048		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	<p>BOF length register</p> <p>IR-IrDA and IR-CIR modes only. In IR-IrDA SIR operation, this register specifies the number of BOF + xBOFs to transmit. Value set into this register must account for the BOF character; therefore, to send only one BOF with no XBOF, this register must be set to 1. To send one BOF with N XBOF, this register must be set to N + 1. The value 0 sends 1 BOF plus 255 XBOF. In IR-IrDA MIR mode, this register specifies the number of additional start flags (MIR protocol mandates a minimum of 2 start flags). In IR-CIR mode, this register specifies the number of consecutive 0s to be received before generating the RX_STOP interrupt (<a href="#">UART_IIR[2]</a>). All received 0s are stored in the RX FIFO. When the register is set to 0, this feature is deactivated and always in reception state, which can be disabled by setting the ACREG[5] to 1.</p> <p><b>Note:</b> If the RX_STOP interrupt occurs before a byte boundary, the remaining bits of the last byte are filled with 0s and passed into the RX FIFO.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EBLR															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	EBLR	<p>IR-IrDA mode: This register allows definition of up to 176 xBOFs, the maximum required by IrDA specification.</p> <p>IR-CIR mode: This register specifies the number of consecutive 0s to be received before generating the RX_STOP interrupt (<a href="#">UART_IIR[2]</a>).</p> <p>0x00: Feature disabled</p> <p>0x01: Generate RX_STOP interrupt after receiving one zero bit.</p> <p>...</p> <p>0xFF: generate RX_STOP interrupt after receiving 255 zero bits.</p>	RW	0x00

**Table 23-237. Register Call Summary for Register UART\_EBLR**

## UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[2\] \[3\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[4\] \[5\]](#)
- [Transmit: \[6\]](#)
- [Transmit: \[7\]](#)
- [Transmit: \[8\]](#)
- [UART/IrDA/CIR Register Summary: \[9\] \[10\]](#)
- [UART/IrDA/CIR Registers: \[11\]](#)

**Table 23-238. UART\_MVR**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Module version register The reset value is fixed by hardware and corresponds to the RTL revision of this module. A reset has no effect on the value returned.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															

Bits	Field Name	Description	Type	Reset
31:0	REV	Revision Number	R	0x-- TI Internal data

**Table 23-239. Register Call Summary for Register UART\_MVR**

UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\] \[2\]](#)
- [Registers Available for the UART Function: \[3\] \[4\] \[5\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[6\] \[7\] \[8\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[9\] \[10\] \[11\]](#)
- [UART/IrDA/CIR Register Summary: \[12\] \[13\]](#)

**Table 23-240. UART\_SYSC**

<b>Address Offset</b>	0x0000 0054		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	System configuration register The auto idle bit controls a power-saving technique to reduce the logic power consumption of the open-core protocol (OCP) interface. When the feature is enabled, the clock is gated off until an OCP command for this device is detected. When the software reset bit is set high, it causes a full device reset.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	IDLEMODE	ENAWAKEUP	SOFTRESET	AUTOIDLE											

UART/IrDA/CIR

www.ti.com

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:5	RESERVED	Read returns 0.	R	0x0
4:3	IDLEMODE	Power management req/ack control ref: OCP Design Guidelines Version 1.1  0x0: Force idle: Idle request is acknowledged unconditionally. 0x1: No-idle: Idle request is never acknowledged. 0x2: Smart idle: Idle request is acknowledged based in module internal activity. 0x3: Smart idle Wakeup: Acknowledgement to an idle request is given based in the internal activity of the module. The module is allowed to generate wakeup request.	RW	0x0
2	ENAWAKEUP	Wake-Up feature control 0x0: Wakeup is disabled. 0x1: Wake-Up capability is enabled.	RW	0
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. This bit is automatically reset by the hardware. Read returns 0. 0x0: Normal mode 0x1: The module is reset.	RW	0
0	AUTOIDLE	Internal OCP clock gating strategy 0x0: Clock is running. 0x1: Automatic OCP clock gating strategy is applied, based on OCP interface activity	RW	0

**Table 23-241. Register Call Summary for Register UART\_SYSS**

UART/IrDA/CIR

- [Clock Configuration: \[0\]](#)
- [Software Reset: \[1\]](#)
- [System Power Saving: \[2\]](#)
- [Local Power Management: \[3\] \[4\] \[5\]](#)
- [Registers Available for the Register Access Modes: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [Registers Available for the UART Function: \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[24\] \[25\] \[26\] \[27\] \[28\] \[29\]](#)
- [Software Reset: \[30\]](#)
- [UART/IrDA/CIR Register Summary: \[31\] \[32\]](#)

**Table 23-242. UART\_SYSS**

<b>Address Offset</b>	0x0000 0058		<b>Instance</b>	UART																											
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .																														
<b>Description</b>	System status register																														
<b>Type</b>	R																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED											RESETDONE				



Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:1	RESERVED	Read returns 0.	R	0x00
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is ongoing. Read 0x1: Reset complete	R	0

**Table 23-243. Register Call Summary for Register UART\_SYSS**

UART/IrDA/CIR

- Registers Available for the Register Access Modes: [0] [1] [2]
- Registers Available for the UART Function: [3] [4] [5]
- Registers Available for the IrDA Function (UART3 Only): [6] [7] [8]
- Registers Available for the CIR Function (UART3 Only): [9] [10] [11]
- Software Reset: [12]
- UART/IrDA/CIR Register Summary: [13] [14]

**Table 23-244. UART\_WER**

<b>Address Offset</b>	0x0000 005C		
<b>Physical Address</b>	See Table 23-166 to Table 23-167.	<b>Instance</b>	UART
<b>Description</b>	Wake-Up enable register  The UART wake-up enable register is used to mask and unmask a UART event that would subsequently notify the system. an event is any activity in the logic that could cause an interrupt and/or an activity that would require the system to wake up. Even if the wakeup is disabled for certain events, if these events are also an interrupt to the UART, the UART registers the interrupt.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
RESERVED																<table border="1"> <tr> <td>TX_WAKEUP_EN</td> <td>EVENT_6_RECEIVER_LINE_STATUS_INTERRUPT</td> <td>EVENT_5_RHR_INTERRUPT</td> <td>EVENT_4_RX_ACTIVITY</td> <td>EVENT_3_DCD_CD_ACTIVITY</td> <td>EVENT_2_RI_ACTIVITY</td> <td>EVENT_1_DSR_ACTIVITY</td> <td>EVENT_0_CTS_ACTIVITY</td> </tr> </table>																	TX_WAKEUP_EN	EVENT_6_RECEIVER_LINE_STATUS_INTERRUPT	EVENT_5_RHR_INTERRUPT	EVENT_4_RX_ACTIVITY	EVENT_3_DCD_CD_ACTIVITY	EVENT_2_RI_ACTIVITY	EVENT_1_DSR_ACTIVITY	EVENT_0_CTS_ACTIVITY
TX_WAKEUP_EN	EVENT_6_RECEIVER_LINE_STATUS_INTERRUPT	EVENT_5_RHR_INTERRUPT	EVENT_4_RX_ACTIVITY	EVENT_3_DCD_CD_ACTIVITY	EVENT_2_RI_ACTIVITY	EVENT_1_DSR_ACTIVITY	EVENT_0_CTS_ACTIVITY																																	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	TX_WAKEUP_EN	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system: it can be THR_IT or TX_DMA request and/or TX_SATUS_IT.	RW	0
6	EVENT_6_RECEIVER_LINE_STATUS_INTERRUPT	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1

Bits	Field Name	Description	Type	Reset
5	EVENT_5_RHR_INTERRUPT	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
4	EVENT_4_RX_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
3	EVENT_3_DCD_CD_ACTIVITY	0x0: Event is not allowed to wake up the system 0x1: Event can wake up the system	RW	1
2	EVENT_2_RI_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
1	EVENT_1_DSR_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
0	EVENT_0_CTS_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1

**Table 23-245. Register Call Summary for Register UART\_WER**

UART/IrDA/CIR

- [System Power Saving: \[0\]](#)
- [Registers Available for the Register Access Modes: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Registers Available for the UART Function: \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]](#)
- [UART/IrDA/CIR Register Summary: \[25\] \[26\]](#)

**Table 23-246. UART\_CFPS**

<b>Address Offset</b>	0x0000 0060																
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b> UART															
<b>Description</b>	Carrier frequency prescaler  Because the consumer IR works at modulation rates of 30 - 56.8 kHz, the 48-MHz clock must be prescaled before the clock can drive the IR logic. This register sets the divisor rate to give a range to accommodate the remote-control requirements in baud multiples of 12x. The value of the CFPS at reset is 0105 decimal, which equals 38.1 KHz output from starting conditions. The 48-MHz carrier is prescaled by the CFPS, which is then divided by the 12x baud multiple.																
<b>Type</b>	RW																
	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																
	RESERVED											CFPS					

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	CFPS	System clock frequency prescaler at (12x multiple). Examples for CFPS values:	RW	0x69
		<b>Target Freq (kHz)</b>	<b>CFPS (decimal)</b>	<b>Actual Freq (kHz)</b>
		30	133	30.08
		32.75	122	32.79
		36	111	36.04
		36.7	109	36.69
		38*	105	38.1
		40	100	40
		56.8	70	57.14
		*configured at reset to this value		
		<b>Note:</b> CFPS = 0 is not supported.		

**Table 23-247. Register Call Summary for Register UART\_CFPS**

UART/IrDA/CIR

- Registers Available for the Register Access Modes: [0] [1] [2] [3] [4] [5]
- Registers Available for the CIR Function (UART3 Only): [6] [7] [8] [9] [10] [11]
- CIR Mode Clock Generation: [12] [13] [14]
- UART/IrDA/CIR Register Summary: [15] [16]

**Table 23-248. UART\_RXFIFO\_LVL**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	UART
<b>Physical Address</b>	See Table 23-166 to Table 23-167.		
<b>Description</b>	Level of the RX FIFO		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RXFIFO_LVL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	RXFIFO_LVL	Shows the number of received bytes in the RX FIFO	R	0x00

**Table 23-249. Register Call Summary for Register UART\_RXFIFO\_LVL**

UART/IrDA/CIR

- Registers Available for the Register Access Modes: [0] [1] [2] [3] [4] [5]
- Registers Available for the UART Function: [6] [7] [8] [9] [10] [11]
- Registers Available for the IrDA Function (UART3 Only): [12] [13] [14] [15] [16] [17]
- Registers Available for the CIR Function (UART3 Only): [18] [19] [20] [21] [22] [23]
- UART/IrDA/CIR Register Summary: [24] [25]

**Table 23-250. UART\_TXFIFO\_LVL**

<b>Address Offset</b>	0x0000 0068	<b>Instance</b>	UART
<b>Physical Address</b>	See Table 23-166 to Table 23-167.		
<b>Description</b>	Level of the TX FIFO		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TXFIFO_LVL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	TXFIFO_LVL	Shows the number of written bytes in the TX FIFO	R	0x00

**Table 23-251. Register Call Summary for Register UART\_TXFIFO\_LVL**

UART/IrDA/CIR

- Registers Available for the Register Access Modes: [0] [1] [2] [3] [4] [5]
- Registers Available for the UART Function: [6] [7] [8] [9] [10] [11]
- Registers Available for the IrDA Function (UART3 Only): [12] [13] [14] [15] [16] [17]
- Registers Available for the CIR Function (UART3 Only): [18] [19] [20] [21] [22] [23]
- UART/IrDA/CIR Register Summary: [24] [25]

**Table 23-252. UART\_IER2**

<b>Address Offset</b>	0x0000 006C	
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b> UART
<b>Description</b>	Enables RX/TX FIFOs empty corresponding interrupts	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EN_TXFIFO_EMPTY		EN_RXFIFO_EMPTY													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
1	EN_TXFIFO_EMPTY	Enables TX FIFO empty corresponding interrupt 0x0: Enables EN_TXFIFO_EMPTY interrupt 0x1: Disables EN_TXFIFO_EMPTY interrupt	RW	0
0	EN_RXFIFO_EMPTY	Enables RX FIFO empty corresponding interrupt 0x0: Enables EN_RXFIFO_EMPTY interrupt 0x1: Disables EN_RXFIFO_EMPTY interrupt	RW	0

**Table 23-253. Register Call Summary for Register UART\_IER2**

UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Registers Available for the UART Function: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [UART/IrDA/CIR Register Summary: \[24\] \[25\]](#)

**Table 23-254. UART\_ISR2**

<b>Address Offset</b>	0x0000 0070	
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b> UART
<b>Description</b>	Status of RX/TX FIFOs empty corresponding interrupts	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TXFIFO_EMPTY_STS		RXFIFO_EMPTY_STS													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
1	TXFIFO_EMPTY_STS	Used to generate interrupt if the TX_FIFO is empty (software flow control) 0x0: TXFIFO_EMPTY interrupt not pending. 0x1: TXFIFO_EMPTY interrupt pending.	RW	1
0	RXFIFO_EMPTY_STS	Used to generate interrupt if the RX_FIFO is empty (software flow control) 0x0: RXFIFO_EMPTY interrupt not pending. 0x1: RXFIFO_EMPTY interrupt pending.	RW	1

**Table 23-255. Register Call Summary for Register UART\_ISR2**

UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Registers Available for the UART Function: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [UART/IrDA/CIR Register Summary: \[24\] \[25\]](#)

**Table 23-256. UART\_FREQ\_SEL**

<b>Address Offset</b>	0x0000 0074	
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b> UART
<b>Description</b>	Sample per bit selector.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FREQ_SEL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
7:0	FREQ_SEL	Sets the sample per bit if non default frequency is used. MDR3[1] must be set to 1 after this value is set. Must be equal or higher then 6.	RW	0x1A

**Table 23-257. Register Call Summary for Register UART\_FREQ\_SEL**

UART/IrDA/CIR

- [Registers Available for the Register Access Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [Registers Available for the UART Function: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [UART/IrDA/CIR Register Summary: \[24\] \[25\]](#)

**Table 23-258. UART\_MDR3**

<b>Address Offset</b>	0x0000 0080		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Mode definition register 3.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SET_DMA_TX_THRESHOLD	NONDEFAULT_FREQ	DISABLE_CIR_RX_DEMOD	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
2	SET_DMA_TX_THRESHOLD	Enable to set different TXDMA threshold in <a href="#">UART_TX_DMA_THRESHOLD</a> register.	RW	0
1	NONDEFAULT_FREQ	Used to enable the NONDEFAULT fclk frequencies. 0x0: Disables using NONDEFAULT fclk frequencies. 0x1: Enables using NONDEFAULT fclk frequencies (set <a href="#">FREQ_SEL</a> and <a href="#">DLH/DLL</a> ).	RW	0
0	DISABLE_CIR_RX_DEMOD	Used to enable CIR RX demodulation. 0x0: Enables CIR RX demodulation. 0x1: Disables CIR RX demodulation.	RW	0

**Table 23-259. Register Call Summary for Register UART\_MDR3**

## UART/IrDA/CIR

- [FIFO DMA Mode Operation: \[0\]](#)
- [Registers Available for the Register Access Modes: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Registers Available for the UART Function: \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [Registers Available for the IrDA Function \(UART3 Only\): \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [Registers Available for the CIR Function \(UART3 Only\): \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]](#)
- [UART/IrDA/CIR Register Summary: \[25\] \[26\]](#)
- [UART/IrDA/CIR Registers: \[27\]](#)

**Table 23-260. UART\_TX\_DMA\_THRESHOLD**

<b>Address Offset</b>	0x0000 0084		
<b>Physical Address</b>	See <a href="#">Table 23-166</a> to <a href="#">Table 23-167</a> .	<b>Instance</b>	UART
<b>Description</b>	Use to manually set the TX DMA threshold level. <a href="#">UART_MDR3[2]</a> SET_TX_DMA_THRESHOLD must be one and must be value + tx_trigger_level <= 64 (TX FIFO size). If not, 64-tx_trigger_level will be used w/o modifying the value of this register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TX_DMA_THRESHOLD															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	RW	0x0000000
4:0	TX_DMA_THRESHOLD	Use to manually set the TX DMA threshold level.	RW	0x00

**Table 23-261. Register Call Summary for Register UART\_TX\_DMA\_THRESHOLD**

UART/IrDA/CIR

- FIFO DMA Mode Operation: [0]
- Registers Available for the Register Access Modes: [1] [2] [3] [4] [5] [6]
- Registers Available for the UART Function: [7] [8] [9] [10] [11] [12]
- Registers Available for the IrDA Function (UART3 Only): [13] [14] [15] [16] [17] [18]
- Registers Available for the CIR Function (UART3 Only): [19] [20] [21] [22] [23] [24]
- UART/IrDA/CIR Register Summary: [25] [26]
- UART/IrDA/CIR Registers: [27]

## 23.4 Multichannel Serial Port Interface (MCSPi)

This section describes the four multichannel serial port interface (MCSPi) modules for the device.

### 23.4.1 MCSPi Overview

The multichannel serial port interface (MCSPi) is a master/slave synchronous serial bus. The MCSPi has four separate modules (SPI1, SPI2, SPI3, and SPI4) in the device (see [Figure 23-85](#)). The MCSPi modules differ as follows: SPI1 supports up to four peripherals, SPI2 and SPI3 support up to two peripherals, and SPI4 supports only one peripheral.

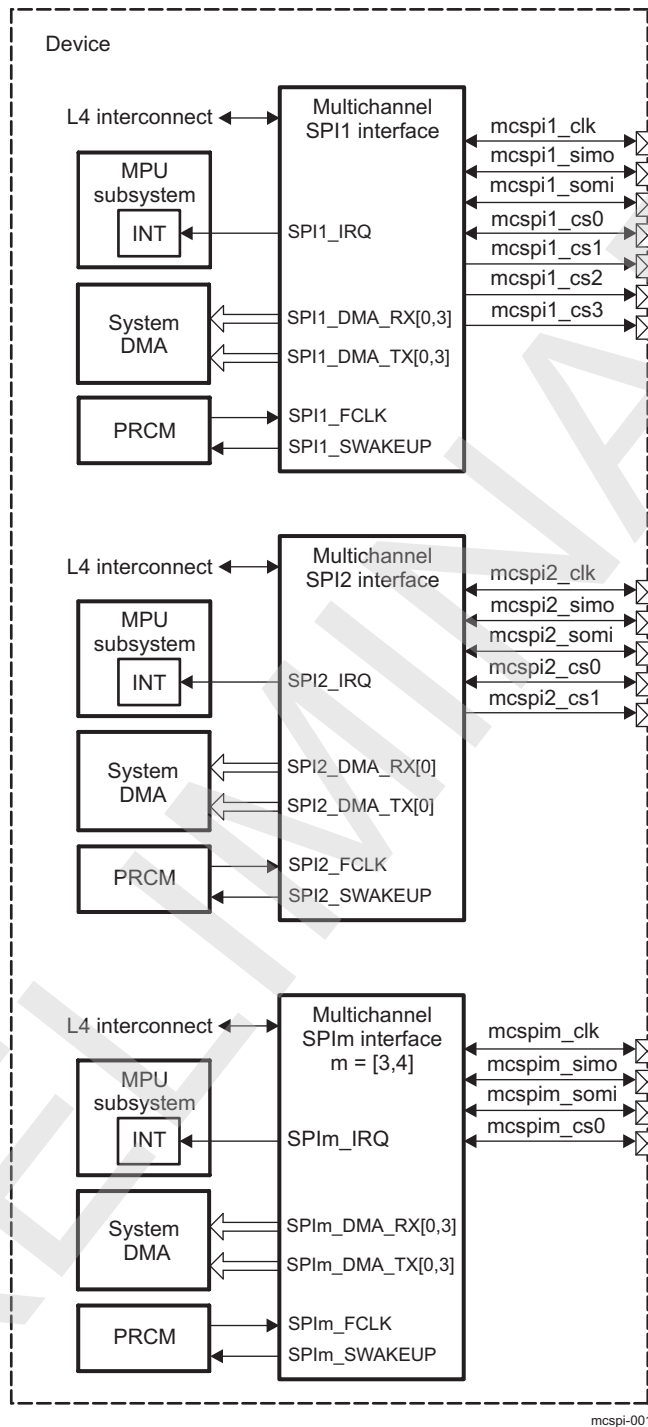
---

**NOTE:** In this chapter,  $m = [1,4]$  represents the module instance and  $x$  represents the channel in signal and register naming. There MCSPi has four instances, with each module instance having different channel numbers:

- SPI1: 4 channels (if  $m = 1, x = 4$ )
  - SPI2: 2 channels (if  $m = 2, x = 2$ )
  - SPI3: 1 channel (if  $m = 3, x = 1$ )
  - SPI4: 1 channel (if  $m = 4, x = 1$ )
-



Figure 23-85. Multichannel Modules SPI1, SPI2, SPI3, and SPI4



The MCSIPI instances include the following main features:

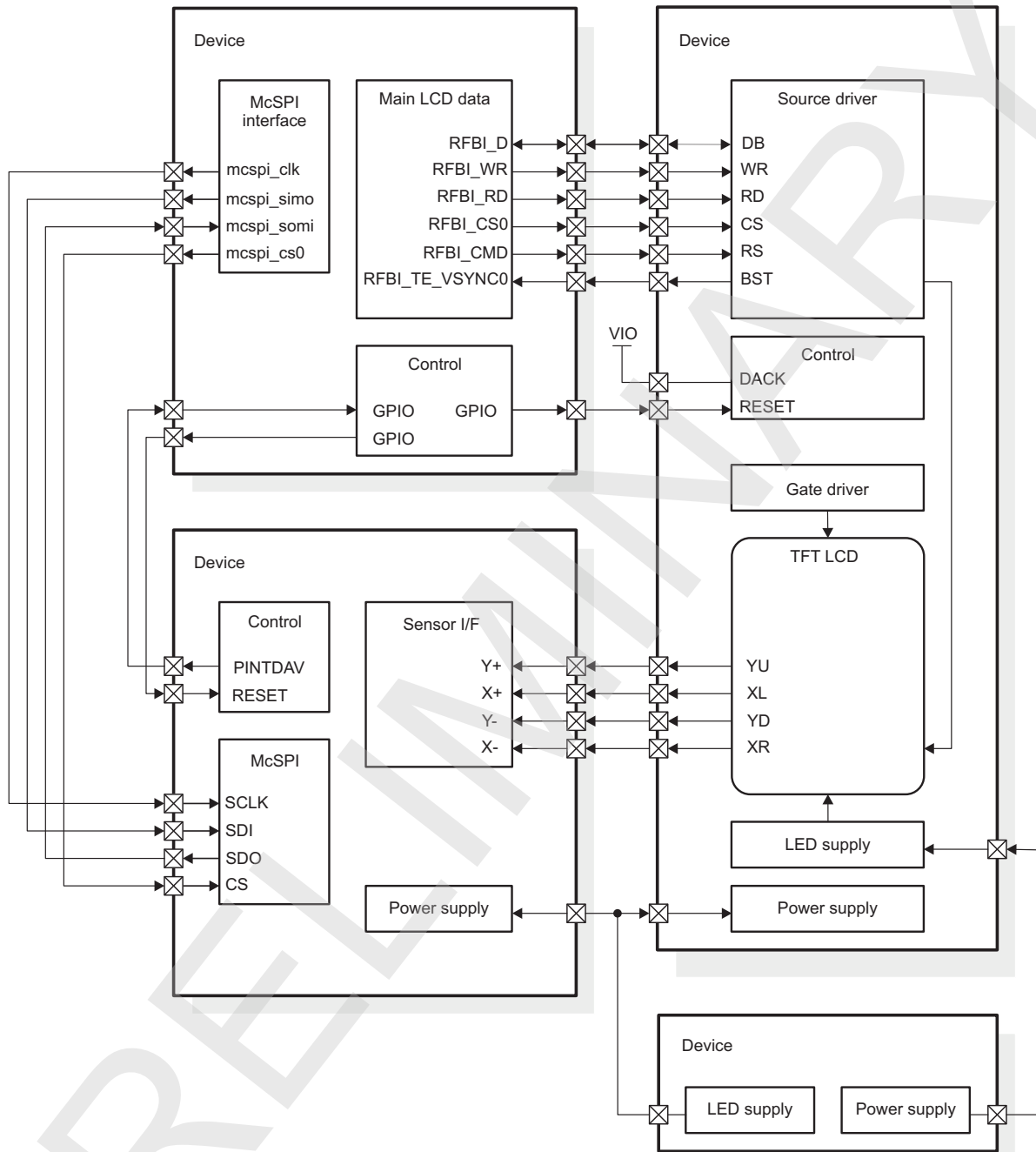
- Serial clock with programmable frequency, polarity, and phase for each channel
- Wide selection of SPI word lengths, ranging from 4 to 32 bits
- Up to four master channels, or single channel in slave mode
- Master multichannel mode:
  - Full duplex/half duplex
  - Transmit-only/receive-only/transmit-and-receive modes

- Flexible I/O port controls per channel
- Two direct memory access (DMA) requests (read/write) per channel
- Single interrupt line for multiple interrupt source events
- Power management through wake-up capabilities
- Enable the addition of a programmable start-bit for SPI transfer per channel (start-bit mode)
- Supports start-bit write command
- Supports start-bit pause and break sequence
- 64-byte built-in FIFO available for a single channel
- Force CS mode for continuous transfers

### 23.4.2 MCSPi Environment

Figure 23-86 shows a simplified overview of a typical application system using the MCSPi. This example is based on a TFS chip set, including a 2.2-inch color-active matrix thin-film transistor (TFT) liquid crystal display (LCD) with a light-emitting diode (LED) front light, a four-wire resistive touch-screen panel, and LCD controllers. This chip set is associated with a touch-screen controller, powered by a power-management unit, and driven by the device. The MCSPi device interface is set to master mode; the touch-screen MCSPi controller interface operates in slave mode.

Figure 23-86. Typical Application Using the MCSPi



mcspi-034

### 23.4.2.1 MCSPi Interface

#### 23.4.2.1.1 Basic MCSPi Pins for Master Mode

Figure 23-87 shows all of the MCSPi interface signals in master mode.

**Figure 23-87. MCSPi Interface Signals in Master Mode**

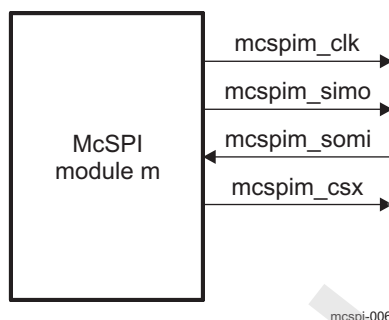


Table 23-262 describes the MCSPi I/O in master mode.

**Table 23-262. MCSPi I/O Description (Master Mode)**

Signal Name	I/O	Description	Reset <sup>(1)</sup>
mcspim_clk	O	SPIm module serial clock <sup>(2)</sup>	Unknown
mcspim_simo	O	SPIm module serial data master out (slave input, master output)	Unknown
mcspim_somi	I	SPIm module serial data master input (slave output, master input)	-
mcspim_csx	O	SPIm module chip-select x output	Low

<sup>(1)</sup> After reset, the SPI modules are in slave mode by default. This paragraph implies that the MCSPi module is configured in slave mode. (See the MCSPi\_MODULCTRL[2] MS bit in the module control register [MCSPi\_MODULCTRL]).

<sup>(2)</sup> This output signal is also used as re-timing input.

#### 23.4.2.1.2 Basic MCSPi Pins for Slave Mode

Figure 23-88 shows all of the MCSPi interface signals in slave mode.

**Figure 23-88. MCSPi Interface Signals in Slave Mode**

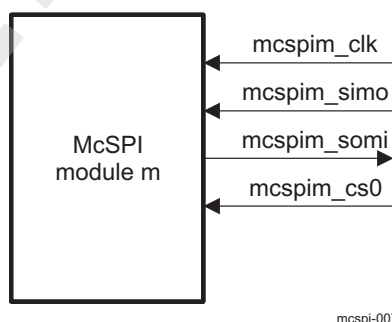


Table 23-263 describes the MCSPi I/O in slave mode.

**Table 23-263. MCSPi I/O Description (Slave Mode)**

Signal Name	I/O	Description	Reset <sup>(1)</sup>
mcspim_clk	I	SPIm module serial clock	Unknown

<sup>(1)</sup> After reset, the SPI modules are in slave mode by default. This paragraph implies that the MCSPi module is configured in slave mode. (See the MCSPi\_MODULCTRL[2] MS bit in the module control register (MCSPi\_MODULCTRL).)

**Table 23-263. MCSPI I/O Description (Slave Mode) (continued)**

Signal Name	I/O	Description	Reset <sup>(1)</sup>
mcspim_simo	I	SPIm module serial data master out (slave input, master output)	Unknown
mcspim_somi	O	SPIm module serial data master input (slave output, master input)	-
mcspim_csx	I	SPIm module chip-select x output	Low

**23.4.2.1.3 Multichannel SPI Protocol and Data Format**

The synchronous SPI protocol allows a master device to initiate serial data transfers to a slave device. A slave select line (mcspim\_csx) allows selection of an individual slave SPI device. Slave devices that are not selected do not interfere with SPI bus activities.

MCSPi offers the flexibility to modify the following parameters to adapt to the device features:

- Word length  
 MCSPi supports any SPI word ranging from 4 bits to 32 bits long (SPIm.MCSPi\_CHxCONF[11:7] WL field).  
 SPI word length can be changed between transmissions to allow the master device to communicate with peripheral slaves that have different requirements.
- SPI enable (mcspim\_csx, for channel x of instance m)  
 The polarity of the SPI enable signals is programmable (SPIm.MCSPi\_CHxCONF[6] EPOL bit). mcspim\_csx signals can be active high or low.  
 The assertion of the mcspim\_csx signals is programmable and can be manually or automatically asserted. The manual assertion mode is available in single master mode only. mcspim\_csx can be kept active between words with the SPIm.MCSPi\_CHxCONF[20] FORCE bit.  
 Two consecutive words for two different slave devices can go along with active mcspim\_csx signals with different polarity. .
- Programmable start-bit  
 In start-bit mode a start-bit is added before the SPI word length to indicate how the next SPI word must be handled. The start-bit is enabled by setting SPIm.MCSPi\_CHxCONF[23] SBE bit to 1. The SPIm.MCSPi\_CHxCONF[24] SBPOL bit defines the polarity of the start-bit.
- Programmable SPI clock
  - Bit rate  
 In master mode, the baud rate of the SPI serial clock is programmable using the 48-MHz reference clock (from the power, reset, and clock management [PRCM] module). Table 23-264 gives the spim\_clk bit rates obtained for data transfer when programming the clock divider (SPIm.MCSPi\_CHxCONF[5:2] CLKD bit field).

**Table 23-264. SPI Master Clock Rates**

Divider	Clock Rate
1	48 MHz
2	24 MHz
4	12 MHz
8	6 MHz
16	3 MHz
32	1.5 MHz
64	750 kHz
128	375 kHz
256	~187 kHz
512	~93.7 kHz
1024	~46.8 kHz
2048	~23.4 kHz

**Table 23-264. SPI Master Clock Rates (continued)**

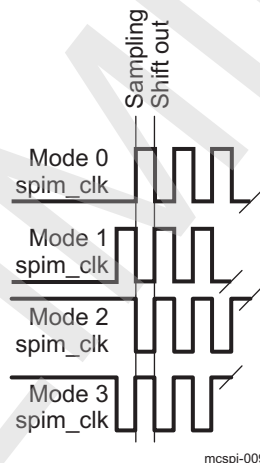
Divider	Clock Rate
4096	~11.7 kHz

– Polarity and phase

The polarity (the SPIm.MCSPi\_CHxCONF[1] POL bit) and the phase (the SPIm.MCSPi\_CHxCONF[0] PHA bit) of the SPI serial clock (mcspim\_clk) are configurable to offer four combinations. Software selects the right combination, depending on the device. See [Table 23-265](#) and [Figure 23-89](#).

**Table 23-265. Phase and Polarity Combinations**

Polarity (POL)	Phase (PHA)	SPI Mode	Comments
0	0	Mode 0	mcspim_clk is active high and sampling occurs on the rising edge.
0	1	Mode 1	mcspim_clk is active high and sampling occurs on the falling edge.
1	0	Mode 2	mcspim_clk is active low and sampling occurs on the falling edge.
1	1	Mode 3	mcspim_clk is active low and sampling occurs on the rising edge.

**Figure 23-89. Phase and Polarity Combinations****23.4.2.1.3.1 Transfer Format**

In master and slave modes, MCSPi drives the data lines when spim\_csx is asserted.

Each word is transmitted starting with the most-significant bit (MSB).

This section explains the two cases of data transmission determined by the clock phase (PHA) and the type of data transmission using a start-bit (SBE) called the start-bit mode:

- Transmission in mode 0 and mode 2 (PHA = 0)

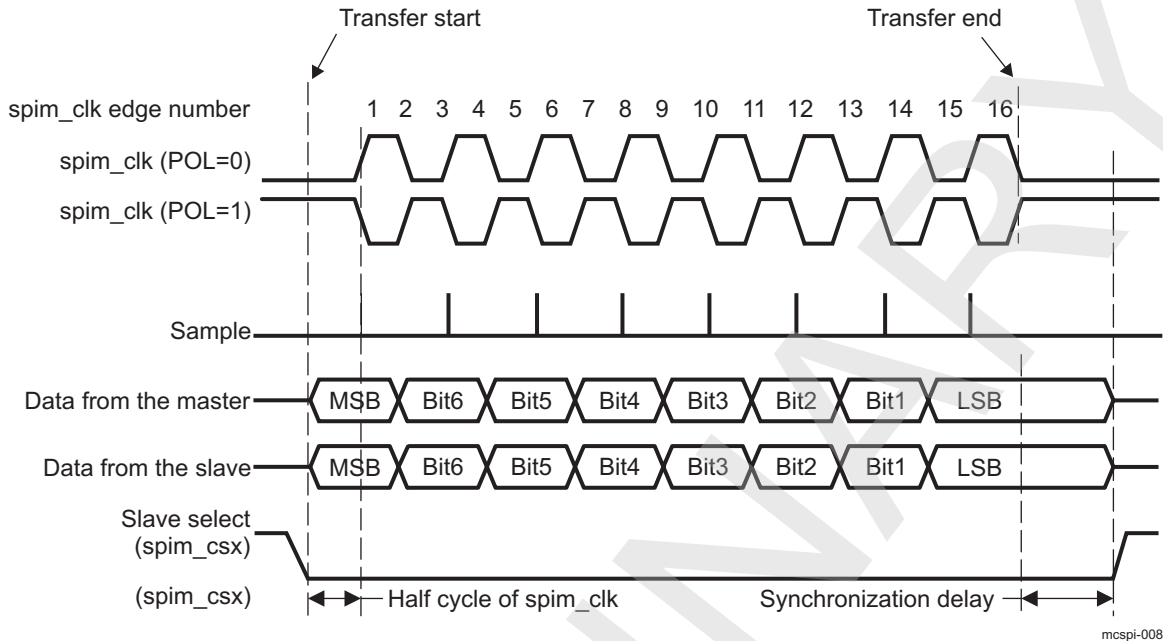
When PHA = 0, the first bit of the SPI word to transmit (on the master or the slave data output pin) is valid one half cycle of spim\_clk after the spim\_csx assertion.

Therefore, the first edge of the mcspim\_clk line is used by the master to sample the first data bit sent by the slave. On the same edge, the first data bit sent by the master is sampled by the slave.

On the next mcspim\_clk edge, the received data bit is shifted into the receive shift register and a new data bit is transmitted on the serial data line.

This process continues for a number of pulses on the spim\_clk line defined by the SPI word length programmed in the master device, with data being latched on odd-numbered edges and shifted on even-numbered edges. See [Figure 23-90](#).

**Figure 23-90. Full-Duplex Transfer Format With PHA = 0**



- Transmission in mode 1 and mode 3 (PHA = 1)

When PHA = 1, the first bit of the SPI word to transmit (on the master or the slave data output pin) is valid on the following mcsnim\_clk edge (one half cycle later). This is the sampling edge for the master and slave. A synchronization delay is added between the mcsnim\_csx activation and the first mcsnim\_clk edge.

The received data bit is shifted into the shift register on the third mcsnim\_clk edge.

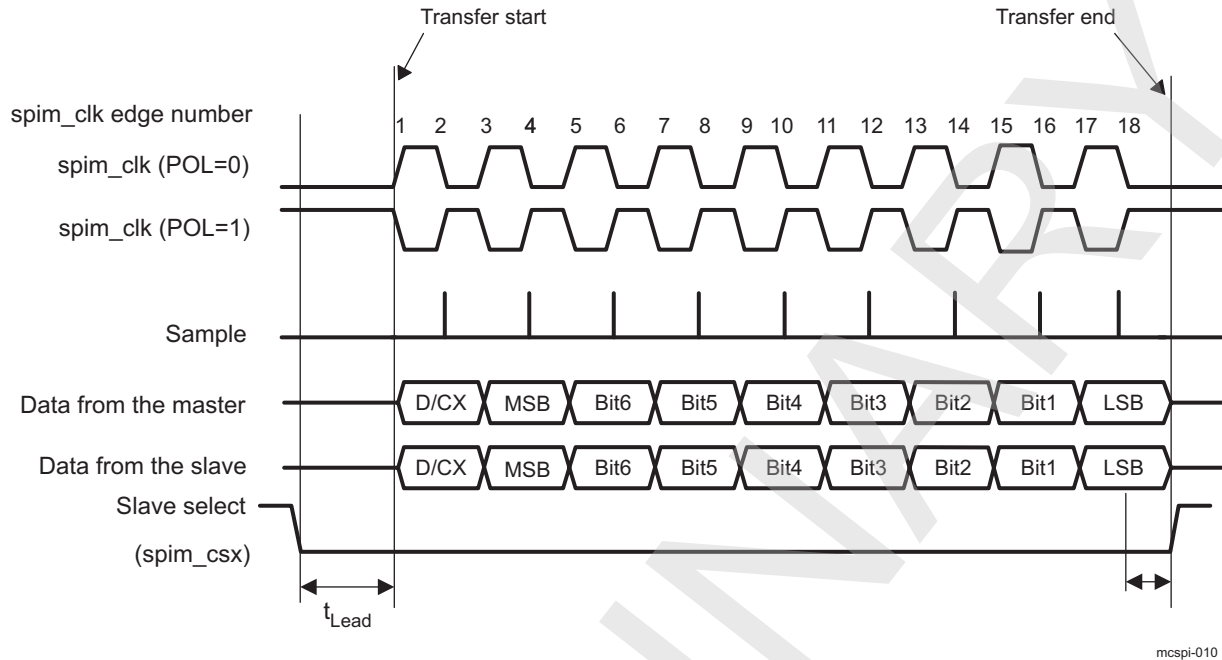
This process continues for a number of pulses on the mcsnim\_clk line defined by the SPI word length programmed in the master device, with data being latched on even-numbered edges and shifted on odd-numbered edges.

**NOTE:** The minimum synchronization delay is one cycle of mcsnim\_clk, if the mcsnim\_clk frequency equals the SPI<sub>m</sub>\_FCLK (MCSPIm functional clock) frequency in master mode. The minimum synchronization delay is one-half cycle of mcsnim\_clk, if the mcsnim\_clk frequency is lower than the SPI<sub>m</sub>\_FCLK frequency in the master and slave modes.

- Transmission with a start-bit (SBE = 1)

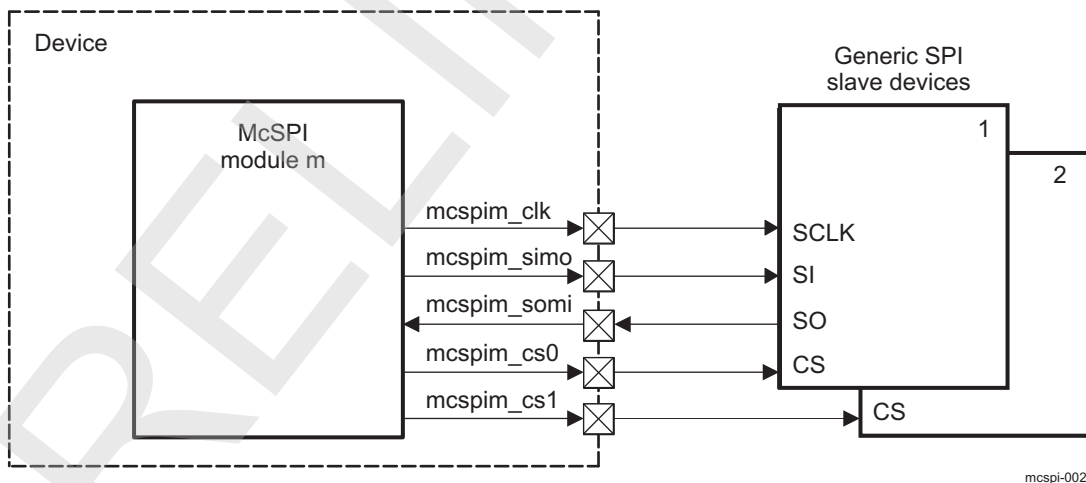
When the SPI<sub>m</sub>.MCSPI\_CHxCONF[23] SBE bit = 1, a start-bit is added before the MSB to indicate whether the next SPI word must be handled as a command or as data.

Figure 23-91 shows an example of a data transfer with an extra start-bit.

**Figure 23-91. Extended SPI Transfer With a Start-Bit (SBE = 1)**

### 23.4.2.2 SPI in Master Mode

Figure 23-92 shows a case in master mode (full-duplex) where the MCSPi module is connected with two slave devices.

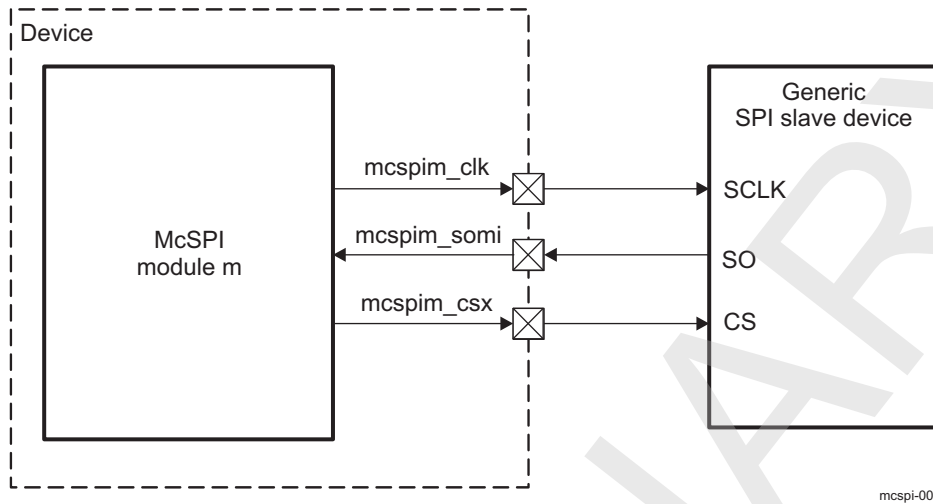
**Figure 23-92. MCSPi Master Mode (Full-Duplex)**

**NOTE:** In this case  $m = [1,3]$ .

Figure 23-93 shows the master single mode, which can also be configured in receive-only mode.



Figure 23-93. MCSPi Master Single Mode (Receive-Only)



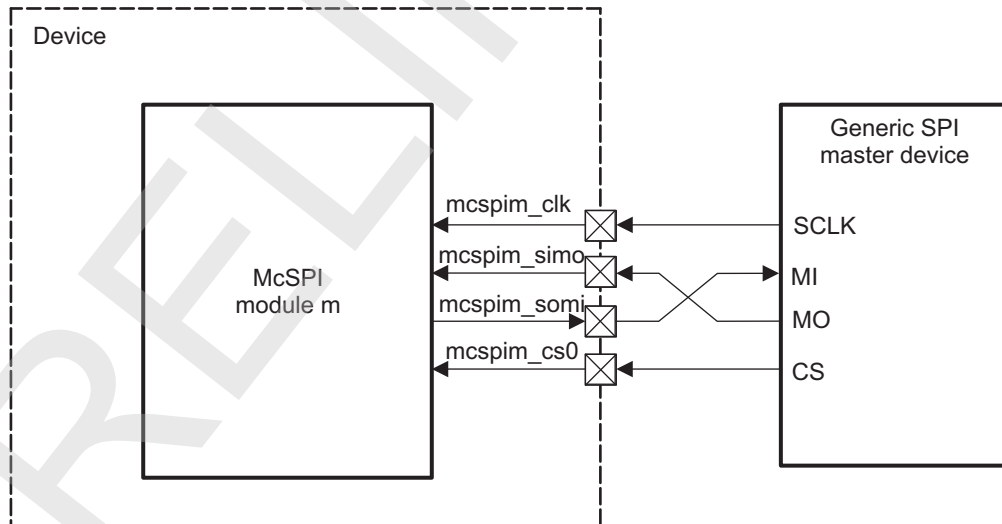
mcspi-003

### 23.4.2.3 SPI in Slave Mode

Figure 23-94 shows a case in slave mode (full-duplex).

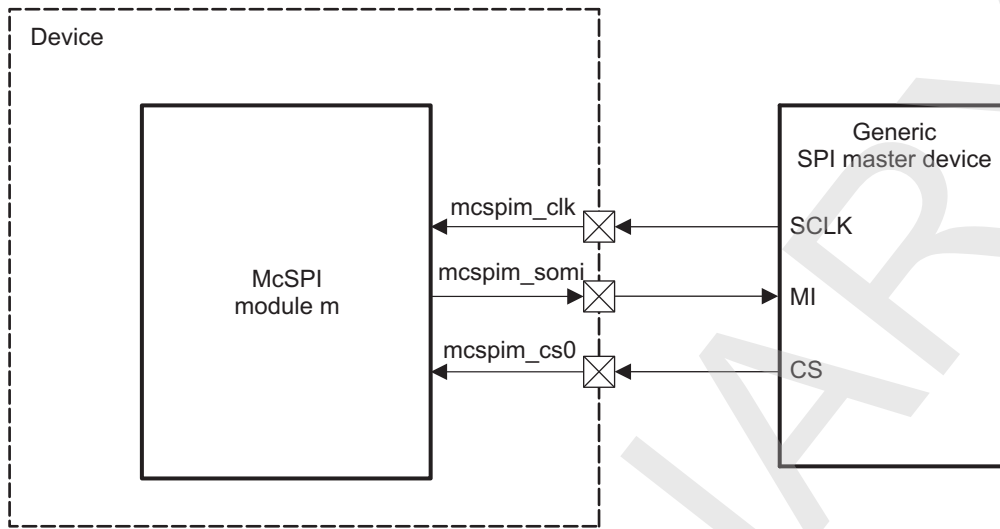
**NOTE:** Only channel 0 can be configured as slave, but the chip-enable signal can be connected to any mcspim\_csx pin and then rerouted internally to channel 0 (SPIm.MCSPi\_CHxCONF[22:21] SPIENSLV bit field [where x = 0]). For more information, see Section 23.4.4.3, *Slave Mode*.

Figure 23-94. MCSPi Slave Mode (Full Duplex)



mcspi-004

Figure 23-95 shows the slave single mode, which can also be configured in transmit-only mode.

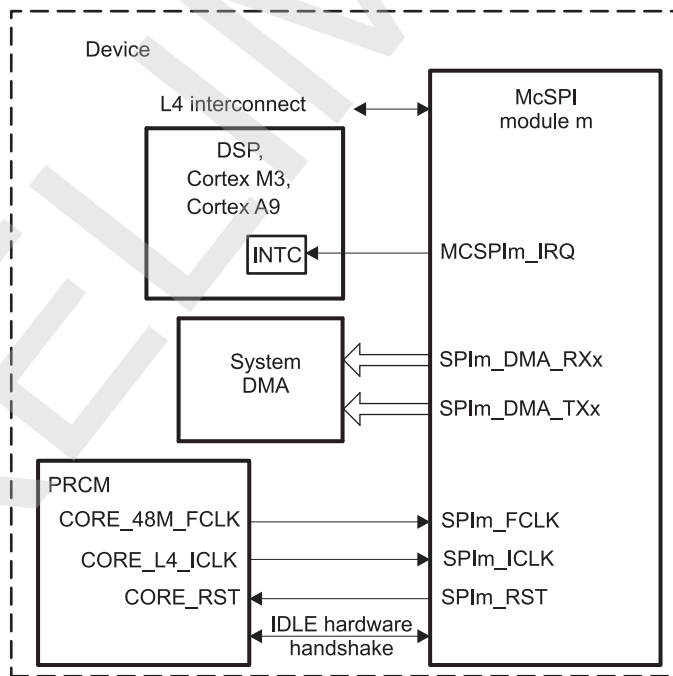
**Figure 23-95. MCSPi Slave Single Mode (Transmit Only)**

mcspi-005

### 23.4.3 MCSPi Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 23-96 shows **MCSPi** integration.

**Figure 23-96. MCSPi Integration**

mcspi-011

**NOTE:** For more information about the IDLE hardware handshake and the wake-up request, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**NOTE:** For the names of the signals going from the MCSPi and to the DSP, Cortex™-M3, and Cortex™-A9 INTC, see [Table 23-268](#).

[Table 23-266](#) through [Table 23-268](#) summarize the integration of the module in the device.

**Table 23-266. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
MCSPi1	PD_L4_PER	L4_PER
MCSPi2	PD_L4_PER	L4_PER
MCSPi3	PD_L4_PER	L4_PER
MCSPi4	PD_L4_PER	L4_PER

**Table 23-267. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
MCSPi1	SPI1_ICLK	PER_L4_ICLK	PRCM	Interface clock
	SPI1_FCLK	PER_48M_FCLK	PRCM	Functional clock
MCSPi2	SPI2_ICLK	PER_L4_ICLK	PRCM	Interface clock
	SPI2_FCLK	PER_48M_FCLK	PRCM	Functional clock
MCSPi3	SPI3_ICLK	PER_L4_ICLK	PRCM	Interface clock
	SPI3_FCLK	PER_48M_FCLK	PRCM	Functional clock
MCSPi4	SPI4_ICLK	PER_L4_ICLK	PRCM	Interface clock
	SPI4_FCLK	PER_48M_FCLK	PRCM	Functional clock
Resets				
MCSPi1	SPI1_RST	L4_PER_RST	PRCM	Reset from the PRCM
MCSPi2	SPI2_RST	L4_PER_RST	PRCM	Reset from the PRCM
MCSPi3	SPI3_RST	L4_PER_RST	PRCM	Reset from the PRCM
MCSPi4	SPI4_RST	L4_PER_RST	PRCM	Reset from the PRCM

**Table 23-268. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
MCSPi1	SPI1_IRQ	D_IRQ_87	DSP INTC	Interrupt request to the DSP interrupt controller (INTC)
	SPI1_IRQ	MA_IRQ_65	Cortex-A9 INTC	Interrupt request to the Cortex-A9 INTC
	SPI1_IRQ	MM_IRQ_41	Cortex-M3 INTC	Interrupt request to the Cortex-M3 INTC
MCSPi2	SPI2_IRQ	MA_IRQ_66	Cortex-A9 INTC	Interrupt request to the Cortex-A9 INTC
	SPI2_IRQ	MM_IRQ_42	Cortex-M3 INTC	Interrupt request to the Cortex-M3 INTC
MCSPi3	SPI3_IRQ	MA_IRQ_91	Cortex-A9 INTC	Interrupt request to the Cortex-A9 INTC
MCSPi4	SPI4_IRQ	MA_IRQ_48	Cortex-A9 INTC	Interrupt request to the Cortex-A9 INTC
DMA Requests				
MCSPi1	SPI1_DMA_TX0	S_DMA_34	sDMA	Destination is system DMA (sDMA).
	SPI1_DMA_RX0	S_DMA_35	sDMA	Destination is system DMA (sDMA).

**Table 23-268. Hardware Requests (continued)**

	SPI1_DMA_TX1	S_DMA_36	sDMA	Destination is system DMA (sDMA).
	SPI1_DMA_RX1	S_DMA_37	sDMA	Destination is system DMA (sDMA).
	SPI1_DMA_TX2	S_DMA_38	sDMA	Destination is system DMA (sDMA).
	SPI1_DMA_RX2	S_DMA_39	sDMA	Destination is system DMA (sDMA).
	SPI1_DMA_TX3	S_DMA_40	sDMA	Destination is system DMA (sDMA).
	SPI1_DMA_RX3	S_DMA_41	sDMA	Destination is system DMA (sDMA).
MCSPI2	SPI2_DMA_TX0	S_DMA_42	sDMA	Destination is system DMA (sDMA).
	SPI2_DMA_RX0	S_DMA_43	sDMA	Destination is system DMA (sDMA).
	SPI2_DMA_TX1	S_DMA_44	sDMA	Destination is system DMA (sDMA).
	SPI2_DMA_RX1	S_DMA_45	sDMA	Destination is system DMA (sDMA).
MCSPI3	SPI3_DMA_TX0	S_DMA_14	sDMA	Destination is system DMA (sDMA).
	SPI3_DMA_RX0	S_DMA_15	sDMA	Destination is system DMA (sDMA).
	SPI3_DMA_TX1	S_DMA_22	sDMA	Destination is system DMA (sDMA).
	SPI3_DMA_RX1	S_DMA_23	sDMA	Destination is system DMA (sDMA).
MCSPI4	SPI4_DMA_TX0	S_DMA_69	sDMA	Destination is system DMA (sDMA).
	SPI4_DMA_RX0	S_DMA_70	sDMA	Destination is system DMA (sDMA).

Table 23-269 lists the wake-up requests.

**Table 23-269. Wake-Up Requests**

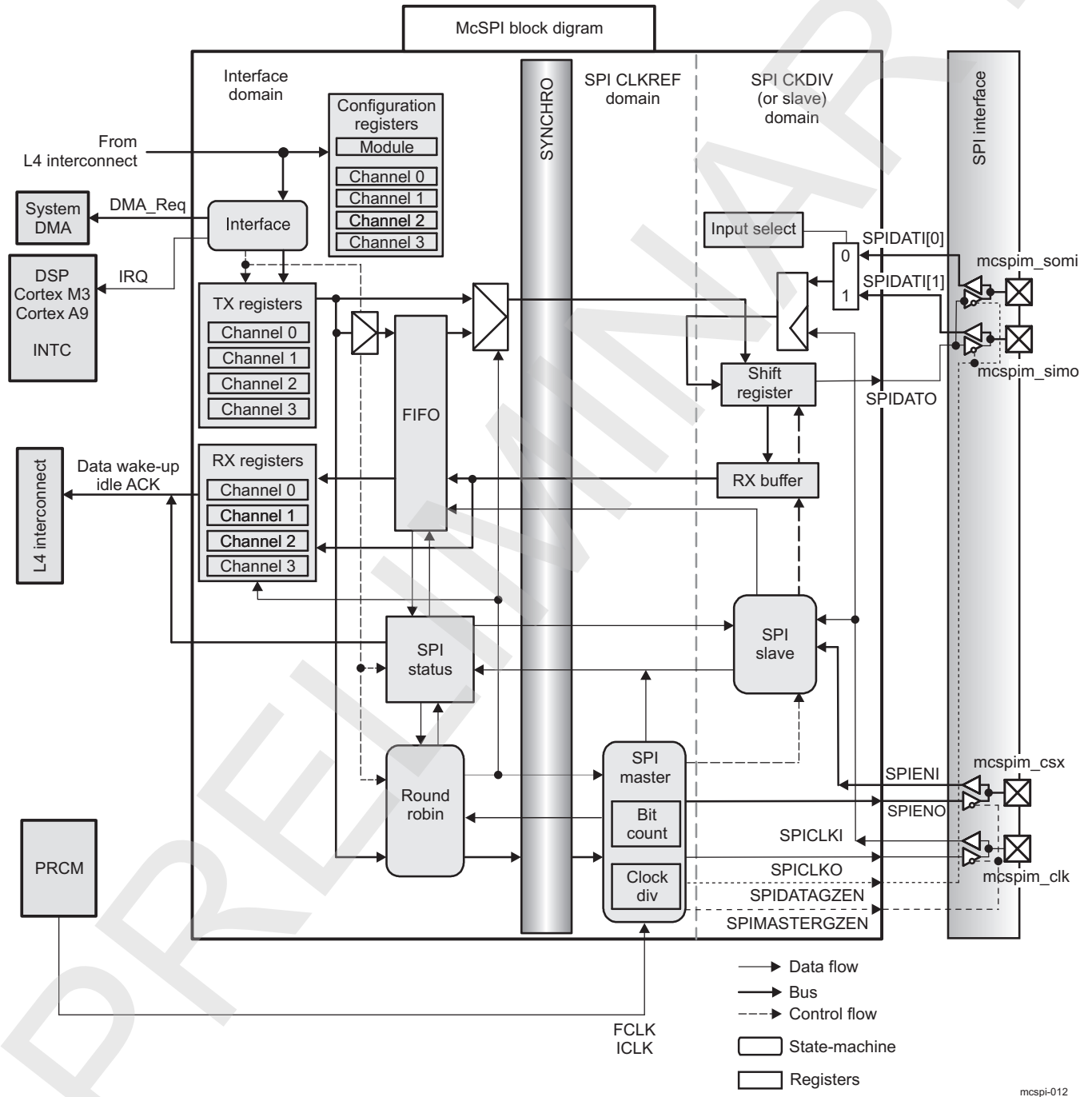
Attributes	Wake Request	Name	Mapping	Comments
SPI1	1	spi1_cs0	SPI1_SWAKEUP	Destination is the PRCM module.
SPI2	1	spi2_cs0	SPI2_SWAKEUP	Destination is the PRCM module.
SPI3	1	spi3_cs0	SPI3_SWAKEUP	Destination is the PRCM module.
SPI4	1	spi4_cs0	SPI4_SWAKEUP	Destination is the PRCM module.

### 23.4.4 MCSPi Functional Description

#### 23.4.4.1 MCSPi Block Diagram

Figure 23-97 shows the MCSPi module.

Figure 23-97. MCSPi Block Diagram



mcs pi-012

## 23.4.4.2 Master Mode

### 23.4.4.2.1 Master Mode Features

The MCSPI master mode supports multichannel communication with up to four independent SPI communication channel contexts. The MCSPI initiates a data transfer on the data lines (mcs pim\_simo and mcs pim\_somi) and generates clock (mcs pim\_clk) and control signals (mcs pim\_csx).

Connected to multiple external devices, the MCSPI exchanges data with one SPI device at a time through two main modes (available in slave mode):

- Two-data-pins interface mode (transmit-and-receive mode for full-duplex transmission)
- Single-data-pin interface mode (recommended for half-duplex transmission)

Two DMA request events (read and write) allow synchronized accesses of the DMA controller with the activity of MCSPI.

Three interrupt events can be used for data transmission and reception in master mode (for more information about interrupts, see [Section 23.4.4.5.1, Interrupt Events in Master Mode](#)).

### 23.4.4.2.2 Master Transmit-and-Receive Mode (Full Duplex)

In full-duplex transmission, data is transmitted (shifted out serially on mcs pim\_simo) and received (shifted in serially on mcs pim\_somi) simultaneously on separate data lines.

The master transmit-and-receive mode is programmable per channel (the SPIm.MCSPI\_CHxCONF[13:12] TRM field).

Channel access to the shift registers for transmission/reception is based on the MCSPI\_TXx transmitter register state, the MCSPI\_RXx receiver register state, and round robin arbitration.

Channels that meet the following rules are included in the round-robin list of active channels scheduled for transmission and/or reception. The arbiter skips channels that do not meet the rules and searches in the rotation for the next enabled channel.

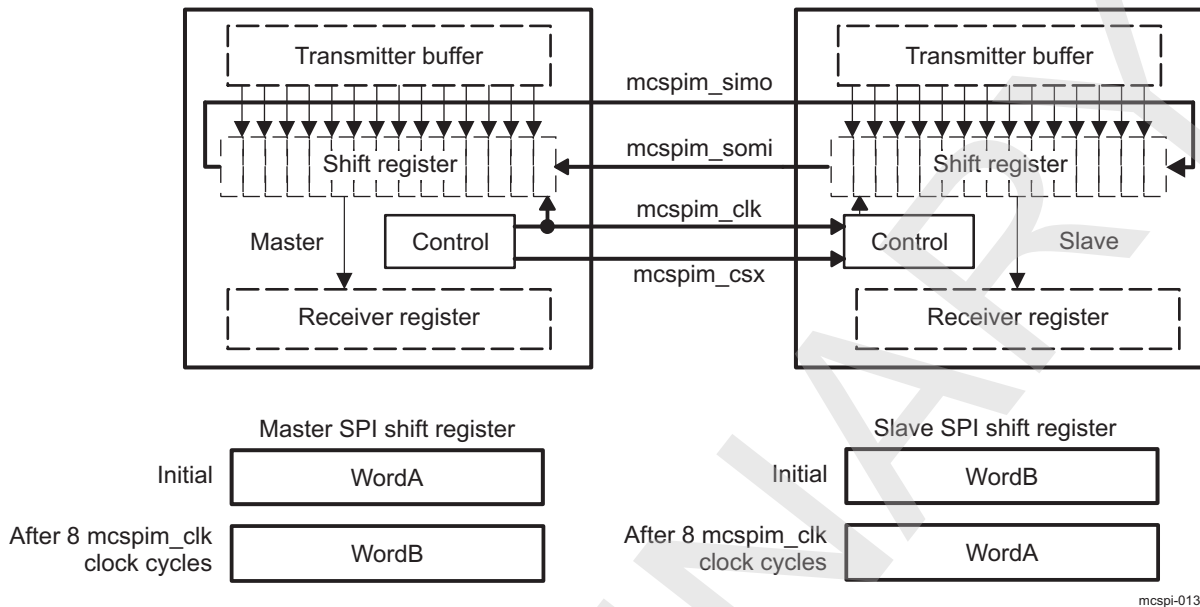
- Rule 1: Only enabled channels (the SPIm.MCSPI\_CHxCTRL[0] EN bit) can be scheduled for transmission and/or reception.
- Rule 2: If its MCSPI\_TXx transmitter register is not empty (the SPIm.MCSPI\_CHxSTAT[1] TXS bit), an enabled channel can be scheduled when the shift register is assigned. If the MCSPI\_TXx register is empty when the shift register is assigned, the TXx\_UNDERFLOW event is activated, and the next enabled channel with new data to transmit is scheduled (see also the transmit-only mode).
- Rule 3: An enabled channel can be scheduled if its receive register is not full (the SPIm.MCSPI\_CHxSTAT[0] RXS bit) when the shift register is assigned (see also the receive-only mode). Therefore, the MCSPI\_RXx register cannot be overwritten. The SPI1.MCSPI\_IRQSTATUS[3] RX0\_OVERFLOW bit is never set to this mode.

When SPI word transfer completes (the SPIm.MCSPI\_CHxSTAT[2] EOT bit is set), the updated MCSPI\_TXx register of the next scheduled channel is loaded into the shift register. The serialization (transmit-and-receive) starts depending on the channel communication configuration. When serialization completes, the received data transfers to the channel receive register.

The serial clock (mcs pim\_clk) synchronizes shifting and sampling of the information on the two serial data lines (mcs pim\_simo and mcs pim\_somi). Each time a bit transfers out from the master, 1 bit transfers in from the slave.

[Figure 23-98](#) shows an example of a full-duplex system with a master device (MCSPI module *m*) on the left and a slave device on the right. After eight cycles of the serial clock mcs pim\_clk, WordA transfers from the master to the slave. At the same time, WordB transfers from the slave to the master.

Figure 23-98. SPI Full-Duplex Transmission (Example)



mcs pi-013

### 23.4.4.2.3 Master Transmit-Only Mode (Half Duplex)

The master transmit-only mode prevents the MPU from reading the `MCSPI_RXx` register (minimizing data movement) when only transmission is meaningful.

The master transmit-only mode is programmable per channel (the `SPIm.MCSPI_CHxCONF[13:12]` TRM field). Transmission starts only after data is loaded into the `MCSPI_TXx` register.

Rule 1 and Rule 2, defined in Section 23.4.4.2.2, are applicable in this mode.

Rule 3, defined in Section 23.4.4.2.2, is not applicable.

In master transmit-only mode, the `MCSPI_RXx` register state FULL does not prevent transmission and the `MCSPI_RXx` register is always overwritten with the new SPI word. This event is not significant when only transmission is meaningful. Thus, the `RX0_OVERFLOW` bit in the `SPIm.MCSPI_IRQSTATUS` register is never set in this mode.

The hardware automatically disables the `RX_FULL` interrupt and the DMA read requests.

The transfer status is given by the `SPIm.MCSPI_CHxSTAT[2]` EOT bit.

### 23.4.4.2.4 Master Receive-Only Mode (Half Duplex)

The master receive mode prevents the MPU from refilling the `MCSPI_TXx` register (minimizing data movement) when only reception is meaningful.

The master receive mode is programmable per channel (the `SPIm.MCSPI_CHxCONF[13:12]` TRM field).

The master receive-only mode enables channel scheduling only on the empty state of the `MCSPI_RXx` register.

Rule 1 and Rule 3, defined in Section 23.4.4.2.2, are applicable in this mode.

Rule 2, defined in Section 23.4.4.2.2, is not applicable.

In the master receive-only mode, software must write dummy data to the `MCSPI_TXx` register. Only one dummy write is enough to receive any number of words from the slave. Software should ensure that the `MCSPI_TXx` register is always full (the `TXx_EMPTY` bits of `SPIm.MCSPI_IRQSTATUS`) when receiving. The content of the `MCSPI_TXx` register is always loaded into the shift register when the shift register is assigned. After writing the dummy data to the `MCSPI_TXx` register, the `TXx_EMPTY` and `TXx_UNDERFLOW` bits in the `SPIm.MCSPI_IRQSTATUS` register are never set in receive-only mode.

The SPIm.MCSPI\_CHxSTAT[2] EOT bit gives the status of serialization. The RXx\_FULL bits of the SPIm.MCSPI\_IRQSTATUS register are set when received data is loaded from the shift register to the corresponding MCSPI\_RXx register. The SPIm.MCSPI\_IRQSTATUS[3] RX0\_OVERFLOW bit is never set in this mode.

#### 23.4.4.2.5 Single-Channel Master Mode

When the MCSPI is configured as a master device with a single enabled channel, the assertion of the mcsnim\_csx signal can be controlled in two different ways:

- If the MCSPI\_MODULCTRL[0] SINGLE bit is set to 0, mcsnim\_csx assertion/deassertion after each SPI word is automatically controlled by the MCSPI module (see subsections of Section 23.4.4.2.1, *Master Mode Features*).
- If the MCSPI\_MODULCTRL[0] SINGLE bit and the MCSPI\_CHxCONF[20] FORCE bit are set to 1: mcsnim\_csx assertion/deassertion is controlled by software (see Section 23.4.4.2.5.1, *Programming Tips When Switching to Another Channel*).

##### 23.4.4.2.5.1 Programming Tips When Switching to Another Channel

When a single channel is enabled and data transfer is ongoing:

- Wait for completion of the SPI word transfer (wait until the SPIm.MCSPI\_CHxSTAT[2] EOT bit is set to 1) before disabling the current channel and enabling a different channel.
- Disable the current channel first, and then enable the other channel.

##### 23.4.4.2.5.2 Force mcsnim\_csx Mode

Continuous transfers are allowed manually by keeping the mcsnim\_csx signal active for successive SPI words transfer. Several sequences (configuration/enable/disable of the channel) can be run without deactivating the mcsnim\_csx line. This mode is supported by all channels and any master sequence can be used (transmit-receive, transmit-only, receive-only).

Keeping the mcsnim\_csx active mode is supported when:

- A single channel is used (with the SPIm.MCSPI\_MODULCTRL[0] SINGLE bit set to 1).
- Transfer parameters are loaded in the configuration register of the appropriate channel (SPIm.MCSPI\_CHxCONF).

The state of the mcsnim\_csx signal is programmable.

- Writing 1 to the SPIm.MCSPI\_CHxCONF[20] FORCE bit drives the mcsnim\_csx line high when the SPIm.MCSPI\_CHxCONF[6] EPOL bit is set to 0. mcsnim\_csx is driven low when the SPIm.MCSPI\_CHxCONF[6] EPOL bit is set to 1.
- Writing 0 to the SPIm.MCSPI\_CHxCONF[20] FORCE bit drives the mcsnim\_csx line low when the SPIm.MCSPI\_CHxCONF[6] EPOL bit is set to 0. mcsnim\_csx is driven high when the SPIm.MCSPI\_CHxCONF[6] EPOL bit is set to 1.
- A single channel is enabled (the SPIm.MCSPI\_CHxCTRL[0] EN bit is set to 1). The first enabled channel activates the mcsnim\_csx line.

When the channel is enabled, the mcsnim\_csx signal activates with the programmed polarity. As in the multichannel master mode, the transfer start depends on the status of the MCSPI\_TXx register (the SPIm.MCSPI\_CHxSTAT[1] TXS bit), the status of the MCSPI\_RXx register (the SPIm.MCSPI\_CHxSTAT[1] RXS bit), and the defined mode (the SPIm.MCSPI\_CHxCONF[13:12] TRM field) of the channel enabled.

The SPIm.MCSPI\_CHxSTAT[2] EOT bit gives the transfer status of each SPI word. The RXx\_FULL bit in the SPIm.MCSPI\_IRQSTATUS register is set when received data is loaded from the shift register to the MCSPI\_RXx register.

A change in the configuration parameters is propagated directly on the SPI interface. If the mcsnim\_csx signal is activated, ensure that the configuration is changed only between SPI words to avoid corrupting the current transfer.



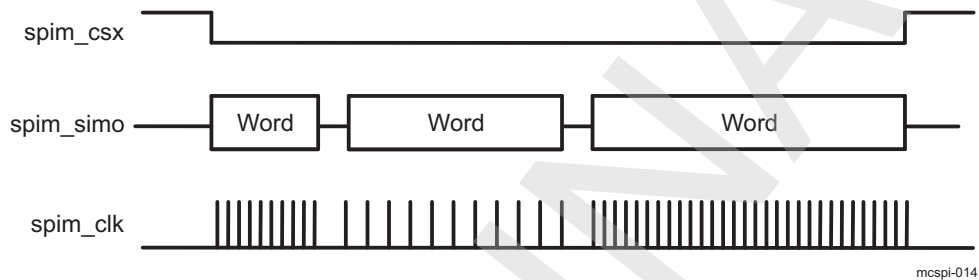
**NOTE:** To avoid data corruption, `mcs pim_clk` phase and `mcs pim_clk` polarity must not be modified when the `mcs pim_csx` signal is activated.

A delay between SPI words that requires the connected SPI slave device to switch from one configuration (transmit-only for instance) to another (receive-only for instance) must be handled by software.

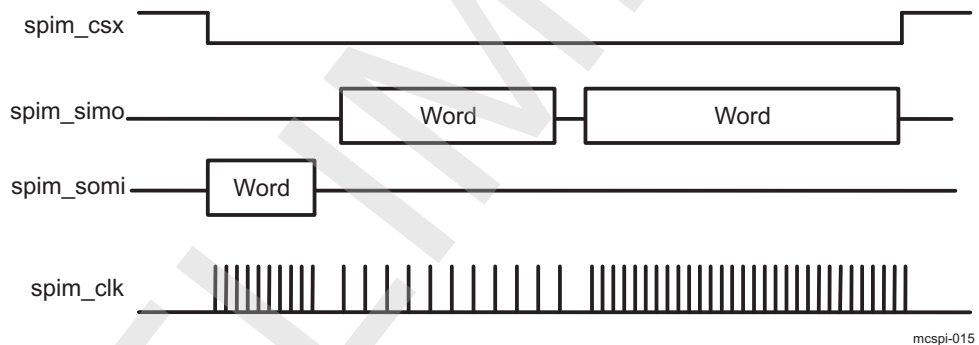
At the end of the last SPI word, the channel must be deactivated (the `SPI m.MCSPi_CHxCTRL[0]` EN bit set to 0) and `mcs pim_csx` can be forced to its inactive state using the `SPI m.MCSPi_CHxCONF[20]` FORCE bit.

Figure 23-99 and Figure 23-100 show successive transfers with `mcs pim_csx` maintained active low with a different configuration for each SPI word in single-data-pin and dual-data-pin interface modes, respectively.

**Figure 23-99. Continuous Transfers With `mcs pim_csx` Maintained Active (Single-Data-Pin Interface Mode)**



**Figure 23-100. Continuous Transfers With `mcs pim_csx` Maintained Active (Dual-Data-Pin Interface Mode)**



**NOTE:** The turbo mode described in Section 23.4.4.2.5.3, *Turbo Mode*, maintains `mcs pim_csx` in active mode when the following conditions are met:

- A single channel is explicitly used (the `SPI m.MCSPi_MODULCTRL[0]` SINGLE bit is set to 1).
- Turbo mode is enabled in the configuration of the channel. (The `SPI m.MCSPi_CHxCONF[19]` TURBO bit is set to 1.)

### 23.4.4.2.5.3 Turbo Mode

Turbo mode improves the throughput of the SPI interface when a single channel is enabled by allowing transfers until the shift register and the `MCSPi_RXx` register are full. Turbo mode is useful (time saving) when a transfer exceeds two words. This mode is programmable per channel (through the `SPI1.MCSPi_CHxCONF[9]` TURBO bit).

When several channels are enabled, the TURBO bit has no effect and the channel access to the shift registers remains as previously described.

In turbo mode, Rule 1 and Rule 2 apply, but Rule 3 does not (see Section 23.4.4.2.2). An enabled channel can be scheduled if its receive register is full (the `SPI m.MCSPi_CHxSTAT[0]` RXS bit) at the time of the shift-register assignment until the shift register is full.

The `MCSPi_RXx` register cannot be overwritten in turbo mode. Consequently, the `SPIm.MCSPi_IRQSTATUS[3] RX0_OVERFLOW` bit is never set in this mode.

#### 23.4.4.2.6 Start-Bit Mode

In start-bit mode, an extended bit is added before the SPI word to indicate whether the next SPI word must be handled as a command or as data. This feature is available only in master mode. Start-bit mode cannot be used at the same time as turbo mode and/or force `mcsxim_csx` mode. In this case, only one channel can be used; round-robin arbitration is not possible.

This mode is programmable per channel by setting the `SPIm.MCSPi_CHxCONF[23] SBE` bit to 1. The polarity of the extended bit is programmable per channel. When the `SPIm.MCSPi_CHxCONF[24] SBPOL` bit is set to 0, the SPI word must be handled as a command. When the `SPIm.MCSPi_CHxCONF[24] SBPOL` bit is set to 1, the SPI word must be handled as data. Moreover, start-bit polarity can be changed dynamically during start-bit transfer without disabling the channel for reconfiguration; in this case, users must configure the `SPIm.MCSPi_CHxCONF[24] SBPOL` bit before writing the SPI word to be transmitted to the TX register.

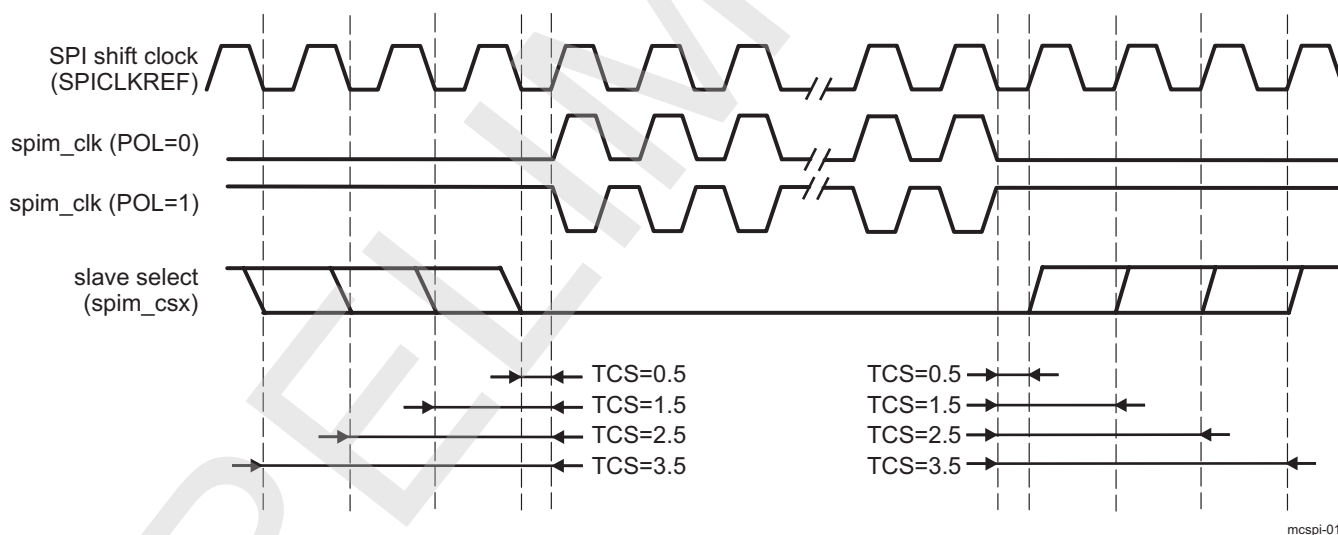
#### 23.4.4.2.7 Chip-Select Timing Control

The chip-select (CS) timing control is available only in master mode with automatic CS generation (the `MCSPi_MODULCTRL[0] SINGLE` bit field set to 0) to add a programmable delay between CS assertion and first clock edge, or CS removal and last clock edge.

This mode is programmable per channel (the `TCS` bit of the `SPIm.MCSPi_CHxCONF` register).

Figure 23-101 shows the CS SPIEN timing control.

**Figure 23-101. CS SPIEN Timing Controls**



**NOTE:** Because of the design implementation for transfers using a clock divider ratio set to 1 (clock bypassed), a half cycle must be added to the value between CS assertion and the first clock edge with `PHA = 1` or between CS removal and the last clock edge with `PHA = 0`.

#### 23.4.4.2.8 Programmable SPI Clock (`mcsxim_clk`)

In master mode, the baud rate of the SPI serial clock is programmable.

An internal reference clock, `SPIm_FCLK`, is used as input of a programmable divider (the `SPIm.MCSPi_CHxCONF[5:2] CLKD` bit field) to generate the bit rate of the serial output clock `mcsxim_clk`. Table 23-270 summarizes the supported divisor values.

**Table 23-270. SPI Master Clock Rates**

Divider	Clock Rate
1	48 MHz
2	24 MHz
4	12 MHz
8	6 MHz
16	3 MHz
32	1.5 MHz
64	750 kHz
128	375 kHz
256	~187 kHz
512	~93.7 kHz
1024	~46.8 kHz
2048	~23.4 kHz
4096	~11.7 kHz
8192 and higher: Division not supported	-

**23.4.4.2.8.1 Clock Ratio Granularity**

By default, the clock division ratio is defined by the SPIm.MCSPi\_CHxCONF[5:2] CLKD bit field with power-of-two granularity leading to a clock division in the range 1 to 4096; in this case, the duty cycle is always 50 percent. With the SPIm.MCSPi\_CHxCONF[29] CLKG bit, clock division granularity can be changed to one clock cycle; in that case the SPIm.MCSPi\_CHxCTRL[15:8] EXTCLK bit field is concatenated with the SPIm.MCSPi\_CHxCONF[5:2] CLKD bit field to give a 12-bit-wide division ratio in the range 1 to 4096.

When granularity is one clock cycle (CLKG set to 1), for the odd value of the clock ratio, the clock high level lasts one clock cycle more than the low level, depending on the SPIm.MCSPi\_CHxCONF[1] POL and SPIm.MCSPi\_CHxCONF[0] PHA bits (see Table 23-271).

**Table 23-271. CLKSPiO High/Low Time Computation**

Clock Ratio Fratio	CLKSPiO High Time	CLKSPiO Low Time
1	T <sub>high_ref</sub>	T <sub>low_ref</sub>
Even >=2	T <sub>ref</sub> * (Fratio/2)	T <sub>ref</sub> * (Fratio/2)
Odd >= (POL=PHA)	T <sub>ref</sub> * (Fratio - 1)/2	T <sub>ref</sub> * (Fratio + 1)/2
Odd >= (POL != PHA)	T <sub>ref</sub> * (Fratio + 1)/2	T <sub>ref</sub> * (Fratio - 1)/2

**NOTE:** Fratio = spi1\_clk frequency (Fout) division ratio.  
 T<sub>high</sub> = spi1\_clk high time period.  
 T<sub>low</sub> = spi1\_clk low time period.  
 T<sub>ref</sub> = SPI1\_FCLK period.  
 T<sub>high\_ref</sub> = SPI1\_FCLK high time period.  
 T<sub>low\_ref</sub> = SPI1\_FCLK low time period.

If CLKG = 1: Fratio = EXTCLK concatenated with CLKD + 1

For odd ratio values, the duty cycle is calculated as follows:

$$\text{Duty\_cycle} = (1 - 1/\text{Fratio})/2$$

Table 23-272 shows examples of clock granularity with a clock source frequency of 48 MHz.

**Table 23-272. Clock Granularity Examples**

EXTCLK	CLKD	CLKG	Fratio	PHA	POL	Thigh (ns)	Tlow (ns)	Tperiod (ns)	Duty Cycle	Fout (MHz)
X	0	0	1	X	X	10.4	10.4	20.8	50–50	48
X	1	0	2	X	X	20.8	20.8	41.6	50–50	24
X	2	0	4	X	X	41.6	41.6	83.2	50–50	12
X	3	0	8	X	X	83.2	83.2	166.4	50–50	6
0	0	1	1	X	X	10.4	10.4	20.8	50–50	48
0	1	1	2	X	X	20.8	20.8	41.6	50–50	24
0	2	1	3	1	0	41.6	20.8	62.4	66–33	16
0	2	1	3	1	1	20.8	41.6	62.4	33–66	16
0	3	1	4	X	X	41.6	41.6	83.2	50–50	12
5	0	1	81	1	0	852.8	832	1684.8	50.6–49.4	0.592
5	7	1	88	X	X	915.2	915.2	1830.4	50–50	0.545

#### 23.4.4.3 Slave Mode

To select the MCSPi slave mode, set the SPI1.MCSPi\_MODULCTRL[2] MS bit.

A MCSPi slave device can be connected to up to four external SPI master devices but handles transactions with one SPI master device at a time.

In slave mode, the MCSPi initiates data transfer on the data lines (mcspim\_simo and mcspim\_somi) when it is selected by an active control signal (mcspim\_csx) and receives an SPI clock (mcspim\_clk) from the external SPI master device. Only channel 0 can be configured as a slave. In slave mode, the MCSPi uses the edge of mcspim\_csx to detect word length. For this reason, mcspim\_csx must become inactive between each word.

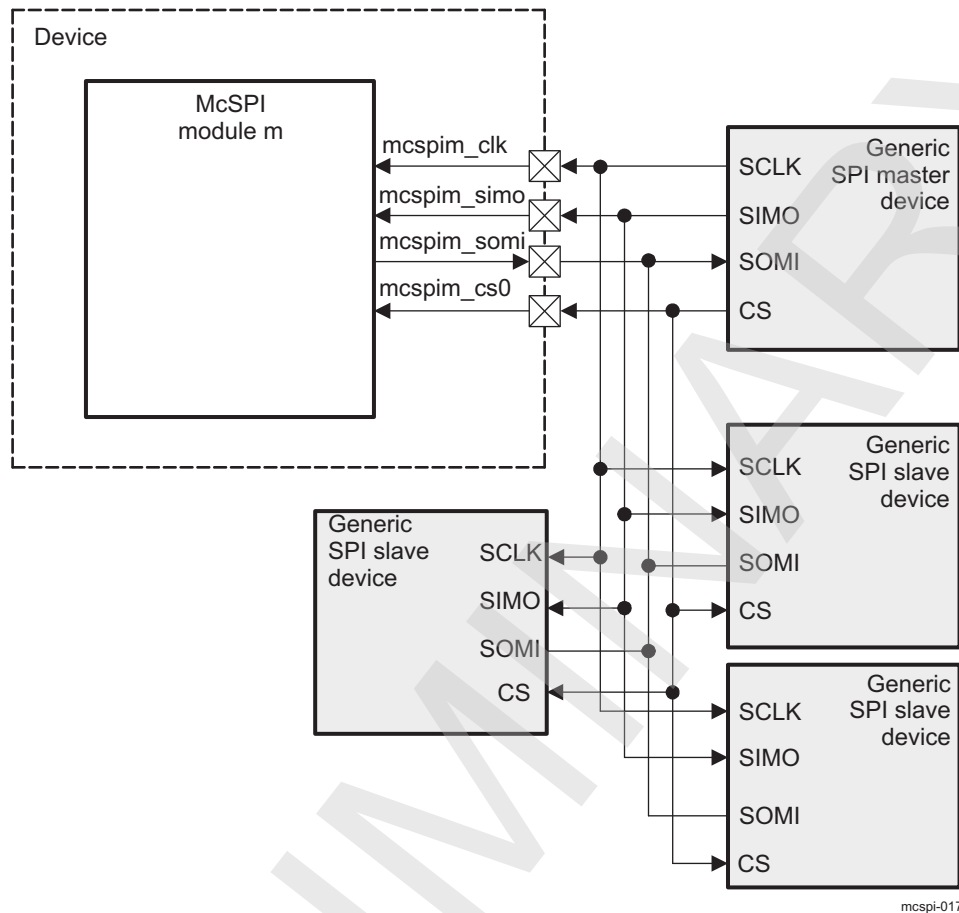
The MCSPi does not support mcspim\_csx active between SPI words. It uses the edge to detect word length.

##### 23.4.4.3.1 Dedicated Resources

Only channel 0 can be enabled in slave mode. In this section register names such as SPI1.MCSPi\_CHxCTRL stand for SPI1.MCSPi\_CH0CTRL, where x = 0 (channel 0 control register).

Figure 23-102 shows an example of four slaves wired on a single master device.

Figure 23-102. Example of MCSPi Slave With One Master and Multiple Slave Devices on Channel 0



The channel 0 in slave mode has the following resources:

- Its own channel enable, programmable with the SPIm.MCSPi\_CHxCTRL[0] EN bit (where x = 0). This channel must be enabled before transmission and reception.
- For this mode, the slave-select signal can be detected on any one of the mcspim\_csx ports. This is programmable with the SPIm.MCSPi\_CHxCONF[22:21] SPIENSLV bit field (where x = 0).
- Its own transmitter register, SPIm.MCSPi\_TXx (where x = 0), on top of the common transmit shift register. If the MCSPi\_TXx register is empty, the SPIm.MCSPi\_CHxSTAT[1] TXS bit (where x = 0) is set. If MCSPi is selected by an external master (the active signal on the mcspim\_csx port assigned to channel 0), the MCSPi\_TXx register content of channel 0 is always loaded into the shift register, whether its content is updated or not. The MCSPi\_TXx register must be loaded before MCSPi is selected by a master.
- Its own receiver register, SPIm.MCSPi\_RXx (where x = 0), on top of the common receive shift register. If the MCSPi\_RXx register is full, the SPIm.MCSPi\_CHxSTAT[0] RXS bit (where x = 0) is set.

**NOTE:** The MCSPi\_TXx register and MCSPi\_RXx registers of the other channels are not used. Reading from or writing to a channel register other than channel 0 has no effect.

- Its own communication configuration with the following parameters through the SPIm.MCSPi\_CHxCONF register (where x = 0):
  - Transmit and receive modes, programmable with the TRM field
  - Interface mode (two data pins or single data pin) and data pins assignment, both programmable with the IS and DPE bits. (The SPIm modules are in slave mode after reset and must be properly configured for the modules to act in master mode.)
  - SPI word length, programmable with the WL bit

- mcspim\_csx polarity, programmable with the EPOL bit
- mcspim\_clk polarity, programmable with the POL bit
- mcspim\_clk phase, programmable with the PHA bit

The mcspim\_clk frequency of a transfer is controlled by the external SPI master connected to the MCSPi slave device. The SPIm.MCSPi\_CHxCONF[5:2] CLKD bit field (where x = 0) is not used in slave mode.

---

**NOTE:** The configuration of the channel can be loaded in the SPIm.MCSPi\_CHxCONF register (where x = 0) only when the channel is disabled.

---

- Two DMA request events, read and write, synchronize read/write accesses of the DMA controller with the activity of MCSPi. DMA requests are asserted using the SPIm.MCSPi\_CHxCONF[15] DMAR bit (where x = 0) for reading and the SPIm.MCSPi\_CHxCONF[14] DMAW bit (where x = 0) for writing.
- Four interrupt events (see [Section 23.4.4.5.2, Interrupt Events in Slave Mode](#))

#### 23.4.4.3.2 Slave Transmit-and-Receive Mode

The slave receive mode is programmable (set the SPIm.MCSPi\_CHxCONF[13:12] TRM bit field [where x = 0] to 0x0).

In slave transmit-and-receive mode, the MCSPi\_TXx register must be loaded before MCSPi is selected by an external SPI master device.

After a channel is enabled, transmission and reception proceed with interrupt and DMA request events.

The MCSPi\_TXx register content is always loaded in the shift register whether it is updated or not. The event TXx\_UNDERFLOW is activated accordingly and does not prevent transmission.

When an SPI word transfer completes (the SPIm.MCSPi\_CHxSTAT0[2] EOT bit [where x = 0] set to 1), the received data is transferred to the channel receive register.

To use MCSPi as a slave transmit-only device, the RXx\_FULL and RX0\_OVERFLOW interrupts and DMA read requests must be disabled due to the MCSPi\_RXx register state (see [Section 23.4.4.5.2, Interrupt Events in Slave Mode](#)).

#### 23.4.4.3.3 Slave Transmit-Only Mode

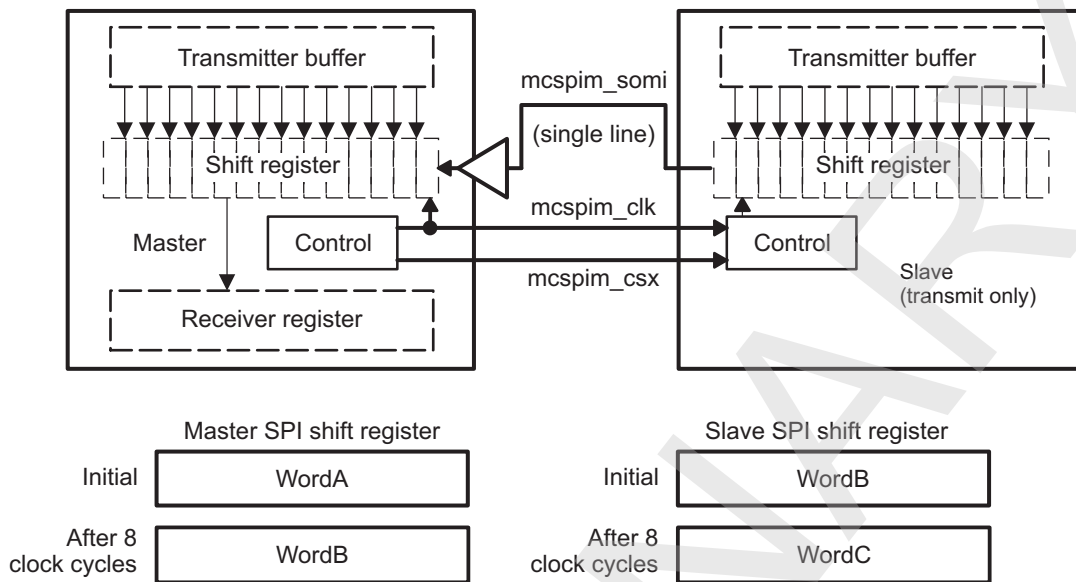
The slave transmit-only mode is programmable (set the SPIm.MCSPi\_CHxCONF[13:12] TRM bit field [where x = 0] to 0x2) and avoids the requirement for the MPU to read the MCSPi\_RXx register (minimizing data movement) only when transmission is meaningful.

To use the MCSPi as a slave transmit-only device, the RXx\_FULL and RX0\_OVERFLOW interrupts and DMA read requests must be disabled because of the MCSPi\_RXx register state.

When the SPI word transfer completes, the SPIm.MCSPi\_CHxSTAT[2] EOT bit is set (where x = 0).

[Figure 23-103](#) shows a half-duplex system with a master device on the left and a transmit-only slave device on the right. Each time a bit transfers out from the slave device, 1 bit transfers in the master. After eight cycles of the serial clock mcspim\_clk, WordB transfers from the slave to the master.

Figure 23-103. SPI Half-Duplex Transmission (Transmit-Only Slave)



#### 23.4.4.3.4 Slave Receive-Only Mode

The slave receive mode is programmable (set the `SPI m.MCSPI_CHxCONF[13:12]` TRM bit field [where  $x = 0$ ] to 0x1).

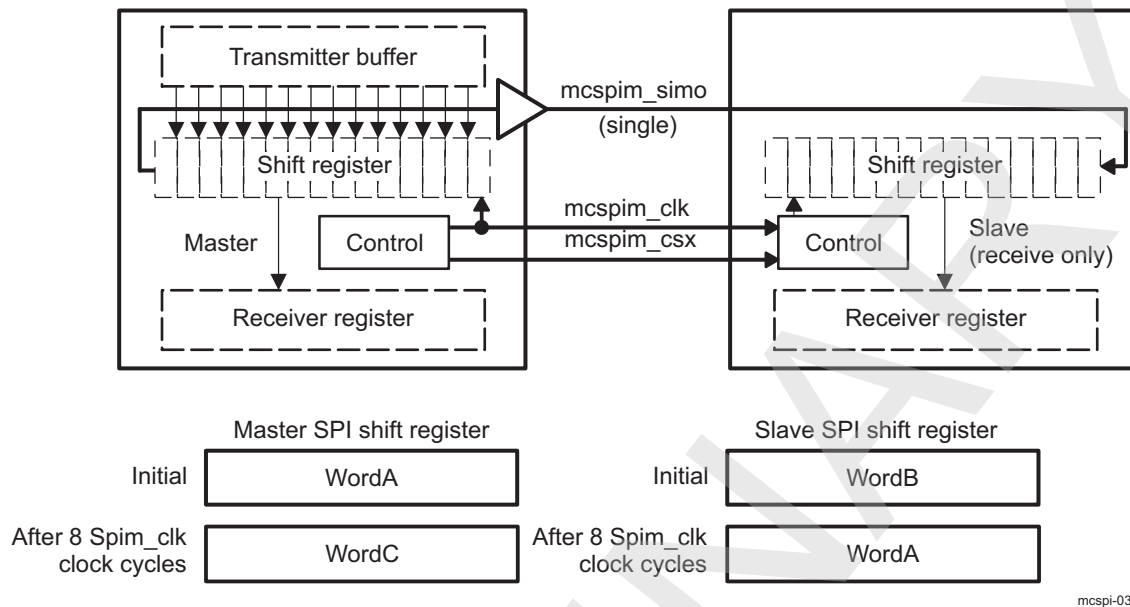
In receive-only mode, the `MCSPI_Tx` register must be loaded before the MCSPi is selected by an external SPI master device. The `MCSPI_Tx` register content is always loaded into the shift register whether it is updated or not. The `Tx_UNDERFLOW` event is activated accordingly and does not prevent transmission.

When an SPI word transfer completes (the `SPI m.MCSPI_CHxSTAT0[2]` EOT bit [where  $x = 0$ ] is set to 1), the received data is transferred to the channel receive register.

To use the MCSPi as a slave receive-only device, the `Tx_EMPTY` and `Tx_UNDERFLOW` interrupts and the DMA write requests must be disabled due to the `MCSPI_Tx` register state.

For a full-duplex transmission, the serial clock (`mcs pim_clk`) synchronizes shifting and sampling of the information on the single serial data line. For full duplex, two data lines are required. If `mcs pim_clk` synchronizes on a single serial data line, the data line should be half-duplex.

Figure 23-104 shows an example of a half-duplex system with a master device on the left and a receive-only slave device on the right. Each time a bit transfers out from the master, 1 bit transfers in from the slave. After eight cycles of the serial clock `mcs pim_clk`, Word A transfers from the master to the slave.

**Figure 23-104. SPI Half-Duplex Transmission (Receive-Only Slave)**

#### 23.4.4.4 FIFO Buffer Management

The MCSPI controller has a built-in 64-byte buffer to unload the DMA or interrupt handler and improve data throughput.

This buffer can be used by only one channel at a time and is selected by setting the `SPIm.MCSPI_CHxCONF[28] FFER` or `SPIm.MCSPI_CHxCONF[27] FFEW` bits to 1. If several channels are selected and several FIFO enable bit fields are set to 1, the controller forces the buffer not to be used; the driver must set only one FIFO enable bit field.

The buffer can be used in the following modes:

- Master or slave mode
- Transmit-only, receive-only, or transmit/receive mode
- Single channel or turbo mode, or normal round-robin mode. In round-robin mode the buffer is used by only one channel.

Every word length (`SPIm.MCSPI_CHxCONF[11:7] WL`) is supported.

In transmit/receive mode, the buffer can be used in transmit (see [Figure 23-105](#)) or receive (see [Figure 23-106](#)) directions, or in both directions. If only one direction is chosen in transmit/receive mode, the full buffer is used for this direction. In both directions, the buffer is split into two 32-byte buffers, one for each direction. See [Figure 23-107](#).

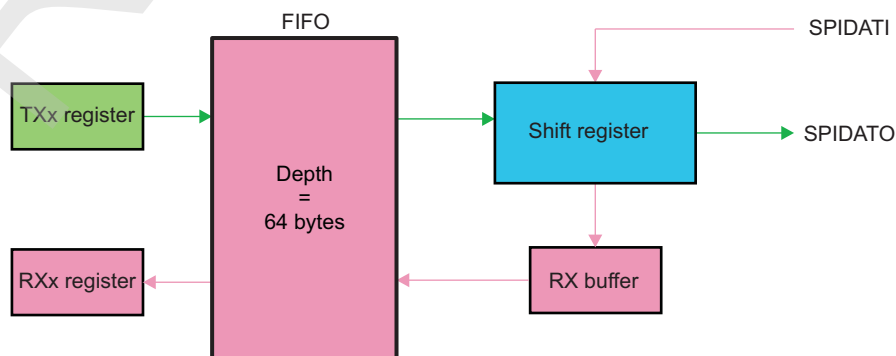
**Figure 23-105. Buffer Used in Transmit Direction Only**



Figure 23-106. Buffer Used in Receive Direction Only

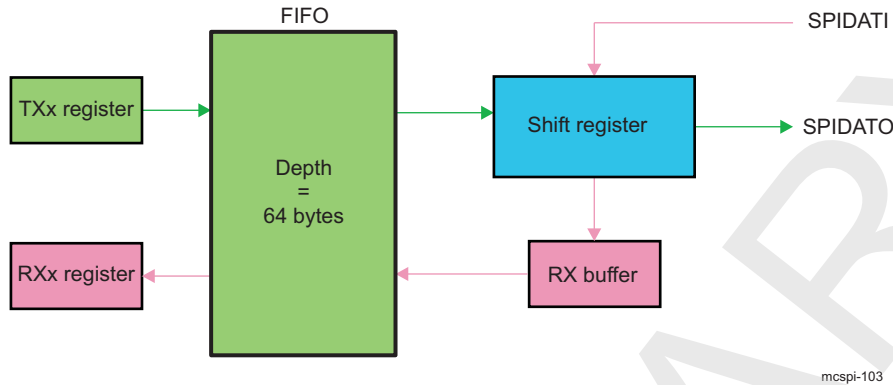
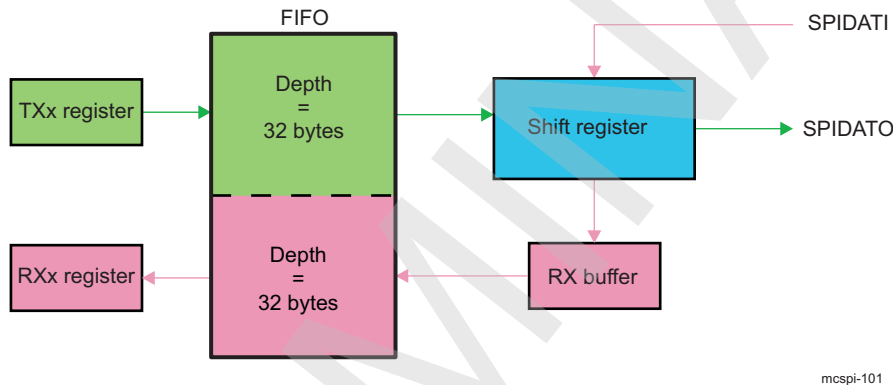


Figure 23-107. Buffer Used for Transmit and Receive Directions



Two levels (SPI<sub>m</sub>.MCSPI\_XFERLEVEL[5:0] AEL and SPI<sub>m</sub>.MCSPI\_XFERLEVEL[13:8] AFL) rule the buffer management. The granularity of these levels is 1 byte; it is not aligned with the SPI word length. The driver must set these values as a multiple of the SPI word length defined in WL. Table 23-273 shows the number of bytes written in the FIFO, depending on the word length.

Table 23-273. FIFO Writes, Word Length Relationship

SPI Word Length WL	3 ≤ WL ≤ 7	8 ≤ WL ≤ 15	16 ≤ WL ≤ 31
Number of byte written in the FIFO	1 byte	2 bytes	4 bytes

The FIFO buffer pointers are reset when the corresponding channel is enabled or the FIFO configuration changes.

#### 23.4.4.4.1 Buffer Almost Full

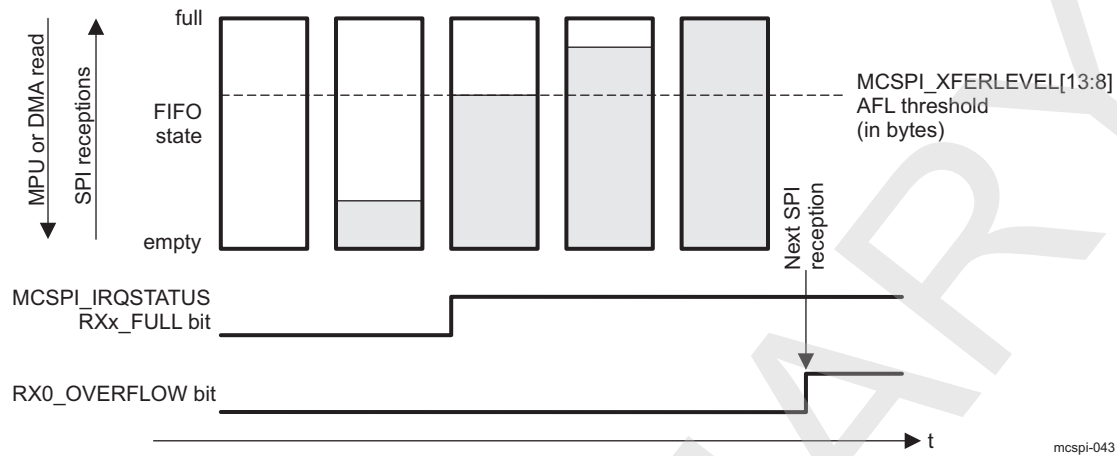
The MCSPI\_XFERLEVEL[15:8] AFL bit field is needed when the buffer is used to receive an SPI word from a slave (the MCSPI\_CHxCONF[28] FFER bit must be set to 1). It defines the almost-full buffer status. See Figure 23-108.

When the FIFO pointer reaches this level an interrupt or a DMA request is sent to the MPU to enable the system to read AFL + 1 bytes from the receive register.

**NOTE:** AFL + 1 must correspond to a multiple value of the MCSPI\_CHxCONF[11:7] WL bit field.

When DMA is used, the request is deasserted after the first receive register read.

No new request will be asserted again as long as the system has not performed the right number of read accesses.

**Figure 23-108. Buffer Almost Full Level (AFL)**

**NOTE:** The [MCSPI\\_IRQSTATUS](#) register bits are not available in DMA mode. In DMA mode, the SPIIm\_DMA\_RXx request is asserted on the same conditions as the [MCSPI\\_IRQSTATUS](#) RXx\_FULL flag.

#### 23.4.4.4.2 Buffer Almost Empty

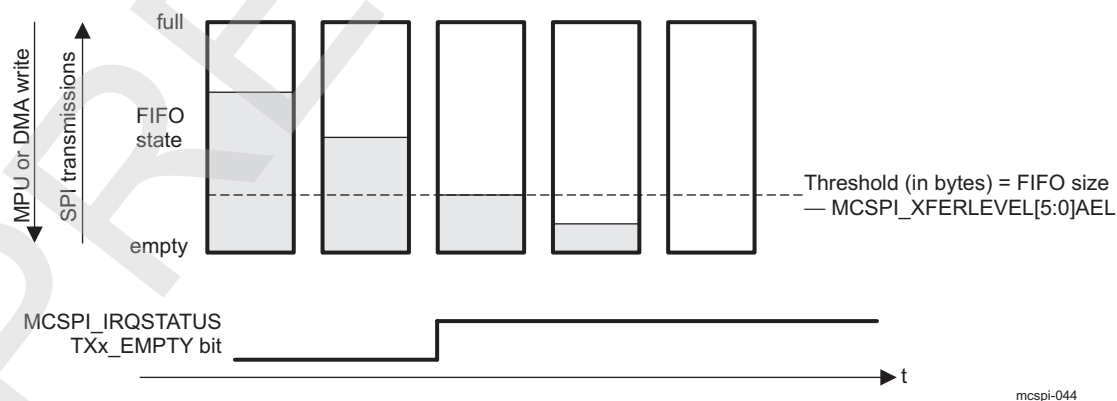
The [MCSPI\\_XFERLEVEL\[7:0\]](#) AEL bit field is needed when the buffer is used to transmit an SPI word to a slave (the [MCSPI\\_CHxCONF\[27\]](#) FFEW bit must be set to 1). It defines the almost-empty buffer status. See [Figure 23-109](#).

When the FIFO pointer has not reached this level, an interrupt or a DMA request is sent to the MPU to enable the system to write AEL + 1 bytes to the transmit register.

**NOTE:** AEL + 1 must correspond to a multiple value of the [MCSPI\\_CHxCONF\[11:7\]](#) WL bit field.

When DMA is used, the request is deasserted after the first transmit register write.

No new request will be asserted again as long as the system has not performed the right number of write accesses.

**Figure 23-109. Buffer Almost Empty Level (AEL)**

**NOTE:** The [MCSPI\\_IRQSTATUS](#) register bits are not available in DMA mode. In DMA mode, the SPIIm\_DMA\_TXx request is asserted on the same conditions as the [MCSPI\\_IRQSTATUS](#) TXx\_EMPTY flag.

#### 23.4.4.4.3 End of Transfer Management

When the FIFO buffer is enabled for a channel, the user must previously configure in the [MCSPi\\_XFERLEVEL](#) register the AEL and AFL levels and especially the [MCSPi\\_XFERLEVEL\[31:16\]](#) WCNT bit field to define the number of SPI words to be transferred using the FIFO before enabling the channel.

This counter allows the controller to stop the transfer correctly after a defined number of SPI word transfers. If WCNT is set to 0x0000, the counter is not used and the user must stop the transfer manually by disabling the channel; in this case, the user does not know how many SPI transfers have been done. For received words, software must poll the CHxSTAT[5] RXFFE bit and read the [MCSPi\\_RXx](#) receive register to empty the FIFO buffer.

When the end-of-word count interrupt is generated (the [MCSPi\\_IRQSTATUS\[17\]](#) EOW bit is set), the user can disable the channel and poll the [MCSPi\\_CHxSTAT\[5\]](#) RXFFE bit to know it lasts SPI words in the FIFO buffer and read them.

No new request will be asserted as long as the system has not performed the right number of write accesses.

#### 23.4.4.5 Interrupts

Each channel can issue interrupt events.

Each interrupt event has status bits in the SPIm.[MCSPi\\_IRQSTATUS](#) register (RXx\_FULL, TXx\_UNDERFLOW, TXx\_EMPTY, ...) [where x = 0, 3] that indicate whether service is required. Each status bit has an interrupt enable bit (a mask) in the SPIm.[MCSPi\\_IRQENABLE](#) register (RXx\_FULL\_ENABLE, TXx\_UNDERFLOW\_ENABLE, TXx\_EMPTY\_ENABLE, ...).

When an interrupt occurs and a mask is later applied on it, the interrupt line is not asserted again, even if the interrupt source is not serviced.

The MCSPi supports interrupt-driven and polling operations.

##### 23.4.4.5.1 Interrupt Events in Master Mode

In master mode, the interrupt events related to the [MCSPi\\_TXx](#) register state are TXx\_EMPTY and TXx\_UNDERFLOW. The interrupt event related to the [MCSPi\\_RXx](#) register state is RXx\_FULL.

###### 23.4.4.5.1.1 TXx\_EMPTY

The TXx\_EMPTY event is activated when a channel is enabled and its [MCSPi\\_TXx](#) register is empty (transient event). Enabling a channel automatically triggers this event, except in master receive-only mode (see [Section 23.4.4.2.4, Master Receive-Only Mode](#)). When the FIFO buffer is enabled (the [MCSPi\\_CHxCONF\[27\]](#) FFEW bit is set to 1), the [MCSPi\\_IRQSTATUS](#) TXx\_EMPTY bit is set as soon as there is enough space in the buffer to write a number of bytes defined by the [MCSPi\\_XFERLEVEL\[5:0\]](#) AEL bit field.

The [MCSPi\\_TXx](#) register must be loaded with data to remove the source of the interrupt; the SPIm.[MCSPi\\_IRQSTATUS](#) TXx\_EMPTY interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new TXx\_EMPTY event will be asserted as soon as the MPU has not performed the number of writes into the [MCSPi\\_TXx](#) register defined by the [MCSPi\\_XFERLEVEL\[5:0\]](#) AEL bit field. The MPU must perform the right number of writes.

###### 23.4.4.5.1.2 TXx\_UNDERFLOW

The event TXx\_UNDERFLOW is activated when the channel is enabled and if the [MCSPi\\_TXx](#) register or the FIFO is empty (not updated with new data) when an external master device starts a data transfer with the MCSPi (transmit and receive).

The TXx\_UNDERFLOW is a harmless warning in master mode.

To avoid having a TXx\_UNDERFLOW event at the beginning of a transmission, the event TXx\_UNDERFLOW is not activated when no data has been loaded into the MCSPI\_TXx register, because the channel has been enabled. To avoid having a TXx\_UNDERFLOW event, the MCSPI\_TXx register must seldom be loaded.

The SPI.MCSPI\_IRQSTATUS TXx\_UNDERFLOW interrupt status bit must be cleared for interrupt line deassertion (if event enabled as the interrupt source).

#### 23.4.4.5.1.3 RXx\_FULL

The RXx\_FULL event is activated when channel is enabled and MCSPI\_RXx register becomes filled (transient event). When the FIFO buffer is enabled (the MCSPI\_CHxCONF[28] FFER bit is set to 1), RXx\_FULL is asserted as soon as the number of bytes holds in the FIFO to be read reaches the MCSPI\_XFERLEVEL[13:8] AFL threshold.

The MCSPI\_RXx register must be read to remove the source of the interrupt; the MCSPI\_IRQSTATUS RXx\_FULL interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new RXx\_FULL event will be asserted as soon as the MPU has not performed AFL + 1 reads into MCSPI\_RXx. The MPU must perform the right number of reads.

#### 23.4.4.5.1.4 End Of Word Count

The MCSPI\_IRQSTATUS[17] EOW event (end of word count) is activated when the channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller performs the number of transfers defined in the MCSPI\_XFERLEVEL[31:16] WCNT bit field. If WCNT is set to 0x0000, the counter is not enabled and this interrupt is not generated.

The end of word count interrupt also indicates that the SPI transfer is halted on the channel using the FIFO buffer as soon as MCSPI\_XFERLEVEL[31:16] WCNT is not reloaded and the channel is not re-enabled.

The MCSPI\_IRQSTATUS[17] EOW interrupt status bit must be cleared for interrupt line deassertion (if the event enabled as interrupt source).

#### 23.4.4.5.2 Interrupt Events in Slave Mode

In slave mode, the interrupt events related to the MCSPI\_TXx register state are TXx\_EMPTY and TXx\_UNDERFLOW. The interrupt events related to the MCSPI\_RXx register state are RXx\_FULL and RX0\_OVERFLOW (channels 1, 2, and 3 do not have a receiver overflow status bit). See the MCSPI\_IRQSTATUS register.

##### 23.4.4.5.2.1 TXx\_EMPTY

The TXx\_EMPTY event is activated when a channel is enabled and its MCSPI\_TXx register is empty. Enabling the channel automatically raises this event. If the FIFO buffer is enabled (the MCSPI\_CHxCONF[27] FFEW bit is set to 1), the TXx\_EMPTY event will be asserted as soon as there is enough space in buffer to write a number of bytes defined by the MCSPI\_XFERLEVEL[5:0] AEL bit field.

The MCSPI\_TXx register must be loaded with data to remove the source of the interrupt; the SPI.MCSPI\_IRQSTATUS TXx\_EMPTY interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new TXx\_EMPTY event will be asserted as soon as the MPU has not performed the number of write into the MCSPI\_TXx register defined by MCSPI\_XFERLEVEL[5:0] AEL bit field. The MPU must perform the right number of writes.

##### 23.4.4.5.2.2 TXx\_UNDERFLOW

The TXx\_UNDERFLOW event is activated when a channel is enabled and if the MCSPI\_TXx register is empty (not updated with new data) when an external master device starts a data transfer with the MCSPI (transmit and receive).

When FIFO is enabled, the data emitted while the underflow event is raised is not the last data written in the FIFO but the next data in the FIFO (an old transmitted value or a dummy data in the FIFO has been reset).

TXx\_UNDERFLOW indicates an error (data loss) in slave mode.

To avoid having a TXx\_UNDERFLOW event at the beginning of a transmission, the TXx\_UNDERFLOW event is not activated when data is not loaded into the **MCSPi\_TXx** register because the channel is enabled.

The SPIm.**MCSPi\_IRQSTATUS** TXx\_UNDERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

#### **23.4.4.5.2.3 RXx\_FULL**

The RXx\_FULL event is activated when a channel is enabled and the **MCSPi\_RXx** register is being filled (transient event). When the FIFO buffer is enabled (the **MCSPi\_CHxCONF**[28] FFER bit is set to 1), RXx\_FULL will be asserted as soon as there is a number of bytes holds in buffer to read defined by the **MCSPi\_XFERLEVEL**[13:8] AFL bit field.

The **MCSPi\_RXx** register must be read to remove the source of the interrupt; the SPIm.**MCSPi\_IRQSTATUS** RXx\_FULL interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new RXx\_FULL event will be asserted as soon as the MPU has not performed AFL + 1 reads into **MCSPi\_RXx**. The MPU must perform the right number of reads.

#### **23.4.4.5.2.4 RX0\_OVERFLOW**

The RX0\_OVERFLOW event is activated in slave mode in either transmit-and-receive or receive-only mode, when a channel is enabled and the **MCSPi\_RXx** register or FIFO is full when a new SPI word is received. The **MCSPi\_RXx** register is always overwritten with the new SPI word. If the FIFO is enabled data within the FIFO are overwritten, it must be considered as corrupted. The RX0\_OVERFLOW event should not appear in slave mode using the FIFO.

The RX0\_OVERFLOW indicates an error (data loss) in slave mode.

The **MCSPi\_IRQSTATUS**[3] RX0\_OVERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

#### **23.4.4.5.2.5 End Of Word Count**

The **MCSPi\_IRQSTATUS**[17] EOW event (end of word count) is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfer defined in the **MCSPi\_XFERLEVEL**[31:16] WCNT bit field. If WCNT is set to 0x0000, the counter is not enabled and this interrupt is not generated.

The end of word count interrupt also indicates that the SPI transfer is halted on the channel using the FIFO buffer as soon as WCNT is not reloaded and channel re-enabled.

The **MCSPi\_IRQSTATUS**[17] EOW interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

#### **23.4.4.5.3 Interrupt-Driven Operation**

An interrupt enable bit in the SPIm.**MCSPi\_IRQENABLE** register can be set to enable each event to generate interrupt requests when the corresponding event occurs. Status bits are automatically set by hardware logic conditions.

When an event occurs (the single interrupt line is asserted), the MPU must:

1. Read the SPIm.**MCSPi\_IRQSTATUS** register to identify which event occurred.

2. Read the [MCSPi\\_RXx](#) register that corresponds to the event to remove the source of an [RXx\\_FULL](#) event or write into the [MCSPi\\_TXx](#) register that corresponds to the event to remove the source of a [TXx\\_EMPTY](#) event. No action is required to remove the source of the [WKS](#) (wake-up), [TXx\\_UNDERFLOW](#), and [RX0\\_OVERFLOW](#) events.
3. Write 1 into the corresponding bit of the SPIm.[MCSPi\\_IRQSTATUS](#) register to clear an interrupt status and then release the interrupt line.

The interrupt status bit must always be reset after channel enabling and before events are enabled as interrupt sources.

#### 23.4.4.5.4 Polling

When the interrupt capability of an event is disabled in the SPIm.[MCSPi\\_IRQENABLE](#) register, the interrupt line is not asserted, but the status bits in the SPIm.[MCSPi\\_IRQSTATUS](#) register can be polled by software to detect when the corresponding event occurs.

Once the expected event occurs:

- [RXx\\_FULL](#): To remove the source of the event, the MPU must read the corresponding [MCSPi\\_RXx](#) register.
- [TXx\\_EMPTY](#): To remove the source of the event, the MPU must write into the corresponding [MCSPi\\_TXx](#) register.
- [WKS](#) (wake-up), [TXx\\_UNDERFLOW](#), and [RX0\\_OVERFLOW](#): No action is required to remove the source of the event.

To clear an interrupt, set the corresponding status bit of the SPIm.[MCSPi\\_IRQSTATUS](#) register to 1. This does not affect the interrupt line state.

#### 23.4.4.6 DMA Requests

The sDMA controller module manages DMA accesses. The sDMA controller advantage is to lower the MPU charge for data transfers.

Each MCSPi channel can issue DMA requests if they are enabled. There are two DMA request lines per MCSPi channel (one for read and one for write).

The DMA read request line is asserted when the MCSPi channel is enabled and new data is available in the receive register of the MCSPi channel. A DMA read request can be individually masked with the SPI1.[MCSPi\\_CHxCONF](#)[15] DMAR bit. The DMA read request line is deasserted on read completion of the [MCSPi\\_RXx](#) register of the MCSPi channel.

The DMA write request line is asserted when the MCSPi channel is enabled and the [MCSPi\\_TXx](#) register of the MCSPi channel is empty. A DMA write request can be individually masked with the SPI1.[MCSPi\\_CHxCONF](#)[14] DMAW bit. The DMA write request line is deasserted on load completion of the [MCSPi\\_TXx](#) register of the channel.

#### 23.4.4.7 Power Saving Management

Power consumption can be optimized by switching off internal clocks (interface and functional clock) when there is no activity. The MCSPi is compliant with the idle and wake-up system handshake protocol.

##### 23.4.4.7.1 Normal Mode

In normal mode, internal SPI module clocks are automatically switched off (autogating) when there is no activity in slave or master mode.

Autogating of the module interface clock and functional clock occurs when the following conditions are met:

- The SPIm.[MCSPi\\_SYSCONFIG](#)[0] AUTOIDLE bit is set.
- In master mode, there is no data to transmit or receive in all channels.
- In slave mode, the MCSPi is not selected by the external master and there are no register accesses.

Autogating of the module interface clock and functional clock stops when the following conditions are met:

- In master mode, an internal access occurs.
- In slave mode, an internal access occurs or the MCSPi is selected by the external master.

#### **23.4.4.7.2 Idle Mode**

At the PRCM module level, when all conditions are met to shut off the CORE\_48M\_FCLK or CORE\_L4\_ICLK output clocks (see [Chapter 3, Power, Reset, and Clock Management](#), for details), the PRCM module automatically launches a hardware handshake protocol to ensure that the MCSPi is ready to have its clocks switched off. Namely, the PRCM module asserts an idle request to the MCSPi.

Although this handshake is completely hardware-oriented and out of software control, the method in which the MCSPi module acknowledges the PRCM idle request is configurable through the MCSPi.SYSCONFIG[4:3] SIDLEMODE bit.

The settings of the SIDLEMODE bit and the related acknowledgment modes are:

- Force-idle mode (the SPIm.MCSPi\_SYSCONFIG[4:3] SIDLEMODE bit is set to 0x0): The MCSPi module acknowledges unconditionally the idle request from the PRCM module, regardless of its internal operations. This mode must be used carefully in this case because it does not prevent the loss of data when the clock is switched off.
- No-idle mode the (SIDLEMODE bit is set to 0x1): The MCSPi module never acknowledges an idle request from the PRCM module and is safe from a module point of view because it ensures that the clocks remain active. However, it is not efficient to save power because it does not allow the PRCM output clock to be shut off and thus the power domain to be set to a lower power state.
- Smart-idle mode (the SIDLEMODE bit set to 0x2): The MCSPi module acknowledges the idle request, basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, IRQs, or DMA requests are treated. This is the best approach for efficient system power management.

When configured in smart-idle mode, the MCSPi module also offers an additional granularity on the CORE\_48M\_FCLK and CORE\_L4\_ICLK gating. The SPIm.SYSCONFIG[9:8] CLOCKACTIVITY bit field determines which clock shuts down (the CORE\_48M\_FCLK, CORE\_L4\_ICLK, neither clock, or both clocks).

The CLOCKACTIVITY setting is used internally to MCSPi to determine on which part of the module the conditions to acknowledge the PRCM idle request are tested. For example, if CORE\_48M\_FCLK is not shut down on a PRCM idle request, MCSPi considers only CORE\_L4\_ICLK and the associated pending activities before acknowledging the request.

Some MCSPi features are associated with CORE\_L4\_ICLK and others with CORE\_48M\_FCLK. Using the CLOCKACTIVITY bit field along with the smart-idle mode ensures that the features associated with the clock that remains active are always enabled, even if MCSPi acknowledges an idle request.

The CLOCKACTIVITY settings and the associated features are:

- CLOCKACTIVITY set to 00: ICLK OFF and FCLK OFF, ICLK and FCLK are considered for generating the acknowledge. This setting also means that FCLK and ICLK are likely to be shut down on a PRCM idle request.
- CLOCKACTIVITY set to 01: ICLK ON and FCLK OFF, ICLK is not shut down on a PRCM idle request; only FCLK is concerned.
- CLOCKACTIVITY set to 10: ICLK OFF and FCLK ON, FCLK is not shut down on a PRCM idle request; only ICLK is concerned.
- CLOCKACTIVITY set to 11: ICLK ON and FCLK ON, none of the clocks are shut down. This means MCSPi can potentially acknowledge the idle request without checking the internal functions linked to its clocks.

**CAUTION**

The PRCM module does not have a hardware means of reading the CLOCKACTIVITY settings. Therefore, software must ensure consistent programming between CLOCKACTIVITY and the CORE\_48M\_FCLK and CORE\_L4\_ICLK control bits in the PRCM module. If the MCSPi is disabled in the CM\_FCLKEN and CM\_ICLKEN PRCM registers while CLOCKACTIVITY is set to 11, nothing prevents the PRCM module from asserting its idle request, which is acknowledged regardless of the features associated with the MCSPi clocks. This can lead to unpredictable behavior.

**23.4.4.7.2.1 Wake-Up Event in Smart-Idle Mode**

The module wake-up feature is enabled when the SPIm.MCSPi\_SYSCONFIG[2] ENAWAKEUP and SPIm.MCSPi\_WAKEUPENABLE[0] WKEN bits are set. Wake-up capability is relevant only when the module is configured in slave mode.

The module generates an asynchronous wake-up request to the system power manager to switch back the interface clock and the functional clock. A wake-up is requested when channel 0 is enabled and an asynchronous selection occurs on the mcsxim.csx port associated with channel 0 (see the definition for the SPIm.MCSPi\_CHxCONF SPIENSLV bit field [where x = 0] in the register description table).

After the McSPi wake-up request, the system power manager must reactivate the interface clock:

- Before the beginning of the second SPI word serialization when McSPi is in slave transmit-only mode or in slave transmit-and-receive mode
- Before the end of the second received SPI word in slave receive-only mode. To avoid data loss, the first received SPI word must be read from the SPIm.MCSPi\_RXx register (where x = 0) before the completion of the second SPI word serialization.

Table 23-274 lists the supported cases in wake-up mode.

**Table 23-274. Smart-Idle Mode and Wake-Up Capabilities**

Mode	Interface Clock	SPI Clock Ref	Functionality	Wake-Up Event
Master	Must be maintained	Must be maintained	Full functionality, but the module does not generate a new interrupt or DMA request until the system exits wake-up mode	No wake-up event
Slave	Can be switched off	Can be switched off	An SPI word can be transmitted and/or received, but the module does not generate any new interrupts or DMA requests until the system exits wake-up mode.	The module asynchronously sends a wake-up request if an event on the mcsxim.csx port associated with channel 0 is detected.

In wake-up mode, the interrupt and DMA request lines are no longer asserted.

Any access to the module in wake-up mode generates an error as long as the interface clock is alive.

**23.4.4.7.2.2 Transitions From Smart-Idle Mode to Normal Mode**

The MCSPi detects the end of the wake period through the idle and wake-up hardware handshake protocol.

The interrupt status register (the SPIm.MCSPi\_IRQSTATUS[16] WKS bit) is updated with the event causing the wake-up; the wake-up event at the origin of the transition to the normal mode is converted to its corresponding interrupt when enabled by the SPIm.MCSPi\_IRQENABLE[16] WKE bit or the DMA request.

Interrupts and wake-up events have independent enable and disable controls, accessible through the SPIm.MCSPi\_IRQENABLE and SPIm.MCSPi\_WAKEUPENABLE registers. Software must ensure the overall consistency.



The interrupt status register SPIm.MCSPi\_IRQSTATUS is updated with the event causing the wake-up; the wake-up event at the origin of the transition to normal mode is converted to its corresponding interrupt request or DMA request. The module is fully operational.

#### **23.4.4.7.2.3 Force-Idle Mode**

Force-idle mode is enabled and exited as follows:

- Force-idle mode is enabled when the SPIm.MCSPi\_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x0.

In force-idle mode, the MCSPi responds unconditionally to the idle request by deasserting unconditionally the interrupt and DMA request lines, if asserted. In addition, the wake-up capability is totally inhibited even if the SPIm.MCSPi\_SYSCONFIG[2] ENAWAKEUP and SPIm.MCSPi\_WAKEUPENABLE[0] WKEN bits are set.

The transition from normal mode to idle mode does not affect the interrupt event bits of the SPIm.MCSPi\_IRQSTATUS register.

In force-idle mode, because the module must be disabled, the interrupt and DMA request lines are likely deasserted. The interface clock and SPI clock provided to the MCSPi can be switched off.

An idle request during an SPI data transfer can lead to an unexpected and unpredictable result. Software must avoid such a request.

- The module exits force-idle mode through the idle and wake-up hardware handshake protocol. The module is fully operational. The interrupt and DMA request lines are optionally asserted one clock cycle later.

## 23.4.5 MCSPi Programming Guide

### 23.4.5.1 MCSPi Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the MCSPi module.

#### 23.4.5.1.1 Global Initialization

##### 23.4.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the I<sup>2</sup>C module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the I<sup>2</sup>C. See the *I<sup>2</sup>C Module Integration and Environment* sections for further information.

See [Section 23.4.3](#) for more information.

**Table 23-275. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	MCSPi_FCLK functional clock must be enabled. See <a href="#">Section 3.1.1.1.2</a> , <i>Module Level Clock Management</i> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
L4 Interconnect	For information about L4-PER interconnect configuration, see <a href="#">Section 13.3, L4 Interconnects</a> , in <a href="#">Chapter 13, Interconnect</a> .
sDMA	DMA configuration must be done to enable the module DMA channel requests. See <a href="#">Chapter 16, sDMA</a> .
MPU INTC	Cortex-A9 MPU, Cortex-M3 MPU, or DSP INTC must be configured to enable the interrupt request generation to the Cortex-A9 MPU or Cortex-M3 MPU, or in <a href="#">INTC</a> , in <a href="#">Chapter 5, DSP Subsystem</a> .

##### 23.4.5.1.1.2 MCSPi Global Initialization

###### 23.4.5.1.1.2.1 Main Sequence – MCSPi Global Initialization

This procedure initializes the MCSPi module after a POR or software reset.

**Table 23-276. MCSPi Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Perform a software reset.	SPIm.MCSPi_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is finished?	SPIm.MCSPi_SYSSTATUS[0] RESETDONE	=0x1

##### 23.4.5.1.2 Operational Mode Configuration

###### 23.4.5.1.2.1 MCSPi Operational Modes

The selection of the working mode is done with the SPIm.MCSPi\_CHxCONF register. [Table 23-277](#) through [Table 23-279](#) list the possible operating modes and their configurations.

**Table 23-277. MCSPi Receive Mode initialization**

Step	Register/Bit Field/Programming Model	Value
Set receive mode.	SPIm.MCSPi_CHxCONF[13:12] TRM	0x1
Set the word length.	SPIm.MCSPi_CHxCONF[11:7] WL	0x8
Clock initialization and channel enabling	SPIm.MCSPi_MODULCTRL[2] MS	0x0
	SPIm.MCSPi_CHxCTRL[0] EN	0x1
Channels activated low during active state	SPIm.MCSPi_CHxCONF[6] EPOL	0x1
Clock held high during active state	SPIm.MCSPi_CHxCONF[1] POL	0x0

**Table 23-277. MCSPi Receive Mode initialization (continued)**

Step	Register/Bit Field/Programming Model	Value
Data latched on odd-numbered edges of the SPI clock	SPIm.MCSPi_CHxCONF[0] PHA	0x0
Reset the status bits.	SPIm.MCSPi_IRQSTATUS	0x0

**Table 23-278. MCSPi Transmit Mode initialization**

Step	Register/Bit Field/Programming Model	Value
Set transmit mode.	SPIm.MCSPi_CHxCONF[13:12] TRM	0x2
Set the word length.	SPIm.MCSPi_CHxCONF[11:7] WL	0x8
Clock initialization and channel enabling	SPIm.MCSPi_MODULCTRL[2] MS SPIm.MCSPi_CHxCTRL[0] EN	0x0 0x1
Channels activated low during active state	SPIm.MCSPi_CHxCONF[6] EPOL	0x1
Clock held high during active state	SPIm.MCSPi_CHxCONF[1] POL	0x0
Data latched on odd-numbered edges of the SPI clock	SPIm.MCSPi_CHxCONF[0] PHA	0x0
Reset the status bits.	SPIm.MCSPi_IRQSTATUS	0x0

**Table 23-279. MCSPi Transmit and Receive Mode initialization**

Step	Register/Bit Field/Programming Model	Value
Set transmit and receive mode.	SPIm.MCSPi_CHxCONF[13:12] TRM	0x0
Set the word length.	SPIm.MCSPi_CHxCONF[11:7] WL	0x8
Clock Initialization and channel enabling	SPIm.MCSPi_MODULCTRL[2] MS SPIm.MCSPi_CHxCTRL[0] EN	0x0 0x1
Channels activated low during active state	SPIm.MCSPi_CHxCONF[6] EPOL	0x1
Clock held high during active state	SPIm.MCSPi_CHxCONF[1] POL	0x0
Data latched on odd-numbered edges of the SPI clock	SPIm.MCSPi_CHxCONF[0] PHA	0x0
Reset the status bits.	SPIm.MCSPi_IRQSTATUS	0x0

### 23.4.5.1.2.1.1 Common Transfer Procedures Without FIFO – Polling Method

#### 23.4.5.1.2.1.1.1 Receive-Only Procedure – Polling Method

Table 23-280 lists the receive-only procedure using the polling method. The MCSPi is acting as slave.

**Table 23-280. Receive-Only Procedure – Polling Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode	See Table 23-277	
Start the channel	SPIm.MCSPi_CHxCTRL[0] EN	0x1
Wait until receive register is full?	SPIm.MCSPi_RXx	=0x1
Stop the channel	SPIm.MCSPi_CHxCTRL[0] EN	0x0

#### 23.4.5.1.2.1.1.2 Receive-Only Procedure – Interrupt Method

Table 23-281 lists the receive-only procedure using the interrupt method. The MCSPi is acting as slave.

**Table 23-281. Receive-Only Procedure – Interrupt Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 23-277	
Start the channel.	SPIm.MCSPi_CHxCTRL[0] EN	0x1
Enable the interrupt for the receiver register.	SPIm.MCSPi_IRQENABLE[2] RX_FULL_ENABLE	0x1

**Table 23-281. Receive-Only Procedure – Interrupt Method (continued)**

Step	Register/Bit Field/Programming Model	Value
Read the status register.	SPIm.MCSPI_IRQSTATUS[2] RX_FULL	0x0
Disable the interrupt.	SPIm.MCSPI_IRQENABLE[2] RX_FULL_ENABLE	0x0
Stop the channel.	SPIm.MCSPI_CHXCTRL[0] EN	0x0
Read the receiver register.	SPIm.MCSPI_RXx	xxxx

#### 23.4.5.1.2.1.1.3 Transmit-Only Procedure - Polling Method

Table 23-282 lists the transmit-only procedure using the polling method. The MCSPI is acting as master.

**Table 23-282. Transmit-Only Procedure – Polling Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 23-278	
Start the channel.	SPIm.MCSPI_CHXCTRL[0] EN	0x1
Wait until end of transfer?	SPIm.MCSPI_CHXSTAT[2:1]	=0x2
Stop the channel.	SPIm.MCSPI_CHXCTRL[0] EN	0x0

#### 23.4.5.1.2.1.1.4 Transmit and Receive Procedure – Polling Method

Table 23-282 lists the transmit and receive procedure using the polling method. The MCSPI is acting as master and slave.

**Table 23-283. Transmit and Receive Procedure – Polling Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 23-279	
Start the channel.	SPIm.MCSPI_CHXCTRL[0] EN	0x1
Wait until transmit/receive word?	SPIm.MCSPI_CHXSTAT[2:0]	=0x3
Stop the channel.	SPIm.MCSPI_CHXCTRL[0] EN	0x0

#### 23.4.5.1.3 Common Transfer Procedures With FIFO – Polling Method

When using FIFO the MCSPI module can start the transfer only after the first write request is released by writing the MCSPI\_TXx register. The first write request can be managed by the IRQ routine or DMA handler. The end of transfer is more complex and depends on the transfer type. See Table 23-282 through Table 23-289.

##### 23.4.5.1.3.1 Receive-Only Procedure With Word Count – Polling Method

**Table 23-284. Receive-Only Procedure With Word Count – Polling Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 23-277	
Start the channel.	SPIm.MCSPI_CHXCTRL[0] EN	0x1
Wait until end of word count?	SPIm.MCSPI_IRQSTATUS[17] EOW	=0x1
Stop the channel.	SPIm.MCSPI_CHXCTRL[0] EN	0x0
Read from the receiver register.	SPIm.MCSPI_RXx	xxxx

##### 23.4.5.1.3.2 Transmit-Only Procedure With and Without Word Count – Polling Method

**Table 23-285. Transmit-Only Procedure Without Word Count – Polling Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See <a href="#">Table 23-278</a>	
Start the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x1
Wait until end of word count?	SPIm.MCSPI_IRQSTATUS[17] EOW	=0x1
Wait until end of transfer?	SPIm.MCSPI_CHxSTAT[2] EOT SPIm.MCSPI_CHxSTAT[3] TXFFE	=0x1 =0x1
Stop the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x0

### 23.4.5.1.3.3 Transmit-Only Procedure With and Without Word Count – Interrupt Method

**Table 23-286. Transmit-Only Procedure With Word Count – Interrupt Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See <a href="#">Table 23-278</a>	
Start the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x1
Enable the interrupt for the transmit register.	SPIm.MCSPI_IRQENABLE[4] TX_EMPTY_ENABLE	0x1
End of word count.	SPIm.MCSPI_IRQSTATUS[17] EOW	=0x1
End of transfer	SPIm.MCSPI_CHxSTAT[2] EOT SPIm.MCSPI_CHxSTAT[3] TXFFE	=0x1 =0x1
Clear the interrupt.	SPIm.MCSPI_IRQENABLE[17] EOW_ENABLE	=0x0
Disable the interrupt for the transmit register.	SPIm.MCSPI_IRQENABLE[4] TX_EMPTY_ENABLE	0x0
Stop the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x0

### 23.4.5.1.3.4 Transmit and Receive Procedure with Word Count – Polling Method

**Table 23-287. Transmit and Receive Procedure With Word Count – Polling Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See <a href="#">Table 23-279</a>	
Start the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x1
Wait until end of word count?	SPIm.MCSPI_IRQSTATUS[17] EOW	=0x1
Stop the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x0
Read from the receiver register.	SPIm.MCSPI_RXx	xxxx

### 23.4.5.1.3.5 Transmit and Receive Procedure with Word Count – Interrupt Method

**Table 23-288. Transmit and Receive Procedure With Word Count – Interrupt Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See <a href="#">Table 23-279</a>	
Start the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x1
Enable the interrupt for the receiver register.	SPIm.MCSPI_IRQENABLE[2] RX_FULL_ENABLE	0x1
Enable the interrupt for the transmit register.	SPIm.MCSPI_IRQENABLE[4] TX_EMPTY_ENABLE	0x1
End of word count?	SPIm.MCSPI_IRQSTATUS[17] EOW	=0x1
Clear the interrupt.	SPIm.MCSPI_IRQENABLE[17] EOW_ENABLE	=0x0
Disable the interrupt for the receiver register.	SPIm.MCSPI_IRQENABLE[4] TX_EMPTY_ENABLE	0x0
Disable the interrupt for the transmit register.	SPIm.MCSPI_IRQENABLE[2] RX_FULL_ENABLE	0x0
Stop the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x0

**Table 23-288. Transmit and Receive Procedure With Word Count – Interrupt Method (continued)**

Step	Register/Bit Field/Programming Model	Value
Read from the receiver register.	SPIm.MCSPI_RXx	xxxx

**23.4.5.1.3.6 Transmit and Receive Procedure Without Word Count – Polling Method****Table 23-289. Transmit and Receive Procedure Without Word Count – Polling Method**

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See <a href="#">Table 23-279</a>	
Start the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x1
Wait until end of transfer?	SPIm.MCSPI_CHxSTAT[2] EOT SPIm.MCSPI_CHxSTAT[3] TXFFE	=0x1 =0x1
Stop the channel.	SPIm.MCSPI_CHxCTRL[0] EN	0x0
Read from the receiver register.	SPIm.MCSPI_RXx	xxxx

## 23.4.6 MCSPi Register Manual

### 23.4.6.1 MCSPi Instance Summary

**Table 23-290. MCSPi Instance Summary**

Module Name	Base Address	Size
MCSPi1	0x4809 8000	4KB
MCSPi2	0x4809 A000	4KB
MCSPi3	0x480B 8000	4KB
MCSPi4	0x480B A000	4KB

### 23.4.6.2 MCSPi Registers

#### 23.4.6.2.1 MCSPi Register Summary

Table 23-291 lists the MCSPi registers. Each register has a 32-bit width.

**Table 23-291. MCSPi Register Summary**

Register	Type	Offset Address	MCSPi1 L3 Physical Address	MCSPi2 L3 Physical Address	MCSPi3 L3 Physical Address	MCSPi4 L3 Physical Address
MCSPi_HL_REV	Rw	0x00	0x4809 8000	0x4809 A000	0x480B 8000	0x480B A000
MCSPi_HL_HWINFO	Rw	0x04	0x4809 8004	0x4809 A004	0x480B 8004	0x480B A004
MCSPi_HL_SYSCONFIG	Rw	0x10	0x4809 8010	0x4809 A010	0x480B 8010	0x480B A010
MCSPi_REVISION	R	0x100	0x4809 8100	0x4809 A100	0x480B 8100	0x480B A100
MCSPi_SYSCONFIG	RW	0x110	0x4809 8110	0x4809 A110	0x480B 8110	0x480B A110
MCSPi_SYSSTATUS	R	0x114	0x4809 8114	0x4809 A114	0x480B 8114	0x480B A114
MCSPi_IRQSTATUS	RW	0x118	0x4809 8118	0x4809 A118	0x480B 8118	0x480B A118
MCSPi_IRQENABLE	RW	0x1C	0x4809 811C	0x4809 A11C	0x480B 811C	0x480B A11C
MCSPi_WAKEUPENABLE	RW	0x120	0x4809 8120	0x4809 A120	0x480B 8120	0x480B A120
MCSPi_SYST	RW	0x124	0x4809 8124	0x4809 A124	0x480B 8124	0x480B A124
MCSPi_MODULCTRL	RW	0x128	0x4809 8128	0x4809 A128	0x480B 8128	0x480B A128
MCSPi_CHxCONF <sup>(1)</sup>	RW	0x12C + (0x14 * x)	0x4809 812C + (0x14 * x)	0x4809 A12C + (0x14 * x)	0x480B 812C + (0x14 * x)	0x480B A12C + (0x14 * x)
MCSPi_CHxSTAT <sup>(1)</sup>	R	0x130 + (0x14 * x)	0x4809 8130 + (0x14 * x)	0x4809 A130 + (0x14 * x)	0x480B 8130 + (0x14 * x)	0x480B A130 + (0x14 * x)
MCSPi_CHxCTRL <sup>(1)</sup>	RW	0x134 + (0x14 * x)	0x4809 8134 + (0x14 * x)	0x4809 A134 + (0x14 * x)	0x480B 8134 + (0x14 * x)	0x480B A134 + (0x14 * x)
MCSPi_TXx <sup>(1)</sup>	RW	0x138 + (0x14 * x)	0x4809 8138 + (0x14 * x)	0x4809 A138 + (0x14 * x)	0x480B 8138 + (0x14 * x)	0x480B A138 + (0x14 * x)
MCSPi_RXx <sup>(1)</sup>	R	0x13C + (0x14 * x)	0x4809 813C + (0x14 * x)	0x4809 A13C + (0x14 * x)	0x480B 813C + (0x14 * x)	0x480B A13C + (0x14 * x)
MCSPi_XFERLEVEL	RW	0x17C	0x4809 817C	0x4809 A17C	0x480B 817C	0x480B A17C

<sup>(1)</sup> x = 0 to 3 for MCSPi1  
x = 0 to 1 for MCSPi2  
x = 0 for MCSPi3  
x = 0 for MCSPi4

### 23.4.6.2.2 MCSPi Register Description

Table 23-292 through Table 23-324 describe the individual MCSPi register bits.

**Table 23-292. MCSPi\_HL\_REV**

<b>Address Offset</b>	0x00	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Physical Address</b>	See Table 23-291		
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

**Table 23-293. Register Call Summary for Register MCSPi\_HL\_REV**

Multichannel Serial Port Interface (MCSPi)

- [MCSPi Register Summary: \[0\]](#)

**Table 23-294. MCSPi\_HL\_HWINFO**

<b>Address Offset</b>	0x04	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Physical Address</b>	See Table 23-291		
<b>Description</b>	Information about the IP module's hardware configuration, that is, typically the module's HDL generics (if any). Actual field format and encoding is up to the module's designer to decide.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								RETMODE	FFNBYTE	USEFIFO					

Bits	Field Name	Description	Type	Reset
31:7	RSVD	Reserved These bits are initialized to zero, and writes to them are ignored.	R	0x0000000
6	RETMODE	Retention Mode generic parameter This bit field indicates whether the retention mode is supported using the pin PIRFFRET. 0x0: Retention mode disabled 0x1: Retention mode enabled	R	RETMODE



Bits	Field Name	Description	Type	Reset
5:1	FFNBYTE	FIFO number of byte generic parameter This register defines the value of FFNBYTE generic parameter, only MSB bits from 8 down to 4 are taken into account. Read 0x1: FIFO 16 bytes depth Read 0x2: FIFO 32 bytes depth Read 0x4: FIFO 64 bytes depth Read 0x8: FIFO 128 bytes depth Read 0x10: FIFO 256 bytes depth	R	0x04
0	USEFIFO	Use of a FIFO enable: This bit field indicates if a FIFO is integrated within controller design with its management. Read 0x0: FIFO not implemented in design Read 0x1: FIFO and its management implemented in design with depth defined by FFNBYTE generic.	R	0

**Table 23-295. Register Call Summary for Register MCSPi\_HL\_HWINFO**

- Multichannel Serial Port Interface (MCSPi)
- [MCSPi Register Summary: \[0\]](#)

**Table 23-296. MCSPi\_HL\_SYSCONFIG**

<b>Address Offset</b>	0x10	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Physical Address</b>	See <a href="#">Table 23-291</a>		
<b>Description</b>	Clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																IDLEMODE		FREEEMU		SOFTRESET											

Bits	Field Name	Description	Type	Reset
31:4	RSVD		R	0x0000000
3:2	IDLEMODE	<p>Configuration of the local target state management mode.</p> <p>By definition, target can handle read/write transaction as long as it is out of IDLE state.</p> <p>0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. Backup mode, for debug only.</p> <p>0x1: No-idle mode: local target never enters idle state. Backup mode, for debug only.</p> <p>0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wake-up events.</p> <p>0x3: Smart-idle wake-up-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wake-up events when in idle state. Mode is only relevant if the appropriate IP module "swake-up" output(s) is (are) implemented.</p>	RW	0x2
1	FREEEMU	<p>Sensitivity to emulation (debug) suspend input signal.</p> <p>0x0: IP module is sensitive to emulation suspend</p> <p>0x1: IP module is not sensitive to emulation suspend</p>	RW	0
0	SOFTRESET	<p>Software reset. (Optional)</p> <p>Write 0x0: No action</p> <p>Read 0x0: Reset done, no pending action</p> <p>Read 0x1: Reset (software or other) ongoing</p> <p>Write 0x1: Initiate software reset</p>	RW	0

**Table 23-297. Register Call Summary for Register MCSPi\_HL\_SYSCONFIG**

Multichannel Serial Port Interface (MCSPi)

- [MCSPi Register Summary: \[0\]](#)

**Table 23-298. MCSPi\_REVISION**

<b>Address Offset</b>	0x100		
<b>Physical Address</b>	See <a href="#">Table 23-291</a>	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Description</b>	This register contains the revision number.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI Internal data

**Table 23-299. Register Call Summary for Register MCSPi\_REVISION**

Multichannel Serial Port Interface (MCSPi)

- [MCSPi Register Summary: \[0\]](#)

**Table 23-300. MCSPi\_SYSCONFIG**

<b>Address Offset</b>	0x110		
<b>Physical Address</b>	See <a href="#">Table 23-291</a>	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Description</b>	This register allows controlling various parameters of the OCP interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLOCKACTIVITY		RESERVED			SIDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE						

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reads returns 0	RW	0x000000
9:8	CLOCKACTIVITY	Clocks activity during wake-up mode period 0x0: OCP and Functional clocks may be switched off. 0x1: OCP clock is maintained. Functional clock may be switched-off. 0x2: Functional clock is maintained. OCP clock may be switched-off. 0x3: OCP and Functional clocks are maintained.	RW	0x0
7:5	RESERVED	Reads returns 0	RW	0x0
4:3	SIDLEMODE	Power management 0x0: If an idle request is detected, the McSPi acknowledges it unconditionally and goes in Inactive mode. Interrupt, DMA requests and wake-up lines are unconditionally de-asserted and the module wake-up capability is deactivated even if the bit <a href="#">MCSPi_SYSCONFIG[EnaWakeUp]</a> is set. 0x1: If an idle request is detected, the request is ignored and the module does not switch to wake-up mode, and keeps on behaving normally. 0x2: If an idle request is detected, the module will switch to wake-up mode based on its internal activity, and the wake-up capability can be used if the bit <a href="#">MCSPi_SYSCONFIG[EnaWakeUp]</a> is set. 0x3: Reserved - do not use.	RW	0x2
2	ENAWAKEUP	Wake-up feature control 0x0: Wake-up capability is disabled 0x1: Wake-up capability is enabled	RW	1
1	SOFTRESET	Software reset. During reads it always returns 0. 0x0: (write) Normal mode 0x1: (write) Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware.	RW	0
0	AUTOIDLE	Internal OCP Clock gating strategy 0x0: OCP clock is free-running 0x1: Automatic OCP clock gating strategy is applied, based on the OCP interface activity	RW	1

**Table 23-301. Register Call Summary for Register MCSPi\_SYSCONFIG**

Multichannel Serial Port Interface (MCSPi)

- [Normal Mode: \[0\]](#)
- [Idle Mode: \[1\]](#)
- [Wake-Up Event in Smart-Idle Mode: \[2\]](#)
- [Force-Idle Mode: \[3\] \[4\]](#)
- [MCSPi Global Initialization: \[5\]](#)
- [MCSPi Register Summary: \[6\]](#)
- [MCSPi Register Description: \[7\] \[8\]](#)

**Table 23-302. MCSPi\_SYSSTATUS**

<b>Address Offset</b>	0x114	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Physical Address</b>	See <a href="#">Table 23-291</a>		
<b>Description</b>	This register provides status information about the module excluding the interrupt status information		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESETDONE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved for module specific status information. Read returns 0	R	0x0000 0000
0	RESETDONE	Internal Reset Monitoring Read 0x0: Internal module reset is on-going Read 0x1: Reset completed	R	0

**Table 23-303. Register Call Summary for Register MCSPi\_SYSSTATUS**

Multichannel Serial Port Interface (MCSPi)

- [MCSPi Global Initialization: \[0\]](#)
- [MCSPi Register Summary: \[1\]](#)

**Table 23-304. MCSPi\_IRQSTATUS**

<b>Address Offset</b>	0x118	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Physical Address</b>	See <a href="#">Table 23-291</a>		
<b>Description</b>	The interrupt status regroups all the status of the module internal events that can generate an interrupt		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																EOW	WKS	RESERVED	RX3_FULL	TX3_UNDERFLOW	TX3_EMPTY	RESERVED	RX2_FULL	TX2_UNDERFLOW	TX2_EMPTY	RESERVED	RX1_FULL	TX1_UNDERFLOW	TX1_EMPTY	RX0_OVERFLOW	RX0_FULL	TX0_UNDERFLOW	TX0_EMPTY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reads returns 0	RW	0x0000
17	EOW	End of word count event when a channel is enabled using the FIFO buffer and the channel had sent the number of SPI word defined by <a href="#">MCSPI_XFERLEVEL[WCNT]</a> . Write 0x0: w:Event status bit unchanged Read 0x0: r: Event false Read 0x1: r: Event is pending Write 0x1: w:Event status bit is reset	RW W1toClr	0
16	WKS	Wake Up event in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CH0CONF[SPIENSLV] Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
15	RESERVED	Reads returns 0	RW	0
14	RX3_FULL	Receiver register is full or almost full. Only when Channel 3 is enabled Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
13	TX3_UNDERFLOW	Transmitter register underflow. Only when Channel 3 is enabled. The transmitter register is empty (not updated by Host or DMA with new data) before its time slot assignment. Exception: No TX_underflow event when no data has been loaded into the transmitter register since channel has been enabled. Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
12	TX3_EMPTY	Transmitter register is empty or almost empty. Note: Enabling the channel automatically rises this event. Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
11	RESERVED	Reads returns 0	RW	0

## Multichannel Serial Port Interface (MCSPi)

www.ti.com

Bits	Field Name	Description	Type	Reset
10	RX2_FULL	Receiver register full or almost full. Channel 2 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
9	TX2_UNDERFLOW	Transmitter register underflow. Channel 2 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
8	TX2_EMPTY	Transmitter register empty or almost empty. Channel 2 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
7	RESERVED	Reads returns 0	RW	0
6	RX1_FULL	Receiver register full or almost full. Channel 1 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
5	TX1_UNDERFLOW	Transmitter register underflow. Channel 1 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
4	TX1_EMPTY	Transmitter register empty or almost empty. Channel 1 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
3	RX0_OVERFLOW	Receiver register overflow (slave mode only). Channel 0 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
2	RX0_FULL	Receiver register full or almost full. Channel 0 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0
1	TX0_UNDERFLOW	Transmitter register underflow. Channel 0 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
0	TX0_EMPTY	Transmitter register empty or almost empty. Channel 0 Read 0x0: r: Event false Write 0x0: w:Event status bit unchanged Write 0x1: w:Event status bit is reset Read 0x1: r: Event is pending	RW W1toClr	0

**Table 23-305. Register Call Summary for Register MCSPi\_IRQSTATUS**

Multichannel Serial Port Interface (MCSPi)

- [Master Transmit-and-Receive Mode \(Full Duplex\): \[0\]](#)
- [Master Transmit-Only Mode \(Half Duplex\): \[1\]](#)
- [Master Receive-Only Mode \(Half Duplex\): \[2\] \[3\] \[4\] \[5\]](#)
- [Force mcs pim\\_c sx Mode: \[6\]](#)
- [Turbo Mode: \[7\]](#)
- [Buffer Almost Full: \[8\] \[9\]](#)
- [Buffer Almost Empty: \[10\] \[11\]](#)
- [End of Transfer Management: \[12\]](#)
- [Interrupts: \[13\]](#)
- [TXx\\_EMPTY: \[14\] \[15\]](#)
- [TXx\\_UNDERFLOW: \[16\]](#)
- [RXx\\_FULL: \[17\]](#)
- [End Of Word Count: \[18\] \[19\]](#)
- [Interrupt Events in Slave Mode: \[20\]](#)
- [TXx\\_EMPTY: \[21\]](#)
- [TXx\\_UNDERFLOW: \[22\]](#)
- [RXx\\_FULL: \[23\]](#)
- [RX0\\_OVERFLOW: \[24\]](#)
- [End Of Word Count: \[25\] \[26\]](#)
- [Interrupt-Driven Operation: \[27\] \[28\]](#)
- [Polling: \[29\] \[30\]](#)
- [Transitions From Smart-Idle Mode to Normal Mode: \[31\] \[32\]](#)
- [Force-Idle Mode: \[33\]](#)
- [MCSPi Operational Modes: \[34\] \[35\] \[36\] \[37\]](#)
- [Receive-Only Procedure With Word Count- Polling Method: \[38\]](#)
- [Transmit-Only Procedure With and Without Word Count- Polling Method: \[39\]](#)
- [Transmit-Only Procedure With and Without Word Count- Interrupt Method: \[40\]](#)
- [Transmit and Receive Procedure with Word Count- Polling Method: \[41\]](#)
- [Transmit and Receive Procedure with Word Count - Interrupt Method: \[42\]](#)
- [MCSPi Register Summary: \[43\]](#)
- [MCSPi Register Description: \[44\] \[45\] \[46\]](#)

**Table 23-306. MCSPi\_IRQENABLE**

<b>Address Offset</b>	0x11C		
<b>Physical Address</b>	See <a href="#">Table 23-291</a>	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Description</b>	This register allows to enable/disable the module internal sources of interrupt, on an event-by-event basis.		
<b>Type</b>	RW		

## Multichannel Serial Port Interface (MCSPi)

www.ti.com

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																EOW_ENABLE	WKE	RESERVED	RX3_FULL_ENABLE	TX3_UNDERFLOW_ENABLE	TX3_EMPTY_ENABLE	RESERVED	RX2_FULL_ENABLE	TX2_UNDERFLOW_ENABLE	TX2_EMPTY_ENABLE	RESERVED	RX1_FULL_ENABLE	TX1_UNDERFLOW_ENABLE	TX1_EMPTY_ENABLE	RX0_OVERFLOW_ENABLE	RX0_FULL_ENABLE	TX0_UNDERFLOW_ENABLE	TX0_EMPTY_ENABLE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reads return 0	RW	0x0000
17	EOW_ENABLE	End of Word count Interrupt Enable. 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
16	WKE	Wake Up event interrupt Enable in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPi_CHOCONF[SPIENSLV] 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
15	RESERVED	Reads returns 0	RW	0
14	RX3_FULL_ENABLE	Receiver register Full Interrupt Enable. Ch 3 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
13	TX3_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Ch 3 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
12	TX3_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Ch3 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
11	RESERVED	Reads return 0	RW	0
10	RX2_FULL_ENABLE	Receiver register Full Interrupt Enable. Ch 2 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
9	TX2_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Ch 2 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
8	TX2_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Ch 2 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
7	RESERVED	Reads return 0	RW	0
6	RX1_FULL_ENABLE	Receiver register Full Interrupt Enable. Ch 1 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
5	TX1_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Ch 1 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
4	TX1_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Ch 1 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0



Bits	Field Name	Description	Type	Reset
3	RX0_OVERFLOW_ENABLE	Receiver register Overflow Interrupt Enable. Ch 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
2	RX0_FULL_ENABLE	Receiver register Full Interrupt Enable. Ch 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
1	TX0_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Ch 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
0	TX0_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Ch 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0

**Table 23-307. Register Call Summary for Register MCSPI\_IRQENABLE**

Multichannel Serial Port Interface (MCSPi)

- [Interrupts: \[0\]](#)
- [Interrupt-Driven Operation: \[1\]](#)
- [Polling: \[2\]](#)
- [Transitions From Smart-Idle Mode to Normal Mode: \[3\] \[4\]](#)
- [MCSPi Operational Modes: \[5\] \[6\]](#)
- [Transmit-Only Procedure With and Without Word Count- Interrupt Method: \[7\] \[8\] \[9\]](#)
- [Transmit and Receive Procedure with Word Count - Interrupt Method: \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [MCSPi Register Summary: \[15\]](#)

**Table 23-308. MCSPI\_WAKEUPENABLE**

<b>Address Offset</b>	0x120	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Physical Address</b>	See <a href="#">Table 23-291</a>		
<b>Description</b>	The wake-up enable register allows to enable/disable the module internal sources of wake-up on event-by-event basis.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKEN															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	RW	0x0000 0000
0	WKEN	Wake-up functionality in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CHOCONF[SPIENSLV]  0x0: The event is not allowed to wake-up the system, even if the global control bit MCSPI_SYSCONF[EnaWakeUp] is set.  0x1: The event is allowed to wake-up the system if the global control bit MCSPI_SYSCONF[EnaWakeUp] is set.	RW	0

**Table 23-309. Register Call Summary for Register MCSPi\_WAKEUPENABLE**

Multichannel Serial Port Interface (MCSPi)

- [Wake-Up Event in Smart-Idle Mode: \[0\]](#)
- [Transitions From Smart-Idle Mode to Normal Mode: \[1\]](#)
- [Force-Idle Mode: \[2\]](#)
- [MCSPi Register Summary: \[3\]](#)

**Table 23-310. MCSPi\_SYST**

<b>Address Offset</b>	0x124	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Physical Address</b>	See <a href="#">Table 23-291</a>		
<b>Description</b>	This register is used to check the correctness of the system interconnect either internally to peripheral bus, or externally to device IO pads, when the module is configured in system test (SYSTEST) mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SSB	SPIENDIR	SPIDATDIR1	SPIDATDIR0	WAKD	SPICLK	SPIDAT_1	SPIDAT_0	SPIEN_3	SPIEN_2	SPIEN_1	SPIEN_0				

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reads returns 0	RW	0x00000
11	SSB	Set status bit  0x0: No action. Writing 0 does not clear already set status bits; This bit must be cleared prior attempting to clear a status bit of the <MCSPi_IRQSTATUS> register.  0x1: Force to 1 all status bits of MCSPi_IRQSTATUS register. Writing 1 into this bit sets to 1 all status bits contained in the <MCSPi_IRQSTATUS> register.	RW	0
10	SPIENDIR	Set the direction of the SPIEN[3:0] lines and SPICLK line 0x0: output (as in master mode) 0x1: input (as in slave mode)	RW	0
9	SPIDATDIR1	Set the direction of the SPIDAT[1] 0x0: output 0x1: input	RW	0
8	SPIDATDIR0	Set the direction of the SPIDAT[0] 0x0: output 0x1: input	RW	0
7	WAKD	SWAKEUP output (signal data value of internal signal to system). The signal is driven high or low according to the value written into this register bit. 0x0: The pin is driven low. 0x1: The pin is driven high.	RW	0
6	SPICLK	SPICLK line (signal data value) If MCSPi_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the CLKSPi line (high or low), and a write into this bit has no effect. If MCSPi_SYST[SPIENDIR] = 0 (output mode direction), the CLKSPi line is driven high or low according to the value written into this register.	RW	0

Bits	Field Name	Description	Type	Reset
5	SPIDAT_1	SPIDAT[1] line (signal data value) If <a href="#">MCSPi_SYST[SPIDATDIR1]</a> = 0 (output mode direction), the SPIDAT[1] line is driven high or low according to the value written into this register. If <a href="#">MCSPi_SYST[SPIDATDIR1]</a> = 1 (input mode direction), this bit returns the value on the SPIDAT[1] line (high or low), and a write into this bit has no effect.	RW	0
4	SPIDAT_0	SPIDAT[0] line (signal data value) If <a href="#">MCSPi_SYST[SPIDATDIR0]</a> = 0 (output mode direction), the SPIDAT[0] line is driven high or low according to the value written into this register. If <a href="#">MCSPi_SYST[SPIDATDIR0]</a> = 1 (input mode direction), this bit returns the value on the SPIDAT[0] line (high or low), and a write into this bit has no effect.	RW	0
3	SPIEN_3	SPIEN[3] line (signal data value) If <a href="#">MCSPi_SYST[SPIENDIR]</a> = 0 (output mode direction), the SPIEN[3] line is driven high or low according to the value written into this register. If <a href="#">MCSPi_SYST[SPIENDIR]</a> = 1 (input mode direction), this bit returns the value on the SPIEN[3] line (high or low), and a write into this bit has no effect.	RW	0
2	SPIEN_2	SPIEN[2] line (signal data value) If <a href="#">MCSPi_SYST[SPIENDIR]</a> = 0 (output mode direction), the SPIEN[2] line is driven high or low according to the value written into this register. If <a href="#">MCSPi_SYST[SPIENDIR]</a> = 1 (input mode direction), this bit returns the value on the SPIEN[2] line (high or low), and a write into this bit has no effect.	RW	0
1	SPIEN_1	SPIEN[1] line (signal data value) If <a href="#">MCSPi_SYST[SPIENDIR]</a> = 0 (output mode direction), the SPIEN[1] line is driven high or low according to the value written into this register. If <a href="#">MCSPi_SYST[SPIENDIR]</a> = 1 (input mode direction), this bit returns the value on the SPIEN[1] line (high or low), and a write into this bit has no effect.	RW	0
0	SPIEN_0	SPIEN[0] line (signal data value) If <a href="#">MCSPi_SYST[SPIENDIR]</a> = 0 (output mode direction), the SPIEN[0] line is driven high or low according to the value written into this register. If <a href="#">MCSPi_SYST[SPIENDIR]</a> = 1 (input mode direction), this bit returns the value on the SPIEN[0] line (high or low), and a write into this bit has no effect.	RW	0

**Table 23-311. Register Call Summary for Register MCSPi\_SYST**

Multichannel Serial Port Interface (MCSPi)

- [MCSPi Register Summary: \[0\]](#)
- [MCSPi Register Description: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

**Table 23-312. MCSPi\_MODULCTRL**

<b>Address Offset</b>	0x128		
<b>Physical Address</b>	See <a href="#">Table 23-291</a>	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Description</b>	This register is dedicated to the configuration of the serial port interface.		
<b>Type</b>	RW		

## Multichannel Serial Port Interface (MCSPi)

www.ti.com

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FDAA	MOA	INITDLY				SYSTEM_TEST	MS	PIN34	SINGLE						

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reads returns 0	RW	0x000000
8	FDAA	<p>FIFO DMA Address 256-bit aligned</p> <p>This register is used when a FIFO is managed by the module and DMA connected to the controller provides only 256 bit aligned address. If this bit is set the enabled channel which uses the FIFO has its data managed through MCSPI_DAFTX and MCSPI_DAFRX registers instead of MCSPI_TX(i) and MCSPI_RX(i) registers.</p> <p>0x0: FIFO data managed by MCSPI_TX(i) and MCSPI_RX(i) registers.</p> <p>0x1: FIFO data managed by MCSPI_DAFTX and MCSPI_DAFRX registers.</p>	RW	0
7	MOA	<p>Multiple word ocp access:</p> <p>This register can only be used when a channel is enabled using a FIFO. It allows the system to perform multiple SPI word access for a single 32-bit OCP word access. This is possible for WL &lt; 16.</p> <p>0x0: Multiple word access disabled</p> <p>0x1: Multiple word access enabled with FIFO</p>	RW	0
6:4	INITDLY	<p>Initial spi delay for first transfer:</p> <p>This register is an option only available in SINGLE master mode, The controller waits for a delay to transmit the first spi word after channel enabled and corresponding TX register filled. This Delay is based on SPI output frequency clock, No clock output provided to the boundary and chip select is not active in 4 pin mode within this period.</p> <p>0x0: No delay for first spi transfer.</p> <p>0x1: The controller wait 4 spi bus clock</p> <p>0x2: The controller wait 8 spi bus clock</p> <p>0x3: The controller wait 16 spi bus clock</p> <p>0x4: The controller wait 32 spi bus clock</p>	RW	0x0
3	SYSTEM_TEST	<p>Enables the system test mode</p> <p>0x0: Functional mode</p> <p>0x1: System test mode (SYSTEST)</p>	RW	0
2	MS	<p>Master/Slave</p> <p>0x0: Master - The module generates the SPICLK and SPIEN[3:0]</p> <p>0x1: Slave - The module receives the SPICLK and SPIEN[3:0]</p>	RW	1
1	PIN34	<p>Pin mode selection:</p> <p>This register is used to configure the SPI pin mode, in master or slave mode. If asserted the controller only use SIMO,SOMI and SPICLK clock pin for spi transfers.</p> <p>0x0: SPIEN is used as a chip-select.</p> <p>0x1: SPIEN is not used. In this mode all related option to chip-select have no meaning</p>	RW	0

Bits	Field Name	Description	Type	Reset
0	SINGLE	Single channel/Multi Channel (master mode only) 0x0: More than one channel will be used in master mode. 0x1: Only one channel will be used in master mode. This bit must be set in Force SPIEN mode.	RW	0

**Table 23-313. Register Call Summary for Register MCSPI\_MODULCTRL**

Multichannel Serial Port Interface (MCSPi)

- [Single-Channel Master Mode: \[0\] \[1\]](#)
- [Force mcs pim\\_csx Mode: \[2\] \[3\]](#)
- [Chip-Select Timing Control: \[4\]](#)
- [Slave Mode: \[5\]](#)
- [MCSPi Operational Modes: \[6\] \[7\] \[8\]](#)
- [MCSPi Register Summary: \[9\]](#)
- [MCSPi Register Description: \[10\] \[11\]](#)

**Table 23-314. MCSPI\_CHxCONF**

<b>Address Offset</b>	0x12C + (0x14 * x)	<b>Index</b>	x= 0 to 3 for MCSPi1. x= 0 to 1 for MCSPi2 . x= 0 for MCSPi4 and MCSPi3.
<b>Physical Address</b>	See <a href="#">Table 23-291</a>	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Description</b>	This register is dedicated to the configuration of the channel 0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	CLKG	FFER	FFEW	TCS0	SBPOL	SBE	SPIENSLV	FORCE	TURBO	IS	DPE1	DPE0	DMAR	DMAW	TRM									EPOL			CLKD		POL	PHA

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	read returns 0	R	0x0
29	CLKG	Clock divider granularity This register defines the granularity of channel clock divider: power of two or one clock cycle granularity. When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio. Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values 0x0: Clock granularity of power of two 0x1: One clock cycle granularity	RW	0
28	FFER	FIFO enabled for receive: Only one channel can have this bit field set. 0x0: The buffer is not used to receive data. 0x1: The buffer is used to receive data.	RW	0
27	FFEW	FIFO enabled for Transmit: Only one channel can have this bit field set. 0x0: The buffer is not used to transmit data. 0x1: The buffer is used to transmit data.	RW	0

## Multichannel Serial Port Interface (MCSPi)

www.ti.com

Bits	Field Name	Description	Type	Reset
26:25	TCS0	<p>Chip Select Time Control This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock.</p> <p>0x0: 0.5 clock cycle 0x1: 1.5 clock cycle 0x2: 2.5 clock cycle 0x3: 3.5 clock cycle</p>	RW	0x0
24	SBPOL	<p>Start bit polarity 0x0: Start bit polarity is held to 0 during SPI transfer. 0x1: Start bit polarity is held to 1 during SPI transfer.</p>	RW	0
23	SBE	<p>Start bit enable for SPI transfer 0x0: Default SPI transfer length as specified by WL bit field 0x1: Start bit D/CX added before SPI transfer polarity is defined by MCSPi_CH0CONF[SBPOL]</p>	RW	0
22:21	SPIENSLV	<p>Channel 0 only and slave mode only: SPI slave select signal detection. Reserved bits for other cases.</p> <p>0x0: Detection enabled only on SPIEN[0] 0x1: Detection enabled only on SPIEN[1] 0x2: Detection enabled only on SPIEN[2] 0x3: Detection enabled only on SPIEN[3]</p>	RW	0x0
20	FORCE	<p>Manual SPIEN assertion to keep SPIEN active between SPI words. (single channel master mode only)</p> <p>0x0: Writing 0 into this bit drives low the SPIEN line when MCSPi_CHCONF(i)[EPOL]=0, and drives it high when MCSPi_CHCONF(i)[EPOL]=1. 0x1: Writing 1 into this bit drives high the SPIEN line when MCSPi_CHCONF(i)[EPOL]=0, and drives it low when MCSPi_CHCONF(i)[EPOL]=1</p>	RW	0
19	TURBO	<p>Turbo mode 0x0: Turbo is deactivated (recommended for single SPI word transfer) 0x1: Turbo is activated to maximize the throughput for multi SPI words transfer.</p>	RW	0
18	IS	<p>Input Select 0x0: Data Line0 (SPIDAT[0]) selected for reception. 0x1: Data Line1 (SPIDAT[1]) selected for reception</p>	RW	1
17	DPE1	<p>Transmission Enable for data line 1 (SPIDATAGZEN[1]) 0x0: Data Line1 (SPIDAT[1]) selected for transmission 0x1: No transmission on Data Line1 (SPIDAT[1])</p>	RW	1
16	DPE0	<p>Transmission Enable for data line 0 (SPIDATAGZEN[0]) 0x0: Data Line0 (SPIDAT[0]) selected for transmission 0x1: No transmission on Data Line0 (SPIDAT[0])</p>	RW	0
15	DMAR	<p>DMA Read request The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. The DMA Read request line is deasserted on read completion of the receive register of the channel.</p> <p>0x0: DMA Read Request disabled 0x1: DMA Read Request enabled</p>	RW	0

<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>
14	DMAW	DMA Write request. The DMA Write request line is asserted when The channel is enabled and the transmitter register of the channel is empty. The DMA Write request line is deasserted on load completion of the transmitter register of the channel. 0x0: DMA Write Request disabled 0x1: DMA Write Request enabled	RW	0
13:12	TRM	Transmit/Receive modes 0x0: Transmit and Receive mode 0x1: Receive only mode 0x2: Transmit only mode 0x3: Reserved	RW	0x0
11:7	WL	SPI word length 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: The SPI word is 4-bits long 0x4: The SPI word is 5-bits long 0x5: The SPI word is 6-bits long 0x6: The SPI word is 7-bits long 0x7: The SPI word is 8-bits long 0x8: The SPI word is 9-bits long 0x9: The SPI word is 10-bits long 0xA: The SPI word is 11-bits long 0xB: The SPI word is 12-bits long 0xC: The SPI word is 13-bits long 0xD: The SPI word is 14-bits long 0xE: The SPI word is 15-bits long 0xF: The SPI word is 16-bits long 0x10: The SPI word is 17-bits long 0x11: The SPI word is 18-bits long 0x12: The SPI word is 19-bits long 0x13: The SPI word is 20-bits long 0x14: The SPI word is 21-bits long 0x15: The SPI word is 22-bits long 0x16: The SPI word is 23-bits long 0x17: The SPI word is 24-bits long 0x18: The SPI word is 25-bits long 0x19: The SPI word is 26-bits long 0x1A: The SPI word is 27-bits long 0x1B: The SPI word is 28-bits long 0x1C: The SPI word is 29-bits long 0x1D: The SPI word is 30-bits long 0x1E: The SPI word is 31-bits long 0x1F: The SPI word is 32-bits long	RW	0x00
6	EPOL	SPIEN polarity 0x0: SPIEN is held high during the active state. 0x1: SPIEN is held low during the active state.	RW	0

## Multichannel Serial Port Interface (MCSPI)

www.ti.com

Bits	Field Name	Description	Type	Reset
5:2	CLKD	<p>Frequency divider for SPICLK. (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (CLKSPIREF) with a 4-bit value, and results in a new clock SPICLK available to shift-in and shift-out data. By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared, Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] register. The value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is set to 0.</p> <p>0x0: 1            0x1: 2            0x2: 4            0x3: 8            0x4: 16            0x5: 32            0x6: 64            0x7: 128            0x8: 256            0x9: 512            0xA: 1024            0xB: 2048            0xC: 4096            0xD: 8192            0xE: 16384            0xF: 32768</p>	RW	0x0
1	POL	<p>SPICLK polarity</p> <p>0x0: SPICLK is held high during the active state            0x1: SPICLK is held low during the active state</p>	RW	0
0	PHA	<p>SPICLK phase</p> <p>0x0: Data are latched on odd numbered edges of SPICLK.            0x1: Data are latched on even numbered edges of SPICLK.</p>	RW	0



**Table 23-315. Register Call Summary for Register MCSPI\_CHxCONF**

Multichannel Serial Port Interface (MCSPi)

- Multichannel SPI Protocol and Data Format: [0] [1] [2] [3] [4] [5] [6] [7]
- Transfer Format: [8]
- SPI Interface in Slave Mode: [9]
- Master Transmit-and-Receive Mode (Full Duplex): [10]
- Master Transmit-Only Mode (Half Duplex): [11]
- Master Receive-Only Mode (Half Duplex): [12]
- Single-Channel Master Mode: [13]
- Force mcsxim\_csx Mode: [14] [15] [16] [17] [18] [19] [20] [21] [22] [23]
- Turbo Mode: [24]
- Start Bit Mode: [25] [26] [27] [28]
- Chip-Select Timing Control: [29]
- Programmable SPI Clock (mcsxim\_clk): [30]
- Clock Ratio Granularity: [31] [32] [33] [34] [35]
- Dedicated Resources: [36] [37] [38] [39] [40] [41]
- Slave Transmit-and-Receive Mode: [42]
- Slave Transmit-Only Mode: [43]
- Slave Receive-Only Mode: [44]
- FIFO Buffer Management: [45] [46] [47]
- Buffer Almost Full: [48] [49]
- Buffer Almost Empty: [50] [51]
- TXx\_EMPTY: [52]
- RXx\_FULL: [53]
- TXx\_EMPTY: [54]
- RXx\_FULL: [55]
- DMA Requests: [56] [57]
- Wake-Up Event in Smart-Idle Mode: [58]
- MCSPi Operational Modes: [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74]
- MCSPi Register Summary: [75]

**Table 23-316. MCSPI\_CHxSTAT**

<b>Address Offset</b>	0x130 + (0x14 * x)	<b>Index</b>	x= 0 to 3 for MCSPI1. x= 0 to 1 for MCSPI2 . x= 0 for MCSPI4 and MCSPI3.
<b>Physical Address</b>	See Table 23-291	<b>Instance</b>	MCSPI1 MCSPI2 MCSPI3 MCSPI4
<b>Description</b>	This register provides status information about transmitter and receiver registers of channel 0		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RXFFF	RXFFE	TXFFF	TXFFE	EOT	TXS	RXS		

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Read returns 0	R	0x0000000
6	RXFFF	Channel "i" FIFO Receive Buffer Full Status Read 0x0: FIFO Receive Buffer is not full Read 0x1: FIFO Receive Buffer is full	R	0
5	RXFFE	Channel "i" FIFO Receive Buffer Empty Status Read 0x0: FIFO Receive Buffer is not empty Read 0x1: FIFO Receive Buffer is empty	R	0

Bits	Field Name	Description	Type	Reset
4	TXFFF	Channel "i" FIFO Transmit Buffer Full Status Read 0x0: FIFO Transmit Buffer is not full Read 0x1: FIFO Transmit Buffer is full	R	0
3	TXFFE	Channel "i" FIFO Transmit Buffer Empty Status Read 0x0: FIFO Transmit Buffer is not empty Read 0x1: FIFO Transmit Buffer is empty	R	0
2	EOT	Channel "i" End of transfer Status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (Transmit/Receive modes, Turbo mode). See dedicated chapters for details.  Read 0x0: This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer).  Read 0x1: This flag is automatically set to one at the end of an SPI transfer.	R	0
1	TXS	Channel "i" Transmitter Register Status Read 0x0: Register is full Read 0x1: Register is empty	R	0
0	RXS	Channel "i" Receiver Register Status Read 0x0: Register is empty Read 0x1: Register is full	R	0

**Table 23-317. Register Call Summary for Register MCSPi\_CHxSTAT**

## Multichannel Serial Port Interface (MCSPi)

- [Master Transmit-and-Receive Mode \(Full Duplex\): \[0\] \[1\] \[2\]](#)
- [Master Transmit-Only Mode \(Half Duplex\): \[3\]](#)
- [Master Receive-Only Mode \(Half Duplex\): \[4\]](#)
- [Programming Tips When Switching to Another Channel: \[5\]](#)
- [Force mcs pim\\_csx Mode: \[6\] \[7\] \[8\]](#)
- [Turbo Mode: \[9\]](#)
- [Dedicated Resources: \[10\] \[11\]](#)
- [Slave Transmit-and-Receive Mode: \[12\]](#)
- [Slave Transmit-Only Mode: \[13\]](#)
- [Slave Receive-Only Mode: \[14\]](#)
- [End of Transfer Management: \[15\]](#)
- [MCSPi Operational Modes: \[16\] \[17\]](#)
- [Transmit-Only Procedure With and Without Word Count- Polling Method: \[18\] \[19\]](#)
- [Transmit-Only Procedure With and Without Word Count- Interrupt Method: \[20\] \[21\]](#)
- [Transmit and Receive Procedure without Word Count- Polling Method: \[22\] \[23\]](#)
- [MCSPi Register Summary: \[24\]](#)

**Table 23-318. MCSPi\_CHxCTRL**

<b>Address Offset</b>	0x134 + (0x14 * x)	<b>Index</b>	x= 0 to 3 for MCSPi1. x= 0 to 1 for MCSPi2 . x= 0 for MCSPi4 and MCSPi3.
<b>Physical Address</b>	See <a href="#">Table 23-291</a>	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Description</b>	This register is dedicated to enable the channel 0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																EXTCLK								RESERVED								Z

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Read returns 0	RW	0x0000
15:8	EXTCLK	Clock ratio extension: This register is used to concatenate with MCSPi_CHCONF[CLKD] register for clock ratio only when granularity is one clock cycle (MCSPi_CHCONF[CLKG] set to 1). Then the max value reached is 4096 clock divider ratio.  0x0: Clock ratio is CLKD + 1 0x1: Clock ratio is CLKD + 1 + 16 0xFF: Clock ratio is CLKD + 1 + 4080	RW	0x00
7:1	RESERVED	Read returns 0	RW	0x00
0	EN	Channel Enable  0x0: Channel "i" is not active 0x1: Channel "i" is active	RW	0

**Table 23-319. Register Call Summary for Register MCSPi\_CHxCTRL**

Multichannel Serial Port Interface (MCSPi)

- [Master Transmit-and-Receive Mode \(Full Duplex\): \[0\]](#)
- [Force mcspim\\_csx Mode: \[1\] \[2\]](#)
- [Clock Ratio Granularity: \[3\]](#)
- [Dedicated Resources: \[4\] \[5\]](#)
- [MCSPi Operational Modes: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\]](#)
- [Receive-Only Procedure With Word Count- Polling Method: \[17\] \[18\]](#)
- [Transmit-Only Procedure With and Without Word Count- Polling Method: \[19\] \[20\]](#)
- [Transmit-Only Procedure With and Without Word Count- Interrupt Method: \[21\] \[22\]](#)
- [Transmit and Receive Procedure with Word Count- Polling Method: \[23\] \[24\]](#)
- [Transmit and Receive Procedure with Word Count - Interrupt Method: \[25\] \[26\]](#)
- [Transmit and Receive Procedure without Word Count- Polling Method: \[27\] \[28\]](#)
- [MCSPi Register Summary: \[29\]](#)

**Table 23-320. MCSPi\_TXx**

<b>Address Offset</b>	0x138 + (0x14 * x)	<b>Index</b>	x= 0 to 3 for MCSPi1. x= 0 to 1 for MCSPi2 . x= 0 for MCSPi4 and MCSPi3.
<b>Physical Address</b>	See <a href="#">Table 23-291</a>	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Description</b>	This register contains a single SPI word to transmit on the serial link, what ever SPI word length is.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDATA																															

Bits	Field Name	Description	Type	Reset
31:0	TDATA	Channel 0 Data to transmit	RW	0x0000 0000

**Table 23-321. Register Call Summary for Register MCSPi\_TXx**

## Multichannel Serial Port Interface (MCSPi)

- [Master Transmit-and-Receive Mode \(Full Duplex\): \[0\] \[1\] \[2\] \[3\]](#)
- [Master Transmit-Only Mode \(Half Duplex\): \[4\]](#)
- [Master Receive-Only Mode \(Half Duplex\): \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [Force mcsxim\\_csx Mode: \[10\]](#)
- [Dedicated Resources: \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [Slave Transmit-and-Receive Mode: \[16\] \[17\]](#)
- [Slave Receive-Only Mode: \[18\] \[19\] \[20\]](#)
- [Interrupt Events in Master Mode: \[21\]](#)
- [TXx\\_EMPTY: \[22\] \[23\] \[24\]](#)
- [TXx\\_UNDERFLOW: \[25\] \[26\] \[27\]](#)
- [Interrupt Events in Slave Mode: \[28\]](#)
- [TXx\\_EMPTY: \[29\] \[30\] \[31\]](#)
- [TXx\\_UNDERFLOW: \[32\] \[33\]](#)
- [Interrupt-Driven Operation: \[34\]](#)
- [Polling: \[35\]](#)
- [DMA Requests: \[36\] \[37\]](#)
- [Common Transfer Procedures with FIFO- Polling Method: \[38\]](#)
- [MCSPi Register Summary: \[39\]](#)

**Table 23-322. MCSPi\_RXx**

<b>Address Offset</b>	0x13C + (0x14 * x)	<b>Index</b>	x= 0 to 3 for MCSPi1. x= 0 to 1 for MCSPi2 . x= 0 for MCSPi4 and MCSPi3.
<b>Physical Address</b>	See <a href="#">Table 23-291</a>	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Description</b>	This register contains a single SPI word received through the serial link, what ever SPI word length is.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA																															

Bits	Field Name	Description	Type	Reset
31:0	RDATA	Channel 0 Received Data	R	0x0000 0000

**Table 23-323. Register Call Summary for Register MCSPI\_RXx**

Multichannel Serial Port Interface (MCSPi)

- Master Transmit-and-Receive Mode (Full Duplex): [0] [1]
- Master Transmit-Only Mode (Half Duplex): [2] [3] [4]
- Master Receive-Only Mode (Half Duplex): [5] [6]
- Force mcs pim\_csx Mode: [7] [8]
- Turbo Mode: [9] [10]
- Dedicated Resources: [11] [12] [13]
- Slave Transmit-and-Receive Mode: [14]
- Slave Transmit-Only Mode: [15] [16]
- End of Transfer Management: [17]
- Interrupt Events in Master Mode: [18]
- RXx\_FULL: [19] [20] [21]
- Interrupt Events in Slave Mode: [22]
- RXx\_FULL: [23] [24] [25]
- RX0\_OVERFLOW: [26] [27]
- Interrupt-Driven Operation: [28]
- Polling: [29]
- DMA Requests: [30]
- Wake-Up Event in Smart-Idle Mode: [31]
- MCSPi Operational Modes: [32] [33]
- Receive-Only Procedure With Word Count- Polling Method: [34]
- Transmit and Receive Procedure with Word Count- Polling Method: [35]
- Transmit and Receive Procedure with Word Count - Interrupt Method: [36]
- Transmit and Receive Procedure without Word Count- Polling Method: [37]
- MCSPi Register Summary: [38]

**Table 23-324. MCSPI\_XFERLEVEL**

<b>Address Offset</b>	0x17C	<b>Instance</b>	MCSPi1 MCSPi2 MCSPi3 MCSPi4
<b>Physical Address</b>	See Table 23-291		
<b>Description</b>	This register provides transfer levels needed while using FIFO buffer during transfer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WCNT								AFL								AEL															

Bits	Field Name	Description	Type	Reset
31:16	WCNT	Spi word counter. This register holds the programmable value of number of SPI word to be transferred on channel which is using the FIFO buffer. When transfer had started, a read back in this register returns the current SPI word transfer index.  0x0: Counter not used 0x1: one word 0xFFFFE: 65534 spi word 0xFFFFF: 65535 spi word	RW	0x0000

Bits	Field Name	Description	Type	Reset
15:8	AFL	<p>Buffer Almost Full</p> <p>This register holds the programmable almost full level value used to determine almost full buffer condition. If the user wants an interrupt or a DMA read request to be issued during a receive operation when the data buffer holds at least n bytes, then the buffer <a href="#">MCSPi_MODULCTRL[AFL]</a> must be set with n-1. The size of this register is defined by the generic parameter FFBYTE.</p> <p>0x0: one byte 0x1: 2 bytes 0xFE: 255bytes 0xFF: 256bytes</p>	RW	0x00
7:0	AEL	<p>Buffer Almost Empty. This register holds the programmable almost empty level value used to determine almost empty buffer condition. If the user wants an interrupt or a DMA write request to be issued during a transmit operation when the data buffer is able to receive n bytes, then the buffer <a href="#">MCSPi_MODULCTRL[AEL]</a> must be set with - 1.</p> <p>0x0: one byte 0x1: 2 bytes 0xFE: 255 bytes 0xFF: 256bytes</p>	RW	0x00

**Table 23-325. Register Call Summary for Register MCSPi\_XFERLEVEL**

Multichannel Serial Port Interface (MCSPi)

- [FIFO Buffer Management: \[0\] \[1\]](#)
- [Buffer Almost Full: \[2\]](#)
- [Buffer Almost Empty: \[3\]](#)
- [End of Transfer Management: \[4\] \[5\]](#)
- [TXx\\_EMPTY: \[6\] \[7\]](#)
- [RXx\\_FULL: \[8\]](#)
- [End Of Word Count: \[9\] \[10\]](#)
- [TXx\\_EMPTY: \[11\] \[12\]](#)
- [RXx\\_FULL: \[13\]](#)
- [End Of Word Count: \[14\]](#)
- [MCSPi Register Summary: \[15\]](#)
- [MCSPi Register Description: \[16\]](#)

## 23.5 Multichannel Buffered Serial Port (McBSP)

This section introduces the Multi-Channel Buffered Serial Port.

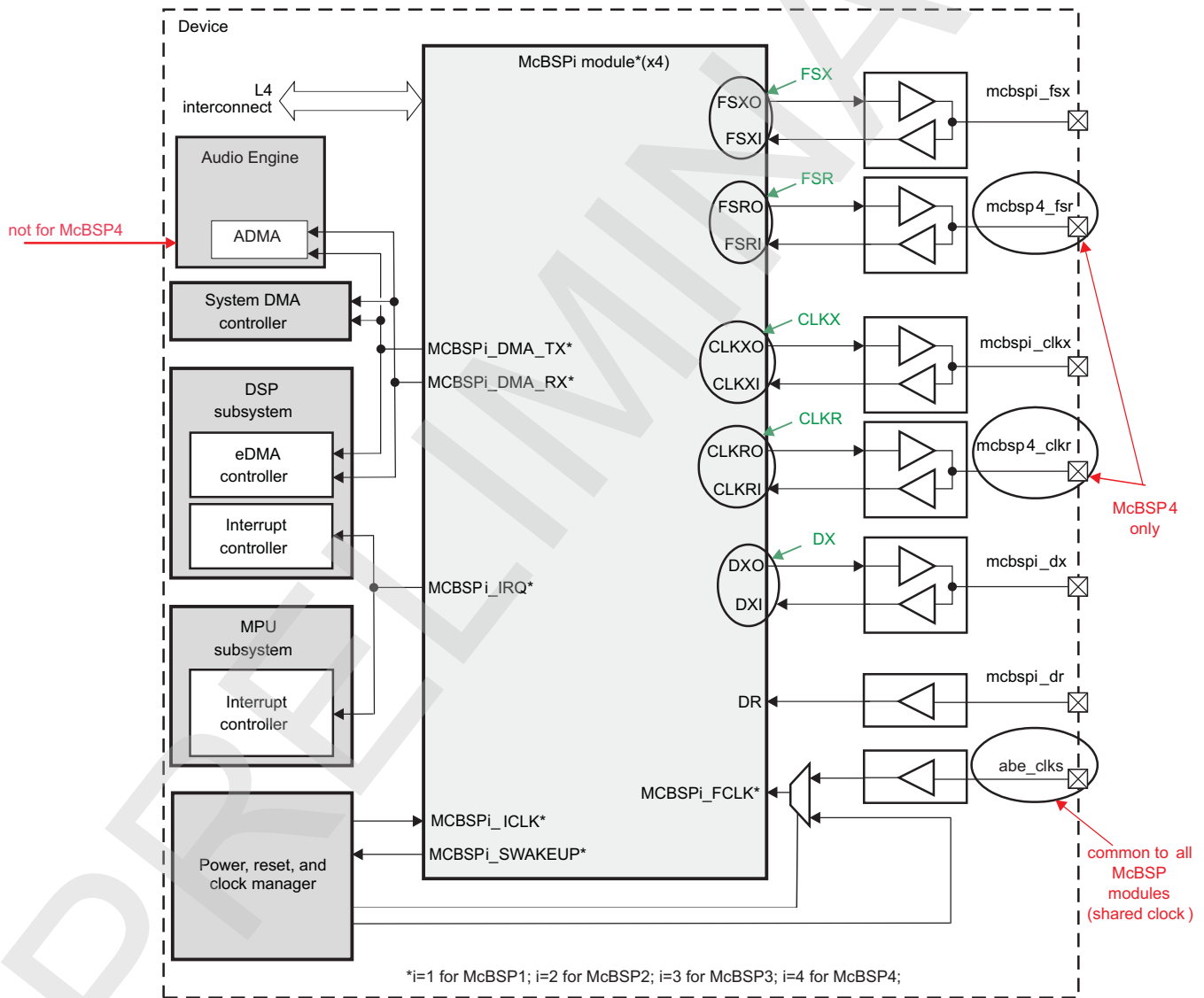
### 23.5.1 McBSP Overview

The multichannel buffered serial port (McBSP) provides a full-duplex direct serial interface between the device and other devices in a system such as other application chips (digital base band), audio and voice codec, etc. Because of its high level of versatility, it can accommodate a wide range of peripherals and clocked frame oriented protocols (for details, see [Section 23.5.1.1, McBSP Features](#)).

The device provides four instances of the McBSP module.

Figure 23-110 shows the McBSP overview in the device.

Figure 23-110. McBSP Highlight



mcbasp-001

#### 23.5.1.1 McBSP Features

The main features of the McBSP modules are:

- L4 interconnect slave interface supports:

- 32-bit data bus width
- 32-bit access supported
- 16-/8-bit access not supported
- 10-bit address bus width
- Burst mode not supported
- Write nonposted transaction mode supported
- 128 × 32-bit words (512 bytes) for each buffer for transmit/receive operations
- Transmit and receive DMA requests triggered with programmable FIFO thresholds
- Serial interface description
  - 6 pin configuration (McBSP 4 only)
  - 4 pin configuration (McBSP1,2, 3)
  - Full-duplex communication
  - Multichannel selection modes
    - Support to enable or block transfers in each of the channels
    - 128 channels for transmission and for reception
  - Direct interface to industry-standard codecs, analog interface chips (AICs), and other serially connected A/D and D/A devices:
    - Inter-IC sound (I2S) compliant devices
    - Pulse code modulation (PCM) devices
    - Time division multiplexed (TDM) bus devices

**CAUTION**

McBSP modules do not offer support for  $\mu$ -law and A-law companding, two partitions mode dynamic reassignment, AC'97, and SPI protocol.

- A wide selection of data sizes: 8, 12, 16, 20, 24, and 32 bits
- Bit reordering (send/receive least significant bit [LSB])
- Clock and frame-synchronization generation support:
  - Independent clocking and framing for reception and for transmission up to 48 MHz
  - Support for external generation of clock signals and frame-synchronization (frame-sync) signals
  - A programmable sample rate generator for internal generation and control of clock signals and frame-sync signals
  - Programmable polarity for frame-sync pulses and for clock signals

**NOTE:**

- McBSP modules do not support features such as retransmit or re-receive of an erroneous frame or word.
- McBSP modules support dual phase frames to provide I2S fully compliant capabilities. But, this dual phase mode is limited at one channel (or word) for each phase instead of 128 channels max for single phase mode.



### 23.5.2 McBSP Environment

This section describes the intended functions for McBSP module from an environment point of view (that is, external connections). It presents the McBSP connectivity options, lists the possible interfaces, and details the protocol and data format used in each case.

#### 23.5.2.1 McBSP Functions

The device provides four instances of the McBSP module, called McBSP1, McBSP2, McBSP3 and McBSP4.

List of recommended usage (non exhaustive) per McBSP modules in the device:

- McBSP1: Digital baseband (DBB) Data
- McBSP2: Audio data
- McBSP3: Bluetooth voice data
- McBSP4: Midi data

[Table 23-326](#) describes the functions and the corresponding application fields.

**Table 23-326. Functions Description**

Function	Application field	Recommended McBSP module	Description
Control and Data	Digital Base Band (DBB) Data	McBSP1	Serial interface to transfer data
Audio Data	Audio Data without audio buffer and sidetone feature	McBSP2	Audio interface to transfer audio data with Inter-IC Sound codec (I2S)
	Midi Data	McBSP4	
Voice Data	Bluetooth Voice Data without sidetone feature	McBSP3	Voice interface to transfer voice data with Pulse Code Modulation codec (PCM)

#### 23.5.2.2 McBSP Signals Descriptions

The four McBSP modules consist of a data-flow path and a control path connected to external devices by a serial interface with 6 pins configuration (McBSP 4 only) or 4 pins configuration (McBSP1,2, 3).

For a McBSP module with 6 pins configuration, an internal loop back capability between Transmitter and Receiver clock signals, and both frame synchronisation signals enables using the McBSP module with 4 pins configuration. The related internal multiplexers are controlled through the System Control Module on the device (see [Section 23.5.3, McBSP Integration](#)).

**Table 23-327. Input/Output Description**

Pin Name	I/O	Description	Internal Signal Name	Reset Value	Control and Data	Audio Data	Voice Data
mcbasp_clks	I	External clock (shared by all McBSP modules)	CLKS	-	✓	✓	✓
abe_mcbspi_dr	I	Receive serial data	DR	-	✓	✓	✓
abe_mcbspi_dx	(I)O see Note below	Transmit serial data	DX	0	✓	✓	✓
abe_mcbspi_clkx	I/O <sup>(1)</sup>	Transmit clock <sup>(2)</sup>	CLKX	0	✓	✓	✓
abe_mcbspi_fsx	I/O <sup>(1)</sup>	Transmit frame synchronization <sup>(3)</sup>	FSX	0	✓	✓	✓
mcbasp4_dr	I	Receive serial data	DR	-	✓	✓	✓
mcbasp4_dx	(I)O see Note below	Transmit serial data	DX	0	✓	✓	✓
mcbasp4_clkx	I/O <sup>(1)</sup>	Transmit clock <sup>(2)</sup>	CLKX	0	✓	✓	✓
mcbasp4_fsx	I/O <sup>(1)</sup>	Transmit frame synchronization <sup>(3)</sup>	FSX	0	✓	✓	✓
mcbasp4_clkcr	I/O <sup>(1)</sup>	Receive clock <sup>(4)</sup>	CLKR	1	✓		
mcbasp4_fsr	I/O <sup>(1)</sup>	Receive frame synchronization <sup>(5)</sup>	FSR	0	✓		

<sup>(1)</sup> For details of the input/output selection, see [Section 23.5.2.3.1, McBSP Modes](#).

<sup>(2)</sup> This signal is also used as CLKR when it is configured as output.

<sup>(3)</sup> This signal is also used as FSR when it is configured as output.

<sup>(4)</sup> This signal is also used as CLKX when it is configured as input.

<sup>(5)</sup> This signal is also used as FSX when it is configured as input.

Legend: i= 1 to 3, I=Input, O=Output

- The mcbasp\_clks pin can be used to inject an external clock. This clock is used to generate control signals depending on the module internal configuration (see [Section 23.5.4.9, McBSP SRG](#)). The CLKS signal of the McBSP modules is linked to an external signal through the mcbasp\_clks pin, but the CLKS signal can also be linked to an internal clock provided by PRCM of the device. For more information, see [Section 23.5.3, McBSP Integration](#).
- Data are transmitted to external devices interfacing with McBSP modules via the mcbspi\_dx pin. Data from those devices are received on the mcbspi\_dr pin.

**NOTE:** The mcbspi\_dx pin is an input/output signal to use the McBSP module in half-duplex mode.

- Control information is communicated via the following pins: mcbspi\_clkx (transmit clock), mcbasp4\_clkcr (receive clock), mcbspi\_fsx (transmit frame-sync), and mcbasp4\_fsr (receive frame-sync).

#### CAUTION

External pins mcbasp4\_clkcr and mcbasp4\_fsr are connected to pads only for McBSP4 module; other McBSP modules don't have these connections. For these modules, CLKR and FSR signals sources are mcbspi\_clkx and mcbspi\_fsx pins, respectively. Consequently, there is a light restriction on other McBSP modules when used in full-duplex mode. Both reception and transmission use the same clock signal and the same frame synchronization signal.

### 23.5.2.3 McBSP Functions Description

#### 23.5.2.3.1 McBSP Modes

For all McBSP functions, McBSP modules can operate in master or slave mode. The difference between these modes is the definition of the source of McBSP clocks and McBSP frames synchronization:

- Master mode: McBSP module provides them to the external device
- Slave mode: McBSP module receives them from the external device

The choice between the two modes depends of technical data of the external device and the type of interface (protocols and data formats). For one McBSP module, there are four possible functions:

1. Transmit and receive master mode
2. Transmit and receive slave mode
3. Transmit master mode and receive slave mode
4. Transmit slave mode and receive master mode

**NOTE:** If the McBSP module has a serial interface with 4 pins configuration (McBSP1, 2, 3), only modes 1 or 2 are possible.

Figure 23-111 shows the connection between the McBSP4 module (6 pins configuration) and an external device in transmit master mode and receive slave mode.

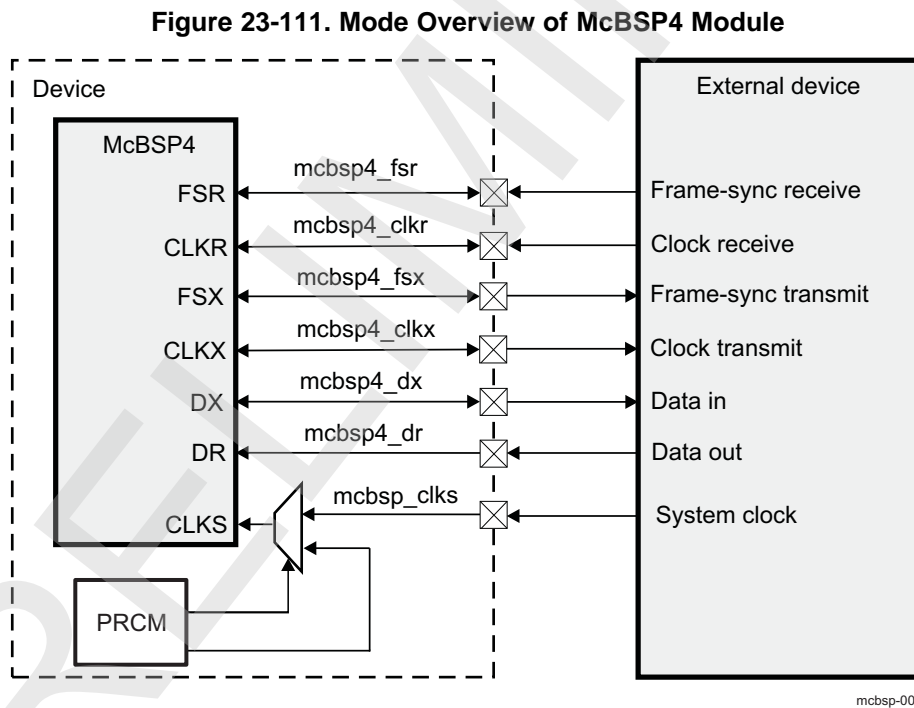
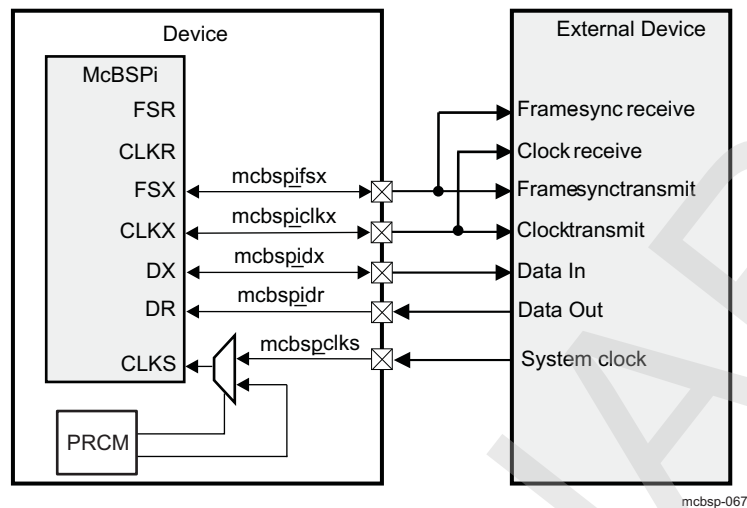


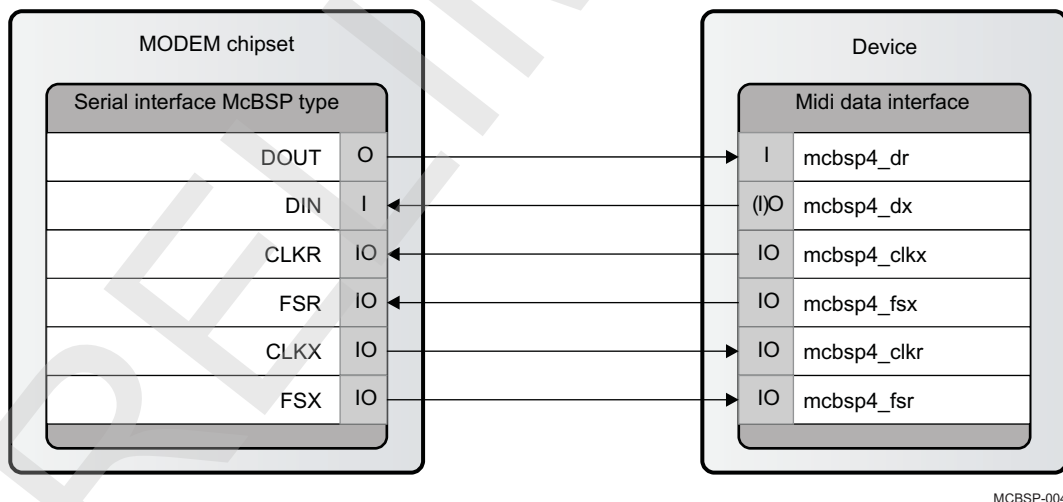
Figure 23-112 shows the connection between the McBSPi module, with i = 1, 2 or 3 (4 pins configuration) and an external device in transmit and receive master mode.

**Figure 23-112. Mode Overview of McBSPi Module**

### 23.5.2.3.2 McBSP Functions

#### 23.5.2.3.2.1 McBSP Function 1: Control and Data

In full-duplex mode (reception and transmission use independent clock signals and frame synchronization signals), the McBSP module can be used to exchange control and data with an external chipset, allowing the device to be interfaced with a modem device. Figure 23-113 shows typical connections between device and modem chipset to illustrate Midi data application.

**Figure 23-113. Midi Data Application**

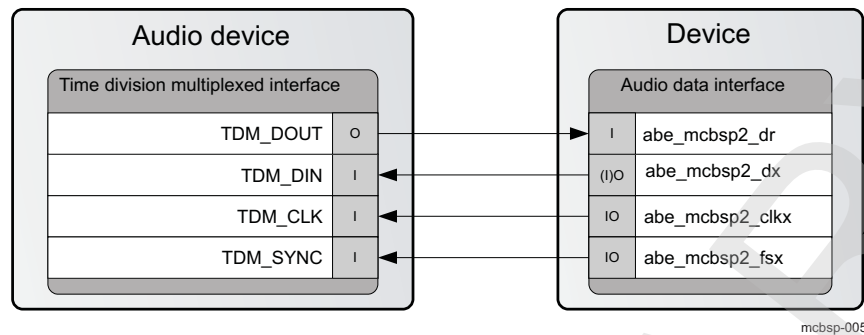
In Figure 23-113, the McBSP4 module is configured in transmit master mode and in receive slave mode.

#### 23.5.2.3.2.2 McBSP Function 2: Audio Data

The McBSP module is connected to audio devices through the I2S interface. The I2S link serial interface is a TDM slot based serial interface that is used to transfer audio data. Those audio devices can be either AICs or other serially connected A/D and D/A devices.

Figure 23-114 shows typical connections between device and a typical device of analog audio interface to illustrate the audio data application. The typical device contains several audio analog inputs and outputs, as well as digital microphone inputs.

Figure 23-114. Audio Data Application



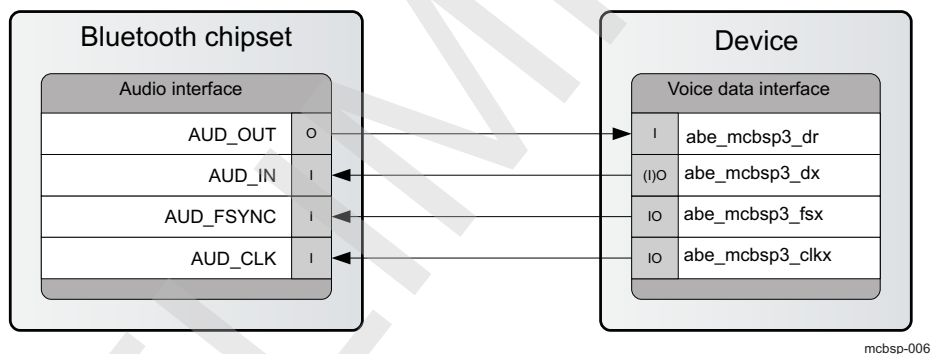
In Figure 23-114, the McBSP2 module is configured in transmit and receive master mode.

### 23.5.2.3.2.3 McBSP Function 3: Voice Data

The McBSP module is connected to a voice device through the PCM interface. The PCM link serial interface is a TDM slot based serial interface that is used to transfer voice data. The voice devices can be either modem chipsets, Bluetooth chipsets or others devices with voice data interface.

Figure 23-115 shows typical connections between device and Bluetooth chipset (TI BRF6300 or TI BRF6350) to illustrate voice data application.

Figure 23-115. Voice Data Application



In Figure 23-115, the McBSP3 module is configured in transmit and receive master mode.

### 23.5.2.4 McBSP Protocols and Data Formats

The McBSP module can use one of the three protocols with associated data formats:

- Serial protocol to exchange serial data
- Audio protocol to exchange the audio samples
- Voice protocol to exchange the voice samples

The McBSP modules offer the flexibility to modify the following parameters to adapt to the device features as described in the following subsections.

### 23.5.2.4.1 Words, Frames, and Phases Definitions

#### 23.5.2.4.1.1 Words or Channels

The data bits are transferred (transmission or reception) in a group called a serial word or channel. The number of bits in a word (length) is programmable via bits field (McBSPi.MCBSPLP\_RCR1\_REG[7:5] RWDLEN1 field and McBSPi.MCBSPLP\_RCR2\_REG[7:5] RWDLEN2 field, McBSPi.MCBSPLP\_XCR1\_REG[7:5] XWDLEN1 field and McBSPi.MCBSPLP\_XCR2\_REG[7:5] XWDLEN2 field) and can be 8, 12, 16, 20, 24 or 32 bits (see [Section 23.5.4.8.3, Clocking and Framing Data](#)). The McBSP module uses clock signals to control the time for each bit transfer. Data are sampled/driven on rising or falling edge of clock signals. This clock polarity is programmable via bits field of pin-control register (McBSPi.MCBSPLP\_PCR\_REG).

For more information, see [Section 23.5.4.8.4, Frame Phases \(Dual-Phase Frame I2S Support\)](#).

#### 23.5.2.4.1.2 Frames

One or more words (max 128) are transferred in a group called a frame. The McBSP module can transmit / receive a maximum of 128 words per frame, programmable via bits field of transmit and receive control registers (McBSPi.MCBSPLP\_XCR1\_REG/McBSPi.MCBSPLP\_XCR2\_REG and McBSPi.MCBSPLP\_RCR1\_REG/McBSPi.MCBSPLP\_RCR2\_REG). For more details, see [Section 23.5.4.8.3, Clocking and Framing Data](#).

All the words in a frame are sent in a continuous stream. However, there can be pauses between frame transfers. The McBSP module uses frame-synchronization signals to determine when each frame is received/transmitted. When a pulse occurs on a frame-synchronization signal, the McBSP module begins receiving/transmitting a frame of data. When the next pulse occurs, the McBSP module receives/transmits the next frame, and so on. Frame-synchronization pulse is active high or low. This pulse polarity is programmable via bits field of pin-control register (McBSPi.MCBSPLP\_PCR\_REG).

Each frame transfer can be delayed by 0, 1, or 2 clock cycles, depending on the value of bits for transmit and receive control registers (McBSPi.MCBSPLP\_XCR2\_REG and McBSPi.MCBSPLP\_RCR2\_REG). For more information, see [Section 23.5.4.10.6.3, Preventing Unexpected Transmit Frame-sync Pulses](#) and [Section 23.5.4.10.3.3, Preventing Unexpected Receive Frame-sync Pulses](#).

#### 23.5.2.4.1.3 Phases

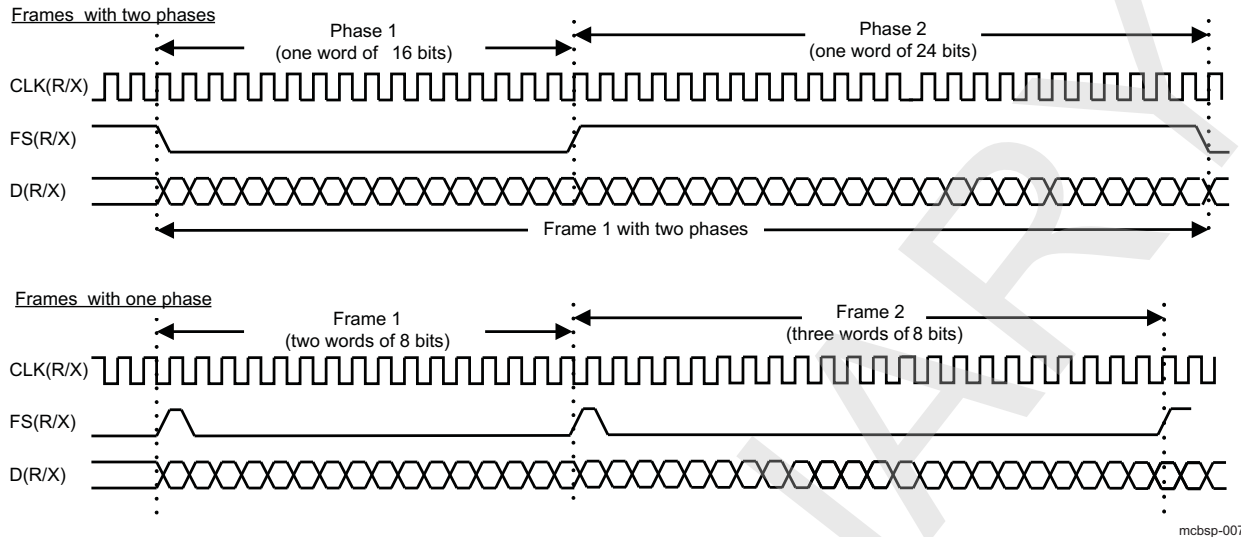
The McBSP module allows configuring each frame to contain one or two phases. The McBSP module supports dual phase frames to provide I2S fully compliant capabilities. These two phases represent left and right channels of audio stereo signals.

The limitation on dual phase frame is that the number of words per phase must be set to one for both first and second phase. But, the number of bits per word can be specified differently for each of the two phases of a frame, allowing greater flexibility in structuring data transfers.

For example, software may define a frame composed of a first phase with one 12-bit word and a second phase with one 16-bit word. This configuration allows the software to compose frames for custom applications. For more details, see [Section 23.5.4.8.4, Frame Phases \(Dual-Phase Frame I2S Support\)](#).

[Figure 23-116](#) shows signal activity for two possible reception/transmission scenarios.

Figure 23-116. McBSP Reception/Transmission Signal Activity



mcbasp-007

### 23.5.2.4.2 Serial Protocol and Data Formats

#### 23.5.2.4.2.1 Protocol

The serial protocol is used to send and receive control data without specific formats. This allows McBSP module to accommodate all serial devices and their protocols.

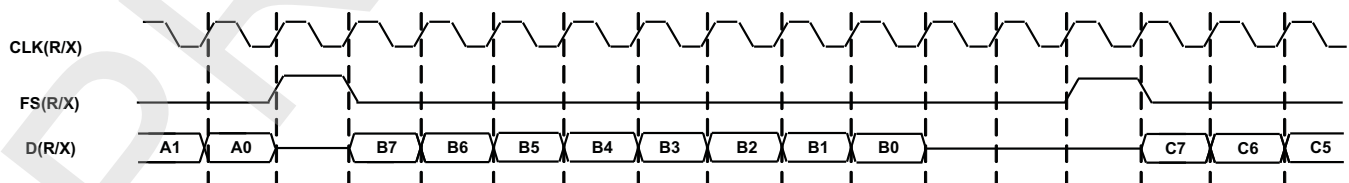
#### 23.5.2.4.2.2 Data Format

Figure 23-117 shows typical operation of the McBSP clock and frame sync signals. Serial clocks CLKR and CLKX define the boundaries between bits for receive and transmit, respectively. Similarly, frame-sync signals FSR and FSX define the beginning of an element and/or frame transfer. The McBSP module allows the configuration of the following parameters for data and frame synchronization:

- Polarity of FSR, FSX, CLKX, and CLKR
- The number of words per frame
- The number of bits per word
- Whether subsequent frame synchronization restarts the serial data stream or is ignored
- The data delay from frame synchronization to first data bit which can be 0-, 1-, or 2-bit delays

The configuration is independent for receive and transmit parts. For more details and configuration examples, see Section 23.5.4, *McBSP Functional Description* and Section 23.5.5, *McBSP Basic Programming Model*.

Figure 23-117. Serial Data Formats



mcbasp-009

### 23.5.2.4.3 Audio Protocol and Data Formats

#### 23.5.2.4.3.1 Protocol

The I2S protocol is used to send and receive audio data from 8 KHz up to 48 KHz sampling rate (frame-sync frequency), with 16 bits or 32 bits per words (Supported frequencies are 8, 11.025, 12, 16, 22.05, 24, 32, 44.1 and 48 KHz).

The frame-synchronization signal defines the frame length in the I2S protocol. Each frame consists of a fixed number of words. In dual-phase frame, the frame-synchronization signal is low for the left phase time slot and is high for the right phase time slot. In addition, the frame-synchronization signal is synchronous to the falling edge of the clock signal.

#### 23.5.2.4.3.2 Data Formats

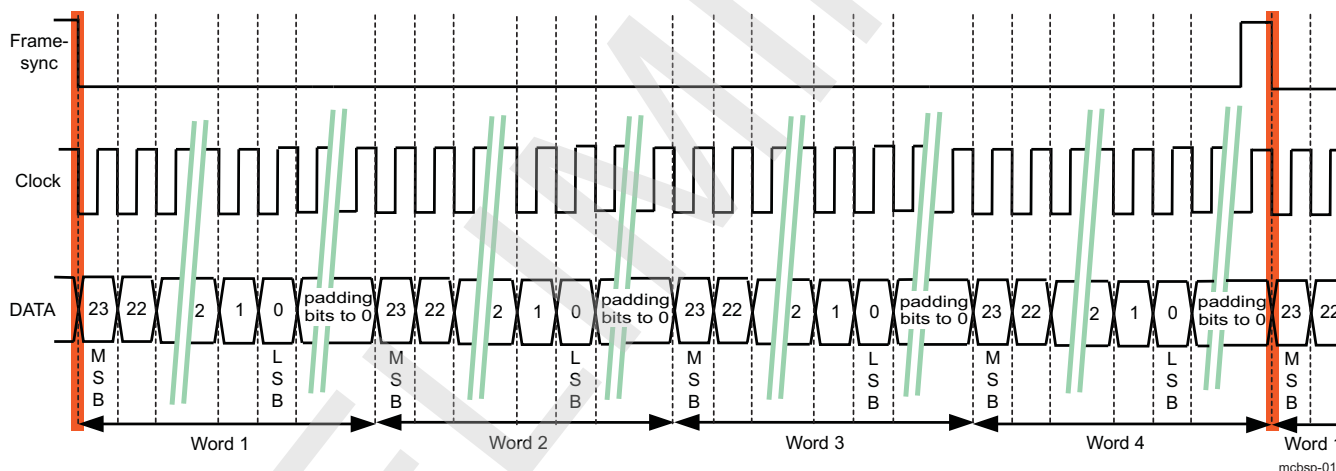
The I2S protocol supports TDM, I2S, left justified, and right justified data formats.

Bits of each word (sample) are clocked using clock signal. For each word, MSB is first. LSBs are padded to 0 when the data length (8, 12, 16, 20, or 24 bits) is less than the sample word width (16 or 32 bits).

##### 23.5.2.4.3.2.1 TDM Data Format

Figure 23-118 shows that each frame of TDM data format is composed of four words (or channels).

**Figure 23-118. TDM Data Format; Word Width: 32 Bits; Data Length: 24 Bits**

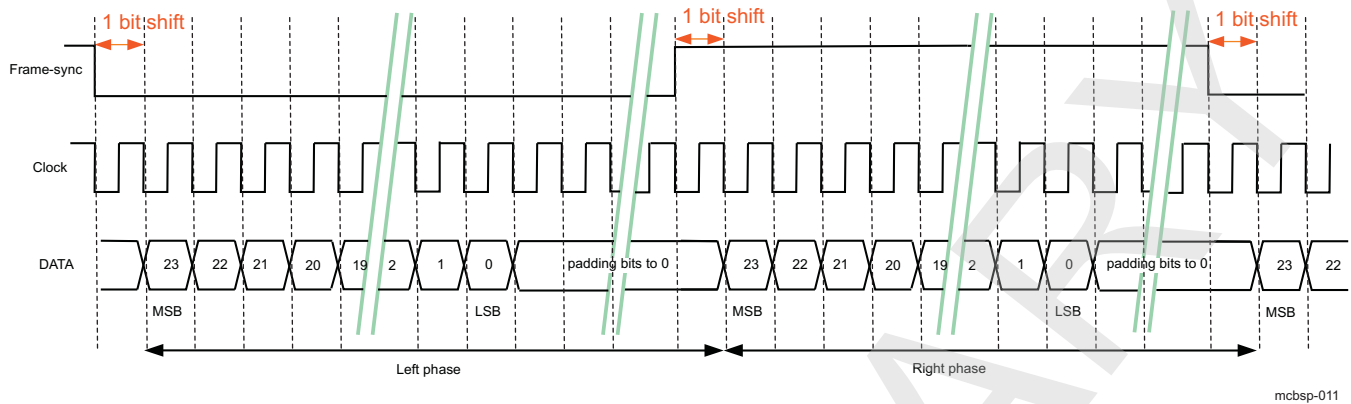


##### 23.5.2.4.3.2.2 I2S Data Format

Figure 23-119 shows an example with 24 bits data (MSB first) and 8 padding bits at '0'.



Figure 23-119. I2S Data Format; Word Width: 32 Bits; Data Length: 24 Bits

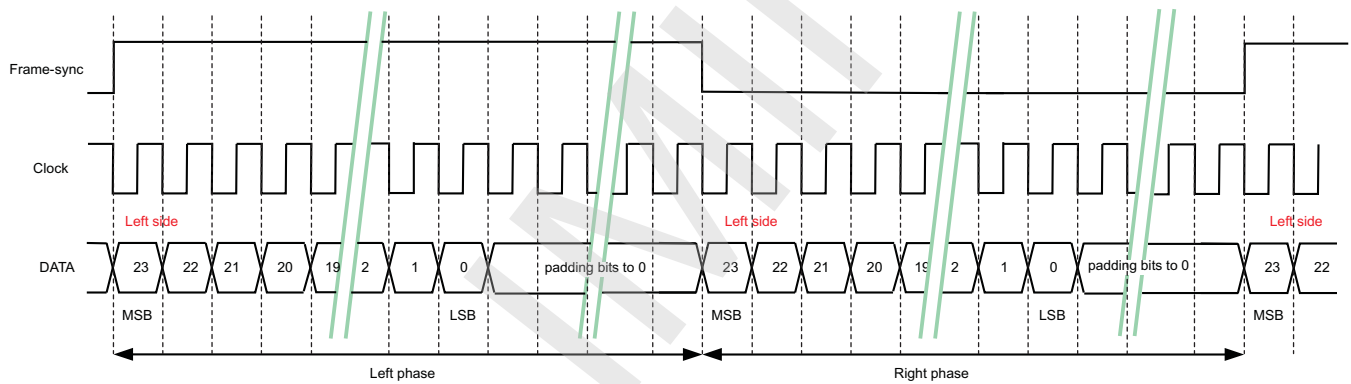


mcbasp-011

### 23.5.2.4.3.2.3 Left Justified Data Format

Figure 23-120 shows an example with 24 bits data (MSB first) and 8 padding bits at '0'.

Figure 23-120. Left Justified Data Format; Word Width: 32 Bits; Data Length: 24 Bits

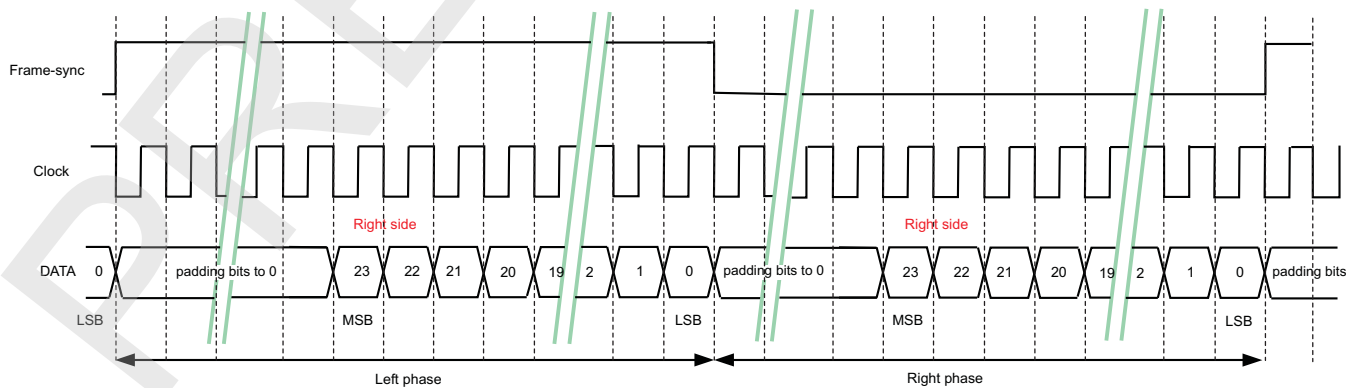


mcbasp-012

### 23.5.2.4.3.2.4 Right Justified Data Format

Figure 23-121 shows an example with 8 padding bits at '0' and 24 bits data (MSB first).

Figure 23-121. Right Justified Data Format; Word Width: 32 Bits; Data Length: 24 Bits



mcbasp-013

### 23.5.2.4.4 Voice Protocol and Data Formats

#### 23.5.2.4.4.1 Protocol

The PCM protocol is intended to transfer voice data at 8 kHz (default narrowband mode) or 16 kHz (wideband mode) sample rates (frame-sync frequency). PCM protocol can act as a slave or master, and is used by the Bluetooth interface and the modem generic interface. The frame-synchronization defines the frame length in the PCM protocol. Bits are clocked using PCM clock signal, with MSB first.

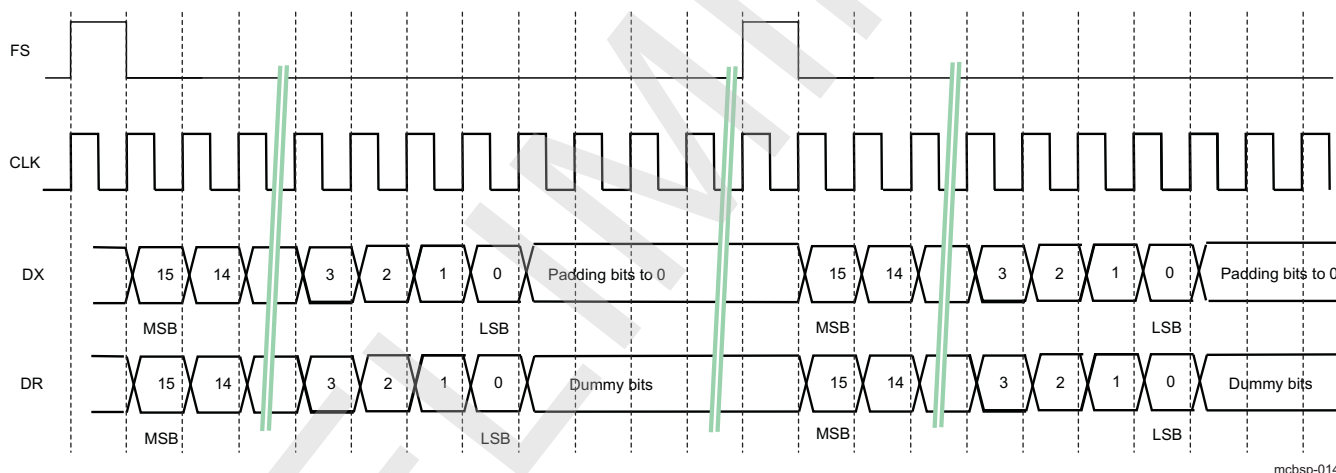
#### 23.5.2.4.4.2 Data Formats

Two modes are available for the PCM protocol: mode 1 and mode 2. For these both modes, it has two types of operations: Mono or stereo channels. The difference between PCM mode 1 and PCM mode 2 is in the way they use either the rising or the falling edge of clock signal, and the frame-synchronization polarity.

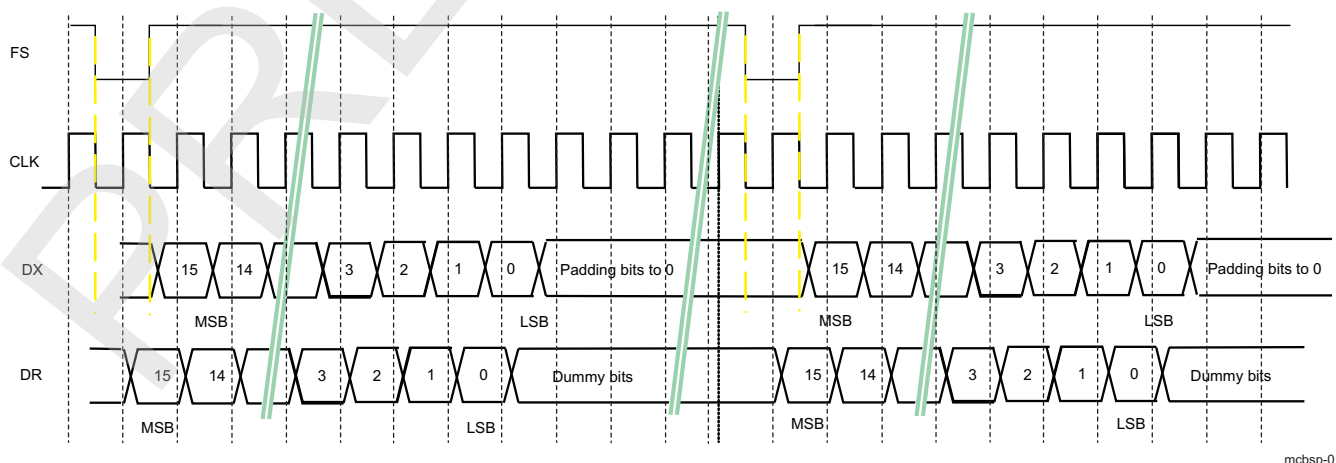
- PCM Mode 1: Input data is latched on the falling edge of the clock, and the transmitted data starts on the rising edge of the clock. Frame-synchronization pulse is active high.
- PCM Mode 2: Input data is latched on the falling edge of the clock, and the transmitted data starts on the falling edge of the clock. Frame-synchronization pulse is active low.

Figure 23-122 and Figure 23-123 shows an example of PCM protocol, mode 1 and mode 2, respectively, for a frame composed one word (width: 32 bits) with 16 bits data.

**Figure 23-122. PCM Protocol - Mode 1 Data Format**



**Figure 23-123. PCM Protocol - Mode 2 Data Format**



### 23.5.3 McBSP Integration

This section describes the McBSP modules integration in the device, including information about clocks, resets, and hardware requests.

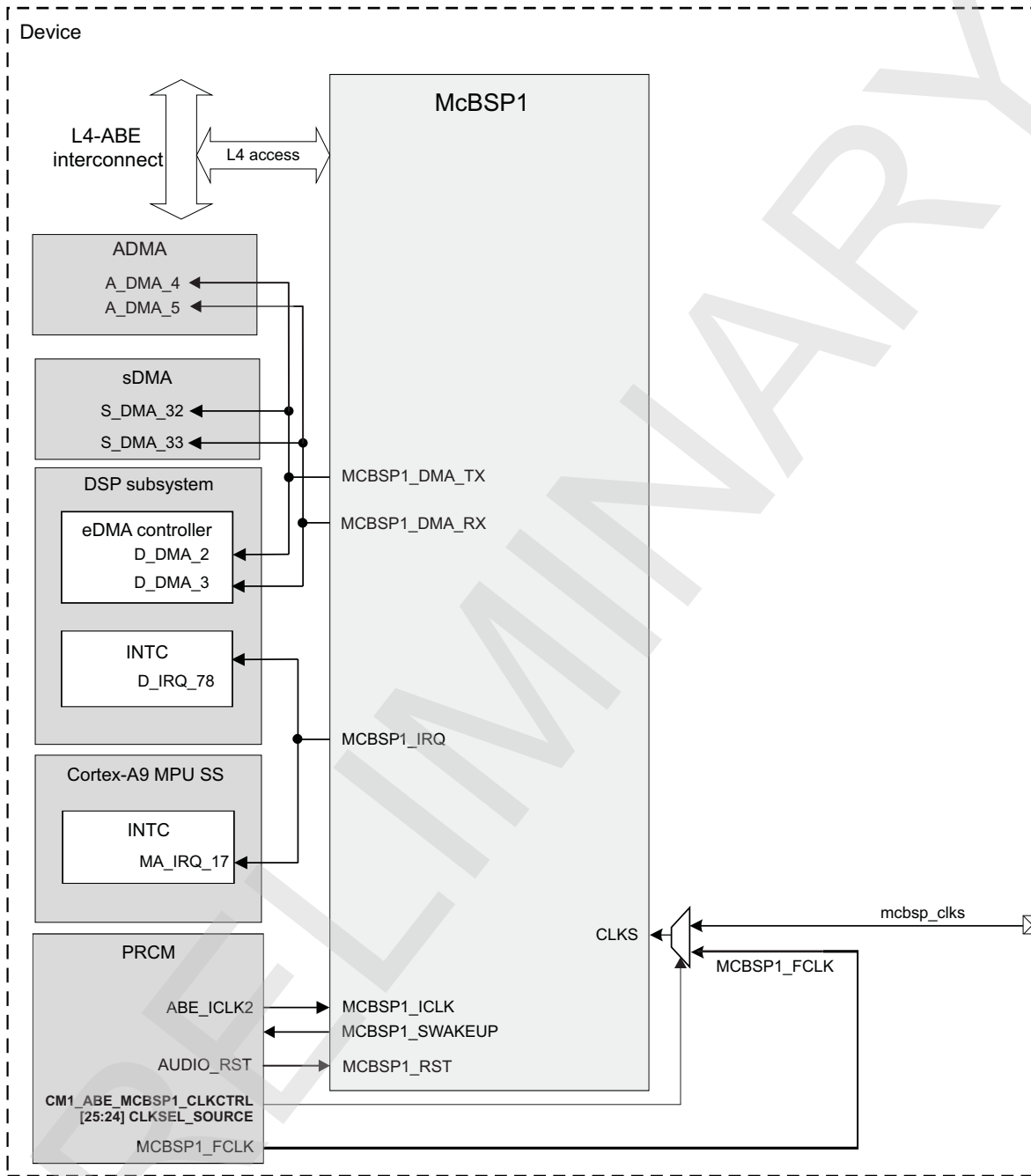
McBSP modules are divided into 2 families: McBSP modules that are gated in AUDIO power domain (McBSP1,2 and 3), and McBSP4 module which is gated in PER power domain.

[Figure 23-124](#) through [Figure 23-127](#) highlight the integration of the McBSP modules in the device including interrupt handlers, DMA requests, clock generators, and interconnections.

McBSP modules features:

- Wake-up request
- Two DMA requests
- One IRQ
- One functional clock
- One interface clock

Figure 23-124. McBSP1 Integration



mcbbsp-074

Figure 23-125. McBSP2 Integration

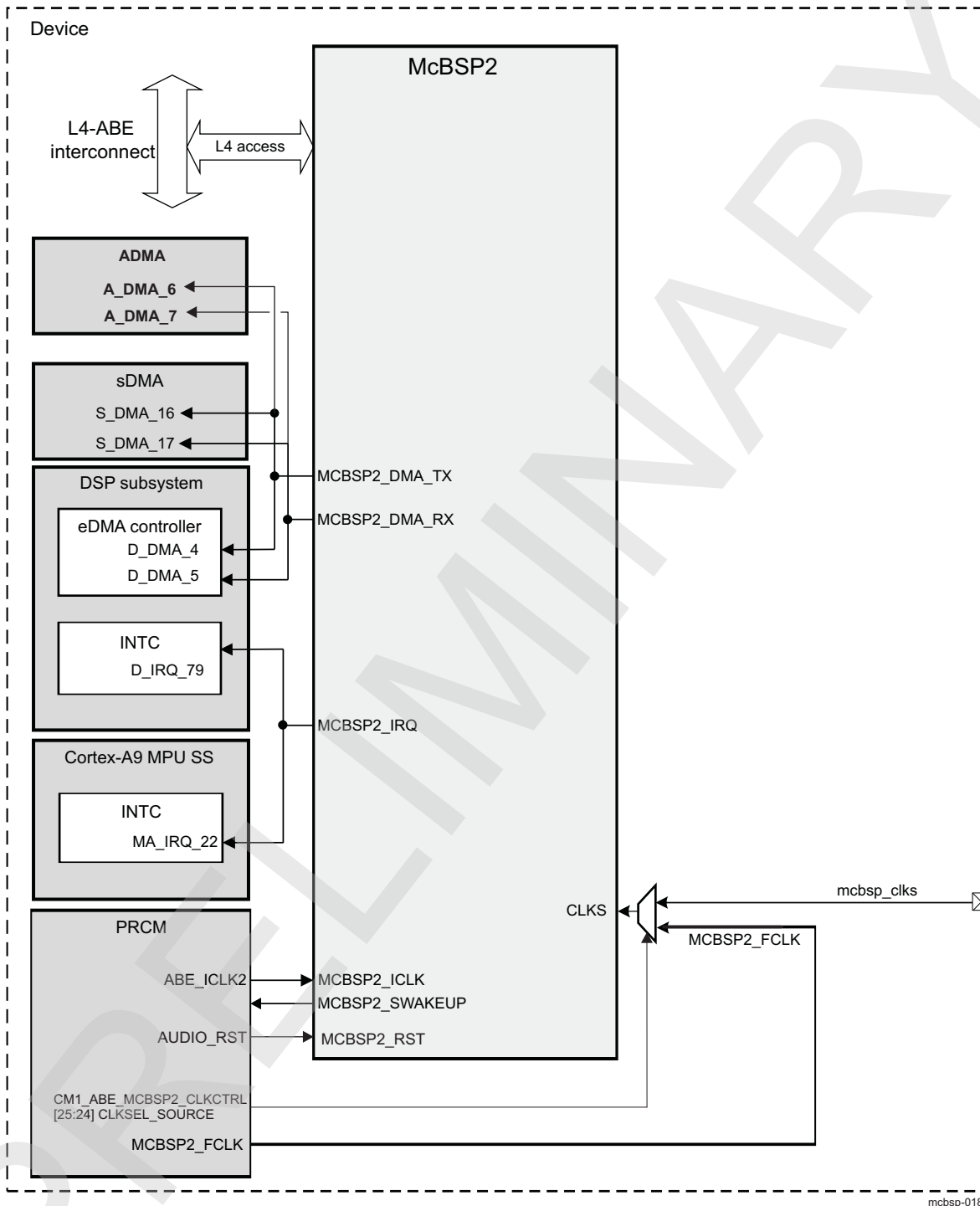
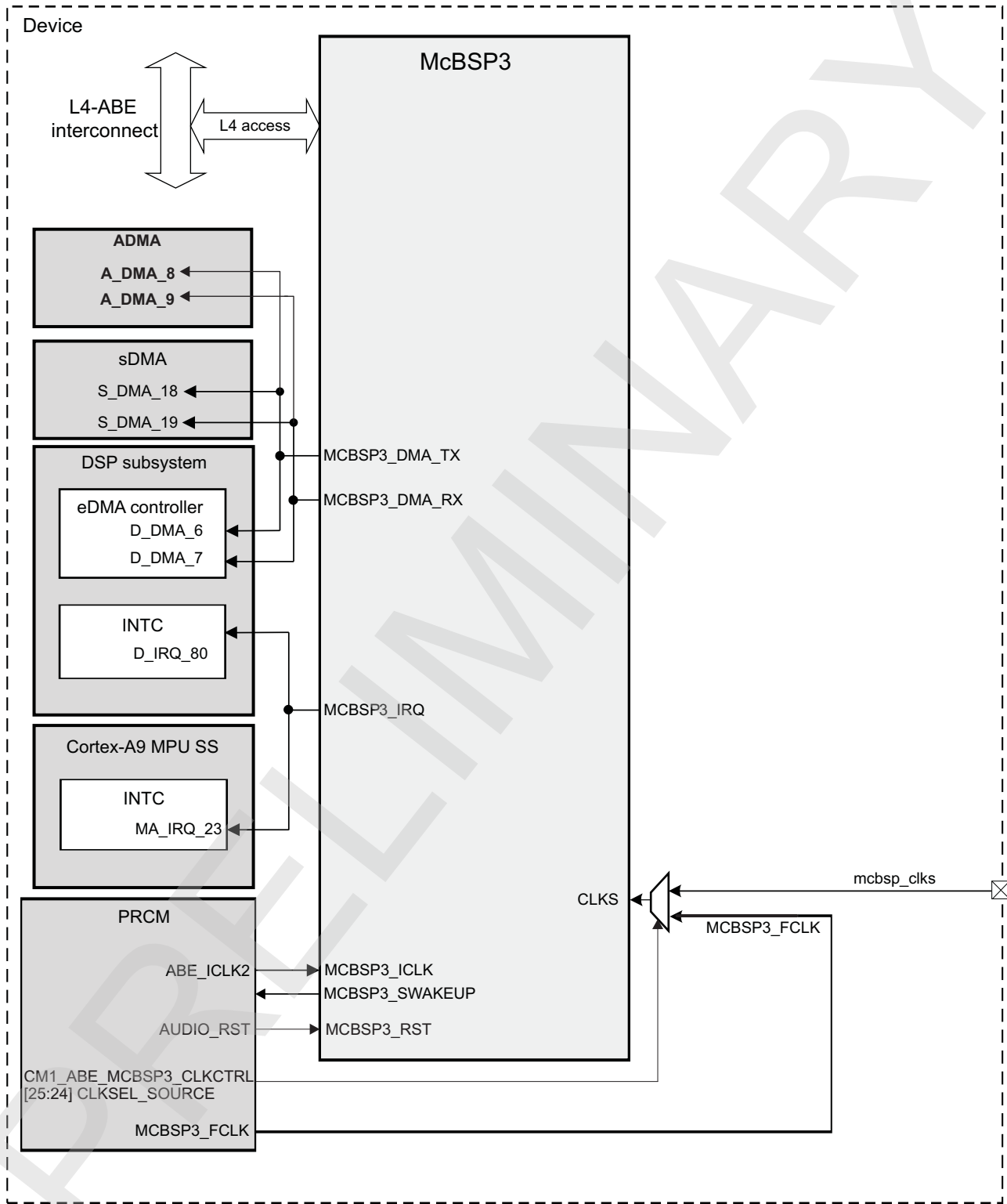
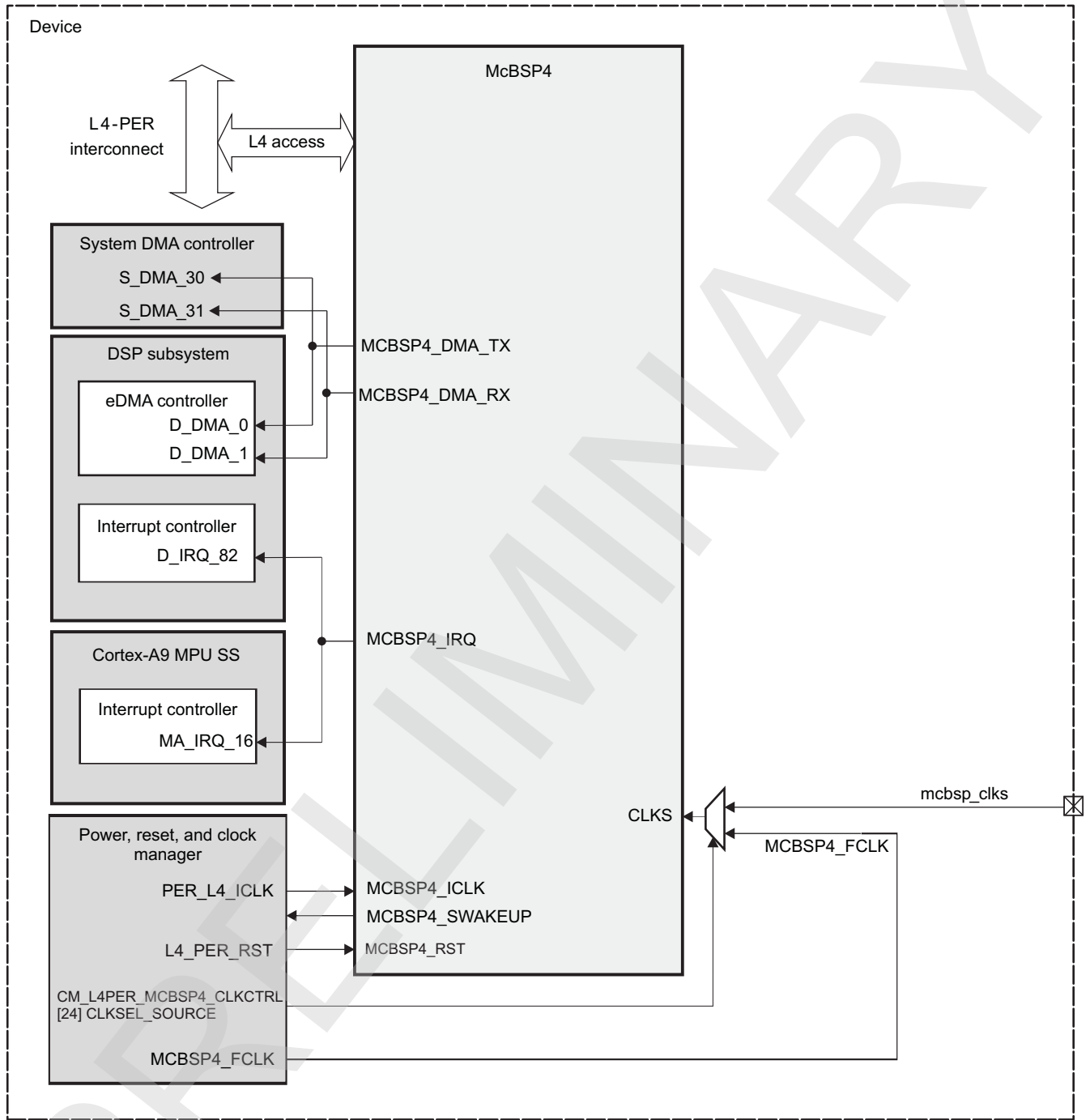


Figure 23-126. McBSP3 Integration



mcbbsp-019

Figure 23-127. McBSP4 Integration



mcbssp-017

**NOTE:** The selection between the FSX and FSR signals is made in the CONTROL\_MCBSP4[31] ALBCTRLRX\_FSX bit in the control module register. The selection between the CLKR and CLKX signals is made in the CONTROL\_MCBSP4[30] ALBCTRLRX\_CLKX bit in the control module register.

Table 23-328 through Table 23-330 summarize the integration of the module in the device.

**Table 23-328. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-up Capability	Interconnect
MCBSP1	PD_AUDIO	Yes	L4_ABE
MCBSP2	PD_AUDIO	Yes	L4_ABE
MCBSP3	PD_AUDIO	Yes	L4_ABE
MCBSP4	PD_L4_PER	Yes	L4_PER

**Table 23-329. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
MCBSP1	MCBSP1_ICLK	ABE_ICLK2	PRCM	MCBSP1 Interface Clock
	MCBSP1_FCLK	MCBSP1_FCLK	PRCM	MCBSP1 Functional Clock
MCBSP2	MCBSP2_ICLK	ABE_ICLK2	PRCM	MCBSP2 Interface Clock
	MCBSP2_FCLK	MCBSP2_FCLK	PRCM	MCBSP2 Functional Clock
MCBSP3	MCBSP3_ICLK	ABE_ICLK2	PRCM	MCBSP3 Interface Clock
	MCBSP3_FCLK	MCBSP3_FCLK	PRCM	MCBSP3 Functional Clock
MCBSP4	MCBSP4_ICLK	PER_L4_ICLK	PRCM	MCBSP4 Interface Clock
	MCBSP4_FCLK	MCBSP4_FCLK	PRCM	MCBSP4 Functional Clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
MCBSP1	MCBSP1_RST	AUDIO_RST	PRCM	AUDIO Power Domain Reset
MCBSP2	MCBSP2_RST	AUDIO_RST	PRCM	AUDIO Power Domain Reset
MCBSP3	MCBSP3_RST	AUDIO_RST	PRCM	AUDIO Power Domain Reset
MCBSP4	MCBSP4_RST	L4_PER_RST	PRCM	L4_PER Power Domain Reset

**Table 23-330. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
MCBSP1	MCBSP1_IRQ	D_IRQ_78	DSP INTC	McBSP 1: Common Synchronous Interrupt Request line
	MCBSP1_IRQ	MA_IRQ_17	Cortex-A9 MPU INTC	McBSP 1: Common Synchronous Interrupt Request line
MCBSP2	MCBSP2_IRQ	D_IRQ_79	DSP INTC	McBSP 2: Common Synchronous Interrupt Request line
	MCBSP2_IRQ	MA_IRQ_22	Cortex-A9 MPU INTC	McBSP 2: Common Synchronous Interrupt Request line
MCBSP3	MCBSP3_IRQ	D_IRQ_80	DSP INTC	McBSP 3: Common Synchronous Interrupt Request line
	MCBSP3_IRQ	MA_IRQ_23	Cortex-A9 MPU INTC	McBSP 3: Common Synchronous Interrupt Request line
MCBSP4	MCBSP4_IRQ	D_IRQ_82	DSP INTC	McBSP 4: Common Synchronous Interrupt Request line
	MCBSP4_IRQ	MA_IRQ_16	Cortex-A9 MPU INTC	McBSP 4: Common Synchronous Interrupt Request line
DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
MCBSP1	MCBSP1_DMA_TX	A_DMA_4	ADMA	MCBSP module 1 - transmit DMA request
	MCBSP1_DMA_RX	A_DMA_5	ADMA	MCBSP module 1 - receive DMA request
	MCBSP1_DMA_TX	S_DMA_32	sDMA	MCBSP module 1 - transmit DMA request



**Table 23-330. Hardware Requests (continued)**

	MCBSP1_DMA_RX	S_DMA_33	sDMA	MCBSP module 1 - receive DMA request
	MCBSP1_DMA_TX	D_DMA_2	DDMA	MCBSP module 1 - transmit DMA request
	MCBSP1_DMA_RX	D_DMA_3	DDMA	MCBSP module 1 - receive DMA request
MCBSP2	MCBSP2_DMA_TX	A_DMA_6	ADMA	MCBSP module 2 - transmit DMA request
	MCBSP2_DMA_RX	A_DMA_7	ADMA	MCBSP module 2 - receive DMA request
	MCBSP2_DMA_TX	S_DMA_16	sDMA	MCBSP module 2 - transmit DMA request
	MCBSP2_DMA_RX	S_DMA_17	sDMA	MCBSP module 2 - receive DMA request
	MCBSP2_DMA_TX	D_DMA_4	DDMA	MCBSP module 2 - transmit DMA request
	MCBSP2_DMA_RX	D_DMA_5	DDMA	MCBSP module 2 - receive DMA request
MCBSP3	MCBSP3_DMA_TX	A_DMA_8	ADMA	MCBSP module 3 - transmit DMA request
	MCBSP3_DMA_RX	A_DMA_9	ADMA	MCBSP module 3 - receive DMA request
	MCBSP3_DMA_TX	S_DMA_18	sDMA	MCBSP module 3 - transmit DMA request
	MCBSP3_DMA_RX	S_DMA_19	sDMA	MCBSP module 3 - receive DMA request
	MCBSP3_DMA_TX	D_DMA_6	DDMA	MCBSP module 3 - transmit DMA request
	MCBSP3_DMA_RX	D_DMA_7	DDMA	MCBSP module 3 - receive DMA request
MCBSP4	MCBSP4_DMA_TX	S_DMA_30	sDMA	MCBSP module 4 - transmit DMA request
	MCBSP4_DMA_RX	S_DMA_31	sDMA	MCBSP module 4 - receive DMA request
	MCBSP4_DMA_TX	D_DMA_0	DDMA	MCBSP module 4 - transmit DMA request
	MCBSP4_DMA_RX	D_DMA_1	DDMA	MCBSP module 4 - receive DMA request

**NOTE:**

- About interrupt source description, see [Section 23.5.4.6, Interrupt Requests](#).
- About DMA source description, see [Section 23.5.4.7, DMA Requests](#).

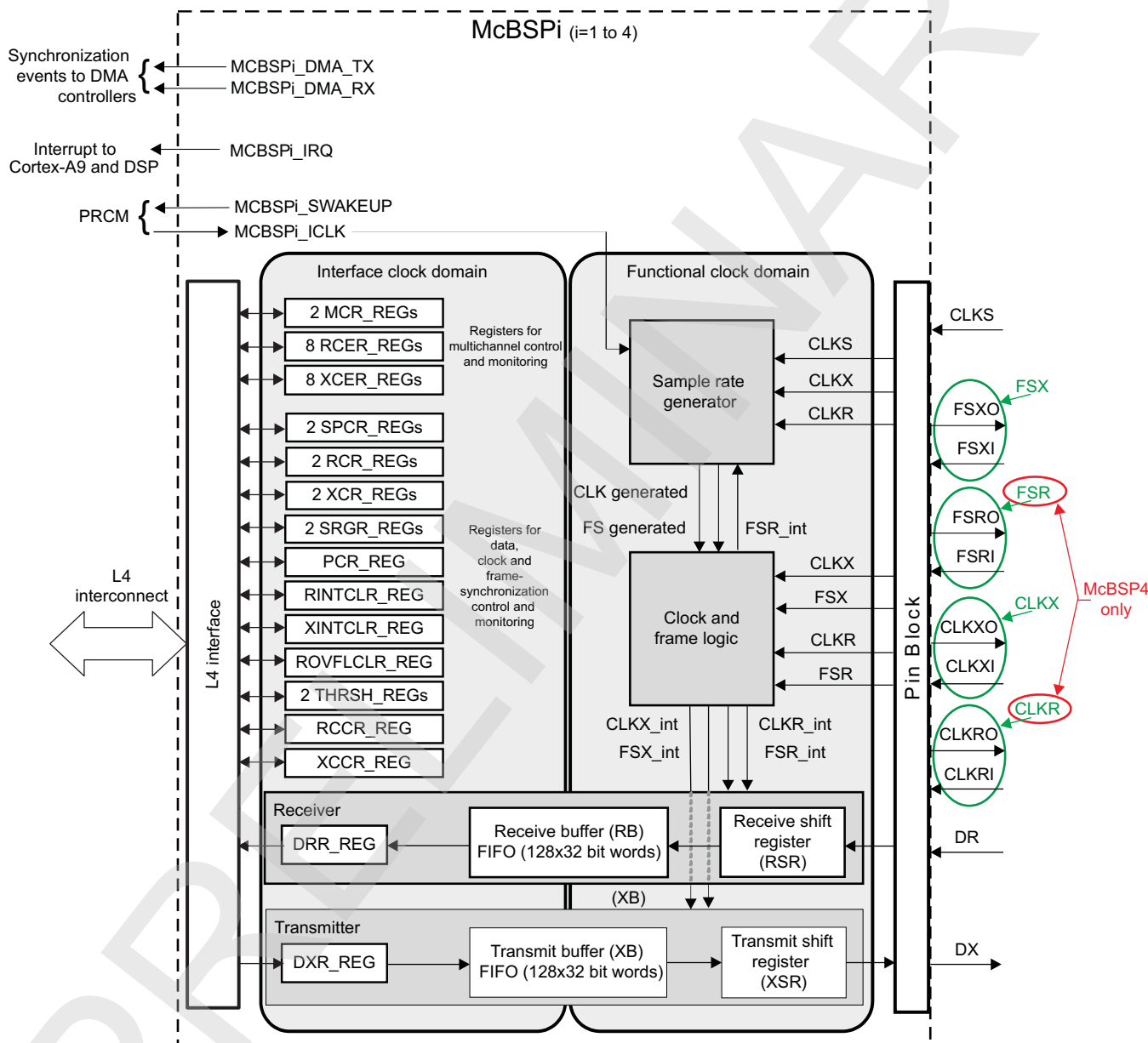
### 23.5.4 McBSP Functional Description

This section is a functional description of the McBSP module.

#### 23.5.4.1 Block Diagram

Figure 23-128 shows functional block diagram of the four instances of the McBSP modules.

Figure 23-128. McBSP iBlock Diagram



mcbasp-021

#### 23.5.4.2 Signal Source Control

The CLKX, DX, DR, FSX, FSR and CLKR signals sources are defined by the system control module. The control registers of the system control module are used to select these signals sources. For more information, see Chapter 18, Control Module.

### 23.5.4.3 McBSP Clock Configuration

#### 23.5.4.3.1 McBSP1 Clocks

The McBSP1 module is clocked by a functional clock (CLKS and CLKX) and an interface clock (MCBSP1\_ICLK).

- The functional clock is used to generate control signals depending on the module internal configuration (see [Section 23.5.4](#)). For McBSP1 module, the functional clock comes from the CLKS signal and the CLKX signal. The choice between these two clocks is defined by the SCLKME bit of the MCBSP1.MCBSPLP\_PCR\_REG[7] register and the CLKSM bit of the MCBSP1.MCBSPLP\_SRGR2\_REG[13] register.

The CLKS signal of the McBSP1 module is linked to an internal clock (MCBSP1\_FCLK) provided by PRCM. The CLKS signal can also be linked to an external signal through the mcbbsp\_clks pin of the device boundary.

---

**NOTE:** When the McBSP1 module does not require the functional clock anymore, the software can disable it at the PRCM level by configuring the MODULEMODE bit field to 0x0 (PRCM.CM1\_ABE\_MCBSP1\_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At PRCM level, when all the conditions to shut-off MCBSP1\_FCLK clock are met the PRCM automatically launches a hardware handshake protocol to ensure McBSP is ready to have this clock switched off. Namely, the PRCM asserts an IDLE request to McBSP module. For more details, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

---

Only, the CLKX signal is connected by mcbbsp4\_clkx pads. The CLKR signal is connected to the CLKX signal. These signals are used like functional clocks by the intermediary of SRG.

- The MCBSP1\_ICLK runs at the L4 core interconnect clock speed. It is used to trigger access to the McBSP1 L4 interface and McBSP1 configuration interface via the Cortex-A9 MPU/DSP shared bus. It can also be an input clock for the McBSP sample-rate generator (clock divider), depending on the module configuration (see [Section 23.5.4.9, McBSP SRG](#)). Its source is the ABE\_ICLK2 signal.

---

**NOTE:** When the McBSP1 module does not require the interface clock anymore, the software can disable it at the PRCM level by configuring the MODULEMODE bit field (PRCM.CM1\_ABE\_MCBSP1\_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At PRCM level, when all the conditions to shut-off ABE\_ICLK2 clock are met the PRCM automatically launches a hardware handshake protocol to ensure McBSP is ready to have this clock switched off. Namely, the PRCM asserts an IDLE request to McBSP module. For more details, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power Reset and Clock Management](#).

---

#### 23.5.4.3.2 McBSP2 Clocks

The McBSP2 module is clocked by a functional clock (CLKS or CLKX) and an interface clock (MCBSP2\_ICLK).

- The functional clock is used to generate control signals depending on the module internal configuration (see [Section 23.5.4, McBSP Functional Description](#)). For McBSP2 module, the functional clock comes from the CLKS signal and CLKX signal. The choice between these two clocks is defined by the SCLKME bit of the MCBSP2.MCBSPLP\_PCR\_REG[7] register and the CLKSM bit of the MCBSP2.MCBSPLP\_SRGR2\_REG[13] register.

**NOTE:** When the McBSP2 module does not require the functional clock anymore, the software can disable it at the PRCM level by configuring the MODULEMODE bit field to 0x0 (PRCM.CM1\_ABE\_MCBSP2\_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At PRCM level, when all the conditions to shut-off MCBSP2\_FCLK clock are met the PRCM automatically launches a hardware handshake protocol to ensure McBSP is ready to have this clock switched off. Namely, the PRCM asserts an IDLE request to McBSP module. For more details, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

Only, the CLKX signal is connected by mcbasp2\_clkx pads. The CLKR signal is connected to the CLKX signal. These signals are used like functional clocks by the intermediary of the SRG.

- The MCBSP2\_ICLK runs at the L4 core interconnect clock speed. It is used to trigger access to the McBSP2 L4 interface and McBSP2 configuration interface via the Cortex-A9 MPU/DSP shared bus. It can also be an input clock for the McBSP sample-rate generator (clock divider), depending on the module configuration (see [Section 23.5.4.9, McBSP SRG](#)). Its source is either the ABE\_ICLK2 signal.

**NOTE:** When the McBSP2 module does not require the interface clock anymore, the software can disable it at the PRCM level by configuring the MODULEMODE bit field (PRCM.CM1\_ABE\_MCBSP2\_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At PRCM level, when all the conditions to shut-off ABE\_ICLK2 clock are met the PRCM automatically launches a hardware handshake protocol to ensure McBSP is ready to have this clock switched off. Namely, the PRCM asserts an IDLE request to McBSP module. For more details, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

### 23.5.4.3.3 McBSP3 Clocks

The McBSP3 module is clocked by a functional clock (CLKS or CLKX) and an interface clock (MCBSP3\_ICLK).

- The functional clock is used to generate control signals depending on the module internal configuration (see [Section 23.5.4, McBSP Functional Description](#)). For McBSP3 module, the functional clock comes from the CLKS signal and CLKX signal. The choice between these two clocks is defined by the SCLKME bit of the MCBSP3.MCBSPLP\_PCR\_REG[7] register and the CLKSM bit of the MCBSP3.MCBSPLP\_SRGR2\_REG[13] register.

**NOTE:** When the McBSP3 module does not require the functional clock anymore, the software can disable it at the PRCM level by configuring the MODULEMODE bit field to 0x0 (PRCM.CM1\_ABE\_MCBSP3\_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At PRCM level, when all the conditions to shut-off MCBSP3\_FCLK clock are met the PRCM automatically launches a hardware handshake protocol to ensure McBSP is ready to have this clock switched off. Namely, the PRCM asserts an IDLE request to McBSP module. For more details, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

Only, the CLKX signal is connected by mcbasp3\_clkx pads. The CLKR signal is connected to the CLKX signal. These signals are used like functional clocks by the intermediary of SRG.

- The MCBSP3\_ICLK runs at the L4 core interconnect clock speed. It is used to trigger access to the McBSP3 L4 interface and McBSP3 configuration interface via the Cortex-A9 MPU/DSP shared bus. It can also be an input clock for the McBSP sample-rate generator (clock divider), depending on the module configuration (see [Section 23.5.4.9, McBSP SRG](#)). Its source is either the ABE\_ICLK2 signal.

**NOTE:** When the McBSP2 module does not require the interface clock anymore, the software can disable it at the PRCM level by configuring the MODULEMODE bit field (PRCM.CM1\_ABE\_MCBSP3\_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At PRCM level, when all the conditions to shut-off ABE\_ICLK2 clock are met the PRCM automatically launches a hardware handshake protocol to ensure McBSP is ready to have this clock switched off. Namely, the PRCM asserts an IDLE request to McBSP module. For more details, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

It is also possible to activate an autoidle mode for this clock (PRCM.CM\_AUTOIDLE\_PER[1] register AUTO\_MCBSP3 bit set to 1). In this case, McBSP3\_ICLK follows the AUDIO\_L4 clock domain behavior on the device. For more information, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

#### 23.5.4.3.4 McBSP4 Clocks

The McBSP4 module is clocked by a functional clock (CLKS, CLKX or CLKR) and an interface clock (MCBSP4\_ICLK).

- The functional clock is used to generate control signals depending on the module internal configuration (see [Section 23.5.4, McBSP Functional Description](#)). For McBSP4 module, the functional clock comes from the CLKS signal, CLKX signal, or CLKR signal. The choice between these three clocks is defined by the SCLKME bit of the MCBSP4.MCBSPLP\_PCR\_REG[7] register and the CLKSM bit of the MCBSP4.MCBSPLP\_SRGR2\_REG[13] register.

**NOTE:** When the McBSP4 module does not require the PRCM functional clock anymore, the software can disable it at the PRCM level by setting the MODULEMODE bit (PRCM.CM\_L4PER\_MCBSP4\_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At PRCM level, when all the conditions to shut-off MCBSP4\_FCLK clock are met the PRCM automatically launches a Hardware handshake protocol to ensure McBSP is ready to have this clock switched off. Namely, the PRCM asserts an IDLE request to McBSP module. For more details, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

The CLKX and CLKR signals are connected either by mcbbsp4\_clkx or mcbbsp4\_clkr pads. These signals are used like functional clocks by the intermediary of the sample rate generator (SRG).

- The McBSP4\_ICLK runs at the L4 core interconnect clock speed. It is used to trigger access to the McBSP4 L4 interface and McBSP4 configuration interface via the Cortex-A9 MPU/DSP shared bus. It can also be an input clock for the McBSP sample-rate generator (clock divider), depending on the module configuration (see [Section 23.5.4.9, McBSP SRG](#)). Its source is the PER\_L4\_ICLK signal.

**NOTE:** When the McBSP4 module does not require the interface clock anymore, the software can disable it at the PRCM level by setting the MODULEMODE bit (PRCM.CM\_L4PER\_MCBSP4\_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At PRCM level, when all the conditions to shut-off PER\_L4\_ICLK clock are met the PRCM automatically launches a hardware handshake protocol to ensure McBSP is ready to have this clock switched off. Namely, the PRCM asserts an IDLE request to McBSP module. For more details, see [Section 3.1.1.1.2, Module Level Clock Management](#) in [Chapter 3, Power, Reset, and Clock Management](#).

#### 23.5.4.4 McBSP Software Reset

McBSP4 module belong to the PER domain and its reset signal is the L4\_PER\_RST from the PRCM module. The McBSP1, 2 and 3 modules belong to the AUDIO domain and their reset signal is the AUDIO\_RST from the PRCM module.

**Table 23-331. Software Reset Signals to All McBSP Modules**

Type	Bit Field	Register Source	Activation	Description
Software	SOFTRESET	MCBSPi.MCBSPLP_SYSCONFIG_REG[1]	Active high	McBSP global software reset
	RRST	MCBSPi.MCBSPLP_SPCR1_REG[0]	Active low	This resets and disables the receiver, including the RB
	XRST	MCBSPi.MCBSPLP_SPCR2_REG[0]		This resets and disables the transmitter, including the XB
	GRST	MCBSPi.MCBSPLP_SPCR2_REG[6]		SRG is reset
	FRST	MCBSPi.MCBSPLP_SPCR2_REG[7]	Frame-sync logic is reset. Frame-sync generated signal is not generated by the SRG	

See [Section 23.5.5.1, McBSP Initialization Procedure](#) for a complete description of the McBSP initialization procedure.

#### 23.5.4.5 McBSP Power Management

##### 23.5.4.5.1 McBSP Operating States

Two operating states are defined for all the McBSP modules:

- **ACTIVE** state: The module is running synchronously on the interface and functional clock, interrupts and DMA requests can be generated according to the configuration (register, master or slave mode, etc) and the external signals.
- **IDLE** state: As part of the system power management, the PRCM module can request the McBSP modules to enter IDLE state. Depending on the configured acknowledgment mode: Force Idle, No Idle and Smart Idle modes, a McBSP module will effectively enter IDLE state or not. As soon as a McBSP module enters IDLE state, it doesn't have anymore activities except those unrelated to clock activity (for example wakeup features) and its clocks are likely to be switched off at PRCM level.

**NOTE:** IDLE request and IDLE acknowledge are only internal signals, with no means to observe or to control. The signals generation and control is purely hardware (managed automatically by the PRCM and the McBSP module depending on the SIDLEMODE settings).

##### 23.5.4.5.2 McBSP Acknowledgment Modes

During initialization or configuration of the McBSP module, the software must configure how the McBSP module will answer an IDLE solicitation from the PRCM module (that is, the way idle acknowledge will be asserted following an idle request assertion).

Each McBSP module can be configured via the MCBSPi.MCBSPLP\_SYSCONFIG\_REG[4:3] SIDLEMODE field as one of the following acknowledgment modes:

- **Force Idle mode** (SIDLEMODE bit = 0x0): An idle request is acknowledged unconditionally, regardless of the internal state of the module. The McBSP module immediately enters Idle state (no activity), interface and PRCM functional clock can be stopped, no interrupts and DMA requests can be generated. In this mode, the McBSP module freezes all the internal activity when the PRCM clocks are switched off by the PRCM module, leading to a potential loss of data.

#### CAUTION

In Force Idle mode, the wake-up feature is inhibited.

**CAUTION**

If the McBSP functional part, transmitter or receiver, is running within this period of time (the functional clock source is not the PRCM functional clock), the internal state of the McBSP module will not be idle (FSM states, processes, etc.), and when the McBSP module exits from the Force Idle state unexpected behavior may happen in both receiver and transmitter. To avoid this, both receive and transmit parts, must be disabled by software prior to idle request assertion (all functional clock external sources must be disabled).

- No Idle mode (SIDLEMODE bit = 0x1): An idle request is never acknowledged, meaning it will prevent the PRCM from switching off its related clocks and from putting in a lower power state than the power domain it belongs to. The McBSP module never enters Idle state (is active).
- Smart Idle mode (SIDLEMODE bit = 0x2): Acknowledgement to an idle request is given based on the internal activity of the McBSP module. The McBSP module is in a waiting state, interface / functional clocks can be stopped, no interrupt can be generated, a wake-up signal can be generated according to the configuration (See [Section 23.5.4.5.4, Analysis of the Receiver Smart Idle Behavior](#)) and external signals.

**NOTE:** The value `MCBSPi.MCBSPLP_SYSCONFIG_REG[4:3]` SIDLEMODE field = 0x3 must not be used.

When configured in Smart Idle mode, McBSP module also offers an additional granularity on `MCBSPi_FCLK` and `MCBSPi_ICLK` gating. `MCBSPi.MCBSPLP_SYSCONFIG_REG[9:8]` CLOCKACTIVITY bit field is used to determine which clock will be shut down (`MCBSPi_FCLK`, `MCBSPi_ICLK`, none of them or both of them).

CLOCKACTIVITY setting is used in the McBSP module to determine on which part of the module the conditions to acknowledge the PRCM IDLE request will be tested. As an example, if `MCBSPi_FCLK` is said not to be shut down upon a PRCM IDLE request, this means McBSP module will only consider `MCBSPi_ICLK` and the associated pending activities before acknowledging the request.

**NOTE:** Some of McBSP features are associated to `MCBSPi_ICLK` and others to `MCBSPi_FCLK`. Using the CLOCKACTIVITY along with the Smart Idle mode ensures that the features associated with the clock that will remain active are always enabled, even if McBSP has acknowledged an IDLE request. For more information, see [Section 23.5.4.5.4, Analysis of the Receiver Smart Idle Behavior](#).

[Table 23-332](#) lists the value of the bit field and indicates if the interface (`MCBSPi_ICLK`) and PRCM functional (`MCBSPi_FCLK`) clocks can be switched-off or not when an Idle request is received by McBSP module.

**Table 23-332. State of Clocks When the Module is in Idle State**

CLOCKACTIVITY Value	Interface Clock (MCBSPi_ICLK)	PRCM Functional Clock
0b00	OFF	OFF
0b01	OFF	ON
0b10	ON	OFF
0b11	ON	ON

**NOTE:** OFF means this clock can be switched-off.

ON means this clock must be maintained during wake up period.

#### CAUTION

The PRCM module does not have any hardware mean to read CLOCKACTIVITY settings. It is thus software responsibility to ensure a consistent programming between McBSPi.MCBSPLP\_SYSCONFIG\_REG[9:8] CLOCKACTIVITY bit field and PRCM McBSPi\_FCLK and L4\_ICLK control bits (see Notes in previous Section 23.5.4.3, *McBSP Clock Configuration*). If McBSP module is disabled while CLOCKACTIVITY is set to 0x3, nothing prevents the PRCM module from asserting its IDLE request which will be acknowledged regardless of the features associated with the McBSP clocks. This may lead to unpredictable behaviors.

#### 23.5.4.5.3 Wake-Up Capability

When configured in Smart Idle mode, the sources for wake-up generation are a subset of the interrupt sources. The wake up sources are enabled by setting the McBSPi.MCBSPLP\_SYSCONFIG\_REG[2] ENAWAKEUP bit (wake up feature control):

- Set to '0', wake up capability is disabled
- Set to '1', wake up capability is enabled

The McBSPi\_SWAKEUP signal is the McBSP module asynchronous wake-up signal sent to the PRCM module when a wake-up generation is requested.

The wake up configurations are defined by setting the corresponding bits in the McBSPi.MCBSPLP\_WAKEUPEN\_REG register.

##### 23.5.4.5.3.1 Receive Wakeup

There are 4 receive possible wake up configurations:

- McBSPi.MCBSPLP\_WAKEUPEN\_REG[3] RRDYEN bit: The McBSP module asserts the McBSPi\_SWAKEUP request when the receive buffer reaches the high threshold value (RTHRESHOLD value + 1) of the McBSPi.MCBSPLP\_THRSH1\_REG register. If the McBSPi.MCBSPLP\_IRQENABLE\_REG[3] RRDYEN bit is set to 1, the McBSP module sends an interrupt (McBSPi\_IRQ) request to the MPU or DSP subsystems when exiting from idle mode (interrupt will be asserted once the McBSPi.MCBSPLP\_IRQSTATUS\_REG[3] RRDY bit changes from '0' to '1', indicating that received data is ready to be read).
- McBSPi.MCBSPLP\_WAKEUPEN\_REG[2] REOFEN bit: The McBSP module asserts the McBSPi\_SWAKEUP request at the end of the frame. If the McBSPi.MCBSPLP\_IRQENABLE\_REG[2] REOFEN bit is set to 1, the McBSP module sends an interrupt (McBSPi\_IRQ) request to the MPU or DSP subsystems when exiting idle mode.
- McBSPi.MCBSPLP\_WAKEUPEN\_REG[1] RFSREN bit: The McBSP module sends a McBSPi\_SWAKEUP request to the PRCM module when a receive frame-sync pulse is detected while the McBSP module is in idle mode. If the McBSPi.MCBSPLP\_IRQENABLE\_REG[1] RFSREN bit is set to 1, the McBSP module sends an interrupt (McBSPi\_IRQ) request to the MPU or DSP subsystems when exiting idle mode.
- McBSPi.MCBSPLP\_WAKEUPEN\_REG[0] RSYNCERREN bit: The McBSP module asserts the McBSPi\_SWAKEUP request when an unexpected receive frame-sync pulse is detected. If the McBSPi.MCBSPLP\_IRQENABLE\_REG[0] RSYNCERREN bit is set to 1, the McBSP module sends an interrupt (McBSPi\_IRQ) request to the MPU or DSP subsystems when exiting from idle mode (interrupt is asserted once the McBSPi.MCBSPLP\_IRQSTATUS\_REG[0] RSYNCERR bit changes from '0' to '1', indicating that a receive error occurred).



### 23.5.4.5.3.2 Transmit Wakeup

For transmit, there are also 5 possible wakeup configuration scenarios:

- McBSPi.MCBSPLP\_WAKEUPEN\_REG[14] XEMPTYEOFEN bit: The McBSP module asserts the McBSPi\_SWAKEUP request when a complete frame was transmitted and the transmit buffer is empty. If the McBSPi.MCBSPLP\_IRQENABLE\_REG[14] XEMPTYEOFEN bit is set to 1, the McBSP module sends an interrupt (McBSPi\_IRQ) request to the MPU or DSP subsystems when exiting from idle mode.
- McBSPi.MCBSPLP\_WAKEUPEN\_REG[10] XRDYEN bit: The McBSP module asserts the McBSPi\_SWAKEUP request when the transmit buffer reaches the high threshold value (XTHRESHOLD value + 1) of the McBSPi.MCBSPLP\_THRSH2\_REG register. If the McBSPi.MCBSPLP\_IRQENABLE\_REG[10] XRDYEN bit is set to 1, the McBSP module sends an interrupt (McBSPi\_IRQ) request to the MPU or DSP subsystems when exiting from idle mode (interrupt is asserted once the McBSPi.MCBSPLP\_IRQSTATUS\_REG[10] XRDY bit changes from '0' to '1', indicating that transmit buffer data is ready to accept new data).
- McBSPi.MCBSPLP\_WAKEUPEN\_REG[9] XEOFEN bit: The McBSP module asserts the McBSPi\_SWAKEUP request at the end of the frame. If the McBSPi.MCBSPLP\_IRQENABLE\_REG[9] XEOFEN bit is set to 1, the McBSP module sends an interrupt (McBSPi\_IRQ) request to the MPU or DSP subsystems when exiting from idle mode.
- McBSPi.MCBSPLP\_WAKEUPEN\_REG[8] XFSXEN bit: The McBSP module sends a McBSPi\_SWAKEUP request when a transmit frame-sync pulse is detected while the module is in idle mode. If the McBSPi.MCBSPLP\_IRQENABLE\_REG[8] XFSXEN bit is set to 1, the McBSP module sends an interrupt (McBSPi\_IRQ) request to the MPU or DSP subsystems when exiting from idle mode.
- McBSPi.MCBSPLP\_WAKEUPEN\_REG[7] XSYNCERREN bit: The McBSP module asserts the McBSPi\_SWAKEUP request when an unexpected transmits frame-sync pulse is detected. If the McBSPi.MCBSPLP\_IRQENABLE\_REG[7] XSYNCERREN bit is set to 1, the McBSP module sends an interrupt (McBSPi\_IRQ) request to the MPU or DSP subsystems when exiting from idle mode (interrupt will be asserted once the McBSPi.MCBSPLP\_IRQSTATUS\_REG[7] XSYNCERR bit changes from '0' to '1', indicating that a transmit error occurred).

### 23.5.4.5.3.3 Notes

When mcbbsp4\_fsr/mcbbsp\_i\_fsx pins is configured as an output, the FSR/FSX wakeup generation makes no sense (the module cannot be in Smart Idle mode).

Detection of RSYNCERR/XSYNCERR during idle mode can be used only when mcbbsp4\_fsr/mcbbsp\_i\_fsx pins is configured as an input and the remote system knows to assert such an error to trigger the wake up of the McBSP module.

The module does not implement interrupt request (IRQ) assertion when configured as (GPIO). Pins that can be used to accept input signals and/or send output signals but are not linked to specific uses; also a wake up capability in this mode is not available.

### 23.5.4.5.4 Analysis of the Receiver Smart Idle Behavior

The analysis of the power mode behavior is shown in [Table 23-333](#) :

In this table, the CLKRM bit is in the McBSPi.MCBSPLP\_PCR\_REG register on position 8, CLKXM bit in the McBSPi.MCBSPLP\_PCR\_REG[9] register, and CLOCKACTIVITY bit in the McBSPi.MCBSPLP\_SYSCONFIG\_REG[9:8] register.

The value X signifies that the bit value is not significant.

**Table 23-333. McBSP Smart Idle Mode Configuration Behavior**

CLKRM Bit	CLKXM Bit	McBSP Mode	Source of Functional Clock	CLOCKACTIVITY Bit	Behavior
0	0	Slave	outside	0bXX	The module acknowledges the idle request as soon as there is no pending DMA, interrupt request or transmit buffer threshold synchronization (only when wake-up event is set on transmit threshold reached), regardless of the CLOCKACTIVITY settings or receive and transmit activity.
0	1	Transmit Master	MCBSPi_ICLK	0b0X	The McBSP will not acknowledge the Idle request unless: <ul style="list-style-type: none"> <li>The transmit part is disabled (XDISABLE) or under software reset (XRST).</li> <li>Both transmit and receive parts are disabled (XDISABLE/RDISABLE) or under software reset (XRST/RRST)</li> </ul> The idle acknowledge is asserted as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached) and the pending transmit and/or receive frames were completed in case of transmit and/or receive disable.
			CLKS	0bX0	
			MCBSPi_ICLK	0b1X	The module acknowledges the idle request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached).
			CLKS	0bX1	
			CLKR (outside)	0bXX	The module acknowledges the idle request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached), regardless of the CLOCKACTIVITY settings.
			1	0	Receive master
CLKS	0bX0				
			MCBSPi_ICLK	0b1X	The module acknowledges the idle request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached).
			CLKS	0bX1	
			CLKX	0bXX	When CLKX is used as source (functional clock is provided from outside) then the module acknowledges the idle request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached), regardless of the CLOCKACTIVITY settings.

**Table 23-333. McBSP Smart Idle Mode Configuration Behavior (continued)**

CLKRM Bit	CLKXM Bit	McBSP Mode	Source of Functional Clock	CLOCKACTIVITY Bit	Behavior
1	1	Transmit and Receive Master	MCBSPi_ICLK	0b0X	The McBSP will not acknowledge the idle request unless: Both transmit and receive parts are disabled (XDISABLE/RDISABLE) or under software reset (XRST/RRST) The idle acknowledge is asserted as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached) and the pending transmit and/or receive frames were completed in case of transmit and/or receive disable. Note that no wake-up event is available in this mode since the entire McBSP and remote device activity is frozen.
			CLKS	0bX0	
			MCBSPi_ICLK	0b1X	The module acknowledges the idle request as soon as there is no pending DMA, interrupt request or transmit/receive buffer threshold synchronization (only when wake-up event is set on transmit/receive threshold reached).
			CLKS	0bX1	

**NOTE:** The RFSREN/XFSXEN mode is suitable for wakeup generation when both clocks (PRCM functional and interface) are switched off and `mcbasp4_fsr/mcbaspi_fsx` pins is configured as input. The frame-sync pulse is asynchronously detected during idle.

The RSYNCERREN/XSYNCERREN mode can be used to wakeup the McBSP module only by a remote module implementing such a feature, to trigger a wake up. This mode requires functional clock to be active.

#### 23.5.4.6 McBSP Interrupt Requests

Each of the four McBSP modules can generate one common interrupt (MCBSPi\_IRQ), shared between the Cortex-A9 MPU subsystem and DSP subsystem interrupt controllers.

An event can generate an interrupt request when the corresponding mask bit in the `McBSPi.MCBSPLP_IRQENABLE_REG` register is set to '1'.

Once an interrupt request is generated, the software must read the `McBSPi.MCBSPLP_IRQSTATUS_REG` register to check what event has caused the interrupt request generation, and acknowledge each processed event by writing a 1 to the corresponding bit in the `McBSPi.MCBSPLP_IRQSTATUS_REG` register.

#### 23.5.4.7 McBSP DMA Requests

The DMA requests are shared between the DSP subsystem DMA controller (eDMA), system DMA controller (sDMA) and Audio Engine DMA controller (ADMA). Each of the four McBSP modules can generate two DMA events:

- `MCBSPi_DMA_TX` : McBSPi module transmit DMA request
- `MCBSPi_DMA_RX` : McBSPi module receive DMA request

Table 23-330 summarizes the DMA events with the mapping on both DMA controllers.

The receive and transmit DMA requests can be individually disabled by setting the `McBSPi.MCBSPLP_RCCR_REG[3]` RDMAEN, `McBSPi.MCBSPLP_XCCR_REG[3]` XDMAEN bits to 0. When disabling the DMA, the DMA request line is de-asserted even if a DMA transfer is pending and the DMA state-machine is not reset.

See Section 23.5.4.11, *McBSP DMA Configuration* for more details.

### 23.5.4.8 McBSP Data Transfer Process

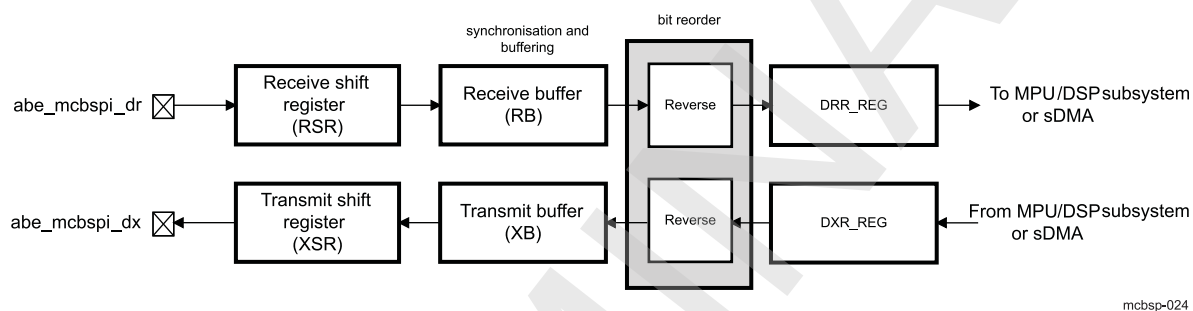
For McBSP modules, receive and transmit operations are triple-buffered (512 bytes buffers organized in 32-bit words are used).

All registers of McBSP data transfer paths are 32-bits wide. Figure 23-129 shows the McBSP data transfer paths.

**CAUTION**

The McBSP registers (DRR\_REG and DXR\_REG) are limited to 32-bit data accesses (L4 Interconnect). 16-bit and 8-bit is not allowed and can corrupt register content.

**Figure 23-129. McBSP Data Transfer Paths**



#### 23.5.4.8.1 Data Transfer Process for 8- / 12- / 16- / 20- / 24- / 32-bits Long Words

**CAUTION**

For each data word length, one data occupies one 32-bit buffer word.

Receive data arrives on the mcbspi\_dr pin and is shifted into the receive shift register (RSR). When a full word (depending on the data length configuration) is received, the content of the shift register is copied into the receive buffer (RB) if it is not full. When the RB threshold is reached the McBSP module asserts DMA or interrupt request and the RB content is then transferred (the sDMA or the eDMA controller reads the data receive register McBSPi.MCBSPLP\_DRR\_REG).

Transmit data is written by the MPU subsystem, the DSP subsystem or the DMA controller to data transmit register (McBSPi.MCBSPLP\_DXR\_REG) using the McBSPi.MCBSPLP\_SPCR2\_REG[1] XRDY bit enable input (when a byte is not enabled, the byte value in the memory will contain the previous written value). If there is no previous data in transmit shift register (XSR), the value from the transmit buffer (XB) is copied to XSR; otherwise, the content is copied to the XSR when the last bit of the previous data is shifted out on the mcbspi\_dx pin.

#### 23.5.4.8.2 Bit Reordering (Option to Transfer LSB First)

Generally, the McBSP module transmits or receives all data with the MSB first. However, some data protocols require the LSB to be transferred first.

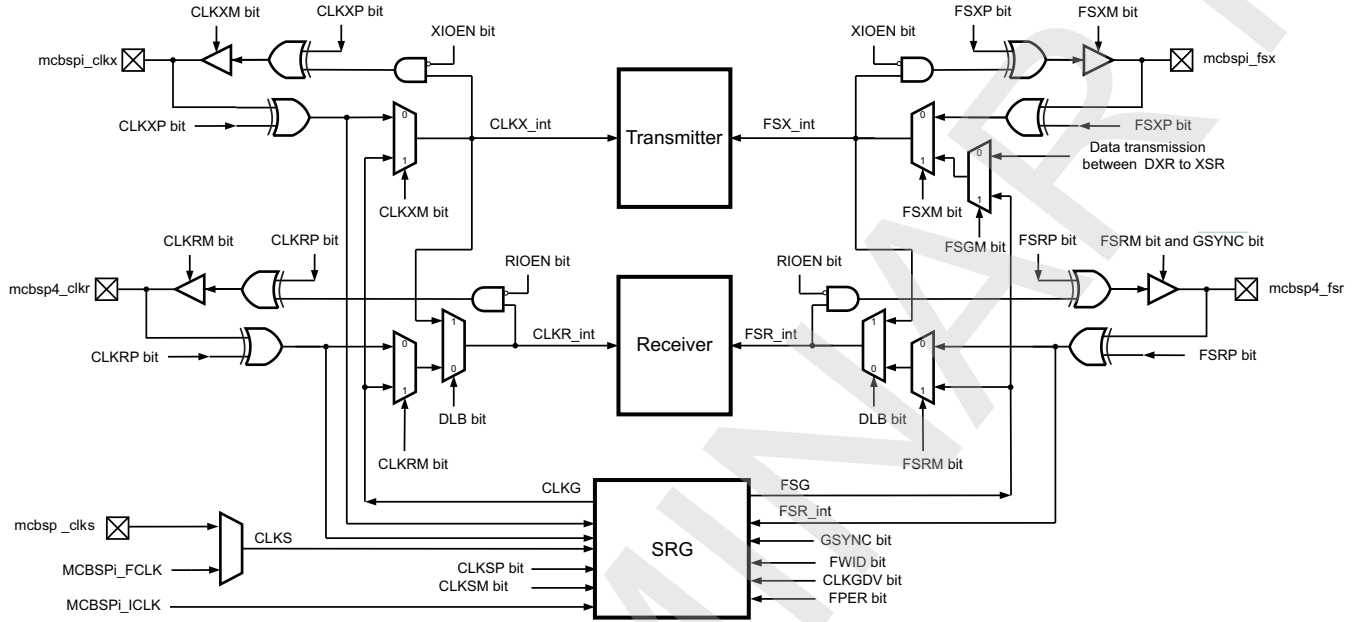
If you set McBSPi.MCBSPLP\_XCR2\_REG[4:3] XREVERSE=0b01, the bit ordering of the data words is reversed (LSB first) before being sent to the serial port. If you set McBSPi.MCBSPLP\_RCR2\_REG[4:3] RREVERSE=0b01, the bit ordering of the data words is reversed during reception (LSB first).

This feature is available for all the data formats from 8 up to 32-bit data length.

23.5.4.8.3 Clocking and Framing Data

This section explains basic concepts and terminology important for understanding how McBSP data transfers are timed and delimited.

Figure 23-130. Conceptual Block Diagram for Clock and Frame Generation



mcbasp-025

23.5.4.8.3.1 Clocking

Data is shifted one bit at a time from the mcbspi\_dr pin to the RSRs or from the XSRs to the mcbspi\_dx pin. The time for each bit transfer is controlled by the rising or falling edge of a clock signal.

The receive clock signal (CLKR\_int) controls bit transfers from the mcbspi\_dr pin to the RSRs. The transmit clock signal (CLKX\_int) controls bit transfers from the XSRs to the mcbspi\_dx pin. CLKR\_int or CLKX\_int signals can be derived from a pin at the boundary of the McBSP module (mcbasp4\_clkr and mcbspi\_clkx respectively) or derived from inside the McBSP module (see Figure 23-130). The clocks source is selected by programming the McBSPi.MCBSPLP\_PCR\_REG[9] CLKXM bit and the McBSPi.MCBSPLP\_PCR\_REG[8] CLKRM bit respectively.

When the McBSPi.MCBSPLP\_PCR\_REG[9] CLKXM bit (transmitter clock mode) is set to:

- '0', CLKX\_int is driven by an external clock and mcbspi\_clkx is an input pin.
- '1', CLKX\_int is driven by the internal sample rate generator and mcbspi\_clkx is an output pin.

For the McBSPi.MCBSPLP\_PCR\_REG[8] CLKRM bit (receiver clock mode), see Table 23-334.

Table 23-334. Receiver Clock Mode

Value	Digital loop back mode	mcbasp4_clkr pin	Description
0x0	McBSPi.MCBSPLP_XCCR_REG[5] DLB bit =0	input	CLKR_int is driven by an external clock
	McBSPi.MCBSPLP_XCCR_REG[5] DLB bit =1	High-impedance	CLKR_int is driven by CLKX_int. The CLKX_int is derived based on the CLKXM value.
0x1	McBSPi.MCBSPLP_XCCR_REG[5] DLB bit =0	output	CLKR_int is driven by the internal sample rate generator
	McBSPi.MCBSPLP_XCCR_REG[5] DLB bit =1	output	CLKR_int is driven by CLKX_int. The CLKX_int is derived based on the CLKXM value.

The polarities of CLKR and CLKX signals are configured in McBSPi.MCBSPLP\_PCR\_REG register.

The McBSPi.MCBSPLP\_PCR\_REG[1] CLKXP defines the transmit clock polarity:

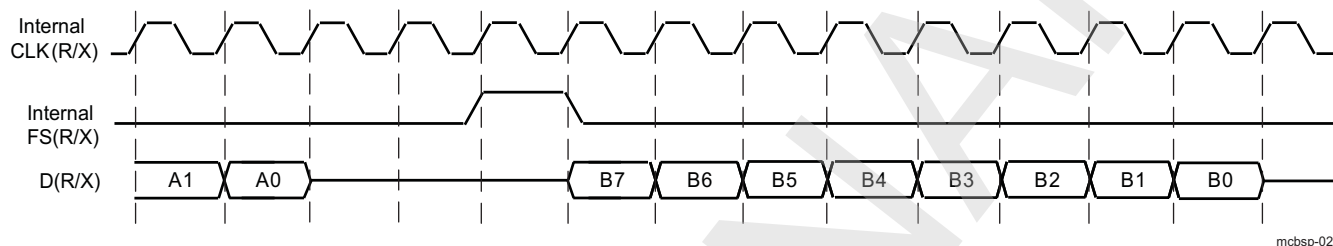
- When set to 0, transmit data driven on rising edge of CLKX signal
- When set to 1, transmit data driven on falling edge of CLKX signal

The McBSPi.MCBSPLP\_PCR\_REG[0] CLKRP defines the receive clock polarity:

- When set to 0, receive data sampled on falling edge of CLKX signal
- When set to 1, transmit data sampled on rising edge of CLKX signal

Figure 23-131 gives an example where the clock signal controls the timing of each bit transfer on the pin.

**Figure 23-131. Clock Signal Control of Bit Transfer Timing**



**NOTE:** The McBSP module is constrained to operate at an internal functional frequency of up to L4 interface frequency divided by 2. When driving CLKX or CLKR at the pin, choose an appropriate input clock frequency. When using the internal sample rate generator for CLKX/CLKR/CLKS, choose an appropriate input clock frequency (up to L4 interface frequency) and divide down value by programming the McBSPi.MCBSPLP\_SRGR1\_REG[7:0] CLKGDV bit field.

### 23.5.4.8.3.2 Serial Words

Bits traveling between a shift register (RSR or XSR) and a data pin (mcbspi\_dr or mcbspi\_dx) are transferred in a group called a serial word. The software defines how many bits are in a word by programming:

- For the receiver: the McBSPi.MCBSPLP\_RCR1\_REG[7:5] RWDLEN1 field and the McBSPi.MCBSPLP\_RCR2\_REG[7:5] RWDLEN2 field.
- For the transmitter: the McBSPi.MCBSPLP\_XCR1\_REG[7:5] XWDLEN1 field and the McBSPi.MCBSPLP\_XCR2\_REG[7:5] XWDLEN2 field.

The difference of use is explained to [Section 23.5.4.8.4.1, Number of Phases, Words, and Bits per Frame](#).

The various possibilities of word length are 8, 12, 16, 20, 24, and 32 bits (for field values, see [Section 23.5.6, McBSP Register Manual](#))

Bits coming from the mcbspi\_dr pin are held in the RSR until it holds a full serial word, then the word is passed to the RB and to the McBSPi.MCBSPLP\_DRR\_REG register.

During transmission, XSR accepts new data from XB after a full serial word has been passed from XSR to the mcbspi\_dx pin.

In the example in [Figure 23-134](#), an 8-bit word size was defined (see the transfer of the 8-bit word B).

### 23.5.4.8.3.3 Frames and Frame Synchronization

One or more words (up to 128) are transferred in a group called a frame. The software defines how many words are in a frame by programming:

- For the receiver: The McBSPi.MCBSPLP\_RCR1\_REG[14:8] RFRLLEN1 field and the McBSPi.MCBSPLP\_RCR2\_REG[14:8] RFRLLEN2 field.

- For the transmitter: The McBSPi.MCBSPLP\_XCR1\_REG[14:8] XFRLEN1 field and the McBSPi.MCBSPLP\_XCR2\_REG[14:8] XFRLEN2 field.

The difference between these registers is explained to [Section 23.5.4.8.4.1, Number of Phases, Words, and Bits per Frame](#). For the corresponding between field value and words number, see [Section 23.5.6, McBSP Register Manual](#).

All the words in a frame are sent in a continuous stream. However, there can be pauses between frame transfers. The McBSP module uses frame-synchronization signals (FSG) to determine when each frame is received/transmitted. When a pulse occurs on a frame-synchronization signal, the McBSP module begins receiving/transmitting a frame of data. When the next pulse occurs, the McBSP module receives/transmits the next frame, and so on.

Pulses on the receive frame-synchronization (FSR\_int) signal initiate frame transfers on mcbspi\_dr. Pulses on the transmit frame-sync (FSX\_int) signal initiate frame transfers on mcbspi\_dx. FSR\_int or FSX\_int signals can be derived from a pin at the boundary of the McBSP module (mcbbsp4\_fsr and mcbspi\_fsx respectively) or derived from inside the McBSP module (see [Figure 23-129](#)). The frame-sync source is selected by programming the McBSPi.MCBSPLP\_PCR\_REG[11] FSXM bit and the McBSPi.MCBSPLP\_PCR\_REG[10] FSRM bit respectively.

When the McBSPi.MCBSPLP\_PCR\_REG[11] FSXM bit (transmitter frame-sync mode) is set to:

- '0', FSX\_int is derived from an external source and mcbspi\_fsx is an input pin.
- '1', FSX\_int is determined by the McBSPi.MCBSPLP\_SRGR2\_REG[12] FSGM bit and mcbspi\_fsx is an output pin.

For the McBSPi.MCBSPLP\_PCR\_REG[10] FSRM bit (receiver frame-sync mode), is set to:

- '0', FSR\_int is generated by an external source and mcbbsp4\_fsr is an input pin
- '1', FSR\_int is generated internally by sample rate generator. The mcbbsp4\_fsx is an output pin except when McBSPi.MCBSPLP\_SRGR2\_REG[15] GSYNC bit = 0x1

In the example in [Figure 23-130](#), a one-word frame is transferred when a frame-synchronization pulse occurs. The polarities of FSR and FSX signals are programmable by bits on McBSPi.MCBSPLP\_PCR\_REG register.

The McBSPi.MCBSPLP\_PCR\_REG[3] FSXP defines the transmit frame-sync polarity:

- When set to '0', frame-sync pulse FSX is active high
- When set to '1', frame-sync pulse FSX is active low

The McBSPi.MCBSPLP\_PCR\_REG[2] FSRP defines the receive frame-sync polarity:

- When set to '0', frame-sync pulse FSR is active high
- When set to '1', frame-sync pulse FSR is active low

In McBSP operation, the inactive-to-active transition of the frame-synchronization signal indicates the start of the next frame. For this reason, the frame-synchronization signal may be high for an arbitrary number of clock cycles. Only after the signal is recognized to have gone inactive, and then active again, does the next frame synchronization occur.

#### **23.5.4.8.3.4 Detecting Frame-Synchronization Pulses, Even in Reset State**

The McBSP module can generate receive and transmit interrupts to the Cortex-A9 MPU/DSP subsystems to indicate specific events in the McBSP module. To facilitate detection of frame synchronization, these interrupts can be sent in response to frame-synchronization pulses (see [Section 23.5.5, McBSP Basic Programming Model](#) for further details).

Unlike other serial port interrupt modes, this mode can operate while the associated portion of the serial port is in reset (such as activating receive interrupt when the receiver is in reset). In this case, McBSPi.MCBSPLP\_PCR\_REG[0] FSRM bit/McBSPi.MCBSPLP\_PCR\_REG[1] FSXM bit and McBSPi.MCBSPLP\_PCR\_REG[2] FSRP bit/McBSPi.MCBSPLP\_PCR\_REG[3] FSXP bit still select the appropriate source and polarity of frame synchronization. Thus, even when the serial port is in the reset state, these signals are synchronized to the interface clock (MCBSPi\_ICLK) and then sent to the Cortex-A9 MPU/DSP subsystem in the form of receive interrupt and transmit interrupt at the point where they feed the receiver and transmitter of the serial port. Consequently, a new frame-synchronization pulse can be detected, and then, the Cortex-A9 MPU/DSP subsystem can take the serial port out of reset safely.

### 23.5.4.8.3.5 Ignoring Frame-Synchronization Pulses

The McBSP module ignores transmit and/or receive frame-synchronization pulses if the frame transfer was started by a previous frame-synchronization pulse (unexpected frame-synchronization pulses). The McBSP module does not support features like retransmit or re-receive of an erroneous frame or word. The receiver or transmitter ignores frame-synchronization pulses until the desired frame length or number of words is reached. For more details on unexpected frame-synchronization pulses, see [Section 23.5.4.10.3, Unexpected Receive Frame-sync Pulse](#) or [Section 23.5.4.10.6, Unexpected Transmit Frame-sync Pulse](#)

### 23.5.4.8.3.6 Frame Frequency

The frame frequency is determined by the period between frame-synchronization pulses and is defined as shown in the following equation:

$$\text{Frame frequency} = \text{Clock frequency} / (\text{Number of clock cycles between two rising edges [or falling edges] of two consecutive frame synchronization pulses})$$

The frame frequency can be increased by decreasing the time between frame-synchronization pulses (limited only by the number of bits per frame). As the frame transmit frequency increases, the inactivity period between the data packets for adjacent transfers decreases to zero.

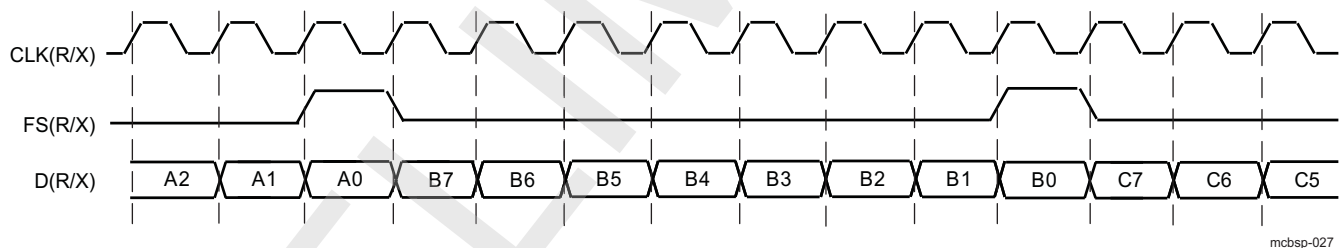
### 23.5.4.8.3.7 Maximum Frame Frequency

The minimum number of clock cycles between frame synchronization pulses is equal to the number of bits transferred per frame. The maximum frame frequency is defined as shown in the following equation:

$$\text{Maximum frame frequency} = \text{clock frequency} / \text{Number of bits per frame}$$

[Figure 23-132](#) below shows the McBSP operating at maximum packet frequency. At maximum packet frequency, the data bits in consecutive packets are transmitted contiguously with no inactivity between bits.

**Figure 23-132. McBSP Operating at Maximum Packet Frequency**



If there is a 1-bit data delay as shown in [Figure 23-132](#), the frame-synchronization pulse overlaps the last bit transmitted in the previous frame. Effectively, this permits a continuous stream of data, back-to-back transfers.

---

**NOTE:** For McBSPi.MCBSPLP\_XCR2\_REG[1:0] XDATDLY = 0x0 (0-bit data delay), the first bit of data is transmitted asynchronously to the internal transmit clock signal (CLKX\_int). For more details, see [Table 23-384](#).

---

### 23.5.4.8.4 Frame Phases (Dual-Phase Frame I2S Support)

The McBSP module allows you to configure each frame to contain one or two phases. The support for dual-phase frames is required to provide I2S fully compliant capabilities (audio left and right channels—stereo audio stream).



**CAUTION**

The limitation on dual-phase frame support is that the number of words per phase must be set to 1 for both first and second phase. It is the only possible value for word per frame when using the dual phase frame.

The number of bits per word can be specified differently for each of the two phases of a frame, allowing greater flexibility in structuring data transfers. For example, a user might define a frame as consisting of one phase containing one word of 16 bits, followed by a second phase consisting of one word of 32 bits. This configuration allows the user to compose frames for custom applications such as I2S protocol.

**23.5.4.8.4.1 Number of Phases, Words, and Bits per Frame**

Table 23-335 below shows which bit fields in the receive control registers (McBSPi.MCBSPLP\_RCR1\_REG and McBSPi.MCBSPLP\_RCR2\_REG) and in the transmit control registers (McBSPi.MCBSPLP\_XCR1\_REG and McBSPi.MCBSPLP\_XCR2\_REG) determine the number of phases per frame, the number of words per frame, and the number of bits per word for each phase, for both receiver and transmitter. The maximum number of words per frame is limited to 2 when using dual-phase frames (one word for each phase), and to 128 for a single-phase frame. The number of bits per word can be 8, 12, 16, 20, 24, or 32 bits.

The following legend applies to the table:

- RPHASE => McBSPi.MCBSPLP\_RCR2\_REG[15] RPHASE bit
- XPHASE => McBSPi.MCBSPLP\_XCR2\_REG[15] XPHASE bit
- RFLEN1 => McBSPi.MCBSPLP\_RCR1\_REG[14:8] RFLEN1 field
- RFLEN2 => McBSPi.MCBSPLP\_RCR2\_REG[14:8] RFLEN2 field
- XFLEN1 => McBSPi.MCBSPLP\_XCR1\_REG[14:8] XFLEN1 field
- XFLEN2 => McBSPi.MCBSPLP\_XCR2\_REG[14:8] XFLEN2 field
- RWDLEN1 => McBSPi.MCBSPLP\_RCR1\_REG[7:5] RWDLEN1 field
- RWDLEN2 => McBSPi.MCBSPLP\_RCR2\_REG[7:5] RWDLEN2 field
- XWDLEN1 => McBSPi.MCBSPLP\_XCR1\_REG[7:5] XWDLEN1 field
- XWDLEN2 => McBSPi.MCBSPLP\_XCR2\_REG[7:5] XWDLEN2 field

**Table 23-335. Phases, Words and Bits per Frame Control Bit**

Operation	Number of phases	Words per frame set with	Bits per word set with
Reception	1 (RPHASE=0)	RFLEN1	RWDLEN1
Reception	2 (RPHASE=1)	RFLEN1=0x0 and RFLEN2=0x0	RWDLEN1 for phase 1 RWDLEN2 for phase 2
Transmission	1 (XPHASE=0)	XFLEN1	XWDLEN1
Transmission	2 (XPHASE=1)	XFLEN1=0x0 and XFLEN2=0x0	XWDLEN1 for phase 1 XWDLEN2 for phase 2

**23.5.4.8.4.2 Single-Phase Frame Example**

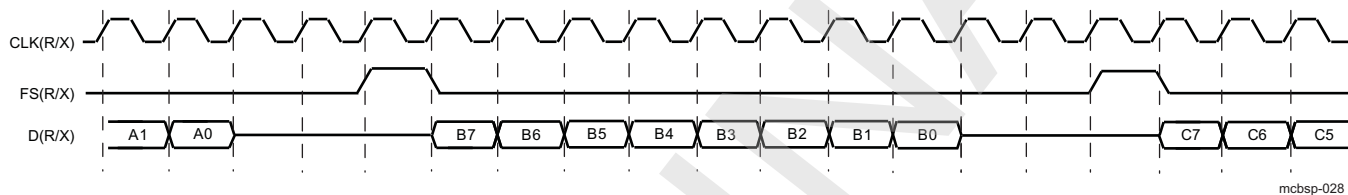
Figure 23-133 below shows an example of a single-phase data frame containing one 8-bit word. Because the transfer is configured for one data bit delay, the data on the mcbspi\_dx and mcbspi\_dr pins are available one clock cycle after FS(R/X) goes active. The following table shows the assumptions used in the example of this figure.

**Table 23-336. Assumptions for the Single-Phase Frame Example**

Assumption	Value	Bit or Field Name
Single-phase frame	'0'	McBSPi.MCBSPLP_RCR2_REG[15] RPHASE bit
		McBSPi.MCBSPLP_XCR2_REG[15] XPHASE bit
One word per frame	0x0	McBSPi.MCBSPLP_RCR1_REG[14:8] RFLEN1 field
		McBSPi.MCBSPLP_XCR1_REG[14:8] XFLEN1 field

**Table 23-336. Assumptions for the Single-Phase Frame Example (continued)**

Assumption	Value	Bit or Field Name
8-bit word length	0x0	McBSPi.MCBSPLP_RCR1_REG[7:5] RWDLEN1 field
		McBSPi.MCBSPLP_XCR1_REG[7:5] XWDLEN1 field
word length in register2	ignored	McBSPi.MCBSPLP_RCR2_REG[14:8] RFRLLEN2 field
		McBSPi.MCBSPLP_XCR2_REG[14:8] XWDLEN2 field
Receive data clocked on falling edge	'0'	McBSPi.MCBSPLP_PCR_REG[0] CLKRP bit
Transmit data clocked on rising edge		McBSPi.MCBSPLP_PCR_REG[1] CLKXP bit
Active-high frame-sync signals	'0'	McBSPi.MCBSPLP_PCR_REG[2] FSRP bit
		McBSPi.MCBSPLP_PCR_REG[3] FSXP bit
1-bit data delay	01b	McBSPi.MCBSPLP_RCR2_REG[1:0] RDATDLY field
		McBSPi.MCBSPLP_XCR2_REG[1:0] XDARDLY field

**Figure 23-133. Single-Phase Frame for a McBSP Data Transfer**

mcbasp-028

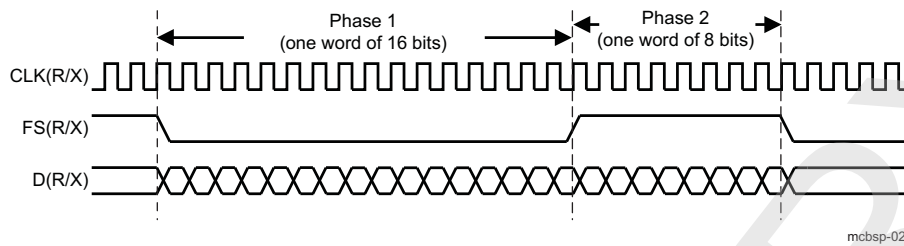
**23.5.4.8.4.3 Dual-Phase Frame Example**

Figure 23-134 below shows an example of a frame. The first phase consists of one word of 16 bits, followed by a second phase of one word of 8 bits. The entire bit stream in the frame is contiguous. There are no gaps between words/phases. Table 23-337 shows the assumptions used to the example in Figure 23-134.

**Table 23-337. Assumptions for the Dual-Phase Frame Example**

Assumption	Value	Bit or Field name
Single-phase frame	'1'	McBSPi.MCBSPLP_RCR2_REG[15] RPHASE bit
		McBSPi.MCBSPLP_XCR2_REG[15] XPHASE bit
One word per frame	0x0	McBSPi.MCBSPLP_RCR1_REG[14:8] RFRLLEN1 field
		McBSPi.MCBSPLP_XCR1_REG[14:8] XFRLEN1 field
16-bit word length	0x0	McBSPi.MCBSPLP_RCR1_REG[7:5] RWDLEN1 field
		McBSPi.MCBSPLP_XCR1_REG[7:5] XWDLEN1 field
8-bit word length	0x2	McBSPi.MCBSPLP_RCR2_REG[14:8] RFRLLEN2 field
		McBSPi.MCBSPLP_XCR2_REG[14:8] XWDLEN2 field
Receive data clocked on falling edge	'0'	McBSPi.MCBSPLP_PCR_REG[0] CLKRP bit
Transmit data clocked on rising edge		McBSPi.MCBSPLP_PCR_REG[1] CLKXP bit
Active-high frame-sync signals	'0'	McBSPi.MCBSPLP_PCR_REG[2] FSRP bit
		McBSPi.MCBSPLP_PCR_REG[3] FSXP bit
0-bit data delay	00b	McBSPi.MCBSPLP_RCR2_REG[1:0] RDATDLY field
		McBSPi.MCBSPLP_XCR2_REG[1:0] XDARDLY field

**Figure 23-134. Dual-Phase Frame for a McBSP Data Transfer**

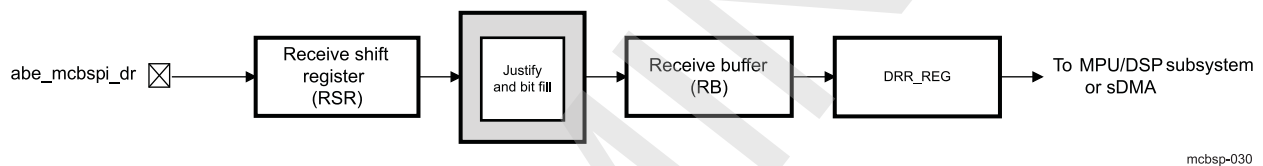


**23.5.4.8.5 McBSP Reception**

This section explains the fundamental process of reception in the McBSP module. For details about how to program the McBSP receiver, see [Section 23.5.5, McBSP Basic Programming Model](#), [Section 23.5.5.4, Interrupt Configuration](#), and [Section 23.5.5.5, Receiver Configuration](#).

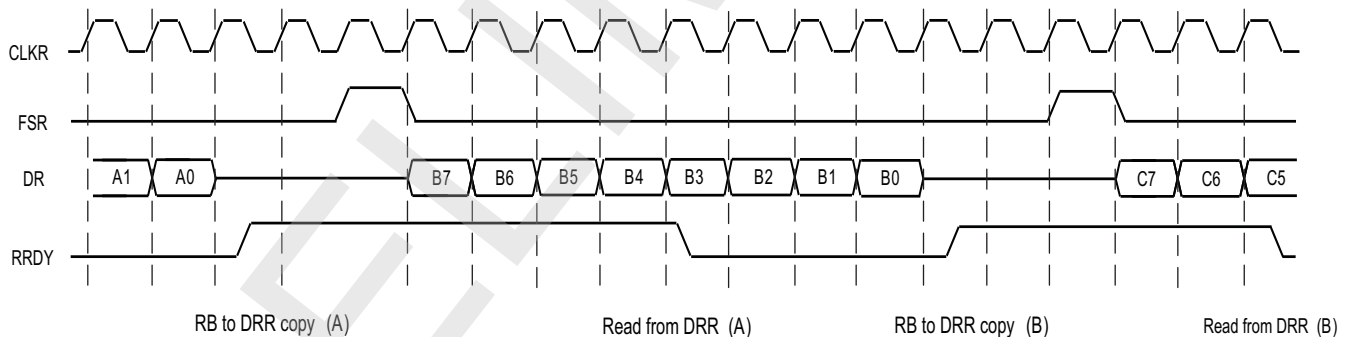
[Figure 23-135](#) and [Figure 23-136](#) below show how reception occurs in the McBSP module. A description of the process follows the figures. [Figure 23-135](#) shows the physical path for the data.

**Figure 23-135. McBSP Reception Physical Data Path**



[Figure 23-136](#) is a timing diagram showing signal activity for one possible reception scenario.

**Figure 23-136. McBSP Reception Signal Activity**



RRDY: Status of receiver ready bit (high is 1)

The following process describes how data travels from the mcbspi\_dr pin to the MPU/DSP subsystem or to the sDMA controller:

1. The McBSP module waits for a receive frame-synchronization pulse on FSR\_int.
2. When the pulse arrives, the McBSP module inserts the appropriate data delay that is selected with the McBSPi.MCBSPLP\_RCR2\_REG[1:0] RDATDLY bits. In the preceding timing diagram a 1-bit data delay is selected.
3. The McBSP module accepts data bits on the mcbspi\_dr pin and shifts them into the RSR. For details on choosing a word length, see [Section 23.5.5.6, Transmitter Configuration](#).
4. When a full word is received, the McBSP module copies the contents of the RSR to the RB, provided that RB is not full.
5. When the programmed receive threshold is reached (McBSPi.MCBSPLP\_THRSH1\_REG[6:0] RTHRESHOLD field), the McBSP module asserts the receiver ready bit

(McBSPi.MCBSPLP\_SPCR1\_REG[1] RRDY bit). This indicates that receive data is ready to be read by the Cortex-A9 MPU/DSP subsystem or the sDMA controller by accessing McBSPi.MCBSPLP\_DRR\_REG register.

The data copied from RB to McBSPi.MCBSPLP\_DRR\_REG is justified and bit filled according to the McBSPi.MCBSPLP\_SPCR1\_REG[14:13] RJUST field.

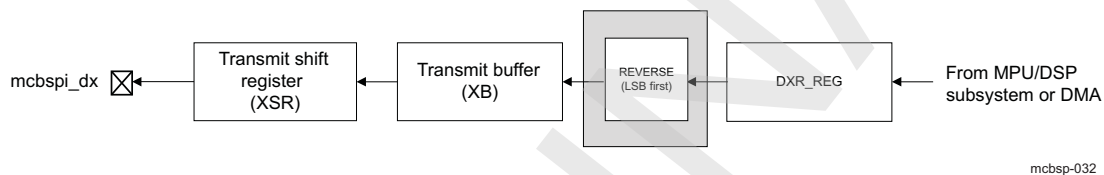
- The Cortex-A9 MPU/DSP subsystem or the sDMA controller reads the data from the data receive register. When the RB is empty, McBSPi.MCBSPLP\_SPCR1\_REG[1] RRDY bit is cleared.

#### 23.5.4.8.6 McBSP Transmission

This section explains the fundamental process of transmission in the McBSP module. For details about how to program the McBSP transmitter, see [Section 23.5.5, McBSP Basic Programming Model](#) and [Section 23.5.5.6, Transmitter Configuration](#).

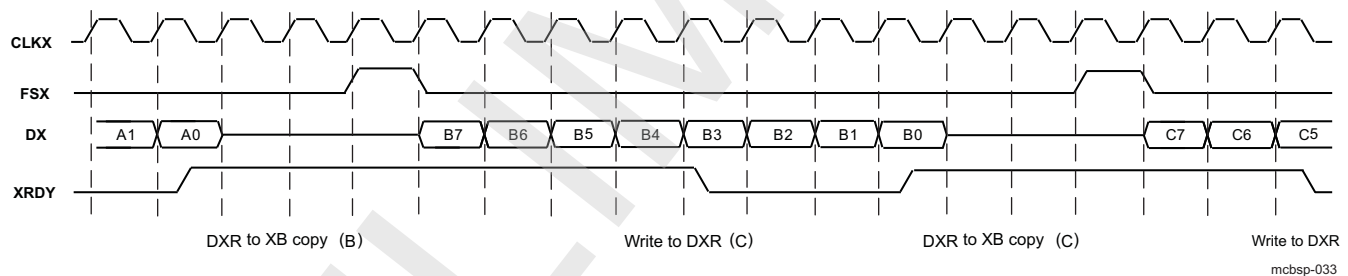
Figures below show how transmission occurs in the McBSP module. A description of the process follows the figures. [Figure 23-137](#) shows the physical path for the data.

**Figure 23-137. McBSP Transmission Physical Data Path**



[Figure 23-138](#) is a timing diagram showing signal activity for one possible transmission scenario.

**Figure 23-138. McBSP Transmission Signal Activity**



**XRDY: Status of transmitter ready bit (high is 1)**

- The Cortex-A9 MPU/DSP subsystem or the sDMA controller writes data to the data transmit register (McBSPi.MCBSPLP\_DXR\_REG). When the XB is reached the transmitter ready bit (McBSPi.MCBSPLP\_SPCR2\_REG[1] XRDY bit) is cleared to indicate that the transmitter is not ready for new data. For details on choosing a word length, see [Table 23-380](#).
- When new data arrives in McBSPi.MCBSPLP\_DXR\_REG register, the McBSP module copies the content of the data transmit register to the XB. In addition, the transmit ready bit (McBSPi.MCBSPLP\_SPCR2\_REG[1] XRDY bit) is set as long as the buffer contains at least the transmit threshold number of free locations (McBSPi.MCBSPLP\_THRSH2\_REG[6:0] XTHRESHOLD field). This indicates that the transmitter is ready to accept new data from the MPU/DSP subsystem or the sDMA controller.
- The McBSP module waits for a transmit frame-synchronization pulse on FSX\_int.
- When the pulse arrives, the McBSP module inserts the appropriate data delay that is selected with the McBSPi.MCBSPLP\_XCR2\_REG[1:0] XDADLY field.  
In the preceding timing diagram, a 1-bit data delay is selected.
- The McBSP module shifts data bits from the XSR to the mcbspi\_dx pin.

#### 23.5.4.8.7 Enable/Disable the Transmit and Receive Processes

The McBSP module has the option to stop-resume the transmit/receive process while the module is in functional mode (out of transmit/receive reset).

When the transmit/receive disable bit (McBSPi.MCBSPLP\_XCCR\_REG[0] XDISABLE/McBSPi.MCBSPLP\_RCCR\_REG[0] RDISABLE) is set, the McBSP module stops the transmit/receive operation at the next frame boundary (frame corruption avoided).

During the receive disable state, the frames that are sent (when FSR signal is asserted while receive disable) by the remote device are lost, and receive buffer overflow status bit (McBSPi.MCBSPLP\_IRQSTATUS\_REG[5] ROVFLSTAT) is not set. Also, the frames received by the remote device while McBSPi.MCBSPLP\_XCCR\_REG[0] XDISABLE bit is set (when FSX signal is asserted while transmit disable) are meaningless undefined data frames, and transmit buffer underflow status bit (McBSPi.MCBSPLP\_IRQSTATUS\_REG[11] XUNDFLSTAT) is not set. The presence of the frame synchronization, while transmit/receive process is disabled, can be checked by reading the transmit/receive Frame-sync interrupt status: McBSPi.MCBSPLP\_IRQSTATUS\_REG[8] XFSX / McBSPi.MCBSPLP\_IRQSTATUS\_REG[1] RFSR bits.

As soon as the McBSPi.MCBSPLP\_XCCR\_REG[0] XDISABLE/McBSPi.MCBSPLP\_RCCR\_REG[0] RDISABLE bit is cleared, the transmit/receive process resumes at the next frame boundary.

---

**NOTE:** It is not recommended to use this mechanism together with the possibility to interrogate the transmit/receive buffer status register (McBSPi.MCBSPLP\_XBUFFSTAT\_REG[7:0] XBUFFSTAT/McBSPi.MCBSPLP\_RBUFFSTAT\_REG[7:0] RBUFFSTAT field indicating the occupied/available buffer locations), since this register is an interface clock (MCBSPi\_ICLK) synchronous register and does not reflect the exact number of occupied/free locations available on the functional clock domain.

---

#### **23.5.4.8.8** *MCBSP Data Transfer Mode*

---

**NOTE:** For all examples in this section, the configured CLKX edge is the rising edge (McBSPi.MCBSPLP\_PCR\_REG[1] CLKXP bit = 0x0) and the configured CLKR edge is the falling edge (McBSPi.MCBSPLP\_PCR\_REG[0] CLKRP bit = 0x0). These are the reset values.

In timing diagrams below, a 1-bit data delay is selected (McBSPi.MCBSPLP\_RCR2\_REG[1:0] RDATDLY bit field = 0x01 and McBSPi.MCBSPLP\_XCR2\_REG[1:0] XDATDLY bit field = 0x01), because data often follows a 1-cycle active frame-synchronisation.

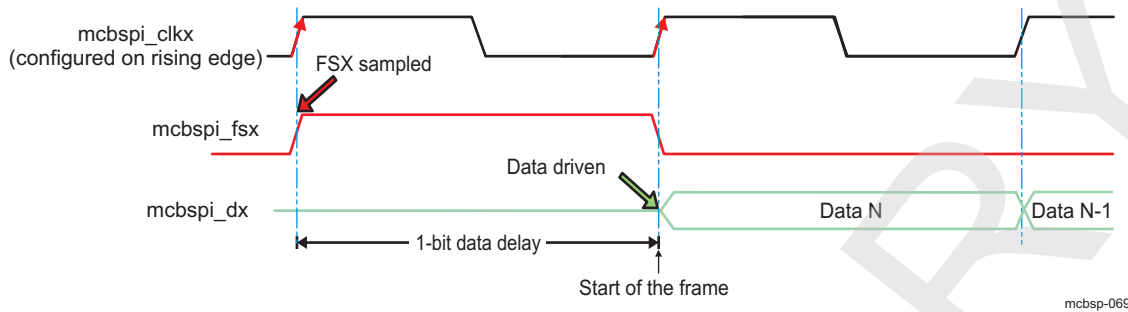
---

McBSP modules can support 2 edge selection modes for transmit and receive data transfer at the system level:

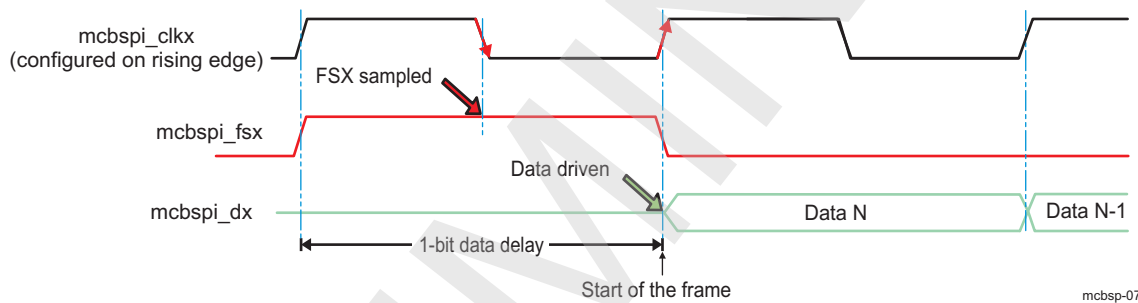
- The full cycle mode, for which one clock period is used to transfer the data, generated on one edge and captured on the same edge (one clock period later).
- The half cycle mode, for which one half clock period is used to transfer the data, generated on one edge and captured on the opposite edge (one half clock period later). Note that a new data is generated only every clock period, which permits to guarantee the required hold time.

#### **23.5.4.8.8.1** *Transmit Full Cycle Mode*

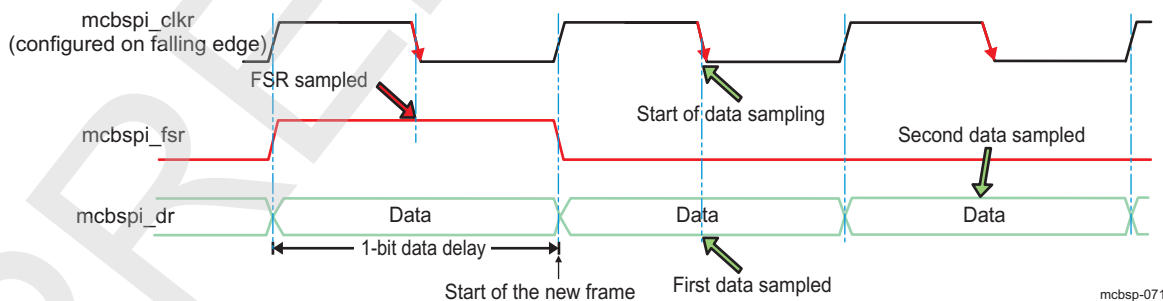
When configured in full cycle mode (McBSPi.MCBSPLP\_XCCR\_REG[11] XFULL\_CYCLE bit = 0x1), the FSX signal is sampled on the configured CLKX edge and the data is driven on the same configured edge. See [Figure 23-139](#).

**Figure 23-139. Transmit Full Cycle Timing Diagram****23.5.4.8.8.2 Transmit Half Cycle Mode**

When configured in half cycle mode (McBSPi.MCBSPLP\_XCCR\_REG[11] XFULL\_CYCLE bit = 0x0, reset value), the FSX signal is sampled on the opposite configured CLKX edge and the data is driven on the next configured edge. See [Figure 23-140](#).

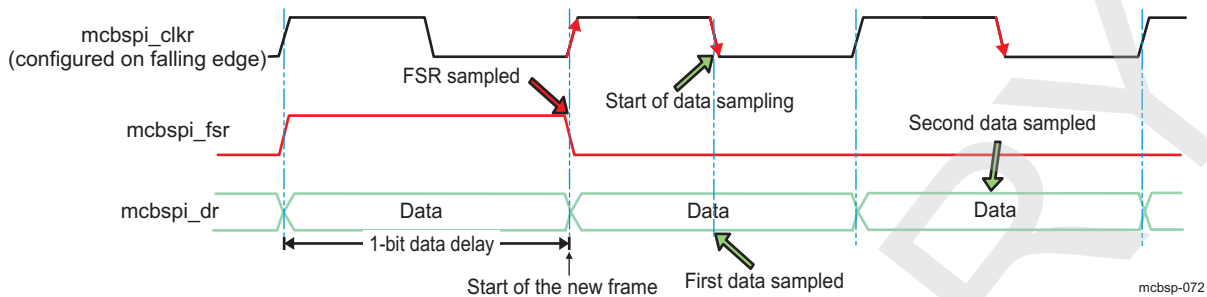
**Figure 23-140. Transmit Half Cycle Timing Diagram****23.5.4.8.8.3 Receive Full Cycle Mode**

When configured in full cycle mode (McBSPi.MCBSPLP\_RCCR\_REG[11] RFULL\_CYCLE bit = 0x1, reset value), the FSR signal is sampled on the configured CLKR edge and the data is driven on the same configured edge. See [Figure 23-141](#).

**Figure 23-141. Receive Full Cycle Timing Diagram****23.5.4.8.8.4 Receive Half Cycle Mode**

When configured in half cycle mode (McBSPi.MCBSPLP\_RCCR\_REG[11] RFULL\_CYCLE bit = 0x0), the FSR signal is sampled on the opposite configured CLKR edge and the data is driven on the next configured edge. See [Figure 23-142](#).

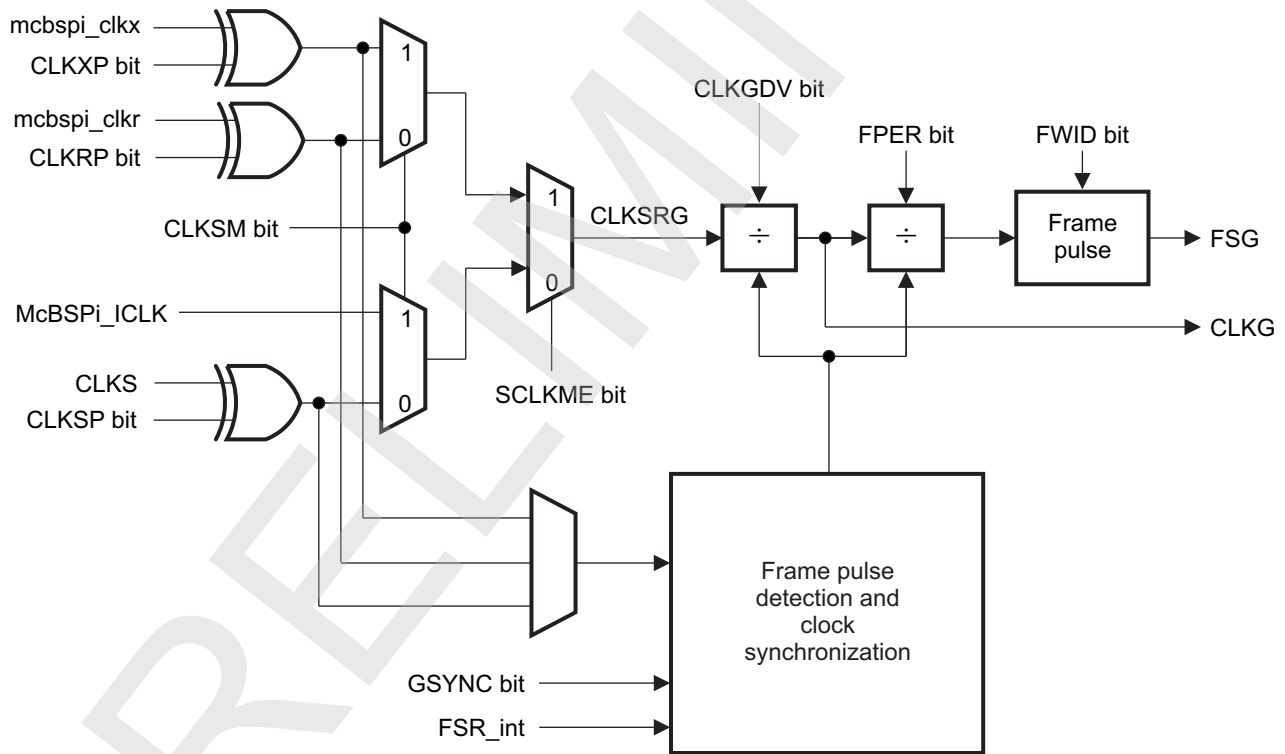
Figure 23-142. Receive Half Cycle Timing Diagram



23.5.4.9 McBSP SRG

The McBSP module contains an internal SRG that can be used to generate an internal data clock (CLKG) and an internal frame-synchronization signal (FSG). CLKG can be used for bit shifting on the data receive pin (mcbspi\_dr) and/or the data transmit pin (mcbspi\_dx). FSG can be used to initiate frame transfers on mcbspi\_dr pin and/or mcbspi\_dx pin. Figure 23-143 is a conceptual block diagram of the SRG.

Figure 23-143. Conceptual Block Diagram of the Sample Rate Generator



The source clock for the SRG (labeled **CLKSRG** in the diagram) can be supplied by either the interface clock (**McBSPi\_ICLK**), or the functional clock (**CLKS** input), or by an external pin (**mcbspi\_clkx**, or **mcbbsp4\_clkr**). The source is selected with the **McBSPi.MCBSPLP\_PCR\_REG[7]** **SCLKME bit** and the **McBSPi.MCBSPLP\_SRGR2\_REG[13]** **CLKSM bit**.

If a pin or **CLKS** signal is used, the polarity of the incoming signal can be inverted with the appropriate polarity bit (**McBSPi.MCBSPLP\_SRGR2\_REG[14]** **CLKSP bit**, **McBSPi.MCBSPLP\_PCR\_REG[1]** **CLKXP bit**, or **McBSPi.MCBSPLP\_PCR\_REG[0]** **CLKRP bit**).

The SRG has a three-stage clock divider that gives **CLKG** and **FSG** programmability.

The three stages provide:

- Clock divide-down: The source clock (CLKSRG) is divided according to the McBSPi.MCBSPLP\_SRGR1\_REG[7:0] CLKGDV field to produce CLKG signal
- Frame period divide-down: CLKG is divided according to the McBSPi.MCBSPLP\_SRGR2\_REG[11:0] FPER field to control the period from the start of a frame-pulse to the start of the next pulse
- Frame-synchronization pulse-width countdown: CLKG cycles are counted according to the McBSPi.MCBSPLP\_SRGR1\_REG[15:8] FWID field to control the width of each frame-synchronization pulse

---

**NOTE:** The McBSP module cannot operate at an internal functional frequency faster than L4 interface frequency divided by 2. Choose an input clock frequency and a McBSPi.MCBSPLP\_SRGR1\_REG[7:0] CLKGDV value such that CLKG is less than or equal to L4 interface frequency divided by 2.

---

In addition to the three-stage clock divider, the sample rate generator has a frame-synchronization pulse detection and clock synchronization module that allows synchronization of the clock divide down with an incoming frame-synchronization pulse on the mcbbsp4\_fsr pin. This feature is enabled or disabled with the McBSPi.MCBSPLP\_SRGR2\_REG[15] GSYNC bit.

CLKG is used as source to generate the output clocks CLKX/CLKR when the McBSPi.MCBSPLP\_PCR\_REG[9] CLKXM / McBSPi.MCBSPLP\_PCR\_REG[8] CLKRM bit indicates that the clock is an output. The output CLKX/CLKR is generated according to the clock polarity setting (see [Figure 23-130](#)).

For details on getting the sample rate generator ready for operation, see [Section 23.5.5, McBSP Basic Programming Model](#).

#### 23.5.4.9.1 Clock Generation in the SRG

The SRG can produce a clock signal (CLKG) for use by the receiver, the transmitter, or both. Use of the SRG to drive clocking is controlled by the clock mode bits (McBSPi.MCBSPLP\_PCR\_REG[9] CLKXM and McBSPi.MCBSPLP\_PCR\_REG[8] CLKRM) and polarity mode bits (McBSPi.MCBSPLP\_PCR\_REG[1] CLKXP and McBSPi.MCBSPLP\_PCR\_REG[0] CLKRP).

When a clock mode bit is set to 1 (CLKRM=1 for reception, CLKXM=1 for transmission), the corresponding data clock (CLKR for reception, CLKX for transmission) is driven by the internal SRG output clock (CLKG) according to the polarity setting.

The effects of this setting on the McBSP module are partially affected by the use of the digital loopback (DLB) mode, the analog loop back (ALB) mode and by the synchronous receive/transmit setting, respectively, as described in [Table 23-338](#). The ALB mode is selected with the McBSPi.MCBSPLP\_SPCR1\_REG[15] ALB bit. The DLB mode is selected with the McBSPi.MCBSPLP\_XCCR\_REG[5] DLB bit. The synchronous setting is controlled by input signals. These signals are defined by the control registers of the Control Module (more details, see [Chapter 18, Control Module](#)).

When using the SRG as a clock source, make sure the SRG is enabled (McBSPi.MCBSPLP\_SPCR2\_REG[6] GRST bit =1).



**Table 23-338. Effects of DLB and ALB Bits on Clock Modes**

Mode Bit Settings		Effect
CLKRM=1	DLB=0 and ALB = 0 (Digital and analog loop back mode disabled)	mcbbsp1_clkx is an output pin driven by the SRG output clock (CLKG).
	DLB=0 and ALB = 1 (Digital loop back mode disabled and Analog loop back mode enabled)	mcbbsp1_clkx is an output pin driven by the SRG output clock (CLKG). The receiver functional part internal clock is driven by CLKX input signal provided by mcbbsp1_clkx pin. The source of CLKX depends on the CLKXM bit. The receive frame synchronization is driven by FSX input signal provided by mcbbsp1_fsx pin. The receive data is driven by the DX input loop-back pin (mcbbsp1_dx).
	DLB=1 & ALB = 0 (Digital loop back mode enabled and Analog loop back mode disabled)	The SRG and the frame synchronization generator must be enabled. The internal transmit and receive clocks are driven by the SRG (CLKG having the appropriate CLKXP polarity). The transmit and receive frame synchronization signals are driven by FSG (having the appropriate FSXP polarity). The transmit data is connected to the DR input data. Note that in digital loop back mode no serial link activity will be seen by the remote device.
	DLB=1 & ALB = 1 (reserved mode)	Undefined functionality.
CLKXM=1	DLB=0 & ALB = 0 (Digital & analog loop back mode disabled)	mcbbsp1_clkx is an output pin driven by the SRG output clock (CLKG).
	DLB=0 & ALB = 1 (Digital loop back mode disabled and Analog loop back mode enabled)	mcbbsp1_clkx is an output pin driven by the SRG output clock (CLKG).
	DLB=1 & ALB = 0 (Digital loop back mode enabled and Analog loop back mode disabled)	The SRG and the frame synchronization generator need to be enabled. The internal transmit and receive clocks are driven by the SRG (CLKG having the appropriate CLKXP polarity). The transmit and receive frame synchronization signals are driven by FSG (having the appropriate FSXP polarity). The transmit data is connected to the DR input data. Note that in digital loop back mode no serial link activity will be seen by the remote device.
	DLB=1 & ALB = 1 (reserved mode)	Undefined functionality.
	CONTROL_MCBSPPLP[30] ALBCTRLRX_CLKX bit =1 (synchronous setting and DLB = 0 & ALB = 0)	CLKX is an output pin driven by the SRG output clock (CLKG). CLKR is connected to the CLKX.

#### 23.5.4.9.2 Frame Sync Generation in the SRG

The SRG can produce a frame-synchronization signal (FSG) for use by the receiver, the transmitter, or both.

If you want the receiver to use FSG for frame synchronization, make sure McBSPi.MCBSPPLP\_PCR\_REG[10] FSRMbit =1. (When FSRM=0, receive frame synchronization is supplied via the mcbbsp4\_fsr pin.)

If you want the transmitter to use FSG for frame synchronization, you must set:

- McBSPi.MCBSPPLP\_PCR\_REG[11] FSXM = 1: This indicates that transmit frame synchronization is supplied by the McBSP module itself rather than from the mcbbsp1\_fsx pin.
- MCBSPPLP\_SRGR2\_REG[12] FSGM=1: This indicates that when FSXM=1, transmit frame synchronization is supplied by the SRG.

**NOTE:** When FSGM=0 and FSXM=1, the transmit frame-sync signal (FSX) is generated when XB is not empty. When FSGM = 0, McBSPi.MCBSPLP\_SRGR2\_REG[11:0] FPER and McBSPi.MCBSPLP\_SRGR1\_REG[15:8] FWID field are used to determine the frame synchronization period and width (external FSX is gated by the buffer empty condition).

In either case, the SRG must be enabled (McBSPi.MCBSPLP\_SPCR2\_REG[6] GRST bit=1) and the frame-synchronization logic in the SRG must be enabled (McBSPi.MCBSPLP\_SPCR2\_REG[7] FRST bit=0).

#### 23.5.4.9.2.1 Choosing the Width of the Frame-sync Pulse

Each pulse on FSG has a programmable width. You program the McBSPi.MCBSPLP\_SRGR1\_REG[15:8] FWID field, and the resulting pulse width is (FWID+1)CLKGcycles, where CLKG is the output clock of the SRG. The range is from 1 to 256 clock periods.

#### 23.5.4.9.2.2 Controlling the Period Between the Starting Edges of Frame Sync Pulses

You can control the amount of time from the starting edge of one FSG pulse to the starting edge of the next FSG pulse. This period is controlled in one of two ways, depending on the configuration of the SRG:

- If the SRG is using an external input clock and McBSPi.MCBSPLP\_SRGR2\_REG[15] GSYNCbit =1, FSG pulses in response to an inactive-to-active transition on the mcbasp4\_fsr pin. Thus, an external device controls the frame-synchronization period.
- Otherwise, the software program the McBSPi.MCBSPLP\_SRGR2\_REG[11:0] FPER field, and the resulting frame-synchronization period is (FPER+1)CLKGcycles, where CLKG is the output clock of the SRG. The range is from 1 to 4096 clock periods.

#### 23.5.4.9.2.3 Keeping FSG Synchronized to an External Clock

When an external signal is selected to drive the SRG, the McBSPi.MCBSPLP\_SRGR2\_REG[15] GSYNC bit and the mcbasp4\_fsr pin can be used to configure the timing of FSG pulses.

McBSPi.MCBSPLP\_SRGR2\_REG[15] GSYNC=1 ensures that the McBSP module and an external device are dividing down the input clock with the same phase relationship.

If McBSPi.MCBSPLP\_SRGR2\_REG[15] GSYNC=1, an inactive-to-active transition on the mcbasp4\_fsr pin triggers a resynchronization of CLKG and generation of FSG.

#### 23.5.4.9.3 Synchronizing SRG Outputs to an External Clock

The SRG can produce a clock signal (CLKG) and a FSG based on an input clock signal that is either the interface clock signal (MCBSPi\_ICLK), or the CLKS signal (PRCM functional clock or mcbasp\_clks), or a signal at the mcbasp4\_clkr, or mcbspi\_clkx pin. When an external clock (mcbasp\_clks, or mcbasp4\_clkr, or mcbspi\_clkx) is selected to drive the SRG, the McBSPi.MCBSPLP\_SRGR2\_REG[15] GSYNC bit and the mcbasp4\_fsr pin can be used to control the timing of CLKG and the pulsing of FSG relative to the chosen input clock. Make GSYNC=1 so that the McBSP module and an external device divide down the input clock with the same phase relationship.

If the McBSPi.MCBSPLP\_SRGR2\_REG[15] GSYNC bit=1:

- An inactive-to-active transition on the mcbasp4\_fsr pin triggers a resynchronization of CLKG signal and a pulsing of FSG signal.
- CLKG signal always begins with a high state after synchronization.
- FSR signal is always detected at the same edge of the input clock signal that generates CLKG signal, no matter how long the FSR pulse is.
- The McBSPi.MCBSPLP\_SRGR2\_REG[11:0] FPER field are ignored because the frame-synchronization period on FSG is determined by the arrival of the next frame-synchronization pulse on the mcbasp4\_fsr pin.

If the McBSPi.MCBSPLP\_SRGR2\_REG[15] GSYNC bit=0, CLKG signal runs freely and is not resynchronized, and the frame-synchronization period on FSG signal is determined by McBSPi.MCBSPLP\_SRGR2\_REG[11:0] FPER field.

**23.5.4.9.3.1 Operating the Transmitter Synchronously with the Receiver**

When the `McBSPi.MCBSPLP_SRGR2_REG[15]` GSYNC bit=1, the transmitter can operate synchronously with the receiver, provided that the FSX signal is programmed to be driven by FSG signal (`McBSPi.MCBSPLP_SRGR2_REG[12]` FSGM=1 and `McBSPi.MCBSPLP_PCR_REG[11]` FSXM='1'). If the FSR input signal has appropriate timing so that it can be sampled by the falling edge of CLKG signal, it can be used, instead, by setting `McBSPi.MCBSPLP_PCR_REG[11]` FSXM=0 and connecting FSR signal to FSX externally.

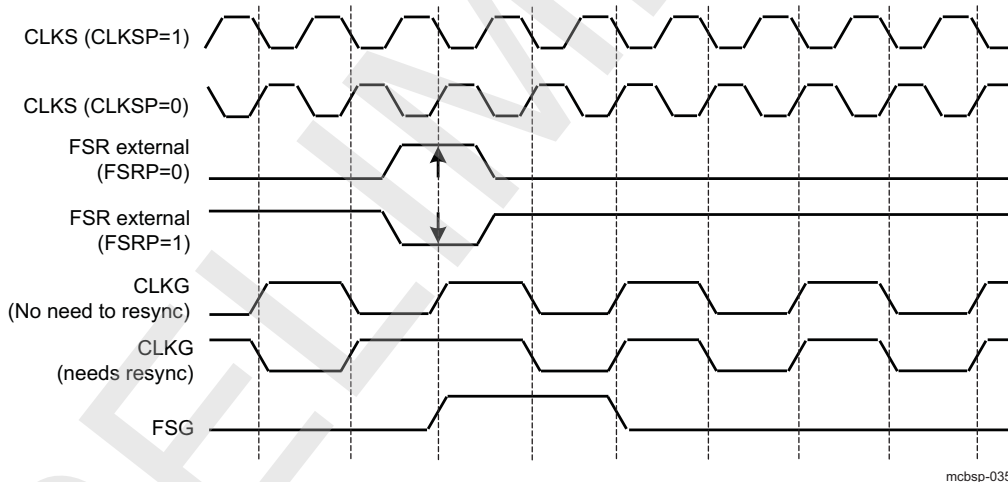
The SRG clock drives the transmit and receive clocking (`McBSPi.MCBSPLP_PCR_REG[8]` CLKRMbit and `McBSPi.MCBSPLP_PCR_REG[9]` CLKXM bit are set to '1'). Therefore, the CLK(R/X) pin must not be driven by any other driving source.

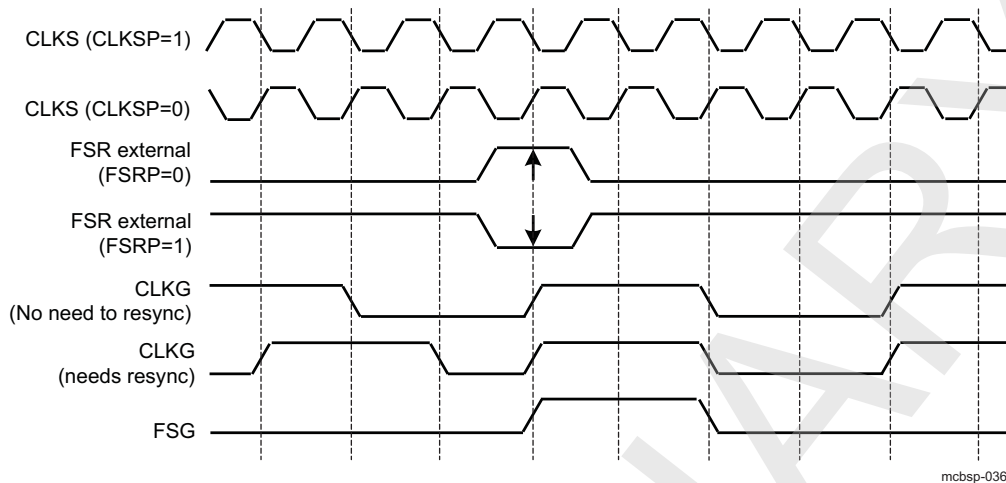
**23.5.4.9.3.2 Synchronization Examples**

Figure 23-144 and Figure 23-145 show the clock and frame-synchronization operation with various polarities of CLKS (the chosen input clock) and FSR signals. These figures assume `McBSPi.MCBSPLP_SRGR1_REG[15:8]` FWID = 0x0, for an FSG pulse that is oneCLKGcycle wide. The `McBSPi.MCBSPLP_SRGR2_REG[11:0]` FPER field are not programmed; the period from the start of a frame-synchronization pulse to the start of the next pulse is determined by the arrival of the next inactive-to-active transition on the `mcbasp4_fsr` pin.

Each figure shows what happens to CLKG signal when the `McBSPi.MCBSPLP_SRGR2_REG[15]` GSYNC bit = 1 and if it is initially synchronized or if it is not initially synchronized. Figure 23-145 has a slower CLKG frequency (it has a larger divide-down value in the `McBSPi.MCBSPLP_SRGR1_REG[7:0]` CLKGDV field).

**Figure 23-144. CLKG Synchronization and FSG Generation (GSYNC = 1 and CLKGDV = 0x1)**



**Figure 23-145. CLKG Synchronization and FSG Generation (GSYNC = 1 and CLKGDV = 0x3)**

### 23.5.4.10 McBSP Exception/Error Conditions

#### 23.5.4.10.1 Introduction

There are several serial port events that can constitute a system error. Any error conditions can be a source of an interrupt:

- Receiver overrun (McBSPi.MCBSPLP\_IRQSTATUS\_REG[5] ROVFLSTAT bit is set to '1', and legacy mode McBSPi.MCBSPLP\_SPCR1\_REG[2] RFULLbit is set to '1')

This occurs when RB is full and RSR are full with another new word shifted in from mcbspi\_dr. Therefore, McBSPi.MCBSPLP\_IRQSTATUS\_REG[5] ROVFLSTAT (McBSPi.MCBSPLP\_SPCR1\_REG[2] RFULL) indicates an error condition wherein any new data that can arrive at this time on mcbspi\_dr replaces the contents of the RSR, and the previous word is lost. The RSR continues to be overwritten as long as new data arrives on mcbspi\_dr and McBSPi.MCBSPLP\_DRR\_REG register is not read. For more details about overrun in the receiver, see [Section 23.5.4.10.2, Overrun in the Receiver](#).
- Unexpected receive frame-synchronization pulse (McBSPi.MCBSPLP\_IRQSTATUS\_REG[0] RSYNCERRbit is set to '1', and legacy mode McBSPi.MCBSPLP\_SPCR1\_REG[3] RSYNCERR bit is set to '1')

This occurs during reception when an unexpected frame-synchronization pulse arrives. An unexpected frame-synchronization pulse is one that is supposed to begin the next frame transfer before all the bits of the current frame have been received. Such a pulse is ignored by the receiver, but sets the McBSPi.MCBSPLP\_SPCR1\_REG[3] RSYNCERR bit. For more details about receive frame-synchronization errors, see [Section 23.5.4.10.3, Unexpected Receive Frame-Sync Pulse](#).
- Receiver underflow (McBSPi.MCBSPLP\_IRQSTATUS\_REG[4] RUNDLSTAT bit is set to '1')

This occurs when sDMA controller or MPU/DSP subsystem reads data from an empty receive buffer. For more details about underflow in the receiver, see [Section 23.5.4.10.4, Underflow in the Receiver](#).
- Transmitter underflow (McBSPi.MCBSPLP\_IRQSTATUS\_REG[11] XUNDLSTAT bit is set to '1', and legacy mode McBSPi.MCBSPLP\_SPCR2\_REG[2] XEMPTY bit is set to '0')

If a new frame-synchronization signal arrives when XB is empty, the previous data in the XSR is sent again. This procedure continues for every new frame-synchronization pulse that arrives until McBSPi.MCBSPLP\_DXR\_REG register is loaded with new data (and the XB is no longer empty). For more details about underflow in the transmitter, see [Section 23.5.4.10.5, Underflow in the Transmitter](#).
- Unexpected transmit frame-synchronization pulse (McBSPi.MCBSPLP\_IRQSTATUS\_REG[7] XSYNCERRbit is set to '1', and legacy mode McBSPi.MCBSPLP\_SPCR2\_REG[3] XSYNCERRbit is set to '1')

This occurs during transmission when an unexpected frame-synchronization pulse arrives. An unexpected frame-synchronization pulse is one that is supposed to begin the next frame transfer before all the bits of the current frame have been transferred. Such a pulse is ignored by the

transmitter, but sets the McBSPi.MCBSPLP\_SPCR2\_REG[3] XSYNCERR bit. For more details see [Section 23.5.4.10.6, Unexpected Transmit Frame-sync Pulse](#).

- Transmitter overflow (McBSPi.MCBSPLP\_IRQSTATUS\_REG[12] XOVLSTAT bit is set to '1') This occurs when sDMA controller or MPU/DSP subsystem writes data to a full XB. For more details about underflow in the receiver, see [Section 23.5.4.10.7, Overflow in the Transmitter](#).

### 23.5.4.10.2 Overrun in the Receiver

When McBSPi.MCBSPLP\_IRQSTATUS\_REG[5] ROVFLSTAT bit set to '1', and McBSPi.MCBSPLP\_SPCR1\_REG[2] RFULL bit set to '1' (legacy mode) indicates that the receiver has experienced overrun and is in an error condition. Receive overrun is set when all of the following conditions are met:

1. McBSPi.MCBSPLP\_DRR\_REG is not read even if the McBSPi.MCBSPLP\_IRQSTATUS\_REG[3] RRDY bit is set (legacy mode) and DMA or interrupt request has been asserted.
2. RB is full
3. RSR is full

As previously described, data arriving on mcbspi\_dr is continuously shifted into the Receive Shift Register (RSR). Once a complete word is shifted into the RSR, an RSR-to-RB copy can occur only if the RB is not full.

Either of the following events clears the legacy mode McBSPi.MCBSPLP\_SPCR1\_REG[2] RFULL bit and allows subsequent transfers to be read properly:

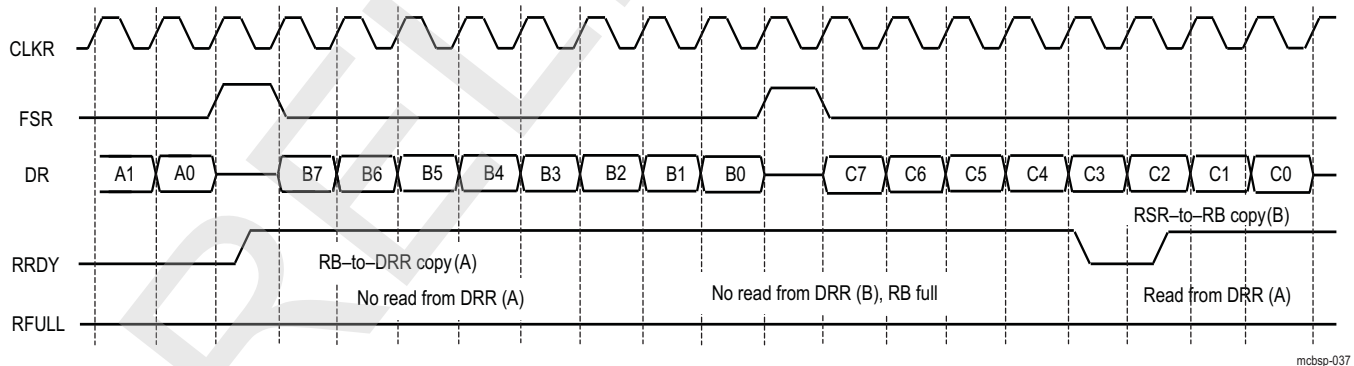
- The MPU/DSP subsystems or sDMA controller reads McBSPi.MCBSPLP\_DRR\_REG.
- The receiver is reset individually (McBSPi.MCBSPLP\_SPCR1\_REG[0] RRSTbit =0) or as part of a global reset.

Another frame-synchronization pulse is required to restart the receiver.

According to the McBSPi.MCBSPLP\_IRQENABLE\_REG register setting, this condition can generate the McBSPi\_IRQ line to be asserted low. Writing 1 to the corresponding bit in McBSPi.MCBSPLP\_IRQSTATUS\_REG register clears the interrupt.

Figure 23-146 shows the receive overrun condition.

Figure 23-146. Overrun in the McBSP Receiver



### 23.5.4.10.3 Unexpected Receive Frame-sync Pulse

#### 23.5.4.10.3.1 Possible Responses to Receive Frame-sync Pulses

If a frame-synchronization pulse starts the transfer of a new frame before the current frame is fully received, this pulse is treated as an unexpected frame-synchronization pulse, and the receiver sets the receive frame-synchronization error bit McBSPi.MCBSPLP\_IRQSTATUS\_REG[0] RSYNCERR (and the legacy McBSPi.MCBSPLP\_SPCR1\_REG[3] RSYNCERR bit).

According to the `McBSPi.MCBSPLP_IRQENABLE_REG` register settings this condition can generate the `McBSPi_IRQ` line to be asserted low. Writing 1 to the corresponding bit in `McBSPi.MCBSPLP_IRQSTATUS_REG` register clears the interrupt.

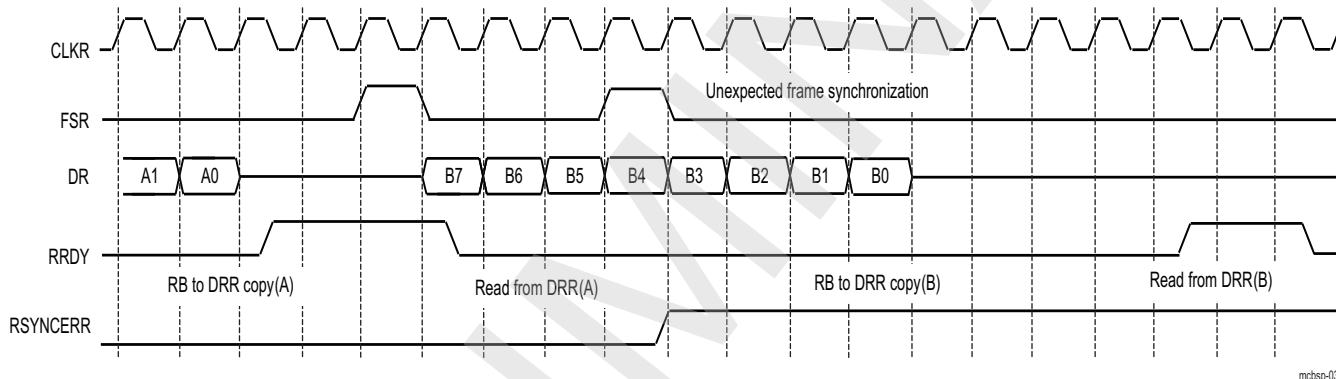
Using the legacy mode, `McBSPi.MCBSPLP_SPCR1_REG[3]` `RSYNCERR` bit can be cleared only by a receiver reset or by writing 0 to this bit. If you want the McBSP module to notify the MPU/DSP subsystem of receive frame-synchronization errors, set the legacy mode receive interrupt with the `McBSPi.MCBSPLP_SPCR1_REG[5:4]` `RINTM` field. When `RINTM=0b11`, the McBSP module sends a receive interrupt (legacy mode) request to the MPU/DSP subsystems each time that `RSYNCERR` is set.

### 23.5.4.10.3.2 Example of an Unexpected Receive Frame-sync Pulse

Figure 23-147 shows an unexpected receive frame-synchronization pulse during normal operation of the serial port, with time intervals between data packets.

**NOTE:** The unexpected receive frame-synchronization pulse does not influence the data receive process, being ignored by the data receive state-machine.

**Figure 23-147. Unexpected Frame-sync Pulse During a McBSP Reception**

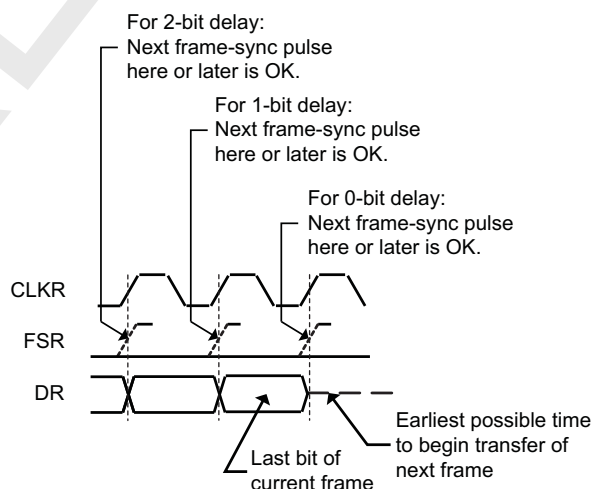


mcbasp-038

### 23.5.4.10.3.3 Preventing Unexpected Receive Frame-sync Pulses

Each frame transfer can be delayed by 0, 1, or 2 `CLKR` cycles, depending on the value of the `McBSPi.MCBSPLP_RCR2_REG[1:0]` `RDATDLY` field. For each possible data delay, Figure 23-148 shows when a new frame-synchronization pulse on `FSR` can safely occur relative to the last bit of the current frame.

**Figure 23-148. Proper Positioning of Receive Frame-sync Pulses**



mcbasp-039

#### **23.5.4.10.4 Underflow in the Receiver**

The McBSP module indicates a receiver underflow condition by setting the McBSPi.MCBSPLP\_IRQSTATUS\_REG[4] RUNDLSTAT bit. This error occurs when sDMA controller or MPU/DSP subsystem reads data from an empty RB this happens only if the MPU/DSP subsystem or sDMA controller does not respect the DMA length, does not wait for DMA request, or does not check the buffer status before reading data. According to the McBSPi.MCBSPLP\_IRQENABLE\_REG register settings this condition can generate the McBSPi\_IRQ line to be asserted low. Writing 1 to the corresponding bit in McBSPi.MCBSPLP\_IRQSTATUS\_REG register clears the interrupt.

#### **23.5.4.10.5 Underflow in the Transmitter**

The McBSP module indicates a transmitter empty (or underflow) condition by setting the McBSPi.MCBSPLP\_IRQSTATUS\_REG[11] XUNDLSTAT bit. Also the legacy mode McBSPi.MCBSPLP\_SPCR2\_REG[2] XEMPTY bit is cleared. Either of the following events activates XEMPTY bit (XEMPTY = 0):

- McBSPi.MCBSPLP\_DXR\_REG has not been loaded and XB is empty, and all bits of the data word in the XSR have been shifted out on the mcbspi\_dx pin.
- The transmitter is reset (by forcing McBSPi.MCBSPLP\_SPCR2\_REG[0] XRST=0, or by an global reset) and is then restarted.

XEMPTY bit is deactivated (XEMPTY=1) when a new word in McBSPi.MCBSPLP\_DXR\_REG is transferred to Transmit Buffer (XB). If McBSPi.MCBSPLP\_PCR\_REG[11] FSXM=1 and McBSPi.MCBSPLP\_SRGR2\_REG[12] FSGM=0, the transmit frame-sync signal (FSX) is generated when Transmit Buffer (XB) is not empty. When McBSPi.MCBSPLP\_SRGR2\_REG[12] FSGM=0, McBSPi.MCBSPLP\_SRGR2\_REG[11:0] FPER and McBSPi.MCBSPLP\_SRGR1\_REG[15:8] FWID are used to determine the frame synchronization period and width (external FSX is gated by the buffer empty condition). Otherwise, the transmitter waits for the next frame-synchronization pulse before sending out the next frame on mcbspi\_dx.

When the transmitter is taken out of reset (McBSPi.MCBSPLP\_SPCR2\_REG[0] XRST=1), it is in a transmitter ready state (McBSPi.MCBSPLP\_SPCR2\_REG[1] XRDYbit =1) and transmitter empty (McBSPi.MCBSPLP\_SPCR2\_REG[2] XEMPTY=0) state. If McBSPi.MCBSPLP\_DXR\_REG is loaded by the MPU/DSP subsystem or the sDMA controller before internal FSX goes active high, a valid XB-to-XSR transfer occurs. This allows for the first word of the first frame to be valid even before the transmit frame-synchronization pulse is generated or detected. Alternatively, if a transmit frame-synchronization pulse is detected before McBSPi.MCBSPLP\_DXR\_REG is loaded, zeros are output on mcbspi\_dx.

The McBSPi.MCBSPLP\_IRQSTATUS\_REG[11] XUNDLSTAT bit indicates a real underflow condition, in which the frame is corrupted due to lack of data availability during transmit process. According to the McBSPi.MCBSPLP\_IRQENABLE\_REG register settings this condition can generate the McBSPi\_IRQ line to be asserted low. Writing 1 to the corresponding bit in McBSPi.MCBSPLP\_IRQSTATUS\_REG register clears the interrupt.

#### **23.5.4.10.6 Unexpected Transmit Frame-sync Pulse**

##### **23.5.4.10.6.1 Possible Responses to Transmit Frame-sync Pulses**

If a frame-synchronization pulse starts the transfer of a new frame before the current frame is fully transmitted, this pulse is treated as an unexpected frame-synchronization pulse, and the transmitter sets the transmit frame-synchronization error bit McBSPi.MCBSPLP\_IRQSTATUS\_REG[7] XSYNCERR (and the legacy McBSPi.MCBSPLP\_SPCR2\_REG[3] XSYNCERR bit).

According to the McBSPi.MCBSPLP\_IRQENABLE\_REG register settings, this condition can generate the McBSPi\_IRQ line to be asserted low. Writing 1 to the corresponding bit in status register clears the interrupt.

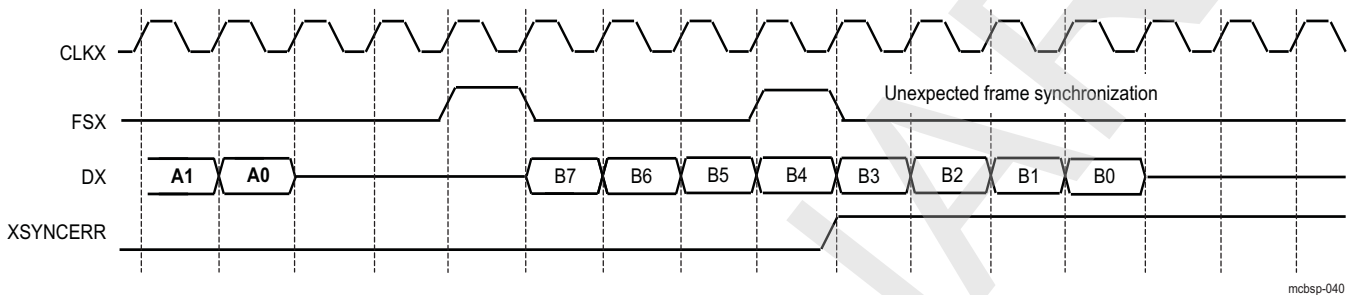
Using the legacy mode, McBSPi.MCBSPLP\_SPCR2\_REG[3] XSYNCERR bit can be cleared only by a transmitter reset or by a write of 0 to this bit. If you want the McBSP module to notify the MPU/DSP subsystem of frame-synchronization errors, you can set a special transmit interrupt mode with the McBSPi.MCBSPLP\_SPCR2\_REG[5:4] XINTM field. When XINTM=0b11, the McBSP module sends a transmit interrupt request to the MPU/DSP subsystem each time that XSYNCERR is set.

### 23.5.4.10.6.2 Example of Unexpected Transmit Frame-Synchronization Pulse

Figure 23-149 shows an unexpected transmit frame–synchronization pulse during normal operation of the serial port with intervals between the data packets.

**NOTE:** The unexpected transmit frame-synchronization pulse does not influence the data transmit process, being ignored by the data transmit state-machine.

**Figure 23-149. Unexpected Frame-sync Pulse During a McBSP Transmission**

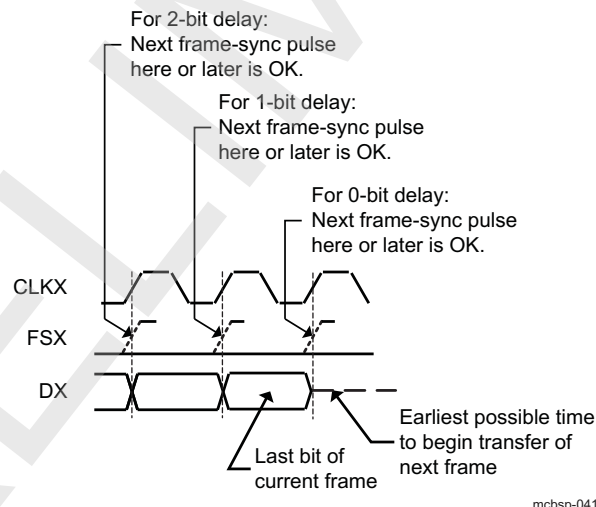


mcbasp-040

### 23.5.4.10.6.3 Preventing Unexpected Transmit Frame-sync Pulses

Each frame transfer can be delayed by 0, 1, or 2 CLKX cycles, depending on the value in the McBSPi.MCBSPLP\_XCR2\_REG[1:0] XDATDLY field. For each possible data delay, Figure 23-150 shows when a new frame–synchronization pulse on FSX can safely occur relative to the last bit of the current frame.

**Figure 23-150. Proper Positioning of Transmit Frame-sync Pulses**



mcbasp-041

### 23.5.4.10.7 Overflow in the Transmitter

The McBSP module indicates a transmitter overflow condition by setting the McBSPi.MCBSPLP\_IRQSTATUS\_REG[12] XOVLSTAT bit. This error occurs when sDMA controller or MPU/DSP subsystem write data to a full XB (this may happen only if the MPU/DSP subsystem or sDMA controller does not respect the DMA length, does not wait for DMA request or does not check the buffer status before writing data). According to the McBSPi.MCBSPLP\_IRQENABLE\_REG register settings this condition can generate the McBSPi\_IRQ line to be asserted low. Writing 1 to the corresponding bit in status register clears the interrupt.



### 23.5.4.11 McBSP DMA Configuration

The McBSP receive and transmit data DMA requests are active after the receive McBSPi.MCBSPLP\_SPCR1\_REG[0] RRDY and transmit McBSPi.MCBSPLP\_SPCR2\_REG[0] XRDY are released. After reset the default DMA threshold (and length) is one.

The receive and transmit DMA requests can be individually disabled by setting the McBSPi.MCBSPLP\_RCCR\_REG[3] RDMAEN, McBSPi.MCBSPLP\_XCCR\_REG[3] XDMAEN bits to 0. When disabling the DMA, the DMA request line is de-asserted even if a DMA transfer is pending and the DMA state-machine is not reset.

The DMA threshold and length configuration is done through McBSPi.MCBSPLP\_THRSH1\_REG and McBSPi.MCBSPLP\_THRSH2\_REG registers as follows:

- (THRSH1\_REG + 1) value represents the required receive DMA request length (the length of the transfer is the same as the threshold value plus one). As long as the RB occupied locations level is above or equal to the THRSH1\_REG value + 1, the DMA request is asserted. After transferring the configured (THRSH1\_REG + 1) number of words, the receive DMA request is de-asserted and reasserted as soon as the conditions are met again.
- (THRSH2\_REG + 1) value represents the required transmit DMA request length (the length of the transfer is the same as the threshold value plus one). As long as the XB free locations level is above or equal to the THRSH2\_REG value + 1, the DMA request is asserted. After transferring the configured (THRSH2\_REG + 1) number of words, the transmit DMA request is de-asserted and reasserted as soon as the conditions are met again.

---

**NOTE:** The MPU/DSP subsystem can decide not to use the DMA to transfer the data. In this case, the DMA must be disabled (or the DMA request can be ignored by MPU/DSP subsystem) and the common interrupt line (MCBSPi\_IRQ) can be used. The McBSPi.MCBSPLP\_SPCR1\_REG[1] RRDY bit for receive and McBSPi.MCBSPLP\_SPCR2\_REG[1] XRDY bit for transmit will indicate when the threshold values are reached. Also, by reading the receive buffer status McBSPi.MCBSPLP\_RBUFFSTAT\_REG register and transmit buffer status McBSPi.MCBSPLP\_XBUFFSTAT\_REG register, the MPU/DSP subsystem can decide to transfer data even if the threshold is not reached. This mechanism is useful on the last transfer on receive side when the threshold value is bigger than the occupied locations inside the receive buffer and the MPU/DSP subsystem needs to read this data. Since no interrupt or DMA request is asserted the only option in this case is to read the RB status register value and to transfer the remaining data without using the DMA or interrupt indication.

---

### 23.5.4.12 Multi-channel Selection Modes

#### 23.5.4.12.1 Channels, Blocks and Partitions

A McBSP channel is a time slot for shifting in/out the bits of one serial word. The McBSP module supports up to 128 channels for reception and 128 channels for transmission. In the receiver and in the transmitter, the 128 available channels are divided into eight blocks that contain 16 contiguous channels each (see [Table 23-339](#)):

**Table 23-339. McBSP Channels**

<b>Block 0: Channels 0–15</b>	<b>Block 4: Channels 64–79</b>
<b>Block 1: Channels 16–31</b>	<b>Block 5: Channels 80–95</b>
<b>Block 2: Channels 32–47</b>	<b>Block 6: Channels 96–111</b>
<b>Block 3: Channels 48–63</b>	<b>Block 7: Channels 112–127</b>

The blocks are assigned to partitions according to the selected partition mode. In the two-partition mode described in [Section 23.5.4.12.6, Using Two Partitions \(Legacy Only\)](#), you assign one even-numbered block (0, 2, 4, or 6) to partition A and one odd-numbered block (1, 3, 5, or 7) to partition B. In the 8-partition mode, see [Section 23.5.4.12.4, Using Eight Partitions](#), blocks 0 through 7 are automatically assigned to partitions A through H, respectively.

The number of partitions for reception and the number of partitions for transmission are independent of each other. For example, it is possible to use two receive partitions (A and B) and eight transmit partitions (A–H).

### 23.5.4.12.2 Multi-channel Selection

When a McBSP module uses a time-division multiplexed (TDM) data stream while communicating with other McBSP modules or serial devices, the McBSP module may need to receive and/or transmit on only a few channels. To save memory and bus bandwidth, you can use a multi-channel selection mode to prevent data flow in some of the channels.

Each channel partition has a dedicated channel enable register. If the appropriate multi-channel selection mode is on, each bit in the register controls whether data flow is allowed or prevented in one of the channels that is assigned to that partition.

The McBSP module has one receive multi-channel selection mode [Section 23.5.4.12.5, Receive Multi-channel Selection Mode](#) and three transmit multi-channel selection modes [Section 23.5.4.12.7, Transmit Multi-channel Selection Modes](#).

### 23.5.4.12.3 Configuring a Frame for Multi-channel Selection

Before enabling a multi-channel selection mode, make sure you properly configure the data frame:

- Select a single-phase frame (McBSPi.MCBSPLP\_RCR2\_REG[15] RPHASE bit and McBSPi.MCBSPLP\_XCR2\_REG[15] XPHASE bit=0). Each frame represents a TDM data stream.
- Set a frame length (in McBSPi.MCBSPLP\_RCR1\_REG[14:8] RFLEN1 field and in McBSPi.MCBSPLP\_XCR1\_REG[14:8] XFLEN1 field) that includes the highest-numbered channel to be used. For example, if you plan to use channels 0, 15, and 39 for reception, the receive frame length must be at least 40 (RFLEN1=39). If XFLEN1=39 in this case, the receiver creates 40 time slots per frame but only receives data during time slots 0, 15, and 39 of each frame.

### 23.5.4.12.4 Using Eight Partitions

For multi-channel selection operation in the receiver and/or the transmitter, you can use eight partitions or two partitions (as previously described). If you choose the 8-partition mode (McBSPi.MCBSPLP\_MCR1\_REG[9] RMCME=1 for reception, McBSPi.MCBSPLP\_MCR2\_REG[9] XMCME=1 for transmission), McBSP channels are activated in the following order: A, B, C, D, E, F, G, H.

In response to a frame-synchronization pulse, the receiver or transmitter begins with the channels in partition A and then continues with the other partitions in order until the complete frame has been transferred. When the next frame-synchronization pulse occurs, the next frame is transferred, beginning with the channels in partition A.

In the 8-partition mode, the McBSPi.MCBSPLP\_MCR1\_REG[6:5] RPABLK/McBSPi.MCBSPLP\_MCR2\_REG[6:5] XPABLK and McBSPi.MCBSPLP\_MCR1\_REG[8:7] RPBBLK/McBSPi.MCBSPLP\_MCR2\_REG[8:7] XPBBLK bits are ignored and the 16-channel blocks are assigned to the partitions as shown in [Table 23-340](#) through [Table 23-341](#). These assignments cannot be changed. The tables also show the registers used to control the channels in the partitions.

**Table 23-340. Eight Partitions – Receive Channel Assignment and Control**

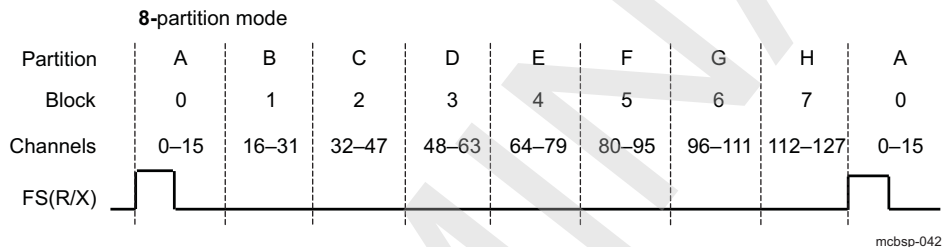
Receive Partition	Assigned Block of Receive Channels	Register Used for Channel Control
A	Block 0: channels 0 through 15	McBSPi.MCBSPLP_RCERA_REG
B	Block 1: channels 16 through 31	McBSPi.MCBSPLP_RCERB_REG
C	Block 2: channels 32 through 47	McBSPi.MCBSPLP_RCERC_REG
D	Block 3: channels 48 through 63	McBSPi.MCBSPLP_RCERD_REG
E	Block 4: channels 64 through 79	McBSPi.MCBSPLP_RCERE_REG
F	Block 5: channels 80 through 95	McBSPi.MCBSPLP_RCERF_REG
G	Block 6: channels 96 through 111	McBSPi.MCBSPLP_RCERG_REG
H	Block 7: channels 112 through 127	McBSPi.MCBSPLP_RCERH_REG

**Table 23-341. Eight Partitions – Transmit Channel Assignment and Control**

Transmit Partition	Assigned Block of Receive Channels	Register Used for Channel Control
A	Block 0: channels 0 through 15	McBSPi.MCBSPLP_XCERA_REG
B	Block 1: channels 16 through 31	McBSPi.MCBSPLP_XCERB_REG
C	Block 2: channels 32 through 47	McBSPi.MCBSPLP_XCERC_REG
D	Block 3: channels 48 through 63	McBSPi.MCBSPLP_XCERD_REG
E	Block 4: channels 64 through 79	McBSPi.MCBSPLP_XCERE_REG
F	Block 5: channels 80 through 95	McBSPi.MCBSPLP_XCERF_REG
G	Block 6: channels 96 through 111	McBSPi.MCBSPLP_XCERG_REG
H	Block 7: channels 112 through 127	McBSPi.MCBSPLP_XCERH_REG

Figure 23-151 shows an example of the McBSP using the 8-partition mode. In response to a frame-synchronization pulse, the McBSP module begins a frame transfer with partition A and then activates B, C, D, E, F, G, and H to complete a 128-word frame.

**Figure 23-151. McBSP Data Transfer in the 8-partition Mode**



#### 23.5.4.12.5 Receive Multi-channel Selection Mode

The McBSPi.MCBSPLP\_MCR1\_REG[0] RMCM bit determines whether all channels or only selected channels are enabled for reception.

- When RMCM=0, all 128 receive channels are enabled and cannot be disabled.
- When RMCM=1, the receive multi-channel selection mode is enabled. In this mode:
  - Channels can be individually enabled or disabled. The enabled channels are those selected in the appropriate receive channel enable registers (McBSPi.MCBSPLP\_RCERA\_REG/McBSPi.MCBSPLP\_RCERH\_REG). The channels are assigned to the McBSPi.MCBSPLP\_RCERA\_REG/McBSPi.MCBSPLP\_RCERH\_REG registers depends on the number of receive channel partitions (2 or 8), as defined by the McBSPi.MCBSPLP\_MCR1\_REG[9] RMCME bit.
  - If a receive channel is disabled, any bits received in that channel are not transferred to the RB, and as a result, the receiver ready bit (RRDY) is not set. Therefore, no DMA synchronization event is generated and, if the receiver interrupt mode depends on RRDY (McBSPi.MCBSPLP\_SPCR1\_REG[5:4] RINTM=0b00), no interrupt is generated.

As an example of how the McBSP module behaves in the receive multi-channel selection mode, suppose you enable only channels 0, 15, and 39 and that the frame length is 40. The McBSP module:

1. Accepts bits shifted in from the mcbspi\_dr pin in channel 0
2. Ignores bits received in channels 1–14
3. Accepts bits shifted in from the mcbspi\_dr pin in channel 15
4. Ignores bits received in channels 16–38
5. Accepts bits shifted in from the mcbspi\_dr pin in channel 39

#### 23.5.4.12.6 Using Two Partitions (Legacy Only)

For multi-channel selection operation in the receiver and/or the transmitter, you can use two partitions or eight partitions. If you choose the 2-partition mode (McBSPi.MCBSPLP\_MCR1\_REG[9] RMCME=0 for

reception, McBSPi.MCBSPLP\_MCR2\_REG[9] XMCME=0 for transmission), McBSP channels are activated using an alternating scheme. In response to a frame-synchronization pulse, the receiver or transmitter begins with the channels in partition A and then alternates between partitions B and A until the complete frame has been transferred. When the next frame-synchronization pulse occurs, the next frame is transferred beginning with the channels in partition A.

For reception, any two of the eight receive-channel blocks can be assigned to receive partitions A and B, which means up to 32 receive channels can be enabled at any given point. Similarly, any two of the eight transmit-channel blocks (up to 32 enabled transmit channels) can be assigned to transmit partitions A and B.

For reception:

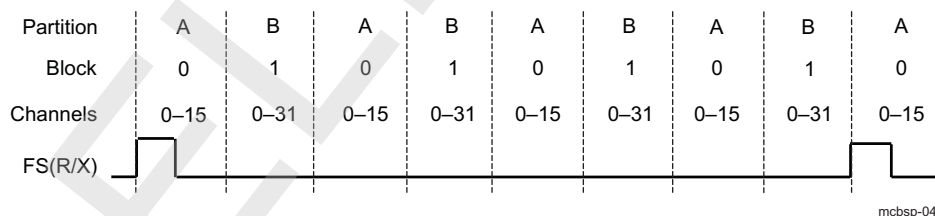
- Assign an even-numbered channel block (0, 2, 4, or 6) to receive partition A by writing to the McBSPi.MCBSPLP\_MCR1\_REG[6:5] RPABLK field. In the receive multi-channel selection mode, the channels in this partition are controlled by receive channel enable register A (McBSPi.MCBSPLP\_RCERA\_REG).
- Assign an odd-numbered block (1, 3, 5, or 7) to receive partition B with the McBSPi.MCBSPLP\_MCR1\_REG[8:7] RPBBLK field. In the receive multi-channel selection mode, the channels in this partition are controlled by receive channel enable register B (McBSPi.MCBSPLP\_RCERB\_REG).

For transmission:

- Assign an even-numbered channel block (0, 2, 4, or 6) to transmit partition A by writing to the McBSPi.MCBSPLP\_MCR2\_REG[6:5] XPABLK fields. In one of the transmit multi-channel selection modes, the channels in this partition are controlled by transmit channel enable register A (McBSPi.MCBSPLP\_XCERA\_REG).
- Assign an odd-numbered block (1, 3, 5, or 7) to transmit partition B with the McBSPi.MCBSPLP\_MCR1\_REG[8:7] XPBBLK field. In one of the transmit multi-channel selection modes, the channels in this partition are controlled by transmit channel enable register B (McBSPi.MCBSPLP\_XCERB\_REG).

Figure 23-152 shows an example of alternating between the channels of partition A and the channels of partition B. Channels 0–15 have been assigned to partition A, and channels 16–31 have been assigned to partition B. In response to a frame-synchronization pulse, the McBSP module begins a frame transfer with partition A and then alternates between partitions B and A until the complete frame is transferred.

**Figure 23-152. Alternating Between Partitions A and B Channels**



#### 23.5.4.12.7 Transmit Multi-channel Selection Modes

The McBSPi.MCBSPLP\_MCR2\_REG[1:0] XMCM field determine whether all channels or only selected channels are enabled and unmasked for transmission. The McBSP module has three transmit multi-channel selection modes (XMCM=0b01, XMCM=0b10, and XMCM=0b11), which are described in Table 23-342.

**Table 23-342. Selecting a Transmit Multi-Channel Selection Mode with the XMCM Bit Field**

XMCM	Transmit Multi-Channel Selection Mode
0b00	No transmit multi-channel selection mode is on. All channels are enabled and unmasked. No channels can be disabled or masked.
0b01	All channels are disabled unless they are selected in the appropriate transmit channel enable registers (McBSPi.MCBSPLP_XCERA_REG/McBSPi.MCBSPLP_XCERH_REG). If enabled, a channel in this mode is also unmasked. The McBSPi.MCBSPLP_MCR2_REG[9] XMCME bit determines whether 32 channels or 128 channels are selectable in the McBSPi.MCBSPLP_XCERA_REG/McBSPi.MCBSPLP_XCERH_REG registers.

**Table 23-342. Selecting a Transmit Multi-Channel Selection Mode with the XMCM Bit Field (continued)**

XMCM	Transmit Multi-Channel Selection Mode
0b10	All channels are enabled, but they are masked unless they are selected in the appropriate transmit channel enable registers (McBSPi.MCBSPLP_XCERA_REG/McBSPi.MCBSPLP_XCERH_REG). The McBSPi.MCBSPLP_MCR2_REG[9] XMCME bit determines whether 32 channels or 128 channels are selectable in the McBSPi.MCBSPLP_XCERA_REG/McBSPi.MCBSPLP_XCERH_REG registers.
0b11	This mode is used for symmetric transmission and reception. All channels are disabled for transmission unless they are enabled for reception in the appropriate receive channel enable registers (McBSPi.MCBSPLP_RCERA_REG/McBSPi.MCBSPLP_RCERH_REG). Once enabled, they are masked unless they are also selected in the appropriate transmit channel enable registers (McBSPi.MCBSPLP_XCERA_REG/McBSPi.MCBSPLP_XCERH_REG). The McBSPi.MCBSPLP_MCR2_REG[9] XMCME bit determines whether 32 channels or 128 channels are selectable in McBSPi.MCBSPLP_RCERA_REG/McBSPi.MCBSPLP_RCERH_REG registers and McBSPi.MCBSPLP_XCERA_REG/McBSPi.MCBSPLP_XCERH_REG registers.

As an example of how the McBSP module behaves in a transmit multi-channel selection mode, suppose that XMCM=0b01 (all channels disabled unless individually enabled) and that you have enabled only channels 0, 15, and 39. Suppose also that the frame length is 40. The McBSP module:

1. Shifts data to the mcbspi\_dx pin in channel 0
2. Places the mcbspi\_dx pin in the high impedance state in channels 1–14
3. Shifts data to the mcbspi\_dx pin in channel 15
4. Places the mcbspi\_dx pin in the high impedance state in channels 16–38
5. Shifts data to the mcbspi\_dx pin in channel 39

**23.5.4.12.7.1 Disabling/Enabling Versus Masking/Unmasking**

For transmission, a channel can be:

- Enabled and unmasked (transmission can begin and can be completed)
- Enabled but masked (transmission can begin but cannot be completed)
- Disabled (transmission cannot occur)

The definitions in [Table 23-343](#) explain the channel control options:

**Table 23-343. McBSP Channel Control Options**

<b>Enabled channel</b>	A channel that can begin transmission by passing data from the data transmit register (McBSPi.MCBSPLP_DXR_REG) to the XSR through XB.
<b>Masked channel</b>	A channel that cannot complete transmission. The mcbspi_dx pin is held in the high impedance state; data cannot be shifted out on the mcbspi_dx pin. In systems where symmetric transmit and receive provide software benefits, this feature allows transmit channels to be disabled on a shared serial bus. A similar feature is not needed for reception because multiple receptions cannot cause serial bus contention.
<b>Disabled channel</b>	A channel that is not enabled. A disabled channel is also masked. Because no DXR-to-XB copy occurs, the McBSPi.MCBSPLP_SPCR2_REG[1] XRDY bit is not set. Therefore, no DMA synchronization event is generated, and if the transmit interrupt mode depends on XRDY (McBSPi.MCBSPLP_SPCR2_REG[5:4] XINTM=00b), no interrupt is generated. The McBSPi.MCBSPLP_SPCR2_REG[2] XEMPTY bit is not affected.
<b>Unmasked channel</b>	A channel that is not masked. Data in the XSR(s) is shifted out on the mcbspi_dx pin.

**23.5.4.12.7.2 Activity on McBSP Pins for Different Values of XMCM**

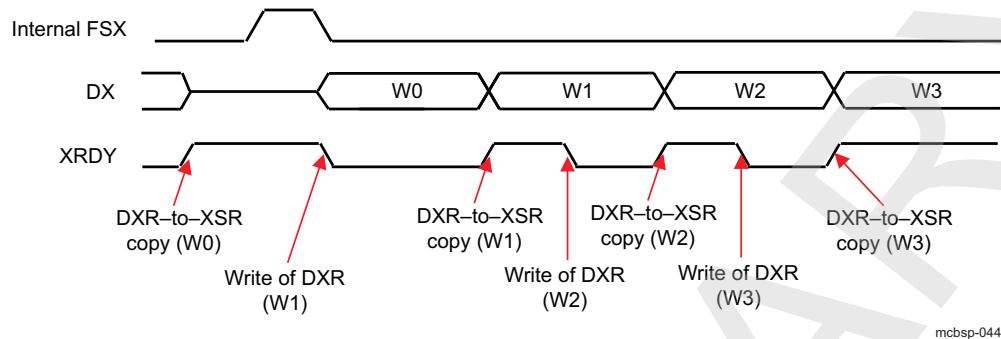
[Figure 23-153](#) shows the activity on the McBSP pins for the various McBSPi.MCBSPLP\_MCR2\_REG[1:0] XMCM values. In all cases, the transmit frame is configured as follows:

- XPHASE=0: Single-phase frame (required for multi-channel selection modes)
- XFRLLEN1=0b0000011: 4 words per frame
- XWDLEN1=0b000: 8 bits per word
- XMCME=0: 2-partition mode (only partitions A and B used)

In the case where McBSPi.MCBSPLP\_MCR2\_REG[1:0] XMCM=0b11, transmission and reception are symmetric, which means the corresponding bits for the receiver (RPHASE, RFRLLEN1, RWDLEN1, and RMCME) must have the same values as XPHASE, XFRLLEN1, and XWDLEN1, respectively.

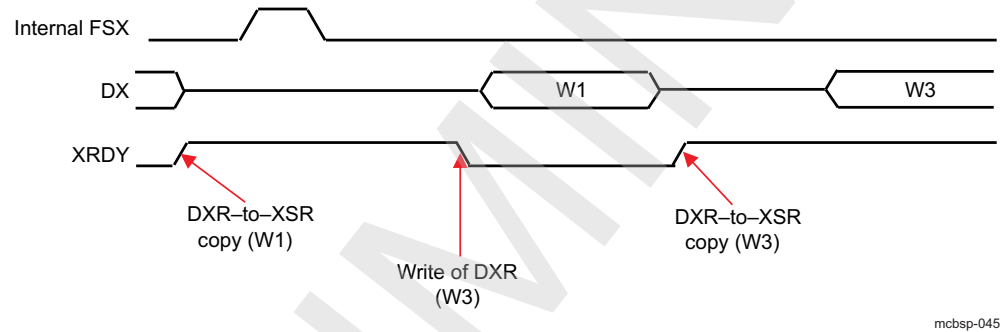
In [Figure 23-153](#), the arrows showing where the various events occur are only sample indications. Wherever possible, there is a time window in which these events can occur.

**Figure 23-153. Activity on McBSP Pins When XMCM=0b00**



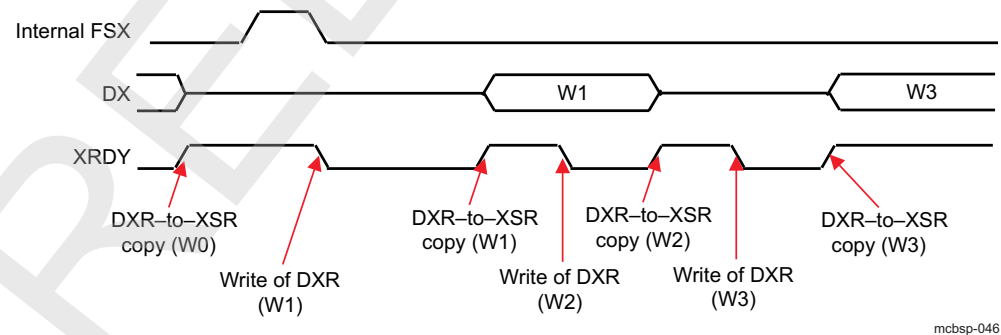
If XMCM = 0b00, all channels are enabled and unmasked. Words W0, W1, W2, and W3 are written to the XB, then, from the XB, there are transferred by mcbspi\_dx.

**Figure 23-154. Activity on McBSP Pins When XMCM=0b01**



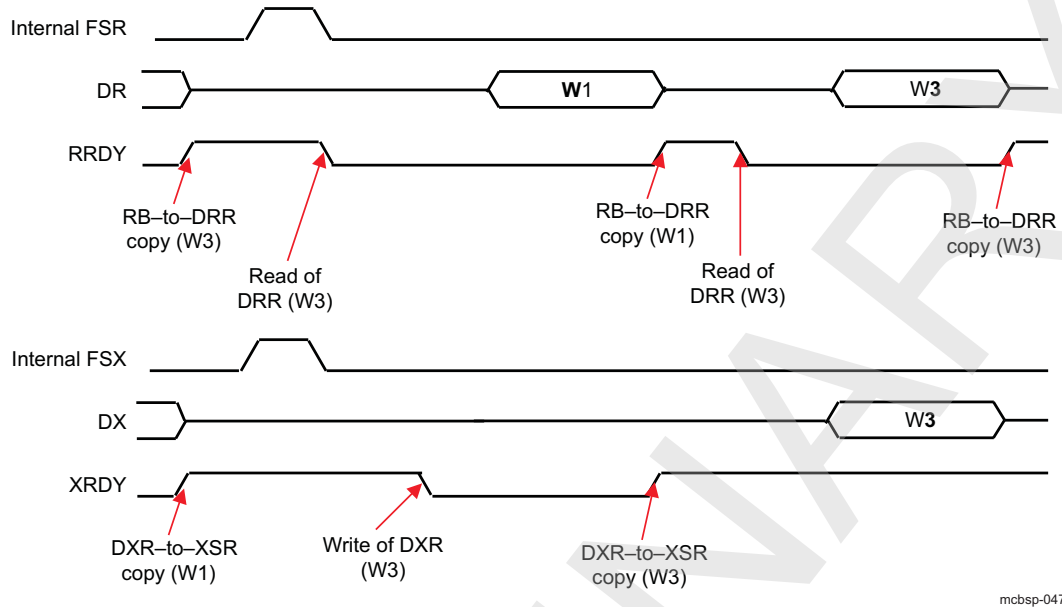
In [Figure 23-154](#) if XMCM = 0b01, XPABLK = 0b00, and XCERA = 0b1010, only channels 1 and 3 are enabled and unmasked. Words W1 and W3 are written to the XB, then, from the XB, there are transferred by mcbspi\_dx.

**Figure 23-155. Activity on McBSP Pins When XMCM=0b10**



In [Figure 23-155](#) if XMCM = 0b10, XPABLK = 0b00, and XCERA = 0b1010, all channels are enabled, only 1 and 3 unmasked. Words W0, W1, W2, and W3 are written to the XB, but only W1 and W3, from the XB, there are transferred by mcbspi\_dx.

Figure 23-156. Activity on McBSP Pins When XMCM=0b11



In Figure 23-156 if XMCM = 0b11, RPABLK = 0b00, XPABLK = 0bX, RCERA = 0b1010 and XCERA = 0b1000, channels 1 and 3 are enabled in receive and transmit mode, but only 3 unmasked. Words W1 and W3 are written to the XB, but only W3, from the XB, there are transferred by mcbspi\_dx.

### 23.5.5 McBSP Basic Programming Model

This section describes the programming model of typical McBSP module.

#### CAUTION

For all descriptions in this section, the McBSPi.MCBSPLP\_XCCR\_REG[11] XFULL\_CYCLE bit and the McBSPi.MCBSPLP\_RCCR\_REG[11] RFULL\_CYCLE bit are their reset value (XFULL\_CYCLE bit=0 and RFULL\_CYCLE bit=1).

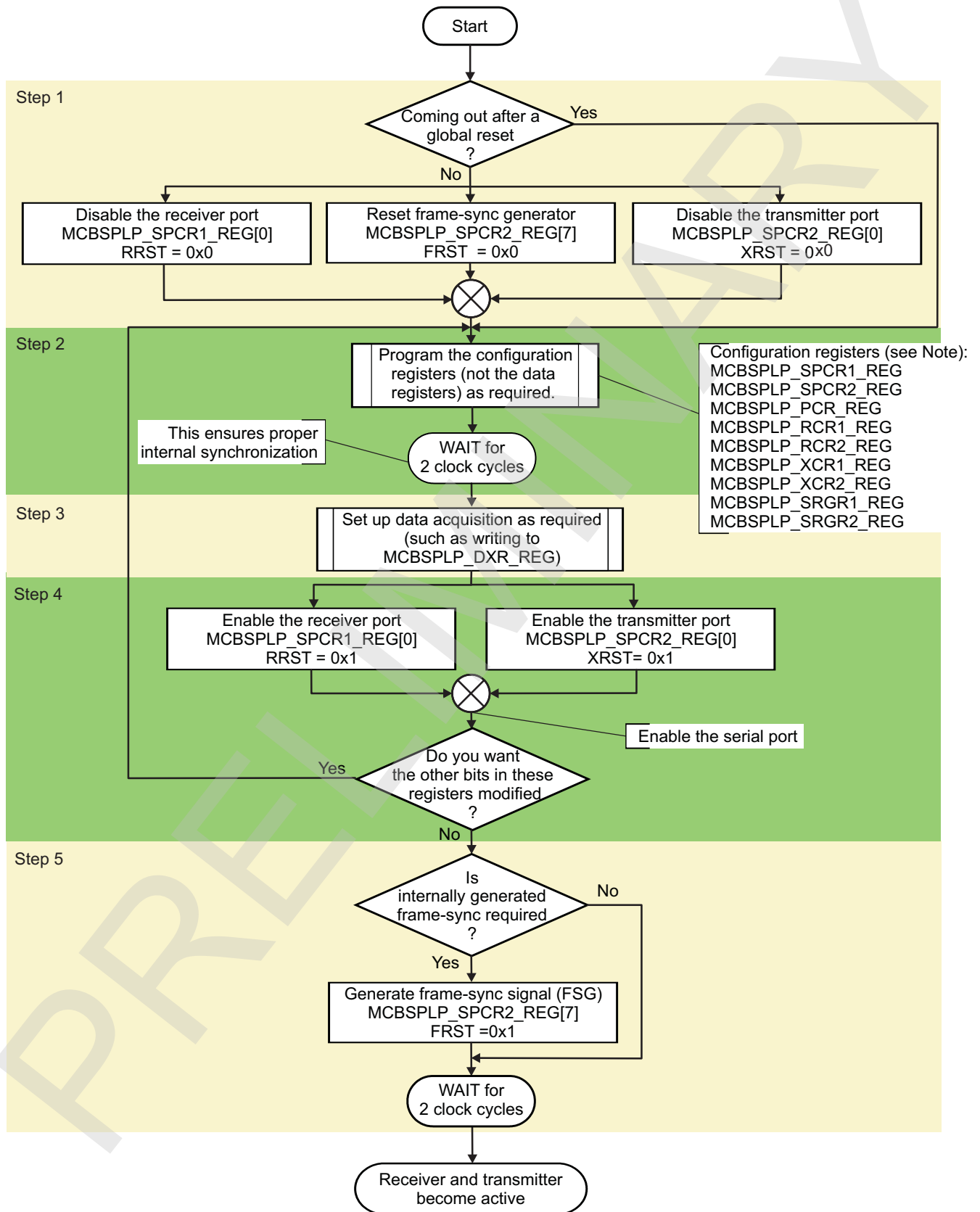
#### 23.5.5.1 McBSP Initialization Procedure

This procedure for reset/initialization can be applied in general when the Receiver or Transmitter has to be reset during its normal operation, and also when the Sample Rate Generator is not used for either operation.

Figure 23-157 presents the serial port initialization procedure for master mode.



Figure 23-157. Flow Diagram of McBSP Initialization Procedure for Master Mode

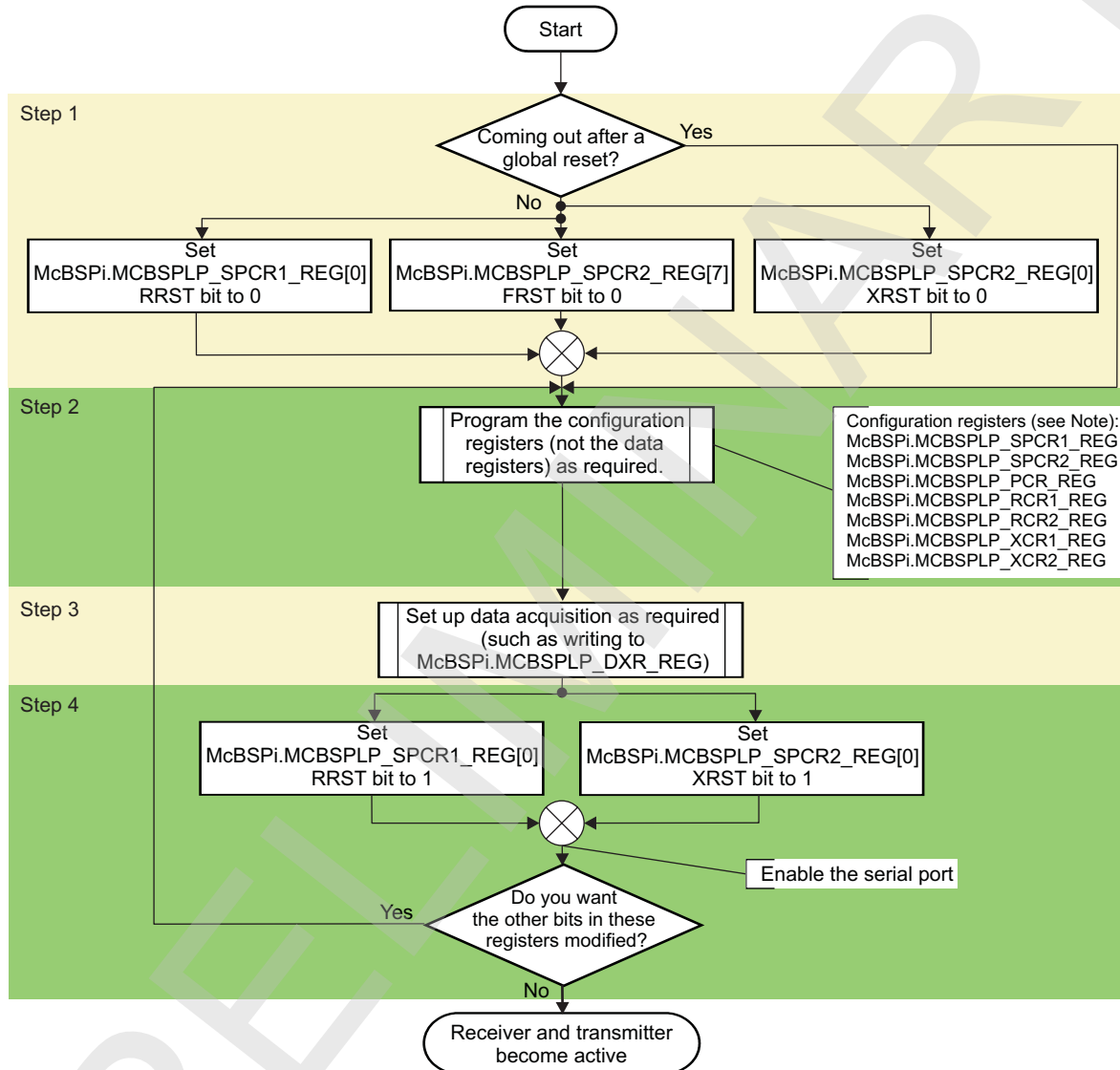


mcbsp-062

Alternatively, on write (step 1 or 4), the transmitter and receiver can be placed in or taken out of reset by modifying the McBSPi.MCBSPLP\_SPCR2\_REG[0] XRST bit and the McBSPi.MCBSPLP\_SPCR1\_REG[0] RRST bit, respectively.

Figure 23-158 shows the flow diagram of the McBSP initialization procedure for slave mode.

**Figure 23-158. Flow Diagram of McBSP Initialization Procedure for Slave Mode**



mcbasp-075

**NOTE:**

- Alternatively, on write (step 1 or 4), the transmitter and receiver can be placed in or taken out of reset by modifying the McBSPi.MCBSPLP\_SPCR2\_REG[0] XRST bit and the McBSPi.MCBSPLP\_SPCR1\_REG[0] RRST bit, respectively.
- The necessary duration of the active-low period of XRST or RRST is at least two CLKR/CLKX cycles.
- The appropriate bits in serial port configuration registers (McBSPi.MCBSPLP\_SPCR1\_REG, McBSPi.MCBSPLP\_SPCR2\_REG, McBSPi.MCBSPLP\_PCR\_REG, McBSPi.MCBSPLP\_RCR1\_REG, McBSPi.MCBSPLP\_RCR2\_REG, McBSPi.MCBSPLP\_XCR1\_REG, McBSPi.MCBSPLP\_XCR2\_REG, McBSPi.MCBSPLP\_THRSH2\_REG, McBSPi.MCBSPLP\_XCCR\_REG, McBSPi.MCBSPLP\_SYSCONFIG\_REG, McBSPi.MCBSPLP\_SRGR1\_REG and McBSPi.MCBSPLP\_SRGR2\_REG) should only be modified when the affected portion of the serial port is in its reset state.
- In most cases, the data transmit register (McBSPi.MCBSPLP\_DXR\_REG) should be loaded by the MPU/DSP subsystem or the sDMA controller only when the transmitter is enabled (McBSPi.MCBSPLP\_SPCR2\_REG[0] XRST = 1). An exception to this rule is when these registers are used for loopback internal data.
- The bits of the channel control registers (McBSPi.MCBSPLP\_MCR1\_REG, McBSPi.MCBSPLP\_MCR2\_REG, McBSPi.MCBSPLP\_RCER{A-H}\_REG and McBSPi.MCBSPLP\_XCER{A-H}\_REG) can be modified at any time as long as they are not being used by the current reception/transmission in a multichannel selection mode.
- The SRG is reset by setting McBSPi.MCBSPLP\_SPCR2\_REG[6] GRST bit to 0.
- It is not necessary to wait if SRG is not used
- The necessary duration of the active-low period of XRST or RRST is at least two
- Modification on-the-fly has no effect if a reset is not done first.

**Table 23-344. Register Call Summary for Flow Diagram of McBSP Initialization Procedure**

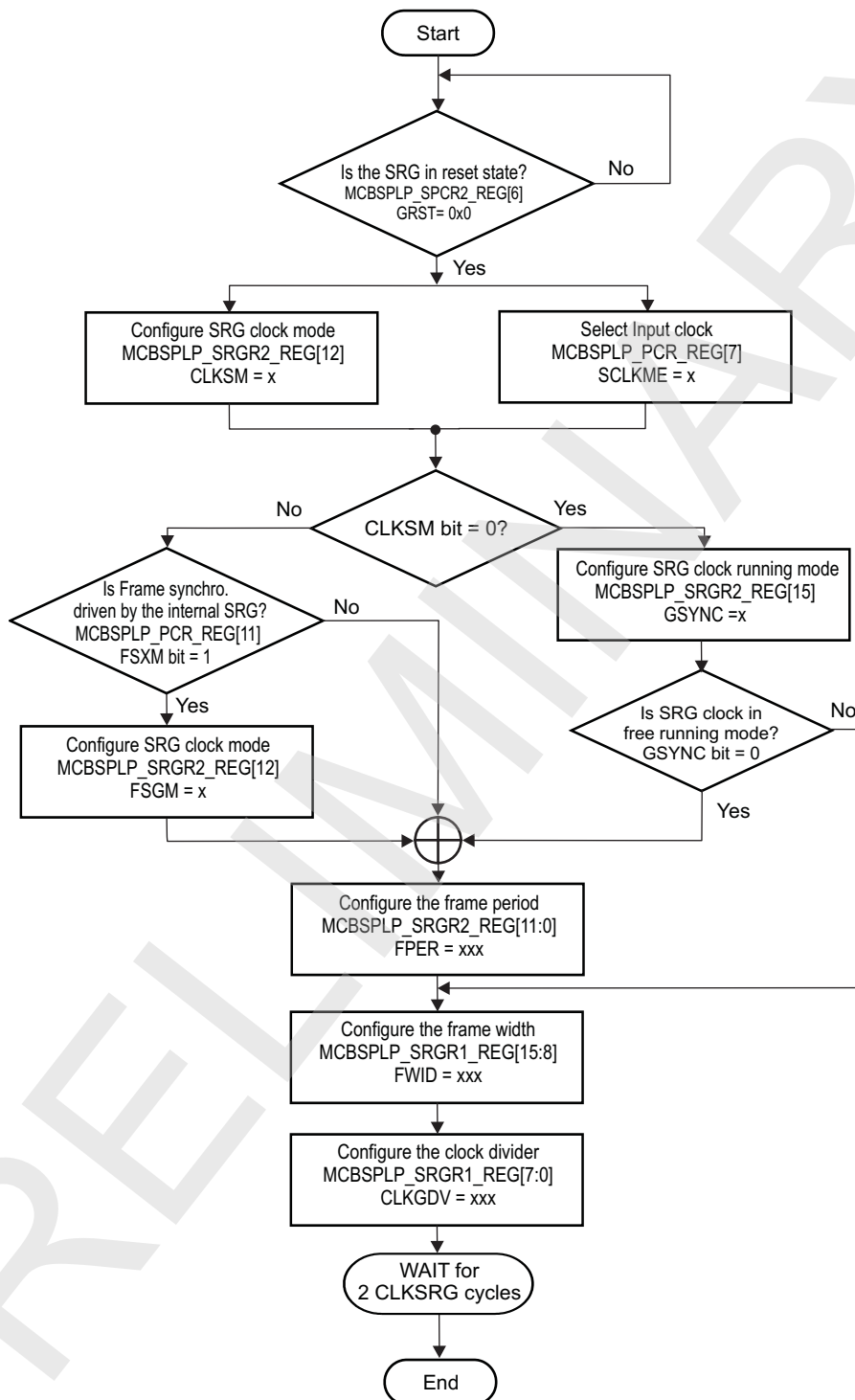
Register Name	Register Name	Register Name
MCBSPLP_SPCR1_REG	MCBSPLP_RCR1_REG	MCBSPLP_XCR2_REG
MCBSPLP_SPCR2_REG	MCBSPLP_RCR2_REG	MCBSPLP_SRGR1_REG
MCBSPLP_PCR_REG	MCBSPLP_XCR1_REG	MCBSPLP_SRGR2_REG

**23.5.5.2 Reset and Initialization Procedure for the Sample Rate Generator**

To reset and initialize the Sample Rate Generator:

- Place the McBSP Sample Rate Generator in RESET.
- Program the registers that affect the Sample Rate Generator.
- Enable the Sample Rate Generator (take it out of reset).
- If necessary, enable the receiver and/or the transmitter.
- If necessary, remove the receiver and/or transmitter from reset

Figure 23-159. Flow Diagram for the SRG Registers Programming



mcbsp-073

Table 23-345. Register Call Summary for Flow Diagram for the SRG Registers Programming

Register Name	Register Name
MCBSPLP_SPCR2_REG	MCBSPLP_PCR_REG
MCBSPLP_SRGR2_REG	MCBSPLP_SRGR1_REG

### 23.5.5.3 Data Transfer DMA Request Configuration

This procedure configures the McBSP receive/transmit data DMA requests (MCBSPi\_DMA\_RX and MCBSPi\_DMA\_TX).

**Table 23-346. Data Transfer DMA Request Configuration**

Step	Register / Bit Field / Programming Model	Value
Write the required receive DMA request length	MCBSPLP_THRSH1_REG[6:0] RTHRESHOLD	x..x
Write the required transmit DMA request length	MCBSPLP_THRSH2_REG[6:0] RTHRESHOLD	x..x

**NOTE:**

- The length of the transfer is the same as the threshold value + 1
- In case of a number of transfers that exceed the number of the programmed DMA length the McBSP module will respond to the command, and will perform the transfer regardless of the receive buffer empty condition.
- In case of a number of transfers that exceed the number of the programmed DMA length the McBSP module will respond to the command, and will perform the transfer regardless of the transmit buffer full condition.

### 23.5.5.4 Interrupt Configuration

This procedure configures the common receive/transmit interrupt request line.

**Table 23-347. Interruption Configuration**

Step	Register / Bit Field / Programming Model	Value
Enable the required IRQ events	MCBSPLP_IRQENABLE_REG	x..x

### 23.5.5.5 Receiver Configuration

To configure the McBSP receiver, perform the following steps:

- Step 1. Place the McBSP receiver in reset .
- Step 2. Program the McBSP registers for the desired receiver operation.
- Step 3. Take the receiver out of reset.

#### 23.5.5.5.1 Place the receiver in reset (Step 1)

**Table 23-348. Receiver Reset**

Step	Register / Bit Field / Programming Model	Value
Place the receiver in Reset	MCBSPLP_SPCR1_REG[0] RST	0x0

### 23.5.5.2 Programming the McBSP Registers for the Desired Receiver Configuration (Step 2)

Followings items detail important procedures to be performed when the software is configuring the McBSP receiver.

- **Global Configuration**

**Table 23-349. Global Configuration**

Step	Register / Bit Field / Programming Model	Value
Set the Receiver Pins to Operate as McBSP Pins	MCBSPLP_PCR_REG[12] RIOEN	0x0
Enable/ Disable the Digital Loop Back Mode <sup>(1)</sup>	MCBSPLP_XCCR_REG[5] DLB	0x0 or 0x1
Enable/ Disable the Analog Loop Back Mode <sup>(1)</sup>	MCBSPLP_SPCR1_REG[5] ALB	0x0 or 0x1
Select clock used in the synchronous transmit-receive mode (McBSP4 only) <sup>(1)</sup>	CONTROL_MCBSPLP[30] ALBCTRLRX_CLKX	0x0 or 0x1
Select Frame Synchronization used in the synchronous transmit-receive mode (McBSP4 only) <sup>(1)</sup>	CONTROL_MCBSPLP[31] ALBCTRLRX_FSX	0x0 or 0x1
Enable/ Disable the Receive Multichannel Selection Mode <sup>(1)</sup>	MCBSPLP_MCR1_REG[0] RMCN	0x0 or 0x1

<sup>(1)</sup> Software decision

**NOTE:** In digital Loopback mode the SRG and the Frame Synchronization generator need to be enabled to generate the CLKX and FSX signals.

- **Data Configuration**

**Table 23-350. Data Configuration**

Step	Register / Bit Field / Programming Model	Value
Select single/dual-phase frame <sup>(1)</sup>	MCBSPLP_RCR2_REG[15] RPHASE	0x0 or 0x1
Set the Receive Word Length(s) for Phase1 <sup>(1)</sup>	MCBSPLP_RCR1_REG[7:5] RWDLEN1	xxx
Set the Receive Word Length(s) for Phase2 <sup>(1)</sup>	MCBSPLP_RCR2_REG[7:5] XWDLEN2	xxx
Set the Receive Frame length*	MCBSPLP_RCR1_REG[14:8] RFRLEN1 MCBSPLP_RCR2_REG[14:8] RFRLEN2	xxx
Set the Receive Reverse Mode <sup>(1)</sup>	MCBSPLP_RCR2_REG[4:3] RREVERSE	xx
Set the Receive Data Delay <sup>(1)</sup>	MCBSPLP_RCR2_REG[1:0] RDATLY	xx
Set the Receive Sign-Extension and Justification Mode <sup>(1)</sup>	MCBSPLP_SPCR1_REG[14:13] RJUST	xx
Enable the serial Receiver Port	MCBSPLP_SPCR1_REG[0] RRST	0x1
Configure the Receive buffer threshold value <sup>(1)</sup>	MCBSPLP_THRSH1_REG[6:0] RTHRESHOLD	x..x

<sup>(1)</sup> Software decision

**NOTE:**

- When dual-phase is selected the number of words per phase must be set to one.
- If a single-phase frame is selected, RWDLEN1 selects the length for every serial word received in the frame. If a dual-phase frame is selected, RWDLEN1 and RWDLEN2 must be set to select the both length. These both bits can have values different.
- If a dual-phase is selected, the frame length must be two words.

- **Frame-Sync Configuration**

**Table 23-351. Frame Sync Configuration**

Step	Register / Bit Field / Programming Model	Value
Set the Receive Frame-Sync Mode <sup>(1)</sup>	MCBSPLP_PCR_REG[10] FSRM	0x0 or 0x1
Select the Sample Rate Generator Synchronization <sup>(1)</sup>	MCBSPLP_SRGR2_REG[15] GSYNC	0x0 or 0x1
Set the Receive Frame-Sync Polarity <sup>(1)</sup>	MCBSPLP_PCR_REG[2] FSRP	0x0 or 0x1

<sup>(1)</sup> Software decision

**Table 23-351. Frame Sync Configuration (continued)**

Step	Register / Bit Field / Programming Model	Value
Set the SRG Frame-Sync Period <sup>(1)</sup>	MCBSPLP_SRGR2_REG[11:0] FPER	x..x
Set the SRG Frame-Sync Pulse Width <sup>(1)</sup>	MCBSPLP_SRGR1_REG[15:8] FWID	x..x

- **Clock Configuration**

**Table 23-352. Clock Configuration**

Step	Register / Bit Field / Programming Model	Value
Set the Receive Clock Mode <sup>(1)</sup>	MCBSPLP_PCR_REG[8] CLKRM	0x0 or 0x1
Set the Receive Clock Polarity <sup>(1)</sup>	MCBSPLP_PCR_REG[0] CLKRP	0x0 or 0x1
Set the SRG Clock Divide-Down Value <sup>(1)</sup>	MCBSPLP_SRGR1_REG[7:0] CLKGDV	x..x
Set the SRG Clock Synchronization Mode <sup>(1)</sup>	MCBSPLP_SRGR2_REG[15] GSYNC	0x0 or 0x1
Set the SRG Input Clock Mode <sup>(1)</sup>	MCBSPLP_PCR_REG[7] SCKLME MCBSPLP_SRGR2_REG[13] CLKSM	0x0 or 0x1
Set the SRG Input Clock Polarity <sup>(1)</sup>	MCBSPLP_SRGR2_REG[14] CLKSP MCBSPLP_PCR_REG[1] CLKXP MCBSPLP_PCR_REG[0] CLKRP	0x0 or 0x1

<sup>(1)</sup> Software decision

**NOTE:** CLKRP=CLKXP in a system where the same clock (internal or external) is used to clock the receiver and transmitter. The receiver uses the opposite edge as the transmitter to ensure valid setup and hold of data around this edge.

### 23.5.5.5.3 Take the Receiver Out of Reset (Step 3)

**Table 23-353. Take the Receiver Out of Reset**

Step	Register / Bit Field / Programming Model	Value
Enable the receiver	MCBSPLP_SPCR1_REG[0] RST	0x1

### 23.5.5.6 Transmitter Configuration

To configure the McBSP transmitter, perform the following steps:

- Step 1. Place the McBSP transmitter in reset
- Step 2. Program the McBSP registers for the desired transmitter operation
- Step 3. Take the transmitter out of reset

#### 23.5.5.6.1 Place the Transmitter in Reset (Step 1)

**Table 23-354. Transmitter Reset**

Step	Register / Bit Field / Programming Model	Value
Place the transmitter in Reset	MCBSPLP_SPCR2_REG[0] RST	0x0

#### 23.5.5.6.2 Programming the McBSP Registers for the Desired Transmitter Operation (Step 2)

Followings items detail important procedures to be performed when the software is configuring the McBSP transmitter.

- **Global Configuration**

**Table 23-355. Global Configuration**

Step	Register / Bit Field / Programming Model	Value
Set the transmitter Pins to Operate as McBSP Pins	MCBSPLP_PCR_REG[13] XIOEN	0x0
Enable/ Disable the Digital Loop Back Mode <sup>(1)</sup>	MCBSPLP_XCCR_REG[5] DLB	0x0 or 0x1
Enable/ Disable the Analog Loop Back Mode <sup>(1)</sup>	MCBSPLP_SPCR1_REG[5] ALB	0x0 or 0x1
Select clock used in the synchronous transmit-receive mode (McBSP4 only) <sup>(1)</sup>	CONTROL_MCBSPLP[30] ALBCTRLRX_CLKX	0x0 or 0x1
Select Frame Synchronization used in the synchronous transmit-receive mode (McBSP4 only) <sup>(1)</sup>	CONTROL_MCBSPLP[31] ALBCTRLRX_FSX	0x0 or 0x1
Enable/ Disable the Transmit Multichannel Selection Mode <sup>(1)</sup>	MCBSPLP_MCR1_REG[0] RCMCM	0x0 or 0x1

<sup>(1)</sup> Software decision

- **Data Configuration**

**Table 23-356. Data Configuration**

Step	Register / Bit Field / Programming Model	Value
Select single/dual-phase frame <sup>(1)</sup>	MCBSPLP_RCR2_REG[15] XPHASE	0x0 or 0x1
Set the Transmit Word Length(s) for Phase1 <sup>(1)</sup>	MCBSPLP_XCR1_REG[7:5] XWDLEN1	xxx
Set the Transmit Word Length(s) for Phase2 <sup>(1)</sup>	MCBSPLP_XCR2_REG[7:5] XWDLEN2	xxx
Set the Transmit Frame length <sup>(1)</sup>	MCBSPLP_XCR1_REG[14:8] XFRLEN1 MCBSPLP_XCR2_REG[14:8] XFRLEN2	xxx
Set the Transmit Reverse Mode <sup>(1)</sup>	MCBSPLP_XCR2_REG[4:3] XREVERSE	xx
Set the Transmit Data Delay <sup>(1)</sup>	MCBSPLP_XCR2_REG[1:0] XDATLY	xx
Set the extra delay (DX delay) mode <sup>(1)</sup>	MCBSPLP_SPCR1_REG[7] DXENA	xx
Select the inserted delay value <sup>(1)</sup>	MCBSPLP_XCCR_REG[13:12] DXENDLY	xx
Set the Interrupt Line	MCBSPLP_IRQENABLE_REG	x..x
Configure the Transmit buffer threshold value <sup>(1)</sup>	MCBSPLP_THRSH2_REG[6:0] XTHRESHOLD	x..x

<sup>(1)</sup> Software decision**NOTE:**

- When dual-phase is selected the number of words per phase must be set to one.
- If a single-phase frame is selected, XWDLEN1 selects the length for every serial word transmit in the frame. If a dual-phase frame is selected, XWDLEN1 and XWDLEN2 must be set to select the both length. These both bits can have values different.
- If a dual-phase is selected, the frame length must be two words.

The DXENA bit controls the delay enabler on the mcbasp\_dx pin. Set DXENA to enable an extra delay for turn-on time. Because this bit does not control the data itself, only the first bit is delayed (the delay is given by a combinatorial delay buffer). The inserted delay, 80 ps, 160 ps (default), 240 ps, or 320 ps, can be set using the McBSPi.MCBSPLP\_XCCR\_REG[13:12] DXENDLY bit field. If the mcbasp\_dx pins of multiple McBSP modules are tied together, ensure that DXENA = 1 to avoid having more than one McBSP transmitting at a time on the data line.

- **Frame-Sync Configuration**

**Table 23-357. Frame-Synchronization Configuration**

Step	Register / Bit Field / Programming Model	Value
Set the Transmit Frame-Sync Mode <sup>(1)</sup>	MCBSPLP_PCR_REG[11] FSXM MCBSPLP_SRGR2_REG[12] FSGM	0x0 or 0x1
Set the Transmit Frame-Sync Polarity <sup>(1)</sup>	MCBSPLP_PCR_REG[3] FSXP	0x0 or 0x1
Set the SRG Frame-Sync Period <sup>(1)</sup>	MCBSPLP_SRGR2_REG[11:0] FPER	x..x
Set the SRG Frame-Sync Pulse Width <sup>(1)</sup>	MCBSPLP_SRGR1_REG[15:8] FWID	x..x

<sup>(1)</sup> Software decision



- **Clock Configuration**

**Table 23-358. Clock Configuration**

Step	Register / Bit Field / Programming Model	Value
Set the Transmit Clock Mode <sup>(1)</sup>	MCBSPLP_PCR_REG[9] CLKXM	0x0 or 0x1
Set the Transmit Clock Polarity <sup>(1)</sup>	MCBSPLP_PCR_REG[1] CLKXP	0x0 or 0x1
Set the SRG Clock Divide-Down Value <sup>(1)</sup>	MCBSPLP_SRGR1_REG[7:0] CLKGDV	x..x
Set the SRG Clock Synchronization Mode <sup>(1)</sup>	MCBSPLP_SRGR2_REG[15] GSYNC	0x0 or 0x1
Set the SRG Input Clock Mode <sup>(1)</sup>	MCBSPLP_PCR_REG[7] SCKLME MCBSPLP_SRGR2_REG[13] CLKSM	0x0 or 0x1
Set the SRG Input Clock Polarity <sup>(1)</sup>	MCBSPLP_SRGR2_REG[14] CLKSP MCBSPLP_PCR_REG[1] CLKXP MCBSPLP_PCR_REG[0] CLKRP	0x0 or 0x1

<sup>(1)</sup> Software decision

**NOTE:** CLKRP=CLKXP in a system where the same clock (internal or external) is used to clock the receiver and transmitter. The receiver uses the opposite edge as the transmitter to ensure valid setup and hold of data around this edge.

### 23.5.5.6.3 Take the receiver out of reset (Step 3)

**Table 23-359. Take the Receiver Out of Reset**

Step	Register / Bit Field / Programming Model	Value
Enable the transmitter	MCBSPLP_SPCR2_REG[0] RST	0x1

### 23.5.5.7 General-Purpose I/O on the McBSP Pins (Legacy Only)

To use McBSP pins as general-purpose I/O pins rather than as serial port pins, follow these steps:

- **For the receive pins (mcbasp\_clkr, mcbasp\_fsr, and mcbasp\_dr as)**

**Table 23-360. Use Serial Receive Pins as GPIO Pins**

Step	Register / Bit Field / Programming Model	Value
Place the receiver in Reset	MCBSPLP_SPCR1_REG[0] RRST	0x0
Enabled GPIO for the serial port receiver	MCBSPLP_PCR_REG[12] RIOEN	0x1

- **For the transmitter pins (mcbasp\_clkx, mcbasp\_fsx, and mcbasp\_dx)**

**Table 23-361. Use Serial Transmit Pins as GPIO Pins**

Step	Register / Bit Field / Programming Model	Value
Place the transmit in Reset	MCBSPLP_SPCR2_REG[0] XRST	0x0
Enabled GPIO for the serial port transmitter	MCBSPLP_PCR_REG[12] XIOEN	0x1

- **For the external clock pins (mcbasp\_clks)**

For the mcbasp\_clks pin (common to all McBSP modules), all of the reset and I/O conditions must be met as follow:

**Table 23-362. Use External Clock Pin as GPIO Pin**

Step	Register / Bit Field / Programming Model	Value
Place the receiver in Reset	MCBSPLP_SPCR1_REG[0] RRST	0x0
Place the transmit in Reset	MCBSPLP_SPCR2_REG[0] XRST	0x0
Enabled GPIO for the serial port receiver	MCBSPLP_PCR_REG[12] RIOEN	0x1
Enabled GPIO for the serial port transmitter	MCBSPLP_PCR_REG[12] XIOEN	0x1

### 23.5.6 MCBSP Register Manual

Table 23-363 shows the base address and address space for the device module instances.

#### 23.5.6.1 MCBSP Instance Summary

**Table 23-363. MCBSP Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-A9 Private Access	Base Address DSP Private Access	Base Address L4-PER Interconnect	Size
MCBSP1	0x4902 2000	0x4012 2000	0x2 2000	N/A	4KB
MCBSP2	0x4902 4000	0x4012 4000	0x2 4000	N/A	4KB
MCBSP3	0x4902 6000	0x4012 6000	0x2 6000	N/A	4KB
MCBSP4	N/A	N/A	N/A	0x4809 6000	4KB

**NOTE:** Private Access is an access which is not using the L3/L4 interconnects.

#### 23.5.6.2 MCBSP Registers

##### CAUTION

The McBSP registers are limited to 32-bit data accesses. 16-bit and 8-bit are not allowed and can corrupt register content.

##### 23.5.6.2.1 MCBSP Register Summary

**Table 23-364. MCBSP1 Register Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
MCBSPLP_DRR_REG	R	32	0x0000 0000	0x4902 2000	0x4012 2000	0x2 2000
MCBSPLP_DXR_REG	W	32	0x0000 0008	0x4902 2008	0x4012 2008	0x2 2008
MCBSPLP_SPCR2_REG	RW	32	0x0000 0010	0x4902 2010	0x4012 2010	0x2 2010
MCBSPLP_SPCR1_REG	RW	32	0x0000 0014	0x4902 2014	0x4012 2014	0x2 2014
MCBSPLP_RCR2_REG	RW	32	0x0000 0018	0x4902 2018	0x4012 2018	0x2 2018
MCBSPLP_RCR1_REG	RW	32	0x0000 001C	0x4902 201C	0x4012 201C	0x2 201C
MCBSPLP_XCR2_REG	RW	32	0x0000 0020	0x4902 2020	0x4012 2020	0x2 2020
MCBSPLP_XCR1_REG	RW	32	0x0000 0024	0x4902 2024	0x4012 2024	0x2 2024
MCBSPLP_SRGR2_REG	RW	32	0x0000 0028	0x4902 2028	0x4012 2028	0x2 2028
MCBSPLP_SRGR1_REG	RW	32	0x0000 002C	0x4902 202C	0x4012 202C	0x2 202C
MCBSPLP_MCR2_REG	RW	32	0x0000 0030	0x4902 2030	0x4012 2030	0x2 2030
MCBSPLP_MCR1_REG	RW	32	0x0000 0034	0x4902 2034	0x4012 2034	0x2 2034
MCBSPLP_RCERA_REG	RW	32	0x0000 0038	0x4902 2038	0x4012 2038	0x2 2038
MCBSPLP_RCERB_REG	RW	32	0x0000 003C	0x4902 203C	0x4012 203C	0x2 203C
MCBSPLP_XCERA_REG	RW	32	0x0000 0040	0x4902 2040	0x4012 2040	0x2 2040
MCBSPLP_XCERB_REG	RW	32	0x0000 0044	0x4902 2044	0x4012 2044	0x2 2044
MCBSPLP_PCR_REG	RW	32	0x0000 0048	0x4902 2048	0x4012 2048	0x2 2048
MCBSPLP_RCERC_REG	RW	32	0x0000 004C	0x4902 204C	0x4012 204C	0x2 204C
MCBSPLP_RCERD_REG	RW	32	0x0000 0050	0x4902 2050	0x4012 2050	0x2 2050
MCBSPLP_XCERC_REG	RW	32	0x0000 0054	0x4902 2054	0x4012 2054	0x2 2054

**Table 23-364. McBSP1 Register Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
MCBSPLP_XCERD_REG	RW	32	0x0000 0058	0x4902 2058	0x4012 2058	0x2 2058
MCBSPLP_RCERE_REG	RW	32	0x0000 005C	0x4902 205C	0x4012 205C	0x2 205C
MCBSPLP_RCERF_REG	RW	32	0x0000 0060	0x4902 2060	0x4012 2060	0x2 2060
MCBSPLP_XCERE_REG	RW	32	0x0000 0064	0x4902 2064	0x4012 2064	0x2 2064
MCBSPLP_XCERF_REG	RW	32	0x0000 0068	0x4902 2068	0x4012 2068	0x2 2068
MCBSPLP_RCERG_REG	RW	32	0x0000 006C	0x4902 206C	0x4012 206C	0x2 206C
MCBSPLP_RCERH_REG	RW	32	0x0000 0070	0x4902 2070	0x4012 2070	0x2 2070
MCBSPLP_XCERG_REG	RW	32	0x0000 0074	0x4902 2074	0x4012 2074	0x2 2074
MCBSPLP_XCERH_REG	RW	32	0x0000 0078	0x4902 2078	0x4012 2078	0x2 2078
MCBSPLP_REV_REG	R	32	0x0000 007C	0x4902 207C	0x4012 207C	0x2 207C
MCBSPLP_RINTCLR_REG	RW	32	0x0000 0080	0x4902 2080	0x4012 2080	0x2 2080
MCBSPLP_XINTCLR_REG	RW	32	0x0000 0084	0x4902 2084	0x4012 2084	0x2 2084
MCBSPLP_ROVFLCLR_REG	RW	32	0x0000 0088	0x4902 2088	0x4012 2088	0x2 2088
MCBSPLP_SYSCONFIG_REG	RW	32	0x0000 008C	0x4902 208C	0x4012 208C	0x2 208C
MCBSPLP_THRSH2_REG	RW	32	0x0000 0090	0x4902 2090	0x4012 2090	0x2 2090
MCBSPLP_THRSH1_REG	RW	32	0x0000 0094	0x4902 2094	0x4012 2094	0x2 2094
MCBSPLP_IRQSTATUS_REG	RW	32	0x0000 00A0	0x4902 20A0	0x4012 20A0	0x2 20A0
MCBSPLP_IRQENABLE_REG	RW	32	0x0000 00A4	0x4902 20A4	0x4012 20A4	0x2 20A4
MCBSPLP_WAKEUPEN_REG	RW	32	0x0000 00A8	0x4902 20A8	0x4012 20A8	0x2 20A8
MCBSPLP_XCCR_REG	RW	32	0x0000 00AC	0x4902 20AC	0x4012 20AC	0x2 20AC
MCBSPLP_RCCR_REG	RW	32	0x0000 00B0	0x4902 20B0	0x4012 20B0	0x2 20B0
MCBSPLP_XBUFFSTAT_REG	R	32	0x0000 00B4	0x4902 20B4	0x4012 20B4	0x2 20B4
MCBSPLP_RBUFFSTAT_REG	R	32	0x0000 00B8	0x4902 20B8	0x4012 20B8	0x2 20B8

**Table 23-365. McBSP2 Register Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
MCBSPLP_DRR_REG	R	32	0x0000 0000	0x4902 4000	0x4012 4000	0x2 4000
MCBSPLP_DXR_REG	W	32	0x0000 0008	0x4902 4008	0x4012 4008	0x2 4008
MCBSPLP_SPCR2_REG	RW	32	0x0000 0010	0x4902 4010	0x4012 4010	0x2 4010
MCBSPLP_SPCR1_REG	RW	32	0x0000 0014	0x4902 4014	0x4012 4014	0x2 4014
MCBSPLP_RCR2_REG	RW	32	0x0000 0018	0x4902 4018	0x4012 4018	0x2 4018
MCBSPLP_RCR1_REG	RW	32	0x0000 001C	0x4902 401C	0x4012 401C	0x2 401C
MCBSPLP_XCR2_REG	RW	32	0x0000 0020	0x4902 4020	0x4012 4020	0x2 4020
MCBSPLP_XCR1_REG	RW	32	0x0000 0024	0x4902 4024	0x4012 4024	0x2 4024
MCBSPLP_SRGR2_REG	RW	32	0x0000 0028	0x4902 4028	0x4012 4028	0x2 4028
MCBSPLP_SRGR1_REG	RW	32	0x0000 002C	0x4902 402C	0x4012 402C	0x2 402C
MCBSPLP_MCR2_REG	RW	32	0x0000 0030	0x4902 4030	0x4012 4030	0x2 4030
MCBSPLP_MCR1_REG	RW	32	0x0000 0034	0x4902 4034	0x4012 4034	0x2 4034
MCBSPLP_RCERA_REG	RW	32	0x0000 0038	0x4902 4038	0x4012 4038	0x2 4038
MCBSPLP_RCERB_REG	RW	32	0x0000 003C	0x4902 403C	0x4012 403C	0x2 403C

**Table 23-365. MCBSP2 Register Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
MCBSPLP_XCERA_REG	RW	32	0x0000 0040	0x4902 4040	0x4012 4040	0x2 4040
MCBSPLP_XCERB_REG	RW	32	0x0000 0044	0x4902 4044	0x4012 4044	0x2 4044
MCBSPLP_PCR_REG	RW	32	0x0000 0048	0x4902 4048	0x4012 4048	0x2 4048
MCBSPLP_RCERC_REG	RW	32	0x0000 004C	0x4902 404C	0x4012 404C	0x2 404C
MCBSPLP_RCERD_REG	RW	32	0x0000 0050	0x4902 4050	0x4012 4050	0x2 4050
MCBSPLP_XCERC_REG	RW	32	0x0000 0054	0x4902 4054	0x4012 4054	0x2 4054
MCBSPLP_XCERD_REG	RW	32	0x0000 0058	0x4902 4058	0x4012 4058	0x2 4058
MCBSPLP_RCERE_REG	RW	32	0x0000 005C	0x4902 405C	0x4012 405C	0x2 405C
MCBSPLP_RCERF_REG	RW	32	0x0000 0060	0x4902 4060	0x4012 4060	0x2 4060
MCBSPLP_XCERE_REG	RW	32	0x0000 0064	0x4902 4064	0x4012 4064	0x2 4064
MCBSPLP_XCERF_REG	RW	32	0x0000 0068	0x4902 4068	0x4012 4068	0x2 4068
MCBSPLP_RCERG_REG	RW	32	0x0000 006C	0x4902 406C	0x4012 406C	0x2 406C
MCBSPLP_RCERH_REG	RW	32	0x0000 0070	0x4902 4070	0x4012 4070	0x2 4070
MCBSPLP_XCERG_REG	RW	32	0x0000 0074	0x4902 4074	0x4012 4074	0x2 4074
MCBSPLP_XCERH_REG	RW	32	0x0000 0078	0x4902 4078	0x4012 4078	0x2 4078
MCBSPLP_REV_REG	R	32	0x0000 007C	0x4902 407C	0x4012 407C	0x2 407C
MCBSPLP_RINTCLR_REG	RW	32	0x0000 0080	0x4902 4080	0x4012 4080	0x2 4080
MCBSPLP_XINTCLR_REG	RW	32	0x0000 0084	0x4902 4084	0x4012 4084	0x2 4084
MCBSPLP_ROVFLCLR_REG	RW	32	0x0000 0088	0x4902 4088	0x4012 4088	0x2 4088
MCBSPLP_SYSCONFIG_REG	RW	32	0x0000 008C	0x4902 408C	0x4012 408C	0x2 408C
MCBSPLP_THRSH2_REG	RW	32	0x0000 0090	0x4902 4090	0x4012 4090	0x2 4090
MCBSPLP_THRSH1_REG	RW	32	0x0000 0094	0x4902 4094	0x4012 4094	0x2 4094
MCBSPLP_IRQSTATUS_REG	RW	32	0x0000 00A0	0x4902 40A0	0x4012 40A0	0x2 40A0
MCBSPLP_IRQENABLE_REG	RW	32	0x0000 00A4	0x4902 40A4	0x4012 40A4	0x2 40A4
MCBSPLP_WAKEUPEN_REG	RW	32	0x0000 00A8	0x4902 40A8	0x4012 40A8	0x2 40A8
MCBSPLP_XCCR_REG	RW	32	0x0000 00AC	0x4902 40AC	0x4012 40AC	0x2 40AC
MCBSPLP_RCCR_REG	RW	32	0x0000 00B0	0x4902 40B0	0x4012 40B0	0x2 40B0
MCBSPLP_XBUFFSTAT_REG	R	32	0x0000 00B4	0x4902 40B4	0x4012 40B4	0x2 40B4
MCBSPLP_RBUFFSTAT_REG	R	32	0x0000 00B8	0x4902 40B8	0x4012 40B8	0x2 40B8

**Table 23-366. MCBSP3 Register Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
MCBSPLP_DRR_REG	R	32	0x0000 0000	0x4902 6000	0x4012 6000	0x2 6000
MCBSPLP_DXR_REG	W	32	0x0000 0008	0x4902 6008	0x4012 6008	0x2 6008
MCBSPLP_SPCR2_REG	RW	32	0x0000 0010	0x4902 6010	0x4012 6010	0x2 6010
MCBSPLP_SPCR1_REG	RW	32	0x0000 0014	0x4902 6014	0x4012 6014	0x2 6014
MCBSPLP_RCR2_REG	RW	32	0x0000 0018	0x4902 6018	0x4012 6018	0x2 6018
MCBSPLP_RCR1_REG	RW	32	0x0000 001C	0x4902 601C	0x4012 601C	0x2 601C
MCBSPLP_XCR2_REG	RW	32	0x0000 0020	0x4902 6020	0x4012 6020	0x2 6020
MCBSPLP_XCR1_REG	RW	32	0x0000 0024	0x4902 6024	0x4012 6024	0x2 6024

**Table 23-366. MCBSP3 Register Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
MCBSPLP_SRGR2_REG	RW	32	0x0000 0028	0x4902 6028	0x4012 6028	0x2 6028
MCBSPLP_SRGR1_REG	RW	32	0x0000 002C	0x4902 602C	0x4012 602C	0x2 602C
MCBSPLP_MCR2_REG	RW	32	0x0000 0030	0x4902 6030	0x4012 6030	0x2 6030
MCBSPLP_MCR1_REG	RW	32	0x0000 0034	0x4902 6034	0x4012 6034	0x2 6034
MCBSPLP_RCERA_REG	RW	32	0x0000 0038	0x4902 6038	0x4012 6038	0x2 6038
MCBSPLP_RCERB_REG	RW	32	0x0000 003C	0x4902 603C	0x4012 603C	0x2 603C
MCBSPLP_XCERA_REG	RW	32	0x0000 0040	0x4902 6040	0x4012 6040	0x2 6040
MCBSPLP_XCERB_REG	RW	32	0x0000 0044	0x4902 6044	0x4012 6044	0x2 6044
MCBSPLP_PCR_REG	RW	32	0x0000 0048	0x4902 6048	0x4012 6048	0x2 6048
MCBSPLP_RCERC_REG	RW	32	0x0000 004C	0x4902 604C	0x4012 604C	0x2 604C
MCBSPLP_RCERD_REG	RW	32	0x0000 0050	0x4902 6050	0x4012 6050	0x2 6050
MCBSPLP_XCERC_REG	RW	32	0x0000 0054	0x4902 6054	0x4012 6054	0x2 6054
MCBSPLP_XCERD_REG	RW	32	0x0000 0058	0x4902 6058	0x4012 6058	0x2 6058
MCBSPLP_RCERE_REG	RW	32	0x0000 005C	0x4902 605C	0x4012 605C	0x2 605C
MCBSPLP_RCERF_REG	RW	32	0x0000 0060	0x4902 6060	0x4012 6060	0x2 6060
MCBSPLP_XCERE_REG	RW	32	0x0000 0064	0x4902 6064	0x4012 6064	0x2 6064
MCBSPLP_XCERF_REG	RW	32	0x0000 0068	0x4902 6068	0x4012 6068	0x2 6068
MCBSPLP_RCERG_REG	RW	32	0x0000 006C	0x4902 606C	0x4012 606C	0x2 606C
MCBSPLP_RCERH_REG	RW	32	0x0000 0070	0x4902 6070	0x4012 6070	0x2 6070
MCBSPLP_XCERG_REG	RW	32	0x0000 0074	0x4902 6074	0x4012 6074	0x2 6074
MCBSPLP_XCERH_REG	RW	32	0x0000 0078	0x4902 6078	0x4012 6078	0x2 6078
MCBSPLP_REV_REG	R	32	0x0000 007C	0x4902 607C	0x4012 607C	0x2 607C
MCBSPLP_RINTCLR_REG	RW	32	0x0000 0080	0x4902 6080	0x4012 6080	0x2 6080
MCBSPLP_XINTCLR_REG	RW	32	0x0000 0084	0x4902 6084	0x4012 6084	0x2 6084
MCBSPLP_ROVFLCLR_REG	RW	32	0x0000 0088	0x4902 6088	0x4012 6088	0x2 6088
MCBSPLP_SYSCONFIG_REG	RW	32	0x0000 008C	0x4902 608C	0x4012 608C	0x2 608C
MCBSPLP_THRSH2_REG	RW	32	0x0000 0090	0x4902 6090	0x4012 6090	0x2 6090
MCBSPLP_THRSH1_REG	RW	32	0x0000 0094	0x4902 6094	0x4012 6094	0x2 6094
MCBSPLP_IRQSTATUS_REG	RW	32	0x0000 00A0	0x4902 60A0	0x4012 60A0	0x2 60A0
MCBSPLP_IRQENABLE_REG	RW	32	0x0000 00A4	0x4902 60A4	0x4012 60A4	0x2 60A4
MCBSPLP_WAKEUPEN_REG	RW	32	0x0000 00A8	0x4902 60A8	0x4012 60A8	0x2 60A8
MCBSPLP_XCCR_REG	RW	32	0x0000 00AC	0x4902 60AC	0x4012 60AC	0x2 60AC
MCBSPLP_RCCR_REG	RW	32	0x0000 00B0	0x4902 60B0	0x4012 60B0	0x2 60B0
MCBSPLP_XBUFFSTAT_REG	R	32	0x0000 00B4	0x4902 60B4	0x4012 60B4	0x2 60B4
MCBSPLP_RBUFFSTAT_REG	R	32	0x0000 00B8	0x4902 60B8	0x4012 60B8	0x2 60B8

**Table 23-367. MCBSP4 Register Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L4-PER Interconnect
MCBSPLP_DRR_REG	R	32	0x0000 0000	0x4809 6000
MCBSPLP_DXR_REG	W	32	0x0000 0008	0x4809 6008
MCBSPLP_SPCR2_REG	RW	32	0x0000 0010	0x4809 6010

**Table 23-367. MCBSP4 Register Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L4-PER Interconnect
MCBSPLP_SPCR1_REG	RW	32	0x0000 0014	0x4809 6014
MCBSPLP_RCR2_REG	RW	32	0x0000 0018	0x4809 6018
MCBSPLP_RCR1_REG	RW	32	0x0000 001C	0x4809 601C
MCBSPLP_XCR2_REG	RW	32	0x0000 0020	0x4809 6020
MCBSPLP_XCR1_REG	RW	32	0x0000 0024	0x4809 6024
MCBSPLP_SRGR2_REG	RW	32	0x0000 0028	0x4809 6028
MCBSPLP_SRGR1_REG	RW	32	0x0000 002C	0x4809 602C
MCBSPLP_MCR2_REG	RW	32	0x0000 0030	0x4809 6030
MCBSPLP_MCR1_REG	RW	32	0x0000 0034	0x4809 6034
MCBSPLP_RCERA_REG	RW	32	0x0000 0038	0x4809 6038
MCBSPLP_RCERB_REG	RW	32	0x0000 003C	0x4809 603C
MCBSPLP_XCERA_REG	RW	32	0x0000 0040	0x4809 6040
MCBSPLP_XCERB_REG	RW	32	0x0000 0044	0x4809 6044
MCBSPLP_PCR_REG	RW	32	0x0000 0048	0x4809 6048
MCBSPLP_RCERC_REG	RW	32	0x0000 004C	0x4809 604C
MCBSPLP_RCERD_REG	RW	32	0x0000 0050	0x4809 6050
MCBSPLP_XCERC_REG	RW	32	0x0000 0054	0x4809 6054
MCBSPLP_XCERD_REG	RW	32	0x0000 0058	0x4809 6058
MCBSPLP_RCERE_REG	RW	32	0x0000 005C	0x4809 605C
MCBSPLP_RCERF_REG	RW	32	0x0000 0060	0x4809 6060
MCBSPLP_XCERE_REG	RW	32	0x0000 0064	0x4809 6064
MCBSPLP_XCERF_REG	RW	32	0x0000 0068	0x4809 6068
MCBSPLP_RCERG_REG	RW	32	0x0000 006C	0x4809 606C
MCBSPLP_RCERH_REG	RW	32	0x0000 0070	0x4809 6070
MCBSPLP_XCERG_REG	RW	32	0x0000 0074	0x4809 6074
MCBSPLP_XCERH_REG	RW	32	0x0000 0078	0x4809 6078
MCBSPLP_REV_REG	R	32	0x0000 007C	0x4809 607C
MCBSPLP_RINTCLR_REG	RW	32	0x0000 0080	0x4809 6080
MCBSPLP_XINTCLR_REG	RW	32	0x0000 0084	0x4809 6084
MCBSPLP_ROVFLCLR_REG	RW	32	0x0000 0088	0x4809 6088
MCBSPLP_SYSCONFIG_REG	RW	32	0x0000 008C	0x4809 608C
MCBSPLP_THRSH2_REG	RW	32	0x0000 0090	0x4809 6090
MCBSPLP_THRSH1_REG	RW	32	0x0000 0094	0x4809 6094
MCBSPLP_IRQSTATUS_REG	RW	32	0x0000 00A0	0x4809 60A0
MCBSPLP_IRQENABLE_REG	RW	32	0x0000 00A4	0x4809 60A4
MCBSPLP_WAKEUPEN_REG	RW	32	0x0000 00A8	0x4809 60A8
MCBSPLP_XCCR_REG	RW	32	0x0000 00AC	0x4809 60AC
MCBSPLP_RCCR_REG	RW	32	0x0000 00B0	0x4809 60B0
MCBSPLP_XBUFFSTAT_REG	R	32	0x0000 00B4	0x4809 60B4
MCBSPLP_RBUFFSTAT_REG	R	32	0x0000 00B8	0x4809 60B8

**23.5.6.2.2 MCBSP Register Description**

**Table 23-368. MCBSPLP\_DRR\_REG**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP data receive register	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRR																															

Bits	Field Name	Description	Type	Reset
31:0	DRR	Data receive register	R	0x0000 0000

**Table 23-369. Register Call Summary for Register MCBSPLP\_DRR\_REG**

- Multichannel Buffered Serial Port (McBSP)
- [McBSP Data Transfer Process: \[0\] \[1\] \[2\] \[3\]](#)
  - [McBSP Exception/Error Conditions: \[4\] \[5\] \[6\]](#)
  - [MCBSP Registers: \[7\] \[8\] \[9\] \[10\]](#)

**Table 23-370. MCBSPLP\_DXR\_REG**

<b>Address Offset</b>	0x0000 0008	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP data transmit register	
<b>Type</b>	W	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DXR																															

Bits	Field Name	Description	Type	Reset
31:0	DXR	Data transmit register	W	0x0000 0000

**Table 23-371. Register Call Summary for Register MCBSPLP\_DXR\_REG**

- Multichannel Buffered Serial Port (McBSP)
- [McBSP Data Transfer Process: \[0\] \[1\] \[2\]](#)
  - [McBSP Exception/Error Conditions: \[3\] \[4\] \[5\] \[6\] \[7\]](#)
  - [Multi-channel Selection Modes: \[8\]](#)
  - [McBSP Initialization Procedure: \[9\]](#)
  - [MCBSP Registers: \[10\] \[11\] \[12\] \[13\]](#)

**Table 23-372. MCBSPLP\_SPCR2\_REG**

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP serial port control register 2	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FREE	SOFT	FRST	GRST	XINTM	XSYNCERR	XEMPTY	XRDY	XRST							

## Multichannel Buffered Serial Port (McBSP)

www.ti.com

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x000000
9	FREE	Free Running Mode (When this bit is set, the module ignores the Msuspend input) 0x0: Free running mode is disabled 0x1: Free running mode is enabled	RW	0
8	SOFT	Soft Bit 0x0: SOFT mode is disabled: the module stops its activity immediately following MSuspend assertion 0x1: SOFT mode is enabled: the module freezes its state after completion of the current operation when MSuspend is asserted	RW	0
7	FRST	Frame-Sync Generator Reset 0x0: Frame-synchronization logic is reset. Frame-sync signal FSG is not generated by the sample-rate generator 0x1: Frame-sync signal FSG is generated after (FPER+1) number of CLKG clocks; that is, all frame counters are loaded with their programmed values	RW	0
6	GRST	Sample-Rate Generator Reset 0x0: Sample rate generator is reset 0x1: Sample rate generator is pulled out of reset. CLKG is driven as per programmed value in sample rate generator registers (SRGR[1,2])	RW	0
5:4	XINTM	Transmit Interrupt Mode (legacy) 0x0: XINT is driven by XRDY 0x1: XINT generated by end-of-frame 0x2: XINT generated by a new frame synchronization 0x3: XINT generated by XSYNCERR	RW	0x0
3	XSYNCERR	Transmit Synchronization Error (writing 0 to this bit clear the legacy transmit interrupt if asserted due to XSYNCERR condition) 0x0: No synchronization error 0x1: Synchronization error detected by McBSP	RW	0
2	XEMPTY	Transmit Shift Register XSR Empty Read 0x0: XSR is empty Read 0x1: XSR is not empty	R	0
1	XRDY	Transmitter ready Read 0x0: Transmitter is not ready. Read 0x1: Transmitter is ready for new data in DXR	R	0
0	XRST	Transmitter reset. This resets and enables the transmitter. 0x0: The serial port transmitter is disabled and in reset state. 0x1: The serial port transmitter is enabled.	RW	0



**Table 23-373. Register Call Summary for Register MCBSP\_SPCR2\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Software Reset](#): [0] [1] [2]
- [McBSP Data Transfer Process](#): [3] [4] [5]
- [McBSP SRG](#): [6] [7] [8]
- [McBSP Exception/Error Conditions](#): [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19]
- [McBSP DMA Configuration](#): [20] [21]
- [Multi-channel Selection Modes](#): [22] [23] [24]
- [McBSP Initialization Procedure](#): [25] [26] [27] [28] [29] [30]
- [Reset and Initialization Procedure for the Sample Rate Generator](#): [31]
- [Transmitter Configuration](#): [32] [33]
- [General-Purpose I/O on the McBSP Pins \(Legacy Only\)](#): [34] [35]
- [MCBSP Registers](#): [36] [37] [38] [39]

**Table 23-374. MCBSP\_SPCR1\_REG**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	McBSPLP serial port control register 1		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ALB	RJUST	RESERVED					DXENA	RESERVED	RINTM	RSYNCERR	RFULL	RRDY	RRST		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15	ALB	Analog Loop Back Mode 0x0: Analog loop back mode disabled 0x1: Analog loop back mode enabled	RW	0
14:13	RJUST	Receive Sign-Extension and Justification Mode 0x0: Right-justify and zero-fill MSBs in DRR 0x1: Right-justify and sign-extend MSBs in DRR 0x2: Left-justify and zero-fill LSBs in DRR 0x3: Reserved	RW	0x0
12:8	RESERVED	Reserved	R	0x00
7	DXENA	DX Enabler 0x0: DX enabler is off 0x1: DX enabler is on	RW	0
6	RESERVED	Reserved	R	0
5:4	RINTM	Receive Interrupt Mode (legacy) 0x0: RINT driven by RRDY (that is, end of word) and end of frame in A-bis mode 0x1: RINT generated by end-of-block or end-of-frame in multichannel operation 0x2: RINT generated by a new frame synchronization 0x3: RINT generated by RSYNCERR	RW	0x0

## Multichannel Buffered Serial Port (McBSP)

www.ti.com

Bits	Field Name	Description	Type	Reset
3	RSYNCERR	Receive Synchronization Error (writing 0 to this bit clear the legacy receive interrupt if asserted due to RSYNCERR condition)  0x0: No synchronization error 0x1: Synchronization error detected by McBSP	RW	0
2	RFULL	Receive Shift Register (RSR) Full  Read 0x0: DRR is not read, RB is full and RSR is also full with new word Read 0x1: RB is not in overrun condition	R	0
1	RRDY	Receiver Ready  Read 0x0: Receiver is not ready Read 0x1: Receiver is ready with data to be read from DRR	R	0
0	RRST	Receiver reset. This resets and enables the receiver.  0x0: The serial port receiver is disabled and in reset state. 0x1: The serial port receiver is enabled.	RW	0

**Table 23-375. Register Call Summary for Register MCBSP\_LP\_SPCR1\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Software Reset: \[0\]](#)
- [McBSP Data Transfer Process: \[1\] \[2\] \[3\]](#)
- [McBSP SRG: \[4\]](#)
- [McBSP Exception/Error Conditions: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [McBSP DMA Configuration: \[15\] \[16\]](#)
- [Multi-channel Selection Modes: \[17\]](#)
- [McBSP Initialization Procedure: \[18\] \[19\] \[20\] \[21\]](#)
- [Receiver Configuration: \[22\] \[23\] \[24\] \[25\] \[26\]](#)
- [Transmitter Configuration: \[27\] \[28\]](#)
- [General-Purpose I/O on the McBSP Pins \(Legacy Only\): \[29\] \[30\]](#)
- [MCBSP Registers: \[31\] \[32\] \[33\] \[34\]](#)

**Table 23-376. MCBSP\_LP\_RCR2\_REG**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	McBSP_LP receive control register 2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RPHASE	RFRLLEN2						RWDLEN2	RREVERSE	RESERVED	RDATDLY					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15	RPHASE	Receive Phases  0x0: Single-phase frame 0x1: Dual-phase frame	RW	0

Bits	Field Name	Description	Type	Reset
14:8	RFLEN2	Receive Frame Length 2 Single-phase frame selected: RFLEN2=don't care Dual-phase frame selected: RFLEN2=000 0000 - 1 word per second phase (other values are reserved)	RW	0x00
7:5	RWDLEN2	Receive Word Length 2 0x0: 8 bits 0x1: 12 bits 0x2: 16 bits 0x3: 20 bits 0x4: 24 bits 0x5: 32 bits 0x6: Reserved (do not use) 0x7: Reserved (do not use)	RW	0x0
4:3	RREVERSE	Receive reverse mode. 0x0: Data transfer starts with MSB first. 0x1: Data transfer starts with LSB first. 0x2: Reserved (do not use) 0x3: Reserved (do not use)	RW	0x0
2	RESERVED	Reserved	R	0
1:0	RDATDLY	Receive Data Delay 0x0: 0-bit data delay 0x1: 1-bit data delay 0x2: 2-bit data delay 0x3: Reserved	RW	0x0

**Table 23-377. Register Call Summary for Register MCBSP\_LP\_RCR2\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Protocols and Data Formats: \[0\] \[1\] \[2\]](#)
- [McBSP Data Transfer Process: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [McBSP Exception/Error Conditions: \[18\]](#)
- [Multi-channel Selection Modes: \[19\]](#)
- [McBSP Initialization Procedure: \[20\] \[21\]](#)
- [Receiver Configuration: \[22\] \[23\] \[24\] \[25\] \[26\]](#)
- [Transmitter Configuration: \[27\]](#)
- [MCBSP Registers: \[28\] \[29\] \[30\] \[31\]](#)

**Table 23-378. MCBSP\_LP\_RCR1\_REG**

<b>Address Offset</b>	0x0000 001C																														
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>										<b>Instance</b>						MCBSP <sub>i</sub> (i = 1 to 4)														
<b>Description</b>	McBSP_LP receive control register 1																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RFLEN1				RWDLEN1				RESERVED											

## Multichannel Buffered Serial Port (McBSP)

www.ti.com

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14:8	RFRLN1	Receive Frame Length 1 Single-phase frame selected: RFRLN1=000 0000 - 1 word per frame RFRLN1=000 0001 - 2 words per frame RFRLN1=111 1111 - 128 words per frame Dual-phase frame selected: RFRLN1=000 0000 - 1 word per phase (other values are reserved)	RW	0x00
7:5	RWDLEN1	Receive Word Length 1  0x0: 8 bits 0x1: 12 bits 0x2: 16 bits 0x3: 20 bits 0x4: 24 bits 0x5: 32 bits 0x6: Reserved (do not use) 0x7: Reserved (do not use)	RW	0x0
4:0	RESERVED	Reserved	R	0x00

**Table 23-379. Register Call Summary for Register MCBSP\_LP\_RCR1\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Protocols and Data Formats: \[0\] \[1\]](#)
- [McBSP Data Transfer Process: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [Multi-channel Selection Modes: \[11\]](#)
- [McBSP Initialization Procedure: \[12\] \[13\]](#)
- [Receiver Configuration: \[14\] \[15\]](#)
- [MCBSP Registers: \[16\] \[17\] \[18\] \[19\]](#)

**Table 23-380. MCBSP\_LP\_XCR2\_REG**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	MCBSP <sub>i</sub> (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	McBSP_LP transmit control register 2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XPHASE	XFRLN2						XWDLEN2	XREVERSE	RESERVED	XDATDLY					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15	XPHASE	Transmit Phases 0x0: Single-phase frame 0x1: Dual-phase frame	RW	0
14:8	XFRLN2	Transmit Frame Length 2 Single-phase frame selected: XFRLN2=don't care Dual-phase frame selected: XFRLN2=000 0000 - 1 word per second phase (other values are reserved)	RW	0x00

Bits	Field Name	Description	Type	Reset
7:5	XWDLEN2	Transmit Word Length 2 0x0: 8 bits 0x1: 12 bits 0x2: 16 bits 0x3: 20 bits 0x4: 24 bits 0x5: 32 bits 0x6: Reserved (do not use) 0x7: Reserved (do not use)	RW	0x0
4:3	XREVERSE	Transmit reverse mode. 0x0: Data transfer starts with MSB first. 0x1: Data transfer starts with LSB first. 0x2: Reserved (do not use) 0x3: Reserved (do not use)	RW	0x0
2	RESERVED	Reserved	R	0
1:0	XDATDLY	Transmit Data Delay 0x0: 0-bit data delay 0x1: 1-bit data delay 0x2: 2-bit data delay 0x3: Reserved	RW	0x0

**Table 23-381. Register Call Summary for Register MCBSP\_LP\_XCR2\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Protocols and Data Formats: \[0\] \[1\] \[2\]](#)
- [McBSP Data Transfer Process: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [McBSP Exception/Error Conditions: \[19\]](#)
- [Multi-channel Selection Modes: \[20\]](#)
- [McBSP Initialization Procedure: \[21\] \[22\]](#)
- [Transmitter Configuration: \[23\] \[24\] \[25\] \[26\]](#)
- [MCBSP Registers: \[27\] \[28\] \[29\] \[30\]](#)

**Table 23-382. MCBSP\_LP\_XCR1\_REG**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	MCBSP <sub>i</sub> (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	McBSP_LP transmit control register 1		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XFRLN1						XWDLEN1			RESERVED						

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14:8	XFRLN1	Transmit Frame Length 1 Single-phase frame selected: XFRLN1=000 0000 - 1 word per frame XFRLN1=000 0001 - 2 words per frame XFRLN1=111 1111 - 128 words per frame Dual-phase frame selected: XFRLN1=000 0000 - 1 word per phase (other values are reserved)	RW	0x00

Bits	Field Name	Description	Type	Reset
7:5	XWDLEN1	Transmit Word Length 1 0x0: 8 bits 0x1: 12 bits 0x2: 16 bits 0x3: 20 bits 0x4: 24 bits 0x5: 32 bits 0x6: Reserved (do not use) 0x7: Reserved (do not use)	RW	0x0
4:0	RESERVED	Reserved	R	0x00

**Table 23-383. Register Call Summary for Register MCBSP\_LP\_XCR1\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Protocols and Data Formats: \[0\] \[1\]](#)
- [McBSP Data Transfer Process: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [Multi-channel Selection Modes: \[11\]](#)
- [McBSP Initialization Procedure: \[12\] \[13\]](#)
- [Transmitter Configuration: \[14\] \[15\]](#)
- [MCBSP Registers: \[16\] \[17\] \[18\] \[19\]](#)

**Table 23-384. MCBSP\_LP\_SRGR2\_REG**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	McBSP_LP sample rate generator register 2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																GSYNC	CLKSP	CLKSM	FSGM	FPER											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15	GSYNC	0x0: The sample rate generator clock (CLKG) is free running. 0x1: The sample rate generator clock (CLKG) is running. But CLKG is resynchronized and frame-sync signal (FSG) is generated only after detecting the receive frame-synchronization signal (FSR). Also, frame period, FPER, is a don't care because the period is dictated by the external frame-sync pulse.	RW	0
14	CLKSP	CLKS Polarity Clock Edge Select Only used when the external clock CLKS drives the sample rate generator clock (CLKSM=0). 0x0: Rising edge of CLKG and FSG. 0x1: Falling edge of CLKG and FSG.	RW	0
13	CLKSM	McBSP_LP Sample Rate Generator Clock Mode 0x0: SCLKME=0: Sample rate generator clock derived from the CLKS pin. SCLKME=1: Sample rate generator clock derived from the CLKRI pin. 0x1: SCLKME=0: Sample rate generator clock derived from the CPU clock. SCLKME=1: Sample rate generator clock derived from the CLKXI clock.	RW	1

Bits	Field Name	Description	Type	Reset
12	FSGM	Sample Rate Generator Transmit Frame-Synchronization Mode Used when FSXM=1 in the PCR.  0x0: Transmit frame-sync signal (FSX) is generated when transmit buffer is not empty When FSGM=0, FPER and FWID are used to determine the frame synchronization period and width (external FSX is gated by the buffer empty condition).  0x1: Transmit frame-sync signal driven by the sample rate generator frame-sync signal, FSG.	RW	0
11:0	FPER	Frame Period. This field plus 1 determines when the next frame-sync signal becomes active. Range: 1 to 4096 CLKG periods	RW	0x000

**Table 23-385. Register Call Summary for Register MCBSPPLP\_SRGR2\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Clock Configuration: \[0\] \[1\] \[2\] \[3\]](#)
- [McBSP Data Transfer Process: \[4\] \[5\]](#)
- [McBSP SRG: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\]](#)
- [McBSP Exception/Error Conditions: \[26\] \[27\] \[28\]](#)
- [McBSP Initialization Procedure: \[29\] \[30\]](#)
- [Reset and Initialization Procedure for the Sample Rate Generator: \[31\]](#)
- [Receiver Configuration: \[32\] \[33\] \[34\] \[35\] \[36\]](#)
- [Transmitter Configuration: \[37\] \[38\] \[39\] \[40\] \[41\]](#)
- [MCBSP Registers: \[42\] \[43\] \[44\] \[45\]](#)

**Table 23-386. MCBSPPLP\_SRGR1\_REG**

<b>Address Offset</b>	0x0000 002C	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPPLP sample rate generator register 1	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FWID						CLKGDV									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:8	FWID	Frame Width. This field plus 1 determines the width of the frame-sync pulse, FSG, during its active period. Range: 1 to 256 CLKG periods.	RW	0x00
7:0	CLKGDV	Sample Rate Generator Clock Divider This value is used as the divide-down number to generate the required sample rate generator clock frequency. Default value is 1.	RW	0x01

**Table 23-387. Register Call Summary for Register MCBSPPLP\_SRGR1\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Data Transfer Process: \[0\]](#)
- [McBSP SRG: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [McBSP Exception/Error Conditions: \[8\]](#)
- [McBSP Initialization Procedure: \[9\] \[10\]](#)
- [Reset and Initialization Procedure for the Sample Rate Generator: \[11\]](#)
- [Receiver Configuration: \[12\] \[13\]](#)
- [Transmitter Configuration: \[14\] \[15\]](#)
- [MCBSP Registers: \[16\] \[17\] \[18\] \[19\]](#)

**Table 23-388. MCBSP\_LP\_MCR2\_REG**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP multi channel register 2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							XMCME	XPBBLK	XPABLK	RESERVED	XMCM				

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x000000
9	XMCME	<p>0x0: 2-partition mode: Only partitions A and B are used. You can control up to 32 channels in the transmit multichannel selection mode selected with the XMCM bits. If XMCM = 01b or 10b, assign 16 channels to partition A with the XPABLK bits. Assign 16 channels to partition B with the XPBBLK bits. If XMCM = 11b (for symmetric transmission and reception), assign 16 channels to receive partition A with the RPABLK bits. Assign 16 channels to receive partition B with the RPBBLK bits. You control the channels with the appropriate transmit channel enable registers: XCERA: Channels in partition A XCERB: Channels in partition B</p> <p>0x1: 8-partition mode: All partitions (A through H) are used. You can control up to 128 channels in the transmit multichannel selection mode selected with the XMCM bits. You control the channels with the appropriate transmit channel enable registers: XCERA: Channels 0 through 15 XCERB: Channels 16 through 31 XCERC: Channels 32 through 47 XCERD: Channels 48 through 63 XCERE: Channels 64 through 79 XCERF: Channels 80 through 95 XCERG: Channels 96 through 111 XCERH: Channels 112 through 127</p>	RW	0
8:7	XPBBLK	<p>Transmit Partition B Block (legacy)</p> <p>0x0: Block 1. Channel 16 to channel 31 0x1: Block 3. Channel 48 to channel 63 0x2: Block 5. Channel 80 to channel 95 0x3: Block 7. Channel 112 to channel 127</p>	RW	0x0
6:5	XPABLK	<p>Transmit Partition A Block (legacy)</p> <p>0x0: Block 0. Channel 0 to channel 15 0x1: Block 2. Channel 32 to channel 47 0x2: Block 4. Channel 64 to channel 79 0x3: Block 6. Channel 96 to channel 111</p>	RW	0x0
4:2	RESERVED	Reserved	R	0x0



Bits	Field Name	Description	Type	Reset
1:0	XMCM	<p>Transmit Multichannel Selection Enable</p> <p>0x0: All channels enabled without masking (DX is always driven during transmission of data).</p> <p>0x1: All channels disabled and therefore masked by default. Required channels are selected by enabling XP(A/B)BLK and XCER(A/B) appropriately. Also, these selected channels are not masked and therefore DX is always driven.</p> <p>0x2: All channels enabled, but masked. Selected channels enabled via XP(A/B)BLK and XCER(A/B) are unmasked.</p> <p>0x3: All channels disabled and therefore masked by default. Required channels are selected by enabling RP(A/B)BLK and RCER(A/B) appropriately. Selected channels can be unmasked by RP(A/B)BLK and XCER(A/B). This mode is used for symmetric transmit and receive operation.</p>	RW	0x0

**Table 23-389. Register Call Summary for Register MCBSP\_LP\_MCR2\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [McBSP Initialization Procedure: \[11\]](#)
- [MCBSP Registers: \[12\] \[13\] \[14\] \[15\]](#)

**Table 23-390. MCBSP\_LP\_MCR1\_REG**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	MCBSP <sub>i</sub> (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	McBSPLP multi channel register 1		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RMCME	RPBBLK	RPABLK	RESERVED				RMCM								

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x000000
9	RMCME	<p>(legacy)</p> <p>0x0: 2-partition mode. Only partitions A and B are used. You can control up to 32 channels in the receive multichannel selection mode (RMCM = 1). Assign 16 channels to partition A with the RPABLK bits. Assign 16 channels to partition B with the RPBBLK bits. You control the channels with the appropriate receive channel enable registers: RCERA: Channels in partition A RCERB: Channels in partition B</p> <p>0x1: 8-partition mode: All partitions (A through H) are used. You can control up to 128 channels in the receive multichannel selection mode. You control the channels with the appropriate receive channel enable registers: RCERA: Channels 0 through 15 RCERB: Channels 16 through 31 RCERC: Channels 32 through 47 RCERD: Channels 48 through 63 RCERE: Channels 64 through 79 RCERF: Channels 80 through 95 RCERG: Channels 96 through 111 RCERH: Channels 112 through 127</p>	RW	0

## Multichannel Buffered Serial Port (McBSP)

www.ti.com

Bits	Field Name	Description	Type	Reset
8:7	RPBBLK	Receive Partition B Block (legacy) 0x0: Block 1. Channel 16 to channel 31 0x1: Block 3. Channel 48 to channel 63 0x2: Block 5. Channel 80 to channel 95 0x3: Block 7. Channel 112 to channel 127	RW	0x0
6:5	RPABLK	Receive Partition A Block (legacy) 0x0: Block 0. Channel 0 to channel 15 0x1: Block 2. Channel 32 to channel 47 0x2: Block 4. Channel 64 to channel 79 0x3: Block 6. Channel 96 to channel 111	RW	0x0
4:1	RESERVED	Reserved	R	0x0
0	RMCM	Receive Multichannel Selection Enable 0x0: All 128 channels 0x1: All channels disabled by default. Required channels are selected by enabling RP(A/B)BLK and RCER(A/B) appropriately	RW	0

**Table 23-391. Register Call Summary for Register MCBSP\_LP\_MCR1\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [McBSP Initialization Procedure: \[9\]](#)
- [Receiver Configuration: \[10\]](#)
- [Transmitter Configuration: \[11\]](#)
- [MCBSP Registers: \[12\] \[13\] \[14\] \[15\]](#)

**Table 23-392. MCBSP\_LP\_RCERA\_REG**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	MCBSP <sub>i</sub> (i = 1 to 4)																																																												
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>																																																														
<b>Description</b>	McBSP_LP receive channel enable register partition A																																																														
<b>Type</b>	RW																																																														
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="12">RCERA</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																RCERA											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
RESERVED																RCERA																																															
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																											
31:16	RESERVED	Reserved	R	0x0000																																																											
15:0	RCERA	Receive Channel Enable RCERA n=0 Disables reception of n-th channel in an even-numbered block in partition A RCERA n=1 Enables reception of n-th channel in an even-numbered block in partition A	RW	0x0000																																																											

**Table 23-393. Register Call Summary for Register MCBSP\_LP\_RCERA\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [MCBSP Registers: \[6\] \[7\] \[8\] \[9\]](#)

**Table 23-394. MCBSPLP\_RCERB\_REG**

<b>Address Offset</b>	0x0000 003C	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive channel enable register partition B	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RCERB															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	RCERB	Receive Channel Enable RCERB n=0 Disables reception of n-th channel in a even-numbered block in partition B RCERB n=1 Enables reception of n-th channel in a even-numbered block in partition B	RW	0x0000

**Table 23-395. Register Call Summary for Register MCBSPLP\_RCERB\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\] \[1\]](#)
- [MCBSP Registers: \[2\] \[3\] \[4\] \[5\]](#)

**Table 23-396. MCBSPLP\_XCERA\_REG**

<b>Address Offset</b>	0x0000 0040	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit channel enable register partition A	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XCERA															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	XCERA	Transmit Channel Enable XCERA n=0 Disables transmission of n-th channel in an event-numbered block in partition A XCERA n=1 Enables transmission of n-th channel in an event-numbered block in partition A	RW	0x0000

**Table 23-397. Register Call Summary for Register MCBSPLP\_XCERA\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [MCBSP Registers: \[8\] \[9\] \[10\] \[11\]](#)

**Table 23-398. MCBSPLP\_XCERB\_REG**

<b>Address Offset</b>	0x0000 0044	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit channel enable register partition B	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XCERB															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	XCERB	Transmit Channel Enable XCERB n=0 Disables transmission of n-th channel in an even-numbered block in partition B XCERB n=1 Enables transmission of n-th channel in an even-numbered block in partition B	RW	0x0000

**Table 23-399. Register Call Summary for Register MCBSPLP\_XCERB\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\] \[1\]](#)
- [McBSP Registers: \[2\] \[3\] \[4\] \[5\]](#)

**Table 23-400. MCBSPLP\_PCR\_REG**

<b>Address Offset</b>	0x0000 0048	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP pin control register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLE_EN	XIOEN	RIOEN	FSXM	FSRM	CLKXM	CLKRM	SCLKME	CLKS_STAT	DX_STAT	DR_STAT	FSXP	FSRP	CLKXP	CLKRP	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	IDLE_EN	Idle enable. This bit allows stopping all the clocks in the McBSPLP. (legacy)  0x0: The McBSP is running 0x1: The clocks in the McBSP are shut off when both IDLE_EN=1 and peripheral domain is in idle mode	RW	0
13	XIOEN	Transmit General Purpose I/O Mode only when XRST=0 in SPCR[1,2] (legacy)  0x0: DX, FSX and CLKX are configured as serial port pins and do not function as general-purpose I/Os. 0x1: DX pin is a general purpose output. FSX and CLKX are general purpose I/Os. These serial port pins do not perform serial port operation.	RW	0

Bits	Field Name	Description	Type	Reset
12	RIOEN	<p>Receive General Purpose I/O Mode when RRST=0 in SPCR[1,2] (legacy)</p> <p>0x0: DR, FSR, CLKR and CLKS are configured as serial port pins and do not function as general-purpose I/Os.</p> <p>0x1: DR and CLKS pins are general purpose inputs; FSR and CLKR are general purpose I/Os. These serial port pins do not perform serial port operation. The CLKS pin is affected by a combination of RRST and RIOEN signals of the receiver.</p>	RW	0
11	FSXM	<p>Transmit Frame-Synchronization Mode</p> <p>0x0: Frame-synchronization signal derived from an external source</p> <p>0x1: Frame synchronization is determined by the sample rate generator frame-synchronization mode bit FSGM in SRGR2.</p>	RW	0
10	FSRM	<p>Receive Frame-Synchronization Mode</p> <p>0x0: Frame-Synchronization pulses generated by an external device. FSR is an input pin.</p> <p>0x1: Frame synchronization generated internally by sample rate generator. FSR is an output pin except when GSYNC=1 in SRGR.</p>	RW	0
9	CLKXM	<p>Transmitter Clock Mode</p> <p>0x0: Transmitter clock is driven by an external clock with CLKX as an input pin.</p> <p>0x1: CLKX is an output pin and is driven by the internal sample rate generator.</p>	RW	0
8	CLKRM	<p>Receiver Clock Mode</p> <p>0x0: Case 1: Digital loop back mode not set (DLB=0) in SPCR1: Receive clock (CLKR) is an input driven by an external clock. Case 2: Digital loop back mode set (DLB=1) in SPCR1: Receive clock (not the CLKR pin) is driven by transmit clock (CLKX) which is based on the CLKXM bit in the PCR. CLKR pin is in high-impedance.</p> <p>0x1: Case 1: Digital loop back mode not set (DLB=0) in SPCR1: CLKR is an output pin and is driven by the internal sample rate generator. Case 2: Digital loop back mode set (DLB=1) in SPCR1: CLKR is an output pin and is driven by the transmit clock. The transmit clock is derived based on the CLKRM bit in the PCR.</p>	RW	0
7	SCLKME	<p>The frequency of CLKG is: <math>CLKG \text{ frequency} = (\text{Input clock frequency}) / (\text{CLKGDV} + 1)</math> SCLKME is used in conjunction with the CLKSM bit to select the input clock:</p> <p>0x0: CLKSM = 0: Signal on CLKS pin CLKSM = 1: CPU clock</p> <p>0x1: CLKSM = 0: Signal on CLKR pin CLKSM = 1: Signal on CLKX pin</p>	RW	0
6	CLKS_STAT	<p>CLKS pin status. Reflects value on CLKS pin when selected as a general purpose input. (legacy)</p> <p>Read 0x0: The signal on the CLKS pin is low</p> <p>Read 0x1: The signal on the CLKS pin is high</p>	R	0
5	DX_STAT	<p>DX pin status. Reflects value driven on to DX pin when selected as a general purpose output. (legacy)</p> <p>0x0: Drive the signal on the DX pin low</p> <p>0x1: Drive the signal on the DX pin high</p>	RW	0
4	DR_STAT	<p>DR pin status. Reflects value on DR pin when selected as a general purpose input. (legacy)</p> <p>Read 0x0: The signal on DR pin is low</p> <p>Read 0x1: The signal on DR pin is high</p>	R	0

## Multichannel Buffered Serial Port (McBSP)

www.ti.com

Bits	Field Name	Description	Type	Reset
3	FSXP	Transmit Frame-Synchronization Polarity 0x0: Frame-synchronization pulse FSX is active high 0x1: Frame-synchronization pulse FSX is active low	RW	0
2	FSRP	Receive Frame-Synchronization Polarity 0x0: Frame-synchronization pulse FSR is active high 0x1: Frame-synchronization pulse FSR is active low	RW	0
1	CLKXP	Transmit Clock Polarity 0x0: Transmit data driven on rising edge of CLKX 0x1: Transmit data driven on falling edge of CLKX	RW	0
0	CLKRP	Receive Clock Polarity 0x0: Receive data sampled on falling edge of CLKR 0x1: Receive data sampled on rising edge of CLKR	RW	0

**Table 23-401. Register Call Summary for Register MCBSP\_LP\_PCR\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Protocols and Data Formats: \[0\] \[1\]](#)
- [McBSP Clock Configuration: \[2\] \[3\] \[4\] \[5\]](#)
- [McBSP Power Management: \[6\] \[7\]](#)
- [McBSP Data Transfer Process: \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\]](#)
- [McBSP SRG: \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\] \[44\] \[45\] \[46\] \[47\] \[48\] \[49\] \[50\]](#)
- [McBSP Exception/Error Conditions: \[51\]](#)
- [McBSP Initialization Procedure: \[52\] \[53\]](#)
- [Reset and Initialization Procedure for the Sample Rate Generator: \[54\]](#)
- [Receiver Configuration: \[55\] \[56\] \[57\] \[58\] \[59\] \[60\] \[61\] \[62\]](#)
- [Transmitter Configuration: \[63\] \[64\] \[65\] \[66\] \[67\] \[68\] \[69\] \[70\]](#)
- [General-Purpose I/O on the McBSP Pins \(Legacy Only\): \[71\] \[72\] \[73\] \[74\]](#)
- [MCBSP Registers: \[75\] \[76\] \[77\] \[78\]](#)

**Table 23-402. MCBSP\_LP\_RCERC\_REG**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	MCBSP <sub>i</sub> (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	McBSP_LP receive channel enable register partition C		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RCERC															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	RCERC	Receive Channel Enable RCERC n=0 Disables reception of n-th channel in an even-numbered block in partition C RCERC n=1 Enables reception of n-th channel in an even-numbered block in partition C	RW	0x0000

**Table 23-403. Register Call Summary for Register MCBSP\_LP\_RCERC\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\]](#)
- [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-404. MCBSP\_LP\_RCERD\_REG**

<b>Address Offset</b>	0x0000 0050	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive channel enable register partition D	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RCERD															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	RCERD	Receive Channel Enable RCERD n=0 Disables reception of n-th channel in an even-numbered block in partition D RCERD n=1 Enables reception of n-th channel in an even-numbered block in partition D	RW	0x0000

**Table 23-405. Register Call Summary for Register MCBSP\_LP\_RCERD\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\]](#)
- [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-406. MCBSP\_LP\_XCERC\_REG**

<b>Address Offset</b>	0x0000 0054	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit channel enable register partition C	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XCERC															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	XCERC	Transmit Channel Enable XCERC n=0 Disables transmission of n-th channel in an event-numbered block in partition C XCERC n=1 Enables transmission of n-th channel in an event-numbered block in partition C	RW	0x0000

**Table 23-407. Register Call Summary for Register MCBSP\_LP\_XCERC\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\]](#)
- [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-408. MCBSPLP\_XCERD\_REG**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit channel enable register partition D		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XCERD															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	XCERD	Transmit Channel Enable XCERD n=0 Disables transmission of n-th channel in an even-numbered block in partition D XCERD n=1 Enables transmission of n-th channel in an even-numbered block in partition D	RW	0x0000

**Table 23-409. Register Call Summary for Register MCBSPLP\_XCERD\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\]](#)
- [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-410. MCBSPLP\_RCERE\_REG**

<b>Address Offset</b>	0x0000 005C		
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive channel enable register partition E		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RCERE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	RCERE	Receive Channel Enable RCERE n=0 Disables reception of n-th channel in an even-numbered block in partition E RCERE n=1 Enables reception of n-th channel in an even-numbered block in partition E	RW	0x0000

**Table 23-411. Register Call Summary for Register MCBSPLP\_RCERE\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\]](#)
- [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)



**Table 23-412. MCBSPLP\_RCERF\_REG**

<b>Address Offset</b>	0x0000 0060	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive channel enable register partition F	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RCERF															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	RCERF	Receive Channel Enable RCERF n=0 Disables reception of n-th channel in an even-numbered block in partition F RCERF n=1 Enables reception of n-th channel in an even-numbered block in partition F	RW	0x0000

**Table 23-413. Register Call Summary for Register MCBSPLP\_RCERF\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\]](#)
- [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-414. MCBSPLP\_XCERE\_REG**

<b>Address Offset</b>	0x0000 0064	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit channel enable register partition E	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XCERE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	XCERE	Transmit Channel Enable XCERE n=0 Disables transmission of n-th channel in an event-numbered block in partition E XCERE n=1 Enables transmission of n-th channel in an event-numbered block in partition E	RW	0x0000

**Table 23-415. Register Call Summary for Register MCBSPLP\_XCERE\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\]](#)
- [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-416. MCBSPLP\_XCERF\_REG**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit channel enable register partition F		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XCERF															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	XCERF	Transmit Channel Enable XCERF n=0 Disables transmission of n-th channel in an even-numbered block in partition F XCERF n=1 Enables transmission of n-th channel in an even-numbered block in partition F	RW	0x0000

**Table 23-417. Register Call Summary for Register MCBSPLP\_XCERF\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\]](#)
- [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-418. MCBSPLP\_RCERG\_REG**

<b>Address Offset</b>	0x0000 006C		
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive channel enable register partition G		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RCERG															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	RCERG	Receive Channel Enable RCERG n=0 Disables reception of n-th channel in an even-numbered block in partition G RCERG n=1 Enables reception of n-th channel in an even-numbered block in partition G	RW	0x0000

**Table 23-419. Register Call Summary for Register MCBSPLP\_RCERG\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\]](#)
- [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-420. MCBSP\_LP\_RCERH\_REG**

<b>Address Offset</b>	0x0000 0070	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive channel enable register partition H	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RCERH															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	RCERH	Receive Channel Enable RCERH n=0 Disables reception of n-th channel in an even-numbered block in partition H RCERH n=1 Enables reception of n-th channel in an even-numbered block in partition H	RW	0x0000

**Table 23-421. Register Call Summary for Register MCBSP\_LP\_RCERH\_REG**

- Multichannel Buffered Serial Port (McBSP)
- [Multi-channel Selection Modes: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
  - [MCBSP Registers: \[5\] \[6\] \[7\] \[8\]](#)

**Table 23-422. MCBSP\_LP\_XCERG\_REG**

<b>Address Offset</b>	0x0000 0074	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit channel enable register partition G	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XCERG															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	XCERG	Transmit Channel Enable XCERG n=0 Disables transmission of n-th channel in an event-numbered block in partition G XCERG n=1 Enables transmission of n-th channel in an event-numbered block in partition G	RW	0x0000

**Table 23-423. Register Call Summary for Register MCBSP\_LP\_XCERG\_REG**

- Multichannel Buffered Serial Port (McBSP)
- [Multi-channel Selection Modes: \[0\]](#)
  - [MCBSP Registers: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-424. MCBSPLP\_XCERH\_REG**

<b>Address Offset</b>	0x0000 0078	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit channel enable register partition H	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XCERH															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	XCERH	Transmit Channel Enable XCERH n=0 Disables transmission of n-th channel in an even-numbered block in partition H XCERH n=1 Enables transmission of n-th channel in an even-numbered block in partition H	RW	0x0000

**Table 23-425. Register Call Summary for Register MCBSPLP\_XCERH\_REG**

Multichannel Buffered Serial Port (McBSP)

- [Multi-channel Selection Modes: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [McBSP Registers: \[7\] \[8\] \[9\] \[10\]](#)

**Table 23-426. MCBSPLP\_REV\_REG**

<b>Address Offset</b>	0x0000 007C	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP Revision number register	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REV															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000000
7:0	REV	Revision number	R	0x11

**Table 23-427. Register Call Summary for Register MCBSPLP\_REV\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Registers: \[0\] \[1\] \[2\] \[3\]](#)

**Table 23-428. MCBSPLP\_RINTCLR\_REG**

<b>Address Offset</b>	0x0000 0080	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> McBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive interrupt clear	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RINTCLR																															

Bits	Field Name	Description	Type	Reset
31:0	RINTCLR	Read from this register will clear the IRQ generated by receive end-of-frame indication or MCBSP.LP.FSR detection. Write to this register has no effect. (legacy)	RW	0x0000 0000

**Table 23-429. Register Call Summary for Register MCBSP.LP\_RINTCLR\_REG**

Multichannel Buffered Serial Port (McBSP)

- [MCBSP Registers: \[0\] \[1\] \[2\] \[3\]](#)

**Table 23-430. MCBSP.LP\_XINTCLR\_REG**

<b>Address Offset</b>	0x0000 0084	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	McBSP.LP transmit interrupt clear (legacy)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XINTCLR																															

Bits	Field Name	Description	Type	Reset
31:0	XINTCLR	Read from this register will clear the IRQ generated by transmit end-of-frame indication or MCBSP.LP.FSX detection. Write to this register has no effect.	RW	0x0000 0000

**Table 23-431. Register Call Summary for Register MCBSP.LP\_XINTCLR\_REG**

Multichannel Buffered Serial Port (McBSP)

- [MCBSP Registers: \[0\] \[1\] \[2\] \[3\]](#)

**Table 23-432. MCBSP.LP\_ROVFLCLR\_REG**

<b>Address Offset</b>	0x0000 0088	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	McBSP.LP receive overflow interrupt clear		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROVFLCLR																															

Bits	Field Name	Description	Type	Reset
31:0	ROVFLCLR	Read from this register will clear the IRQ generated by the receive overflow condition. Write to this register has no effect.	RW	0x0000 0000

**Table 23-433. Register Call Summary for Register MCBSP.LP\_ROVFLCLR\_REG**

Multichannel Buffered Serial Port (McBSP)

- [MCBSP Registers: \[0\] \[1\] \[2\] \[3\]](#)

**Table 23-434. MCBSPi\_SYSCONFIG\_REG**

<b>Address Offset</b>	0x0000 008C		
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPi System Configuration register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLOCKACTIVITY	RESERVED			SIDLEMODE	ENAWAKEUP	SOFTRESET	RESERVED	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0000000
9:8	CLOCKACTIVITY	0x0: The MCBSPi_ICLK clock can be switched off. The PRCM functional clock can be switched off. 0x1: The MCBSPi_ICLK clock must be maintained during wakeup. The PRCM functional clock can be switched off. 0x2: The MCBSPi_ICLK clock can be switched off. The PRCM functional clock must be maintained during wakeup. 0x3: The MCBSPi_ICLK clock must be maintained during wakeup. The PRCM functional clock must be maintained during wakeup	RW	0x0
7:5	RESERVED	Reserved	R	0x0
4:3	SIDLEMODE	Slave interface power management, req/ack control: 0x0: Force-idle. An idle request is acknowledged unconditionally. 0x1: No-idle. An idle request is never acknowledged. 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module 0x3: Reserved	RW	0x0
2	ENAWAKEUP	WakeUp feature control: 0x0: WakeUp is disabled 0x1: WakeUp capability is enabled	RW	0
1	SOFTRESET	McBSPi global software reset 0x0: NO soft reset 0x1: Soft reset triggered	RW	0
0	RESERVED	Reserved	R	0

**Table 23-435. Register Call Summary for Register MCBSPi\_SYSCONFIG\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Software Reset: \[0\]](#)
- [McBSP Power Management: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [McBSP Initialization Procedure: \[7\]](#)
- [McBSP Registers: \[8\] \[9\] \[10\] \[11\]](#)

**Table 23-436. MCBSP\_LP\_THRSH2\_REG**

<b>Address Offset</b>	0x0000 0090	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit buffer threshold (DMA or IRQ trigger)	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x00000000
6:0	XTHRESHOLD	Transmit buffer threshold value. The DMA request (if enabled) of interrupt assertion (if enabled) will be triggered if the number of free locations inside transmit buffer are above or equal to the XTHRESHOLD value + 1. Also, this value (XTHRESHOLD value + 1) indicates the number of words transferred during a transmit data DMA request, if transmit DMA is enabled	RW	0x00

**Table 23-437. Register Call Summary for Register MCBSP\_LP\_THRSH2\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Power Management: \[0\]](#)
- [McBSP Data Transfer Process: \[1\]](#)
- [McBSP DMA Configuration: \[2\]](#)
- [McBSP Initialization Procedure: \[3\]](#)
- [Data Transfer DMA Request Configuration: \[4\]](#)
- [Transmitter Configuration: \[5\]](#)
- [MCBSP Registers: \[6\] \[7\] \[8\] \[9\]](#)

**Table 23-438. MCBSP\_LP\_THRSH1\_REG**

<b>Address Offset</b>	0x0000 0094	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive buffer threshold (DMA or IRQ trigger)	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x00000000
6:0	RTHRESHOLD	Receive buffer threshold value. The DMA request (if enabled) of interrupt assertion (if enabled) will be triggered if the number of occupied locations inside receive buffer are above or equal to the RTHRESHOLD value + 1. Also, this value (RTHRESHOLD value + 1) indicates the number of words transferred during a receive data DMA request, if receive DMA is enabled.	RW	0x00

**Table 23-439. Register Call Summary for Register MCBSP\_LP\_THRSH1\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Power Management: \[0\]](#)
- [McBSP Data Transfer Process: \[1\]](#)
- [McBSP DMA Configuration: \[2\]](#)
- [Data Transfer DMA Request Configuration: \[3\]](#)
- [Receiver Configuration: \[4\]](#)
- [MCBSP Registers: \[5\] \[6\] \[7\] \[8\]](#)

**Table 23-440. MCBSP\_LP\_IRQSTATUS\_REG**

<b>Address Offset</b>	0x0000 00A0	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>		
<b>Description</b>	MCBSP_LP Interrupt Status register (interconnect compliant IRQ line)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XEMPTYEOF	RESERVED	XOVFLSTAT	XUNDFLSTAT	XRDY	XEOF	XFSX	XSYNCERR	RESERVED	ROVFLSTAT	RUNDFLSTAT	RRDY	REOF	RFSR	RSYNCERR	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	XEMPTYEOF	Transmit Buffer Empty at end of frame (XEMPTYEOF is set to one when a complete frame was transmitted and the transmit buffer is empty).  0x0: XEMPTYEOF is NOT set to when a complete frame was transmitted and the transmit buffer is empty 0x1: XEMPTYEOF is set to one when a complete frame was transmitted and the transmit buffer is empty. Writing 1 to this bit clears the bit.	RW	0
13	RESERVED	Reserved	R	0
12	XOVFLSTAT	Transmit Buffer Overflow (XOVFLSTAT bit is set to one when transmit buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit clears the bit.  0x0: Transmit buffer NOT overflow 0x1: Transmit buffer overflow; Writing 1 to this bit clears the bit.	RW	0
11	XUNDFLSTAT	Transmit Buffer Underflow (XUNDFLSTAT bit is set to one when the transmit data buffer is empty new data is required to be transmitted). Writing 1 to this bit clears the bit.  0x0: the transmit data buffer is NOT empty new data is required to be transmitted. 0x1: the transmit data buffer is empty new data is required to be transmitted. Writing 1 to this bit clears the bit.	RW	0



Bits	Field Name	Description	Type	Reset
10	XRDY	<p>Transmit Buffer Threshold Reached (XRDY bit is set to one when the transmit buffer free locations are equal or above the THRSH2_REG value). Writing 1 to this bit clears the bit.</p> <p>0x0: Transmit buffer occupied locations are below the THRSH2_REG value).</p> <p>0x1: Transmit buffer occupied locations are equal or above the THRSH2_REG value). Writing 1 to this bit clears the bit.</p>	RW	0
9	XEOF	<p>Transmit End Of Frame (XEOF is set to one when a complete frame was transmitted). Writing 1 to this bit clears the bit.</p> <p>0x0: complete frame was NOT transmitted</p> <p>0x1: complete frame was transmitted; Writing 1 to this bit clears the bit.</p>	RW	0
8	XFSX	<p>Transmit Frame Synchronization (XFSX bit is set to one when a new transmit frame synchronization is asserted). Writing 1 to this bit clears the bit.</p> <p>0x0: new transmit frame synchronization is NOT asserted</p> <p>0x1: new transmit frame synchronization is asserted; Writing 1 to this bit clears the bit.</p>	RW	0
7	XSYNCERR	<p>Transmit Frame Synchronization Error (XSYNCERR is set to one when a transmit frame synchronization error is detected). Writing 1 to this bit clears the bit.</p> <p>0x0: Transmit frame synchronization error is NOT detected</p> <p>0x1: Transmit frame synchronization error is detected. Writing 1 to this bit clears the bit.</p>	RW	0
6	RESERVED	Reserved	R	0
5	ROVFLSTAT	<p>Receive Buffer Overflow (ROVFLSTAT bit is set to one when receive buffer overflow; the data which is written while overflow condition is discarded). Writing 1 to this bit clears the bit.</p> <p>0x0: receive buffer NOT overflow</p> <p>0x1: receive buffer overflow; Writing 1 to this bit clears the bit.</p>	RW	0
4	RUNDFLSTAT	<p>Receive Buffer Underflow (RUNDFLSTAT bit is set to one when read operation is performed to the receive data register while receive buffer is empty; data read while underflow condition is undefined). Writing 1 to this bit clears the bit.</p> <p>0x0: read operation is performed to the receive data register while receive buffer is NOT empty</p> <p>0x1: read operation is performed to the receive data register while receive buffer is empty; Writing 1 to this bit clears the bit.</p>	RW	0
3	RRDY	<p>Receive Buffer Threshold Reached (RRDY bit is set to one when the receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit clears the bit.</p> <p>0x0: receive buffer occupied locations are below the THRSH1_REG value).</p> <p>0x1: receive buffer occupied locations are equal or above the THRSH1_REG value). Writing 1 to this bit clears the bit.</p>	RW	0

Bits	Field Name	Description	Type	Reset
2	REOF	Receive End Of Frame (REOF is set to one when a complete frame was received). Writing 1 to this bit clears the bit.  0x0: complete frame was NOT received 0x1: complete frame was received; Writing 1 to this bit clears the bit.	RW	0
1	RFSR	Receive Frame Synchronization (RFSR bit is set to one when a new receive frame synchronization is asserted). Writing 1 to this bit clears the bit.  0x0: new receive frame synchronization is NOT asserted 0x1: new receive frame synchronization is asserted; Writing 1 to this bit clears the bit.	RW	0
0	RSYNCERR	Receive Frame Synchronization Error (RSYNCERR is set to one when a receive frame synchronization error is detected). Writing 1 to this bit clears the bit.  0x0: receive frame synchronization error is NOT detected 0x1: receive frame synchronization error is detected. Writing 1 to this bit clears the bit.	RW	0

**Table 23-441. Register Call Summary for Register MCBSP\_LP\_IRQSTATUS\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Power Management: \[0\] \[1\] \[2\] \[3\]](#)
- [McBSP Interrupt Requests: \[4\] \[5\]](#)
- [McBSP Data Transfer Process: \[6\] \[7\] \[8\] \[9\]](#)
- [McBSP Exception/Error Conditions: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\]](#)
- [MCBSP Registers: \[29\] \[30\] \[31\] \[32\]](#)

**Table 23-442. MCBSP\_LP\_IRQENABLE\_REG**

<b>Address Offset</b>	0x0000 00A4																																																																																
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSP <sub>i</sub> (i = 1 to 4)																																																																															
<b>Description</b>	McBSP_LP Interrupt Enable register (interconnect compliant IRQ line)																																																																																
<b>Type</b>	RW																																																																																
<table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> </thead> <tbody> <tr> <td colspan="16">RESERVED</td> <td>XEMPTYEOFEN</td> <td>RESERVED</td> <td>XOVFLEN</td> <td>XUNDFLEN</td> <td>XRDYEN</td> <td>XEOFEN</td> <td>XFSXEN</td> <td>XSYNCERREN</td> <td>RESERVED</td> <td>ROVFLEN</td> <td>RUNDFLEN</td> <td>RRDYEN</td> <td>REOFEN</td> <td>RFSREN</td> <td>RSYNCERREN</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																XEMPTYEOFEN	RESERVED	XOVFLEN	XUNDFLEN	XRDYEN	XEOFEN	XFSXEN	XSYNCERREN	RESERVED	ROVFLEN	RUNDFLEN	RRDYEN	REOFEN	RFSREN	RSYNCERREN																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																		
RESERVED																XEMPTYEOFEN	RESERVED	XOVFLEN	XUNDFLEN	XRDYEN	XEOFEN	XFSXEN	XSYNCERREN	RESERVED	ROVFLEN	RUNDFLEN	RRDYEN	REOFEN	RFSREN	RSYNCERREN																																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																																													
31:15	RESERVED	Reserved	R	0x00000																																																																													
14	XEMPTYEOFEN	Transmit buffer empty at end of frame enable bit.  0x0: Transmit Buffer Empty at End Of Frame NOT enabled 0x1: Transmit Buffer Empty at End Of Frame enabled	RW	0																																																																													
13	RESERVED	Reserved	R	0																																																																													
12	XOVFLEN	Transmit Buffer Overflow enable bit.  0x0: Transmit Buffer Overflow NOT enabled 0x1: Transmit Buffer Overflow enabled	RW	0																																																																													

Bits	Field Name	Description	Type	Reset
11	XUNDFLEN	Transmit Buffer Underflow enable bit. 0x0: Transmit Buffer Underflow NOT enabled 0x1: Transmit Buffer Underflow enabled	RW	0
10	XRDYEN	Transmit Buffer Threshold Reached enable bit. 0x0: Transmit Buffer Threshold Reached NOT enabled 0x1: Transmit Buffer Threshold Reached enabled	RW	0
9	XEOFEN	Transmit End Of Frame enable bit. 0x0: Transmit End Of Frame NOT enabled 0x1: Transmit End Of Frame enabled	RW	0
8	XFSXEN	Transmit Frame Synchronization enable bit. 0x0: Transmit Frame Synchronization NOT enabled 0x1: Transmit Frame Synchronization enabled	RW	0
7	XSYNCERREN	Transmit Frame Synchronization Error enable bit. 0x0: Transmit Frame Synchronization Error NOT enabled 0x1: Transmit Frame Synchronization Error enabled	RW	0
6	RESERVED	Reserved	R	0
5	ROVFLEN	Receive Buffer Overflow enable bit. 0x0: Receive Buffer Overflow NOT enabled 0x1: Receive Buffer Overflow enabled	RW	0
4	RUNDFLEN	Receive Buffer Underflow enable bit. 0x0: Receive Buffer Underflow NOT enabled 0x1: Receive Buffer Underflow enabled	RW	0
3	RRDYEN	Receive Buffer Threshold enable bit. 0x0: Receive Buffer Threshold NOT enabled 0x1: Receive Buffer Threshold enabled	RW	0
2	REOFEN	Receive End Of Frame enable bit. 0x0: Receive End Of Frame NOT enabled 0x1: Receive End Of Frame enabled	RW	0
1	RFSREN	Receive Frame Synchronization enable bit. RW 0x0: Receive Frame Synchronization NOT enabled 0x1: Receive Frame Synchronization enabled	RW	0
0	RSYNCERREN	Receive Frame Synchronization Error enable bit. 0x0: Receive Frame Synchronization Error NOT enabled 0x1: Receive Frame Synchronization Error enabled	RW	0

**Table 23-443. Register Call Summary for Register MCBSP\_LP\_IRQENABLE\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Power Management: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [McBSP Interrupt Requests: \[9\]](#)
- [McBSP Exception/Error Conditions: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [Interrupt Configuration: \[16\]](#)
- [Transmitter Configuration: \[17\]](#)
- [MCBSP Registers: \[18\] \[19\] \[20\] \[21\]](#)

**Table 23-444. MCBSPLP\_WAKEUPEN\_REG**

<b>Address Offset</b>	0x0000 00A8		
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP Wakeup Enable register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																XEMPTYEOFEN	RESERVED				XRDYEN	XEOFEN	XFSXEN	XSYNCERREN	RESERVED				RRDYEN	REOFEN	RFSREN	RSYNCERREN

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	XEMPTYEOFEN	Transmit Buffer Empty at End Of Frame enable bit. 0x0: Transmit Buffer Empty at End Of Frame WK enable is NOT active 0x1: Transmit Buffer Empty at End Of Frame WK enable is active	RW	0
13:11	RESERVED	Reserved	R	0x0
10	XRDYEN	Transmit Buffer Threshold Reached WK enable bit. 0x0: Transmit Buffer Threshold WK enable is NOT active 0x1: Transmit Buffer Threshold WK enable is active	RW	0
9	XEOFEN	Transmit End Of Frame WK enable bit. 0x0: Transmit End Of Frame WK enable is NOT active 0x1: Transmit End Of Frame WK enable is active	RW	0
8	XFSXEN	Transmit Frame Synchronization WK enable bit. 0x0: Transmit Frame Synchronization WK enable is NOT active 0x1: Transmit Frame Synchronization WK enable is active	RW	0
7	XSYNCERREN	Transmit Frame Synchronization Error WK enable bit. 0x0: Transmit Frame Synchronization Error WK enable is NOT active 0x1: Transmit Frame Synchronization Error WK enable is active	RW	0
6:4	RESERVED	Reserved	R	0x0
3	RRDYEN	Receive Buffer Threshold wakeup enable bit. 0x0: Receive Buffer Threshold WK enable is NOT active 0x1: Receive Buffer Threshold WK enable is active	RW	0
2	REOFEN	Receive End Of Frame WK enable bit. 0x0: Receive End Of Frame WK enable is NOT active 0x1: Receive End Of Frame WK enable is active	RW	0
1	RFSREN	Receive Frame Synchronization WK enable bit. 0x0: Receive Frame Synchronization WK enable is NOT active 0x1: Receive Frame Synchronization WK enable is active	RW	0

Bits	Field Name	Description	Type	Reset
0	RSYNCERREN	Receive Frame Synchronization Error WK enable bit. 0x0: Receive Frame Synchronization Error WK enable is NOT active 0x1: Receive Frame Synchronization Error WK enable is active	RW	0

**Table 23-445. Register Call Summary for Register MCBSP\_LP\_WAKEUPEN\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Power Management: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [MCBSP Registers: \[10\] \[11\] \[12\] \[13\]](#)

**Table 23-446. MCBSP\_LP\_XCCR\_REG**

<b>Address Offset</b>	0x0000 00AC	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSP_LP transmit configuration control register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EXTCLKGATE	PPCONNECT	DXENDLY	XFULL_CYCLE	RESERVED						DLB	RESERVED	XDMAEN	RESERVED	XDISABLE	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15	EXTCLKGATE	External clock gating enable (CLKX and FSX master only). When this bit is set and the transmit clock and FSX are set as output, the CLKX is enabled when FSX is active plus 3 clock cycles after (clock is provided for FWID + 4 clock cycles, assuming that the FSX width, active, is FWID + 1 clock cycles); outside this window the external transmit clock is gated. The receive use the same gated transmit clock and transmit frame synchronization signals regardless of the CLKRM/FSRM settings. When using this mode the frame synchronization signal must be active during reception of the entire frame (FWID must be programmed accordingly) to ensure the proper receive process, which requires at least 3 cycles after the frame complete to transfer the data into the receive buffer. 0x0: External clock gating disabled. 0x1: External clock gating enable.	RW	0
14	PPCONNECT	Pair to pair connection. When set the DXENO pin is always set to 0 regardless of the frame boundary, setting the tree state buffer as output 0x0: Pair to pair connection. When set the DXENO pin is always set to 0 regardless of the frame boundary, setting the tree state buffer as output 0x1: Pair to pair connection. When set the DXENO pin is always set to 0 regardless of the frame boundary, setting the tree state buffer as output	RW	0

## Multichannel Buffered Serial Port (McBSP)

www.ti.com

Bits	Field Name	Description	Type	Reset
13:12	DXENDLY	When DXENA bit in SPCR1 is set to one this field selects the added delay as follow: 0x0: 80 ps 0x1: 160 ps (default) 0x2: 240 ps 0x3: 320 ps	RW	0x1
11	XFULL_CYCLE	Transmit full cycle mode select. 0x0: McBSPLP operates in transmit half-cycle mode (transmit frame synchronization is sampled by the opposite edge of the clock used to drive transmit data). 0x1: McBSPLP operates in transmit full-cycle mode (transmit frame synchronization is sampled by the same edge of the clock used to drive transmit data).	RW	0
10:6	RESERVED	Reserved	R	0x00
5	DLB	Digital Loop-Back 0x0: No DLB 0x1: DLB	RW	0
4	RESERVED	Reserved	R	0
3	XDMAEN	Transmit DMA Enable bit. When set to zero this bit will gate the external transmit DMA request, without resetting the DMA state machine. It is recommended to change this bit value only during transmit reset. 0x0: When set to zero this bit will gate the external transmit DMA request, 0x1: When set to one this bit will NOT gate the external transmit DMA request,	RW	1
2:1	RESERVED	Reserved	R	0
0	XDISABLE	Transmit Disable bit. When this bit is set the transmit process will stop at the next frame boundary. 0x0: The transmit process will NOT stop at the next frame boundary. 0x1: The transmit process will stop at the next frame boundary.	RW	0

**Table 23-447. Register Call Summary for Register MCBSP\_LP\_XCCR\_REG**

## Multichannel Buffered Serial Port (McBSP)

- [McBSP DMA Requests: \[0\]](#)
- [McBSP Data Transfer Process: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [McBSP SRG: \[10\]](#)
- [McBSP DMA Configuration: \[11\]](#)
- [McBSP Basic Programming Model: \[12\]](#)
- [McBSP Initialization Procedure: \[13\]](#)
- [Receiver Configuration: \[14\]](#)
- [Transmitter Configuration: \[15\] \[16\]](#)
- [MCBSP Registers: \[17\] \[18\] \[19\] \[20\]](#)

**Table 23-448. MCBSP\_RCCR\_REG**

<b>Address Offset</b>	0x0000 00B0		
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b>	MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive configuration control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											RFULL_CYCLE	RESERVED						RDMAEN	RESERVED	RDISABLE											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11	RFULL_CYCLE	Receive full cycle mode select. 0x0: McBSPLP operates in receive half-cycle mode (receive frame synchronization is sampled by the opposite edge of the clock used to sample receive data). 0x1: McBSPLP operates in receive full-cycle mode (receive frame synchronization is sampled by the same edge of the clock used to sample receive data).	RW	1
10:4	RESERVED	Reserved	R	0x00
3	RDMAEN	Receive DMA Enable bit. When set to zero this bit will gate the external transmit DMA request, without resetting the DMA state machine. It is recommended to change this bit value only during receive reset. 0x0: When set to zero this bit will gate the external transmit DMA request 0x1: When set to one this bit will NOT gate the external transmit DMA request	RW	1
2:1	RESERVED	Reserved	R	0
0	RDISABLE	Receive Disable bit. When this bit is set the receive process will stop at the next frame boundary. 0x0: the receive process will NOT stop at the next frame boundary. 0x1: When this bit is set the receive process will stop at the next frame boundary.	RW	0

**Table 23-449. Register Call Summary for Register MCBSP\_RCCR\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP DMA Requests: \[0\]](#)
- [McBSP Data Transfer Process: \[1\] \[2\] \[3\] \[4\]](#)
- [McBSP DMA Configuration: \[5\]](#)
- [McBSP Basic Programming Model: \[6\]](#)
- [McBSP Registers: \[7\] \[8\] \[9\] \[10\]](#)

**Table 23-450. MCBSPLP\_XBUFFSTAT\_REG**

<b>Address Offset</b>	0x0000 00B4	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP transmit buffer status	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XBUFFSTAT															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:0	XBUFFSTAT	Transmit Buffer Status (indicates the number of free locations inside transmit buffer). The XBUFFSTAT value reflects the buffer status on the interface clock domain and it can be smaller than the number of free locations which are seen by the transmit state machine.	R	0x80

**Table 23-451. Register Call Summary for Register MCBSPLP\_XBUFFSTAT\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Data Transfer Process: \[0\]](#)
- [McBSP DMA Configuration: \[1\]](#)
- [MCBSP Registers: \[2\] \[3\] \[4\] \[5\]](#)

**Table 23-452. MCBSPLP\_RBUFFSTAT\_REG**

<b>Address Offset</b>	0x0000 00B8	
<b>Physical Address</b>	See <a href="#">Section 23.5.6.2.1</a>	<b>Instance</b> MCBSPi (i = 1 to 4)
<b>Description</b>	McBSPLP receive buffer status	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RBUFFSTAT															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:0	RBUFFSTAT	Receive Buffer Status (indicates the number of occupied locations inside receive buffer). The RBUFFSTAT value reflects the buffer status on the interface clock domain and it can be smaller than the real number of the occupied locations which are seen by the receive state machine.	R	0x00

**Table 23-453. Register Call Summary for Register MCBSPLP\_RBUFFSTAT\_REG**

Multichannel Buffered Serial Port (McBSP)

- [McBSP Data Transfer Process: \[0\]](#)
- [McBSP DMA Configuration: \[1\]](#)
- [MCBSP Registers: \[2\] \[3\] \[4\] \[5\]](#)

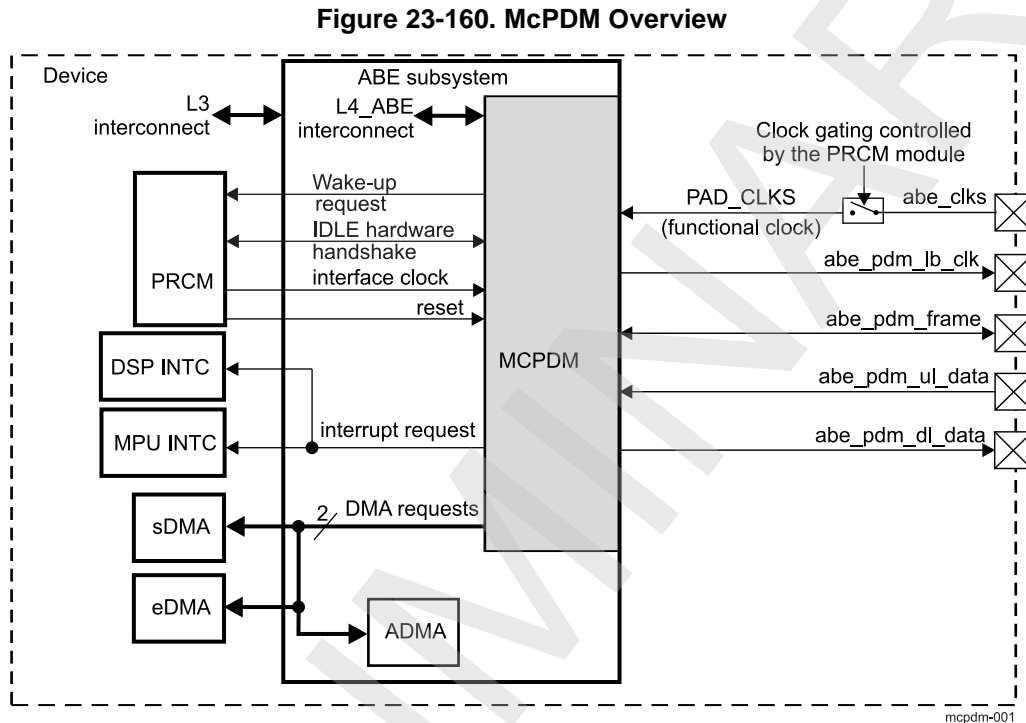


## 23.6 Multichannel PDM Controller

### 23.6.1 McPDM Overview

The multichannel pulse density modulation (McPDM) module consists in a proprietary audio module dedicated to mobile phone terminals. It is composed of audio uplink and downlink paths communicating with an external audio companion chip through a dedicated interface.

Figure 23-160 is an overview of the McPDM module.



The McPDM module can support the following features:

- Full duplex communication:
  - Five audio downlink channels with up to 3.84 Mbps each
  - Three audio uplink channels with up to 3.84 Mbps each
  - One command channel with up to 3.84 Mbps
  - One status channel with up to 3.84 Mbps
- Uplink and downlink first in first out (FIFO) operations:
  - 16 x 96-bit words for FIFO uplink, where a 96-bit word contains three 24-bit words for the audio uplink channels (one 24-bit word per audio uplink channel) and one 16-bit word for status information.
  - 16 x 144-bit words for FIFO downlink, where a 144-bit word contains five 24-bit words for the audio downlink channels (one 24-bit word per audio downlink channel) and one 16-bit word for the command data.
- Interrupt request to microprocessor unit (MPU) and digital signal processor (DSP) subsystems
- Direct memory access (DMA) requests with programmable FIFO thresholds:
  - One DMA request for downlink path
  - One DMA request for uplink path
- Up-sampling and pulse-density modulators for each audio downlink channel (five channels)
- Offset cancellation feature for audio downlink channels 1 and 2
- Decimation filter and downsampling for each audio uplink channel (three channels)
- Idle state hardware handshake

- Wake-up request

### 23.6.2 McPDM Environment

This section describes the McPDM application fields from an environment point of view (external connections). It describes McPDM connectivity options, lists all possible interfaces, and describes the protocol and data format used in each case.

#### 23.6.2.1 McPDM Modes

Table 23-454 describes the modes and application fields of the McPDM module.

**Table 23-454. Modes**

Mode	Description
Normal mode	Up to five audio channels are transmitted to, and up to three audio channels are received from, the audio companion chip.
Command mode	In addition to the audio channels, commands are transmitted to the audio companion chip.
Status mode	In addition to the audio channels, status information is received from the audio companion chip.

#### 23.6.2.2 McPDM Signals

Table 23-455 describes the McPDM module signals and specifies their links to modes.

**Table 23-455. I/O**

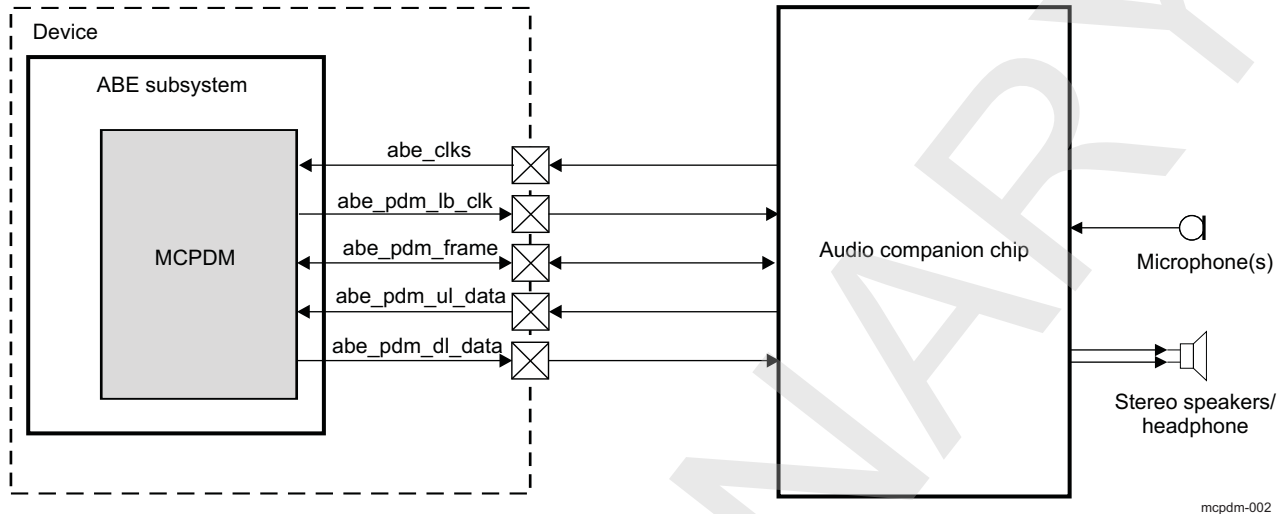
Signal	I/O (1)	Description	Reset Value	Normal Mode	Command Mode	Status Mode
abe_clks	I	Clock received from the audio companion chip. The clock frequency is: <ul style="list-style-type: none"> <li>• 17.64 MHz for a sampling frequency of 88.2/176.4 kHz</li> <li>• 19.2 MHz for a sampling frequency of 96/192 kHz</li> </ul>	-	x	x	x
abe_pdm_lb_clk	O	Clock loop-back to the audio companion chip from the abe_clks clock signal. The frequency is the same as abe_clks frequency	0	x	x	x
abe_pdm_frame	I/O	Frame synchronization and command/status data.	0	x	x	x
abe_pdm_ul_data	I	Audio uplink path data.	-	x	x	x
abe_pdm_dl_data	O	Audio downlink path data.	0	x	x	x

(1) I = Input; O = Output

### 23.6.2.3 McPDM Configurations

Figure 23-161 shows the McPDM generic configuration.

Figure 23-161. McPDM Generic Configuration



mcpdm-002

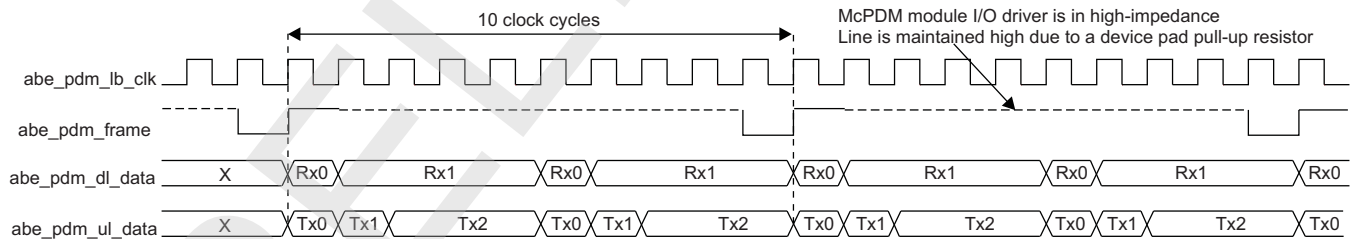
### 23.6.2.4 Protocols and Data Formats

#### 23.6.2.4.1 Protocols and Data Formats in Normal Mode

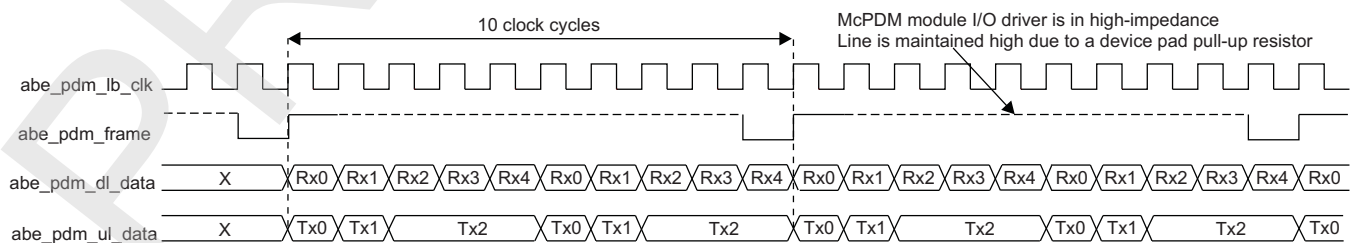
This section describes data transfer on the interface in normal mode. In this mode, the `abe_pdm_frame` signal is only used for frame synchronization purposes. Figure 23-162 shows the protocols and data formats in normal mode for two examples:

- (a) With two downlink and three uplink channels enabled
- (b) With five downlink and three uplink channels enabled

Figure 23-162. Protocols and Data Formats in Normal Mode



a) Normal mode with 2 downlink and 3 uplink channels enabled



b) Normal mode with 5 downlink and 3 uplink channels enabled

mcpdm-003

In normal mode, the frequency ratio between the `abe_pdm_frame` and `abe_pdm_lb_clk` signals is 10. This ratio is static and the `abe_pdm_frame` signal low-pulse width is one clock period longer than the `abe_pdm_lb_clk` signal.

The `abe_pdm_frame` signal is driven by the McPDM module in the device to the external audio companion chip. The McPDM module drives the `abe_pdm_frame` signal low during one clock period of the `abe_pdm_lb_clk` signal, then it drives it high during one clock period of the `abe_pdm_lb_clk` signal before releasing the `abe_pdm_frame` I/O driver in high-impedance state. The McPDM module pad internal pull-up resistor allows keeping a high-level state on the `abe_pdm_frame` line when the `abe_pdm_frame` I/O driver is in high-impedance state.

A maximum of two samples  $\times$  five downlink channels (10 samples) can be transmitted by the McPDM module to the external audio companion chip during a frame period.

A maximum of two samples  $\times$  three uplink channels (six samples) can be received by the McPDM module from the external audio companion chip during a frame period.

Each downlink and uplink channel can be independently enabled or disabled by setting to 1 or clearing to 0 the corresponding bit in the `MCPDM_CTRL` register.

#### 23.6.2.4.2 Protocols and Data Formats in Command Mode

This section describes data transfer on the interface in command mode. In this mode, the `abe_pdm_frame` signal is used to transmit command words to the external audio companion chip. It allows configuring audio registers in the external audio companion chip (that is, amplifier gains) without using the legacy inter-integrated circuit (I<sup>2</sup>C™) interface, which could be busy and add latency to gain correction.

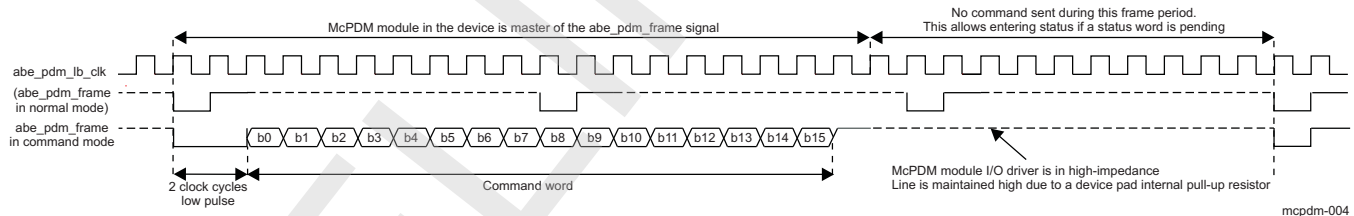
---

**NOTE:** In command mode, uplink channel samples are transmitted to and downlink channel samples are received from the external audio companion chip as in normal mode.

---

Figure 23-163 shows the protocols and data formats in command mode.

**Figure 23-163. Protocols and Data Formats in Command Mode**



##### 23.6.2.4.2.1 Entering the Command Mode

Command mode can be entered when command data are pending and no status mode is ongoing or detected.

If both of these conditions are met and the `MCPDM_CTRL`[9] `CMD_INT` bit is set to 1, the McPDM module automatically enters the command mode.

To enter the command mode, the McPDM module must drive the `abe_pdm_frame` signal low during two periods of the `abe_pdm_lb_clk` clock signal.

The external audio companion chip detects a command mode request from the McPDM module when a low state has been detected during two periods of the `abe_pdm_lb_clk` clock signal. Once a command mode request is detected, the external audio companion chip enters the command mode. In this mode, the status mode cannot be entered during at least four frame cycles.

If a status mode is detected or ongoing while a command is ready to be sent, the command data are stored in an internal register.

If one or more command data are written before the previous one is sent, the previous one is overwritten with the new one and only the last stored 16-bit command data are sent.

If a new command data are available while the system is already in command mode, the current command completes normally and exits before re-entering the command mode and transmitting the new command data a frame period later.

#### **23.6.2.4.2.2 Command Data Transmission**

The 16-bit command data are transmitted directly after the `abe_pdm_frame` signal is driven low during two periods of the `abe_pdm_lb_clk` clock signal.

When the `MCPDM_CTRL[9]` `CMD_INT` bit is set to 1, 16-bit command data must be written to the `MCPDM_DN_DATA` register. Only the 16 lower bits are significant. When no command must be transmitted on the `abe_pdm_frame` line, a null word (0x00000000) must be written as the command data.

When the `MCPDM_CTRL[9]` `CMD_INT` bit is cleared to 0, no 16-bit command data has to be written to the `MCPDM_DN_DATA` register.

#### **23.6.2.4.2.3 Exiting the Command Mode**

After the 16-bit command data has been sent, the McPDM module drives the `abe_pdm_frame` signal high during one period of the `abe_pdm_lb_clk` clock signal before releasing the `abe_pdm_frame` I/O driver in high-impedance state. The McPDM module pad internal pullup resistor allows keeping a high-level state on the `abe_pdm_frame` line when the `abe_pdm_frame` I/O driver is in high-impedance state.

The McPDM module exits automatically from command mode one period of the `abe_pdm_lb_clk` clock signal after the last bit of the command data has been transmitted on the `abe_pdm_frame` line.

---

**NOTE:** Due to this protocol mechanism, frame synchronization is lost (replaced by the ninth bit of the command data) during one frame period. The external audio companion chip must internally emulate a frame synchronization to send uplink channel samples and received downlink channel samples as in normal mode while receiving command data on the `abe_pdm_frame` line.

---

#### **23.6.2.4.2.4 Multiple Command Data Access**

As shown in [Figure 23-163](#), two consecutive command data can be sent every three frames. If no status mode is entered after a command data has been sent, the next command data is sent three frames after starting to send the current command data.

#### **23.6.2.4.3 Protocols and Data Formats in Status Mode**

This section describes data transfer on the interface in status mode. In this mode, the `abe_pdm_frame` signal is used to receive status information from the external audio companion chip.

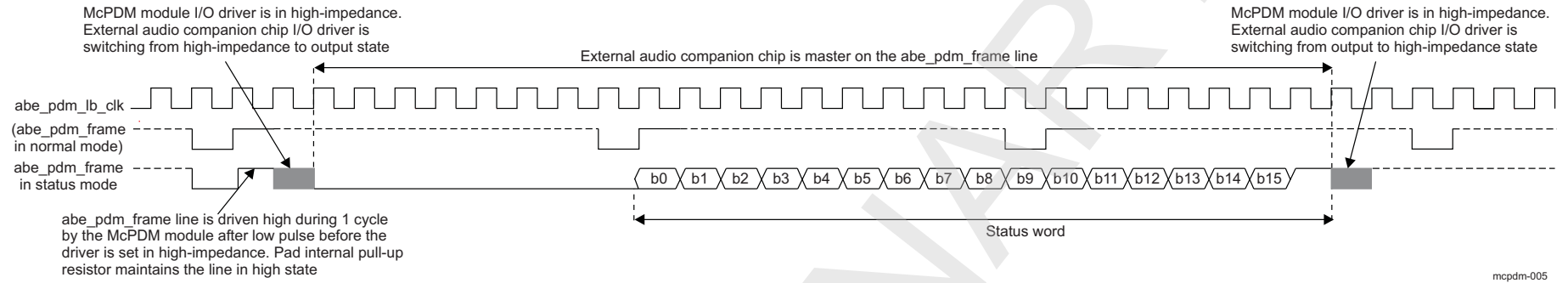
---

**NOTE:** In status mode, uplink channel samples are transmitted to and downlink channel samples are received from the external audio companion chip as in normal mode.

---

Figure 23-164 shows the protocols and data formats in status mode.

**Figure 23-164. Protocols and Data Formats in Status Mode**



mcpdm-005

#### **23.6.2.4.3.1 Entering the Status Mode**

Status mode can be entered when no command mode is detected during the last frame low pulse and no command mode is ongoing.

If both of these conditions are met and the `MCPDM_CTRL[10]` `STATUS_INT` bit is set to 1, the McPDM module automatically enters the status mode when the command mode is complete.

In status mode, the status information data are sent by the external audio companion chip to the McPDM module. The status information is stored in an internal register.

If one or more status information packages are received before the previous status information has been read, the status information stored in the internal register is overwritten with the new one.

In status mode, the external audio companion chip drives the `abe_pdm_frame` line low until the next frame synchronization low pulse. Then, it sends the 16-bit status information data on the `abe_pdm_frame` line. During the transmission of the status information, the McPDM module does not generate any frame synchronization low pulse on the `abe_pdm_frame` line for one frame period.

#### **23.6.2.4.3.2 Status Information Data Reception**

The 16-bit status information data are received directly after the second frame synchronization low pulse on the `abe_pdm_frame` line.

When the `MCPDM_CTRL[10]` `STATUS_INT` bit is set to 1, 16 bits of status information data must be read from the `MCPDM_UP_DATA` register. Only the 16 lower bits are significant. When the value of the status information is a null word (0x00000000), it means that no status information has been sent by the external audio companion chip.

When the `MCPDM_CTRL[10]` `STATUS_INT` bit is cleared to 0, no status information data has to be read from the `MCPDM_UP_DATA` register.

#### **23.6.2.4.3.3 Exiting the Status Mode**

After the 16-bit status information data has been sent, the external audio companion chip drives the `abe_pdm_frame` line high during one period of the `abe_pdm_lb_clk` clock signal before releasing its I/O driver in high-impedance state.

The McPDM module automatically exits the status mode one period of the `abe_pdm_lb_clk` clock signal after the last bit of the status information data has been transmitted on the `abe_pdm_frame` line. After exiting the status mode, the McPDM module generates frame synchronization low pulses.

---

**NOTE:** Due to the protocol mechanism, frame synchronization is lost during three frame periods. The external audio companion chip must internally emulate a frame synchronization to send uplink channel samples and received downlink channel samples as in normal mode while transmitting status information data on the `abe_pdm_frame` line.

---

#### **23.6.2.4.3.4 Multiple Status Information Data Access**

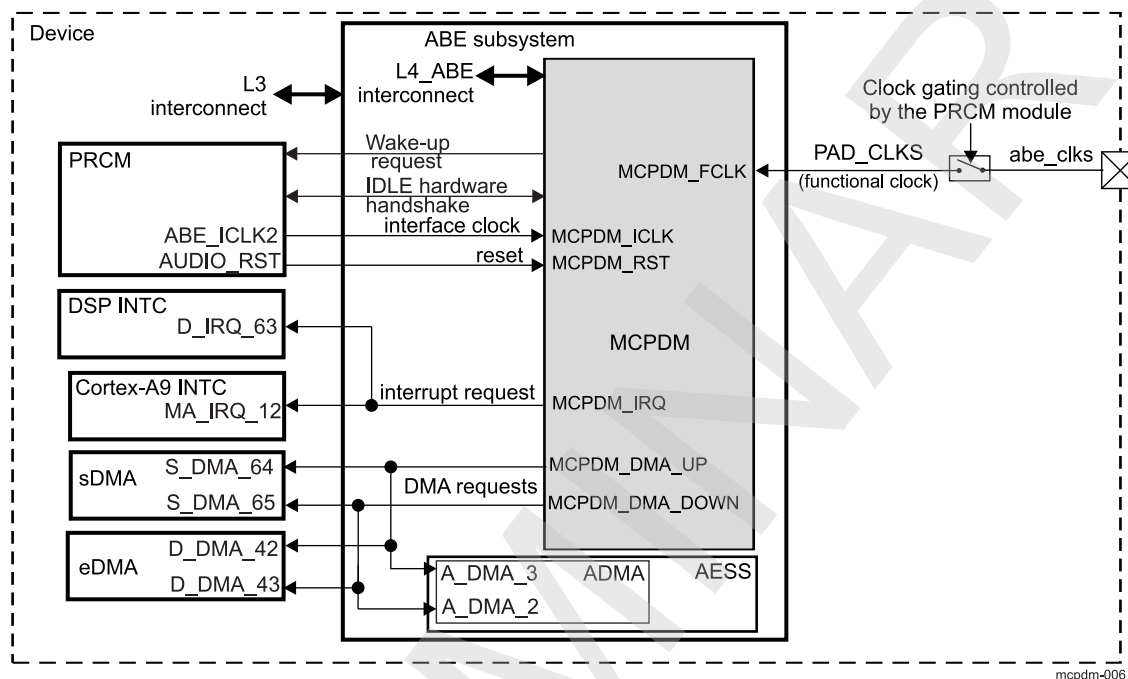
Multiple status information data can be sent by the external audio companion chip to the McPDM module every four frames.

### 23.6.3 McPDM Integration

This section describes the McPDM module integration in the device, including information about clocks, resets, and hardware requests.

Figure 23-165 shows the McPDM integration.

**Figure 23-165. McPDM Integration**



**NOTE:** For more information about the idle hardware handshake and the wake-up request, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

Table 23-456 through Table 23-458 summarize the integration of the module in the device.

**Table 23-456. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
MCPDM	PD_AUDIO	YES	L4_ABE

**Table 23-457. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
MCPDM	MCPDM_ICLK	ABE_ICLK2	PRCM module	McPDM interface clock
	MCPDM_FCLK	PAD_CLKS	PRCM module	Common external functional clock
Resets				
MCPDM	MCPDM_RST	AUDIO_RST	PRCM module	McPDM hardware reset

**Table 23-458. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description



**Table 23-458. Hardware Requests (continued)**

MCPDM	MCPDM_IRQ	D_IRQ_63	DSP	McPDM interrupt request to the DSP interrupt controller (INTC)
	MCPDM_IRQ	MA_IRQ_112	Cortex™-A9	McPDM interrupt request to the Cortex-A9 MPU INTC
<b>DMA Requests</b>				
MCPDM	MCPDM_DMA_UP	S_DMA_64	sDMA	McPDM uplink path DMA request to the system DMA (sDMA)
	MCPDM_DMA_DOWN	S_DMA_65	sDMA	McPDM downlink path DMA request to the sDMA
	MCPDM_DMA_UP	D_DMA_42	DDMA	McPDM uplink path DMA request to the eDMA
	MCPDM_DMA_DOWN	D_DMA_43	DDMA	McPDM downlink path DMA request to the eDMA
	MCPDM_DMA_UP	A_DMA_3	ADMA	McPDM uplink path DMA request to the ADMA
	MCPDM_DMA_DOWN	A_DMA_2	ADMA	McPDM downlink path DMA request to the ADMA

**NOTE:**

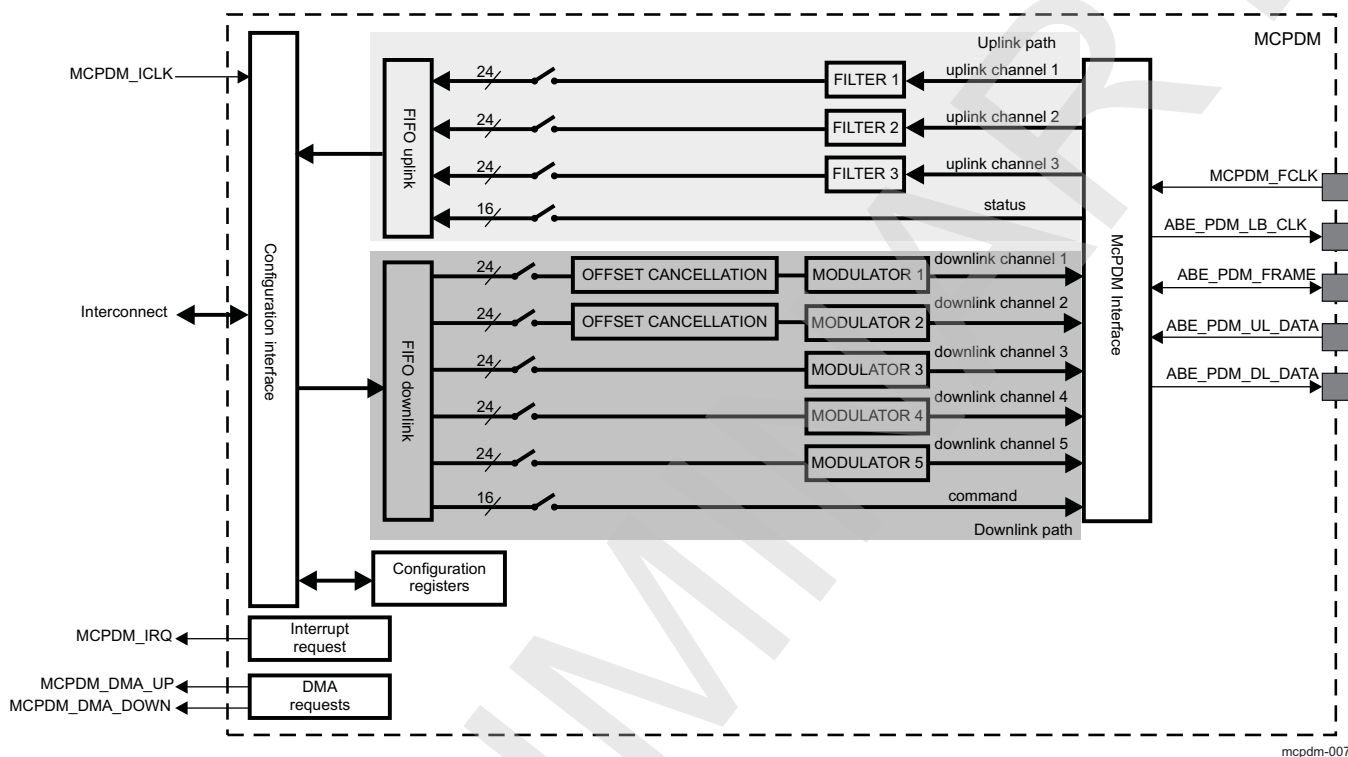
- For the description of the interrupt source, see [Section 23.6.4.5, Interrupt Requests](#).
- For the description of the DMA source, see [Section 23.6.4.6, DMA Requests](#).

## 23.6.4 McPDM Functional Description

### 23.6.4.1 McPDM Block Diagram

Figure 23-166 shows the McPDM block diagram.

Figure 23-166. McPDM Block Diagram



### 23.6.4.2 McPDM Clock Configuration

The functional clock MCPDM\_FCLK is provided by the external audio companion chip. Its frequency can be either 17.64 MHz for a sampling frequency of 88.2/176.4 kHz, or 19.2 MHz for a sampling frequency of 96/192 kHz.

The functional clock is also looped-back to the audio companion chip through ABE\_PDM\_LB\_CLK for resynchronization purposes.

### 23.6.4.3 McPDM Software Reset

To perform a software reset of the McPDM module, the MCPDM\_SYSCONFIG[0] SOFTRESET bit must be set to 1. Reading the MCPDM\_SYSCONFIG[0] SOFTRESET bit gives the status of the software reset:

- Read 1: The software reset is ongoing.
- Read 0: The software reset is complete.

Software must ensure that the software reset completes before doing McPDM operations. Moreover, uplink and downlink paths can be independently software reset by setting the MCPDM\_CTRL[11] SW\_UP\_RST bit for the uplink path, and the MCPDM\_CTRL[12] SW\_DN\_RST bit for the downlink path to 1. The software reset is stopped by clearing the MCPDM\_CTRL[11] SW\_UP\_RST bit for the uplink path, and the MCPDM\_CTRL[12] SW\_DN\_RST bit for the downlink path to 0.

The software reset of the McPDM module, after completion, put all McPDM registers in their default state. The downlink/uplink reset, resets only the downlink/uplink hardware logic (no registers are affected).

### 23.6.4.4 McPDM Power Management

Table 23-459 describes the power-management features available to the McPDM module.

**Table 23-459. Local Power-Management Features**

Feature	Registers	Description
Clock autogating	N/A	Feature not available
Slave idle modes	MCPDM_SYSCONFIG[3:2] IDLEMODE bit field	Force-idle, no-idle, smart-idle, and smart-idle wake-up capable modes are available for the interface clock
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	N/A	Feature not available
Wake-up sources enable	MCPDM_IRQWAKEEN and MCPDM_DMAWAKEEN registers	Each register holds one active-high enable bit per event source able to generate wake-up signal
Audio channel enable	MCPDM_CTRL[0] PDM_UP1_EN to MCPDM_CTRL[2] PDM_UP3_EN bits for audio uplink channels MCPDM_CTRL[3] PDM_DN1_EN to MCPDM_CTRL[7] PDM_DN5_EN bits for audio downlink channels	Each of these bits allows enabling (setting to 1) or disabling (clearing to 0) the corresponding audio channel
Status channel enable	MCPDM_CTRL[10] STATUS_INT bit for the status channel	This bit allows enabling (set to 1) or disabling (clear to 0) the status channel
Command channel enable	MCPDM_CTRL[9] CMD_INT bit for the command channel	This bit allows enabling (set to 1) or disabling (clear to 0) the command channel

The MCPDM\_ICLK interface clock is controlled through the power, reset, and clock management (PRCM) module by using idle request and acknowledge signals. The MCPDM\_SYSCONFIG[3:2] IDLEMODE bit field allows controlling the interface clock gating in idle mode.

For the uplink path, the interface clock can be put in idle after uplink path has been enabled. Wakeup is available according to the MCPDM\_IRQWAKEEN and MCPDM\_DMAWAKEEN register settings and the FIFO uplink threshold configured in the MCPDM\_FIFO\_CTRL\_UP[3:0] UP\_TRESH bit field. The wake-up signal is asserted when the FIFO uplink level is equal to, or greater than the FIFO uplink threshold value.

For the downlink path, the interface clock can be put in idle after the FIFO downlink level has reached at least once the FIFO downlink threshold. Wakeup is available according to the MCPDM\_IRQWAKEEN and MCPDM\_DMAWAKEEN register settings and the FIFO downlink threshold configured in the MCPDM\_FIFO\_CTRL\_DN[3:0] DN\_TRESH bit field. The wake-up signal is asserted when the FIFO downlink level is below the FIFO downlink threshold value.

### 23.6.4.5 McPDM Interrupt Requests

The McPDM can generate interrupt requests to the Cortex-A9 MPU and DSP interrupt controllers through the MCPDM\_IRQ signal.

Table 23-460 lists the nonmaskable and maskable event flags and their mask that can cause the McPDM module to generate interrupts.

Table 23-460. Interrupt Events

Nonmaskable Event Flag <sup>(1)</sup>	Maskable Event Flag	Event Mask Bit	Event Unmask Bit	Description
MCPDM_IRQSTATUS_RAW[11] UP_IRQ_FULL	MCPDM_IRQSTATUS[11] UP_IRQ_FULL	MCPDM_IRQENABLE_CLR[11] UP_IRQ_FULL_MASK	MCPDM_IRQENABLE_SET[11] UP_IRQ_FULL_MASK	FIFO-uplink-full event. This event occurs when FIFO uplink is already full and a new word is written to the FIFO uplink by the uplink part of the McPDM module.
MCPDM_IRQSTATUS_RAW[10] UP_IRQ_ALST_FULL	MCPDM_IRQSTATUS[10] UP_IRQ_ALST_FULL	MCPDM_IRQENABLE_CLR[10] UP_IRQ_ALST_FULL_MASK	MCPDM_IRQENABLE_SET[10] UP_IRQ_ALST_FULL_MASK	FIFO-uplink-almost-full event. This event occurs when FIFO uplink contains (FIFO uplink size - 1) words and a new word is written to the FIFO uplink by the uplink part of the McPDM module.
MCPDM_IRQSTATUS_RAW[9] UP_IRQ_EMPTY	MCPDM_IRQSTATUS[9] UP_IRQ_EMPTY	MCPDM_IRQENABLE_CLR[9] UP_IRQ_EMPTY_MASK	MCPDM_IRQENABLE_SET[9] UP_IRQ_EMPTY_MASK	FIFO-uplink-empty event. This event occurs when FIFO uplink is already empty and a new word is read from the FIFO uplink by the software.
MCPDM_IRQSTATUS_RAW[8] UP_IRQ	MCPDM_IRQSTATUS[8] UP_IRQ	MCPDM_IRQENABLE_CLR[8] UP_IRQ_MASK	MCPDM_IRQENABLE_SET[8] UP_IRQ_MASK	FIFO-uplink-read-request event. This event occurs when the number of words stored in the FIFO uplink is equal to or greater than the FIFO uplink threshold.
MCPDM_IRQSTATUS_RAW[3] DN_IRQ_FULL	MCPDM_IRQSTATUS[3] DN_IRQ_FULL	MCPDM_IRQENABLE_CLR[3] DN_IRQ_FULL_MASK	MCPDM_IRQENABLE_SET[3] DN_IRQ_FULL_MASK	FIFO-downlink-full event. This event occurs when FIFO downlink is already full and a new word is written to the FIFO downlink by the software.
MCPDM_IRQSTATUS_RAW[2] DN_IRQ_ALST_EMPTY	MCPDM_IRQSTATUS[2] DN_IRQ_ALST_EMPTY	MCPDM_IRQENABLE_CLR[2] DN_IRQ_ALST_EMPTY_MASK	MCPDM_IRQENABLE_SET[2] DN_IRQ_ALST_EMPTY_MASK	FIFO-downlink-almost-empty event. This event occurs when FIFO downlink contains one word and a new word is read from the FIFO downlink by the downlink part of the McPDM module.
MCPDM_IRQSTATUS_RAW[1] DN_IRQ_EMPTY	MCPDM_IRQSTATUS[1] DN_IRQ_EMPTY	MCPDM_IRQENABLE_CLR[1] DN_IRQ_EMPTY_MASK	MCPDM_IRQENABLE_SET[1] DN_IRQ_EMPTY_MASK	FIFO-downlink-empty event. This event occurs when FIFO downlink is empty and a new word is read from the FIFO downlink by the downlink part of the McPDM module.
MCPDM_IRQSTATUS_RAW[0] DN_IRQ	MCPDM_IRQSTATUS[0] DN_IRQ	MCPDM_IRQENABLE_CLR[0] DN_IRQ_MASK	MCPDM_IRQENABLE_SET[0] DN_IRQ_MASK	FIFO-downlink-write-request event. This event occurs when the number of words stored in the FIFO downlink is less than the FIFO downlink threshold.

<sup>(1)</sup> The MCPDM\_IRQSTATUS\_RAW register is mainly used for debug purposes.

### CAUTION

Once an event generating the interrupt request has been processed by the software, this event must be cleared by writing a logical 1 to the corresponding bit of the **MCPDM\_IRQSTATUS** register.

Writing a logical 1 in a bit of the **MCPDM\_IRQSTATUS** register also clears the corresponding bit in the **MCPDM\_IRQSTATUS\_RAW** register to 0.

An event can generate an interrupt request when a logical 1 is written to the corresponding mask bit in the **MCPDM\_IRQENABLE\_SET** register. Events are reported in the **MCPDM\_IRQSTATUS** and **MCPDM\_IRQSTATUS\_RAW** registers.

An event stops generating interrupt requests when a logical 1 is written to the corresponding mask bit in the `MCPDM_IRQENABLE_CLR` register. Events are only reported in the `MCPDM_IRQSTATUS_RAW` register.

In case of the `MCPDM_IRQSTATUS_RAW` register, the event is reported in the corresponding bit, even if the interrupt request generation is disabled for this event.

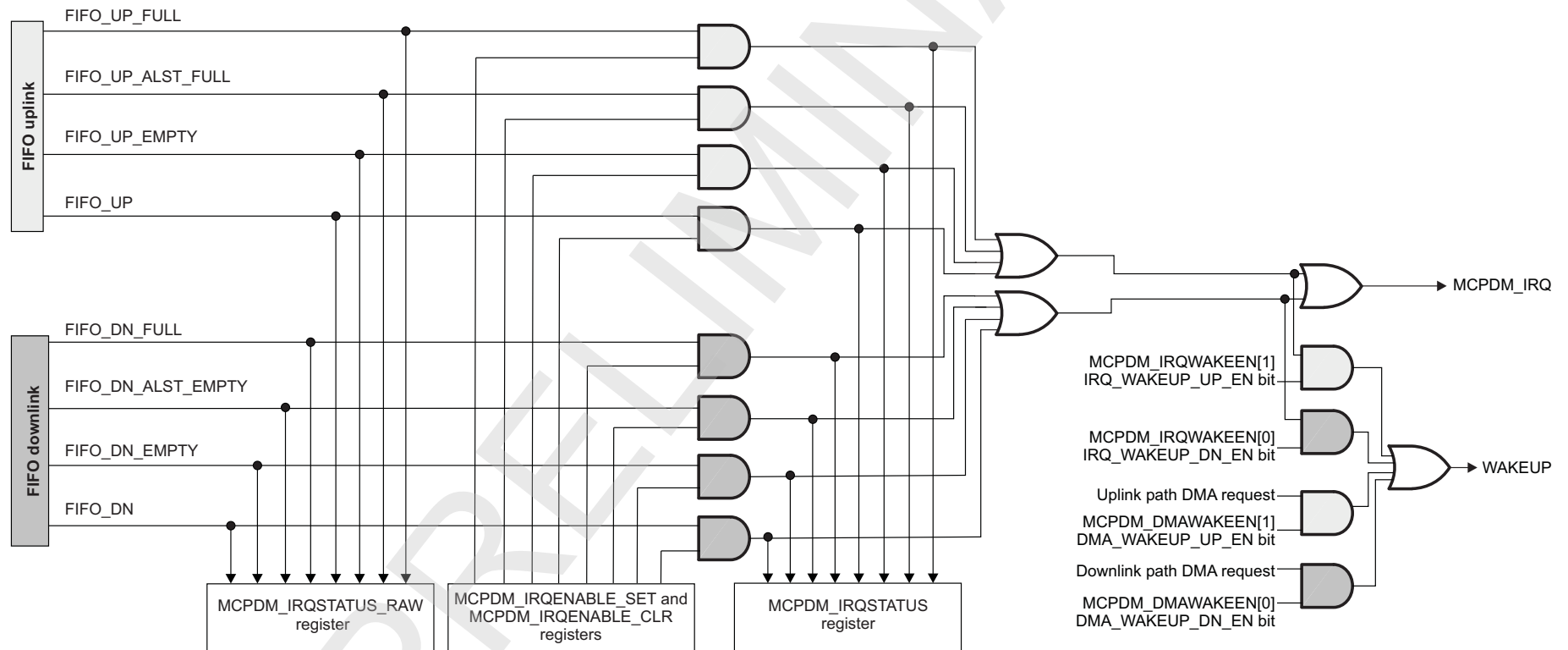
Writing a logical 1 in a bit of the `MCPDM_IRQSTATUS_RAW` register sets the corresponding bit in the `MCPDM_IRQSTATUS` register to 1.

The `MCPDM_IRQSTATUS_RAW` register is mostly used for debug purposes.

For more information about interrupt request signal mapping, see [Table 23-458](#).

[Figure 23-167](#) shows the interrupt tree of the McPDM module.

**Figure 23-167. Interrupt Tree**



mcpdm-008

### 23.6.4.6 McPDM DMA Requests

The McPDM module can generate two DMA requests to the system DMA (sDMA), audio DMA (ADMA), and DSP DMA (eDMA) controllers:

- The uplink path can generate a DMA request through the MCPDM\_DMA\_UP signal.
- The downlink path can generate a DMA request through the MCPDM\_DMA\_DOWN signal.

[Table 23-461](#) lists the DMA requests of the McPDM module.

**Table 23-461. McPDM DMA Requests**

DMA Request Signal	DMA Request Generation Enabling Bit	DMA Request Generation Disabling Bit	Description
MCPDM_DMA_UP	<a href="#">MCPDM_DMAENABLE_SET</a> [1] DMA_UP_ENABLE	<a href="#">MCPDM_DMAENABLE_CLR</a> [1] DMA_UP_ENABLE	Uplink path DMA request. This request is generated when the number of words in the FIFO uplink is equal to, or above the FIFO uplink threshold.
MCPDM_DMA_DOWN	<a href="#">MCPDM_DMAENABLE_SET</a> [0] DMA_DN_ENABLE	<a href="#">MCPDM_DMAENABLE_CLR</a> [0] DMA_DN_ENABLE	Downlink path DMA request. This request is generated when the number of words in the FIFO downlink is below the FIFO downlink threshold.

To enable or disable the DMA request generation on the MCPDM\_DMA\_UP signal, one of the following actions must be performed:

- Write a logical 1 to the [MCPDM\\_DMAENABLE\\_SET](#)[1] DMA\_UP\_ENABLE bit to enable the DMA request generation on the MCPDM\_DMA\_UP signal.
- Write a logical 1 to the [MCPDM\\_DMAENABLE\\_CLR](#)[1] DMA\_UP\_ENABLE bit to disable the DMA request generation on the MCPDM\_DMA\_UP signal.

To enable or disable the DMA request generation on the MCPDM\_DMA\_DOWN signal, one of the following actions must be performed:

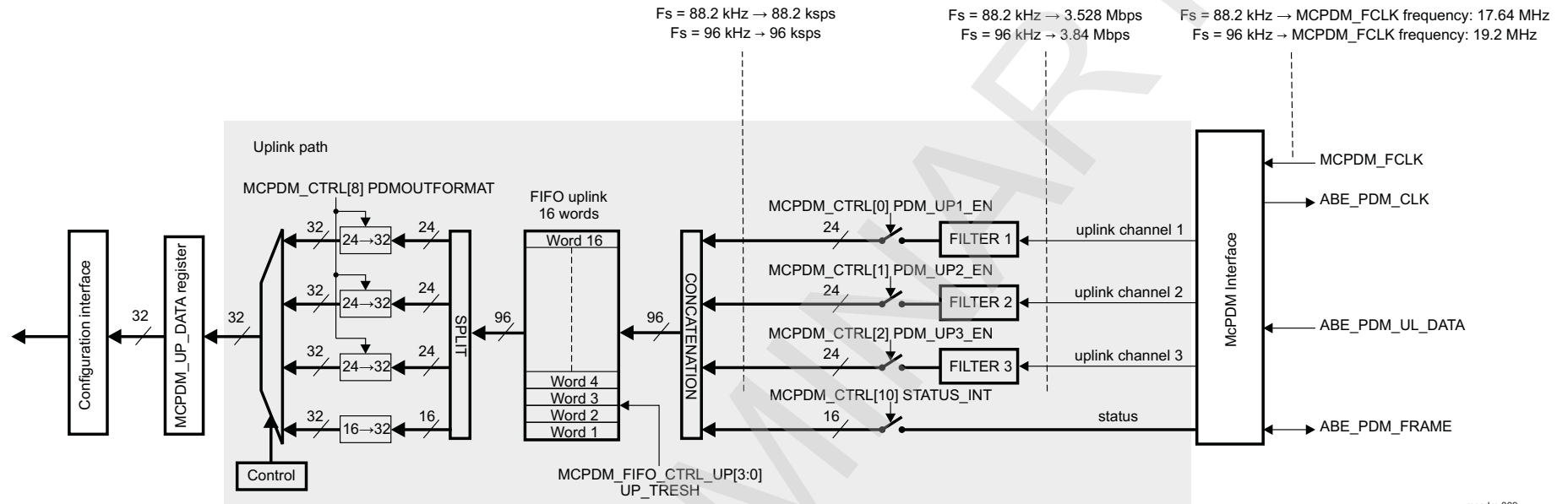
- Write a logical 1 to the [MCPDM\\_DMAENABLE\\_SET](#)[0] DMA\_DN\_ENABLE bit to enable the DMA request generation on the MCPDM\_DMA\_DOWN signal.
- Write a logical 1 to the [MCPDM\\_DMAENABLE\\_CLR](#)[0] DMA\_DN\_ENABLE bit to disable the DMA request generation on the MCPDM\_DMA\_DOWN signal.

For more information about DMA request signals mapping, see [Table 23-458](#).

### 23.6.4.7 Uplink Path

[Figure 23-168](#) is a detailed block diagram of the uplink path in the McPDM module.

Figure 23-168. Uplink Path Detailed Block Diagram



mcpdm-009

### 23.6.4.7.1 Uplink Path Features

The uplink path has the following features:

- Three audio uplink channels
- One status channel
- Decimation filter for each audio uplink channel
- 16-word FIFO uplink with threshold setting
- Audio channel data formatting
- High-frequency sampling feature

### 23.6.4.7.2 Uplink Path Description

#### 23.6.4.7.2.1 Uplink Data Received From the External Audio Companion Chip

The McPDM interface receives audio uplink channel data and status information data from the external audio companion chip at a clock rate of 17.64 MHz for a sampling frequency of 88.2 kHz, or 19.2 MHz for a sampling frequency of 96 kHz. This is the low-frequency sampling feature which is enabled by default. The uplink path support high-frequency sampling feature to work with an ultrasonic external microphone. This feature is enabled by setting the `MCPDM_CTRL[13] DIV_SEL` bit to 1. Thus, the McPDM interface receives audio uplink channel data and status information data from the external audio companion chip at a clock rate of 17.64 MHz for a sampling frequency of 176,4 kHz, or 19.2 MHz for a sampling frequency of 192 kHz.

The McPDM interface extracts each data bit related to audio and sends it to the decimation filter of the corresponding channel. So, for each audio uplink channel, the McPDM interface generates one bit data flow to the corresponding decimation filter at a rate of 3.528 Mbps for a sampling frequency of 88.2 kHz, or 3.84 Mbps for a sampling frequency of 96 kHz.

Moreover, when status information data are detected on the frame synchronization signal (`ABE_PDM_FRAME`), the status information are recovered and sent to the status channel. When no status information data is detected on the frame synchronization signal, a null word (0x0000) is sent on the status channel.

See [Section 23.6.2.4.1, Protocols and Data Formats in Normal Mode](#), for more information about protocols and data formats in normal mode, and [Section 23.6.2.4.3, Protocols and Data Formats in Status Mode](#), for more information about protocols and data formats in status mode.

#### 23.6.4.7.2.2 Decimation Filter

Each bit data flow related to an audio uplink channel and provided by the McPDM interface is processed by a decimation filter. The aim of this filter is to band-limit the noise and to downsample the incoming bit data flow to generate a 24-bit wide signed data flow at a rate of 88.2 kbps for a sampling frequency of 88.2 kHz, or 96 kbps for a sampling frequency of 96 kHz.

#### 23.6.4.7.2.3 FIFO Uplink Management

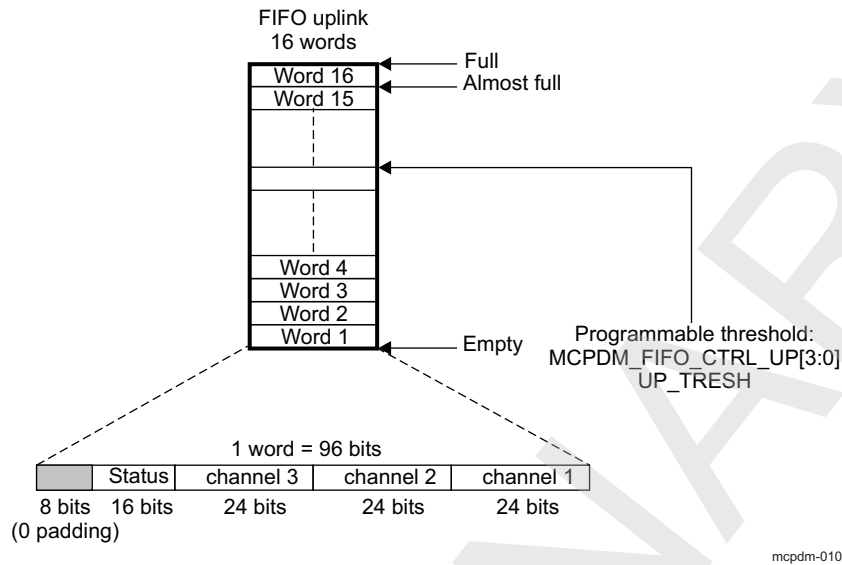
##### 23.6.4.7.2.3.1 General Description

The McPDM module implements a FIFO for the uplink path for communication with the host processor in the device. The FIFO uplink can store up to 16 words. Each word is 96 bits wide and is a combination of 24-bit wide signed audio uplink channel data coming from decimation filters and 16-bit wide status information data coming from the McPDM interface.

[Figure 23-169](#) shows the FIFO uplink overview.



Figure 23-169. FIFO Uplink Overview



23.6.4.7.2.3.2 FIFO Operations

The FIFO uplink is filled with audio uplink channel data coming from decimation filters and status information data from the McPDM interface. Once the FIFO uplink threshold (`MCPDM_FIFO_CTRL_UP[3:0] UP_TRESH` bit field) is reached and depending on conditions, an event occurs and generates an interrupt or DMA request.

**NOTE:** See Section 23.6.4.5, *McPDM Interrupt Requests*, for more information about events generating an interrupt request, and Section 23.6.4.6, *McPDM DMA Requests*, for more information about conditions for DMA request generation.

When an interrupt request upon a FIFO-uplink-read-request event or a `MCPDM_DMA_UP` request occurs, the host processor or the DMA controller must read the words from the FIFO uplink (`MCPDM_UP_DATA` register). The number of read accesses from the host/DMA is determined by the following equation:

$$\text{Num\_read\_access} = (\text{PDM\_UP1\_EN} + \text{PDM\_UP2\_EN} + \text{PDM\_UP3\_EN} + \text{STAT\_INT}) * (\text{UP\_THRESH})$$
 where the `PDM_UP_THRESHOLD` is programmed through the `MCPDM_FIFO_CTRL_UP[3:0] UP_TRESH` bit field. `PDM_UP1_EN`, `PDM_UP2_EN`, `PDM_UP3_EN`, and `STAT_INT` are boolean values representing whether the corresponding channel is enable or not.

For example, if audio uplink channels 1 and 3 are enabled, audio uplink channel 2 is disabled, and status channel is enabled, and the uplink threshold is programmed for one line only (`MCPDM_FIFO_CTRL_UP[3:0] UP_TRESH = 0x0`):

- `Num_read_access = 3` because two audio uplink and the status channels are enabled.
- The first reading of the `McPDM_UP_DATA` register should correspond to the audio uplink channel 1 data.
- The second reading of the `McPDM_UP_DATA` register should correspond to the audio uplink channel 3 data.
- The third reading of the `McPDM_UP_DATA` register should correspond to the status information data.

---

**NOTE:** No status information has been sent by the external audio companion chip if the status channel is enabled and the 16 lower bits of the corresponding data read from the `MCPDM_UP_DATA` register is 0x0000 .

The audio uplink and status channels cannot be enabled/disabled on the fly during uplink signaling. The uplink path must first be forced to reset by setting the `MCPDM_CTRL[11]` `SW_UP_RST` bit to 1 before enabling or disabling any channels.

---

Because the audio uplink channel data are 24 bits wide and the `MCPDM_UP_DATA` register is 32 bits wide, the `MCPDM_CTRL[8]` `PDMOUTFORMAT` bit allows selecting the justification of the 24-bit audio uplink channel data in the `MCPDM_UP_DATA` register:

- When the `MCPDM_CTRL[8]` `PDMOUTFORMAT` bit is cleared to 0, the 24-bit wide signed audio uplink channel data are left-justified with eight 0 padding bits for the lower bits.
  - When the `MCPDM_CTRL[8]` `PDMOUTFORMAT` is set to 1, the 24-bit wide signed audio uplink channel data are right-justified and sign-extended.
- 

**NOTE:** The `MCPDM_CTRL[8]` `PDMOUTFORMAT` bit does not affect the status information format. When the status information data are read from the `MCPDM_UP_DATA` register, only the 16 lower bits are significant.

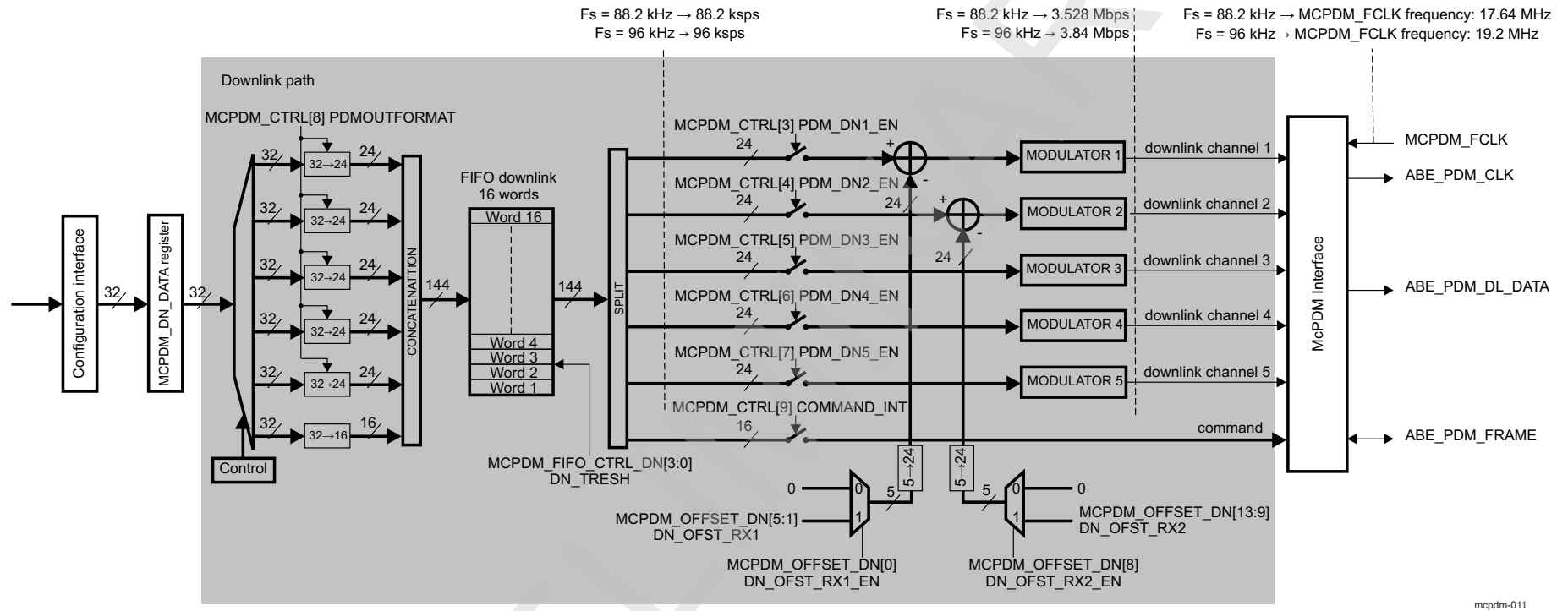
The setting of the `MCPDM_CTRL[8]` `PDMOUTPUTFORMAT` bit also affects the audio downlink channel data format.

---

23.6.4.8 Downlink Path

Figure 23-170 is a detailed block diagram of the downlink path in the McPDM module.

Figure 23-170. Downlink Path Detailed Block Diagram



mcpdm-011

### 23.6.4.8.1 Downlink Path Features

The downlink path has the following features:

- Five audio downlink channels
- Offset cancellation for audio downlink channels 1 and 2
- One command channel
- Delta-sigma modulator for each audio downlink channel
- 16-word FIFO downlink with threshold setting
- Audio channel data formatting

### 23.6.4.8.2 Downlink Path Description

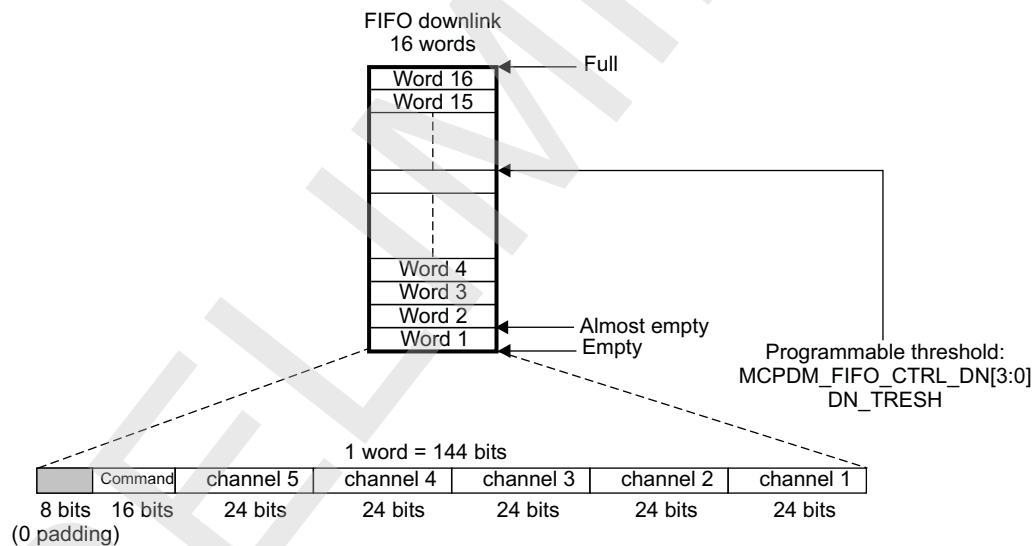
#### 23.6.4.8.2.1 FIFO Downlink Management

##### 23.6.4.8.2.1.1 General Description

The McPDM module implements a FIFO for the downlink path for communication with the host processor in the device. The FIFO downlink can store up to 16 words. Each word is 144 bits wide and is a combination of 24-bit wide signed audio downlink channels data and 16-bit wide command information data.

Figure 23-171 shows the FIFO downlink overview.

**Figure 23-171. FIFO Downlink Overview**



mcpdm-012

##### 23.6.4.8.2.1.2 FIFO Operations

The FIFO downlink is read by the McPDM module and data related to the audio downlink channels are sent to the delta-sigma modulators, while the data related to the command information is sent to the McPDM interface. Once the FIFO pointer goes below the programmed threshold (`MCPDM_FIFO_CTRL_DN[3:0] DN_TRESH`), a DMA or interrupt request is sent to the DMA or interrupt handler and, depending on conditions, an event occurs and generates an interrupt or DMA request.

**NOTE:** See [Section 23.6.4.5, McPDM Interrupt Requests](#), for more information about events generating an interrupt request, and [Section 23.6.4.6, McPDM DMA Requests](#), for more information about conditions for DMA request generation.

When an interrupt request upon a FIFO downlink write-request event or a MCPDM\_DMA\_DOWN request occurs, it indicates that the next words are available to be written to the [MCPDM\\_DN\\_DATA](#) register. The minimum number of write accesses from the host/DMA is determined by the following equation:

$$\text{Num\_min\_write\_access} = \text{PDM\_DN1\_EN} + \text{PDM\_DN2\_EN} + \text{PDM\_DN3\_EN} + \text{PDM\_DN4\_EN} + \text{PDM\_DN5\_EN} + \text{CMD\_INT}$$

PDM\_DN1\_EN, PDM\_DN2\_EN, PDM\_DN3\_EN, PDM\_DN4\_EN, PDM\_DN5\_EN and CMD\_INT are boolean values representing whether the corresponding channel is enable or not.

The maximum number of write accesses (before overwriting the data in the FIFO still to be transmitted) from the host/DMA is determined by the following equation:

$$\text{Num\_max\_write\_access} = (\text{FIFO\_SIZE} - \text{DN\_THRESH} + 1) * (\text{PDM\_DN1\_EN} + \text{PDM\_DN2\_EN} + \text{PDM\_DN3\_EN} + \text{PDM\_DN4\_EN} + \text{PDM\_DN5\_EN} + \text{CMD\_INT})$$

FIFO\_SIZE is the size of the downlink FIFO and DN\_THRESH is the downlink FIFO threshold, programmed through the [MCPDM\\_FIFO\\_CTRL\\_DN\[3:0\]](#) DN\_THRESHOLD bit field.

For example, if audio downlink channels 1, 2, and 3 are enabled, audio downlink channels 4 and 5 are disabled, and the command channel is enabled:

- Num\_min\_write\_access = 4, because three audio downlinks and the command channels are enabled.
- The first writing to the McPDM\_DN\_DATA register should correspond to the audio downlink channel 1 data.
- The second writing to the McPDM\_DN\_DATA register should correspond to the audio downlink channel 2 data.
- The third writing to the McPDM\_DN\_DATA register should correspond to the audio downlink channel 3 data.
- The fourth writing to the McPDM\_DN\_DATA register should correspond to the command information data.

---

**NOTE:** When the command channel is enabled and no command data needs to be sent to the external audio companion chip, 0x0000 must be written to the 16 lower bits of the [MCPDM\\_DN\\_DATA](#) register as command data.

The audio downlink and command channels cannot be enabled/disabled on the fly during downlink signaling. The downlink path must first be forced to reset by setting the [MCPDM\\_CTRL\[12\]](#) SW\_DN\_RST bit to 1 before enabling or disabling any channels.

---

Because the audio downlink channel data are 24 bits wide and the [MCPDM\\_DN\\_DATA](#) register is 32 bits wide, the [MCPDM\\_CTRL\[8\]](#) PDMOUTFORMAT bit indicates the justification of the 24-bit audio downlink channel data in the [MCPDM\\_DN\\_DATA](#) register:

- When the [MCPDM\\_CTRL\[8\]](#) PDMOUTFORMAT is cleared to 0, the 24-bit wide signed audio downlink channel data are left-justified with eight 0 padding bits for the lower bits.
- When [MCPDM\\_CTRL\[8\]](#) PDMOUT FORMAT is set to 1, the 24-bit wide signed audio downlink channel data are right-justified and sign-extended.

---

**NOTE:** The [MCPDM\\_CTRL\[8\]](#) PDMOUTFORMAT bit does not affect the command information format. When the command information data are written to the [MCPDM\\_DN\\_DATA](#) register, only the 16 lower bits are significant.

The setting of the [MCPDM\\_CTRL\[8\]](#) PDMOUTPUTFORMAT bit also affects the audio uplink channel data format.

---

### 23.6.4.8.2.1.3 Offset Cancellation

Because audio downlink channels 1 and 2 are related to the analog headset downlink path within the external audio companion chip, an offset cancellation feature is available for these channels to eliminate the offset of the analog headset downlink path.

When the offset cancellation feature is enabled, an offset value is subtracted from the audio data before being sent to the pulse-density modulator. The offset values are read from registers within the external audio companion chip and configured in the `MCPDM_DN_OFFSET[5:1]` `DN_OFST_RX1` bit field for the audio downlink channel 1, and in the `MCPDM_DN_OFFSET[13:9]` `DN_OFST_RX2` bit field for the audio downlink channel 2.

The offset cancellation feature is enabled or disabled by configuring the `MCPDM_DN_OFFSET[0]` `DN_OFST_RX1_EN` bit for the audio downlink channel 1, and the `MCPDM_DN_OFFSET[8]` `DN_OFST_RX2_EN` bit for the audio downlink channel 2 (00: Disabled, 10: Enabled).

#### 23.6.4.8.2.1.4 Pulse-Density Modulators

Each 24-bit wide data related to an audio downlink channel is processed by a pulse-density modulator. The aim of this modulator is to upsample the incoming audio data and then perform a 1-bit pulse-density modulation to generate a 1-bit data flow at a rate of 3.528 Mbps for a sampling frequency of 88.2 kHz, or 3.84 Mbps for a sampling frequency of 96 kHz. The 1-bit pulse-density modulation allows using a 1-bit analog-to-digital converter implemented in the external audio companion chip to convert the audio digital data into an audio analog signal.

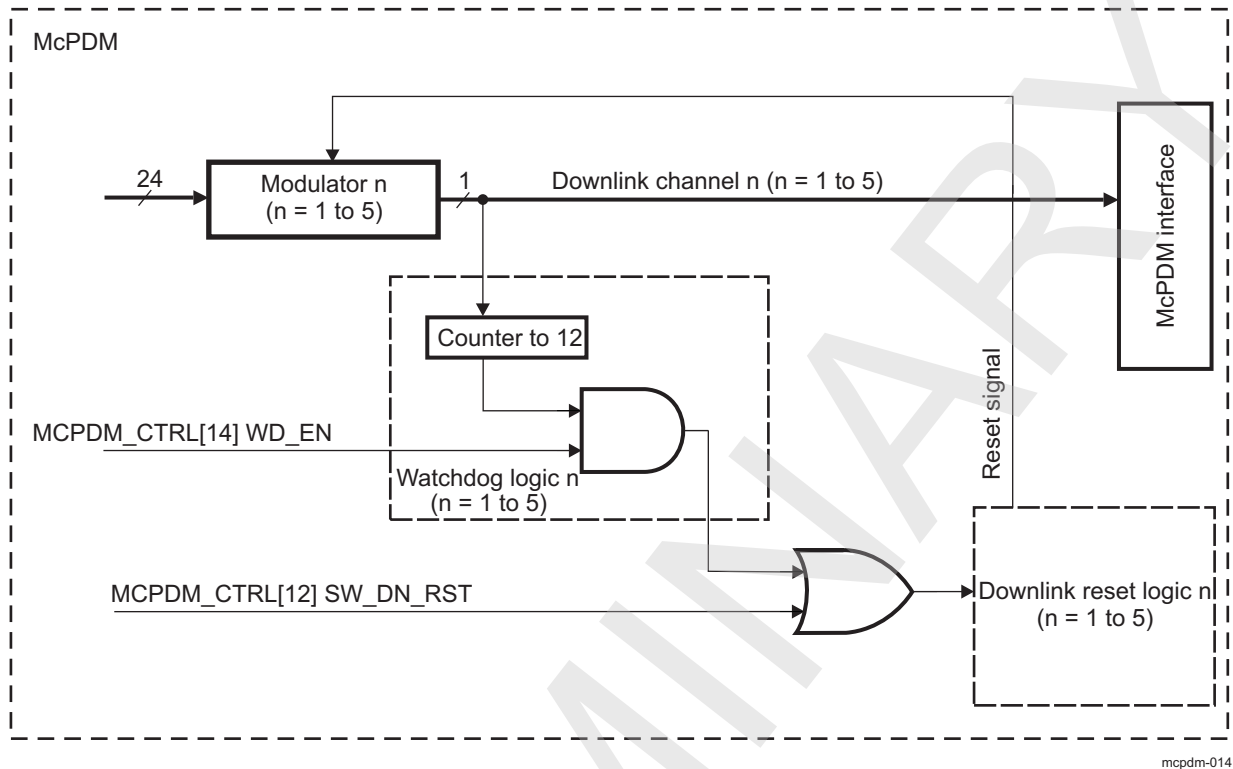
##### 23.6.4.8.2.1.4.1 Pulse-Density Modulators watchdog logic

If the 1-bit data flow coming out of the pulse-density modulator has twelve or more consecutive 1s or 0s, the pulse-density modulator had entered into unstable or saturated condition. Unstable or saturated condition leads to unpredictable behavior of the pulse-density modulator, thereby producing random noise. To prevent this unstable or saturated condition from occurring, a watchdog logic is used to monitor the 1-bit output of the pulse-density modulator for twelve consecutive 1s or twelve consecutive 0s. If entry to unstable condition is detected, the watchdog logic asserts the reset of the McPDM downlink paths for one clock pulse to put the McPDM hardware logic, related to the downlink mechanism, to initial state.

The watchdog logic is disabled by default. To enable it, the `MCPDM_CTRL[14]` `WD_EN` bit must be set to 0x1. If the watchdog logic is disabled, only the assertion of the downlink path reset `MCPDM_CTRL[12]` `SW_DN_RST` bit is allowed to reset the McPDM hardware logic, which is related to the downlink mechanism. If the watchdog logic is enabled, the downlink mechanism logic is reset when the downlink path reset or the watchdog reset is asserted.

Figure 23-172 shows the watchdog logic mechanism.

Figure 23-172. McPDM Watchdog Logic



mcpdm-014

#### 23.6.4.8.2.1.5 Downlink Data Sent to the External Audio Companion Chip

The McPDM interface transmits the audio downlink channel data and command data to the external audio companion chip at a clock rate of 17.64 MHz for a sampling frequency of 88.2 kHz, or 19.2 MHz for a sampling frequency of 96 kHz.

The command data are sent on the frame synchronization signal (ABE\_PDM\_FRAME) when the command channel is enabled. When the command channel is disabled or the command data is 0x0000, no command data is sent on the frame synchronization signal.

See [Section 23.6.2.4.1, Protocols and Data Formats in Normal Mode](#), for more information about protocols and data formats in normal mode, and [Section 23.6.2.4.2, Protocols and Data Formats in Command Mode](#), for more information about protocols and data formats in command mode.

#### 23.6.4.9 Error Reporting

During normal operation, when interrupt and DMA requests are properly served, no FIFO-uplink-full and no FIFO-downlink-empty event should occur.

Nevertheless, if one of these events occurs, the corresponding path must be forced to reset by software before being reconfigured. For more information about events servicing, see [Section 23.6.5.1.2, McPDM Events Servicing](#).

Moreover, a FIFO-uplink-empty or FIFO-downlink-full event occurs only in the case where all the channels of the corresponding path have been disabled during normal operation.

**NOTE:** Channels must not be enabled or disabled during normal operation.

Channels can be enabled or disabled when the corresponding path is forced to reset. For more information about McPDM software reset, see [Section 23.6.4.3, McPDM Software Reset](#).

## 23.6.5 McPDM Programming Guide

### 23.6.5.1 McPDM Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the module.

#### 23.6.5.1.1 Global Initialization

##### 23.6.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules in the device when the McPDM module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the environment and the integration of the McPDM. See [Section 23.6.2, McPDM Environment](#), and [Section 23.6.3, McPDM Integration](#), for more information.

[Table 23-462](#) describes the global initialization for surrounding modules.

**Table 23-462. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM module	The McPDM interface clock must be enabled and the functional clock must not be gated.
Control module	The McPDM specific pad muxing must be configured in the device control module.
(Optional) Cortex A9 MPU INTC (or DSP INTC)	The Cortex A9 MPU (or DSP) INTC must be configured to enable the interrupt request generation to the Cortex A9 MPU (or DSP) subsystem when interrupt requests are generated by the McPDM module.
(Optional) sDMA (or eDMA or ADMA)	The sDMA (or eDMA or ADMA) controller must be configured to account for the DMA requests generated by the McPDM module in the case of autonomous data transfers from/to memory.
(Optional) Interconnect	For more information about the interconnect configuration, see <a href="#">Chapter 13, Interconnect</a> .
External audio chip	Initialize the external audio chip according to the specific needs.
I <sup>2</sup> C controller	For more information about the I <sup>2</sup> C initialization, see <a href="#">Section 23.1, Multimaster High-Speed I<sup>2</sup>C Controller</a> .

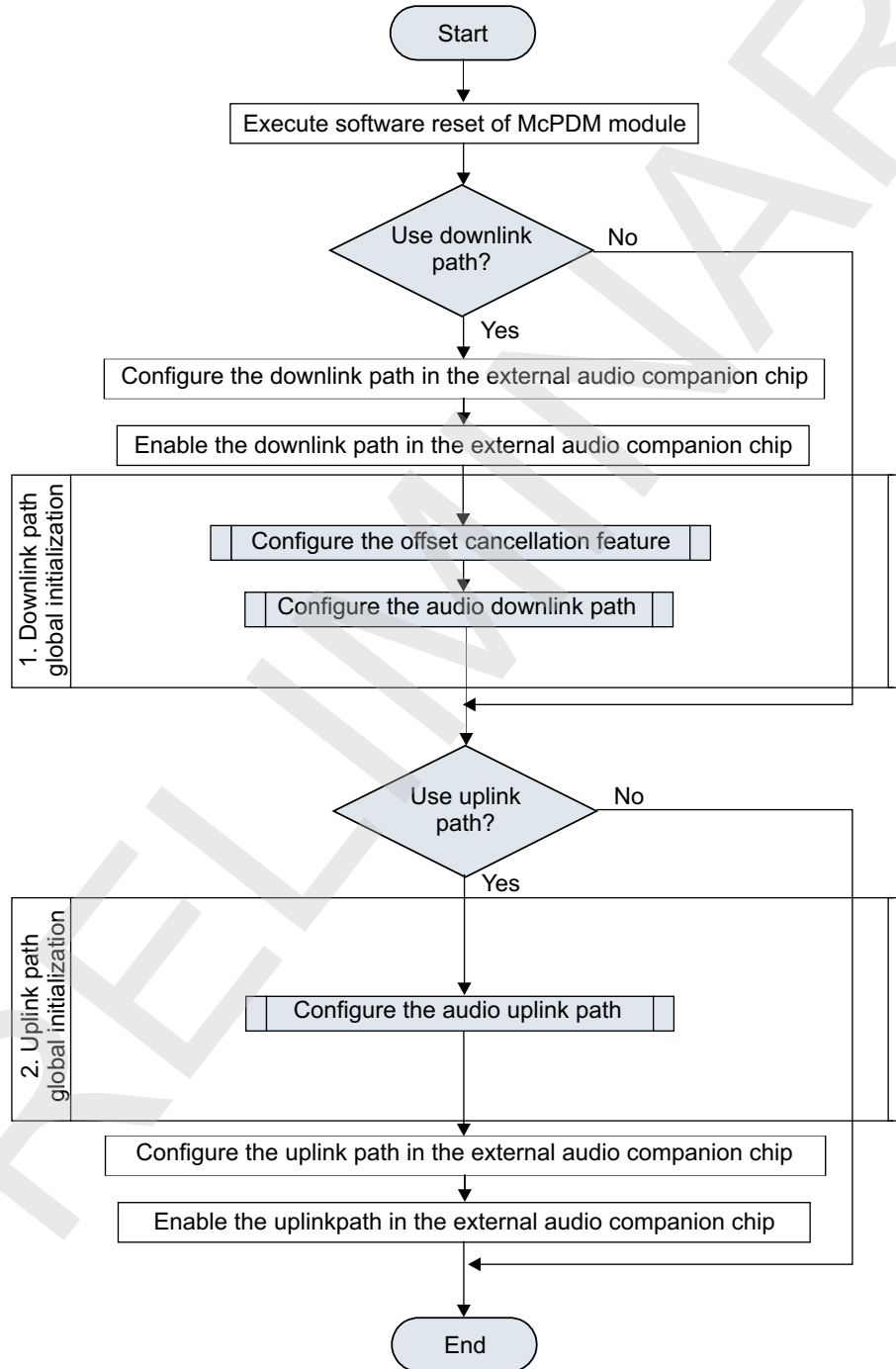


23.6.5.1.1.2 McPDM Global Initialization

23.6.5.1.1.2.1 Main Sequence – McPDM Global Initialization

This procedure initializes the McPDM after a power-on reset (POR) or software reset. Figure 23-173 shows a flow chart for the global initialization of the McPDM module.

Figure 23-173. McPDM Global Initialization Flow Chart



mcpdm-013

The McPDM software reset is executed by writing 0x1 in the `MCPDM_SYSCONFIG[0] SOFTRESET` bit. To detect the reset done event, user software must check this bit until it is cleared to 0x0 by the hardware.

Configuration of the offset cancellation feature is optional. The use of that feature depends on the characteristics of the external audio chip attached to the McPDM interface. If the external chip adds some DC offset on its output, then the offset cancellation feature of the McPDM must be used.

Table 23-463 describes the steps of the downlink path global initialization.

**Table 23-463. McPDM Downlink Path Global Initialization**

Procedure	Step	Access Type	Register/Bit Field/Programming Model	Value
Configure the offset cancellation feature. Only downlink path 1 and downlink path 2 can use the offset cancellation feature.	1. Read the offset value for the audio downlink path 1 from the external audio companion chip via the I <sup>2</sup> C interface.	R	Offset register 1 in the external audio companion chip	OFFSET1
	2. Configure the offset of the audio downlink path 1.	W	MCPDM_DN_OFFSET[5:1] DN_OFST_RX1	OFFSET1
	3. Enable the offset cancellation feature for the audio downlink path 1.	W	MCPDM_DN_OFFSET[0] DN_OFST_RX1_EN	0x1
	4. Read the offset value for the audio downlink path 2 from the external audio companion chip via the I <sup>2</sup> C interface.	R	Offset register 2 in the external audio companion chip	OFFSET2
	5. Configure the offset of the audio downlink path 2.	W	MCPDM_DN_OFFSET[13:9] DN_OFST_RX2	OFFSET2
	6. Enable the offset cancellation feature for the audio downlink path 2.	W	MCPDM_DN_OFFSET[8] DN_OFST_RX2_EN	0x1
Configure the audio downlink path.	1. Enable the interrupt request generation for the downlink path.	W	a. MCPDM_IRQENABLE_SET[3] DN_IRQ_FULL_MASK b. MCPDM_IRQENABLE_SET[2] DN_IRQ_ALMST_EMPTY_MASK c. MCPDM_IRQENABLE_SET[1] DN_IRQ_EMPTY_MASK d. MCPDM_IRQENABLE_SET[0] DN_IRQ_MASK	When downlink DMA request not used: 0xF When downlink DMA request used: 0xE
	2. Configure the FIFO downlink threshold.	W	MCPDM_FIFO_CTRL_DN[3:0] DN_TRESH	FIFO downlink threshold
	3. Enable the DMA request generation for the downlink path (optional: only if DMA request generation for the downlink path is required).	W	MCPDM_DMAENABLE_SET[0] DMA_DN_ENABLE	0x1
	4. Enable the downlink channels.	W	MCPDM_CTRL	See Table 23-459

Table 23-464 describes the steps of the uplink path global initialization.

**Table 23-464. McPDM Uplink Path Global Initialization**

Step	Access Type	Register/Bit Field/Programming Model	Value
1. Enable the interrupt request generation for the uplink path.	W	a. MCPDM_IRQENABLE_SET[11] UP_IRQ_FULL_MASK b. MCPDM_IRQENABLE_SET[10] UP_IRQ_ALMST_FULL_MASK c. MCPDM_IRQENABLE_SET[9] UP_IRQ_EMPTY_MASK d. MCPDM_IRQENABLE_SET[8] UP_IRQ_MASK	When uplink DMA request not used: 0xF When uplink DMA request used: 0xE
2. Configure the FIFO uplink threshold.	W	MCPDM_FIFO_CTRL_UP[3:0] UP_TRESH	FIFO uplink threshold
3. Enable the DMA request generation for the uplink path (optional: only if DMA request generation for the uplink path is required).	W	MCPDM_DMAENABLE_SET[1] DMA_UP_ENABLE	0x1
4. Configure the FIFO uplink sampling frequency.	W	MCPDM_CTRL[13] DIV_SEL	0x-

**Table 23-464. McPDM Uplink Path Global Initialization (continued)**

Step	Access Type	Register/Bit Field/Programming Model	Value
5. Enable the uplink channels.	W	MCPDM_CTRL	See Table 23-459

**23.6.5.1.2 McPDM Events Servicing**

**23.6.5.1.2.1 Downlink Path Event Servicing**

Table 23-465 to Table 23-468 describe the downlink path event servicing.

When the four downlink interrupts are enabled and then the downlink channel(s) is enabled, the FIFO-downlink-write request (MCPDM\_IRQSTATUS = 0x1), instead of the FIFO-downlink-empty request (MCPDM\_IRQSTATUS = 0x2), is generated. The first audio data to be sent is written from the interrupt handler. The McPDM module can work in two ways:

- Enable the desired downlink channels, and then write in the downlink FIFO the audio data that is going to be sent.
- Write the audio data in the downlink FIFO, and then enable the desired downlink channels to send this audio data.

In either way, the FIFO-downlink-write request is generated.

**Table 23-465. FIFO-Downlink-Full Event Servicing**

Step	Access Type	Register/Bit Field/Programming Model	Value
<b>TEST TA: Is FIFO downlink full?</b>	R	MCPDM_IRQSTATUS[3] DN_IRQ_FULL	
<b>IF: TRUE</b>			0x1
Acknowledge event.	W	MCPDM_IRQSTATUS[3] DN_IRQ_FULL	0x1
Downlink path reset	W	MCPDM_CTRL[12] SW_DN_RST	0x1
Downlink path global initialization		See Table 23-463.	
Downlink path reset release	W	MCPDM_CTRL[12] SW_DN_RST	0x0
END			
<b>ELSE</b>			0x0
END			
<b>END TEST TA</b>			

**Table 23-466. FIFO-Downlink-Empty Event Servicing**

Step	Access Type	Register/Bit Field/Programming Model	Value
<b>TEST TA: Is FIFO downlink empty?</b>	R	MCPDM_IRQSTATUS[1] DN_IRQ_EMPTY	
<b>IF: TRUE</b>			0x1
Acknowledge event.	W	MCPDM_IRQSTATUS[1] DN_IRQ_EMPTY	0x1
Downlink path reset	W	MCPDM_CTRL[12] SW_DN_RST	0x1
Downlink path reconfiguration		See Table 23-463.	
Downlink path reset release	W	MCPDM_CTRL[12] SW_DN_RST	0x0
END			
<b>ELSE</b>			0x0
END			
<b>END TEST TA</b>			

**Table 23-467. FIFO-Downlink-Almost-Empty Event Servicing**

Step	Access Type	Register/Bit Field/Programming Model	Value
<b>TEST TA:</b> Is FIFO downlink almost empty?	R	<a href="#">MCPDM_IRQSTATUS</a> [2] DN_IRQ_ALST_EMPTY	
<b>IF: TRUE</b>			0x1
Acknowledge event.	W	<a href="#">MCPDM_IRQSTATUS</a> [2] DN_IRQ_ALST_EMPTY	0x1
Downlink path reset	W	<a href="#">MCPDM_CTRL</a> [12] SW_DN_RST	0x1
Downlink path reconfiguration		See <a href="#">Table 23-463</a> .	
Downlink path reset release	W	<a href="#">MCPDM_CTRL</a> [12] SW_DN_RST	0x0
END			
<b>ELSE</b>			0x0
END			
<b>END TEST TA</b>			

**Table 23-468. FIFO Downlink Write-Request Event Servicing**

Step	Access Type	Register/Bit Field/Programming Model	Value
<b>TEST TA:</b> Is FIFO downlink write requested?	R	<a href="#">MCPDM_IRQSTATUS</a> [0] DN_IRQ	
<b>IF: TRUE</b>			0x1
Write FIFO downlink register (see <a href="#">Section 23.6.4.8.2.1.2</a> for more information).	W	<a href="#">MCPDM_DN_DATA</a>	
Acknowledge event	W	<a href="#">MCPDM_IRQSTATUS</a> [0] DN_IRQ	0x1
END			
<b>ELSE</b>			0x0
END			
<b>END TEST TA</b>			

**23.6.5.1.2.2 Uplink Path Events Servicing**

[Table 23-469](#) to [Table 23-471](#) describe the downlink path event servicing.

**Table 23-469. FIFO-Uplink-Full Event Servicing**

Step	Access Type	Register/Bit Field/Programming Model	Value
<b>TEST TA:</b> Is FIFO-uplink-full?	R	<a href="#">MCPDM_IRQSTATUS</a> [11] UP_IRQ_FULL	
<b>IF: TRUE</b>			0x1
Acknowledge event.	W	<a href="#">MCPDM_IRQSTATUS</a> [11] UP_IRQ_FULL	0x1
Uplink path reset	W	<a href="#">MCPDM_CTRL</a> [11] SW_UP_RST	0x1
Uplink path global initialization		See <a href="#">Table 23-464</a>	
Uplink path reset release	W	<a href="#">MCPDM_CTRL</a> [11] SW_UP_RST	0x0
END			
<b>ELSE</b>			0x0
END			
<b>END TEST TA</b>			

**Table 23-470. FIFO-Uplink-Almost-Full Event Servicing**

Step	Access Type	Register/Bit Field/Programming Model	Value
<b>TEST TA:</b> Is FIFO-uplink-almost-full?	R	<a href="#">MCPDM_IRQSTATUS</a> [10] UP_IRQ_ALST_FULL	
<b>IF: TRUE</b>			0x1
Acknowledge event.	W	<a href="#">MCPDM_IRQSTATUS</a> [10] UP_IRQ_ALST_FULL	0x1

**Table 23-470. FIFO-Uplink-Almost-Full Event Servicing (continued)**

Step	Access Type	Register/Bit Field/Programming Model	Value
Downlink path reset.	W	<a href="#">MCPDM_CTRL[12] SW_DN_RST</a>	0x1
Downlink path reconfiguration.		See <a href="#">Table 23-463</a> .	
Downlink path reset release.	W		0x0
END			
<b>ELSE</b>			0x0
END			
<b>END TEST TA</b>			

**Table 23-471. FIFO-Uplink-Empty Event Servicing**

Step	Access Type	Register/Bit Field/Programming Model	Value
<b>TEST TA:</b> Is FIFO uplink empty?	R	<a href="#">MCPDM_IRQSTATUS[9] UP_IRQ_EMPTY</a>	
<b>IF: TRUE</b>			0x1
Acknowledge event.	W	<a href="#">MCPDM_IRQSTATUS[9] UP_IRQ_EMPTY</a>	0x1
Uplink path reset	W	<a href="#">MCPDM_CTRL[11] SW_UP_RST</a>	0x1
Uplink path reconfiguration		See <a href="#">Table 23-464</a>	
Uplink path reset release	W	<a href="#">MCPDM_CTRL[11] SW_UP_RST</a>	0x0
END			
<b>ELSE</b>			0x0
END			
<b>END TEST TA</b>			

**Table 23-472. FIFO-Uplink-Read-Request Event Servicing**

Step	Access Type	Register/Bit Field/Programming Model	Value
<b>TEST TA:</b> Is FIFO uplink read requested?	R	<a href="#">MCPDM_IRQSTATUS[8] UP_IRQ</a>	
<b>IF: TRUE</b>			0x1
Read FIFO uplink register (see <a href="#">Section 23.6.4.7.2.3.2</a> for more information).	R	<a href="#">MCPDM_UP_DATA</a>	
Acknowledge event.	W	<a href="#">MCPDM_IRQSTATUS[8] UP_IRQ</a>	0x1
END			
<b>ELSE</b>			0x0
END			
<b>END TEST TA</b>			

## 23.6.6 McPDM Register Manual

### 23.6.6.1 McPDM Instance Summary

Table 23-473 summarizes the McPDM instance.

**Table 23-473. McPDM Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-A9 Private Access	Base Address DSP Private Access	Size
McPDM	0x4903 2000	0x4013 2000	0x3 2000	4KB

**NOTE:** Private access is an access that does not use the L3/L4 interconnects.

### 23.6.6.2 McPDM Registers

#### 23.6.6.2.1 McPDM Register Summary

Table 23-474 summarizes the McPDM register mapping.

**Table 23-474. McPDM Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
MCPDM_REVISION	R	32	0x0000 0000	0x4903 2000	0x4013 2000	0x3 2000
MCPDM_SYSCONFIG	RW	32	0x0000 0010	0x4903 2010	0x4013 2010	0x3 2010
RESERVED	RW	32	0x0000 0020	0x4903 2020	0x4013 2020	0x3 2020
MCPDM_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4903 2024	0x4013 2024	0x3 2024
MCPDM_IRQSTATUS	RW	32	0x0000 0028	0x4903 2028	0x4013 2028	0x3 2028
MCPDM_IRQENABLE_SET	RW	32	0x0000 002C	0x4903 202C	0x4013 202C	0x3 202C
MCPDM_IRQENABLE_CLR	RW	32	0x0000 0030	0x4903 2030	0x4013 2030	0x3 2030
MCPDM_IRQWAKEEN	RW	32	0x0000 0034	0x4903 2034	0x4013 2034	0x3 2034
MCPDM_DMAENABLE_SET	RW	32	0x0000 0038	0x4903 2038	0x4013 2038	0x3 2038
MCPDM_DMAENABLE_CLR	RW	32	0x0000 003C	0x4903 203C	0x4013 203C	0x3 203C
MCPDM_DMAWAKEEN	RW	32	0x0000 0040	0x4903 2040	0x4013 2040	0x3 2040
MCPDM_CTRL	RW	32	0x0000 0044	0x4903 2044	0x4013 2044	0x3 2044
MCPDM_DN_DATA	RW	32	0x0000 0048	0x4903 2048	0x4013 2048	0x3 2048
MCPDM_UP_DATA	R	32	0x0000 004C	0x4903 204C	0x4013 204C	0x3 204C
MCPDM_FIFO_CTRL_DN	RW	32	0x0000 0050	0x4903 2050	0x4013 2050	0x3 2050
MCPDM_FIFO_CTRL_UP	RW	32	0x0000 0054	0x4903 2054	0x4013 2054	0x3 2054
MCPDM_DN_OFFSET	RW	32	0x0000 0058	0x4903 2058	0x4013 2058	0x3 2058
RESERVED	RW	32	0x0000 005C	0x4903 205C	0x4013 205C	0x3 205C
RESERVED	RW	32	0x0000 0060	0x4903 2060	0x4013 2060	0x3 2060
RESERVED	RW	32	0x0000 0064	0x4903 2064	0x4013 2064	0x3 2064
RESERVED	R	32	0x0000 0068	0x4903 2068	0x4013 2068	0x3 2068
RESERVED	RW	32	0x0000 006C	0x4903 206C	0x4013 206C	0x3 206C
RESERVED	RW	32	0x0000 0070	0x4903 2070	0x4013 2070	0x3 2070
RESERVED	R	32	0x0000 0074	0x4903 2074	0x4013 2074	0x3 2074

#### 23.6.6.2.2 McPDM Register Description

Table 23-475 through Table 23-505 describe the individual McPDM registers.

**Table 23-475. MCPDM\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b>	McPDM
<b>Description</b>	IP revision identifier (X.Y.R) used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

**Table 23-476. Register Call Summary for Register MCPDM\_REVISION**

- Multichannel PDM Controller
- [McPDM Register Summary: \[0\]](#)

**Table 23-477. MCPDM\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b>	McPDM
<b>Description</b>	This register allows controlling various parameters of the OCP interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLEMODE	FREEMU	SOFTRESET													

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:2	IDLEMODE	Configuration of the idle mode. 0x0: FORCE idle. idle request is acknowledged unconditionally and immediately. No wake-up capability 0x1: NO idle. Idle request is never acknowledged. No wake-up capability 0x2: SMART idle. The acknowledgment to an idle request is given based on the internal activity. No wake-up capability 0x3: Idle request acknowledged pending internal conditions, asynchronous wake-up enabled. Wakeup capability.	RW	0x2
1	FREEMU	Sensitivity to emulation (debug) suspend input signal. 0x0: McPDM module is sensitive to emulation suspend 0x1: McPDM module is not sensitive to emulation suspend	RW	0
0	SOFTRESET	McPDM software Reset. Write 0x0: No Action Read 0x0: No ongoing software reset Read 0x1: Reset is ongoing Write 0x1: Start software reset	RW	0

**Table 23-478. Register Call Summary for Register MCPDM\_SYSCONFIG**

Multichannel PDM Controller

- [McPDM Software Reset: \[0\] \[1\]](#)
- [McPDM Power Management: \[2\] \[3\]](#)
- [McPDM Global Initialization: \[4\]](#)
- [McPDM Register Summary: \[5\]](#)

**Table 23-479. MCPDM\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	McPDM
<b>Physical Address</b>	See <a href="#">Table 23-474</a>		
<b>Description</b>	Interrupt request raw status register (for debug purpose).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
RESERVED																RESERVED				RESERVED				UP_IRQ_FULL		UP_IRQ_ALST_FULL		UP_IRQ_EMPTY		UP_IRQ		DN_IRQ_FULL		DN_IRQ_ALST_EMPTY		DN_IRQ_EMPTY		DN_IRQ	

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11	UP_IRQ_FULL	FIFO-uplink-full signal appears when a write access is performed and the FIFO uplink is already full. Read 0x0: No event pending Write 0x0: No action Write 0x1: Set the event (for debug) Read 0x10: Event pending	RW	0
10	UP_IRQ_ALST_FULL	FIFO uplink almost-full signal appears when the FIFO uplink contains (FIFO uplink size - 1) elements. Read 0x0: No event pending Write 0x0: No action Write 0x1: Set the event (for debug) Read 0x10: Event pending	RW	0
9	UP_IRQ_EMPTY	FIFO uplink empty signal appears when a read access is done and FIFO uplink already empty. Read 0x0: No event pending Write 0x0: No action Write 0x1: Set the event (for debug) Read 0x10: Event pending	RW	0
8	UP_IRQ	FIFO uplink interrupt appears when the number of data present in the FIFO uplink has reached the value of the FIFO uplink threshold. Read 0x0: No event pending Write 0x0: No action Write 0x1: Set the event (for debug) Read 0x10: Event pending	RW	0
7:4	RESERVED	reserved	R	0x0



Bits	Field Name	Description	Type	Reset
3	DN_IRQ_FULL	FIFO downlink full signal appears when the FIFO uplink is full and another a write access is performed. Read 0x0: No event pending Write 0x0: No action Write 0x1: Set the event (for debug) Read 0x10: Event pending	RW	0
2	DN_IRQ_ALST_EMPTY	FIFO downlink almost empty signal appears when the FIFO downlink contains only one element. Read 0x0: No event pending Write 0x0: No action Write 0x1: Set the event (for debug) Read 0x10: Event pending	RW	0
1	DN_IRQ_EMPTY	FIFO-downlink-empty signal appears when read access is performed and FIFO downlink is already empty. Read 0x0: No event pending Write 0x0: No action Write 0x1: Set the event (for debug) Read 0x10: Event pending	RW	0
0	DN_IRQ	FIFO downlink status is set when the number of data is below the FIFO downlink threshold value. Read 0x0: No event pending Write 0x0: No action Write 0x1: Set the event (for debug) Read 0x10: Event pending	RW	0

**Table 23-480. Register Call Summary for Register MCPDM\_IRQSTATUS\_RAW**

Multichannel PDM Controller

- [McPDM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)
- [McPDM Register Summary: \[14\]](#)

**Table 23-481. MCPDM\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	McPDM
<b>Physical Address</b>	See <a href="#">Table 23-474</a>		
<b>Description</b>	Interrupt request status register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED				RESERVED				UP_IRQ_FULL	UP_IRQ_ALST_FULL	UP_IRQ_EMPTY	UP_IRQ	DN_IRQ_FULL	DN_IRQ_ALST_EMPTY	DN_IRQ_EMPTY	DN_IRQ

## Multichannel PDM Controller

www.ti.com

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11	UP_IRQ_FULL	FIFO-uplink-full signal appears when a write access is performed and the FIFO uplink is already full. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event Read 0x10: Event pending	RW	0
10	UP_IRQ_ALST_FULL	FIFO uplink almost-full signal appears when the FIFO uplink contains (FIFO uplink size - 1) elements. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event Read 0x10: Event pending	RW	0
9	UP_IRQ_EMPTY	FIFO uplink empty signal appears when a read access is done and FIFO uplink already empty. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event Read 0x10: Event pending	RW	0
8	UP_IRQ	FIFO uplink interrupt appears when the number of data present in the FIFO uplink has reached the value of the FIFO uplink threshold. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event Read 0x10: Event pending	RW	0
7:4	RESERVED		R	0x0
3	DN_IRQ_FULL	FIFO downlink full signal appears when the FIFO uplink is full and another a write access is performed. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event Read 0x10: Event pending	RW	0
2	DN_IRQ_ALST_EMPTY	FIFO downlink almost empty signal appears when the FIFO downlink contains only one element. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event Read 0x10: Event pending	RW	0
1	DN_IRQ_EMPTY	FIFO-downlink-empty signal appears when read access is performed and FIFO downlink is already empty. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event Read 0x10: Event pending	RW W1toSet	0
0	DN_IRQ	FIFO downlink status is set when the number of data is below the FIFO downlink threshold value. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event Read 0x10: Event pending	RW	0

**Table 23-482. Register Call Summary for Register MCPDM\_IRQSTATUS**

Multichannel PDM Controller

- [McPDM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [Downlink Path Event Servicing: \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\]](#)
- [Uplink Path Events Servicing: \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\]](#)
- [McPDM Register Summary: \[30\]](#)

**Table 23-483. MCPDM\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	McPDM
<b>Physical Address</b>	See <a href="#">Table 23-474</a>		
<b>Description</b>	Interrupt request enable set register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																UP_IRQ_FULL_MASK				RESERVED				DN_IRQ_FULL_MASK							
																UP_IRQ_ALST_FULL_MASK								DN_IRQ_ALST_EMPTY_MASK							
																UP_IRQ_EMPTY_MASK								DN_IRQ_EMPTY_MASK							
																UP_IRQ_MASK															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x000000
11	UP_IRQ_FULL_MASK	FIFO-uplink-full event interrupt enabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Enable interrupt Read 0x1: Interrupt enabled	RW	0
10	UP_IRQ_ALST_FULL_MASK	FIFO-uplink-almost-full event interrupt enabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Enable interrupt Read 0x1: Interrupt enabled	RW	0
9	UP_IRQ_EMPTY_MASK	FIFO-uplink-empty event interrupt enabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Enable interrupt Read 0x1: Interrupt enabled	RW	0
8	UP_IRQ_MASK	FIFO-uplink-read-request event interrupt enabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Enable interrupt Read 0x1: Interrupt enabled	RW	0
7:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3	DN_IRQ_FULL_MASK	FIFO-downlink-full event interrupt enabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Enable interrupt Read 0x1: Interrupt enabled	RW	0
2	DN_IRQ_ALST_EMPTY_MASK	FIFO downlink almost-empty event interrupt enabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Enable interrupt Read 0x1: Interrupt enabled	RW	0
1	DN_IRQ_EMPTY_MASK	FIFO-downlink-empty event interrupt enabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Enable interrupt Read 0x1: Interrupt enabled	RW	0
0	DN_IRQ_MASK	FIFO downlink write-request event interrupt enabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Enable interrupt Read 0x1: Interrupt enabled	RW	0

**Table 23-484. Register Call Summary for Register MCPDM\_IRQENABLE\_SET**

Multichannel PDM Controller

- [McPDM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [McPDM Global Initialization: \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\]](#)
- [McPDM Register Summary: \[17\]](#)

**Table 23-485. MCPDM\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 0030	
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b> McPDM
<b>Description</b>	Interrupt request enable clear register.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																				
RESERVED																UP_IRQ_FULL_MASK				UP_IRQ_ALST_FULL_MASK				UP_IRQ_EMPTY_MASK				UP_IRQ_MASK				RESERVED				DN_IRQ_FULL_MASK				DN_IRQ_ALST_EMPTY_MASK				DN_IRQ_EMPTY_MASK				DN_IRQ_MASK			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11	UP_IRQ_FULL_MASK	FIFO-uplink-full event interrupt disabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Disable interrupt Read 0x1: Interrupt enabled	RW	0
10	UP_IRQ_ALST_FULL_MASK	FIFO-uplink-almost-full event interrupt disabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Disable interrupt Read 0x1: Interrupt enabled	RW	0
9	UP_IRQ_EMPTY_MASK	FIFO-uplink-empty event interrupt disabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Disable interrupt Read 0x1: Interrupt enabled	RW	0
8	UP_IRQ_MASK	FIFO-uplink-read-request event interrupt disabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Disable interrupt Read 0x1: Interrupt enabled	RW	0
7:4	RESERVED		R	0x0
3	DN_IRQ_FULL_MASK	FIFO-downlink-full event interrupt disabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Disable interrupt Read 0x1: Interrupt enabled	RW	0
2	DN_IRQ_ALST_EMPTY_MASK	FIFO downlink almost-empty event interrupt disabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Disable interrupt Read 0x1: Interrupt enabled	RW	0
1	DN_IRQ_EMPTY_MASK	FIFO-downlink-empty event interrupt disabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Disable interrupt Read 0x1: Interrupt enabled	RW	0
0	DN_IRQ_MASK	FIFO downlink write-request event interrupt disabling bit. Read 0x0: Interrupt disabled Write 0x0: No action Write 0x10: Disable interrupt Read 0x1: Interrupt enabled	RW	0

**Table 23-486. Register Call Summary for Register MCPDM\_IRQENABLE\_CLR**

Multichannel PDM Controller

- [McPDM Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [McPDM Register Summary: \[9\]](#)

**Table 23-487. MCPDM\_IRQWAKEEN**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b>	McPDM
<b>Description</b>	Interrupt request wake-up enable register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IRQ_WAKEUP_UP_EN		IRQ_WAKEUP_DN_EN													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	IRQ_WAKEUP_UP_EN	Enabling/disabling bit for wake-up request generation upon a FIFO-uplink-read-request event. 0x00: Disable the wake-up request generation 0x10: Enable the wake-up request generation	RW	1
0	IRQ_WAKEUP_DN_EN	Enabling/disabling bit for wake-up by FIFO downlink write-request event. 0x00: Disable the wake-up request generation 0x10: Enable the wake-up request generation	RW	1

**Table 23-488. Register Call Summary for Register MCPDM\_IRQWAKEEN**

Multichannel PDM Controller

- [McPDM Power Management: \[0\] \[1\] \[2\]](#)
- [McPDM Register Summary: \[3\]](#)

**Table 23-489. MCPDM\_DMAENABLE\_SET**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b>	McPDM
<b>Description</b>	DMA request enable set register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DMA_UP_ENABLE		DMA_DN_ENABLE													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	DMA_UP_ENABLE	Uplink path DMA request generation enabling bit. Read 0x00: DMA request disabled Write 0x0: No action Write 0x10: Enable DMA request Read 0x10: DMA request enabled	RW	0
0	DMA_DN_ENABLE	Downlink path DMA request generation enabling bit. Read 0x00: DMA request disabled Write 0x0: No action Write 0x10: Enable DMA request Read 0x10: DMA request enabled	RW	0

**Table 23-490. Register Call Summary for Register MCPDM\_DMAENABLE\_SET**

- Multichannel PDM Controller
- [McPDM DMA Requests: \[0\] \[1\] \[2\] \[3\]](#)
  - [McPDM Global Initialization: \[4\] \[5\]](#)
  - [McPDM Register Summary: \[6\]](#)

**Table 23-491. MCPDM\_DMAENABLE\_CLR**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	McPDM
<b>Physical Address</b>	See <a href="#">Table 23-474</a>		
<b>Description</b>	DMA request enable clear register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DMA_UP_ENABLE		DMA_DN_ENABLE													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	DMA_UP_ENABLE	Uplink path DMA request generation disabling bit. Read 0x00: DMA request disabled Write 0x0: No action Write 0x10: Disable DMA request Read 0x10: DMA request enabled	RW	0
0	DMA_DN_ENABLE	Downlink path DMA request generation disabling bit. Read 0x00: DMA request disabled Write 0x0: No action Write 0x10: Disable DMA request Read 0x10: DMA request enabled	RW	0

**Table 23-492. Register Call Summary for Register MCPDM\_DMAENABLE\_CLR**

- Multichannel PDM Controller
- [McPDM DMA Requests: \[0\] \[1\] \[2\] \[3\]](#)
  - [McPDM Register Summary: \[4\]](#)

**Table 23-493. MCPDM\_DMAWAKEEN**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b>	McPDM
<b>Description</b>	DMA request wake-up enable register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DMA_WAKEUP_UP_EN		DMA_WAKEUP_DN_EN													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	DMA_WAKEUP_UP_EN	Enabling/disabling bit for wake-up request generation upon an uplink path DMA request.  0x00: Disable the wake-up request generation 0x10: Enable the wake-up request generation	RW	1
0	DMA_WAKEUP_DN_EN	Enabling/disabling bit for wake-up request generation upon a downlink path DMA request.  0x00: Disable the wake-up request generation 0x10: Enable the wake-up request generation	RW	1

**Table 23-494. Register Call Summary for Register MCPDM\_DMAWAKEEN**

Multichannel PDM Controller

- [McPDM Power Management: \[0\] \[1\] \[2\]](#)
- [McPDM Register Summary: \[3\]](#)

**Table 23-495. MCPDM\_CTRL**

<b>Address Offset</b>	0x0000 0044		
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b>	McPDM
<b>Description</b>	MCPDM control register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WD_EN	DIV_SEL	SW_DN_RST	SW_UP_RST	STATUS_INT	CMD_INT	PDMOUTFORMAT	PDM_DN5_EN	PDM_DN4_EN	PDM_DN3_EN	PDM_DN2_EN	PDM_DN1_EN	PDM_UP3_EN	PDM_UP2_EN	PDM_UP1_EN	



Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	WD_EN	This bit is used to enable or disable the pulse-density modulator watchdog logic 0x0: Disabled (Default after reset) 0x1: Enabled	RW	0
13	DIV_SEL	Multiply by 2 the FS of the uplink path 0x0: FS = 88.2 kHz, or 96 kHz 0x1: FS = 176.4 kHz 192 kHz	RW	0
12	SW_DN_RST	Software reset of the downlink path. 0x00: Downlink path is out of reset 0x1: reset of the downlink path	RW	0
11	SW_UP_RST	Software reset of the uplink path. 0x0: uplink path is out of reset. 0x1: reset of the uplink path	RW	0
10	STATUS_INT	Status channel enabling/disabling bit. 0x0: status channel is disabled 0x1: status channel is enabled	RW	0
9	CMD_INT	Command channel enabling/disabling bit. 0x0: command channel is disabled 0x1: command channel is enabled	RW	0
8	PDMOUTFORMAT	Audio format selection: 0x0: left justification with eight "0" bit padding added for the less significant bits 0x1: right justification with sign bit extended to the eight most significant bits	RW	0
7	PDM_DN5_EN	Audio downlink channel 5 enabling/disabling bit.	RW	0
6	PDM_DN4_EN	Audio downlink channel 4 enabling/disabling bit.	RW	0
5	PDM_DN3_EN	Audio downlink channel 3 enabling/disabling bit.	RW	0
4	PDM_DN2_EN	Audio downlink channel 2 enabling/disabling bit.	RW	0
3	PDM_DN1_EN	Audio downlink channel 1 enabling/disabling bit.	RW	0
2	PDM_UP3_EN	Audio uplink channel 3 enabling/disabling bit.	RW	0
1	PDM_UP2_EN	Audio uplink channel 2 enabling/disabling bit.	RW	0
0	PDM_UP1_EN	Audio uplink channel 1 enabling/disabling bit.	RW	0

**Table 23-496. Register Call Summary for Register MCPDM\_CTRL**
**Multichannel PDM Controller**

- [Protocols and Data Formats in Normal Mode: \[0\]](#)
- [Entering the Command Mode: \[1\]](#)
- [Command Data Transmission: \[2\] \[3\]](#)
- [Entering the Status Mode: \[4\]](#)
- [Status Information Data Reception: \[5\] \[6\]](#)
- [McPDM Software Reset: \[7\] \[8\] \[9\] \[10\]](#)
- [McPDM Power Management: \[11\] \[12\] \[13\] \[14\] \[15\] \[16\]](#)
- [Uplink Data Received From the External Audio Companion Chip: \[17\]](#)
- [FIFO Uplink Management: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [FIFO Downlink Management: \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\]](#)
- [McPDM Global Initialization: \[32\] \[33\] \[34\]](#)
- [Downlink Path Event Servicing: \[35\] \[36\] \[37\] \[38\] \[39\] \[40\]](#)
- [Uplink Path Events Servicing: \[41\] \[42\] \[43\] \[44\] \[45\]](#)
- [McPDM Register Summary: \[46\]](#)

**Table 23-497. MCPDM\_DN\_DATA**

<b>Address Offset</b>	0x0000 0048	
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b> McPDM
<b>Description</b>	Downlink path data register.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DN_DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DN_DATA	Downlink path data value.	RW	0x0000 0000

**Table 23-498. Register Call Summary for Register MCPDM\_DN\_DATA**

Multichannel PDM Controller

- [Command Data Transmission: \[0\] \[1\]](#)
- [FIFO Downlink Management: \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Downlink Path Event Servicing: \[7\]](#)
- [McPDM Register Summary: \[8\]](#)

**Table 23-499. MCPDM\_UP\_DATA**

<b>Address Offset</b>	0x0000 004C	
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b> McPDM
<b>Description</b>	Uplink path data register.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UP_DATA																															

Bits	Field Name	Description	Type	Reset
31:0	UP_DATA	Uplink path data value.	R	0x0000 0000

**Table 23-500. Register Call Summary for Register MCPDM\_UP\_DATA**

Multichannel PDM Controller

- [Status Information Data Reception: \[0\] \[1\]](#)
- [FIFO Uplink Management: \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Uplink Path Events Servicing: \[7\]](#)
- [McPDM Register Summary: \[8\]](#)

**Table 23-501. MCPDM\_FIFO\_CTRL\_DN**

<b>Address Offset</b>	0x0000 0050	
<b>Physical Address</b>	See <a href="#">Table 23-474</a>	<b>Instance</b> McPDM
<b>Description</b>	FIFO downlink control register.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DN_TRESH							

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000
3:0	DN_TRESH	FIFO downlink threshold value.	RW	0x2

**Table 23-502. Register Call Summary for Register MCPDM\_FIFO\_CTRL\_DN**

Multichannel PDM Controller

- [McPDM Power Management: \[0\]](#)
- [FIFO Downlink Management: \[1\] \[2\]](#)
- [McPDM Global Initialization: \[3\]](#)
- [McPDM Register Summary: \[4\]](#)

**Table 23-503. MCPDM\_FIFO\_CTRL\_UP**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	McPDM
<b>Physical Address</b>	See <a href="#">Table 23-474</a>		
<b>Description</b>	FIFO uplink control register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																UP_TRESH															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000
3:0	UP_TRESH	FIFO uplink threshold value.	RW	0x2

**Table 23-504. Register Call Summary for Register MCPDM\_FIFO\_CTRL\_UP**

Multichannel PDM Controller

- [McPDM Power Management: \[0\]](#)
- [FIFO Uplink Management: \[1\] \[2\] \[3\]](#)
- [McPDM Global Initialization: \[4\]](#)
- [McPDM Register Summary: \[5\]](#)

**Table 23-505. MCPDM\_DN\_OFFSET**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	McPDM
<b>Physical Address</b>	See <a href="#">Table 23-474</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DN_OFST_RX2		DN_OFST_RX2_EN	RESERVED	DN_OFST_RX1		DN_OFST_RX1_EN									

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x00000
13:9	DN_OFST_RX2	Offset value for the audio downlink channel 2.	RW	0x00
8	DN_OFST_RX2_EN	Offset cancellation feature enabling/disabling bit for the audio downlink channel 2. 0x0: Offset cancellation disabled. 0x1: Offset cancellation enabled.	RW	0
7:6	RESERVED		R	0x0
5:1	DN_OFST_RX1	Offset value for the audio downlink channel 1.	RW	0x00
0	DN_OFST_RX1_EN	Offset cancellation feature enabling/disabling bit for the audio downlink channel 1. 0x0: Offset cancellation disabled. 0x1: Offset cancellation enabled.	RW	0

**Table 23-506. Register Call Summary for Register MCPDM\_DN\_OFFSET**

Multichannel PDM Controller

- [FIFO Downlink Management: \[0\] \[1\] \[2\] \[3\]](#)
- [McPDM Global Initialization: \[4\] \[5\] \[6\] \[7\]](#)
- [McPDM Register Summary: \[8\]](#)

## 23.7 Digital Microphone Module

This section describes the digital microphone (DMIC) module.

### 23.7.1 DMIC Overview

The digital microphone (DMIC) module consists of an audio module dedicated to a mobile telephone terminal. The DMIC allows the support of up to six digital microphones. It provides an interface between the audio backend (ABE) module and microphones connected through the 6-wire DMIC serial interface.

The DMIC supports six paths. Because each path is composed of two digital microphone channels, the DMIC can be used with three stereo or six mono microphones. Each path is enabled or disabled independently.

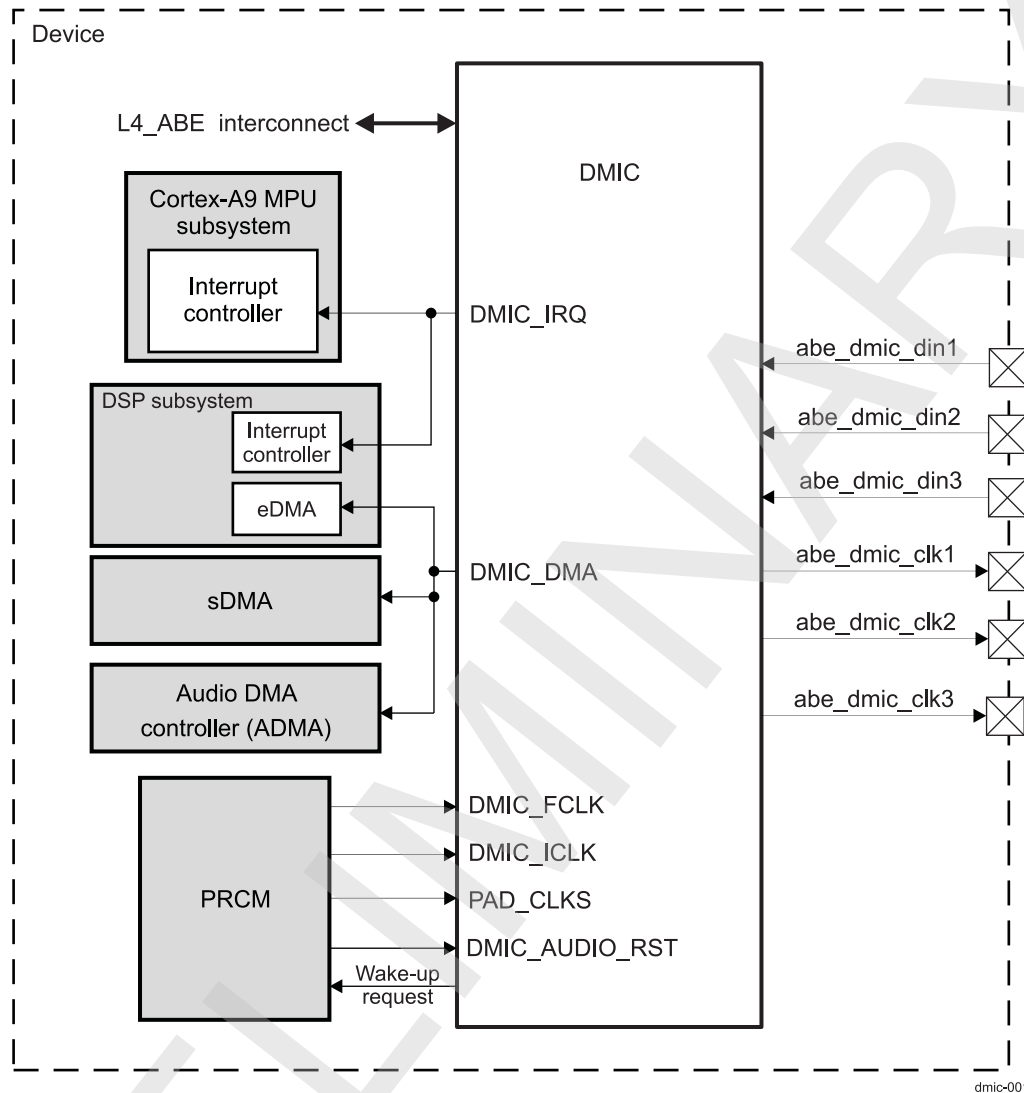
The DMIC generates a pulse-density modulated (PDM) stream of bits and transfers it in one period or one half-period of the clock provided by the DMIC (oversampling clock).

Each microphone is directly connected to a filter decimator to extract the audio samples at the desired accuracy and sample rate. Data are stored in the first in first out (FIFO), which is controlled by interrupt request (IRQ) and direct memory access (DMA) request.

The DMIC pads are supplied with 1.8 V. These pads assure the transition to the host supply.

[Figure 23-174](#) shows the DMIC in the device.

Figure 23-174. DMIC



The main features of the DMIC are:

- Six external pin connections (three data lines and three clock lines)
- Using the three clock lines, the DMIC delivers three clock signals (same frequency, individually gateable) for all digital microphones.
- Using the three data lines, the DMIC receives audio data from external microphones.
- Supports stereo (up to three) and mono (up to six) digital microphones
- Each line can support two microphones working in clock phase opposition.
- Selectable functional clock source
- Programmable output clock frequency ( $32 \times FS$ ,  $50 \times FS$ ,  $64 \times FS$ , or  $80 \times FS$ , where  $FS = 48 \text{ kHz}$ )
- Programmable data sampling sensibility (rising or falling edge)
- One RX FIFO (16 words of 144 bits)
- Supports idle request/acknowledge power, reset, and clock management (PRCM) module protocol
- Interconnect sample format: 32 bits (only 24 are significant)
- One IRQ to the microprocessor unit (Cortex™-A9 microprocessor unit [MPU]) and digital signal processor (DSP) subsystems
- One DMA request capability on programmable FIFO threshold

### 23.7.2 DMIC Environment

This section describes the DMIC application fields from an environment point of view (that is, external connections). It presents the DMIC connectivity options, lists possible interfaces, and details the protocol and data format used in each case.

#### 23.7.2.1 DMIC Interface With External MIC

##### 23.7.2.1.1 DMIC Interface Overview

The DMIC interface is a 6-wire interface. The DMIC module can support up to three external stereo and six mono digital microphones. A DMIC is an external module that can deliver data. A clock is sent from the host to the microphone.

Figure 23-175 and Figure 23-176 show two cases of use of the DMIC, but all configurations are possible with stereo and mono digital microphones. The output of the DMIC is in 32-bit format.

Figure 23-175 shows the DMIC used with three stereo DMICs.

Figure 23-175. DMIC With Three Stereo DMICs

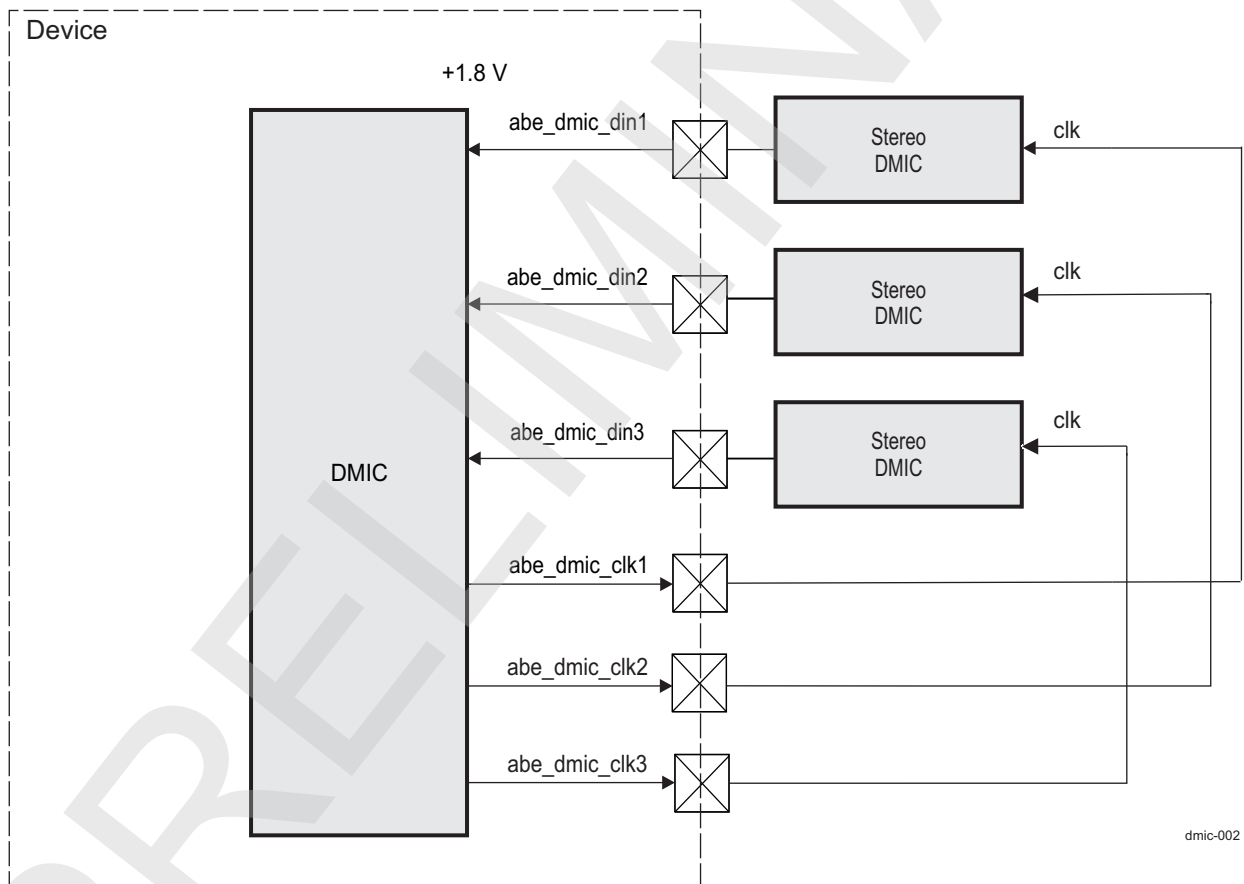
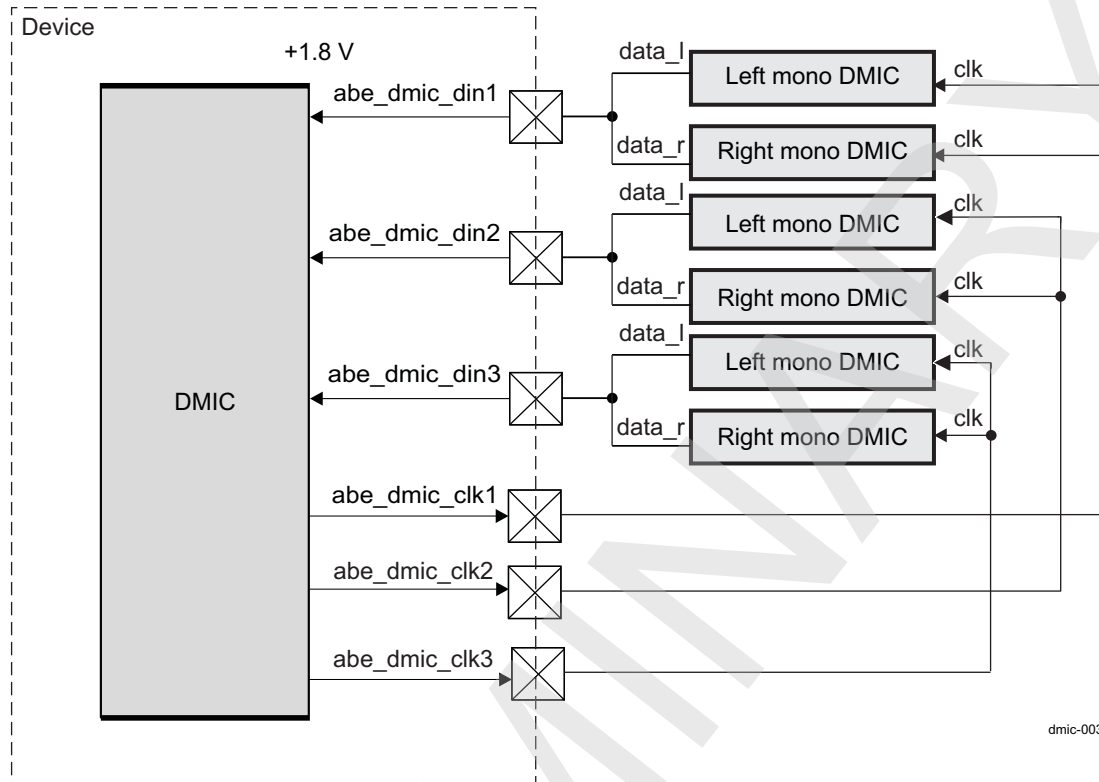


Figure 23-176 shows the DMIC used with six mono DMICs.

Figure 23-176. DMIC With Six Mono DMICs



### 23.7.2.1.2 DMIC Signals

Table 23-507 describes the signals in the DMIC module.

Table 23-507. Input/Output

Signal Name	I/O <sup>(1)</sup>	Description	Reset Value <sup>(1)</sup>
abe_dmic_clk1	O	Digital (stereo) microphone clock output 1	0
abe_dmic_clk2	O	Digital (stereo) microphone clock output 2	0
abe_dmic_clk3	O	Digital (stereo) microphone clock output 3	0
abe_dmic_din1	I	Digital (stereo) microphone data input 1	Hi-Z
abe_dmic_din2	I	Digital (stereo) microphone data input 2	Hi-Z
abe_dmic_din3	I	Digital (stereo) microphone data input 3	Hi-Z

<sup>(1)</sup> I = Input; O = Output; Hi-Z = High impedance

All DMIC external connections have pullup or pulldown resistors, selected and configured in the control module registers.

### 23.7.2.1.3 Serial Data Format

The DMIC generates a PDM stream of bits and transfers it in one period or one half-period of the clock. Then, the filter decimator extracts the 24-bit audio samples.

#### 23.7.2.1.3.1 DMIC Clock Generation and Synchronization

When DMIC\_POLAR is set to 0, the microphone left data is generated on the abe\_dmic\_clk low level. When DMIC\_POLAR is set to 1, the microphone left data is generated on the abe\_dmic\_clk high level. There are three DMIC\_POLARx bits (where x = 1, 2, or 3) corresponding to the three stereo paths:

- [DMIC\\_CTRL\[4\]](#) for path 1

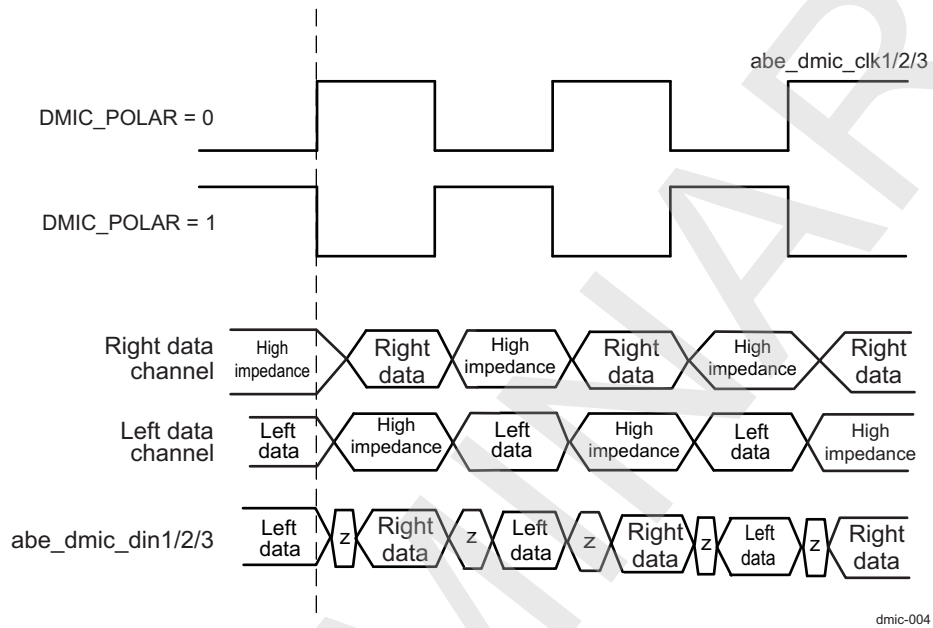


- [DMIC\\_CTRL\[5\]](#) for path 2
- [DMIC\\_CTRL\[6\]](#) for path 3

The DMIC\_POLARx bits are set depending on the configuration of the external digital microphone.

[Figure 23-177](#) is a diagram of the DMIC interface with a DMIC.

**Figure 23-177. DMIC Interface Diagram With External DMICs**



### 23.7.2.2 DMIC Functional Path

The DMIC path processes, decimates, and filters the data from the DMIC interface, and stores the processed data in the FIFO. The FIFO is controlled by IRQ and DMA requests and is fed outside of the DMIC.

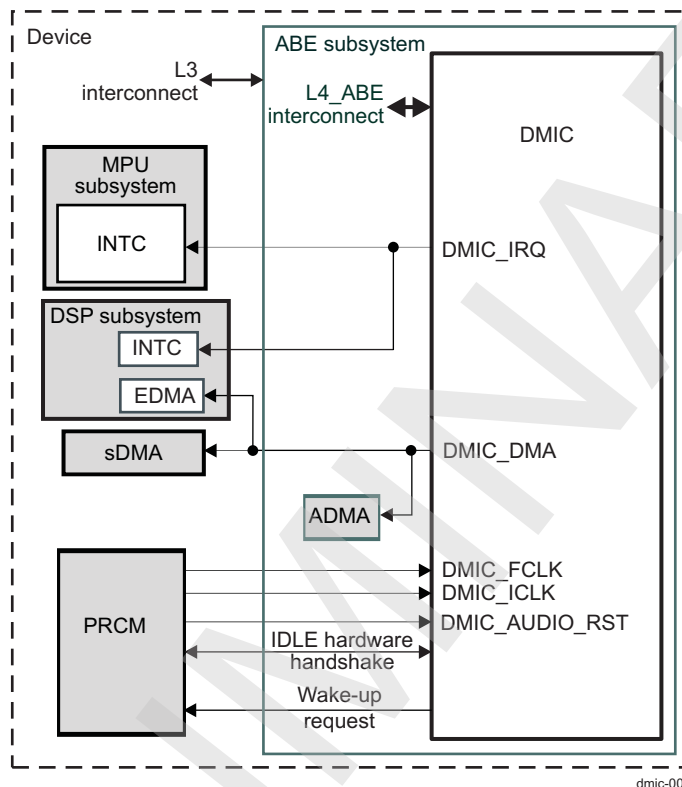
For more information about microphone data processing, see [Section 23.7.4.7, Audio Data Processing](#).

### 23.7.3 DMIC Integration

This section describes the integration of the DMIC in the device, and includes information about clocks, resets, and hardware requests.

Figure 23-178 shows the DMIC integration.

**Figure 23-178. DMIC Integration**



**NOTE:** For more information about the IDLE hardware handshake and the wake-up request, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

The DMIC integration includes:

- One functional clock
- One interface clock
- One IRQ
- One sDMA/ADMA/eDMA request
- IDLE hardware handshake
- Wake-up request

[Table 23-508](#) through [Table 23-510](#) summarize the integration of the module in the device.

**Table 23-508. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
DMIC	PD_AUDIO	N/A	L4_ABE

**Table 23-509. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DMIC	DMIC_ICLK	ABE_ICLK2	PRCM	Interface clock
	DMIC_FCLK	DMIC_ABE_FCLK	PRCM	Functional clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
DMIC	DMIC_AUDIO_RST	AUDIO_RST	PRCM	Module hardware reset, nonretention

**Table 23-510. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
DMIC	DMIC_IRQ	D_IRQ_83	DSP	DMIC interrupt to the DSP subsystem
	DMIC_IRQ	MA_IRQ_114	Cortex™-A9 MPU	DMIC interrupt to the Cortex-A9 MPU subsystem
DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
DMIC	DMIC_DMA	S_DMA_66	sDMA	DMA request to the system DMA (sDMA)
	DMIC_DMA	A_DMA_1	ADMA	DMA request to the audio DMA (ADMA)
	DMIC_DMA	D_DMA_44	eDMA	DMA request to the DSP DMA (eDMA)

**NOTE:**

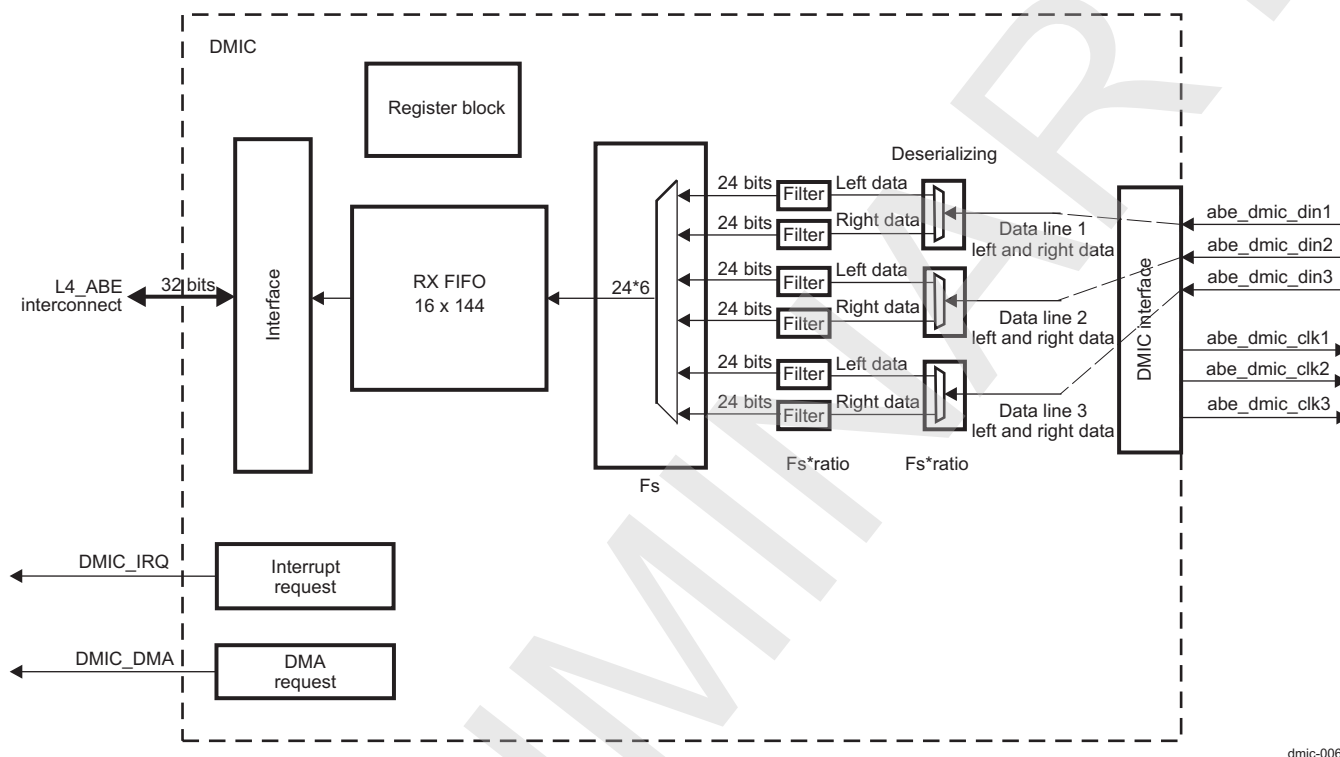
- For a description of the interrupt source, see [Section 23.7.4.5, Interrupt Requests](#).
- For a description of the DMA source, see [Section 23.7.4.6, DMA Requests](#).
- For information about PRCM clock gating and management and PRCM reset sources and distribution, see [Section 3.1.1.2.1, Power Domain](#), in [Chapter 3, Power, Reset, and Clock Management](#).

## 23.7.4 DMIC Functional Description

### 23.7.4.1 Block Diagram

Figure 23-179 is the DMIC block diagram.

Figure 23-179. DMIC Block Diagram



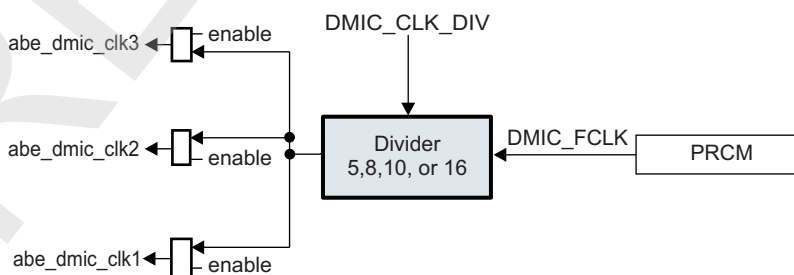
dmic-006

### 23.7.4.2 DMIC Clock Configuration

The DMIC operates from one interface clock to interface with the L4-ABE interconnect and one selectable (among three) functional clock. These clocks are provided by the PRCM module. For the interface clock, request and acknowledge are set up between the PRCM module and the DMIC.

Figure 23-180 shows the DMIC clock selection.

Figure 23-180. DMIC Clock Selection



dmic-007

DMIC\_FCLK is the source clock signal. Several source clocks are available and are selected according to the configuration of the `DMIC_CTRL[9:7]` DMIC\_CLK\_DIV bit field. For more information about

DMIC\_FCLK selection, see [Figure 23-180](#). `abe_dmic_clkx` are the output clocks generated in the DMIC and are fed to the external microphones. They are also the internal clocks of the external microphones. The performance in terms of signal-to-noise ratio (SNR) depends on these clocks. Users must take care about the value of `abe_dmic_clkx` (the analog-to-digital converter [ADC] clock) when selecting the source clock (DMIC\_FCLK). [Table 23-511](#) describes the `abe_dmic_clkx` generation.

**Table 23-511. `abe_dmic_clkx` Generation**

DMIC_FCLK	DMIC_CLK_DIV	DMIC_FCLK/ <code>abe_dmic_clk</code>	<code>abe_dmic_clk</code> = ADC Clock	Ratio = <code>abe_dmic_clk</code> /FS = 48 kHz
19.2 MHz	000	8	2.4 MHz	50
	001	5	3.84 MHz	80
24 MHz	010	10	2.4 MHz	50
24.576 MHz	011	8	3.072 MHz	64
	100	16	1.536 MHz	32
12 MHz	101	5	2.4 MHz	50

The maximum clock on the interface is 3.84 MHz, with a source clock of 19.2 MHz.

### 23.7.4.3 DMIC Software Reset

[Table 23-512](#) lists the different types of software reset available at the module level.

**Table 23-512. Reset Description**

Type	Name	Source	Description
Software	<code>DMIC_SYSCONFIG</code> [1] SOFTRESET	–	Module software reset
Software	<code>DMIC_CTRL</code> [10] SW_DMIC_RST	–	Reset-enabled DMIC paths

The software path reset is gated by:

- `DMIC_CTRL`[0] `DMIC_UP1_EN` for path 1
- `DMIC_CTRL`[1] `DMIC_UP2_EN` for path 2
- `DMIC_CTRL`[2] `DMIC_UP3_EN` for path 3
- `DMIC_CTRL`[10] SW\_DMIC\_RST

When the `DMIC_CTRL`[10] SW\_DMIC\_RST bit is set to 1, the corresponding enabled DMIC paths are reset. Resets are cleared by writing 0.

**NOTE:** A software reset is required to change the configuration of enabled paths; for example, to disable path 3, a software reset of path 3 is mandatory for the change to be considered.

A module software reset can be performed by setting the `DMIC_SYSCONFIG`[1] SOFTRESET bit. This action starts a soft reset sequence.

The software reset status of the module can be checked by reading the `DMIC_SYSCONFIG`[0] SOFTRESET bit (0: reset is done; 1: reset is ongoing).

### 23.7.4.4 DMIC Power Management

[Table 23-513](#) describes the power-management features available for the DMIC.

**Table 23-513. Local Power-Management Features**

Feature	Registers	Description
Clock autogating	N/A	Feature not available
Slave idle modes	<code>DMIC_SYSCONFIG</code> [3:2] SIDLEMODE	Force-idle, no-idle, and smart-idle modes are available.
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available

**Table 23-513. Local Power-Management Features (continued)**

Feature	Registers	Description
Global wake-up enable	N/A	Feature not available
Wake-up sources enable	<a href="#">DMIC_IRQWAKEEN</a> [0] IRQ_WAKEUP_EN <a href="#">DMIC_DMAWAKEEN</a> [0] DMA_WAKEUP_EN	This register holds one active-high enable bit per event source that can generate a wake-up signal.

#### 23.7.4.5 DMIC Interrupt Requests

[Figure 23-181](#) lists the event flags, and their mask, that can cause module interrupts.

Table 23-514. Interrupt Events

Nonmaskable Event Flag <sup>(1)</sup>	Maskable Event Flag	Event Mask Bit	Event Unmask Bit	Description
DMIC_IRQSTATUS_RAW[3] DMIC_IRQ_EMPTY	DMIC_IRQSTATUS[3] DMIC_IRQ_EMPTY	DMIC_IRQENABLE_CLR[3] DMIC_IRQ_EMPTY_MASK	DMIC_IRQENABLE_SET[3] DMIC_IRQ_EMPTY_MASK	FIFO empty event. This event occurs when the FIFO is already empty and a new word is read from the FIFO by software.
DMIC_IRQSTATUS_RAW[2] DMIC_IRQ_ALST_FULL	DMIC_IRQSTATUS[2] DMIC_IRQ_ALST_FULL	DMIC_IRQENABLE_CLR[2] DMIC_IRQ_ALST_FULL_MASK	DMIC_IRQENABLE_SET[2] DMIC_IRQ_ALST_FULL_MASK	FIFO almost-full event. This event occurs when the FIFO contains (FIFO size – 1) words and a new word is written to the FIFO.
DMIC_IRQSTATUS_RAW[1] DMIC_IRQ_FULL	DMIC_IRQSTATUS[1] DMIC_IRQ_FULL	DMIC_IRQENABLE_CLR[1] DMIC_IRQ_FULL_MASK	DMIC_IRQENABLE_SET[1] DMIC_IRQ_FULL_MASK	FIFO full event. This event occurs when the FIFO is already full and a new word is written to the FIFO.
DMIC_IRQSTATUS_RAW[0] DMIC_IRQ	DMIC_IRQSTATUS[0] DMIC_IRQ	DMIC_IRQENABLE_CLR[0] DMIC_IRQ_MASK	DMIC_IRQENABLE_SET[0] DMIC_IRQ_MASK	FIFO read request event. This event occurs when the number of words stored in the FIFO is equal to or greater than the FIFO threshold.

<sup>(1)</sup> The [DMIC\\_IRQSTATUS\\_RAW](#) register is used primarily for debug purposes.

The interrupt line is asserted (active high) when one of the following signals appears:

- DMIC\_IRQ\_EMPTY
- DMIC\_IRQ\_ALST\_FULL
- DMIC\_IRQ\_FULL
- DMIC\_IRQ

Read the [DMIC\\_IRQSTATUS](#) register to determine which event occurs.

**CAUTION**

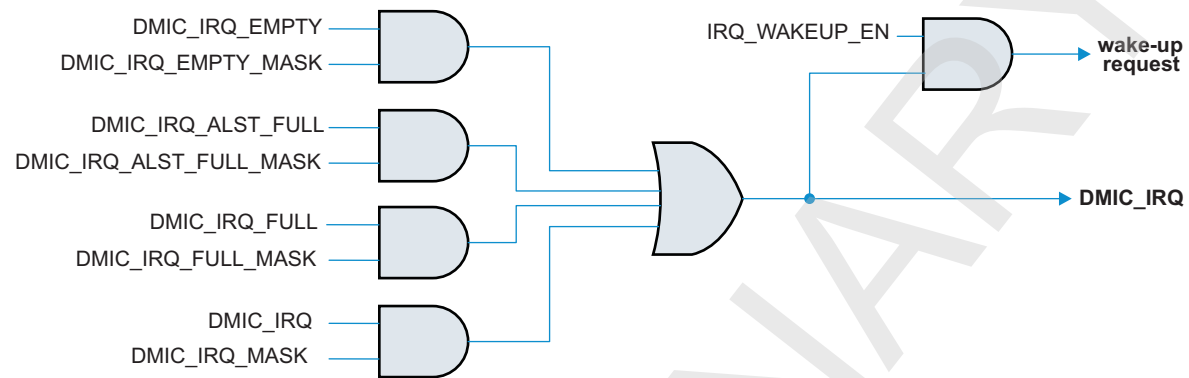
Interrupt lines must be cleared by software. Write 1 in the [DMIC\\_IRQSTATUS](#) register to clear the corresponding event.

An event can generate an IRQ when the corresponding mask bit in the [DMIC\\_IRQENABLE\\_SET](#) register is set to 1. Events are register bits that belong to the [DMIC\\_IRQSTATUS](#) and [DMIC\\_IRQSTATUS\\_RAW](#) registers (the same register accessed by two interconnect addresses). In the case of the [DMIC\\_IRQSTATUS\\_RAW](#) register, status event is set even if it is not enabled (used primarily for debug purposes). The [DMIC\\_IRQSTATUS](#) register is set by writing 1 to the [DMIC\\_IRQSTATUS\\_RAW](#) register.

Each internal interrupt signal can be masked by enabling the corresponding mask in the [DMIC\\_IRQENABLE\\_SET](#) register or disabling it in the [DMIC\\_IRQENABLE\\_CLR](#) register. This mechanism is defined in [Section 23.7.6, DMIC Register Manual](#).

[Figure 23-181](#) shows the interrupt tree with the event generated and the mask that enables or disables the event. All events are summarized to give the single interrupt generation.

Figure 23-181. Interrupt Tree



dmic-008

#### 23.7.4.6 DMIC DMA Requests

The DMIC can generate one DMA request to the sDMA, ADMA, and eDMA controllers (see [Table 23-515](#)).



**Table 23-515. DMIC DMA Request**

Name	Description	Enable DMA Bit	Disable DMA Bit
DMIC_DMA_ENABLE	Data ready in FIFO (threshold is reached)	<a href="#">DMIC_DMAENABLE_SET</a> [0] DMA_ENABLE	<a href="#">DMIC_DMAENABLE_CLR</a> [0] DMA_ENABLE

The DMA lines can be enabled or disabled depending on the value of the bits in the [DMIC\\_DMAENABLE\\_SET](#) and [DMIC\\_DMAENABLE\\_CLR](#) registers.

A DMA request is sent when data is ready in the FIFO. This request is used only to notify the host that data are ready in the FIFO.

### 23.7.4.7 Audio Data Processing

The DMIC supports six audio paths and can be used with three stereo or six mono microphones. Each path is independent and is enabled or disabled by using the following bits:

- [DMIC\\_CTRL](#)[0] DMIC\_UP1\_EN
- [DMIC\\_CTRL](#)[1] DMIC\_UP2\_EN
- [DMIC\\_CTRL](#)[2] DMIC\_UP3\_EN

#### 23.7.4.7.1 Filtering

The external DMIC generates a PDM stream of bits and transfers it in one period or one half-period of the clock provided by the DMIC inside the device. The aim of the DMIC audio path is to process data from the DMIC interface, decimate and filter the data, and store the processed data in the FIFO.

The format of the DMIC data is 24 bits (the transfer format is 32 bits long). The following bits allow switching the polarity of the interface:

- [DMIC\\_CTRL](#)[4] DMIC\_POLAR1
- [DMIC\\_CTRL](#)[5] DMIC\_POLAR2
- [DMIC\\_CTRL](#)[6] DMIC\_POLAR3

For more information, see [Section 23.7.2.1.3.1](#), *DMIC Clock Generation and Synchronization*.

Processed data from the DMIC interface are stored in the FIFO. The FIFO is controlled by the IRQ and DMA request and externally fed by the DMIC using the following interconnect format.

The three uplink paths are identical. Each uplink path is composed of a left and a right channel. The DMIC interface delivers six parallel data of 1 bit (3 × 2 paths of stereo data that can be used at six mono paths). Each bit goes to a filter. The aim of the filter is to limit the noise and perform decimation by factor D. Factor D depends on the [DMIC\\_CLK\\_DIV](#) register bits, as described in the ratio defined in [Table 23-511](#). The filters are sinc filter order 4. They are chosen with regard to the nature of the analog converter.

The format of the filter output is 24 bits.

### 23.7.4.8 FIFO Management

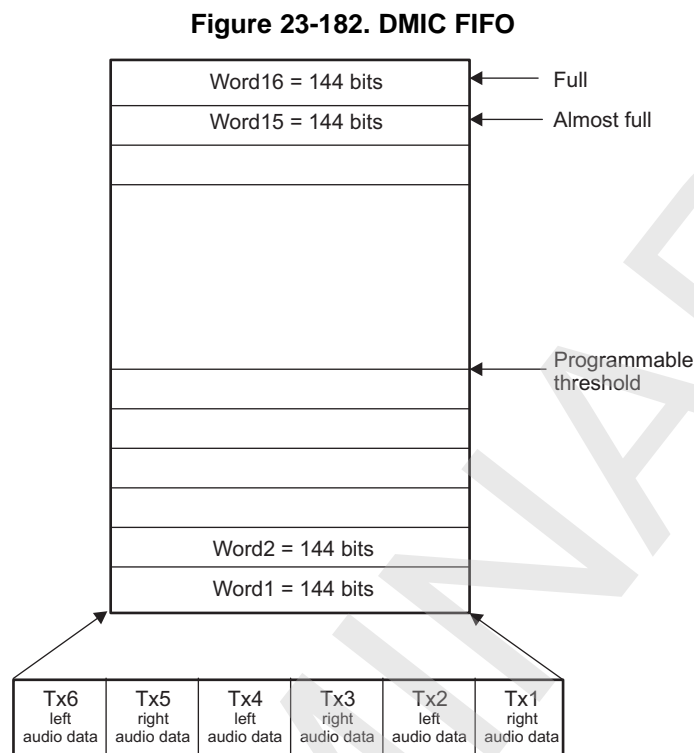
The DMIC implements an internal FIFO controlled by IRQ and DMA request. The host must read words in the FIFO. The interrupt line is used not only to inform the host that data are ready in the FIFO, but also to warn the host about events that can occur, such as FIFO full, almost full, or empty. A DMA request is used only to inform the host that data are ready in the FIFO.

#### 23.7.4.8.1 FIFO General Description

The DMIC communicates with the host through L4-ABE interconnect access. DMIC audio data coming from external microphones are stored in the FIFO. The DMIC FIFO allows the exchange of information between the DMIC and the host. Each audio data is 24 bits long. The DMIC FIFO contains 16 words. Each word is the combination of six audio data (24 × 6 = 144 bits) coming from the interface. The FIFO is controlled by interrupt lines and DMA requests. For more information about IRQs and DMA controls, see [Section 23.7.4.5](#), *DMIC Interrupt Requests*, and [Section 23.7.4.6](#), *DMIC DMA Requests*.

FIFO write access is done in the functional clock domain, and read access is done in the interface clock domain.

Figure 23-182 is an overview of the DMIC FIFO.



### 23.7.4.8.2 FIFO Operations

Audio data are filled in the FIFO from DMIC internal filters. The host must read words in the FIFO. The number of reads depends on the application that is configured by the following bits:

- [DMIC\\_CTRL\[0\]](#) DMIC\_UP1\_EN
- [DMIC\\_CTRL\[2\]](#) DMIC\_UP3\_EN
- [DMIC\\_CTRL\[1\]](#) DMIC\_UP2\_EN
- [DMIC\\_FIFO\\_CTRL\[3:0\]](#) DMIC\_TRESH (FIFO threshold)

If only uplink path 2 and path 3 are enabled, for example, the Tx1 and Tx2 words are left and right data for path 2, and Tx3 and Tx4 are left and right data for path 3.

When the FIFO threshold is reached, and depending on conditions, an event is sent to the DMA handler. The host must read the exact number of words in the FIFO.

Two accesses are required to read a complete stereo audio path.

The number of read accesses from the host is determined by the following equation:

$$\text{Num\_rd\_access} = (\text{DMIC\_UP1\_EN} + \text{DMIC\_UP2\_EN} + \text{DMIC\_UP3\_EN}) \times 2 \times (\text{DMIC\_TRESH})$$

The FIFO threshold is programmed through the [DMIC\\_FIFO\\_CTRL\[3:0\]](#) DMIC\_TRESH bit field.

---

**NOTE:** If DMIC\_TRESH = 1, the host is expected to perform  $1 \times (\text{DMIC\_UP1\_EN} + \text{DMIC\_UP2\_EN} + \text{DMIC\_UP3\_EN}) \times 2$  read access every 96 kHz ( $2 \times \text{FS}$ ).

If DMIC\_TRESH = 2, the host is expected to perform  $(\text{DMIC\_UP1\_EN} + \text{DMIC\_UP2\_EN} + \text{DMIC\_UP3\_EN}) \times 4$  read access every 48 kHz (FS).

If DMIC\_TRESH = 12, the host is expected to perform  $(\text{DMIC\_UP1\_EN} + \text{DMIC\_UP2\_EN} + \text{DMIC\_UP3\_EN}) \times 12$  access every 8 kHz ( $\text{FS}/6$ ).

---

### 23.7.4.8.3 FIFO Output Range Definition

The [DMIC\\_CTRL\[3\]](#) DMICOUTFORMAT bit allows determining the format of the output of the DMIC uplink module.

When DMICOUTFORMAT = 0, the data going from the FIFO are left-shifted by 8 bits. When DMICOUTFORMAT = 1, the data are signed-extended on 32 bits.

The DMIC uplink modules process audio data based on twos complement format.

---

**NOTE:** When DMICOUTFORMAT = 0, the maximum positive swing analog voltage corresponds to a final DMIC output of 0x7FFF FF00. The maximum negative swing analog voltage corresponds to a final DMIC output of 0x8000 0000.

When DMICOUTFORMAT = 1, the sign is extended: The maximum positive swing analog voltage corresponds to a final DMIC output of 0x007F FFFF. The maximum negative swing analog voltage corresponds to a final DMIC output of 0xFF80 0000.

---

Reading the value of the corresponding audio data in the FIFO requires access to the [DMIC\\_DATA\\_REG](#) register. The first access gives the value of Tx1 right data, the second gives the value of Tx2 left data, up to Tx6 (the left audio data of the last audio path). If the path is disabled, 0x0 is read. All data are duplicated (for debug purposes) in their corresponding left or right data registers. [Table 23-516](#) describes the audio data path registers.

**Table 23-516. Audio Data Path Registers**

Stereo Audio Path	Mono Audio Path	Data Register
Right path 1	Mono path 1	<a href="#">DMIC_FIFO_DMIC1R_DATA[23:0]</a>
Left path 1	Mono path 2	<a href="#">DMIC_FIFO_DMIC1L_DATA[23:0]</a>
Right path 2	Mono path 3	<a href="#">DMIC_FIFO_DMIC2R_DATA[23:0]</a>
Left path 2	Mono path 4	<a href="#">DMIC_FIFO_DMIC2L_DATA[23:0]</a>
Right path 3	Mono path 5	<a href="#">DMIC_FIFO_DMIC3R_DATA[23:0]</a>
Left path 3	Mono path 6	<a href="#">DMIC_FIFO_DMIC3L_DATA[23:0]</a>

The output of the DMIC path for interconnect access is in 32-bit format.

## 23.7.5 DMIC Programming Guide

### 23.7.5.1 DMIC Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the module.

#### 23.7.5.1.1 Global Initialization

##### 23.7.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the DMIC is used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DMIC. For more information, see [Section 23.7.3, DMIC Integration](#) and [Section 23.7.2, DMIC Environment](#).

[Table 23-517](#) describes the global initialization of surrounding modules.

**Table 23-517. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. See <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Control module	Module-specific pad muxing must be set in the control module. See <a href="#">Chapter 18 Control Module</a> .
(optional) Cortex-A9 MPU interrupt controller (INTC) (or DSP INTC)	INTC configuration must be done to enable the interrupts from the DMIC. See <a href="#">Chapter 17, Interrupt Controllers</a> .
(optional) sDMA (or eDMA or ADMA)	DMA configuration must be done to enable the DMIC DMA channel requests.
(optional) Interconnect	For more information about the interconnect configuration, see <a href="#">Chapter 13, Interconnect</a> .

**NOTE:** The INTC and DMA configurations are necessary if the interrupt and DMA-based communication modes are used.

#### 23.7.5.1.1.2 DMIC Global Initialization

##### 23.7.5.1.1.2.1 Main Sequence – DMIC Global Initialization

The procedure in [Table 23-518](#) initializes the DMIC after a power-on reset (POR).

**Table 23-518. DMIC Global Initialization**

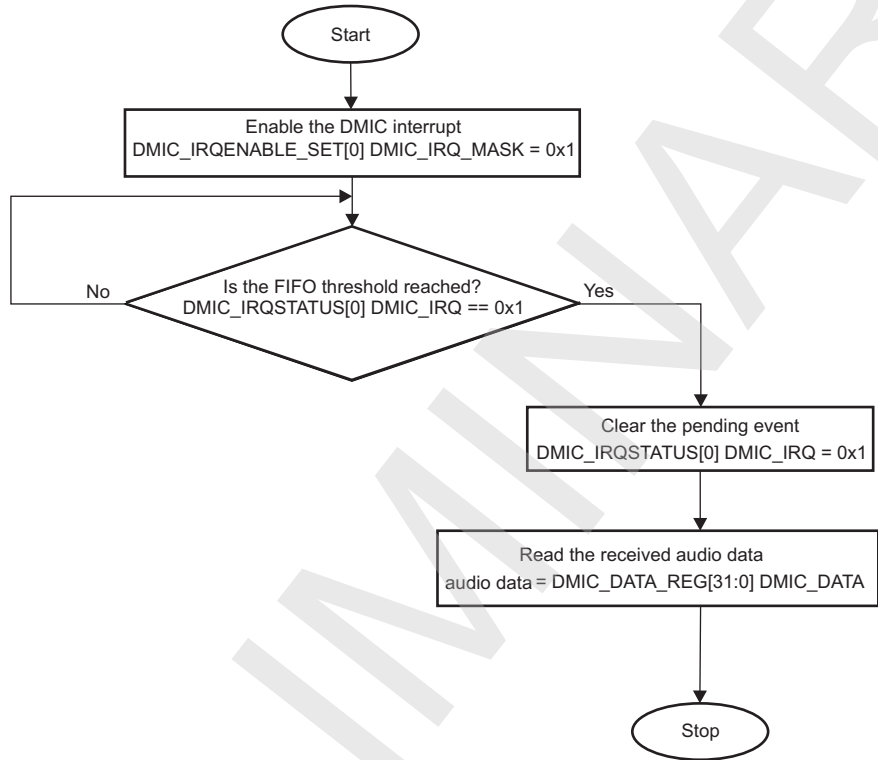
Step	Register/Bit Field/Programming Model	Value
Execute a software reset.	DMIC_SYSCONFIG[0] SOFTRESET	0x1
Wait for the software reset completion by polling.	DMIC_SYSCONFIG[0] SOFTRESET	==0x0
Configure the local power management.	DMIC_SYSCONFIG[3:2] SIDLEMODE	0x-
Select the DMIC FIFO threshold.	DMIC_FIFO_CTRL[3:0] DMIC_TRESH	0x-
Select the DMIC output clock frequency.	DMIC_CTRL[9:7] DMIC_CLK_DIV	0x-
Select the DMIC FIFO data output format.	DMIC_CTRL[3] DMICOUTFORMAT	0x-
Select the level (high or low) of the clock signal <code>abe_dmic_clk_i</code> when the left data is to be generated (extracted) from the external microphone <code>i</code> (where <code>i = 1 to 3</code> ).	DMIC_CTRL[j] DMIC_POLAR <sub>i</sub> (where <code>j = 4 to 6</code> , and <code>i = 1 to 3</code> )	0x-
Enable audio path <code>i</code> (where <code>i = 1 to 3</code> ).	DMIC_CTRL[k] DMIC_POLAR <sub>i</sub> (where <code>k = 0 to 2</code> , and <code>i = 1 to 3</code> )	0x1

23.7.5.1.2 Operational Mode Configuration

23.7.5.1.2.1 DMIC Polling Mode

Figure 23-183 shows the polling mode. Table 23-519 summarizes the register call for the polling mode. For the number of read accesses used while reading the received audio data, see Section 23.7.4.8, FIFO Management.

Figure 23-183. Polling Mode



dmic-010

Table 23-519. Register Call Summary for Polling Mode

Register Name	Register Name	Register Name
DMIC_IRQENABLE_SET	DMIC_IRQSTATUS	DMIC_DATA_REG

23.7.5.1.2.2 DMIC Interrupt Mode

Table 23-520 lists the procedure to initialize interrupt mode.

Table 23-520. Interrupt Mode

Step	Register/Bit Field/Programming Model	Value
Enable the DMIC interrupts.	DMIC_IRQENABLE_SET[3:0]	0xF

23.7.5.1.2.3 DMIC DMA Mode

Table 23-521 lists the procedure to initialize DMA mode.

Table 23-521. DMA Mode

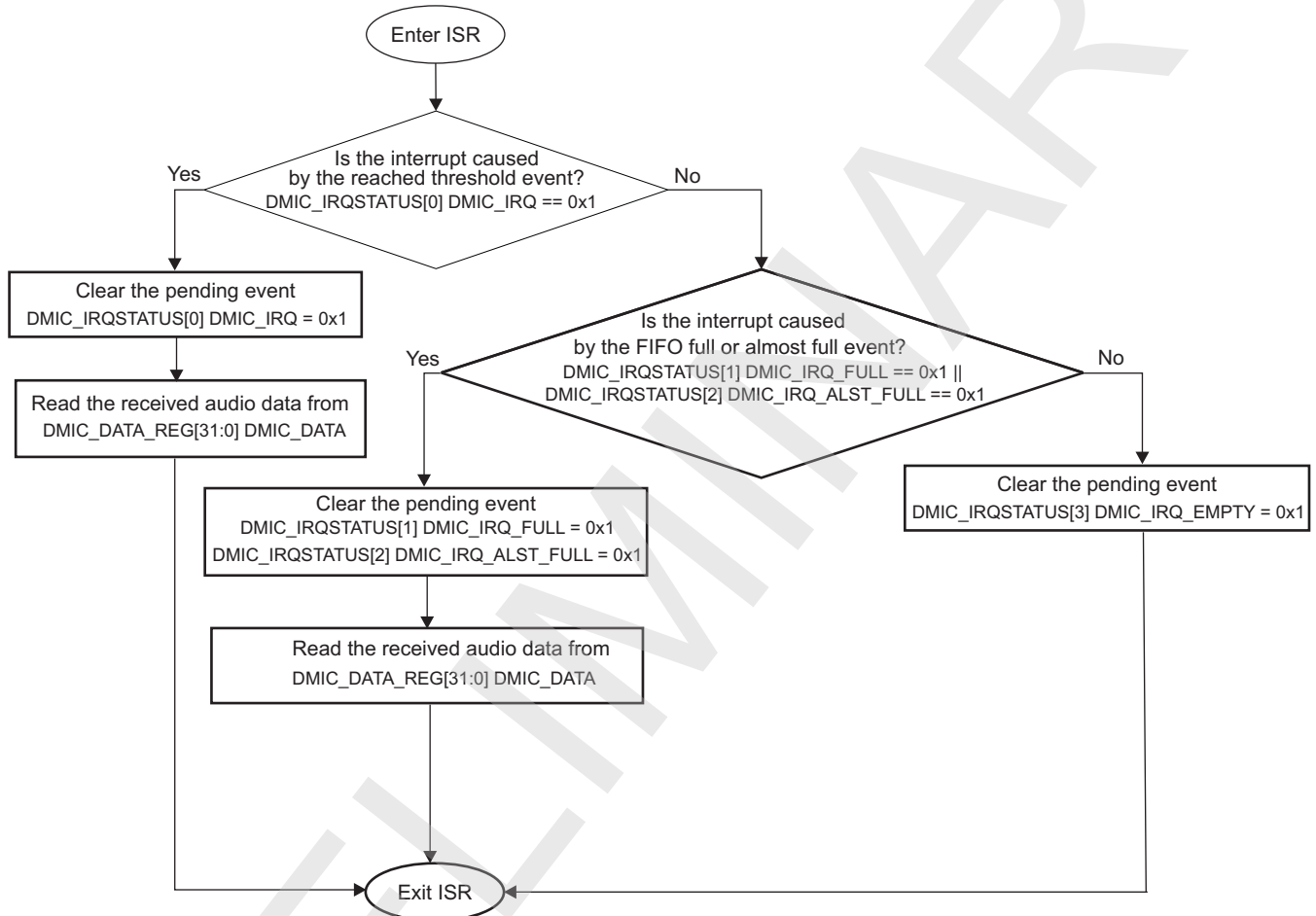
Step	Register/Bit Field/Programming Model	Value
Enable the DMIC DMA interrupt event.	DMIC_DMAENABLE_SET[0] DMA_ENABLE	0x1

### 23.7.5.1.3 DMIC Events Servicing

#### 23.7.5.1.3.1 DMIC Interrupt Servicing

This section describes the interrupt event servicing of the module. Figure 23-184 shows the interrupt handler.

**Figure 23-184. Interrupt Handler**



dmic-011

When the interrupt is caused by the reached threshold event, for the number of read accesses used while reading the received audio data, see Section 23.7.4.8, *FIFO Management*.

When the interrupt is caused by a FIFO full or almost-full event, software must read all the available data in the FIFO. If a FIFO full event occurred, the number of read accesses must be the maximum:

$$\text{Num\_rd\_access} = (\text{DMIC\_UP1\_EN} + \text{DMIC\_UP2\_EN} + \text{DMIC\_UP3\_EN}) \times 2 \times 15$$

If a FIFO almost-full event occurred, the number of read accesses must be:

$$\text{Num\_rd\_access} = (\text{DMIC\_UP1\_EN} + \text{DMIC\_UP2\_EN} + \text{DMIC\_UP3\_EN}) \times 2 \times 14$$

Table 23-522 summarizes the register call for the interrupt handler.

**Table 23-522. Register Call Summary for the Interrupt Handler**

Register Name	Register Name
DMIC_IRQSTATUS	DMIC_DATA_REG

PRELIMINARY

## 23.7.6 DMIC Register Manual

### 23.7.6.1 DMIC Instance Summary

Table 23-523 is the DMIC instance summary.

**Table 23-523. DMIC Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-A9 Private Access	Base Address DSP Private Access	Size
DMIC	0x4902 E000	0x4012 E000	0x2 E000	4KB

**NOTE:** Private access is an access that does not use the L3/L4 interconnects.

### 23.7.6.2 DMIC Registers

#### 23.7.6.2.1 DMIC Register Summary

Table 23-524 summarizes the DMIC register mapping.

**Table 23-524. DMIC Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
DMIC_REVISION	R	32	0x0000 0000	0x4902 E000	0x4012 E000	0x2 E000
DMIC_SYSCONFIG	RW	32	0x0000 0010	0x4902 E010	0x4012 E010	0x2 E010
RESERVED	RW	32	0x0000 0020	0x4902 E020	0x4012 E020	0x2 E020
DMIC_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4902 E024	0x4012 E024	0x2 E024
DMIC_IRQSTATUS	RW	32	0x0000 0028	0x4902 E028	0x4012 E028	0x2 E028
DMIC_IRQENABLE_SET	RW	32	0x0000 002C	0x4902 E02C	0x4012 E02C	0x2 E02C
DMIC_IRQENABLE_CLR	RW	32	0x0000 0030	0x4902 E030	0x4012 E030	0x2 E030
DMIC_IRQWAKEEN	RW	32	0x0000 0034	0x4902 E034	0x4012 E034	0x2 E034
DMIC_DMAENABLE_SET	RW	32	0x0000 0038	0x4902 E038	0x4012 E038	0x2 E038
DMIC_DMAENABLE_CLR	RW	32	0x0000 003C	0x4902 E03C	0x4012 E03C	0x2 E03C
DMIC_DMAWAKEEN	RW	32	0x0000 0040	0x4902 E040	0x4012 E040	0x2 E040
DMIC_CTRL	RW	32	0x0000 0044	0x4902 E044	0x4012 E044	0x2 E044
DMIC_DATA_REG	R	32	0x0000 0048	0x4902 E048	0x4012 E048	0x2 E048
DMIC_FIFO_CTRL	RW	32	0x0000 004C	0x4902 E04C	0x4012 E04C	0x2 E04C
DMIC_FIFO_DMIC1R_DATA	R	32	0x0000 0050	0x4902 E050	0x4012 E050	0x2 E050
DMIC_FIFO_DMIC1L_DATA	R	32	0x0000 0054	0x4902 E054	0x4012 E054	0x2 E054
DMIC_FIFO_DMIC2R_DATA	R	32	0x0000 0058	0x4902 E058	0x4012 E058	0x2 E058
DMIC_FIFO_DMIC2L_DATA	R	32	0x0000 005C	0x4902 E05C	0x4012 E05C	0x2 E05C
DMIC_FIFO_DMIC3R_DATA	R	32	0x0000 0060	0x4902 E060	0x4012 E060	0x2 E060
DMIC_FIFO_DMIC3L_DATA	R	32	0x0000 0064	0x4902 E064	0x4012 E064	0x2 E064



23.7.6.2.2 DMIC Register Descriptions

Table 23-525 through Table 23-561 describe the DMIC registers.

Table 23-525. DMIC\_REVISION

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See Table 23-524.	<b>Instance</b>	DMIC
<b>Description</b>	IP Revision Identifier (X.Y.R) used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See <sup>(1)</sup> .

<sup>(1)</sup> TI internal data

Table 23-526. Register Call Summary for Register DMIC\_REVISION

DMIC Register Manual

- [DMIC Register Summary: \[0\]](#)

Table 23-527. DMIC\_SYSCONFIG

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See Table 23-524.	<b>Instance</b>	DMIC
<b>Description</b>	This register allows controlling various parameters of the DMIC interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	SIDLEMODE	FREEMU	SOFTRESET												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3:2	SIDLEMODE	Configuration of the local target state management (idlereq/idleack control)  0x0: Force-idle. IDLE request is acknowledged unconditionally and immediately. 0x1: No-idle. IDLE request is never acknowledged. 0x2: Smart-idle. The acknowledgment to an IDLE request is given based on the internal activity. 0x3: Reserved - do not use- no wake-up capability	RW	0x2
1	FREEMU	Sensitivity to emulation (debug) suspend input signal  0x0: IP module is sensitive to emulation suspend. 0x1: IP module is not sensitive to emulation suspend.	RW	0

Bits	Field Name	Description	Type	Reset
0	SOFTRESET	<p>Module software reset. The bit is automatically reset by the hardware. It has same effect as the main hardware reset.</p> <p>Write 0x0: No action</p> <p>Read 0x0: Reset done, no pending action</p> <p>Read 0x1: Reset (software or other) ongoing</p> <p>Write 0x1: Initiate software reset.</p>	RW	0

**Table 23-528. Register Call Summary for Register DMIC\_SYSCONFIG**

## DMIC Functional Description

- [DMIC Software Reset: \[0\] \[1\] \[2\]](#)
- [DMIC Power Management: \[3\]](#)

## DMIC Programming Guide

- [Main Sequence – DMIC Global Initialization: \[4\] \[5\] \[6\]](#)

## DMIC Register Manual

- [DMIC Register Summary: \[7\]](#)

**Table 23-529. DMIC\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	DMIC
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .		
<b>Description</b>	Component (that is, main) interrupt request status. Check the corresponding secondary status register. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											DMIC_IRQ_EMPTY	DMIC_IRQ_ALST_FULL	DMIC_IRQ_FULL	DMIC_IRQ	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	DMIC_IRQ_EMPTY	<p>Read 0x0: No event pending</p> <p>Write 0x0: No action</p> <p>Write 0x1: Set</p> <p>Read 0x1: IRQ event pending</p>	RW	0
2	DMIC_IRQ_ALST_FULL	<p>Read 0x0: No event pending</p> <p>Write 0x0: No action</p> <p>Write 0x1: Set</p> <p>Read 0x1: IRQ event pending</p>	RW W1toSet	0
1	DMIC_IRQ_FULL	<p>Read 0x0: No event pending</p> <p>Write 0x0: No action</p> <p>Write 0x1: Set</p> <p>Read 0x1: IRQ event pending</p>	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
0	DMIC_IRQ	Read 0x0: No event pending Write 0x0: No action Write 0x1: Set Read 0x1: IRQ event pending	RW W1toSet	0

**Table 23-530. Register Call Summary for Register DMIC\_IRQSTATUS\_RAW**

DMIC Functional Description

- [DMIC Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[8\]](#)

**Table 23-531. DMIC\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .	<b>Instance</b>	DMIC
<b>Description</b>	Component (that is, main) interrupt request status. Check the corresponding secondary status register. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DMIC_IRQ_EMPTY		DMIC_IRQ_ALST_FULL		DMIC_IRQ_FULL		DMIC_IRQ									

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	DMIC_IRQ_EMPTY	Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending	RW	0
2	DMIC_IRQ_ALST_FULL	This interrupt status is set when only one FIFO space is still available. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending	RW W1toClr	0
1	DMIC_IRQ_FULL	This interrupt status is set when FIFO is full and a new write access has been performed by filter. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event, if any. Read 0x1: IRQ event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
0	DMIC_IRQ	This interrupt status is set when FIFO threshold value defined in <a href="#">DMIC_FIFO_CTRL</a> is reached. Read 0x0: No event pending Write 0x0: No action Write 0x1: Clear pending event, if any. Read 0x1: IRQ event pending	RW W1toClr	0

**Table 23-532. Register Call Summary for Register DMIC\_IRQSTATUS**

DMIC Functional Description

- [DMIC Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)

DMIC Programming Guide

- [DMIC Polling Mode: \[8\]](#)
- [DMIC Interrupt Servicing: \[9\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[10\]](#)

**Table 23-533. DMIC\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	DMIC
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .		
<b>Description</b>	Component (that is, main) interrupt request enable. Write 1 to set (enable interrupt). Readout equal to corresponding _SET register. _SET register is cleared when writing 1 to _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DMIC_IRQ_EMPTY_MASK	DMIC_IRQ_ALST_FULL_MASK	DMIC_IRQ_FULL_MASK	DMIC_IRQ_MASK												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3	DMIC_IRQ_EMPTY_MASK	Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Enable interrupt Read 0x1:	RW W1toSet	0
2	DMIC_IRQ_ALST_FULL_MASK	Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled	RW W1toSet	0
1	DMIC_IRQ_FULL_MASK	Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
0	DMIC_IRQ_MASK	Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled	RW W1toSet	0

**Table 23-534. Register Call Summary for Register DMIC\_IRQENABLE\_SET**

DMIC Functional Description

- [DMIC Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)

DMIC Programming Guide

- [DMIC Polling Mode: \[6\]](#)
- [DMIC Interrupt Mode: \[7\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[8\]](#)

**Table 23-535. DMIC\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	DMIC
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .		
<b>Description</b>	Component (that is, main) interrupt request enable Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register. _SET register is cleared when writing 1 to _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DMIC_IRQ_EMPTY_MASK	DMIC_IRQ_ALST_FULL_MASK	DMIC_IRQ_FULL_MASK	DMIC_IRQ_MASK												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	DMIC_IRQ_EMPTY_MASK	Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled	RW W1toClr	0
2	DMIC_IRQ_ALST_FULL_MASK	Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled	RW W1toClr	0
1	DMIC_IRQ_FULL_MASK	Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
0	DMIC_IRQ_MASK	Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled	RW W1toClr	0

**Table 23-536. Register Call Summary for Register DMIC\_IRQENABLE\_CLR**

DMIC Functional Description

- [DMIC Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[5\]](#)

**Table 23-537. DMIC\_IRQWAKEEN**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .	<b>Instance</b>	DMIC
<b>Description</b>	This register allows to enable the wake-up capability on interrupt event.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															IRQ_WAKEUP_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	IRQ_WAKEUP_EN	Write 1 to allow wake-up by IRQ source (register threshold value reached).	RW	1

**Table 23-538. Register Call Summary for Register DMIC\_IRQWAKEEN**

DMIC Functional Description

- [DMIC Power Management: \[0\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[1\]](#)

**Table 23-539. DMIC\_DMAENABLE\_SET**

<b>Address Offset</b>	0x0000 0038		
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .	<b>Instance</b>	DMIC
<b>Description</b>	Component DMA enable (1 bit per DMA-capable channel)/Write 1 to set (enable DMA). Readout equal to corresponding _SET register. _SET register is cleared when writing 1 to _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DMA_ENABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	DMA_ENABLE	Write 1 to set (enable DMA request) Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Enable interrupt Read 0x1: Interrupt enabled	RW W1toSet	0

**Table 23-540. Register Call Summary for Register DMIC\_DMAENABLE\_SET**

DMIC Functional Description

- [DMIC DMA Requests: \[0\] \[1\]](#)

DMIC Programming Guide

- [DMIC DMA Mode: \[2\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[3\]](#)

**Table 23-541. DMIC\_DMAENABLE\_CLR**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	DMIC
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .		
<b>Description</b>	Component DMA enable (1 bit per DMA-capable channel)/Write 1 to clear (disable DMA). Readout equal to corresponding _SET register. _SET register is cleared when writing 1 to _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												DMA_ENABLE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	DMA_ENABLE	Write 1 to clear (disable DMA request) Read 0x0: Interrupt disabled Write 0x0: No action Write 0x1: Disable interrupt Read 0x1: Interrupt enabled	RW W1toClr	0

**Table 23-542. Register Call Summary for Register DMIC\_DMAENABLE\_CLR**

DMIC Functional Description

- [DMIC DMA Requests: \[0\] \[1\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[2\]](#)

**Table 23-543. DMIC\_DMAWAKEEN**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .	<b>Instance</b>	DMIC
<b>Description</b>	This register allows to enable the wake-up capability on DMA request event.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DMA_WAKEUP_EN															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	DMA_WAKEUP_EN	Write 1 to allow wakeup by DMA source (register threshold value reached).	RW	1

**Table 23-544. Register Call Summary for Register DMIC\_DMAWAKEEN**

DMIC Functional Description

- [DMIC Power Management: \[0\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[1\]](#)

**Table 23-545. DMIC\_CTRL**

<b>Address Offset</b>	0x0000 0044		
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .	<b>Instance</b>	DMIC
<b>Description</b>	This register configures the various parameters of the DMIC module.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SW_DMIC_RST	DMIC_CLK_DIV	DMIC_POLAR3	DMIC_POLAR2	DMIC_POLAR1	DMICOUTFORMAT	DMIC_UP3_EN	DMIC_UP2_EN	DMIC_UP1_EN							

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x000000
10	SW_DMIC_RST	Software reset of the DMIC path. When 1, the DMIC path is reset. Clearing the reset is done by writing 0 to the register.	RW	0
9:7	DMIC_CLK_DIV	Select the DMIC output clock frequency. See <a href="#">Table 23-511</a> for details.	RW	0x0
6	DMIC_POLAR3	0x0: When 0, the left data is generated in the external microphone 3 on <code>abe_dmic_clk3</code> low level. 0x1: When 1, the left data is generated in the external microphone 3 on <code>abe_dmic_clk3</code> high level.	RW	0



Bits	Field Name	Description	Type	Reset
5	DMIC_POLAR2	0x0: When 0, the left data is generated in the external microphone 2 on abe_dmic_clk2 low level. 0x1: When 1, the left data is generated in the external microphone 2 on abe_dmic_clk2 high level.	RW	0
4	DMIC_POLAR1	0x0: When 0, the left data is generated in the external microphone 1 on abe_dmic_clk1 low level. 0x1: When 1, the left data is generated in the external microphone 1 on abe_dmic_clk1 high level.	RW	0
3	DMICOUTFORMAT	When 0, the data going out from the FIFO are left shifted from 8 bits. When 1, the data going are signed extended on 32 bits	RW	0
2	DMIC_UP3_EN	When 1, uplink path 3 is powered up.	RW	0
1	DMIC_UP2_EN	When 1, uplink path 2 is powered up.	RW	0
0	DMIC_UP1_EN	When 1, uplink path 1 is powered up.	RW	0

**Table 23-546. Register Call Summary for Register DMIC\_CTRL**

DMIC Environment
<ul style="list-style-type: none"> <li>• <a href="#">DMIC Clock Generation and Synchronization: [0] [1] [2]</a></li> </ul>
DMIC Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">DMIC Clock Configuration: [3]</a></li> <li>• <a href="#">DMIC Software Reset: [4] [5] [6] [7] [8] [9]</a></li> <li>• <a href="#">Audio Data Processing: [10] [11] [12]</a></li> <li>• <a href="#">Filtering: [13] [14] [15]</a></li> <li>• <a href="#">FIFO Operations: [16] [17] [18]</a></li> <li>• <a href="#">FIFO Output Range Definition: [19]</a></li> </ul>
DMIC Programming Guide
<ul style="list-style-type: none"> <li>• <a href="#">Main Sequence – DMIC Global Initialization: [20] [21] [22] [23]</a></li> </ul>
DMIC Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">DMIC Register Summary: [24]</a></li> </ul>

**Table 23-547. DMIC\_DATA\_REG**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	DMIC																																																																
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .																																																																		
<b>Description</b>	DMIC FIFO data																																																																		
<b>Type</b>	R																																																																		
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">DMIC_DATA</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DMIC_DATA																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
DMIC_DATA																																																																			
<b>Bits</b>	31:0	<b>Field Name</b>	DMIC_DATA																																																																
		<b>Description</b>	DMIC FIFO data																																																																
		<b>Type</b>	R																																																																
		<b>Reset</b>	0x000000																																																																

**Table 23-548. Register Call Summary for Register DMIC\_DATA\_REG**

DMIC Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">FIFO Output Range Definition: [0]</a></li> </ul>
DMIC Programming Guide
<ul style="list-style-type: none"> <li>• <a href="#">DMIC Polling Mode: [1]</a></li> <li>• <a href="#">DMIC Interrupt Servicing: [2]</a></li> </ul>
DMIC Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">DMIC Register Summary: [3]</a></li> </ul>

**Table 23-549. DMIC\_FIFO\_CTRL**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	DMIC
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .		
<b>Description</b>	This register sets the FIFO threshold for the data-ready event.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DMIC_TRESH															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000
3:0	DMIC_TRESH	Uplink FIFO threshold	RW	0x2

**Table 23-550. Register Call Summary for Register DMIC\_FIFO\_CTRL**

DMIC Functional Description

- [FIFO Operations: \[0\] \[1\]](#)

DMIC Programming Guide

- [Main Sequence – DMIC Global Initialization: \[2\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[3\]](#)
- [DMIC Register Descriptions: \[4\]](#)

**Table 23-551. DMIC\_FIFO\_DMIC1R\_DATA**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	DMIC
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .		
<b>Description</b>	Data of the first FIFO DMIC right channel		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIFO_DMIC1R_DAT																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	FIFO_DMIC1R_DAT	Data of the right FIFO DMIC path 1	R	0x000000

**Table 23-552. Register Call Summary for Register DMIC\_FIFO\_DMIC1R\_DATA**

DMIC Functional Description

- [FIFO Output Range Definition: \[0\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[1\]](#)

**Table 23-553. DMIC\_FIFO\_DMIC1L\_DATA**

<b>Address Offset</b>	0x0000 0054
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .
<b>Description</b>	Data of the first FIFO DMIC left channel
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIFO_DMIC1R_DAT																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	FIFO_DMIC1R_DAT	Data of the left FIFO DMIC path 1	R	0x000000

**Table 23-554. Register Call Summary for Register DMIC\_FIFO\_DMIC1L\_DATA**

- DMIC Functional Description
- [FIFO Output Range Definition: \[0\]](#)
- DMIC Register Manual
- [DMIC Register Summary: \[1\]](#)

**Table 23-555. DMIC\_FIFO\_DMIC2R\_DATA**

<b>Address Offset</b>	0x0000 0058
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .
<b>Description</b>	Data of the second FIFO DMIC right channel
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIFO_DMIC2R_DAT																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	FIFO_DMIC2R_DAT	Data of the right FIFO DMIC path 2	R	0x000000

**Table 23-556. Register Call Summary for Register DMIC\_FIFO\_DMIC2R\_DATA**

- DMIC Functional Description
- [FIFO Output Range Definition: \[0\]](#)
- DMIC Register Manual
- [DMIC Register Summary: \[1\]](#)

**Table 23-557. DMIC\_FIFO\_DMIC2L\_DATA**

<b>Address Offset</b>	0x0000 005C
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .
<b>Description</b>	Data of the second FIFO DMIC left channel
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIFO_DMIC2L_DAT																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	FIFO_DMIC2L_DAT	Data of the left FIFO DMIC path 2	R	0x000000

**Table 23-558. Register Call Summary for Register DMIC\_FIFO\_DMIC2L\_DATA**

DMIC Functional Description

- [FIFO Output Range Definition: \[0\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[1\]](#)

**Table 23-559. DMIC\_FIFO\_DMIC3R\_DATA**

<b>Address Offset</b>	0x0000 0060	<b>Instance</b>	DMIC
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .		
<b>Description</b>	Data of the third FIFO DMIC right channel		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIFO_DMIC3R_DAT																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	FIFO_DMIC3R_DAT	Data of the right FIFO DMIC path 3	R	0x000000

**Table 23-560. Register Call Summary for Register DMIC\_FIFO\_DMIC3R\_DATA**

DMIC Functional Description

- [FIFO Output Range Definition: \[0\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[1\]](#)

**Table 23-561. DMIC\_FIFO\_DMIC3L\_DATA**

<b>Address Offset</b>	0x0000 0064	<b>Instance</b>	DMIC
<b>Physical Address</b>	See <a href="#">Table 23-524</a> .		
<b>Description</b>	Data of the third FIFO DMIC left channel		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIFO_DMIC3R_DAT																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:0	FIFO_DMIC3R_DAT	Data of the left FIFO DMIC path 3	R	0x000000

**Table 23-562. Register Call Summary for Register DMIC\_FIFO\_DMIC3L\_DATA**

DMIC Functional Description

- [FIFO Output Range Definition: \[0\]](#)

DMIC Register Manual

- [DMIC Register Summary: \[1\]](#)

PRELIMINARY

## 23.8 Multichannel Audio Serial Port

This section describes the multichannel audio serial port (McASP).

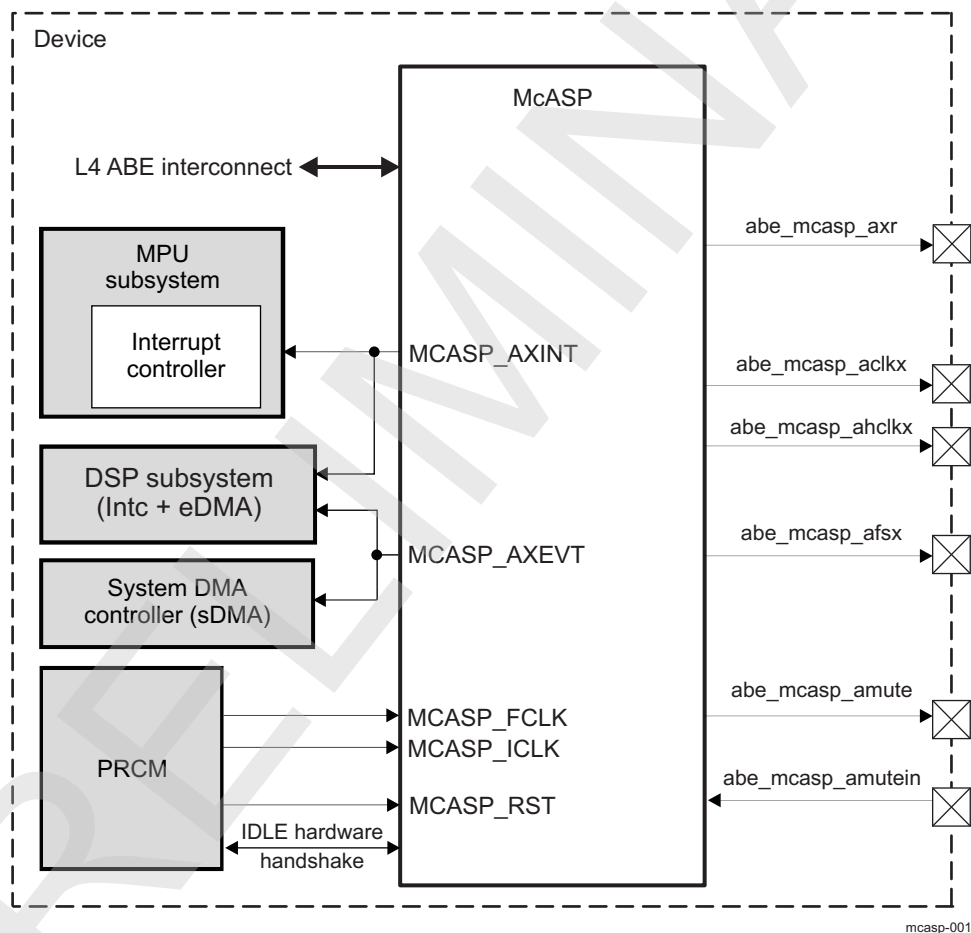
### 23.8.1 McASP Overview

This section introduces the multichannel audio serial port (McASP) module and describes its main functions and connections in the device.

The McASP functions as a general-purpose audio serial port optimized to the requirements of various audio applications. The McASP is useful for intercomponent digital audio interface transmission (DIT). The McASP has the flexibility to gluelessly connect to a Sony/Philips digital interface (S/PDIF) transmit physical layer component. The McASP module operates in transmit mode only; it has no receive capabilities.

Figure 23-185 shows the McASP module in the device.

**Figure 23-185. McASP Module**



The McASP module includes the following main features:

- Support of the idle request/acknowledge protocol
- Buffer for transmit operations
- One transmit direct memory access (DMA) request linked with a 32-bit register, and one transmit interrupt request
- One transmit channel
- One serializer

## 23.8.2 McASP Environment

This section describes the McASP application fields from an environment point of view (external connections), along with the McASP connectivity options. This section also lists all of the possible interfaces and describes the protocol and data format used in each case.

### 23.8.2.1 McASP Signals

Table 23-563 describes the McASP signals and specifies their links to functions.

**Table 23-563. I/O**

Signal	I/O <sup>(1)</sup>	Description
abe_mcasp_axr	O	Audio transmit data
abe_mcasp_aclkx	O	Transmit bit clock
abe_mcasp_ahclkx	O	Transmit high-frequency master clock
abe_mcasp_afsx	O	Transmit frame synchronization
abe_mcasp_amute	O	Mute out to external
abe_mcasp_amutein	I	Mute in from external

<sup>(1)</sup> I = Input; O = Output

All McASP pins on the device can be individually programmed as general-purpose input/output (GPIO) if they are not used for serial port functions.

The McASP includes the following pins:

- Serializer:
  - Data pin abe\_mcasp\_axr
- Transmit clock generator:
  - abe\_mcasp\_aclkx: McASP transmit bit clock
  - abe\_mcasp\_ahclkx: McASP transmit high-frequency master clock
- Transmit frame-sync generator:
  - abe\_mcasp\_afsx: McASP transmit frame sync
- Mute in/out:
  - abe\_mcasp\_amutein: McASP mute input (from external device)
  - abe\_mcasp\_amute: McASP mute output

### 23.8.2.2 Protocols and Data Formats

#### 23.8.2.2.1 Protocols Supported

The McASP transmit section uses the S/PDIF industry format, which is supported by the DIT transfer mode of the module.

Each transmit section can be programmed to support the following options on the basic serial protocol:

- Programmable clock and frame-sync polarity (rising or falling edge): abe\_mcasp\_aclkx, abe\_mcasp\_ahclkx, and abe\_mcasp\_afsx
- Slot length (number of bits per time slot): 8, 12, 16, 20, 24, 28, 32 bits supported
- Word length (bits per word): 8, 12, 16, 20, 24, 28, 32 bits; always less than or equal to the time slot length
- First-bit data delay: 0-, 1-, 2-bit clocks
- Left/right alignment of word inside slot
- Bit order: Most-significant bit (MSB) first or least-significant bit (LSB) first
- Bit mask/pad/rotate function
- Automatically aligns data in Q31 or integer format
- Automatically masks nonsignificant bits (sets to 0, 1, or extends value of another bit)

The DIT transfer mode includes the following additional transmitter features:

- Transmit-only mode: 384 time slots (subframe) per frame
- Biphase encoded LVCMOS output
- Channel status RAM (384 bits)
- User data RAM (384 bits)
- Separate valid bit (V) for subframes A and B

---

**NOTE:** An external transceiver must be connected to the McASP port in the device to translate the electrical signals delivered by the McASP (1.2 V or 1.8 V LVCMOS levels) to the electrical levels of the S/PDIF standard.

---

### 23.8.2.2.2 Definition of Terms

The serial bit stream transmitted by the McASP is a long sequence of 1s and 0s on an audio transmit pin (`abe_mcasp_axr`). However, the sequence has a hierarchical organization that can be described in terms of frames of data, slots, words, and bits.

A basic synchronous serial interface consists of three important components: clock, frame sync, and data. [Figure 23-186](#) shows two of the three basic components: the clock (`abe_mcasp_aclkx`) and the data (`abe_mcasp_axr`). In operation, the transmitter uses `abe_mcasp_aclkx` as serial clock.

- Bit:

A bit is the smallest entity in the serial data stream. The beginning and end of each bit is marked by an edge of the serial clock. The duration of a bit is a serial clock period. A 1 is represented by a logic high on the `abe_mcasp_axr` pin for the entire duration of the bit. A 0 is represented by a logic low on the `abe_mcasp_axr` pin for the entire duration of the bit.

- Word:

A word is a group of bits that make up the data being transferred between the McASP and the external device. [Figure 23-186](#) shows an 8-bit word.

- Slot:

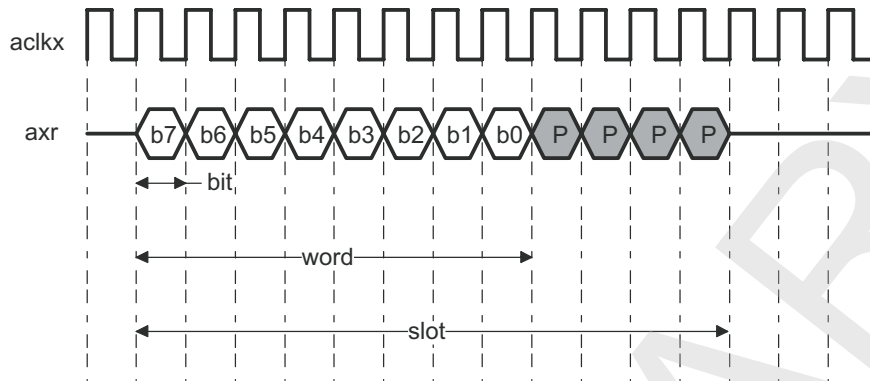
A slot consists of the bits that make up the word and can consist of additional bits used to pad the word to a convenient number of bits for the interface between the McASP and the external device. In [Figure 23-186](#), the audio data consists of only 8 bits of useful data (8-bit word), but it is padded with four 0s (12-bit slot) to satisfy the desired protocol in interfacing to an external device. Within a slot, the bits can be shifted out of the McASP on the `abe_mcasp_axr` pin with either MSB or LSB first.

When the word size is smaller than the slot size, the word can be aligned to the left of the slot (beginning) or to the right of the slot (end). The additional bits in the slot not belonging to the word can be padded with 0, 1, or with one of the bits (typically, the MSB or LSB) from the data word.

[Figure 23-187](#) shows these options.



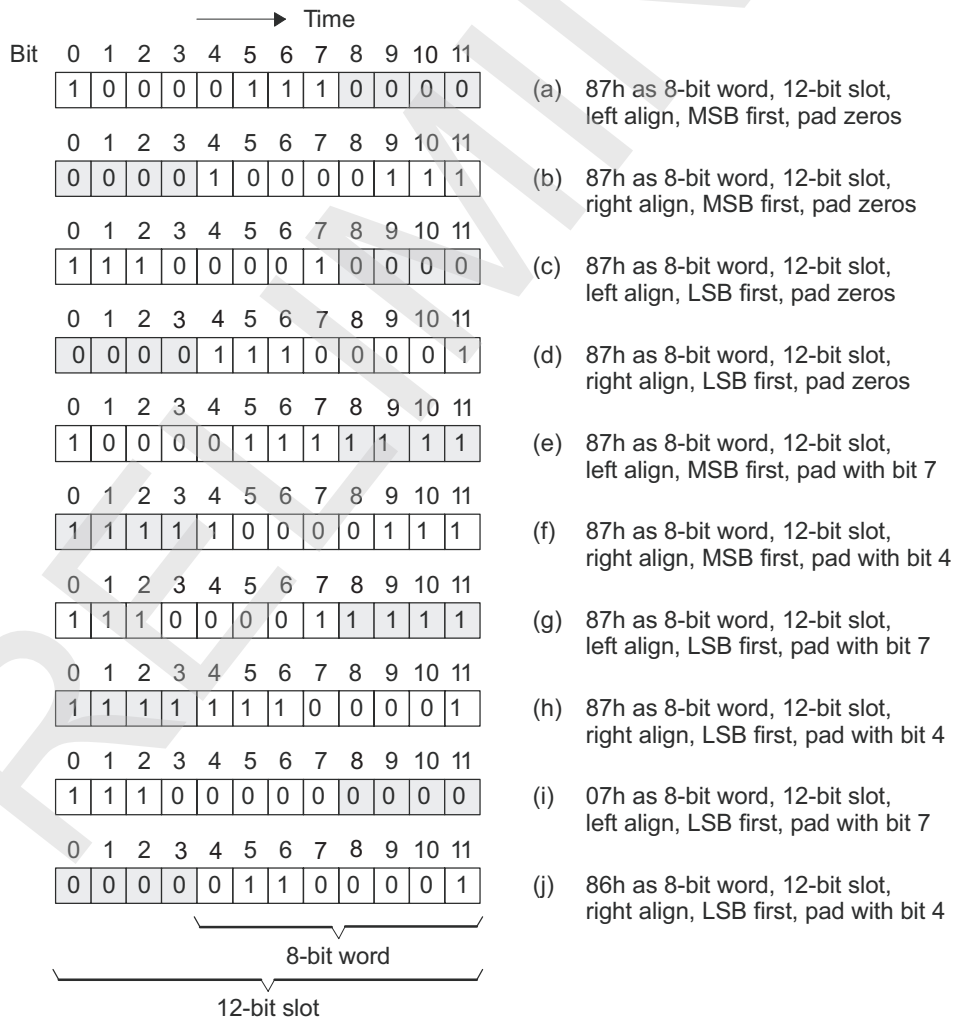
Figure 23-186. Definition of Bit, Word, and Slot



- (1) b7:b0 - bits. Bits b7 to b0 form a word.
- (2) P - pad bits. Bits b7 to b0, together with the 4 pad bits, form a slot.
- (3) In this example, the data is transmitted MSB first, left-aligned.

mccasp-010

Figure 23-187. Bit Order and Word Alignment Within a Slot Examples



- Unshaded: bit belongs to word
- Shaded: bit is a pad bit

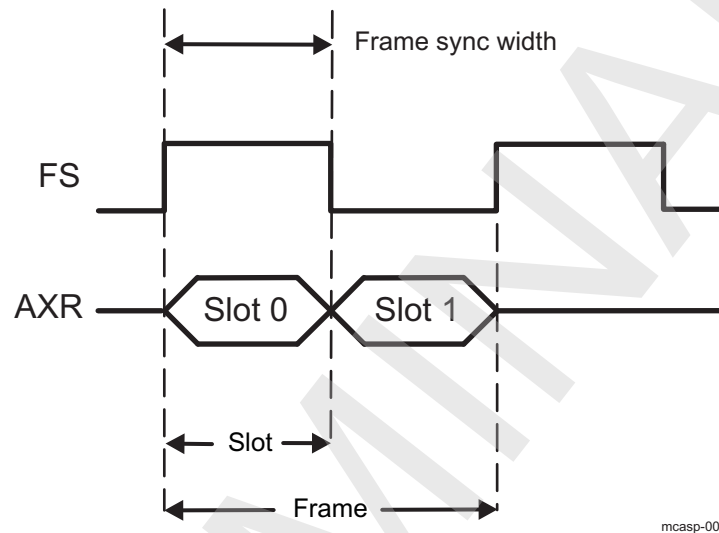
mccasp-008

- Frame

The third basic element of a synchronous serial interface is the frame synchronization signal, also referred to as frame sync in this document. A frame contains one or multiple slots, as determined by the desired protocol. [Figure 23-188](#) shows an example frame of data and the frame definitions. In operation, the transmitter uses `abe_mcaspx_axr`. This example shows two slots in a frame and the frame-sync (FS) duration of the slot length.

This section shows only the generic definition of the frame sync. For more information about the frame-sync formats required for the transfer modes and protocols (DIT mode and S/PDIF format), see [Section 23.8.2.2.3, Frame Format](#).

**Figure 23-188. Definition of Frame and Frame-Sync Width**



The following terms are used throughout this chapter:

- DIT: Digital audio interface transmit. The McASP supports transmitting in S/PDIF format on the `abe_mcaspx_axr` data pin.
- Slot or time slot: For DIT format, a McASP time slot corresponds to a DIT subframe.

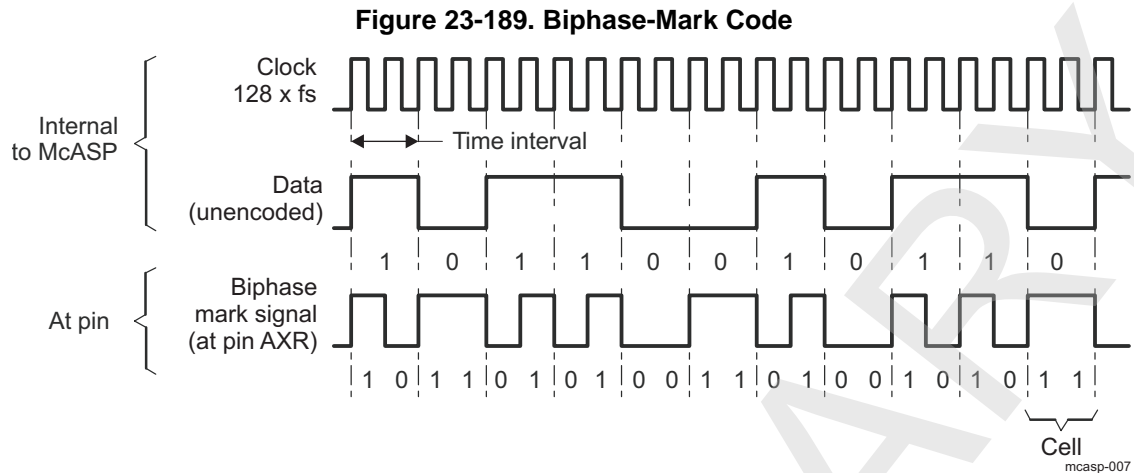
### 23.8.2.2.3 S/PDIF Coding Format

The McASP transmitter supports the S/PDIF format with 3.3-V biphasemark encoded output. The S/PDIF format is supported by the DIT transfer mode of the McASP. This section briefly discusses the S/PDIF coding format.

#### 23.8.2.2.3.1 Biphasemark Code

In S/PDIF format, the digital signal is coded using the biphasemark code (BMC). The clock, frame, and data are embedded in only one signal: data pin `abe_mcaspx_axr`. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. These two logical states form a cell. The duration of the cell, which equals the duration of the data bit, is called a time interval. A logical 1 is represented by two transitions of the signal within a time interval, which corresponds to a cell with logical states 01 or 10. A logical 0 is represented by one transition within a time interval, which corresponds to a cell with logical states 00 or 11. In addition, the logical level at the start of a cell is inverted from the level at the end of the previous cell. [Figure 23-189](#) and [Table 23-564](#) show how data is encoded to the BMC format.

As shown in [Figure 23-189](#), the clock frequency is twice the unencoded data bit rate. In addition, the clock is always programmed to  $128 \times f_s$ , where  $f_s$  is the sample rate (see [Section 23.8.2.2.3.3, Frame Format](#), for details on how this clock rate is derived based on the S/PDIF format).



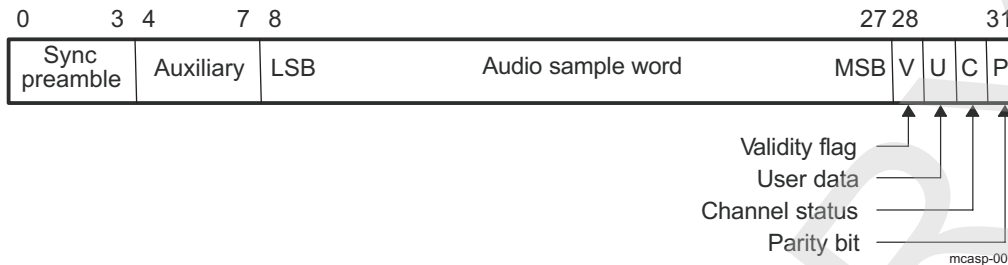
**Table 23-564. Biphase-Mark Encoder**

Data (Unencoded)	Previous State at Pin abe_mcasep_axr	BMC-Encoded Cell Output at abe_mcasep_axr
0	0	11
0	1	00
1	0	10
1	1	01

**23.8.2.2.3.2 Subframe Format**

Every audio sample transmitted in a subframe consists of 32 S/PDIF time intervals (or cells), numbered 0 to 31. [Figure 23-190](#) shows a subframe.

- Time intervals 0–3 carry one of the three permitted preambles to signify the type of audio sample in the current subframe. The preamble is not encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0 or 1 logical states in a row. See [Table 23-565](#).
- Time intervals 4–27 carry the audio sample word in linear 2s-complement representation. The MSB is carried by time interval 27. When a 24-bit coding range is used, the LSB is in time interval 4. When a 20-bit coding range is used, time intervals 8–27 carry the audio sample word with the LSB in time interval 8. Time intervals 4–7 may be used for other applications and are designated auxiliary sample bits.
- If the source provides fewer bits than the interface allows (20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field can carry any other information.
- Time interval 28 carries the validity bit (V) associated with the main data field in the subframe.
- Time interval 29 carries the user data channel (U) associated with the main data field in the subframe.
- Time interval 30 carries the channel status information (C) associated with the main data field in the subframe. The channel status indicates if the data in the subframe is digital audio or some other type of data.
- Time interval 31 carries a parity bit (P) such that time intervals 4–31 carry an even number of 1s and an even number of 0s (even parity). As listed in [Table 23-565](#), the preambles (time intervals 0–3) are also defined with even parity.

**Figure 23-190. S/PDIF Subframe Format**

As listed in [Table 23-565](#), the McASP DIT generates only one polarity of preambles, and it assumes the previous logical state is 0. This is because the McASP assures an even-polarity encoding scheme when transmitting in DIT mode. If an underrun condition occurs, the DIT resynchronizes to the correct logic level on the `abe_mcasp_axr` pin before continuing with the next transmission.

**Table 23-565. Preamble Codes**

Preamble Code <sup>(1)</sup>	Previous Logical State	Logical States on pin <code>abe_mcasp_axr</code> <sup>(2)</sup>	Description
B (or Z)	0	1110 1000	Start of a block and subframe 1
M (or X)	0	1110 0010	Subframe 1
W (or Y)	0	1110 0100	Subframe 2

<sup>(1)</sup> Historically, preamble codes are referred to as B, M, and W. For use in professional applications, preambles are referred to as Z, X, and Y, respectively.

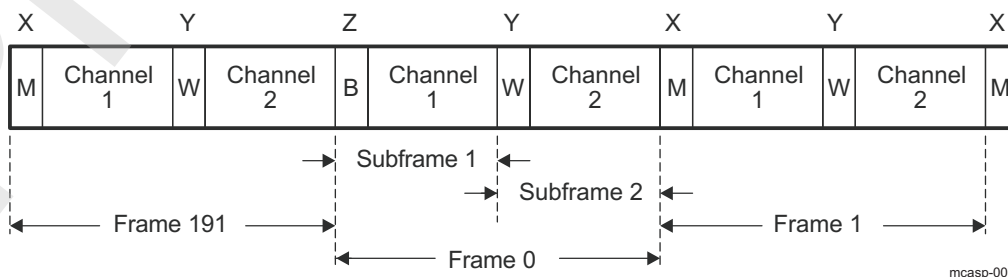
<sup>(2)</sup> The preamble is not BMC-encoded. Each logical state is synchronized to the serial clock. These eight logical states make up time slots (cells) 0 to 3 in the S/PDIF stream.

### 23.8.2.2.3.3 Frame Format

An S/PDIF frame is composed of two subframes (see [Figure 23-191](#)). For linear coded audio applications, the rate of frame transmission normally corresponds exactly to the source sampling frequency  $f_s$ . The S/PDIF format clock rate is therefore  $128 \times f_s$  ( $128 = 32$  cells per subframe  $\times 2$  clocks per cell  $\times 2$  subframes per sample). For example, for an S/PDIF stream at a 192-kHz sampling frequency, the serial clock is  $128 \times 192$  kHz = 24.58 MHz.

In 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive subframes. Both subframes contain valid data. The first subframe (left or A channel in stereophonic operation and primary channel in monophonic operation) normally starts with preamble M. However, the preamble of the first subframe changes to preamble B once every 192 frames to identify the start of the block structure used to organize the channel status information. The second subframe (right or B channel in stereophonic operation and secondary channel in monophonic operation) always starts with preamble W.

In single-channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried in the first subframe and may be duplicated in the second subframe. If the second subframe is not carrying duplicate data, cell 28 (validity bit) is set to logical 1.

**Figure 23-191. S/PDIF Frame Format**

### 23.8.3 McASP Integration

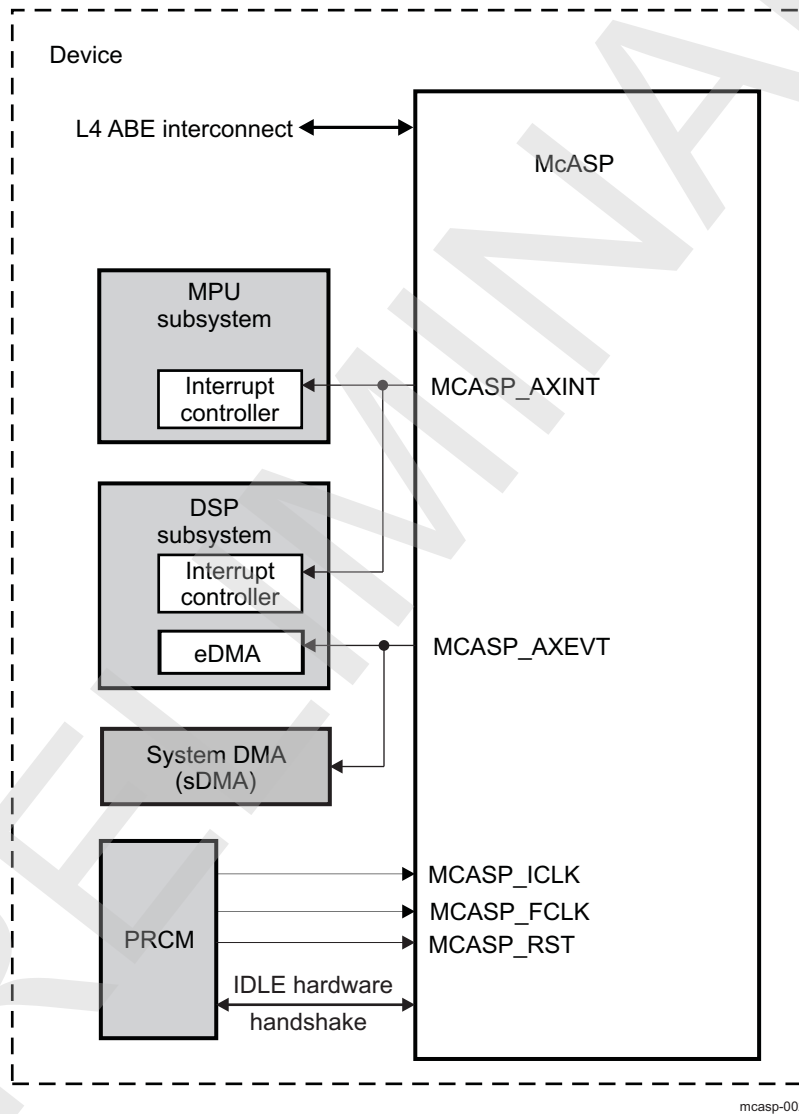
This section describes module integration in the device, including information about clocks, resets, and hardware requests.

The McASP includes the following features:

- IDLE hardware handshake
- One DMA request
- One interrupt request (IRQ)

Figure 23-192 shows McASP integration.

Figure 23-192. McASP Integration



**NOTE:** For more information about the IDLE hardware handshake, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

[Table 23-566](#) through [Table 23-568](#) summarize the integration of the module in the device.

**Table 23-566. Integration Attributes**

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
McASP	PD_AUDIO	NO	L4_ABE

**Table 23-567. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
McASP	MCASP_ICLK	ABE_ICLK2	PRCM	McASP interface clock
	MCASP_FCLK	MCASP1_FCLK	PRCM	McASP functional clock
Resets				
McASP	MCASP_RST	AUDIO_RST	PRCM	AUDIO power domain reset

**Table 23-568. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
McASP	MCASP_AXINT	D_IRQ_92	DSP	McASP transmit interrupt
	MCASP_AXINT	MA_IRQ_109	Cortex™-A9	McASP transmit interrupt
DMA Requests				
McASP	MCASP_AXEVT	S_DMA_7	sDMA	McASP data transmit DMA request line to the system DMA (sDMA)
	MCASP_AXEVT	D_DMA_36	eDMA	McASP data transmit DMA request line to the DSP DMA (eDMA)

**NOTE:**

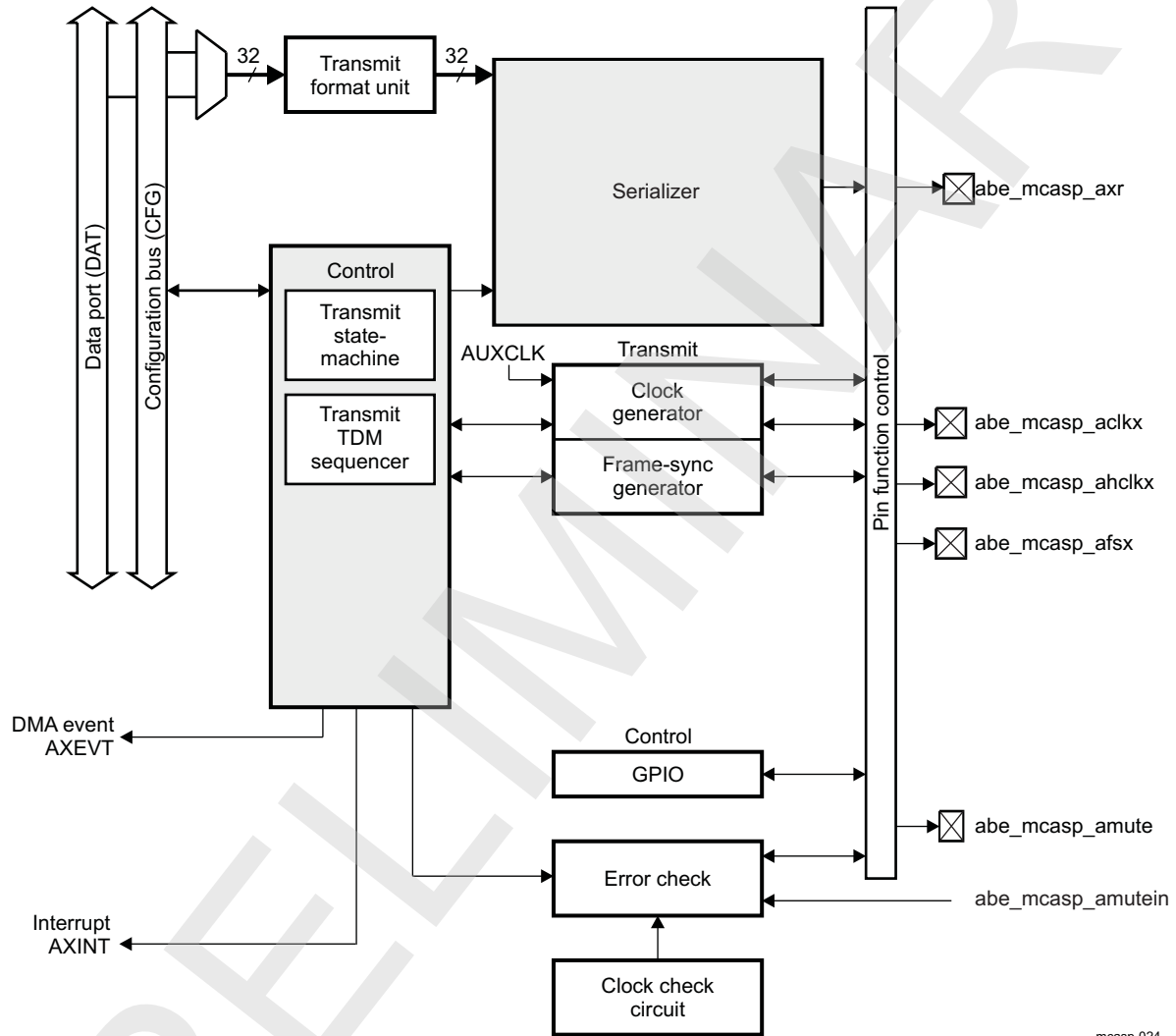
- For the description of the interrupt source, see [Section 23.8.4.11, Interrupt Requests](#).
- For the description of the DMA source, see [Section 23.8.4.12, DMA Requests](#).

### 23.8.4 McASP Functional Description

#### 23.8.4.1 Block Diagram

Figure 23-193 shows the major blocks of the McASP. The McASP has a transmit clock generator and a frame-sync generator, serializer, format unit, and error-checking logic.

Figure 23-193. McASP Block Diagram



mcasp-024

#### 23.8.4.2 McASP Clock and Frame-Sync Configuration

The transmit serial clock (clock at the bit rate) can be sourced internally by passing through two clock dividers off the internal clock source (AUXCLK). The bit rate clock is generated internally and is driven out on the `abe_mcasp_aclkx` pin. An internally generated high-frequency clock can be driven out onto the `abe_mcasp_ahclkx` pin to serve as a reference clock for other components in the system.

##### 23.8.4.2.1 Transmit Clock

The transmit bit clock, `abe_mcasp_aclkx`, is generated internally. For this purpose, the `MCASP_ACLKXCTL[5]` CLKXM bit must be set to 1. Thus, the clock is divided down by a programmable bit clock divider (the `MCASP_ACLKXCTL[4:0]` CLKXDIV bit field) from the source signal.

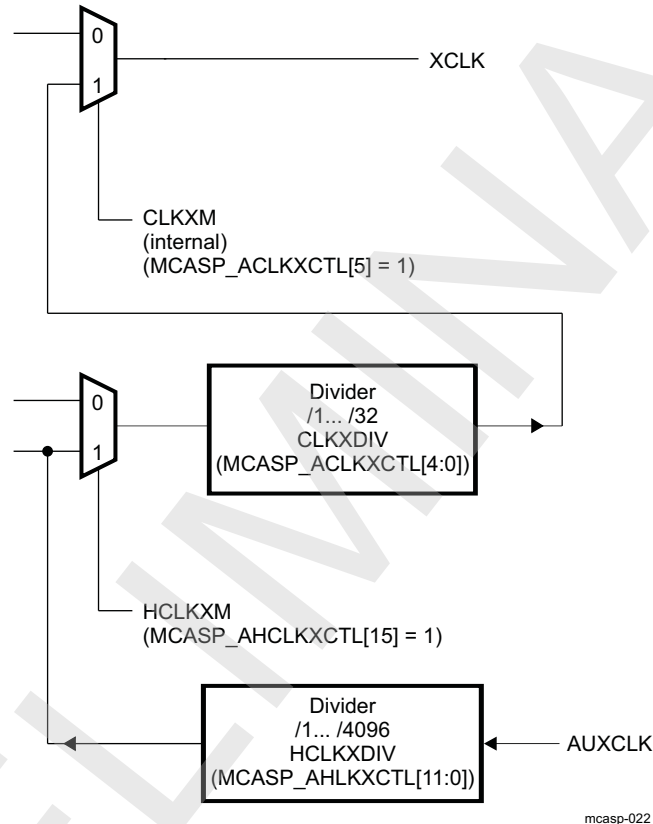
The transmit high-frequency master clock, `abe_mcasp_ahclkx`, is also generated internally. For this purpose, the `MCASP_AHCLKXCTL[15]` HCLKXM bit must be set to 1. Thus, the clock is divided down by a programmable high-clock divider (the `MCASP_AHCLKXCTL[11:0]` HCLKXDIV bit field) from the McASP internal clock source AUXCLK.

The transmit clock configuration is controlled by the following registers:

- `MCASP_ACLKXCTL`
- `MCASP_AHCLKXCTL`

Figure 23-194 is the block diagram of the transmit clock generator.

**Figure 23-194. Transmit Clock Generator Block Diagram**



### 23.8.4.2.2 Frame-Sync Generator

The transmit frame-sync pin is `abe_mcasp_afsx`. The frame-sync generator must be configured through the control register (`MCASP_TXFMCTL`) as:

- Internally generated `MCASP_TXFMCTL[1]` FSXM = 1
- Frame-sync polarity: Rising edge `MCASP_TXFMCTL[0]` FSXP = 0
- Frame-sync width: Single bit `MCASP_TXFMCTL[4]` FXWID = 0
- 384-slot, `MCASP_TXFMCTL[15:7]` XMOD = 1 1000 0000b

### 23.8.4.3 Serializers

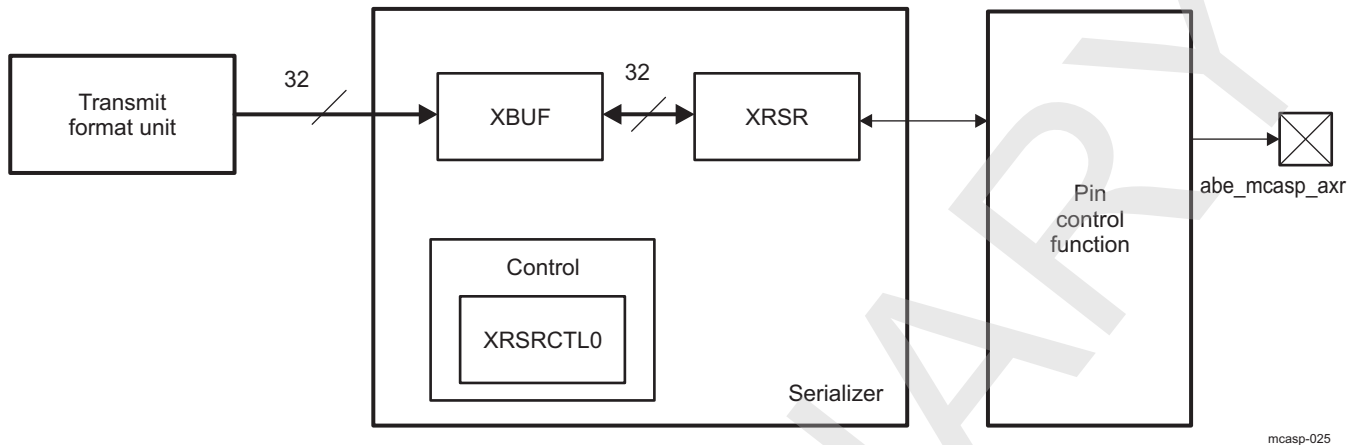
The serializer shifts serial data out of the McASP. It consists of a shift register (XRSR), a data buffer (XBUF), a control register (`MCASP_XRSRCTL0`), a serial data pin (`abe_mcasp_axr`), and logic to support the data alignment options of the McASP.

The serializer must be configured to transmit through the control register, thus shifting out data to `abe_mcasp_axr`. The serializer is clocked from the transmit section clock (`abe_mcasp_aclkx`) if configured to transmit. A serializer configured to transmit operates in lockstep.



Figure 23-195 is the serializer block diagram.

**Figure 23-195. Individual Serializer and Connections Within McASP**



For transmit, the microprocessor unit (MPU) services the McASP by writing data into the XBUF register, which is an alias of the [MCASP\\_TXBUF0](#) register. The data automatically passes through the transmit format unit before reaching the XBUF register in the serializer. The data is then copied from XBUF to XRSR and shifted out from `abe_mcasp_axr` synchronously to the serial clock.

In addition to the data, the serializer shifts out other DIT-specific information accordingly (preamble, user data, etc.). The serializer configuration is controlled by [MCASP\\_XRSRCTL0\[1:0\]](#) SRMOD = 0x1. Thus, the serializer is configured to operate in transmit mode.

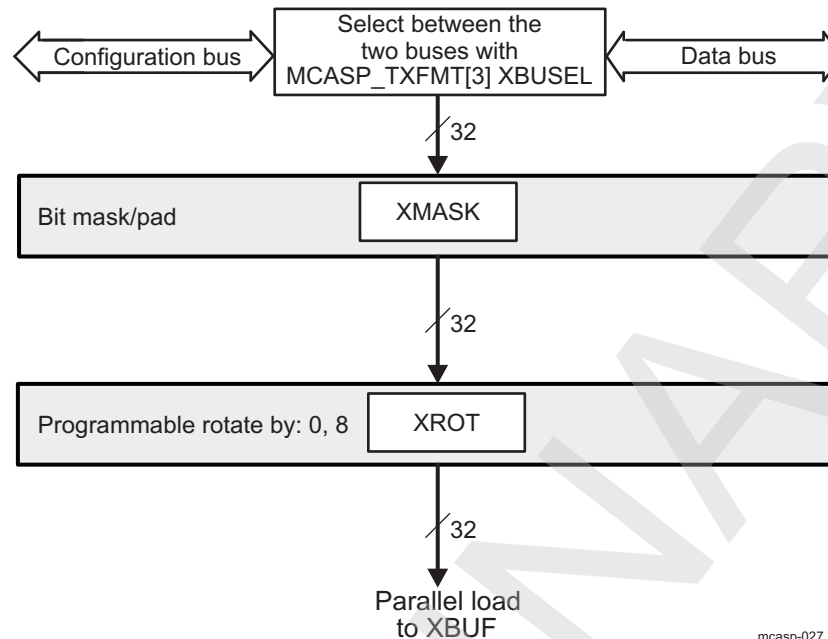
#### 23.8.4.4 Format Unit

The McASP has one transmit data formatting unit. This unit automatically remaps the data bits within the transmitted words between a natural format for the MPU (such as a Q31 representation) and the required format for the external serial device. During the remapping process, the format unit can also mask off certain bits.

This formatting unit consists of two stages:

- Bit mask (masks off bits)
- Rotate right (aligns data within word)

Figure 23-196 shows the transmit formatting unit.

**Figure 23-196. Transmit Format Unit**

As shown in [Figure 23-196](#), the data to the transmit format unit can come from the configuration bus or the data bus. The selection is made through the [MCASP\\_TXFMT\[3\] XBUSEL](#) bit.

The bit mask stage includes a full 32-bit mask register, allowing selected individual bits to pass through the stage unchanged or be masked off.

The rotate-right stage performs bitwise rotation by 0 bits for right-aligned Q31 data and bitwise rotation by 8 bits for left-aligned Q31 data, programmable by the [MCASP\\_TXFMT\[2:0\] XROT](#) bit field. This is a rotation process, not a shifting process, so bit 0 gets shifted back into bit 31 during the process.

The [MCASP\\_TXFMT\[17:16\] XDATLY](#) bit field must be set to a 0-bit delay (0x0 value).

For left-aligned Q31 data, the following transmit format unit settings process the data into right-aligned 24-bit audio data, ready for transmission:

- [MCASP\\_TXFMT\[2:0\] XROT](#) = 0x2 (rotate right by 8 bits)
- [MCASP\\_TXFMT\[15\] XRVRS](#) = 0x0 – Bit reversal is not enabled; the McASP naturally transmits and receives in an LSB-first order.
- [MCASP\\_TXMASK\[32\] XMASK](#) = 0xFFFFFFFF00 – 0xFFFF0000
- [MCASP\\_TXFMT\[14:13\] XPAD](#) = 0x0 (Pad extra bits with 0s.)

For right-aligned data, the following transmit format unit settings process the data into right-aligned 24-bit audio data ready for transmission:

- [MCASP\\_TXFMT\[2:0\] XROT](#) = 0x0 (rotate right by 0 bits)
- [MCASP\\_TXFMT\[15\] XRVRS](#) = 0x0 – Bit reversal is not enabled; the McASP naturally transmits and receives in an LSB-first order.
- [MCASP\\_TXMASK\[32\] XMASK](#) = 0x00FFFFFF – 0x0000FFFF
- [MCASP\\_TXFMT\[14:13\] XPAD](#) = 0x0 (Pad extra bits with 0s.)

[Table 23-589](#) provides more details and specific examples. The examples use internal representation in integer and Q31 notation, but other fractional notations are also possible.

#### 23.8.4.5 Transmit State-Machine

The transmit state-machine controls the interactions between the various units in the transmit section. In addition, the state-machine keeps track of error conditions and serial port status. No serial transfers can occur until the transmit state-machine is released from reset.

The transmit state-machine is controlled by the transmit bit stream format register ([MCASP\\_TXFMT](#)) and it reports the McASP status and error conditions in the transmitter status register ([MCASP\\_TXSTAT](#)).

#### 23.8.4.6 Transmit TDM Sequencer

The time division multiplexing (TDM) sequencer is used to count the 384 subframes (slots) in the DIT block. If currently transmitting slot 1, slot 2 (next value of the TDM slot counter) should be used during the encode phase to select the appropriate C, V, and U bit, because the data encoded and written to the [MCASP\\_TXBUF0](#) register during the current time slot (slot 1) is actually shifted out on the next time slot.

The transmit TDM sequencer is controlled by the [MCASP\\_TXTDM](#) register and reports the current transmit slot to the [MCASP\\_TXTDMSLOT\[9:0\]](#) XSLOT CNT bit field.

#### 23.8.4.7 McASP Software Reset

The McASP can be put into reset through the global transmit control register ([MCASP\\_GBLCTL](#)). A valid serial clock must be supplied to the McASP to assert the software reset bits in the [MCASP\\_GBLCTL](#) register.

#### 23.8.4.8 McASP Power Management

[Table 23-569](#) describes power-management features available to the McASP.

**Table 23-569. Local Power-Management Features**

Feature	Registers	Description
Slave idle modes	<a href="#">MCASP_SYSCONFIG[1:0]</a> IDLE_MODE	Force-idle, no-idle, and smart-idle modes are available.

#### 23.8.4.9 Transfer Modes

##### 23.8.4.9.1 DIT Transfer Mode

The DIT transfer mode of the McASP also supports transmission of audio data in S/PDIF, AES-3, and IEC-60958 formats. These formats are designed to carry audio data between different systems through an optical or coaxial cable. The DIT mode applies only to a serializer configured as transmitter, not receiver. For a description of the S/PDIF format, see [Section 23.8.2.2.3, S/PDIF Coding Format](#).

##### 23.8.4.9.1.1 Transmit DIT Encoding

When the McASP operates in DIT mode, the data transmitted is output as a biphase-mark encoded bit stream, with preamble, channel status, user data, validity, and parity automatically stuffed into the bit stream by the McASP. The McASP includes separate validity bits for even/odd subframes and two 384-bit RAM modules to hold channel status and user data bits.

The transmit TDM time slot register ([MCASP\\_TXTDM](#)) should be programmed to all 1s during DIT mode. TDM functionality is not supported in DIT mode, except that the TDM slot counter counts the DIT subframes.

To transmit data in DIT mode, the following pins are typically required:

- `abe_mcaspl_ahclkx` – transmit high-frequency master clock (The internal clock source can be used instead.)
- One serial data pin, `abe_mcaspl_axr`, with serializer configured to transmit

The actual 24-bit audio data must always be in bit positions 23–0 after passing through the first three stages of the transmit format unit.

##### 23.8.4.9.1.2 Transmit DIT Clock and Frame-Sync Generation

The DIT transmitter works only in the following configuration:

- In the transmit frame control register ([MCASP\\_TXFMCTL](#)):

- Internally generated transmit frame sync, FSXM = 1
- Rising-edge frame sync, FSXP = 0
- Bit-width frame sync, FXWID = 0
- 384-slot TDM, XMOD = 1 1000 0000b
- In the transmit clock control register ([MCASP\\_ACLKXCTL](#)), ASYNC = 1
- In the transmit bit stream format register ([MCASP\\_TXFMT](#)), XSSZ = 1111 (32-bit slot size)

All combinations of `abe_mcasp_ahclkx` and `abe_mcasp_aclkx` are supported.

The following summarizes the register configurations required for DIT mode. DIT mode-specific bit fields are in bold face:

- [MCASP\\_PFUNC](#): The data pin must be configured for McASP function. If `abe_mcasp_ahclkx` is used, it must also be configured for McASP function. Other pins can be configured to function as GPIOs, if desired.
- [MCASP\\_PDIR](#): The data pin must be configured as output. If internal clock source AUXCLK is used as the reference clock, it may be output on the `abe_mcasp_ahclkx` pin by configuring `abe_mcasp_ahclkx` as an output.
- [MCASP\\_GBLCTL](#): Global initialization
- [MCASP\\_AMUTE](#): Program all fields according to the mute control desired.
- [MCASP\\_TXDITCTL](#): The **DITEN** bit must be set to 1 to enable DIT mode. Configure other bits as desired.
- [MCASP\\_TXMASK](#): Mask the desired bits, depending upon left-aligned or right-aligned internal data.
- [MCASP\\_TXFMT](#): **XDATDLY** = 0. **XRVR** = 0. **XPAD** = 0. **XSSZ** = Fh (32-bit slot). **XBUSEL** = configured as desired. The **XROT** bit is configured either 0 or 8-bit rotate.
- [MCASP\\_TXFMCTL](#): Configure the bits according to former discussions.
- [MCASP\\_ACLKXCTL](#): **ASYNC** = 1. Program the CLKXDIV bits to obtain the bit clock rate desired. CLKXM = 1.
- [MCASP\\_AHCLKXCTL](#): Program the HCLKXDIV bits to obtain the high-frequency bit clock rate desired.
- [MCASP\\_TXTDM](#): Set to FFFF FFFFh for all active slots for DIT transfers.
- [MCASP\\_TXEVTCTL](#): Program all fields according to the interrupts desired.
- [MCASP\\_TXCLKCHK](#): Program all fields according to the clock checking desired.
- [MCASP\\_XRSRCTL0](#): Set **SRMOD** = 1 (transmitter) for the DIT pins.
- [MCASP\\_DITCSRAi](#) and [MCASP\\_DITCSRBi](#): Program the channel status bits as desired.
- [MCASP\\_DITUDRAi](#) and [MCASP\\_DITUDRBi](#): Program the user data bits as desired.

#### 23.8.4.9.1.3 DIT Channel Status and User Data Register Files

The channel status registers ([MCASP\\_DITCSRAi](#) and [MCASP\\_DITCSRBi](#)) and user data registers ([MCASP\\_DITUDRAi](#) and [MCASP\\_DITUDRBi](#)) are not double-buffered. Typically, programmers use one of the synchronizing interrupts, such as the last slot, to create an event at a safe time so the register may be updated. In addition, the MPU reads the transmit TDM slot counter to determine which word of the register is being used.

It is a software requirement to avoid writing to the word of user data and channel status that are being used to encode the current time slot; otherwise, it is undetermined whether old or new data is used to encode the bit stream.

The DIT subframe format is defined in [Section 23.8.2.2.3.2, Subframe Format](#). The channel status information (C) and user data (U) are defined in the following DIT control registers:

- [MCASP\\_DITCSRA0](#) to [MCASP\\_DITCSRA5](#): The 192 bits in these six registers contain the channel status information for the left channel within each frame.
- [MCASP\\_DITCSRB0](#) to [MCASP\\_DITCSRB5](#): The 192 bits in these six registers contain the channel status information for the right channel within each frame.
- [MCASP\\_DITUDRA0](#) to [MCASP\\_DITUDRA5](#): The 192 bits in these six registers contain the user data information for the left channel within each frame.

- MCASP\_DITUDRB0 to MCASP\_DITUDRB5: The 192 bits in these six registers contain the user data information for the right channel within each frame.
- The S/PDIF block format is shown in [Figure 23-191](#). There are 192 frames within a block (frame 0 to frame 191). There are two subframes within each frame (subframes 1 and 2 for the left and right channels, respectively).

The channel status and user data information sent on each subframe is summarized in [Table 23-570](#).

**Table 23-570. Channel Status and User Data for Each DIT Block**

Frame	Subframe	Preamble	Channel Status Defined in:	User Data Defined in:
<b>Defined by DITCSRA0, DITCSRB0, DITUDRA0, DITUDRB0</b>				
0	1 (L)	B	DITCSRA0[0]	DITUDRA0[0]
0	2 (R)	W	DITCSRB0[0]	DITUDRB0[0]
1	1 (L)	M	DITCSRA0[1]	DITUDRA0[1]
1	2 (R)	W	DITCSRB0[1]	DITUDRB0[1]
2	1 (L)	M	DITCSRA0[2]	DITUDRA0[2]
2	2 (R)	W	DITCSRB0[2]	DITUDRB0[2]
...	...	...	...	...
31	1 (L)	M	DITCSRA0[31]	DITUDRA0[31]
31	2 (R)	W	DITCSRB0[31]	DITUDRB0[31]
<b>Defined by DITCSRA1, DITCSRB1, DITUDRA1, DITUDRB1</b>				
32	1 (L)	M	DITCSRA1[0]	DITUDRA1[0]
32	2 (R)	W	DITCSRB1[0]	DITUDRB1[0]
...	...	...	...	...
63	1 (L)	M	DITCSRA1[31]	DITUDRA1[31]
63	2 (R)	W	DITCSRB1[31]	DITUDRB1[31]
<b>Defined by DITCSRA2, DITCSRB2, DITUDRA2, DITUDRB2</b>				
64	1 (L)	M	DITCSRA2[0]	DITUDRA2[0]
64	2 (R)	W	DITCSRB2[0]	DITUDRB2[0]
...	...	...	...	...
95	1 (L)	M	DITCSRA2[31]	DITUDRA2[31]
95	2 (R)	W	DITCSRB2[31]	DITUDRB2[31]
<b>Defined by DITCSRA3, DITCSRB3, DITUDRA3, DITUDRB3</b>				
96	1 (L)	M	DITCSRA3[0]	DITUDRA3[0]
96	2 (R)	W	DITCSRB3[0]	DITUDRB3[0]
...	...	...	...	...
127	1 (L)	M	DITCSRA3[31]	DITUDRA3[31]
127	2 (R)	W	DITCSRB3[31]	DITUDRB3[31]
<b>Defined by DITCSRA4, DITCSRB4, DITUDRA4, DITUDRB4</b>				
128	1 (L)	M	DITCSRA4[0]	DITUDRA4[0]
128	2 (R)	W	DITCSRB4[0]	DITUDRB4[0]
...	...	...	...	...
159	1 (L)	M	DITCSRA4[31]	DITUDRA4[31]
159	2 (R)	W	DITCSRB4[31]	DITUDRB4[31]
<b>Defined by DITCSRA5, DITCSRB5, DITUDRA5, DITUDRB5</b>				
160	1 (L)	M	DITCSRA5[0]	DITUDRA5[0]
160	2 (R)	W	DITCSRB5[0]	DITUDRB5[0]
...	...	...	...	...
191	1 (L)	M	DITCSRA5[31]	DITUDRA5[31]
191	2 (R)	W	DITCSRB5[31]	DITUDRB5[31]

### 23.8.4.10 Data Transmission

The MPU services the McASP by writing data to the [MCASP\\_TXBUF0](#) register for transmit operations. The McASP sets status flags and notifies the MPU whenever data is ready to be transmitted. [Section 23.8.4.10.1](#), *Data Ready Status and Event/Interrupt Generation*, discusses data-ready status in detail.

The [MCASP\\_TXBUF0](#) register can be accessed through one of the two peripheral ports of the device:

- Data port (DAT): This port is dedicated to data transfers on the device.
- Configuration bus (CFG): This port is used for data transfers and peripheral configuration control on the device.

[Section 23.8.4.10.1.2](#), *Transfers Through the Data Port (DAT)*, and [Section 23.8.4.10.1.3](#), *Transfers Through the Configuration Bus (CFG)*, discuss how to perform transfers through the data port and the configuration bus, respectively.

The MPU or the DMA can be used to service the McASP through either of these peripheral ports. The MPU and DMA usages are discussed in [Section 23.8.4.10.1.4](#), *Using the MPU for McASP Servicing*, and [Section 23.8.4.10.1.5](#), *Using the DMA for McASP Servicing*, respectively.

#### 23.8.4.10.1 Data Ready Status and Event/Interrupt Generation

##### 23.8.4.10.1.1 Transmit Data Ready

The transmit data ready flag XDATA bit in the [MCASP\\_TXSTAT](#) register reflects the status of the [MCASP\\_TXBUF0](#) register. The XDATA flag is set when data is transferred from the [MCASP\\_TXBUF0\[n\]](#) buffer to the XRSR[n] shift register, thus indicating that the [MCASP\\_TXBUF0](#) is empty and ready to accept new data from the MPU. This flag is cleared when the XDATA bit is written with 1, or when the serializer configured as transmitter is written by the MPU.

Whenever XDATA is set, the DMA AXEVT event is automatically generated to notify the DMA of the [MCASP\\_TXBUF0](#) empty status. An AXINTn interrupt is also generated if the XDATA interrupt is enabled in the [MCASP\\_EVTCTLX](#) register (for details, see [Section 23.8.4.11.1](#), *Transmit Data Ready Interrupt*).

For DMA requests, the McASP does not require that [MCASP\\_TXSTAT](#) be read between DMA events. This means that, even if [MCASP\\_TXSTAT](#) already has the XDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

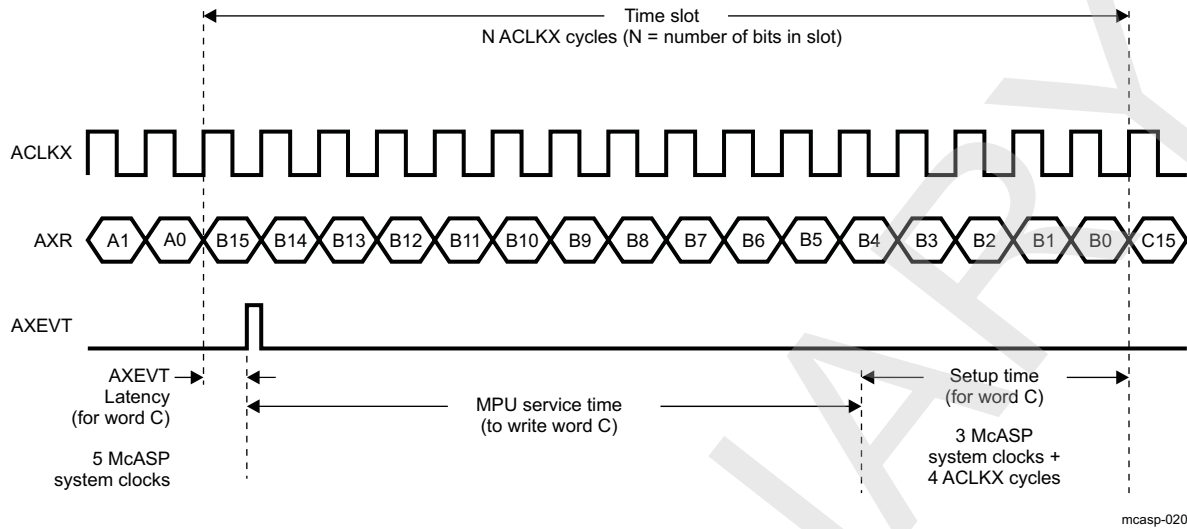
Because the serializer acts in lockstep, only one DMA event is generated to indicate that the transmit serializer is ready to be written to with new data.

[Figure 23-197](#) shows the timing details of when AXEVT is generated at the McASP boundary. In this example, as soon as the last bit (A0) of word A is transmitted, the McASP sets the XDATA flag and generates an AXEVT event. However, it takes up to five McASP system clocks (AXEVT latency) before AXEVT is active at the McASP boundary. Upon AXEVT, the MPU can begin servicing the McASP by writing word C into the [MCASP\\_TXBUF0](#) (service time). The MPU must write word C into the [MCASP\\_TXBUF0](#) within the setup time required by the McASP (setup time).

The maximum service time (see [Figure 23-197](#)) can be calculated as:

$$\text{Service Time} = \text{Time Slot} - \text{AXEVT Latency} - \text{Setup Time}$$

Figure 23-197. MPU Service Time Upon Transmit DMA Event (AXEVT)



23.8.4.10.1.2 Transfers Through the Data Port (DAT)

**CAUTION**

To perform internal transfers through the data port, clear the XBUSEL bit to 0 in the [MCASP\\_TXFMT](#) register. Failure to do so may result in software malfunction.

Typically, the [MCASP\\_TXBUF0](#) register is accessed through the data port. To access through the data port, simply have the MPU or DMA access [MCASP\\_TXBUF0](#) through its data port location.

**NOTE:** DAT allows the MPU, DSP subsystem, or DMA to access only the transmit buffer [MCASP\\_TXBUF0](#) register. There is no way to access the McASP configuration registers within the MPU/DSP subsystem/DMA DAT address ranges.

For transmit operations through the data port, the DMA/MPU must write to the [MCASP\\_TXBUF0](#) data port address to service the active transmit serializer.

When transmitting, the DMA/MPU must write data to the serializer and transmit within each time slot. Failure to do so may result in a buffer underrun condition (see [Section 23.8.4.14.1, Buffer Underrun Error – Transmitter](#)).

23.8.4.10.1.3 Transfers Through the Configuration Bus (CFG)

**CAUTION**

To perform internal transfers through the configuration bus, set the XBUSEL bit to 1 in the [MCASP\\_TXFMT](#) register. Failure to do so may result in software malfunction.

In this method, the DMA/MPU accesses the [MCASP\\_TXBUF0](#) register through the configuration bus address.

When transmitting, the DMA/MPU must write data to the serializer and transmit within each time slot. Failure to do so may result in a buffer underrun condition ([Section 23.8.4.14.1, Buffer Underrun Error – Transmitter](#)).

#### 23.8.4.10.1.4 Using the MPU for McASP Servicing

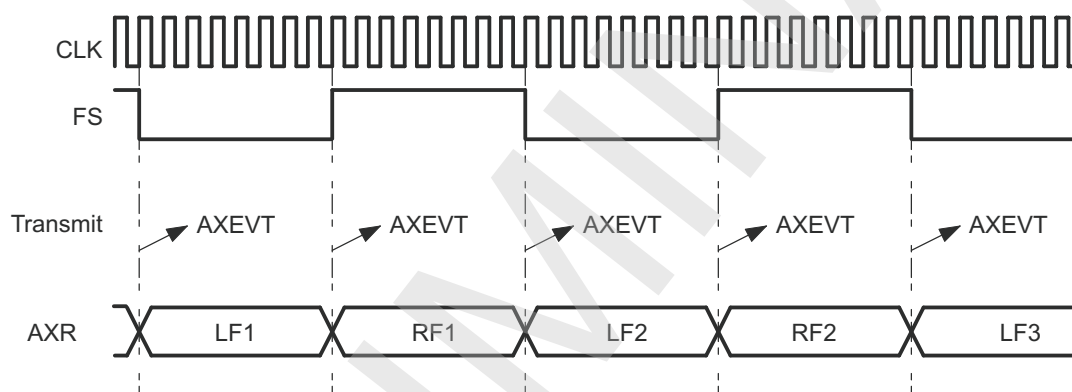
The MPU can be used to service the McASP through interrupts (upon MCASP\_AXINT interrupt) or through polling the XDATA bit in the [MCASP\\_TXSTAT](#) register. As discussed in [Section 23.8.4.10.1.2, Transfers Through the Data Port \(DAT\)](#), and [Section 23.8.4.10.1.3, Transfers Through the Configuration Bus \(CFG\)](#), the MPU can access through the data port or the configuration bus.

To use the MPU to service the McASP through interrupts, the XSTAT bit must be enabled in the [MCASP\\_TXEVTCTL](#) register to generate the MCASP\_AXINT interrupt to the MPU upon data ready.

#### 23.8.4.10.1.5 Using the DMA for McASP Servicing

The typical scenario is to use the DMA to service the McASP through the data port, although the DMA can also service the McASP through the configuration bus.

**Figure 23-198. DMA Transmit Event in an Audio Example – One Event**



mcasep-013

The DMA event AXEVT, which is triggered upon each XDATA transition from 0 to 1, is used to service the McASP transmit buffer (XBUF). [Figure 23-198](#) is an example of an audio system with two audio channels (LF and RF) transmitted through the `abe_mcasp_axr` pin on the McASP. It shows the points at which event AXEVT is triggered.

The DMA AXEVT event is triggered on each time slot. In the example, AXEVT is triggered for each transmit audio channel time slot (time slot for channel LF, and time slot for channel RF). DMA events are generated automatically upon data ready, whereas MPU interrupt generation must be enabled in the [MCASP\\_EVTCTLX](#) register.

#### 23.8.4.11 Interrupt Requests

[Table 23-571](#) lists the event flags that can cause module interrupts.

**Table 23-571. TX Events**

Event Mask	Event Flag	Map to	Description
<a href="#">MCASP_EVTCTLX[0]</a> XUNDRN	<a href="#">MCASP_TXSTAT[0]</a> XUNDRN	MCASP_AXINT	Transmit underrun
<a href="#">MCASP_EVTCTLX[1]</a> XSYNCERR	<a href="#">MCASP_TXSTAT[1]</a> XSYNCERR	MCASP_AXINT	Unexpected transmit frame sync
<a href="#">MCASP_EVTCTLX[2]</a> XCKFAIL	<a href="#">MCASP_TXSTAT[2]</a> XCKFAIL	MCASP_AXINT	Transmit clock failure
<a href="#">MCASP_EVTCTLX[4]</a> XLAST	<a href="#">MCASP_TXSTAT[4]</a> XLAST	MCASP_AXINT	Transmit last slot interrupt
<a href="#">MCASP_EVTCTLX[5]</a> XDATA	<a href="#">MCASP_TXSTAT[5]</a> XDATA	MCASP_AXINT	Transmit data-ready interrupt
<a href="#">MCASP_EVTCTLX[7]</a> XSTAFRM	<a href="#">MCASP_TXSTAT[7]</a> XSTAFRM	MCASP_AXINT	Transmit start of frame interrupt



An interrupt line is asserted (active high) when one of these events occurs. Read the [MCASP\\_TXSTAT](#) register to determine which event occurs.

#### **23.8.4.11.1 Transmit Data Ready Interrupt**

The transmit data-ready interrupt (XDATA) is generated if XDATA is 1 in the [MCASP\\_TXSTAT](#) register and XDATA is enabled in [MCASP\\_EVTCTLX](#). [Section 23.8.4.10.1, Data Ready Status and Event/Interrupt Generation](#), provides details on when XDATA is set in the [MCASP\\_TXSTAT](#) register.

A transmit-start-of-frame interrupt (XSTAFRM) is triggered by the recognition of a transmit frame sync.

A transmit-last-slot interrupt (XLAST) is a qualified version of the data-ready interrupt (XDATA). It has the same behavior as the data-ready interrupt, but is further qualified by having the data requested belonging to the last slot (the slot that just ended is the next-to-last TDM slot, the current slot is the last slot).

#### **23.8.4.11.2 Error Interrupt**

Upon detection, the following error conditions generate interrupt flags:

In the transmit status register ([MCASP\\_TXSTAT](#)):

- Transmit underrun (XUNDRN)
- Transmit DMA error (XDMAERR)

Each interrupt source also has a corresponding enable bit in the transmit interrupt control register ([MCASP\\_EVTCTLX](#)). If the enable bit is set, an interrupt is requested when the interrupt flag is set in [MCASP\\_TXSTAT](#). If the enable bit is not set, no interrupt request is generated. However, the interrupt flag may be polled.

#### **23.8.4.12 DMA Requests**

The McASP can generate one DMA request to the sDMA and eDMA controllers to transmit data ([MCASP\\_AXEVT](#)).

A DMA request to transmit data is generated if the XDATDMA bit in the [MCASP\\_TXEVTCTL](#) register is cleared.

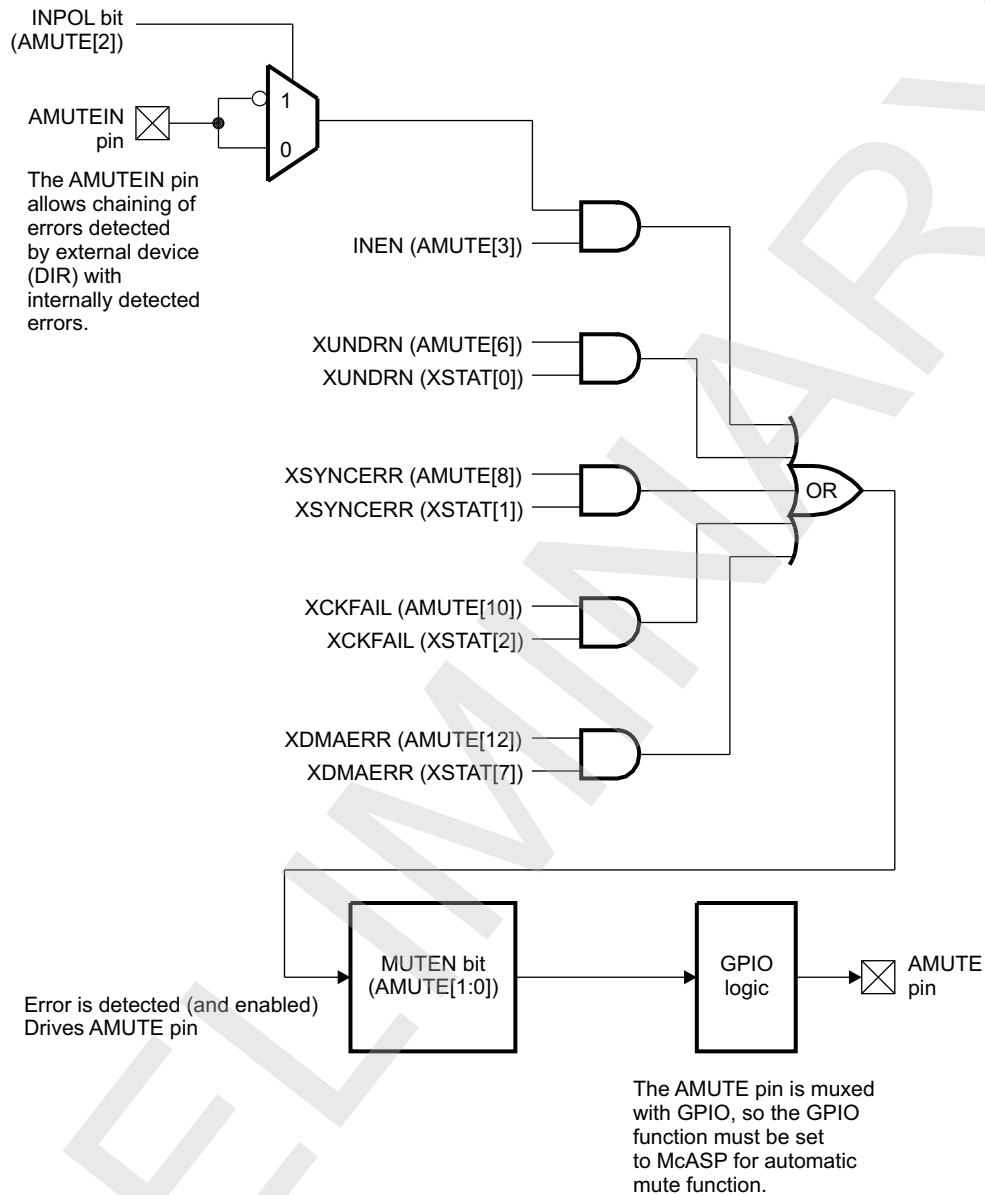
#### **23.8.4.13 Audio Mute (AMUTE) Function**

The McASP includes an automatic audio mute function (see [Figure 23-199](#)) that asserts, in hardware, the AMUTE pin to a preprogrammed output state, as selected by the MUTEN bit in the audio mute control register ([MCASP\\_AMUTE](#)). The AMUTE pin is asserted when one of the interrupt flags is set or an external device issues an error signal on the `abe_mcasp_amutein` input. Typically, the `abe_mcasp_amutein` input is shared with a device interrupt pin.

The `abe_mcasp_amutein` input allows the on-chip logic to consider a mute input from other devices in the system so that all errors may be considered. The `abe_mcasp_amutein` input has a programmable polarity to allow it to adapt to different devices, as selected by the INPOL bit in the [MCASP\\_AMUTE](#) register, and it must be explicitly enabled.

In addition to the external `abe_mcasp_amutein` input, the `abe_mcasp_amute` pin output may be asserted when one of the error interrupt flags is set and its mute function is enabled in the [MCASP\\_AMUTE](#) register.

When one or more errors are detected and enabled, the `abe_mcasp_amute` pin is driven to an active state that is selected by MUTEN in the [MCASP\\_AMUTE](#) register. The active polarity of the AMUTE pin is programmable by the MUTEN bit (the inactive polarity is the opposite of the active polarity). The `abe_mcasp_amute` pin remains driven active until software clears all the error interrupt flags that are enabled to mute, and until the `abe_mcasp_amutein` is inactive.

**Figure 23-199. Audio Mute (AMUTE) Block Diagram**

#### 23.8.4.14 Error Reporting

The McASP includes error-checking capability for the serial protocol and data underrun. In addition, the McASP includes a timer that continually measures the high-frequency master clock every 32 AHCLKX clock cycles. The value of the timer can be read to get a measurement of the clock frequency and has a minimum and maximum range setting that can set an error flag if the master clock goes out of a specified range.

When one or more errors (software selectable) are detected or the `abe_mccasp_amutein` input pin is asserted, the `abe_mccasp_amute` output pin may be asserted to a high or low level to immediately mute the audio output. In addition, an interrupt can be generated if desired, based on one or more error sources.

#### **23.8.4.14.1 Buffer Underrun Error –Transmitter**

A buffer underrun occurs when the serializer is instructed by the transmit state-machine to transfer data from `MCASP_TXBUF0` to XRSR, but `MCASP_TXBUF0` has not yet been written with new data since the last time the transfer occurred. When this occurs, the transmit state-machine sets the XUNDRN flag.

An underrun is checked only once per time slot. The `MCASP_TXSTAT[0]` XUNDRN flag is set when an underrun condition occurs. Once set, the XUNDRN flag remains set until the host explicitly writes 1 to the XUNDRN bit to clear it.

A pair of BMC zeros is shifted out when an underrun occurs (four bit times at 128 bps). By shifting out a pair of zeros, a clock can be recovered on the receiver. To recover, reset the McASP and restart with the proper initialization.

#### **23.8.4.14.2 DMA Error –Transmitter**

A transmit DMA error, as indicated by the XDMAERR flag in the `MCASP_TXSTAT` register, occurs when the DMA (or MPU) writes more words to the data port of the McASP than it should.

XDMAERR indicates that the DMA (or MPU) wrote too many words to the McASP for a given transmit DMA event. Writing too few words results in a transmit underrun error setting XUNDRN in `MCASP_TXSTAT`.

While XDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or MPU. The McASP transmitter and the DMA must be reinitialized to resynchronize them.

#### **23.8.4.14.3 Clock Failure Detection**

##### **23.8.4.14.3.1 Clock Failure Check Startup**

It is initially expected of the clock failure circuits to generate an error until at least one measurement is taken. Therefore, the clock failure interrupts, clock switch, and mute functions should not be enabled immediately, but only after a specific startup procedure.

To start the transmit clock failure check procedure:

1. Configure the transmit clock failure detect logic (XMIN, XMAX, XPS) in the transmit clock check control register (`MCASP_TXCLKCHK`).
2. Clear the transmit clock failure flag (XCKFAIL) in the transmit status register (`MCASP_TXSTAT`).
3. Wait until the first measurement is taken (> 32 AHCLKX clock periods).
4. Verify that no clock failure is detected.
5. Repeat Step 2 through Step 4 until the clock is running and is no longer issuing clock failure errors.
6. After the transmit clock is measured and falls within the acceptable range, the following can be enabled:
  - (a) The transmit clock failure interrupt enable bit (XCKFAIL) in the transmitter interrupt control register (`MCASP_EVTCTLX`)
  - (b) The mute option (XCKFAIL) in the mute control register (`MCASP_AMUTE`)

##### **23.8.4.14.3.2 Transmit Clock Failure Check and Recovery**

The transmit clock failure check circuit (see [Figure 23-200](#)) works off the internal McASP system clock and the external high-frequency serial clock (`abe_mcasp_ahclkx`). It continually counts the number of system clocks for every 32 high-rate serial clock (`abe_mcasp_ahclkx`) periods, and stores the count in XCNT of the transmit clock check control register (`MCASP_TXCLKCHK`) every 32 high-rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (XMIN), and automatically flags an interrupt (XCKFAIL in `MCASP_TXSTAT`) when an out-of-range condition occurs. An

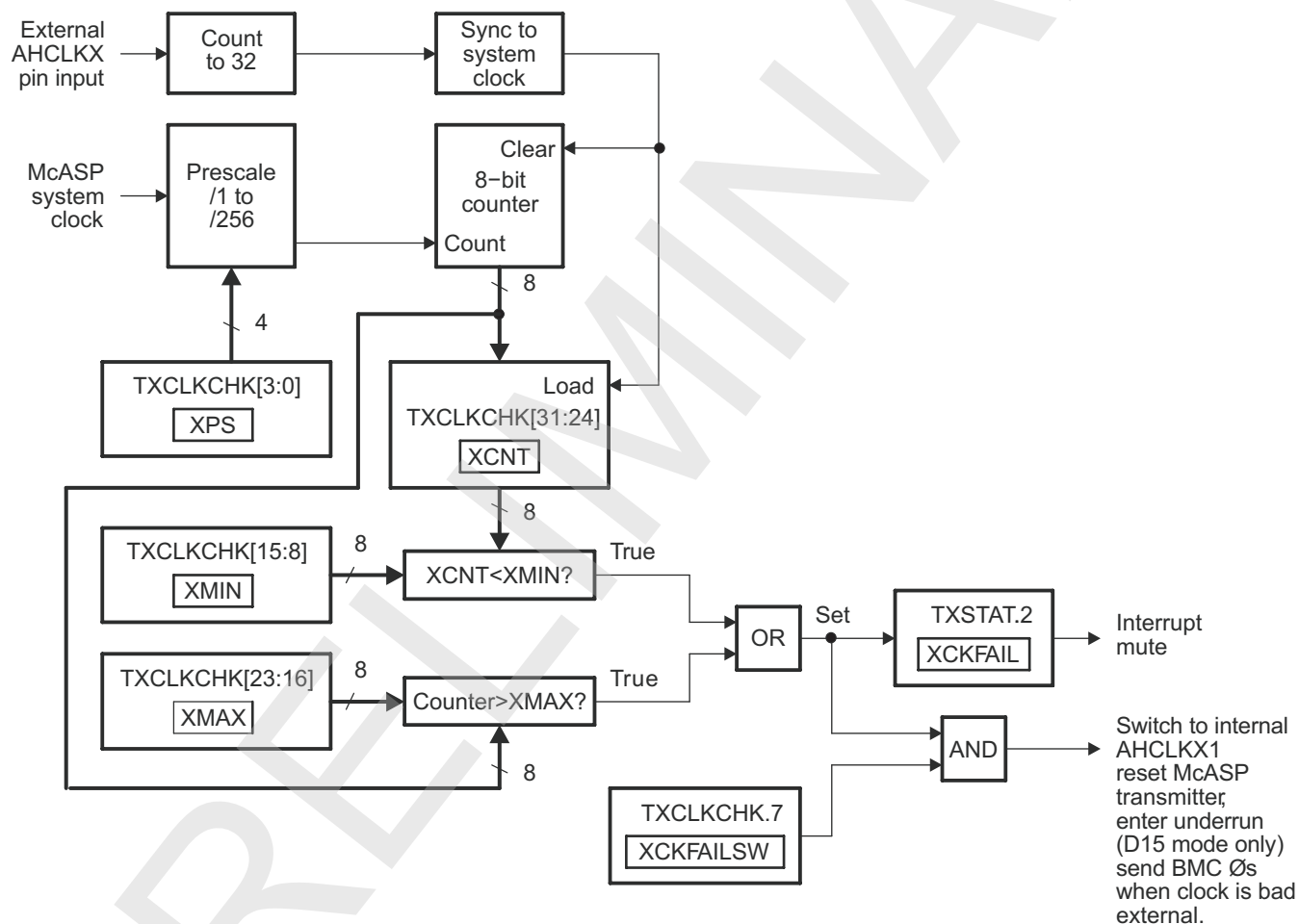
out-of-range minimum condition occurs when the count is less than XMIN. The logic continually compares the current count (from the running system clock counter) to the maximum allowable boundary (XMAX). This is so that if the external clock completely stops, the counter value is not copied to XCNT. An out-of-range maximum condition occurs when the count is greater than XMAX. The XMIN and XMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate that an unstable clock was detected or that the audio source has changed and a new sample rate is being used.

For the transmit clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset.

If a clock failure is detected, the transmit clock failure flag (XCKFAIL) in `MCASP_TXSTAT` is set. This causes an interrupt if the transmit clock failure interrupt enable bit (XCKFAIL) in `MCASP_EVTCTLX` is set.

**Figure 23-200. Transmit Clock Failure Detection Circuit Block Diagram**



mcasep-014

## 23.8.5 McASP Low-Level Programming Model

This section describes the low-level hardware programming sequences for the configuration and use of the McASP module.

### 23.8.5.1 Global Initialization

#### 23.8.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the McASP module is used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the McASP (for more information, see the [Section 23.8.3](#), *McASP Integration*, and [Section 23.8.2](#), *McASP Environment*).

[Table 23-572](#), describes the global initialization of surrounding modules.

**Table 23-572. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	Module functional and interface clocks must be enabled. (See <a href="#">Chapter 3</a> , <i>Power, Reset, and Clock Management</i> .)
Control module	Module-specific pad muxing and other pad configurations must be set in the control module. (See <a href="#">Chapter 18</a> , <i>Control Module</i> .)
(Optional) Cortex-A9 MPU INTC or DSP INTC	INTC configuration must be done to enable the interrupts from the McASP. (See <a href="#">Chapter 17</a> , <i>Interrupt Controllers</i> .)
(Optional) sDMA (or eDMA or ADMA)	DMA configuration must be done to enable the McASP DMA data channel requests.
(Optional) ABE_L4 and L3 Interconnects	For more information about the interconnect configuration, see <a href="#">Chapter 13</a> , <i>Interconnect</i> .

**NOTE:** The MPU/DSP INTC and the sDMA/eDMA/ADMA configurations are required when the interrupt and DMA-based communication modes are used.

#### 23.8.5.1.2 McASP Global Initialization

##### 23.8.5.1.2.1 Main Sequence – McASP Global Initialization

The procedure in [Table 23-573](#) initializes the McASP after a power-on reset (POR).

**Table 23-573. McASP Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Apply software reset to different McASP components.	<a href="#">MCASP_GBLCTL</a> [12:8]	0x00
Poll the bits to ensure the active reset value (0x00) is successfully latched into the register.	<a href="#">MCASP_GBLCTL</a> [12:8]	=0x00
Configure the local power management.	<a href="#">MCASP_SYSCONFIG</a> [1:0] IDLE_MODE	0x-
Configure the transmit format unit.	See <a href="#">Section 23.8.5.1.2.2</a> .	
Configure the transmit frame sync generator.	See <a href="#">Section 23.8.5.1.2.3</a> .	
Configure the transmit clock generator.	See <a href="#">Section 23.8.5.1.2.4</a> .	
Configure the TDM sequencer—set all slots active.	<a href="#">MCASP_TXTDM</a> [31:0] XTDMs	0xFFFF FFFF
Configure the serializer for transmit mode operation.	<a href="#">MCASP_XRSRCTL0</a> [1:0] SRMOD	0x1
Configure the McASP pins functionality.	See <a href="#">Section 23.8.5.1.2.5</a> .	
Configure the McASP mute input/output conditions.	See <a href="#">Table 23-578</a> .	
Enable the McASP DIT-transmission mode.	<a href="#">MCASP_TXDITCTL</a> [0] DITEN	0x1
Configure DIT-specific subframe fields.	See <a href="#">Table 23-579</a> .	

**Table 23-573. McASP Global Initialization (continued)**

Step	Register/Bit Field/Programming Model	Value
Release from reset state the divider that outputs the <code>abe_mcasp_ahclkx</code> clock. <sup>(1)</sup>	<code>MCASP_GBLCTL[9]</code> XHCLKRST	0x1
Poll the bit to ensure that it is successfully latched in the register.	<code>MCASP_GBLCTL[9]</code> XHCLKRST	=0x1
Release from reset state the divider that outputs the <code>abe_mcasp_aclkx</code> clock. <sup>(1)</sup>	<code>MCASP_GBLCTL[8]</code> XCLKRST	0x1
Poll the bit to ensure that it is successfully latched in the register.	<code>MCASP_GBLCTL[8]</code> XCLKRST	=0x1

<sup>(1)</sup> During reset state the local McASP internal clock dividers maintain a 1:1 ratio at their outputs. The values stored in the `MCASP_AHCLKX` and `MCASP_ACLKX` registers are ignored; hence, the transmission clock does not stop during the reset state of the dividers.

### 23.8.5.1.2.2 Subsequence – Transmit Format Unit Configuration

The procedure in [Table 23-574](#) configures the transmit frame format unit of the McASP module.

**NOTE:**

- The first transmit data bit always has a 0-bit delay.
- The bit stream is always transmitted in least-significant-bit (LSB)-first order.
- Pad value for extra bits in a certain slot is always 0.

**Table 23-574. Transmit Format Unit Configuration**

Step	Register/Bit Field/Programming Model	Value
Configure the slot size to 32 bits.	<code>MCASP_TXFMT[7:4]</code> XSSZ	0xF
<b>IF:</b> the MPU or DMA data to transmit is left- aligned	Software test condition	
Set data mask in the range 0xFFFF FF00 – 0xFFFF 0000.	<code>MCASP_TXMASK[31:0]</code> XMASK	0x-
Rotate data right by 8-bit positions.	<code>MCASP_TXFMT[2:0]</code> XROT	0x2
<b>ELSE</b>		
Set data mask in the range 0x00FF FFFF– 0x0000 FFFF.	<code>MCASP_TXMASK[31:0]</code> XMASK	0x-
Rotate data right by 0-bit positions.	<code>MCASP_TXFMT[2:0]</code> XROT	0x0
<b>ENDIF</b>		

### 23.8.5.1.2.3 Subsequence – Transmit Frame Synchronization Generator Configuration

The procedure in [Table 23-575](#) configures the transmit frame synchronization generator of the McASP module.

**NOTE:** The frame synchronization signal is always rising-edge active and always has a single-bit width.

**Table 23-575. Transmit Frame-Synchronization Generator Configuration**

Step	Register/Bit Field/Programming Model	Value
Select 384-slot size block.	<code>MCASP_TXFMCTL[15:7]</code> XMOD	0x180
Select internally-generated transmit frame sync.	<code>MCASP_TXFMCTL[1]</code> FSXM	0x1

### 23.8.5.1.2.4 Subsequence – Transmit Clock Generator Configuration

**NOTE:** By default, the ACLKX and AHCLKX clocks are generated only from the McASP internal clock source.

The procedure in [Table 23-576](#) configures the transmit clock generator of the McASP module.

**Table 23-576. Transmit Clock Generator Configuration**

Step	Register/Bit Field/Programming Model	Value
Set the divisor for the internally generated high frequency clock– <code>abe_mcasp_ahclkx</code> .	<code>MCASP_AHCLKXCTL[11:0]</code> HCLKXDIV	0x-
Set the divisor for the internally generated transmission clock– <code>abe_mcasp_aclkx</code> .	<code>MCASP_ACLKXCTL[4:0]</code> CLKXDIV	0x-
Configure the transmit clock failure detect logic.	See <a href="#">Section 23.8.4.14.3.1</a> , <i>Clock Failure Check Startup</i> .	

### 23.8.5.1.2.5 Subsequence—McASP Pins Functional Configuration

The procedure in [Table 23-577](#) configures the McASP pins for McASP functionality.

**Table 23-577. McASP Pins Functional Configuration**

Step	Register/Bit Field/Programming Model	Value
Configure module different pins to have McASP functionality.	<code>MCASP_PFUNC[31:0]</code>	0x0
Configure the McASP pins as outputs:	<code>MCASP_PDIR[28]</code> AFSX;	0x1
<code>abe_mcasp_afsx</code>	<code>MCASP_PDIR[27]</code> AHCLKX;	0x1
<code>abe_mcasp_ahclkx</code>	<code>MCASP_PDIR[26]</code> ACLKX;	0x1
<code>abe_mcasp_aclkx</code>	<code>MCASP_PDIR[25]</code> AMUTE;	0x1
<code>abe_mcasp_amute</code>	<code>MCASP_PDIR[0]</code> AXR0;	0x1
<code>abe_mcasp_axr</code>		

### 23.8.5.1.2.6 Subsequence – McASP Mute Input/Output Trigger Condition Settings

The procedure in [Table 23-578](#) configures the behavior of the mute associated I/Os and different trigger conditions for the McASP mute output functionality.

**Table 23-578. McASP Mute Input/Output Trigger Condition Settings**

Step	Register/Bit Field/Programming Model	Value
Optional: Select to drive AMUTE active enable bit on transmit DMA error.	<code>MCASP_AMUTE[12]</code> XDMAERR	0x1
Optional: Select to drive AMUTE active enable bit on transmit clock failure.	<code>MCASP_AMUTE[10]</code> XCKFAIL	0x1
Optional: Select to drive AMUTE active enable bit on transmit frame synchronization error.	<code>MCASP_AMUTE[8]</code> XSYNCERR	0x1
Optional: Select to drive AMUTE active enable bit on transmit underrun error.	<code>MCASP_AMUTE[6]</code> XUNDRN	0x1
Optional: Enable sensitivity of the <code>abe_mcasp_amute</code> output to <code>abe_mcasp_amutein</code> error source.	<code>MCASP_AMUTE[3]</code> INEN	0x1
Optional: Select the <code>abe_mcasp_amutein</code> input polarity.	<code>MCASP_AMUTE[2]</code> INPOL	0x1
Optional: Configure the behavior of the <code>abe_mcasp_amute</code> output.	<code>MCASP_AMUTE[1:0]</code> MUTEN	0x1

### 23.8.5.1.2.7 Subsequence – DIT-Specific Subframe Fields Configuration

**NOTE:** The McASP integrated in the device supports only the DIT transmission mode that uses the S/PDIF protocol.

The procedure in [Table 23-579](#) configures the DIT-specific subframe fields as part of the S/PDIF format data.

**Table 23-579. DIT-Specific Subframe Fields Configuration**

Step	Register/Bit Field/Programming Model	Value
Configure the valid bit value for odd time slots.	MCASP_TXDITCTL[3] VB	0x-
Configure the valid bit value for even time slots.	MCASP_TXDITCTL[2] VA	0x-
Configure the user data bit for each subframe A and B in a 384-slot S/PDIF block.	MCASP_DITUDRAi[31:0] DITUDRAi, where i = 0 to 5	0x-
	MCASP_DITUDRBi[31:0] DITUDRBi, where i = 0 to 5	0x-
Configure the channel status bit for each subframe A and B in a 384-slot S/PDIF block.	MCASP_DITCSRAi[31:0], where i = 0 to 5	0x-
	MCASP_DITCSRBi[31:0], where i = 0 to 5	0x-

## 23.8.5.2 Operational Modes Configuration

### 23.8.5.2.1 McASP DIT Transmission Mode

#### 23.8.5.2.1.1 Main Sequence – McASP DIT Polling Transmission Method

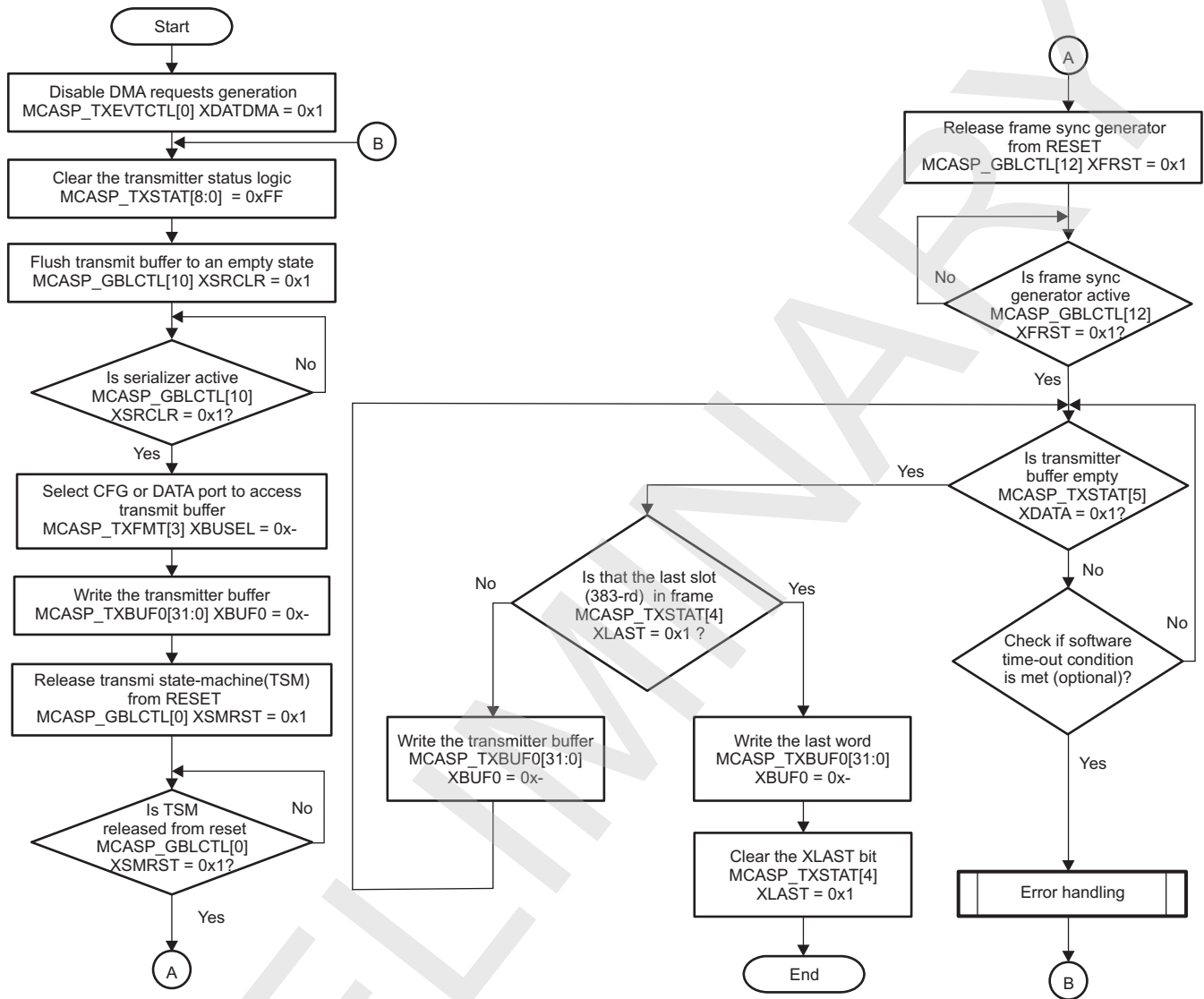
[Figure 23-201](#) shows the McASP DIT polling method.

**NOTE:**

- The McASP polling transmission model considers the MPU/DSP as the source of audio data for the McASP transmission buffer.
- The DMA request is disabled and the XDMAERR event is not analyzed.
- The flow chart shown in [Figure 23-201](#) shows the sequence for transmitting one DIT frame.



Figure 23-201. McASP DIT Transmission Polling Method



mcasp-028

Table 23-580 summarizes the register call for the polling mode.

Table 23-580. Register Call Summary for Main Sequence – McASP DIT Transmission Polling Method

Register Name	Register Name	Register Name	Register Name	Register Name
MCASP_TXSTAT	MCASP_GBLCTL	MCASP_TXBUF0	MCASP_TXFMT	MCASP_TXEVTCTL

Table 23-581 summarizes the subprocess call for the polling mode.

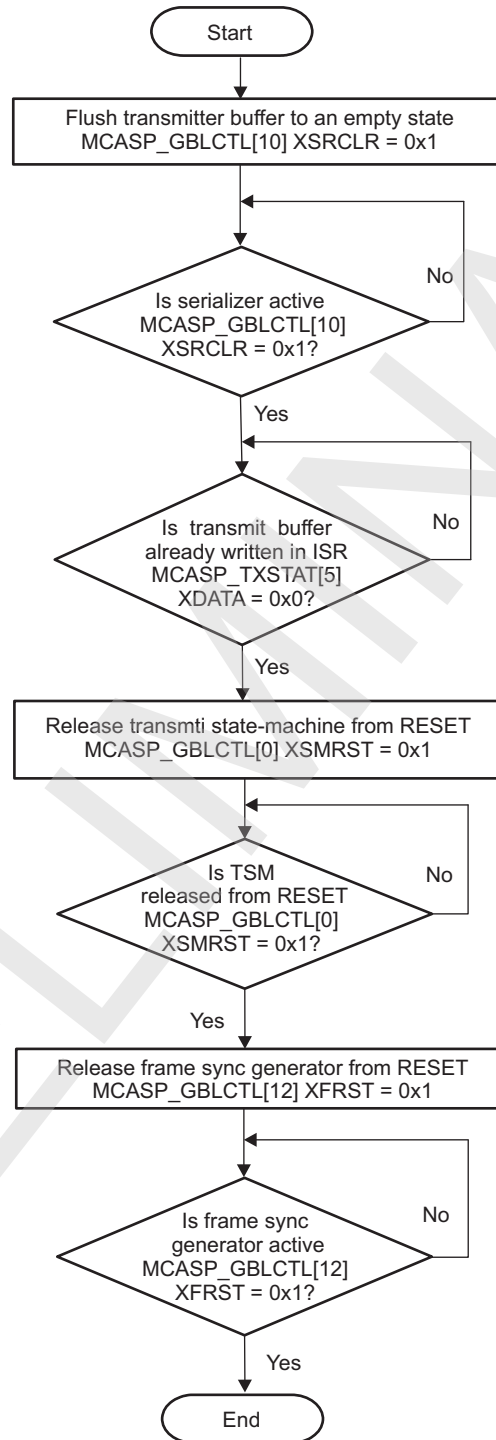
Table 23-581. Subprocess Call Summary for Main Sequence – McASP DIT Transmission Polling Method

Subprocess Name	Cross-Reference
Error handling	Figure 23-205

### 23.8.5.2.1.2 Main Sequence – McASP DIT Interrupt Transmission Method

Figure 23-202 shows the initial setup for interrupt-based transmission.

**Figure 23-202. Subsequence – DIT Transmission Startup Procedure**



mcaspp-029

Table 23-582 shows the configuration of the McASP using the McASP DIT interrupt method for transmission.

**Table 23-582. McASP Interrupt Transmission Model**

Step	Register/Bit Field/Programming Model	Value
Select the CFG or DATA port to access the transmit buffer.	MCASP_TXFMT[3] XBUSEL	0x-
Disable DMA requests generation.	MCASP_TXEVTCTL[0] XDATDMA	0x1
Clear the transmitter status logic.	MCASP_TXSTAT[8:0]	0xFF
Enable the data ready event transmit interrupt.	MCASP_EVTCTLX[5] XDATA	0x1
Optional: Enable the transmit error event interrupts.	MCASP_EVTCTLX[2] XCKFAIL MCASP_EVTCTLX[1] XSYNCERR MCASP_EVTCTLX[0] XUNDRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt.	MCASP_EVTCTLX [7] XSTAFRM MCASP_EVTCTLX[4] XLAST	0x1 0x1
Disable the DMA error event.	MCASP_EVTCTLX[7] XDMAERR	0x0
DIT transmission startup procedure	See <a href="#">Figure 23-202</a> .	

[Table 23-583](#) summarizes the register call to initialize the McASP to transmit using interrupt events.

**Table 23-583. Register Call Summary for Subsequence – McASP DIT Transmission Startup Procedure**

Register Name	Register Name
MCASP_GBLCTL	MCASP_TXBUF0

**23.8.5.2.1.3 Main Sequence –McASP DIT DMA Transmission Method**

[Table 23-584](#) shows the configuration of the McASP using the DMA method for transmission. Possible interrupt error event servicing is also considered. [Table 23-583](#) shows the initial setup for DMA-based transmission.

**NOTE:** Because of the DATA port burst access capability with the DMA method, It is preferable for transfers to be initiated through the DATA port rather than through the CFG port.

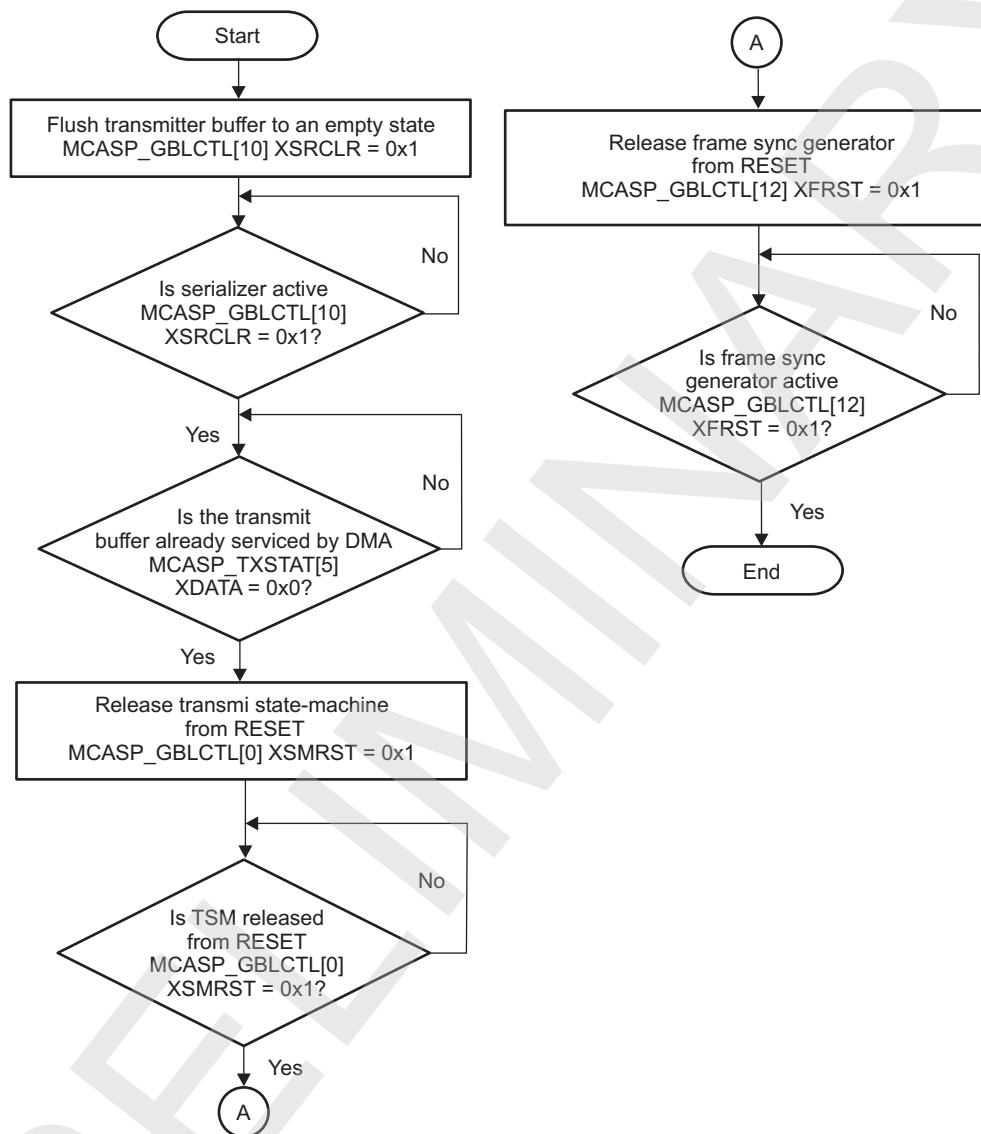
**Table 23-584. McASP DMA Transmission Model with Interrupt Events Servicing**

Step	Register/Bit Field/Programming Model	Value
Select DATA port to access the transmit buffer.	MCASP_TXFMT[3] XBUSEL	0x0
Enable the DMA requests generation.	MCASP_TXEVTCTL[0] XDATDMA	0x0
Clear the transmitter status logic.	MCASP_TXSTAT[8:0]	0xFF
Enable the DMA error event.	MCASP_EVTCTLX[7] XDMAERR	0x1
Optional: Enable the transmit error event interrupts.	MCASP_EVTCTLX[2] XCKFAIL MCASP_EVTCTLX[1] XSYNCERR MCASP_EVTCTLX[0] XUNDRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt.	MCASP_EVTCTLX [7] XSTAFRM MCASP_EVTCTLX[4] XLAST	0x1 0x1
Disable the data ready event transmit interrupt.	MCASP_EVTCTLX[5] XDATA	0x0
DMA transmission procedure	See <a href="#">Section 23.8.5.2.1.4</a> .	

### 23.8.5.2.1.4 Subsequence – DIT DMA Transmission Procedure

Figure 23-203 shows the McASP initial setup when the DMA method is used for transmission.

**Figure 23-203. Subsequence – DIT DMA Transmission Procedure**



mcasep-030

Table 23-585 shows the register call summary for the McASP DMA transmission procedure subsequence.

**Table 23-585. Register Call Summary for Subsequence – McASP DMA Transmission Procedure**

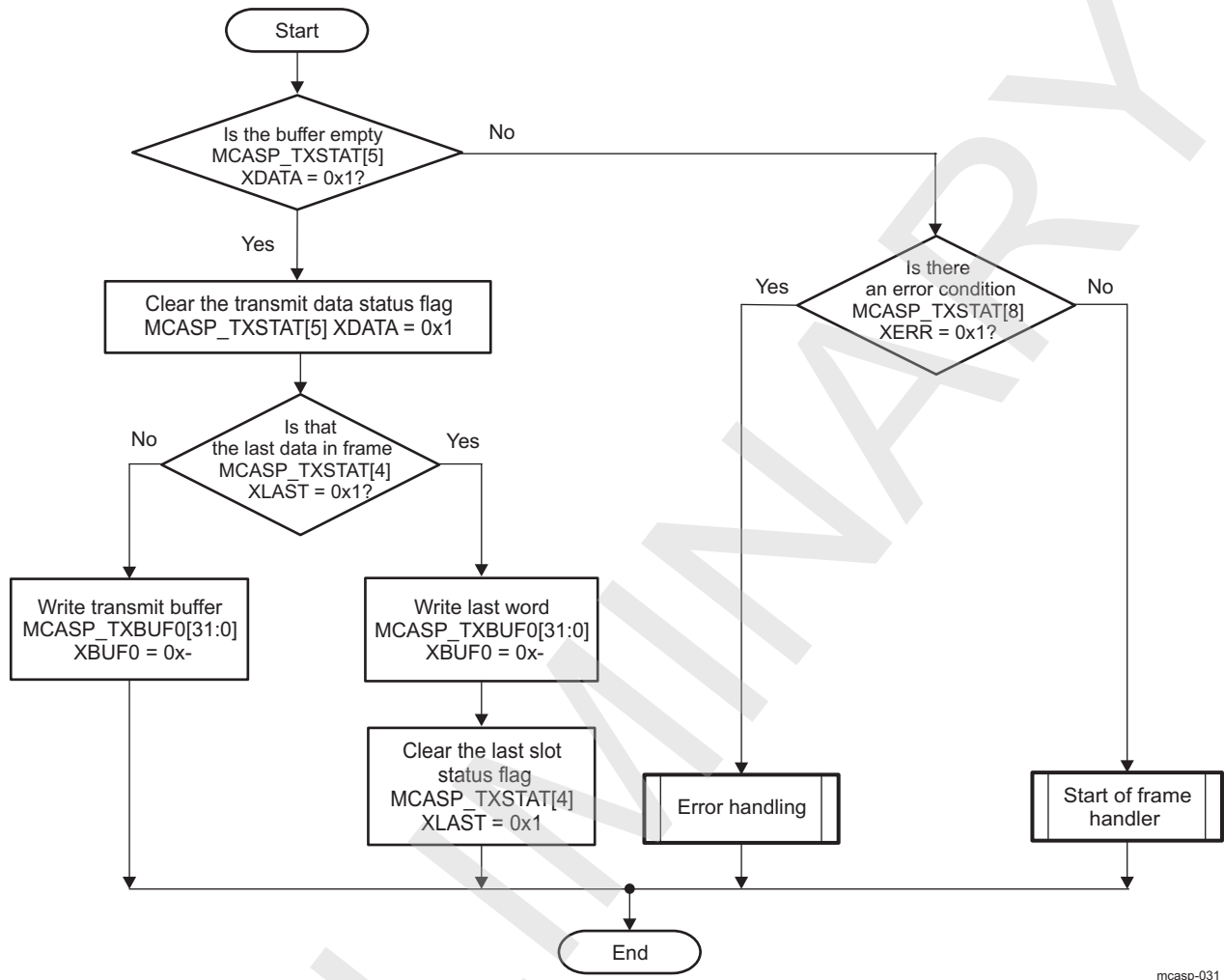
Register Name	Register Name
MCASP_TXSTAT	MCASP_GBLCTL

### 23.8.5.2.2 McASP Event Servicing

#### 23.8.5.2.2.1 McASP Interrupt Events Servicing

Figure 23-204 shows the flow of interrupt events servicing for the McASP module.

Figure 23-204. McASP Interrupt Events Servicing



mcasp-031

Table 23-586 lists the register call summary for the interrupt event servicing.

Table 23-586. Register Call Summary for McASP Interrupt Events Servicing

Register Name	Register Name
MCASP_TXSTAT	MCASP_TXBUF0

Table 23-587 lists the subprocess call summary for interrupt events servicing.

Table 23-587. Subprocess Call Summary for Interrupt Events Servicing

Subprocess Name	Cross-Reference
Error handling	Figure 23-205
Start of frame handling	Section 23.8.4.11.1

### 23.8.5.2.2.2 Subsequence – Error Handling

Figure 23-205 shows the error handling schema for the McASP, which can be implemented as part of the interrupt service routine or as part of polling sequence.

**Figure 23-205. McASP Error Handling**

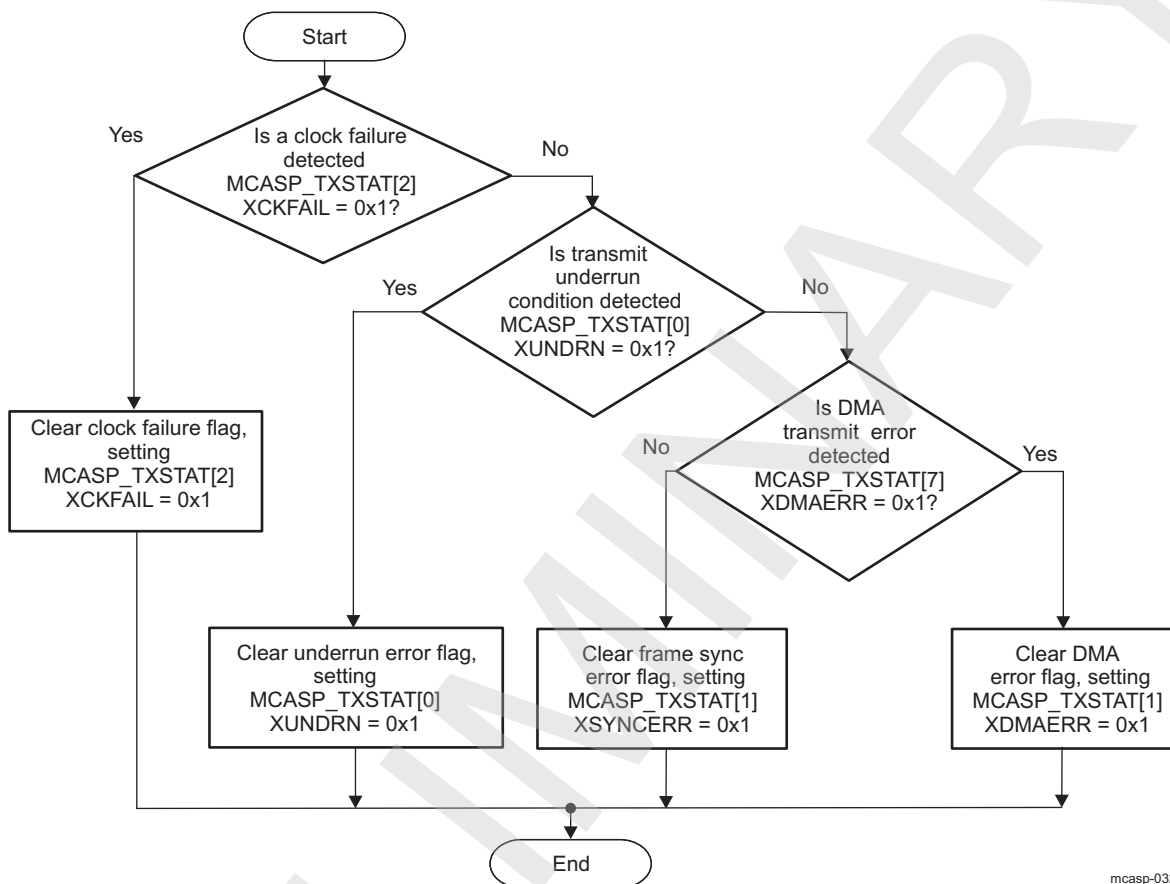


Table 23-588 lists the register call summary for the McASP error handling.

**Table 23-588. Register Call Summary for McASP Error Handling**

Register Name
MCASP_TXSTAT

**NOTE:**

- For more information about clock failure handling, see [Section 23.8.4.14.3](#).
- For more information about transmit buffer underrun handling, see [Section 23.8.4.14.1](#).
- For more information about DMA error handling, see [Section 23.8.4.14.2](#).

## 23.8.6 McASP Register Manual

### 23.8.6.1 McASP Instance Summary

Table 23-589 summarizes the McASP instances.

**Table 23-589. McASP Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-A9 Private Access	Base Address DSP Private Access	Size
McASP	0x4902 8000	0x4012 8000	0x01D2 8000	4KB

**NOTE:** Private access is an access that does not use the L3 / L4 interconnects.

### 23.8.6.2 McASP Registers

#### 23.8.6.2.1 McASP Register Summary

Table 23-590 summarizes the McASP register mapping.

**Table 23-590. McASP Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
MCASP_PID	R	32	0x0000 0000	0x4902 8000	0x4012 8000	0x01D2 8000
MCASP_SYSCONFIG	RW	32	0x0000 0004	0x4902 8004	0x4012 8004	0x01D2 8004
MCASP_PFUNC	RW	32	0x0000 0010	0x4902 8010	0x4012 8010	0x01D2 8010
MCASP_PDIR	RW	32	0x0000 0014	0x4902 8014	0x4012 8014	0x01D2 8014
MCASP_PDOUT	RW	32	0x0000 0018	0x4902 8018	0x4012 8018	0x01D2 8018
MCASP_PDIN	R	32	0x0000 001C	0x4902 801C	0x4012 801C	0x01D2 801C
MCASP_PDSET	W	32	0x0000 001C	0x4902 801C	0x4012 801C	0x01D2 801C
MCASP_PDCLR	RW	32	0x0000 0020	0x4902 8020	0x4012 8020	0x01D2 8020
RESERVED	RW	32	0x0000 0030	0x4902 8030	0x4012 8030	0x01D2 8030
RESERVED	RW	32	0x0000 0034	0x4902 8034	0x4012 8034	0x01D2 8034
RESERVED	RW	32	0x0000 0038	0x4902 8038	0x4012 8038	0x01D2 8038
MCASP_GBLCTL	RW	32	0x0000 0044	0x4902 8044	0x4012 8044	0x01D2 8044
MCASP_AMUTE	RW	32	0x0000 0048	0x4902 8048	0x4012 8048	0x01D2 8048
RESERVED	RW	32	0x0000 004C	0x4902 804C	0x4012 804C	0x01D2 804C
MCASP_TXDITCTL	RW	32	0x0000 0050	0x4902 8050	0x4012 8050	0x01D2 8050
RESERVED	RW	32	0x0000 0060	0x4902 8060	0x4012 8060	0x01D2 8060
RESERVED	RW	32	0x0000 0064	0x4902 8064	0x4012 8064	0x01D2 8064
RESERVED	RW	32	0x0000 0068	0x4902 8068	0x4012 8068	0x01D2 8068
RESERVED	RW	32	0x0000 006C	0x4902 806C	0x4012 806C	0x01D2 806C
RESERVED	RW	32	0x0000 0070	0x4902 8070	0x4012 8070	0x01D2 8070
RESERVED	RW	32	0x0000 0074	0x4902 8074	0x4012 8074	0x01D2 8074
RESERVED	RW	32	0x0000 0078	0x4902 8078	0x4012 8078	0x01D2 8078
RESERVED	RW	32	0x0000 007C	0x4902 807C	0x4012 807C	0x01D2 807C
RESERVED	RW	32	0x0000 0080	0x4902 8080	0x4012 8080	0x01D2 8080
RESERVED	R	32	0x0000 0084	0x4902 8084	0x4012 8084	0x01D2 8084
RESERVED	RW	32	0x0000 0088	0x4902 8088	0x4012 8088	0x01D2 8088
RESERVED	RW	32	0x0000 008C	0x4902 808C	0x4012 808C	0x01D2 808C

**Table 23-590. McASP Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
RESERVED	RW	32	0x0000 00A0	0x4902 80A0	0x4012 80A0	0x01D2 80A0
MCASP_TXMASK	RW	32	0x0000 00A4	0x4902 80A4	0x4012 80A4	0x01D2 80A4
MCASP_TXFMT	RW	32	0x0000 00A8	0x4902 80A8	0x4012 80A8	0x01D2 80A8
MCASP_TXFMCTL	RW	32	0x0000 00AC	0x4902 80AC	0x4012 80AC	0x01D2 80AC
MCASP_ACLKXCTL	RW	32	0x0000 00B0	0x4902 80B0	0x4012 80B0	0x01D2 80B0
MCASP_AHCLKXCTL	RW	32	0x0000 00B4	0x4902 80B4	0x4012 80B4	0x01D2 80B4
MCASP_TXTDM	RW	32	0x0000 00B8	0x4902 80B8	0x4012 80B8	0x01D2 80B8
MCASP_EVTCTLX	RW	32	0x0000 00BC	0x4902 80BC	0x4012 80BC	0x01D2 80BC
MCASP_TXSTAT	RW	32	0x0000 00C0	0x4902 80C0	0x4012 80C0	0x01D2 80C0
MCASP_TXTDMSLOT	R	32	0x0000 00C4	0x4902 80C4	0x4012 80C4	0x01D2 80C4
MCASP_TXCLKCHK	RW	32	0x0000 00C8	0x4902 80C8	0x4012 80C8	0x01D2 80C8
MCASP_TXEVTCTL	RW	32	0x0000 00CC	0x4902 80CC	0x4012 80CC	0x01D2 80CC
RESERVED	RW	32	0x0000 00D0	0x4902 80D0	0x4012 80D0	0x01D2 80D0
MCASP_DITCSRAi <sup>(1)</sup>	RW	32	0x0000 0100 + (0x04*i)	0x4902 8100 + (0x04*i)	0x4012 8100 + (0x04*i)	0x01D2 8100 + (0x04*i)
MCASP_DITCSRBi <sup>(1)</sup>	RW	32	0x0000 0118 + (0x04*i)	0x4902 8118 + (0x04*i)	0x4012 8118 + (0x04*i)	0x01D2 8118 + (0x04*i)
MCASP_DITUDRAi <sup>(1)</sup>	RW	32	0x0000 0130 + (0x04*i)	0x4902 8130 + (0x04*i)	0x4012 8130 + (0x04*i)	0x01D2 8130 + (0x04*i)
MCASP_DITUDRBi <sup>(1)</sup>	RW	32	0x0000 0148 + (0x04*i)	0x4902 8148 + (0x04*i)	0x4012 8148 + (0x04*i)	0x01D2 8148 + (0x04*i)
MCASP_XRSRCTL0	RW	32	0x0000 0180	0x4902 8180	0x4012 8180	0x01D2 8180
RESERVED	RW	32	0x0000 0184 + (0x04*j) <sup>(2)</sup>	0x4902 8184 + (0x04*j)	0x4012 8184 + (0x04*j)	0x01D2 8184 + (0x04*j)
MCASP_TXBUF0	RW	32	0x0000 0200	0x4902 8200	0x4012 8200	0x01D2 8200
RESERVED	RW	32	0x0000 0204 + (0x04*k) <sup>(3)</sup>	0x4902 8204 + (0x04*k)	0x4012 8204 + (0x04*k)	0x01D2 8204 + (0x04*k)

<sup>(1)</sup> i = 0 to 5 for McASP\_L3Interconnect  
i = 0 to 5 for McASP\_Cortex-A9  
i = 0 to 5 for McASP\_DSP

<sup>(2)</sup> j = 0 to 14

<sup>(3)</sup> k = 0 to 30

**NOTE:** The address locations listed in [Table 23-590, McASP Register Mapping Summary](#), are relevant for accessing all McASP configuration registers and McASP\_TXBUF0 registers through the McASP peripheral configuration (CFG) port.

McASP DAT allows the L3-interconnect, DSP subsystem, and MPU to access the McASP\_TXBUF0 register at locations 0x4902 A200, 0x01D2 A200, and 0x4012 A200, respectively.

### 23.8.6.2.2 McASP Register Description

[Table 23-591](#) through [Table 23-645](#) describe the individual register bits.



**Table 23-591. MCASP\_PID**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .	<b>Instance</b> McASP
<b>Description</b>	Peripheral identification register	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

**Table 23-592. Register Call Summary for Register MCASP\_PID**

Multichannel Audio Serial Port

- [McASP Registers: \[0\]](#)

**Table 23-593. MCASP\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0004	
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .	<b>Instance</b> McASP
<b>Description</b>	Power idle module configuration register.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IDLE_MODE															

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x00000000
1:0	IDLE_MODE	0x0: Force-idle mode 0x1: No-idle mode 0x2: Smart-idle mode - default state 0x3: Reserved	RW	0x2

**Table 23-594. Register Call Summary for Register MCASP\_SYSCONFIG**

Multichannel Audio Serial Port

- [McASP Power Management: \[0\]](#)
- [Global Initialization: \[1\]](#)
- [McASP Registers: \[2\]](#)

**Table 23-595. MCASP\_PFUNC**

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .	<b>Instance</b> McASP
<b>Description</b>	Specifies the function of the pins as either a McASP pin or a GPIO pin	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	AXR0

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	RW	0
28	AFSX	Determines if <code>abe_mcasp_afsx</code> pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
27	AHCLKX	Determines if <code>abe_mcasp_ahclkx</code> pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
26	ACLKX	Determines if <code>abe_mcasp_aclkx</code> pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
25	AMUTE	Determines if <code>abe_mcasp_amute</code> pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
24:1	RESERVED	Reserved	RW	0x000
0	AXR0	Determines if <code>abe_mcasp_axr</code> pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0

**Table 23-596. Register Call Summary for Register MCASP\_PFUNC**

Multichannel Audio Serial Port

- [Transfer Modes: \[0\]](#)
- [Global Initialization: \[1\]](#)
- [McASP Registers: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\]](#)

**Table 23-597. MCASP\_PDIR**

<b>Address Offset</b>	0x0000 0014	
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .	<b>Instance</b> McASP
<b>Description</b>	Pin direction register - specifies the direction of the McASP pins as either an input or an output pin. For the module to operate properly, configure them as outputs.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	AXR0

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	RW	0
28	AFSX	Determines if <code>abe_mcasp_afsx</code> pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
27	AHCLKX	Determines if <code>abe_mcasp_ahclkx</code> pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
26	ACLKX	Determines if <code>abe_mcasp_aclkx</code> pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
25	AMUTE	Determines if <code>abe_mcasp_amute</code> pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
24:1	RESERVED	Reserved	RW	0x000
0	AXR0	Determines if <code>abe_mcasp_axr</code> pin functions as an input or output. 0x0: Input 0x1: Output	RW	0

**Table 23-598. Register Call Summary for Register MCASP\_PDIR**

Multichannel Audio Serial Port

- [Transfer Modes: \[0\]](#)
- [Global Initialization: \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [McASP Registers: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)

**Table 23-599. MCASP\_PDOUT**

<b>Address Offset</b>	0x0000 0018
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .
<b>Description</b>	<p>Pin data output register - holds a value for data out at all times, and may be read back at all times. The value held by <a href="#">MCASP_PDOUT</a> is not affected by writing to <a href="#">MCASP_PDIR</a> and <a href="#">MCASP_PFUNC</a>. However, the data value in <a href="#">MCASP_PDOUT</a> is driven out onto the McASP pin only if the corresponding bit in <a href="#">MCASP_PFUNC</a> is set to 1 (GPIO function) and the corresponding bit in <a href="#">MCASP_PDIR</a> is set to 1 (output).</p> <p>When reading data, it returns the corresponding bit value in <a href="#">MCASP_PDOUT[n]</a>; it does not return the input from the I/O pin.</p> <p>When writing data, writes to the corresponding <a href="#">MCASP_PDOUT[n]</a> bit.</p> <p>PDOUT has these aliases or alternate addresses:</p> <ul style="list-style-type: none"> <li><a href="#">MCASP_PDSET</a> - when written to at this address, writing a 1 to a bit in <a href="#">MCASP_PDSET</a> sets the corresponding bit in <a href="#">MCASP_PDOUT</a> to 1; writing a 0 has no effect and keeps the bits in <a href="#">MCASP_PDOUT</a> unchanged.</li> <li><a href="#">MCASP_PDCLR</a> - when written to at this address, writing a 1 to a bit in <a href="#">MCASP_PDCLR</a> clears the corresponding bit in <a href="#">MCASP_PDOUT</a> to 0; writing a 0 has no effect and keeps the bits in <a href="#">MCASP_PDOUT</a> unchanged.</li> </ul> <p>There is only one set of data-out bits, <a href="#">MCASP_PDOUT[31:0]</a>. The other registers, <a href="#">MCASP_PDSET</a> and <a href="#">MCASP_PDCLR</a>, are just different addresses for the same control bits, with different behaviors during writes.</p>
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	AFSX	AHCLKX	ACLKX	AMUTE		RESERVED														AXR0									

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	RW	0
28	AFSX	Determines drive on AFSX output pin when the corresponding <a href="#">MCASP_PFUNC[28]</a> and <a href="#">MCASP_PDIR[28]</a> bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
27	AHCLKX	Determines drive on AHCLKX output pin when the corresponding <a href="#">MCASP_PFUNC[27]</a> and <a href="#">MCASP_PDIR[27]</a> bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
26	ACLKX	Determines drive on ACLKX output pin when the corresponding <a href="#">MCASP_PFUNC[26]</a> and <a href="#">MCASP_PDIR[26]</a> bits are set to 1 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
25	AMUTE	Determines drive on AMUTE output pin when the corresponding <a href="#">MCASP_PFUNC[25]</a> and <a href="#">MCASP_PDIR[25]</a> bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
24:1	RESERVED	Reserved	RW	0x000
0	AXR0	Determines drive on <code>abe_mcasp_axr</code> output pin when the corresponding <a href="#">MCASP_PFUNC[0]</a> and <a href="#">MCASP_PDIR[0]</a> bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0

**Table 23-600. Register Call Summary for Register MCASP\_PDOUT**

Multichannel Audio Serial Port

- [McASP Registers: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\]](#)

**Table 23-601. MCASP\_PDIN**

<b>Address Offset</b>	0x0000 001C
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .
<b>Description</b>	Pin data input register - holds the state of all the McASP pins. <b>MCASP_PDIN</b> allows reading the actual value of the pin, regardless of the state of <b>MCASP_PFUNC</b> and <b>MCASP_PDIR</b> .
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED	RESERVED																AXR0							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0
28	AFSX	Logic level on <code>abe_mcasp_afsx</code> pin. 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
27	AHCLKX	Logic level on <code>abe_mcasp_ahclkx</code> pin. 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
26	ACLKX	Logic level on <code>abe_mcasp_aclkx</code> pin. 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
25	AMUTE	Logic level on <code>abe_mcasp_amute</code> pin. 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
24:1	RESERVED	Reserved	R	0x000
0	AXR0	Logic level on <code>abe_mcasp_axr</code> pin. 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0

**Table 23-602. Register Call Summary for Register MCASP\_PDIN**

- Multichannel Audio Serial Port
- [McASP Registers: \[0\] \[1\]](#)

**Table 23-603. MCASP\_PDSET**

<b>Address Offset</b>	0x0000 001C
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .
<b>Description</b>	The pin data set register is an alias of the pin data output register ( <b>MCASP_PDOUT</b> ) for writes only. Writing a 1 to the <b>MCASP_PDSET</b> bit sets the corresponding bit in <b>MCASP_PDOUT</b> and, if <b>MCASP_PFUNC</b> = 1 (GPIO function) and <b>MCASP_PDIR</b> = 1 (output), drives a logic high on the pin.
<b>Type</b>	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED	RESERVED																AXR0							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	W	0
28	AFSX	Allows the corresponding AFSX bit in <a href="#">MCASP_PDOUT</a> to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [28] bit is set to 1.	W	0
27	AHCLKX	Allows the corresponding AHCLKX bit in <a href="#">MCASP_PDOUT</a> to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [27] bit is set to 1.	W	0
26	ACLKX	Allows the corresponding ACLKX bit in <a href="#">MCASP_PDOUT</a> to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [26] bit is set to 1.	W	0
25	AMUTE	Allows the corresponding AMUTE bit in <a href="#">MCASP_PDOUT</a> to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [25] bit is set to 1.	W	0
24:1	RESERVED	Reserved	W	0x000
0	AXR0	Allows the AXR0 bit in <a href="#">MCASP_PDOUT</a> to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [0] bit is set to 1.	W	0

**Table 23-604. Register Call Summary for Register MCASP\_PDSET**

Multichannel Audio Serial Port

- [McASP Registers: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-605. MCASP\_PDCLR**

<b>Address Offset</b>	0x0000 0020
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .
<b>Description</b>	The pin data clear register is an alias of the pin data output register ( <a href="#">MCASP_PDOUT</a> ) for writes only. Writing a 1 to the <a href="#">MCASP_PDCLR</a> bit clears the corresponding bit in <a href="#">MCASP_PDOUT</a> and, if <a href="#">MCASP_PFUNC</a> = 1 (GPIO function) and <a href="#">MCASP_PDIR</a> = 1 (output), drives a logic low on the pin.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	AFSX	AHCLKX	ACLKX	AMUTE	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	AXR0

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	RW	0
28	AFSX	Allows the corresponding AFSX bit in <a href="#">MCASP_PDOUT</a> to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [28] bit is cleared to 0.	RW	0
27	AHCLKX	Allows the corresponding AHCLKX bit in <a href="#">MCASP_PDOUT</a> to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [27] bit is cleared to 0.	RW	0

Bits	Field Name	Description	Type	Reset
26	ACLKX	Allows the corresponding ACLKX bit in <a href="#">MCASP_PDOUT</a> to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [26] bit is cleared to 0.	RW	0
25	AMUTE	Allows the corresponding AMUTE bit in <a href="#">MCASP_PDOUT</a> to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [25] bit is cleared to 0.	RW	0
24:1	RESERVED	Reserved	RW	0x000
0	AXR0	Allows the AXR0 bit in <a href="#">MCASP_PDOUT</a> to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: <a href="#">MCASP_PDOUT</a> [0] bit is cleared to 0.	RW	0

**Table 23-606. Register Call Summary for Register MCASP\_PDCLR**

- Multichannel Audio Serial Port
- [McASP Registers: \[0\] \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-607. MCASP\_GBLCTL**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Global transmit control register - provides initialization of the transmit section. The bit fields in <a href="#">MCASP_GBLCTL</a> are synchronized and latched by the <code>abe_mcasp_aclkx</code> clock. Before programming <a href="#">MCASP_GBLCTL</a> , ensure that the serial clocks are running. If the corresponding external serial clock, <code>abe_mcasp_clkx</code> , is not yet running, select the internal serial clock source in <code>AHCLKXCTL</code> and <code>ACLKXCTL</code> before programming <a href="#">MCASP_GBLCTL</a> . Also, after programming any bits in <a href="#">MCASP_GBLCTL</a> , do not proceed until reading back from <a href="#">MCASP_GBLCTL</a> and verifying that the bits in <a href="#">MCASP_GBLCTL</a> are latched.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST	RESERVED										

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	RW	0x00000
12	XFRST	Transmit frame-sync generator reset enable bit 0x0: The transmit frame-sync generator is reset. 0x1: The transmit frame-sync generator is active. When released from reset, the transmit frame-sync generator begins counting serial clocks and generating frame sync as programmed.	RW	0

Bits	Field Name	Description	Type	Reset
11	XSMRST	<p>Transmit state-machine reset enable bit</p> <p>0x0: The transmit state-machine is held in reset.</p> <p>abe_mcasp_axr pin state: If <a href="#">MCASP_PFUNC</a>[0] = 0 and <a href="#">MCASP_PDIR</a>[0] = 1, the serializer drives the abe_mcasp_axr pin to the state specified for inactive time slot</p> <p>0x1: The transmit state-machine is released from reset. When released from reset, the transmit state-machine immediately transfers data from XBUF[n] to XRSR[n]. The transmit state-machine sets the underrun flag (XUNDRN) in MCASP_XSTAT, if XBUF[n] have not been preloaded with data before reset is released. The transmit state-machine also immediately begins detecting frame sync and is ready to transmit. Transmission of TDM time slot begins at slot 0 after reset is released.</p>	RW	0
10	XSRCLR	<p>Transmit serializer clear enable bit. By clearing and then setting this bit, the transmit buffer is flushed to an empty state (XDATA = 1). If XSMRST = 1, XSRCLR = 1, XDATA = 1, and XBUF is not loaded with new data before the start of the next active time slot, an underrun occurs.</p> <p>0x0: The transmit serializer is cleared.</p> <p>0x1: The transmit serializer is active. When the transmit serializer is first taken out of reset (XSRCLR changes from 0 to 1), the transmit data ready bit (XDATA) in MCASP_XSTAT is set to indicate XBUF is ready to be written.</p>	RW	0
9	XHCLKRST	<p>Transmit high-frequency clock divider reset enable bit</p> <p>0x0: The transmitter high-frequency clock divider is held in reset.</p> <p>0x1: The transmitter high-frequency clock divider is running.</p>	RW	0
8	XCLKRST	<p>Transmit clock divider reset enable bit</p> <p>0x0: The transmit clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input.</p> <p>0x1: The transmit clock divider is running.</p>	RW	0
7:0	RESERVED	Reserved	RW	0x0

**Table 23-608. Register Call Summary for Register MCASP\_GBLCTL**

Multichannel Audio Serial Port

- [McASP Software Reset: \[0\] \[1\]](#)
- [Transfer Modes: \[2\]](#)
- [Global Initialization: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [Operational Modes Configuration: \[9\] \[10\] \[11\]](#)
- [McASP Registers: \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)

**Table 23-609. MCASP\_AMUTE**

<b>Address Offset</b>	0x0000 0048																														
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .																														
<b>Description</b>	Mute control register - Controls the McASP mute output pin (abe_mcasp_amute)																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XDMAERR	RESERVED	XCKFAIL	RESERVED	XSYNCERR	RESERVED	XUNDRN	RESERVED	INSTAT	INEN	INPOL	MUTEN				



Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	RW	0x00000
12	XDMAERR	Drives AMUTE active enable bit on transmit DMA error (XDMAERR). 0x0: Drive is disabled. Detection of transmit DMA error is ignored by <code>abe_mcaspp_amute</code> . 0x1: Drive is enabled (active). Upon detection of transmit DMA error, <code>abe_mcaspp_amute</code> is active and is driven according to MUTEN bits.	RW	0
11	RESERVED	Reserved	RW	0
10	XCKFAIL	XMT bad clock. Drives AMUTE active enable bit on transmit clock failure (XCKFAIL). 0x0: Drive is disabled. Detection of transmit clock failure is ignored by <code>abe_mcaspp_amute</code> . 0x1: Drive is enabled (active). Upon detection of transmit clock failure, <code>abe_mcaspp_amute</code> is active and is driven according to MUTEN bits.	RW	0
9	RESERVED	Reserved	RW	0
8	XSYNCERR	XMT unexpected FS. Drives AMUTE active enable bit on unexpected transmit frame-sync error (XSYNCERR). 0x0: Drive is disabled. Detection of unexpected transmit frame-sync error is ignored by <code>abe_mcaspp_amute</code> . 0x1: Drive is enabled (active). Upon detection of unexpected transmit frame-sync error, <code>abe_mcaspp_amute</code> is active and is driven according to MUTEN bit.	RW	0
7	RESERVED	Reserved	RW	0
6	XUNDRN	XMT underrun occurs. Drives AMUTE active enable bit on transmit underrun error (XUNDRN). 0x0: Drive is disabled. Detection of transmit underrun error is ignored by <code>abe_mcaspp_amute</code> . 0x1: Drive is enabled (active). Upon detection of transmit underrun error, <code>abe_mcaspp_amute</code> is active and is driven according to MUTEN bit.	RW	0
5	RESERVED	Reserved	RW	0
4	INSTAT	Status of mute in pin, determines drive on <code>abe_mcaspp_axr</code> pin when the <code>MCASP_PFUNC[0]</code> and <code>MCASP_PDIR[0]</code> bits are set to 1. Read 0x0: Inactive Read 0x1: Active. Audio mute in error is detected.	R	0
3	INEN	Drive <code>abe_mcaspp_amute</code> active when <code>abe_mcaspp_amutein</code> error is active. 0x0: Drive is disabled. <code>abe_mcaspp_amutein</code> is ignored by <code>abe_mcaspp_amute</code> . 0x1: Drive is enabled (active). <code>INSTAT = 1</code> drives <code>abe_mcaspp_amute</code> active.	RW	0
2	INPOL	Audio mute in ( <code>abe_mcaspp_amutein</code> ) polarity select bit 0x0: Polarity is active high. A high on <code>abe_mcaspp_amutein</code> sets <code>INSTAT</code> to 1. 0x1: Polarity is active low. A low on <code>abe_mcaspp_amutein</code> sets <code>INSTAT</code> to 1.	RW	0
1:0	MUTEN	<code>abe_mcaspp_amute</code> pin enable bit field (unless overridden by GPIO registers) 0x0: Disabled, pin goes to 3-state condition. 0x1: Pin is driven high if error is detected. 0x2: Pin is driven low if error is detected.	RW	0x0

**Table 23-610. Register Call Summary for Register MCASP\_AMUTE**

Multichannel Audio Serial Port

- [Transfer Modes](#): [0]
- [Audio Mute \(AMUTE\) Function](#): [1] [2] [3] [4]
- [Error Reporting](#): [5]
- [Global Initialization](#): [6] [7] [8] [9] [10] [11] [12]
- [McASP Registers](#): [13]

**Table 23-611. MCASP\_TXDITCTL**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Transmit DIT mode control register, controls DIT operations of the McASP		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VB		VA		RESERVED		DITEN									

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	RW	0x00000000
3	VB	Valid bit for odd time slots (DIT right subframe). 0x0: V bit is 0 during odd DIT subframes. 0x1: V bit is 1 during odd DIT subframes.	RW	0
2	VA	Valid bit for even time slots (DIT left subframe). 0x0: V bit is 0 during even DIT subframes. 0x1: V bit is 1 during even DIT subframes.	RW	0
1	RESERVED	Reserved	RW	0
0	DITEN	DIT mode enable bit 0x0: DIT mode is disabled. 0x1: DIT mode is enabled. Transmitter operates in DIT encoded mode.	RW	0

**Table 23-612. Register Call Summary for Register MCASP\_TXDITCTL**

Multichannel Audio Serial Port

- [Transfer Modes](#): [0]
- [Global Initialization](#): [1] [2] [3]
- [McASP Registers](#): [4]

**Table 23-613. MCASP\_TXMASK**

<b>Address Offset</b>	0x0000 00A4	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Transmit format unit bit mask register - Determines which bits of the transmitted data are masked off before being shifted out the McASP		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMASK31	XMASK30	XMASK29	XMASK28	XMASK27	XMASK26	XMASK25	XMASK24	XMASK23	XMASK22	XMASK21	XMASK20	XMASK19	XMASK18	XMASK17	XMASK16	XMASK15	XMASK14	XMASK13	XMASK12	XMASK11	XMASK10	XMASK9	XMASK8	XMASK7	XMASK6	XMASK5	XMASK4	XMASK3	XMASK2	XMASK1	XMASK0

Bits	Field Name	Description	Type	Reset
31:0	XMASK[31:0]	Transmit data mask enable bit  0x0: The corresponding bit of transmit data is masked out and then transmitted out the McASP in place of the original bit. 0x1: The corresponding bit of transmit data is transmitted out the McASP.	RW	0

**Table 23-614. Register Call Summary for Register MCASP\_TXMASK**

Multichannel Audio Serial Port

- [Format Unit: \[0\] \[1\]](#)
- [Transfer Modes: \[2\]](#)
- [Global Initialization: \[3\] \[4\]](#)
- [McASP Registers: \[5\]](#)

**Table 23-615. MCASP\_TXFMT**

<b>Address Offset</b>	0x0000 00A8	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Transmit bit stream format register - configures the transmit data format		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														XDATDLY	XRVR	XPAD	RESERVED				XSSZ			XBUSEL	XROT						

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	RW	0x0000
17:16	XDATDLY	Transmit sync bit delay  0x0: 0 bit delay - The first transmit data bit, <code>abe_mcaspx_axr</code> , occurs in the same <code>abe_mcaspx_aclckx</code> cycle as the transmit frame sync ( <code>abe_mcaspx_afx</code> ). 0x1: Reserved 0x2: Reserved	RW	0x0
15	XRVR	Transmit serial bit stream order 0x0: Bit stream is LSB first. No bit reversal is performed in transmit format unit. 0x1: Reserved	RW	0x0
14:13	XPAD	Pad value for extra bits in slot not belonging to word defined by XMASK. This field only applies to bits when XMASK[n] = 0. 0x0: Pad extra bits with 0. 0x1 to 0x3: Reserved	RW	0x00
12:8	RESERVED	Reserved	RW	0x0
7:4	XSSZ	Transmit slot size 0x0 to 0xE: Reserved 0xF: Slot size is 32 bits.	RW	0x0
3	XBUSEL	Selects whether writes to the serializer buffer XBUF originate from the peripheral configuration port or the DMA port. 0x0: Writes to XBUF originate from the data port. Writes to XBUF from the peripheral configuration port are ignored with no effect on the McASP. 0x1: Writes to XBUF originate from the peripheral configuration port. Writes to XBUF from the data port are ignored with no effect on the McASP.	RW	0

Bits	Field Name	Description	Type	Reset
2:0	XROT	Right-rotation value for transmit rotate right format unit 0x0: Rotate right by 0 (no rotation). 0x1: Reserved 0x2: Rotate right by 8 bit positions. 0x3 to 0x7: Reserved	RW	0x0

**Table 23-616. Register Call Summary for Register MCASP\_TXFMT**

Multichannel Audio Serial Port

- [Format Unit: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [Transmit State-Machine: \[9\]](#)
- [Transfer Modes: \[10\] \[11\]](#)
- [Data Transmission: \[12\] \[13\]](#)
- [Global Initialization: \[14\] \[15\] \[16\]](#)
- [Operational Modes Configuration: \[17\] \[18\] \[19\]](#)
- [McASP Registers: \[20\]](#)

**Table 23-617. MCASP\_TXFMCTL**

<b>Address Offset</b>	0x0000 00AC		
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .	<b>Instance</b>	McASP
<b>Description</b>	Transmit frame-sync control register - configures the transmit frame sync (abe_mcasp_afsx).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XMOD						RESERVED	FXWID	RESERVED	FSXM	FSXP					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW	0x0000
15:7	XMOD	Transmit frame-sync mode select bits 0x180: 384-slot DIT mode All other: Reserved	RW	0x000
6:5	RESERVED	Reserved	RW	0x0
4	FXWID	The transmit frame-sync width select bit indicates the width of the transmit frame sync (abe_mcasp_afsx) during its active period. 0x0: Single bit 0x1: Reserved	RW	0
3:2	RESERVED	Reserved	RW	0x0
1	FSXM	Transmit frame-sync generation select bit 0x0: Reserved 0x1: Internally-generated transmit frame sync	RW	0
0	FSXP	Transmit frame-sync polarity select bit 0x0: Rising Edge - A rising edge on transmit frame sync (abe_mcasp_afsx) indicates the beginning of a frame. 0x1: Reserved	RW	0

**Table 23-618. Register Call Summary for Register MCASP\_TXFMCTL**

Multichannel Audio Serial Port

- [McASP Clock and Frame-Sync Configuration: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [Transfer Modes: \[5\] \[6\]](#)
- [Global Initialization: \[7\] \[8\]](#)
- [McASP Registers: \[9\]](#)

**Table 23-619. MCASP\_ACLKXCTL**

<b>Address Offset</b>	0x0000 00B0	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Transmit clock control register - Configures the transmit bit clock (abe_mcasp_aclkx) and the transmit clock generator.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ASYNC		CLKXM		CLKXDIV											

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	RW	0x00
6	ASYNC	Transmit operation asynchronous enable bit 0x0: Reserved 0x1: Asynchronous	RW	1
5	CLKXM	Transmit bit clock source bit 0x0: Reserved 0x1: Internal (output of divider)	RW	1
4:0	CLKXDIV	Transmit bit clock divide ratio bits, determine the divide-down ratio from AHCLKX to ACLKX. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 to 0x1F: Divide-by-3 to divide-by-32	RW	0x00

**Table 23-620. Register Call Summary for Register MCASP\_ACLKXCTL**

Multichannel Audio Serial Port

- [McASP Clock and Frame-Sync Configuration: \[0\] \[1\] \[2\]](#)
- [Transfer Modes: \[3\] \[4\]](#)
- [Global Initialization: \[5\]](#)
- [McASP Registers: \[6\]](#)

**Table 23-621. MCASP\_AHCLKXCTL**

<b>Address Offset</b>	0x0000 00B4	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	High-frequency transmit clock control register - Configures the transmit high-frequency master clock (abe_mcasp_ahclkx) and the transmit clock generator.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														HCLKXM		RESERVED		HCLKXDIV													

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW	0x000
15	HCLKXM	Transmit high-frequency clock source bit 0x0: Reserved 0x1: Internal transmit high-frequency clock source from output of programmable high clock divider	RW	1
14:12	RESERVED	Reserved	RW	0x0
11:0	HCLKXDIV	Transmit high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to abe_mcaspl_ahclkx. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 to 0xFFFF: Divide-by-3 to divide-by-4096	RW	0x000

**Table 23-622. Register Call Summary for Register MCASP\_AHCLKXCTL**

Multichannel Audio Serial Port

- [McASP Clock and Frame-Sync Configuration: \[0\] \[1\] \[2\]](#)
- [Transfer Modes: \[3\]](#)
- [Global Initialization: \[4\]](#)
- [McASP Registers: \[5\]](#)

**Table 23-623. MCASP\_TXTDM**

<b>Address Offset</b>	0x0000 00B8	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Transmit TDM slot 0-31 register - TDM time slot counter range is to 384 slots (to support SPDIF blocks of 384 subframes).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XTDMS31	XTDMS30	XTDMS29	XTDMS28	XTDMS27	XTDMS26	XTDMS25	XTDMS24	XTDMS23	XTDMS22	XTDMS21	XTDMS20	XTDMS19	XTDMS18	XTDMS17	XTDMS16	XTDMS15	XTDMS14	XTDMS13	XTDMS12	XTDMS11	XTDMS10	XTDMS9	XTDMS8	XTDMS7	XTDMS6	XTDMS5	XTDMS4	XTDMS3	XTDMS2	XTDMS1	XTDMS0

Bits	Field Name	Description	Type	Reset
31:0	XTDMS[31:0]	Transmitter mode during TDM time slot n 0x0: Reserved 0x1: The transmit TDM time slot n is active. The transmit serializer shifts out data during this slot according to the serializer control register (MCASP_SRCTL).	RW	0

**Table 23-624. Register Call Summary for Register MCASP\_TXTDM**

Multichannel Audio Serial Port

- [Transmit TDM Sequencer: \[0\]](#)
- [Transfer Modes: \[1\] \[2\]](#)
- [Global Initialization: \[3\]](#)
- [McASP Registers: \[4\]](#)

**Table 23-625. MCASP\_EVTCTLX**

<b>Address Offset</b>	0x0000 00BC
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .
<b>Instance</b>	McASP
<b>Description</b>	Transmitter Interrupt control register - controls generation of the McASP transmit interrupt (XINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates XINT.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XSTAFRM	RESERVED	XDATA	XLAST	XDMAERR	XCKFAIL	XSYNCERR	XUNDRN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	RW	0x000000
7	XSTAFRM	Transmit start of frame interrupt enable bit 0x0: Interrupt is disabled. A transmit-start-of-frame interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit-start-of-frame interrupt generates a McASP transmit interrupt (XINT).	RW	0
6	RESERVED	Reserved	RW	0
5	XDATA	Transmit data-ready interrupt enable bit 0x0: Interrupt is disabled. A transmit data-ready interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit data-ready interrupt generates a McASP transmit interrupt (XINT).	RW	0
4	XLAST	Transmit last slot interrupt enable bit 0x0: Interrupt is disabled. A transmit-last-slot interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit-last-slot interrupt generates a McASP transmit interrupt (XINT).	RW	0
3	XDMAERR	Transmit DMA error interrupt enable bit 0x0: Interrupt is disabled. A transmit DMA error interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit DMA error interrupt generates a McASP transmit interrupt (XINT).	RW	0
2	XCKFAIL	Transmit clock failure interrupt enable bit 0x0: Interrupt is disabled. A transmit clock failure interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit clock failure interrupt generates a McASP transmit interrupt (XINT).	RW	0
1	XSYNCERR	Unexpected transmit frame-sync interrupt enable bit 0x0: Interrupt is disabled. An unexpected transmit frame-sync interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. An unexpected transmit frame-sync interrupt generates a McASP transmit interrupt (XINT).	RW	0
0	XUNDRN	Transmitter underrun interrupt enable bit 0x0: Interrupt is disabled. A transmitter underrun interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmitter underrun interrupt generates a McASP transmit interrupt (XINT).	RW	0

**Table 23-626. Register Call Summary for Register MCASP\_EVTCTLX**

Multichannel Audio Serial Port

- [Data Transmission](#): [0] [1]
- [Interrupt Requests](#): [2] [3] [4] [5] [6] [7] [8] [9]
- [Error Reporting](#): [10] [11]
- [Operational Modes Configuration](#): [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25]
- [McASP Registers](#): [26] [27] [28] [29] [30] [31] [32] [33]

**Table 23-627. MCASP\_TXSTAT**

<b>Address Offset</b>	0x0000 00C0	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Transmitter status register - If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes the generation of a new interrupt request.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XERR	XDMAERR	XSTAFRM	XDATA	XLAST	XTDMSLOT	XCKFAIL	XSYNCERR	XUNDRN							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	RW	0x000000
8	XERR	XERR bit always returns a logic-OR of: XUNDRN   XSYNCERR   XCKFAIL   XDMAERR. Allows a single bit to be checked to determine if a transmitter error interrupt has occurred.  0x0: No errors have occurred. 0x1: An error has occurred.	RW	0
7	XDMAERR	Transmit DMA error flag. XDMAERR is set when the CPU or DMA writes more words to the data port of the McASPin a given time slot than it should. Causes a transmit interrupt (XINT) if this bit and XDMAERR in <a href="#">MCASP_EVTCTLX</a> are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect.  0x0: Transmit DMA error did not occur. 0x1: Transmit DMA error occurred.	RW	0
6	XSTAFRM	Transmit start of frame flag. Causes a transmit interrupt (XINT) if this bit and XSTAFRM in <a href="#">MCASP_EVTCTLX</a> are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect.  0x0: No new transmit frame sync (AFSX) is detected. 0x1: A new transmit frame sync (AFSX) is detected.	RW	0
5	XDATA	Transmit data ready flag. Causes a transmit interrupt (XINT) if this bit and XDATA in <a href="#">MCASP_EVTCTLX</a> are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect  0x0: XBUF is written and is full  0x1: Data is copied from XBUF to XRSR. XBUF is empty and ready to be written. XDATA is also set when the transmit serializers are taken out of reset. When XDATA is set, it always causes a DMA event (AXEVT).	RW	0
4	XLAST	Transmit last slot flag. XLAST, along with XDATA, are set if the current slot is the last slot in a frame. Causes a transmit interrupt (XINT) if this bit and XLAST in <a href="#">MCASP_EVTCTLX</a> are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect.  0x0: Current slot is not the last slot in a frame. 0x1: Current slot is the last slot in a frame. XDATA is also set.	RW	0



Bits	Field Name	Description	Type	Reset
3	XTDMSLOT	Returns the LSB of XSLOT. Allows a single read of XSTAT to determine whether the current TDM time slot is even or odd. read 0x0: Current TDM time slot is odd. read 0x1: Current TDM time slot is even.	R	0
2	XCKFAIL	Transmit clock failure flag. XCKFAIL is set when the transmit clock failure detection circuit reports an error. Causes a transmit interrupt (XINT) if this bit and XCKFAIL in <a href="#">MCASP_EVTCTLX</a> are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect. 0x0: Transmit clock failure did not occur. 0x1: Transmit clock failure occurred	RW	0
1	XSYNCERR	Unexpected transmit frame-sync flag. XSYNCERR is set when a new transmit frame sync (AFSX) occurs before it is expected. Causes a transmit interrupt (XINT) if this bit and XSYNCERR in <a href="#">MCASP_EVTCTLX</a> are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect. 0x0: Unexpected transmit frame sync did not occur 0x1: Unexpected transmit frame sync occurred.	RW	0
0	XUNDRN	Transmitter underrun flag. XUNDRN is set when the transmit serializer is instructed to transfer data from XBUF to XRSR, but XBUF has not yet been serviced with new data since the last transfer. Causes a transmit interrupt (XINT) if this bit and XUNDRN in <a href="#">MCASP_EVTCTLX</a> are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect. 0x0: Transmitter underrun did not occur 0x1: Transmitter underrun occurred.	RW	0

**Table 23-628. Register Call Summary for Register MCASP\_TXSTAT**

Multichannel Audio Serial Port

- [Transmit State-Machine: \[0\]](#)
- [Data Transmission: \[1\] \[2\] \[3\] \[4\]](#)
- [Interrupt Requests: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)
- [Error Reporting: \[16\] \[17\] \[18\] \[19\] \[20\] \[21\]](#)
- [Operational Modes Configuration: \[22\] \[23\] \[24\] \[25\] \[26\] \[27\]](#)
- [McASP Registers: \[28\]](#)

**Table 23-629. MCASP\_TXTDMSLOT**

<b>Address Offset</b>	0x0000 00C4	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Current transmit TDM time slot register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XSLOTCNT															

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x000000
9:0	XSLOTCNT	Current transmit time slot count. the value of this register is 0b0101111111 (0x17f) during reset and 0 after reset.	R	0x000

**Table 23-630. Register Call Summary for Register MCASP\_TXTDMSLOT**

Multichannel Audio Serial Port

- [Transmit TDM Sequencer: \[0\]](#)
- [McASP Registers: \[1\]](#)

**Table 23-631. MCASP\_TXCLKCHK**

<b>Address Offset</b>	0x0000 00C8																														
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .								<b>Instance</b>	McASP																					
<b>Description</b>	Transmit clock check control register - configures the transmit clock failure detection circuit.																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCNT								XMAX								XMIN				RESERVED				XPS							
<b>Bits</b>	<b>Field Name</b>		<b>Description</b>														<b>Type</b>	<b>Reset</b>													
31:24	XCNT		Transmit clock count value (from previous measurement). The clock circuit continually counts the number of system clocks for every 32 transmit high-frequency master clock (abe_mcaspl_ahclkx) signals, and stores the count in XCNT until the next measurement is taken														R	0x00													
23:16	XMAX		0x0 to 0xFF: Transmit clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (abe_mcaspl_ahclkx) signals have been received. If the current counter value is greater than XMAX after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.														RW	0x00													
15:8	XMIN		0x0 to 0xFF: Transmit clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (abe_mcaspl_ahclkx) signals have been received. If XCNT is less than XMIN after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.														RW	0x00													
7:4	RESERVED		Reserved														RW	0x0													
3:0	XPS		Transmit clock check prescaler value 0x0: McASP system clock divided by 1 0x1: McASP system clock divided by 2 0x2: McASP system clock divided by 4 0x3: McASP system clock divided by 8 0x4: McASP system clock divided by 16 0x5: McASP system clock divided by 32 0x6: McASP system clock divided by 64 0x7: McASP system clock divided by 128 0x8: McASP system clock divided by 256 0x9 to 0xF: Reserved														RW	0x0													

**Table 23-632. Register Call Summary for Register MCASP\_TXCLKCHK**

Multichannel Audio Serial Port

- [Transfer Modes: \[0\]](#)
- [Error Reporting: \[1\] \[2\]](#)
- [McASP Registers: \[3\]](#)

**Table 23-633. MCASP\_TXEVTCTL**

<b>Address Offset</b>	0x0000 00CC																														
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .								<b>Instance</b>	McASP																					
<b>Description</b>	Transmitter DMA event control register																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XDATDMA															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	RW	0x0000 0000
0	XDATDMA	Transmit data DMA request enable bit. When writing to this field, always write the default value of 0.  0x0: The transmit data DMA request is enabled. 0x1: The transmit data DMA request is disabled.	RW	0

**Table 23-634. Register Call Summary for Register MCASP\_TXEVTCTL**

Multichannel Audio Serial Port

- [Transfer Modes: \[0\]](#)
- [Data Transmission: \[1\]](#)
- [DMA Requests: \[2\]](#)
- [Operational Modes Configuration: \[3\] \[4\] \[5\]](#)
- [McASP Registers: \[6\]](#)

**Table 23-635. MCASP\_DITCSRAi**

<b>Address Offset</b>	0x0000 0100 + (0x04*i)
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .
<b>Description</b>	DIT left channel status register - All six 32-bit registers (I = 0 to 5) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register file before a different set of data needs to be sent.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITCSRAi																															

Bits	Field Name	Description	Type	Reset
31:0	DITCSRAi	Left (even TDM slot ) channel status	RW	0x0000 0000

**Table 23-636. Register Call Summary for Register MCASP\_DITCSRAI**

Multichannel Audio Serial Port

- [Transfer Modes: \[0\] \[1\]](#)
- [Global Initialization: \[2\]](#)
- [McASP Registers: \[3\]](#)

**Table 23-637. MCASP\_DITCSRBi**

<b>Address Offset</b>	0x0000 0118+ (0x04*i)
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .
<b>Description</b>	DIT right channel status register - All six 32-bit registers (I = 0 to 5) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register file before a different set of data needs to be sent.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITCSRBi																															

Bits	Field Name	Description	Type	Reset
31:0	DITCSRBi	Right (odd TDM slot ) channel status	RW	0x0000 0000

**Table 23-638. Register Call Summary for Register MCASP\_DITCSRBi**

Multichannel Audio Serial Port

- [Transfer Modes: \[0\] \[1\]](#)
- [Global Initialization: \[2\]](#)
- [McASP Registers: \[3\]](#)

**Table 23-639. MCASP\_DITUDRAi**

<b>Address Offset</b>	0x0000 0130 + (0x04*i)	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	DIT left channel user data register - provides the user data of each left channel (even TDM time slot). All six 32-bit registers (I = 0 to 5) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register before a different set of data needs to be sent.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITUDRAi																															

Bits	Field Name	Description	Type	Reset
31:0	DITUDRAi	Left (even TDM slot ) user data	RW	0x0000 0000

**Table 23-640. Register Call Summary for Register MCASP\_DITUDRAi**

Multichannel Audio Serial Port

- [Transfer Modes: \[0\] \[1\]](#)
- [Global Initialization: \[2\]](#)
- [McASP Registers: \[3\]](#)

**Table 23-641. MCASP\_DITUDRBi**

<b>Address Offset</b>	0x0000 0148+ (0x04*i)	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	DIT right user data register - provides the user data of each right channel (odd TDM time slot). All six 32-bit registers (I = 0 to 5) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register before a different set of data needs to be sent.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITUDRBi																															

Bits	Field Name	Description	Type	Reset
31:0	DITUDRBi	Right (odd TDM slot ) user data	RW	0x0000 0000

**Table 23-642. Register Call Summary for Register MCASP\_DITUDRBi**

Multichannel Audio Serial Port

- [Transfer Modes: \[0\] \[1\]](#)
- [Global Initialization: \[2\]](#)
- [McASP Registers: \[3\]](#)

**Table 23-643. MCASP\_XRSRCTL0**

<b>Address Offset</b>	0x0000 0180	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Serializer control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XRDY		RESERVED		SRMOD											

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	RW	0x00000000
4	XRDY	Transmit buffer ready bit Read 0x0: The transmit buffer ( <a href="#">MCASP_TXBUF0</a> ) contains data. Read 0x1: The transmit buffer ( <a href="#">MCASP_TXBUF0</a> ) is empty and needs to be written before the start of the next time slot or a transmit underrun occurs.	R	0
3:2	RESERVED	Reserved	RW	0x0
1:0	SRMOD	Serializer mode bit 0x0:The serializer is inactive 0x1:The serializer is operating in transmit mode. 0x2: Reserved 0x3: Reserved	RW	0x0

**Table 23-644. Register Call Summary for Register MCASP\_XRSRCTL0**

Multichannel Audio Serial Port

- [Serializers: \[0\] \[1\]](#)
- [Transfer Modes: \[2\]](#)
- [Global Initialization: \[3\]](#)
- [McASP Registers: \[4\]](#)

**Table 23-645. MCASP\_TXBUF0**

<b>Address Offset</b>	0x0000 0200	<b>Instance</b>	McASP
<b>Physical Address</b>	See <a href="#">Table 23-590</a> .		
<b>Description</b>	Transmit buffer - The transmit buffer for the serializer holds data from the transmit format unit.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XBUF0																															

Bits	Field Name	Description	Type	Reset
31:0	XBUF0	Transmit buffer	RW	0x0000 0000

**Table 23-646. Register Call Summary for Register MCASP\_TXBUF0**

## Multichannel Audio Serial Port

- Serializers: [0]
- Transmit TDM Sequencer: [1]
- Data Transmission: [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14]
- Error Reporting: [15] [16]
- Operational Modes Configuration: [17] [18] [19]
- McASP Registers: [20] [21] [22]

### 23.9 Serial Low-Power Inter-Chip Media Bus Controller

This section describes the features and functions of the serial low-power inter-chip media bus (SLIMbus®) controller.

#### 23.9.1 SLIMbus Overview

There are two SLIMbus module instantiations, SLIMBUS1 and SLIMBUS2, in the device. SLIMbus1 is implemented in the audio backend (ABE) subsystem and is intended for the connection of external audio peripherals. SLIMBUS2 is at the top-level of the device and is intended for the connection of other types of peripherals.

The SLIMbus controller provides a bidirectional, multidrop, multichannel two-line serial interface between the device and other SLIMbus components in a system, such as audio codecs, Bluetooth® modules, and FM radio receivers/transmitters. Because of its versatility, the SLIMbus controller can play the role of a wide range of peripherals and clocked frame-oriented protocols ( I2S™, PCM, and TDM).

Figure 23-206 shows the SLIMbus1 controller highlight.

Figure 23-206. SLIMbus1 Highlight

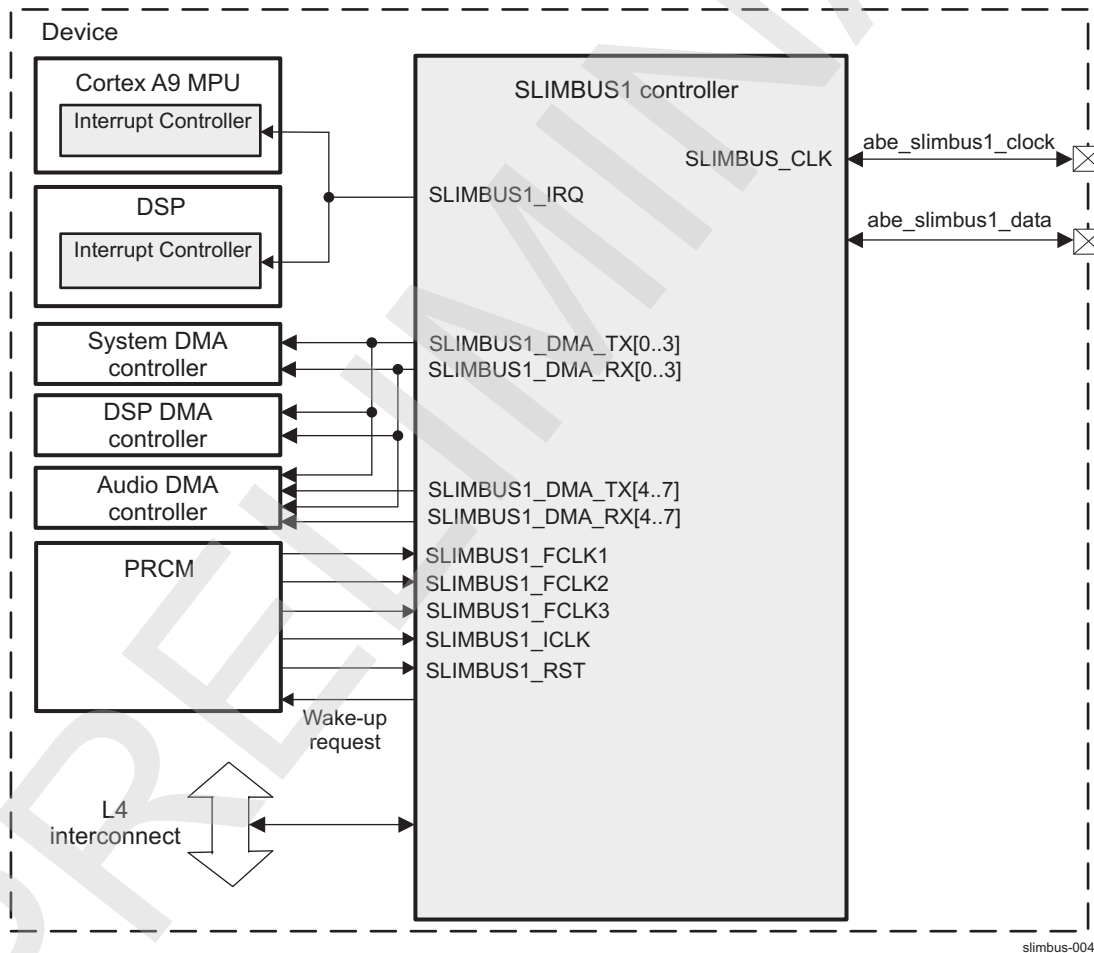
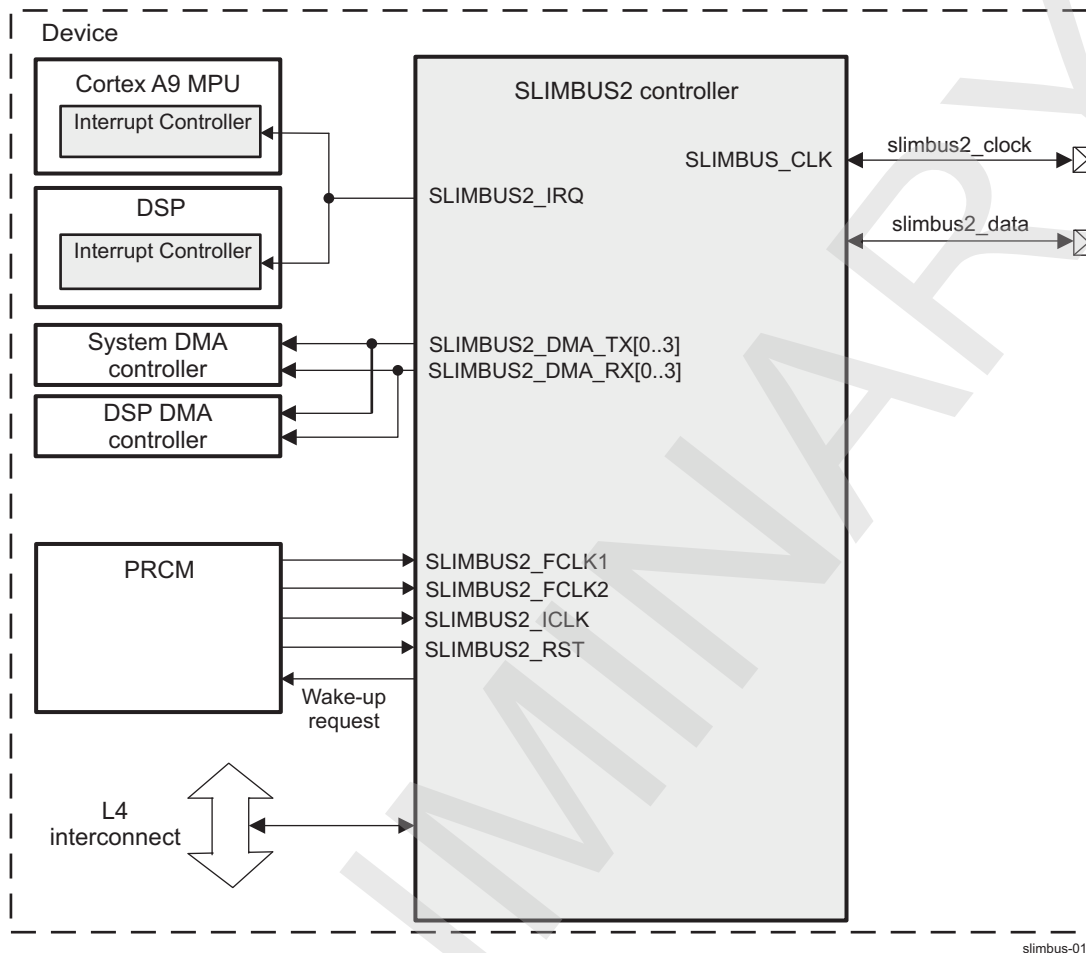


Figure 23-207 shows the SLIMbus2 controller highlight.

**Figure 23-207. SLIMbus2 Highlight**

### 23.9.1.1 Terminology

The SLIMbus module is by default the SLIMbus controller IP. The module implements a SLIMbus component by integrating together several submodules: a SLIMbus component contains a set of devices and the necessary logic needed to provide bus access to these devices.

Every component contains one Interface device and can contain additional devices belonging to other device classes (framer, manager, or generic devices).

- An interface device is the central status and control location of the component. All components have only one interface device.
- A framer device can generate the clock and synchronization information for the entire SLIMbus. Consequently, only one framer device (for all SLIMbus components) at a time can be active on the bus.
- A manager device controls the bus scheduling at high level: It assigns logical addresses to any device that is authorized to communicate using the bus, gives the bandwidth allocation, and so on. Only one manager device at a time can be active on the bus.
- A generic device implements the "useful" functionality of the component by sourcing/sinking the payload data. A component may include several generic devices so that each appears as a separate, standardized entity to the software driver.

**NOTE:** Typically, generic devices are used to transport data (rather than the specialized manager, framer, or interface devices, which typically do not feature data ports).



A device is a separately addressable entity within a SLIMbus component. Each device belongs to a single device class. After enumeration each device is addressed with a single logical address. A device (and especially a generic device) contains data ports to transmit and/or receive data. Each port is identified within the device by its port number (PN). A port is identified within the SLIMbus by its device address (logical address or enumeration address) and PN.

An enumeration address (EA) is a 48-bit value (composed of the manufacturer ID, product code, device ID, and instance value) that allows a device to receive messages.

A logical address (LA) is an 8-bit number used to uniquely identify an enumerated device on the bus.

### 23.9.1.2 Main Features

Each SLIMbus module includes the following main features:

- SLIMbus controller compliant with the MIPI SLIMbus standard version 1.01.00
- Support of seven SLIMbus devices:
  - One Interface device
  - One framer device
  - One manager device
  - Four generic devices
- SLIMBUS1: Eight TX and eight RX independent ports (data channels)
- SLIMBUS2: Four TX and four RX independent ports (data channels)
- FIFO access (one FIFO for transmit and one for receive), with DMA request (one line for transmit port and one line for receive port)
- SLIMBUS1: Message FIFO size is 32 bytes for transmit and 32 bytes for receive.
- SLIMBUS2: Message FIFO size is 32 bytes for transmit and 32 bytes for receive, by default. The message FIFO size can be reprogrammed to 64 (or 16 or 48) bytes if needed. However, typical use never exceeds 32 bytes.
- Data channel FIFO size is programmable for both modules.
- Programmable data length (sample size) from 4 to 32 bits
- Programmable port assignment to either of the four generic devices
- Supported transport protocols: Isochronous, pulled, pushed, asynchronous, and extended asynchronous transport protocols (the latter two in simplex and one-half duplex mode)
- No support for locked protocol
- SLIMbus clock rate: From 0 to 24.576 MHz

Each SLIMbus module has the following characteristics:

- The component is implemented by a digital controller (the current module), plus a physical layer (a set of CLK + DATA bidirectional input/output [I/O] drivers).
- Devices are logical entities implemented by a combination of hardware (automatic address decoding for message reception) and software (message decode, encode, etc.).
- Ports (and their mapping to data channels) are supported by the module hardware-implemented data channel agents. Each agent features a unidirectional FIFO, a DMA request, support for a number of transfer protocols, etc. One agent is used per port and per data direction: Transfer protocols are unidirectional and require only one data channel agent per data channel.

## 23.9.2 SLIMbus Environment

### 23.9.2.1 SLIMbus Component Connection Example

Figure 23-208 shows an example of a system with the SLIMbus1 controller connected to an audio peripheral component.

**Figure 23-208. SLIMbus1 Typical Application System Overview**

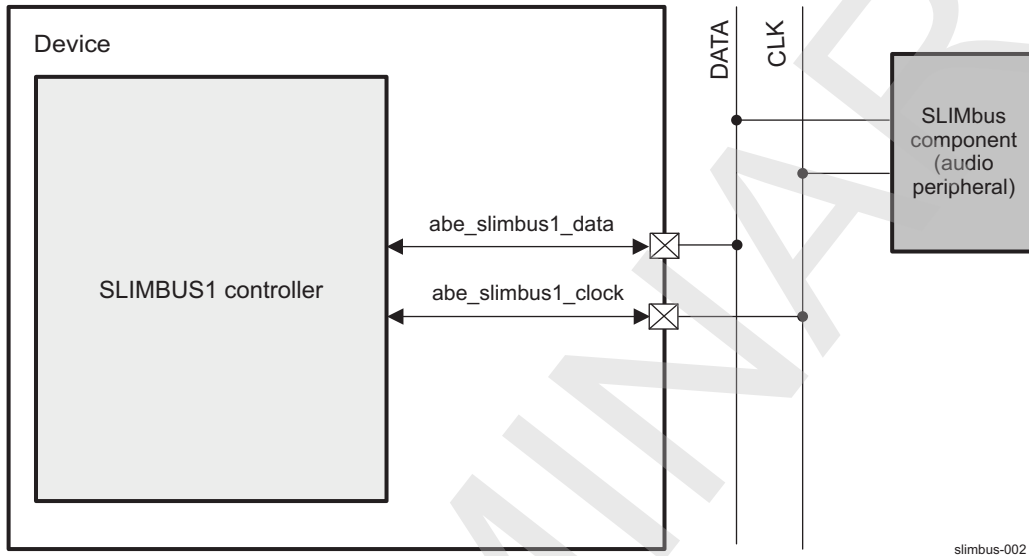
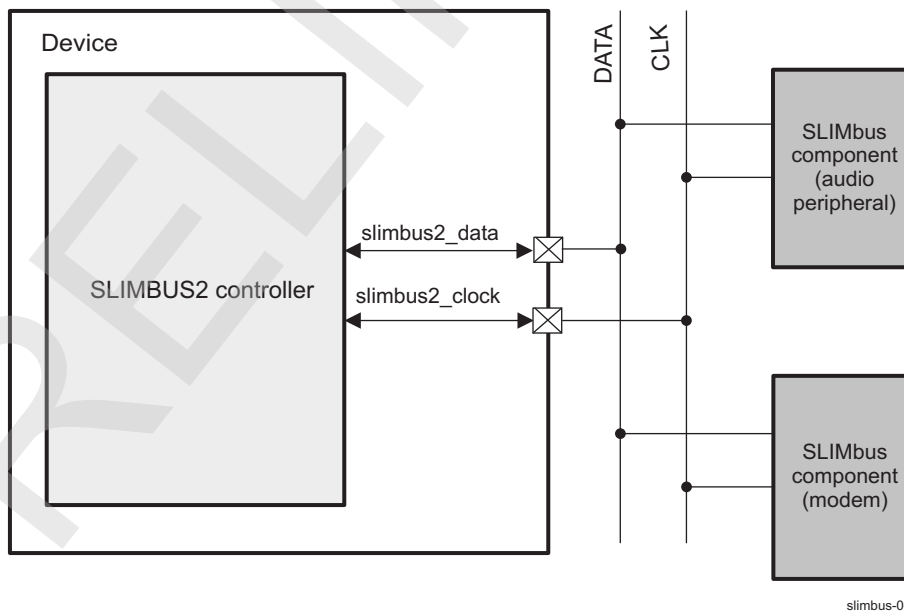


Figure 23-209 shows an example of a system with the SLIMbus2 controller connected to one audio peripheral component and one modem component.

**Figure 23-209. SLIMbus2 Typical Application System Overview**



The physical medium of SLIMbus is composed of two lines: CLK and DATA. All components of the bus are attached to both lines. The clock line (CLK) is driven by the (component containing the) active framer, and received (and used) by all other components on the bus. The data line (DATA) is driven alternately by all components, following the SLIMbus protocol.

The SLIMbus uses non-return-to-zero inverted (NRZI) signalling:

- A logical 1 is represented by a transition (a change) in the physical level.
- A logical 0 is represented by the same physical level as the previous bit.

### 23.9.2.2 SLIMbus Functional Interface

Table 23-647 describes the SLIMbus1 interface.

**Table 23-647. SLIMbus1 Signals Description**

Signal	I/O	Description	Value at Reset
abe_slimbus1_data	I/O	SLIMbus1 DATA pad	1
abe_slimbus1_clock	I/O	SLIMbus1 CLOCK pad	1

Table 23-648 describes the SLIMbus2 interface.

**Table 23-648. SLIMbus2 Signals Description**

Signal	I/O	Description	Value at Reset
slimbus2_data	I/O	SLIMbus2 DATA pad	1
slimbus2_clock	I/O	SLIMbus2 CLOCK pad	1

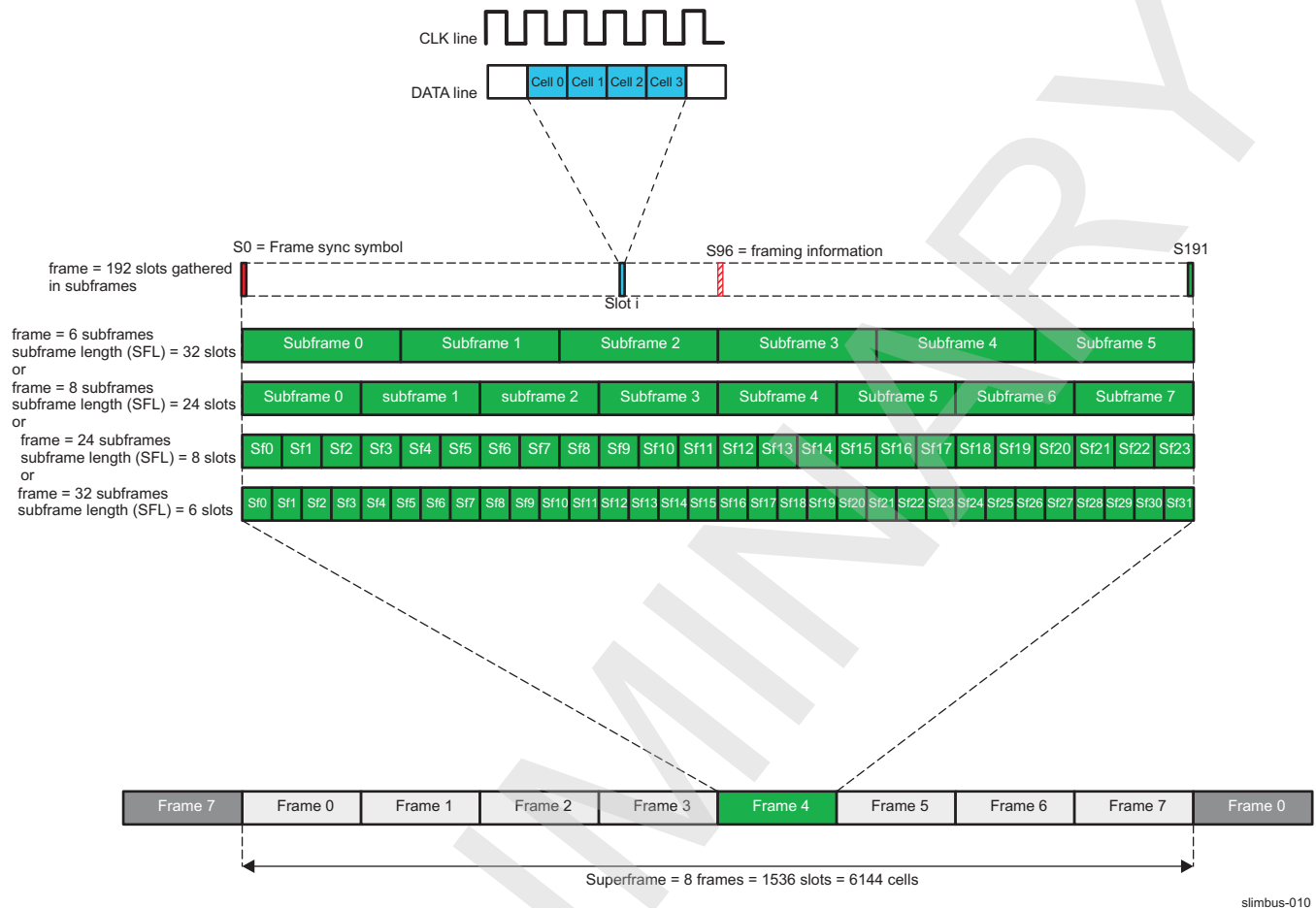
### 23.9.2.3 SLIMbus Protocols and Data Formats

#### 23.9.2.3.1 Frame Structure

The SLIMbus bitstream is transferred in time-division multiplexing (TDM) organization. TDM is a type of multiplexing in which two or more bitstreams are transferred apparently simultaneously as subchannels in one communication channel, but are physically taking turns on the channel. This organization allows SLIMbus to transfer control, data, and configuration information through only one DATA signal. The organization of information on the bus is called a frame structure. The frame structure is composed of five building blocks: cells, slots, subframes, frames and superframes.

- A cell is the smallest division of the SLIMbus data stream. A cell is defined as the region of the DATA line signal that lies between two consecutive positive edges of the CLK signal.
- A slot is a group of four contiguous cells. It is at the slot level that the SLIMbus DATA signal is time division-multiplexed between different channels.
- A frame is a group of 192 contiguous slots (768 cells). The first slot of each frame carries the frame-sync symbol.
- A subframe is the division of the frame structure at which control space and, if present, data space are interleaved. The subframe length can be 6, 8, 24, or 32 slots (24, 32, 96, and 128 cells).
- A superframe is composed of eight consecutive frames and contains a complete set of Framing Information.

Figure 23-210 shows the organization of the frame structure.

**Figure 23-210. SLIMbus Frame Structure Overview**

### 23.9.2.3.2 Control and Data Channels

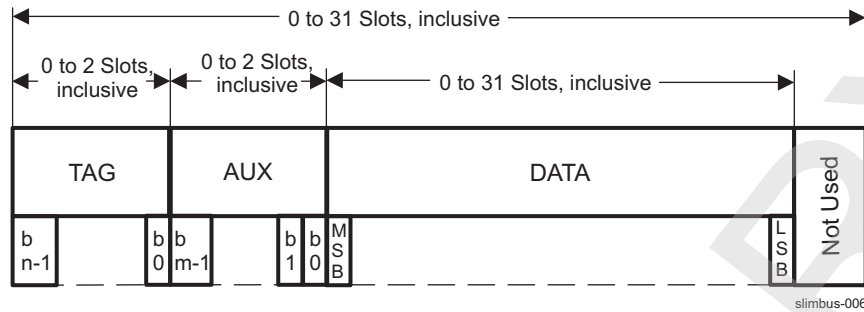
A frame is divided into two parts: the control space and the data space. The control space contains the information needed for a device to discover the bus state, to track messages, and for the messages themselves. The control space size is programmable by setting the control space width (number of control space slots per subframe) and the subframe length. Data space is interleaved with control space at the subframe level. Slots not explicitly allocated to control space are considered data space. The active manager can organize these slots into data channels to transport application-specific data streams between devices on the bus. Data space can contain up to 256 data channels.

A data channel is a stream of one or more contiguous slots organized in a consistent data structure that repeats at a fixed interval. The largest possible group of contiguous slots within a data channel is known as a segment.

A data channel connects one data source port and one or several data sink ports, depending on the protocol used. A port can be connected only to a single data channel at a time.

A segment is composed of three fields: TAG, auxiliary information (AUX), and data fields. These fields are organized within the segment, as shown in [Figure 23-211](#). The exact composition of the segment depends on the transport protocol.

Figure 23-211. Segment Organization



### 23.9.2.3.3 SLIMbus Transfer Protocols

The SLIMbus module supports seven the following protocols:

- Isochronous
- Pushed
- Pulled
- Asynchronous
  - Simplex
  - Half-duplex
- Extended asynchronous
  - Simplex
  - Half-duplex

Table 23-649. SLIMbus Transfer Protocols

Transfer Protocol	Multicast Capability
Isochronous transfer protocol	Yes
Pushed transfer protocol	Yes
Pulled transfer protocol	No
Asynchronous transfer protocols	No
Extended asynchronous transfer protocols	No

#### 23.9.2.3.3.1 Isochronous Transfer Protocol

The isochronous transfer protocol is a unidirectional, streaming, multicast communication mode. Data is to be present in all allocated segments (this transfer protocol has no TAG field).

- 100 percent of the bandwidth is usable for AUX + DATA slots (no control TAG overhead).
- Because no flow control exists, multiple data sinks may be connected to the same data source (multicast capability).
- The range of possible presence rates (number of segments per second) is limited to simple ratios of the SLIMbus clock. If several data streams with incompatible presence rates must be multiplexed over the same bus, no combination of isochronous-based schedulings works. In those cases, the pushed or pulled transfer protocol must be used.

#### 23.9.2.3.3.2 Pushed Transfer Protocol

The pushed transfer protocol is unidirectional, multicast, and controlled by the data source (transmitter), which pushes data toward the data sink (receiver). The presence of valid data in a segment is indicated by the presence (P) bit in the TAG field, driven by the source. The TAG field is composed of four bits: one reserved, one strobe (STR) bit, one reserved, and one presesnce (P) bit.

- The pushed data source is the master of the data flow through the presence (P) bit .
- The pushed data sink is slaved to the presence (P) bit..

- The TAG control field creates a 1-slot overhead per segment. The rest of the segment is usable for AUX + DATA slots.
- Because the flow control is generated by the source with no feedback from the sink, multiple data sinks may be connected to that source (multicast capability).
- The presence rate can be modulated at will from 0 (no data at all) to the full segment rate (data present in all segments) according to the requirement. Intermediate settings allow several streams with incompatible presence rates to be multiplexed over the same bus, at the cost of less than 100 percent bandwidth utilization (for example, a 44.1-kHz sample flow transmitted over a reserved 48-kHz data channel).

### 23.9.2.3.3.3 Pulled Transfer Protocol

The pulled transfer protocol is unidirectional, unicast, and controlled by the data sink (receiver), which pulls data from the data transmitter (source). The presence of valid data in a segment is indicated by the presence bit (P) in the TAG field, driven by the source. The TAG field is composed of four bits: one reserved, one sample request (SRQ) bit, one reserved, and one presence (P) bit.

- The pulled data sink is the master of the data flow through the (SRQ bit, itself determining the value of the presence bit (P), driven by the data source.
- The pulled data source is slaved to the SRQ bit.
- The TAG control field creates a 1-slot overhead per segment. The rest of the segment is usable for AUX + DATA slots.
- Because the flow control is generated by the sink with feedback from the source, only a single data sink may be connected to that source (unicast-only, no multicast capability).
- The presence rate can be modulated at will from 0 (no data at all) to the full segment rate (data present in all segments) according to the requirement. Intermediate settings allow several streams with incompatible presence rates to be multiplexed over the same bus, at the cost of less than 100 percent bandwidth utilization (for example, a 44.1-kHz sample flow transmitted over a reserved 48-kHz data channel).

### 23.9.2.3.3.4 Asynchronous Transfer Protocols

The asynchronous transfer protocols are typically used for on-demand, packet-based data transfers (that is, nonstreaming). Because the SLIMbus protocol and architecture are optimized for data streaming scenarios, asynchronous protocols are considered a useful extension rather than a central feature of the SLIMbus. An asynchronous data channel gets the same periodic, reserved, fixed-bandwidth slot allocation as a streaming (for example, isochronous) data channel.

An asynchronous data channel is a point-to-point (unicast) communication link between a primary owner device and a secondary owner device.

A half-duplex asynchronous channel is a bidirectional data channel.

- A handshake protocol is used to determine the direction of the channel through its ownership. The channel owner is the data source, and the other is the sink.
- Once the ownership (data direction) is determined, both source and sink can regulate the flow of data, depending on data availability (source) and free buffer availability (sink).
- Half-duplex ports can be used when bandwidth is sparse (compare simplex and full duplex).
- However, each half-duplex port requires two data channel agents (with FIFO and control logic) inside the controller, one in each direction, contrary to all other asynchronous modes and transfer protocols. Those two associated agents must be configured coherently.

A simplex asynchronous channel is a unidirectional data channel, a simplified version of the half-duplex channel defined perviously. The ownership is never given to the secondary owner.

- Secondary owner never requests the channel. As a consequence, the roles are never exchanged: the primary owner is always data source, and the secondary owner always data sink.
- Simplex ports are used in pairs of opposite directions to implement full-duplex communication, where each direction has its own reserved bandwidth. As far as the controller is concerned, the two simplex links are treated as fully independent data channels, with independent configurations; only the higher layers (software) associate the two.

- Two simplex ports use the same number of data channel agents (that is, FIFOs and control logic) as a single half-duplex port (two per controller in each case; the only difference is in the bandwidth management).

#### **23.9.2.3.3.5 Extended Asynchronous Transfer Protocols**

The extended asynchronous transfer protocols are extensions of the plain asynchronous transfer protocol. The main features are retained: point-to-point (unicast) between a primary and a secondary owner, simplex (unidirectional) or half-duplex (bidirectional), and so on.

The difference lies in a more flexible management of the DATA slots: In the plain asynchronous TP, each segment contains no data (presence [P] bit = 0) or a fixed amount of data ( $P = 1$ ), preconfigured for all segments. In the extended asynchronous mode, a number of valid bytes (NVB) 4-bit field is added before each segment, in a second TAG slot, to indicate a variable number of significant DATA slots (the granularity is 1 byte = 2 slots).

---

**NOTE:** AUX slots (when present; not in the current implementation) are not counted in NVB, just as they are not counted in DL in the plain asynchronous TP.

---

#### **23.9.2.3.4 Half-Duplex Versus Full-Duplex Asynchronous Configurations**

Half-duplex and full-duplex modes are similar in their operation, in that they use two data channel agents (FIFOs and associated control hardware) in opposite directions (one TX, one RX) to transmit and receive data over a bidirectional, unicast (point-to-point) asynchronous link. However, there are differences:

- The half-duplex setup uses a single data channel (in the SLIMbus sense); that is, a single reserved data space segment allocation, used alternately in both directions, according to the dynamic ownership negotiated in the TAG slots.
- The full-duplex setup uses two data channels (in the SLIMbus sense); that is, two separate reserved data space segment allocations, each used in a fixed direction (opposite of the other) (that is, with no ownership change).

### 23.9.3 SLIMbus Integration

Figure 23-212 and Figure 23-213 show the integration of the two SLIMbus module instantiations in the device. The SLIMBUS1 module is part of the ABE module

Figure 23-212. SLIMBUS1 Integration

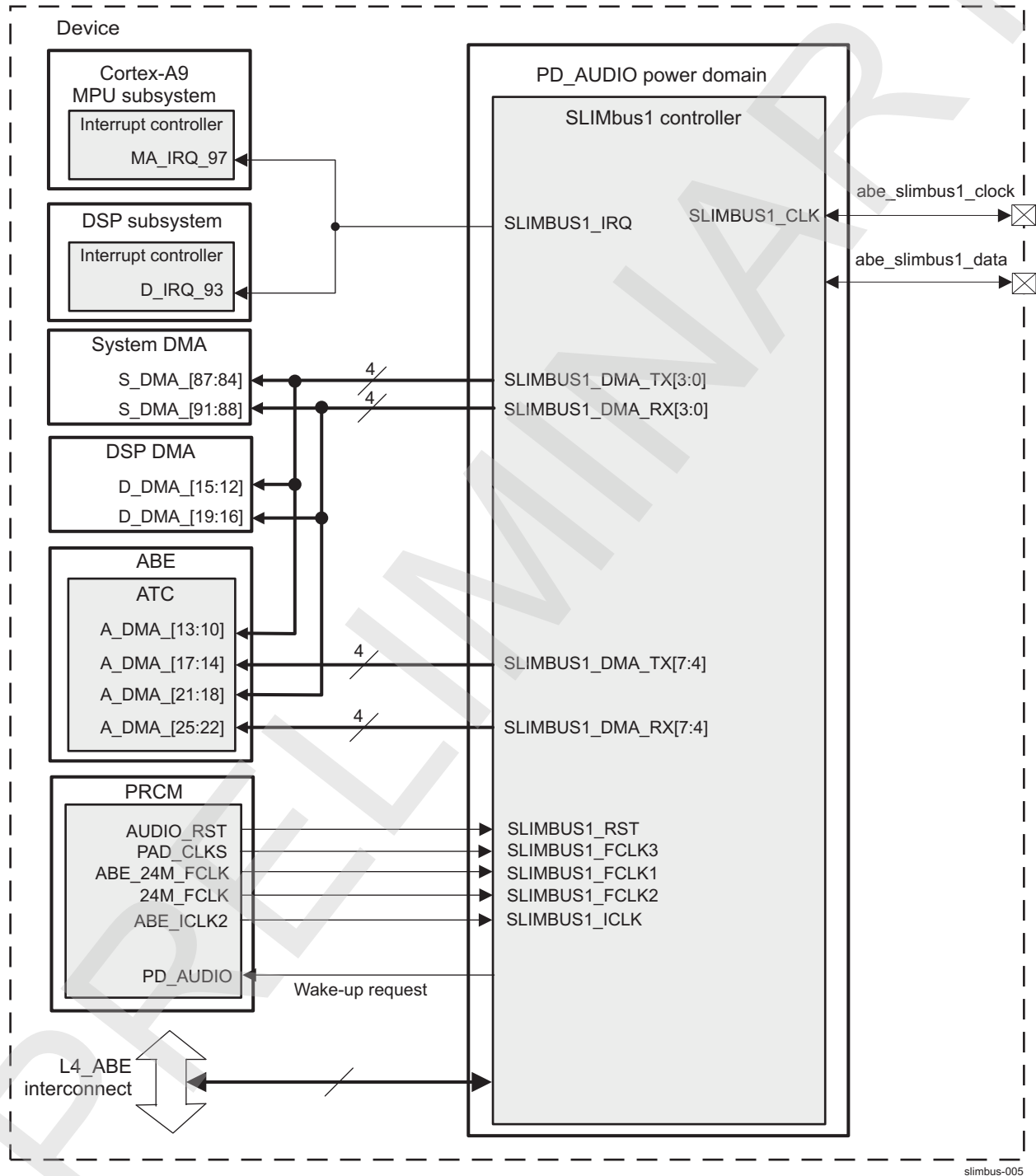
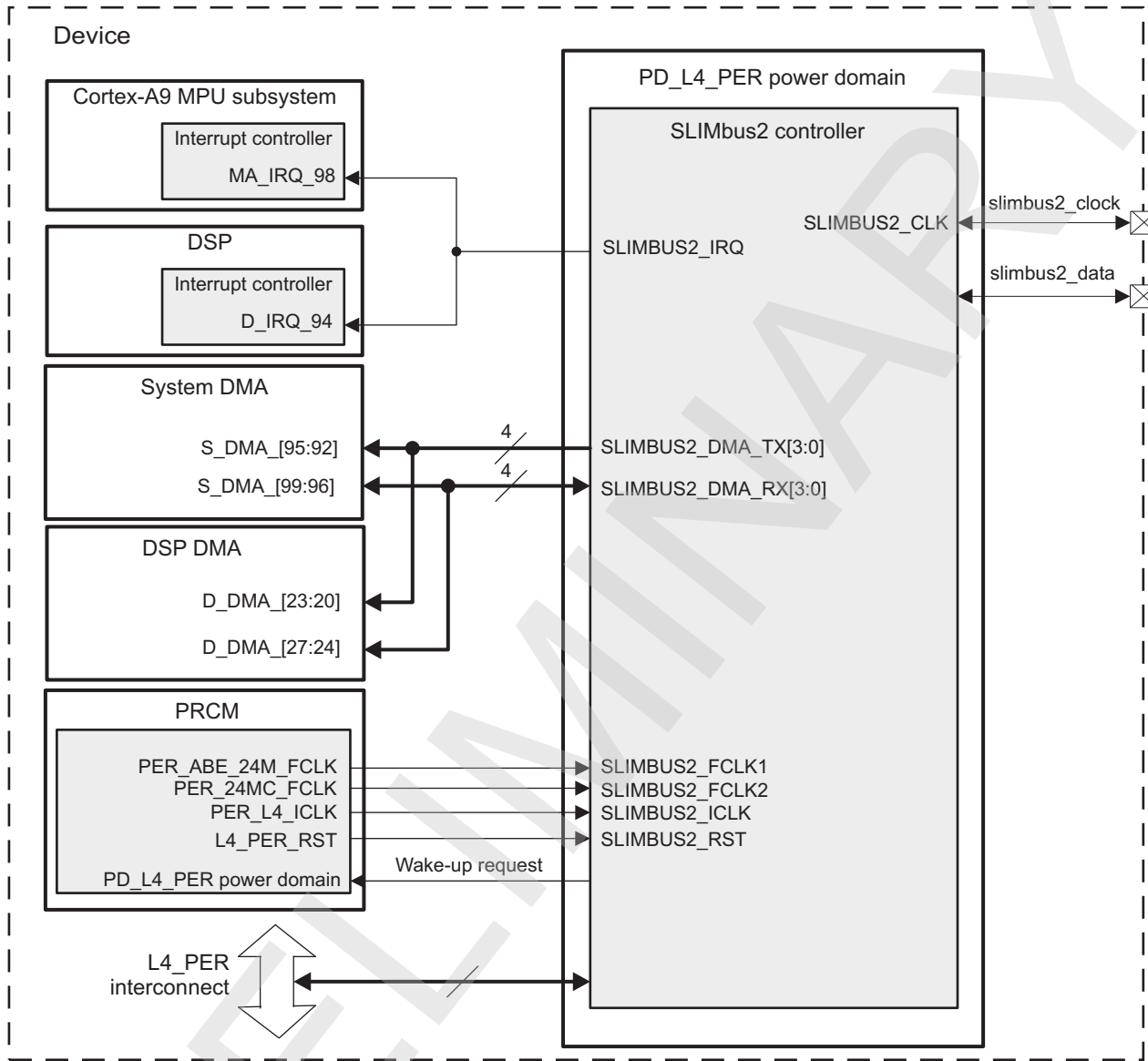




Figure 23-213. SLIMBUS2 Integration



slimbus-003

Table 23-650 through Table 23-652 summarize the integration of the modules in the device.

Table 23-650. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
SLIMBUS1	PD_AUDIO	Yes	L4_ABE
SLIMBUS2	PD_L4_PER	Yes	L4_PER

**Table 23-651. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SLIMBUS1	SLIMBUS1_FCLK3	PAD_CLKS	PRCM	Functional clock, divided by 4 or more inside SLIMbus module. The source is the abe_clks pin gated by the PRCM module.
	SLIMBUS1_FCLK1	ABE_24M_FCLK	PRCM	Functional clock, divided by 4 or more inside SLIMbus module
	SLIMBUS1_ICLK	ABE_ICLK2	PRCM	Interface clock
	SLIMBUS1_FCLK2	24M_FCLK	PRCM	Functional clock, divided by 4 or more inside SLIMbus module
	SLIMBUS1_CLK	SLIMBUS_UCLKS	abe_slimbus1_clock pin	SLIMBUS clock, may or may not be locally generated
SLIMBUS2	SLIMBUS2_ICLK	PER_L4_ICLK	PRCM	Interface clock
	SLIMBUS2_CLK	PAD_SLIMBUS_CLKS	slimbus2_clock pin	SLIMBUS clock, may or may not be locally generated
	SLIMBUS2_FCLK2	PER_24MC_FCLK	PRCM	Functional clock, divided by 4 or more inside SLIMbus module
	SLIMBUS2_FCLK1	PER_ABE_24M_FCLK	PRCM	Functional clock, divided by 4 or more inside SLIMbus module
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
SLIMBUS1	SLIMBUS1_RST	AUDIO_RST	PRCM	Reset to SLIMbus1 module
SLIMBUS2	SLIMBUS2_RST	L4_PER_RST	PRCM	Reset to SLIMbus2 module

**Table 23-652. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
SLIMBUS1	SLIMBUS1_IRQ	MA_IRQ_97	Cortex-A9 INTC	SLIMBUS1 interrupt to Cortex-A9 MPU
	SLIMBUS1_IRQ	D_IRQ_93	DSP INTC	SLIMBUS1 interrupt to DSP
SLIMBUS2	SLIMBUS2_IRQ	MA_IRQ_98	Cortex-A9 INTC	SLIMBUS2 interrupt to Cortex-A9 MPU
	SLIMBUS2_IRQ	D_IRQ_94	DSP INTC	SLIMBUS2 interrupt to DSP
DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
SLIMBUS1	SLIMBUS1_DMA_TX0	A_DMA_10	ADMA (Audio DMA)	SLIMbus module 1 – transmit request channel 0
	SLIMBUS1_DMA_TX1	A_DMA_11	ADMA	SLIMbus module 1 – transmit request channel 1
	SLIMBUS1_DMA_TX2	A_DMA_12	ADMA	SLIMbus module 1 – transmit request channel 2
	SLIMBUS1_DMA_TX3	A_DMA_13	ADMA	SLIMbus module 1 – transmit request channel 3
	SLIMBUS1_DMA_TX4	A_DMA_14	ADMA	SLIMbus module 1 – transmit request channel 4
	SLIMBUS1_DMA_TX5	A_DMA_15	ADMA	SLIMbus module 1 – transmit request channel 5
	SLIMBUS1_DMA_TX6	A_DMA_16	ADMA	SLIMbus module 1 – transmit request channel 6
	SLIMBUS1_DMA_TX7	A_DMA_17	ADMA	SLIMbus module 1 – transmit request channel 7
	SLIMBUS1_DMA_RX0	A_DMA_18	ADMA	SLIMbus module 1 – receive request channel 0

**Table 23-652. Hardware Requests (continued)**

	SLIMBUS1_DM A_RX1	A_DMA_19	ADMA	SLIMbus module 1 – receive request channel 1
	SLIMBUS1_DM A_RX2	A_DMA_20	ADMA	SLIMbus module 1 – receive request channel 2
	SLIMBUS1_DM A_RX3	A_DMA_21	ADMA	SLIMbus module 1 – receive request channel 3
	SLIMBUS1_DM A_RX4	A_DMA_22	ADMA	SLIMbus module 1 – receive request channel 4
	SLIMBUS1_DM A_RX5	A_DMA_23	ADMA	SLIMbus module 1 – receive request channel 5
	SLIMBUS1_DM A_RX6	A_DMA_24	ADMA	SLIMbus module 1 – receive request channel 6
	SLIMBUS1_DM A_RX7	A_DMA_25	ADMA	SLIMbus module 1 – receive request channel 7
	SLIMBUS1_DM A_TX0	S_DMA_84	sDMA (System DMA)	SLIMbus module 1 – transmit request channel 0
	SLIMBUS1_DM A_TX1	S_DMA_85	sDMA	SLIMbus module 1 – transmit request channel 1
	SLIMBUS1_DM A_TX2	S_DMA_86	sDMA	SLIMbus module 1 – transmit request channel 2
	SLIMBUS1_DM A_TX3	S_DMA_87	sDMA	SLIMbus module 1 – transmit request channel 3
	SLIMBUS1_DM A_RX0	S_DMA_88	sDMA	SLIMbus module 1 – receive request channel 0
	SLIMBUS1_DM A_RX1	S_DMA_89	sDMA	SLIMbus module 1 – receive request channel 1
	SLIMBUS1_DM A_RX2	S_DMA_90	sDMA	SLIMbus module 1 – receive request channel 2
	SLIMBUS1_DM A_RX3	S_DMA_91	sDMA	SLIMbus module 1 – receive request channel 3
	SLIMBUS1_DM A_TX0	D_DMA_12	DDMA (DSP DMA)	SLIMbus module 1 – transmit request channel 0
	SLIMBUS1_DM A_TX1	D_DMA_13	DDMA	SLIMbus module 1 – transmit request channel 1
	SLIMBUS1_DM A_TX2	D_DMA_14	DDMA	SLIMbus module 1 – transmit request channel 2
	SLIMBUS1_DM A_TX3	D_DMA_15	DDMA	SLIMbus module 1 – transmit request channel 3
	SLIMBUS1_DM A_RX0	D_DMA_16	DDMA	SLIMbus module 1 – receive request channel 0
	SLIMBUS1_DM A_RX1	D_DMA_17	DDMA	SLIMbus module 1 – receive request channel 1
	SLIMBUS1_DM A_RX2	D_DMA_18	DDMA	SLIMbus module 1 – receive request channel 2
	SLIMBUS1_DM A_RX3	D_DMA_19	DDMA	SLIMbus module 1 – receive request channel 3
SLIMBUS2	SLIMBUS2_DM A_TX0	D_DMA_20	DDMA	SLIMbus module 2 – transmit request channel 0
	SLIMBUS2_DM A_TX1	D_DMA_21	DDMA	SLIMbus module 2 – transmit request channel 1
	SLIMBUS2_DM A_TX2	D_DMA_22	DDMA	SLIMbus module 2 – transmit request channel 2
	SLIMBUS2_DM A_TX3	D_DMA_23	DDMA	SLIMbus module 2 – transmit request channel 3
	SLIMBUS2_DM A_RX0	D_DMA_24	DDMA	SLIMbus module 2 – receive request channel 0

**Table 23-652. Hardware Requests (continued)**

SLIMBUS2_DM A_RX1	D_DMA_25	DDMA	SLIMbus module 2 – receive request channel 1
SLIMBUS2_DM A_RX2	D_DMA_26	DDMA	SLIMbus module 2 – receive request channel 2
SLIMBUS2_DM A_RX3	D_DMA_27	DDMA	SLIMbus module 2 – receive request channel 3
SLIMBUS2_DM A_TX0	S_DMA_92	sDMA	SLIMbus module 2 – transmit request channel 0
SLIMBUS2_DM A_TX1	S_DMA_93	sDMA	SLIMbus module 2 – transmit request channel 1
SLIMBUS2_DM A_TX2	S_DMA_94	sDMA	SLIMbus module 2 – transmit request channel 2
SLIMBUS2_DM A_TX3	S_DMA_95	sDMA	SLIMbus module 2 – transmit request channel 3
SLIMBUS2_DM A_RX0	S_DMA_96	sDMA	SLIMbus module 2 – receive request channel 0
SLIMBUS2_DM A_RX1	S_DMA_97	sDMA	SLIMbus module 2 – receive request channel 1
SLIMBUS2_DM A_RX2	S_DMA_98	sDMA	SLIMbus module 2 – receive request channel 2
SLIMBUS2_DM A_RX3	S_DMA_99	sDMA	SLIMbus module 2 – receive request channel 3

**NOTE:**

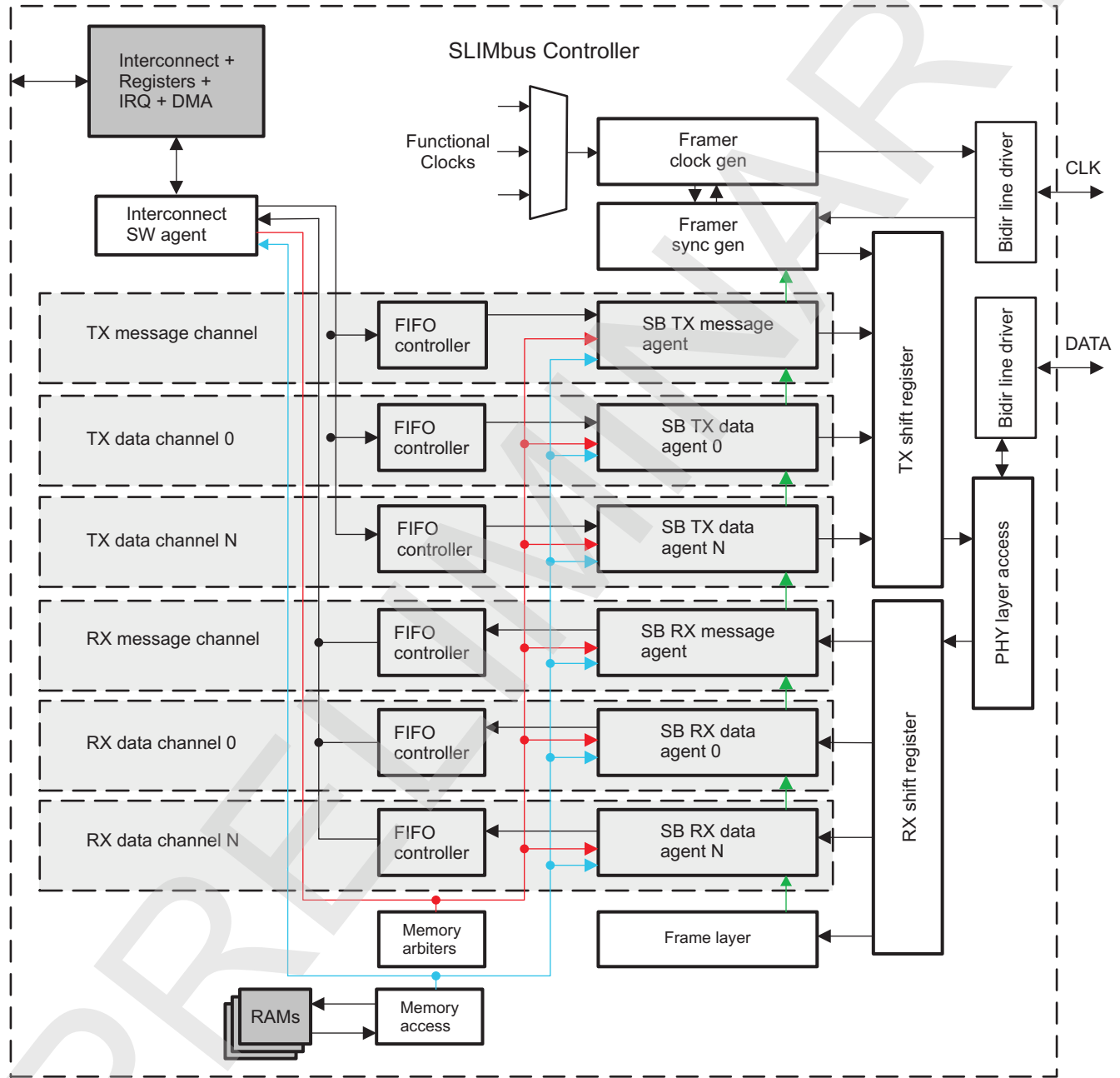
- For the description of the interrupt, see [Section 23.9.4.12, Interrupt Service Routines](#), and [Chapter 17, Interrupt Controllers](#).
- For the description of the DMA, see [Section 23.9.4.5.2, Data Input/Output: DMA Request Mode](#), and [Chapter 16, SDMA](#).

### 23.9.4 SLIMbus Functional Description

#### 23.9.4.1 SLIMbus Block Description

Figure 23-214 shows the the overall architecture of a SLIMbus module.

Figure 23-214. SLIMbus Block Diagram



- The red arrows represent the shared memory arbitration interface, from the arbiter to each agent.
- The blue arrows represent the shared memory access interface, from the memory access module to each agent.
- The green arrows represent the frame layer sync information broadcasted to all SB-side agents.

### 23.9.4.2 SLIMbus Clock Configuration

Most of the SLIMbus module runs off two main clocks: the L4 interface clock for the data input and output registers, and the control and status control registers; and the SLIMbus clock, taken from the serial interface (CLK line) for the SLIMbus-side logic.

The SLIMbus controller operates as clock source component (active framer), which drives the SLIMbus clock line CLK, or as clock receiver component, which gets its clock from the same CLK line.

- As a clock receiver, the module has by definition no control over the clock activity and can perform required operations based on that. Even clock pause is mostly transparent as far as the receiver logic is concerned (that is, everything happens as if time had stopped at the clock pause, and activity resumes seamlessly upon clock restart).
- As a clock source (also called active framer), the module relies on a local functional clock to drive the SLIMbus clock line. Such a functional clock is selected out of the available clock inputs. The appropriate input is expected to be active whenever it is needed (that is, when selected and used to run the SLIMbus clock and/or the module internal logic). In this case, the SLIMbus clock output is generated by selecting and dividing one functional clock input provided to the controller. Internal logic selects and divides the appropriate one depending on the SLIMbus configuration.

### 23.9.4.3 Software Reset

The [SLIMBUS\\_CMP\\_SYSCONFIG\[1\] SOFTRESET](#) bit controls the software reset; writing 1 to this bit starts a soft reset sequence.

### 23.9.4.4 System Power Management and Wakeup

#### 23.9.4.4.1 Wake-Up Request

Both SLIMbus controllers are wake-up capable.

There are several levels of wake-up capability:

- Most IRQ events are potential wake-up sources, because they can occur while the external interface is idle, assuming the external and SLIMbus clocks are still running.
- The clock restart IRQ event (the [SLIMBUS\\_FR\\_INFO\[5\] CLOCK\\_RESTART](#) bit) following a clock pause is the only one that can take place when all module clocks are stopped.

#### 23.9.4.4.2 Power Management

[Table 23-653](#) describes power-management features available for the SLIMbus module.

---

**NOTE:** For the description of the IdleMode feature, see [Chapter 3, Power, Reset, and Clock Management](#).

---

**Table 23-653. Local Power-Management Features**

Feature	Registers	Description
Clock auto gating	<a href="#">SLIMBUS_CMP_SYSCONFIG[8] AUTOGATINGDISABLE</a>	Internal clock autogating is intended to conserve power when the module is not in use.
Slave idle modes	<a href="#">SLIMBUS_CMP_SYSCONFIG[3:2] IDLEMODE</a>	Force-idle, no-idle, smart-idle, and smart-idle wake-up-capable modes are available.
Wake-up sources enable	<a href="#">SLIMBUS_FR_INFO[5] CLOCK_RESTART</a> and most IRQ events	See <a href="#">Section 23.9.4.4.1</a>

#### 23.9.4.4.3 SLIMbus Boot and Reset Procedures

The SLIMbus standard defines different levels of resets and default states from which to boot. They are implemented by a mix of software and hardware.

**NOTE:** This does not include the module-wide software or hardware resets of the controller, which simply wipe off the entire logic with no consideration for any potentially ongoing SLIMbus operation.

Table 23-654 summarizes the control bit fields to set in the various reset and/or boot sequences. Any combination other than the ones described here may yield unpredictable results.

**Table 23-654. Reset and Boot Control Summary**

Register / Field	SLIMBUS_FR_CONTROL		SLIMBUS_FL_CONTROL		
	Boot	bus_reset	Boot	bus_shutdown	force_reconfigure <sup>(1)</sup>
Clock receiver boot <sup>(2)</sup>			1		1
Clock source boot	1				1
Clock receiver component reset <sup>(2) (3)</sup>			1		1
Clock source component reset		1			1
Bus reset <sup>(4) (5)</sup>		1			
Bus shutdown <sup>(4)</sup>				1	

- (1) When used, the SLIMBUS\_FL\_CONTROL[2] FORCE\_RECONFIGURE bit must be set (written to 1) last, because its action can be immediate. Other control bits can be set in any order.
- (2) The controls for boot and component reset look identical for the clock receiver. The difference is in the initial state of the component: in the first case, the component is in the undefined state (not started), in the second case it is operational.
- (3) The clock receiver component reset case shown here corresponds to the reception by the component interface of the RESET\_DEVICE. In the two other cases, a bus reset is detected and the hardware automatically goes to the correct state.
- (4) The bus reset and shutdown belong to a reconfiguration (message) sequence and take place on the reconfiguration boundary: the SLIMBUS\_FL\_CONTROL[2] FORCE\_RECONFIGURE is therefore not used.
- (5) The bus reset is executed only by the clock source (active framer).

**23.9.4.4.3.1 Clock Receiver Component Boot**

A clock receiver boot sequence is initiated when a component that is not the default active framer decides to join the SLIMbus. Provided the clock line is driven (that is, a framer is active), the clock receiver must eventually acquire the synchronization.

Clock receiver boot sequence:

- Initial state: Out of hardware module reset or following a SLIMbus shutdown
- Write 1 to the SLIMBUS\_FL\_CONTROL[0] BOOT bit to start the clock receiver.
- Write 1 to the SLIMBUS\_FL\_CONTROL[2] FORCE\_RECONFIGURE bit.

**23.9.4.4.3.2 Clock Source Component Boot**

Source component is, by definition, the component including the active framer (that is, the clock driver). A clock source boot sequence is initiated when a component that is the default active framer joins the SLIMbus. The clock source drives the clock line and runs the SLIMbus boot sequence, allowing the connected clock receivers to synchronize to it.

**NOTE:** The component knows in advance that it is the only default active framer, in charge of booting the bus: That information must be available for the component, as well as the default SM, CG, and RF configuration parameters.

Clock source boot sequence:

- Initial state: The component is the designated default active framer and is out of hardware module reset or follows a SLIMbus shutdown.
- Write 1 to the SLIMBUS\_FR\_CONTROL[0] BOOT bit.
- Write the basic bus configuration, to be used by the active framer device to generate the clock and the framing channel contents:
  - Subframe mode (M) in the SLIMBUS\_FL\_SM[4:0] SM bit field: Sets the control space width and subframe length

- Clock gear (G) in the [SLIMBUS\\_FL\\_CG\[3:0\]](#) CG bit field: The root/SLIMbus frequency ratio is defined as  $2^{(10-CG)}$
- Root frequency (F) in the [SLIMBUS\\_FL\\_RF\[3:0\]](#) RF bit field: Sets the root frequency
- Clock source selection, in the [SLIMBUS\\_FR\\_CLOCK\\_SOURCE\[5:3\]](#) CLKSEL and [SLIMBUS\\_FR\\_CLOCK\\_SOURCE\[2:0\]](#) CLKDIV bit fields
- The relationships (and limitations) between CG, RF, and the clock source selection also apply here.
- Write 1 to the [SLIMBUS\\_FL\\_CONTROL\[2\]](#) FORCE\_RECONFIGURE bit to trigger the sequence execution
- The clock source starts immediately and eventually drives the CLK line and allows other components (clock receivers) to join the bus.

#### 23.9.4.4.3.3 Bus Reset

A bus reset is a specific DATA and CLK sequence run by the active framer, scheduled after the bus has already booted, and which causes all components of the bus to reset. For more information about the bus reset detection mechanism, see [Section 23.9.4.4.3.4, Component Reset](#).

The bus reset is requested by the manager device through a reconfiguration sequence, and is run by the active framer starting at the following reconfiguration boundary.

Bus reset sequence, to be executed by all components:

- Write 1 to the [SLIMBUS\\_FR\\_CONTROL\[1\]](#) BUS\_RESET bit.
- Wait for the end of the reconfiguration sequence to execute:
  - On the clock source (active framer), this must trigger a bus reset sequence.
  - On any other device (that is, a clock receiver), this must have no direct effect (other than clearing the [SLIMBUS\\_FR\\_CONTROL\[1\]](#) BUS\_RESET bit).
  - Because all bus devices are being reset (bus is rebooted, almost from the beginning), the configuration (M, G, F parameters) can change (because those parameters must be reacquired by all attached devices as part of the resynchronization sequence, this is not a problem).

#### 23.9.4.4.3.4 Component Reset

A component reset implies the (device) reset of all devices within that component. When a component reset condition is detected, software must therefore run the device reset sequence defined in [Section 23.9.4.4.3.5, Device Reset](#), as many times as necessary. Additional operations are also required.

A component reset has two possible causes:

- Bus reset sequence detected
- Interface device of the local component receives a reset order. (This is not a reconfiguration message; that is, the reset is effective immediately upon reception. Nevertheless, a specific action is needed to perform the device reset:)
  - Use the [SLIMBUS\\_FL\\_CONTROL\[1\]](#) COMPONENT\_RESET bit.
  - If the component is a clock receiver (clock receiver FSM), the sequence is :
    - Write 1 to the [SLIMBUS\\_FL\\_CONTROL\[0\]](#) BOOT bit.
    - Write 1 to the [SLIMBUS\\_FL\\_CONTROL\[2\]](#) FORCE\_RECONFIGURE bit to trigger the reset execution.
  - If the component is the active framer (clock source FSM), the sequence is :
    - Write 1 to the [SLIMBUS\\_FR\\_CONTROL\[1\]](#) BUS\_RESET bit.
    - Write 1 to the [SLIMBUS\\_FL\\_CONTROL\[2\]](#) FORCE\_RECONFIGURE bit to trigger the reset execution.
    - The effect of a clock source component reset is the same as a bus reset, except it takes place immediately instead of waiting for a reconfiguration boundary.

#### 23.9.4.4.3.5 Device Reset

A device reset can have several causes:



- Component reset of the receiver component containing that device, caused by a bus reset reconfiguration message. If the cause of the device reset is a component reset, the sequence defined in [Section 23.9.4.4.3.4, Component Reset](#), is applied.
- A reset order targeted at the device itself. (This is not a reconfiguration message; that is, it is effective immediately upon reception. In this case, no additional action is required.)

The following sequence must be executed for all devices:

- Release the logical address (LA) of the device by clearing (writing 0 to) the [SLIMBUS\\_DEV\\_LA\\_i\[8\]](#) LA\_ENABLE bit. It is assumed that the [SLIMBUS\\_DEV\\_LA\\_i\[9\]](#) EA\_ENABLE bit remains set (1). The device returns to the default, nonenumerated state where it is identified by its hard-coded enumeration address (EA). The [SLIMBUS\\_DEV\\_LA\\_i\[7:0\]](#) LA bit may be assigned to another device by the bus manager (that is, keeping track of all device resets).
- The following hardware bits must also be cleared (by writing 1) when the device to which they belong (identified by the message target LA) is being reset:
  - The [SLIMBUS\\_FR\\_INFO\[2\]](#) FS, [SLIMBUS\\_FR\\_INFO\[1\]](#) FI, and [SLIMBUS\\_FR\\_INFO\[0\]](#) GC\_TX\_COL bits (framer device-specific IEs)
  - The [SLIMBUS\\_FL\\_INFO\[4\]](#) DATA\_SLOT\_OVERLAP bit (interface device-specific IE)
  - The [SLIMBUS\\_FL\\_INFO\[2\]](#) LOST\_FS, [SLIMBUS\\_FL\\_INFO\[1\]](#) LOST\_SFS, and [SLIMBUS\\_FL\\_INFO\[0\]](#) LOST\_MS bits (interface device-specific IEs)
  - The [SLIMBUS\\_SMR\\_INFO\[0\]](#) MC\_TX\_COL, [SLIMBUS\\_SMT\\_INFO\[0\]](#) MC\_TX\_COL bit (primitives of interface device-specific IE)
  - The [SLIMBUS\\_DCT\\_INFO\\_j\[0\]](#)/[SLIMBUS\\_DCR\\_INFO\\_j\[0\]](#) DATA\_TX\_COL bit (generic device IEs)
- Reset all pending reconfiguration parameters by writing 1 to the [SLIMBUS\\_FL\\_CONTROL\[7\]](#) CLEAR\_RECONFIGURATION bit (see [Section 23.9.4.4.4](#)).
- Reset all device ports (if any), as explained in [Section 23.9.4.4.3.6, Port Reset](#).

#### **23.9.4.4.3.6 Port Reset**

A port reset can have several causes:

- Device reset of the device to which the port belongs, which may be part of a wider component of the bus reset
- Synchronization loss. Although all ports are reset, this is not equivalent to a device reset, because logical addresses are not cleared.
- Specific software messages reception

#### **23.9.4.4.3.7 Bus Shutdown**

A bus shutdown stops definitively all SLIMbus activity (CLK and DATA) and brings the bus back to its original (preboot) state. When the shutdown is executed, the system must reboot from scratch.

The bus shutdown is requested by the manager device through a reconfiguration sequence.

Bus shutdown sequence:

- Write 1 to the [SLIMBUS\\_FL\\_CONTROL\[6\]](#) BUS\_SHUTDOWN bit. This is valid for an active framer (clock source) or a clock receiver.
- Wait for the end of the reconfiguration sequence to execute.
- Beyond that point, the bus is down and can be restarted only by a boot sequence.

#### **23.9.4.4.4 SLIMbus Reconfigure Parameters**

During a reconfiguration message sequence, the controller accumulates parameter changes to be applied at the following reconfiguration boundary. If the sequence does not complete correctly, all changes are discarded. In specific boot and reset cases, the parameters are also set or reset automatically.

This section provides an overview of all the configuration register fields and their use.

Writing 1 to the [SLIMBUS\\_FL\\_CONTROL\[7\]](#) CLEAR\_RECONFIGURATION bit has an immediate effect on the configuration registers.

- The following registers fields revert to their current value; that is, the value currently used for SLIMbus operations. The value previously written in is discarded.
  - The [SLIMBUS\\_FL\\_SM\[4:0\]](#) SM bit field: Sets the control space width and subframe length.
  - The [SLIMBUS\\_FL\\_CG\[3:0\]](#) CG bit field: The root/SLIMbus frequency ratio is defined as  $2^{(10-G)}$ .
  - The [SLIMBUS\\_FL\\_RF\[3:0\]](#) RF bit field: Sets the root frequency
  - The [SLIMBUS\\_FR\\_CLOCK\\_SOURCE\[5:3\]](#) CLKSEL: Clock input source selection
  - [SLIMBUS\\_FR\\_CLOCK\\_SOURCE\[2:0\]](#) CLKDIV bit field: Root divider ratio, applied on clock input to obtain root clock, to be used at next reconfiguration boundary. Input/ root frequency ratio is defined as  $2^{(CLKDIV-1)}$ .
  - The [SLIMBUS\\_DCT\\_CONFIG1\\_j\[31\]/SLIMBUS\\_DCR\\_CONFIG1\\_j\[31\]](#) ENABLE bit (this field is also cleared when the superframe or frame sync is lost): Data channel agent enabling control
  - The [SLIMBUS\\_DCT\\_CONFIG1\\_j\[15\]/SLIMBUS\\_DCR\\_CONFIG1\\_j\[15\]](#) CL bit: Channel link configuration mode selection
  - The [SLIMBUS\\_DCT\\_CONFIG1\\_j\[26:20\]/SLIMBUS\\_DCR\\_CONFIG1\\_j\[26:20\]](#) PR bit field: Presence rate configuration
  - The [SLIMBUS\\_DCT\\_CONFIG1\\_j\[3:0\]/SLIMBUS\\_DCR\\_CONFIG1\\_j\[3:0\]](#) TP bit field: Transfer protocol selection
  - The [SLIMBUS\\_DCT\\_CONFIG2\\_j\[11:0\]/SLIMBUS\\_DCR\\_CONFIG2\\_j\[11:0\]](#) SD bit field: Segment distribution
- The following register bits are cleared to 0. They correspond to features enabled for the upcoming reconfiguration boundary, and that are discarded here.
  - The [SLIMBUS\\_FR\\_CONTROL\[1\]](#) BUS\_RESET bit: Bus reset request control
  - The [SLIMBUS\\_FR\\_CONTROL\[0\]](#) BOOT bit: Initiate an active framer (clock source) boot sequence for the component when the component is default active framer.
  - The [SLIMBUS\\_FL\\_CONTROL\[6\]](#) BUS\_SHUTDOWN
  - [SLIMBUS\\_FL\\_CONTROL\[0\]](#) BOOT bit: Boot the component when not in active framer mode.
  - The [SLIMBUS\\_FL\\_CONTROL\[4\]](#) KILL\_SFS bit is set when the reconfiguration sequence terminates incorrectly (in that case the bit must be set after the clear-up). Forces immediate frame synchronization loss.
  - The [SLIMBUS\\_FL\\_CONTROL\[7\]](#) CLEAR\_RECONFIGURATION bit itself; that is, the bit never reads out 1.
  - The [SLIMBUS\\_FR\\_FRAMER\\_HANDBOVER\[12\]](#) HANDBOVER\_ENABLE bit: Enable framer handover, upon NEXT\_ACTIVE\_FRAMER(NCO,NCI) reception.
  - The [SLIMBUS\\_FR\\_CLOCK\\_PAUSE\[0\]](#) CLOCK\_PAUSE bit: Controls clock pause/restart
- The following bit fields do not change, although they are updated as part of the reconfiguration message sequence:
  - The [SLIMBUS\\_FR\\_FRAMER\\_HANDBOVER\[11:0\]](#) NCO\_NCI bit field: The value is don't care as long as [SLIMBUS\\_FR\\_FRAMER\\_HANDBOVER\[12\]](#) HANDBOVER\_ENABLE = 0, and is overwritten if a new handover is scheduled.
  - The [SLIMBUS\\_FR\\_CLOCK\\_PAUSE\[2:1\]](#) TRT bit field: The value is don't care as long as [SLIMBUS\\_FR\\_CLOCK\\_PAUSE\[0\]](#) CLOCK\_PAUSE = 0, and is overwritten if a new pause is scheduled.

#### 23.9.4.5 Data Channels (Ports) Management

Ports (contained in devices) are associated with data channels that connect several ports, typically one output port on a data source device and one or more inputs ports on a data sink device, over periodically scheduled data segments and using a given transfer protocol (TP).

##### 23.9.4.5.1 Data input/output: manual mode

A data channel agent FIFO stores a whole number of FIFO words. A FIFO word is a chunk of data that contains the AUX and DATA information of one SLIMbus data segment, with the following properties:

- The FIFO word size minimum is 4 bits (one slot) in the current implementation, as defined in the MIPI SLIMbus standard 1.01.00; they are of little practical use.

- The FIFO word size maximum is 32 bits (eight slots) in the current implementation, equal to one physical FIFO RAM word and to one read/write access. Larger segment sizes (up to 31 slots = 124 bits) segments are defined in the MIPI SLIMbus standard version 1.01.00, but they are not implemented here.
- Between the minimum and maximum values, the FIFO word size is programmable, with a slot (that is, 4 bits) granularity.
- The AUX information is a field of 0, 1, or 2 slots (that is, 0, 4, or 8 bits) transmitted immediately before the DATA slots. All sizes are supported by current implementation. It is stored and managed along with the DATA by the module FIFOs.
- The DATA information is a field of 0 to 31 slots in the MIPI SLIMbus standard version 1.01.00, with slot granularity. This is implemented within the limits of the FIFO word size given previously, with AUX size + DATA size = FIFO word size; therefore, the DATA information is a field of 0 to 8 slots.
- The extended asynchronous transfer protocols (TP; that is, simplex and half-duplex) are exceptions to the previously discussed rules. In that case, the source device fills each segment with as many bytes as it deems necessary (provided it fits in the segment), and lets the sink device know how many through the TAG NVB field:
  - The source transmits as many whole FIFO words as it deems necessary. This implies that the FIFO word is a byte multiple (that is, an even number of 4-bit slots).
  - The sink receives a whole number of bytes, which it stores in byte-size FIFO words.
  - AUX is not applicable and is not supported in these modes.

Each read (or write) access to the appropriate [SLIMBUS\\_DCR\\_DATA\\_j](#) (respectively, [SLIMBUS\\_DCT\\_DATA\\_j](#)) removes (respectively, adds) one FIFO word from RX (TX) FIFO #N. [Table 23-655](#) lists examples of possible FIFO word mappings (with the AUX and DATA slots indicated as A() and D()) inside 32-bit read and write accesses.

**NOTE:** The transmission over the SLIMbus serial interface is always A(0) to A(N1), and then D(0) to D(N2); that is, MSB to LSB.

**Table 23-655. Examples of AUX + DATA Mapping in 32-bit Read/Write Accesses**

FIFO Format	Word	FIFO Word Size (in Slots)	32-bit Access Read/Write Data (Bit Range)									
			31..28	27..24	23..20	19..16	15..12	11..8	7..4	3..0		
4-bit + 0 aux		1 (min)									D(0)	
8-bit + 0 aux		2									D(0)	D(1)
12-bit + 0 aux		3								D(0)	D(1)	D(2)
16-bit + 0 aux		4						D(0)	D(1)	D(2)	D(3)	
24-bit + 0 aux		6			D(0)	D(1)	D(2)	D(3)	D(4)	D(5)		
32-bit + 0 aux		8 (max)	D(0)	D(1)	D(2)	D(3)	D(4)	D(5)	D(6)	D(7)		
4-bit + 4 aux		2								A(0)	D(0)	
24-bit + 4 aux		7		A(0)	D(0)	D(1)	D(2)	D(3)	D(4)	D(5)		
28-bit + 4 aux		8 (max)	A(0)	D(0)	D(1)	D(2)	D(3)	D(4)	D(5)	D(6)		
4-bit + 8 aux		3							A(0)	A(1)	D(0)	
24-bit + 8 aux		8 (max)	A(0)	A(1)	D(0)	D(1)	D(2)	D(3)	D(4)	D(5)		

### 23.9.4.5.2 Data Input/Output: DMA Request Mode

The DMA mode reuses all the features of the manual mode defined in [Section 23.9.4.5.1](#), but adds a DMA request generation mechanism to offload the data manipulation task to an external (off-module) DMA controller. As far as the FIFOs themselves are concerned, the accesses are unchanged. For more information about the DMA controller configuration see [Chapter 16, SDMA](#).

DMA data channels can be enabled by the [SLIMBUS\\_CMP\\_DMAENABLE\\_SET](#) register and disabled by the [SLIMBUS\\_CMP\\_DMAENABLE\\_CLR](#) register.

### 23.9.4.5.3 Data Channel Agent FIFO Control and Status

Each data channel agent is associated with a FIFO, fixed in either TX or RX direction. All TX FIFOs are mapped to a shared buffer space, all RX FIFOs to another shared buffer space. FIFO sizes are software-programmable.

A FIFO must be configured before the data channel is enabled, depending on system-related parameters. For instance, more or less FIFO depth may be required. More FIFO is desirable to prevent underflow/overflow errors in a jittery transmission, but may also carry an unacceptable latency penalty in a voice stream. In any case, FIFO depth is beyond the SLIMbus protocol.

### 23.9.4.6 FIFO Management

The SLIMbus controller includes two FIFOs, used to buffer transmit/receive data/messages. The SLIMBUS1 FIFO is 32 bytes for TX and 32 bytes for RX. The SLIMBUS2 FIFO is 64 bytes for TX and 64 bytes for RX. FIFO size information is given by the `SLIMBUS_CMP_HWINFO[19:12]` RRSZ and [11:4] TRSZ bit fields.

FIFOs are connected to SLIMbus-side agents, which transmit and receive data on the serial SLIMbus DATA line. All ports are synchronous to the SLIMbus clock. The numbers of data channel agents are given by `SLIMBUS_CMP_HWINFO[27:24]` TDC and [23:20] RDC bit fields.

A segment (see [Section 23.9.2.3.2](#)) passes through the internal RX and TX FIFOs, where it is stored in FIFO words. The size of the FIFO word, expressed in 4-bit slots, is configurable by software. The FIFO word size is hardware-calculated with the following parameters:

- The `SLIMBUS_DCR_CONFIG1_j[9:5]`/`SLIMBUS_DCT_CONFIG1_j[9:5]` SL bit field: Channel segment length, in slots
- The `SLIMBUS_DCR_CONFIG1_j[14:10]`/`SLIMBUS_DCT_CONFIG1_j[14:10]` DL bit field: Channel data length, integer
- The `SLIMBUS_DCR_CONFIG1_j[3:0]`/`SLIMBUS_DCT_CONFIG1_j[3:0]` TP (tag\_length) bit field: Length of segment TAG field, function of the channel transfer protocol, in slots.
- The `SLIMBUS_DCR_CONFIG1_j[19:16]`/`SLIMBUS_DCT_CONFIG1_j[19:16]` AF (aux\_length) bit field: Length of segment AUX field, function of the channel auxiliary format, in slots.

The FIFO word size is computed as follows by the module hardware:

- if DL is equal to 0, FIFO word size = SL tag\_length(TP)
- if DL is not equal to 0, FIFO word size = aux\_length(AF) + DL

#### 23.9.4.6.1 FIFO Mapping

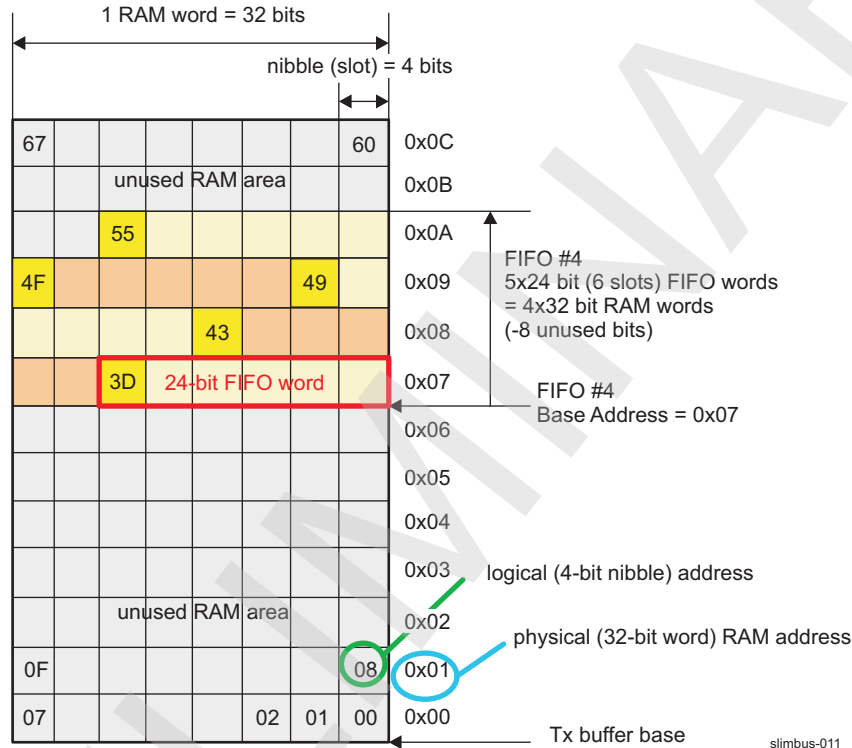
Each FIFO must be mapped in a free zone of the shared buffer before enabling, as follows:

- The `SLIMBUS_DCR_FIFO_SETUP2_j[7:0]` SIZE and/or `SLIMBUS_DCT_FIFO_SETUP2_j[7:0]` SIZE + 1: Capacity of FIFO in FIFO words. (The minimum setting recommended by the SLIMbus protocol is 4-word deep.)
- The `SLIMBUS_SMT_FIFO_SETUP[6:5]` SIZE\_HI/`SLIMBUS_SMR_FIFO_SETUP[6:5]` SIZE\_HI and `SLIMBUS_SMT_FIFO_SETUP[4:1]` SIZE\_LO/`SLIMBUS_SMR_FIFO_SETUP[4:1]` SIZE\_LO bit fields: Capacity of shared TX/RX FIFOs, in bytes, minus 1.
- The `SLIMBUS_DCR_FIFO_SETUP1_j[11:5]` BASE\_ADDR and/or `SLIMBUS_DCT_FIFO_SETUP1_j[11:5]` BASE\_ADDR bit fields: RX and/or TX data channel agent FIFO base address within shared RX/TX RAM
  - FIFO base nibble address = <FIFO base physical add> \* 8
  - FIFO top nibble address = <FIFO base nibble add> + (<FIFO size> \* <FIFO word size>) – 1
  - FIFO top physical address = <FIFO top nibble add> / 8
  - The next base physical address available to map another FIFO is the current FIFO top physical address + 1.
- The shared RX and TX buffers are independent; that is, the RX and TX FIFOs are mapped to separate address spaces.

[Figure 23-215](#) is an example mapping, with the following numerical configuration:

- Isochronous (SLIMBUS\_DCT\_CONFIG1\_4[3:0] TP = 0x0) TX FIFO using TX data channel agent 4
- FIFO word size: 6 slots = 24 bits, 20 DATA + 4 AUX (AF = 0xB, DL = 0x5)
- FIFO size: SLIMBUS\_DCT\_FIFO\_SETUP2\_4[7:0] SIZE = 4 (4 + 1 = 5 FIFO words stored in FIFO)
- FIFO base physical address: SLIMBUS\_DCT\_FIFO\_SETUP1\_4[11:5] BASE\_ADDR = 0x7 (= 7)
- FIFO base nibble address = 0x7 \* 8 = 0x38 (56)
- FIFO top nibble address = 0x38 + (0x5 \* 0x6) - 1 = 0x55 (= 85)
- FIFO top physical address = 0x55 / 0x8 = 0xA (= 10, this is a integer division)
- Next base physical address available to map another FIFO: 0xA + 1 = 0xB (= 11)

Figure 23-215. Example of FIFO Mapping in Shared Buffer



### 23.9.4.6.2 FIFO Status

The following receive-side status is available from the register fields:

- The `SLIMBUS_DCR_FIFO_STATUS_j[11:2]` RD\_LEVEL bit field: Number of FIFO words stored in RX FIFO #n. (This field is 1 bit longer than the size and threshold fields, because it ranges from 0 to the FIFO size, whereas the other fields range only from 0 to the FIFO size minus 1).
- The `SLIMBUS_DCR_FIFO_STATUS_j[0]` EMPTYbit: Active (1) status when RX FIFO #n contains 0 words (`rd_level = 0`); otherwise, low (0). A read may cause an underflow.
- The `SLIMBUS_DCR_FIFO_STATUS_j[1]` HILEVEL bit: Active (1) status when FIFO (read) level is (strictly) above the high threshold (that is, when there is more than a certain maximum of FIFO words stored in the FIFO). Equivalent to full in case `dma_hi_threshold = <FIFO size> - 1` (the maximum allowed value).

Table 23-656 shows numerical examples for all valid DMA threshold values and all levels in an RX FIFO of `<FIFO size> = 4` FIFO words (`SLIMBUS_DCR_FIFO_SETUP2_j[7:0] SIZE = 0x3`).

**NOTE:** The valid threshold range is from 0 to `<FIFO size> - 1`: the case `threshold = <FIFO size>` is functionally useless and impossible to program for maximum-sized FIFOs (because the `rd_level` field is 1 bit wider than the threshold fields).

**Table 23-656. HILEVEL/Empty Status of a 4-Word Receive FIFO, for all DMA Threshold Values**

rd_level (FIFO Words)	Empty	HILEVEL			
		thr = 0	thr = 1	thr = 2	thr = 3
0 (empty FIFO)	1	0	0	0	0
1	0	1	0	0	0
2	0	1	1	0	0
3	0	1	1	1	0
4 (full FIFO)	0	1	1	1	1

The following transmit-side status is available from the following bit fields:

- **SLIMBUS\_DCT\_FIFO\_STATUS\_j[11:2] WR\_LEVEL:** Number of free FIFO words in TX FIFO #n. (This field is 1 bit longer than the size and threshold fields, because it ranges from 0 to the FIFO size, whereas the other fields range only from 0 to the FIFO size minus 1).
- **SLIMBUS\_DCT\_FIFO\_STATUS\_j[0] FULL:** Active (1) status when TX FIFO #n contains <FIFO size> words (wr\_level = 0); otherwise, low (0). A write may cause an overflow.
- **SLIMBUS\_DCT\_FIFO\_STATUS\_j[1] LOLEVEL:** Active (1) status when FIFO (write) level is under (or equal to) the low threshold; that is, when there is less than a certain minimum of free FIFO words in the FIFO. Equivalent to full in case dma\_lo\_threshold = 0.

Table 23-657 shows numerical examples for all valid DMA threshold values and all levels in a TX FIFO of <FIFO size> = 4 FIFO words (SLIMBUS\_DCT\_FIFO\_SETUP2\_j[7:0] SIZE = 0x3).

**NOTE:** The valid threshold range is from 0 to <FIFO size> – 1: the case threshold = <FIFO size> is functionally useless and impossible to program for maximum-sized FIFOs (because the wr\_level field is 1 bit wider than the threshold fields).

**Table 23-657. LOLEVEL/Full Status of a 4-Word Transmit FIFO, for all DMA Threshold Values**

wr_level (FIFO Words)	Full	LOLEVEL			
		thr = 3	thr = 2	thr = 1	thr = 0
4 (empty FIFO)	1	1	1	1	0
3	0	1	1	1	0
2	0	0	1	1	0
1	0	0	0	1	0
0 (full FIFO)	0	0	0	0	1

### 23.9.4.6.3 FIFO Configuration

The bit fields used to configure the FIFOs are:

- **SLIMBUS\_DCR\_FIFO\_SETUP1\_j[11:5] BASE\_ADDR** and/or **SLIMBUS\_DCT\_FIFO\_SETUP1\_j[11:5] BASE\_ADDR:** RX and/or TX data channel agent FIFO base address within shared RX/TX RAM
- **SLIMBUS\_DCR\_FIFO\_SETUP1\_j[4:1] DMA\_REQ\_SIZE** and/or **SLIMBUS\_DCT\_FIFO\_SETUP1\_j[4:1] DMA\_REQ\_SIZE:** DMA request size minus 1, in accesses
- **SLIMBUS\_DCR\_FIFO\_SETUP2\_j[7:0] SIZE** and/or **SLIMBUS\_DCT\_FIFO\_SETUP2\_j[7:0] SIZE:** Capacity of FIFO in FIFO words. (The FIFO can be 1-word deep [the minimum possible value], but it is not recommended. The minimum setting recommended by the SLIMbus protocol is 4-word deep).
- **SLIMBUS\_DCR\_FIFO\_SETUP2\_j[15:8] SB\_TRESHOLD:** SLIMbus-side RX data channel agent FIFO threshold, used for SLIMbus reception.
- **SLIMBUS\_DCT\_FIFO\_SETUP2\_j[15:8] SB\_TRESHOLD:** SLIMbus-side TX data channel agent FIFO threshold, used for SLIMbus transmission.
- **SLIMBUS\_DCR\_FIFO\_SETUP2\_j[23:16] DMA\_TRESHOLD:** Software-side RX data channel agent FIFO low threshold. DMA write requests stop when **SLIMBUS\_DCR\_FIFO\_STATUS\_j[11:2] RD\_LEVEL** goes under threshold.
- **SLIMBUS\_DCT\_FIFO\_SETUP2\_j[23:16] DMA\_TRESHOLD:** Software-side TX data channel agent

FIFO low threshold. DMA write requests stop when `SLIMBUS_DCT_FIFO_STATUS_j[11:2]` `WR_LEVEL` goes under threshold.

- `SLIMBUS_DCR_FIFO_SETUP2_j[23:16]` `DMA_TRESHOLD`: Software-side RX data channel agent FIFO high threshold. DMA read requests start when `SLIMBUS_DCR_FIFO_STATUS_j[11:2]` `RD_LEVEL` goes above threshold.
- `SLIMBUS_DCT_FIFO_SETUP2_j[23:16]` `DMA_TRESHOLD`: Software-side TX data channel agent FIFO high threshold. DMA read requests start when `SLIMBUS_DCT_FIFO_STATUS_j[11:2]` `WR_LEVEL` goes above threshold.
- `SLIMBUS_DCR_FIFO_STATUS_j[0]` `EMPTY`: Active (1) status when RX FIFO #n contains 0 words (`rd_level = 0`), else low (0). A read may cause an underflow.
- `SLIMBUS_DCT_FIFO_STATUS_j[0]` `FULL`: Active (1) status when TX FIFO #n contains <FIFO size> words (`wr_level = 0`), else low (0). A write may cause an overflow.
- `SLIMBUS_DCT_FIFO_STATUS_j [1]` `LOLEVEL`: Active (1) status when FIFO (write) level is under (or equal to) the low threshold, that is, when there is less than a certain minimum of free FIFO words in the FIFO. Equivalent to full in case `dma_lo_threshold = 0`.
- `SLIMBUS_DCR_FIFO_STATUS_j[1]` `HILEVEL`: Active (1) status when FIFO (read) level is (strictly) above the high threshold, that is, when there is more than a certain maximum of FIFO words stored in the FIFO. Equivalent to full in case `dma_hi_threshold = <FIFO size> - 1` (the maximum allowed value).
- `SLIMBUS_DCR_FIFO_STATUS_j[11:2]` `RD_LEVEL`: Number of FIFO words stored in RX FIFO #n. (This field is 1 bit longer than the size and threshold fields, because it ranges from 0 to the FIFO size, whereas the other fields only range from 0 to the FIFO size minus 1).
- `SLIMBUS_DCT_FIFO_STATUS_j[11:2]` `WR_LEVEL`: Number of free FIFO words in TX FIFO #n. (This field is 1 bit longer than the size and threshold fields, because it ranges from 0 to the FIFO size, whereas the other fields only range from 0 to the FIFO size minus 1).

#### 23.9.4.7 Data Channel Segment Composition

The allocated size of a data channel segment is set by segment length (SL) parameter. The segment can be broken down into: TAG slots, AUX slots, DATA slots, and unused slots. The parameters listed in [Table 23-658](#) configure the different lengths.

**Table 23-658. Segment Composition**

Parameter	Register Field	Segment Field	Size Range (slots)	Comments
Segment length	<code>SLIMBUS_DCR_CONFIG1_j[9:5]</code> / <code>SLIMBUS_DCT_CONFIG1_j[9:5]</code> SL	Segment	1 to 31	Encoding 0x0 is reserved.
Transport protocol	<code>SLIMBUS_DCR_CONFIG1_j[3:0]</code> / <code>SLIMBUS_DCT_CONFIG1_j[3:0]</code> TP	TAG	0, 1, 2	N/A
AUX format	<code>SLIMBUS_DCR_CONFIG1_j[19:16]</code> / <code>SLIMBUS_DCT_CONFIG1_j[19:16]</code> AF	AUX	0, 1, 2	N/A
Data length	<code>SLIMBUS_DCR_CONFIG1_j[14:10]</code> / <code>SLIMBUS_DCT_CONFIG1_j[14:10]</code> DL	DATA	0 to 31	Encoding 0x0 is "all remaining slots," don't care if NVB is present
Number of valid bytes	NVB (not in a register field, send in the message)	DATA	0 to 28, even only	Encoded in AUX field, overrides DL, exists only for TP = extended asynchronous

**NOTE:**

- Explicit data length (DL) range is 1 to 31 slots. When encoding 0x0 is used, the DATA field occupies all remaining slots of the segment; that is, not occupied by TAG and AUX. This allows to encode a 0-slot data length.
- NVB is available only in the extended asynchronous transfer protocols (TP), which uses two TAG slots: the maximum remaining length is 31 (SL max) – 2 = 29 slots; that is, the maximum NVB is 14 bytes = 28 slots =< 29 (because NVB is byte-granular and cannot encode odd slot counts)
- Not all combinations are legal. For example, the DATA field of a pushed channel (TP = 0x1: 1 TAG slot) with no AUX information (AF = 0x0) and a 5-slot overall segment length (SL = 0x5) can have only 1 to 4 slots, encoded respectively as DL = 0x1, 0x2, 0x3 and 0x4/0x0.

The SL, TP, AF, and DL setup parameters are received from the appropriate control messages (inside or outside reconfiguration sequences), and programmed into appropriate control registers. They are typically static over long periods.



### 23.9.4.8 SLIMbus Transfer Protocols

Several registers are used to configure each Transfer Protocol. [Table 23-659](#) shows a description of these registers and fields.

**Table 23-659. Register Fields Used to Configure Transfer Protocols**

Register	Field	Description
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [3:0]	TP	Transfer protocol choice
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [4]	TP_QUALIFIER	Used only for asynchronous TPs (qualifier for plain/extended half-duplex asynchronous TPs)
<a href="#">SLIMBUS_DCT_CONFIG1_j</a> [4]	TP_QUALIFIER	Used only for asynchronous TPs (qualifier for plain/extended half-duplex asynchronous TPs)
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [9:5]	SL	Total segment length (SL), in 4-bit slots, $SL \geq TAG\ length + AUX\ length + DATA\ length$
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [26:23]	PR_HI	4-MSB presence rate (PR). Selects the base frequency and the calculation mode of N, with: $PR = 2^N \times base\ frequency$ .
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [22:20]	PR_LO	3-LSB presence rate (PR). Sets the multiple of the base frequency for the presence frequency.
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [14:10]	DL	Segment data length, in 4-bits slots. AUX+DATA length must fit in a FIFO word.
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [15]	CL	N/A
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [31]	ENABLE	Enable/disable data channel
<a href="#">SLIMBUS_DCR_CONFIG2_j / SLIMBUS_DCT_CONFIG2_j</a> [11:0]	SD	Segment periodicity and offset (segment distribution)

#### 23.9.4.8.1 Isochronous Transfer Protocol

[Table 23-660](#) lists the register fields to program isochronous data transfer protocol.

**Table 23-660. Register Fields Used for Isochronous Transfer Protocol**

Register	Field	Value
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [3:0]	TP	0x0 (isochronous transfer protocol)
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [4]	TP_QUALIFIER	Unused in this mode
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [9:5]	SL	2 to 8, see <a href="#">Section 23.9.4.7</a>
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [26:23]	PR_HI	N/A, only for information
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [22:20]	PR_LO	N/A
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [14:10]	DL	2 to 8, see <a href="#">Section 23.9.4.7</a>
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [15]	CL	N/A
<a href="#">SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j</a> [31]	ENABLE	1 (enabled)
<a href="#">SLIMBUS_DCR_CONFIG2_j / SLIMBUS_DCT_CONFIG2_j</a> [11:0]	SD	Segment periodicity and offset

##### 23.9.4.8.1.1 Isochronous Segment Composition

Like for all other TPs, the TDM allocation of a pushed data channel is composed of periodic segments of equal length, with the 5-bit segment length (SL) defining the reserved length, in slots (1 to 31), and the segment distribution (SD) defining the segmen offset and periodicity.

Isochronous has no explicit presence bit because data is always assumed to be present. The SL-slots segment is therefore always composed of the following elements:

- 0, 1, or 2 AUX slots, as defined by the 4-bit auxiliary format (AF) configuration field
- 2 to 8 DATA slots, as defined by the 5-bit data length (DL) configuration field. Full range defined by standard is 1 to 31.
- 0 or more unused slots (should be 0, because this is wasted bandwidth)

**NOTE:**

- Maximum supported SL and DL are limited by the maximum FIFO word size (the actual implementation of the agent), because each segment (with data present) contains one FIFO word of DATA + AUX. In the current case, FIFO word size is 2 to 8 slots (8 to 32 bits):
  - DL range is 2 to 8 (assuming no AUX slot, or AUX slots stored with the DATA in the FIFO)
  - SL range is 2 to 8 (= DL, since there is no TAG in isochronous protocol).
  - DL = 0 is a special encoding where all slots after the AUX are considered to be DATA slots. In that case, there are by definition no unused slots.
- DL = 0 is a special encoding where all slots after the AUX are considered to be DATA slots. In that case, there are by definition no unused slots.
- Not all combinations of legal field values are legal. The condition is that AUX and DATA slots must fit within the allocated segment, that is, the sum must not be (strictly) greater than SL.
  - DL = 31 cannot be used with 1-slot AUX contents because 1 AUX slot + 31 DATA slots (DL) > 31 segment slots (SL max) (assuming no more FIFO word size limit)
  - DL = 30 cannot be used with 2-slot AUX contents (assuming no more FIFO word size limit)
  - ...

**23.9.4.8.2 Pushed Transfer Protocol**

Table 23-661 lists the register fields to program to have a pushed data transfer protocol.

**Table 23-661. Register Fields Used for Pushed Transfer Protocol**

Register	Field	Value
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[3:0]	TP	0x1 (pushed Transfer Protocol)
SLIMBUS_DCR_CONFIG1_j[4]	TP_QUALIFIER	unused in this mode
SLIMBUS_DCT_CONFIG1_j[4]	TP_QUALIFIER	unused in this mode
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[9:5]	SL	3 to 9, see <a href="#">Section 23.9.4.7</a>
SLIMBUS_DCR_CONFIG1_j[26:23]	PR_HI	N/A, only for information
SLIMBUS_DCR_CONFIG1_j[22:20]	PR_LO	N/A, only for information
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[14:10]	DL	2 to 8, see <a href="#">Section 23.9.4.7</a>
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[15]	CL	N/A
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[31]	ENABLE	1 (enabled)
SLIMBUS_DCR_CONFIG2_j / SLIMBUS_DCT_CONFIG2_j[11:0]	SD	segment periodicity and offset

**23.9.4.8.2.1 Pushed Segment Composition**

Like for all other TPs, the TDM allocation of a pushed data channel is composed of periodic segments of equal length, with the 5-bit segment length (SL) defining the reserved length, in slots (1 to 31), and the segment distribution (SD) defining the segment's offset and periodicity.

When data is present (P=1), the SL-slots segment is composed of the following elements:

- 1 TAG slot for STR, P control bits (Sample TRansmit, Presence)
- 0, 1 or 2 AUX slots, as defined by the 4-bit auxiliary format (AF) configuration field
- 2 to 8 DATA slots, as defined by the 5-bit data length (DL) configuration field. Full range defined by standard is 1 to 30
- 0 or more unused slots. (should be 0, as this is wasted bandwidth)

When no data is present (P=0), all slots except the TAG are unused and undriven.

**NOTE:**

- Maximum supported SL and DL are limited by the maximum FIFO word size (that is, the agent's actual implementation), as each segment (with data present) contains one FIFO word of DATA + AUX. In the current case, FIFO word size is 2 to 8 slots (8 to 32 bits), that is,
  - DL range is 2 to 8 (assuming no AUX slot, or AUX slots stored with the DATA in the FIFO)
  - SL range is 3 to 9 (= DL + 1 TAG)
- DL=0 is a special encoding where all slots after the AUX are considered to be DATA slots. In that case, there are by definition no unused slots.
- Not all combinations of legal field values are legal. The condition is that AUX and DATA slots must fit within the allocated segment, that is, the sum must not be (strictly) greater than SL.
  - DL=31 cannot be used at all for pushed (although it is legal for other TPs) because 1 TAG slot + 31 DATA slots (DL) > 31 segment slots (SL max).
  - DL=30 cannot be used with 1-slot AUX contents (assuming no more FIFO word size limit)
  - DL=29 cannot be used with 2-slot AUX contents (assuming no more FIFO word size limit)
  - ...

**23.9.4.8.3 Pulled Transfer Protocol**

Table 23-662 lists the register fields to program to have a pulled data transfer protocol.

**Table 23-662. Register fields used for Pulled Transfer Protocol**

Register	Field	Value
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [3:0]	TP	0x2 (pulled Transfer Protocol)
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [4]	TP_QUALIFIER	unused in this mode
<a href="#">SLIMBUS_DCT_CONFIG1_j</a> [4]	TP_QUALIFIER	unused in this mode
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [9:5]	SL	3 to 9, see <a href="#">Section 23.9.4.7</a>
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [26:23]	PR_HI	N/A, only for information
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [22:20]	PR_LO	N/A, only for information
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [14:10]	DL	2 to 8, see <a href="#">Section 23.9.4.7</a>
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [15]	CL	N/A
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [31]	ENABLE	1 (enabled)
<a href="#">SLIMBUS_DCR_CONFIG2_j</a> / <a href="#">SLIMBUS_DCT_CONFIG2_j</a> [11:0]	SD	segment periodicity and offset

### 23.9.4.8.3.1 Pulled Segment Composition

Like for all other TPs, the TDM allocation of a pushed data channel is composed of periodic segments of equal length, with the 5-bit segment length (SL) defining the reserved length, in slots (1 to 31), and the segment distribution (SD) defining the segment's offset and periodicity.

When data is present (P=1), the SL-slots segment is composed of the following elements:

- 1 TAG slot for SRQ, P control bits (Sample ReQuest, Presence)
- 0, 1 or 2 AUX slots, as defined by the 4-bit auxiliary format (AF) configuration field
- 2 to 8 DATA slots, as defined by the 5-bit data length (DL) configuration field. Full range defined by standard is 1 to 30.
- 0 or more unused slots. (should be 0, as this is wasted bandwidth)

When no data is present (P=0), all slots except the TAG are unused and undriven.

#### NOTE:

- Maximum supported SL and DL are limited by the maximum FIFO word size (that is, the agent's actual implementation), as each segment (with data present) contains one FIFO word of DATA + AUX. In the current case, FIFO word size is 2 to 8 slots (8 to 32 bits), that is,
  - DL range is 2 to 8 (assuming no AUX slot, or AUX slots stored with the DATA in the FIFO)
  - SL range is 3 to 9 (= DL + 1 TAG)
- DL=0 is a special encoding where all slots after the AUX are considered to be DATA slots. In that case, there are by definition no unused slots.
- Not all combinations of legal field values are legal. The condition is that AUX and DATA slots must fit within the allocated segment, that is, the sum must not be (strictly) greater than SL.
  - DL=31 cannot be used at all for pushed (although it is legal for other TPs) because 1 TAG slot + 31 DATA slots (DL) > 31 segment slots (SL max).
  - DL=30 cannot be used with 1-slot AUX contents (assuming no more FIFO word size limit)
  - DL=29 cannot be used with 2-slot AUX contents (assuming no more FIFO word size limit)
  - ...

### 23.9.4.8.4 Asynchronous Transfer Protocol

Table 23-665 shows the configuration required for a (plain) asynchronous data channel agent, whether in simplex or in half-duplex mode.

One data channel requires one agent, except for 1/2 duplex where 2 agents are required (one in each direction).

**Table 23-663. "Plain" Asynchronous Endpoint Configurations**

Agent Setup	Type	TP	TP_qualifier	SD, SL, DL, ENABLE	Notes
TX simplex	TX	0x4 = "plain" asynchronous, simplex	0b0 = primary owner	independent	always primary owner
TX 1/2 duplex, primary	TX	0x5 = "plain" asynchronous, 1/2 duplex	0b0 = primary owner	same than RX (programming identical in both associated agent)	associated with RX agent
TX 1/2 duplex, secondary	TX	0x5 = "plain" asynchronous, 1/2 duplex	0b1 = secondary owner	same than RX (programming identical in both associated agent)	associated with RX agent

**Table 23-663. "Plain" Asynchronous Endpoint Configurations (continued)**

Agent Setup	Type	TP	TP_qualifier	SD, SL, DL, ENABLE	Notes
RX simplex	RX	0x4 = "plain" asynchronous, simplex	0b1 = secondary owner	independent	always secondary owner
RX 1/2 duplex, primary	RX	0x5 = "plain" asynchronous, 1/2 duplex	0b0 = primary owner	same than TX (programming identical in both associated agent)	associated with TX agent
RX 1/2 duplex secondary	RX	0x5 = "plain" asynchronous, 1/2 duplex	0b1 = secondary owner	same than TX (programming identical in both associated agent)	associated with TX agent

Table 23-666 lists the register fields to program to have a "plain" asynchronous transfer protocol.

**Table 23-664. Register fields used for "Plain" Asynchronous Transfer Protocol**

Register (for a given data channel agent)	Field	Value
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[3:0]	TP	0x4 (simplex) or 0x5 (half duplex)
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[4]	TP_QUALIFIER	primary or secondary ownership setup
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[9:5]	SL	3 to 9, see <a href="#">Section 23.9.4.7</a>
SLIMBUS_DCR_CONFIG1_j[26:23]	PR_HI	Presence Rate, only for information
SLIMBUS_DCR_CONFIG1_j[22:20]	PR_LO	N/A, only for information
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[14:10]	DL	2 to 8, see <a href="#">Section 23.9.4.7</a>
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[15]	CL	0 (disabled: makes no sense in asynchronous context)
SLIMBUS_DCR_CONFIG1_j / SLIMBUS_DCT_CONFIG1_j[31]	ENABLE	1 (enabled)
SLIMBUS_DCR_CONFIG2_j / SLIMBUS_DCT_CONFIG2_j[11:0]	SD	segment periodicity and offset

#### 23.9.4.8.4.1 Asynchronous Segment Composition

Like for all other TPs, the TDM allocation of an asynchronous data channel is composed of periodic segments of equal length, with the 5-bit segment length (SL) defining the reserved length, in slots (2 to 31), and the segment distribution (SD) defining the segment's offset and periodicity.

When data is present (P=1), the SL-slots segment is composed of the following elements:

- 1 TAG slot for T1, T2, CTS, P control bits
- 0, 1 or 2 AUX slots, as defined by the 4-bit auxiliary format (AF) configuration field
- 2 to 8 DATA slots, as defined by the 5-bit data length (DL) configuration field. Full range defined by standard is 1 to 30.
- 0 or more unused slots. (should be 0, as this is wasted bandwidth)

When no data is present (P=0), only the TAG slot is (partially) driven, depending on the ownership control and data flow control protocol phases. All other slots are unused and undriven.

**NOTE:**

- Maximum supported SL and DL are limited by the maximum FIFO word size (that is, the agent's actual implementation), as each segment (with data present) contains one FIFO word of DATA + AUX. In the current case, FIFO word size is 2 to 8 slots (8 to 32 bits), that is,
  - DL range is 2 to 8 (assuming no AUX slot, or AUX slots stored with the DATA in the FIFO)
  - SL range is 3 to 9 (= DL + 1 TAG)
- DL=0 is a special encoding where all slots after the AUX are considered to be DATA slots. In that case, there are by definition no unused slots. When used, the actual length of the segment data (in slots) is determined by hardware, by subtracting the TAG and AUX slots from the total segment length (SL). This case is strictly equivalent to the "direct" encoding, that is, where DL is a non-zero figure.
- Not all combinations of legal field values are legal, even in an ideal implementation. The condition is that TAG, AUX and DATA slots must fit within the allocated segment, that is, the sum must not be (strictly) greater than SL.
  - DL=31 cannot be used at all for asynchronous (although it is legal for other TPs) because 1 TAG slot + 31 DATA slots (DL) > 31 segment slots (SL max).
  - DL=30 cannot be used with 1-slot AUX contents (assuming no more FIFO word size limit)
  - DL=29 cannot be used with 2-slot AUX contents (assuming no more FIFO word size limit)
  - SL = 1 is forbidden, since the smallest asynchronous segment is 1 TAG slots + 1 DATA slot = 2 slots long.
  - ...

**23.9.4.8.4.2 Programming Specificities for Half-Duplex Protocol**

Contrary to all other standard transfer protocols (TP), the half-duplex asynchronous ones (plain or extended) are bidirectional, that is, DATA slots can go in both directions. This means that 1 half-duplex asynchronous port (mapped on 1 half-duplex asynchronous data channel) uses 2 unidirectional data channel agents (FIFOs and logic) in the same controller.

The two agents are of opposite directions (1 TX + 1 RX), and the configuration register fields must always be programmed simultaneously and with matching values for both, so that both own the exact same slots (the data channel's allocated distribution).

**23.9.4.8.5 Extended Asynchronous Transfer Protocol**

Table 23-665 shows the configuration required for an extended asynchronous data channel agent, whether in simplex or in half-duplex mode.

One data channel requires one agent, except for 1/2 duplex where 2 agents are required (one in each direction).

**Table 23-665. Extended Asynchronous Endpoint Configurations**

Agent Setup	Type	TP	TP_qualifier	SD, SL, DL, ENABLE	Notes
TX simplex	TX	0x6 = extended asynchronous, simplex	0b0 = primary owner	independent	always primary owner
TX 1/2 duplex, primary	TX	0x7 = extended asynchronous, 1/2 duplex	0b0 = primary owner	same than RX (programming identical in both associated agent)	associated with RX agent

**Table 23-665. Extended Asynchronous Endpoint Configurations (continued)**

Agent Setup	Type	TP	TP_qualifier	SD, SL, DL, ENABLE	Notes
TX 1/2 duplex, secondary	TX	0x7 = extended asynchronous, 1/2 duplex	0b1 = secondary owner	same than RX (programming identical in both associated agent)	associated with RX agent
RX simplex	RX	0x6 = extended asynchronous, simplex	0b1 = secondary owner	independent	always secondary owner
RX 1/2 duplex, primary	RX	0x7 = extended asynchronous, 1/2 duplex	0b0 = primary owner	same than TX (programming identical in both associated agent)	associated with TX agent
RX 1/2 duplex secondary	RX	0x7 = extended asynchronous, 1/2 duplex	0b1 = secondary owner	same than TX (programming identical in both associated agent)	associated with TX agent

The data length within the segment is now determined in the TAG NVB field, so that the SLIMbus data length (DL) parameter is not needed in the SLIMbus definition. The corresponding configuration register field [SLIMBUS\\_DCR\\_CONFIG1\\_j](#) / [SLIMBUS\\_DCT\\_CONFIG1\\_j](#)[14:10] DL is still needed to set the FIFO word size: it must always be programmed to 0x2 (slots) = 1 byte, in both the data sinks and sources, which corresponds to the NVB granularity unit.

[Table 23-666](#) lists the register fields to program to have a "plain" asynchronous transfer protocol.

**Table 23-666. Register fields used for Extended Asynchronous Transfer Protocol**

Register (for a given data channel agent)	Field	Value
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [3:0]	TP	0x6 (simplex) or 0x7 (half duplex)
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [4]	TP_QUALIFIER	primary or secondary ownership setup
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [9:5]	SL	4 to 31, see <a href="#">Section 23.9.4.7</a>
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [26:23]	PR_HI	Presence Rate, only for information
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> [22:20]	PR_LO	N/A, only for information
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [14:10]	DL	2: NVB data granularity (not the Slimbus "data length")
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [15]	CL	0 (disabled: makes no sense in asynchronous context)
<a href="#">SLIMBUS_DCR_CONFIG1_j</a> / <a href="#">SLIMBUS_DCT_CONFIG1_j</a> [31]	ENABLE	1 (enabled)
<a href="#">SLIMBUS_DCR_CONFIG2_j</a> / <a href="#">SLIMBUS_DCT_CONFIG2_j</a> [11:0]	SD	segment periodicity and offset

### 23.9.4.8.5.1 Extended Asynchronous Segment Composition

Like for all other TPs, the TDM allocation of an extended asynchronous data channel is composed of periodic segments of equal length, with the 5-bit segment length (SL) defining the reserved length, in slots (1 to 31), and the segment distribution (SD) defining the segment's offset and periodicity.

When data is present (P=1), the segment is composed of the following elements:

- 2 TAG slots for T1, T2, CTS, P control bits + 4-bit NVB field
- no (0) AUX slots. The current implementation does not support auxiliary fields in the extended asynchronous TP, that is, the only legal value in the 4-bit auxiliary format (AF) is 0b0000. Other values are theoretically allowed by the SLIMbus standard, but for this TP they have little functional use.
- 0 to 28 DATA slots, as defined by the 4-bit number of valid bytes (NVB) field
- 0 or more unused slots. (this is wasted bandwidth, but as NVB can change for each segment this is

normal)

When no data is present (P=0):

- 1st TAG slot is (partially) driven, depending on the ownership / data flow control protocol phases.
- 2nd TAG slot is either driven to all-zero, or undriven, that is, it always reads out as all-zero.
- All other slots are unused and undriven

---

**NOTE:**

- SL should be configured so that the maximum number of DATA slots is even, otherwise there will always be at least one unused slot.
  - Largest supported SL is not limited by the maximum FIFO word size (that is, the agent's actual implementation), unlike all other transfer protocols, as several FIFO words can be concatenated inside the same data segment.
  - Not all combinations of legal field values are legal. The condition is that TAG, AUX and DATA slots must fit within the allocated segment, that is, the sum must not be (strictly) greater than SL.
    - NVB = 15 (15 valid data bytes = 30 slots) cannot be used, because 2 TAG slots + 30 DATA slots (NVB x 2) = 32 > 31 segment slots (SL max).
    - SL = 1,2,3 are forbidden, since the smallest extended asynchronous segment is 2 TAG slots + 1 DATA byte (NVB = 1) = 4 slots long.
    - ...
- 

#### 23.9.4.8.5.2 Programming Specificities for Half-Duplex Protocol

Contrary to all other standard transfer protocols (TP), the half-duplex asynchronous ones (plain or extended) are bidirectional, that is, DATA slots can go in both directions. This means that 1 half-duplex asynchronous port (mapped on 1 half-duplex asynchronous data channel) uses 2 unidirectional data channel agents (FIFOs and logic) in the same controller.

The two agents are of opposite directions (1 TX + 1 RX), and the configuration register fields must always be programmed simultaneously and with matching values for both, so that both own the exact same slots (the data channel's allocated distribution).

#### 23.9.4.9 Paired-up channels

Two (or more) data channels with common properties can be paired up to operate in lockstep, like e.g. the left and right channels of a stereo audio flow: paired up data samples are generated together by a data source and need to arrive together at the data sink after going over the SLIMbus interface.

Assuming that pairing up has been implemented in both the sink and the source SLIMbus controllers of the data channel, it is enabled by configuring the channel as follows:

- Same segment windows: same channel rate, also expressed as segment interval (expressed in slots). This is encoded in the channel's Segment Distribution ([SLIMBUS\\_DCT\\_CONFIG2\\_j](#) / [SLIMBUS\\_DCR\\_CONFIG2\\_j](#)[11:0] SD).
- Same Segment Length ([SLIMBUS\\_DCT\\_CONFIG1\\_j](#) / [SLIMBUS\\_DCR\\_CONFIG1\\_j](#)[9:5] SL)
- Segment offset (also encoded in SD) is necessarily different, otherwise the two channels would be scheduled in the same slots. SD is therefore different.
- Sequential Channel Numbers (CN) are assigned to the channels to be paired up
- Data channel numbering and scheduling are defined by the active manager device, through the broadcasts of the NEXT\_DEFINE\_CHANNEL() message.

Contents must be aligned, as follows:

- Same Transfer Protocol ([SLIMBUS\\_DCT\\_CONFIG1\\_j](#) / [SLIMBUS\\_DCR\\_CONFIG1\\_j](#)[3:0] TP). Asynchronous protocols (extended or not) do not support pairing.
- Same Aux Format ([SLIMBUS\\_DCT\\_CONFIG1\\_j](#) / [SLIMBUS\\_DCR\\_CONFIG1\\_j](#)[19:16] AF)
- Same Data Length ([SLIMBUS\\_DCT\\_CONFIG1\\_j](#) / [SLIMBUS\\_DCR\\_CONFIG1\\_j](#)[14:10] DL)
- Pairing can be enabled through the Channel Link ([SLIMBUS\\_DCT\\_CONFIG1\\_j](#) /



SLIMBUS\_DCR\_CONFIG1\_j[15] CL) data channel content bit. Pairing between channel #N-1 and channel #N is enabled by setting the Channel Link (CL) content bit to 1 on channel #N. For example, to pair up channels #1 and #2, CL = 1 in channel #2 (thus linking it to #1) and CL = 0 in channel #1 (thus keeping it separate from channel #0).

- Data channel contents are defined by active manager device, through the broadcasting of the NEXT\_DEFINE\_CONTENT() and CHANGE\_CONTENT() messages.

#### 23.9.4.9.1 Paired-up behaviour for Isochronous TP

Data is by definition present in all segments, so that isochronous data channels with the same channel rate (as extracted from SD) are naturally paired up in normal operation. However, pairing also ensures that transmit underflows are also coupled:

- On the source side, if one of the channels has no data to transmit (TX underflow error, reported by IRQ) the other channels won't transmit any data either (even if they have some). This prevents the transmitter from getting out of sync.
- On the sink side, the underflow error will not be detected since no presence indication is transmitted, but the channels will remain in sync when the correct data flow resumes.

#### 23.9.4.9.2 Paired-up behaviour for Pushed TP

Data presence in each segment is flagged by the STR/P bits (strobe/presence) driven by the data source in the TAG slot. On the source side, paired up channels must have the same P value in a given segment window, that is, data must only be transmitted (STR=P=1) when both channels have data to send. No requirements on the sink side.

#### 23.9.4.9.3 Paired-up behaviour for Pulled TP

Data presence in a segment is requested by the SRQ TAG bit (sample request), driven by the sink, and confirmed by the P TAG bit (presence), driven by the source, so that SRQ and P are always equal in normal operation. The main pairing up requirement is therefore that the SRQ are aligned for all channel sinks, that is, that data is requested on all channels only if all sinks are ready to receive it, otherwise it is requested on none.

Pairing also ensures that transmit underflows are also coupled between channels: if for some reason a source cannot fulfill a request to drive data (SRQ=1, P=0: transmit underflow error, reported by IRQ), none of the other channels must drive data either, and all must flag P=0. The source side can only implement correct pairing if the sink side is correct in the first place; indeed, unless an underflow occurs, it is sufficient that the sink side implements pairing for the channels to be paired.

#### 23.9.4.10 Device and Addressing Management

One of the manager device tasks is to keep track of devices present on the bus and to enumerate them, that is, assign a (different) Logical Address (LA) to each.

The manager device itself is implicitly enumerated by default, to Logical Address (LA) 0xFF. This is the only case where a device enumerates without ASSIGN\_LOGICAL\_ADDRESS message transmitted over the bus. Therefore for the manager device the following default configuration must be applied :

- SLIMBUS\_DEV\_LA\_i[7:0] LA = 0xFF
- SLIMBUS\_DEV\_LA\_i[8] LA\_ENABLE = 1
- SLIMBUS\_DEV\_LA\_i[9] EA\_ENABLE = 1 (although manager is never addressed by Enumeration Address, EA)

The SLIMbus controller implements several devices, each of which is identified by its address. Addresses are used to exchange control messages over the shared message channel (SMC). The controller will recognize, store and respond (e.g. acknowledge) to any message on the SMC targeted at one of the component's devices in the RX message FIFO.

The Enumeration Addresses (EA) composition is described in [Table 23-667](#), with:

- MI, PC and IV are Software-programmable and have the same value for all devices of the component with SLIMBUS\_CMP\_MI\_PC[15:0] MI, SLIMBUS\_CMP\_MI\_PC[31:16] PC and SLIMBUS\_CMP\_IV[7:0] IV.

- DI is hardwired and unique for each device within the component. It is given by [SLIMBUS\\_DCT\\_MAP\\_j](#) / [SLIMBUS\\_DCR\\_MAP\\_j](#)[7:0] DI bit field.
- Each device individual EA is Software-accessible, for reference, in read-only registers ([SLIMBUS\\_DEV\\_EA\\_LO\\_i](#) and [SLIMBUS\\_DEV\\_EA\\_HI\\_i](#) registers)

**Table 23-667. Enumeration Address (EA) makeup**

EA Subfield	Size (bits)	Top Bit	Bottom Bit	Source
MI: Manufacturer ID	16	47	32	<a href="#">SLIMBUS_CMP_MI_PC</a> [15:0] MI
PC: Product Code	16	31	16	<a href="#">SLIMBUS_CMP_MI_PC</a> [31:16] PC
DI: Device Index	8	15	8	hardwired
IV: Instance Value	8	7	0	<a href="#">SLIMBUS_CMP_IV</a> [7:0] IV
Total: EA	48	47	0	<a href="#">SLIMBUS_DEV_EA_HI_i</a> / <a href="#">SLIMBUS_DEV_EA_LO_i</a>

The Logical Address (LA) assigned by the manager range from 0x00 to 0xEF, with the exception of the active manager located by definition at LA = 0xFF. It is configured through [SLIMBUS\\_DEV\\_LA\\_i](#)[7:0] LA for device m.

By default, a device uses a unique, static 48-bit Enumeration Address (EA). The EA is used only for enumeration, that is, the acquisition a dynamic Logical Address (LA) assigned by the active manager device.

---

**NOTE:** The port number is given by [SLIMBUS\\_DCT\\_MAP\\_j](#) / [SLIMBUS\\_DCR\\_MAP\\_j](#)[13:8] PN bit field.

---

### 23.9.4.11 Message Management

#### 23.9.4.11.1 Transmission

##### 23.9.4.11.1.1 TX Message Sequence

The message transmission software sequence is the following:

- Initial state:
  - [SLIMBUS\\_SMT\\_CONTROL](#)[0] MESSAGE\_ENABLE = 0: No ongoing TX message. This also implies that TX message FIFO can receive another message.
  - [SLIMBUS\\_SMT\\_INFO](#)[5:0] = 0 : TX message event flags must be cleared (to 0) by Software by writing 1 to the corresponding bit(s). Failing to meet this condition will not prevent the message transmission, but will prevent a proper readout of the message transmission's outcome
  - Optional: If the IRQ method is to be used (instead of polling), it must be cleared and enabled, that is, [SLIMBUS\\_CMP\\_IRQENABLE\\_SET](#)[2] SMT\_INFO\_EN = 1 and [SLIMBUS\\_CMP\\_IRQSTATUS](#)[2] SMT\_INFO = 0. For more information about interrupts see [Section 23.9.4.12](#)
- FIFO loading:
  - The message is written byte by byte to register [SLIMBUS\\_SMT\\_MESSAGE](#)[7:0]. SLIMbus endianness is retained (big-endian) that is, serial message starts with the MSB (bit 7) of the first byte written into the FIFO and ends with the LSB (bit 0) of the last written byte.
  - The minimum message size is 6 bytes (5 entered to the FIFO), for a short-arbitration (that is, logical source address [SLIMBUS\\_DEV\\_LA\\_i](#)[7:0] LA = 1 byte), broadcast (that is, no destination address), no-payload message.
  - The maximum theoretical message size is 39 bytes (38 entered to the FIFO), for a long-arbitration (that is, enumeration source address EA = 6 bytes), maximum remaining length (RL = 5b11111 = 31 bytes) message. The TX message FIFO can hold 32 bytes, that is, most messages (of 33 bytes or less).
  - An overflow event is flagged ([SLIMBUS\\_SMT\\_INFO](#)[1] OVERFLOW is set) if too many bytes are written in. The TX message FIFO is then cleared before another TX message sequence attempt is made.

- Transmission enabling:
  - Once the message has been loaded into the TX message FIFO, software must write 1 on field [SLIMBUS\\_SMT\\_CONTROL\[0\] MESSAGE\\_ENABLE](#).
- Transmission:
  - From that point on, the device will attempt to transmit the message stored in the FIFO. Actual transmission may be delayed because of Message Sync acquisition time (if not already operational), Ongoing message on the SMC, and Arbitration(s) lost to higher-priority messages.
- Transmission completion and reporting:
  - Transmission is considered complete if one of the following -mutually exclusive- conditions is met, and the corresponding bit of [SLIMBUS\\_SMT\\_INFO](#) is set : Negative response [SLIMBUS\\_SMT\\_INFO\[3\] NACK = 1](#), or Positive response : [SLIMBUS\\_SMT\\_INFO\[2\] PACK = 1](#), or No response: [SLIMBUS\\_SMT\\_INFO\[4\] NORE = 1](#), or Illegal response: [SLIMBUS\\_SMT\\_INFO\[5\] UDEF = 1](#), or Transmit collision: [SLIMBUS\\_SMT\\_INFO\[0\] MC\\_TX\\_COL = 1](#).
  - If enabled ([SLIMBUS\\_CMP\\_IRQENABLE\\_SET\[2\] SMT\\_INFO\\_EN = 1](#)), an interrupt request flag is set ([SLIMBUS\\_CMP\\_IRQSTATUS\[2\] SMT\\_INFO = 1](#)) and the IRQ line asserted.
  - Additionally, bit [SLIMBUS\\_SMT\\_CONTROL\[0\] MESSAGE\\_ENABLE](#) must auto-clear. Polling this bit will also work as an alternative to waiting for the IRQ.
  - Following the transmission, all message bytes have been removed from the TX message FIFO.

#### **23.9.4.11.1.2 TX Message FIFO Clearing**

The TX message FIFO must not be cleared as part of the normal message transmission operation, between messages. However, the clearing becomes necessary in case of error:

- FIFO overflow: the bit [SLIMBUS\\_SMT\\_INFO\[1\] OVERFLOW](#) is set
- FIFO underflow: the bit [SLIMBUS\\_SMR\\_INFO\[1\] UNDERFLOW](#) is set
- TX collision: the bit [SLIMBUS\\_SMT\\_INFO\[0\] MC\\_TX\\_COL](#) or the bit [SLIMBUS\\_SMR\\_INFO\[0\] MC\\_TX\\_COL](#) is set
- Undefined error condition, unknown FIFO content

Clearing is done by writing 0 to bit field [SLIMBUS\\_SMT\\_CONTROL\[0\] MESSAGE\\_ENABLE](#).

#### **23.9.4.11.2 Message reception**

##### **23.9.4.11.2.1 Message Responding**

The message responding and message receiving are separate but intertwined processes. Message responding starts as soon as a message is received, and consists in sending back information, including PACK or NACK.

##### **23.9.4.11.2.2 RX Message Sequence**

The message reception software sequence is the following:

- Initial state
  - [SLIMBUS\\_SMR\\_INFO = 0x0](#): Pending (at 1) RX message event flags must be cleared (to 0) by Software by writing 1 to the corresponding bit(s).
  - Optional: If the IRQ method is to be used (instead of polling), it must be cleared and enabled, that is, [SLIMBUS\\_CMP\\_IRQENABLE\\_SET\[3\] SMR\\_INFO\\_EN = 1](#) and [SLIMBUS\\_CMP\\_IRQSTATUS\[3\] SMR\\_INFO = 0](#). For more information about interrupts see [Section 23.9.4.12](#)
- Message reception
  - Message reception requires no intervention: Messages with target addresses enabled in [SLIMBUS\\_DEV\\_LA\\_i](#) registers are automatically loaded into the RX message FIFO.
- FIFO unloading
  - The message is read byte by byte off the FIFO and put into register [SLIMBUS\\_SMR\\_MESSAGE\[7:0\]](#). SLIMbus endianness is used (big-endian) that is, serial message starts with the MSB (bit 7) of the first byte written into the FIFO and ends with the LSB (bit 0) of the last written byte.

- The minimum message size is 6 bytes, for a short-arbitration (that is, logical source address LA = 1 byte), broadcast (that is, no destination address), no-payload message.
- The maximum theoretical message size is 39 bytes, for a long-arbitration (that is, enumeration source address EA = 6 bytes), maximum remaining length (RL = 5b11111 = 31 bytes) message. The RX message FIFO can hold 32 bytes, that is, most message types.
- An overflow event is flagged ([SLIMBUS\\_SMR\\_INFO\[1\] OVERFLOW](#) bit is set) if too many bytes are received, which will overwrite previous bytes.
- An underflow event is flagged ([SLIMBUS\\_SMR\\_INFO\[1\] UNDERFLOW](#) bit is set) if too many bytes are read out. The RX message FIFO is cleared before another RX message sequence attempt is made.
- Message decoding and servicing
  - This phase takes place in parallel with the unloading phase above. In some cases, a partial decoding can lead the software to skip the end of the unloading and flush the FIFO. The software will check Device Association (thanks to the [SLIMBUS\\_SMR\\_INFO\[2\] RECEIVED\\_MESSAGE](#) bit) and Message Support.

#### 23.9.4.11.2.3 RX Message FIFO Clearing

- While the software is reading the message off the FIFO, the receive message agent unconditionally answers NACK, forcing remote side to retransmit any new incoming message.
- Only once the message has been read out of the FIFO, the software writes 1 on [SLIMBUS\\_SMR\\_INFO\[2\] RECEIVED\\_MESSAGE](#) bit. This will automatically clear the FIFO.

The RX message FIFO is cleared as part of the normal message reception operation, between messages, although it is not strictly necessary. However, the clearing becomes compulsory in case of error:

- FIFO overflow: the bit [SLIMBUS\\_SMT\\_INFO\[1\] OVERFLOW](#) is set
- FIFO underflow: the bit [SLIMBUS\\_SMR\\_INFO\[1\] UNDERFLOW](#) is set
- Undefined error condition, unknown FIFO content

Clearing is done by writing 1 to register bit field [SLIMBUS\\_SMR\\_INFO\[2\] RECEIVED\\_MESSAGE](#) bit.

#### 23.9.4.12 Interrupt Service Routines

This section lists specific software programming sequences, to be applied at specific phases of the SLIMbus protocol. This section covers the low-level software service of IRQs. More information about interrupts in the device can be found in [Chapter 17, Interrupt Controllers](#).

The SLIMbus controller has a single IRQ output line, controlled through 9-bit registers [SLIMBUS\\_CMP\\_IRQSTATUS](#), [SLIMBUS\\_CMP\\_IRQENABLE\\_SET](#) and [SLIMBUS\\_CMP\\_IRQENABLE\\_CLR](#). Each bit of this 1st level IRQ management points to a 2nd level information register. There are thus 9 such information registers, each connected with a specific submodule (framer device, interface device, message receiver, message transmitter, and one per generic data device).

2nd level information register and status register are written independently each other. When an event occurs, both bits are set to 1, and when the software writes 1 on either register the corresponding bit is cleared to 0. IRQ line is the status register output.

- When an event takes place, the corresponding 2nd level information bit must be set to 1 (if already 1: no change)
- When an event takes place, the corresponding 1st level IRQ status bit must be set to 1 (if enabled and if not already 1) and trigger an interrupt (if not already active). Note that this does not depend on the information bit's value: if the information bit is already high, the event must still potentially set the IRQ status bit.
- Writing 1 to a set IRQ status bit clears the bit and deasserts the IRQ line (if another status bit is set, the IRQ line will reassert immediately). Note that the clearing of the IRQ status does not depend on the information bit, that is, it will work even if the information bit has not been cleared.
- Writing 1 to a set information bit will clear the bit, but have no impact on the IRQ status

**CAUTION**

The software sequence should always be as follows:

1. Clear the status bit
2. Read the relative information register
3. When needed and allowed, clear the information register

**23.9.4.12.1 Interrupt Requests**

Table 23-668 lists the event flags, and their mask, that can cause module interrupts.

**Table 23-668. Events**

Interrupt Line	Event Flag	Event Mask
SLIMBUS1_IRQ / SLIMBUS2_IRQ	SLIMBUS_CMP_IRQSTATUS_RAW[11:0] and SLIMBUS_CMP_IRQSTATUS[11:0]	SLIMBUS_CMP_IRQENABLE_SET[11:0] and SLIMBUS_CMP_IRQENABLE_CLR[11:0]

**23.9.4.12.2 Possible Interrupts**

Table 23-669 lists all the interrupts that can occur and the related registers.

**Table 23-669. Interrupts summary**

Interrupt	Register	Field name	Description
Transmit collision in data channel port	SLIMBUS_DCT_INFO_j[0]	DATA_TX_COL	TX collision in TX data channel agent
Data segment overlap	SLIMBUS_FL_INFO[4]	DATA_SLOT_OVERLAP	Overlap detected between a data segment and either the control space or another segment.
Lost message synchronization	SLIMBUS_FL_INFO[2]	LOST_MS	Message Synchronization lost
Lost superframe synchronization	SLIMBUS_FL_INFO[1]	LOST_SFS	Super Frame Synchronization lost
Lost frame synchronization	SLIMBUS_FL_INFO[0]	LOST_FS	Frame Synchronization lost
Transmit collision in message body, Transmit collision in message response	SLIMBUS_SMT_INFO[0]	MC_TX_COL	Transmit collision in Message Channel during message transmission.
Transmit collision in guide channel	SLIMBUS_FR_INFO[2]	GC_TX_COL	Collision during guide byte transmit
Transmit collision in frame sync symbol	SLIMBUS_FR_INFO[1]	FI_TX_COL	Collision during Framing Information transmit
Transmit collision in frame sync symbol	SLIMBUS_FR_INFO[0]	FS_TX_COL	Collision during Frame Sync symbol transmit
Framer device un-activated	SLIMBUS_FR_INFO[4]	FRAMER_UNACTIVATED	Set on ACTIVE_FRAMER information element deassertion
Framer device activated	SLIMBUS_FR_INFO[3]	FRAMER_ACTIVATED	Set on ACTIVE_FRAMER information element assertion
Reconfigured	SLIMBUS_FL_INFO[6]	RECONFIGURED	Reconfiguration boundary crossed
Restart after clock pause	SLIMBUS_FR_INFO[5]	CLOCK_RESTART	Clock restart request event detected
TX message FIFO overflow	SLIMBUS_SMT_INFO[1]	OVERFLOW	Overflow in TX message FIFO
Received positive acknowledge message response	SLIMBUS_SMT_INFO[2]	PACK	TX message was PACK: all recipient devices accepted the message.

**Table 23-669. Interrupts summary (continued)**

Interrupt	Register	Field name	Description
Received negative acknowledge message response	<a href="#">SLIMBUS_SMT_INFO</a> [3]	NACK	TX message was NACK: at least one of the recipient devices requested message retransmission.
Received no message response	<a href="#">SLIMBUS_SMT_INFO</a> [4]	NORE	TX message was NORE: no or all-zero message response
Received undefined message response	<a href="#">SLIMBUS_SMT_INFO</a> [5]	UDEF	TX message was UDEF: undefined response, protocol error.
RX message FIFO underflow	<a href="#">SLIMBUS_SMR_INFO</a> [1]	UNDERFLOW	Underflow in RX message FIFO
Message received	<a href="#">SLIMBUS_SMR_INFO</a> [2]	RECEIVED_MESSAGE	Message available in the RX message FIFO. Clear bit to clear the FIFO and allow the reception of further messages.
Reconfigure_now message received	<a href="#">SLIMBUS_SMR_INFO</a> [3]	RECEIVED_RECONFIGURE	RECONFIGURE_NOW message received, available in RX message FIFO.
Data channel FIFO overflow	<a href="#">SLIMBUS_DCT_INFO_j</a> [1], <a href="#">SLIMBUS_DCR_INFO_j</a> [1]	OVERFLOW	Overflow in TX data channel agent FIFO.
Data channel FIFO underflow	<a href="#">SLIMBUS_DCT_INFO_j</a> [2], <a href="#">SLIMBUS_DCR_INFO_j</a> [2]	UNDERFLOW	Underflow in TX data channel agent FIFO.

### 23.9.4.12.3 Interface Device ISRs

The Interface Device ISRs are applied when a bit of status register [SLIMBUS\\_FL\\_INFO](#) is set, which also triggers the assertion of bit field [SLIMBUS\\_CMP\\_IRQSTATUS](#)[0] FL\_INFO and of the IRQ line, when enabled in bit field [SLIMBUS\\_CMP\\_IRQENABLE\\_SET](#)[0] FL\_INFO\_EN. Service routines must be applied and the FL\_INFO bit field cleared (by writing 1 to it).

#### 23.9.4.12.3.1 Lost frame synchronization (LOST\_FS)

##### 23.9.4.12.3.1.1 Root Event

This interrupt occurs when a loss of frame synchronization happens. All data channels automatically stop, that is, all active [SLIMBUS\\_DCT\\_CONFIG1\\_j](#) / [SLIMBUS\\_DCR\\_CONFIG1\\_j](#)[31] ENABLE bits are cleared. Component automatically reacquires frame, superframe and message synchronization.

##### 23.9.4.12.3.1.2 Interrupt Service Routine

Frame synchronization (FS) loss implies the loss of higher-level superframe and message synchronizations:

- Messages cannot be expected to be transmitted / received until the component is operational again. Additionally, any ongoing reconfiguration sequence must be interrupted and made invalid by MS loss. No action required if no reconfiguration sequence is ongoing.
- Implicit superframe synchronization loss triggers a reset of all component ports: see [Section 23.9.4.4.3.6](#).

This field, [SLIMBUS\\_FL\\_INFO](#)[0] LOST\_FS, is not cleared by software into the ISR. Instead, it stays set until the component receives a REQUEST\_CLEAR\_INFORMATION message by the remote manager (Receiving a message implies of course that message synchronization has been restored in the mean time). Only at this point, software can apply the message and clear the designated field.

### **23.9.4.12.3.2 Lost superframe synchronization (LOST\_SFS)**

#### **23.9.4.12.3.2.1 Root Event**

This interrupt occurs when a loss of superframe synchronization happens. The loss of lower-level frame synchronization also implies the loss of superframe synchronization, if already acquired, but does not trigger the assertion of LOST\_SFS Information Element. All data channels automatically stop, that is, all active [SLIMBUS\\_DCT\\_CONFIG1\\_j](#) / [SLIMBUS\\_DCR\\_CONFIG1\\_j](#)[31] ENABLE bits must be cleared. Component automatically reacquires superframe and message synchronization.

#### **23.9.4.12.3.2.2 Interrupt Service Routine**

Superframe synchronization (SFS) loss will trigger the reset of all ports of the component: see [Section 23.9.4.4.3.6](#).

Superframe synchronization loss implies the loss of message synchronization, that is, messages cannot be expected to be transmitted / received until the component is operational again. Additionally, any ongoing reconfiguration sequence must be interrupted and made invalid by MS loss. No action required if no reconfiguration sequence is ongoing.

This field, [SLIMBUS\\_FL\\_INFO](#)[1] LOST\_SFS, is not cleared by software into the ISR. Instead, it stays set until the component receives a REQUEST\_CLEAR\_INFORMATION message by the remote manager (Receiving a message implies of course that message synchronization has been restored in the mean time). Only at this point, software can apply the message and clear the designated field.

### **23.9.4.12.3.3 Lost message synchronization (LOST\_MS)**

#### **23.9.4.12.3.3.1 Root Event**

This interrupt occurs when a loss of message synchronization happens. The loss of either lower-level frame or superframe synchronization also implies the loss of message synchronization, if already acquired, but does not trigger the assertion of LOST\_MS Information Element. Component automatically reacquires message synchronization, by seeking for Guide Channel information.

#### **23.9.4.12.3.3.2 Interrupt Service Routine**

Any ongoing reconfiguration sequence must be interrupted and made invalid by the Message Synchronisation (MS) loss. No action required here if no reconfiguration sequence is ongoing.

This field, [SLIMBUS\\_FL\\_INFO](#)[2] LOST\_MS, is not cleared by software into the ISR. Instead, it stays set until the component receives a REQUEST\_CLEAR\_INFORMATION message by the remote manager, requiring to clear it (Receiving a message implies of course that message synchronization has been restored in the mean time). Only at this point, software can apply the message and clear the designated field.

### **23.9.4.12.3.4 Found message synchronization (FOUND\_MS)**

#### **23.9.4.12.3.4.1 Root Event**

This interrupt occurs when acquisition of message synchronization. In that state, messages may be transmitted and received over the SMC.

#### **23.9.4.12.3.4.2 Interrupt Service Routine**

If MS is reacquired, following a previous LOST\_MS event, then a REPORT\_INFORMATION message must be transmitted, as follows:

- Source: interface device
- Destination: SLIMbus manager device ([SLIMBUS\\_DEV\\_LA\\_i](#))
- Information Element: [SLIMBUS\\_FL\\_INFO](#)[2] LOST\_MS.

If MS is reacquired, following a previous LOST\_FS event, then a REPORT\_INFORMATION message must be transmitted, as follows:

- Source: interface device
- Destination: SLIMbus manager device (`SLIMBUS_DEV_LA_i[7:0]` LA = 0xFF)
- Information Element: `SLIMBUS_FL_INFO[2]` LOST\_FS.

If MS is reacquired, following a previous LOST\_SFS event, then a REPORT\_INFORMATION message must be transmitted, as follows:

- Source: interface device
- Destination: SLIMbus manager device (`SLIMBUS_DEV_LA_i[7:0]` LA = 0xFF)
- Information Element: `SLIMBUS_FL_INFO[2]` LOST\_SFS.

If MS is reacquired, following a previous MC\_TX\_COL event, then a REPORT\_INFORMATION message must be transmitted, as follows:

- Source: interface device
- Destination: SLIMbus manager device (`SLIMBUS_DEV_LA_i[7:0]` LA = 0xFF)
- Information Element: `SLIMBUS_SMT_INFO[0]` MC\_TX\_COL, `SLIMBUS_SMR_INFO[0]` MC\_TX\_COL.

If MS is reacquired, following a previous UDEF (Received undefined message response) event, the message that caused UDEF must be retransmitted until it succeeds or until the retransmit maximum (fully managed by software) is reached.

In any case, software must clear the field `SLIMBUS_FL_INFO[3]` FOUND\_MS. From that point on, any pending messages in the TX FIFO will be sent.

### 23.9.4.12.3.5 Data segment overlap (DATA\_SLOT\_OVERLAP)

#### 23.9.4.12.3.5.1 Root Event

This interrupt occurs when an internal conflict detected between several data channels or between a data channel and the control space, resulting from an incorrect scheduling (incorrect SD, SL for a given M). Note that no actual collision takes place on the bus.

#### 23.9.4.12.3.5.2 Interrupt Service Routine

A REPORT\_INFORMATION message must be transmitted, as follows:

- Sender: interface device
- Destination: SLIMbus manager device (`SLIMBUS_DEV_LA_i[7:0]` LA = 0xFF)
- Information Element: `SLIMBUS_FL_INFO[4]` DATA\_SLOT\_OVERLAP.

### 23.9.4.12.3.6 Reconfigured (RECONFIGURED)

#### 23.9.4.12.3.6.1 Root Event

This interrupt occurs when crossing of a reconfiguration boundary. In case of clock pause, where the SLIMbus clock stops at the boundary (1st rising edge of the new superframe), the bit (and therefore the IRQ) is asserted at the clock pause, not at the clock restart.

#### 23.9.4.12.3.6.2 Interrupt Service Routine

Completes the software-controller reconfiguration sequence. No reconfiguration must be applied to the control registers between the RECONFIGURE\_NOW reception and the reconfiguration boundary. A attempt by the manager to start a new reconfiguration sequence during that interval is an error.

In the case of framer release handover and/or root frequency change, the active framer must start using a new clock input selection at the reconfiguration boundary (as per register field `SLIMBUS_FR_CLOCK_SOURCE[5:3]` CLKSEL). All non-selected SLIMbus clock inputs of active framer components and all SLIMbus clock inputs for clock receiver components (inactive framers), and that may have been used until the boundary, can be turned off at that point.



Field [SLIMBUS\\_FL\\_INFO\[6\]](#) RECONFIGURED must be software cleared (by writing 1 to it).

#### **23.9.4.12.4 Framer Device ISRs**

The Framer Device ISRs are applied when a bit of status register FR\_INFO is set, which also triggers the assertion of bit field [SLIMBUS\\_CMP\\_IRQSTATUS\[1\]](#) FR\_INFO and of the IRQ line, when enabled in bit field [SLIMBUS\\_CMP\\_IRQENABLE\\_SET\[1\]](#) FR\_INFO\_EN. Service routines must be applied, and the fr\_info bit field cleared (by writing 1 to it). Some interrupt generating events are collisions, which occur when the framer detects on DATA line a different value than it's actively driving.

##### **23.9.4.12.4.1 Transmit collision in frame sync symbol (FS\_TX\_COL)**

###### **23.9.4.12.4.1.1 Root Event**

Transmit collision detected while driving the frame sync symbol (as active framer) of the framing channel.

###### **23.9.4.12.4.1.2 Interrupt Service Routine**

A REPORT\_INFORMATION message must be transmitted, as follows:

- Sender: framer device
- Destination: SLIMbus manager device ([SLIMBUS\\_DEV\\_LA\\_i\[7:0\]](#) LA = 0xFF)
- Information Element: [SLIMBUS\\_FR\\_INFO\[0\]](#) FS\_TX\_COL.

This field is not cleared by software into the ISR. Instead, it stays set until the component receives a REQUEST\_CLEAR\_INFORMATION message by the remote manager (Receiving a message implies of course that message synchronization has been restored in the mean time). Only at this point, software can apply the message and clear the designated field.

##### **23.9.4.12.4.2 Transmit collision in frame sync symbol (FI\_TX\_COL)**

###### **23.9.4.12.4.2.1 Root Event**

Transmit collision detected while driving the framing information (as active framer) of the framing channel.

###### **23.9.4.12.4.2.2 Interrupt Service Routine**

A REPORT\_INFORMATION message must be transmitted, as follows:

- Sender: framer device
- Destination: SLIMbus manager device ([SLIMBUS\\_DEV\\_LA\\_i\[7:0\]](#) LA = 0xFF)
- Information Element: [SLIMBUS\\_FR\\_INFO\[1\]](#) FI\_TX\_COL.

This field is not cleared by software into the ISR. Instead, it stays set until the component receives a REQUEST\_CLEAR\_INFORMATION message by the remote manager (Receiving a message implies of course that message synchronization has been restored in the mean time). Only at this point, software can apply the message and clear the designated field.

##### **23.9.4.12.4.3 Transmit collision in guide channel (GC\_TX\_COL)**

###### **23.9.4.12.4.3.1 Root Event**

Transmit collision detected while driving the guide byte (as active framer), that is, during the guide channel.

###### **23.9.4.12.4.3.2 Interrupt Service Routine**

A REPORT\_INFORMATION message must be transmitted, as follows:

- Sender: framer device
- Destination: SLIMbus manager device ([SLIMBUS\\_DEV\\_LA\\_i\[7:0\]](#) LA = 0xFF)

- Information Element: [SLIMBUS\\_FR\\_INFO](#)[2] GC\_TX\_COL.

This field is not cleared by software into the ISR. Instead, it stays set until the component receives a REQUEST\_CLEAR\_INFORMATION message by the remote manager (Receiving a message implies of course that message synchronization has been restored in the mean time). Only at this point, software can apply the message and clear the designated field.

#### **23.9.4.12.4.4 Framer device activated (FRAMER\_ACTIVATED)**

##### **23.9.4.12.4.4.1 Root Event**

Acquisition of active framer role by the component. This event is only possible on bus boot (when default active framer) and incoming framer handover (when inactive framer).

##### **23.9.4.12.4.4.2 Interrupt Service Routine**

Assert software implemented ACTIVE\_FRAMER information element.

Field [SLIMBUS\\_FR\\_INFO](#)[4] FRAMER\_UNACTIVATED must be software cleared (by writing 1 to it).

Note that the ACTIVE\_FRAMER information element's assertion does not trigger the transmission by the framer device of a REPORT\_INFORMATION message, since the information is already "publicized" by the framer handover process.

#### **23.9.4.12.4.5 Framer device un-activated (FRAMER\_UNACTIVATED)**

##### **23.9.4.12.4.5.1 Root Event**

Loss of active framer role by the component. This event is only possible on bus shutdown and outgoing framer handover (when active framer).

##### **23.9.4.12.4.5.2 Interrupt Service Routine**

Deassert software-implemented ACTIVE\_FRAMER information element.

Field [SLIMBUS\\_FR\\_INFO](#)[3] FRAMER\_ACTIVATED must be software cleared (by writing 1 to it).

Note that the ACTIVE\_FRAMER information element's deassertion does not trigger the transmission by the framer device of a REPORT\_INFORMATION message, since the information is already "publicized" by the framer handover process.

#### **23.9.4.12.4.6 Restart after clock pause (CLOCK\_RESTART)**

##### **23.9.4.12.4.6.1 Root Event**

Transition detected on the DATA line during a clock pause, driven by any of the connected components and seen by the clock source (active framer) as a restart request. Note that this field is set for all clock restarts, even if the local framer is inactive.

##### **23.9.4.12.4.6.2 Interrupt Service Routine**

Field [SLIMBUS\\_FR\\_INFO](#)[5] CLOCK\_RESTART must be software cleared (by writing 1 to it).

---

**NOTE:** Restart event is a source of asynchronous wakeup.

---

#### **23.9.4.12.5 Shared Message Channel Transmit ISRs**

These ISRs are applied when a bit of status register SMT\_INFO is set, which also triggers the assertion of bit field [SLIMBUS\\_CMP\\_IRQSTATUS](#)[2] SMT\_INFO and of the IRQ line, when enabled in bit field [SLIMBUS\\_CMP\\_IRQENABLE\\_SET](#)[2] SMT\_INFO\_EN. Service routines must be applied, and the fr\_info bit field cleared (by writing 1 to it)

### **23.9.4.12.5.1 Transmit collision in message body (MC\_TX\_COL)**

#### **23.9.4.12.5.1.1 Root Event**

Transmit collision detected while transmitting a message, that is, writing any subfield except message response (arbitration, header, payload, or message integrity). The device immediately (before next cell) stop transmission and the component automatically lose message synchronization. The message has to be sent again, once message synchronization has been regained.

#### **23.9.4.12.5.1.2 Interrupt Service Routine**

This field, [SLIMBUS\\_SMT\\_INFO\[0\]](#) MC\_TX\_COL, is not cleared by software into the ISR. Instead, it stays set until the component receives a REQUEST\_CLEAR\_INFORMATION message by the remote manager (Receiving a message implies of course that message synchronization has been restored in the mean time). Only at this point, software can apply the message and clear the designated field.

### **23.9.4.12.5.2 TX message FIFO overflow (OVERFLOW)**

#### **23.9.4.12.5.2.1 Root Event**

Overflow occurs when too many bytes are written into the TX message FIFO by the software, either because the entered message is too long or because the FIFO still contains bytes from previous messages. The state of the FIFO after an overflow event is undetermined, but the other FIFOs are unaffected.

#### **23.9.4.12.5.2.2 Interrupt Service Routine**

TX message FIFO must be cleared as explained in [Section 23.9.4.11.1.2](#).

Field [SLIMBUS\\_SMT\\_INFO\[1\]](#) OVERFLOW must be software cleared (by writing 1 to it).

Message transmission must resume normally.

### **23.9.4.12.5.3 Received positive acknowledge message response (PACK)**

#### **23.9.4.12.5.3.1 Root Event**

Message written in the TX message FIFO was transmitted after unlimited, automatic arbitration retries, and removed from the FIFO. Received message response (MR) is PACK.

#### **23.9.4.12.5.3.2 Interrupt Service Routine**

No action required for this field: [SLIMBUS\\_SMT\\_INFO\[2\]](#) PACK. FIFO is now available for another message.

### **23.9.4.12.5.4 Received negative acknowledge message response (NACK)**

#### **23.9.4.12.5.4.1 Root Event**

Message written in the TX message FIFO was transmitted after unlimited, automatic arbitration retries, and removed from the FIFO. Received message response (MR) is NACK.

#### **23.9.4.12.5.4.2 Interrupt Service Routine**

Current message must be eventually retransmitted until it succeeds or until the retransmit maximum (fully managed by software) is reached.

Field [SLIMBUS\\_SMT\\_INFO\[3\]](#) NACK must be software cleared (by writing 1 to it).

FIFO is available for another message, which can be the same or another one.

### **23.9.4.12.5.5 Received no message response (NORE)**

#### **23.9.4.12.5.5.1 Root Event**

Message written in the TX message FIFO was transmitted after unlimited, automatic arbitration retries, and removed from the FIFO. Received message response (MR) is NORE.

#### **23.9.4.12.5.5.2 Interrupt Service Routine**

Current message must be eventually retransmitted until it succeeds or until the retransmit maximum (fully managed by software) is reached.

Field [SLIMBUS\\_SMT\\_INFO](#)[4] NORE must be software cleared (by writing 1 to it).

FIFO is available for another message, which can be the same or another one.

### **23.9.4.12.5.6 Received undefined message response (UDEF)**

#### **23.9.4.12.5.6.1 Root Event**

Message written in the TX message FIFO was transmitted after unlimited, automatic arbitration retries, and removed from the FIFO. Received message response (MR) is UDEF, that is, one of several undefined MR codes that are not combinations of legal responses, implying that an error has taken place. Component loses message synchronization, and reacquires it automatically.

#### **23.9.4.12.5.6.2 Interrupt Service Routine**

Field [SLIMBUS\\_SMT\\_INFO](#)[5] UDEF must be software cleared (by writing 1 to it).

FIFO is available for another message, which can be the same or another one.

### **23.9.4.12.5.7 TX message FIFO underflow (UNDERFLOW)**

#### **23.9.4.12.5.7.1 Root Event**

Underflow takes place when the message transmission logic attempts to read more bytes than the FIFO contains. The state of the FIFO after an underflow event is undetermined, but the other FIFOs are unaffected.

#### **23.9.4.12.5.7.2 Interrupt Service Routine**

TX message FIFO must be cleared as explained in [Section 23.9.4.11.1.2](#).

Field must be software-cleared (by writing 1 to it).

Message transmission must resume normally, making sure the error is not repeated.

### **23.9.4.12.6 Shared Message Channel Receive ISRs**

These ISRs are applied when a bit of status register [SMR\\_INFO](#) is set, which also triggers the assertion of bit field [SLIMBUS\\_CMP\\_IRQSTATUS](#)[3] [SMR\\_INFO](#) and of the IRQ line, when enabled in bit field [SLIMBUS\\_CMP\\_IRQENABLE\\_SET](#)[3] [SMR\\_INFO\\_EN](#). Service routines must be applied, and the [FR\\_INFO](#) bit field cleared (by writing 1 to it)

### **23.9.4.12.6.1 Transmit collision in message response (MC\_TX\_COL)**

#### **23.9.4.12.6.1.1 Root Event**

Transmit collision detected while receiving a message, that is, writing the message response subfield. The component automatically loses message synchronization, and eventually reacquires it.

### **23.9.4.12.6.1.2 Interrupt Service Routine**

The field [SLIMBUS\\_SMR\\_INFO\[0\] MC\\_TX\\_COL](#) must not be unconditionally software cleared, as it implements a Boolean IE that can only be cleared under specific conditions (and together with the [SLIMBUS\\_SMT\\_INFO\[0\] MC\\_TX\\_COL](#)).

### **23.9.4.12.6.2 RX message FIFO underflow (UNDERFLOW)**

#### **23.9.4.12.6.2.1 Root Event**

Underflow takes place when too many bytes are read out of the RX message FIFO by the software, because of an error when estimating the message length, as defined in [Section 23.9.4.11.2.2](#). The state of the FIFO after an underflow event is undetermined, but the other FIFOs are unaffected.

#### **23.9.4.12.6.2.2 Interrupt Service Routine**

RX message FIFO must be cleared as explained in [Section 23.9.4.11.2.3](#).

Field [SLIMBUS\\_SMR\\_INFO\[1\] UNDERFLOW](#) must be software-cleared (by writing 1 to it).

---

**NOTE:** This software error is not supposed to take place, since the message itself is assumed to be correct and could affect the correct running of the SLIMbus protocol, if the received message cannot be recovered.

---

### **23.9.4.12.6.3 RX message FIFO overflow (OVERFLOW)**

#### **23.9.4.12.6.3.1 Root Event**

Overflow occurs when too many bytes are received into the RX message FIFO because the entered message is too long. The state of the FIFO after an overflow event is undetermined, but the other FIFOs are unaffected.

#### **23.9.4.12.6.3.2 Interrupt Service Routine**

RX message FIFO must be cleared as explained in [Section 23.9.4.11.2.3](#).

Field must be software cleared (by writing 1 to it).

Message transmission must resume normally.

### **23.9.4.12.6.4 Message received (RECEIVED\_MESSAGE)**

#### **23.9.4.12.6.4.1 Root Event**

A message has been received and checked by the hardware: target address matches a local device (LA or EA), message length is coherent with AT/RL fields, integrity is correct for both PI and MI, all target devices acknowledged the message (PACK MR transmitted and received).

#### **23.9.4.12.6.4.2 Interrupt Service Routine**

Received message must be unloaded from RX message FIFO, parsed, and the appropriate action taken, as explained in [Section 23.9.4.11.2.2](#).

Field [SLIMBUS\\_SMR\\_INFO\[2\] RECEIVED\\_MESSAGE](#) must be software cleared (by writing 1 to it). This will also clear the message FIFO and allow the local component to respond PACK to new incoming messages.

### 23.9.4.12.6.5 Reconfigure\_now message received (RECEIVED\_RECONFIGURE)

#### 23.9.4.12.6.5.1 Root Event

A message has been received as in RECEIVED\_MESSAGE event above, with the additional condition that it has been recognized by hardware as a RECONFIGURE\_NOW() message. Note that by the time the ISR is executed, the reconfiguration may have already taken place.

#### 23.9.4.12.6.5.2 Interrupt Service Routine

Field [SLIMBUS\\_SMR\\_INFO\[3\]](#) RECEIVED\_CONFIGURE must be software cleared (by writing 1 to it).

### 23.9.4.12.7 Data Channel Port ISRs

These ISRs are applied when a bit of a status register [SLIMBUS\\_DCT\\_INFO\\_j](#) and/or [SLIMBUS\\_DCR\\_INFO\\_j](#) is set, which also triggers the assertion of bit field [SLIMBUS\\_CMP\\_IRQSTATUS\[11:4\]](#) DCT\_INFO\_j/DCR\_INFO\_j and of the IRQ line, when enabled in bit field [SLIMBUS\\_CMP\\_IRQENABLE\\_SET\[11:4\]](#) DCT/DCR\_INFO\_j. Service routines must be applied, and the [SLIMBUS\\_DCT\\_INFO\\_j](#) and/or [SLIMBUS\\_DCR\\_INFO\\_j](#) bit fields cleared (by writing 1 to it).

#### 23.9.4.12.7.1 Transmit collision in data channel port (DATA\_TX\_COL)

##### 23.9.4.12.7.1.1 Root Event

Transmit collision detected while writing to the considered data channel, in any of the TAG, AUX, or DATA fields. Note that transmit collisions are possible even for receive ports, as those port may also write some TAG cells (e.g. for pulled TP), where transmit collisions are possible. The device must immediately (before next cell) stop writing to the bus, only resuming on next segment.

##### 23.9.4.12.7.1.2 Interrupt Service Routine

The REPORT\_INFORMATION message must be transmitted, from the port's stereo or voice device (that is, with that device's LA in source address of arbitration field), to the active manager (that is, with manager's target [SLIMBUS\\_DEV\\_LA\\_j\[7:0\]](#) LA in header = 0xFF), and containing the information element (IE) of the same name (DATA\_TX\_COL).

The field [SLIMBUS\\_DCT\\_INFO\\_j\[0\]](#) DATA\_TX\_COL and/or [SLIMBUS\\_DCR\\_INFO\\_j\[0\]](#) DATA\_TX\_COL must not be unconditionally software cleared, as it implements a boolean IE that can only be cleared under specific conditions.

#### 23.9.4.12.7.2 Data channel FIFO overflow (OVERFLOW)

##### 23.9.4.12.7.2.1 Root Event

Overflow takes place when the data rate into the FIFO exceeds the data rate out of the FIFO for a sufficient amount of time, which can happen in both TX and RX ports. The state of the FIFO after an overflow event is undetermined, but the other FIFOs are unaffected.

##### 23.9.4.12.7.2.2 Interrupt Service Routine

The SLIMbus protocol does not specify the handling of such an event. However, since data integrity cannot be guaranteed following an that event, stopping the channel is an option.

Field [SLIMBUS\\_DCT\\_INFO\\_j\[1\]](#) / [SLIMBUS\\_DCR\\_INFO\\_j\[1\]](#) OVERFLOW must be software cleared (by writing 1 to it).

### **23.9.4.12.7.3 5.5.2 Data channel FIFO underflow (UNDERFLOW)**

#### **23.9.4.12.7.3.1 Root Event**

Overflow takes place when the data rate out of the FIFO exceeds the data rate into the FIFO for a sufficient amount of time, which can happen in both TX and RX ports. The state of the FIFO after an underflow event is undetermined, but the other FIFOs are unaffected.

#### **23.9.4.12.7.3.2 Interrupt Service Routine**

The SLIMbus protocol does not specify the handling of such an event. However, since data integrity cannot be guaranteed following an that event, stopping the channel is an option.

Field `SLIMBUS_DCT_INFO_j[2]` / `SLIMBUS_DCR_INFO_j[2]` UNDERFLOW must be software cleared (by writing 1 to it).

## 23.9.5 SLIMbus Programming Guide

### 23.9.5.1 SLIMbus Low-level Programming Models

This section covers the low-level programming sequences for configuration and usage of the SLIMbus module.

#### 23.9.5.1.1 Global Initialization

##### 23.9.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the SLIMbus module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the SLIMbus. See [Section 23.9.3](#) and [Section 23.9.2](#) for further information.

**Table 23-670. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	All SLIMbus clocks must be enabled. For SLIMbus clocks summary see <a href="#">Table 23-651</a> . See <a href="#">Chapter 3, Power, Reset, and Clock Management</a>
L4 Interconnect	For information about L4-PER interconnect configuration, see <a href="#">Chapter 13, Interconnect</a> .
sDMA	sDMA configuration must be done to enable the module DMA channel requests. See <a href="#">Chapter 16 SDMA</a> .
INTC	Cortex-A9 MPU and DSP subsystems interrupt controller must be configured to enable the interrupt request generation to the Cortex-A9 MPU, <a href="#">Chapter 4</a> or to the DSP subsystem, <a href="#">Chapter 5</a> .

#### 23.9.5.1.1.2 SLIMbus Global Initialization

##### 23.9.5.1.1.2.1 SLIMbus module software reset

This procedure performs software reset to the SLIMbus module.

**Table 23-671. SLIMbus component reset**

Step	Register / Bit Field / Programming Model	Value
Perform a software Reset	SLIMBUS_CMP_SYSCONFIG[0] SOFTRESET	0x1
Wait until Reset is finished?	SLIMBUS_CMP_SYSCONFIG[0] SOFTRESET	=0x0

##### 23.9.5.1.1.2.2 Power Management and Wakeup Configuration

This procedure configures the idle mode for SLIMbus component. By definition, target can handle read/write transaction as long as it is out of IDLE state.

**Table 23-672. SLIMbus idle mode configuration**

Step	Register / Bit Field / Programming Model	Value
Set slave idle mode	SLIMBUS_CMP_SYSCONFIG[3:2] IDLEMODE	0x-
Turn on/off internal clock autogating	SLIMBUS_CMP_SYSCONFIG[8] AUTOGATINGDISABLE	0x-

##### 23.9.5.1.1.2.3 SLIMbus FIFOs mapping and setup

This procedure maps each used FIFO to a free zone of the shared buffer prior to enabling. Also sets the FIFO thresholds used to trigger loading/unloading the FIFOs.



**Table 23-673. SLIMbus FIFOs mapping and setup**

Step	Register / Bit Field / Programming Model	Value
Set the capacity of data FIFO for RX channel j	SLIMBUS_DCR_FIFO_SETUP2_j[7:0] SIZE	0x--
Set the capacity of data FIFO for TX channel j	SLIMBUS_DCT_FIFO_SETUP2_j[7:0] SIZE	0x--
Select shared message channel RX FIFO size	SLIMBUS_SMR_FIFO_SETUP[6:5] SIZE_HI	0x-
Select shared message channel TX FIFO size	SLIMBUS_SMT_FIFO_SETUP[6:5] SIZE_HI	0x-
Set RX data channel agent j FIFO base address within shared RAM	SLIMBUS_DCR_FIFO_SETUP1_j[11:5] BASE_ADDR	0x--
Set TX data channel agent j FIFO base address within shared RAM	SLIMBUS_DCT_FIFO_SETUP1_j[11:5] BASE_ADDR	0x--
Set SLIMbus-side RX data channel agent j FIFO threshold, used for SLIMbus reception	SLIMBUS_DCR_FIFO_SETUP2_j[15:8] SB_THRESHOLD	0x--
Set SLIMbus-side TX data channel agent j FIFO threshold, used for SLIMbus transmission	SLIMBUS_DCT_FIFO_SETUP2_j[15:8] SB_THRESHOLD	0x--
Set software-side RX data channel agent's j FIFO threshold.	SLIMBUS_DCR_FIFO_SETUP2_j[23:16] DMA_THRESHOLD	0x--
Set software-side TX data channel agent's j FIFO threshold.	SLIMBUS_DCT_FIFO_SETUP2_j[23:16] DMA_THRESHOLD	0x--

### 23.9.5.1.2 Operational Modes Configuration

#### 23.9.5.1.2.1 SLIMbus transfer protocol configuration

This procedure configures the data channel(s) for a specific transfer protocol use. Values for each bit field vary depending on transfer protocol as shown in table [Table 23-675](#).

**Table 23-674. SLIMbus transfer protocol configuration sequence**

Step	Register / Bit Field / Programming Model	Value
Select asynchronous mode for RX data channel j	SLIMBUS_DCR_CONFIG1_j[3:0] TP	0x-
Select asynchronous mode for TX data channel j	SLIMBUS_DCT_CONFIG1_j[3:0] TP	0x-
Select ownership for RX data channel j	SLIMBUS_DCR_CONFIG1_j[4] TP_QUALIFIER	0x-
Select ownership for TX data channel j	SLIMBUS_DCT_CONFIG1_j[4] TP_QUALIFIER	0x-
Set segment length RX for data channel j	SLIMBUS_DCR_CONFIG1_j[9:5] SL	0x--
Set segment length TX for data channel j	SLIMBUS_DCT_CONFIG1_j[9:5] SL	0x--
Select number of AUX slots in segment for RX data channel j	SLIMBUS_DCR_CONFIG1_j[19:16] AF	0x--
Select number of AUX slots in segment for TX data channel j	SLIMBUS_DCT_CONFIG1_j[19:16] AF	0x--
Set data length (data slots in segment) for RX data channel j	SLIMBUS_DCR_CONFIG1_j[14:10] DL	0x--
Set data length (data slots in segment) for TX data channel j	SLIMBUS_DCT_CONFIG1_j[14:10] DL	0x--
Enable/Disable RX data channel n pairing with RX channel j-1	SLIMBUS_DCR_CONFIG1_j[15] CL	0x-
Enable/Disable TX data channel n pairing with TX channel j-1	SLIMBUS_DCT_CONFIG1_j[15] CL	0x-
Enable RX data channel j	SLIMBUS_DCR_CONFIG1_j[31] ENABLE	0x1
Enable TX data channel j	SLIMBUS_DCT_CONFIG1_j[31] ENABLE	0x1
Set segment periodicity and offset for RX data channel j	SLIMBUS_DCR_CONFIG2_j[11:0] SD	0x---
Set segment periodicity and offset for TX data channel j	SLIMBUS_DCT_CONFIG2_j[11:0] SD	0x---

**Table 23-675. SLIMbus transfer protocol configuration**

Register / Bit Field	Isochronous TP	Pushed TP	Pulled TP	"Plain" Asynchronous simplex TP	"Plain" Asynchronous 1/2 duplex TP	Extended Asynchronous simplex TP	Extended Asynchronous 1/2 duplex TP
SLIMBUS_DCR_CONFIG1_j[3:0] TP SLIMBUS_DCT_CONFIG1_j[3:0] TP	0x0	0x1	0x2	0x4	0x5	0x6	0x7
SLIMBUS_DCR_CONFIG1_j[4] TP_QUALIFIER SLIMBUS_DCT_CONFIG1_j[4] TP_QUALIFIER	unused	unused	unused	0x0, 0x1	0x0, 0x1	0x0, 0x1	0x0, 0x1
SLIMBUS_DCR_CONFIG1_j[9:5] SL SLIMBUS_DCT_CONFIG1_j[9:5] SL	0x2...0x8	0x3...0x9	0x3...0x9	0x3...0x9	0x3...0x9	0x4...0x1F	0x4...0x1F
SLIMBUS_DCR_CONFIG1_j[19:16] AF SLIMBUS_DCT_CONFIG1_j[19:16] AF	0x0...0x2	0x0...0x2	0x0...0x2	0x0...0x2	0x0...0x2	0x0	0x0
SLIMBUS_DCR_CONFIG1_j[14:10] DL SLIMBUS_DCT_CONFIG1_j[14:10] DL	0x2...0x8	0x2...0x8	0x2...0x8	0x2...0x8	0x2...0x8	0x2	0x2
SLIMBUS_DCR_CONFIG1_j[15] CL SLIMBUS_DCT_CONFIG1_j[15] CL	0x0, 0x1	0x0, 0x1	0x0, 0x1	0x0	0x0	0x0	0x0
SLIMBUS_DCR_CONFIG2_j[11:0] SD SLIMBUS_DCT_CONFIG2_j[11:0] SD	0x---	0x---	0x---	0x---	0x---	0x---	0x---

More about SLIMbus protocol configuration see in [Section 23.9.4.8 SLIMbus Transfer Protocols](#).

### 23.9.5.1.2.2 SLIMbus interrupts configuration

This procedure selects the required interrupt sources for generating interrupts.

**Table 23-676. SLIMbus interrupts enabling**

Step	Register / Bit Field / Programming Model	Value
Enable required interrupt sources (for SLIMbus1)	SLIMBUS_CMP_IRQENABLE_SET[19:0]	0x----
Enable required interrupt sources (for SLIMbus2)	SLIMBUS_CMP_IRQENABLE_SET[11:0]	0x---
Disable the rest of interrupt sources (for SLIMbus1)	SLIMBUS_CMP_IRQENABLE_CLR[19:0]	0x----
Disable the rest of interrupt sources (for SLIMbus2)	SLIMBUS_CMP_IRQENABLE_CLR[11:0]	0x---
Clear the interrupt status register (for SLIMbus1)	SLIMBUS_CMP_IRQSTATUS[19:0]	0xFFFFF
Clear the interrupt status register (for SLIMbus2)	SLIMBUS_CMP_IRQSTATUS[11:0]	0xFFF

### 23.9.5.1.2.3 SLIMbus DMA configuration

This procedure selects and configures the channels for DMA mode use.

**Table 23-677. SLIMbus DMA configuration**

Step	Register / Bit Field / Programming Model	Value
Select DMA request size minus 1, in FIFO words for RX data channel j	SLIMBUS_DCR_FIFO_SETUP1_j[4:1] DMA_REQ_SIZE	0x-
Select DMA request size minus 1, in FIFO words for TX data channel j	SLIMBUS_DCT_FIFO_SETUP1_j[4:1] DMA_REQ_SIZE	0x-
Enable DMA mode for required data channels (for SLIMbus1) <sup>(1)</sup>	SLIMBUS_CMP_DMAENABLE_SET [19:4]	0x----
Enable DMA mode for required data channels (for SLIMbus2)	SLIMBUS_CMP_DMAENABLE_SET [11:4]	0x--
Disable DMA mode for the rest of channels (for SLIMbus1) <sup>(1)</sup>	SLIMBUS_CMP_DMAENABLE_CLR [19:4]	0x----
Disable DMA mode for the rest of channels (for SLIMbus2)	SLIMBUS_CMP_DMAENABLE_CLR [11:4]	0x--

<sup>(1)</sup> Component level registers (e.g. with CMP in name) composition (bit fields used) depends on the SLIMbus component instantiation - SLIMbus1 or SLIMbus2.

### 23.9.5.1.2.4 SLIMbus Component Boot

Component boot sequence is initiated when the component decides to join the bus. More about SLIMbus boot and reset procedures see in [Section 23.9.4.4.3](#)

#### 23.9.5.1.2.4.1 Clock Receiver Component Boot

This procedure boots the SLIMbus configuration when the component is in clock receiver mode.

**Table 23-678. Clock Receiver Component Boot**

Step	Register / Bit Field / Programming Model	Value
Set clock receiver component boot flag	SLIMBUS_FL_CONTROL[0] BOOT	0x1
Initiate the sequence	SLIMBUS_FL_CONTROL[2] FORCE_RECONFIGURE	0x1

#### 23.9.5.1.2.4.2 Clock Source Component Boot

This procedure boots the SLIMbus when the component is in clock source mode (includes the active framer).

**Table 23-679. Clock Source Component Boot**

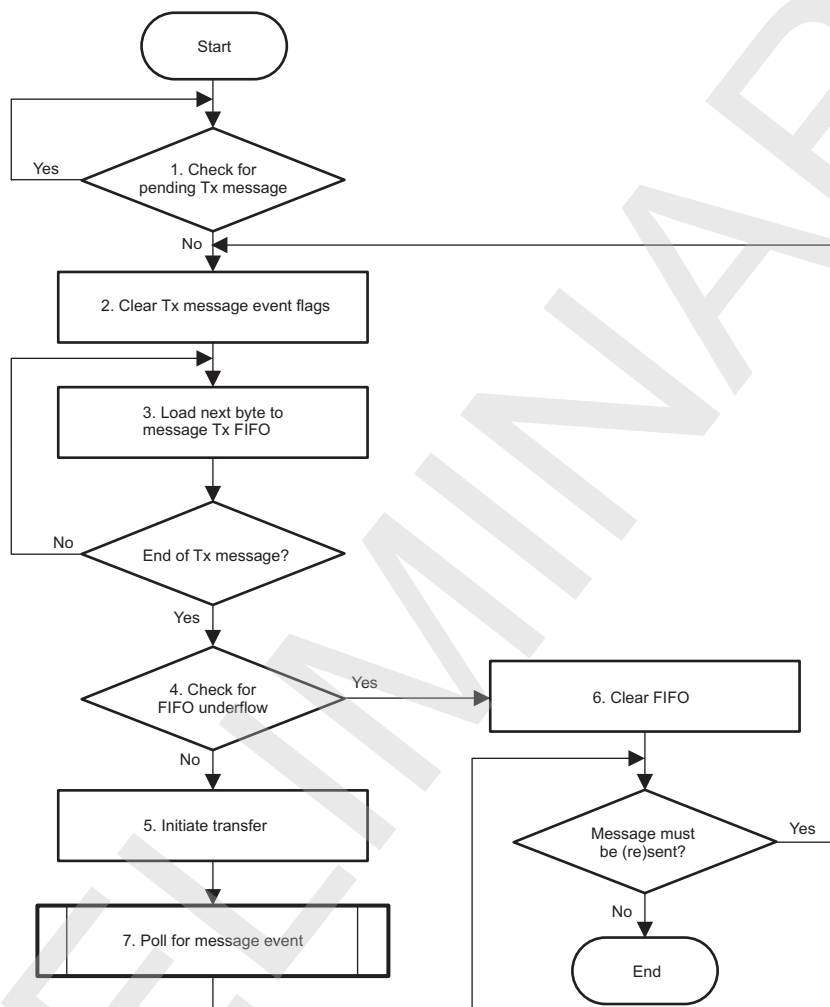
Step	Register / Bit Field / Programming Model	Value
Set clock source component boot flag	SLIMBUS_FR_CONTROL[0] BOOT	0x1
Select control space width and subframe length	SLIMBUS_FL_SM[4:0] SM	0x--
Select Root / SLIMbus frequency ratio (Clock Gear)	SLIMBUS_FL_CG[3:0] CG	0x-
Set the Root frequency	SLIMBUS_FL_RF[3:0] RF	0x-
Select input clock source	SLIMBUS_FR_CLOCK_SOURCE[5:3] CLKSEL	0x-
Select Root divider ratio to obtain the Root frequency	SLIMBUS_FR_CLOCK_SOURCE[2:0] CLKDIV	0x-
Initiate the sequence	SLIMBUS_FL_CONTROL[2] FORCE_RECONFIGURE	0x1

### 23.9.5.1.3 SLIMbus Transfer Procedures

#### 23.9.5.1.3.1 Transmitting Message, Polling Mode

This procedure transmits a message over the SLIMbus without using interrupts.

**Figure 23-216. Transmitting Message, Polling Mode Procedure**



slimbus-012

**Table 23-680. Register Settings for the SLIMbus Message Transmit Procedure, Polling Method**

Procedure Name	Description	Register settings
1. Check for pending TX message	This procedure checks whether ongoing TX message is in progress. This also implies that TX message FIFO can receive another message. <code>SLIMBUS_SMT_CONTROL[0] MESSAGE_ENABLE = 0</code> means there is no ongoing TX message in FIFO.	Check <code>SLIMBUS_SMT_CONTROL[0] MESSAGE_ENABLE = 0</code> ?
2. Clear TX message event flags	This procedure clears TX message event flags by writing 1 to the corresponding bit(s). Failing to meet this condition will not prevent the message transmission, but will prevent a proper readout of the message transmission's outcome.	Write <code>SLIMBUS_SMT_INFO[5:0] = 0x3F</code>

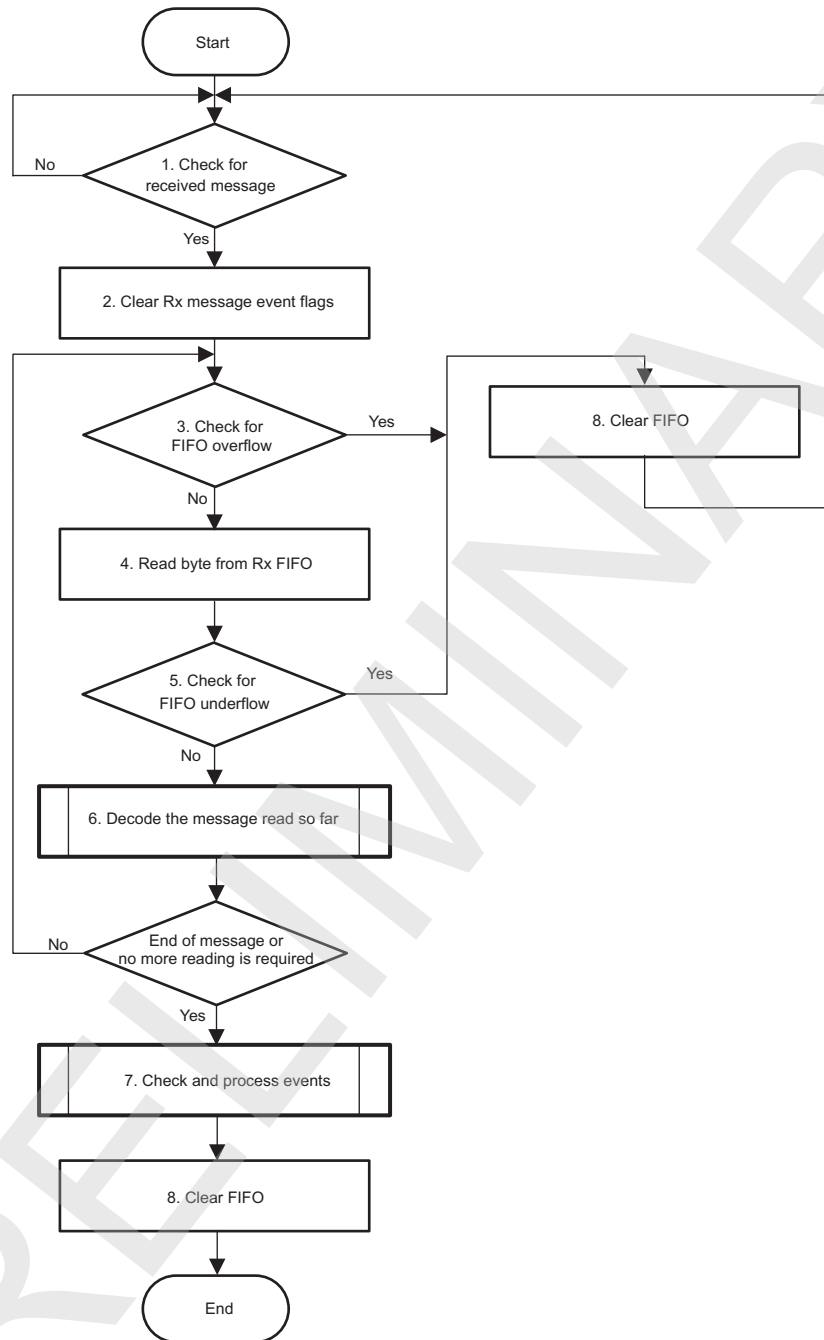
**Table 23-680. Register Settings for the SLIMbus Message Transmit Procedure, Polling Method  
(continued)**

Procedure Name	Description	Register settings
3. Load byte to TX FIFO	The message is written byte by byte to register <a href="#">SLIMBUS_SMT_MESSAGE</a> [7:0]. Serial message starts with the MSB (bit 7) of the first byte written into the FIFO and ends with the LSB (bit 0) of the last written byte.	Write <a href="#">SLIMBUS_SMT_MESSAGE</a> [7:0] = 0x--
4. Check for FIFO underflow	This procedure checks if an underflow event is flagged (stored message is shorter than encoded in its own AT, RL fields).	Check <a href="#">SLIMBUS_SMT_INFO</a> [6] UNDERFLOW = 1?
5. Initiate transfer	This procedure initiates the transmit sequence. From that point on, the device will attempt to transmit the message stored in the FIFO.	Write <a href="#">SLIMBUS_SMT_CONTROL</a> [0] MESSAGE_ENABLE = 1
6. Clear FIFO	This procedure clears the FIFO in case of error.	Write <a href="#">SLIMBUS_SMT_CONTROL</a> [0] MESSAGE_ENABLE = 0
7. Poll message events	Transmission is considered complete if one of the following -mutually exclusive- conditions is met, and the corresponding bit of <a href="#">SLIMBUS_SMT_INFO</a> is set: Negative response: <a href="#">SLIMBUS_SMT_INFO</a> [3] NACK = 1, or Positive response: <a href="#">SLIMBUS_SMT_INFO</a> [2] PACK = 1, or No response: <a href="#">SLIMBUS_SMT_INFO</a> [4] NORE = 1, or Illegal response: <a href="#">SLIMBUS_SMT_INFO</a> [5] UDEF = 1, or Transmit collision: <a href="#">SLIMBUS_SMT_INFO</a> [0] MC_TX_COL = 1	<ol style="list-style-type: none"> <li>1. Poll <a href="#">SLIMBUS_SMT_INFO</a>[7:0]</li> <li>2. Process the event as required</li> <li>3. Clear the corresponding info bit</li> </ol>

**NOTE:** For more information about SMT events see [Section 23.9.4.12.5 Shared Message Channel Transmit ISRs](#).  
For information about all SLIMbus events see [Section 23.9.4.12.2 Possible Interrupts](#).

### 23.9.5.1.3.2 Receiving Message, Polling Mode

This procedure reads any received in FIFO message without using interrupts.

**Figure 23-217. Receiving Message, Polling Mode Procedure**

slimbus-013

**Table 23-681. Register Settings for the SLIMbus message receive procedure, Polling Method**

Procedure Name	Description	Register settings
1. Check for received message	This procedure checks whether a message is received in FIFO.	Check <code>SLIMBUS_SMR_INFO[2] RECEIVED_MESSAGE = 1?</code>
2. Clear RX message event flags	This procedure clears pending (at 1) RX message event flags by writing 1 to the corresponding bit(s).	Write <code>SLIMBUS_SMR_INFO[4:0] = 0x1F</code>

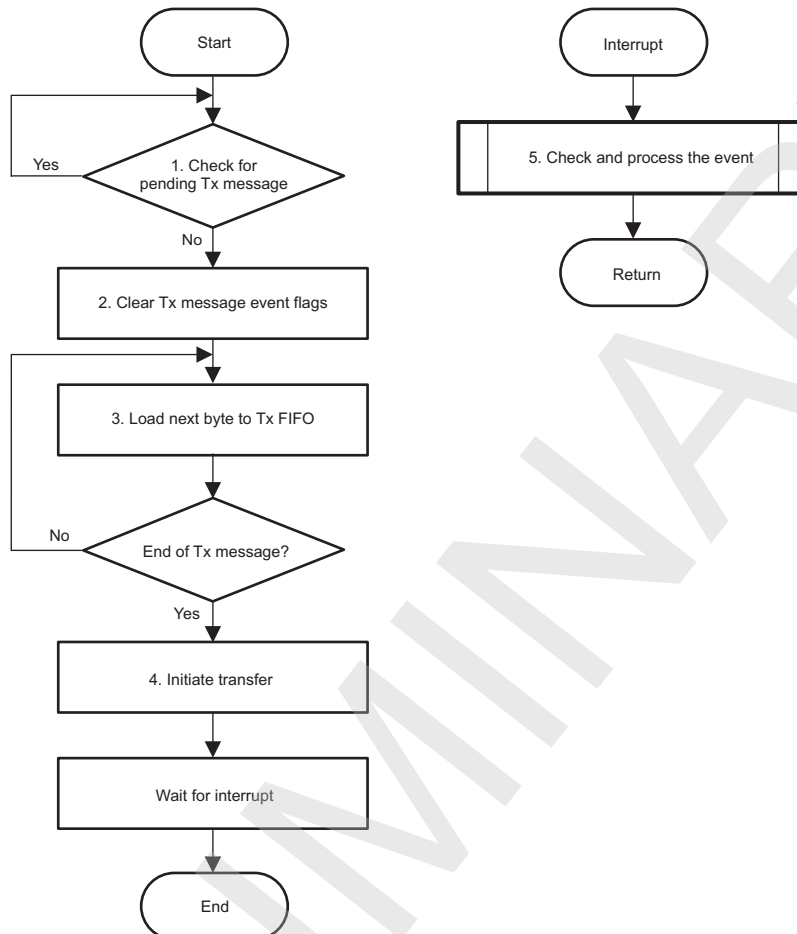
**Table 23-681. Register Settings for the SLIMbus message receive procedure, Polling Method (continued)**

Procedure Name	Description	Register settings
3. Check for FIFO overflow	This procedure checks whether overflow event is flagged ( too many bytes are received, which will overwrite previous bytes.)	Check <code>SLIMBUS_SMR_INFO[1] OVERFLOW = 1?</code>
4. Read byte from RX FIFO	The message is read byte by byte off the FIFO and put into register <code>SLIMBUS_SMR_MESSAGE[7:0]</code> .	Read <code>SLIMBUS_SMR_MESSAGE[7:0]</code>
5. Check for FIFO underflow	This procedure checks if an underflow event is flagged (too many bytes are read out ).	Check <code>SLIMBUS_SMR_INFO[1] UNDERFLOW = 1?</code>
6. Decode the read so far message	This procedure takes place in parallel with the unloading phase. In some cases, a partial decoding can lead the software to skip the end of the unloading and flush the FIFO.	Decoding the message is entirely done by software
7. Check and process events	This procedure checks and processes any pending events.	<ol style="list-style-type: none"> <li>1. Poll <code>SLIMBUS_SMR_INFO[4:0]</code></li> <li>2. Process events as required</li> <li>3. Clear the corresponding info bit</li> </ol>
8. Clear FIFO	This procedure clears the FIFO in order to be ready to receive a new message, or in case of error	Write <code>SLIMBUS_SMR_INFO[2] RECEIVED_MESSAGE = 1</code>

**NOTE:** For more information about SMR events see [Section 23.9.4.12.6 Shared Message Channel Receive ISRs](#).  
For information about all SLIMbus events see [Section 23.9.4.12.2 Possible Interrupts](#).

### 23.9.5.1.3.3 Transmitting message, interrupt mode

This procedure transmits a message over the SLIMbus using interrupts for event handling. Interrupts must be enabled as shown in [Section 23.9.5.1.2.2 SLIMbus interrupts configuration](#).

**Figure 23-218. Transmitting message, interrupt mode procedure**

slimbus-014

**Table 23-682. Register Settings for the SLIMbus message transmit procedure, Interrupt Method**

Procedure Name	Description	Register settings
<b>Main routine</b>		
1. Check for pending TX message	This procedure checks whether ongoing TX message is in progress. This also implies that TX message FIFO can receive another message. <b>SLIMBUS_SMT_CONTROL[0] MESSAGE_ENABLE = 0</b> means there is no ongoing TX message in FIFO.	Check <b>SLIMBUS_SMT_CONTROL[0] MESSAGE_ENABLE = 0?</b>
2. Clear TX message event flags	This procedure clears TX message event flags by writing 1 to the corresponding bit(s). Failing to meet this condition will not prevent the message transmission, but will prevent a proper readout of the message transmission's outcome	Write <b>SLIMBUS_SMT_INFO[5:0] = 0x3F</b>
3. Load byte to TX FIFO	The message is written byte by byte to register <b>SLIMBUS_SMT_MESSAGE[7:0]</b> . Serial message starts with the MSB (bit 7) of the first byte written into the FIFO and ends with the LSB (bit 0) of the last written byte.	Write <b>SLIMBUS_SMT_MESSAGE[7:0] = 0x--</b>
4. Initiate transfer	This procedure initiates the transmit sequence. From that point on, the device will attempt to transmit the message stored in the FIFO.	Write <b>SLIMBUS_SMT_CONTROL[0] MESSAGE_ENABLE = 1</b>
<b>Interrupt subroutine</b>		



**Table 23-682. Register Settings for the SLIMbus message transmit procedure, Interrupt Method (continued)**

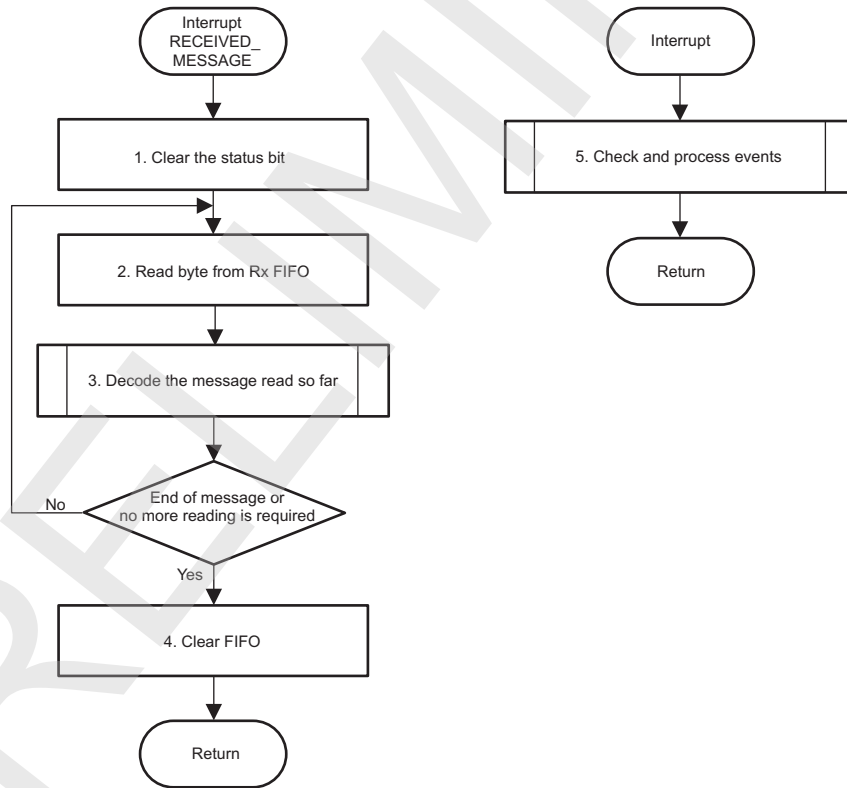
Procedure Name	Description	Register settings
5. Check and process events	<ol style="list-style-type: none"> <li>1. Clear the corresponding status bit</li> <li>2. Check the pending event by polling the related information register's flags.</li> <li>3. Process the event as required. For more information see <a href="#">Section 23.9.4.12.5</a> SLIMbus Shared Message Channel Transmit ISRs</li> <li>4. When needed and allowed, clear the information register</li> </ol>	<ol style="list-style-type: none"> <li>1. Write <code>SLIMBUS_CMP_IRQSTATUS[2]</code> <code>SMT_INFO = 1</code></li> <li>2. Poll <code>SLIMBUS_SMT_INFO[7:0]</code></li> <li>3. Process the event as required</li> <li>4. Write <code>SLIMBUS_SMT_INFO[7:0] = 0xFF</code></li> </ol>

**NOTE:** For information about all SLIMbus events see [Section 23.9.4.12.2](#) Possible interrupts.

**23.9.5.1.3.4 Receiving message, interrupt mode**

This procedure reads the received in FIFO message and processes any pending events using interrupts. Interrupts must be enabled as shown in [Section 23.9.5.1.2.2](#) SLIMbus interrupts configuration.

**Figure 23-219. Receiving message, interrupt mode procedure**



slimbus-015

**Table 23-683. Register Settings for the SLIMbus message receive procedure, Transmit Method**

Procedure Name	Description	Register settings
Main interrupt routine		
1. Clear the status bit	This procedure clears the interrupt status bit, if it is not already cleared.	Write <code>SLIMBUS_CMP_IRQSTATUS[3]</code> <code>SMR_INFO = 1</code>

**Table 23-683. Register Settings for the SLIMbus message receive procedure, Transmit Method (continued)**

Procedure Name	Description	Register settings
2. Read byte from RX FIFO	The message is read byte by byte off the FIFO and put into register <code>SLIMBUS_SMR_MESSAGE[7:0]</code> .	Read <code>SLIMBUS_SMR_MESSAGE[7:0]</code>
3. Decode the read so far message	This procedure takes place in parallel with the unloading phase. In some cases, a partial decoding can lead the software to skip the end of the unloading and flush the FIFO.	Decoding of the message is entirely done by software
4. Clear FIFO	This procedure clears the FIFO in order to be ready to receive a new message, or in case of error.	Write <code>SLIMBUS_SMR_INFO[2] RECEIVED_MESSAGE = 1</code>
<b>Interrupt subroutine</b>		
5. Check and process events	<ol style="list-style-type: none"> <li>1. Clear the corresponding status bit</li> <li>2. Check the pending event by polling the related information register's flags.</li> <li>3. Process the event as required. For more information see <a href="#">Section 23.9.4.12.6</a> SLIMbus Shared Message Channel Receive ISRs</li> <li>4. When needed and allowed, clear the information register</li> </ol>	<ol style="list-style-type: none"> <li>1. Write <code>SLIMBUS_CMP_IRQSTATUS[3] SMR_INFO = 1</code></li> <li>2. Poll <code>SLIMBUS_SMR_INFO[4:0]</code></li> <li>3. Process the event as required.</li> <li>4. Write <code>SLIMBUS_SMR_INFO[4:0] = 0x1F</code></li> </ol>

**NOTE:** For information about all SLIMbus events see [Section 23.9.4.12.2](#) Possible interrupts.

### 23.9.5.1.3.5 Receiving data

This procedure reads data word received in FIFO from a SLIMbus device.

Interrupts must be enabled in order to generate interrupt, when data is ready for reading from FIFO. More about enabling interrupts see in [Section 23.9.5.1.2.2](#) SLIMbus interrupts configuration.

Data receive channel(s) must be configured to generate DMA requests, when DMA mode is used for data transfer. See [Section 23.9.5.1.2.3](#) SLIMbus DMA configuration.

**Table 23-684. Data word receiving**

Step	Register / Bit Field / Programming Model	Value
Check FIFO level	<code>SLIMBUS_DCR_INFO_j[3] HILEVEL</code>	=0x1
Read data word from FIFO	<code>SLIMBUS_DCR_DATA_j</code>	See <sup>(1)</sup>

<sup>(1)</sup> FIFO data word size varies between 1 and 8 slots (4 and 32bits). See [Table 23-655](#), Examples of AUX + DATA mapping in 32-bit read/write accesses.

### 23.9.5.1.3.6 Transmitting data

This procedure loads data word to FIFO for transmitting to a SLIMbus device.

Interrupts must be enabled in order to generate interrupt, when FIFO is ready for more data loading. More about enabling interrupts see in [Section 23.9.5.1.2.2](#) SLIMbus interrupts configuration.

Data transmit channel(s) must be configured to generate DMA requests, when DMA mode is used for data transfer. See [Section 23.9.5.1.2.3](#) SLIMbus DMA configuration.

**Table 23-685. Data word transmitting**

Step	Register / Bit Field / Programming Model	Value
Check FIFO level	<code>SLIMBUS_DCT_INFO_j[3] LOLEVEL</code>	=0x1
Write data word to FIFO	<code>SLIMBUS_DCT_DATA_j</code>	See <sup>(1)</sup>

<sup>(1)</sup> FIFO data word size varies between 1 and 8 slots (4 and 32bits). See [Table 23-655](#), Examples of AUX + DATA mapping in 32-bit read/write accesses.

#### **23.9.5.1.4 Event handling**

For Information about all SLIMbus events handling refer to [Section 23.9.4.12.1 Interrupt Requests](#).

## 23.9.6 SLIMbus Register Manual

### 23.9.6.1 SLIMbus Register Overview

The registers described in this section are prefixed with one of the following:

- **CMP:** Component- (that is, module) level control and status.
- **SMT:** Shared Messages Transmit: All messages transmitted to the serial SLIMbus interface. This resource is shared by all devices within the component (all devices can transmit messages).
- **SMR:** Shared Messages Receive: All messages received from the serial SLIMbus interface. This resource is shared by all devices within the component (all devices can receive messages).
- **FL:** Frame Layer status and control. The frame layer is the hardware-implemented protocol layer located just above the physical layer, and shared by all devices of a component.
- **FR:** Framer device-specific status and control. The framer device is one of the component's devices (logical entities of the SLIMbus); exactly one framer device may optionally be implemented in each SLIMbus component. Note that the framer device is not always active (in which case the related registers aren't used either), and that some of the framer's control and status is shared with the (always-present, always-active) interface device.
- **DEV:** Device control and status. The same register structure is repeated as many times as there are devices supported by the component (that is, the module), including the (compulsory) interface device, the framer device (optional), the manager device (optional), and the generic devices (optional, one or several instances).
- **DCT:** Data Channel Transmit: The same register structure is repeated as many times as there are transmit data channels supported, that is, FIFOs and associated logic. Devices (typically generic devices) are associated to a data channel for the transmissions of data towards the SLIMbus serial interface.
- **DCR:** Data Channel Receive: The same register structure is repeated as many times as there are receive data channels supported, that is, FIFOs and associated logic. Devices (typically generic devices) are associated to a data channel for the reception of data from the SLIMbus serial interface.

### 23.9.6.2 SLIMbus Instance Summary

Table 23-686 shows the base address and address space for the SLIMbus module instances.

**Table 23-686. SLIMbus Instance Summary**

Module Name	Base Address L3 Interconnect	Base Address Cortex-A9 Private Access	Base Address DSP Private Access	Size
SLIMBUS1	0x4902 C000	0x4012 C000	0x2 C000	4 KB
SLIMBUS2	0x4807 6000	-	-	4 KB

**NOTE:** Private Access is an access which is not using the L3/L4 interconnects.

### 23.9.6.3 SLIMbus Registers

#### 23.9.6.3.1 SLIMBUS1 Register Summary

**Table 23-687. SLIMBUS1 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
<a href="#">SLIMBUS_CMP_REVISION</a>	R	32	0x0000 0000	0x4902 C000	0x4012 C000	0x2 C000
<a href="#">SLIMBUS_CMP_HWINFO</a>	R	32	0x0000 0004	0x4902 C004	0x4012 C004	0x2 C004
<a href="#">SLIMBUS_CMP_SYSCONFIG</a>	RW	32	0x0000 0010	0x4902 C010	0x4012 C010	0x2 C010
RESERVED	RW	32	0x0000 0020	0x4902 C020	0x4012 C020	0x2 C020

**Table 23-687. SLIMBUS1 Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
SLIMBUS_CMP_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4902 C024	0x4012 C024	0x2 C024
SLIMBUS_CMP_IRQSTATUS	RW	32	0x0000 0028	0x4902 C028	0x4012 C028	0x2 C028
SLIMBUS_CMP_IRQENABLE_SET	RW	32	0x0000 002C	0x4902 C02C	0x4012 C02C	0x2 C02C
SLIMBUS_CMP_IRQENABLE_CLR	RW	32	0x0000 0030	0x4902 C030	0x4012 C030	0x2 C030
SLIMBUS_CMP_DMAENABLE_SET	RW	32	0x0000 0034	0x4902 C034	0x4012 C034	0x2 C034
SLIMBUS_CMP_DMAENABLE_CLR	RW	32	0x0000 0038	0x4902 C038	0x4012 C038	0x2 C038
SLIMBUS_CMP_IV	RW	32	0x0000 0040	0x4902 C040	0x4012 C040	0x2 C040
SLIMBUS_CMP_MI_PC	RW	32	0x0000 0044	0x4902 C044	0x4012 C044	0x2 C044
SLIMBUS_SMT_INFO	RW	32	0x0000 0050	0x4902 C050	0x4012 C050	0x2 C050
SLIMBUS_SMT_MESSAGE	RW	32	0x0000 0054	0x4902 C054	0x4012 C054	0x2 C054
SLIMBUS_SMT_CONTROL	RW	32	0x0000 0058	0x4902 C058	0x4012 C058	0x2 C058
SLIMBUS_SMT_FIFO_SETUP	RW	32	0x0000 005C	0x4902 C05C	0x4012 C05C	0x2 C05C
SLIMBUS_SMR_INFO	RW	32	0x0000 0060	0x4902 C060	0x4012 C060	0x2 C060
SLIMBUS_SMR_MESSAGE	R	32	0x0000 0064	0x4902 C064	0x4012 C064	0x2 C064
SLIMBUS_SMR_CONTROL	RW	32	0x0000 0068	0x4902 C068	0x4012 C068	0x2 C068
SLIMBUS_SMR_FIFO_SETUP	RW	32	0x0000 006C	0x4902 C06C	0x4012 C06C	0x2 C06C
SLIMBUS_FL_INFO	RW	32	0x0000 0070	0x4902 C070	0x4012 C070	0x2 C070
SLIMBUS_FL_CONTROL	RW	32	0x0000 0074	0x4902 C074	0x4012 C074	0x2 C074
SLIMBUS_FL_SM	RW	32	0x0000 0078	0x4902 C078	0x4012 C078	0x2 C078
SLIMBUS_FL_CG	RW	32	0x0000 007C	0x4902 C07C	0x4012 C07C	0x2 C07C
SLIMBUS_FL_RF	RW	32	0x0000 0080	0x4902 C080	0x4012 C080	0x2 C080
SLIMBUS_FR_INFO	RW	32	0x0000 0090	0x4902 C090	0x4012 C090	0x2 C090
SLIMBUS_FR_CLOCK_SOURCE	RW	32	0x0000 0094	0x4902 C094	0x4012 C094	0x2 C094
SLIMBUS_FR_CONTROL	RW	32	0x0000 0098	0x4902 C098	0x4012 C098	0x2 C098
SLIMBUS_FR_FRAMER_HANDOVER	RW	32	0x0000 009C	0x4902 C09C	0x4012 C09C	0x2 C09C
SLIMBUS_FR_CLOCK_PAUSE	RW	32	0x0000 00A0	0x4902 C0A0	0x4012 C0A0	0x2 C0A0
SLIMBUS_DEV_LA <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0100 + (0x10 * i)	0x4902 C100 + (0x10 * i)	0x4012 C100 + (0x10 * i)	0x2 C100 + (0x10 * i)
SLIMBUS_DEV_EA_LO <sub>i</sub> <sup>(1)</sup>	R	32	0x0000 0104 + (0x10 * i)	0x4902 C104 + (0x10 * i)	0x4012 C104 + (0x10 * i)	0x2 C104 + (0x10 * i)
SLIMBUS_DEV_EA_HI <sub>i</sub> <sup>(1)</sup>	R	32	0x0000 0108 + (0x10 * i)	0x4902 C108 + (0x10 * i)	0x4012 C108 + (0x10 * i)	0x2 C108 + (0x10 * i)
SLIMBUS_DCT_INFO <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0200 + (0x20 * j)	0x4902 C200 + (0x20 * j)	0x4012 C200 + (0x20 * j)	0x2 C200 + (0x20 * j)
SLIMBUS_DCT_FIFO_SETUP1 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0204 + (0x20 * j)	0x4902 C204 + (0x20 * j)	0x4012 C204 + (0x20 * j)	0x2 C204 + (0x20 * j)
SLIMBUS_DCT_FIFO_SETUP2 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0208 + (0x20 * j)	0x4902 C208 + (0x20 * j)	0x4012 C208 + (0x20 * j)	0x2 C208 + (0x20 * j)

<sup>(1)</sup> i = 0 to 6

<sup>(2)</sup> j = 0 to 7

**Table 23-687. SLIMBUS1 Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect	Physical Address Cortex-A9 Private Access	Physical Address DSP Private Access
SLIMBUS_DCT_FIFO_STATU S <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 020C + (0x20 * j)	0x4902 C20C + (0x20 * j)	0x4012 C20C + (0x20 * j)	0x2 C20C + (0x20 * j)
SLIMBUS_DCT_MAP <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0210 + (0x20 * j)	0x4902 C210 + (0x20 * j)	0x4012 C210 + (0x20 * j)	0x2 C210 + (0x20 * j)
SLIMBUS_DCT_CONFIG1 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0214 + (0x20 * j)	0x4902 C214 + (0x20 * j)	0x4012 C214 + (0x20 * j)	0x2 C214 + (0x20 * j)
SLIMBUS_DCT_CONFIG2 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0218 + (0x20 * j)	0x4902 C218 + (0x20 * j)	0x4012 C218 + (0x20 * j)	0x2 C218 + (0x20 * j)
SLIMBUS_DCT_DATA <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 021C + (0x20 * j)	0x4902 C21C + (0x20 * j)	0x4012 C21C + (0x20 * j)	0x2 C21C + (0x20 * j)
SLIMBUS_DCR_INFO <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0300 + (0x20 * j)	0x4902 C300 + (0x20 * j)	0x4012 C300 + (0x20 * j)	0x2 C300 + (0x20 * j)
SLIMBUS_DCR_FIFO_SETU P1 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0304 + (0x20 * j)	0x4902 C304 + (0x20 * j)	0x4012 C304 + (0x20 * j)	0x2 C304 + (0x20 * j)
SLIMBUS_DCR_FIFO_SETU P2 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0308 + (0x20 * j)	0x4902 C308 + (0x20 * j)	0x4012 C308 + (0x20 * j)	0x2 C308 + (0x20 * j)
SLIMBUS_DCR_FIFO_STAT US <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 030C + (0x20 * j)	0x4902 C30C + (0x20 * j)	0x4012 C30C + (0x20 * j)	0x2 C30C + (0x20 * j)
SLIMBUS_DCR_MAP <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0310 + (0x20 * j)	0x4902 C310 + (0x20 * j)	0x4012 C310 + (0x20 * j)	0x2 C310 + (0x20 * j)
SLIMBUS_DCR_CONFIG1 <sub>j</sub> <sup>(3)</sup>	RW	32	0x0000 0314 + (0x20 * j)	0x4902 C314 + (0x20 * j)	0x4012 C314 + (0x20 * j)	0x2 C314 + (0x20 * j)
SLIMBUS_DCR_CONFIG2 <sub>j</sub> <sup>(3)</sup>	RW	32	0x0000 0318 + (0x20 * j)	0x4902 C318 + (0x20 * j)	0x4012 C318 + (0x20 * j)	0x2 C318 + (0x20 * j)
SLIMBUS_DCR_DATA <sub>j</sub> <sup>(3)</sup>	R	32	0x0000 031C + (0x20 * j)	0x4902 C31C + (0x20 * j)	0x4012 C31C + (0x20 * j)	0x2 C31C + (0x20 * j)

<sup>(3)</sup> j = 0 to 7**23.9.6.3.2 SLIMBUS2 Register Summary****Table 23-688. SLIMBUS2 Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect
SLIMBUS_CMP_REVISION	R	32	0x0000 0000	0x4807 6000
SLIMBUS_CMP_HWINFO	R	32	0x0000 0004	0x4807 6004
SLIMBUS_CMP_SYSCONFIG	RW	32	0x0000 0010	0x4807 6010
RESERVED	RW	32	0x0000 0020	0x4807 6020
SLIMBUS_CMP_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4807 6024
SLIMBUS_CMP_IRQSTATUS	RW	32	0x0000 0028	0x4807 6028
SLIMBUS_CMP_IRQENABLE_SET	RW	32	0x0000 002C	0x4807 602C
SLIMBUS_CMP_IRQENABLE_CLR	RW	32	0x0000 0030	0x4807 6030
SLIMBUS_CMP_DMAENABLE_SET	RW	32	0x0000 0034	0x4807 6034
SLIMBUS_CMP_DMAENABLE_CLR	RW	32	0x0000 0038	0x4807 6038
SLIMBUS_CMP_IV	RW	32	0x0000 0040	0x4807 6040
SLIMBUS_CMP_MI_PC	RW	32	0x0000 0044	0x4807 6044
SLIMBUS_SMT_INFO	RW	32	0x0000 0050	0x4807 6050
SLIMBUS_SMT_MESSAGE	RW	32	0x0000 0054	0x4807 6054
SLIMBUS_SMT_CONTROL	RW	32	0x0000 0058	0x4807 6058
SLIMBUS_SMT_FIFO_SETUP	RW	32	0x0000 005C	0x4807 605C

**Table 23-688. SLIMBUS2 Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address L3 Interconnect
SLIMBUS_SMR_INFO	RW	32	0x0000 0060	0x4807 6060
SLIMBUS_SMR_MESSAGE	R	32	0x0000 0064	0x4807 6064
SLIMBUS_SMR_CONTROL	RW	32	0x0000 0068	0x4807 6068
SLIMBUS_SMR_FIFO_SETUP	RW	32	0x0000 006C	0x4807 606C
SLIMBUS_FL_INFO	RW	32	0x0000 0070	0x4807 6070
SLIMBUS_FL_CONTROL	RW	32	0x0000 0074	0x4807 6074
SLIMBUS_FL_SM	RW	32	0x0000 0078	0x4807 6078
SLIMBUS_FL_CG	RW	32	0x0000 007C	0x4807 607C
SLIMBUS_FL_RF	RW	32	0x0000 0080	0x4807 6080
SLIMBUS_FR_INFO	RW	32	0x0000 0090	0x4807 6090
SLIMBUS_FR_CLOCK_SOURCE	RW	32	0x0000 0094	0x4807 6094
SLIMBUS_FR_CONTROL	RW	32	0x0000 0098	0x4807 6098
SLIMBUS_FR_FRAMER_HANDOVER	RW	32	0x0000 009C	0x4807 609C
SLIMBUS_FR_CLOCK_PAUSE	RW	32	0x0000 00A0	0x4807 60A0
SLIMBUS_DEV_LA <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0100 + (0x10 * i)	0x4807 6100 + (0x10 * i)
SLIMBUS_DEV_EA_LO <sub>j</sub> <sup>(1)</sup>	R	32	0x0000 0104 + (0x10 * i)	0x4807 6104 + (0x10 * i)
SLIMBUS_DEV_EA_HI <sub>j</sub> <sup>(1)</sup>	R	32	0x0000 0108 + (0x10 * i)	0x4807 6108 + (0x10 * i)
SLIMBUS_DCT_INFO <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0200 + (0x20 * j)	0x4807 6200 + (0x20 * j)
SLIMBUS_DCT_FIFO_SETUP1 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0204 + (0x20 * j)	0x4807 6204 + (0x20 * j)
SLIMBUS_DCT_FIFO_SETUP2 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0208 + (0x20 * j)	0x4807 6208 + (0x20 * j)
SLIMBUS_DCT_FIFO_STATUS <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 020C + (0x20 * j)	0x4807 620C + (0x20 * j)
SLIMBUS_DCT_MAP <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0210 + (0x20 * j)	0x4807 6210 + (0x20 * j)
SLIMBUS_DCT_CONFIG1 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0214 + (0x20 * j)	0x4807 6214 + (0x20 * j)
SLIMBUS_DCT_CONFIG2 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0218 + (0x20 * j)	0x4807 6218 + (0x20 * j)
SLIMBUS_DCT_DATA <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 021C + (0x20 * j)	0x4807 621C + (0x20 * j)
SLIMBUS_DCR_INFO <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0280 + (0x20 * j)	0x4807 6280 + (0x20 * j)
SLIMBUS_DCR_FIFO_SETUP1 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0284 + (0x20 * j)	0x4807 6284 + (0x20 * j)
SLIMBUS_DCR_FIFO_SETUP2 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0288 + (0x20 * j)	0x4807 6288 + (0x20 * j)
SLIMBUS_DCR_FIFO_STATUS <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 028C + (0x20 * j)	0x4807 628C + (0x20 * j)
SLIMBUS_DCR_MAP <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0290 + (0x20 * j)	0x4807 6290 + (0x20 * j)
SLIMBUS_DCR_CONFIG1 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0294 + (0x20 * j)	0x4807 6294 + (0x20 * j)
SLIMBUS_DCR_CONFIG2 <sub>j</sub> <sup>(2)</sup>	RW	32	0x0000 0298 + (0x20 * j)	0x4807 6298 + (0x20 * j)
SLIMBUS_DCR_DATA <sub>j</sub> <sup>(2)</sup>	R	32	0x0000 029C + (0x20 * j)	0x4807 629C + (0x20 * j)

<sup>(1)</sup> i = 0 to 6

<sup>(2)</sup> j = 0 to 3

### 23.9.6.3.3 SLIMbus Register Descriptions

**Table 23-689. SLIMBUS\_CMP\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	IP Revision Identifier Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Revision Number	R	0x---- ---- <sup>(1)</sup>

<sup>(1)</sup> TI internal data.

**Table 23-690. Register Call Summary for Register SLIMBUS\_CMP\_REVISION**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMBUS1 Register Summary: \[0\]](#)
- [SLIMBUS2 Register Summary: \[1\]](#)

**Table 23-691. SLIMBUS\_CMP\_HWINFO**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Those read-only, hardcoded fields display the values of the eponymous hardware configuration depending on module instantiation - SLIMBUS1 or SLIMBUS2.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEV		TDC		RDC		RRSZ				TRSZ				PSZ																	

Bits	Field Name	Description	Type	Reset
31:28	DEV	Device count: number of devices supported by the component, most notably through the automatically reception of the messages addressed to them. Sets the number of instances of the "DEV_XYZ" registers.	R	0x7
27:24	TDC	Transmit Data agent Count: Number of transmit hardware data agents (FIFO and associated logic) Sets the number of instances of the "DCT_XYZ" registers.	R	See <a href="#">Table 23-693</a>
23:20	RDC	Receive Data agent Count: Number of receive hardware data agents (FIFO and associated logic) Sets the number of instances of the "DCR_XYZ" registers.	R	See <a href="#">Table 23-693</a>
19:12	RRSZ	Size of receive FIFO RAM, in 32-bit words. RX message FIFO and all RX data FIFOs map to that shared RAM.	R	See <a href="#">Table 23-693</a>
11:4	TRSZ	Size of transmit FIFO RAM, in 32-bit words. TX message FIFO and all TX data FIFOs map to that shared RAM.	R	See <a href="#">Table 23-693</a>



Bits	Field Name	Description	Type	Reset
3:0	PSZ	Size of (both TX and RX) FIFO segment word pointers, in bits. Determines the maximum number of (variable-sized) segment words a data agent FIFO can contain. Sets the width of FIFO control and status fields: size, threshold, level.	R	0x8

**Table 23-692. Register Call Summary for Register SLIMBUS\_CMP\_HWINFO**

Serial Low-Power Inter-Chip Media Bus Controller

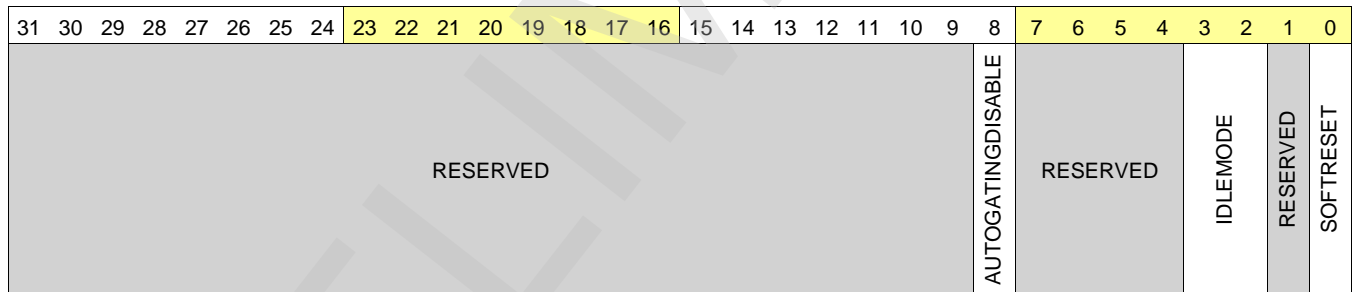
- [FIFO Management: \[0\] \[1\]](#)
- [SLIMBUS1 Register Summary: \[2\]](#)
- [SLIMBUS2 Register Summary: \[3\]](#)

**Table 23-693. Reset values for SLIMBUS\_CMP\_HWINFO**

	TDC	RDC	RRSZ	TRSZ
SLIMBUS1	0x8	0x8	0x38	0x38
SLIMBUS2	0x4	0x4	0x80	0x80

**Table 23-694. SLIMBUS\_CMP\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	This register allows controlling various parameters of the interconnect interface		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	AUTOGATINGDISABLE	Control of the internal clock autogating for safety / debug only. No impact on module functionality, only on dynamic power. 0x0: Internal clocks are gated off when unused, power is optimized. 0x1: Internal clocks are free-running, maximum dynamic power.	RW	0
7:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:2	IDLEMODE	<p>Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.</p> <p>0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. Backup mode, for debug only.</p> <p>0x1: No-idle mode: local target never enters idle state. Backup mode, for debug only.</p> <p>0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.</p> <p>0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events.</p>	RW	0x2
1	RESERVED		R	0
0	SOFTRESET	<p>Software reset. (Optional)</p> <p>Write 0x0: No action</p> <p>Write 0x1: Initiate software reset</p> <p>Read 0x1: Reset (software or other) ongoing</p> <p>Read 0x0: Reset done, no pending action</p>	RW	0

**Table 23-695. Register Call Summary for Register SLIMBUS\_CMP\_SYSCONFIG**

Serial Low-Power Inter-Chip Media Bus Controller

- [Software Reset: \[0\]](#)
- [Power Management: \[1\] \[2\]](#)
- [SLIMbus Global Initialization: \[3\] \[4\] \[5\] \[6\]](#)
- [SLIMBUS1 Register Summary: \[7\]](#)
- [SLIMBUS2 Register Summary: \[8\]](#)

**Table 23-696. SLIMBUS\_CMP\_IRQSTATUS\_RAW**

<b>Address Offset</b>	0x0000 0024																															
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>																															
<b>Description</b>	<p>Component (that is, main) interrupt request status. Check the corresponding secondary status register. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.</p>																															
<b>Type</b>	RW																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED								DCR_INFO_7	DCR_INFO_6	DCR_INFO_5	DCR_INFO_4	DCR_INFO_3	DCR_INFO_2	DCR_INFO_1	DCR_INFO_0	DCT_INFO_7	DCT_INFO_6	DCT_INFO_5	DCT_INFO_4	DCT_INFO_3	DCT_INFO_2	DCT_INFO_1	DCT_INFO_0	SMR_INFO	SMT_INFO	FR_INFO	FL_INFO				

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19	DCR_INFO_7 <sup>(1)</sup>	IRQ status for RX Data agent 7 <sup>(1)</sup> If implemented: Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
18	DCR_INFO_6 <sup>(1)</sup>	IRQ status for RX Data agent 6 <sup>(1)</sup> If implemented: Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
17	DCR_INFO_5 <sup>(1)</sup>	IRQ status for RX Data agent 5 <sup>(1)</sup> If implemented: Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
16	DCR_INFO_4 <sup>(1)</sup>	IRQ status for RX Data agent 4 <sup>(1)</sup> If implemented: Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
15	DCR_INFO_3 <sup>(1)</sup>	IRQ status for RX Data agent 3 <sup>(1)</sup> If implemented: Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
14	DCR_INFO_2 <sup>(1)</sup>	IRQ status for RX Data agent 2 <sup>(1)</sup> If implemented: Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
13	DCR_INFO_1 <sup>(1)</sup>	IRQ status for RX Data agent 1 <sup>(1)</sup> If implemented: Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
12	DCR_INFO_0 <sup>(1)</sup>	IRQ status for RX Data agent 0 <sup>(1)</sup> If implemented: Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
11	DCT_INFO_7 <sup>(2)</sup>	IRQ status for TX Data agent 7 <sup>(2)</sup> Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0

<sup>(1)</sup> This bit field is RESERVED in SLIMBUS2

<sup>(2)</sup> This bit field is DCR\_INFO\_3 (IRQ status for RX Data agent 3) in SLIMBUS2

Bits	Field Name	Description	Type	Reset
10	DCT_INFO_6 <sup>(3)</sup>	IRQ status for TX Data agent 6 <sup>(3)</sup> Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
9	DCT_INFO_5 <sup>(4)</sup>	IRQ status for TX Data agent 5 <sup>(4)</sup> Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
8	DCT_INFO_4 <sup>(5)</sup>	IRQ status for TX Data agent 4 <sup>(5)</sup> Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
7	DCT_INFO_3	IRQ status for TX Data agent 3 Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
6	DCT_INFO_2	IRQ status for TX Data agent 2 Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
5	DCT_INFO_1	IRQ status for TX Data agent 1 Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
4	DCT_INFO_0	IRQ status for TX Data agent 0 Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
3	SMR_INFO	IRQ status for message receive Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
2	SMT_INFO	IRQ status for message transmit Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0

<sup>(3)</sup> This bit field is DCR\_INFO\_2 (IRQ status for RX Data agent 2) in SLIMBUS2

<sup>(4)</sup> This bit field is DCR\_INFO\_1 (IRQ status for RX Data agent 1) in SLIMBUS2

<sup>(5)</sup> This bit field is DCR\_INFO\_0 (IRQ status for RX Data agent 0) in SLIMBUS2

Bits	Field Name	Description	Type	Reset
1	FR_INFO	IRQ status for Framer device, when active Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0
0	FL_INFO	IRQ status for Frame Layer Write 0x0: No action Write 0x1: Trigger IRQ event by software Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toSet	0

**Table 23-697. Register Call Summary for Register SLIMBUS\_CMP\_IRQSTATUS\_RAW**

Serial Low-Power Inter-Chip Media Bus Controller

- [Interrupt Requests: \[0\]](#)
- [SLIMBUS1 Register Summary: \[1\]](#)
- [SLIMBUS2 Register Summary: \[2\]](#)

**Table 23-698. SLIMBUS\_CMP\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0028
<b>Physical Address</b>	See <b>Instance</b> <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>
<b>Description</b>	Component (that is, main) interrupt request status. Check the corresponding secondary status register. Enabled status isn't set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED																DCR_INFO_7	DCR_INFO_6	DCR_INFO_5	DCR_INFO_4	DCR_INFO_3	DCR_INFO_2	DCR_INFO_1	DCR_INFO_0	DCT_INFO_7	DCT_INFO_6	DCT_INFO_5	DCT_INFO_4	DCT_INFO_3	DCT_INFO_2	DCT_INFO_1	DCT_INFO_0	SMR_INFO	SMT_INFO	FR_INFO	FL_INFO

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19	DCR_INFO_7 <sup>(1)</sup>	IRQ status for RX Data agent 7 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
18	DCR_INFO_6 <sup>(1)</sup>	IRQ status for RX Data agent 6 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0

<sup>(1)</sup> This bit field is RESERVED in SLIMBUS2

Bits	Field Name	Description	Type	Reset
17	DCR_INFO_5 <sup>(1)</sup>	IRQ status for RX Data agent 5 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
16	DCR_INFO_4 <sup>(1)</sup>	IRQ status for RX Data agent 4 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
15	DCR_INFO_3 <sup>(1)</sup>	IRQ status for RX Data agent 3 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
14	DCR_INFO_2 <sup>(1)</sup>	IRQ status for RX Data agent 2 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
13	DCR_INFO_1 <sup>(2)</sup>	IRQ status for RX Data agent 1 <sup>(2)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
12	DCR_INFO_0 <sup>(2)</sup>	IRQ status for RX Data agent 0 <sup>(2)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
11	DCT_INFO_7 <sup>(3)</sup>	IRQ status for TX Data agent 7 <sup>(3)</sup> Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
10	DCT_INFO_6 <sup>(4)</sup>	IRQ status for TX Data agent 6 <sup>(4)</sup> Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
9	DCT_INFO_5 <sup>(5)</sup>	IRQ status for TX Data agent 5 <sup>(5)</sup> Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0

<sup>(2)</sup> This bit field is RESERVED in SLIMBUS2

<sup>(3)</sup> This bit field is DCR\_INFO\_3 (IRQ status for RX Data agent 3) in SLIMBUS2

<sup>(4)</sup> This bit field is DCR\_INFO\_2 (IRQ status for RX Data agent 2) in SLIMBUS2

<sup>(5)</sup> This bit field is DCR\_INFO\_1 (IRQ status for RX Data agent 1) in SLIMBUS2

Bits	Field Name	Description	Type	Reset
8	DCT_INFO_4 <sup>(6)</sup>	IRQ status for TX Data agent 4 <sup>(6)</sup> Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
7	DCT_INFO_3	IRQ status for TX Data agent 3 Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
6	DCT_INFO_2	IRQ status for TX Data agent 2 Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
5	DCT_INFO_1	IRQ status for TX Data agent 1 Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
4	DCT_INFO_0	IRQ status for TX Data agent 0 Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
3	SMR_INFO	IRQ status for message receive Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
2	SMT_INFO	IRQ status for message transmit Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
1	FR_INFO	IRQ status for Framer device, when active Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
0	FL_INFO	IRQ status for Frame Layer Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0

<sup>(6)</sup> This bit field is DCR\_INFO\_0 (IRQ status for RX Data agent 0) in SLIMBUS2

**Table 23-699. Register Call Summary for Register SLIMBUS\_CMP\_IRQSTATUS**

Serial Low-Power Inter-Chip Media Bus Controller

- TX Message Sequence: [0] [1]
- RX Message Sequence: [2]
- Interrupt Service Routines: [3]
- Interrupt Requests: [4]
- Interface Device ISRs: [5]
- Framing Device ISRs: [6]
- Shared Message Channel Transmit ISRs: [7]
- Shared Message Channel Receive ISRs: [8]
- Data Channel Port ISRs: [9]
- SLIMbus interrupts configuration: [10] [11]
- Transmitting message, interrupt mode: [12]
- Receiving message, interrupt mode: [13] [14]
- SLIMBUS1 Register Summary: [15]
- SLIMBUS2 Register Summary: [16]

**Table 23-700. SLIMBUS\_CMP\_IRQENABLE\_SET**

<b>Address Offset</b>	0x0000 002C		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Component (that is, main) interrupt request enable Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DCR_INFO_7_EN	DCR_INFO_6_EN	DCR_INFO_5_EN	DCR_INFO_4_EN	DCR_INFO_3_EN	DCR_INFO_2_EN	DCR_INFO_1_EN	DCR_INFO_0_EN	DCT_INFO_7_E7	DCT_INFO_6_EN	DCT_INFO_5_EN	DCT_INFO_4_EN	DCT_INFO_3_EN	DCT_INFO_2_EN	DCT_INFO_1_EN	DCT_INFO_0_EN	SMR_INFO_EN	SMT_INFO_EN	FR_INFO_EN	FL_INFO_EN				

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19	DCR_INFO_7_EN <sup>(1)</sup>	IRQ enable for RX Data agent 7 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
18	DCR_INFO_6_EN <sup>(1)</sup>	IRQ enable for RX Data agent 6 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
17	DCR_INFO_5_EN <sup>(1)</sup>	IRQ enable for RX Data agent 5 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0

<sup>(1)</sup> This bit field is RESERVED in SLIMBUS2



Bits	Field Name	Description	Type	Reset
16	DCR_INFO_4_EN <sup>(1)</sup>	IRQ enable for RX Data agent 4 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
15	DCR_INFO_3_EN <sup>(1)</sup>	IRQ enable for RX Data agent 3 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
14	DCR_INFO_2_EN <sup>(1)</sup>	IRQ enable for RX Data agent 2 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
13	DCR_INFO_1_EN <sup>(2)</sup>	IRQ enable for RX Data agent 1 <sup>(2)</sup> . If implemented: Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
12	DCR_INFO_0_EN <sup>(2)</sup>	IRQ enable for RX Data agent 0 <sup>(2)</sup> . If implemented: Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
11	DCT_INFO_7_EN <sup>(3)</sup>	IRQ enable for TX Data agent 7 <sup>(3)</sup> Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
10	DCT_INFO_6_EN <sup>(4)</sup>	IRQ enable for TX Data agent 6 <sup>(4)</sup> Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
9	DCT_INFO_5_EN <sup>(5)</sup>	IRQ enable for TX Data agent 5 <sup>(5)</sup> Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
8	DCT_INFO_4_EN <sup>(6)</sup>	IRQ enable for TX Data agent 4 <sup>(6)</sup> Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0

<sup>(2)</sup> This bit field is RESERVED in SLIMBUS2

<sup>(3)</sup> This bit field is DCR\_INFO\_3\_EN (IRQ enable for RX Data agent 3) in SLIMBUS2

<sup>(4)</sup> This bit field is DCR\_INFO\_2\_EN (IRQ enable for RX Data agent 2) in SLIMBUS2

<sup>(5)</sup> This bit field is DCR\_INFO\_1\_EN (IRQ enable for RX Data agent 1) in SLIMBUS2

<sup>(6)</sup> This bit field is DCR\_INFO\_0\_EN (IRQ enable for RX Data agent 0) in SLIMBUS2

Bits	Field Name	Description	Type	Reset
7	DCT_INFO_3_EN	IRQ enable for TX Data agent 3 Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
6	DCT_INFO_2_EN	IRQ enable for TX Data agent 2 Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
5	DCT_INFO_1_EN	IRQ enable for TX Data agent 1 Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
4	DCT_INFO_0_EN	IRQ enable for TX Data agent 0 Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
3	SMR_INFO_EN	IRQ enable for message receive Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
2	SMT_INFO_EN	IRQ enable for message transmit Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
1	FR_INFO_EN	IRQ enable for Framer device, when active Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0
0	FL_INFO_EN	IRQ enable for Frame Layer Write 0x0: No action Write 0x1: Set IRQ enable (that is, enable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toSet	0

**Table 23-701. Register Call Summary for Register SLIMBUS\_CMP\_IRQENABLE\_SET**

Serial Low-Power Inter-Chip Media Bus Controller

- TX Message Sequence: [0] [1]
- RX Message Sequence: [2]
- Interrupt Service Routines: [3]
- Interrupt Requests: [4]
- Interface Device ISRs: [5]
- Framers Device ISRs: [6]
- Shared Message Channel Transmit ISRs: [7]
- Shared Message Channel Receive ISRs: [8]
- Data Channel Port ISRs: [9]
- SLIMbus interrupts configuration: [10] [11]
- SLIMBUS1 Register Summary: [12]
- SLIMBUS2 Register Summary: [13]

**Table 23-702. SLIMBUS\_CMP\_IRQENABLE\_CLR**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Component (that is, main) interrupt request enable Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED																DCR_INFO_7_EN	DCR_INFO_6_EN	DCR_INFO_5_EN	DCR_INFO_4_EN	DCR_INFO_3_EN	DCR_INFO_2_EN	DCR_INFO_1_EN	DCR_INFO_0_EN	DCT_INFO_7_E7	DCT_INFO_6_EN	DCT_INFO_5_EN	DCT_INFO_4_EN	DCT_INFO_3_EN	DCT_INFO_2_EN	DCT_INFO_1_EN	DCT_INFO_0_EN	SMR_INFO_EN	SMT_INFO_EN	FR_INFO_EN	FL_INFO_EN

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19	DCR_INFO_7_EN <sup>(1)</sup>	IRQ enable for RX data agent 7 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
18	DCR_INFO_6_EN <sup>(1)</sup>	IRQ enable for RX data agent 6 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
17	DCR_INFO_5_EN <sup>(1)</sup>	IRQ enable for RX data agent 5 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0

<sup>(1)</sup> This bit field is RESERVED in SLIMBUS2

Bits	Field Name	Description	Type	Reset
16	DCR_INFO_4_EN <sup>(1)</sup>	IRQ enable for RX data agent 4 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
15	DCR_INFO_3_EN <sup>(1)</sup>	IRQ enable for RX data agent 3 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
14	DCR_INFO_2_EN <sup>(1)</sup>	IRQ enable for RX data agent 2 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
13	DCR_INFO_1_EN <sup>(2)</sup>	IRQ enable for RX data agent 1 <sup>(2)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
12	DCR_INFO_0_EN <sup>(2)</sup>	IRQ enable for RX data agent 0 <sup>(2)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
11	DCT_INFO_7_EN <sup>(3)</sup>	IRQ enable for TX Data agent 7 <sup>(3)</sup> Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
10	DCT_INFO_6_EN <sup>(4)</sup>	IRQ enable for TX Data agent 6 <sup>(4)</sup> Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
9	DCT_INFO_5_EN <sup>(5)</sup>	IRQ enable for TX Data agent 5 <sup>(5)</sup> Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
8	DCT_INFO_4_EN <sup>(6)</sup>	IRQ enable for TX Data agent 4 <sup>(6)</sup> Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0

<sup>(2)</sup> This bit field is RESERVED in SLIMBUS2

<sup>(3)</sup> This bit field is DCR\_INFO\_3\_EN (IRQ enable for RX Data agent 3) in SLIMBUS2

<sup>(4)</sup> This bit field is DCR\_INFO\_2\_EN (IRQ enable for RX Data agent 2) in SLIMBUS2

<sup>(5)</sup> This bit field is DCR\_INFO\_1\_EN (IRQ enable for RX Data agent 1) in SLIMBUS2

<sup>(6)</sup> This bit field is DCR\_INFO\_0\_EN (IRQ enable for RX Data agent 0) in SLIMBUS2

Bits	Field Name	Description	Type	Reset
7	DCT_INFO_3_EN	IRQ enable for TX Data agent 3 Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
6	DCT_INFO_2_EN	IRQ enable for TX Data agent 2 Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
5	DCT_INFO_1_EN	IRQ enable for TX Data agent 1 Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
4	DCT_INFO_0_EN	IRQ enable for TX Data agent 0 Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
3	SMR_INFO_EN	IRQ enable for message receive Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
2	SMT_INFO_EN	IRQ enable for message transmit Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
1	FR_INFO_EN	IRQ enable for Framer device, when active Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0
0	FL_INFO_EN	IRQ enable for Frame Layer Write 0x0: No action Write 0x1: Clear IRQ enable (that is, disable event) Read 0x1: IRQ event is enabled Read 0x0: IRQ event is disabled	RW W1toClr	0

**Table 23-703. Register Call Summary for Register SLIMBUS\_CMP\_IRQENABLE\_CLR**

Serial Low-Power Inter-Chip Media Bus Controller

- [Interrupt Service Routines: \[0\]](#)
- [Interrupt Requests: \[1\]](#)
- [SLIMbus interrupts configuration: \[2\] \[3\]](#)
- [SLIMBUS1 Register Summary: \[4\]](#)
- [SLIMBUS2 Register Summary: \[5\]](#)

**Table 23-704. SLIMBUS\_CMP\_DMAENABLE\_SET**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Components DMA enable (1 bit per DMA-capable channel) Write 1 to set (enable DMA). Readout equal to corresponding _CLR register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DCR_DMA_7_EN	DCR_DMA_6_EN	DCR_DMA_5_EN	DCR_DMA_4_EN	DCR_DMA_3_EN	DCR_DMA_2_EN	DCR_DMA_1_EN	DCR_DMA_0_EN	DCT_DMA_7_EN	DCT_DMA_6_EN	DCT_DMA_5_EN	DCT_DMA_4_EN	DCT_DMA_3_EN	DCT_DMA_2_EN	DCT_DMA_1_EN	DCT_DMA_0_EN	RESERVED							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19	DCR_DMA_7_EN <sup>(1)</sup>	DMA enable for RX Data agent 7 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet.	0
18	DCR_DMA_6_EN <sup>(1)</sup>	DMA enable for RX Data agent 6 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet.	0
17	DCR_DMA_5_EN <sup>(1)</sup>	DMA enable for RX Data agent 5 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet.	0
16	DCR_DMA_4_EN <sup>(1)</sup>	DMA enable for RX Data agent 4 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet.	0
15	DCR_DMA_3_EN <sup>(1)</sup>	DMA enable for RX Data agent 3 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet.	0
14	DCR_DMA_2_EN <sup>(1)</sup>	DMA enable for RX Data agent 2 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet.	0

<sup>(1)</sup> This bit field is RESERVED in SLIMBUS2

Bits	Field Name	Description	Type	Reset
13	DCR_DMA_1_EN <sup>(1)</sup>	DMA enable for RX Data agent 1 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet.	0
12	DCR_DMA_0_EN <sup>(1)</sup>	DMA enable for RX Data agent 0 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet.	0
11	DCT_DMA_7_EN <sup>(2)</sup>	DMA enable for TX Data agent 7 <sup>(2)</sup> Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet	0
10	DCT_DMA_6_EN <sup>(3)</sup>	DMA enable for TX Data agent 6 <sup>(3)</sup> Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet	0
9	DCT_DMA_5_EN <sup>(4)</sup>	DMA enable for TX Data agent 5 <sup>(4)</sup> Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet	0
8	DCT_DMA_4_EN <sup>(5)</sup>	DMA enable for TX Data agent 4 <sup>(5)</sup> Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet	0
7	DCT_DMA_3_EN	DMA enable for TX Data agent 3 Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet	0
6	DCT_DMA_2_EN	DMA enable for TX Data agent 2 Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet	0
5	DCT_DMA_1_EN	DMA enable for TX Data agent 1 Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet	0

<sup>(2)</sup> This bit field is DCR\_DMA\_3\_EN (DMA enable for RX Data agent 3) in SLIMBUS2

<sup>(3)</sup> This bit field is DCR\_DMA\_2\_EN (DMA enable for RX Data agent 2) in SLIMBUS2

<sup>(4)</sup> This bit field is DCR\_DMA\_1\_EN (DMA enable for RX Data agent 1) in SLIMBUS2

<sup>(5)</sup> This bit field is DCR\_DMA\_0\_EN (DMA enable for RX Data agent 0) in SLIMBUS2

Bits	Field Name	Description	Type	Reset
4	DCT_DMA_0_EN	DMA enable for TX Data agent 0 Write 0x0: No action Write 0x1: Set DMA enable (that is, enable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toSet	0
3:0	RESERVED		R	0x0

**Table 23-705. Register Call Summary for Register SLIMBUS\_CMP\_DMAENABLE\_SET**

Serial Low-Power Inter-Chip Media Bus Controller

- [Data Input/Output: DMA Request Mode: \[0\]](#)
- [SLIMbus DMA configuration: \[1\] \[2\]](#)
- [SLIMBUS1 Register Summary: \[3\]](#)
- [SLIMBUS2 Register Summary: \[4\]](#)

**Table 23-706. SLIMBUS\_CMP\_DMAENABLE\_CLR**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Components DMA enable (1 bit per DMA-capable channel) Write 1 to clear (disable DMA). Readout equal to corresponding _SET register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DCR_DMA_7_EN	DCR_DMA_6_EN	DCR_DMA_5_EN	DCR_DMA_4_EN	DCR_DMA_3_EN	DCR_DMA_2_EN	DCR_DMA_1_EN	DCR_DMA_0_EN	DCT_DMA_7_EN	DCT_DMA_6_EN	DCT_DMA_5_EN	DCT_DMA_4_EN	DCT_DMA_3_EN	DCT_DMA_2_EN	DCT_DMA_1_EN	DCT_DMA_0_EN	RESERVED							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19	DCR_DMA_7_EN <sup>(1)</sup>	DMA enable for RX Data agent 7 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
18	DCR_DMA_6_EN <sup>(1)</sup>	DMA enable for RX Data agent 6 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
17	DCR_DMA_5_EN <sup>(1)</sup>	DMA enable for RX Data agent 5 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0

<sup>(1)</sup> This bit field is RESERVED in SLIMBUS2



Bits	Field Name	Description	Type	Reset
16	DCR_DMA_4_EN <sup>(1)</sup>	DMA enable for RX Data agent 4 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
15	DCR_DMA_3_EN <sup>(1)</sup>	DMA enable for RX Data agent 3 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
14	DCR_DMA_2_EN <sup>(1)</sup>	DMA enable for RX Data agent 2 <sup>(1)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
13	DCR_DMA_1_EN <sup>(2)</sup>	DMA enable for RX Data agent 1 <sup>(2)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
12	DCR_DMA_0_EN <sup>(2)</sup>	DMA enable for RX Data agent 0 <sup>(2)</sup> . If implemented: Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
11	DCT_DMA_7_EN <sup>(3)</sup>	DMA enable for TX Data agent 7 <sup>(3)</sup> Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
10	DCT_DMA_6_EN <sup>(4)</sup>	DMA enable for TX Data agent 6 <sup>(4)</sup> Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
9	DCT_DMA_5_EN <sup>(5)</sup>	DMA enable for TX Data agent 5 <sup>(5)</sup> Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
8	DCT_DMA_4_EN <sup>(6)</sup>	DMA enable for TX Data agent 4 <sup>(6)</sup> Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0

<sup>(2)</sup> This bit field is RESERVED in SLIMBUS2

<sup>(3)</sup> This bit field is DCR\_DMA\_3\_EN (DMA enable for RX Data agent 3) in SLIMBUS2

<sup>(4)</sup> This bit field is DCR\_DMA\_2\_EN (DMA enable for RX Data agent 2) in SLIMBUS2

<sup>(5)</sup> This bit field is DCR\_DMA\_1\_EN (DMA enable for RX Data agent 1) in SLIMBUS2

<sup>(6)</sup> This bit field is DCR\_DMA\_0\_EN (DMA enable for RX Data agent 0) in SLIMBUS2

Bits	Field Name	Description	Type	Reset
7	DCT_DMA_3_EN	DMA enable for TX Data agent 3 Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
6	DCT_DMA_2_EN	DMA enable for TX Data agent 2 Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
5	DCT_DMA_1_EN	DMA enable for TX Data agent 1 Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
4	DCT_DMA_0_EN	DMA enable for TX Data agent 0 Write 0x0: No action Write 0x1: Clear DMA enable (that is, disable channel) Read 0x1: DMA channel is enabled Read 0x0: DMA channel is disabled	RW W1toClr	0
3:0	RESERVED		R	0x0

**Table 23-707. Register Call Summary for Register SLIMBUS\_CMP\_DMAENABLE\_CLR**

Serial Low-Power Inter-Chip Media Bus Controller

- [Data Input/Output: DMA Request Mode: \[0\]](#)
- [SLIMbus DMA configuration: \[1\] \[2\]](#)
- [SLIMBUS1 Register Summary: \[3\]](#)
- [SLIMBUS2 Register Summary: \[4\]](#)

**Table 23-708. SLIMBUS\_CMP\_IV**

<b>Address Offset</b>	0x0000 0040		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Table 23-687 to Table 23-688 Component IV field, used as lower 8 bits of the devices' Enumeration Address (EA[47:0]). Identical for all devices of the component.		
<b>Type</b>	RW		
	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	RESERVED		IV
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b> <b>Reset</b>
31:8	RESERVED		R      0x000000
7:0	IV	Device component's Instance Value (IV[7:0])	RW      0x00

**Table 23-709. Register Call Summary for Register SLIMBUS\_CMP\_IV**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device and Addressing Management: \[0\] \[1\]](#)
- [SLIMBUS1 Register Summary: \[2\]](#)
- [SLIMBUS2 Register Summary: \[3\]](#)

**Table 23-710. SLIMBUS\_CMP\_MI\_PC**

<b>Address Offset</b>	0x0000 0044		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Component MI and PC fields, used as upper 32 bits of the devices' Enumeration Address (EA[47:0]). Identical for all devices of the component.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC																MI															

Bits	Field Name	Description	Type	Reset
31:16	PC	Device component's Product Code (PC[15:0])	RW	0x0000
15:0	MI	Device component's Manufacturer Index (MI[15:0])	RW	0x0102

**Table 23-711. Register Call Summary for Register SLIMBUS\_CMP\_MI\_PC**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device and Addressing Management: \[0\] \[1\] \[2\] \[3\]](#)
- [SLIMBUS1 Register Summary: \[4\]](#)
- [SLIMBUS2 Register Summary: \[5\]](#)

**Table 23-712. SLIMBUS\_SMT\_INFO**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	IRQ status for Shared Message Transmit channel. Write 1 to a bit to clear it.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							ABORT	RESERVED	UDEF	NORE	NACK	PACK	OVERFLOW	MC_TX_COL	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	ABORT	TX message was aborted before completion because of message sync loss. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
6	RESERVED		R	0
5	UDEF	TX message was UDEF'ed: undefined response, protocol error. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
4	NORE	TX message was NORE'ed: no or all-zero message response Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
3	NACK	TX message was NACK'ed: at least one of the recipient devices requested message retransmission. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
2	PACK	TX message was PACK'ed: all recipient devices accepted the message. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
1	OVERFLOW	Overflow in TX message FIFO: SW tried to write more bytes than available in the FIFO. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
0	MC_TX_COL	Transmit collision in Message Channel (MC) during message transmission. Interface device class-specific information element (IE) Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0

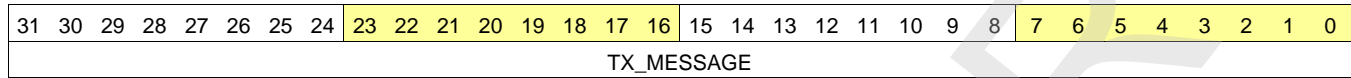
**Table 23-713. Register Call Summary for Register SLIMBUS\_SMT\_INFO**

## Serial Low-Power Inter-Chip Media Bus Controller

- [Device reset: \[0\]](#)
- [TX Message Sequence: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [TX Message FIFO Clearing: \[9\] \[10\]](#)
- [RX Message FIFO Clearing: \[11\]](#)
- [Possible Interrupts: \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [Found message synchronization \(FOUND\\_MS\): \[18\]](#)
- [Transmit collision in message body \(MC\\_TX\\_COL\): \[19\]](#)
- [Tx message FIFO overflow \(OVERFLOW\): \[20\]](#)
- [Received positive acknowledge message response \(PACK\): \[21\]](#)
- [Received negative acknowledge message response \(NACK\): \[22\]](#)
- [Received no message response \(NORE\): \[23\]](#)
- [Received undefined message response \(UDEF\): \[24\]](#)
- [Transmit collision in message response \(MC\\_TX\\_COL\): \[25\]](#)
- [Transmitting Message, Polling Mode: \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\]](#)
- [Transmitting message, interrupt mode: \[35\] \[36\] \[37\]](#)
- [SLIMBUS1 Register Summary: \[38\]](#)
- [SLIMBUS2 Register Summary: \[39\]](#)

**Table 23-714. SLIMBUS\_SMT\_MESSAGE**

<b>Address Offset</b>	0x0000 0054		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Shared Message Transmit FIFO input		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:0	TX_MESSAGE	TX message byte(s), depending on packing mode. Reads return 0. Enter the whole message except last byte (Message Integrity MI and Response MR), and with a don't care placeholder for Primary Integrity (PI).	RW	0x0000 0000

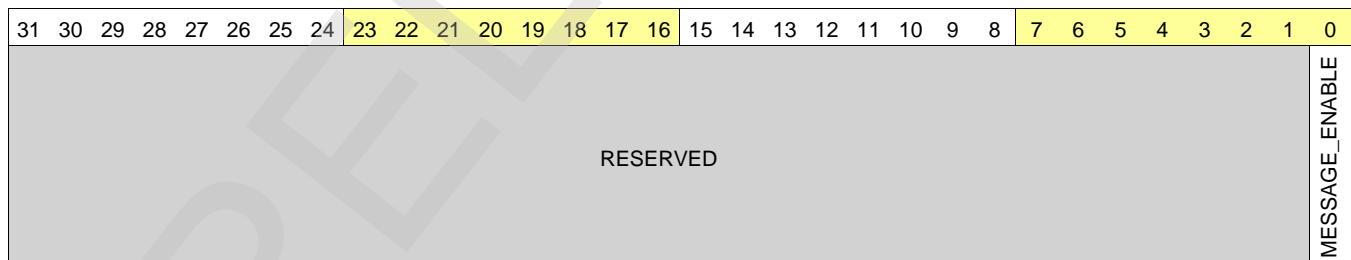
**Table 23-715. Register Call Summary for Register SLIMBUS\_SMT\_MESSAGE**

Serial Low-Power Inter-Chip Media Bus Controller

- [TX Message Sequence: \[0\]](#)
- [Transmitting Message, Polling Mode: \[1\] \[2\]](#)
- [Transmitting message, interrupt mode: \[3\] \[4\]](#)
- [SLIMBUS1 Register Summary: \[5\]](#)
- [SLIMBUS2 Register Summary: \[6\]](#)

**Table 23-716. SLIMBUS\_SMT\_CONTROL**

<b>Address Offset</b>	0x0000 0058		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Control of Shared Message channel Transmission.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MESSAGE_ENABLE	Enable the transmission of the message(s) previously written into the TX Message FIFO, with unlimited arbitration auto-retries. Self-cleared after either a message not get PACK'ed or the FIFO is empty. Write 0x0: No effect Write 0x1: Request transmission of stored messages Read 0x1: Message transmission is ongoing Read 0x0: Message transmission is stopped / complete	RW	0

**Table 23-717. Register Call Summary for Register SLIMBUS\_SMT\_CONTROL**

Serial Low-Power Inter-Chip Media Bus Controller

- [TX Message Sequence](#): [0] [1] [2]
- [TX Message FIFO Clearing](#): [3]
- [Transmitting Message, Polling Mode](#): [4] [5] [6] [7]
- [Transmitting message, interrupt mode](#): [8] [9] [10]
- [SLIMBUS1 Register Summary](#): [11]
- [SLIMBUS2 Register Summary](#): [12]

**Table 23-718. SLIMBUS\_SMT\_FIFO\_SETUP**

<b>Address Offset</b>	0x0000 005C		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a> Shared Message Transmit channel configuration. To be kept static during operation. Note that message segment word size is 2 slots = 1 byte (= message size granularity) and that message FIFO base address is always 0x0.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PACKING	SIZE_HI	SIZE_LO	FIFO_CLEAR												

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	PACKING	Packing enable control for TX message FIFO 0x0: Packing disabled: one byte per access only (LSByte), byte enable is don't care. 0x1: Packing enabled: one or several bytes per access, depending on byte enable.	RW	0
6:5	SIZE_HI	(Upper bits of the) capacity of shared TX message FIFO, in bytes, minus one. To be set for largest supported (transmitted) message(s). 0x0: 16 bytes (minimum size) 0x1: 32 bytes (recommended) 0x3: 64 bytes (max size) 0x2: 48 bytes (large messages)	RW	0x1
4:1	SIZE_LO	(Lower bits of the) capacity of shared TX message FIFO, in bytes, minus one. Read-only, for reference only.	R	0xF
0	FIFO_CLEAR	Returns FIFO to initial empty state. Write 0x0: No effect Write 0x1: Clear FIFO Read 0x1: Ongoing Clear. Completion requires the SLIMbus clock. Read 0x0: No event, last clear completed	RW	0

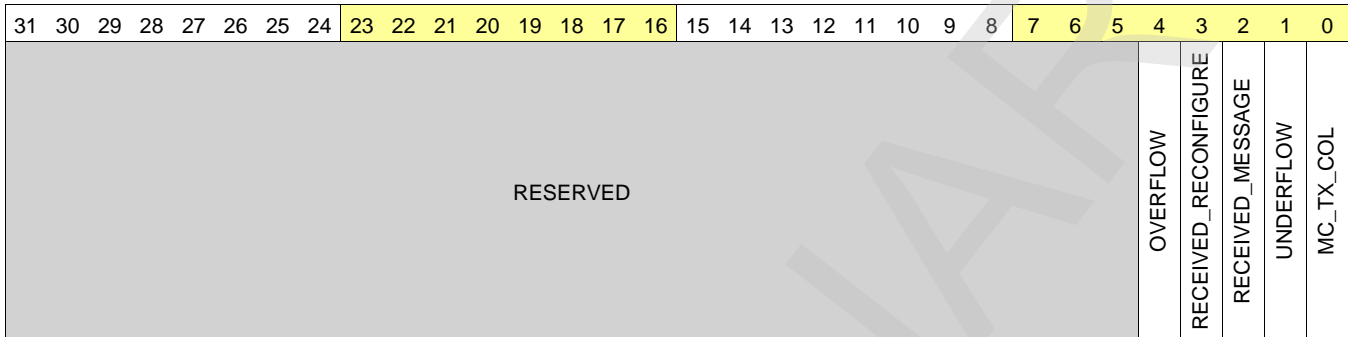
**Table 23-719. Register Call Summary for Register SLIMBUS\_SMT\_FIFO\_SETUP**

Serial Low-Power Inter-Chip Media Bus Controller

- [FIFO Mapping](#): [0] [1]
- [SLIMbus Global Initialization](#): [2]
- [SLIMBUS1 Register Summary](#): [3]
- [SLIMBUS2 Register Summary](#): [4]

**Table 23-720. SLIMBUS\_SMR\_INFO**

<b>Address Offset</b>	0x0000 0060		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	Table 23-687 to Table 23-688		
<b>Description</b>	IRQ status for Shared Message Receive channel. Write 1 to a bit to clear it.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x00000000
4	OVERFLOW	Overflow in RX message FIFO: received message is larger than the FIFO. Write 0x0: no action Write 0x1: clear event Read 0x1: event pending Read 0x0: no event pending	RW W1toClr	0
3	RECEIVED_RECONFIGURE	RECONFIGURE_NOW message received, available in RX message FIFO Write 0x0: no action Write 0x1: clear event Read 0x1: event pending Read 0x0: no event pending	RW W1toClr	0
2	RECEIVED_MESSAGE	Message available in the RX message FIFO. Clear bit to clear the FIFO and allow the reception of further messages (incoming messages will be NACK'ed until then). Write 0x0: no action Write 0x1: clear event, clear FIFO, allow further messages. Read 0x1: event pending Read 0x0: no event pending	RW W1toClr	0
1	UNDERFLOW	Underflow in RX message FIFO: SW tried to read out more bytes than available. Write 0x0: no action Write 0x1: clear event Read 0x1: event pending Read 0x0: no event pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
0	MC_TX_COL	Transmit collision in Message Channel (MC) during RX message during message reception. Interface device class-specific information element (IE)  Write 0x0: no action Write 0x1: clear event Read 0x1: event pending Read 0x0: no event pending	RW W1toClr	0

**Table 23-721. Register Call Summary for Register SLIMBUS\_SMR\_INFO**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device reset: \[0\]](#)
- [TX Message FIFO Clearing: \[1\] \[2\]](#)
- [RX Message Sequence: \[3\] \[4\] \[5\] \[6\]](#)
- [RX Message FIFO Clearing: \[7\] \[8\] \[9\]](#)
- [Possible Interrupts: \[10\] \[11\] \[12\]](#)
- [Found message synchronization \(FOUND\\_MS\): \[13\]](#)
- [Transmit collision in message response \(MC\\_TX\\_COL\): \[14\]](#)
- [Rx message FIFO underflow \(UNDERFLOW\): \[15\]](#)
- [Message received \(RECEIVED\\_MESSAGE\): \[16\]](#)
- [Reconfigure\\_now message received \(RECEIVED\\_RECONFIGURE\): \[17\]](#)
- [Receiving Message, POLLing Mode: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [Receiving message, interrupt mode: \[24\] \[25\] \[26\]](#)
- [SLIMBUS1 Register Summary: \[27\]](#)
- [SLIMBUS2 Register Summary: \[28\]](#)

**Table 23-722. SLIMBUS\_SMR\_MESSAGE**

<b>Address Offset</b>	0x0000 0064		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Shared Message Receive channel FIFO output.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_MESSAGE																															

Bits	Field Name	Description	Type	Reset
31:0	RX_MESSAGE	RX message byte(s), depending on packing mode. Writes have no effect. Readout value from an empty FIFO is undefined, that is, there is no defined "reset value". Message is guaranteed to have destination address matching a local device address (LA or EA), correct Primary and Message Integrity (PI & MI) and PACK Message Response (MR). Whole message available for readout, including the last byte (Message Integrity MI and Response MR)	R	0x0000 0000

**Table 23-723. Register Call Summary for Register SLIMBUS\_SMR\_MESSAGE**

Serial Low-Power Inter-Chip Media Bus Controller

- [RX Message Sequence: \[0\]](#)
- [Receiving Message, POLLing Mode: \[1\] \[2\]](#)
- [Receiving message, interrupt mode: \[3\] \[4\]](#)
- [SLIMBUS1 Register Summary: \[5\]](#)
- [SLIMBUS2 Register Summary: \[6\]](#)



**Table 23-724. SLIMBUS\_SMR\_CONTROL**

<b>Address Offset</b>	0x0000 0068		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Control of Shared Message channel Reception.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MESSAGE_DISABLE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MESSAGE_DISABLE	Disables the reception of incoming messages (that is, broadcast or locally addressed) into the RX Message FIFO. Same effect for the same value than SMR_INFO[2] RECEIVED_MESSAGE, but this bit can be set to 1. 0x0: Enabled: incoming messages will be PACK'ed 0x1: Disabled: incoming messages will be NACK'ed	RW	0

**Table 23-725. Register Call Summary for Register SLIMBUS\_SMR\_CONTROL**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMBUS1 Register Summary: \[0\]](#)
- [SLIMBUS2 Register Summary: \[1\]](#)

**Table 23-726. SLIMBUS\_SMR\_FIFO\_SETUP**

<b>Address Offset</b>	0x0000 006C		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Shared Message Receive channel setup. To be kept static during operation. Note that message segment word size is 2 slots = 1 byte (= message size granularity) and that message FIFO base address is always 0x0.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PACKING	SIZE_HI	SIZE_LO	FIFO_CLEAR												

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7	PACKING	Packing enable control for RX message FIFO 0x0: Packing disabled: one byte per access only (LSByte), byte enable is don't care. 0x1: Packing enabled: one or several bytes per access, depending on byte enable.	RW	0
6:5	SIZE_HI	(Upper bits of the) capacity of shared RX message FIFO, in bytes, minus one. To be set for largest supported (received) message. 0x0: 16 bytes (minimum size) 0x1: 32 bytes (recommended) 0x3: 64 bytes (max size) 0x2: 48 bytes (large messages)	RW	0x1
4:1	SIZE_LO	(Lower bits of the) capacity of shared RX message FIFO, in bytes, minus one. Read-only, for reference only.	R	0xF
0	FIFO_CLEAR	Returns message FIFO to initial empty state. Write 0x0: No effect Write 0x1: Clear FIFO Read 0x1: Ongoing Clear. Completion requires the SLIMbus clock. Read 0x0: No event, last clear completed	RW	0

**Table 23-727. Register Call Summary for Register SLIMBUS\_SMR\_FIFO\_SETUP**

Serial Low-Power Inter-Chip Media Bus Controller

- [FIFO Mapping: \[0\] \[1\]](#)
- [SLIMbus Global Initialization: \[2\]](#)
- [SLIMBUS1 Register Summary: \[3\]](#)
- [SLIMBUS2 Register Summary: \[4\]](#)

**Table 23-728. SLIMBUS\_FL\_INFO**

<b>Address Offset</b>	0x0000 0070																																																								
<b>Physical Address</b>	See	<b>Instance</b> SLIMBUS1 SLIMBUS2																																																							
<b>Description</b>	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a> Frame Layer IRQ status. Write 1 to a bit to clear it. Note: FS loss implies SFS loss, and SFS loss implies MS loss, but only the "strongest" loss event is logged for a given event.																																																								
<b>Type</b>	RW																																																								
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 2.5%;">31</th><th style="width: 2.5%;">30</th><th style="width: 2.5%;">29</th><th style="width: 2.5%;">28</th><th style="width: 2.5%;">27</th><th style="width: 2.5%;">26</th><th style="width: 2.5%;">25</th><th style="width: 2.5%;">24</th><th style="width: 2.5%;">23</th><th style="width: 2.5%;">22</th><th style="width: 2.5%;">21</th><th style="width: 2.5%;">20</th><th style="width: 2.5%;">19</th><th style="width: 2.5%;">18</th><th style="width: 2.5%;">17</th><th style="width: 2.5%;">16</th><th style="width: 2.5%;">15</th><th style="width: 2.5%;">14</th><th style="width: 2.5%;">13</th><th style="width: 2.5%;">12</th><th style="width: 2.5%;">11</th><th style="width: 2.5%;">10</th><th style="width: 2.5%;">9</th><th style="width: 2.5%;">8</th><th style="width: 2.5%;">7</th><th style="width: 2.5%;">6</th><th style="width: 2.5%;">5</th><th style="width: 2.5%;">4</th><th style="width: 2.5%;">3</th><th style="width: 2.5%;">2</th><th style="width: 2.5%;">1</th><th style="width: 2.5%;">0</th> </tr> </thead> <tbody> <tr> <td colspan="16" style="text-align: left;">RESERVED</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">RECONFIGURED</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">SAW_BUS_RESET</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">RESERVED</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">FOUND_MS</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">LOST_MS</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">LOST_SFS</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">LOST_FS</td> </tr> </tbody> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																RECONFIGURED	SAW_BUS_RESET	RESERVED	FOUND_MS	LOST_MS	LOST_SFS	LOST_FS
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																										
RESERVED																RECONFIGURED	SAW_BUS_RESET	RESERVED	FOUND_MS	LOST_MS	LOST_SFS	LOST_FS																																			

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6	RECONFIGURED	Reconfiguration boundary crossed Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
5	SAW_BUS_RESET	Bus reset sequence detected on the bus: no activity on DATA line (all-zero) for 2 to 4 frames. Clock receiver FSM has returned to Reset state. Should trigger a local component reset (SW sequence). Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
4	RESERVED		R	0
3	FOUND_MS	Message sync was acquired: operational state reached Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
2	LOST_MS	Message Synchronization was lost Interface device class-specific information element (IE) Note: Not asserted in case of FS or SFS loss. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
1	LOST_SFS	Super Frame Synchronization was lost Interface device class-specific information element (IE) Note: Not asserted in case of FS loss. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
0	LOST_FS	Frame Synchronization was lost Interface device class-specific information element (IE) Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0

**Table 23-729. Register Call Summary for Register SLIMBUS\_FL\_INFO**

Serial Low-Power Inter-Chip Media Bus Controller

- Device reset: [0] [1] [2] [3]
- Possible Interrupts: [4] [5] [6] [7] [8]
- Interface Device ISRs: [9]
- Lost frame synchronization (LOST\_FS): [10]
- Lost superframe synchronization (LOST\_SFS): [11]
- Lost message synchronization (LOST\_MS): [12]
- Found message synchronization (FOUND\_MS): [13] [14] [15] [16]
- Data segment overlap (DATA\_SLOT\_OVERLAP): [17]
- Reconfigured (RECONFIGURED): [18]
- SLIMBUS1 Register Summary: [19]
- SLIMBUS2 Register Summary: [20]

**Table 23-730. SLIMBUS\_FL\_CONTROL**

<b>Address Offset</b>	0x0000 0074		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	Table 23-687 to Table 23-688		
<b>Description</b>	Frame Layer control. (Self-cleared bits)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VALIDATE_DCMAP	CLEAR_RECONFIGURATION	BUS_SHUTDOWN	KILL_FS	KILL_SFS	KILL_MS	FORCE_RECONFIGURE	COMPONENT_RESET	BOOT							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	VALIDATE_DCMAP	Validates the mapping of data channels on devices: to be used after modifying the DC*_MAP* fields (DI and PN). Write 0x0: No action Write 0x1: Validates all DCT_MAP / DCR_MAP values	RW	0
7	CLEAR_RECONFIGURATION	Clear all updates cumulated since the last reconfiguration, return reconfiguration fields to currently active value. Self-cleared immediately. Write 0x0: No action Write 0x1: Clear reconfiguration	RW	0

Bits	Field Name	Description	Type	Reset
6	BUS_SHUTDOWN	<p>Sets both the clock source or the clock receiver FSMs back to "Undefined" state at the next reconfiguration boundary.</p> <p>Executed (and bit self-cleared) on next reconfiguration (forced or not); also cleared by a "clear_reconfiguration".</p> <p>Write 0x0: Cancel pending request (not used in normal operation) if any, otherwise no action</p> <p>Write 0x1: Assert request for next reconfiguration</p> <p>Read 0x1: Request pending for next reconfiguration</p> <p>Read 0x0: No request pending for next reconfiguration</p>	RW	0
5	KILL_FS	<p>Force immediate Frame Synchronization loss (implying as well message and superframe sync)</p> <p>When in "Operational", "SeekingMessageSync" or "SeekingSuperFrameSync" states, component shall go to "SeekingFrameSync" and start reaquiring frame sync. No effect otherwise.</p> <p>For debug use.</p> <p>Write 0x0: No action</p> <p>Write 0x1: Force frame sync loss</p> <p>Read 0x0: Always reads 0 because of immediate action.</p>	RW	0
4	KILL_SFS	<p>Force SuperFrame Synchronization loss at next reconfiguration boundary (implying as well message sync loss).</p> <p>When in "Operational" or "SeekingMessageSync" states, component shall go to "SeekingSuperFrameSync" and start reaquiring superframe sync. No effect otherwise.</p> <p>Used by clock receiver component upon imcompletely received reconfiguration sequence.</p> <p>Write 0x0: No action</p> <p>Write 0x1: Schedule superframe sync loss</p> <p>Read 0x1: Superframe sync loss pending</p> <p>Read 0x0: No pending superframe sync loss</p>	RW	0
3	KILL_MS	<p>Force an immediate Message Synchronization loss.</p> <p>When in "Operational" state, component shall go to "SeekingMessageSync" and start reaquiring message sync. No effect otherwise.</p> <p>For debug use.</p> <p>Write 0x0: No action</p> <p>Write 0x1: Force message sync loss</p> <p>Read 0x0: Always reads 0 because of immediate action.</p>	RW	0
2	FORCE_RECONFIGURE	<p>Force a reconfiguration boundary at next superframe boundary, rather than wait for a RECONFIGURE_NOW() message.</p> <p>Write 0x0: Cancel pending request (not used in normal operation) if any, otherwise no action</p> <p>Write 0x1: Assert reconfiguration request for next superframe boundary</p> <p>Read 0x1: Reconfiguration request pending for next superframe boundary</p> <p>Read 0x0: No request pending</p>	RW	0
1	COMPONENT_RESET	<p>Component reset request control: Transitions clock receiver FSM to state "Reset" (from any other state). Immediate action.</p> <p>Set upon reception of the RESET_DEVICE() message by the interface device, which requires a component-level reset.</p> <p>Note that a component-level reset is also required in case of bus reset, but the transition to "Reset" state happens automatically in that case and this bit is not used.</p> <p>Write 0x1: Assert request for next reconfiguration</p>	RW	0

Bits	Field Name	Description	Type	Reset
0	BOOT	Boot the component when not in active framer mode (don't care when in active framer mode). Transitions clock receiver FSM from state "Undefined" to "Reset".  0x0: No action 0x1: Enable clock receiver boot sequence	RW	0

**Table 23-731. Register Call Summary for Register SLIMBUS\_FL\_CONTROL**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMbus Boot and Reset Procedures: \[0\]](#)
- [Clock receiver component boot: \[1\] \[2\]](#)
- [Clock source component boot: \[3\]](#)
- [Component reset: \[4\] \[5\] \[6\] \[7\]](#)
- [Device reset: \[8\]](#)
- [Bus Shutdown: \[9\]](#)
- [SLIMbus Reconfigure Parameters: \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [SLIMbus Component Boot: \[15\] \[16\] \[17\]](#)
- [SLIMBUS1 Register Summary: \[18\]](#)
- [SLIMBUS2 Register Summary: \[19\]](#)

**Table 23-732. SLIMBUS\_FL\_SM**

<b>Address Offset</b>	0x0000 0078		
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Subframe mode (SM: defines control vs. data space partition) control & status SW must write in values : a) to use when booting in active framer mode. b) extracted from NEXT_SUBFRAME_MODE(SM), to use at following reconfiguration boundary. HW-updated upon superframe sync acquisition, when booting in clock receiver mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SM															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000000
4:0	SM	<p>Subframe Mode, which sets Control Space Width (CSW) and Subframe Length (SFL), both expressed in slots. Note that we always have CSW =&lt; SFL since the control is included in the subframe.</p> <p>0xD: CSW=3; SFL=8 (37.5% control)</p> <p>0x15: CSW=6; SFL=8 (75% control)</p> <p>0x1E: reserved. Same effect than 0b00000</p> <p>0x8: CSW=2; SFL=6 (33.3% control)</p> <p>0x5: CSW=1; SFL=8 (12.5% control)</p> <p>0x1B: CSW=12; SFL=32 (37.5% control)</p> <p>0x2: reserved. Same as 0x00</p> <p>0x4: CSW=1; SFL=6 (16.7% control)</p> <p>0x6: CSW=1; SFL=24 (4.17% control)</p> <p>0x1: reserved. Same as 0x05</p> <p>0x1D: CSW=16; SFL=32 (50% control)</p> <p>0x0: CSW=8; SFL=8 (100% control vs. 0% data)</p> <p>0xB: CSW=2; SFL=32 (6.25% control)</p> <p>0x3: reserved. Same as 0x07</p> <p>0x17: CSW=6; SFL=32 (18.8% control)</p> <p>0x11: CSW=4; SFL=8 (50% control)</p> <p>0xA: CSW=2; SFL=24 (8.33% control)</p> <p>0x9: CSW=2; SFL=8 (25% control)</p> <p>0x10: CSW=4; SFL=6 (66.7% control)</p> <p>0x12: CSW=4; SFL=24 (16.7% control)</p> <p>0x13: CSW=4; SFL=32 (12.5% control)</p> <p>0x18: CSW=8; SFL=24 (33.3% control)</p> <p>0x14: reserved. Same as 0x00</p> <p>0xE: CSW=3; SFL=24 (12.5% control)</p> <p>0x16: CSW=6; SFL=24 (25% control)</p> <p>0x1C: CSW=16; SFL=24 (66.7% control)</p> <p>0x7: CSW=1; SFL=32 (3.13% control)</p> <p>0x19: CSW=8; SFL=32 (25% control)</p> <p>0x1F: CSW=24; SFL=32 (75% control)</p> <p>0x1A: CSW=12; SFL=24 (50% control)</p> <p>0xF: CSW=3; SFL=32 (9.38% control)</p> <p>0xC: CSW=3; SFL=6 (50% control)</p>	RW	0x00

**Table 23-733. Register Call Summary for Register SLIMBUS\_FL\_SM**

Serial Low-Power Inter-Chip Media Bus Controller

- [Clock source component boot: \[0\]](#)
- [SLIMbus Reconfigure Parameters: \[1\]](#)
- [SLIMbus Component Boot: \[2\]](#)
- [SLIMBUS1 Register Summary: \[3\]](#)
- [SLIMBUS2 Register Summary: \[4\]](#)

**Table 23-734. SLIMBUS\_FL.CG**

<b>Address Offset</b>	0x0000 007C		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Clock Gear (CG) control and status SW must write in values : a) to use when booting in active framer mode. b) extracted from NEXT_CLOCK_GEAR(CG), to use at following reconfiguration boundary. HW-updated upon superframe sync acquisition, when booting in clock receiver mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CG							

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	CG	Clock Gear (CG) to be used after the next reconfiguration boundary. Root / SLIMbus frequency ratio is defined as $2^{(10-CG)}$ 0x6: Ratio to root frequency: 16 0x1: Ratio to root frequency: 512 0xA: Ratio to root frequency: 1 (undivided) 0x7: Ratio to root frequency: 8 0x3: Ratio to root frequency: 128 0x2: Ratio to root frequency: 256 0x8: Ratio to root frequency: 4 0x9: Ratio to root frequency: 2 0x5: Ratio to root frequency: 32 0x4: Ratio to root frequency: 64	RW	0x0

**Table 23-735. Register Call Summary for Register SLIMBUS\_FL.CG**

Serial Low-Power Inter-Chip Media Bus Controller

- [Clock source component boot: \[0\]](#)
- [SLIMbus Reconfigure Parameters: \[1\]](#)
- [SLIMbus Component Boot: \[2\]](#)
- [SLIMBUS1 Register Summary: \[3\]](#)
- [SLIMBUS2 Register Summary: \[4\]](#)



**Table 23-736. SLIMBUS\_FL\_RF**

<b>Address Offset</b>	0x0000 0080		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Root frequency (RF = SLIMbus clock frequency when in gear 10) control & status SW must write in values : a) to use when booting in active framer mode. b) extracted from NEXT_ROOT_FREQUENCY(RF), to use at following reconfiguration boundary. HW-updated upon superframe sync acquisition, when booting in clock receiver mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RF															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3:0	RF	Root frequency 0x6: 24 MHz 0x1: 24.576 MHz 0x7: 25 MHz 0x0: Root frequency not indicated 0x2: 22.5792 MHz 0x8: 26 MHz 0x9: 27 MHz 0x4: 16.8 MHz 0x5: 19.2 MHz 0x3: 15.36 MHz	RW	0x0

**Table 23-737. Register Call Summary for Register SLIMBUS\_FL\_RF**

Serial Low-Power Inter-Chip Media Bus Controller

- [Clock source component boot: \[0\]](#)
- [SLIMbus Reconfigure Parameters: \[1\]](#)
- [SLIMbus Component Boot: \[2\]](#)
- [SLIMBUS1 Register Summary: \[3\]](#)
- [SLIMBUS2 Register Summary: \[4\]](#)

**Table 23-738. SLIMBUS\_FR\_INFO**

<b>Address Offset</b>	0x0000 0090		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Framer device status (when active). Write 1 to a bit to clear it.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLOCK_RESTART		FRAMER_UNACTIVATED		FRAMER_ACTIVATED		GC_TX_COL		FI_TX_COL		FS_TX_COL					

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5	CLOCK_RESTART	Clock restart request event detected (that is, asynchronous DATA line transition during clock pause) 0x0: no event pending 0x1: event pending	RW W1toClr	0
4	FRAMER_UNACTIVATED	Framer has left clock source operational state. Set on ACTIVE_FRAMER information element deassertion. (framer device class-specific IE) 0x0: no event pending 0x1: event pending	RW W1toClr	0
3	FRAMER_ACTIVATED	Framer has reached clock source operational state. Set on ACTIVE_FRAMER information element assertion. (framer device class-specific IE) 0x0: no event pending 0x1: event pending	RW W1toClr	0
2	GC_TX_COL	Collision during guide byte transmit (Guide Channel) Framer device class-specific information element (IE) 0x0: no event pending 0x1: event pending	RW W1toClr	0
1	FI_TX_COL	Collision during Framing Information transmit (framing channel) Framer device class-specific information element (IE) 0x0: no event pending 0x1: event pending	RW W1toClr	0
0	FS_TX_COL	Collision during Frame Sync symbol transmit (framing channel) Framer device class-specific information element (IE) 0x0: no event pending 0x1: event pending	RW W1toClr	0

**Table 23-739. Register Call Summary for Register SLIMBUS\_FR\_INFO**

Serial Low-Power Inter-Chip Media Bus Controller

- [Wake-up request](#): [0]
- [Power Management](#): [1]
- [Device reset](#): [2] [3] [4]
- [Possible Interrupts](#): [5] [6] [7] [8] [9] [10]
- [Transmit collision in frame sync symbol \(FS\\_TX\\_COL\)](#): [11]
- [Transmit collision in frame sync symbol \(FI\\_TX\\_COL\)](#): [12]
- [Transmit collision in guide channel \(GC\\_TX\\_COL\)](#): [13]
- [Framer device activated \(FRAMER\\_ACTIVATED\)](#): [14]
- [Framer device un-activated \(FRAMER\\_UNACTIVATED\)](#): [15]
- [Restart after clock pause \(CLOCK\\_RESTART\)](#): [16]
- [SLIMBUS1 Register Summary](#): [17]
- [SLIMBUS2 Register Summary](#): [18]

**Table 23-740. SLIMBUS\_FR\_CLOCK\_SOURCE**

<b>Address Offset</b>	0x0000 0094		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Table 23-687 to Table 23-688 Root clock configuration for active framer mode, used at next reconfiguration boundary. Unused when not active framer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLKSEL			CLKDIV												

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00000000
6:4	CLKSEL	Selected clock in clock input vector. Indexed from 0 to FCNT-1 (for FCNT clocks).  0x0: Select input clock #0  0x7: Select input clock #7 (assuming the maximum clock input count of 8)	RW	0x0
3:0	CLKDIV	Root divider ratio, applied on clock input to obtain root clock, to be used at next reconfiguration boundary. Input/ root frequency ratio is defined as $2^{(CLKDIV-5)}$ With CG the SLIMbus clock gear, Input / SLIMbus frequency ratio is therefore $2^{(5+CLKDIV-CG)}$ WARNING: that last ratio has to be 1 or greater, to be implementable by the actual clock divider.  0x6: Ratio is 2  0x1: Ratio is 1/16: Gears 7 to 10 forbidden  0xA: Ratio is 32  0x7: Ratio is 4  0x0: Ratio is 1/32: Gears 6 to 10 forbidden  0x2: Ratio is 1/8: Gears 8 to 10 forbidden  0x8: Ratio is 8  0x9: Ratio is 16  0x4: Ratio is 1/2: Gear 10 forbidden.  0x5: Ratio is 1 (root freq. = input freq.)  0x3: Ratio is 1/4: Gears 9-10 forbidden	RW	0x0

**Table 23-741. Register Call Summary for Register SLIMBUS\_FR\_CLOCK\_SOURCE**

Serial Low-Power Inter-Chip Media Bus Controller

- Clock source component boot: [0] [1]
- SLIMbus Reconfigure Parameters: [2] [3]
- Reconfigured (RECONFIGURED): [4]
- SLIMbus Component Boot: [5] [6]
- SLIMBUS1 Register Summary: [7]
- SLIMBUS2 Register Summary: [8]

**Table 23-742. SLIMBUS\_FR\_CONTROL**

<b>Address Offset</b>	0x0000 0098		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Framer Device control. Unused when not active framer. (Self-cleared bits)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUS_RESET		BOOT													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1	BUS_RESET	<p>Bus reset request control, when framer is already active. Set upon reception of the NEXT_RESET_BUS() message.</p> <p>Transitions clock source FSM to state "StartingClock" (from "Operational").</p> <p>Write 0x0: Cancel pending request (not used in normal operation) if any, otherwise no action</p> <p>Write 0x1: Assert request for next reconfiguration</p> <p>Read 0x1: Request pending for next reconfiguration</p> <p>Read 0x0: No request pending for next reconfiguration</p>	RW	0
0	BOOT	<p>Initiate an active framer (clock source) boot sequence for the component, that is, when component is default active framer.</p> <p>Transitions clock source FSM from state "Undefined" to "CheckingDataLine".</p> <p>Transitions clock receiver FSM from state "Undefined" to "Reset".</p> <p>No effect if bus has already booted.</p> <p>Write 0x0: Cancel pending request (not used in normal operation) if any, otherwise no action</p> <p>Write 0x1: Assert request for next reconfiguration</p> <p>Read 0x1: Request pending for next reconfiguration</p> <p>Read 0x0: No request pending for next reconfiguration</p>	RW	0

**Table 23-743. Register Call Summary for Register SLIMBUS\_FR\_CONTROL**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMbus Boot and Reset Procedures: \[0\]](#)
- [Clock source component boot: \[1\]](#)
- [Bus reset: \[2\] \[3\]](#)
- [Component reset: \[4\]](#)
- [SLIMbus Reconfigure Parameters: \[5\] \[6\]](#)
- [SLIMbus Component Boot: \[7\]](#)
- [SLIMBUS1 Register Summary: \[8\]](#)
- [SLIMBUS2 Register Summary: \[9\]](#)

**Table 23-744. SLIMBUS\_FR\_FRAMER\_HANOVER**

<b>Address Offset</b>	0x0000 009C		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Framer handover control (outgoing if currently active, incoming if currently inactive)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HANDOVER_ENABLE	NCO_NCI														

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12	HANDOVER_ENABLE	Enable framer handover, upon NEXT_ACTIVE_FRAMER(NCi,NCo) reception. Self-cleared upon handover, that is, at next reconfiguration boundary. 0x0: No framer handover 0x1: Framer handover at next configuration boundary.	RW	0
11:0	NCO_NCI	NCo[11:0] / NCi[11:0] for outgoing / incoming framer respectively, as extracted from NEXT_ACTIVE_FRAMER(NCi,NCo) Used on framer handover at next reconfiguration boundary.	RW	0x000

**Table 23-745. Register Call Summary for Register SLIMBUS\_FR\_FRAMER\_HANOVER**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMbus Reconfigure Parameters: \[0\] \[1\] \[2\]](#)
- [SLIMBUS1 Register Summary: \[3\]](#)
- [SLIMBUS2 Register Summary: \[4\]](#)

**Table 23-746. SLIMBUS\_FR\_CLOCK\_PAUSE**

<b>Address Offset</b>	0x0000 00A0		
<b>Physical Address</b>	See	<b>Instance</b>	SLIMBUS1 SLIMBUS2
	<a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>		
<b>Description</b>	Clock pause & restart control. Applied at following reconfiguration boundary, that is, where the clock is paused. Note that programming is identical for an active framer (clock source) and a clock receiver.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RT_HI				RT		CLOCK_PAUSE									

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:3	RT_HI	MSBs of 8-bit SLIMbus parameter. Constant. Read 0x0: Only legal, unreserved value.	R	0x00
2:1	RT	Restart Time (RT) for recovery after clock pause, as extracted from NEXT_PAUSE_CLOCK(RT) LSBs of 8-bit SLIMbus parameter. Unused when not active framer.  0x0: Fast recovery (less than 4 cells) 0x1: Constant-phase recovery (superframe phase preserved) 0x2: Unspecified recovery delay	RW	0x0
0	CLOCK_PAUSE	Control clock pause / restart. Self-cleared upon restart. 0x0: Force clock restart if paused, no effect otherwise. 0x1: Pause clock (clock source) or expect clock pause (clock receiver) at next configuration boundary	RW	0

**Table 23-747. Register Call Summary for Register SLIMBUS\_FR\_CLOCK\_PAUSE**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMbus Reconfigure Parameters: \[0\] \[1\] \[2\]](#)
- [SLIMBUS1 Register Summary: \[3\]](#)
- [SLIMBUS2 Register Summary: \[4\]](#)

**Table 23-748. SLIMBUS\_DEV\_LA\_i**

<b>Address Offset</b>	0x0000 0100 + (0x10 * i)	<b>Index</b>	i = 0 to 6
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Device Logical Address control		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EA_ENABLE	LA_ENABLE	LA													

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	EA_ENABLE	Enables the reception of "long-header" messages to the device's hardwired Enumeration Address (EA). Don't care if LA_ENABLE is 1.  0x0: EA is not valid 0x1: EA is used for message reception	RW	0
8	LA_ENABLE	Enables the reception of "short-header" messages to the device's Logical Address (LA).  0x0: LA is not valid 0x1: LA valid, to be used for message reception	RW	0
7:0	LA	Device Logical Address, assigned by manager device at enumeration (0x00 through 0xEF). The active manager device itself has address 0xFF by default.  0xFF: active manager device address	RW	0x00

**Table 23-749. Register Call Summary for Register SLIMBUS\_DEV\_LA\_i**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device reset](#): [0] [1] [2]
- [Device and Addressing Management](#): [3] [4] [5] [6]
- [TX Message Sequence](#): [7]
- [RX Message Sequence](#): [8]
- [Found message synchronization \(FOUND\\_MS\)](#): [9] [10] [11] [12]
- [Data segment overlap \(DATA\\_SLOT\\_OVERLAP\)](#): [13]
- [Transmit collision in frame sync symbol \(FS\\_TX\\_COL\)](#): [14]
- [Transmit collision in frame sync symbol \(FI\\_TX\\_COL\)](#): [15]
- [Transmit collision in guide channel \(GC\\_TX\\_COL\)](#): [16]
- [Transmit collision in data channel port \(DATA\\_TX\\_COL\)](#): [17]
- [SLIMBUS1 Register Summary](#): [18]
- [SLIMBUS2 Register Summary](#): [19]

**Table 23-750. SLIMBUS\_DEV\_EA\_LO\_i**

<b>Address Offset</b>	0x0000 0104 + (0x10 * i)	<b>Index</b>	i = 0 to 6
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Device's Enumeration Address (EA[47:0]), lower 32 bits. For reference.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EA_PC								EA_DI								EA_IV															

Bits	Field Name	Description	Type	Reset
31:16	EA_PC	Product Code (PC[15:0]), shared by all devices of the component	R	0x0000
15:8	EA_DI	Device Index (DI[15:0]), hardcoded from 0 upwards (0x00,0x01,0x02, etc...) each device of the component. <b>WARNING:</b> value is incorrectly given as 0x00 for all indexes.	R	0x00
7:0	EA_IV	Component's Instance Value (IV[7:0]), shared by all devices of the component	R	0x00

**Table 23-751. Register Call Summary for Register SLIMBUS\_DEV\_EA\_LO\_i**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device and Addressing Management](#): [0] [1]
- [SLIMBUS1 Register Summary](#): [2]
- [SLIMBUS2 Register Summary](#): [3]

**Table 23-752. SLIMBUS\_DEV\_EA\_HI\_i**

<b>Address Offset</b>	0x0000 0108 + (0x10 * i)	<b>Index</b>	i = 0 to 6
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Device's Enumeration Address (EA[47:0]), upper 16 bits. For reference.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EA_MI															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	EA_MI	Manufacturer Index (MI[15:0]), shared by all devices	R	0x0102

**Table 23-753. Register Call Summary for Register SLIMBUS\_DEV\_EA\_HI\_i**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device and Addressing Management: \[0\] \[1\]](#)
- [SLIMBUS1 Register Summary: \[2\]](#)
- [SLIMBUS2 Register Summary: \[3\]](#)

**Table 23-754. SLIMBUS\_DCT\_INFO\_j**

<b>Address Offset</b>	0x0000 0200 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	TX data agent status. Write 1 to a bit to clear it.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LOLEVEL	UNDERFLOW	OVERFLOW	DATA_TX_COL												

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x00000000
3	LOLEVEL	Write level of TX data agent FIFO has gone above DMA threshold (that is, writes required to fill FIFO) Note: Defaults to 0 (even though FIFO is then empty) as it sets to 1 only when read-only status bit DCT_FIFO_STATUS.LOLEVEL transitions from 0 to 1, that is, when threshold is actually crossed downward. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
2	UNDERFLOW	Underflow in TX data agent FIFO. Note: never asserted in pushed or async TPs, by construction. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
1	OVERFLOW	Overflow in TX data agent FIFO. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0



Bits	Field Name	Description	Type	Reset
0	DATA_TX_COL	TX collision in TX data agent. Core information element (IE). Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0

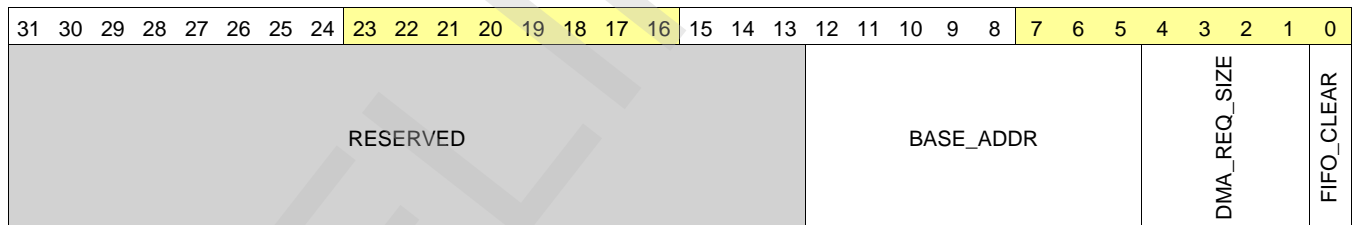
**Table 23-755. Register Call Summary for Register SLIMBUS\_DCT\_INFO\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device reset: \[0\]](#)
- [Possible Interrupts: \[1\] \[2\] \[3\]](#)
- [Data Channel Port ISRs: \[4\] \[5\]](#)
- [Transmit collision in data channel port \(DATA\\_TX\\_COL\): \[6\]](#)
- [Data channel FIFO overflow \(OVERFLOW\): \[7\]](#)
- [5.5.2 Data channel FIFO underflow \(UNDERFLOW\): \[8\]](#)
- [Transmitting data: \[9\]](#)
- [SLIMBUS1 Register Summary: \[10\]](#)
- [SLIMBUS2 Register Summary: \[11\]](#)

**Table 23-756. SLIMBUS\_DCT\_FIFO\_SETUP1\_j**

<b>Address Offset</b>	0x0000 0204 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	TX data agent FIFO setup. To be kept static during channel operation.		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:5	BASE_ADDR	TX data agent FIFO base address within shared TX RAM. (physical address of the 32-bit wide RAM array)	RW	0x00
4:1	DMA_REQ_SIZE	DMA request size minus 1, in write accesses. In counting (packet) mode, last request may be smaller. Only applicable in precise DMA mode.  0xF: 16 accesses per DMA request 0x0: 1 access per DMA request	RW	0x0
0	FIFO_CLEAR	Returns FIFO to initial empty state. Write 0x0: No effect Write 0x1: Clear FIFO  Read 0x1: Ongoing Clear. Completion requires the SLIMbus clock. Read 0x0: No event, last clear completed	RW	0

**Table 23-757. Register Call Summary for Register SLIMBUS\_DCT\_FIFO\_SETUP1\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [FIFO Mapping: \[0\]](#)
- [FIFO Configuration: \[1\] \[2\]](#)
- [SLIMbus Global Initialization: \[3\]](#)
- [SLIMbus DMA configuration: \[4\]](#)
- [SLIMBUS1 Register Summary: \[5\]](#)
- [SLIMBUS2 Register Summary: \[6\]](#)

**Table 23-758. SLIMBUS\_DCT\_FIFO\_SETUP2\_j**

<b>Address Offset</b>	0x0000 0208 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	TX data agent FIFO setup, continued. To be kept static during channel operation.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PACKING	MSB_ALIGNED	RESERVED						DMA_THRESHOLD						SB_THRESHOLD						SIZE											

Bits	Field Name	Description	Type	Reset
31	PACKING	Packing enable control  0x0: Packing disabled: one segment word per write access. Byte enable is don't care.  0x1: Packing enabled: one or several segment words per write access, depending on segment word size and byte enable. Only works in LSB-aligned mode. Behaviour in LSB-aligned mode is undefined.	RW	0
30	MSB_ALIGNED	PLACEHOLDER, NO EFFECT: DATA ALWAYS LSB-aligned LSB/MSB-alignment of FIFO input (write) TX data.  0x0: Input data is LSB-aligned 0x1: Input data is MSB-aligned	RW	0
29:24	RESERVED		R	0x00
23:16	DMA_THRESHOLD	SW-side TX data agent FIFO threshold. DMA write requests get (re-)asserted when WR_LEVEL is above threshold. Range: 0 to SIZE field value (included).  0xFF: DMA writes requested whenever FIFO is empty (assuming maximum FIFO size) 0x0: DMA writes requested whenever FIFO is not full	RW	0x00
15:8	SB_THRESHOLD	SLIMbus-side TX data agent FIFO threshold, used for SLIMbus transmission.  0xFF: Threshold condition = FIFO is full (assuming maximum FIFO size) 0x0: Threshold condition = FIFO is not empty	RW	0x00

Bits	Field Name	Description	Type	Reset
7:0	SIZE	Capacity of FIFO in segment words, minus one. 0xFF: Maximum FIFO size 0x0: FIFO is 1-word deep (not recommended) 0x3: FIFO is 4-word deep. Recommended minimum setting (SLIMbus protocol).	RW	0x00

**Table 23-759. Register Call Summary for Register SLIMBUS\_DCT\_FIFO\_SETUP2\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- FIFO Mapping: [0]
- FIFO Status: [1]
- FIFO Configuration: [2] [3] [4] [5]
- SLIMbus Global Initialization: [6] [7] [8]
- SLIMBUS1 Register Summary: [9]
- SLIMBUS2 Register Summary: [10]

**Table 23-760. SLIMBUS\_DCT\_FIFO\_STATUS\_j**

<b>Address Offset</b>	0x0000 020C + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See Table 23-687 to Table 23-688	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	TX data agent FIFO status, for SW (non-DMA) FIFO management.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT_EN	ACC_CNT															IMPRECISE_EN	RESERVED						WR_LEVEL						LOLEVEL	FULL	

Bits	Field Name	Description	Type	Reset
31	COUNT_EN	Down-counter mode control. Only applicable in precise DMA mode.  Write 0x0: Disable down-counter mode. Should typically be used only when ACC_CNT = 0. If ACC_CNT > 0, DMA request gets deasserted immediately.  Write 0x1: Enable down-counter mode. ACC_CNT is typically set to the desired value in the same write access.  Read 0x1: Down-counter mode is enabled. DMA request can only be active when ACC_CNT > 0  Read 0x0: Down-counter mode is disabled: DMA request follows FIFO level, and is active during no-tx periods.	RW	0

Bits	Field Name	Description	Type	Reset
30:16	ACC_CNT	<p>"Packet mode" down-counter of segment words, decremented on DCT_DATA write accesses. In that mode, DMA request shall deassert when ACC_CNT = 0 (that is, packet fully transmitted / stored in TX FIFO).</p> <p>Write 0x0: Deasserts current DMA request by forcing the count to zero (not typical operation)</p> <p>Write 0x7F: Set largest packet size (in accesses) to transfer before DMA request deassertion. Any lower value is also valid to enter.</p> <p>Read 0x7F: Maximum number of accesses still to be transferred before DMA deassertion. Unless interrupted, counter shall go through all values down to zero.</p> <p>Read 0x0: No access left to transfer, DMA request currently inactive.</p>	RW	0x0000
15	IMPRECISE_EN	<p>Precise vs. Imprecise DMA requesting mode control. (A DMA request is a single assertion-deassertion cycle.)</p> <p>0x0: DMA request contains a precise number of accesses</p> <p>0x1: DMA request is not expected to trigger a fixed number of accesses</p>	RW	0
14:11	RESERVED		R	0x0
10:2	WR_LEVEL	<p>Number of free segment words in FIFO, that could be filled by writes.</p> <p>Read 0x100: FIFO is empty and entirely available for writes (assuming maximum FIFO size)</p> <p>Read 0x0: FIFO is full and cannot be written to</p>	R	0x001
1	LOLEVEL	<p>Indicator of write level (WR_LEVEL: number of free words in FIFO) with respect to (write) threshold (DMA_threshold).</p> <p>Activates DMA write requests when high.</p> <p>Read 0x1: WR_LEVEL &gt; DMA_THRESHOLD</p> <p>Read 0x0: WR_LEVEL &lt;= DMA_THRESHOLD</p>	R	1
0	FULL	<p>FIFO full indicator</p> <p>Read 0x1: no space left in FIFO to write a word (WR_LEVEL = 0)</p> <p>Read 0x0: some space left in FIFO to write at least one word (WR_LEVEL &gt; 0)</p>	R	0

**Table 23-761. Register Call Summary for Register SLIMBUS\_DCT\_FIFO\_STATUS\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [FIFO Status: \[0\] \[1\] \[2\]](#)
- [FIFO Configuration: \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [SLIMBUS1 Register Summary: \[8\]](#)
- [SLIMBUS2 Register Summary: \[9\]](#)

**Table 23-762. SLIMBUS\_DCT\_MAP\_j**

<b>Address Offset</b>	0x0000 0210 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	Associates a device and port to the TX data agent. Write to field FL_CONTROL.validate_dcmmap to validate changes.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PN						DI									

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:8	PN	Port Number (PN), identifies the data channel's port for the device. Shall be unique within a device. Note that a half-duplex (bidirectional) port has 2 data agents with the same PN, 1 TX + 1 Rx.	RW	0x00
7:0	DI	Index (DI) of the device this data agent belongs to. Devices are hard-indexed from 0 upwards.	RW	0x00

**Table 23-763. Register Call Summary for Register SLIMBUS\_DCT\_MAP\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device and Addressing Management: \[0\] \[1\]](#)
- [SLIMBUS1 Register Summary: \[2\]](#)
- [SLIMBUS2 Register Summary: \[3\]](#)

**Table 23-764. SLIMBUS\_DCT\_CONFIG1\_j**

<b>Address Offset</b>	0x0000 0214 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	TX data agent configuration, applied at the next reconfiguration boundary		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE	CTS_LAST_VALUE	RESERVED	DT_LPCM	RESERVED								AF	CL	DL				SL	TP_QUALIFIER	TP											

Bits	Field Name	Description	Type	Reset
31	ENABLE	Data agent enabling control. Auto-cleared upon frame or superframe sync loss. Read 0x1: Data agent is enabled Read 0x0: Data agent is disabled	RW	0

Bits	Field Name	Description	Type	Reset
30	CTS_LAST_VALUE	Last received value of the CTS TAG bit. Only updated for asynchronous TP, when primary owner (that is, transmitter) of the channel.  Read 0x1: Local sink signals TAG.CTS=1 that is, it can receive data (and remote source can transmit some)  Read 0x0: Local sink signals TAG.CTS=0 that is, it cannot receive any data (and remote source shall not transmit any)	R	0
29	RESERVED		R	0
28	DT_LPCM	Unsigned-to-OSAM encoding enable for Data Type (DT) = LPCM audio (0x1)  0x0: DT!=0x1: encoding disabled 0x1: DT=0x1: encoding enabled	RW	0
27:20	RESERVED		R	0x00
19:16	AF	Auxillary Format (AF) used in segment word size calculation. Non-zero values not supported in extended asynchronous TPs.  0xF: User-defined 8-bit AUX (2 AUX slot per segment) 0x0: No AUX information (0 AUX slots per segment) 0x1: ZCUV for tunneling IEC60958 (1 AUX slot per segment) 0xB: User-defined 4-bit AUX (1 AUX slot per segment)	RW	0x0
15	CL	Channel Link (CL) enable: Pairing up of current channel with previous one (that is, channel index below) Tied-0, read-only field for channel index 0, since there is no "previous" channel in that case. TP and segment interval (encoded in SD) must be equal between the linked channels.  0x0: Current channel independent from previous one. 0x1: Current channel paired up with previous one.	RW	0
14:10	DL	Segment Data Length (DL), in 4-bit slots (except when 0). AUX + DATA length shall fit in a segment word, that is, 1 to 32 bits. In extended async TPs, sets segment word size (0 and odd values illegal) instead.  0x6: 6 slots (24 bits) 0x1: 1 slot (4 bits) 0x7: 7 slots (28 bits), requiring at most 1 AUX slot to fit in max-size 32-bit segment word 0x0: Length not indicated: Segment slots that are not TAG or AUX are DATA (that is, no unused slot) 0x2: 2 slots (8 bits) 0x8: 8 slots (32 bits), requiring zero AUX slot, to fit in a max-size 32-bit segment word 0x4: 4 slots (16 bits) 0x5: 5 slots (20 bits) 0x3: 3 slots (12 bits)	RW	0x00
9:5	SL	total Segment Length (SL), in 4-bit slots. SL >= TAG length + AUX length + DATA length  0x0: Reseved value	RW	0x00
4	TP_QUALIFIER	Qualifier for plain/extended half-duplex asynchronous TPs (owner ID). Unused and don't care for other TPs.  0x0: 1/2 duplex TPs: Local device is primary channel owner, uses T1 token  0x1: 1/2 duplex TPs: Local device is secondary channel owner, uses T2 token	RW	0

Bits	Field Name	Description	Type	Reset
3:0	TP	Transport Protocol (TP) Determines also the TAG length (from 0 to 2 slots). 0x6: Extended asynchronous-half-duplex (unicast, 2-slot TAG) 0x1: Pushed (multicast, 1-slot TAG) 0x7: Extended asynchronous-simplex (unicast, 2-slot TAG) 0x0: Isochronous (multicast, no TAG) 0x2: Pulled (unicast, 1-slot TAG) 0x4: Asynchronous-simplex (unicast, 1-slot TAG) 0x5: Asynchronous-half-duplex (unicast, 1-slot TAG) 0xF: User-defined protocol 2 NOT SUPPORTED, RESERVED VALUE 0x3: Locked (multicast, no TAG) NOT SUPPORTED, RESERVED VALUE 0xE: User-defined protocol 1 NOT SUPPORTED, RESERVED VALUE	RW	0x0

**Table 23-765. Register Call Summary for Register SLIMBUS\_DCT\_CONFIG1\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMbus Reconfigure Parameters: \[0\] \[1\] \[2\] \[3\]](#)
- [FIFO Management: \[4\] \[5\] \[6\] \[7\]](#)
- [Data Channel Segment Composition: \[8\] \[9\] \[10\] \[11\]](#)
- [SLIMbus Transfer Protocols: \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
- [Isochronous Transfer Protocol: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)
- [Pushed Transfer Protocol: \[24\] \[25\] \[26\] \[27\] \[28\] \[29\]](#)
- [Pulled Transfer Protocol: \[30\] \[31\] \[32\] \[33\] \[34\] \[35\]](#)
- [Asynchronous Transfer Protocol: \[36\] \[37\] \[38\] \[39\] \[40\] \[41\]](#)
- [Extended Asynchronous Transfer Protocol: \[42\] \[43\] \[44\] \[45\] \[46\] \[47\] \[48\]](#)
- [Paired-up channels: \[49\] \[50\] \[51\] \[52\] \[53\]](#)
- [Lost frame synchronization \(LOST\\_FS\): \[54\]](#)
- [Lost superframe synchronization \(LOST\\_SFS\): \[55\]](#)
- [SLIMbus transfer protocol configuration: \[56\] \[57\] \[58\] \[59\] \[60\] \[61\] \[62\] \[63\] \[64\] \[65\] \[66\] \[67\] \[68\]](#)
- [SLIMBUS1 Register Summary: \[69\]](#)
- [SLIMBUS2 Register Summary: \[70\]](#)

**Table 23-766. SLIMBUS\_DCT\_CONFIG2\_j**

<b>Address Offset</b>	0x0000 0218 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	TX data agent configuration (continued), applied at the next reconfiguration boundary		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SD															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	SD	Segment Distribution	RW	0x000

**Table 23-767. Register Call Summary for Register SLIMBUS\_DCT\_CONFIG2\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMbus Reconfigure Parameters: \[0\]](#)
- [SLIMbus Transfer Protocols: \[1\]](#)
- [Isochronous Transfer Protocol: \[2\]](#)
- [Pushed Transfer Protocol: \[3\]](#)
- [Pulled Transfer Protocol: \[4\]](#)
- [Asynchronous Transfer Protocol: \[5\]](#)
- [Extended Asynchronous Transfer Protocol: \[6\]](#)
- [Paired-up channels: \[7\]](#)
- [SLIMbus transfer protocol configuration: \[8\] \[9\]](#)
- [SLIMBUS1 Register Summary: \[10\]](#)
- [SLIMBUS2 Register Summary: \[11\]](#)

**Table 23-768. SLIMBUS\_DCT\_DATA\_j**

<b>Address Offset</b>	0x0000 021C + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	TX data agent DATA FIFO input. Reads have no effect on FIFO.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data segment word, single-word or packed, LSB or MSB-aligned.	RW	0x0000 0000

**Table 23-769. Register Call Summary for Register SLIMBUS\_DCT\_DATA\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [Data input/output: manual mode: \[0\]](#)
- [Transmitting data: \[1\]](#)
- [SLIMBUS1 Register Summary: \[2\]](#)
- [SLIMBUS2 Register Summary: \[3\]](#)

**Table 23-770. SLIMBUS\_DCR\_INFO\_j**

<b>Address Offset</b>	0x0000 0300 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	RX data agent status. Write 1 to a bit to clear it.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																HILEVEL	UNDERFLOW	OVERFLOW	DATA_TX_COL												



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3	HILEVEL	Read level of RX data agent FIFO has gone above DMA threshold (that is, reads required to empty FIFO) Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
2	UNDERFLOW	Underflow in RX data agent FIFO Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
1	OVERFLOW	Overflow in RX data agent FIFO Note: never asserted in pulled or async TPs, by construction. Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0
0	DATA_TX_COL	TX collision in RX data agent. Core information element (IE). Write 0x0: No action Write 0x1: Clear pending event, if any Read 0x1: IRQ event pending Read 0x0: No event pending	RW W1toClr	0

**Table 23-771. Register Call Summary for Register SLIMBUS\_DCR\_INFO\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device reset](#): [0]
- [Possible Interrupts](#): [1] [2]
- [Data Channel Port ISRs](#): [3] [4]
- [Transmit collision in data channel port \(DATA\\_TX\\_COL\)](#): [5]
- [Data channel FIFO overflow \(OVERFLOW\)](#): [6]
- [5.5.2 Data channel FIFO underflow \(UNDERFLOW\)](#): [7]
- [Receiving data](#): [8]
- [SLIMBUS1 Register Summary](#): [9]
- [SLIMBUS2 Register Summary](#): [10]

**Table 23-772. SLIMBUS\_DCR\_FIFO\_SETUP1\_j**

<b>Address Offset</b>	0x0000 0304 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	RX data agent FIFO setup. To be kept static during channel operation.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE_ADDR								DMA_REQ_SIZE			FIFO_CLEAR				

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00000
12:5	BASE_ADDR	RX data agent FIFO base address within shared RX RAM. (physical address of the 32-bit wide RAM array)	RW	0x00
4:1	DMA_REQ_SIZE	DMA request size minus 1, in read accesses. In counting (packet) mode, last request may be smaller. Only applicable in precise DMA mode. 0xF: 16 accesses per DMA request 0x0: 1 access per DMA request	RW	0x0
0	FIFO_CLEAR	Returns FIFO to initial empty state. Write 0x0: No effect Write 0x1: Clear FIFO Read 0x1: Ongoing Clear. Completion requires the SLIMbus clock. Read 0x0: No event, last clear completed	RW	0

**Table 23-773. Register Call Summary for Register SLIMBUS\_DCR\_FIFO\_SETUP1\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [FIFO Mapping: \[0\]](#)
- [FIFO Configuration: \[1\] \[2\]](#)
- [SLIMbus Global Initialization: \[3\]](#)
- [SLIMbus DMA configuration: \[4\]](#)
- [SLIMBUS1 Register Summary: \[5\]](#)
- [SLIMBUS2 Register Summary: \[6\]](#)

**Table 23-774. SLIMBUS\_DCR\_FIFO\_SETUP2\_j**

<b>Address Offset</b>	0x0000 0308 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	RX data agent FIFO setup, continued. To be kept static during channel operation.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PACKING	MSB_ALIGNED	RESERVED						DMA_THRESHOLD						SB_THRESHOLD						SIZE											

Bits	Field Name	Description	Type	Reset
31	PACKING	Packing enable control 0x0: Packing disabled: one segment word per read access. Byte enable is don't care. 0x1: Packing enabled: one or several segment words per read access, depending on segment word size and byte enable. Only works in LSB-aligned mode. Behaviour in LSB-aligned mode is undefined.	RW	0
30	MSB_ALIGNED	PLACEHOLDER, NO EFFECT: DATA ALWAYS LSB-aligned LSB/MSB-alignment of FIFO output (read) RX data. 0x0: Output data is LSB-aligned 0x1: Output data is MSB-aligned	RW	0
29:24	RESERVED		R	0x00
23:16	DMA_THRESHOLD	SW-side RX data agent FIFO threshold. DMA read requests get (re-)asserted when RD_LEVEL is above threshold. Range: 0 to SIZE field value (included). 0xFF: DMA reads requested whenever FIFO is full (assuming maximum FIFO size) 0x0: DMA reads requested whenever FIFO is not empty	RW	0x00
15:8	SB_THRESHOLD	SLIMbus-side RX data agent FIFO threshold, used for SLIMbus reception. 0xFF: Threshold condition = FIFO is full (assuming maximum FIFO size) 0x0: Threshold condition = FIFO is not empty	RW	0x00
7:0	SIZE	Capacity of FIFO in segment words, minus one. 0xFF: Maximum FIFO size 0x0: FIFO is 1-word deep (not recommended) 0x3: FIFO is 4-word deep. Recommended minimum setting (SLIMbus protocol).	RW	0x00

**Table 23-775. Register Call Summary for Register SLIMBUS\_DCR\_FIFO\_SETUP2\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [FIFO Mapping: \[0\]](#)
- [FIFO Status: \[1\]](#)
- [FIFO Configuration: \[2\] \[3\] \[4\] \[5\]](#)
- [SLIMbus Global Initialization: \[6\] \[7\] \[8\]](#)
- [SLIMBUS1 Register Summary: \[9\]](#)
- [SLIMBUS2 Register Summary: \[10\]](#)

**Table 23-776. SLIMBUS\_DCR\_FIFO\_STATUS\_j**

Address Offset	0x0000 030C + (0x20 * j)	Index	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
Physical Address	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	Instance	SLIMBUS1 SLIMBUS2
Description	RX data agent FIFO status, for SW (non-DMA) FIFO management.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT_EN		ACC_CNT														IMPRECISE_EN	RESERVED						RD_LEVEL						HILEVEL	EMPTY	

Bits	Field Name	Description	Type	Reset
31	COUNT_EN	<p>Down-counter mode control. Only applicable in precise DMA mode.</p> <p>Write 0x0: Disable down-counter mode. Should typically be used only when ACC_CNT = 0. If ACC_CNT &gt; 0, DMA request gets deasserted immediately.</p> <p>Write 0x1: Enable down-counter mode. ACC_CNT is typically set to the desired value in the same write access.</p> <p>Read 0x1: Down-counter mode is enabled: DMA request can only be active when ACC_CNT &gt; 0</p> <p>Read 0x0: Down-counter mode is disabled: DMA request follows FIFO level, and is active during no-transmission periods.</p>	RW	0
30:16	ACC_CNT	<p>"Packet mode" down-counter of segment words, decremented on DCR_DATA read accesses. In that mode, DMA request shall reassert when ACC_CNT = RD_LEVEL (that is, whole packet received).</p> <p>Write 0x0: Deasserts current DMA request by forcing the count to zero (not typical operation)</p> <p>Write 0x7F: Set largest packet size (in accesses) to transfer before DMA request deassertion. Any lower value is also valid to enter.</p> <p>Read 0x7F: Maximum number of accesses still to be transferred before DMA deassertion. Unless interrupted, counter shall go through all values down to zero.</p> <p>Read 0x0: No access left to transfer, DMA request currently inactive.</p>	RW	0x0000
15	IMPRECISE_EN	<p>Precise vs. Imprecise DMA requesting mode control. (A DMA request is a single assertion-deassertion cycle.)</p> <p>0x0: DMA request contains a defined number of accesses</p> <p>0x1: DMA request is not expected to trigger a fixed number of accesses</p>	RW	0
14:11	RESERVED		R	0x0
10:2	RD_LEVEL	<p>Number of segment words stored in FIFO, that can be read out.</p> <p>Read 0x100: FIFO is full of readable words (assuming maximum size)</p> <p>Read 0x0: FIFO contains no readable word</p>	R	0x000
1	HILEVEL	<p>Indicator of read level (RD_LEVEL: number of readable segment words) with respect to (read) threshold (DMA_THRESHOLD).</p> <p>Activates DMA read requests when high.</p> <p>Read 0x1: RD_LEVEL &gt; DMA_THRESHOLD</p> <p>Read 0x0: RD_LEVEL =&lt; DMA_THRESHOLD</p>	R	0

Bits	Field Name	Description	Type	Reset
0	EMPTY	FIFO empty indicator Read 0x1: FIFO contains no readable word (RD_LEVEL = 0) Read 0x0: FIFO contains at least one readable word (RD_LEVEL > 0)	R	1

**Table 23-777. Register Call Summary for Register SLIMBUS\_DCR\_FIFO\_STATUS\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [FIFO Status: \[0\] \[1\] \[2\]](#)
- [FIFO Configuration: \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [SLIMBUS1 Register Summary: \[8\]](#)
- [SLIMBUS2 Register Summary: \[9\]](#)

**Table 23-778. SLIMBUS\_DCR\_MAP\_j**

Address Offset	0x0000 0310 + (0x20 * j)	Index	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
Physical Address	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	Instance	SLIMBUS1 SLIMBUS2
Description	Associates a device and port to the RX data agent. Write to field FL_CONTROL[8] VALIDATE_DCMAP to validate changes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PN						DI									

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:8	PN	Port Number (PN), identifies the data channel's port for the device. Shall be unique within a device. Note that a half-duplex (bidirectional) port has 2 data agents with the same PN, 1 TX + 1 Rx.	RW	0x00
7:0	DI	Index (DI) of the device this data agent belongs to. Devices are hard-indexed from 0 upwards.	RW	0x00

**Table 23-779. Register Call Summary for Register SLIMBUS\_DCR\_MAP\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [Device and Addressing Management: \[0\] \[1\]](#)
- [SLIMBUS1 Register Summary: \[2\]](#)
- [SLIMBUS2 Register Summary: \[3\]](#)

**Table 23-780. SLIMBUS\_DCR\_CONFIG1\_j**

<b>Address Offset</b>	0x0000 0314 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	RX data agent configuration, applied at the next reconfiguration boundary		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE	CTS_FORCE_VALUE	CTS_FORCE_EN	DT_LPCM	RESERVED								AF	CL	DL				SL	TP_QUALIFIER			TP									

Bits	Field Name	Description	Type	Reset
31	ENABLE	Data agent enabling control. Auto-cleared upon frame or superframe sync loss. 0x0: Data agent is disabled 0x1: Data agent is enabled	RW	0
30	CTS_FORCE_VALUE	Value to force the CTS TAG bit value when in asynchronous TP and secondary channel owner (that is, receiver) and CTS is forced (CTS_force_en=1) Don't care in all other cases. Warning: forcing CTS=1 can result in a FIFO overflow. 0x0: Local sink signals TAG.CTS=0 that is, it cannot receive any data (and remote source shall not transmit any) 0x1: Local sink signals TAG.CTS=1 that is, it can receive data (and remote source can transmit some)	RW	0
29	CTS_FORCE_EN	Forces the CTS TAG bit value when in an asynchronous TP and secondary owner (that is, receiver) of the channel. Don't care in other cases. 0x0: CTS bit generated according to RX FIFO level 0x1: CTS bit forced to CTS_force_value	RW	0
28	DT_LPCM	OSAM-to-unsigned decoding enable for Data Type (DT) = LPCM audio (0x1) 0x0: DT!=0x1: decoding disabled 0x1: DT=0x1: decoding enabled	RW	0
27:20	RESERVED		R	0x00
19:16	AF	Auxillary Format (AF) used in segment word size calculation. Non-zero values not supported in extended asynchronous TPs. 0xF: User-defined 8-bit AUX (2 AUX slot per segment) 0x0: No AUX information (0 AUX slots per segment) 0x1: ZCUV for tunneling IEC60958 (1 AUX slot per segment) 0xB: User-defined 4-bit AUX (1 AUX slot per segment)	RW	0x0

Bits	Field Name	Description	Type	Reset
15	CL	<p>Channel Link (CL) enable: Pairing up of current channel with previous one (that is, channel index below)  Channel index 0 should have CL=0, since there is no "previous" channel in that case.  TP and segment interval (encoded in SD) must be equal between the linked channels.</p> <p>0x0: Current channel independent from previous one.  0x1: Current channel paired up with previous one.</p>	RW	0
14:10	DL	<p>Segment Data Length (DL), in 4-bit slots (except when 0). AUX + DATA length shall fit in a segment word, that is, 1 to 32 bits.  In extended async TPs, sets segment word size (0 and odd values illegal) instead.</p> <p>0x6: 6 slots (24 bits)  0x1: 1 slot (4 bits)  0x7: 7 slots (28 bits), requiring at most 1 AUX slot to fit in max-size 32-bit segment word  0x0: Length not indicated: Segment slots that are not TAG or AUX are DATA (that is, no unused slot)  0x2: 2 slots (8 bits)  0x8: 8 slots (32 bits), requiring zero AUX slot, to fit in a max-size 32-bit segment word  0x4: 4 slots (16 bits)  0x5: 5 slots (20 bits)  0x3: 3 slots (12 bits)</p>	RW	0x00
9:5	SL	<p>total Segment Length (SL), in 4-bit slots.  SL &gt;= TAG length + AUX length + DATA length</p> <p>0x0: Reserved value</p>	RW	0x00
4	TP_QUALIFIER	<p>Qualifier for plain/extended half-duplex asynchronous TPs (owner ID).  Unused and don't care for other TPs</p> <p>0x0: 1/2 duplex TPs: Local device is primary channel owner, uses T1 token  0x1: 1/2 duplex TPs: Local device is secondary channel owner, uses T2 token</p>	RW	0
3:0	TP	<p>Transport Protocol (TP)  Determines also the TAG length (from 0 to 2 slots).</p> <p>0x6: Extended asynchronous-half-duplex (unicast, 2-slot TAG)  0x1: Pushed (multicast, 1-slot TAG)  0x7: Extended asynchronous-simplex (unicast, 2-slot TAG)  0x0: Isochronous (multicast, no TAG)  0x2: Pulled (unicast, 1-slot TAG)  0x4: Asynchronous-simplex (unicast, 1-slot TAG)  0x5: Asynchronous-half-duplex (unicast, 1-slot TAG)  0xF: User-defined protocol 2 NOT SUPPORTED, RESERVED VALUE  0x3: Locked (multicast, no TAG) NOT SUPPORTED, RESERVED VALUE  0xE: User-defined protocol 1 NOT SUPPORTED, RESERVED VALUE</p>	RW	0x0

**Table 23-781. Register Call Summary for Register SLIMBUS\_DCR\_CONFIG1\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMbus Reconfigure Parameters: \[0\] \[1\] \[2\] \[3\]](#)
- [FIFO Management: \[4\] \[5\] \[6\] \[7\]](#)
- [Data Channel Segment Composition: \[8\] \[9\] \[10\] \[11\]](#)
- [SLIMbus Transfer Protocols: \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)
- [Isochronous Transfer Protocol: \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\]](#)
- [Pushed Transfer Protocol: \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\]](#)
- [Pulled Transfer Protocol: \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\]](#)
- [Asynchronous Transfer Protocol: \[44\] \[45\] \[46\] \[47\] \[48\] \[49\] \[50\] \[51\]](#)
- [Extended Asynchronous Transfer Protocol: \[52\] \[53\] \[54\] \[55\] \[56\] \[57\] \[58\] \[59\] \[60\]](#)
- [Paired-up channels: \[61\] \[62\] \[63\] \[64\] \[65\]](#)
- [Lost frame synchronization \(LOST\\_FS\): \[66\]](#)
- [Lost superframe synchronization \(LOST\\_SFS\): \[67\]](#)
- [SLIMbus transfer protocol configuration: \[68\] \[69\] \[70\] \[71\] \[72\] \[73\] \[74\] \[75\] \[76\] \[77\] \[78\] \[79\] \[80\]](#)
- [SLIMBUS1 Register Summary: \[81\]](#)
- [SLIMBUS2 Register Summary: \[82\]](#)

**Table 23-782. SLIMBUS\_DCR\_CONFIG2\_j**

<b>Address Offset</b>	0x0000 0318 + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	RX data agent configuration (continued), applied at the next reconfiguration boundary		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SD															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	SD	Segment Distribution	RW	0x000

**Table 23-783. Register Call Summary for Register SLIMBUS\_DCR\_CONFIG2\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [SLIMbus Reconfigure Parameters: \[0\]](#)
- [SLIMbus Transfer Protocols: \[1\]](#)
- [Isochronous Transfer Protocol: \[2\]](#)
- [Pushed Transfer Protocol: \[3\]](#)
- [Pulled Transfer Protocol: \[4\]](#)
- [Asynchronous Transfer Protocol: \[5\]](#)
- [Extended Asynchronous Transfer Protocol: \[6\]](#)
- [Paired-up channels: \[7\]](#)
- [SLIMbus transfer protocol configuration: \[8\] \[9\]](#)
- [SLIMBUS1 Register Summary: \[10\]](#)
- [SLIMBUS2 Register Summary: \[11\]](#)



**Table 23-784. SLIMBUS\_DCR\_DATA\_j**

<b>Address Offset</b>	0x0000 031C + (0x20 * j)	<b>Index</b>	j = 0 to 7 for SLIMBUS1 j = 0 to 3 for SLIMBUS2
<b>Physical Address</b>	See <a href="#">Table 23-687</a> to <a href="#">Table 23-688</a>	<b>Instance</b>	SLIMBUS1 SLIMBUS2
<b>Description</b>	RX data agent DATA FIFO output. Writes have no effect on FIFO.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data segment word, single-word or packed, LSB or MSB-aligned.	R	0x0000 0000

**Table 23-785. Register Call Summary for Register SLIMBUS\_DCR\_DATA\_j**

Serial Low-Power Inter-Chip Media Bus Controller

- [Data input/output: manual mode: \[0\]](#)
- [Receiving data: \[1\]](#)
- [SLIMBUS1 Register Summary: \[2\]](#)
- [SLIMBUS2 Register Summary: \[3\]](#)

## 23.10 MIPI-HSI

This section describes the Mobile Industry Processor Interface (MIPI) high-speed synchronous serial interface (HSI).

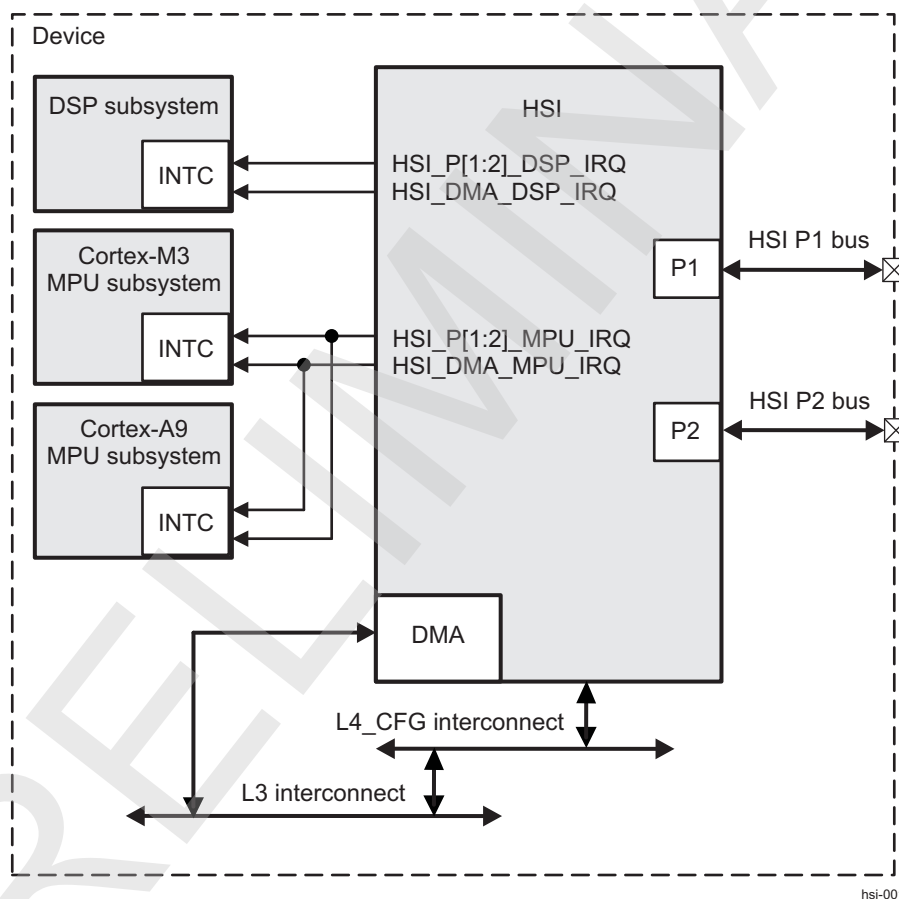
### 23.10.1 HSI Overview

The MIPI high-speed synchronous serial interface (HSI) module is a multichannel and full-duplex serial communications interface, composed of two transmitters (HST) in charge of the transmitted information and two receivers (HSR) in charge of the received information, sharing a 16-channel direct memory access (DMA).

The HSI peripheral is used typically to enable the device to exchange information with an external cellular modem.

Figure 23-220 shows an overview of the HSI module.

**Figure 23-220. HSI Overview**



hsi-001

HSI supports the following features:

- Full duplex
- No static clock matching
- Interface speed up to 192 Mbps on transmitter and up to 225 Mbps on receiver
- Transmission speed and operation mode dynamically configurable
- Logical channels:
  - Up to 16 logical channels on receiver
  - Up to 16 logical channels on transmitter
- One 32-request DMA engine:
  - 16 transmit + 16 receive requests from HSI port

- 16 logical DMA channels
- Interrupt requests:
  - Three to each processor subsystem (two from HSI port, one from DMA)

### 23.10.2 HSI Environment

This section describes the HSI application fields from an environment point of view (external connections).

#### 23.10.2.1 HSI Signals

Table 23-786 describes the HSI module I/O signals.

**Table 23-786. HSI I/O Signals**

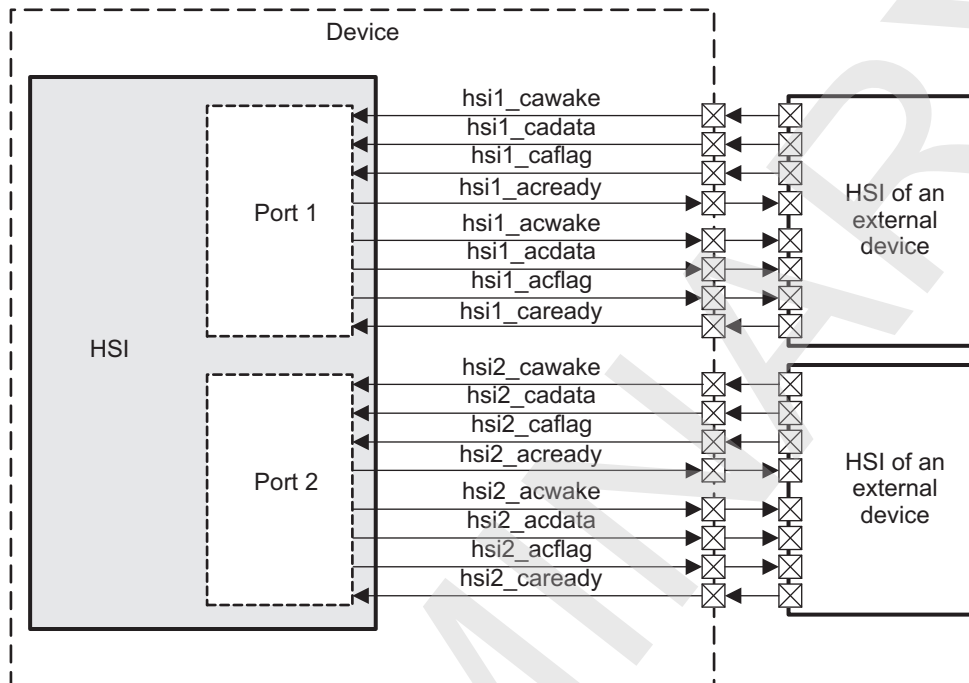
Pin Name	I/O <sup>(1)</sup>	Description	Reset Value
hsi1_cawake	I	HSI port 1 wake-up signal from an external serial transmitter	HiZ
hsi1_cadata	I	HSI port 1 receiver data from an external transmitter	HiZ
hsi1_caflag	I	HSI port 1 receiver flag from an external transmitter	HiZ
hsi1_acready	O	HSI port 1 synchronization signal to an external transmitter	1
hsi1_acwake	O	HSI port 1 wake-up signal to an external serial receiver	0
hsi1_acdata	O	HSI port 1 transmission data (level transmission signaling) to an external receiver	0
hsi1_acflag	O	HSI port 1 transmission flag (bit transition signaling) to an external receiver	1
hsi1_caready	I	HSI port 1 synchronization signal from an external receiver	HiZ
hsi2_cawake	I	HSI port 2 wake-up signal from an external serial transmitter	HiZ
hsi2_cadata	I	HSI port 2 receiver data from an external transmitter	HiZ
hsi2_caflag	I	HSI port 2 receiver flag from an external transmitter	HiZ
hsi2_acready	O	HSI port 2 synchronization signal to an external transmitter	HiZ
hsi2_acwake	O	HSI port 2 wake-up signal to an external serial receiver	HiZ
hsi2_acdata	O	HSI port 2 transmission data (level transmission signaling) to an external receiver	HiZ
hsi2_acflag	O	HSI port 2 transmission flag (bit transition signaling) to an external receiver	HiZ
hsi2_caready	I	HSI port 2 synchronization signal from an external receiver	HiZ

<sup>(1)</sup> I = input; O = output

### 23.10.2.2 HSI Typical Application

Figure 23-221 shows the HSI typical application.

**Figure 23-221. HSI Typical Application**



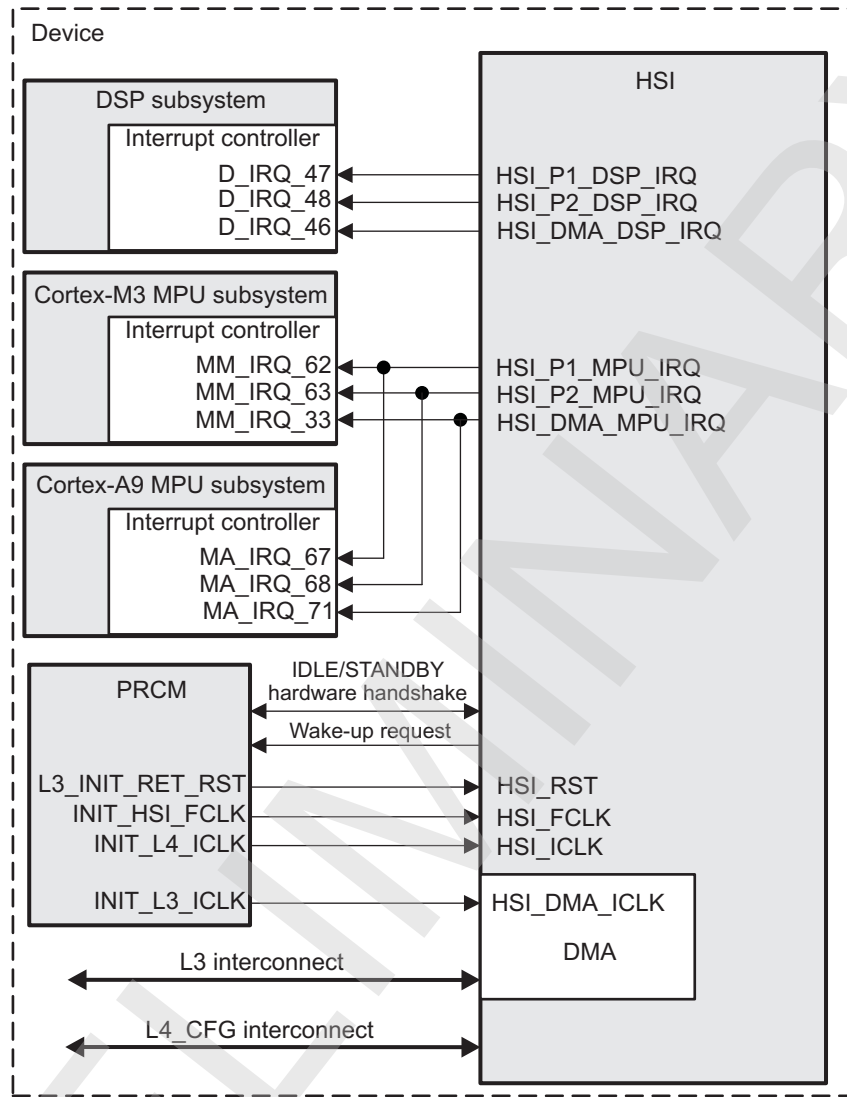
hsi-002

### 23.10.3 HSI Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 23-222 shows the HSI module integration.

Figure 23-222. HSI Integration



hsi-003

**NOTE:** For more information about the IDLE/STANDBY hardware handshake and the wake-up request, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

The following interrupt lines - HSI\_P[1:2]\_MPU\_IRQ and HSI\_DMA\_MPU\_IRQ - are shared between Cortex™-A9 MPU and Cortex™-M3 MPU interrupt controllers (INTCs). It is strongly recommended each interrupt source be unmasked in only one interrupt controller at a time.

[Table 23-787](#) through [Table 23-789](#) summarize the integration of the module in the device.

Table 23-787. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-up Capability	Interconnect
HSI	PD_L3_INIT	NA	L3 L4_CFG

**Table 23-788. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HSI	HSI_FCLK	INIT_HSI_FCLK	PRCM	HSI module functional clock
	HSI_ICLK	INIT_L4_ICLK	PRCM	HSI module interface clock
	HSI_DMA_ICLK	INIT_L3_ICLK	PRCM	HSI DMA engine interface clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HSI	HSI_RST	L3_INIT_RET_RST	PRCM	HSI module asynchronous reset

**Table 23-789. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
HSI	HSI_DMA_MPU_IRQ <sup>(1)</sup>	MM_IRQ_33	Cortex-M3	HSI DMA engine MPU request
	HSI_P1_MPU_IRQ <sup>(1)</sup>	MM_IRQ_62	Cortex-M3	HSI MPU request - Port 1
	HSI_P2_MPU_IRQ <sup>(1)</sup>	MM_IRQ_63	Cortex-M3	HSI MPU request - Port 2
	HSI_DMA_DSP_IRQ	D_IRQ_46	DSP	HSI DMA engine DSP request
	HSI_P1_DSP_IRQ	D_IRQ_47	DSP	HSI DSP request - Port 1
	HSI_P2_DSP_IRQ	D_IRQ_48	DSP	HSI DSP request - Port 2
	HSI_DMA_MPU_IRQ <sup>(1)</sup>	MA_IRQ_71	Cortex-A9	HSI DMA engine MPU request
	HSI_P1_MPU_IRQ <sup>(1)</sup>	MA_IRQ_67	Cortex-A9	HSI MPU request - Port 1
	HSI_P2_MPU_IRQ <sup>(1)</sup>	MA_IRQ_68	Cortex-A9	HSI MPU request - Port 2
No DMA Requests				

<sup>(1)</sup> Interrupts shared between Cortex-A9 MPU INTC and Cortex-M3 MPU INTC

**NOTE:** For a description of interrupt sources, see [Section 23.10.4.5, Interrupt Requests](#)

### 23.10.4 HSI Functional Description

---

**NOTE:** In the functional description section, the following common internal signal names are used for the corresponding external HSI signals (for simplification):

- WAKE: hsi1\_cawake, hsi1\_acwake, hsi2\_cawake, and hsi2\_acwake
  - DATA: hsi1\_cadata, hsi1\_acdata, hsi2\_cadata, and hsi2\_acdata
  - FLAG: hsi1\_caflag, hsi1\_acflag, hsi2\_caflag, and hsi2\_acflag
  - READY: hsi1\_acready, hsi1\_caready, hsi2\_acready, and hsi2\_caready
- 

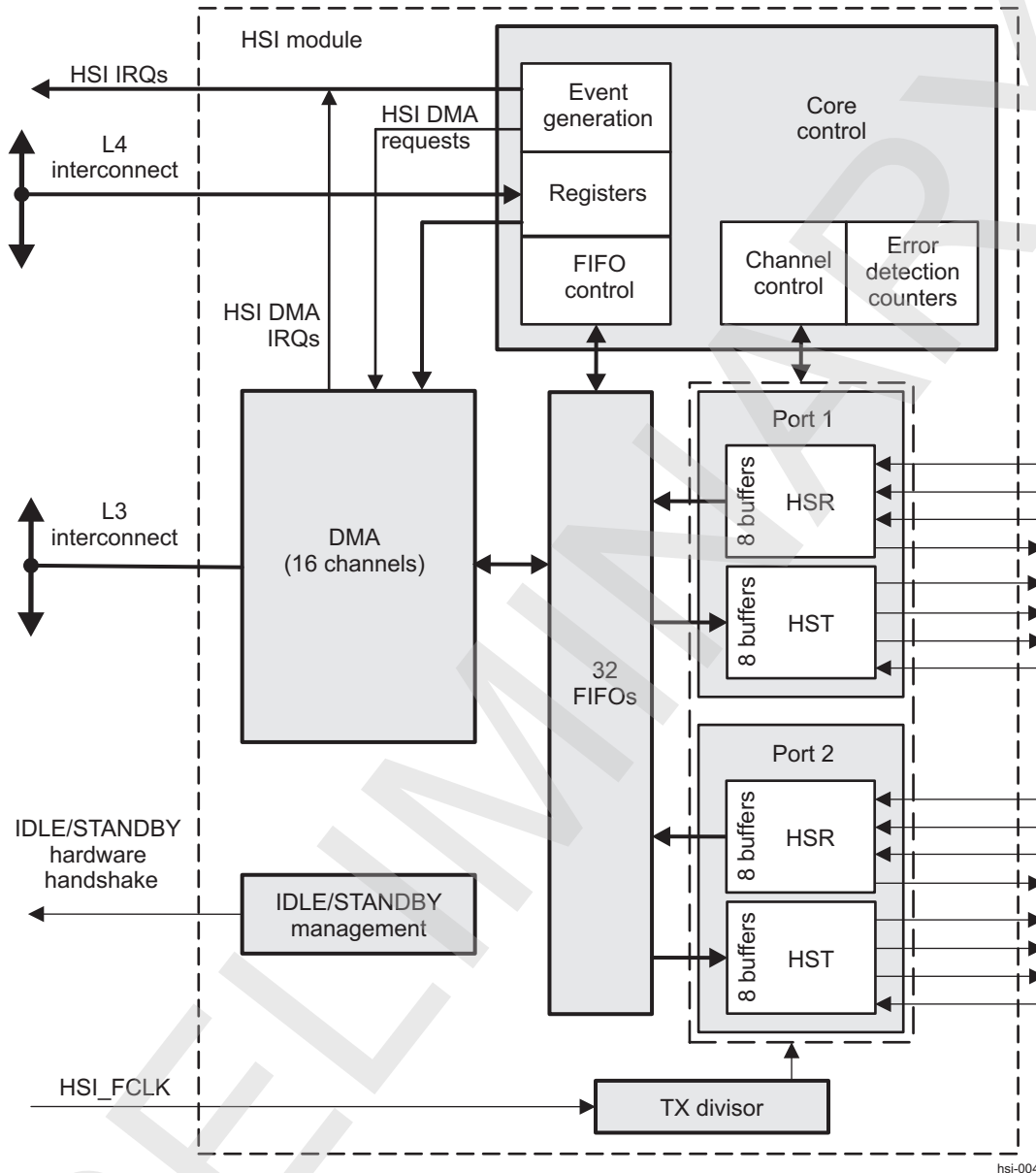
#### 23.10.4.1 Block Diagram

The HSI controller is a subsystem built around two full-duplex multichannel synchronous serial interfaces. It contains a 16-bidirectional-logical-channel DMA in charge of handling data transfers between HSI modules and the L3 interconnect. Also interrupt access to data is available.

**NOTE:** Transmit data cannot be read back and received data cannot be written in DMA and IRQ based transfers.

---

Figure 23-223 shows most HSI module features.

**Figure 23-223. HSI Controller Architecture Overview**

**NOTE:** There is a bypass path in the DMA, which allows a processor (MPU or DSP) to retrieve or provide data received or to be transmitted on an interrupt triggered access, when the DMA is not enabled.

### 23.10.4.2 Clock Configuration

The HSI module has four fully asynchronous clock domains:

- Interface clock domain
- Functional clock domain
- Receive clock domain on port 1
- Receive clock domain on port 2



### 23.10.4.2.1 Interface Clock Domain

The following clocks are part of the interface clock domain:

- HSI\_DMA\_ICLK runs the interface master port (L3). This clock is used for DMA master access to the L3 interconnect and also serves as DMA functional clock.
- HSI\_ICLK runs the interface slave port (L4). This clock is used for configuration of the module and is also used to transmit/receive data through register access in an interrupt based manner.

### 23.10.4.2.2 Functional Clock Domain

The functional clock (HSI\_FCLK) runs the transmitter/receiver logic.

#### 23.10.4.2.2.1 Transmit Clock

The transmit clock is obtained from the functional clock and is used to generate transmitter DATA and FLAG lines, according to the standard. Namely, the DATA line shall always reflect the transmitted bit value, whereas the FLAG line shall provide a transition (0->1, or 1->0) when current DATA value is the same as the previous one.

The transmit data rate is programmable through the [HST\\_DIVISOR\\_Pp\[7:0\]](#) TX\_RATE\_DIV\_VAL bit field.

$$\text{Transmit clock} = \text{HSI\_FCLK} / (\text{TX\_RATE\_DIV\_VAL} + 1).$$

#### 23.10.4.2.2.2 Receive Error Detection Counters Clocks

The functional clock is also used to obtain clocks for Receive Error Counters, which are independently divided versions of the functional clock. Division is set by software through the [HSR\\_DIVISOR\\_Pp\[7:0\]](#) RX\_RATE\_DIV\_VAL bit field, which sets the bit rate clock for frame time-out counter. This clock is further divided to generate an edge per frame to be sent to the frame burst counter.

### 23.10.4.2.3 Receive Clock Domain (Port 1 and Port 2)

The RX clock is retrieved from the DATA and FLAG signals available on the serial interface. A self-clocked source synchronous receiver implementation is done as follows: simple XOR of the DATA and FLAG signals recovers a signal displaying a transition on every bit time. From this signal, two clocks are generated with appropriate phase, which can recover bit values, by sampling the DATA line, where the transmitted bit is always available.

#### 23.10.4.2.3.1 Receiver Clock Recovery

The HSI module uses a delay-locked loop (DLL) to delay the received and recovered clock to produce a delayed version. These clocks are then used to sample the DATA line, not to reconstruct a data flow, but to store received bits on some registers, to load them on receive FIFO as soon as 32 bits (that is, a full word) have arrived.

The DLL module is integrated in the HSR block. The usage of the DLL module is controlled by [HSR\\_DLL\[1\]](#)DLL\_NOT\_BYPASSED bit field. The configurable delay is set through a 10-bit master\_slave\_code value, programmed through the [HSR\\_DLL\[17:8\]](#) INCO bit field. The actual value is:

$$\text{master\_slave\_code} = \text{requested delay} / \text{fine\_delay\_step}.$$

---

**NOTE:** If there is a need to utilize its functionality, care shall be taken while reconfiguring its delay value to prevent invalid clock edges in the RX

---

Software may need to update master\_slave\_code when different OPP are run. In this case, the delay suffers significant variation, and the programming code may need to change, to get the same delay as before. To perform such operation, before operating performance point (OPP) can be changed, software must write 1 on the [HSR\\_DLL\[0\]](#) COCHRE (Code CHange REquest) bit field. Receiver ends the reception of the current frame and drops READY to 0 to prevent transmission of more frames. In case of pipelined

flow, this can take more than one frame time. When READY is 0, the hardware writes 1 on [HSR\\_DLL\[4\] COCHGR](#) (COde CHange GRant) and only at this point the software can program DLL with a the new master\_slave\_code, written in [HSR\\_DLL\[17:8\] INCO](#) (INput COde) at the end of operations, then the [HSR\\_DLL\[0\] COCHRE](#) and [HSR\\_DLL\[4\] COCHGR](#) bit fields are set again to 0. [HSR\\_DLL\[0\] COCHRE](#) are cleared by software, and at this point hardware clears the [HSR\\_DLL\[4\] COCHGR](#) bit.

### 23.10.4.3 Software Reset

The HSI module can be reset by software through the [HSI\\_SYSCONFIG\[1\] SOFTRESET](#) bit. Setting this bit to 1 enables an active global software reset that is functionally equivalent to a hardware reset. The [HSI\\_SYSSTATUS\[0\] RESETDONE](#) bit indicates that the software reset is complete when its value is set to 1. The software must ensure that the software reset completes before doing HSI operations.

The DMA engine can be independently reset by software through the [DMA\\_GRST\[0\] SWRESET](#) bit. Setting this bit to 1 resets the DMA registers and logic. This bit is automatically cleared when the DMA reset completes.

### 23.10.4.4 Power Management

[Table 23-790](#) describes power-management features available for the HSI module.

#### NOTE:

- For descriptions of IdleMode and StandbyMode features, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**Table 23-790. Local Power-Management Features**

Feature	Registers	Description
Clock auto gating	<a href="#">HSI_SYSCONFIG[0] AUTOIDLE</a>	This bit allows a local power optimization inside the module, by gating the HSI_ICLK clock upon the interface activity.
	<a href="#">DMA_GCR[3] AUTOGATING</a>	This bit allows a local power optimization inside the module, by gating the HSI_DMA_ICLK clock upon the interface activity.
Slave idle modes	<a href="#">HSI_SYSCONFIG[4:3] SIDLEMODE</a>	Force-idle, no-idle, smart-idle, and smart-idle wakeup capable modes are available.
Master standby modes	<a href="#">HSI_SYSCONFIG[13:12] MIDLEMODE</a>	Force-standby, no-standby, smart-standby and smart-standby wakeup modes are available.

#### 23.10.4.4.1 HSI Module Power Management Modes

##### 23.10.4.4.1.1 Standby

When asserted, this signal informs the PRCM module that HSI is quiet and does not generate any traffic on the interconnect.

The HSI module supports four STANDBY handshake modes, selected through the [HSI\\_SYSCONFIG\[13:12\] MIDLEMODE](#) bit field:

- Force-standby (0x0): Selecting this mode asserts the standby signal unconditionally of the HSI activity. Software must ensure that HSI is disabled and does not generate any activity before activating this mode.
- No-standby (0x1): Standby signal is never asserted.
- Smart-standby (0x2): Standby signal is asserted when internal HSI DMA does not generate activity anymore (namely, upon DMA idle acknowledge signal assertion).
- Smart-standby wakeup (0x3): Standby signal is asserted when internal HSI DMA does not generate activity anymore (namely, upon DMA idle acknowledge signal assertion). It can also generate a wake-up request.

**23.10.4.4.1.2 Idle**

The PRCM module asserts an IDLE request signal when it requires the HSI slave port to be idled. Any access to this port after the idle request signal has been asserted generates an error.

The HSI module supports four IDLE handshake modes, selected through the [HSI\\_SYSCONFIG\[4:3\] SIDLEMODE](#) bit field:

- Force-idle (0x0): An idle request is acknowledged unconditionally.
- No-idle (0x1): An idle request is never acknowledged.
- Smart-idle (0x2): Acknowledgement to an idle request is given based on the internal HSI module activity.
- Smart-idle Wakeup (0x3): Acknowledgement to an IDLE request is given based on the internal HSI module activity. Additionally, when in this mode, the HSI module is allowed to generate a wake-up request.

**23.10.4.4.1.3 Auto-idle**

The HSI module provides an auto-idle option for the interface clock domain. Auto-idle mode is enabled by writing 1 to [HSI\\_SYSCONFIG\[0\] AUTOIDLE](#) and/or [DMA\\_GCR\[3\] AUTOGATING](#) bits. When this mode is enabled and there is no activity on the interconnect interface, the interface clock is disabled internally to the module to reduce power consumption. When there is new activity on the interconnect interface, the interface clock is restarted without any latency penalty.

**23.10.4.4.1.4 DMA Engine Switch Off Mode**

DMA interface clock (HSI\_DMA\_ICLK) can be switched off by setting the [DMA\\_GCR\[0\] SWITCH\\_OFF](#) bit field to 1. DMA clock activity is then cut off. Software must make sure that ongoing transitions are completed before cutting off clocks. Once the DMA is switched off, only access through configuration port to reset this bit is possible.

**23.10.4.5 Interrupt Requests**

[Table 23-791](#) lists the event flags, and their mask, that can cause module interrupts.

**Table 23-791. Events**

Event Flag	Event Mask	Map to	Description
<a href="#">HSI_Pp_M_IRQrU_STATUS[26]</a> HSR_WAKE	<a href="#">HSI_Pp_M_IRQrU_ENABLE[26]</a> HSR_WAKE	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Wake detected on any channel 8..15.
<a href="#">HSI_Pp_D_IRQrU_STATUS[26]</a> HSR_WAKE	<a href="#">HSI_Pp_D_IRQrU_ENABLE[26]</a> HSR_WAKE	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
<a href="#">HSI_Pp_M_IRQrU_STATUS[25]</a> HSR_BREAK	<a href="#">HSI_Pp_M_IRQrU_ENABLE[25]</a> HSR_BREAK	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Break detected on any channel 8..15.
<a href="#">HSI_Pp_D_IRQrU_STATUS[25]</a> HSR_BREAK	<a href="#">HSI_Pp_D_IRQrU_ENABLE[25]</a> HSR_BREAK	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
<a href="#">HSI_Pp_M_IRQrU_STATUS[24]</a> HSR_ERROR	<a href="#">HSI_Pp_M_IRQrU_ENABLE[24]</a> HSR_ERROR	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Error detected on any channel 8..15
<a href="#">HSI_Pp_D_IRQrU_STATUS[24]</a> HSR_ERROR	<a href="#">HSI_Pp_D_IRQrU_ENABLE[24]</a> HSR_ERROR	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
<a href="#">HSI_Pp_M_IRQrU_STATUS[23:16]</a> HSR_OVERRUN_CHI	<a href="#">HSI_Pp_M_IRQrU_ENABLE[23:16]</a> HSR_OVERRUN_ENI	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Data overrun in real time mode on any channel 8..15
<a href="#">HSI_Pp_D_IRQrU_STATUS[23:16]</a> HSR_OVERRUN_CHI	<a href="#">HSI_Pp_D_IRQrU_ENABLE[23:16]</a> HSR_OVERRUN_ENI	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
<a href="#">HSI_Pp_M_IRQrU_STATUS[15:8]</a> HSR_AVAILABLE_CHI	<a href="#">HSI_Pp_M_IRQrU_ENABLE[15:8]</a> HSR_AVAILABLE_ENI	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Data received on any channel 8..15
<a href="#">HSI_Pp_D_IRQrU_STATUS[15:8]</a> HSR_AVAILABLE_CHI	<a href="#">HSI_Pp_D_IRQrU_ENABLE[15:8]</a> HSR_AVAILABLE_ENI	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	

**Table 23-791. Events (continued)**

Event Flag	Event Mask	Map to	Description
HSI_Pp_M_IRQrU_STATUS[7:0] HST_ACCEPTED_CHI	HSI_Pp_M_IRQrU_ENABLE[7:0] HST_ACCEPTED_ENI	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Data transmitted on any channel 8..15
HSI_Pp_D_IRQrU_STATUS[7:0] HST_ACCEPTED_CHI	HSI_Pp_D_IRQrU_ENABLE[7:0] HST_ACCEPTED_ENI	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
HSI_Pp_M_IRQr_STATUS[26] HSR_WAKE	HSI_Pp_M_IRQr_ENABLE[26] HSR_WAKE	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Wake detected on any channel 0...7
HSI_Pp_D_IRQr_STATUS[26] HSR_WAKE	HSI_Pp_D_IRQr_ENABLE[26] HSR_WAKE	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
HSI_Pp_M_IRQr_STATUS[25] HSR_BREAK	HSI_Pp_M_IRQr_ENABLE[25] HSR_BREAK	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Break detected on any channel 0...7
HSI_Pp_D_IRQr_STATUS[25] HSR_BREAK	HSI_Pp_D_IRQr_ENABLE[25] HSR_BREAK	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
HSI_Pp_M_IRQr_STATUS[24] HSR_ERROR	HSI_Pp_M_IRQr_ENABLE[24] HSR_ERROR	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Error detected on any channel 0...7
HSI_Pp_D_IRQr_STATUS[24] HSR_ERROR	HSI_Pp_D_IRQr_ENABLE[24] HSR_ERROR	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
HSI_Pp_M_IRQr_STATUS[23:16] HSR_OVERRUN_CHI	HSI_Pp_M_IRQr_ENABLE[23:16] HSR_OVERRUN_ENI	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Data overrun in real time mode on any channel 0...7
HSI_Pp_D_IRQr_STATUS[23:16] HSR_OVERRUN_CHI	HSI_Pp_D_IRQr_ENABLE[23:16] HSR_OVERRUN_ENI	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
HSI_Pp_M_IRQr_STATUS[15:8] HSR_AVAILABLE_CHI	HSI_Pp_M_IRQr_ENABLE[15:8] HSR_AVAILABLE_ENI	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Data received on any channel 0...7
HSI_Pp_D_IRQr_STATUS[15:8] HSR_AVAILABLE_CHI	HSI_Pp_D_IRQr_ENABLE[15:8] HSR_AVAILABLE_ENI	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
HSI_Pp_M_IRQr_STATUS[7:0] HST_ACCEPTED_CHI	HSI_Pp_M_IRQr_ENABLE[7:0] HST_ACCEPTED_ENI	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Data transmitted on any channel 0...7
HSI_Pp_D_IRQr_STATUS[7:0] HST_ACCEPTED_CHI	HSI_Pp_D_IRQr_ENABLE[7:0] HST_ACCEPTED_ENI	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
HSI_DMA_M_IRQSTATUS[15:0] DMA_CHI	HSI_DMA_M_IRQENABLE[15:0] DMA_EN_CHI	HSI_P1_MPU_IRQ HSI_P2_MPU_IRQ	Channel 0...15 status
HSI_DMA_D_IRQSTATUS[15:0] DMA_CHI	HSI_DMA_D_IRQENABLE[15:0] DMA_EN_CHI	HSI_P1_DSP_IRQ HSI_P2_DSP_IRQ	
DMA_CSR_CCIR_i[21] BLOCK_IS	DMA_CSR_CCIR_i[5] BLOCK_IE	HSI_DMA_DSP_IRQ HSI_DMA_MPU_IRQ	Block transferred
DMA_CSR_CCIR_i[18] HALF_IS	DMA_CSR_CCIR_i[2] HALF_IE	HSI_DMA_DSP_IRQ HSI_DMA_MPU_IRQ	Half block reached
DMA_CSR_CCIR_i[16] TOUT_IS	DMA_CSR_CCIR_i[0] TOUT_IE	HSI_DMA_DSP_IRQ HSI_DMA_MPU_IRQ	Time-out event

### 23.10.4.5.1 Interrupts

The HSI module outputs six interrupt signals (for more information, see [Section 23.10.3, MIPI HSI Integration](#)). All the interrupts are active-high level-sensitive. Each interrupt line is activated by a set of individual sources. All interrupt sources can individually be enabled and disabled.

For each interrupt, 2 status bits and 2 mask bits are provided, allowing independent masking for the different processors (that is, 1 status bit and 1 mask bit for the MPU [Cortex-A9 or Cortex-M3] interrupts, and 1 status bit and 1 mask bit for the DSP interrupts).

Interrupts in the HSI module are intended to:

- Signal any error condition in reception
- Support the WAKE condition detection
- Support the BREAK condition detection
- Warn of DMA transfer status or problems
- Signal to processors when to read received data and to write data to transmit

DMA has only 16 channels, so 16 RX FIFO and 16 TX FIFO are managed by a mix of DMA driven and interrupt driven read/write policies.

#### **23.10.4.5.1.1 HSI Interrupt Generation**

Five types of interrupt events can be generated by each HSI port:

- Transmit FIFO not full: occurs when TX FIFO is not full, so data can be written for transmission.
- Received data available: occurs when one data frame has been received and is available on the RX FIFO for reading.
- Overrun: occurs when data overrun is detected on HSI port  $p$  channel  $i$
- Error: any error defined by the MIPI HSI protocol activates this source of interrupt if this source is enabled.
- Break condition detected: occurs upon reception of a continuous string of 0s, that exceeds those possible in any FRAME mode frame (data + header). Because FRAME mode always put a 1 before any frame, the maximum number of 0s occurring during normal transmission is clearly 32 plus the base 2 logarithm of channels used. Hardware works out this number based on register information on channels and receive mode, sets the `HSR_BREAK_Pp[0] BREAK_VAL` bit to 1, and issues an interrupt if this source is enabled.

All the events are cleared by software, upon writing 1 on the corresponding register bit. Writing 0 has no effect.

#### **23.10.4.5.1.2 DMA Interrupt Generation**

Three types of interrupt events can be generated by each DMA channel:

- End of Block: Enabled through the `DMA_CSR_CCIR_i[5] BLOCK_IE` bit, it occurs when the last element of the transfer has been written into destination. The `DMA_CEN_i[15:0] SIZE` bit field defines the number of elements in a block.
- Half Block: Enabled through the `DMA_CSR_CCIR_i[2] HALF_IE` bit, it occurs when a half of the elements have been written into destination.
- Time-out error: Enabled through the `DMA_CSR_CCIR_i[0] TOU_IE` bit, it occurs when destination slave answers by an error code.

All interrupt events are generated on the same logical channel dedicated interrupt line. When an interrupt is issued by a logical channel, its `DMA_CSR_CCIR_i` status register is set to record the interrupt cause, if the interrupt event bit is enabled in the relevant `DMA_CSR_CCIR_i` register. The processor interrupt service routine can read this channel status register to identify the sources of the interrupt. The status bits are automatically cleared after they are read by the processor, unless it is an emulation read.

There are two classes of DMA interrupt events:

- Error events: Errors during the transfer
- Status events: DMA transfer status, such as end of block or half-block

The DMA manages the two classes differently.

For an enabled error event, the following sequence occurs:

1. The appropriate bit in the `DMA_CSR_CCIR_i` register is set.
2. An interrupt is generated.
3. The ports are released.
4. Currently active logical channel is disabled and must be reprogrammed.
5. Other channels are not affected, and a new channel can get over, according to programmed priorities.

For an enabled status event, the following sequence occurs:

1. The appropriate bit in the `DMA_CSR_CCIR_i` register is set.
2. An interrupt is generated.
3. A new interrupt cannot be generated until the status register is read and thereby cleared.

All the events are cleared by software, upon writing 1 on the corresponding interrupt status register (`HSI_DMA_M_IRQSTATUS` or `HSI_DMA_D_IRQSTATUS`). Writing 0 has no effect.

### 23.10.4.6 DMA Module

#### 23.10.4.6.1 DMA Overview

The DMA module transfers data between the HSI interfaces and the device on-chip memories. The DMA module supports 16 logical channels and is a dual-port initiator (peripheral port/memory port) with one slave port for configuration. The memory port provides an interface to the memory space and the peripheral port provides an interface to the HSI interfaces. In the HSI controller, the L4\_CFG port accesses can be routed directly to the HSI interfaces. This is done by setting the DMA in bypass mode to route the DMA slave port to the DMA peripheral port.

DMA features:

- One target port, for configuration
- 16 logical channels:  $n$  for transmission and  $16-n$  for reception
- One initiator port, for transfer to/from memory
- One port to/from LCH memory: this memory holds some registers
- 32-bit data handling only
- Constant addressing mode on peripheral port/FIFO
- Post-increment addressing mode with burst support on memory port
- Software enabling
- Hardware activation of data transfer on peripheral port
- Logical channels interleaving with only few-cycles-latency
- Fixed allocation of memory and peripheral port to a logical channel
- LCH first-come-first-served + fixed priority arbitration
- Low-power operation (hardware control for clock domains activity)
- Software reset

#### 23.10.4.6.2 DMA Memories

DMA channel register memory is managed by the DMA. Only the DMA interfaces this memory. A processor must program the DMA registers included in this memory.

The DMA has an access to 16 dual port FIFOs 8 x 32 bits for reception and 16 dual port FIFOs 8 x 32 bits for transmission. They are provided in the form of two separate memories.

#### 23.10.4.6.3 DMA Basic Operation Outline

DMA write and read operations are helped by tags, to manage possibly disorderly answers. Posted or nonposted operations are possible. In any case, the interconnect always manages them as nonposted. This is consistent with DMA operations, that need to know; (for example, if main memory write was successful). If not, the error response from L3 interconnect slave produces a DMA error and the corresponding interrupt. DMA addressing of FIFO memories will always be on a constant address, corresponding to 1 of the 16 `HST_BUFFER_Pp_CHN_i` (or `HSR_BUFFER_Pp_CHN_i`) registers.

These registers are associated to the 16 FIFOs and the naming reflects only the reset value of the association FIFO - HSI logical channel. So at reset, the FIFO addressed on register `HSR_BUFFER_P2_CHN_4` is associated to the fourth logical channel of the second port. Nevertheless, software can change this mapping and associate the same FIFO (for example, to the seventh channel of the first port). This mapping is performed by software through the `HST_MAPPINGf[4:1] CH_NUMBER` or `HSR_MAPPINGf[4:1] CH_NUMBER` bit fields.

#### 23.10.4.6.4 DMA Configuration

Before performing a DMA transfer, the following DMA registers must be configured:

- `DMA_CCR_CSDP_i` - sets source and destination parameters and channel control
  - `DMA_CCR_CSDP_i[15:14] DST_BURST_EN` - destination burst enable (0x0 and 0x1 - single access; 0x2 - burst 4 x 32 bits; 0x3 - burst 8 x 32 bits);
  - `DMA_CCR_CSDP_i[12:9] DST` - transfer destination (0x8 - transfer to memory port; 0x9 - transfer

- to peripheral port);
- [DMA\\_CCR\\_CSDP\\_i\[8:7\]](#) SRC\_BURST\_EN - source burst enable (0x0 and 0x1 - single access; 0x2 - burst 4 x 32 bits; 0x3 - burst 8 x 32 bits);
  - [DMA\\_CCR\\_CSDP\\_i\[5:2\]](#) SRC - transfer source (0x8 - transfer from memory port; 0x9 - transfer from peripheral port);
  - [DMA\\_CCR\\_CSDP\\_i\[31:30\]](#) DST\_ADD\_MODE - destination addressing mode (0x0 - constant address mode; 0x1 - post-incremented address mode);
  - [DMA\\_CCR\\_CSDP\\_i\[29:28\]](#) SRC\_ADD\_MODE - source addressing mode (0x0 - constant address mode; 0x1 - post-incremented address mode);
  - [DMA\\_CSR\\_CCIR\\_i](#) - enables DMA sources of interrupt events.
  - [DMA\\_CSSA\\_i\[31:0\]](#) ADDR - stores 32 bits of the source address
  - [DMA\\_CDSA\\_i\[31:0\]](#) ADDR - stores 32 bits of the destination address
  - [DMA\\_CEN\\_i\[31:0\]](#) SIZE - defines the number of elements in a block. Maximum number of elements is 65535.

To start a DMA transfer, the DMA channel to be used must be enabled by writing 1 to [DMA\\_CCR\\_CSDP\\_i\[23\]](#) ENABLE bit. Once the DMA transfer is finished, the [DMA\\_CCR\\_CSDP\\_i\[23\]](#) ENABLE bit is automatically reset by hardware.

[DMA\\_CDAC\\_CSAC\\_i\[15:0\]](#) ADDRESS\_CSAC (or [DMA\\_CDAC\\_CSAC\\_i\[31:15\]](#) ADDRESS\_CDAC) bit field can be used to monitor the progress of a DMA transfer on channel source (or destination). It is a snapshot of source (or destination) address generated by channel counter. It is incremented on each access made on channels.

### 23.10.4.7 HSR Module

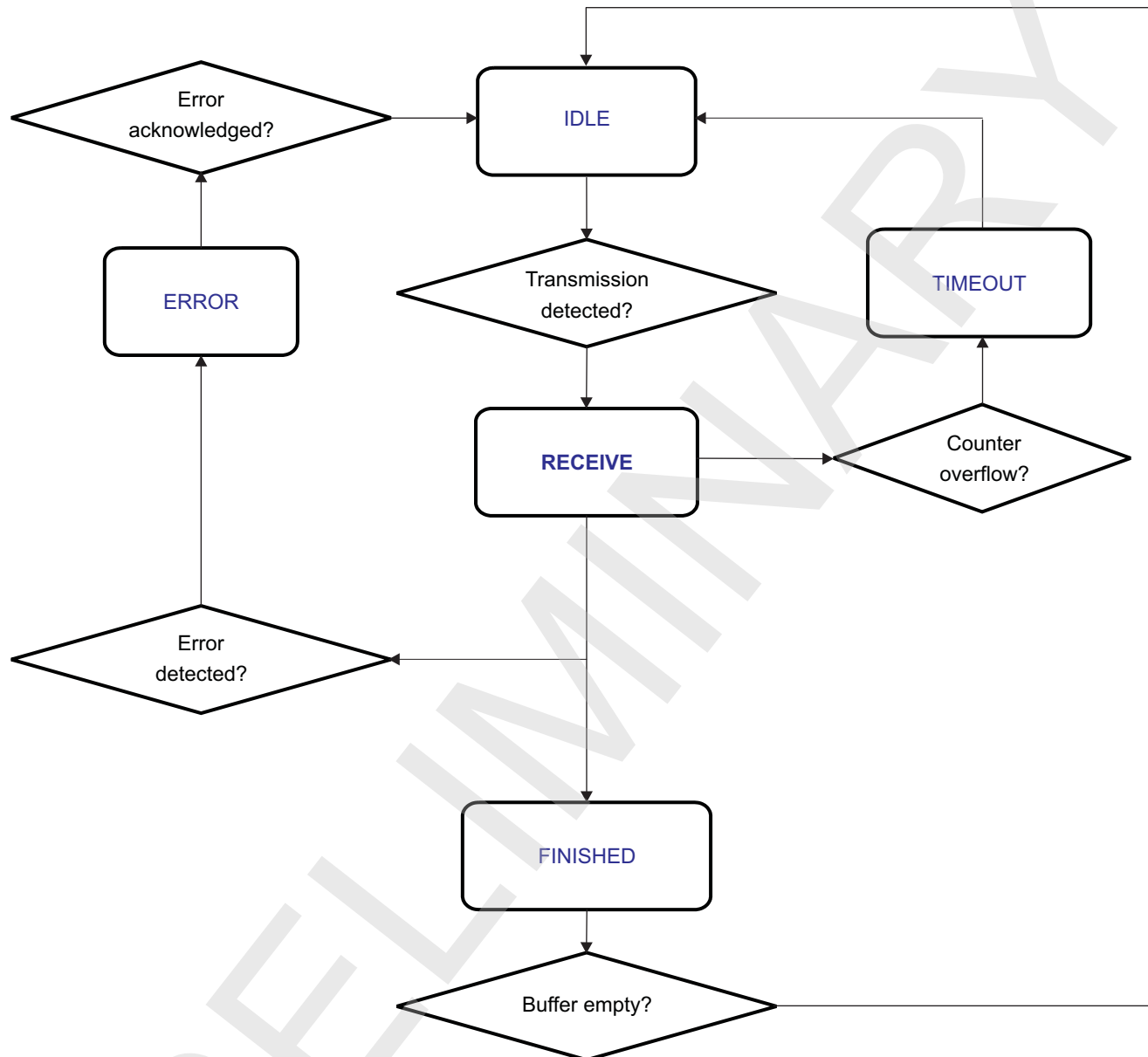
The HSR module is the receiver part of the HSI module. It features 16 channels.

#### 23.10.4.7.1 Receive Data

The value on the DATA line is always the transmitted bit, when sampled by the clock recovered. The double level of buffering allows the data to be safely transferred to the FIFO, while a new frame is stored. Data headers do not belong in FIFO and have to be decoded to load data on the right FIFO (there is one logical FIFO per logical channel).

[Figure 23-224](#) shows the receive state-machine implemented in the HSI module.

Figure 23-224. Receive FSM



hsi-007

### 23.10.4.7.2 Transmission Detection

Reception starts upon two alternate conditions:

- The remote transmitter raises the WAKE line (in this case, it does not actually transmit until the local receiver raises the READY line).
- An activity (in form of transitions) appears on the FLAG/DATA lines, in case the remote transmitter does not implement the WAKE signal. The receiver must then save and store the received data, at least one frame long, before preventing further transmission by setting READY to 0.

### 23.10.4.7.3 Dynamic OPP Changes

If all receive hardware undergoes changes in OPP, operations may have to be stopped, to reprogram a DLL running on different OPP.

A simple software protocol is supported by the [HSR\\_DLL\[4\]](#) COCHGR (change allowed) and [HSR\\_DLL\[0\]](#)



COCHRE (change incoming) bits. When software must change an OPP, it writes 1 on the [HSR\\_DLL\[0\]](#) COCHRE bit. The receiver drops READY to 0 as soon as possible, processes correctly the current frame, and then sets the [HSR\\_DLL\[4\]](#) COCHGR bit. Software could then program the [HSR\\_DLL\[17:8\]](#) INCO bit field or the [HSR\\_DIVISOR\\_Pp\[7:0\]](#) RX\_RATE\_DIV\_VAL bit field, and clear the [HSR\\_DLL\[0\]](#) COCHRE bit when finished. Hardware then clears the [HSR\\_DLL\[4\]](#) COCHGR bit.

### 23.10.4.7.4 Configuration

#### 23.10.4.7.4.1 Mode Register

The [HSR\\_MODE\\_Pp](#) register cannot be changed without stopping the link.

[HSR\\_MODE\\_Pp](#) has the following bit fields:

- [MODE\\_VAL](#) - defines the following operation modes:
  - sleep - specifies whether internal clocks must run and port be operating.
  - stream - receiver does not use start-bits.
  - frame - allows break transmission / reception defined as a longer-than-frame 0 sequence.
- [FLOW\\_VAL](#) - defines the following data flows:
  - Synchronized flow - the receiver drops the READY line every first bit of a received frame, raising it when data is stored.
  - Pipelined flow - the receiver cannot drop the READY line at some times.
  - Real-time flow - the receiver drops the READY at the first received bit and raises it at the last. Data is discarded.

There is one [HST\\_MODE\\_Pp\[3:2\]](#) [FLOW\\_VAL](#) bit field for each active transmit channel and another [HSR\\_MODE\\_Pp\[3:2\]](#) [FLOW\\_VAL](#) bit field for each active receive channel. The local receiver must know its flow mode is (defined by the remote transmitter) to drive the READY line.

#### 23.10.4.7.4.2 Other Registers

- The [HSI\\_Pp\\_SET\\_WAKE\[16\]](#) [HSI\\_3\\_WIRES](#) bit lets hardware know whether WAKE signal is available (when 0x0) or not (when 0x1). Default value is 0x0.
- The [HSR\\_FRAMESIZE\\_Pp](#) register is kept to offer legacy compatibility. It is a read-only register, returning 0x1F.
- The [HSR\\_DIVISOR\\_Pp\[7:0\]](#) [RX\\_RATE\\_DIV\\_VAL](#) bit field sets the clock division ratio for the error detection counters.
- The [HSR\\_CHANNELS\\_Pp\[4:0\]](#) [CHAN\\_NUM\\_VAL](#) bit field sets the number of logical channels. It must be a power of 2. Default value is 0x1. This corresponds to only one logical channel, always the same, and in that case, header is not transmitted. It cannot be changed without stopping the link.

[HSR\\_COUNTERS\\_Pp](#) register sets the value for counters:

- [HSR\\_COUNTERS\\_Pp\[19:0\]](#) FT bit field (frame time-out counter): Maximum number of receive clock cycles to wait for reception of one frame. Afterwards, frame it is considered to have timed out, possibly due to lost synchronization.
- [HSR\\_COUNTERS\\_Pp\[23:20\]](#) TB bit field (tailing bit counter): Maximum number of receive clock cycles to check for possible wrong tailing bit.
- [HSR\\_COUNTERS\\_Pp\[31:24\]](#) FB bit field (frame burst counter): Maximum number of continuous frames to accept to receive in pipelined flow default value for all these values is set to 0, and in that case these checks are disabled.

[HSR\\_MAPPINGf](#) register contains important settings for each FIFO (one register per FIFO):

- [HSR\\_MAPPINGf\[0\]](#) ENABLE bit: FIFO is enabled or not.
- [HSR\\_MAPPINGf\[4:1\]](#) CH\_NUMBER and [HSR\\_MAPPINGf\[7\]](#) PORT\_NUMBER bit fields (channel number and port number): FIFO is assigned to a HSI logical channel on a physical port.
- [HSR\\_MAPPINGf\[13:10\]](#) WORDS: This field holds the number of words to be read (through the L4\_CFG interconnect) upon the interrupt, automatically generated when FIFO pointers are not aligned; therefore, there are data to read. Default value is FIFO enabled and assigned to lower logical channel of each port.

HSR\_DLL register supports a simple software protocol for safe DLL programming:

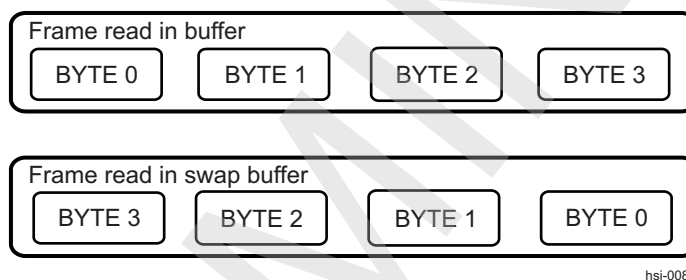
- [HSR\\_DLL\[0\]](#) COCHRE bit: Code Change Request will be set by software when it wants to change DLL code (typically in case of OPP change).
- [HSR\\_DLL\[4\]](#) COCHGR bit: Code Change Grant will be set by hardware when remote side is stopped and no data are incoming.
- [HSR\\_DLL\[17:8\]](#) INCO bit field: Input Code will hold the DLL programming code to be entered into.

#### 23.10.4.7.5 Receive Buffers

For each logical channel, the [HSR\\_BUFFER\\_Pp\\_CHN\\_i\[31:0\]](#) DATA bit field holds one frame. The appropriate bit in the [HSR\\_BUFSTATE\\_Pp](#) register is set to 1 by hardware when the corresponding [HSR\\_BUFFER\\_Pp\\_CHN\\_i](#) register holds one frame. Reading the [HSR\\_BUFFER\\_Pp\\_CHN\\_i\[31:0\]](#) DATA bit field automatically resets the appropriate bit in [HSR\\_BUFSTATE\\_Pp](#). This lets the receiver put another data frame in [HSR\\_BUFFER\\_Pp\\_CHN\\_i](#), setting again [HSR\\_BUFSTATE\\_Pp](#).

Some applications need reversing byte ordering of frames. This is supported by the [HSR\\_SWAPBUFFER\\_Pp\\_CHN\\_i](#) register. By reading this register instead of the corresponding [HSR\\_BUFFER\\_Pp\\_CHN\\_i](#) register on the same channel, byte ordering of the 4 bytes in a frame is reversed.

**Figure 23-225. HSR\_SWAPBUFFER\_Pp\_CHN\_i Register**



Reading MSB of the [HSR\\_SWAPBUFFER\\_Pp\\_CHN\\_i](#) register sets the corresponding bit of [HSR\\_BUFSTATE\\_Pp](#) to empty.

---

**NOTE:** Read access to the register with 0x3 and 0xF byteen removes the receive FIFO-related item. If only 16-bit accesses are used, the upper 2 bytes must be read first (byteen 0xC). Between two consecutive 16-bit FIFO-related accesses there cannot be any other FIFO-related OCP access.

---

#### 23.10.4.7.6 Receive Operations

Logically, the receiver amounts to the receiver shift register, double-buffering and the FSM. Current state of FSM can be observed and even controlled through the [HSR\\_RXSTATE\\_Pp\[2:0\]](#) RXSTATEVAL bit field. For the receive FSM, a functional description is shown in [Figure 23-224](#).

The receiver FSM will be in IDLE state when no transmission is detected, or [HSR\\_MODE\\_Pp\[1:0\]](#) MODE\_VAL bit field = FRAME and no synchronization bit is detected. Otherwise, it stores the first received bit and transitions to RECEIVING state, where it stops until a full frame is received.

Recall that depending on the selected flow control, the READY line may have to go to 0 after the first bit is received.

Upon receiving a full frame, FSM transitions to FINISHED state. Otherwise, FSM transitions to TIMEOUT state as soon as a time-out value (if enabled and programmed in the [HSR\\_COUNTERS\\_Pp\[19:0\]](#) FT bit field is reached. Then it moves to IDLE.

After entering RECEIVED state, FSM will stay there until the received frame has been correctly stored and there is room for another frame. At this point, READY is raised and FSM goes to IDLE. In pipelined flow, flow control by READY is not operating at each byte, and in real time flow it is never operating. In both cases, the FSM transitions right away to IDLE, to receive more data. Illegal transitions or flags rose by error counters (see [Section 23.10.4.11](#), *Error Reporting*), move FSM to ERROR state, and then to HALT, to wait until error is acknowledged by software, warned by an interrupt. Except for pipelined control flow, the READY output is always 1 in IDLE state and 0 anywhere else. In pipelined mode, the READY signal stays to 1, and is cleared by a condition on buffer full. FSM state can be controlled by software by writing in the [HSR\\_RXSTATE\\_Pp\[2:0\]](#) RXSTATEVAL bit field. This can be useful to recover from error condition.

### **23.10.4.7.7 Receive Exceptions**

#### **23.10.4.7.7.1 Break**

Remote transmitter can force synchronization by a long enough string of 0 (equal to the number of header + data bits in the frame + 1). In FRAME mode, this sets the [HSR\\_BREAK\\_Pp\[0\]](#) BREAK\_VAL bit to 1, which in turn sends an interrupt.

#### **23.10.4.7.7.2 Overrun**

Overrun condition is possible during real-time flow, if received data occur when FSM is in FINISHED state. Whenever this happens, the [HSR\\_OVERRUN\\_Pp\[15:0\]](#) OVERRUN\_VAL bit field is set for the corresponding logical channel, which in turn generates an interrupt.

Overrun exception is a legacy condition absolutely needed in previous SSI hardware, where reception without READY signal had to be supported. In HSI we always have READY signal. But it cannot be used during real-time flow and the last 8 bits of a frame during pipelined flow. During real-time data flow, overrun could occur at any time. Main goal of using real-time flow is flushing FIFOs after an error, and in that case overrun interrupt does not seem worth to be raised. Data need not be discarded. The real-time data are discard if receiver cannot follow the transmission rate, otherwise, the real-time data is kept in FIFO.

#### **23.10.4.7.7.3 Errors**

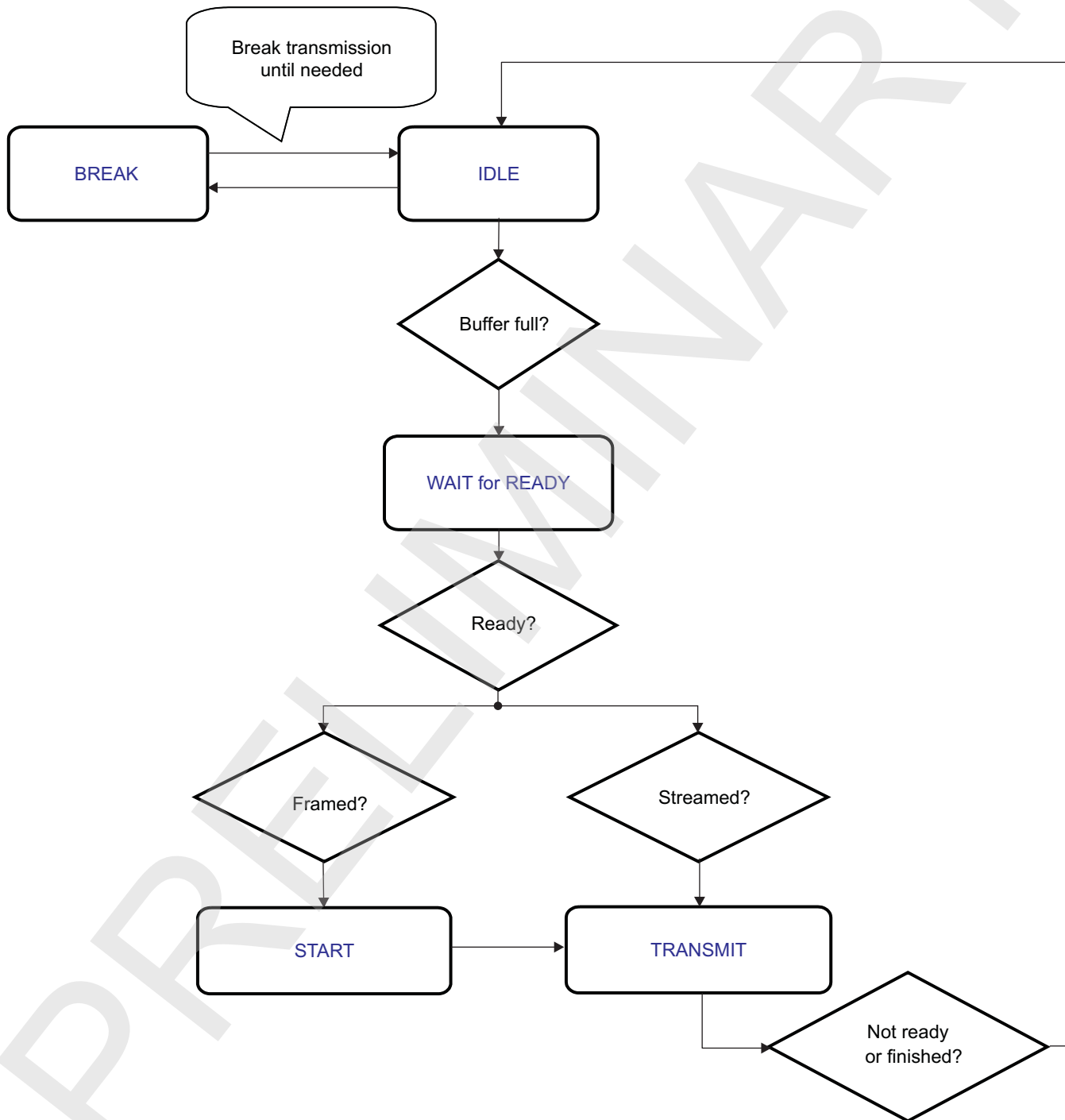
Several sources of errors are possible. In any of them, receiver FSM stays in HALT state until the error interrupt is acknowledged by software; that writes the [HSR\\_ERROR\\_Pp](#) and [HSR\\_ERRORACK\\_Pp](#) registers. Sources of errors include receive FRAME TIMEOUT, other receive errors flagged by error counters, and any other abnormal condition as explained in [Section 23.10.4.11](#), *Error Reporting*.

### 23.10.4.8 HST Block

The HST block is the transmitter part of the HSI module. It features 16 channels.

Figure 23-226 shows the receive state-machine implemented in the HSI module.

**Figure 23-226. Transmit FSM**



hsi-0009

#### 23.10.4.8.1 Configuration

Configuration of the component is expressed in terms of operation mode, number of logical channels, bit rate divisor, and arbitration mode.

### 23.10.4.8.1.1 Mode Register

The `HST_MODE_Pp` register cannot be changed without stopping the link.

There is one `HST_MODE_Pp[3:2]` `FLOW_VAL` bit field for each active transmit channel and another `HSR_MODE_Pp[3:2]` `FLOW_VAL` bit field for each active receive channel. The local transmitter (communicating with the remote receiver) uses the `HST_MODE_Pp[3:2]` `FLOW_VAL` bit field for interpretation of the received `READY` line, for example to signal errors.

The `HST_MODE_Pp` register has the following bit fields:

- `WAKE_CTRL` - selects between software (when 0x0) or automatic (when 0x1) control for the `WAKE` line. Default is software control.
- `MODE_VAL` - defines the following operation modes:
  - sleep - specifies whether internal clocks must run and port be operating.
  - stream - receiver does not use start-bits.
  - frame - allows break transmission / reception defined as a longer-than-frame 0 sequence.
- `FLOW_VAL` - defines the following data flows:
  - Synchronized flow - the `READY` signal from receiver drops after every first frame bit, rising when data are stored, and transmission of a new frame is possible.
  - Pipelined flow - the receiver cannot drop the `READY` line at some times.
  - Real-time flow - the receiver drops the `READY` line at the first received bit and raises it at the last one.

### 23.10.4.8.1.2 Other Registers

- The `HST_FRAME_SIZE_Pp` register is kept to offer legacy compatibility. It is a read-only register, returning 0x1F.
- The `HST_DIVISOR_Pp[7:0]` `RX_RATE_DIV_VAL` bit field sets the transmission bit rate, with 1 transmitted bit per clock cycle. Bit rate can be changed by software at any time.
- The `HST_CHANNELS_Pp[4:0]` `CHAN_NUM_VAL` bit field sets the number of logical channels. It must be a power of 2. Default value is 0x1. This corresponds to only one logical channel, always the same, and in that case, header is not transmitted. It cannot be changed without stopping the link.
- `HST_MAPPINGf` contains important settings for each FIFO (one register per FIFO):
  - `HST_MAPPINGf[0]` `ENABLE` bit: FIFO is enabled or not.
  - `HST_MAPPINGf[4:1]` `CH_NUMBER` and `HSR_MAPPINGf[7]` `PORT_NUMBER` bit fields (channel number and port number): FIFO is assigned to a HSI logical channel on a physical port.
  - `HST_MAPPINGf[13:10]` `THRESHOLD` bit field is the number of empty TX FIFO locations that will activate interrupt automatically.
- The `HST_ARBMODE_Pp[0]` `ARB_VAL` bit allows to choose arbitration policy for time allocation to logical channels, when several channels have to transmit. Round-robin algorithm implements a circular scan: the same channel cannot transmit twice if another channel is ready to transmit. Scan proceeds in the same, unspecified direction (design choice). Fixed priority algorithm grants maximum priority to logical channel 0, minimum priority to channel 15. The right to transmit is assigned to the higher priority channel ready to transmit, independently of previous transmissions.

The choice between algorithms must consider that round-robin sets a maximum wait time for each channel, but it may require a long time to transmit long sequences of frames, on any channel. On the other hand, fixed priority allows to limiting transmission time for high priority channels, while low priority channel might be waiting for a long time.

It is important to understand that this register sets the allocation of one port resource (transmission lines) between 2 up to 16 vying transmit FIFO. Allocation of the 16 transmit FIFO to 32 logical channels is done by the `HST_MAPPINGf` register. For this second kind of allocation, no arbitration makes sense, because software is in charge to ensure against more than 16 logical channels transmitting at the same time.

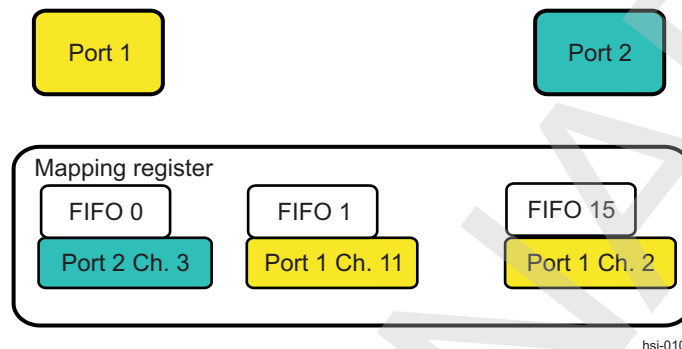
### 23.10.4.8.2 Mapping

Figure 23-227 shows an example of programming the `HST_MAPPINGf` register. No limitation is set on the choice of channel/FIFO/port association.

Each port has its [HST\\_ARBMODE\\_Pp](#) register, so that it is also possible to program alternate algorithms on ports.

While the [HST\\_MAPPINGf](#) register only cares about FIFO, the [HST\\_ARBMODE\\_Pp](#) register is associated to a port and each port has got its own one, which only selects a winner between ready-to-transmit channels of this port. For example, suppose that in [Figure 23-227](#), only the three represented channels need to transmit; then a "yellow" (port 1) arbiter will choose between channel 11 and channel 2, while the "cyan" (port 2) arbiter will have no decision to take and let channel 3 transmit.

**Figure 23-227. MAPPING Register**



#### 23.10.4.8.3 Transmission Buffer

Transmission buffer holds a frame to be transmitted for each channel. The [HST\\_BUFSTATE\\_Pp\[7:0\]](#) [BUFSTATE\\_VAL](#) bit field tells whether [HST\\_BUFFER\\_Pp\\_CHN\\_i\[31:0\]](#) [DATA](#) holds significant data, and it is not be cleared until transmit FSM is in IDLE and arbitration tells the channel is the next in line to transmit. The [HST\\_BUFSTATE\\_Pp](#) register is automatically written by hardware.

---

**NOTE:** Write access to the register with 0x3 and 0xF byteen loads its value into the transmit FIFO. If only 16-bit accesses are used, the upper 2 bytes must be written first (byteen 0xC). Between two consecutive 16-bit FIFO-related accesses there cannot be any other FIFO-related OCP access.

---

Some applications need reversing byte ordering of frames. This is supported by the [HST\\_SWAPBUFFER\\_Pp\\_CHN\\_i](#) register. By reading this register instead of the corresponding [HST\\_BUFFER\\_Pp\\_CHN\\_i](#) register on the same channel, byte ordering of the 4 bytes in a frame is reversed (see [Section 23.10.4.7.5, Receive Buffers](#)).

#### 23.10.4.8.4 Break Pattern Transmission

Break pattern transmission is an alternate option to normal transmission mode, used when trying to recover lost synchronization. It is started by software, by writing the [HST\\_BREAK\\_Pp\[0\]](#) [BREAK\\_VAL](#) bit. As soon as transmit FSM is in IDLE state, (ready to accept another frame for transmission), a nonzero value on the [HST\\_BREAK\\_Pp\[0\]](#) [BREAK\\_VAL](#) bit starts the BREAK sequence, and automatically clears the [HST\\_BREAK\\_Pp\[0\]](#) [BREAK\\_VAL](#) bit. A sequence of 0, longer than a frame is transmitted.

#### 23.10.4.8.5 Transmission Operations

Transmitter adds up to the transmit shift register, loaded from the transmission buffer, and the transmit FSM current state of the FSM can be observed and even controlled through the [HST\\_TXSTATE\\_Pp\[2:0\]](#) [TXSTATEVAL](#) bit field. A functional description of the transmit FSM is shown in [Figure 23-226](#).

After reset, transmitter is in IDLE state. It starts transmission when there is data in one of the transmission [HST\\_BUFFER\\_Pp\\_CHN\\_i](#) registers, depending on the value of [HST\\_BUFSTATE\\_Pp\[7:0\]](#) [BUFSTATE\\_VAL](#) bit field, or when BREAK value is not 0.

When transmitting a data frame (not a BREAK sequence), the frame to be transmitted is clocked from the [HST\\_BUFFER\\_Pp\\_CHN\\_i](#) register to the transmission shift register, and the FSM enters WAIT state, until it gets READY = 1. At this point, it starts driving DATA and FLAG line, either with a preliminary start bit at 1, if in STREAM mode, or with the frame, amounting to header and payload data.

In case of synchronized data flow, TRANSMIT state is left after one frame is transmitted, because READY always drops after the first transmitted bit. Transmit FSM transitions through IDLE and WAIT before starting to transmit again, when READY rises.

In case of pipelined data flow, transmit FSM stays in TRANSMIT state as long as READY = 1, and transmission goes on, always clocking data from the appropriate [HST\\_BUFFER\\_Pp\\_CHN\\_i](#) register, as selected by arbitration. Transmission is stopped by the condition READY = 0, forcing FSM to go into IDLE state. READY is not supposed to switch during the last 8 nominal bit times.

### 23.10.4.8.6 Transmission Exceptions

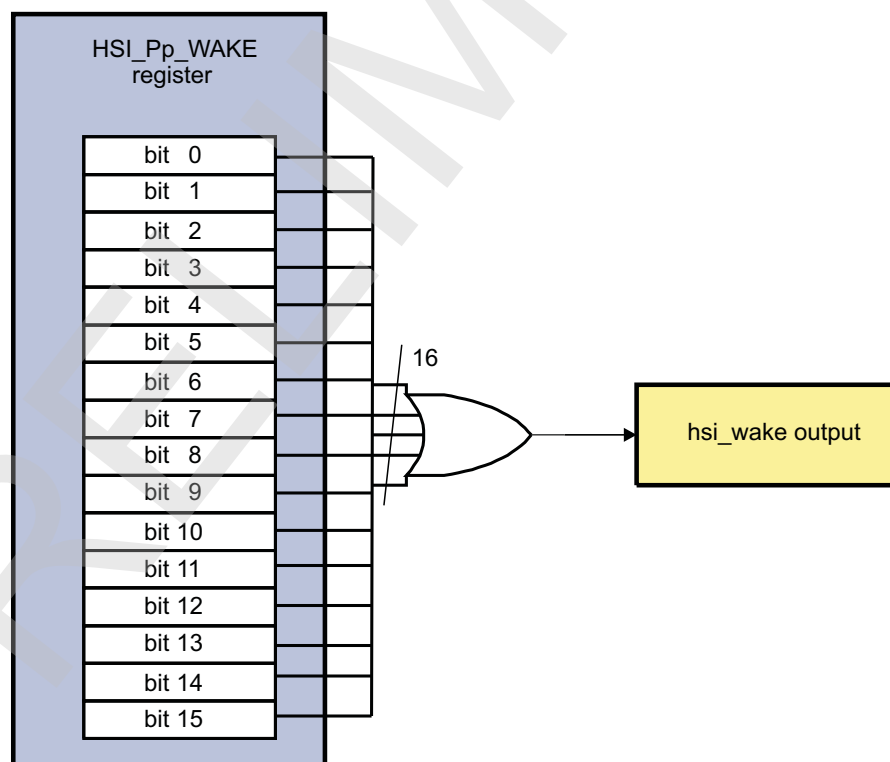
There are no transmission errors. Reception of READY = 0 in the last 8 bits of frames is not flagged. DATA ACCEPTED interrupt is used to tell software that the [HST\\_BUFFER\\_Pp\\_CHN\\_i\[31:0\]](#) DATA bit field is empty and to ask for more data.

### 23.10.4.9 HSI Wake Generator

The WAKE line is used to create a wake-up event for the external module interfacing with HSI before a data transfer through the HSI serial interface.

The 16 bits (1per HSI channel) of the [HSI\\_Pp\\_WAKE\[15:0\]](#) WAKE\_CHI bit field are ORed together to drive the WAKE line, as shown in [Figure 23-228](#).

**Figure 23-228. HSI\_Pp\_WAKE Control in Case of Managed Wake**



hsi-011

The wake generator can be managed by software operations or automatically by hardware. If the wake generator is not managed, the WAKE signal will be driven to 0 and the remote transmitter is in charge of awaking itself on activity detection on DATA/FLAG lines.

### 23.10.4.9.1 Software Management of HSI Wake Generator

Software can get the responsibility to use/manage those bits to create and release wake-up events consistently with HSI use.

Because the [HSI\\_Pp\\_WAKE\[15:0\] WAKE\\_CHI](#) bit field can be managed by different threads/processors, this mechanism allows independent bit management by each thread. Without locking the [HSI\\_Pp\\_WAKE](#) register (like a read-modify-write would do), it avoids one thread to overwrite the register change made by another thread. Each bit of the [HSI\\_Pp\\_SET\\_WAKE\[15:0\] HSI\\_SET\\_WAKE\\_CHI](#) bit field is owned by one thread at a time.

- The thread can set this bit by writing 1 into the corresponding bit of the [HSI\\_Pp\\_SET\\_WAKE\[15:0\] HSI\\_SET\\_WAKE\\_CHI](#) bit field. This sets only this bit, thus leaving the other bits unchanged (no need of a read-modify-write).
- The thread can clear this bit by writing 1 into the corresponding bit of the [HSI\\_Pp\\_CLEAR\\_WAKE\[16:0\] HSI\\_HSI\\_CLEAR\\_WAKE\\_CHI](#) bit field. This clears only this bit, thus leaving the other bits unchanged (no need of a read-modify-write).

### 23.10.4.9.2 Automatic Management of HSI Wake Generator

The wake generator is automatically managed when there is a transmission to be performed. When data are waiting for transmission in a FIFO TX, the hardware sets the corresponding bit of the [HSI\\_Pp\\_WAKE\[15:0\] WAKE\\_CHI](#) bit field, and this raises WAKE signal on port *p*. Then WAKE deassertion is done by the first occurrence of one event between a fixed time-out from end of last transfer, a transition to idle in idle-req protocol, a software clear of register.

### 23.10.4.10 FIFO Subsystem

The FIFO subsystem allows temporary buffering while Receive Data is transferred from Receiver through DMA or the L4 interconnect to main memory, and to store Transmit Data read in main memory through DMA, (or written by L4) before transmitting it. FIFO depth is 8 x 32-bit words for each channel, yielding a total size of 1KB with 16 bidirectional channels.

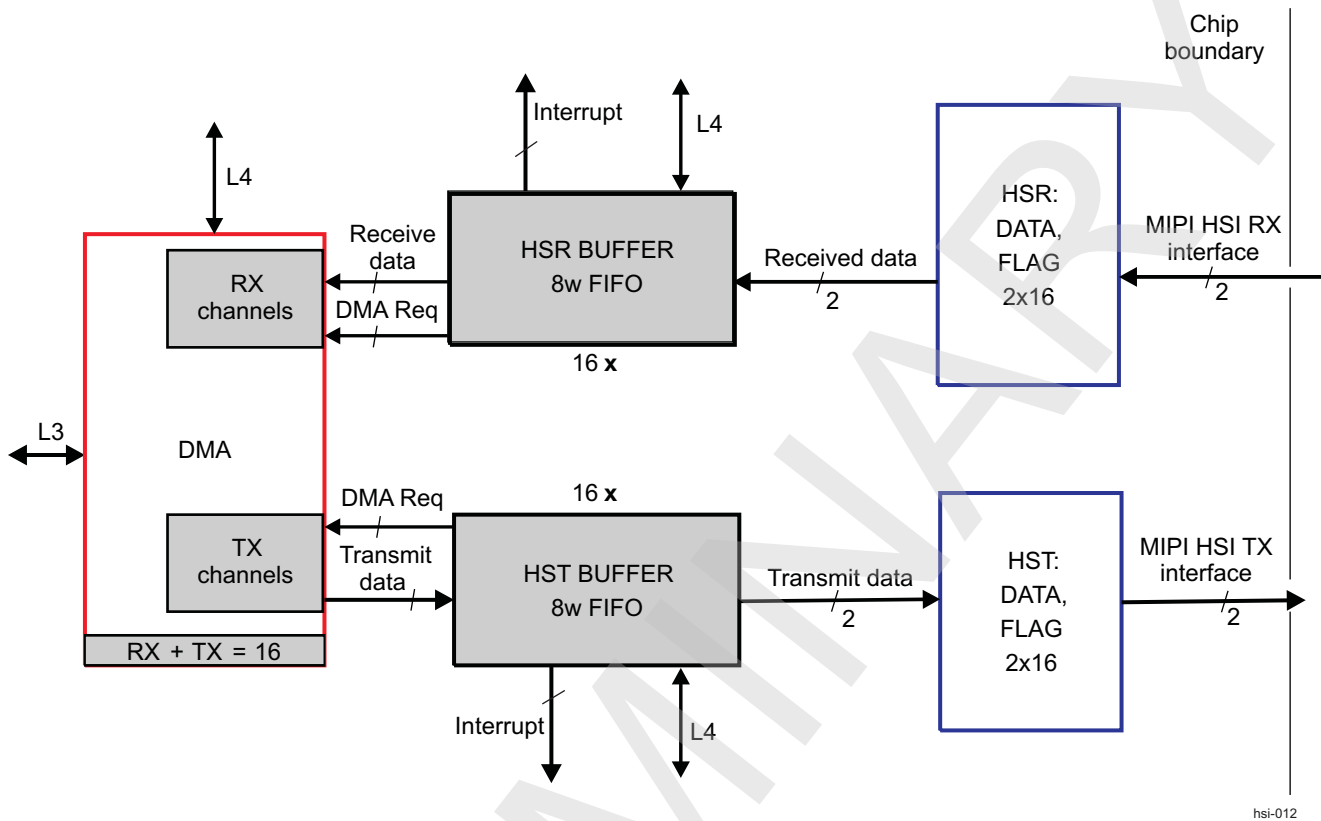
RX FIFOs are clocked by interconnect clock, TX FIFOs are clocked by undivided functional clock.

#### 23.10.4.10.1 FIFO Subsystem Architecture

[Figure 23-229](#) shows the overall FIFO architecture.



Figure 23-229. Overall FIFO Architecture



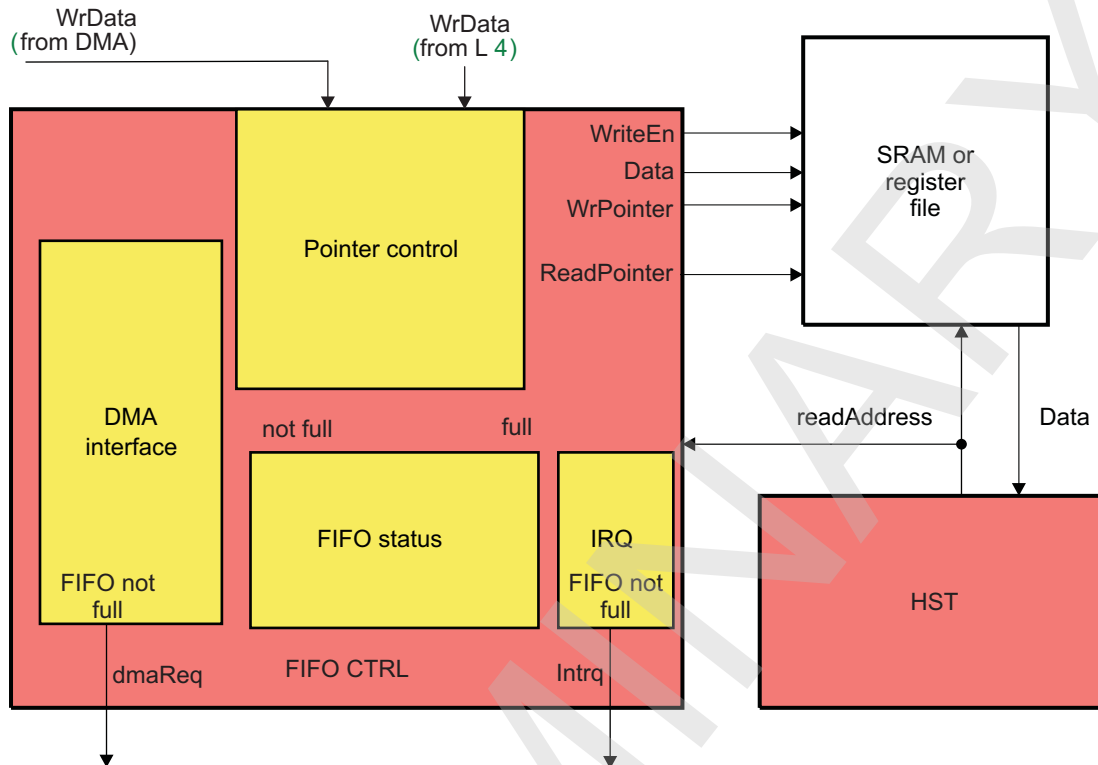
There are 16 total DMA channels available. They can be used for RX or TX in any combination; direction is programmable.

There are 16 RX FIFOs and 16 TX FIFOs. Not all FIFO can be served by DMA, but all RX FIFO (or else TX FIFO) could be served by DMA, if needed. The other FIFOs, if used, must be read/written through the L4\_CFG interconnect.

Both HSI ports provide up to 16 x 2 logical channels. This means that only one half of the HSI logical channels can be accommodated on the existing FIFOs. The others must not be used, and their use has to flag an error. The 16 [HST\\_MAPPINGf](#) or [HSR\\_MAPPINGf](#) registers specify which HSI logical channel number is allowed to be received on RX FIFOs and/or fed for transmission by TX FIFOs. There is one mapping register for transmission and one for reception. Any attempt to transmit or process received data on HSI logical channels not previously mapped on mapping registers results in an error. In other words, some header values (bits 35 down to 32) are not allowed at any given time, and generate an error when received. Software must ensure that the remote side sends payload on the allowed logical channels, specified by header values.

### 23.10.4.10.2 Transmit FIFO Architecture

Figure 23-230 shows the transmission FIFO architecture.

**Figure 23-230. Transmission FIFO Architecture**

hsi-013

FIFO controller flags the condition of "room available in FIFO", or "FIFO not full", at any time when FIFO can host new data for transmission. This condition raises an interrupt request and a DMA request, if enabled. A programmable threshold in the corresponding `HST_MAPPING[13:10] THRESHOLD` bit field reduces the number of requests, ( they might be sent only whether four words can be written in FIFO). Default value for threshold is 0x0 (i.e., request rises as soon as one word can be written in FIFO). See [Table 23-792](#)

**Table 23-792. Threshold for Transmit FIFO Interrupt/DMA Request**

<code>HST_MAPPING[13:10] THRESHOLD</code>	Number of Words in FIFO	Request
0 (legacy default)	Less than 8	Yes
1	Less than 1	Yes
2	Less than 2	Yes
3	Less than 3	Yes
4	Less than 4	Yes
5	Less than 5	Yes
6	Less than 6	Yes
7	Less than 7	Yes
8	Less than 8	Yes
9 to 15	any	No

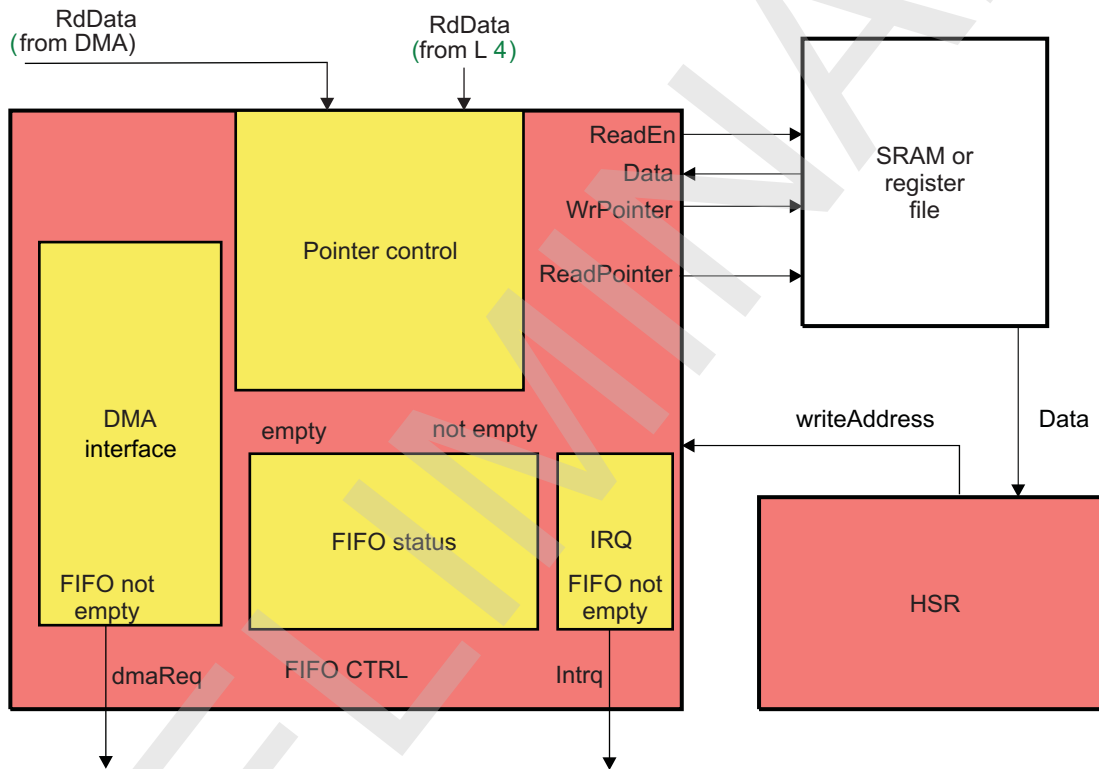
FIFO can be written by DMA and the L4 interconnect configuration port, but the L4 interconnect has fixed priority at any time. Software is in charge of controlling correct access. No hardware mechanism is provided to analyze and block concurrent operation, nor is any hardware control provided on reprogramming DMA registers during a transfer in which they are involved.

The address of FIFO is the corresponding [HST\\_BUFFER\\_Pp\\_CHN\\_i](#) register address. The FIFO controller is in charge of generating actual RAM addresses, namely managing write pointer when DMA or the L4 interconnect access the [HST\\_BUFFER\\_Pp\\_CHN\\_i](#) address, and managing read pointer when HST needs a data to transmit.

**23.10.4.10.3 Receive FIFO Architecture**

Figure 23-231 shows the reception FIFO architecture.

**Figure 23-231. Reception FIFO Architecture**



FIFO controller flags the condition of "unread data in FIFO", or "FIFO not empty", at any time when a FIFO hosts new data, received and not read. This condition can be simply the misalignment of Read Pointer and Write Pointer. The condition raises an interrupt request and a DMA request, if enabled. The [HSR\\_MAPPING\[13:10\] WORDS](#) bit field tells software how many words must be read.

Unlike transmission, it is not possible here to filter requests through a threshold on the number of received words. If such a threshold is bigger than one, and a last single word is received after last read, software would have no clue that more data awaits to be read.

The address of FIFO is the corresponding [HSR\\_BUFFER\\_Pp\\_CHN\\_i](#) register address. The FIFO controller is in charge of generating actual RAM addresses; namely managing write pointer when HSR has received data, and managing read pointer when DMA or the L4 interconnect access the [HSR\\_BUFFER\\_Pp\\_CHN\\_i](#).

#### 23.10.4.10.4 FIFO Addressing

From a software point of view, a FIFO is seen at a single address the corresponding [HST\\_BUFFER\\_Pp\\_CHN\\_i](#) or [HSR\\_BUFFER\\_Pp\\_CHN\\_i](#) register address. The [HST\\_SWAPBUFFER\\_Pp\\_CHN\\_i](#) or [HSR\\_SWAPBUFFER\\_Pp\\_CHN\\_i](#) register address is also provided, to invert data endianness with byte granularity. FIFO overflow in RX is extremely unlikely, because of the huge difference in HSI and L3 interconnect bit rate. However, if this occurs, the hardware prevents any loss of data by deasserting (putting to 0) the READY signal to the remote transmitter. FIFO overflow in TX is not flagged and must be avoided by software drivers.

#### 23.10.4.11 Error Reporting

##### 23.10.4.11.1 Detection of Errors

Error detection on the module is limited to the physical layer. No data corruption is checked, and no error is flagged as long as receiver and transmitter agree on frame boundaries. As both clocking and data is recovered from alternate edges on DATA and FLAG signals, three kinds of error can appear:

- Simultaneous edges
- Missing edges
- Additional edges

##### 23.10.4.11.1.1 Simultaneous Edges

Transmitter is not supposed to produce simultaneous edges on DATA and FLAG signals, but they can show up if transmission rate is far higher than what the receiver can handle. Given to the sampling protocol, this event may introduce errors in sampled data. Detecting this error is not easy, as the recovered clock will be affected as well, and in some configurations no other clock is available. One bit is available in the [HSR\\_ERROR\\_Pp](#) register, to report any hardware abnormal condition in data reception.

##### 23.10.4.11.1.2 Missing Edges

Missing edges result in missing clock cycles, thus causing the receiver to get less data than expected on a given frame.

In synchronized flow, when the transmit side has completed transmission, the receiver side is still waiting for more data, and does not raise READY, thus stopping transmitter from going on with more frames. A frame time-out counter (see [Section 23.10.4.11.2.1, Frame Time-Out Counter](#)) is used to escape this deadlock condition.

In pipelined flow, when the transmit side has completed transmission and has more data to transmit, the receiver accepts and processes this data as if they were part of the previous frame. Frame synchronization is then lost. It may randomly resume when in frame mode, as zeroes at the end of a frame are automatically discarded. But even in this case, a number of frames might be lost or corrupted. When in stream mode, no self-recovery is ever possible, and synchronization is lost forever. A frame burst counter (see [Section 23.10.4.11.2.3, Frame Burst Counter](#)) is used to limit the number of these potentially corrupted frames, simply by limiting the number of frames a transmitter can send back-to-back in a pipelined mode. A tailing bit counter (see [Section 23.10.4.11.2.2, Frame Tailing Bit Counter](#)) is used to check that the receiver and transmitter agree on being at the end of frame.

##### 23.10.4.11.1.3 Additional Edges

Additional edges result in additional clock cycles, thus causing the receiver to get more data than expected on a given frame.

In synchronized flow, when the receive side thinks the frame is over, the transmitter keeps sending data until the end of frame is reached. In frame mode, if these additional bits are all 0s, they are discarded by the receiver, then transmission stops, and the receiver raises READY when it is available to receive a new frame. The transmit side starts a new frame and synchronization is recovered. But if the additional bits are 1 (at least one), this can misguide the receiver that a new frame has already started and READY stays at

0. The transmit side stops, waiting for  $READY=1$ , never occurring condition, as the receiver has lost synchronization. In this case, the link is stuck. This is also the case in stream mode, when the receiver wrongly thinks that a new frame is started, so it is not available to receive a full new frame, and it does not set  $READY=1$ . A frame time-out counter (see [Section 23.10.4.11.2.1, Frame Time-out Counter](#)) is used to escape this deadlock condition.

### 23.10.4.11.2 Error Counters

Clocks for these three counters are provided through a division of functional clock. These counters can be set to a value, and can be stopped when some events occur. Events flag errors and set a corresponding bit in the [HSR\\_ERROR\\_Pp](#) register.

#### 23.10.4.11.2.1 Frame Time-out Counter

This counter is clocked by the functional clock, divided by the factor specified in the [HSR\\_DIVISOR\\_Pp\[7:0\] RX\\_RATE\\_DIV\\_VAL](#) bit field, to provide a clock at receive bit rate. Receive bit rate is always communicated by remote transmitter through protocol layer. This counter starts when the first bit of the frame is found, and stopped when 36 (or 37 depending on the mode) bits are correctly received by receiver. If the counter overflows, an error is generated (the [HSR\\_ERROR\\_Pp\[1\] FTE](#) bit is set to 0). In OPP reduction of functional clock only (receive side kept in fixed voltage domain), software can decide to reduce either the division factor for the clock, or the count, to use a clock consistent with receive rate, or a reduced count of a slower clock. This is transparent for hardware. This counter is not started when receive rate is outside the full error detection rate range.

#### 23.10.4.11.2.2 Frame Tailing Bit Counter

This counter can be clocked by the same clock as frame timeout counter, which has the receiver bit rate frequency. It starts after reception of the last bit of a frame, and  $READY = 0$ ; in pipelined flow, this counter is only started after the end of the last frame received. No event stops the counter, except its own overflow. It counts up to 8. An error is logged (the [HSR\\_ERROR\\_Pp\[4\] TBE](#) bit is set to 0) if a 1 bit is received in frame mode before counter overflow, or any bit is received in stream mode before counter overflow. In OPP reduction of functional clock only (receive side kept in fixed voltage domain), software can decide to reduce either the division factor for the clock, or the count, to use a clock consistent with receive rate, or a reduced count of a slower clock. This is transparent for hardware. This counter is not started when receive rate is outside the full error detection rate range.

#### 23.10.4.11.2.3 Frame Burst Counter

Differently from previous counters, this counter is clocked by a frame synchronization signal obtained by received data directly. Its goal is to reduce the number of contiguous frames sent in pipelined flow. As seen in [Section 23.10.4.11.1, Detection of Errors](#), in some cases after missing/additional clock cycles, synchronization is lost and these pipelined frames can be corrupted. Software sets the count value, which can be as high as 256 (no higher values are supported by hardware). If  $READY$  is dropped to 0, for reasons other than counter expiration, the counter is stopped and restarted as soon as transmission resumes. At the end of the count,  $READY$  drops to 0. The tailing bit counter is started, and if it overflows without error detection, and receiver has enough room for another frame,  $READY$  is reset to 1. No error and no interrupt are generated on this counter.

### 23.10.4.11.3 Error Registers

One error register per port is provided to check the error source that issued an interrupt, with more details than the error bit in the interrupt status register.

The [HSR\\_ERROR\\_Pp\[0\] SIG](#): Signal error bits are kept for legacy reason, and they can be used for any abnormal situation the hardware wants to flag.

The [HSR\\_ERROR\\_Pp\[1\] FTE](#): Frame time-out error bits flag the situation when not even a frame has been received in the time corresponding to two or more frames, depending on FTE counter software setting.

The [HSR\\_ERROR\\_Pp\[4\] FBE](#): Tailing bit error bits flag the situation when at least one spurious bit is received after the end of a frame, in the time corresponding to two or more bit time.

The [HSR\\_ERROR\\_Pp\[7\]](#) RME: Receive mapping error bits flag the situation when data are received on a channel that is not mapped to any enabled FIFO.

The [HSR\\_ERROR\\_Pp\[11\]](#) TME: Transmit mapping error bits flag the situation when DMA tries to access a FIFO that is not mapped to any logical channel, or not enabled.

These two registers have each a companion [HSR\\_ERRORACK\\_Pp](#) register, where software must set the corresponding error bit to 1 to clear it.

PRELIMINARY

## 23.10.5 HSI Programming Guide

### 23.10.5.1 HSI Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the HSI module.

#### 23.10.5.1.1 Global Initialization

##### 23.10.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the HSI module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the HSI.

See [Section 23.10.3, HSI Integration](#), for further information.

**Table 23-793. Global Initialization of Surrounding Modules for HSI**

Surrounding Modules	Comments
PRCM	HSI functional and interface clocks must be enabled. For more information, see <a href="#">Chapter 3</a>
Control Module	Module specific pad muxing must be set in the control module. For more information, see <a href="#">Chapter 18</a> .
(Optional)Interrupt Controller	Cortex-A9 MPU, or Cortex-M3 MPU, or DSP interrupt controller must be configured to enable the interrupt request generation to the Cortex-A9 MPU, or Cortex-M3 MPU, or DSP subsystem. For information about enabling interrupts, see <a href="#">Chapter 5</a> .
(Optional)Interconnect	For information about the L4 interconnect configuration, see <a href="#">Chapter 13, Interconnects</a> For information about L3 interconnect configuration, see <a href="#">Chapter 13, Interconnects</a>

##### 23.10.5.1.1.2 HSI Global Initialization

###### 23.10.5.1.1.2.1 Main Sequence - HSI Global Initialization

This procedure initializes the HSI module after a power-on or software reset.

**Table 23-794. HSI Global Initialization**

Step	Register / Bit Field / Programming Model	Value
Perform a HSI software reset.	<a href="#">HSI_SYSCONFIG</a> [1] SOFTRESET	0x1
<b>WAIT UNTIL</b> HSI reset is complete.	<a href="#">HSI_SYSSTATUS</a> [0] RESETDONE	= 0x1
Set standby mode configuration.	<a href="#">HSI_SYSCONFIG</a> [13:12] MIDDLEMODE	0x-
Set idle mode configuration.	<a href="#">HSI_SYSCONFIG</a> [4:3] SIDLEMODE	0x-
Clock auto gating.	<a href="#">HSI_SYSCONFIG</a> [0] AUTOIDLE	0x-
Set the transmit data rate.	<a href="#">HST_DIVISOR_Pp</a> [7:0] TX_RATE_DIV_VAL	0x-
(Optional) Enable DMA.	<a href="#">DMA_GCR</a> [0] SWITCH_OFF	0x0

##### 23.10.5.1.2 HSI Operational Modes Configuration

###### 23.10.5.1.2.1 HSI Transmit Mode/Receive Mode

###### 23.10.5.1.2.1.1 Main Sequence - Configure HSI Transmitter

**Table 23-795. HSI Configure HSI Transmitter**

Step	Register / Bit Field / Programming Model	Value
Set the mode	<a href="#">HST_MODE_Pp</a> [1:0] MODE_VAL	0x-
Set the data flow type	<a href="#">HST_MODE_Pp</a> [3:2] FLOW_VAL	0x-

**Table 23-795. HSI Configure HSI Transmitter (continued)**

Step	Register / Bit Field / Programming Model	Value
Set the hsi_wake line control	HST_MODE_Pp[4] WAKE_CTRL	0x-
Set the transmission bit rate	HST_DIVISOR_Pp[7:0] TX_RATE_DIV_VAL	0x-
Set the arbitration mode	HST_ARBMODE_Pp[0] ARB_VAL	0x-
Set the number of active HSI logical channels	HST_CHANNELS_Pp[4:0] CHAN_NUM_VAL	0x-
Set the TX FIFO threshold	HST_MAPPING[13:10] THRESHOLD	0x-
Associate the TX FIFO to a HSI port	HST_MAPPING[7] PORT_NUMBER	0x-
Associate the TX FIFO to a HSI logical channel	HST_MAPPING[4:1] CH_NUMBER	0x-
Enable the TX FIFO	HST_MAPPING[0] ENABLE	0x-

### 23.10.5.1.2.1.2 Main Sequence - Configure HSI Receiver

**Table 23-796. HSI Configure HSI Receiver**

Step	Register / Bit Field / Programming Model	Value
Set the mode in Receive Mode	HSR_MODE_Pp[1:0] MODE_VAL	0x-
Set the data flow type in Receive Mode	HSR_MODE_Pp[3:2] FLOW_VAL	0x-
Set a clock frequency for the error detection counters	HSR_DIVISOR_Pp[7:0] RX_RATE_DIV_VAL	0x-
Set the number of active HSI logical channels	HSR_CHANNELS_Pp[4:0] CHAN_NUM_VAL	0x-
Set the frame timeout error counter	HSR_COUNTERS_Pp[19:0] FT	0x-
Set the tailing bit error counter	HSR_COUNTERS_Pp[23:20] TB	0x-
Set the frame burst error counter	HSR_COUNTERS_Pp[31:24] FB	0x-
Associate the RX FIFO to a HSI port	HSR_MAPPING[7] PORT_NUMBER	0x-
Associate the RX FIFO to a HSI logical channel	HSR_MAPPING[4:1] CH_NUMBER	0x-
Enable the RX FIFO	HSR_MAPPING[0] ENABLE	0x-

### 23.10.5.1.2.1.3 Main Sequence - HSI Polling Method

#### 23.10.5.1.2.1.3.1 Sub-sequence - HSI Transmit in Polling Method

Figure 23-232 is a procedure flow chart for HSI Transmit in Polling Method.



Figure 23-232. HSI Transmit in Polling Method

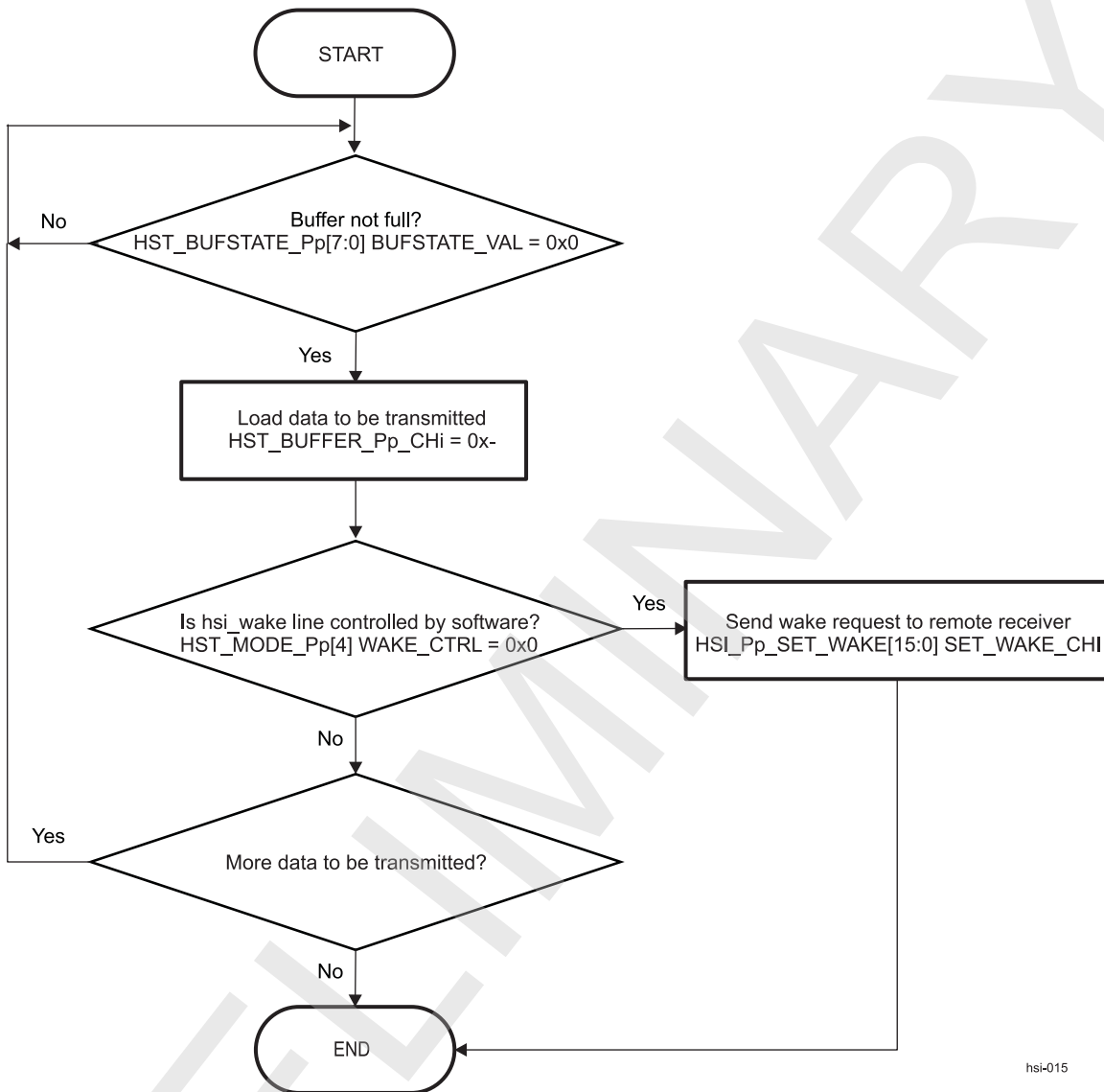


Table 23-797. Register call summary for Sub-sequence - HSI Transmit in Polling Method

Register Name	Register Name	Register Name
HST_BUFSTATE_Pp	HST_BUFFER_Pp_CHN_i	HST_MODE_Pp
HSI_Pp_SET_WAKE		

23.10.5.1.2.1.3.2 Sub-sequence - HSI Receive in Polling Method

Table 23-798. Event Servicing in HSI Receive in Polling Method

Step	Register / Bit Field / Programming Model	Value
IF : Is there data frame to be read?	HSR_BUFSTATE_Pp[7:0]BUFSTATE_VAL	=0x1
Identify number of words available in RX FIFO for reading.	HSR_MAPPING[13:10] WORDS	0x-
Read data from RX FIFO.	HSR_BUFFER_Pp_CHN_i[31:0] DATA	0x-
ENDIF		

### 23.10.5.1.2.1.4 Main Sequence - HSI Interrupt Mode

**Table 23-799. HSI Interrupt Mode**

Step	Register / Bit Field / Programming Model	Value
Disable DMA engine	DMA_GCR[0] SWITCH_OFF	0x1
Configure Transmitter	Refer to <a href="#">Table 23-795</a>	
Configure Receiver	Refer to <a href="#">Table 23-796</a>	
Enable HSI Transmission Interrupts	Refer to <a href="#">Table 23-800</a>	
<b>WHEN HSI INTERRUPT OCCURS</b>	Refer to <a href="#">Figure 23-233</a>	
Enable HSI Reception Interrupts	Refer to <a href="#">Table 23-802</a>	
<b>WHEN HSI INTERRUPT OCCURS</b>	Refer to <a href="#">Table 23-798</a>	

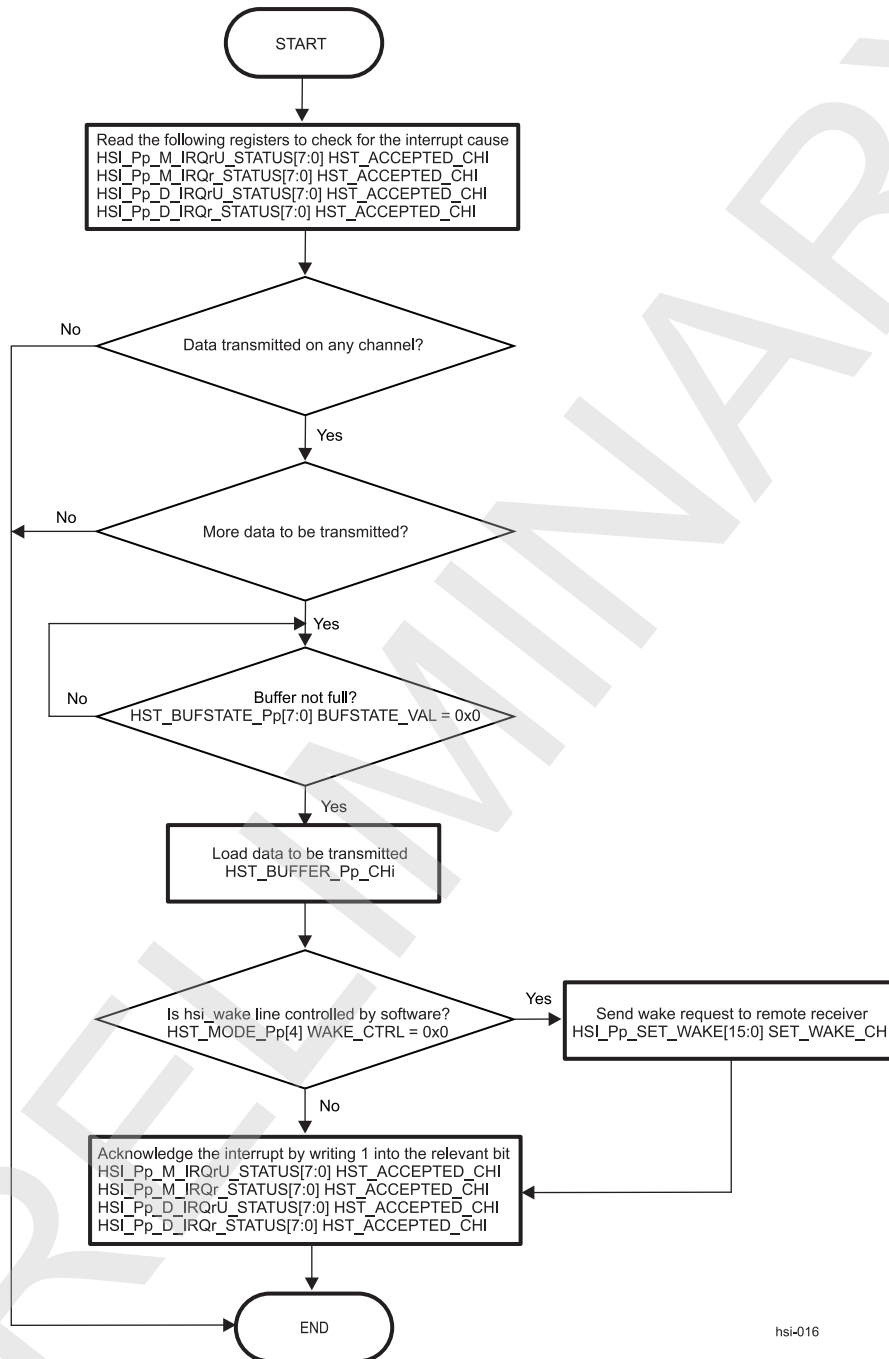
#### 23.10.5.1.2.1.4.1 Sub-sequence - Interrupt Transmit Mode

**Table 23-800. Enable HSI Transmit Interrupts**

Step	Register / Bit Field / Programming Model	Value
Enable data accepted interrupt.	<a href="#">HSI_Pp_M_IRQrU_ENABLE[7:0]</a> HST_ACCEPTED_ENI and/or <a href="#">HSI_Pp_D_IRQrU_ENABLE[7:0]</a> HST_ACCEPTED_ENI and/or <a href="#">HSI_Pp_M_IRQr_ENABLE[7:0]</a> HST_ACCEPTED_ENI and/or <a href="#">HSI_Pp_D_IRQr_ENABLE[7:0]</a> HST_ACCEPTED_ENI	0x-

Figure 23-233 is a procedure flow chart for Event Servicing in HSI Transmission Mode.

Figure 23-233. Event Servicing in HSI Transmission Mode



hsi-016

Table 23-801. Register call summary for Sub-sequence - Event Servicing in HSI Transmission Mode

Register Name	Register Name	Register Name
HST_Pp_M_IRQrU_STATUS[7:0]HST_ACCEPTED_CHI	HST_Pp_D_IRQrU_STATUS[7:0]HST_ACCEPTED_CHI	HST_Pp_M_IRQrU_STATUS[7:0]HST_ACCEPTED_CHI
HST_Pp_D_IRQrU_STATUS[7:0]HST_ACCEPTED_CHI	HST_BUFFER_Pp_CHN_i[31:0]DATA	HST_MODE_Pp[4]WAKE_CTRL

**Table 23-801. Register call summary for Sub-sequence - Event Servicing in HSI Transmission Mode (continued)**

Register Name	Register Name	Register Name
HSI_Pp_SET_WAKE[15:0]	HSI_SET_WAK	
E_CHi		

### 23.10.5.1.2.1.4.2 Sub-sequence - Interrupt Receive Mode

**Table 23-802. Enable HSI Receive Interrupts**

Step	Register / Bit Field / Programming Model	Value
Enable break detected interrupt.	HSI_Pp_M_IRQrU_ENABLE[25] HSR_BREAK and/or HSI_Pp_D_IRQrU_ENABLE[25] HSR_BREAK and/or HSI_Pp_M_IRQr_ENABLE[25] HSR_BREAK and/or HSI_Pp_D_IRQr_ENABLE[25] HSR_BREAK	0x1
Enable wake detected interrupt.	HSI_Pp_M_IRQrU_ENABLE[26] HSR_WAKE and/or HSI_Pp_D_IRQrU_ENABLE[26] HSR_WAKE and/or HSI_Pp_M_IRQr_ENABLE[26] HSR_WAKE and/or HSI_Pp_D_IRQr_ENABLE[26] HSR_WAKE	0x1
Enable overrun interrupt.	HSI_Pp_M_IRQrU_ENABLE[23:16] HSR_OVERRUN_ENI and/or HSI_Pp_D_IRQrU_ENABLE[23:16] HSR_OVERRUN_ENI and/or HSI_Pp_M_IRQr_ENABLE[23:16] HSR_OVERRUN_ENI and/or HSI_Pp_D_IRQr_ENABLE[23:16] HSR_OVERRUN_ENI	0x-
Enable data available interrupt.	HSI_Pp_M_IRQrU_ENABLE[15:8] HSR_AVAILABLE_ENI and/or HSI_Pp_D_IRQrU_ENABLE[15:8] HSR_AVAILABLE_ENI and/or HSI_Pp_M_IRQr_ENABLE[15:8] HSR_AVAILABLE_ENI and/or HSI_Pp_D_IRQr_ENABLE[15:8] HSR_AVAILABLE_ENI	0x-

Figure 23-234 is a procedure flow chart for Event Servicing in HSI Reception Mode.

Figure 23-234. Event Servicing in HSI Reception Mode

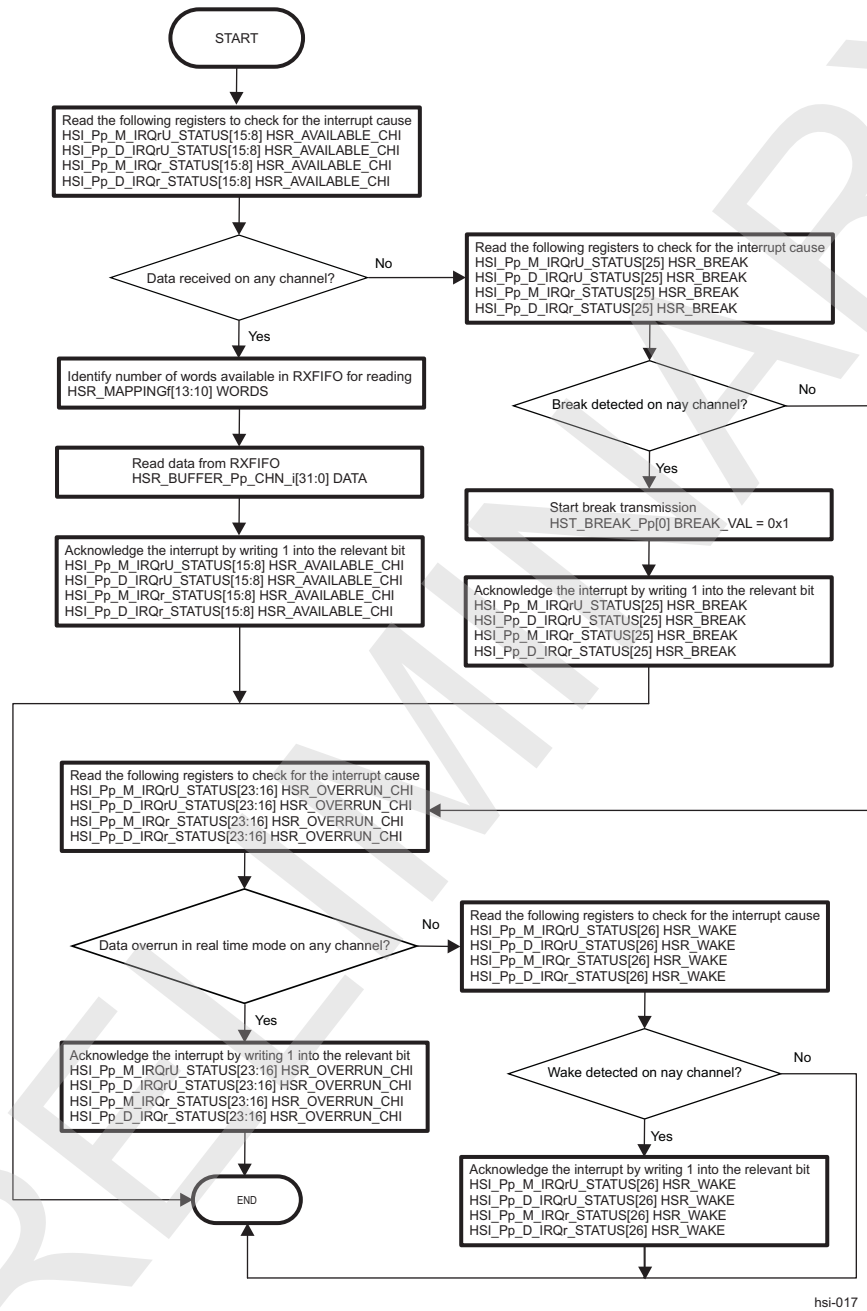


Table 23-803. Register call summary for Sub-sequence - Event Servicing in HSI Reception Mode

Register Name	Register Name	Register Name
HSI_Pp_M_IRQrU_STATUS[15:8] HSR_AVAILABLE_CHI	HSI_Pp_D_IRQrU_STATUS[15:8] HSR_AVAILABLE_CHI	HSI_Pp_M_IRQr_STATUS[15:8] HSR_AVAILABLE_CHI
HSI_Pp_D_IRQr_STATUS[15:8] HSR_AVAILABLE_CHI	HSR_MAPPING[13:10] WORDS	HSR_BUFFER_Pp_CHN_i[31:0] DATA
HSI_Pp_M_IRQrU_STATUS[25] HSR_BREAK	HSI_Pp_D_IRQrU_STATUS[25] HSR_BREAK	HSI_Pp_M_IRQr_STATUS[25] HSR_BREAK
HSI_Pp_D_IRQr_STATUS[25] HSR_BREAK	HST_BREAK_Pp[0] BREAK_VAL	HSI_Pp_M_IRQrU_STATUS[23:16] HSR_OVERRUN_CHI

**Table 23-803. Register call summary for Sub-sequence - Event Servicing in HSI Reception Mode (continued)**

Register Name	Register Name	Register Name
HSI_Pp_D_IRQrU_STATUS[23:16] HSR_OVERRUN_CHI	HSI_Pp_M_IRQr_STATUS[23:16] HSR_OVERRUN_CHI	HSI_Pp_D_IRQr_STATUS[23:16] HSR_OVERRUN_CHI

### 23.10.5.1.2.1.5 Main Sequence -HSI DMA Mode

#### 23.10.5.1.2.1.5.1 Sub-sequence - DMA Configuration

**Table 23-804. HSI DMA Configuration**

Step	Register / Bit Field / Programming Model	Value
Perform a DMA software reset.	DMA_GRST[0] SWRESET	0x1
<b>WAIT UNTIL</b> DMA reset is complete.	DMA_GRST[0] SWRESET	= 0x0
Set autoidle mode configuration for DMA.	DMA_GCR[3] AUTOGATING	0x1
Configure HSI DMI Receiver.	Refer to <a href="#">Table 23-805</a>	
Configure HSI DMI Transmit.	Refer to <a href="#">Table 23-806</a>	
Enable DMA Interrupts.	Refer to <a href="#">Section 23.10.5.1.2.1.5.4</a>	
Enable DMA channel to start DMA transfer.	DMA_CCR_CSDP_[23] ENABLE	0x1
<b>WHEN DMA INTERRUPT OCCURS</b>	Refer to <a href="#">Table 23-799</a>	

#### 23.10.5.1.2.1.5.2 Sub-sequence - DMA Receive Mode

**Table 23-805. Configure DMA Receive Mode**

Step	Register / Bit Field / Programming Model	Value
Set the transfer source to peripheral port.	DMA_CCR_CSDP_[5:2] SRC	0x9
(Optional) Set the source burst enable.	DMA_CCR_CSDP_[8:7] SRC_BURST_EN	0x1
Set the transfer destination to memory port.	DMA_CCR_CSDP_[12:9] DST	0x8
(Optional) Set the destination burst enable.	DMA_CCR_CSDP_[15:14] DST_BURST_EN	0x1
Set the source addressing mode to constant address.	DMA_CCR_CSDP_[29:28] SRC_ADD_MODE	0x0
Set the destination addressing mode to post-incremented address.	DMA_CCR_CSDP_[31:30] DST_ADD_MODE	0x1
Set the source start address .	DMA_CSSA_i[31:0] ADDR	0x-
Set the destination start address.	DMA_CDSA_i[31:0] ADDR	0x-
Set the number of elements in a block.	DMA_CEN_i[15:0] SIZE	0x-

#### 23.10.5.1.2.1.5.3 Sub-sequence - DMA Transmit Mode

**Table 23-806. Configure DMA Transmit Mode**

Step	Register / Bit Field / Programming Model	Value
Set the transfer source to memory port.	DMA_CCR_CSDP_[5:2] SRC	0x8
(Optional) Set the source burst enable.	DMA_CCR_CSDP_[8:7] SRC_BURST_EN	0x1
Set the transfer destination to peripheral port.	DMA_CCR_CSDP_[12:9] DST	0x9
(Optional) Set the destination burst enable to single access.	DMA_CCR_CSDP_[15:14] DST_BURST_EN	0x1
Set the source addressing mode to post-incremented address.	DMA_CCR_CSDP_[29:28] SRC_ADD_MODE	0x1
Set the destination addressing mode to constant address.	DMA_CCR_CSDP_[31:30] DST_ADD_MODE	0x0

**Table 23-806. Configure DMA Transmit Mode (continued)**

Step	Register / Bit Field / Programming Model	Value
Set the source start address.	DMA_CSSA_i[31:0] ADDR	0x-
Set the destination start address.	DMA_CDSA_i[31:0] ADDR	0x-
Set the number of elements in a block.	DMA_CEN_i[15:0] SIZE	0x-

#### 23.10.5.1.2.1.5.4 Sub-sequence - Enable DMA Interrupts

**Table 23-807. Enable DMA Interrupts**

Step	Register / Bit Field / Programming Model	Value
Enable the DMA channels that shall generate an interrupt to a processor (MPU and/or DSP).	HSI_DMA_M_IRQENABLE[15:0] DMA_EN_CHI and/or HSI_DMA_D_IRQENABLE[15:0] DMA_EN_CHI	0x-
Enable the DMA full block interrupt event.	DMA_CSR_CCIR_i[5] BLOCK_IE	0x1
Enable the DMA half block interrupt event.	DMA_CSR_CCIR_i[2] HALF_IE	0x1
Enable the DMA time-out overflow interrupt event.	DMA_CSR_CCIR_i[0] TOUT_IE	0x1

#### 23.10.5.1.2.1.5.5 Sub-sequence - DMA Interrupt Servicing

**Table 23-808. Event Servicing in DMA Transmit/Receive Mode**

Step	Register / Bit Field / Programming Model	Value
Identify which DMA channel has generated an interrupt.	HSI_DMA_M_IRQSTATUS[15:0] DMA_CHI and/or HSI_DMA_D_IRQSTATUS[15:0] DMA_CHI	0x-
Identify the interrupt event type.	DMA_CSR_CCIR_i[21] BLOCK_IS and/or DMA_CSR_CCIR_i[18] HALF_IS and/or DMA_CSR_CCIR_i[16] TOUT_IS	0x-
Acknowledge DMA global interrupt by writing 1 into the relevant bit.	HSI_DMA_M_IRQSTATUS[15:0] DMA_CHI and/or HSI_DMA_D_IRQSTATUS[15:0] DMA_CHI	0x-

#### 23.10.5.1.2.1.6 Main Sequence - HSI Error Reporting

**Table 23-809. Enable HSI Error Interrupts**

Step	Register / Bit Field / Programming Model	Value
Enable error detected interrupt.	HSI_Pp_M_IRQrU_ENABLE[24] HSR_ERROR and/or HSI_Pp_D_IRQrU_ENABLE[24] HSR_ERROR and/or HSI_Pp_M_IRQr_ENABLE[24] HSR_ERROR and/or HSI_Pp_D_IRQr_ENABLE[24] HSR_ERROR	0x1

**Table 23-810. Event Servicing Error**

Step	Register / Bit Field / Programming Model	Value
Read interrupt status register.	HSI_Pp_M_IRQrU_STATUS[24] HSR_ERROR and/or HSI_Pp_D_IRQrU_STATUS[24] HSR_ERROR and/or HSI_Pp_M_IRQr_STATUS[24] HSR_ERROR and/or HSI_Pp_D_IRQr_STATUS[24] HSR_ERROR	0x-
Identify the interrupt event type.	HSR_ERROR_Pp[0] SIG HSR_ERROR_Pp[1] FTE HSR_ERROR_Pp[4] FBE HSR_ERROR_Pp[7] RME HSR_ERROR_Pp[11] TME	0x-

**Table 23-810. Event Servicing Error (continued)**

Step	Register / Bit Field / Programming Model	Value
Clear the event by writing 1 into the relevant bit.	HSR_ERRORACK_Pp[0] SIG HSR_ERRORACK_Pp[1] FTE HSR_ERRORACK_Pp[4] FBE HSR_ERRORACK_Pp[7] RME HSR_ERRORACK_Pp[11] TME	0x1
Acknowledge the interrupt by writing 1 into the relevant bit.	HSI_Pp_M_IRQrU_STATUS[24] HSR_ERROR and/or HSI_Pp_D_IRQrU_STATUS[24] HSR_ERROR and/or HSI_Pp_M_IRQr_STATUS[24] HSR_ERROR and/or HSI_Pp_D_IRQr_STATUS[24] HSR_ERROR	0x1

## 23.10.6 HSI Register Manual

### 23.10.6.1 HSI Instance Summary

**Table 23-811. HSI Instance Summary**

Module Name	Module Base Address	Size
HSI_TOP	0x4A05 8000	5 KB
HSI_DMA_CHANNELS	0x4A05 9800	1 KB
HSI_PORTS	0x4A05 A000	8 KB

### 23.10.6.2 HSI\_TOP Registers

#### 23.10.6.2.1 HSI\_TOP Register Summary

**Table 23-812. HSI\_TOP Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	HSI_TOP Physical Address
HSI_REVISION	R	32	0x0000 0000	0x4A05 8000
HSI_HWINFO	R	32	0x0000 0004	0x4A05 8004
HSI_SYSCONFIG	RW	32	0x0000 0010	0x4A05 8010
HSI_SYSSTATUS	R	32	0x0000 0014	0x4A05 8014
HSI_Pp_M_IRQrU_STATUS <sup>(1)</sup>	RW	32	0x0000 0408 + (0x10 * b) + (0x8 * r)	0x4A05 8408 + (0x10 * b) + (0x8 * r)
HSI_Pp_M_IRQrU_ENABLE <sup>(1)</sup>	RW	32	0x0000 040C + (0x10 * b) + (0x8 * r)	0x4A05 840C + (0x10 * b) + (0x8 * r)
HSI_Pp_D_IRQrU_STATUS <sup>(1)</sup>	RW	32	0x0000 0430 + (0x10 * b) + (0x8 * r)	0x4A05 8430 + (0x10 * b) + (0x8 * r)
HSI_Pp_D_IRQrU_ENABLE <sup>(1)</sup>	RW	32	0x0000 0434 + (0x10 * b) + (0x8 * r)	0x4A05 8434 + (0x10 * b) + (0x8 * r)
HSI_DMA_M_IRQSTATUS	RW	32	0x0000 0800	0x4A05 8800
HSI_DMA_M_IRQENABLE	RW	32	0x0000 0804	0x4A05 8804
HSI_Pp_M_IRQr_STATUS <sup>(1)</sup>	RW	32	0x0000 0808 + (0x10 * b) + (0x8 * r)	0x4A05 8808 + (0x10 * b) + (0x8 * r)
HSI_Pp_M_IRQr_ENABLE <sup>(1)</sup>	RW	32	0x0000 080C + (0x10 * b) + (0x8 * r)	0x4A05 880C + (0x10 * b) + (0x8 * r)
HSI_DMA_D_IRQSTATUS	RW	32	0x0000 0828	0x4A05 8828

<sup>(1)</sup> p = 1 to 2  
r = 0 to 1  
b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)



**Table 23-812. HSI\_TOP Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	HSI_TOP Physical Address
HSI_DMA_D_IRQENABLE	RW	32	0x0000 082C	0x4A05 882C
HSI_Pp_D_IRQr_STATUS <sup>(1)</sup>	RW	32	0x0000 0830 + (0x10 * b) + (0x8 * r)	0x4A05 8830 + (0x10 * b) + (0x8 * r)
HSI_Pp_D_IRQr_ENABLE <sup>(1)</sup>	RW	32	0x0000 0834 + (0x10 * b) + (0x8 * r)	0x4A05 8834 + (0x10 * b) + (0x8 * r)
HSI_Pp_WAKE <sup>(1)</sup>	RW	32	0x0000 0C00 + (0x10 * b)	0x4A05 8C00 + (0x10 * b)
HSI_Pp_CLEAR_WAKE <sup>(1)</sup>	RW	32	0x0000 0C04 + (0x10 * b)	0x4A05 8C04 + (0x10 * b)
HSI_Pp_SET_WAKE <sup>(1)</sup>	RW	32	0x0000 0C08 + (0x10 * b)	0x0005 8C08 + (0x10 * b)
RESERVED	RW	32	0x0000 1000	0x4A05 9000
DMA_GCR	RW	32	0x0000 1100	0x4A05 9100
DMA_GRST	RW	32	0x0000 1200	0x4A05 9200

**23.10.6.2.2 HSI\_TOP Register Description**

**Table 23-813. HSI\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI Internal Data

**Table 23-814. Register Call Summary for Register HSI\_REVISION**

- MIPI-HSI
- [HSI\\_TOP Registers: \[0\]](#)

**Table 23-815. HSI\_HWINFO**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	Information about the IP module's hardware configuration, that is, typically the module's HDL generics (if any). Actual field format and encoding is decided by the module's designer.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HWINFO																															

Bits	Field Name	Description	Type	Reset
31:0	HWINFO	IP module hardware configuration	R	Please refer to <sup>(1)</sup>

<sup>(1)</sup> TI Internal Data

**Table 23-816. Register Call Summary for Register HSI\_HWINFO**

MIPI-HSI

- [HSI\\_TOP Registers: \[0\]](#)

**Table 23-817. HSI\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	HSI_TOP
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>		
<b>Description</b>	This register allows controlling various parameters of the L4_CFG interface		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MIDLEMODE	RESERVED							SIDLEMODE	FREE_EMU	SOFTRESET	AUTOIDLE				

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x00000
13:12	MIDLEMODE	Master interface power management, standby/wait control 0x0: force-standby 0x1: no-standby 0x2: smart-standby 0x3: smart-standby wakeup	RW	0x2
11:5	RESERVED	Reserved	R	0x00
4:3	SIDLEMODE	Slave interface power management, request/acknowledgement control 0x0: force-idle 0x1: no-idle 0x2: smart-idle 0x3: smart-idle wakeup	RW	0x2
2	FREE_EMU	Sensitivity to emulation (debug) suspend input signal 0x0: module is sensitive to emulation suspend 0x1: module is not sensitive to emulation suspend	RW	0x0
1	SOFTRESET	Software reset 0x0: no reset applied 0x1: Software reset applied	RW	0x0
0	AUTOIDLE	Internal interface clock gating strategy 0x0: interface clock is free-running 0x1: automatic interface clock gating strategy, based on interconnect interface activity	RW	0x1

**Table 23-818. Register Call Summary for Register HSI\_SYSCONFIG**

MIPI-HSI

- [Software Reset: \[0\]](#)
- [Power Management: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [HSI Low-level Programming Models: \[7\] \[8\] \[9\] \[10\]](#)
- [HSI\\_TOP Registers: \[11\]](#)

**Table 23-819. HSI\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>
<b>Description</b>	Status on module, (reset done on bit 0, available for more status info)
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESETDONE																															

Bits	Field Name	Description	Type	Reset
31:0	RESETDONE	internal reset monitoring Read 0x0: internal reset is ongoing Read 0x1: internal reset is completed and the module is ready to be used	R	0x0000 0001

**Table 23-820. Register Call Summary for Register HSI\_SYSSTATUS**

MIPI-HSI

- [Software Reset: \[0\]](#)
- [HSI Low-level Programming Models: \[1\]](#)
- [HSI\\_TOP Registers: \[2\]](#)

**Table 23-821. HSI\_Pp\_M\_IRQrU\_STATUS**

<b>Address Offset</b>	0x0000 0408 + (0x10 * b) + (0x8 * r)	<b>Index</b>	p = 1 to 2 r = 0 to 1 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	This register collects status for half of the HSI port p events able to generate interrupt to MPU: Buffers 8..15 Read 0: event has not occurred Read 1: event has occurred Write 0: bit stays unchanged Write 1: bit gets reset to 0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HSR_WAKE	HSR_BREAK	HSR_ERROR	HSR_OVERRUN_CHI								HSR_AVAILABLE_CHI								HST_ACCEPTED_CHI								

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	HSR_WAKE	Wake detected on any channel 8..15	RW	0x0
25	HSR_BREAK	Break detected on any channel 8..15	RW	0x0
24	HSR_ERROR	Error detected on any channel 8..15	RW	0x0
23:16	HSR_OVERRUN_CHI	Data overrun in real time mode channel 8..15 (LSB stands for channel 8 and MSB for channel 15).	RW	0x00
15:8	HSR_AVAILABLE_CHI	Data received on channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0x00
7:0	HST_ACCEPTED_CHI	Data transmitted on channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0xFF

**Table 23-822. Register Call Summary for Register HSI\_Pp\_M\_IRQrU\_STATUS**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [HSI Low-level Programming Models: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [HSI\\_TOP Registers: \[12\]](#)

**Table 23-823. HSI\_Pp\_M\_IRQrU\_ENABLE**

<b>Address Offset</b>	0x0000 040C + (0x10 * b) + (0x8 * r)	p = 1 to 2 r = 0 to 1 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b> HSI_TOP
<b>Description</b>	This register masks and un masks half of HSI port p sources of interrupt to MPU: Write 0: event is masked Write 1: event is enabled	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HSR_WAKE	HSR_BREAK	HSR_ERROR	HSR_OVERRUN_ENI					HSR_AVAILABLE_ENI					HST_ACCEPTED_ENI														

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	HSR_WAKE	Wake interrupt enable for all channels 8..15	RW	0x0
25	HSR_BREAK	Break interrupt enable for all channels 8..15	RW	0x0
24	HSR_ERROR	Error interrupt enable for all channels 8..15	RW	0x0
23:16	HSR_OVERRUN_ENI	Overrun interrupt enable for channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0x00
15:8	HSR_AVAILABLE_ENI	Data available interrupt enable for channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0x00
7:0	HST_ACCEPTED_ENI	Data accepted interrupt enable for channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0x00

**Table 23-824. Register Call Summary for Register HSI\_Pp\_M\_IRQrU\_ENABLE**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [HSI Low-level Programming Models: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [HSI\\_TOP Registers: \[12\]](#)

**Table 23-825. HSI\_Pp\_D\_IRQrU\_STATUS**

<b>Address Offset</b>	0x0000 0430 + (0x10 * b) + (0x8 * r)	<b>Index</b>	p = 1 to 2 r = 0 to 1 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	This register collects status for half of the HSI port p events able to generate interrupt to DSP: Buffers 8..15 Read 0: event has not occurred Read 1: event has occurred Write 0: bit stays unchanged Write 1: bit gets reset to 0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HSR_WAKE	HSR_BREAK	HSR_ERROR	HSR_OVERRUN_CHI					HSR_AVAILABLE_CHI					HST_ACCEPTED_CHI														

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	HSR_WAKE	Wake detected on any channel 8..15	RW	0x0
25	HSR_BREAK	Break detected on any channel 8..15	RW	0x0
24	HSR_ERROR	Error on any channel 8..15	RW	0x0
23:16	HSR_OVERRUN_CHI	Data overrun in real time mode channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0x00
15:8	HSR_AVAILABLE_CHI	Data received on channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0x00
7:0	HST_ACCEPTED_CHI	Data transmitted on channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0xFF

**Table 23-826. Register Call Summary for Register HSI\_Pp\_D\_IRQrU\_STATUS**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [HSI Low-level Programming Models: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [HSI\\_TOP Registers: \[12\]](#)

**Table 23-827. HSI\_Pp\_D\_IRQrU\_ENABLE**

<b>Address Offset</b>	0x0000 0434 + (0x10 * b) + (0x8 * r)	<b>Index</b>	p = 1 to 2 r = 0 to 1 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	This register masks and unmask half of HSI port p sources of interrupt to DSP: Write 0: event is masked Write 1: event is enabled		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HSR_WAKE	HSR_BREAK	HSR_ERROR	HSR_OVERRUN_ENI					HSR_AVAILABLE_ENI					HST_ACCEPTED_ENI														

MIPI-HSI

www.ti.com

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	HSR_WAKE	Wake interrupt enable for channel 8..15	RW	0x0
25	HSR_BREAK	Break interrupt enable for channel 8..15	RW	0x0
24	HSR_ERROR	Error interrupt enable for channel 8..15	RW	0x0
23:16	HSR_OVERRUN_ENI	Overrun interrupt enable for channel 8..15	RW	0x00
15:8	HSR_AVAILABLE_ENI	Data available interrupt enable for channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0x00
7:0	HST_ACCEPTED_ENI	Data accepted interrupt enable for channel 8..15 (LSB stands for channel 8 and MSB for channel 15)	RW	0x00

**Table 23-828. Register Call Summary for Register HSI\_Pp\_D\_IRQrU\_ENABLE**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [HSI Low-level Programming Models: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [HSI\\_TOP Registers: \[12\]](#)

**Table 23-829. HSI\_DMA\_M\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0800	
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b> HSI_TOP
<b>Description</b>	This register collects status for all of the DMA events able to generate interrupt to MPU: Read 0: event hasn't occurred Read 1: event has occurred Write 0: bit stays unchanged Write 1: bit gets reset to 0	
<b>Type</b>	RW	
	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
	RESERVED	DMA_CHI
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>
31:16	RESERVED	Reserved
15:0	DMA_CHI	Channel 0..15 status (LSB stands for channel 0 and MSB for channel 15)

**Table 23-830. Register Call Summary for Register HSI\_DMA\_M\_IRQSTATUS**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\]](#)
- [HSI Low-level Programming Models: \[2\] \[3\]](#)
- [HSI\\_TOP Registers: \[4\]](#)

**Table 23-831. HSI\_DMA\_M\_IRQENABLE**

<b>Address Offset</b>	0x0000 0804		
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	This register masks and unmaskes DMA sources of interrupt to MPU: Write 0: event is masked Write 1: event is enabled		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DMA_EN_CHI															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	DMA_EN_CHI	Channel 0..15 status (LSB stands for channel 0 and MSB for channel 15)	RW	0x0000

**Table 23-832. Register Call Summary for Register HSI\_DMA\_M\_IRQENABLE**

MIPI-HSI

- [Interrupt Requests: \[0\]](#)
- [HSI Low-level Programming Models: \[1\]](#)
- [HSI\\_TOP Registers: \[2\]](#)

**Table 23-833. HSI\_Pp\_M\_IRQr\_STATUS**

<b>Address Offset</b>	0x0000 0808 + (0x10 * b) + (0x8 * r)	<b>Index</b>	p = 1 to 2 r = 0 to 1 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	This register collects status for half of the HSI port p events able to generate interrupt to MPU: Buffers 0..7 Read 0: event has not occurred Read 1: event has occurred Write 0: bit stays unchanged Write 1: bit gets reset to 0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						HSR_WAKE	HSR_BREAK	HSR_ERROR	HSR_OVERRUN_CHI							HSR_AVAILABLE_CHI							HST_ACCEPTED_CHI								

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	HSR_WAKE	Wake detected on channel i	RW	0x0
25	HSR_BREAK	Break detected on channel i	RW	0x0
24	HSR_ERROR	Error on channel i	RW	0x0
23:16	HSR_OVERRUN_CHI	Data overrun in real time mode channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00
15:8	HSR_AVAILABLE_CHI	Data received on channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00
7:0	HST_ACCEPTED_CHI	Data transmitted on channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0xFF

**Table 23-834. Register Call Summary for Register HSI\_Pp\_M\_IRQr\_STATUS**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [HSI Low-level Programming Models: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [HSI\\_TOP Registers: \[12\]](#)

**Table 23-835. HSI\_Pp\_M\_IRQr\_ENABLE**

<b>Address Offset</b>	0x0000 080C + (0x10 * b) + (0x8 * r)	<b>Index</b>	p = 1 to 2 r = 0 to 1 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	This register masks and un masks half of HSI port p sources of interrupt to Mpu: Write 0: event is masked Write 1: event is enabled		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HSR_WAKE	HSR_BREAK	HSR_ERROR	HSR_OVERRUN_ENI				HSR_AVAILABLE_ENI				HST_ACCEPTED_ENI																

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	HSR_WAKE	Wake interrupt enable for channel i	RW	0x0
25	HSR_BREAK	Break interrupt enable for channel i	RW	0x00
24	HSR_ERROR	Error interrupt enable for channel i	RW	0x0
23:16	HSR_OVERRUN_ENI	Overrun interrupt enable for channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00
15:8	HSR_AVAILABLE_ENI	Data available interrupt enable for channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00
7:0	HST_ACCEPTED_ENI	Data accepted interrupt enable for channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00

**Table 23-836. Register Call Summary for Register HSI\_Pp\_M\_IRQr\_ENABLE**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [HSI Low-level Programming Models: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [HSI\\_TOP Registers: \[12\]](#)

**Table 23-837. HSI\_DMA\_D\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0828	<b>Instance</b>	HSI_TOP
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	IRQ status register for all DMA events. Events will generate interrupt for DSP. Read 0: event has not occurred Read 1: event has occurred Write 0: bit stays unchanged Write 1: bit gets reset to 0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_CHI																							



Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	DMA_CHI	IRQ event happened on DMA channel 0..15 (LSB stands for channel 0 and MSB for channel 15)	RW	0x0000

**Table 23-838. Register Call Summary for Register HSI\_DMA\_D\_IRQSTATUS**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\]](#)
- [HSI Low-level Programming Models: \[2\] \[3\]](#)
- [HSI\\_TOP Registers: \[4\]](#)

**Table 23-839. HSI\_DMA\_D\_IRQENABLE**

<b>Address Offset</b>	0x0000 082C	<b>Instance</b>	HSI_TOP
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>		
<b>Description</b>	IRQ enable register for all DMA events signaled to DSP. Write 0: event is masked Write 1: event is enabled		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EN_CHI																							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	DMA_EN_CHI	Channel 0..15 (LSB stands for channel 0 and MSB for channel 15)	RW	0x0000

**Table 23-840. Register Call Summary for Register HSI\_DMA\_D\_IRQENABLE**

MIPI-HSI

- [Interrupt Requests: \[0\]](#)
- [HSI Low-level Programming Models: \[1\]](#)
- [HSI\\_TOP Registers: \[2\]](#)

**Table 23-841. HSI\_Pp\_D\_IRQr\_STATUS**

<b>Address Offset</b>	0x0000 0838 + (0x10 * b) + (0x8 * r)	<b>Index</b>	p = 1 to 2 r = 0 to 1 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	This register collects status for half of the HSI port p events able to generate interrupt to DSP: Buffers 0..7 Read 0: event has not occurred Read 1: event has occurred Write 0: bit stays unchanged Write 1: bit gets reset to 0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HSR_WAKE	HSR_BREAK	HSR_ERROR	HSR_OVERRUN_CHI				HSR_AVAILABLE_CHI				HST_ACCEPTED_CHI																

MIPI-HSI

www.ti.com

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	HSR_WAKE	Wake detected on channel i	RW	0x0
25	HSR_BREAK	Break detected on channel i	RW	0x0
24	HSR_ERROR	Error on channel i	RW	0x0
23:16	HSR_OVERRUN_CHI	Data overrun in real time mode channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00
15:8	HSR_AVAILABLE_CHI	Data received on channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00
7:0	HST_ACCEPTED_CHI	Data transmitted on channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0xFF

**Table 23-842. Register Call Summary for Register HSI\_Pp\_D\_IRQr\_STATUS**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [HSI Low-level Programming Models: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [HSI\\_TOP Registers: \[12\]](#)

**Table 23-843. HSI\_Pp\_D\_IRQr\_ENABLE**

<b>Address Offset</b>	0x0000 083C + (0x10 * b) + (0x8 * r)	<b>Index</b>	p = 1 to 2 r = 0 to 1 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	This register masks and un masks half of HSI port p sources of interrupt to DSP: Write 0: event is masked Write 1: event is enabled		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HSR_WAKE	HSR_BREAK	HSR_ERROR	HSR_OVERRUN_ENI				HSR_AVAILABLE_ENI				HST_ACCEPTED_ENI																

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26	HSR_WAKE	Wake interrupt enable for channel i	RW	0x0
25	HSR_BREAK	Break interrupt enable for channel i	RW	0x0
24	HSR_ERROR	Error interrupt enable for channel i	RW	0x0
23:16	HSR_OVERRUN_ENI	Overrun interrupt enable for channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00
15:8	HSR_AVAILABLE_ENI	Data available interrupt enable for channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00
7:0	HST_ACCEPTED_ENI	Data accepted interrupt enable for channel 0..7 (LSB stands for channel 0 and MSB for channel 7)	RW	0x00

**Table 23-844. Register Call Summary for Register HSI\_Pp\_D\_IRQr\_ENABLE**

MIPI-HSI

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [HSI Low-level Programming Models: \[6\] \[7\] \[8\] \[9\] \[10\] \[11\]](#)
- [HSI\\_TOP Registers: \[12\]](#)

**Table 23-845. HSI\_Pp\_WAKE**

<b>Address Offset</b>	0x0000 0C00 + (0x10 * b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	Programmed wake state for each channel, in port p 0x0: no channel wakeup on WAKE line requested 0x1: channel wakeup requested		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HSI_WAKE_CHI																							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	HSI_WAKE_CHI	Channel0..15 (LSB stands for channel 0 and MSB for channel 15)	R	0x0000

**Table 23-846. Register Call Summary for Register HSI\_Pp\_WAKE**

MIPI-HSI

- [HSI Wake Generator: \[0\] \[1\] \[2\] \[3\]](#)
- [HSI\\_TOP Registers: \[4\]](#)

**Table 23-847. HSI\_Pp\_CLEAR\_WAKE**

<b>Address Offset</b>	0x0000 0C04 + (0x10 * b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	Clear register for programmed wake state on port p HST_WAKE for each channel 0x0: no effect 0x1: clears bit		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HSI_HSI_CLEAR_WAKE_CHI																							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	HSI_CLEAR_WAKE_CHI	Channel 0..15 (LSB stands for channel 0 and MSB for channel 15)	W	0x0000

**Table 23-848. Register Call Summary for Register HSI\_Pp\_CLEAR\_WAKE**

MIPI-HSI

- [HSI Wake Generator: \[0\]](#)
- [HSI\\_TOP Registers: \[1\]](#)

**Table 23-849. HSI\_Pp\_SET\_WAKE**

<b>Address Offset</b>	0x0000 0C08 + (0x10 * b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>	<b>Instance</b>	HSI_TOP
<b>Description</b>	Set function for wake state for each channel, in port p 0x0: no effect 0x1: sets bit		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HSI_READY_LVL		HSI_3_WIRES		HSI_SET_WAKE_CHI																			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17	HSI_READY_LVL	READY default level 0x0: READY defaults to 0 0x1: READY defaults to 1	RW	0x0
16	HSI_3_WIRES	Sets 3 wires behavior 0x0: sets 4 wires behavior (wakeup with WAKE signal) 0x1: sets 3 wire behavior (wakeup on line activity without WAKE signal)	RW	0x0
15:0	HSI_SET_WAKE_CHI	Sets programmed WAKE state for channel i Write 0x0: no effect Write 0x1: sets bit	W	0x0000

**Table 23-850. Register Call Summary for Register HSI\_Pp\_SET\_WAKE**

## MIPI-HSI

- [HSR Module: \[0\]](#)
- [HSI Wake Generator: \[1\] \[2\]](#)
- [HSI Low-level Programming Models: \[3\] \[4\]](#)
- [HSI\\_TOP Registers: \[5\]](#)

**Table 23-851. DMA\_GCR**

<b>Address Offset</b>	0x0000 1100	<b>Instance</b>	HSI_TOP
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>		
<b>Description</b>	Global Control Register: suspend and clock gating		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AUTOGATING		RESERVED		SWITCH_OFF											

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x00000000
3	AUTOGATING	DMA clock autogating enable 0x0: no DMA clock control 0x1: DMA clock control	RW	0x0
2:1	RESERVED	Reserved	R	0x0
0	SWITCH_OFF	DMA global clock control 0x0: DMA Clock released 0x1: DMA Clock cutoff	RW	0x0

**Table 23-852. Register Call Summary for Register DMA\_GCR**

MIPI-HSI

- [Power Management: \[0\] \[1\] \[2\]](#)
- [HSI Low-level Programming Models: \[3\] \[4\] \[5\]](#)
- [HSI\\_TOP Registers: \[6\]](#)

**Table 23-853. DMA\_GRST**

<b>Address Offset</b>	0x0000 1200	<b>Instance</b>	HSI_TOP
<b>Physical Address</b>	Please refer to <a href="#">Table 23-812</a>		
<b>Description</b>	DMA software reset control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SWRESET			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	SWRESET	DMA sw reset control bit 0x1: writing 1 resets the DMA. It is automatically reset to 0 by hardware once software reset is done	RW	0x0

**Table 23-854. Register Call Summary for Register DMA\_GRST**

MIPI-HSI

- [Software Reset: \[0\]](#)
- [HSI Low-level Programming Models: \[1\] \[2\]](#)
- [HSI\\_TOP Registers: \[3\]](#)

### 23.10.6.3 HSI\_DMA\_CHANNELS registers

#### 23.10.6.3.1 HSI\_DMA\_CHANNELS Register Summary

**Table 23-855. HSI\_DMA\_CHANNELS Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	HSI_DMA_CHANNELS Base Address
<a href="#">DMA_CCR_CSDP_i<sup>(1)</sup></a>	RW	32	0x0000 0000 + (0x40 * i)	0x4A05 9800 + (0x40 * i)

<sup>(1)</sup> i = 0 to 15

**Table 23-855. HSI\_DMA\_CHANNELS Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	HSI_DMA_CHANNELS Base Address
<a href="#">DMA_CSR_CCIR_i<sup>(1)</sup></a>	RW	32	0x0000 0004 + (0x40 * i)	0x4A05 9804 + (0x40 * i)
<a href="#">DMA_CSSA_i<sup>(1)</sup></a>	RW	32	0x0000 0008 + (0x40 * i)	0x4A05 9808 + (0x40 * i)
<a href="#">DMA_CDSA_i<sup>(1)</sup></a>	RW	32	0x0000 000C + (0x40 * i)	0x4A05 980C + (0x40 * i)
<a href="#">DMA_CEN_i<sup>(1)</sup></a>	RW	32	0x0000 0010 + (0x40 * i)	0x4A05 9810 + (0x40 * i)
<a href="#">DMA_CDAC_CSAC_i<sup>(1)</sup></a>	R	32	0x0000 0018 + (0x40 * i)	0x4A05 9818 + (0x40 * i)
Reserved	R	32	0x0000 0028 + (0x40 * i)	0x4A05 9828 + (0x40 * i)

**23.10.6.3.2 HSI\_DMA\_CHANNELS Register Description****Table 23-856. DMA\_CCR\_CSDP\_i**

<b>Address Offset</b>	0x0000 0000 + (0x40 * i)	<b>Index</b>	i = 0 to 15
<b>Physical Address</b>	Please refer to <a href="#">Table 23-855</a>	<b>Instance</b>	HSI_DMA_CHANNELS
<b>Description</b>	Stores source and destination parameters and channel control bits		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST_ADD_MODE		SRC_ADD_MODE		RESERVED				ENABLE	RESERVED								DST_BURST_EN		RESERVED	DST			SRC_BURST_EN		RESERVED	SRC			DATA_TYPE		

Bits	Field Name	Description	Type	Reset
31:30	DST_ADD_MODE	Destination addressing mode 0x0: constant address 0x1: post increment address	RW	0x0
29:28	SRC_ADD_MODE	Source addressing mode 0x0: constant address 0x1: post increment address	RW	0x0
27:24	RESERVED	Reserved	R	0
23	ENABLE	Logical channel enable 0x0: transfer stops and is reset 0x1: transfer is enabled. Automatically cleared by hardware once transfer is finished	RW	0
22:16	RESERVED	Reserved	R	0
15:14	DST_BURST_EN	Destination burst enable 0x0: single access 0x1: single access 0x3: burst 8x32 bits (not implemented) 0x2: burst 4x32 bits	RW	0x0
13	RESERVED	Reserved	R	0

Bits	Field Name	Description	Type	Reset
12:9	DST	Transfer destination 0x8: Transfer to Memory port 0x9: Transfer to Peripheral port	RW	0x0
8:7	SRC_BURST_EN	Source burst enable 0x0: single access 0x1: single access 0x3: burst 8x32 bits (not implemented) 0x2: burst 4x32 bits	RW	0x0
6	RESERVED	Reserved	R	0
5:2	SRC	Transfer source 0x8: Transfer From Memory port 0x9: Transfer From Peripheral port	RW	0x0
1:0	DATA_TYPE	Defines data types Implemented bit field but not used	RW	0x0

**Table 23-857. Register Call Summary for Register DMA\_CCR\_CSDP\_i**

MIPI-HSI

- [DMA Module: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- [HSI Low-level Programming Models: \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\]](#)
- [HSI\\_DMA\\_CHANNELS registers: \[22\]](#)

**Table 23-858. DMA\_CSR\_CCIR\_i**

<b>Address Offset</b>	0x0000 0004 + (0x40 * i)	<b>Index</b>	i = 0 to 15
<b>Physical Address</b>	Please refer to <a href="#">Table 23-855</a>	<b>Instance</b>	HSI_DMA_CHANNELS
<b>Description</b>	Interrupt enable and status register Interrupt flag fields will be cleared through read access using the following byteenable values: 0xC 0xF		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED								BLOCK_IS	RESERVED	HALF_IS	RESERVED	TOUT_IS	RESERVED								BLOCK_IE	RESERVED	HALF_IE	RESERVED	TOUT_IE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21	BLOCK_IS	Block transferred Read 0x1: block transferred Read 0x0: no event	R	0
20:19	RESERVED	Reserved	R	0x0
18	HALF_IS	Half block reached Read 0x1: half block transferred Read 0x0: no event	R	0
17	RESERVED	Reserved	R	0
16	TOUT_IS	Time-out overflow event Read 0x1: time-out occurred Read 0x0: no event	R	0
15:6	RESERVED	Reserved	R	0x000

Bits	Field Name	Description	Type	Reset
5	BLOCK_IE	Interrupt is sent when a full block is transferred 0x0: no interrupt 0x1: interrupt enable	RW	0
4:3	RESERVED	Reserved	R	0x0
2	HALF_IE	Interrupt is sent when a half block is transferred 0x0: no interrupt 0x1: interrupt enable	RW	0
1	RESERVED	Reserved	R	0
0	TOUT_IE	Interrupt is sent when a time-out overflow occurs 0x0: no interrupt 0x1: interrupt enable	RW	0

**Table 23-859. Register Call Summary for Register DMA\_CSR\_CCIR\_i**

## MIPI-HSI

- [Interrupt Requests: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)
- [DMA Module: \[13\]](#)
- [HSI Low-level Programming Models: \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)
- [HSI\\_DMA\\_CHANNELS registers: \[20\]](#)

**Table 23-860. DMA\_CSSA\_i**

<b>Address Offset</b>	0x0000 0008 + (0x40 * i)	<b>Index</b>	i = 0 to 15
<b>Physical Address</b>	Please refer to <a href="#">Table 23-855</a>	<b>Instance</b>	HSI_DMA_CHANNELS
<b>Description</b>	Stores source start address If the transfer configured to memory port all bits are taken into account as an address, if the transfer configured to peripheral port the lower four bits will determine the FIFO ID.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	source start address	RW	0x0000 0000

**Table 23-861. Register Call Summary for Register DMA\_CSSA\_i**

## MIPI-HSI

- [DMA Module: \[0\]](#)
- [HSI Low-level Programming Models: \[1\] \[2\]](#)
- [HSI\\_DMA\\_CHANNELS registers: \[3\]](#)



**Table 23-862. DMA\_CDSA\_i**

<b>Address Offset</b>	0x0000 000C + (0x40 * i)	<b>Index</b>	i = 0 to 15
<b>Physical Address</b>	Please refer to <a href="#">Table 23-855</a>	<b>Instance</b>	HSI_DMA_CHANNELS
<b>Description</b>	Stores destination start address If the transfer configured to memory port all bits are taken into account as an address, if the transfer configured to peripheral port the lower four bits will determine the FIFO ID.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	destination start address	RW	0x0000 0000

**Table 23-863. Register Call Summary for Register DMA\_CDSA\_i**

MIPI-HSI

- [DMA Module: \[0\]](#)
- [HSI Low-level Programming Models: \[1\] \[2\]](#)
- [HSI\\_DMA\\_CHANNELS registers: \[3\]](#)

**Table 23-864. DMA\_CEN\_i**

<b>Address Offset</b>	0x0000 0010 + (0x40 * i)	<b>Index</b>	i = 0 to 15
<b>Physical Address</b>	Please refer to <a href="#">Table 23-855</a>	<b>Instance</b>	HSI_DMA_CHANNELS
<b>Description</b>	Stores the number of 4 byte words in a DMA data block. Max is 65536.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SIZE																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	SIZE	Number of elements in a block. Max is 65536.	RW	0x0000

**Table 23-865. Register Call Summary for Register DMA\_CEN\_i**

MIPI-HSI

- [Interrupt Requests: \[0\]](#)
- [DMA Module: \[1\]](#)
- [HSI Low-level Programming Models: \[2\] \[3\]](#)
- [HSI\\_DMA\\_CHANNELS registers: \[4\]](#)

**Table 23-866. DMA\_CDAC\_CSAC\_i**

<b>Address Offset</b>	0x0000 0018 + (0x40 * i)	<b>Index</b>	i = 0 to 15
<b>Physical Address</b>	Please refer to <a href="#">Table 23-855</a>	<b>Instance</b>	HSI_DMA_CHANNELS
<b>Description</b>	Monitors the progress of DMA transfer, by storing the 16-bit counter address for source and destination.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS_CDAC																ADDRESS_CSAC															

Bits	Field Name	Description	Type	Reset
31:16	ADDRESS_CDAC	Destination address	R	0x0000
15:0	ADDRESS_CSAC	Source address	R	0x0000

**Table 23-867. Register Call Summary for Register DMA\_CDAC\_CSAC\_i**

MIPI-HSI

- [DMA Module: \[0\] \[1\]](#)
- [HSI\\_DMA\\_CHANNELS registers: \[2\]](#)

### 23.10.6.4 HSI\_PORTS Registers

#### 23.10.6.4.1 HSI\_PORTS Register Summary

**Table 23-868. HSI\_PORTS Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	HSI_PORTS Physical Address
<a href="#">HST_ID_Pp<sup>(1)</sup></a>	R	32	0x0000 0000 + (0x1000* b)	0x4A05 A000 + (0x1000* b)
<a href="#">HST_MODE_Pp<sup>(1)</sup></a>	RW	32	0x0000 0004 + (0x1000* b)	0x4A05 A004 + (0x1000* b)
<a href="#">HST_FRAME_SIZE_Pp<sup>(1)</sup></a>	R	32	0x0000 0008 + (0x1000* b)	0x4A05 A008 + (0x1000* b)
<a href="#">HST_TXSTATE_Pp<sup>(1)</sup></a>	R	32	0x0000 000C + (0x1000* b)	0x4A05 A00C + (0x1000* b)
<a href="#">HST_BUFSTATE_Pp<sup>(1)</sup></a>	R	32	0x0000 0010 + (0x1000* b)	0x4A05 A010 + (0x1000* b)
<a href="#">HST_DIVISOR_Pp<sup>(1)</sup></a>	RW	32	0x0000 0018 + (0x1000* b)	0x4A05 A018 + (0x1000* b)
<a href="#">HST_BREAK_Pp<sup>(1)</sup></a>	W	32	0x0000 0020 + (0x1000* b)	0x4A05 A020 + (0x1000* b)
<a href="#">HST_CHANNELS_Pp<sup>(1)</sup></a>	RW	32	0x0000 0024 + (0x1000* b)	0x4A05 A024 + (0x1000* b)
<a href="#">HST_ARBMODE_Pp<sup>(1)</sup></a>	RW	32	0x0000 0028 + (0x1000* b)	0x4A05 A028 + (0x1000* b)
<a href="#">HST_BUFFER_Pp_CHN_i<sup>(1)</sup></a>	W	32	0x0000 0080 + (0x1000* b) + (0x4 * i)	0x4A05 A080 + (0x1000* b) + (0x4 * i)
<a href="#">HST_SWAPBUFFER_Pp_CHN_i<sup>(1)</sup></a>	W	32	0x0000 00C0 + (0x1000* b) + (0x4 * i)	0x4A05 A0C0 + (0x1000* b) + (0x4 * i)
<a href="#">HST_MAPPINGf<sup>(1)</sup></a>	RW	32	0x0000 0100 + (0x4 * f)	0x4A05 A100 + (0x4 * f)
RESERVED	RW	32	0x0000 0144	0x4A05 A144

<sup>(1)</sup> p = 1 to 2  
i = 0 to 7  
f = 0 to 15  
b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)

**Table 23-868. HSI\_PORTS Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	HSI_PORTS Physical Address
HSR_ID_Pp <sup>(1)</sup>	R	32	0x0000 0800 + (0x1000* b)	0x4A05 A800 + (0x1000* b)
HSR_MODE_Pp <sup>(1)</sup>	RW	32	0x0000 0804 + (0x1000* b)	0x4A05 A804 + (0x1000* b)
HSR_FRAMESIZE_Pp <sup>(1)</sup>	R	32	0x0000 0808 + (0x1000* b)	0x4A05 A808 + (0x1000* b)
HSR_RXSTATE_Pp <sup>(1)</sup>	R	32	0x0000 080C + (0x1000* b)	0x4A05 A80C + (0x1000* b)
HSR_BUFSTATE_Pp <sup>(1)</sup>	R	32	0x0000 0810 + (0x1000* b)	0x4A05 A810 + (0x1000* b)
HSR_BREAK_Pp <sup>(1)</sup>	R	32	0x0000 081C + (0x1000* b)	0x4A05 A81C + (0x1000* b)
HSR_ERROR_Pp <sup>(1)</sup>	R	32	0x0000 0820 + (0x1000* b)	0x4A05 A820 + (0x1000* b)
HSR_ERRORACK_Pp <sup>(1)</sup>	W	32	0x0000 0824 + (0x1000* b)	0x4A05 A824 + (0x1000* b)
HSR_CHANNELS_Pp <sup>(1)</sup>	RW	32	0x0000 0828 + (0x1000* b)	0x4A05 A828 + (0x1000* b)
HSR_OVERRUN_Pp <sup>(1)</sup>	RW	32	0x0000 082C + (0x1000* b)	0x4A05 A82C + (0x1000* b)
HSR_OVERRUNACK_Pp <sup>(2)</sup>	R	32	0x0000 0830 + (0x1000* b)	0x4A05 A830 + (0x1000* b)
HSR_COUNTERS_Pp <sup>(2)</sup>	W	32	0x0000 0834 + (0x1000* b)	0x4A05 A834 + (0x1000* b)
HSR_BUFFER_Pp_CHN_i <sup>(2)</sup>	R	32	0x0000 0880 + (0x1000* b) + (0x4 * i)	0x4A05 A880 + (0x1000* b) + (0x4 * i)
HSR_SWAPBUFFER_Pp_CHN_i <sup>(2)</sup>	R	32	0x0000 08C0 + (0x1000* b) + (0x4 * i)	0x4A05 A8C0 + (0x1000* b) + (0x4 * i)
HSR_MAPPINGf <sup>(2)</sup>	RW	32	0x0000 0900 + (0x4 * f)	0x4A05 A900 + (0x4 * f)
HSR_DLL	RW	32	0x0000 0944	0x4A05 A944
RESERVED	RW	32	0x0000 0948	0x4A05 A948
HSR_DIVISOR_Pp <sup>(2)</sup>	RW	32	0x0000 094C + (0x1000* b)	0x4A05 A94C + (0x1000* b)

<sup>(2)</sup> p = 1 to 2  
i = 0 to 7  
f = 0 to 15  
b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)

**23.10.6.4.2 HSI\_PORTS Register Description**

**Table 23-869. HST\_ID\_Pp**

<b>Address Offset</b>	0x0000 0000 + (0x1000* b)	<b>Instance</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	legacy identification on port p		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_VAL																															

MIPI-HSI

www.ti.com

Bits	Field Name	Description	Type	Reset
31:0	ID_VAL	Legacy ID	R	Please refer to <sup>(1)</sup>

<sup>(1)</sup> TI Internal Data**Table 23-870. Register Call Summary for Register HST\_ID\_Pp**

MIPI-HSI

- [HSI\\_PORTS](#) registers: [0]

**Table 23-871. HST\_MODE\_Pp**

<b>Address Offset</b>	0x0000 0004 + (0x1000* b)		p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Defines operation mode and data flow on port p		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WAKE_CTRL	FLOW_VAL	MODE_VAL	

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4	WAKE_CTRL	Wake control 0x0: software control 0x1: automatic control	RW	0
3:2	FLOW_VAL	Flow 0x0: synchronized 0x1: pipelined 0x2: real-time	RW	0x0
1:0	MODE_VAL	Mode 0x0: sleep 0x1: stream 0x2: frame 0x3: reserved	RW	0x0

**Table 23-872. Register Call Summary for Register HST\_MODE\_Pp**

MIPI-HSI

- [HSR Module](#): [0]
- [HST Block](#): [1] [2] [3] [4]
- [HSI Low-level Programming Models](#): [5] [6] [7] [8] [9]
- [HSI\\_PORTS](#) registers: [10]

**Table 23-873. HST\_FRAME\_SIZE\_Pp**

<b>Address Offset</b>	0x0000 0008 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	This register is provided for legacy and possible future extension of protocol. Returns 0x1f		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SIZE_VAL															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4:0	SIZE_VAL	In the current implementation it must be always written 31, meaning the frame payload size is 32 bit. Read 0x1F: only possible value	R	0x1F

**Table 23-874. Register Call Summary for Register HST\_FRAME\_SIZE\_Pp**

- MIPI-HSI
- [HST Block: \[0\]](#)
  - [HSI\\_PORTS registers: \[1\]](#)

**Table 23-875. HST\_TXSTATE\_Pp**

<b>Address Offset</b>	0x0000 000C + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Define the state of the transmitter on port p		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TXSTATEVAL															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	TXSTATEVAL	State of the transmitter Read 0x0: idle Read 0x1: wait Read 0x2: transmit Read 0x3: start Read 0x4: break Read 0x5: reserved Read 0x6: reserved Read 0x7: reserved	R	0x0

**Table 23-876. Register Call Summary for Register HST\_TXSTATE\_Pp**

MIPI-HSI

- [HST Block: \[0\]](#)
- [HSI\\_PORTS registers: \[1\]](#)

**Table 23-877. HST\_BUFSTATE\_Pp**

<b>Address Offset</b>	0x0000 0010 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Transmitter state for FIFO 0..7 Each bit gives the state of transmit FIFOs: 0x1 FIFO full 0x0 FIFO not full		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSTATE_VAL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000000
7:0	BUFSTATE_VAL	Each bit gives the state of transmit buffer register: 0x0: buffer is not full 0x1: buffer is full	R	0x00

**Table 23-878. Register Call Summary for Register HST\_BUFSTATE\_Pp**

MIPI-HSI

- [HST Block: \[0\] \[1\] \[2\]](#)
- [HSI Low-level Programming Models: \[3\]](#)
- [HSI\\_PORTS registers: \[4\]](#)

**Table 23-879. HST\_DIVISOR\_Pp**

<b>Address Offset</b>	0x0000 0018 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Transmission bit rate divisor for port p		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TX_RATE_DIV_VAL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000000
7:0	TX_RATE_DIV_VAL	This bit field B has a range [0..255] and it programs a division factor [1..256], B+1 example: writing 0 divides by 1, writing 1 divides by 2, writing 2 divides by 3, etc	RW	0x00

**Table 23-880. Register Call Summary for Register HST\_DIVISOR\_Pp**

- MIPI-HSI
- [Clock Configuration](#): [0]
  - [HST Block](#): [1]
  - [HSI Low-level Programming Models](#): [2] [3]
  - [HSI\\_PORTS registers](#): [4]

**Table 23-881. HST\_BREAK\_Pp**

<b>Address Offset</b>	0x0000 0020 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Transmit break strobe register on port p		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BREAK_VAL															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	W	0x0000 0000
0	BREAK_VAL	Send a break signal on the port Write 0x0: no break Write 0x1: transmit break	W	0x0

**Table 23-882. Register Call Summary for Register HST\_BREAK\_Pp**

- MIPI-HSI
- [HST Block](#): [0] [1] [2]
  - [HSI Low-level Programming Models](#): [3]
  - [HSI\\_PORTS registers](#): [4]

**Table 23-883. HST\_CHANNELS\_Pp**

<b>Address Offset</b>	0x0000 0024 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Number of active channels on port p (this will determine the number of the used channel descriptor bits on the MIPI port as well) It can be 1,2,4,8 or 16.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CHAN_NUM_VAL															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4:0	CHAN_NUM_VAL	Number of active channels up to 16 0x1: One logical channel used, number of the channel descriptor bits is 0. 0x2: Two logical channels used, number of the channel descriptor bits is 1. 0x4: Four logical channels used, number of the channel descriptor bits is 2. 0x8: Eight logical channels used, number of the channel descriptor bits is 3. 0x10: Sixteen logical channels used, number of the channel descriptor bits is 4.	RW	0x01

**Table 23-884. Register Call Summary for Register HST\_CHANNELS\_Pp**

MIPI-HSI

- [HST Block: \[0\]](#)
- [HSI Low-level Programming Models: \[1\]](#)
- [HSI\\_PORTS registers: \[2\]](#)

**Table 23-885. HST\_ARBMODE\_Pp**

<b>Address Offset</b>	0x0000 0028 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Arbitration type for the transmit FIFOs on port p.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																ARB_VAL																

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	ARB_VAL	Mode 0x0: round robin 0x1: priority	RW	0

**Table 23-886. Register Call Summary for Register HST\_ARBMODE\_Pp**

MIPI-HSI

- [HST Block: \[0\] \[1\] \[2\]](#)
- [HSI Low-level Programming Models: \[3\]](#)
- [HSI\\_PORTS registers: \[4\]](#)



**Table 23-887. HST\_BUFFER\_Pp\_CHN\_i**

<b>Address Offset</b>	0x0000 0080 + (0x1000* b) + (0x4 * i)	<b>Index</b>	p = 1 to 2 i = 0 to 7 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Transmit register for FIFO 0..7 Important: Write access to the register with 0xC and 0xF byteen will load its value into the transmit FIFO. In case of using only 16 bit accesses the lower 2 bytes must be written first (byteen 0x3). Between two consecutive 16 bit FIFO related access there can not be any other FIFO related OCP access!		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	32 bits of data	W	0x0000 0000

**Table 23-888. Register Call Summary for Register HST\_BUFFER\_Pp\_CHN\_i**

- MIPI-HSI
- [DMA Module: \[0\]](#)
  - [HST Block: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)
  - [FIFO Subsystem: \[7\] \[8\] \[9\]](#)
  - [HSI Low-level Programming Models: \[10\] \[11\]](#)
  - [HSI\\_PORTS registers: \[12\]](#)

**Table 23-889. HST\_SWAPBUFFER\_Pp\_CHN\_i**

<b>Address Offset</b>	0x0000 00C0 + (0x1000* b) + (0x4 * i)	<b>Index</b>	p = 1 to 2 i = 0 to 7 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Transmit register with byte swapping for FIFO 0..7 Important: Write access to the register with 0x3 and 0xF byteen will load its value into the transmit FIFO. In case of using only 16 bit accesses the upper 2 bytes must be written first (byteen 0xC). Between two consecutive 16 bit FIFO related access there can not be any other FIFO related OCP access!		
<b>Type</b>	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAPDATA																															

Bits	Field Name	Description	Type	Reset
31:0	SWAPDATA	32 bits of byte-swapped data	W	0x0000 0000

**Table 23-890. Register Call Summary for Register HST\_SWAPBUFFER\_Pp\_CHN\_i**

- MIPI-HSI
- [HST Block: \[0\]](#)
  - [FIFO Subsystem: \[1\]](#)
  - [HSI\\_PORTS registers: \[2\]](#)

**Table 23-891. HST\_MAPPINGf**

<b>Address Offset</b>	0x0000 0100 + (0x4 * f)	<b>Index</b>	f = 0 to 15
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	TX FIFO configuration register. One register per FIFO		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																THRESHOLD				RESERVED	PORT_NUMBER	RESERVED	CH_NUMBER			ENABLE					

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x00000
13:10	THRESHOLD	Number of empty TX FIFO locations that will activate interrupt or DMA requests. DMA requests (if enabled) or interrupt assertion (if enabled) is triggered if the number of free locations in FIFO is less than the THRESHOLD value. 0x0: less than 8 words in the FIFO 0x1: less than 1 words in the FIFO 0x2: less than 2 words in the FIFO ..... 0x8: less than 8 words in the FIFO 0x9 through 0xF: no request generated	RW	0x0
9:8	RESERVED	Reserved	R	0x0
7	PORT_NUMBER	Associates the FIFO to a HSI port 0x0: port 1 0x1: port 2	RW	0
6:5	RESERVED	Reserved	R	0x0
4:1	CH_NUMBER	Associates the TX FIFO to a HSI logical channel number (0-15): 0x0: logical channel number 0 0x1: logical channel number 1 ..... 0xF: logical channel number 15	RW	0x0
0	ENABLE	Enables the FIFO 0x0: disabled 0x1: enabled	RW	1

**Table 23-892. Register Call Summary for Register HST\_MAPPINGf**

## MIPI-HSI

- [DMA Module: \[0\]](#)
- [HST Block: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [FIFO Subsystem: \[8\] \[9\] \[10\]](#)
- [HSI Low-level Programming Models: \[11\] \[12\] \[13\] \[14\]](#)
- [HSI\\_PORTS registers: \[15\]](#)

**Table 23-893. HSR\_ID\_Pp**

<b>Address Offset</b>	0x0000 0800 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Legacy Identification		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_VAL																															

Bits	Field Name	Description	Type	Reset
31:0	ID_VAL	Legacy ID	R	Please refer to <sup>(1)</sup>

<sup>(1)</sup> TI Internal Data

**Table 23-894. Register Call Summary for Register HSR\_ID\_Pp**

- MIPI-HSI
- [HSI\\_PORTS registers: \[0\]](#)

**Table 23-895. HSR\_MODE\_Pp**

<b>Address Offset</b>	0x0000 0804 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Defines operation mode and data flow on port p		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WAKE_STATUS	FLOW_VAL	MODE_VAL													

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4	WAKE_STATUS	Level of WAKE line	R	-
3:2	FLOW_VAL	Flow 0x0: synchronized 0x1: pipelined 0x2: real-time	RW	0x0
1:0	MODE_VAL	Mode 0x0: sleep 0x1: stream 0x2: frame 0x3: reserved	RW	0x0

**Table 23-896. Register Call Summary for Register HSR\_MODE\_Pp**

MIPI-HSI

- [HSR Module: \[0\] \[1\] \[2\] \[3\]](#)
- [HST Block: \[4\]](#)
- [HSI Low-level Programming Models: \[5\] \[6\]](#)
- [HSI\\_PORTS registers: \[7\]](#)

**Table 23-897. HSR\_FRAME\_SIZE\_Pp**

<b>Address Offset</b>	0x0000 0808 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	legacy returns 0x1f		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SIZE_VAL															

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4:0	SIZE_VAL	0x1F	R	0x1F

**Table 23-898. Register Call Summary for Register HSR\_FRAME\_SIZE\_Pp**

MIPI-HSI

- [HSR Module: \[0\]](#)
- [HSI\\_PORTS registers: \[1\]](#)

**Table 23-899. HSR\_RXSTATE\_Pp**

<b>Address Offset</b>	0x0000 080C + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Receiver state on port p		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RXSTATEVAL															

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	RXSTATEVAL	RX state Read 0x0: idle Read 0x1: receiving Read 0x2: finished Read 0x3: error Read 0x4: halt Read 0x5: timeout	R	0x0

**Table 23-900. Register Call Summary for Register HSR\_RXSTATE\_Pp**

MIPI-HSI

- [HSR Module: \[0\] \[1\]](#)
- [HSI\\_PORTS registers: \[2\]](#)

**Table 23-901. HSR\_BUFSTATE\_Pp**

<b>Address Offset</b>	0x0000 0810 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	State of receiver buffer register for port p		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSTATE_VAL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	BUFSTATE_VAL	Each bit gives the state of receiver buffer register: 0x0: buffer is not full 0x1: buffer is full	R	0x00

**Table 23-902. Register Call Summary for Register HSR\_BUFSTATE\_Pp**

MIPI-HSI

- [HSR Module: \[0\] \[1\] \[2\] \[3\]](#)
- [HSI Low-level Programming Models: \[4\]](#)
- [HSI\\_PORTS registers: \[5\]](#)

**Table 23-903. HSR\_BREAK\_Pp**

<b>Address Offset</b>	0x0000 081C + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Break detected on port p		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															BREAK_VAL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	BREAK_VAL	Break detected Read 0x0: No break detected Read 0x1: Break detected	R	0

**Table 23-904. Register Call Summary for Register HSR\_BREAK\_Pp**

## MIPI-HSI

- [Interrupt Requests: \[0\]](#)
- [HSR Module: \[1\]](#)
- [HSI\\_PORTS registers: \[2\]](#)

**Table 23-905. HSR\_ERROR\_Pp**

<b>Address Offset</b>	0x0000 0820 + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Error detection state register for port p		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	RESERVED	TME	RESERVED	RME	RESERVED	TBE	RESERVED	FTE	SIG						

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15	RESERVED	Reserved	R	0x0
14:12	RESERVED	Reserved	R	0x0
11	TME	Tx Mapping Error	R	0x0
10:8	RESERVED	Reserved	R	0x0
7	RME	Rx Mapping Error	R	0x0
6:5	RESERVED	Reserved	R	0x0
4	TBE	Tailing Bit Error	R	0x0
3:2	RESERVED	Reserved	R	0x0
1	FTE	Frame Timeout Error	R	0x0
0	SIG	signal error (legacy)	R	0x0

**Table 23-906. Register Call Summary for Register HSR\_ERROR\_Pp**

## MIPI-HSI

- [HSR Module: \[0\]](#)
- [Error Reporting: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [HSI Low-level Programming Models: \[10\] \[11\] \[12\] \[13\] \[14\]](#)
- [HSI\\_PORTS registers: \[15\]](#)

**Table 23-907. HSR\_ERRORACK\_Pp**

<b>Address Offset</b>	0x0000 0824 + (0x1000* b)	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b> HSI_PORTS
<b>Description</b>	Error detection acknowledge register on port p: write 1 to clear	
<b>Type</b>	W	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	RESERVED	TME	RESERVED	RME	RESERVED	TBE	RESERVED	FTE	SIG						

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	W	0x0000
15	RESERVED	Reserved	W	0x0
14:12	RESERVED	Reserved	W	0x0
11	TME	Tx Mapping Error	W	0x0
10:8	RESERVED	Reserved	W	0x0
7	RME	Rx Mapping Error	W	0x0
6:5	RESERVED	Reserved	W	0x0
4	TBE	Tailing Bit Error	W	0x0
3:2	RESERVED	Reserved	W	0x0
1	FTE	Frame Timeout Error	W	0x0
0	SIG	signal error (legacy)	W	0x0

**Table 23-908. Register Call Summary for Register HSR\_ERRORACK\_Pp**

MIPI-HSI

- [HSR Module: \[0\]](#)
- [Error Reporting: \[1\]](#)
- [HSI Low-level Programming Models: \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [HSI\\_PORTS registers: \[7\]](#)

**Table 23-909. HSR\_CHANNELS\_Pp**

<b>Address Offset</b>	0x0000 0828 + (0x1000* b)	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b> HSI_PORTS
<b>Description</b>	Number of active channels on port p (this will determine the number of the used channel descriptor bits on the MIPI port as well) It can be 1,2,4,8 or 16.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CHAN_NUM_VAL							

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4:0	CHAN_NUM_VAL	Number of active channels 0x1: one logical channel used, num. of the channel descriptor bits is 0 0x2: two logical channel used, num. of the channel descriptor bits is 1 0x4: four logical channel used, num. of the channel descriptor bits is 2 0x8: eight logical channel used, num. of the channel descriptor bits is 3 0x10: sixteen logical channel used, num. of the channel descriptor bits is 4	RW	0x01

**Table 23-910. Register Call Summary for Register HSR\_CHANNELS\_Pp**

MIPI-HSI

- [HSR Module: \[0\]](#)
- [HSI Low-level Programming Models: \[1\]](#)
- [HSI\\_PORTS registers: \[2\]](#)

**Table 23-911. HSR\_OVERRUN\_Pp**

<b>Address Offset</b>	0x0000 082C + (0x1000* b)	<b>Instance</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Overflow detection state register for those receive FIFOs which are mapped to port p.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OVERRUN_VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	OVERRUN_VAL	Bit n is set when overrun is detected on channel n Read 0x1: overrun detected Read 0x0: no overrun	R	0x0000

**Table 23-912. Register Call Summary for Register HSR\_OVERRUN\_Pp**

MIPI-HSI

- [HSR Module: \[0\]](#)
- [HSI\\_PORTS registers: \[1\]](#)



**Table 23-913. HSR\_OVERRUNACK\_Pp**

<b>Address Offset</b>	0x0000 0830 + (0x1000* b)	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b> HSI_PORTS
<b>Description</b>	Overflow acknowledge register for those receive FIFOs which are mapped to port p.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OVERRUNACK_VAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	OVERRUNACK_VAL	Bit n is set when overrun is acknowledged on channel n Write 0x0: no action Write 0x1: overrun acknowledge	W	0x0000

**Table 23-914. Register Call Summary for Register HSR\_OVERRUNACK\_Pp**

MIPI-HSI

- [HSI\\_PORTS registers: \[0\]](#)

**Table 23-915. HSR\_COUNTERS\_Pp**

<b>Address Offset</b>	0x0000 0834 + (0x1000* b)	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b> HSI_PORTS
<b>Description</b>	Counters setting register for port p	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB								TB								FT															

Bits	Field Name	Description	Type	Reset
31:24	FB	Setting for Frame Burst Counter. Set n as a value will result the n+1 counter value.	RW	0x00
23:20	TB	Setting for Tailing Bit Counter. Set n as a value will result the n+1 counter value.	RW	0x0
19:0	FT	Setting for Frame Timeout counter. Set n as a value will result the n+1 counter value.	RW	0x00000

**Table 23-916. Register Call Summary for Register HSR\_COUNTERS\_Pp**

MIPI-HSI

- [HSR Module: \[0\] \[1\] \[2\] \[3\] \[4\]](#)
- [HSI Low-level Programming Models: \[5\] \[6\] \[7\]](#)
- [HSI\\_PORTS registers: \[8\]](#)

**Table 23-917. HSR\_BUFFER\_Pp\_CHN\_i**

<b>Address Offset</b>	0x0000 0880 + (0x1000* b) + (0x4 * i)	<b>Index</b>	p = 1 to 2 i = 0 to 7 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Receive register for receive FIFO 0..7 Important: Read access to the register with 0xC and 0xF byteen will remove the receive FIFO related item. In case of using only 16 bit accesses the lower 2 bytes must be read first (byteen 0x3). Between two consecutive 16 bit FIFO related access there can not be any other FIFO related OCP access!		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Received data	R	0x-----

**Table 23-918. Register Call Summary for Register HSR\_BUFFER\_Pp\_CHN\_i**

## MIPI-HSI

- [DMA Module: \[0\]](#)
- [HSR Module: \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- [FIFO Subsystem: \[6\] \[7\] \[8\]](#)
- [HSI Low-level Programming Models: \[9\] \[10\]](#)
- [HSI\\_PORTS registers: \[11\]](#)

**Table 23-919. HSR\_SWAPBUFFER\_Pp\_CHN\_i**

<b>Address Offset</b>	0x0000 08C0 + (0x1000* b) + (0x4 * i)	<b>Index</b>	p = 1 to 2 i = 0 to 7 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Byte swapped receive register for receive FIFO 0..7 Important: Read access to the register with 0x3 and 0xF byteen will remove the receive FIFO related item. In case of using only 16 bit accesses the upper 2 bytes must be read first (byteen 0xC). Between two consecutive 16 bit FIFO related access there can not be any other FIFO related OCP access!		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAPDATA																															

Bits	Field Name	Description	Type	Reset
31:0	SWAPDATA	Received data swapped	R	0x-----

**Table 23-920. Register Call Summary for Register HSR\_SWAPBUFFER\_Pp\_CHN\_i**

## MIPI-HSI

- [HSR Module: \[0\] \[1\]](#)
- [FIFO Subsystem: \[2\]](#)
- [HSI\\_PORTS registers: \[3\]](#)

**Table 23-921. HSR\_MAPPINGf**

<b>Address Offset</b>	0x0000 0900 + (0x4 * f)	<b>Index</b>	f = 0 to 15
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	RX FIFO Configuration register. One register per FIFO		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WORDS						RESERVED	PORT_NUMBER	RESERVED	CH_NUMBER				ENABLE		

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x00000
13:10	WORDS	Number of words available in the RX FIFO for reading	R	0x0
9:8	RESERVED	Reserved	R	0x0
7	PORT_NUMBER	Associates the RX FIFO to a HSI port 0x0: port 1 0x1: port 2	RW	0x0
6:5	RESERVED	Reserved	R	0x0
4:1	CH_NUMBER	Associates RX FIFO to a HSI logical channel 0x0: logical channel number 0 0x1: logical channel number 1 ..... 0xF: logical channel number 15	RW	0x0
0	ENABLE	Enables or disables RX FIFO 0x0: disabled 0x1: enabled	RW	0x1

**Table 23-922. Register Call Summary for Register HSR\_MAPPINGf**

- MIPI-HSI
- [DMA Module: \[0\]](#)
  - [HSR Module: \[1\] \[2\] \[3\] \[4\] \[5\]](#)
  - [HST Block: \[6\]](#)
  - [FIFO Subsystem: \[7\] \[8\]](#)
  - [HSI Low-level Programming Models: \[9\] \[10\] \[11\] \[12\] \[13\]](#)
  - [HSI\\_PORTS registers: \[14\]](#)

**Table 23-923. HSR\_DLL**

<b>Address Offset</b>	0x0000 0944	<b>Index</b>	
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	DLL control: input code, change incoming, change allowed		
<b>Type</b>	RW		

MIPI-HSI

www.ti.com

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																INCO								RESERVED		COCHGR	RESERVED		DLL_NOT_BYPASSED	COCHRE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:8	INCO	Input Code	RW	0x000
7:5	RESERVED	Reserved	R	0x0
4	COCHGR	Code Change Grant	R	0x0
3:2	RESERVED	Reserved	R	0x0
1	DLL_NOT_BYPASSED	Turn off/on DLL usage 0x0: DLL is bypassed, internal clock tree delays have been used 0x1: DLL is used	RW	0x0
0	COCHRE	Code Change Request	RW	0x0

**Table 23-924. Register Call Summary for Register HSR\_DLL**

MIPI-HSI

- [Clock Configuration](#): [0] [1] [2] [3] [4] [5] [6] [7] [8]
- [HSR Module](#): [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19]
- [HSI\\_PORTS registers](#): [20]

**Table 23-925. HSR\_DIVISOR\_Pp**

<b>Address Offset</b>	0x0000 094C + (0x1000* b)	<b>Index</b>	p = 1 to 2 b = 0 to 1 (for p = 1, b = 0 and p = 2, b = 1)
<b>Physical Address</b>	Please refer to <a href="#">Table 23-868</a>	<b>Instance</b>	HSI_PORTS
<b>Description</b>	Receive bit rate divisor for port p. This needs to be set for correct protocol timing detection (tailing timeout, frame timeout).		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_RATE_DIV_VAL															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	RX_RATE_DIV_VAL	The functional clock source divided by this value will be used as a clock base for the receive counters. Set n as a value will result the n+1 divisor value.	RW	0x00

**Table 23-926. Register Call Summary for Register HSR\_DIVISOR\_Pp**

MIPI-HSI

- [Clock Configuration](#): [0]
- [HSR Module](#): [1] [2]
- [Error Reporting](#): [3]
- [HSI Low-level Programming Models](#): [4]
- [HSI\\_PORTS registers](#): [5]

## 23.11 High-Speed Multiport USB Host Subsystem

This section describes the high-speed (HS) universal serial bus (USB) host subsystem of the device.

### 23.11.1 High-Speed Multiport USB Host Subsystem Overview

---

**NOTE:** Copyright ©2004, 2005, 2006, 2007, 2008 Synopsys, Inc. All rights reserved. Used with permission.

---

The high-speed (HS) universal serial bus (USB) host subsystem consists of several ports and several modes per port. The modes are:

- UTMI+ low pin interface (ULPI)
- ULPI transceiverless link (TLL)
- High-speed interchip (HSIC)
- Serial
- Serial TLL

The HS USB host subsystem is composed of:

- HS multiport USB host controller module
- USBTLL module
- Two HSIC digital front end (DFE) modules

The HS USB host controller contains two quasi-independent host controllers that operate in parallel:

- The EHCI controller, based on the *Enhanced Host Controller Interface (EHCI) specification for USB Release 1.0*, is responsible for HS traffic (480 Mbps), over ULPI/USB 2.0 transceiver macrocell interface (UTMI)
- The OHCI controller, based on the *Open Host Controller Interface (OHCI) specification for USB Release 1.0a*, is responsible for full-speed (FS)/low-speed (LS) traffic (12/1.5 Mbps, respectively), over a serial interface

Each of the two external ports of the HS USB host controller is owned by one of the controllers (EHCI or OHCI) at a given time. Each port can work in several modes:

- When the port is owned by the OHCI (FS) host, the 6-pin internal serial interface is used.
- When the port is owned by the EHCI (HS) host, ULPI or UTMI is used.

The current subsystem does not support all dynamic USB speed negotiations, as expected from a standard USB host port at the same time. It supports:

- HS-only (with external HS physical layer [PHY]/HS TLL mode HSIC) on the EHCI
- FS-/LS-only (with external FS PHY/FS TLL mode) on the OHCI

If an FS peripheral is connected to an HS-port (to an HS ULPI PHY), the HS host subsystem negotiates for FS but cannot transfer control to the FS/LS host controller - the PHY interface is not available.

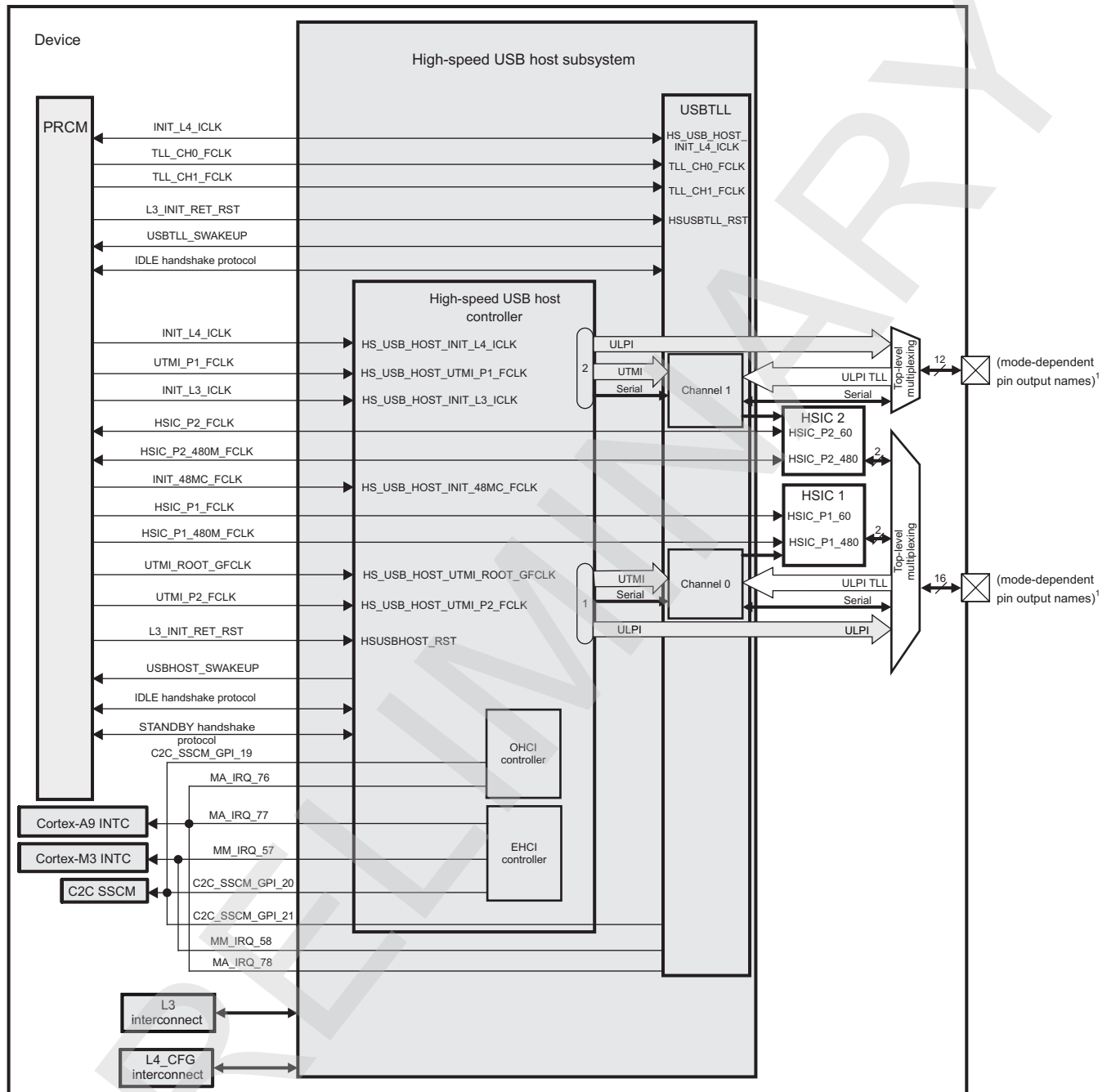
The USBTLL module is the adapter on the host ports. It consists of two channels that are seen as emulating a pair of back-to-back PHYs and has:

- UTMI port, connected to the host
- ULPI port, connected to a ULPI peripheral

The two HSIC DFE plugs into the UTMI port of the host and controls HSIC-capable bidirectional input/output (I/O) drivers driving the HSIC lines (strobe/data).

Figure 23-235 shows the HS USB host subsystem.

Figure 23-235. HS USB Host Controller



usb-006

(1) For more information about output pin names in different modes, see [Section 23.11.2.1, ULPIs](#); [Section 23.11.2.2, FS/LS Serial Interfaces](#); and [Section 23.11.2.3, HSIC Interface](#).

### 23.11.1.1 Main Features

The HS multiport USB host subsystem supports the following features:

- HS USB host controller:
  - One initiator on level 3 (L3)
  - One target on level 4 (L4)
  - Two interrupt requests (IRQs)

- No direct memory access (DMA) request
- Implements a USB 2.0 HS host controller:
  - HS (480 Mbps) communication with HS USB peripherals or hubs
  - FS (12 Mbps) communication with FS USB peripherals or hubs
  - LS (1.5 Mbps) communication with LS USB peripherals
- Support for suspend, resume, remote wakeup
- Two downstream ports (2-port root hub, with each port supporting UTMI, ULPI, and serial interface)
- Heterogeneous port configuration (that is, different HS/FS PHYs with different clocking methods on different ports, supported simultaneously)
- Single FS host controller, compliant with OHCI software application programming interface (API), per-port power switching support
- Interface with UTMI PHYs (transceivers) on all ports
  - 8-bit data path
  - Synchronous (on-chip) interface, independent clock input per port
  - HSIC front-end (after appropriate internal configuration)
- Interface with ULPI PHYs (transceivers) on all ports
  - 8-bit bidirectional data bus (SDR mode)
  - 12-pin interface
  - Independent clock input per port (from ULPI PHY)
  - Supports only HS data transactions

**CAUTION**

The HS USB host subsystem can only supports External Charge pump of PHY (no support of internal charge pump for ULPI PHY)

- Interconnect initiator for built-in DMA operation of EHCI/OHCI
- Interconnect target for configuration
- Link power management (LPM) support in OHCI and EHCI hosts
- Hardware save-and-restore (SAR) feature of the hardware context of the suspended host
- USBTLL module:
  - One interconnect target on L4
  - One IRQ
  - No DMA request
  - HS USB ports and each of them has four ports (A to D):
    - Port A: Host module UTMI+ port. Connected to the local link controller (host module). The UTMI "local" port is used in all configurations (that is, the entire channel can be seen as a protocol converted from that port to one of the other, "remote" ports. The port complies with UTMI+ version 1.0 and supports:
      - 8-data-bit, UTMI (HS/FS-capable)
      - UTMI+ L3 extensions
      - Vcontrol/Vstatus (from UTMI)
      - Serial FS 6-pin mode
    - Port B: Link-side UTMI+ port. Connected to HSIC front-end module.
    - Port C: PHY-side ULPI slave port. Connects to a remote (off-chip) ULPI link controller through I/O pads.
      - SDR ULPI mode (8-bit data width)
      - Supports optional input clocking mode
    - Port D: Serial multimode port.
      - Supports 6-pin unidirectional and 4-/3-/2-pin bidirectional modes
      - All modes are supported for TLL or PHY interface configuration, except 2-pin mode (that is,

TLL-only)

- Supports sideband signals (pullup/down control, speed/suspend enable, etc.)
- HSIC module: Converts UTMI into HSIC standard (2-wire TLL HS protocol)

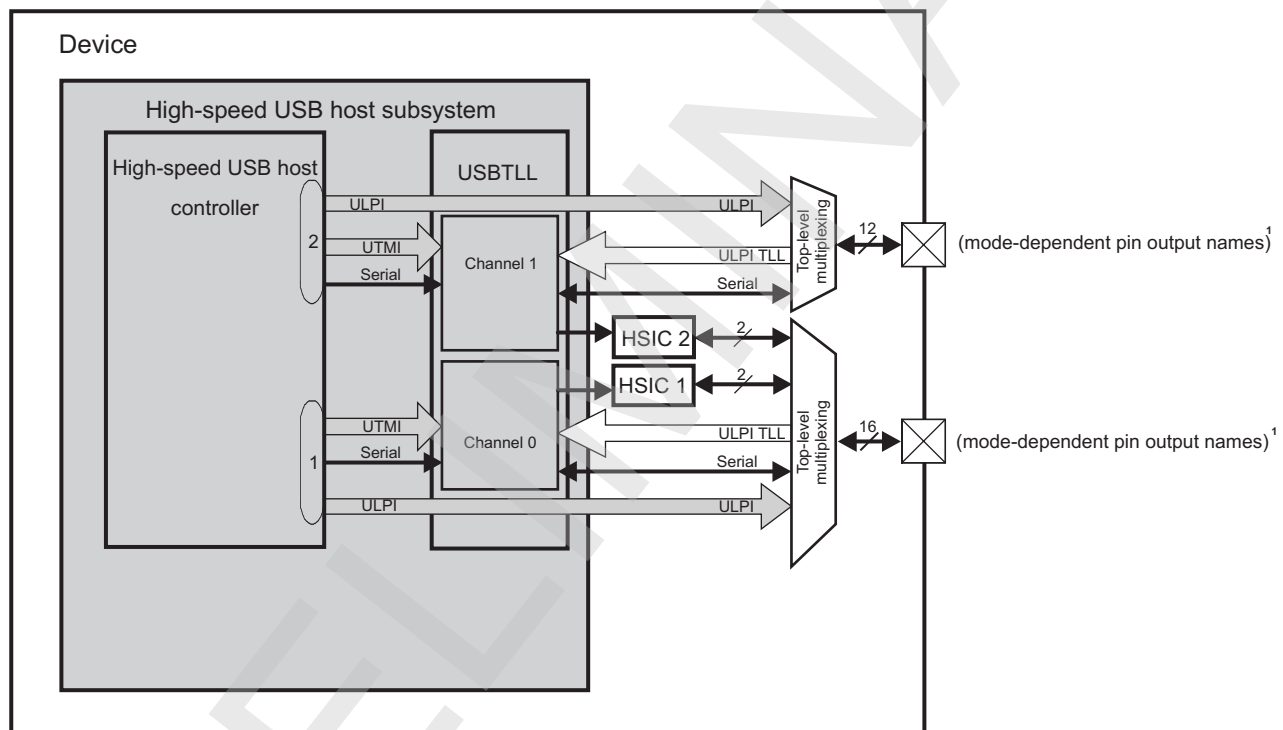
### 23.11.2 HS Multiport USB Host Subsystem Environment

The HS USB host subsystem provides three kinds of interfaces for connection:

- ULPis for HS data transactions (up to 480 Mbps) (see [Section 23.11.2.1](#), *ULPIs*)
- Serial interfaces, using the USBTLL module, for FS and LS data transactions (up to 12 Mbps) (see [Section 23.11.2.2](#), *FS/LS Serial Interfaces*)
- HSIC interface to two HSIC lines over HSIC-capable I/O drivers (see [Section 23.11.2.3](#), *HSIC Interface*)

[Figure 23-236](#) is an overview of the environment of the HS USB host subsystem.

**Figure 23-236. HS Multiport USB Host Subsystem Environment**



usb\_swpu140-072

- (1) For more information about output pin names in different modes, see [Section 23.11.2.1](#), *ULPIs*; [Section 23.11.2.2](#), *FS/LS Serial Interfaces*; and [Section 23.11.2.3](#), *HSIC Interface*.

**NOTE:** The two ports are configured independently and can be in different modes.

For more information how the pins are multiplexed, see [Section 18.4.8](#), *PAD Functional Multiplexing and Configuration*.

#### 23.11.2.1 ULPIs

The HS USB host subsystem supports the following configurations with the ULPIs:

- External USB transceiver:
  - ULPIs: 12-pin/8-bit data SDR version
- TLL:
  - SDR ULPI mode (12-pin/8-bit data width)



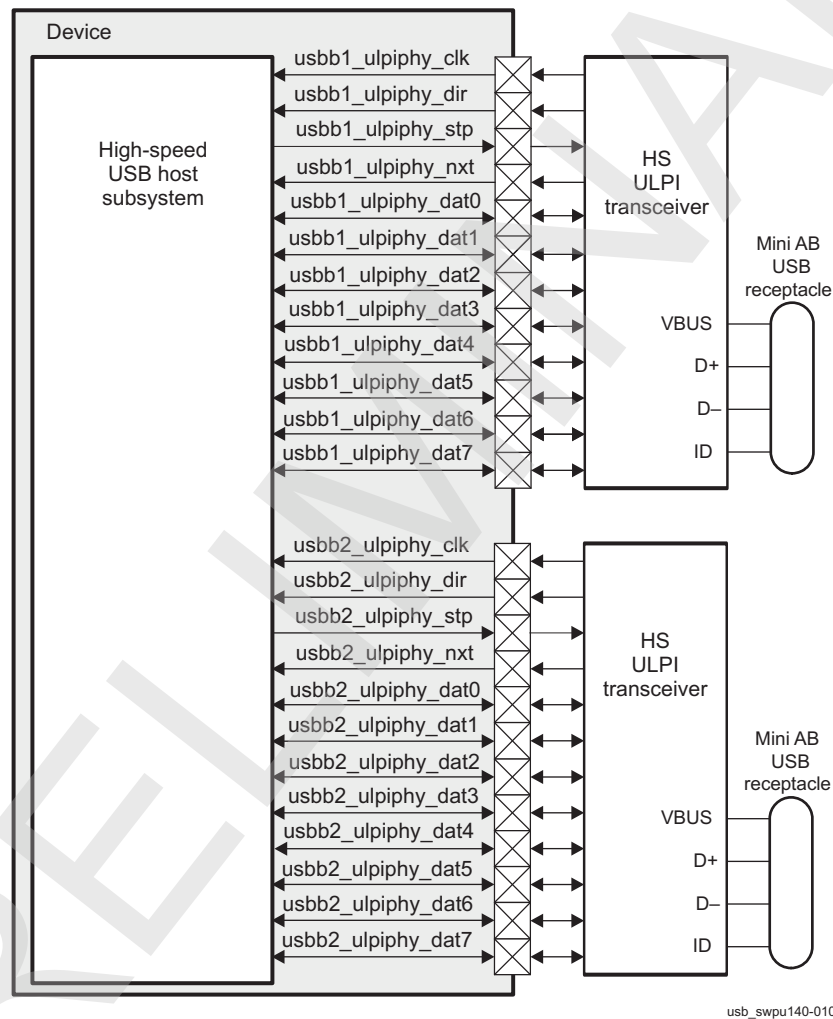
The HS USB host subsystem supports USB ports that use the ULPI mode to connect to an off-chip HS ULPI transceiver (12-pin/8-bit data SDR mode) for data transactions (up to 480 Mbps). FS and LS are not supported over the ULPI.

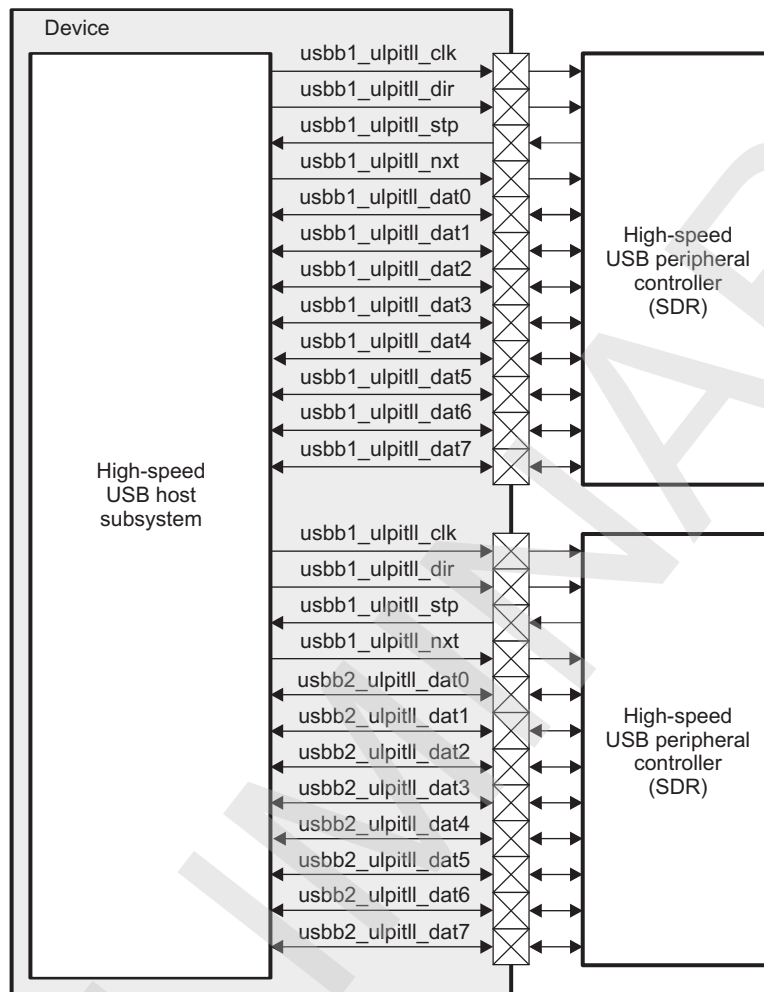
The device supports TLL logic interfaces on its ports in ULPI TLL interface mode. TLL modes enable glueless interconnect to another USB device port.

The external USB transceiver ULPIs and ULPI TLL interfaces cannot be used together on the same port. It is possible for one port to use ULPI and the other port to use ULPI TLL.

Figure 23-237 and Figure 23-238 show typical applications using the HS USB host subsystem with ULPI and the ULPI TLL interface, respectively.

Figure 23-237. External USB Transceiver ULPIs



**Figure 23-238. ULPI TLL Interfaces**

usb\_swpu140-032

The 12-pin ULPI uses an 8-bit data bus with data synchronous to the rising edge of the PHY (transceiver) clock (SDR mode).

#### 23.11.2.1.1 Transceiver Interface Configuration in ULPI Mode

The HS USB host subsystem supports only the 12-pin/8-bit data SDR version of the ULPI mode.

[Figure 23-239](#) shows USB ports using the 12-pin/8-bit data SDR version of the ULPI mode.

Figure 23-239. ULPIs - 12-Pin/8-Bit Data SDR Version

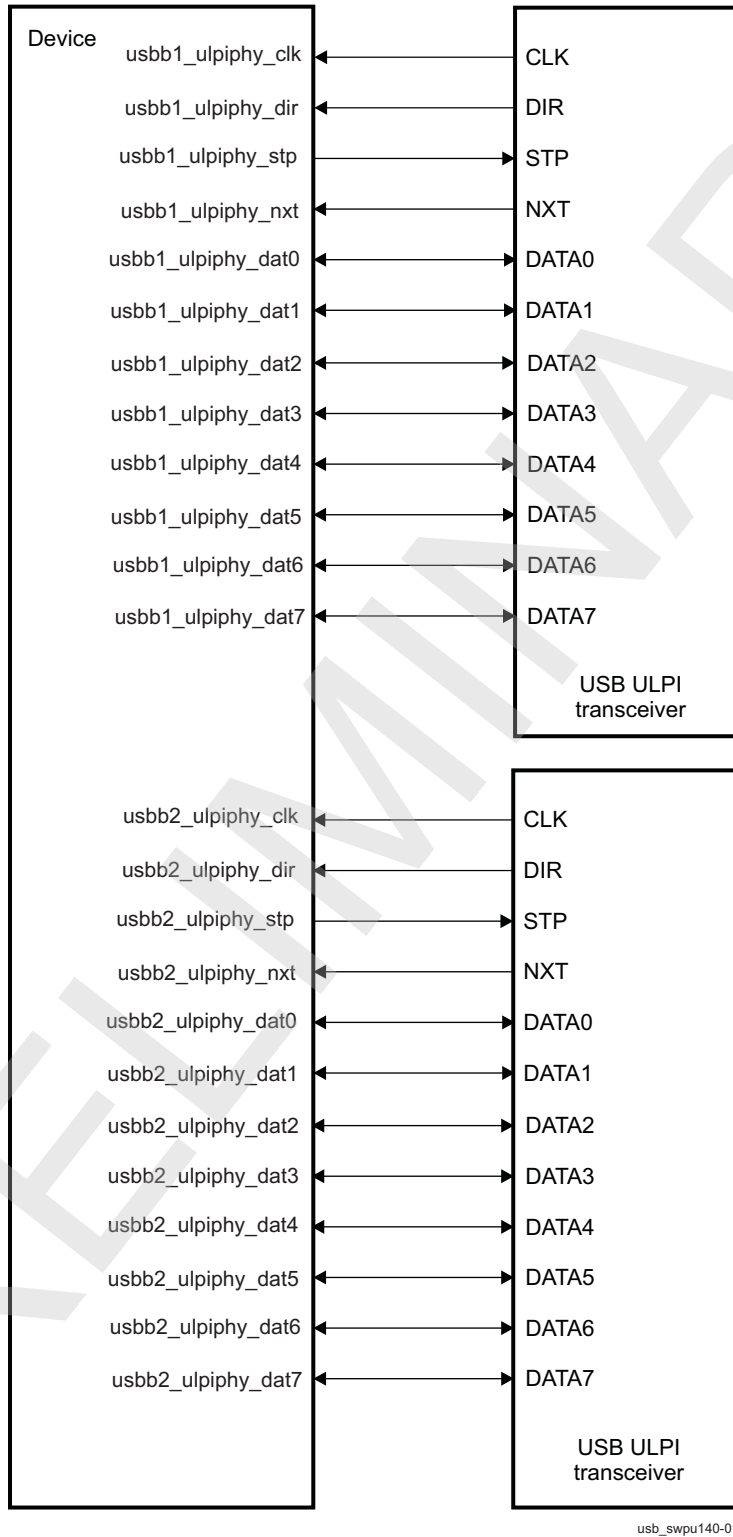


Table 23-927 describes the I/O pins of the USB ports using the 12-pin/8-bit data SDR version of the ULPI mode.

**Table 23-927. ULPI - 12-Pin/8-Bit Data SDR Version I/O Description**

Signal Name	I/O <sup>(1)</sup>	Description	Reset Value
<b>HS USB Host Controller Port 1</b>			
usbb1_ulpiphy_clk	I	Clock input from remote PHY IC <sup>(2)</sup>	0
usbb1_ulpiphy_dir	I	Data direction control from remote PHY IC	Unknown
usbb1_ulpiphy_stp	O	Stop signal to remote PHY IC	1
usbb1_ulpiphy_nxt	I	Next signal from remote PHY IC	Unknown
usbb1_ulpiphy_dat0	I/O	Bidirectional DATA0	Unknown
usbb1_ulpiphy_dat1	I/O	Bidirectional DATA1	Unknown
usbb1_ulpiphy_dat2	I/O	Bidirectional DATA2	Unknown
usbb1_ulpiphy_dat3	I/O	Bidirectional DATA3	Unknown
usbb1_ulpiphy_dat4	I/O	Bidirectional DATA4	Unknown
usbb1_ulpiphy_dat5	I/O	Bidirectional DATA5	Unknown
usbb1_ulpiphy_dat6	I/O	Bidirectional DATA6	Unknown
usbb1_ulpiphy_dat7	I/O	Bidirectional DATA7	Unknown
<b>HS USB Host Controller Port 2</b>			
usbb2_ulpiphy_clk	I	Clock input from remote PHY IC <sup>(2)</sup>	0
usbb2_ulpiphy_dir	I	Data direction control from remote PHY IC	Unknown
usbb2_ulpiphy_stp	O	Stop signal to remote PHY IC	1
usbb2_ulpiphy_nxt	I	Next signal from remote PHY IC	Unknown
usbb2_ulpiphy_dat0	I/O	Bidirectional DATA0	Unknown
usbb2_ulpiphy_dat1	I/O	Bidirectional DATA1	Unknown
usbb2_ulpiphy_dat2	I/O	Bidirectional DATA2	Unknown
usbb2_ulpiphy_dat3	I/O	Bidirectional DATA3	Unknown
usbb2_ulpiphy_dat4	I/O	Bidirectional DATA4	Unknown
usbb2_ulpiphy_dat5	I/O	Bidirectional DATA5	Unknown
usbb2_ulpiphy_dat6	I/O	Bidirectional DATA6	Unknown
usbb2_ulpiphy_dat7	I/O	Bidirectional DATA7	Unknown

<sup>(1)</sup> I = Input; O = Output

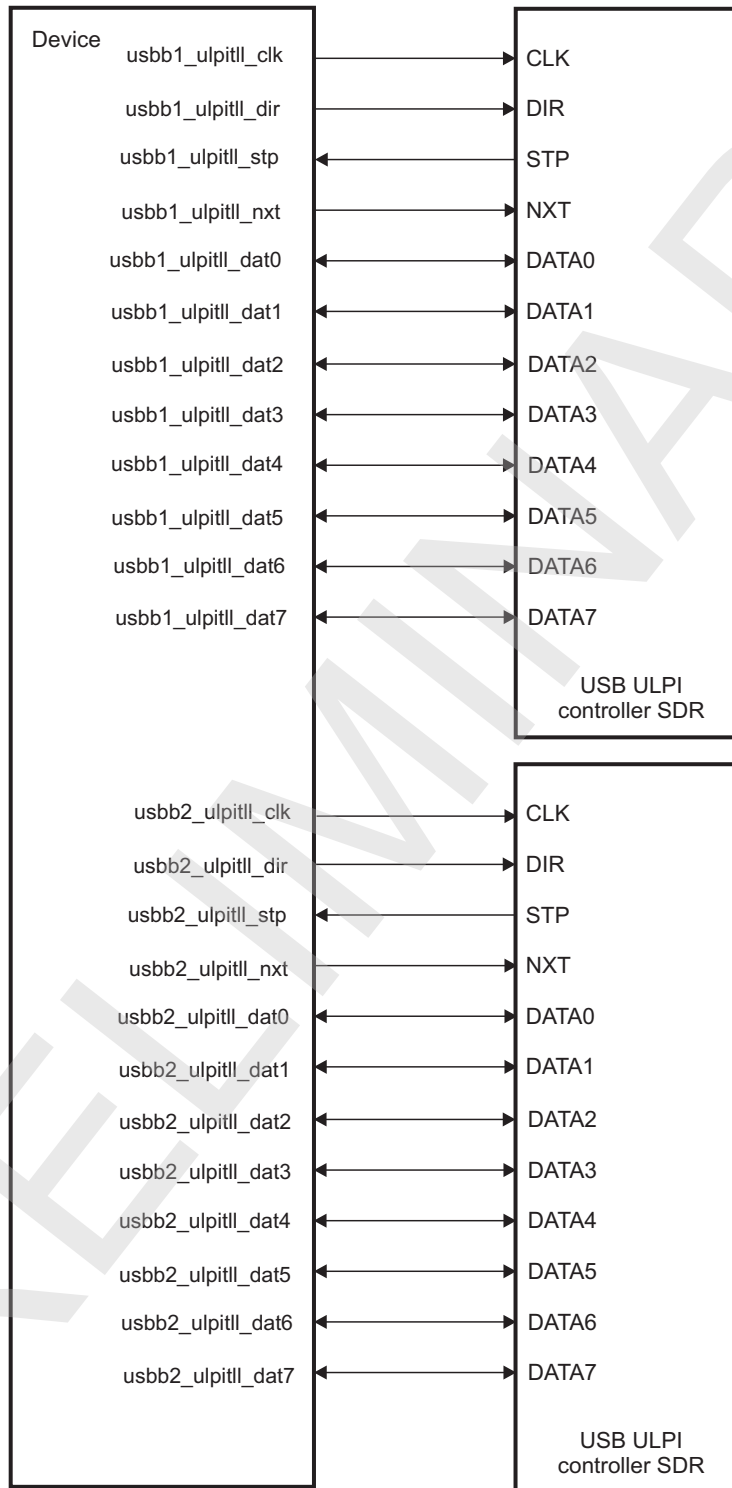
<sup>(2)</sup> This signal is also used as retiming input.

### 23.11.2.1.2 TLL Interface Configuration in ULPI Mode

The HS USB host controller is coupled with the USBTLL module to compose the ULPI TLL interface modes.

Figure 23-240 shows USB ports using the 12-pin/8-bit data SDR version of the ULPI TLL interface mode.

Figure 23-240. ULPI TLL Interfaces - 12-Pin/8-Bit Data SDR Version



usb\_swpu140-012

Table 23-928 describes the I/O pins of the USB ports using the 12-pin/8-bit data SDR version of the ULPI TLL interface mode.

**Table 23-928. ULPI TLL Interfaces - 12-Pin/8-Bit Data SDR Version I/O Description**

Signal Name	I/O <sup>(1)</sup>	Description	Reset Value
<b>HS USB TLL Port 1</b>			
usbb1_ulpitll_clk	O	Clock output to remote ULPI link controller <sup>(2)</sup>	0
usbb1_ulpitll_dir	O	Data direction control from remote ULPI link controller	0
usbb1_ulpitll_stp	I	Stop signal to remote ULPI link controller	Unknown
usbb1_ulpitll_nxt	O	Next signal from remote ULPI link controller	0
usbb1_ulpitll_dat0	I/O	Bidirectional DATA0	0
usbb1_ulpitll_dat1	I/O	Bidirectional DATA1	0
usbb1_ulpitll_dat2	I/O	Bidirectional DATA2	0
usbb1_ulpitll_dat3	I/O	Bidirectional DATA3	0
usbb1_ulpitll_dat4	I/O	Bidirectional DATA4	0
usbb1_ulpitll_dat5	I/O	Bidirectional DATA5	0
usbb1_ulpitll_dat6	I/O	Bidirectional DATA6	0
usbb1_ulpitll_dat7	I/O	Bidirectional DATA7	0
<b>HS USB TLL Port 2</b>			
usbb2_ulpitll_clk	O	Clock output to remote ULPI link controller <sup>(2)</sup>	0
usbb2_ulpitll_dir	O	Data direction control from remote ULPI link controller	0
usbb2_ulpitll_stp	I	Stop signal to remote ULPI link controller	Unknown
usbb2_ulpitll_nxt	O	Next signal from remote ULPI link controller	0
usbb2_ulpitll_dat0	I/O	Bidirectional DATA0	0
usbb2_ulpitll_dat1	I/O	Bidirectional DATA1	0
usbb2_ulpitll_dat2	I/O	Bidirectional DATA2	0
usbb2_ulpitll_dat3	I/O	Bidirectional DATA3	0
usbb2_ulpitll_dat4	I/O	Bidirectional DATA4	0
usbb2_ulpitll_dat5	I/O	Bidirectional DATA5	0
usbb2_ulpitll_dat6	I/O	Bidirectional DATA6	0
usbb2_ulpitll_dat7	I/O	Bidirectional DATA7	0

<sup>(1)</sup> I = Input; O = Output

<sup>(2)</sup> This signal is also used as retiming input.

### 23.11.2.2 FS/LS Serial Interfaces

The HS USB host subsystem supports the following configurations with the serial interfaces:

- External USB transceiver (see [Section 23.11.2.2.3, External USB Transceiver Interface Configurations in Serial Modes](#))
  - Serial 6-pin PHY (transceiver) interfaces: 6-pin mode (TX: DAT/SE0 or TX: DP/DM unidirectional mode), 4-pin mode (DP/DM bidirectional mode), and 3-pin mode (DAT/SE0 bidirectional mode)
- TLL (see [Section 23.11.2.2.4, TLL Interface Configurations in Serial Modes](#))
  - Serial 6-pin TLL interfaces: 6-pin mode (DAT/SE0 and DP/DM unidirectional modes), 4-pin mode (DP/DM bidirectional mode), 3-pin mode (DAT/SE0 bidirectional mode), and 2-pin mode (DAT/SE0 and DP/DM bidirectional modes)

#### CAUTION

Only FS and LS data transactions are possible in serial mode. Transceiver interface is serial (its frequency is that of the actual USB line) and combinatorial (no clock is passed).

Whether in TLL or external transceiver configuration, the serial interface follows the same principles: it is limited to FS/LS, and HS requires a parallel interface.

23.11.2.2.1 Encoding in Serial Mode

23.11.2.2.1.1 Unidirectional

When an external USB transceiver is connected to the device and is used in 6-pin unidirectional DAT/SE0 encoding mode, the encoding described in [Table 23-929](#) is used.

**Table 23-929. Signaling Between HS USB Host Subsystem and 6-Pin Unidirectional USB Transceiver (DAT/SE0 Signaling)**

Logical Signal Name	Device Pin Direction	Transceiver Pin Direction	Description				
TXEN	Output	Input	When low, the USB transceiver drives D+ and D-.				
DAT and SE0	Output	Input	Controls the values output by the USB transceiver on D+ and D- when TXEN is low; ignored when TXEN is high.				
			TXEN	DAT	SE0	D+	D-
			0	0	0	0	1
			0	1	0	1	0
			0	X	1	0	0
1	X	X	Undriven	Undriven			
RCV	Input	Output	Output from transceiver differential receiver				
			D+	D-	RCV		
			0	0	X		
			0	1	0		
			1	0	1		
1	1	X					
DP	Input	Output	Output from transceiver single-ended D+ signal receiver				
			D+	DP			
			0	0			
1	1						
DM	Input	Output	Output from transceiver single-ended D- signal receiver				
			D-	DM			
			0	0			
1	1						

When an external USB transceiver is connected to the device and is used in 6-pin unidirectional DP/DM encoding mode, the encoding described in [Table 23-930](#) is used.

**Table 23-930. Signaling Between HS USB Host Subsystem and 6-Pin Unidirectional USB Transceiver (DP/DM Signaling)**

Logical Signal Name	Device Pin Direction	Transceiver Pin Direction	Description				
TXEN	Output	Input	When low, the USB transceiver drives D+ and D-.				
DAT and SE0	Output	Input	Controls the values output by the USB transceiver on D+ and D- when TXEN is low; ignored when TXEN is high.				
			TXEN	DAT	SE0	D+	D-
			0	0	1	0	1
			0	1	0	1	0
			0	0	0	0	0
1	X	X	Undriven	Undriven			

**Table 23-930. Signaling Between HS USB Host Subsystem and 6-Pin Unidirectional USB Transceiver (DP/DM Signaling) (continued)**

Logical Signal Name	Device Pin Direction	Transceiver Pin Direction	Description		
RCV	Input	Output	Output from transceiver differential receiver		
			D+	D-	RCV
			0	0	X
			0	1	0
			1	0	1
1	1	X			
DP	Input	Output	Output from transceiver single-ended D+ signal receiver		
			D+	DP	
			0	0	
1	1				
DM	Input	Output	Output from transceiver single-ended D-signal receiver		
			D-	DM	
			0	0	
			1	1	

### 23.11.2.2.1.2 Bidirectional

When an external USB or USB On-The-GO (OTG) transceiver is connected to the device and is used in 3-pin bidirectional DAT/SE0 encoding mode, the encoding described in [Table 23-931](#) is used.

**Table 23-931. Signaling Between HS USB Host Subsystem and 3-Pin Bidirectional USB Transceiver Using DAT/SE0 Signaling**

Logical Signal Name	Device Pin Direction	Transceiver Pin Direction	Description				
TXEN	Output	Input	When low, the USB transceiver drives D+ and D-.				
DAT and SE0	Output	Input	When TXEN is low, the device drives DAT and SE0 and the transceiver drives D+ and D- based on the values of DAT and SE0.				
			TXEN	DAT	SE0	D+	D-
			0	0	0	0	1
		1	0	1	0		
		X	1	0	0		
	Input	Output	TXEN	D+	D-	DAT	SE0
			1	0	0	0	1
				0	1	0	0
				1	0	1	0
			1	1	Undefined	0	

**NOTE:** The device does not support 3-wire bidirectional signaling using DP/DM signals.

When an external USB or USB OTG transceiver is connected to the device and is used in 4-pin bidirectional DP/DM encoding mode, the encoding described in [Table 23-932](#) is used.

**Table 23-932. Signaling Between HS USB Host Subsystem and 4-Pin Bidirectional USB Transceiver Using DP/DM Signaling**

Logical Signal Name	Device Pin Direction	Transceiver Pin Direction	Description
TXEN	Output	Input	When low, the USB transceiver drives D+ and D-.



**Table 23-932. Signaling Between HS USB Host Subsystem and 4-Pin Bidirectional USB Transceiver Using DP/DM Signaling (continued)**

Logical Signal Name	Device Pin Direction	Transceiver Pin Direction	Description		
DM	Output	Input	Value driven to or received from D-		
			TXEN	DM	D-
			0	0	0
	0	1	1		
	Input	Output	TXEN	D-	DM
			1	0	0
1			1	1	
DP	Output	Input	Value driven to or received from D+		
			TXEN	DP	D+
			0	0	0
	0	1	1		
	Input	Output	TXEN	D+	DP
			1	0	0
1			1	1	
RCV	Input	Output	Output from transceiver differential receiver		
			D+	D-	RCV
			0	0	X
			0	1	0
			1	0	1
			1	1	X

**23.11.2.2.2 Sideband Signals for Serial Modes**

Serial interfaces carry only the USB data information. Sideband control and status (respectively, to/from the transceiver/TLL or to the bus lines themselves) require additional signals, which are usually implemented in a case-by-case, ad hoc way.

- Sideband control examples: FS/LS (slew rate control), transceiver suspend, connect (D+/D- pullup), pulldown enable, VBUS drive, etc.
- Sideband status example: VBUS level (VBUS valid, session valid, session end, etc.)
- Sideband signal implementations: Dedicated lines (one per sideband information bit), serial bus + interrupt line with register-mapped control/status (inter-integrated circuit [I<sup>2</sup>C™], universal asynchronous receiver/transmitter [UART], etc.)

Figure 23-241 and Figure 23-242 show system integration for sideband signals for two logically identical USB connections: one in transceiver configuration and one in TLL configuration. Although the sideband (purple) arrows are oriented from the controller to the transceiver in the two figures, the sideband information flow is bidirectional (that is, it flows from the controller to the transceiver [control] and from the transceiver to the controller [status]).

Figure 23-241 shows the transceiver configuration, where each side connects the sideband signals to its own transceiver. On the device (containing the USBTLL module), the sideband is decoded/re-encoded. The sideband signals available at the device boundary (and the USBTLL module) are decoded from the standard UTMI+ interface.

- The software-driven VBUS reporting procedure is described in Section 23.11.4.2.6.1.1, *VBUS Management in Serial Transceiver Configurations*.

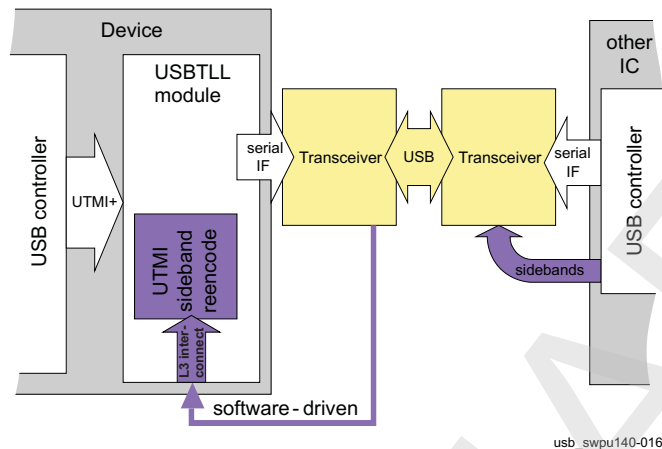
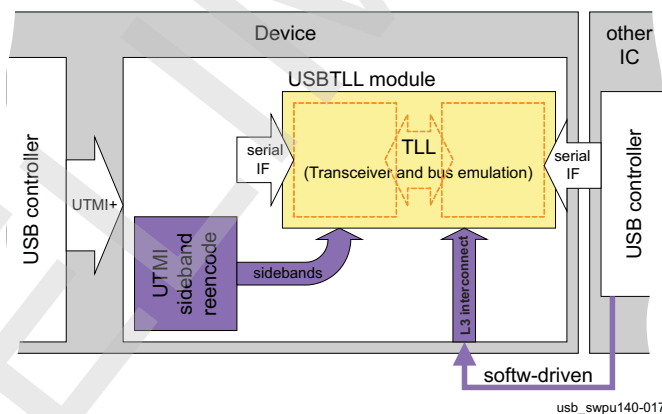
**Figure 23-241. Serial Interface Sideband Integration - Transceiver Configuration**

Figure 23-242 shows the TLL configuration, where both transceivers are emulated inside the USBTLL module.

- The transceiver of the local controller (left side of the figure) is working with the sideband information to/from the UTMI+ port. This is internal to the USBTLL module.
- The transceiver of the remote controller (right side of the figure) must communicate with its controller, which is on another IC. This is done in two ways:
  - The sideband input signals at the TLL module boundary (tlpven, tldrsvbus, tlsvbusvalid, etc.)
  - The software-driven VBUS control procedure described in [Section 23.11.4.2.6.2.2, VBUS Emulation in Serial TLL Modes](#))

**Figure 23-242. Serial Interface Sideband Integration - TLL Configuration**

### 23.11.2.2.3 External USB Transceiver Interface Configurations in Serial Modes

An external USB transceiver is required for each USB port used in the system. It converts between appropriate signaling for the HS USB host subsystem and appropriate signaling for the USB wire.

The serial interface mode of the HS USB host subsystem includes support for several types of USB transceivers. It provides signaling to up to two external USB transceivers.

Several types of external transceiver signaling are supported. Signaling between the HS USB subsystem in serial interface mode and the external USB transceiver for monitoring and controlling the differential USB signal can be done through a 6-, 4-, or 3-wire signaling interface, with two or more control signals provided by additional signals or through an I<sup>2</sup>C link.

The following subsections describe the transceiver interface modes supported by the HS USB host subsystem in the serial interface mode. In each case, the subsystem is connected to external transceivers, on the other side of the USB lines (D+/D-).

**23.11.2.2.3.1 Unidirectional Transceiver Interface Modes: 6-Pin**

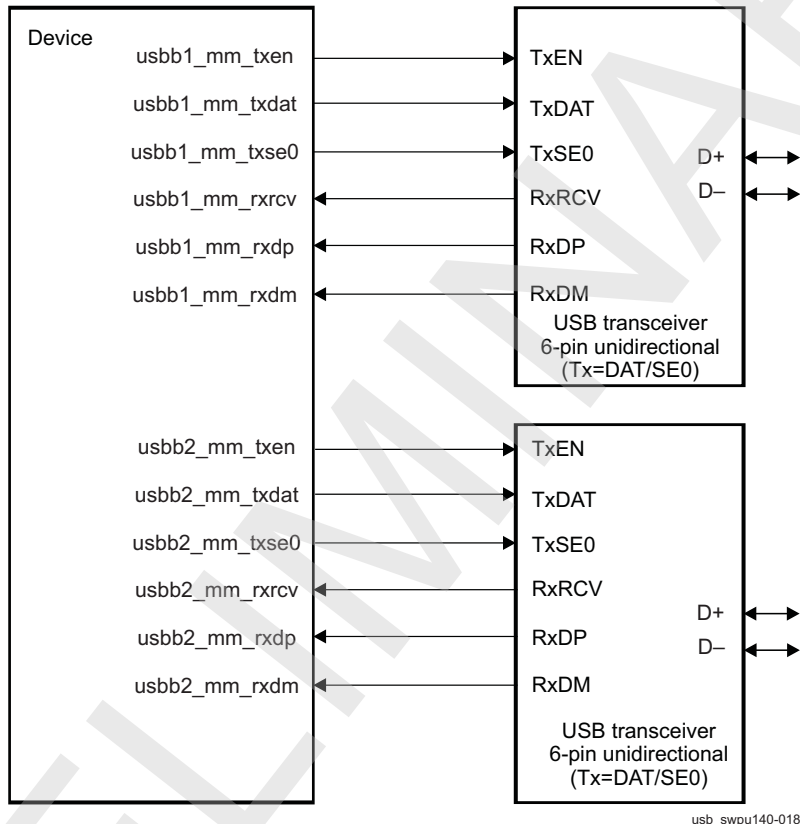
The 6-pin modes are the "natural" transceiver interface modes for the FS transceivers in the sense that they mirror the internal makeup of the transceivers.

Two encodings exist for TX: DAT/SE0 and DP/DM.

When an external USB is connected to the device and is used in 6-pin unidirectional DAT/SE0 signaling mode, the signaling described in [Table 23-929](#) is used.

[Figure 23-243](#) shows a USB port using DAT/SE0 encoding.

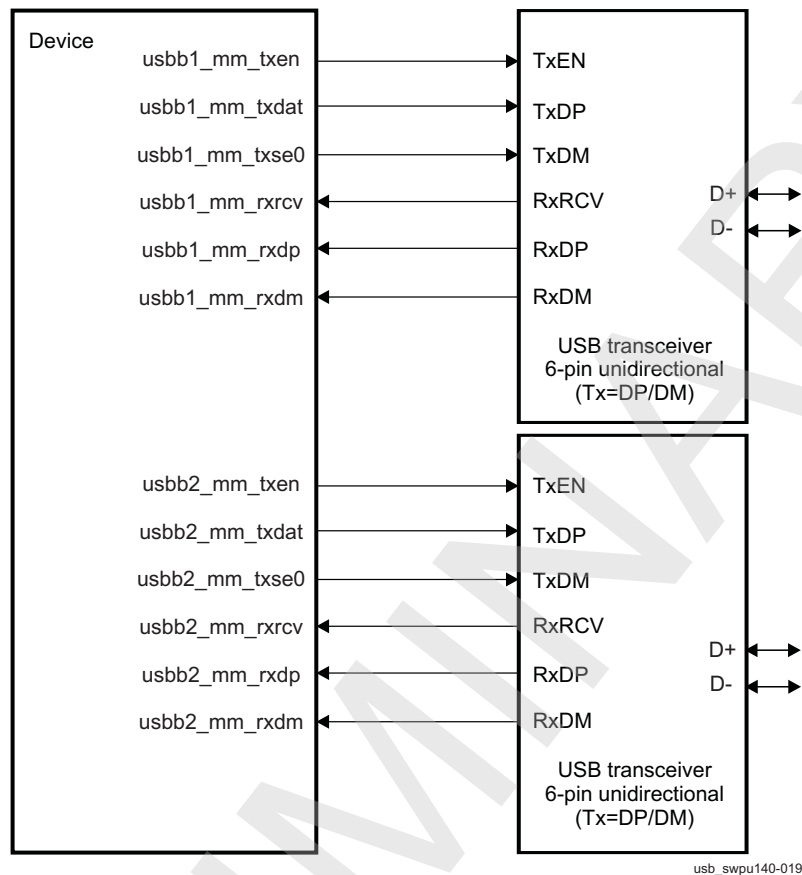
**Figure 23-243. 6-Pin Unidirectional Using DAT/SE0 Signaling**



usb\_swpu140-018

When a USB is connected to the device and is used in 6-pin unidirectional DP/DM signaling mode, the signaling described in [Table 23-930](#) is used.

[Figure 23-244](#) shows a USB port using DP/DM encoding.

**Figure 23-244. 6-Pin Unidirectional Using DP/DM Signaling**

### 23.11.2.2.3.2 Bidirectional Transceiver Interface Modes: 3-Pin, 4-Pin

The bidirectional transceiver interface modes are pin-count optimizations of the unidirectional modes. They take advantage of the fact that a USB port is sending or receiving at any given time, but never both. The TX and RX paths of the unidirectional mode can be multiplexed on bidirectional lines. To prevent glitches at TX/RX turnaround, the same encoding is used for both directions (DAT/SE0 or DP/DM).

The signaling listed in [Table 23-931](#) is used when a USB transceiver is connected to the device and is used in 3-pin bidirectional DAT/SE0 signaling mode.

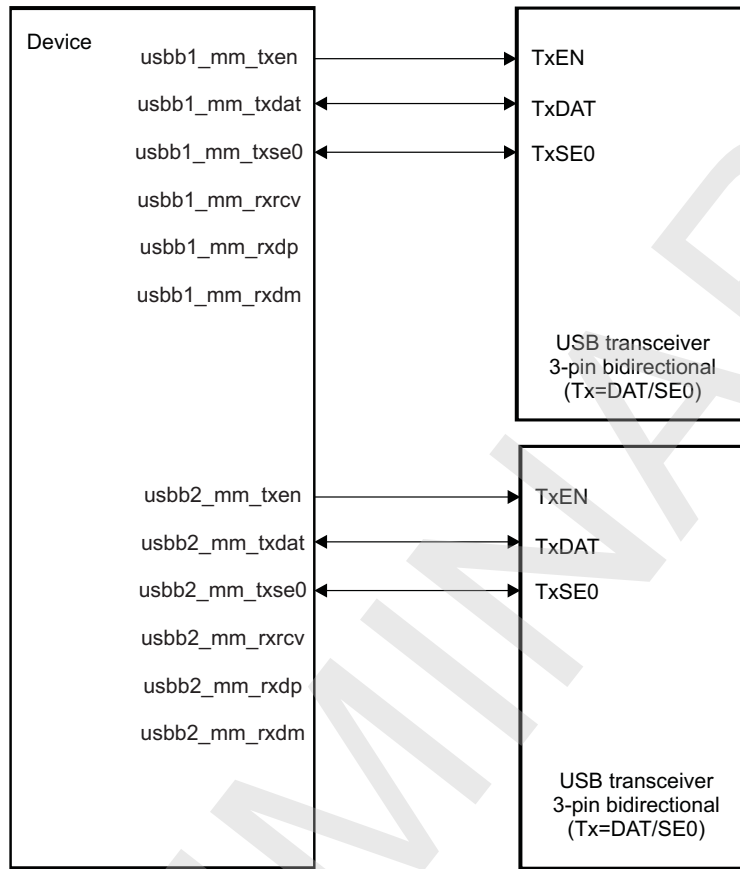
---

**NOTE:** The device does not support 3-wire bidirectional signaling using DP/DM signals.

---

[Figure 23-245](#) shows a USB port using DAT/SE0 encoding.

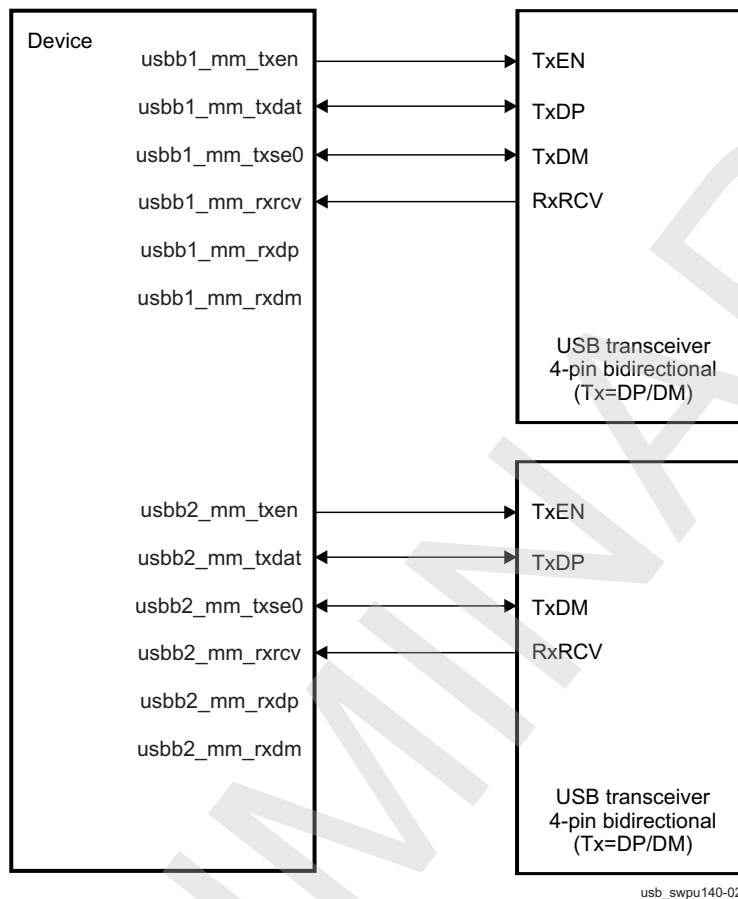
**Figure 23-245. 3-Pin Bidirectional Using DAT/SE0 Signaling**



usb\_swpu140-020

The signaling listed in [Table 23-932](#) is used when a USB transceiver is connected to the device and is used in 4-pin bidirectional DP/DM signaling mode.

[Figure 23-246](#) shows a USB port using DP/DM encoding.

**Figure 23-246. 4-Pin Bidirectional Using DP/DM Signaling**

#### 23.11.2.2.4 TLL Interface Configurations in Serial Modes

The HS USB host subsystem supports unidirectional and bidirectional TLL logic interfaces on its ports. The TLL modes enable glueless interconnect to the USB device port of another device without needing a costly transceiver.

Serial interface modes are FS or LS only. Transceiver interface is serial (its frequency is that of the actual USB line) and combinatorial (no clock is passed).

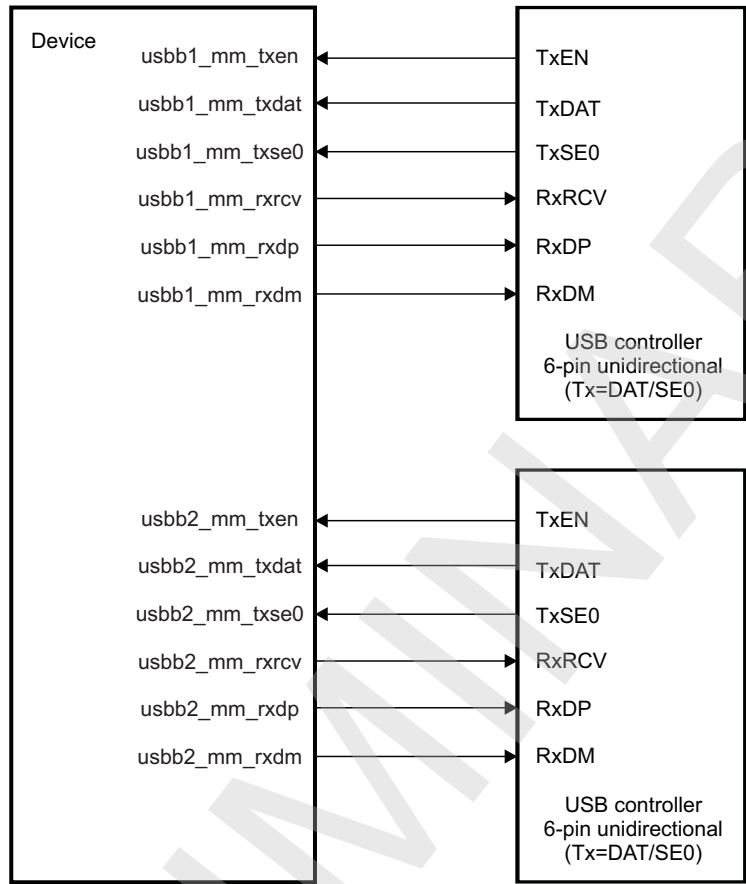
##### 23.11.2.2.4.1 Unidirectional TLL Serial Modes

The 6-pin TLL configurations are mirror images of the 6-pin transceiver configurations previously discussed. The same signals are mapped on the same physical pins, but in opposite directions.

Two possible modes exist, depending on the TX data encoding used by the external device.

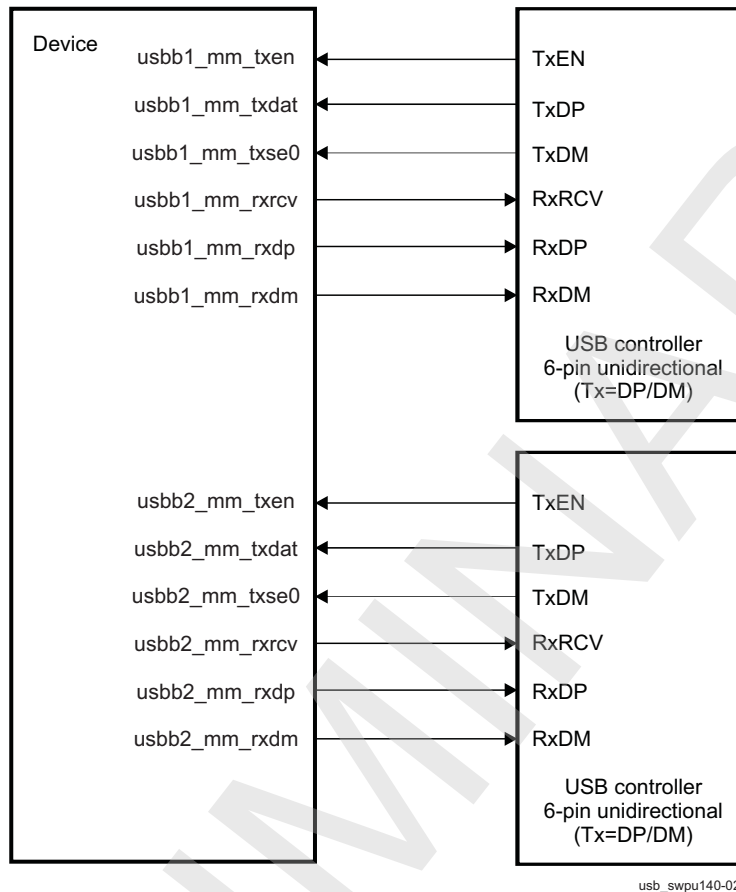
[Figure 23-247](#) shows an external device using DAT/SE0 encoding.

Figure 23-247. 6-Pin Unidirectional TLL Using DAT/SE0 Signaling



usb\_swpu140-022

Figure 23-248 shows an external device using DP/DM encoding.

**Figure 23-248. 6-Pin Unidirectional TLL Using DP/DM Signaling****23.11.2.2.4.2 Bidirectional TLL Serial Modes**

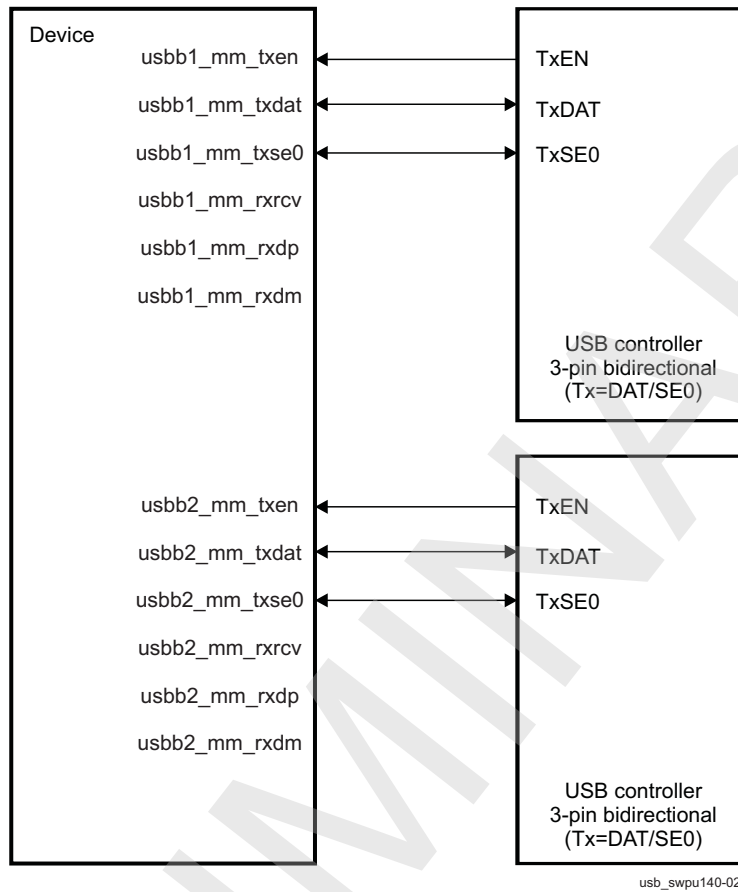
The 3-pin/4-pin TLL configurations are mirror images of the 3-pin/4-pin transceiver configurations previously discussed. The same signals are mapped on the same physical pins, but in opposite directions (bidirectional lines remain bidirectional).

Two possible modes exist, depending on the TX data encoding used by the external device.

[Figure 23-249](#) shows an external device using DAT/SE0 encoding.

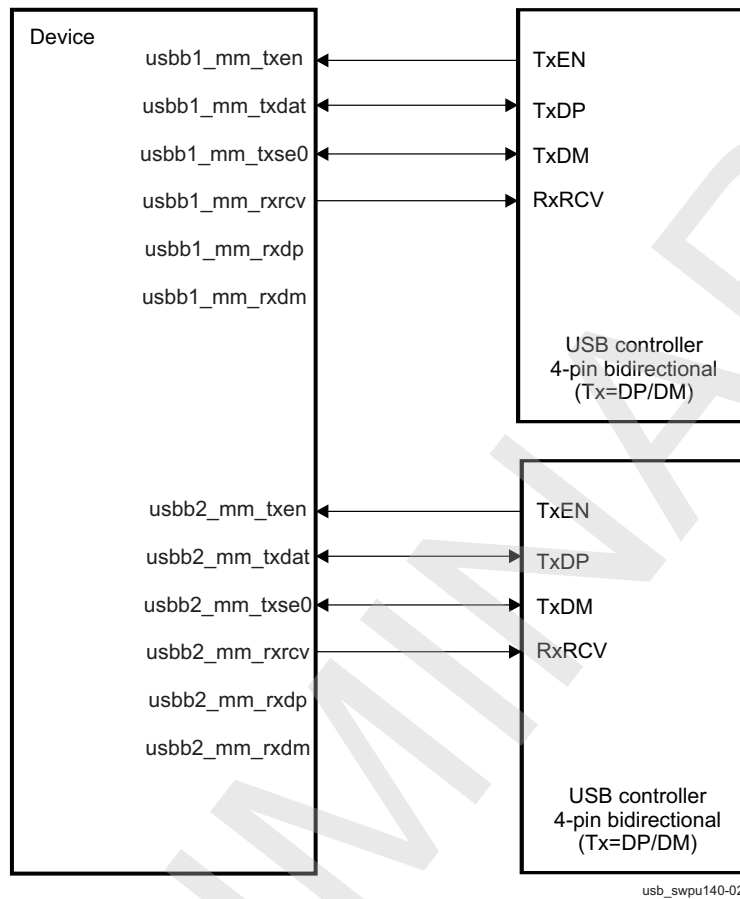


**Figure 23-249. 3-Pin Bidirectional TLL Using DAT/SE0 Signaling**



usb\_swpu140-024

Figure 23-250 shows an external device using DP/DM encoding.

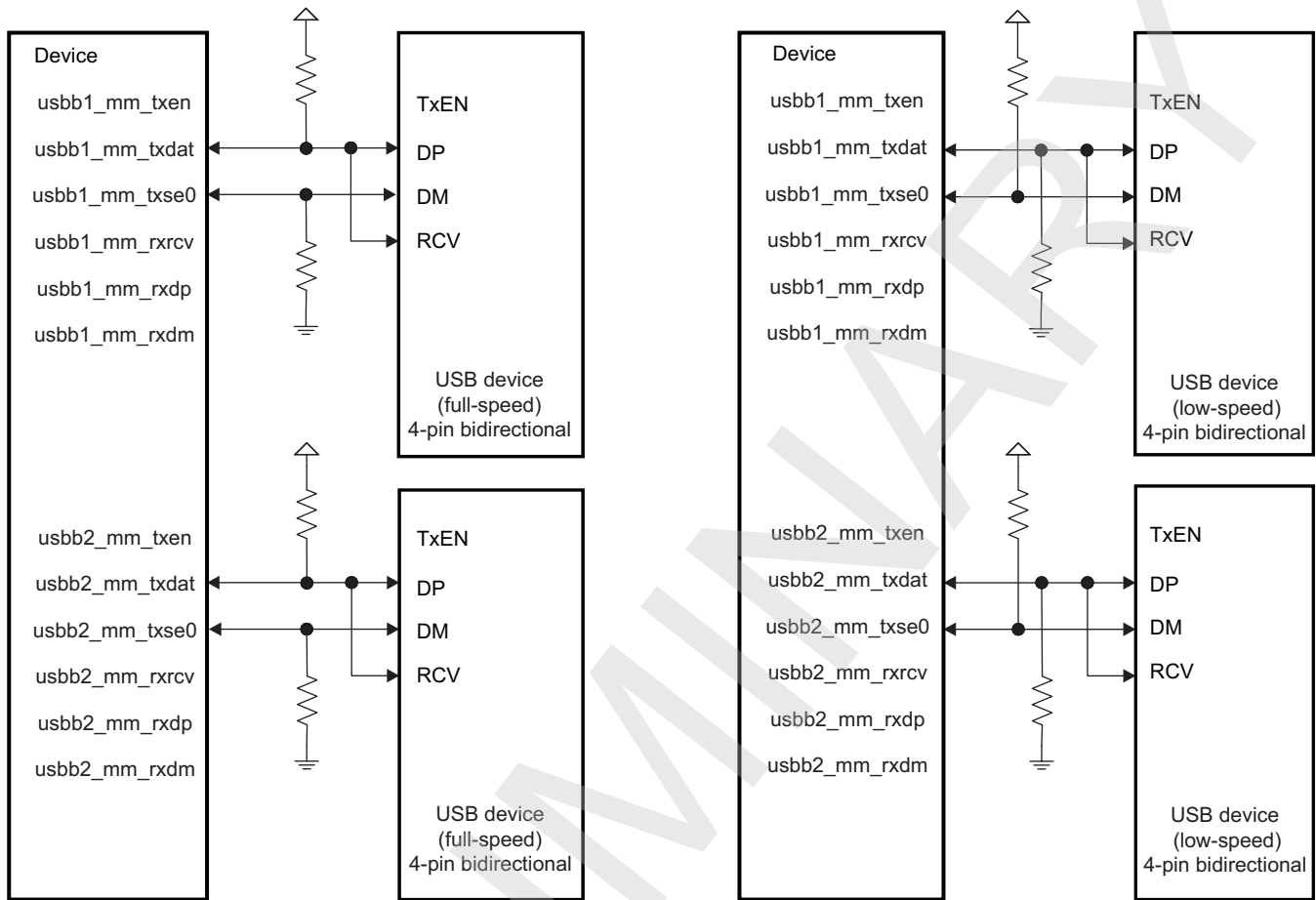
**Figure 23-250. 4-Pin Bidirectional TLL Using DP/DM Signaling**

The 2-pin TLL configurations have unique specifications:

- They require pullups/pulldowns to operate, because the bidirectional lines are not driven at all times like the other serial transceiver interfaces described previously. The connection of pull resistors depends on the speed of the controller.
- The module supports explicit 2-pin TLL modes, with DAT/SE0 or DP/DM encoding.
- Non-TLL modes (that is, transceiver configuration mode) can be used to implement the 2-pin functionality, using a specific connectivity.

Figure 23-251 shows a USB port using DP/DM encoding.

Figure 23-251. 2-Pin Bidirectional TLL Using DP/DM Encoding, With 4-Pin Bidirectional USB Device



usb\_swpu140-026

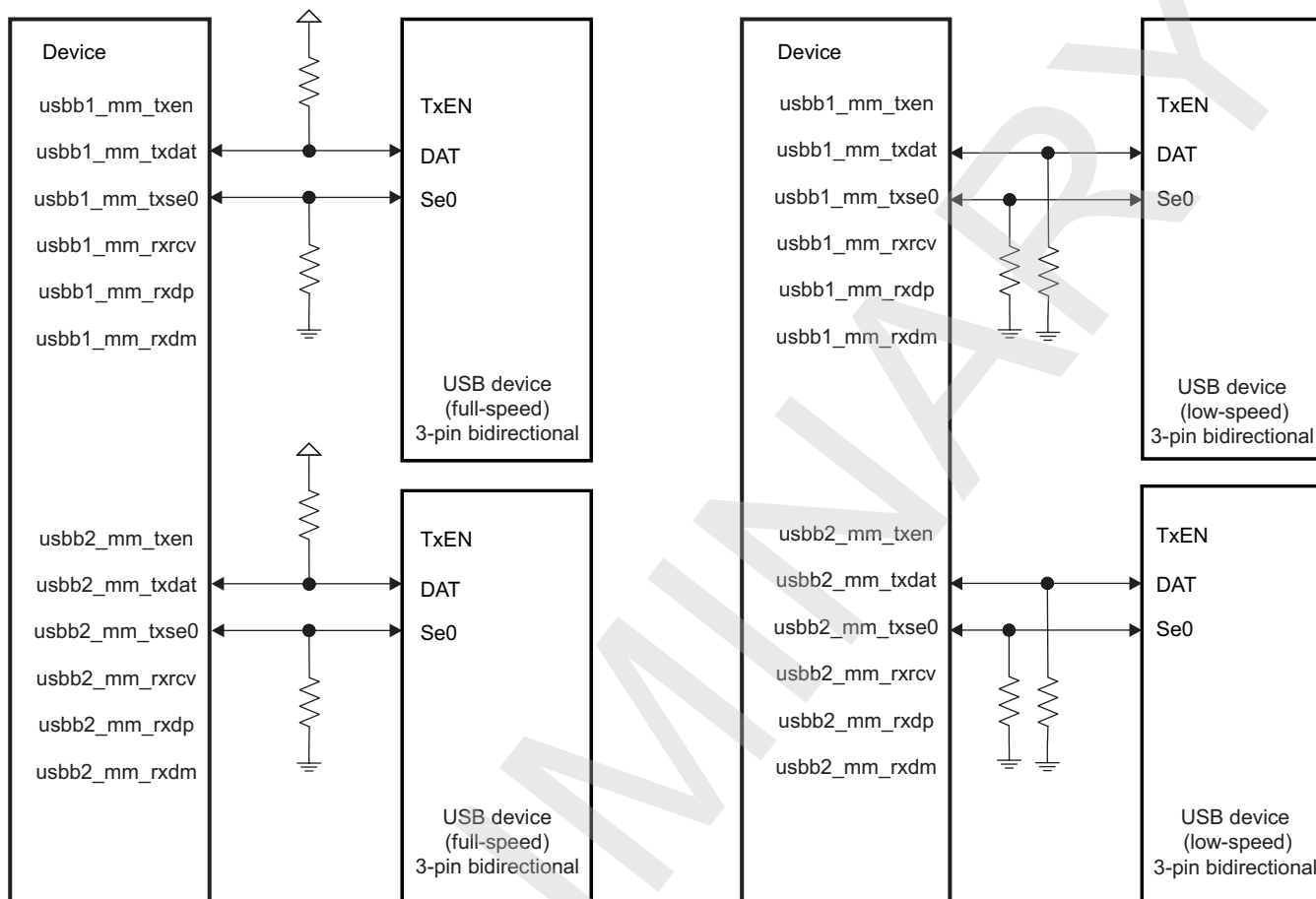
Table 23-933 shows the pullup/pulldown configuration for DP/DM encoding.

Table 23-933. Pullup/Pulldown Configuration for DP/DM Encoding

	Nonconnected Device (Any Speed)	Connected LS Device	Connected FS Device
DP	Pulldown	Pulldown	Pullup
DM	Pulldown	Pullup	Pulldown

Figure 23-252 shows a USB port using DAT/SE0 encoding.

**Figure 23-252. 2-Pin Bidirectional TLL Using DAT/SE0 Encoding, With 3-Pin Bidirectional USB Device**



usb\_swpu140-027

Table 23-934 shows pullup/pulldown configuration for DAT/SE0 encoding.

**Table 23-934. Pullup/Pulldown Configuration for DAT/SE0 Encoding**

	Nonconnected Device (Any Speed)	Connected LS Device	Connected FS Device
DAT	Pulldown	Pulldown	Pullup
SE0	Pullup	Pulldown	Pulldown

**23.11.2.2.5 Interface Description in Serial Modes**

Table 23-935 describes the I/O of the HS USB host subsystem serial interfaces.

**Table 23-935. I/O Description**

Signal Name	I/O <sup>(1)</sup>	Description	Value at Reset
<b>Multiple-Mode FS/LS Serial Interface: Port 1</b>			
usbb1_mm_txse0	I/O	SE0 function in 3-pin bidirectional DAT/SE0 mode	0
	I/O	DM function in 4-pin bidirectional DP/DM mode	
	O	SE0 output in 6-pin unidirectional DAT/SE0 mode	
	O	DM output in 6-pin unidirectional DP/DM mode	
	I/O	SE0-TLL in 2-/3-pin bidirectional DAT/SE0 TLL mode	
	I/O	DM-TLL in 2-/4-pin bidirectional DP/DM TLL mode	
	I	SE0-TLL input in 6-pin unidirectional DAT/SE0 TLL mode	
	I	DM-TLL input in 6-pin unidirectional DP/DM TLL mode	
usbb1_mm_txdat	I/O	DAT function in 3-pin bidirectional DAT/SE0 mode	Unknown
	I/O	DP function in 4-pin bidirectional DP/DM mode	
	O	DAT output in 6-pin unidirectional DAT/SE0 mode	
	O	DP output in 6-pin unidirectional DAT/SE0 mode	
	I/O	DAT-TLL in 2-/3-pin bidirectional DAT/SE0 TLL mode	
	I/O	DP-TLL in 2-/4-pin bidirectional DP/DM TLL mode	
	I	DAT-TLL input in 6-pin unidirectional DAT/SE0 TLL mode	
	I	DP-TLL input in 6-pin unidirectional DP/DM TLL mode	
usbb1_mm_txen	O	Transmit enable in 3-pin bidirectional DAT/SE0 or 4-pin bidirectional DP/DM or 6-pin unidirectional modes	1
	I	Transmit enable in 3-pin bidirectional DAT/SE0 TLL or 4-pin bidirectional DP/DM TLL or 6-pin unidirectional TLL modes (not used in 2-pin bidirectional TLL modes)	
usbb1_mm_rxcv	I	Differential receiver signal input in 4-pin bidirectional DP/DM or 6-pin unidirectional modes (not used in 3-pin bidirectional DAT/SE0 mode)	Unknown
	O	Differential receiver signal output in 4-pin bidirectional DP/DM TLL or 6-pin unidirectional TLL modes (not used in 3-pin bidirectional DAT/SE0 TLL or 2-pin bidirectional TLL modes)	
usbb1_mm_rxdp	I	Single-ended DP receiver signal input in 6-pin unidirectional modes (not used in 3-pin bidirectional DAT/SE0 or 4-pin bidirectional DP/DM modes)	Unknown
	O	Single-ended DP receiver signal output in 6-pin unidirectional TLL modes (not used in 3-pin bidirectional DAT/SE0 TLL or 4-pin bidirectional DP/DM TLL or 2-pin bidirectional TLL modes)	
usbb1_mm_rxdm	I	Single-ended DM receiver signal input in 6-pin unidirectional TLL modes (not used in 3-pin bidirectional DAT/SE0 or 4-pin bidirectional DP/DM or 2-pin bidirectional TLL modes)	Unknown
	O	Single-ended DM receiver signal output in 6-pin unidirectional TLL modes (not used in 3-pin bidirectional DAT/SE0 TLL or 4-pin bidirectional DP/DM TLL or 2-pin bidirectional TLL modes)	
<b>Multiple-Mode FS/LS Serial Interface: Port 2</b>			
usbb2_mm_txse0	I/O	SE0 function in 3-pin bidirectional DAT/SE0 mode	0
	I/O	DM function in 4-pin bidirectional DP/DM mode	
	O	SE0 output in 6-pin unidirectional DAT/SE0 mode	
	O	DM output in 6-pin unidirectional DP/DM mode	
	I/O	SE0-TLL in 2-/3-pin bidirectional DAT/SE0 TLL mode	
	I/O	DM-TLL in 2-/4-pin bidirectional DP/DM TLL mode	
	I	SE0-TLL input in 6-pin unidirectional DAT/SE0 TLL mode	
	I	DM-TLL input in 6-pin unidirectional DP/DM TLL mode	
usbb2_mm_txdat	I/O	DAT function in 3-pin bidirectional DAT/SE0 mode	Unknown
	I/O	DP function in 4-pin bidirectional DP/DM mode	
	O	DAT output in 6-pin unidirectional DAT/SE0 mode	
	O	DP output in 6-pin unidirectional DAT/SE0 mode	

<sup>(1)</sup> I = Input; O = Output

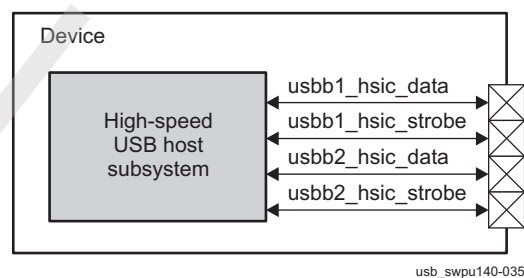
**Table 23-935. I/O Description (continued)**

Signal Name	I/O <sup>(1)</sup>	Description	Value at Reset
<b>Multiple-Mode FS/LS Serial Interface: Port 1</b>			
	I/O	DAT-TLL in 2-/3-pin bidirectional DAT/SE0 TLL mode	
	I/O	DP-TLL in 2-/4-pin bidirectional DP/DM TLL mode	
	I	DAT-TLL input in 6-pin unidirectional DAT/SE0 TLL mode	
	I	DP-TLL input in 6-pin unidirectional DP/DM TLL mode	
usbb2_mm_txen	O	Transmit enable in 3-pin bidirectional DAT/SE0 or 4-pin bidirectional DP/DM or 6-pin unidirectional modes	1
	I	Transmit enable in 3-pin bidirectional DAT/SE0 TLL or 4-pin bidirectional DP/DM TLL or 6-pin unidirectional TLL modes (not used in 2-pin bidirectional TLL modes)	
usbb2_mm_rxcv	I	Differential receiver signal input in 4-pin bidirectional DP/DM or 6-pin unidirectional modes (not used in 3-pin bidirectional DAT/SE0 mode)	Unknown
	O	Differential receiver signal output in the 4-pin bidirectional DP/DM TLL or 6-pin unidirectional TLL modes (not used in 3-pin bidirectional DAT/SE0 TLL or 2-pin bidirectional TLL modes)	
usbb2_mm_rxdp	I	Single-ended DP receiver signal input in 6-pin unidirectional modes (not used in 3-pin bidirectional DAT/SE0 or 4-pin bidirectional DP/DM modes)	Unknown
	O	Single-ended DP receiver signal output in 6-pin unidirectional TLL modes (not used in 3-pin bidirectional DAT/SE0 TLL or 4-pin bidirectional DP/DM TLL or 2-pin bidirectional TLL modes)	
usbb2_mm_rxdm	I	Single-ended DM receiver signal input in 6-pin unidirectional TLL modes (not used in 3-pin bidirectional DAT/SE0 or 4-pin bidirectional DP/DM or 2-pin bidirectional TLL modes)	Unknown
	O	Single-ended DM receiver signal output in 6-pin unidirectional TLL modes (not used in 3-pin bidirectional DAT/SE0 TLL or 4-pin bidirectional DP/DM TLL or 2-pin bidirectional TLL modes)	

**NOTE:** ULPI (PHY) Interfaces and ULPI TLL Interfaces can not be used together: either the ULPI (PHY) Interfaces or the ULPI TLL Interfaces are selected

### 23.11.2.3 HSIC Interface

The HS USB host subsystem supports the configurations with the HSIC interfaces shown in [Figure 23-253](#).

**Figure 23-253. HSIC Interface**

[Table 23-936](#) describes the I/O of the HSIC interface of the HS USB host subsystem.

**Table 23-936. HSIC Interface**

Signal Name	I/O <sup>(1)</sup>	Description	Reset Value
<b>HSUSB HSIC1</b>			
usbb1_hsic_data	I/O	HS interchip data	Unknown

<sup>(1)</sup> I = Input; O = Output

**Table 23-936. HSIC Interface (continued)**

Signal Name	I/O <sup>(1)</sup>	Description	Reset Value
usbb1_hsic_strobe	I/O	HS interchip strobe	Unknown
<b>HSUSB HSIC2</b>			
usbb2_hsic_data	I/O	HS interchip data	Unknown
usbb2_hsic_strobe	I/O	HS interchip strobe	Unknown

### 23.11.3 HS Multiport USB Host Subsystem Integration

This section describes the integration of the HS USB host subsystem, including information about clocks, resets, and hardware requests.

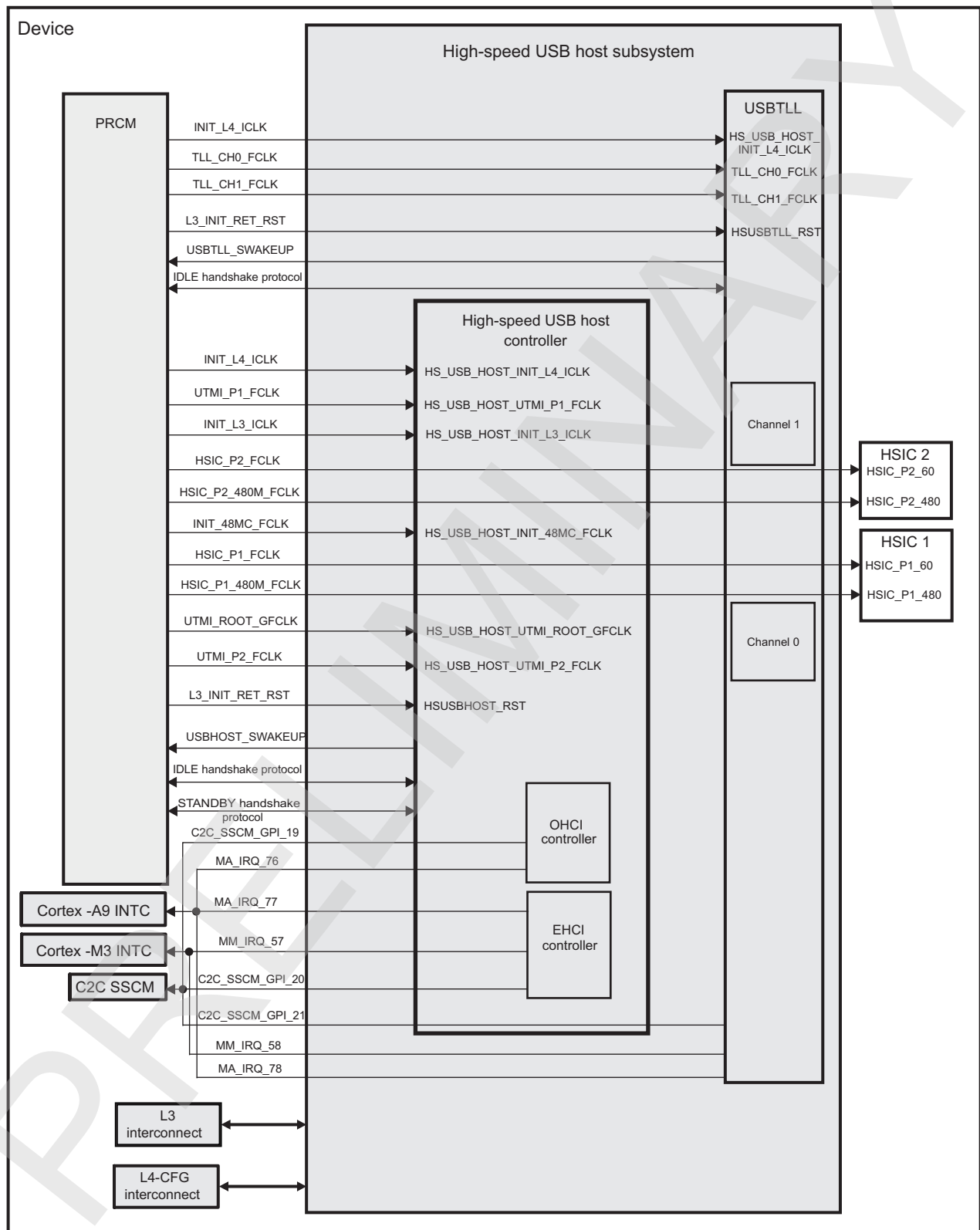
The USB host controller module includes other modules and additional logic attached to the UTMI/ULPI ports. The HS USB host controller is connected to the L3 interconnect master (initiator) and L4\_CFG interconnect slave (target) interfaces.

The elements of the USB HS host subsystem are:

- The multiport USB host controller. Each port uses one of the following modes:
  - ULPI, to a ULPI PHY (typically, off-chip)
  - UTMI, to a local UTMI (typically, on-chip) PHY or emulation of a PHY
- A local phase-locked loop (PLL), able to provide the USB-grade 48-, 60-, 480-MHz clocks
- Two TLL channels, bundled inside a USBTLL module. Each HS TLL channel can be seen as emulating a pair of back-to-back PHYs, and has:
  - A UTMI port, connected to a controller port (the host, in this case)
  - A ULPI port, connected to a (typically, off-chip) ULPI controller (peripheral, in this case)
- Two HSIC DFEs. The DFE plugs into the UTMI port of an HSIC-capable controller (the host, in this case), and controls HSIC-capable bidirectional I/O drivers driving the HSIC lines (strobe/data).

Figure 23-254 shows the HS USB subsystem integration.

Figure 23-254. HS USB Subsystem Integration



usb\_swpu140-028

Table 23-937 through Table 23-939 summarize the integration of the module in the device.



**Table 23-937. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
HSUSBHOST	PD_L3_INIT	L3 L4_CFG
HSUSBTLL	PD_L3_INIT	

**Table 23-938. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HSUSBHOST	HS_USB_HOST_INIT_L4_ICLK	INIT_L4_ICLK	PRCM	L4 interconnect enable signal for the slave interface on L4 (see <a href="#">Section 3.6.3.2</a> )
	HS_USB_HOST_INIT_L3_ICLK	INIT_L3_ICLK	PRCM	L3 interconnect clock, for the L3 master port interface (see <a href="#">Section 3.6.9</a> )
	HSIC_P2_480	HSIC_P2_480M_FCLK	PRCM	480-MHz clock for the USB HSIC (see <a href="#">Section 3.6.9</a> )
	HS_USB_HOST_UTMI_P1_FCLK	UTMI_P1_FCLK/usb1_ulpiphy_clk	PRCM/external PHY IC	Port 1 UTMI interface functional clock (see <a href="#">Section 3.6.9</a> and <a href="#">Section 3.6.3.2.2</a> )
	HSIC_P1_60	HSIC_P1_FCLK	PRCM	60-MHz clock for the USB HSIC (see <a href="#">Section 3.6.9</a> )
	HSIC_P1_480	HSIC_P1_480M_FCLK	PRCM	480-MHz clock for the USB HSIC (see <a href="#">Section 3.6.9</a> )
	HS_USB_HOST_INIT_48MC_FCLK	INIT_48MC_FCLK	PRCM	48-MHz functional clock for OHCI and EHCI (see <a href="#">Section 3.6.9</a> )
	HSIC_P2_60	HSIC_P2_FCLK	PRCM	60-MHz clock for the USB HSIC (see <a href="#">Section 3.6.9</a> )
	HS_USB_HOST_UTMI_ROOT_GFCLK	UTMI_ROOT_GFCLK	PRCM	60-MHz root hub clock (see <a href="#">Section 3.6.9</a> )
	HS_USB_HOST_UTMI_P2_FCLK	UTMI_P2_FCLK/usb2_ulpiphy_clk	PRCM/external PHY IC	Port 2 UTMI interface functional clock (see <a href="#">Section 3.6.9</a> and <a href="#">Section 3.6.3.2.2</a> )
HSUSBTLL	HS_USB_HOST_INIT_L4_ICLK	INIT_L4_ICLK	PRCM	L4 interconnect enable signal for the slave interface on L4 (see <a href="#">Section 3.6.3.2</a> )
	TLL_CH0_FCLK	TLL_CH0_FCLK	PRCM	Functional clock for channel 0 (see <a href="#">Section 3.6.9</a> )
	TLL_CH1_FCLK	TLL_CH1_FCLK	PRCM	Functional clock for channel 1 (see <a href="#">Section 3.6.9</a> )
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HSUSBHOST	HSUSBHOST_RST	L3_INIT_RET_RST	PRCM	The signal resets the HSUSBHOST module.
HSUSBTLL	HSUSBTLL_RST	L3_INIT_RET_RST	PRCM	The signal resets the HSUSBTLL module.

**Table 23-939. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
HSUSBHOST	HSUSB_EHCI_IRQ	MA_IRQ_77	Cortex-A9 INTC	Destination is Cortex™-A9 interrupt controller (INTC).
	HSUSB_OHCI_IRQ	MA_IRQ_76	Cortex-A9 INTC	Destination is Cortex-A9 INTC.
	HSUSB_EHCI_IRQ	MM_IRQ_57	Cortex-M3 INTC	Destination is Cortex-M3 INTC.
	HSUSB_OHCI_IRQ	C2C_SSCM_GPI_19	C2C SSCM	Destination is C2C SSCM.
	HSUSB_EHCI_IRQ	C2C_SSCM_GPI_20	C2C SSCM	Destination is C2C SSCM.
HSUSBTLL	HSUSB_TLL_IRQ	MA_IRQ_78	Cortex-A9 INTC	Destination is Cortex-A9 INTC.

**Table 23-939. Hardware Requests (continued)**

	HSUSB_TLL_IRQ	MM_IRQ_58	Cortex-M3 INTC	Destination is Cortex-M3 INTC.
	HSUSB_TLL_IRQ	C2C_SSCM_GPI_21	C2C SSCM	Destination is C2C SSCM.

The HS USB host controller and TLL module are attached to the L3\_INIT\_RET\_RST reset domain. The L3\_INIT\_RET\_RST signal resets the HS USB host controller and TLL module (see [Chapter 3, Power, Reset, and Clock Management](#)).

### 23.11.4 HS Multiport USB Host Subsystem Functional Description

This section describes the functionality of the HS USB host subsystem by describing the HS USB host controller, USBTLL module.

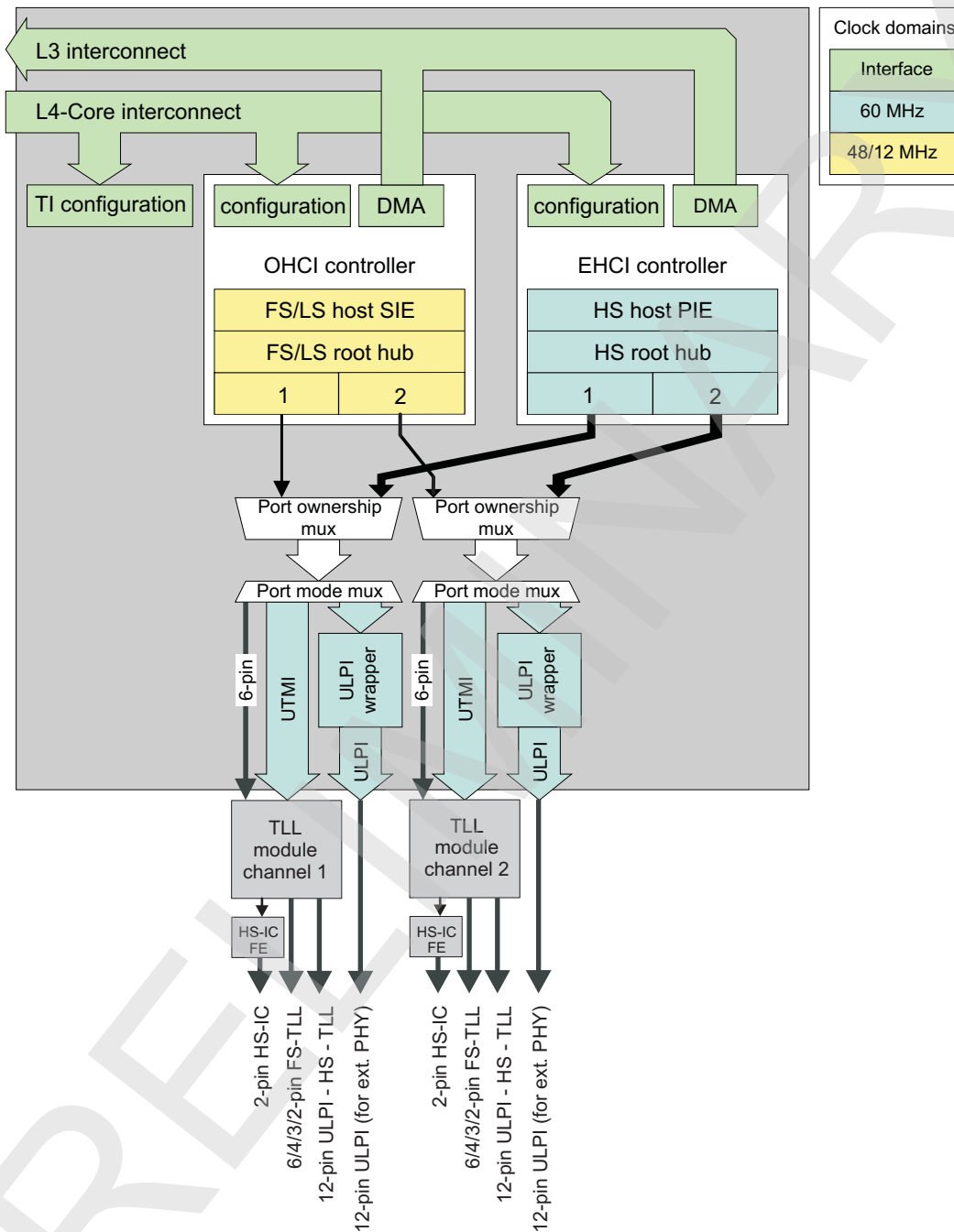
#### 23.11.4.1 USB Host Controller Functionality

The full details of the standard OHCI and EHCI host controller APIs (implemented by the current module) are not repeated here. For more information, see the following specifications:

- *Open Host Controller Interface (OHCI) specification for USB Release 1.0a*
- *Enhanced Host Controller Interface (EHCI) specification for USB Release 1.0*

[Figure 23-255](#) is an overview of the HS USB host controller internal architecture. It contains two independent, 3-port host controllers (OHCI and EHCI) that operate in parallel; however, only two physical ports are mapped in the device processor.

Figure 23-255. HS USB Host Controller Architecture



usb\_swpu140-029

### 23.11.4.1.1 HS USB Host Controller Software Reset

The HS USB host controller has its own software-reset function through the `UHH_SYSCONFIG[0]` `SOFTRESET` bit. Writing 0x1 to this bit resets the HS USB host controller. The bit value of 0x1 remains until the reset completes. When the software reset completes, the `SOFTRESET` bit is automatically set to 0x0 and has the same effect as the hardware reset.

### 23.11.4.1.2 HS USB Host Controller Power Management

The USB protocol defines power-saving modes, where clocks can be shut down. The host asynchronously detects ULPI/UTMI events and asserts the wake-up output, which causes the clocks to restart.

To reduce dynamic power consumption, an efficient idle scheme in the device is based on:

- Efficient local autoclock gating for each module
- Implementation of control sideband signals between the power, reset, and clock management (PRCM) module and each module

This enhanced idle control allows clocks to be activated and deactivated safely without complex software intervention. In both cases, the HS USB host controller power management is applied only to the interface clock domain.

The HS USB host controller has master (initiator) and slave (target) interfaces:

- As an initiator, the HS USB host controller implements the standby handshake protocol to inform the PRCM module when it enters standby mode and does not generate traffic on the interconnect.
- As a target, the HS USB host controller implements the IDLE handshake protocol to allow the PRCM module requiring it to enter idle mode.

#### 23.11.4.1.2.1 Standby Handshake Protocol

The HS USB host controller can go into standby mode, in which case it stops generating transactions on the interconnect. To save power, the module standby leads the PRCM module to disable the USB clocks.

The HS USB host controller has an Mstandby/Mwait handshake mechanism with the PRCM module (see [Chapter 3, Power, Reset, and Clock Management](#)).

The module is ready to enter standby mode (indicated by the Mstandby signal to the PRCM module asserted) when there is no USB activity and the module is idle. It means:

- The module is committed not to start any new transaction on its master interface.
- The whole module is idle and, therefore, the power manager can start the procedure to turn off the interface clock, if needed. This procedure must be implemented using the slave power-management protocol.

The handshake mechanism lets the module go into standby mode based on the [UHH\\_SYSCONFIG\[5:4\] STANDBYMODE](#) bit field.

[Table 23-940](#) lists the settings of standby mode.

**Table 23-940. HS USB Host Controller Standby Mode Settings**

MIDLEMODE Value	Selected Mode	Description
0x0	Force-standby	The HS USB host controller enters standby mode unconditionally (Mstandby is asserted unconditionally).
0x1	No-standby	The HS USB host controller never enters standby mode (Mstandby is never asserted).
0x2	Smart-standby	The HS USB host controller is ready to enter standby mode (Mstandby is asserted) when there is no more activity on the USB master interface of the interconnect.
0x3	Smart-standby + wakeup	The HS USB host controller enters smart mode with asynchronous (master) wakeup enabled.

#### 23.11.4.1.2.2 IDLE Handshake Protocol

The PRCM module handles an IDLE handshake protocol for the HSUSB host controller. The IDLE handshake protocol lets the PRCM module requiring the HS USB host controller to enter idle mode. Although this handshake is completely hardware and out of any software control, the way in which the HS USB host controller acknowledges the PRCM IDLE request is configurable through the [UHH\\_SYSCONFIG\[3:2\] SIDLEMODE](#) bit field. [Table 23-941](#) lists the settings of SIDLEMODE and the related acknowledgment modes.

**Table 23-941. HS USB Host Controller SIDLEMODE Settings**

SIDLEMODE Value	Selected Mode	Description
0x0	Force-idle	The HS USB host controller acknowledges unconditionally the IDLE request from the PRCM module, regardless of its internal operations. Because such a mode does not prevent the loss of data when the clock is switched off, the mode must be used carefully.
0x1	No-idle	The HS USB host controller never acknowledges any IDLE request from the PRCM module. This mode is safe from a module point of view because it ensures the clocks remain active; however, it is not efficient from a power-saving perspective because it does not allow the PRCM output clock to be shut off, and thus the power domain to be set to a lower power state.
0x2	Smart-idle	The HS USB host controller acknowledges the IDLE request basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, IRQs, or DMA requests are treated. This is the best approach for an efficient system power management.
0x3	Smart-idle + wakeup	The HS USB host controller enters smart mode with asynchronous (master) wakeup enabled.

#### 23.11.4.1.2.3 Wake-Up Request

The wake-up request signal USBHOST\_SWAKEUP is generated by the HS USB host controller to the PRCM module.

#### 23.11.4.1.3 HS USB Host Controller Save and Restore

The hardware context of the USB HS host can be frozen, saved, and restored.

The SAR mechanism extracts the hardware context of the HS USB host controller (after all USB activity is suspended) before switching off (saving), saves it to an external always-on memory, and reinjects it later after the module has been switched on again and reset (restored) seamlessly for the USB. For more information, see [Chapter 3, Power, Reset, and Clock Management](#).

#### 23.11.4.1.4 HS USB Host Controller Port Status

The USB port status is given through the [UHH\\_HOSTCONFIG\[10:8\]](#) bit field. The default value of the following bits is 0x1:

- [UHH\\_HOSTCONFIG\[8\]](#) P1\_CONNECT\_STATUS
- [UHH\\_HOSTCONFIG\[9\]](#) P2\_CONNECT\_STATUS

#### CAUTION

These bits show the port status as connected after power on even though no USB device is connected. The USB host controller has operational status registers (for example [HCRHPORTSTATUS\\_1](#) for the OHCI port 1) that indicate the correct port connect status. The USB driver software must read these status bits and check whether or not a port is connected. If the port is not connected the USB driver software must reprogram the [UHH\\_HOSTCONFIG](#) bits to indicate the correct port connect status.

#### 23.11.4.1.5 HS USB Host Controller Burst Control

To avoid buffer underflow, bursts must be enabled by writing 0x7 in the [UHH\\_HOSTCONFIG\[4:2\]](#) bit field, and the [UHH\\_HOSTCONFIG\[5\]](#) ENA\_INCR\_ALIGN bit must be set to 1.

#### 23.11.4.1.6 OHCI Implementation Specifications

Some features of the OHCI API are optional and/or implementation-specific. The choices made in the current implementation, the HS USB host controller, are described in the following list and are reflected in the register descriptions (see [Section 23.11.6.5, OHCI Registers](#)). For all standard features, see the *Open Host Controller Interface (OHCI) Specification for USB Release 1.0a*.

- **HCFMINTERVAL**[30:16] FSMPS bit field (FullSpeedMaxPacketSize) = 0x0000: The host stops scheduling new packets 0 bit times before the end of the frame (that is, by default, there is no scheduling overrun protection). To be updated by the software driver.
- **HCRHDESCRIPTORA**[7:0] NDP bit field (NumberDownstreamPorts) = 0x03 = three ports
- **HCRHDESCRIPTORA**[9] NPS bit (NoPowerSwitching) = 0: Ports are power-switched by default.
- **HCRHDESCRIPTORA**[8] PSM bit (PowerSwitchingMode) = 1: Per-port power switching is supported, although PPCM default setup has all ports controlled globally.
- **HCRHDESCRIPTORA**[31:24] POTPG bit field (PowerOnToPowerGood) = 0x0A = 10: Power ramp-up time is 10 x 2 ms = 20 ms.
- **HCRHDESCRIPTORB**[15:0] DR bit field (DeviceRemovable) = 0x0000: By default, no nonremovable devices (that is, devices attached to any of the ports) are removable.
- **HCRHDESCRIPTORB**[31:16] PPCM bit field (PortPowerControlMask) = 0x0000: By default, all ports are affected only by global power control.

#### 23.11.4.1.7 UTMI Interface

The host controller supports two ports. Each host controller port is configured to be in UTMI or ULPI mode (see the **UHH\_HOSTCONFIG**[21:16] P<sub>x</sub>\_MODE bit field).

In UTMI mode, a port has its UTMI signal set broadcasting the "outgoing" packets (from the host to the peripherals) and gathering the "incoming" ones (from the addressed peripheral to the host). ULPI signals for that port are undefined/don't care on all ports.

The UTMI ports between the HS USB host controller and the USBTLL module are on-chip and remain invisible.

UTMI transparent mode is used to connect with the HSIC, (see the **UHH\_HOSTCONFIG**[17:16] P1\_MODE or **UHH\_HOSTCONFIG**[17:16] P2\_MODE bit fields).

#### 23.11.4.1.8 ULPI Interface

The host supports two ports. Each host controller port is configured to be either in UTMI or in ULPI mode (see the **UHH\_HOSTCONFIG**[17:16] P1\_MODE or **UHH\_HOSTCONFIG**[17:16] P2\_MODE bit fields).

In ULPI mode, a port has its ULPI signal set broadcasting the outgoing packets (that is, from the host to the peripherals) and gathering the incoming packets (that is, from the addressed peripheral to the host). UTMI signal sets are undefined/don't care on all ports.

---

**NOTE:** The ULPI clock (part of the 12-wire ULPI interface) is expected to be routed by the system from the ULPI CLK input I/O pad to the clock input used by all HS modes (ULPI and UTMI) for the current port. [Section 3.6.3.2.2](#)

---

### 23.11.4.2 USBTLL Module Functionality

#### 23.11.4.2.1 USBTLL Software Reset

Writing 0x1 to the **USBTLL\_SYSCONFIG**[1] SOFTRESET bit resets the USBTLL module. The bit value of 0x1 remains until the reset completes. When the software reset completes, the SOFTRESET bit is automatically set to 0x0 and has the same effect as the hardware reset.

#### 23.11.4.2.2 USBTLL Power Management

The USB protocol defines power-saving modes, where clocks can be shut down.

##### 23.11.4.2.2.1 IDLE Handshake Protocol

The PRCM module handles an IDLE handshake protocol for the USBTLL module. The IDLE handshake protocol lets the PRCM module require the HS USBTLL module to enter idle mode.

The way in which the HS USBTLL module acknowledges the PRCM IDLE request is configurable through the [USBTLL\\_SYSCONFIG\[4:3\] SIDLEMODE](#) bit field. [Table 23-942](#) lists related acknowledgment modes.

**Table 23-942. HS USBTLL Module SIDLEMODE Settings**

SIDLEMODE Value	Selected Mode	Description
0x0	Force-idle	The HS USB host controller acknowledges unconditionally the IDLE request from the PRCM module, regardless of its internal operations.
0x1	No-idle	The HS USB host controller never acknowledges any IDLE request from the PRCM module.
0x2	Smart-idle	The HS USB host controller acknowledges the IDLE request, basing its decision on its internal activity.

### 23.11.4.2.3 USBTLL Save and Restore

The SAR mechanism extracts the hardware context of the USBTLL module (after all USB activity is suspended) before switching off (saving), save it to an external always-on memory, and reinjects it later after the module is switched on again and reset (restored) seamlessly for the USB.

Hardware context of the USBTLL module is accessed through the `USBTLL_SAR_CNTX_i[31:0]` CNTX bit field.

The PRCM module enables the SAR mechanism for the USBTLL module. When set, the PRM module initiates the save and/or the restore sequences at the appropriate time. When not set, the USB host is treated as a standard module, and the save/restore sequences do not occur.

For more information, see [Section 3.9.3](#).

[Table 23-943](#) lists the USBTLL registers affected by the SAR context.

---

**NOTE:** Because all addresses give access to the same physical register (that is, to the same piece of context), the ULPI registers with multiple accesses (write, set, clear) are listed only once in the table.

---

**Table 23-943. USBTLL Registers Affected by the SAR Context**

Register Name	Comments on SAR Policy
<a href="#">USBTLL_SYSCONFIG</a>	Except the SOFTRESET bit (write-only)
<a href="#">USBTLL_IRQENABLE</a>	-
<a href="#">TLL_SHARED_CONF</a>	Except the FCLK_REQ bit
<a href="#">TLL_CHANNEL_CONF_i</a>	Except the FSLSLINESTATE bit field
<a href="#">FUNCTION_CTRL_i</a>	-
<a href="#">INTERFACE_CTRL_i</a>	-
<a href="#">OTG_CTRL_i</a>	-
<a href="#">USB_INT_EN_RISE_i</a>	-
<a href="#">USB_INT_EN_FALL_i</a>	-
<a href="#">USB_INT_STATUS_i</a>	-

### 23.11.4.2.4 USBTLL Channels and Ports

Following the same convention as UTMI and ULPI, the current specification is consistently PHY-centric (that is, directions are always given with respect to the transceiver emulated here by the TLL, and not with respect to the link controller): An input goes from the link controller to the TLL (transceiver emulator) (that is, it is an input for the USBTLL module). Reciprocally, an output goes from TLL (transceiver) to the link controller (that is, it is an output for the USBTLL module).

By convention, the local link controller is the controller integrated on the same IC as the USBTLL module: This is the HS USB host controller in the device. The remote link controller is the other controller, located off-chip (that is, on another IC). One controller is always the USB host, the other is the USB peripheral, and they communicate through the USBTLL module.

A channel is defined as a independent USB path through the USBTLL module, which always converts the UTMI+ transceiver interface protocol coming from the local link controller (the HS USB host controller in the device). There are two channels of the USBTLL module in the device.

A USB port is a set of I/O signals that carry the data and control information from/to a USB line. Several port formats exist, with different capabilities. A channel has three ports. If the channel is active, two ports are active at a time, depending on the channel configuration. The mode remains static throughout the USB operation.

#### 23.11.4.2.5 USBTLL Channel Configuration

A channel configuration is a set of software settings that specifies the connection of two of the channel ports through the USBTLL module. USB data and control injected on one side (or port) comes out on the other side (or port) after a certain amount of processing, depending on the mode. [Table 23-944](#), lists the modes.

All configurations connect the PHY UTMI port (attached to the HS USB host controller) to one of the other two ports (attached to a variety of transceivers or controllers on the pad side).

[Table 23-944](#) describes the available modes and the software settings required for each. Channel *i* has the following settings:

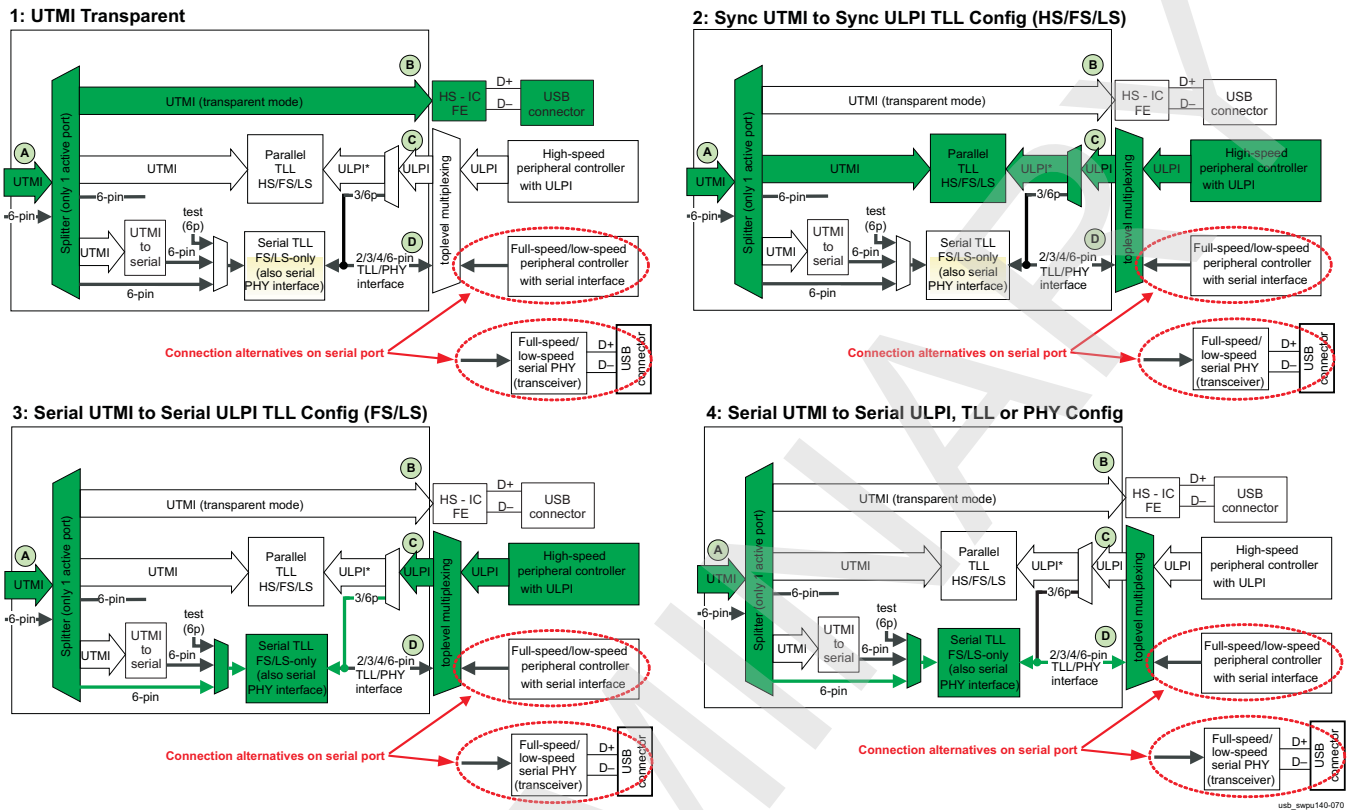
- CHANMODE: [TLL\\_CHANNEL\\_CONF\\_i\[2:1\]](#) CHANMODE bit field
- FLSMODE: [TLL\\_CHANNEL\\_CONF\\_i\[27:24\]](#) FLSMODE bit field
- FLSSEIRIALMODE\_3PIN/6PIN: Either the ULPI PHY-side [INTERFACE\\_CTRL\\_i\[1\]](#) FLSSEIRIALMODE\_3PIN bit or the [INTERFACE\\_CTRL\\_i\[0\]](#) FLSSEIRIALMODE\_6PIN bit (only one at a time can be set to 1)

**Table 23-944. USBTLL Channel Configuration**

Configuration	Mode	CHANMODE	FLSMODE	Other Settings	Ports	Speed	Remote Port Connection
1	UTMI transparent	0	N/A	-	A-B	-	UTMI transparent
2	ULPI synchronous TLL	0	N/A	FLSSEIRIALMODE_3PIN/6PIN = 0	A-C	HFL	ULPI link (peripheral controller)
3	Serial UTMI to serial ULPI TLL	0	N/A	FLSSEIRIALMODE_3PIN/6PIN = 1	A-C	FL	ULPI link (peripheral controller) supporting 3-/6-pin mode
4	Serial UTMI to serial TLL	1	0x4-0x7; 0xA-0xB	-	A-D	FL	Serial link (2-/3-/4-/6-pin)
4	Serial UTMI to serial PHY	1	0x0-0x3	-	A-D	FL	Serial transceiver (2-/3-/4-/6-pin)



Figure 23-256. Per-Configuration Data Path Through USBTLL



23.11.4.2.6 USBTLL VBUS Management and Emulations

In transceiver configurations, a USB cable is present, including a 5-V VBUS supply line. On the other hand, in TLL configurations, the physical USB lines are emulated and have no physical existence. This is especially true for the VBUS line, which distributes the 5-V power provided by the default host (or A-device) to the entire bus. VBUS is also used for signaling purposes, and those features must be emulated:

- A peripheral detects the presence of a host by detecting the presence of VBUS.
- USB OTG defines an elaborate voltage-sensing scheme to dynamically switch on and off VBUS (start and stop sessions). In the context of TLL, this brings no power savings compared to a simple suspend.
- In particular, USB OTG uses VBUS as a wake-up source (VBUS-pulsing SRP) for the default peripheral (or B-device).

For more information on how sideband controls are integrated, see Figure 23-241 and Figure 23-242 and the related explanations.

23.11.4.2.6.1 VBUS Control and Status for Transceiver (Non-TLL) Configurations

In non-TLL modes, VBUS exists, and the problem is to propagate control and status to and from the VBUS manager IC (typically, the transceiver).

Only serial transceiver configurations are concerned in the case of the HS USB host subsystem in the device.

23.11.4.2.6.1.1 VBUS Management in Serial Transceiver Configurations

VBUS management is not standardized in transceiver configurations. The chosen implementation is described in the following list (also see Figure 23-241).

- VBUS control required for host and OTG operation (VBUS drive, VBUS pullup "charge," VBUS pulldown "discharge") is assumed to be handled separately from the USBTLL module (that is, by software and straight to the power IC, which can be the transceiver, especially in OTG cases).
- VBUS status must be sampled by the appropriate hardware (again, most often the transceiver) and reported by software to the USBTLL module, using the [TLL\\_CHANNEL\\_CONF\\_i\[16\]](#) DRVVBUS and [TLL\\_CHANNEL\\_CONF\\_i \[15\]](#) CHRGVBUS bits, as indicated in [Table 23-945](#).

[Table 23-945](#) lists the values to write to the [TLL\\_CHANNEL\\_CONF\\_i](#) register depending on the VBUS status observed by the transceiver on the VBUS line. The same register fields are also used in TLL configuration, and have been named according to that second configuration. In transceiver configurations the actual signification of the fields is:

- DRVVBUS: Set to 1 to report a VBUS level greater than VBUS valid.
- CHRGVBUS: Set to 1 to report a VBUS level greater than session valid.

**Table 23-945. VBUS Level Software Reporting for Serial Transceiver Configuration**

VBUS Status	<a href="#">TLL_CHANNEL_CONF_i[16]</a> DRVVBUS	<a href="#">TLL_CHANNEL_CONF_i[15]</a> CHRGVBUS
VBUS valid	1	1
Session valid (A/B)	0	1
Session not valid	0	0
Session end	0	0

#### 23.11.4.2.6.2 VBUS Emulation for TLL Configurations

The TLL VBUS emulation sums up all actions on the VBUS line and obtains a voltage level, which is reported in the VBUS status bits following the protocol. The level depends on the immediate VBUS actions and has no memory of previous levels, whereas an actual VBUS line behaves like an RC circuit and takes time to charge and discharge. This causes the following differences:

- The TLL level always jumps abruptly from session valid to session end (and back) with no transient time in between (where session is neither valid nor ended) as in real life.
- The charge feature is used for VBUS-pulsing SRP and is enabled long enough to go over the session valid threshold, but without reaching VBUS valid. In the TLL, the transition to session valid is immediate, and VBUS valid is never reached even if the charge is intentionally kept active.
- The discharge feature is used in real life to accelerate the voltage drop of an undriven VBUS toward the session-end level. For TLL, therefore, this is useless (although the UTMI input/ULPI register bit does exist, for compatibility), and is always don't care.

#### 23.11.4.2.6.2.1 VBUS Emulation in ULPI TLL Modes

[Table 23-946](#) summarizes the VBUS emulation in ULPI TLL modes. VBUS controls are writable, static PHY-side registers on the ULPI side, and input signals on the ULPI ports (port A). VBUS status bits are read-only, volatile PHY-side registers on the ULPI, and output signals on the ULPI ports (port A).

**Table 23-946. Emulation of VBUS Levels for UTMI-to-ULPI TLL Mode**

VBUS Controls (Actions)			VBUS Level	VBUS Status		
<a href="#">OTG_CTRL_i[5]</a> DRVVBUS	<a href="#">OTG_CTRL_i[4]</a> CHRGVBUS	<a href="#">OTG_CTRL_i[3]</a> DISCHRGVBUS		<a href="#">USB_INT_STATU TUS_i[1]</a> VBUSVALID bit	<a href="#">USB_INT_STATU S_i[2]</a> SESSVALID bit	<a href="#">USB_INT_STATU S_i[3]</a> SESSSEND bit
1	X	X	VBUS valid	1	1	0
0	1	X	VBUS valid	0	1	0
0	0	X	Session end	0	0	1

### **23.11.4.2.6.2 VBUS Emulation in Serial TLL Modes**

In serial TLL modes, VBUS status and control is implemented with ad hoc sideband signals (see [Figure 23-242](#)).

VBUS control can be done in software, by writing to the following fields of the [TLL\\_CHANNEL\\_CONF\\_i](#) register:

- **DRVVBUS:** Set to 1 to drive VBUS to 5 V (for A-device or host)
- **CHRGVBUS:** Set to 1 to pull up VBUS (for SRP)
- There is no pulldown (discharge) control, because the emulated VBUS has no latency and VBUS level goes to the session-end level as soon as it is neither driven nor pulled up.

Alternatively, VBUS drive can also be hardware-controlled through a dedicated input. (The DRVVBUS bit and input signal are ORed internally.)

VBUS status is available on dedicated output signals. If those outputs are not available at the top level, a software alternative is to use the voltage status reported on the interface (through the standard UTMI+ sideband signals) of the local controller (the HS USB host controller in the device) and to pass it to the remote controller (a peripheral controller) by means of an ad hoc software-controller interface other than the USB. This is based on the fact that the level of VBUS is the same on both extremities of the bus (that is, it does not matter which side does the measurement).

### **23.11.4.2.7 USBTLL Multimode Serial Port**

The multimode serial port requires six bidirectional I/O pads to support all eight defined modes (selected in the [TLL\\_CHANNEL\\_CONF\\_i\[27:24\]](#) FLSMODE bit field when field CHANMODE = 0x1 = UTMI-to-serial). Those modes are FS/LS only (that is, HS is not supported over a serial interface).

The pads are named TXEN, TXDAT, TXSE0, RXRCV, RXDP, and RXDM after their functionality in standard 6-pin mode (mode 0). Each pad has an input, output, and output-enable signal associated with it on the USBTLL entity.

[Table 23-947](#) lists the functionality of each pad in each mode. USBTLL outputs are shown in yellow, inputs in blue, and bidirectional pads in green.

**Table 23-947. Serial Mode Description, Signal Functionality**

Usual Name	6-Pin Mode	6-Pin Mode (Alt)	3-Pin Mode	4-Pin Mode	6-Pin TLL Mode	6-Pin TLL (Alt) Mode	3-Pin TLL Mode	4-Pin TLL Mode	2-Pin TLL Mode	2-Pin TLL (Alt) Mode
<a href="#">TLL_CHANNEL_CONF_i</a> [27:24] FLSMODE field	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0xA	0xB
TX encoding	DAT/SE0	DP/DM	DAT/SE0	DP/DM	DAT/SE0	DP/DM	DAT/SE0	DP/DM	DAT/SE0	DP/DM
RX encoding	DP/DM/RCV	DP/DM/RCV	DAT/SE0	DP/DM/RCV	DP/DM/RCV	DP/DM/RCV	DAT/SE0	DP/DM/RCV	DAT/SE0	DP/DM
Pin use	Unidirectional	Unidirectional	Bidirectional	Bidirectional	Unidirectional	Unidirectional	Bidirectional	Bidirectional	Bidirectional	Bidirectional
Pin count	6	6	3	4	6 or 5 <sup>(1)</sup>	6 or 5 <sup>(1)</sup>	3	4 or 3 <sup>(2)</sup>	2	2
<b>I/O Pad Function Per Mode</b>										
TXEN	TX Enable	TX Enable	TX Enable	TX Enable	TX Enable	TX Enable	TX Enable	TX Enable	N/C	N/C
TXDAT	TX Diff Data	TX SE Plus Data	TX/RX Diff Data	TX/RX SE Plus Data	TX Diff Data	TX SE Plus Data	TX/RX Diff Data	TX/RX Diff Data	TX/RX Diff Data	TX/RX SE Plus Data
TXSE0	TX force SE0	TX SE Minus Data	TX/RX force SE0	TX/RX SE Minus Data	TX force SE0	TX SE Minus Data	TX/RX force SE0	TX/RX force SE0	TX/RX force SE0	TX/RX SE Minus Data
RXRCV	RX Diff Data	RX Diff Data	N/C	RX Diff Data	RX Diff Data	RX Diff Data	N/C	RX Diff Data	N/C	N/C
RXDP	RX SE Plus Data	RX SE Plus Data	N/C	N/C	RX SE Plus Data	RX SE Plus Data	N/C	N/C	N/C	N/C
RXDP	RX SE Minus Data	RX SE Minus Data	N/C	N/C	RX SE Minus Data	RX SE Minus Data	N/C	N/C	N/C	N/C

<sup>(1)</sup> RXRCV and RXDP carry the same information: RXDP can drive both inputs of the remote controller, and RXRCV can be kept unused.

<sup>(2)</sup> Same remark for TXDAT (for outputs) and RXRCV: TXDAT only is enough.

**23.11.4.2.8 USBTLL Attach/Connect Emulation for Serial Modes**

This section applies to all serial TLL modes in the following circumstances:

- In UTMI-to-serial mode (`TLL_CHANNEL_CONF_i[2:1] CHANMODE = 0x1`) for all TLL values of the `TLL_CHANNEL_CONF_i[27:24] FLSMODE` bit field (0x4-0x7; 0xA-0xB)
- In UTMI-to-ULPI TLL mode (`TLL_CHANNEL_CONF_i[2:1] CHANMODE = 0x0`) when the ULPI bus is switched to 6-pin or 3-pin serial modes

In those modes, the USB bus lines are emulated by USBTLL internal logic and are never available on the outside. The pullup/pulldown actions described in the USB specification cannot be applied directly, and the USB cable cannot be physically attached.

Because serial modes do not specify a standard format for those sideband settings, a custom software-controlled one is implemented:

- The `TLL_CHANNEL_CONF_i[4]` TLLATTACH bit emulates the physical attachment of the two controllers through a TLL cable.
  - When this bit is cleared, the local controller RX path shows only the local controller (the HS USB host controller) actions on the bus: TX driving, pullups, pulldowns. The same thing applies for the remote controller RX path (except that test override is not available).
  - When the bit is set, the actions of both sides are applied to the same bus and are resolved, similar to a real bus. The RX path for both sides shows the same bus state.
- The `TLL_CHANNEL_CONF_i[5]` TLLCONNECT bit emulates the USB electrical connect (that is, the pullup by the USB peripheral of one of the two USB lines [by a 1.5 Kohm resistor]), which causes the line state to transition from SE0 to J, which is detected by the USB host. The register bit is ORed with a USBTLL module input signal; the connect control can be software (L4\_CFG interconnect write access) or hardware (input level). The speed of the connection is determined by the TLLFULLSPEED bit.
- The `TLL_CHANNEL_CONF_i[6]` TLLFULLSPEED bit determines the speed (full or low) of the USB connect to be emulated. The connect enable (controlled as previously defined) results in the pulling up of D+ (1 = full speed) or D- (0 = low speed) (see [Table 23-948](#)).
- The 15 Kohm pulldowns are implicit: Because they are supposed to be turned on at least on the host side of the bus, they do not require additional control.

**NOTE:** Sideband control and status actions such as pullups are included in parallel (that is, nonserial) standards (UTMI, ULPI), and do not require custom additions.

**Table 23-948. Pullup Enable Emulation in Serial TLL Modes**

TLL_CHANNEL_CONF_i Fields		Input Signal	Resulting TLL Pullup Emulation	
TLLFULLSPEED	TLLCONNECT	USB State	D+ Pullup	D- Pullup
1	0	Full-speed unconnected	Off	Off
1	1	Full-speed connected	On	Off
0	0	Low-speed unconnected	Off	Off
0	1	Low-speed connected	Off	On

### 23.11.5 HS Multiport USB Host Subsystem Low-Level Programming Model

This section describes the low-level hardware programming sequences for the configuration and use of the HS USB host subsystem.

#### 23.11.5.1 Global Initialization

##### 23.11.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the HS USB host subsystem is used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the HS USB host subsystem.

Table 23-949 describes the global initialization of surrounding modules.

**Table 23-949. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	All clocks must be enabled. See <a href="#">Section 3.6.3.2, CM Clock Source</a> .
Cortex-A9 MPU INTC or Cortex-M3 MPU INTC	The Cortex-A9 MPU or Cortex-M3 MPU INTC must be configured to enable the interrupt request generation to the Cortex-A9 MPU or Cortex-M3 MPU subsystem when interrupt requests are generated by the HS multiport USB host subsystem. See the respective functional description in <a href="#">Chapter 17 Interrupt Controller</a> for the MPUs.
Interconnect (L3 and L4)	For more information, see <a href="#">Section 13.2, L3 Interconnect</a> , and <a href="#">Section 13.3, L4 Interconnect</a> .

##### 23.11.5.1.2 HS Multiport USB Host Subsystem Global Initialization

This procedure in [Table 23-950](#) initializes the HS multiport USB host subsystem after a power-on or software reset.

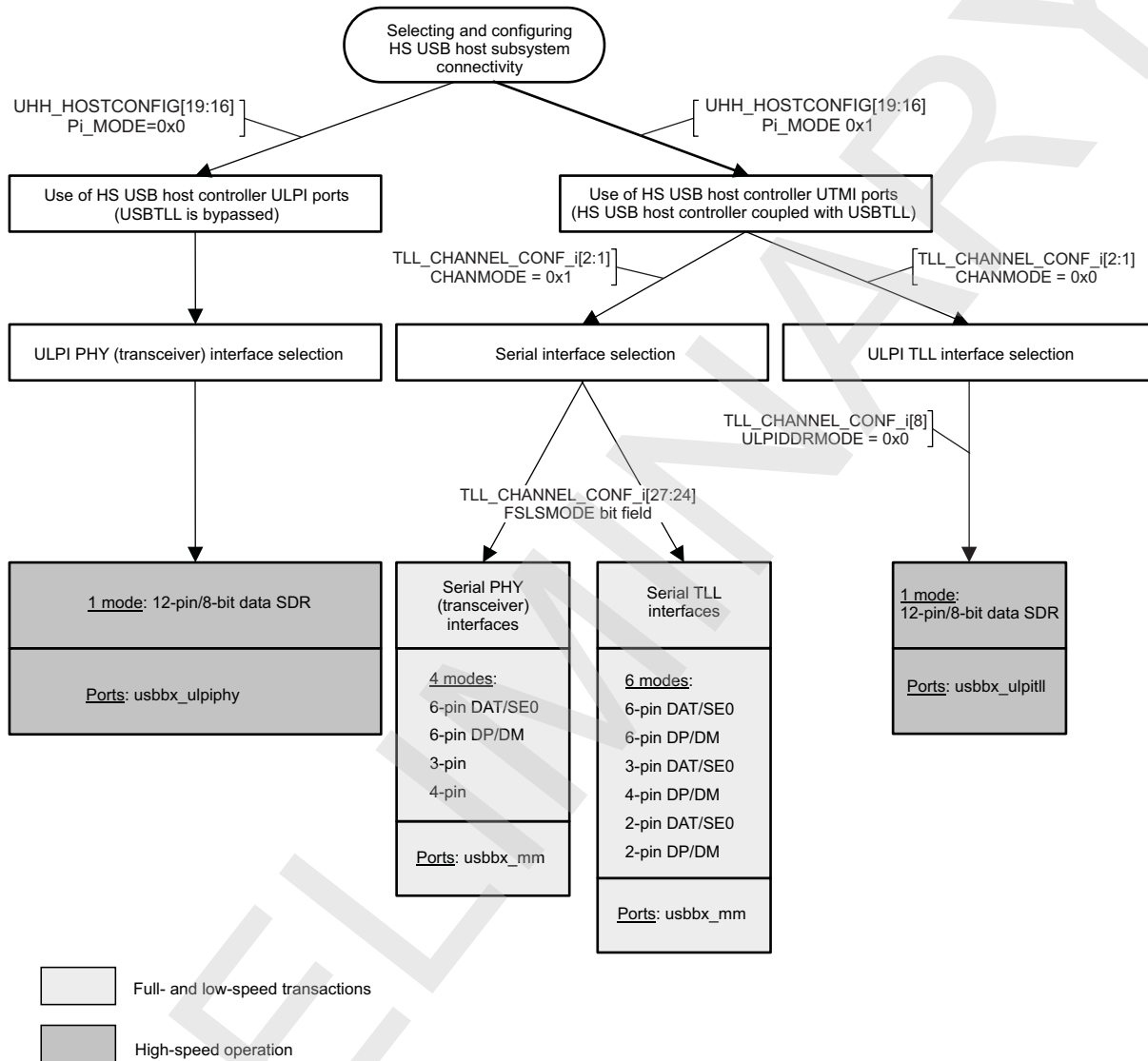
**Table 23-950. HS Multiport USB Host Subsystem Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Perform a software reset on the TLL module.	<a href="#">USBTLL_SYSCONFIG</a> [1] SOFTRESET	0x1
Wait until reset is finished.	<a href="#">USBTLL_SYSSTATUS</a> [0] RESETDONE	=0x1
Enable IRQ event for the TLL module.	<a href="#">USBTLL_IRQENABLE</a> [1] FCLK_END_EN	0x1
Enable IRQ event for the TLL module.	<a href="#">USBTLL_IRQENABLE</a> [0] FCLK_START_EN	=0x1
Perform a software reset on the HS USB controller.	<a href="#">UHH_SYSCONFIG</a> [0] SOFTRESET	0x1
Wait until the EHCI HS host is out of reset.	<a href="#">UHH_SYSSTATUS</a> [2] EHCI_RESETDONE	=0x0
Wait until the OHCI HS host is out of reset.	<a href="#">UHH_SYSSTATUS</a> [1] OHCI_RESETDONE	=0x0
Enable IRQ event for the OHCI HS host.	<a href="#">HCINTERRUPTENABLE</a> [31] MIE	0x1
Enable IRQ event for the EHCI HS host.	<a href="#">USBINTR</a> [0] USBIE	0x1

**23.11.5.2 Operational Modes Configuration (Selecting and Configuring USB Connectivity)**

Figure 23-257 shows how to select and configure the HS USB host subsystem connectivity.

**Figure 23-257. Selecting and Configuring HS USB Host Subsystem Connectivity**



usb\_swpu140-034

**Table 23-951. Register Call Summary for Selecting and Configuring High-Speed USB Host Subsystem Connectivity**

Register Name	
UHH_HOSTCONFIG	TLL_CHANNEL_CONF_i

**23.11.5.3 USBTLL Registers**

The USBTLL module contains two types of software-programmable registers.

**23.11.5.3.1 TLL Control and Status Registers**

These 32-bit registers configure the various channels. These registers are accessed by the MPU through the L4\_CFG interconnect. They are used primarily before USB activity starts. The registers are:

- OCP-standard registers for revision number, IRQ, clocking management, etc.

- TLL-specific registers

### 23.11.5.3.2 ULPI PHY-Side Registers

Each TLL channel emulates a ULPI transceiver and therefore contains this set of 8-bit PHY-side registers, per ULPI specification. The registers are:

- All ULPI-mandatory standard registers and fields
- A selection of ULPI-optional standard registers and fields, when relevant to the TLL context
- Vendor-specific registers, mapped at the addresses specified for that purpose in the ULPI specification

These registers are accessed by the external (that is, off-chip) link controller over the ULPI port of each channel, in the 0x100-byte ULPI address space, using the ULPI register access protocol.

They are accessible by the L4\_CFG interconnect: The ULPI register sets of all channels are mapped side by side in the upper part of the L4\_CFG interconnect address space, where they can be accessed through byte accesses. If a conflict occurs between the two access modes, access over ULPI has priority, but both accesses eventually complete correctly. For normal USB activity, all register accesses are expected to go over ULPI, and register changes caused by L4\_CFG interconnect accesses can compromise proper USB operation. The L4\_CFG interconnect access port is intended for:

- Miscellaneous test and debug
- Nonintrusive observation of ongoing USB operations (test)
- Context restore: During USB suspend periods, the USBTLL module can be switched off to save power. When the module is switched on again, the ULPI register contents, which have been lost, can be restored over the L4\_CFG interconnect before USB operations restart, provided they were saved elsewhere beforehand.



### 23.11.6 HS USB Host Subsystem Register Manual

This section provides the register description of the module.

#### 23.11.6.1 USB\_Host\_HS Instance Summary

Table 23-952 summarizes the USB\_Host\_HS instance.

Table 23-952. USB\_Host\_HS Instance Summary

Module Name	L4_CFG Base Address	Size
USBTLLHS_config	0x4A06 2000	2KB
USBTLLHS_ULPI	0x4A06 2800	2KB
HSUSBHOST	0x4A06 4000	2KB
OHCI	0x4A06 4800	1KB
EHCI	0x4A06 4C00	1KB

#### 23.11.6.2 USBTLLHS\_config Registers

##### 23.11.6.2.1 USBTLLHS\_config Register Summary

Table 23-953 summarizes the USBTLLHS\_config register mapping.

Table 23-953. USBTLLHS\_config Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
USBTLL_REVISION	R	32	0x0000 0000	0x4A06 2000
USBTLL_HWINFO	R	32	0x0000 0004	0x4A06 2004
USBTLL_SYSCONFIG	RW	32	0x0000 0010	0x4A06 2010
USBTLL_SYSSTATUS	R	32	0x0000 0014	0x4A06 2014
USBTLL_IRQSTATUS	RW	32	0x0000 0018	0x4A06 2018
USBTLL_IRQENABLE	RW	32	0x0000 001C	0x4A06 201C
TLL_SHARED_CONF	RW	32	0x0000 0030	0x4A06 2030
TLL_CHANNEL_CONF_j <sup>(1)</sup>	RW	32	0x0000 0040 + (0x4 * i)	0x4A06 2040 + (0x4 * i)
USBTLL_SAR_CNTX_j <sup>(2)</sup>	RW	32	0x0000 0400 + (0x4 * j)	0x4A06 2400 + (0x4 * j)

<sup>(1)</sup> i = 0 to 2

<sup>(2)</sup> j = 0 to 6

#### CAUTION

The USBTLLHS\_config registers are limited to 32-bit data accesses; 16-bit and 8-bit accesses are not allowed and can corrupt register content.

##### 23.11.6.2.2 USBTLLHS\_config Register Description

through describe the USBTLLHS\_config registers.

**Table 23-954. USBTLL\_REVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 23-953</a> .	<b>Instance</b>	USBTLLHS_config
<b>Description</b>	OCP standard revision number, BCD encoded		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Revision number	R	Ti internal data

**Table 23-955. Register Call Summary for Register USBTLL\_REVISION**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_config Register Summary: \[0\]](#)

**Table 23-956. USBTLL\_HWINFO**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 23-953</a> .	<b>Instance</b>	USBTLLHS_config
<b>Description</b>	Information on hardware configuration of host		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SAR_CNTX_SIZE															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0000000
7:0	SAR_CNTX_SIZE	Save-and-Restore context size, in 32-bit words, i.e. number of 32-bit registers with significant context information, mapped from offset 0x400 upward.	R	0x07

**Table 23-957. Register Call Summary for Register USBTLL\_HWINFO**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_config Register Summary: \[0\]](#)
- [USBTLLHS\\_config Register Description: \[1\]](#)

**Table 23-958. USBTLL\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 23-953</a> .	<b>Instance</b>	USBTLLHS_config
<b>Description</b>	OCP standard system configuration register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLOCKACTIVITY	RESERVED	SIDLEMODE	ENAWAKEUP	SOFTRESET	AUTOIDLE										

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	CLOCKACTIVITY	Enable autogating of OCP-derived internal clocks while module is idle. 0x0: OCP-derived internal clocks OFF during idle 0x1: OCP-derived internal clocks ON during idle	RW	0
7:5	RESERVED	Reserved	R	0x0
4:3	SIDLEMODE	Slave interface power management control. Idle Req/ack control 0x0: Force-Idle mode. Sideack[1] asserted after Sidlreq assertion 0x1: No-idle mode. Sideack[1] never asserted. 0x2: Smart-idle mode. Sideack[1] asserted after Sdlreq assertion when no more activity on the OCP.	RW	0x0
2	ENAWAKEUP	Asynchronous wake-up generation control (Swakeup) 0x0: Wake-up generation disabled 0x1: Wake-up generation enabled	RW	0
1	SOFTRESET	Module software reset Write 0x0: no effect Write 0x1: Starts softreset sequence.	W	0
0	AUTOIDLE	Internal autogating control 0x0: Clock always running 0x1: When no activity on OCP, clock is cut off.	RW	1

**Table 23-959. Register Call Summary for Register USBTLL\_SYSCONFIG**

High-Speed Multiport USB Host Subsystem

- [USBTLL Software Reset: \[0\]](#)
- [IDLE Handshake Protocol: \[1\]](#)
- [USBTLL Save and Restore: \[2\]](#)
- [HS Multiport USB Host Subsystem Global Initialization: \[3\]](#)
- [USBTLLHS\\_config Register Summary: \[4\]](#)

**Table 23-960. USBTLL\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	USBTLLHS_config
<b>Physical Address</b>	See <a href="#">Table 23-953</a> .		
<b>Description</b>	OCP standard system status register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											RESETDONE				

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Indicates when the module has entirely come out of reset Read 0x0: Reset is ongoing Read 0x1: Reset is done	R	0

**Table 23-961. Register Call Summary for Register USBTLL\_SYSSTATUS**

High-Speed Multiport USB Host Subsystem

- [HS Multiport USB Host Subsystem Global Initialization: \[0\]](#)
- [USBTLLHS\\_config Register Summary: \[1\]](#)

**Table 23-962. USBTLL\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	USBTLLHS_config
<b>Physical Address</b>	See <a href="#">Table 23-953</a> .		
<b>Description</b>	OCP standard IRQ status vector. Write 1 to clear a bit.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACCESS_ERROR	FCLK_END	FCLK_START													

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	ACCESS_ERROR	Access error to ULPI register over OCP: USB clock must run for that type of access to succeed. 0x0: No event pending 0x1: Event pending	RW	0
1	FCLK_END	Functional clock is no longer requested for USB clocking 0x0: No event pending 0x1: Event pending	RW	0
0	FCLK_START	Functional clock is requested for USB clocking 0x0: No event pending 0x1: Event pending	RW	0

**Table 23-963. Register Call Summary for Register USBTLL\_IRQSTATUS**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_config Register Summary: \[0\]](#)
- [USBTLLHS\\_config Register Description: \[1\] \[2\]](#)

**Table 23-964. USBTLL\_IRQENABLE**

<b>Address Offset</b>	0x0000 001C	<b>Instance</b>	USBTLLHS_config
<b>Physical Address</b>	See <a href="#">Table 23-953</a> .		
<b>Description</b>	OCP standard IRQ enable vector		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												ACCESS_ERROR_EN	FCLK_END_EN	FCLK_START_EN	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x00000000
2	ACCESS_ERROR_EN	Enable IRQ generation upon access error to ULPI register over L3 interconnect 0x0: IRQ event is masked 0x1: IRQ event is enabled	RW	0x0
1	FCLK_END_EN	IRQ event mask for FCLK_END interrupt (see <a href="#">USBTLL_IRQSTATUS[1]</a> ) 0x0: IRQ event is masked 0x1: IRQ event is enabled	RW	0x0
0	FCLK_START_EN	IRQ event mask for FCLK_START interrupt (see <a href="#">USBTLL_IRQSTATUS[0]</a> ) 0x0: IRQ event is masked 0x1: IRQ event is enabled	RW	0x0

**Table 23-965. Register Call Summary for Register USBTLL\_IRQENABLE**

High-Speed Multiport USB Host Subsystem

- [USBTLL Save and Restore: \[0\]](#)
- [HS Multiport USB Host Subsystem Global Initialization: \[1\] \[2\]](#)
- [USBTLLHS\\_config Register Summary: \[3\]](#)

**Table 23-966. TLL\_SHARED\_CONF**

<b>Address Offset</b>	0x0000 0030
<b>Physical Address</b>	See <a href="#">Table 23-953</a> .
<b>Instance</b>	USBTLLHS_config
<b>Description</b>	Common control register for all TLL channels
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												FCLK_REQ	FCLK_IS_ON		

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x00000000
1	FCLK_REQ	Functional clock request, ORed from all channels depending on their respective USB bus state. Combined with the Fclk_is_on status to generate fclk_start/end IRQs. 0x0: Func clock input is not requested by TLL 0x1: Func clock input is requested by TLL	R	0x0

Bits	Field Name	Description	Type	Reset
0	FCLK_IS_ON	Status of the functional clock input, provided by the system to the TLL module. The TLL module will only use that clock if the current status indicated that it is ready. Combined with the Fclk_request to generate fclk_start/end IRQs.  0x0: Functional clock input is not guaranteed ON (can actually be ON, OFF, or unstable) 0x1: Functional clock input is guaranteed ON and stable	RW	0x0

**Table 23-967. Register Call Summary for Register TLL\_SHARED\_CONF**

High-Speed Multiport USB Host Subsystem

- [USBTLL Save and Restore: \[0\]](#)
- [USBTLLHS\\_config Register Summary: \[1\]](#)

**Table 23-968. TLL\_CHANNEL\_CONF\_i**

<b>Address Offset</b>	0x0000 0040 + (0x4 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-953</a> .	<b>Instance</b>	USBTLLHS_config
<b>Description</b>	Control and Status register for channel #N.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		FSLSLINESTATE					FSLSMODE		RESERVED	TESTTXSE0	TESTTXDAT	TESTTXEN	TESTEN	DRVVBUS	CHRGVBUS	RESERVED			ULPINOBITSTUFF	ULPIAUTOIDLE	UTMIAUTOIDLE	RESERVED	ULPIOUTCLKMODE	TLLFULLSPEED	TLLCONNECT	TLLATTACH	UTMISADEV	CHANMODE	CHANEN		

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	FSLSLINESTATE	Line state for Full/Low speed serial modes Bit1 = D- / Bit0 = D+ Read 0x0: Single-ended 0 Read 0x1: Full-Speed J = differential 1 Read 0x2: Full-Speed K = differential 0 Read 0x3: Single-ended 1 (illegal in USB)	R	0x0

Bits	Field Name	Description	Type	Reset
27:24	FSLSMODE	<p>Multiple-mode serial interface's mode select. Only when main channel mode is serial. No effect in other main modes.</p> <p>0x0: 6-pin unidirectional PHY i/f mode. Tx encoding is Dat/Se0 (default)</p> <p>0x1: 6-pin unidirectional PHY i/f mode. Tx encoding is Dp/Dm</p> <p>0x2: 3-pin bidirectional PHY i/f mode.</p> <p>0x3: 4-pin bidirectional PHY i/f mode.</p> <p>0x4: 6-pin unidirectional TLL mode. Tx encoding is Dat/Se0</p> <p>0x5: 6-pin unidirectional TLL mode. Tx encoding is Dp/Dm</p> <p>0x6: 3-pin bidirectional TLL mode.</p> <p>0x7: 4-pin bidirectional TLL mode.</p> <p>0xA: 2-pin bidirectional TLL mode. Encoding is Dat/Se0</p> <p>0xB: 2-pin bidirectional TLL mode. Encoding is Dp/Dm</p>	RW	0x0
23:21	RESERVED		R	0x0
20	TESTTXSE0	<p>Force-Se0 transmit override value for serial mode test Don't care if TestEn = 0 (functional mode) or = TestTxen = 1 (tx = hiz)</p> <p>0x0: drive differential value on Tx according to TestTxDat 0x1: drive SE0 on Tx</p>	RW	0
19	TESTTXDAT	<p>Differential data transmit override value for serial mode test Don't care if TestEn = 0 (functional mode) or = TestTxen = 1 (tx = hiz) or TestSe0 = 1 (tx = se0)</p> <p>0x0: Drive full-speed K = differential 0 0x1: Drive full-speed J = differential 1</p>	RW	0
18	TESTTXEN	<p>Differential data transmit override value for serial mode test Don't care if TestEn = 0 (functional mode)</p> <p>0x0: Drive Tx according to TestTxDat/Se0 0x1: Drive Tx Hiz (no drive: pullups determine line state)</p>	RW	0
17	TESTEN	<p>Enable manual test override for serial mode Tx path (from local controller's UTMI port)</p> <p>0x0: No override. Tx is from local link controller 0x1: Override enabled</p>	RW	0
16	DRVVBUS	<p>VBUS-drive for ChanMode = serial * In TLL config, write 1 to emulate serial-side VBUS drive * In PHY config, write 1 to report "VBUS valid" status (of actual VBUS) to UTMI controller</p> <p>0x0: VBUS not driven 0x1: VBUS driven to 5V</p>	RW	0
15	CHRGVBUS	<p>VBUS-drive for ChanMode = serial * In TLL config, write 1 to emulate serial-side VBUS charge/pullup (OTG) * In PHY config, write 1 to reports "session valid" status (of actual VBUS) to UTMI controller</p> <p>0x0: VBUS not charged, session not valid 0x1: VBUS charged, session valid</p>	RW	0
14:12	RESERVED		R	0x0
11	ULPINOBITSTUFF	<p>Disable bitstuff emulation in ULPI TLL for ULPI ChanMode</p> <p>0x0: Bitstuff enabled, following USB standard 0x1: No bitstuff or associated delays (nonstandard)</p>	RW	0

Bits	Field Name	Description	Type	Reset
10	ULPIAUTOIDLE	For ChanMode = ULPI TLL only. Allow the ULPI output clock to be stopped when ULPI goes into asynchronous mode (low-power, 3-pin serial, 6-pin serial). No effect in ULPI input clock mode.  0x0: ULPI output clock always-on 0x1: ULPI output clock stops during asynchronous ULPI modes	RW	1
9	UTMIAUTOIDLE	For ChanMode = ULPI TLL only. Allow the UTMI clock (output) to be stopped when UTMI goes to suspended mode (suspendm = 0)  0x0: UTMI clock output always on 0x1: UTMI clock output gated upon suspend	RW	1
8	RESERVED		R	0
7	ULPIOUTCLKMODE	ULPI clocking mode select for ULPI TLL ChanMode. Hardcoded, for legacy only.  Read 0x1: ULPI clock provided by PHY side (i.e. TLL, from functional clock). ULPI clock is output	R Rreturns 1s	1
6	TLLFULLSPEED	Sets PHY speed emulation in TLL (full/slow), which determines the line to pull up upon connect. The two connect source controls are: input m(N)_tlpuen, register field TIIConnect.  0x0: Connect is Low-speed: D- pullup 0x1: Connect is Full-Speed: D+ pullup	RW	1
5	TLLCONNECT	Emulation of Full/Low-Speed connect (i.e. D+ resp D-pullup) for serial TLL modes. Speed is determined by field TIIISpeed.  0x0: Unconnected 0x1: Connected	RW	0
4	TLLATTACH	Emulates cable attach/detach for all serial TLL modes: * ChanMode = serial, in TLL mode (FsLsMode) * ChanMode = ULPI, in serial mode (6pin/3pin TLL)  0x0: cable detach emulated on serial TLL 0x1: cable attach emulated on serial TLL	RW	1
3	UTMIISADEV	Select the cable end "seen" by UTMI side of TLL, i.e. the emulated USB cable's orientation. Note that host must always be on A side, Peripheral on B side. Reset value depends on generic DEFUTMIISHOST.  0x0: UTMI side is peripheral, ULPI side is host 0x1: UTMI side is host, ULPI side is peripheral	RW	1
2:1	CHANMODE	Main channel mode selection  0x0: UTMI-to-ULPI TLL mode (HS capable): to ULPI controller 0x1: UTMI-to-serial (FS/LS) mode: to serial controller (TLL) or serial PHY 0x2: Transparent UTMI mode: to UTMI PHY 0x3: No mode selected	RW	0x0
0	CHANEN	Active-high channel enable. A disabled channel is unlocked and kept under reset.  0x0: Channel #N disabled 0x1: Channel #N enabled	RW	0



**Table 23-969. Register Call Summary for Register TLL\_CHANNEL\_CONF\_i**

- High-Speed Multiport USB Host Subsystem
- [USBTLL Save and Restore: \[0\]](#)
  - [USBTLL Channel Configuration: \[1\] \[2\]](#)
  - [VBUS Control and Status for Transceiver \(Non-TLL\) Configurations: \[3\] \[4\] \[5\] \[6\] \[7\]](#)
  - [VBUS Emulation for TLL Configurations: \[8\]](#)
  - [USBTLL Multimode Serial Port: \[9\] \[10\]](#)
  - [USBTLL Attach/Connect Emulation for Serial Modes: \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\]](#)
  - [Operational Modes Configuration \(Selecting and Configuring USB Connectivity\): \[18\]](#)
  - [USBTLLHS\\_config Register Summary: \[19\]](#)

**Table 23-970. USBTLL\_SAR\_CNTX\_j**

<b>Address Offset</b>	0x0000 0400 + (0x4 * j)	<b>Index</b>	j = 0 to 6
<b>Physical Address</b>	See <a href="#">Table 23-953</a> .	<b>Instance</b>	USBTLLHS_config
<b>Description</b>	Save and Restore context array. Array size is indicated in <a href="#">USBTLL_HWINFO</a> . When in SAR mode, read out to save and write to restore. Do not access when not in SAR mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTX																															

Bits	Field Name	Description	Type	Reset
31:0	CNTX	Context bits	RW	0x0000 0000

**Table 23-971. Register Call Summary for Register USBTLL\_SAR\_CNTX\_j**

- High-Speed Multiport USB Host Subsystem
- [USBTLLHS\\_config Register Summary: \[0\]](#)

### 23.11.6.3 USBTLLHS\_ULPI Registers

#### 23.11.6.3.1 USBTLLHS\_ULPI Register Summary

[Table 23-972](#) summarizes the USBTLLHS\_ULPI register mapping.

**Table 23-972. USBTLLHS\_ULPI Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
<a href="#">VENDOR_ID_LO_i<sup>(1)</sup></a>	R	8	0x0000 0000 + (0x100 * i)	0x4A06 2800 + (0x100 * i)
<a href="#">VENDOR_ID_HI_i<sup>(1)</sup></a>	R	8	0x0000 0001 + (0x100 * i)	0x4A06 2801 + (0x100 * i)
<a href="#">PRODUCT_ID_LO_i<sup>(1)</sup></a>	R	8	0x0000 0002 + (0x100 * i)	0x4A06 2802 + (0x100 * i)
<a href="#">PRODUCT_ID_HI_i<sup>(1)</sup></a>	R	8	0x0000 0003 + (0x100 * i)	0x4A06 2803 + (0x100 * i)
<a href="#">FUNCTION_CTRL_i<sup>(1)</sup></a>	RW	8	0x0000 0004 + (0x100 * i)	0x4A06 2804 + (0x100 * i)
<a href="#">FUNCTION_CTRL_SET_i<sup>(1)</sup></a>	R	8	0x0000 0005 + (0x100 * i)	0x4A06 2805 + (0x100 * i)
<a href="#">FUNCTION_CTRL_CLR_i<sup>(1)</sup></a>	R	8	0x0000 0006 + (0x100 * i)	0x4A06 2806 + (0x100 * i)
<a href="#">INTERFACE_CTRL_i<sup>(1)</sup></a>	RW	8	0x0000 0007 + (0x100 * i)	0x4A06 2807 + (0x100 * i)
<a href="#">INTERFACE_CTRL_SET_i<sup>(1)</sup></a>	R	8	0x0000 0008 + (0x100 * i)	0x4A06 2808 + (0x100 * i)
<a href="#">INTERFACE_CTRL_CLR_i<sup>(1)</sup></a>	R	8	0x0000 0009 + (0x100 * i)	0x4A06 2809 + (0x100 * i)
<a href="#">OTG_CTRL_i<sup>(1)</sup></a>	RW	8	0x0000 000A + (0x100 * i)	0x4A06 280A + (0x100 * i)
<a href="#">OTG_CTRL_SET_i<sup>(1)</sup></a>	R	8	0x0000 000B + (0x100 * i)	0x4A06 280B + (0x100 * i)
<a href="#">OTG_CTRL_CLR_i<sup>(1)</sup></a>	R	8	0x0000 000C + (0x100 * i)	0x4A06 280C + (0x100 * i)
<a href="#">USB_INT_EN_RISE_i<sup>(1)</sup></a>	RW	8	0x0000 000D + (0x100 * i)	0x4A06 280D + (0x100 * i)

<sup>(1)</sup> i = 0 to 2

**Table 23-972. USBTLLHS\_ULPI Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
USB_INT_EN_RISE_SET_i <sup>(1)</sup>	R	8	0x0000 000E + (0x100 * i)	0x4A06 280E + (0x100 * i)
USB_INT_EN_RISE_CLR_i <sup>(1)</sup>	R	8	0x0000 000F + (0x100 * i)	0x4A06 280F + (0x100 * i)
USB_INT_EN_FALL_i <sup>(1)</sup>	RW	8	0x0000 0010 + (0x100 * i)	0x4A06 2810 + (0x100 * i)
USB_INT_EN_FALL_SET_i <sup>(1)</sup>	R	8	0x0000 0011 + (0x100 * i)	0x4A06 2811 + (0x100 * i)
USB_INT_EN_FALL_CLR_i <sup>(1)</sup>	R	8	0x0000 0012 + (0x100 * i)	0x4A06 2812 + (0x100 * i)
USB_INT_STATUS_i <sup>(1)</sup>	R	8	0x0000 0013 + (0x100 * i)	0x4A06 2813 + (0x100 * i)
USB_INT_LATCH_i <sup>(1)</sup>	R	8	0x0000 0014 + (0x100 * i)	0x4A06 2814 + (0x100 * i)
DEBUG_i <sup>(1)</sup>	R	8	0x0000 0015 + (0x100 * i)	0x4A06 2815 + (0x100 * i)
SCRATCH_REGISTER_i <sup>(1)</sup>	RW	8	0x0000 0016 + (0x100 * i)	0x4A06 2816 + (0x100 * i)
SCRATCH_REGISTER_SET_i <sup>(1)</sup>	R	8	0x0000 0017 + (0x100 * i)	0x4A06 2817 + (0x100 * i)
SCRATCH_REGISTER_CLR_i <sup>(1)</sup>	R	8	0x0000 0018 + (0x100 * i)	0x4A06 2818 + (0x100 * i)
EXTENDED_SET_ACCESS_i <sup>(1)</sup>	R	8	0x0000 002F + (0x100 * i)	0x4A06 282F + (0x100 * i)
UTMI_VCONTROL_EN_i <sup>(1)</sup>	RW	8	0x0000 0030 + (0x100 * i)	0x4A06 2830 + (0x100 * i)
UTMI_VCONTROL_EN_SET_i <sup>(2)</sup>	R	8	0x0000 0031 + (0x100 * i)	0x4A06 2831 + (0x100 * i)
UTMI_VCONTROL_EN_CLR_i <sup>(2)</sup>	R	8	0x0000 0032 + (0x100 * i)	0x4A06 2832 + (0x100 * i)
UTMI_VCONTROL_STATUS_i <sup>(2)</sup>	RW	8	0x0000 0033 + (0x100 * i)	0x4A06 2833 + (0x100 * i)
UTMI_VCONTROL_LATCH_i <sup>(2)</sup>	R	8	0x0000 0034 + (0x100 * i)	0x4A06 2834 + (0x100 * i)
UTMI_VSTATUS_i <sup>(2)</sup>	RW	8	0x0000 0035 + (0x100 * i)	0x4A06 2835 + (0x100 * i)
UTMI_VSTATUS_SET_i <sup>(2)</sup>	R	8	0x0000 0036 + (0x100 * i)	0x4A06 2836 + (0x100 * i)
UTMI_VSTATUS_CLR_i <sup>(2)</sup>	R	8	0x0000 0037 + (0x100 * i)	0x4A06 2837 + (0x100 * i)
USB_INT_LATCH_NOCLR_i <sup>(2)</sup>	R	8	0x0000 0038 + (0x100 * i)	0x4A06 2838 + (0x100 * i)
VENDOR_INT_EN_i <sup>(2)</sup>	RW	8	0x0000 003B + (0x100 * i)	0x4A06 283B + (0x100 * i)
VENDOR_INT_EN_SET_i <sup>(2)</sup>	R	8	0x0000 003C + (0x100 * i)	0x4A06 283C + (0x100 * i)
VENDOR_INT_EN_CLR_i <sup>(2)</sup>	R	8	0x0000 003D + (0x100 * i)	0x4A06 283D + (0x100 * i)
VENDOR_INT_STATUS_i <sup>(2)</sup>	R	8	0x0000 003E + (0x100 * i)	0x4A06 283E + (0x100 * i)
VENDOR_INT_LATCH_i <sup>(2)</sup>	R	8	0x0000 003F + (0x100 * i)	0x4A06 283F + (0x100 * i)

<sup>(2)</sup> i = 0 to 2**23.11.6.3.2 USBTLLHS\_ULPI Register Description**

through describe the USBTLLHS\_ULPI registers.

**Table 23-973. VENDOR\_ID\_LO\_i**

<b>Address Offset</b>	0x0000 0000 + (0x100 * i)	<b>Index</b>	i = 0 to 2																
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI																
<b>Description</b>	Lower byte of USB-IF-supplied 16-bit vendor ID Value is set for all channels. Default is Texas-Instruments Vendor ID = 0x0451																		
<b>Type</b>	R																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 12.5%;">7</th> <th style="width: 12.5%;">6</th> <th style="width: 12.5%;">5</th> <th style="width: 12.5%;">4</th> <th style="width: 12.5%;">3</th> <th style="width: 12.5%;">2</th> <th style="width: 12.5%;">1</th> <th style="width: 12.5%;">0</th> </tr> </thead> <tbody> <tr> <td colspan="8" style="text-align: center;">VENDOR_ID_LO</td> </tr> </tbody> </table>				7	6	5	4	3	2	1	0	VENDOR_ID_LO							
7	6	5	4	3	2	1	0												
VENDOR_ID_LO																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>																
7:0	VENDOR_ID_LO		R																
			<b>Reset</b>																
			0x51																

**Table 23-974. Register Call Summary for Register VENDOR\_ID\_LO\_i**

- High-Speed Multiport USB Host Subsystem
- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-975. VENDOR\_ID\_HI\_i**

<b>Address Offset</b>	0x0000 0001 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Upper byte of USB-IF-supplied 16-bit vendor ID Value is set for all channels. Default is Texas-Instruments Vendor ID = 0x0451		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
VENDOR_ID_HI							

Bits	Field Name	Description	Type	Reset
7:0	VENDOR_ID_HI		R	0x04

**Table 23-976. Register Call Summary for Register VENDOR\_ID\_HI\_i**

- High-Speed Multiport USB Host Subsystem
- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-977. PRODUCT\_ID\_LO\_i**

<b>Address Offset</b>	0x0000 0002 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Lower byte of 16-bit product ID Value is set for all channels. Default is OMAP4430 Product ID = 0xD00F		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
PRODUCT_ID_LO							

Bits	Field Name	Description	Type	Reset
7:0	PRODUCT_ID_LO		R	0x00

**Table 23-978. Register Call Summary for Register PRODUCT\_ID\_LO\_i**

- High-Speed Multiport USB Host Subsystem
- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-979. PRODUCT\_ID\_HI\_i**

<b>Address Offset</b>	0x0000 0003 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Upper byte of 16-bit product ID Value is set for all channels. Default is OMAP4430 Product ID = 0xD00F		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
PRODUCT_ID_HI							

Bits	Field Name	Description	Type	Reset
7:0	PRODUCT_ID_HI		R	0x00

**Table 23-980. Register Call Summary for Register PRODUCT\_ID\_HI\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-981. FUNCTION\_CTRL\_i**

<b>Address Offset</b>	0x0000 0004 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Controls UTMI function settings of the PHY. Read / Write address.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED	SUSPENDM	RESET	OPMODE		TERMSELECT	XCVRSELECT	

Bits	Field Name	Description	Type	Reset
7	RESERVED		R	0
6	SUSPENDM	Active low PHY suspend: puts the ULPI bus in Low Power Mode. Automatically set back to '1' upon Low Power Mode exit.  0x0: PHY is in low-power mode 0x1: PHY is not in low-power mode	RW	1
5	RESET	Active high UTMI transceiver reset. Auto-cleared. Does not reset the ULPI interface or ULPI register set.  0x0: No ongoing reset/ no action 0x1: Ongoing reset / apply reset	RW	0
4:3	OPMODE	Select the required bit encoding style during transmit  0x0: Normal operation 0x1: Non-driving 0x3: Reserved 0x2: Disable bit-stuff and NRZI encoding	RW	0x0
2	TERMSELECT	Controls the internal 1.5-kΩ HS terminations. Control over bus resistors changes depending on XcvrSelect, OpMode, DpPulldown and DmPulldown.  0x0: HS termination enabled (other conditions) 0x1: FS termination enabled (other conditions)	RW	0

Bits	Field Name	Description	Type	Reset
1:0	XCVRSELECT	Select the required transceiver speed. 0x0: Enable HS transceiver 0x1: Enable FS transceiver 0x3: Enable FS transceiver for LS packets (automatic FS preamble pre-pending) 0x2: Enable LS transceiver	RW	0x1

**Table 23-982. Register Call Summary for Register FUNCTION\_CTRL\_i**

- High-Speed Multiport USB Host Subsystem
- [USBTLL Save and Restore: \[0\]](#)
  - [USBTLLHS\\_ULPI Register Summary: \[1\]](#)

**Table 23-983. FUNCTION\_CTRL\_SET\_i**

<b>Address Offset</b>	0x0000 0005 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Controls UTMI function settings of the PHY. Read / Set address (write 1 to a bit to set it to 1, writing 0 has no effect on bit value). See fields description at the Read/Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED	SUSPENDM	RESET	OPMODE		TERMSELECT	XCVRSELECT	

Bits	Field Name	Description	Type	Reset
7	RESERVED	Reserved	R	0x0
6	SUSPENDM	Active low PHY suspend: Puts the ULPI bus in Low Power Mode. Automatically set back to 1 upon Low Power Mode exit.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
5	RESET	Active high UTMI transceiver reset. Autocleared. Does not reset the ULPI interface or ULPI register set.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
4:3	OPMODE	Select the required bit encoding style during transmit  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
2	TERMSELECT	Controls the internal 1.5kΩ pull-up resistor and 45ohms HS terminations. Control over bus resistors changes depending on XcwrSelect, OpMode, DpPulldown and DmPulldown.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
1:0	XCVRSELECT	Select the required transceiver speed.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0

**Table 23-984. Register Call Summary for Register FUNCTION\_CTRL\_SET\_i**

- High-Speed Multiport USB Host Subsystem
- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-985. FUNCTION\_CTRL\_CLR\_i**

<b>Address Offset</b>	0x0000 0006 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLHS_ULPI
<b>Description</b>	Controls UTMI function settings of the PHY. Read / Clear address (write 1 to a bit to clear it to 0, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED	SUSPENDM	RESET	OPMODE		TERMSELECT	XCVRSELECT	

Bits	Field Name	Description	Type	Reset
7	RESERVED	Reserved	R	0x0
6	SUSPENDM	Active low PHY suspend: Puts the ULPI bus in Low Power Mode. Automatically set back to 1 upon Low Power Mode exit.  Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
5	RESET	Active high UTMI transceiver reset. Autocleared. Does not reset the ULPI interface or ULPI register set.  Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
4:3	OPMODE	Select the required bit encoding style during transmit  Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
2	TERMSELECT	Controls the internal 1.5kΩ pull-up resistor and 45ohms HS terminations. Control over bus resistors changes depending on XcvtSelect, OpMode, DpPulldown and DmPulldown.  Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
1:0	XCVRSELECT	Select the required transceiver speed.  Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0

**Table 23-986. Register Call Summary for Register FUNCTION\_CTRL\_CLR\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-987. INTERFACE\_CTRL\_i**

<b>Address Offset</b>	0x0000 0007 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLHS_ULPI
<b>Description</b>	Enables alternative interfaces and PHY features. Read / Write address.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
INTERFACE_PROTECT_DISABLE	RESERVED		AUTORESUME	CLOCKSUSPENDM	RESERVED	FSLSSERIALMODE_3PIN	FSLSSERIALMODE_6PIN

Bits	Field Name	Description	Type	Reset
7	INTERFACE_PROTECT_DISABLE	Controls circuitry built into the PHY for protecting the ULPI interface when the link tri-states stp and data. 0x0: Enables the interface protect circuit 0x1: Disables the interface protect circuit	RW	0
6:5	RESERVED		R	0x0
4	AUTORESUME	Enables the PHY to automatically drive resume signaling. On by default. 0x0: AutoResume disabled 0x1: AutoResume enabled	RW	1
3	CLOCKSUSPENDM	Active low clock suspend for serial modes (6pin/3-pin). 0x0: ULPI clock will stop during serial modes. 0x1: ULPI clock will run during serial modes.	RW	0
2	RESERVED		R	0
1	FSLSSERIALMODE_3PIN	Sets the ULPI interface to 3-pin (FS/LS only) Serial Mode. Auto-cleared when serial mode is exited. 0x0: ULPI is not in 3-pin mode 0x1: ULPI in 3-pin serial mode	RW	0
0	FSLSSERIALMODE_6PIN	Sets the ULPI interface to 6-pin (FS/LS only) Serial Mode. Auto-cleared when serial mode is exited. 0x0: ULPI is not in 6-pin mode 0x1: ULPI in 6-pin serial mode	RW	0

**Table 23-988. Register Call Summary for Register INTERFACE\_CTRL\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLL Save and Restore: \[0\]](#)
- [USBTLL Channel Configuration: \[1\] \[2\]](#)
- [USBTLLHS\\_ULPI Register Summary: \[3\]](#)

**Table 23-989. INTERFACE\_CTRL\_SET\_i**

<b>Address Offset</b>	0x0000 0008 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Enables alternative interfaces and PHY features. Read / Set address (write 1 to a bit to set it to 1, writing 0 has no effect on bit value). See fields description at the Read/Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0	
INTERFACE_PROTECT_DISABLE	RESERVED		AUTORESUME	CLOCKSSUSPENDM	RESERVED		FSLSSERIALMODE_3PIN	FSLSSERIALMODE_6PIN

Bits	Field Name	Description	Type	Reset
7	INTERFACE_PROTECT_DISABLE	Controls circuitry built into the PHY for protecting the ULPI interface when the link 3-states stp and data.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
6:5	RESERVED	Reserved	R	0x0
4	AUTORESUME	Enables the PHY to automatically drive resume signaling. On by default.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
3	CLOCKSSUSPENDM	Active low clock suspend for serial modes (6pin/3-pin).  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
2	RESERVED	Reserved	R	0x0
1	FSLSSERIALMODE_3PIN	Sets the ULPI interface to 3-pin (FS/LS only) Serial Mode. Autocleared when serial mode is exited.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
0	FSLSSERIALMODE_6PIN	Sets the ULPI interface to 6-pin (FS/LS only) Serial Mode. Autocleared when serial mode is exited.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0

**Table 23-990. Register Call Summary for Register INTERFACE\_CTRL\_SET\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)



**Table 23-991. INTERFACE\_CTRL\_CLR\_i**

<b>Address Offset</b>	0x0000 0009 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Enables alternative interfaces and PHY features. Read / Clear address (write 1 to a bit to clear it to 0, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0	
INTERFACE_PROTECT_DISABLE	RESERVED		AUTORESUME	CLOCKSSUSPENDM	RESERVED		FSLSSERIALMODE_3PIN	FSLSSERIALMODE_6PIN

Bits	Field Name	Description	Type	Reset
7	INTERFACE_PROTECT_DISABLE	Controls circuitry built into the PHY for protecting the ULPI interface when the link 3-states stp and data. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
6:5	RESERVED	Reserved	R	0x0
4	AUTORESUME	Enables the PHY to automatically drive resume signaling. On by default. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
3	CLOCKSSUSPENDM	Active low clock suspend for serial modes (6pin/3-pin). Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
2	RESERVED	Reserved	R	0x0
1	FSLSSERIALMODE_3PIN	Sets the ULPI interface to 3-pin (FS/LS only) Serial Mode. Autocleared when serial mode is exited. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
0	FSLSSERIALMODE_6PIN	Sets the ULPI interface to 6-pin (FS/LS only) Serial Mode. Autocleared when serial mode is exited. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0

**Table 23-992. Register Call Summary for Register INTERFACE\_CTRL\_CLR\_i**

- High-Speed Multiport USB Host Subsystem
- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-993. OTG\_CTRL\_i**

<b>Address Offset</b>	0x0000 000A + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Controls UTMI+ OTG functions of the PHY. Read / Write address.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED		DRVVBUS	CHRGVBUS	DISCHRGVBUS	DMPULLDOWN	DPPULLDOWN	IDPULLUP

Bits	Field Name	Description	Type	Reset
7:6	RESERVED		R	0x0
5	DRVVBUS	Drive 5 V on VBUS 0x0: No action 0x1: Drive VBUS	RW	0
4	CHRGVBUS	Charge VBUS through a resistor for VBUS-pulsing SRP. 0x0: No action 0x1: Charge VBUS	RW	0
3	DISCHRGVBUS	Discharge VBUS through a resistor, until the session-end VBUS state is reached. 0x0: No action 0x1: Discharge VBUS	RW	0
2	DMPULLDOWN	Enables the 15-kΩ pulldown resistor on D- 0x0: Pulldown resistor not connected to D- 0x1: Pulldown resistor connected to D-	RW	1
1	DPPULLDOWN	Enables the 15-kΩ pulldown resistor on D+ 0x0: Pulldown resistor not connected to D+ 0x1: Pulldown resistor connected to D+	RW	1
0	IDPULLUP	Pullup to the (OTG) ID line to allow its sampling 0x0: Disable sampling of ID line. 0x1: Enable sampling of ID line.	RW	0

**Table 23-994. Register Call Summary for Register OTG\_CTRL\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLL Save and Restore: \[0\]](#)
- [VBUS Emulation for TLL Configurations: \[1\] \[2\] \[3\]](#)
- [USBTLLHS\\_ULPI Register Summary: \[4\]](#)

**Table 23-995. OTG\_CTRL\_SET\_i**

<b>Address Offset</b>	0x0000 000B + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLHS_ULPI
<b>Description</b>	Controls UTMI+ OTG functions of the PHY. Read / Set address (write 1 to a bit to set it to 1, writing 0 has no effect on bit value). See fields description at the Read/Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED	DRVVBUS	CHRGVBUS	DISCHRGVBUS	DMPULLDOWN	DPPULLDOWN	IDPULLUP	

Bits	Field Name	Description	Type	Reset
7:6	RESERVED	Reserved	R	0x0
5	DRVVBUS	Drive 5 V on VBUS Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
4	CHRGVBUS	Charge VBUS through a resistor for VBUS-pulsing SRP. Write 0x0: No effect on bit value 0x1: Set the bit to 1.	RW	0x0
3	DISCHRGVBUS	Discharge VBUS through a resistor, until the session-end VBUS state is reached. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
2	DMPULLDOWN	Enables the 15kΩ pull-down resistor on D- Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
1	DPPULLDOWN	Enables the 15kΩ pull-down resistor on D+ Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
0	IDPULLUP	Pull-up to the (OTG) ID line to allow its sampling Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0

**Table 23-996. Register Call Summary for Register OTG\_CTRL\_SET\_i**

- High-Speed Multiport USB Host Subsystem
- [USBTLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-997. OTG\_CTRL\_CLR\_i**

<b>Address Offset</b>	0x0000 000C + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLHS_ULPI
<b>Description</b>	Controls UTMI+ OTG functions of the PHY. Read/clear address (write 1 to a bit to clear it to 0, writing 0 has no effect on bit value). See fields description at the read/write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED	RESERVED	DRVVBUS	CHRGVBUS	DISCHRGVBUS	DMPULLDOWN	DPPULLDOWN	IDPULLUP

Bits	Field Name	Description	Type	Reset
7:6	RESERVED	Reserved	R	0x0
5	DRVVBUS	Drive 5 V on VBUS Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0.	RW	0x0
4	CHRGVBUS	Charge VBUS through a resistor for VBUS-pulsing SRP. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0.	RW	0x0
3	DISCHRGVBUS	Discharge VBUS through a resistor, until the session-end VBUS state is reached. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0.	RW	0x0
2	DMPULLDOWN	Enables the 15k pulldown resistor on D- Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0.	RW	0x0
1	DPPULLDOWN	Enables the 15kΩ pulldown resistor on D+ Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0.	RW	0x0
0	IDPULLUP	Pullup to the (OTG) ID line to allow its sampling Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0.	RW	0x0

**Table 23-998. Register Call Summary for Register OTG\_CTRL\_CLR\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-999. USB\_INT\_EN\_RISE\_i**

<b>Address Offset</b>	0x0000 000D + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Enables an interrupt event notification when the corresponding status bit changes from low to high. By default, all transitions are enabled. Read / Write address.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED			IDGND_RISE	SESEND_RISE	SESSVALID_RISE	VBUSVALID_RISE	HOSTDISCONNECT_RISE

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4	IDGND_RISE	Generate an interrupt event notification when IdGnd changes from low to high. Event is automatically masked if IdPullup bit is clear to 0 and for 50ms after IdPullup is set to 1. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	1
3	SESEND_RISE	Generate an interrupt event notification when SessEnd changes from low to high. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	1
2	SESSVALID_RISE	Generate an interrupt event notification when SessValid changes from low to high. SessValid is the same as UTMI+ AValid. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	1
1	VBUSVALID_RISE	Generate an interrupt event notification when VbusValid changes from low to high. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	1
0	HOSTDISCONNECT_RISE	Generate an interrupt event notification when Hostdisconnect changes from low to high. Applicable only in host mode (DpPulldown and DmPulldown both set to 1b). Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	1

**Table 23-1000. Register Call Summary for Register USB\_INT\_EN\_RISE\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLL Save and Restore: \[0\]](#)
- [USBTLLHS\\_ULPI Register Summary: \[1\]](#)

**Table 23-1001. USB\_INT\_EN\_RISE\_SET\_i**

<b>Address Offset</b>	0x0000 000E + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a>	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Enables an interrupt event notification when the corresponding status bit changes from low to high. Read / Set address (write 1 to a bit to set it to 1, writing 0 has no effect on bit value). See fields description at the Read/Write address of the same register.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
RESERVED			IDGND_RISE	SESEND_RISE	SESSVALID_RISE	VBUSVALID_RISE	HOSTDISCONNECT_RISE

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4	IDGND_RISE	Generate an interrupt event notification when IdGnd changes from low to high. Event is automatically masked if IdPullup bit is clear to 0 and for 50ms after IdPullup is set to 1. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0
3	SESEND_RISE	Generate an interrupt event notification when SessEnd changes from low to high. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0
2	SESSVALID_RISE	Generate an interrupt event notification when SessValid changes from low to high. SessValid is the same as UTMI+ AValid. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0
1	VBUSVALID_RISE	Generate an interrupt event notification when VbusValid changes from low to high. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0
0	HOSTDISCONNECT_RISE	Generate an interrupt event notification when Hostdisconnect changes from low to high. Applicable only in host mode (DpPulldown and DmPulldown both set to 1b). Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0

**Table 23-1002. Register Call Summary for Register USB\_INT\_EN\_RISE\_SET\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1003. USB\_INT\_EN\_RISE\_CLR\_i**

<b>Address Offset</b>	0x0000 000F + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLHS_ULPI
<b>Description</b>	Enables an interrupt event notification when the corresponding status bit changes from low to high. Read / Clear address (write 1 to a bit to clear it to 0, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED			IDGND_RISE	SESEND_RISE	SESSVALID_RISE	VBUSVALID_RISE	HOSTDISCONNECT_RISE

Bits	Field Name	Description	Type	Reset
7:5	RESERVED	Reserved	R	0x0
4	IDGND_RISE	Generate an interrupt event notification when IdGnd changes from low to high. Event is automatically masked if IdPullup bit is clear to 0 and for 50ms after IdPullup is set to 1. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
3	SESEND_RISE	Generate an interrupt event notification when SessEnd changes from low to high. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
2	SESSVALID_RISE	Generate an interrupt event notification when SessValid changes from low to high. SessValid is the same as UTMI+ AValid. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
1	VBUSVALID_RISE	Generate an interrupt event notification when VbusValid changes from low to high. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
0	HOSTDISCONNECT_RISE	Generate an interrupt event notification when Hostdisconnect changes from low to high. Applicable only in host mode (DpPulldown and DmPulldown both set to 1b). Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0

**Table 23-1004. Register Call Summary for Register USB\_INT\_EN\_RISE\_CLR\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1005. USB\_INT\_EN\_FALL\_i**

<b>Address Offset</b>	0x0000 0010 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Enables an interrupt event notification when the corresponding status bit changes from high to low. By default, all transitions are enabled. Read / Write address.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED			IDGND_FALL	SESEND_FALL	SESSVALID_FALL	VBUSVALID_FALL	HOSTDISCONNECT_FALL

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4	IDGND_FALL	Generate an interrupt event notification when IdGnd changes from high to low. Event is automatically masked if IdPullup bit is clear to 0 and for 50ms after IdPullup is set to 1.	RW	1
3	SESEND_FALL	Generate an interrupt event notification when SessEnd changes from high to low.	RW	1
2	SESSVALID_FALL	Generate an interrupt event notification when SessValid changes from high to low. SessValid is the same as UTMI+ AValid.	RW	1
1	VBUSVALID_FALL	Generate an interrupt event notification when VbusValid changes from high to low.	RW	1
0	HOSTDISCONNECT_FALL	Generate an interrupt event notification when Hostdisconnect changes from high to low. Applicable only in host mode (DpPulldown and DmPulldown both set to 1b).	RW	1

**Table 23-1006. Register Call Summary for Register USB\_INT\_EN\_FALL\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLL Save and Restore: \[0\]](#)
- [USBTLLHS\\_ULPI Register Summary: \[1\]](#)

**Table 23-1007. USB\_INT\_EN\_FALL\_SET\_i**

<b>Address Offset</b>	0x0000 0011 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Enables an interrupt event notification when the corresponding status bit changes from high to low. Read / Set address (write 1 to a bit to set it to 1, writing 0 has no effect on bit value). See fields description at the Read/Write address of the same register.		
<b>Type</b>	R		



7	6	5	4	3	2	1	0
RESERVED			IDGND_RISE	SESEND_RISE	SESSVALID_RISE	VBUSVALID_RISE	HOSTDISCONNECT_RISE

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4	IDGND_RISE	Generate an interrupt event notification when IdGnd changes from low to high. Event is automatically masked if IdPullup bit is clear to 0 and for 50ms after IdPullup is set to 1.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0
3	SESEND_RISE	Generate an interrupt event notification when SessEnd changes from low to high.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0
2	SESSVALID_RISE	Generate an interrupt event notification when SessValid changes from low to high. SessValid is the same as UTMI+ AValid.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0
1	VBUSVALID_RISE	Generate an interrupt event notification when VbusValid changes from low to high.  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0
0	HOSTDISCONNECT_RISE	Generate an interrupt event notification when Hostdisconnect changes from low to high. Applicable only in host mode (DpPulldown and DmPulldown both set to 1b).  Write 0x0: No effect on bit value Write 0x1: Set the bit to 1	RW	0x0

**Table 23-1008. Register Call Summary for Register USB\_INT\_EN\_FALL\_SET\_i**

- High-Speed Multiport USB Host Subsystem
- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1009. USB\_INT\_EN\_FALL\_CLR\_i**

<b>Address Offset</b>	0x0000 0012 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLHS_ULPI
<b>Description</b>	Enables an interrupt event notification when the corresponding status bit changes from high to low. Read / Clear address (write 1 to a bit to clear it to 0, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
RESERVED			IDGND_RISE	SESEND_RISE	SESSVALID_RISE	VBUSVALID_RISE	HOSTDISCONNECT_RISE

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4	IDGND_RISE	Generate an interrupt event notification when IdGnd changes from low to high. Event is automatically masked if IdPullup bit is clear to 0 and for 50ms after IdPullup is set to 1. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
3	SESEND_RISE	Generate an interrupt event notification when SessEnd changes from low to high. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
2	SESSVALID_RISE	Generate an interrupt event notification when SessValid changes from low to high. SessValid is the same as UTMI+ AValid. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
1	VBUSVALID_RISE	Generate an interrupt event notification when VbusValid changes from low to high. Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0
0	HOSTDISCONNECT_RISE	Generate an interrupt event notification when Hostdisconnect changes from low to high. Applicable only in host mode (DpPulldown and DmPulldown both set to 1b). Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0

**Table 23-1010. Register Call Summary for Register USB\_INT\_EN\_FALL\_CLR\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1011. USB\_INT\_STATUS\_i**

<b>Address Offset</b>	0x0000 0013 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Indicates the current value of the interrupt source signal.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
RESERVED			IDGND	SESSEND	SESSVALID	VBUSVALID	HOSTDISCONNECT

Bits	Field Name	Description	Type	Reset
7:5	RESERVED	Reserved	R	0x0
4	IDGND	Value of UTMI+ IdDig output. Undefined unless IdPullup = 1 Read 0x1: ID pin is floating = OTG B = default Peripheral Read 0x0: ID pin is grounded = OTG A = default Host	R	0
3	SESSEND	Current value of UTMI+ SessEnd output. Read 0x1: VBUS is below Session-End threshold Read 0x0: VBUS is above Session-End threshold	R	0
2	SESSVALID	Current value of UTMI+ SessValid output. SessValid is the same as UTMI+ AValid. Read 0x1: VBUS is above Session-Valid threshold Read 0x0: VBUS is below Session-Valid threshold	R	0
1	VBUSVALID	Current value of UTMI+ VbusValid output. Read 0x1: VBUS is above Vbus-Valid threshold Read 0x0: VBUS is below Vbus-Valid threshold	R	0
0	HOSTDISCONNECT	Current value of UTMI+ Hostdisconnect output. Applicable only in host mode. Automatically reset to 0 when Low Power Mode is entered. Read 0x1: Peripheral disconnected Read 0x0: Peripheral not disconnected or non-host mode	R	0

**Table 23-1012. Register Call Summary for Register USB\_INT\_STATUS\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLL Save and Restore: \[0\]](#)
- [VBUS Emulation for TLL Configurations: \[1\] \[2\] \[3\]](#)
- [USBTLLHS\\_ULPI Register Summary: \[4\]](#)

**Table 23-1013. USB\_INT\_LATCH\_i**

<b>Address Offset</b>	0x0000 0014 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Set by unmasked changes on the corresponding status bits to generate the ULPI interrupt. Cleared upon read, and when Low Power Mode, Serial Mode or CarKit Mode are entered.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
RESERVED			IDGND_LATCH	SESEND_LATCH	SESSVALID_LATCH	VBUSVALID_LATCH	HOSTDISCONNECT_LATCH

Bits	Field Name	Description	Type	Reset
7:5	RESERVED	Reserved	R	0x0
4	IDGND_LATCH	Set to 1 by the PHY when an unmasked event occurs on IdGnd. Cleared when this register is read.	R	0
3	SESEND_LATCH	Set to 1 by the PHY when an unmasked event occurs on SessEnd. Cleared when this register is read.	R	0
2	SESSVALID_LATCH	Set to 1 by the PHY when an unmasked event occurs on SessValid. Cleared when this register is read. SessValid is the same as UTMI+ AValid.	R	0
1	VBUSVALID_LATCH	Set to 1 by the PHY when an unmasked event occurs on VbusValid. Cleared when this register is read.	R	0
0	HOSTDISCONNECT_LATCH	Set to 1 by the PHY when an unmasked event occurs on Hostdisconnect. Cleared when this register is read. Applicable only in host mode.	R	0

**Table 23-1014. Register Call Summary for Register USB\_INT\_LATCH\_i**

- High-Speed Multiport USB Host Subsystem
- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1015. DEBUG\_i**

<b>Address Offset</b>	0x0000 0015 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Indicates the current value of various signals useful for debugging.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
RESERVED						LINE STATE	

Bits	Field Name	Description	Type	Reset
7:2	RESERVED		R	0x00
1:0	LINE STATE	Current state of the USB line: D+ (bit 0) and D- (bit 1). Read 0x3: SE1 (LS/FS), Invalid (HS/Chirp) Read 0x2: LS: 'J' State, FS: 'K' State, HS: Invalid, Chirp: !Squelch & !HS_Differential_Receiver_Output Read 0x1: LS: 'K' State, FS: 'J' State, HS: !Squelch, Chirp: !Squelch & HS_Differential_Receiver_Output Read 0x0: SE0 (LS/FS), Squelch (HS/Chirp)	R	0x0

**Table 23-1016. Register Call Summary for Register DEBUG\_i**

High-Speed Multiport USB Host Subsystem  
 • [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1017. SCRATCH\_REGISTER\_i**

<b>Address Offset</b>	0x0000 0016 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Register byte for register access testing purposes. Value has no functional effect on PHY. Read / Write address.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
SCRATCH							

Bits	Field Name	Description	Type	Reset
7:0	SCRATCH	Scratch data.	RW	0x00

**Table 23-1018. Register Call Summary for Register SCRATCH\_REGISTER\_i**

High-Speed Multiport USB Host Subsystem  
 • [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1019. SCRATCH\_REGISTER\_SET\_i**

<b>Address Offset</b>	0x0000 0017 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Register byte for register access testing purposes. Value has no functional effect on PHY. Read / Set address (write 1 to a bit to set it to 1, writing 0 has no effect on bit value). See fields description at the Read/Write address of the same register.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
SCRATCH							

Bits	Field Name	Description	Type	Reset
7:0	SCRATCH	Scratch data write 1 to a bit to set it to 1 writing 0 has no effect on bit value	RW	0x00

**Table 23-1020. Register Call Summary for Register SCRATCH\_REGISTER\_SET\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1021. SCRATCH\_REGISTER\_CLR\_i**

<b>Address Offset</b>	0x0000 0018 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Register byte for register access testing purposes. Value has no functional effect on PHY. Read / Clear address (write 1 to a bit to clear it to 0, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
SCRATCH							

Bits	Field Name	Description	Type	Reset
7:0	SCRATCH	Scratch data write 1 to a bit to clear it to 0 writing 0 has no effect on bit value	RW	0x00

**Table 23-1022. Register Call Summary for Register SCRATCH\_REGISTER\_CLR\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1023. EXTENDED\_SET\_ACCESS\_i**

<b>Address Offset</b>	0x0000 002F + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	This address is used to access the extended register set, i.e. addresses above 0x40		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
SET_ACCESS							

Bits	Field Name	Description	Type	Reset
7:0	SET_ACCESS	This bitfield is used to access the extended register set, that is, addresses above 0x40.	RW	0x00

**Table 23-1024. Register Call Summary for Register EXTENDED\_SET\_ACCESS\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1025. UTMI\_VCONTROL\_EN\_i**

<b>Address Offset</b>	0x0000 0030 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Part of nonstandard UTMI-to-ULPI mailbox system, implemented if HDL generic VCS_MAILBOX bit is 1 Enables an interrupt notification when the corresponding vcontrol_status bit changes. Read / Write address. Lowest VCS_CTRL_WIDTH (HDL generic) bits are implemented, others are always-0, read-only. (UTMI standard is 4-bit).		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
VC7_EN	VC6_EN	VC5_EN	VC4_EN	VC3_EN	VC2_EN	VC1_EN	VC0_EN

Bits	Field Name	Description	Type	Reset
7	VC7_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
6	VC6_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
5	VC5_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
4	VC4_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
3	VC3_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
2	VC2_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
1	VC1_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
0	VC0_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0

**Table 23-1026. Register Call Summary for Register UTMI\_VCONTROL\_EN\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1027. UTMI\_VCONTROL\_EN\_SET\_i**

<b>Address Offset</b>	0x0000 0031 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Part of nonstandard UTMI-to-ULPI mailbox system, implemented if HDL generic VCS_MAILBOX bit is 1 Enables an interrupt notification when the corresponding vcontrol_status bit changes. Read / Set address (write 1 to a bit to set it to 1, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
VC7_EN	VC6_EN	VC5_EN	VC4_EN	VC3_EN	VC2_EN	VC1_EN	VC0_EN

Bits	Field Name	Description	Type	Reset
7	VC7_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
6	VC6_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
5	VC5_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
4	VC4_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
3	VC3_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
2	VC2_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
1	VC1_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0
0	VC0_EN	Enable alt_int assertion upon vcontrol_status bit change Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0

**Table 23-1028. Register Call Summary for Register UTMI\_VCONTROL\_EN\_SET\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1029. UTMI\_VCONTROL\_EN\_CLR\_i**

<b>Address Offset</b>	0x0000 0032 + (0x100 * i)	<b>Index</b>	i = 0 to 2																
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI																
<b>Description</b>	Part of nonstandard UTMI-to-ULPI mailbox system, implemented if HDL generic VCS_MAILBOX bit is 1 Enables an interrupt notification when the corresponding vcontrol_status bit changes. Read / Clear address (write 1 to a bit to clear it to 0, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.																		
<b>Type</b>	RW																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 12.5%;">7</th> <th style="width: 12.5%;">6</th> <th style="width: 12.5%;">5</th> <th style="width: 12.5%;">4</th> <th style="width: 12.5%;">3</th> <th style="width: 12.5%;">2</th> <th style="width: 12.5%;">1</th> <th style="width: 12.5%;">0</th> </tr> </thead> <tbody> <tr> <td colspan="8" style="text-align: center;">VC</td> </tr> </tbody> </table>				7	6	5	4	3	2	1	0	VC							
7	6	5	4	3	2	1	0												
VC																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>															
7:0	VC	User-defined UTMI Control data byte	RW	0x00															

**Table 23-1030. Register Call Summary for Register UTMI\_VCONTROL\_EN\_CLR\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)



**Table 23-1031. UTMI\_VCONTROL\_STATUS\_i**

<b>Address Offset</b>	0x0000 0033 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Part of nonstandard UTMI-to-ULPI mailbox system, implemented if HDL generic VCS_MAILBOX bit is 1. UTMI-standard Vcontrol vector byte is sent by the UTMI controller (other side of TLL) to its PHY (emulated here by the TLL). Alternatively, data can be also written directly into the register. Can contain any user-defined data. Vcontrol bit changes can be used to assert the ULPI ALT interrupt. Lowest VCS_CTRL_WIDTH (HDL generic) bits are implemented, others are always-0, read-only. (UTMI standard is 4-bit).		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
VC							

Bits	Field Name	Description	Type	Reset
7:0	VC	User-defined UTMI Control data byte	RW	0x00

**Table 23-1032. Register Call Summary for Register UTMI\_VCONTROL\_STATUS\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1033. UTMI\_VCONTROL\_LATCH\_i**

<b>Address Offset</b>	0x0000 0034 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Part of nonstandard UTMI-to-ULPI mailbox system, implemented if HDL generic VCS_MAILBOX bit is 1. Set by unmasked changes on the corresponding vcontrol_status bits to generate the ULPI ALT interrupt. Cleared upon read, and when Low Power Mode, Serial Mode or CarKit Mode are entered. Lowest VCS_CTRL_WIDTH (HDL generic) bits are implemented, others are always-0, read-only. (UTMI standard is 4-bit).		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
VC7_CHANGE	VC6_CHANGE	VC5_CHANGE	VC4_CHANGE	VC3_CHANGE	VC2_CHANGE	VC1_CHANGE	VC0_CHANGE

Bits	Field Name	Description	Type	Reset
7	VC7_CHANGE	Unmasked change on vcontrol_status bit	R	0
6	VC6_CHANGE	Unmasked change on vcontrol_status bit	R	0
5	VC5_CHANGE	Unmasked change on vcontrol_status bit	R	0
4	VC4_CHANGE	Unmasked change on vcontrol_status bit	R	0
3	VC3_CHANGE	Unmasked change on vcontrol_status bit	R	0
2	VC2_CHANGE	Unmasked change on vcontrol_status bit	R	0
1	VC1_CHANGE	Unmasked change on vcontrol_status bit	R	0
0	VC0_CHANGE	Unmasked change on vcontrol_status bit	R	0

**Table 23-1034. Register Call Summary for Register UTMI\_VCONTROL\_LATCH\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1035. UTMI\_VSTATUS\_i**

<b>Address Offset</b>	0x0000 0035 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Part of nonstandard UTMI-to-ULPI mailbox system, implemented if HDL generic VCS_MAILBOX bit is 1. UTMI-standard Vstatus vector byte is sent by the PHY (emulated here by the TLL) to the UTMI controller (other side of TLL): information written into this register will go directly to the UTMI controller, and can contain any user-defined data. Read / Write address. Lowest VCS_STAT_WIDTH (HDL generic) bits are implemented, others are always-0, read-only. (UTMI standard is 8-bit).		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
VS							

Bits	Field Name	Description	Type	Reset
7:0	VS	User-defined UTMI Status data byte	RW	0x00

**Table 23-1036. Register Call Summary for Register UTMI\_VSTATUS\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1037. UTMI\_VSTATUS\_SET\_i**

<b>Address Offset</b>	0x0000 0036 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Part of nonstandard UTMI-to-ULPI mailbox system, implemented if HDL generic VCS_MAILBOX bit is 1. UTMI-standard Vstatus vector byte is sent by the PHY (emulated here by the TLL) to the UTMI controller (other side of TLL): information written into this register will go directly to the UTMI controller, and can contain any user-defined data. Read / Set address (write 1 to a bit to set it to 1, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
VS							

Bits	Field Name	Description	Type	Reset
7:0	VS	User-defined UTMI status data byte Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x00

**Table 23-1038. Register Call Summary for Register UTMI\_VSTATUS\_SET\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1039. UTMI\_VSTATUS\_CLR\_i**

<b>Address Offset</b>	0x0000 0037 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Part of nonstandard UTMI-to-ULPI mailbox system, implemented if HDL generic VCS_MAILBOX bit is 1. UTMI-standard Vstatus vector byte is sent by the PHY (emulated here by the TLL) to the UTMI controller (other side of TLL): information written into this register will go directly to the UTMI controller, and can contain any user-defined data. Read / Clear address (write 1 to a bit to clear it to 0, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
VS							

Bits	Field Name	Description	Type	Reset
7:0	VS	User-defined UTMI status data byte Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x00

**Table 23-1040. Register Call Summary for Register UTMI\_VSTATUS\_CLR\_i**

- High-Speed Multiport USB Host Subsystem
- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1041. USB\_INT\_LATCH\_NOCLR\_i**

<b>Address Offset</b>	0x0000 0038 + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Set by unmasked changes on the corresponding status bits to generate the ULPI interrupt. Debug, nonstandard address to the standard register: Register is not cleared on read. See fields description at the "clear-on-read" address of the same register.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
RESERVED			IDGND_LATCH	SESEND_LATCH	SESSVALID_LATCH	VBUSVALID_LATCH	HOSTDISCONNECT_LATCH

Bits	Field Name	Description	Type	Reset
7:5	RESERVED	Reserved	R	0x0
4	IDGND_LATCH	Set to 1 by the PHY when an unmasked event occurs on IdGnd.	R	0x0
3	SESEND_LATCH	Set to 1 by the PHY when an unmasked event occurs on SessEnd.	R	0x0
2	SESSVALID_LATCH	Set to 1 by the PHY when an unmasked event occurs on SessValid. SessValid is the same as UTMI+ AValid.	R	0x0
1	VBUSVALID_LATCH	Set to 1 by the PHY when an unmasked event occurs on VbusValid.	R	0x0
0	HOSTDISCONNECT_LATCH	Set to 1 by the PHY when an unmasked event occurs on Hostdisconnect. Applicable only in host mode.	R	0x0

**Table 23-1042. Register Call Summary for Register USB\_INT\_LATCH\_NOCLR\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1043. VENDOR\_INT\_EN\_i**

<b>Address Offset</b>	0x0000 003B + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Vendor-specific interrupt enables (mask) for miscellaneous ULPI alt_int events. Read / Write address.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED							P2P_EN

Bits	Field Name	Description	Type	Reset
7:1	RESERVED		R	0x00
0	P2P_EN	Enable phy-to-phy ULPI wake-up upon inactive UTMI suspendm. 0x0: phy-to-phy wake-up enabled 0x1: phy-to-phy wake-up enabled	RW	0

**Table 23-1044. Register Call Summary for Register VENDOR\_INT\_EN\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1045. VENDOR\_INT\_EN\_SET\_i**

<b>Address Offset</b>	0x0000 003C + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBLLHS_ULPI
<b>Description</b>	Vendor-specific interrupt enable bit (mask) for miscellaneous ULPI alt_int events. Read / Set address (write 1 to a bit to set it to 1, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	RW		

7	6	5	4	3	2	1	0
RESERVED							P2P_EN

Bits	Field Name	Description	Type	Reset
7:1	RESERVED	Reserved	R	0x00
0	P2P_EN	Enable PHY-to-PHY ULPI wakeup upon inactive UTMI suspendm. Write 0x0: No effect on bit value Write 0x1: Set the bit to 1.	RW	0x0

**Table 23-1046. Register Call Summary for Register VENDOR\_INT\_EN\_SET\_i**

High-Speed Multiport USB Host Subsystem

- [USBLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1047. VENDOR\_INT\_EN\_CLR\_i**

<b>Address Offset</b>	0x0000 003D + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Vendor-specific interrupt enables (mask) for miscellaneous ULPI alt_int events. Read / Clear address (write 1 to a bit to clear it to 0, writing 0 has no effect on bit value). See fields description at the Read / Write address of the same register.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
RESERVED							P2P_EN

Bits	Field Name	Description	Type	Reset
7:1	RESERVED	Reserved	R	0x00
0	P2P_EN	Enable PHY-to-PHY ULPI wakeup upon inactive UTMI suspendm.  Write 0x0: No effect on bit value Write 0x1: Clear the bit to 0	RW	0x0

**Table 23-1048. Register Call Summary for Register VENDOR\_INT\_EN\_CLR\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1049. VENDOR\_INT\_STATUS\_i**

<b>Address Offset</b>	0x0000 003E + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLLHS_ULPI
<b>Description</b>	Vendor-specific interrupt sources for miscellaneous ULPI alt_int events.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
RESERVED							UTMI_SUSPENDM

Bits	Field Name	Description	Type	Reset
7:1	RESERVED		R	0x00
0	UTMI_SUSPENDM	UTMI suspendm status (active-low), source of TLL phy-to-phy wake-up interrupt.  Read 0x1: UTMI interface is active (not suspended) Read 0x0: UTMI interface is suspended	R	1

**Table 23-1050. Register Call Summary for Register VENDOR\_INT\_STATUS\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLLHS\\_ULPI Register Summary: \[0\]](#)

**Table 23-1051. VENDOR\_INT\_LATCH\_i**

<b>Address Offset</b>	0x0000 003F + (0x100 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See <a href="#">Table 23-972</a> .	<b>Instance</b>	USBTLHLS_ULPI
<b>Description</b>	Vendor-specific interrupt latches for miscellaneous ULPI alt_int events. Cleared upon read, and when Low Power Mode, Serial Mode or CarKit Mode are entered.		
<b>Type</b>	R		

7	6	5	4	3	2	1	0
RESERVED							P2P_LATCH

Bits	Field Name	Description	Type	Reset
7:1	RESERVED		R	0x00
0	P2P_LATCH	Phy-to-phy ULPI wake-up event latch. Set when ULPI is in low-power mode (suspendm = 0) and UTMI is active (suspendm = 1).  Read 0x1: Phy-to-phy wake-up event was latched, ALT interrupt active  Read 0x0: No phy-to-phy wake-up event latched	R	0

**Table 23-1052. Register Call Summary for Register VENDOR\_INT\_LATCH\_i**

High-Speed Multiport USB Host Subsystem

- [USBTLHLS\\_ULPI Register Summary: \[0\]](#)

### 23.11.6.4 HSUSBHOST Controller Registers

#### 23.11.6.4.1 HSUSBHOST Controller Register Summary

[Table 23-1053](#) summarizes the HSUSBHOST register mapping.

**Table 23-1053. HSUSBHOST Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
<a href="#">UHH_REVISION</a>	R	32	0x0000 0000	0x4A06 4000
<a href="#">UHH_HWINFO</a>	R	32	0x0000 0004	0x4A06 4004
<a href="#">UHH_SYSCONFIG</a>	RW	32	0x0000 0010	0x4A06 4010
<a href="#">UHH_SYSSTATUS</a>	R	32	0x0000 0014	0x4A06 4014
<a href="#">UHH_HOSTCONFIG</a>	RW	32	0x0000 0040	0x4A06 4040
<a href="#">UHH_DEBUG_CSR</a>	RW	32	0x0000 0044	0x4A06 4044
<a href="#">UHH_SAR_CNTX_i<sup>(1)</sup></a>	RW	32	0x0000 0100 + (0x4 * i)	0x4A06 4100 + (0x4 * i)

<sup>(1)</sup> i = 0 to 383**CAUTION**

The UHH\_config registers are limited to 32-bit data accesses; 16-bit and 8-bit accesses are not allowed and can corrupt register content.

#### 23.11.6.4.2 HSUSB Controller Register Description

through describe the HSUSB controller registers.

**Table 23-1054. UHH\_REVISION**

<b>Address Offset</b>	0x0000 0000																																																																																																
<b>Physical Address</b>	See <a href="#">Table 23-1053</a> .																<b>Instance</b>																HSUSBHOST																																																																
<b>Description</b>	USB High-speed Host (UHH) Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility																																																																																																
<b>Type</b>	R																																																																																																
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">REVISION</td> <td colspan="17"></td> </tr> </table>																																	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	REVISION																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																		
REVISION																																																																																																	

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI Internal data

**Table 23-1055. Register Call Summary for Register UHH\_REVISION**

- High-Speed Multiport USB Host Subsystem
- [HSUSBHOST Controller Register Summary: \[0\]](#)

**Table 23-1056. UHH\_HWINFO**

<b>Address Offset</b>	0x0000 0004																																																																																																
<b>Physical Address</b>	See <a href="#">Table 23-1053</a> .																<b>Instance</b>																HSUSBHOST																																																																
<b>Description</b>	Information on host's hardware configuration																																																																																																
<b>Type</b>	R																																																																																																
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="17">SAR_CNTX_SIZE</td> </tr> </table>																																	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																SAR_CNTX_SIZE																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																		
RESERVED																SAR_CNTX_SIZE																																																																																	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9:0	SAR_CNTX_SIZE	Save-and-Restore context size, in 32-bit words, i.e. number of 32-bit registers with significant context information, mapped from offset 0x100 upward.	R	0x180

**Table 23-1057. Register Call Summary for Register UHH\_HWINFO**

- High-Speed Multiport USB Host Subsystem
- [HSUSBHOST Controller Register Summary: \[0\]](#)
  - [HSUSB Controller Register Description: \[1\]](#)

**Table 23-1058. UHH\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010																																																																																			
<b>Physical Address</b>	See <a href="#">Table 23-1053</a> .																<b>Instance</b>																HSUSBHOST																																																			
<b>Description</b>	OCP standard system configuration register																																																																																			
<b>Type</b>	RW																																																																																			
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="1">STANDBYMODE</td> <td colspan="1">IDLEMODE</td> <td colspan="1">RESERVED</td> <td colspan="1">SOFTRESET</td> </tr> </table>																																	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																STANDBYMODE	IDLEMODE	RESERVED	SOFTRESET
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																					
RESERVED																STANDBYMODE	IDLEMODE	RESERVED	SOFTRESET																																																																	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x00000000
5:4	STANDBYMODE	Mstandby/Mwait/[Mwakeup] initiator power management interface configuration.  0x0: Force-standby mode. Mstandby asserted unconditionally. Asynchronous (Master) wake-up disabled.  0x1: No-standby mode. Mstandby never asserted. Asynchronous (Master) wake-up disabled.  0x2: Smart-standby mode. Mstandby asserted when initiator activity stops. Asynchronous (Master) wake-up disabled.  0x3: Smart-standby + wake-up mode, like smart mode with asynchronous (master) wake-up enabled.	RW	0x2
3:2	IDLEMODE	Sidlereq/Sidleack(1:0)/[Swakeup] target power management interface configuration.  0x0: Force-Idle mode. Sidleack asserted after Idlereq assertion. Asynchronous (Slave) wake-up disabled.  0x1: No-idle mode. Sidleack never asserted. Asynchronous (Slave) wake-up disabled.  0x2: Smart-idle mode. Sidleack asserted upon Idlereq assertion, after target activity is over. Asynchronous (Slave) wake-up disabled.  0x3: Smart-idle wake-up mode. Like smart mode with asynchronous (slave) wake-up enabled.	RW	0x2
1	RESERVED		R	0
0	SOFTRESET	Module software reset  Read 0x0: No reset pending Write 0x0: no effect Write 0x1: Starts softreset sequence. Read 0x1: Reset (soft or other) is pending	RW	0

**Table 23-1059. Register Call Summary for Register UHH\_SYSCONFIG**

High-Speed Multiport USB Host Subsystem

- [HS USB Host Controller Software Reset: \[0\]](#)
- [Standby Handshake Protocol: \[1\]](#)
- [IDLE Handshake Protocol: \[2\]](#)
- [HS Multiport USB Host Subsystem Global Initialization: \[3\]](#)
- [HSUSBHOST Controller Register Summary: \[4\]](#)

**Table 23-1060. UHH\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014																																																																																													
<b>Physical Address</b>	See <a href="#">Table 23-1053</a> .																<b>Instance</b>								HSUSBHOST																																																																					
<b>Description</b>	Module-specific system status																																																																																													
<b>Type</b>	R																																																																																													
<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> </thead> <tbody> <tr> <td colspan="28" style="text-align: center;">RESERVED</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">EHCI_RESETDONE</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">OHCI_RESETDONE</td> <td style="writing-mode: vertical-rl; transform: rotate(180deg);">RESERVED</td> </tr> </tbody> </table>																																31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																												EHCI_RESETDONE	OHCI_RESETDONE	RESERVED
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																															
RESERVED																												EHCI_RESETDONE	OHCI_RESETDONE	RESERVED																																																																



Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	EHCI_RESETDONE	Indicates when the EHCI HS host is out of reset Read 0x0: Out of reset Read 0x1: Under reset	R	1
1	OHCI_RESETDONE	Indicates when the OHCI FS/LS host is out of reset Read 0x0: Out of reset Read 0x1: Under reset	R	1
0	RESERVED		R	0

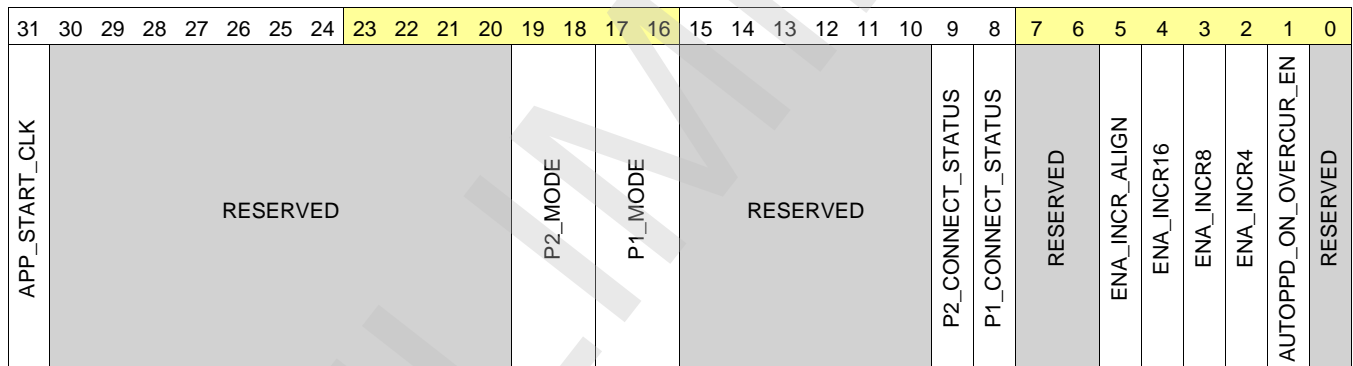
**Table 23-1061. Register Call Summary for Register UHH\_SYSSTATUS**

High-Speed Multiport USB Host Subsystem

- [HS Multiport USB Host Subsystem Global Initialization: \[0\] \[1\]](#)
- [HSUSBHOST Controller Register Summary: \[2\]](#)

**Table 23-1062. UHH\_HOSTCONFIG**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	HSUSBHOST
<b>Physical Address</b>	See <a href="#">Table 23-1053</a> .		
<b>Description</b>	Static configuration of the USB HS host		
<b>Type</b>	RW		



Bits	Field Name	Description	Type	Reset
31	APP_START_CLK	When the OHCI clocks are suspended, the system has to assert this signal to start the clocks (12 and 48 MHz). This must be deasserted after the clocks are started and before the host is suspended again. (Host is suspended means HCFS = SUSPEND or all the OHCI ports are suspended.)	RW	0
30:20	RESERVED		R	0X000
19:18	P2_MODE	Port 2 interface configuration. Each bit corresponds to an internal "strap" signal, and output: Bit 0 = ulpi_bypass Bit 1 = hsic_en  0x0: To external ULPI PHY, HS only 0x1: To UTMI PHY (or ULPI TLL), FS/LS capable 0x2: Forbidden 0x3: To HSIC digital front-end (DFE), HS only	RW	0x0

Bits	Field Name	Description	Type	Reset
17:16	P1_MODE	Port 1 interface configuration. Each bit corresponds to an internal "strap" signal, and output: Bit 0 = ulpi_bypass Bit 1 = hsic_en  0x0: To external ULPI PHY, HS only 0x1: To UTMI PHY (or ULPI TLL), FS/LS capable 0x2: Forbidden 0x3: To HSIC digital front-end (DFE), HS only	RW	0x0
15:10	RESERVED		R	0x00
9	P2_CONNECT_STATUS	Connection status for port 2 Note that reset value does not correspond, on purpose, to the default connection status.  Read 0x0: Disconnected Read 0x1: Peripheral connected and active on port	R	1
8	P1_CONNECT_STATUS	Connection status for port 1 Note that reset value does not correspond, on purpose, to the default connection status.  Read 0x0: Disconnected Read 0x1: Peripheral connected and active on port	R	1
7:6	RESERVED		R	0x0
5	ENA_INCR_ALIGN	Force alignment of bursts to the respective burst-size boundaries <sup>(1)</sup>  0x0: Disable burst type 0x1: Enable burst type	RW	0
4	ENA_INCR16	Control the use of INCR16-type bursts (in AHB sense)  0x0: Disable burst type 0x1: Enable burst type	RW	1
3	ENA_INCR8	Control the use of INCR8-type bursts (in AHB sense)  0x0: Disable burst type 0x1: Enable burst type	RW	1
2	ENA_INCR4	Control the use of INCR4-type bursts (in AHB sense)  0x0: Disable burst type 0x1: Enable burst type	RW	1
1	AUTOPPD_ON_OVERCUR_EN	Configure reaction upon port overcurrent condition. This function is not supported at the device level.  0x0: Port remains on upon overcurrent. 0x1: Port is powered down automatically upon overcurrent.	RW	0X0
0	RESERVED		R	0

<sup>(1)</sup> This bit must be set to 1 to avoid buffer underflow.

**Table 23-1063. Register Call Summary for Register UHH\_HOSTCONFIG**

High-Speed Multiport USB Host Subsystem

- [HS USB Host Controller Port Status: \[0\] \[1\] \[2\] \[3\]](#)
- [HS USB Host Controller Burst Control: \[4\] \[5\]](#)
- [UTMI Interface: \[6\] \[7\] \[8\]](#)
- [ULPI Interface: \[9\] \[10\]](#)
- [Operational Modes Configuration \(Selecting and Configuring USB Connectivity\): \[11\]](#)
- [HSUSBHOST Controller Register Summary: \[12\]](#)

**Table 23-1064. UHH\_DEBUG\_CSR**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	HSUSBHOST
<b>Physical Address</b>	See <a href="#">Table 23-1053</a> .		
<b>Description</b>	Debug control and status for the EHCI, OHCI hosts.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
RESERVED														RESERVED														EHCI_FLADJ							
														OHCI_CCS_2			OHCI_CCS_1			OHCI_GLOBALSUSPEND								OCHI_CNTSEL		EHCI_SIMULATION_MODE					

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Reserved	R	0x000
18	OHCI_CCS_2	Current Connect Status of port 2 Read 0x0: No periph connected Read 0x1: Periph connected	R	0
17	OHCI_CCS_1	Current Connect Status of port 1 Read 0x0: No periph connected Read 0x1: Periph connected	R	0
16	OHCI_GLOBALSUSPEND	OHCI global suspend status, asserted 5 ms after the suspend order. Read 0x0: Host is not suspended Read 0x1: Host is suspended	R	0
15:8	RESERVED		R	0x00
7	OCHI_CNTSEL	Selection of a shorter "1 ms" counter in OHCI host, to speed up long USB phases like reset, resume, etc... (Used only for simulation.) 0x0: Functional mode, 1ms = 12,000 x 12 MHz cycles 0x1: Simulation mode, 1ms = 7 x 12 MHz cycles = 583 ns	RW	0
6	EHCI_SIMULATION_MODE	Sets the PHY to non-driving mode. (Used only for simulation.) 0x0: Functional mode 0x1: PHY set to non-driving	RW	0
5:0	EHCI_FLADJ	EHCI host frame length adjust. Modify only when EHCI Bit Field <a href="#">USBSTS.HCHalted</a> = 1 Field value + 59,488 = 60,000 by default = Number of 60 MHz UTMI/ULPI clock cycles per 1 ms USB frame = Number of 480 MHz HS bits per 125 us HS USB microframe	RW	0x20

**Table 23-1065. Register Call Summary for Register UHH\_DEBUG\_CSR**

High-Speed Multiport USB Host Subsystem

- [HSUSBHOST Controller Register Summary: \[0\]](#)

**Table 23-1066. UHH\_SAR\_CNTX\_i**

<b>Address Offset</b>	0x0000 0100 + (0x4 * i)	<b>Index</b>	i = 0 to 383
<b>Physical Address</b>	See <a href="#">Table 23-1053</a> .	<b>Instance</b>	HSUSBHOST
<b>Description</b>	Save and Restore context array. Array size is indicated in <a href="#">UHH_HWINFO</a> . When in SAR mode, read out to save and write to restore. Do not access when not in SAR mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTX																															

Bits	Field Name	Description	Type	Reset
31:0	CNTX	Context bits	RW	0x0000 0000

**Table 23-1067. Register Call Summary for Register UHH\_SAR\_CNTX\_i**

High-Speed Multiport USB Host Subsystem

- [HSUSBHOST Controller Register Summary: \[0\]](#)

### 23.11.6.5 OHCI Registers

#### 23.11.6.5.1 OHCI Register Summary

[Table 23-1068](#) summarizes the OHCI register mapping.

**Table 23-1068. OHCI Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
<a href="#">HCREVISION</a>	R	32	0x0000 0000	0x4A06 4800
<a href="#">HCCONTROL</a>	RW	32	0x0000 0004	0x4A06 4804
<a href="#">HCCOMMANDSTATUS</a>	RW	32	0x0000 0008	0x4A06 4808
<a href="#">HCINTERRUPTSTATUS</a>	RW	32	0x0000 000C	0x4A06 480C
<a href="#">HCINTERRUPTENABLE</a>	RW	32	0x0000 0010	0x4A06 4810
<a href="#">HCINTERRUPTDISABLE</a>	RW	32	0x0000 0014	0x4A06 4814
<a href="#">HCHCCA</a>	RW	32	0x0000 0018	0x4A06 4818
<a href="#">HCPERIODCURRENTED</a>	R	32	0x0000 001C	0x4A06 481C
<a href="#">HCCONTROLHEADED</a>	RW	32	0x0000 0020	0x4A06 4820
<a href="#">HCCONTROLCURRENTED</a>	RW	32	0x0000 0024	0x4A06 4824
<a href="#">HCBULKHEADED</a>	RW	32	0x0000 0028	0x4A06 4828
<a href="#">HCBULKCURRENTED</a>	RW	32	0x0000 002C	0x4A06 482C
<a href="#">HCDONEHEAD</a>	R	32	0x0000 0030	0x4A06 4830
<a href="#">HCFMINTERVAL</a>	RW	32	0x0000 0034	0x4A06 4834
<a href="#">HCFMREMAINING</a>	R	32	0x0000 0038	0x4A06 4838
<a href="#">HCFMNUMBER</a>	R	32	0x0000 003C	0x4A06 483C
<a href="#">HCPERIODICSTART</a>	RW	32	0x0000 0040	0x4A06 4840
<a href="#">HCLSTHRESHOLD</a>	RW	32	0x0000 0044	0x4A06 4844
<a href="#">HCRHDESCRIPTORA</a>	RW	32	0x0000 0048	0x4A06 4848
<a href="#">HCRHDESCRIPTORB</a>	RW	32	0x0000 004C	0x4A06 484C
<a href="#">HCRHSTATUS</a>	RW	32	0x0000 0050	0x4A06 4850
<a href="#">HCRHPORTSTATUS_1</a>	RW	32	0x0000 0054	0x4A06 4854
<a href="#">HCRHPORTSTATUS_2</a>	RW	32	0x0000 0058	0x4A06 4858
RESERVED	RW	32	0x0000 005C	0x4A06 485C

**CAUTION**

The high-speed USB registers are limited to 32-bit data accesses; 16-bit and 8-bit accesses are not allowed and can corrupt register content.

OHCI register descriptions conform to the OHCI USB standard: Open Host Controller Interface Specification for USB, Release 1.0a. For more information about these registers or for new specification releases, search OHCI on [www.usb.org](http://www.usb.org).

**23.11.6.5.2 OHCI Register Description**

through describe the OHCI registers.

**Table 23-1069. HCREVISION**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	OHCI
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .		
<b>Description</b>	OHCI revision number		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REV															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	REV	OHCI specification revision the OHCI revision number upon which the USB host controller is based. Examples: 0x10 for 1.0 0x21 for 2.1	R	0x10

**Table 23-1070. Register Call Summary for Register HCREVISION**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1071. HCCONTROL**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	OHCI
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .		
<b>Description</b>	HC Operating Mode Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RWE	RWC	IR	HCFS	BLE	CLE	IE	PLE	CBSR	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x000000
10	RWE	Remote wake-up enable This bit is used to enable or disable the remote wakeup feature upon detection of upstream resume signaling.	RW	0
9	RWC	Remote wake-up connected. This bit indicates whether the host controller supports remote wakeup signaling.	RW	0
8	IR	Interrupt routing.	RW	0

Bits	Field Name	Description	Type	Reset
		This bit determines the routing of interrupts generated by events registered in <a href="#">HCINTERRUPTSTATUS</a> . 0x0: All interrupts are routed to the normal host bus interrupt mechanism. 0x1: interrupts are routed to the system management Interrupt.		
7:6	HCFS	Host Controller Functional State 0x0: HCFS: USB Reset 0x1: HCFS: USB Resume 0x2: HCFS: USB Operational 0x3: HCFS: USB Suspend	RW	0x0
5	BLE	Bulk list processing enable. 0x0: Bulk ED list is not processed after the next SOF. 0x1: Enables processing of bulk ED list in the next frame.	RW	0
4	CLE	Control list processing enable. 0x0: Control ED list is not processed after the next SOF. 0x1: Enables processing of control ED list in the next frame.	RW	0
3	IE	Isochronous ED processing enabled by Host Controller Driver. 0x0: Isochronous EDs are not processed 0x1: Enables processing of isochronous EDs.	RW	0
2	PLE	Periodic list enable 0x0: Periodic ED lists are not processed after the next frame. 0x1: Enables processing of periodic ED lists in the next frame.	RW	0
1:0	CBSR	Control/bulk service ratio. Specifies the ratio between control and bulk EDs processed in a frame. 0x0: One control ED per bulk ED 0x1: Two control ED per bulk ED 0x2: Three control ED per bulk ED 0x3: Four control ED per bulk ED	RW	0x0

**Table 23-1072. Register Call Summary for Register HCCONTROL**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1073. HCCOMMANDSTATUS**

<b>Address Offset</b>	0x0000 0008																														
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .																<b>Instance</b> OHCI														
<b>Description</b>	HC Command and Status																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SOC		RESERVED								OCR	BLF	CLF	HCR		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:16	SOC	Scheduling overrun count. This is used to monitor any persistent scheduling problems. These bits are incremented on each scheduling overrun error. It is initialized to 0x0 and wraps around at 0x3.	R	0x0
15:4	RESERVED	Reserved	R	0x000
3	OCR	Ownership change request. This bit is set to request a change of control of the host controller.	RW	0
2	BLF	Bulk list filled. This bit is used to indicate whether there are any TDs on the bulk list. It is set whenever it adds a TD to an ED in the bulk list.	RW	0
1	CLF	Control list filled. This bit is used to indicate whether there are any TDs on the control list. It is set whenever it adds a TD to an ED in the control list.	RW	0
0	HCR	Host controller reset. (software reset) Set this bit to initiate a USB host controller reset. This resets most USB host controller OHCI registers. OHCI register accesses must not be attempted until a read of this register returns a 0. 0x0: No effect 0x1: USB host controller is reset.	RW	0

**Table 23-1074. Register Call Summary for Register HCCOMMANDSTATUS**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)
- [OHCI Register Description: \[1\]](#)

**Table 23-1075. HCINTERRUPTSTATUS**

<b>Address Offset</b>	0x0000 000C	<b>Instance</b>	OHCI
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .		
<b>Description</b>	HC Interrupt Status		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	OC	RESERVED														RHSC	FNO	UE	RD	SF	WDH	SO									

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	OC	Ownership change. This bit is set when the <a href="#">HCCOMMANDSTATUS[3]</a> OCR bit is set. Read 0x1: An ownership change has occurred. Write 0x0: No effect Write 0x1: Clears this bit	R	0
29:7	RESERVED	Reserved	R	0x000000

Bits	Field Name	Description	Type	Reset
6	RHSC	Root hub status change. When 0x1: A root hub status change has occurred. Write 0x0: No effect. Write 0x1: Clears this bit.	RW	0
5	FNO	Frame number overflow. When 0x1: A frame number overflow has occurred. Write 0x0: No effect. Write 0x1: Clears this bit.	RW	0
4	UE	Unrecoverable error. When 0x1: An unrecoverable error has occurred. Write 0x0: No effect. Write 0x1: Clears this bit.	RW	0
3	RD	Resume detected. When 0x1: A downstream device has issued a resume request. Write 0x0: No effect. Write 0x1: Clears this bit.	RW	0
2	SF	Start of frame. When 0x1: A SOF has been issued. Write 0x0: No effect. Write 0x1: Clears this bit.	RW	0
1	WDH	Write done head. When 0x1: The USB host controller has updated the <a href="#">HCDONEHEAD</a> register. Write 0x0: No effect. Write 0x1: Clears this bit.	RW	0
0	SO	Scheduling overrun. When 0x1: A scheduling overrun has occurred. Write 0x0: No effect. Write 0x1: Clears this bit.	RW	0

**Table 23-1076. Register Call Summary for Register HCINTERRUPTSTATUS**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)
- [OHCI Register Description: \[1\]](#)

**Table 23-1077. HCINTERRUPTENABLE**

<b>Address Offset</b>	0x0000 0010																																																								
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI																																																							
<b>Description</b>	HC Interrupt Enable																																																								
<b>Type</b>	RW																																																								
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>MIE</td><td>OC</td><td colspan="14">RESERVED</td><td>RHSC</td><td>FNO</td><td>UE</td><td>RD</td><td>SF</td><td>WDH</td><td>SO</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	MIE	OC	RESERVED														RHSC	FNO	UE	RD	SF	WDH	SO
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																										
MIE	OC	RESERVED														RHSC	FNO	UE	RD	SF	WDH	SO																																			
Bits	Field Name	Description	Type	Reset																																																					
31	MIE	Master interrupt enable. When 0x1: Allows other enabled OHCI interrupt sources to propagate to the device interrupt controller. When 0x0: OHCI interrupt sources are ignored. Write 0x0: No effect. Write 0x1: Sets this bit.	RW	0																																																					
30	OC	Ownership change. Write 0x0: No effect. Write 0x1: Enable interrupt generation due to ownership change.	RW	0																																																					



Bits	Field Name	Description	Type	Reset
29:7	RESERVED	Reserved	R	0x000000
6	RHSC	Root hub status change. When 0x1 and MIE is 0x1: Allows root hub status change interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: Root hub status change interrupts do not propagate. Write 0x0: No effect. Write 0x1: Sets this bit.	RW	0
5	FNO	Frame number overflow. When 0x1 and MIE is 0x1: Allows FNO interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: FNO interrupts do not propagate. Write 0x0: No effect. Write 0x1: Sets this bit.	RW	0
4	UE	Unrecoverable error. When 0x1 and MIE is 0x1: Allows UE interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: UE interrupts do not propagate. Write 0x0: No effect. Write 0x1: Sets this bit.	RW	0
3	RD	Resume detected. When 0x1 and MIE is 0x1: Allows RD interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: RD interrupts do not propagate. Write 0x0: No effect. Write 0x1: Sets this bit.	RW	0
2	SF	Start of frame. When 0x1 and MIE is 0x1: Allows SF interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: SF interrupts do not propagate. Write 0x0: No effect. Write 0x1: Sets this bit.	RW	0
1	WDH	Write done head. When 0x1 and MIE is 0x1: Allows WDH interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: WDH interrupts do not propagate. Write 0x0: No effect. Write 0x1: Sets this bit.	RW	0
0	SO	Scheduling overrun. When 0x1 and MIE is 0x1: Allows SO interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: SO interrupts do not propagate. Write 0x0: No effect. Write 0x1: Sets this bit.	RW	0

**Table 23-1078. Register Call Summary for Register HCINTERRUPTENABLE**

High-Speed Multiport USB Host Subsystem

- [HS Multiport USB Host Subsystem Global Initialization: \[0\]](#)
- [OHCI Register Summary: \[1\]](#)
- [OHCI Register Description: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)

**Table 23-1079. HCINTERRUPTDISABLE**

<b>Address Offset</b>	0x0000 0014		
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b>	OHCI
<b>Description</b>	HC Interrupt Disable		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIE	OC	RESERVED														RHSC	FNO	UE	RD	SF	WDH	SO									

Bits	Field Name	Description	Type	Reset
31	MIE	Master interrupt enable. Always reads 0x0. Write 0x0: No effect. Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> MIE bit.	RW	0
30	OC	Ownership change. Write 0x0: No effect. Write 0x1: Disable interrupt generation due to ownership change.	RW	0
29:7	RESERVED	Reserved	R	0x000000
6	RHSC	Root hub status change. Always reads 0x0. Write 0x0: No effect. Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> RHSC bit.	RW	0
5	FNO	Frame number overflow. Always reads 0x0. Write 0x0: No effect. Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> FNO bit.	RW	0
4	UE	Unrecoverable error. Always reads 0x0. Write 0x0: No effect. Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> UE bit.	RW	0
3	RD	Resume detected. Always reads 0x0. Write 0x0: No effect. Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> RD bit.	RW	0
2	SF	Start of frame. Always reads 0x0. Write 0x0: No effect. Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> SF bit.	RW	0
1	WDH	Write done head. Always reads 0x0. Write 0x0: No effect. Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> WDH bit.	RW	0
0	SO	Scheduling overrun. Always reads 0x0. Write 0x0: No effect. Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> SO bit.	RW	0

**Table 23-1080. Register Call Summary for Register HCINTERRUPTDISABLE**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1081. HCHCCA**

<b>Address Offset</b>	0x0000 0018	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI
<b>Description</b>	HC HCCA Address Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCCA																RESERVED															

Bits	Field Name	Description	Type	Reset
31:8	HCCA	Physical address of the beginning of the HCCA.	RW	0x000000
7:0	RESERVED	Reserved	R	0x00

**Table 23-1082. Register Call Summary for Register HCHCCA**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1083. HCPERIODCURRENTED**

<b>Address Offset</b>	0x0000 001C	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI
<b>Description</b>	HC Current Periodic Register	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	PCED	Physical address of current ED on the periodic ED list.	R	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1084. Register Call Summary for Register HCPERIODCURRENTED**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1085. HCCONTROLHEADED**

<b>Address Offset</b>	0x0000 0020	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a>	<b>Instance</b> OHCI
<b>Description</b>	HC Head Control Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	CHED	Physical address of head ED on the control ED list.	RW	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1086. Register Call Summary for Register HCCONTROLHEADED**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1087. HCCONTROLCURRENTED**

<b>Address Offset</b>	0x0000 0024	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI
<b>Description</b>	HC Current Control Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	CCED	Physical address of current ED on the control ED list.	RW	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1088. Register Call Summary for Register HCCONTROLCURRENTED**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1089. HCBULKHEADED**

<b>Address Offset</b>	0x0000 0028	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI
<b>Description</b>	HC Head Bulk Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BHED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	BHED	Physical address of head ED on the bulk ED list.	RW	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1090. Register Call Summary for Register HCBULKHEADED**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1091. HCBULKCURRENTED**

<b>Address Offset</b>	0x0000 002C	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI
<b>Description</b>	HC Current Bulk Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	BCED	Physical address of current ED on the bulk ED list.	RW	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1092. Register Call Summary for Register HCBULKCURRENTED**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1093. HCDONEHEAD**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	OHCI
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .		
<b>Description</b>	HC Head Done Register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DH																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	DH	Physical address of last TD that was added to the Done queue.	R	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1094. Register Call Summary for Register HCDONEHEAD**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)
- [OHCI Register Description: \[1\]](#)

**Table 23-1095. HCFMINTERVAL**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	OHCI
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .		
<b>Description</b>	HC Frame Interval Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIT	FSMPS																RESERVED	FI													

Bits	Field Name	Description	Type	Reset
31	FIT	Frame interval toggle. This bit is toggled whenever it loads a new value to FI.	RW	0
30:16	FSMPS	Largest data packet size for full-speed packets, bit times. This field specifies a value which is loaded into the largest data packet counter at the beginning of each frame.	RW	0x0000
15:14	RESERVED	Reserved	R	0x0
13:0	FI	Frame interval. Number of 12-MHz clocks in the USB frame. The nominal value is set to 11,999, to give a 1-ms frame.	RW	0x2EDF

**Table 23-1096. Register Call Summary for Register HCFMINTERVAL**

High-Speed Multiport USB Host Subsystem

- [OHCI Implementation Specifications: \[0\]](#)
- [OHCI Register Summary: \[1\]](#)
- [OHCI Register Description: \[2\] \[3\] \[4\]](#)

**Table 23-1097. HCFMREMAINING**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	OHCI
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .		
<b>Description</b>	HC Frame Remaining Register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FR															

Bits	Field Name	Description	Type	Reset
31	FRT	Frame remaining toggle. This bit is used for the synchronization between <a href="#">HCFMINTERVAL[13:0]</a> FI and FR. This bit is loaded from the <a href="#">HCFMINTERVAL[31]</a> FIT field whenever FR reaches 0.	R	0
30:14	RESERVED	Reserved	R	0x00000
13:0	FR	Frame remaining. This counter is decremented at each bit time. When it reaches 0, it is reset by loading the <a href="#">USBHOST.HCFMINTERVAL[13:0]</a> FI value at the next bit time boundary.	R	0x0000

**Table 23-1098. Register Call Summary for Register HCFMREMAINING**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)
- [OHCI Register Description: \[1\]](#)

**Table 23-1099. HCFMNUMBER**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	OHCI
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .		
<b>Description</b>	HC Frame Number Register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FN															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	FN	Frame Number. This is incremented when <a href="#">HCFMREMAINING</a> is reloaded. It is rolled over to 0x0000 after 0xFFFF.	R	0x0000

**Table 23-1100. Register Call Summary for Register HCFMNUMBER**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1101. HCPERIODICSTART**

<b>Address Offset</b>	0x0000 0040	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI
<b>Description</b>	HC Periodic Start Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PS															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x00000
13:0	PS	Periodic start. The host controller driver must program this value to be about 10% less than the frame interval field value so that control and bulk EDs have priority for the first 10% of the frame; then periodic EDs have priority for the remaining 90% of the frame.	RW	0x0000

**Table 23-1102. Register Call Summary for Register HCPERIODICSTART**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1103. HCLSTHRESHOLD**

<b>Address Offset</b>	0x0000 0044	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI
<b>Description</b>	HC Low-Speed Threshold Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LST															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11:0	LST	Low-speed threshold	RW	0x628

**Table 23-1104. Register Call Summary for Register HCLSTHRESHOLD**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1105. HCRHDESCRIPTORA**

<b>Address Offset</b>	0x0000 0048		
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b>	OHCI
<b>Description</b>	HC Root Hub A Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POTPG								RESERVED								NOCP	OCPM	DT	NPS	PSM	NDP										

Bits	Field Name	Description	Type	Reset
31:24	POTPG	Power-on to power-good time. Defines the minimum amount of time (2 ms * POTPG) between the USB host controller turning on power to a downstream port and when the USB host can access the downstream device.	RW	0x0A
23:13	RESERVED	Reserved	R	0x0
12	NOCP	No overcurrent protection. This function is not supported at the device level.  0x0: Overcurrent status is reported collectively for all downstream ports.  0x1: The USB host controller does not implement overcurrent protection inputs.	RW	0x0
11	OCPM	Overcurrent protection mode. This function is not supported at the device level.	RW	0x1
10	DT	Device type. Always reads 0x0: Indicates that the USB host controller implemented is not a compound device.	R	0
9	NPS	No power switching  0x0: VBUS power switching is supported, either per-port or all-port switched per the power  0x1: VBUS power switching is not supported, power is available to all downstream ports.	RW	0
8	PSM	Power switching mode.  0x0: Indicates that all ports are powered at the same time  0x1: Individual port power switching is supported	RW	1
7:0	NDP	Number of downstream ports. These bits specify the number of downstream ports supported by the root hub. It is implementation-specific. The minimum number of ports is 1. The maximum number of ports supported by OHCI is 15.	R	0x03

**Table 23-1106. Register Call Summary for Register HCRHDESCRIPTORA**

High-Speed Multiport USB Host Subsystem

- [OHCI Implementation Specifications: \[0\] \[1\] \[2\] \[3\]](#)
- [OHCI Register Summary: \[4\]](#)



**Table 23-1107. HCRHDESCRIPTORB**

<b>Address Offset</b>	0x0000 004C	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI
<b>Description</b>	HC Root Hub B Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPCM																DR															

Bits	Field Name	Description	Type	Reset
31:16	PPCM	Port power control mask. Each bit defines whether a corresponding downstream port has port power controlled by the global power control. When set the port's power state is only affected by per-port power control. When cleared the port is controlled by the global power switch. If the device is configured to global switch mode this field is not valid. bit 0: reserved, bit 1: Ganged-power mask on port 1, ..., bit 15: Ganged-power mask on port 15	RW	0x0000
15:0	DR	Device removable. Each bit defines whether a corresponding downstream port has a removable device. When cleared the attached device is removable. When set the attached device is not removable. Bit 0: reserved, bit 1 : Device attached to port 1, &, bit 15: Device attached to port 15	RW	0x0000

**Table 23-1108. Register Call Summary for Register HCRHDESCRIPTORB**

- High-Speed Multiport USB Host Subsystem
- [OHCI Implementation Specifications: \[0\] \[1\]](#)
  - [OHCI Register Summary: \[2\]](#)
  - [OHCI Register Description: \[3\] \[4\] \[5\] \[6\]](#)

**Table 23-1109. HCRHSTATUS**

<b>Address Offset</b>	0x0000 0050	
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .	<b>Instance</b> OHCI
<b>Description</b>	HC Root Hub Status Register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRWE	RESERVED										LPSC	DRWE	RESERVED										LPS								

Bits	Field Name	Description	Type	Reset
31	CRWE	Clear remote wake-up enable. Write 0x0: No effect Write 0x1: Clears the device remote wake-up enable bit	W	0
30:17	RESERVED	Reserved	R	0x0000
16	LPSC	Local power status change. Always reads 0x0: The Root Hub does not support the local power status feature Write 0x0: No effect Write 0x1: Sets port power status bits for all ports, if power switching mode is 0. Sets port power status bits for ports with their corresponding port power control mask bits cleared if power switching mode is 1.	RW	0

Bits	Field Name	Description	Type	Reset
15	DRWE	Device remote wake-up enable. Enables a connect status change event as a resume event, causing a USB suspend to USB resume state transition and sets the resume detected interrupt status bit. Read 0x1: Connect status change is a remote wake-up event. Read 0x0: Connect status change is not a remote wake-up event. Write 0x0: No effect Write 0x1: Sets the device remote wake-up enable bit	RW	0
14:1	RESERVED	Reserved	R	0x0000
0	LPS	Local power status. Always reads 0x0 Write 0x0: No effect Write 0x1: When in global power mode (power switching mode = 0), turns off power to all ports. If in per-port power mode (power switching mode = 1), turns of power to those ports whose corresponding port power control mask bit is 0.	RW	0

**Table 23-1110. Register Call Summary for Register HCRHSTATUS**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

**Table 23-1111. HCRHPORTSTATUS\_1**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	OHCI
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .		
<b>Description</b>	HC Port 1 Status and Control Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRSC	RESERVED	PSSC	PESC	CSC	RESERVED								LSDA_CPP	PPS_SPP	RESERVED	PRS_SPR	RESERVED	PSS_SPS	PES_SPE	CCS_CPE			

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved	R	0x000
20	PRSC	Port 1 reset status change. This bit is set when the Port 1 port reset status bit has changed. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
19	RESERVED		R	0
18	PSSC	Port 1 suspendsstatus change. Set when the port leaves the suspend state, i.e. after the full resume sequence has completed. Write 0x0: No effect Read 0x0: No resume completed (since either the last SW clear, or the last port reset) Read 0x1: Resume completed on the port Write 0x1: Clears the bit	RW	0
17	PESC	Port 1 enable status change. This bit is set when the Port 1 port enable status has changed. Write 0x0: No effect Write 0x1: Clears this bit	RW	0

Bits	Field Name	Description	Type	Reset
16	CSC	Port 1 connect status change. This bit is set when the Port1 port current connect status has changed due to a connect or disconnect event. If current connect status is 0 when a set port reset, set port enable, or set port suspend write occurs, this bit is set. Write 0x0: No effect Write 0x1: Clears this bit Note: If the DR bit <a href="#">HCRHDESCRIPTORB[1]</a> is set, this bit is set only after a root hub reset to inform the system that the device is attached.	RW	0
15:10	RESERVED	Reserved	R	0x00
9	LSDA_CPP	Port 1 low-speed device attached/clear port power. This bit is valid only when port 1 current connect status is 1. Read 0x0: A full-speed device is attached to port 1. Read 0x1: A low-speed device is attached to port 1. Write 0x0: No effect Write 0x1: Clears the port 1 port power status	RW	0
8	PPS_SPP	Port 1 port power status/set port power. Read 0x0: Port 1 power is enabled. Read 0x1: Port 1 power is not enabled. Write 0x0: No effect Write 0x1: Sets the port 1 port power status bit	RW	0
7:5	RESERVED	Reserved	R	0x0
4	PRS_SPR	Port 1 port reset status/set port reset. Read 0x0: USB reset is not being sent to port 1. Read 0x1: Port 1 is signaling the USB reset. Write 0x0: No effect Write 0x1: Sets the port 1 port reset status bit and causes the USB host controller to begin signaling USB reset to port 1	RW	0
3	RESERVED		R	0
2	PSS_SPS	Port 1 port suspend status/set port suspend. This bit is cleared automatically at the end of the USB resume sequence and also at the end of the USB reset sequence. Write 0x0: No effect Read 0x0: Port 1 is not in the USB suspend state. Read 0x1: Port 1 is in the USB suspend state or is in the resume sequence. Write 0x1: If port 1 current connect status is 1, sets the port 1 port suspend status bit and places port 1 in USB suspend state. If current connect status is 0, sets instead connect status change to inform the USB host controller driver of an attempt to suspend a disconnected port.	RW	0
1	PES_SPE	Port 1 port enable status/set port enable. This bit is automatically set at completion of port 1 USB reset if it was not already set before the USB reset completed, and is automatically set at the end of a USB suspend if the port was not enabled when the USB resume completed. Read 0x0: Port 1 is not enabled. Read 0x1: Port 1 is enabled. Write 0x0: No effect Write 0x1: When port 1 current connect status is 1 sets the port 1 port enable status bit. When port 1 current status is 0 has no effect.	RW	0
0	CCS_CPE	Port 1 current connection status/clear port enable. Read 0x0: No USB device is attached to port 1. Read 0x1: Port 1 currently has a USB device attached. Write 0x0: No effect Write 0x1: Clears the port 1 port enable bit Note: This bit is set to 1 if the DR bit <a href="#">HCRHDESCRIPTORB[1]</a> is set to indicate a nonremovable device on port 1.	RW	0

**Table 23-1112. Register Call Summary for Register HCRHPORTSTATUS\_1**

High-Speed Multiport USB Host Subsystem

- [HS USB Host Controller Port Status: \[0\]](#)
- [OHCI Register Summary: \[1\]](#)

**Table 23-1113. HCRHPORTSTATUS\_2**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	OHCI
<b>Physical Address</b>	See <a href="#">Table 23-1068</a> .		
<b>Description</b>	HC Port 2 Status and Control Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRSC	RESERVED	PSSC	PESC	CSC	RESERVED								LSDA_CPP	PPS_SPP	RESERVED	PRS_SPR	RESERVED	PSS_SPS	PES_SPE	CCS_CPE			

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved	R	0x000
20	PRSC	Port 2 reset status change. This bit is set when the Port 2 port reset status bit has changed. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
19	RESERVED		RW	0
18	PSSC	Port 2 suspend status change. Set when the port leaves the suspend state, i.e. after the full resume sequence has completed. Write 0x0: No effect Read 0x0: No resume completed (since either the last SW clear, or the last port reset) Read 0x1: Resume completed on the port Write 0x1: Clears the bit	RW	0
17	PESC	Port 2 enable status change. This bit is set when the Port 2 port enable status has changed. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
16	CSC	Port 2 connect status change. This bit is set when the Port 2 port current connect status has changed due to a connect or disconnect event. If current connect status is 0 when a set port reset, set port enable, or set port suspend write occurs, this bit is set. Write 0x0: No effect Write 0x1: Clears this bit Note: If the DR bit <a href="#">HCRHDESCRIPTORB[1]</a> is set, this bit is set only after a root hub reset to inform the system that the device is attached.	RW	0
15:10	RESERVED	Reserved	R	0x00
9	LSDA_CPP	Port 2 low-speed device attached/clear port power. This bit is valid only when port 2 current connect status is 1. Read 0x0: A full-speed device is attached to port 2. Read 0x1: A low-speed device is attached to port 2. Write 0x0: No effect Write 0x1: Clears the port 2 port power status	RW	0

Bits	Field Name	Description	Type	Reset
8	PPS_SPP	Port 2 port power status/set port power. Read 0x0: Port 2 power is enabled. Read 0x1: Port 2 power is not enabled. Write 0x0: No effect Write 0x1: Sets the port 2 port power status bit	RW	0
7:5	RESERVED	Reserved	R	0x0
4	PRS_SPR	Port 2 port reset status/set port reset. Read 0x0: USB reset is not being sent to port 2. Read 0x1: Port 2 is signaling the USB reset. Write 0x0: No effect Write 0x1: Sets the port 2 port reset status bit and causes the USB host controller to begin signaling USB reset to port 2	RW	0
3	RESERVED		RW	0
2	PSS_SPS	Port 2 port suspend status/set port suspend. This bit is cleared automatically at the end of the USB resume sequence and also at the end of the USB reset sequence. Write 0x0: No effect Read 0x0: Port 2 is not in the USB suspend state. Read 0x1: Port 2 is in the USB suspend state or is in the resume sequence. Write 0x1: If port 2 current connect status is 1, sets the port 2 port suspend status bit and places port 2 in USB suspend state. If current connect status is 0, sets instead connect status change to inform the USB host controller driver of an attempt to suspend a disconnected port.	RW	0
1	PES_SPE	Port 2 port enable status/set port enable. This bit is automatically set at completion of port 2 USB reset if it was not already set before the USB reset completed, and is automatically set at the end of a USB suspend if the port was not enabled when the USB resume completed. Read 0x0: Port 2 is not enabled. Read 0x1: Port 2 is enabled. Write 0x0: No effect Write 0x1: When port 2 current connect status is 1 sets the port 2 port enable status bit. When port 2 current status is 0 has no effect.	RW	0
0	CCS_CPE	Port 2 current connection status/clear port enable. Read 0x0: No USB device is attached to port 2. Read 0x1: Port 2 currently has a USB device attached. Write 0x0: No effect Write 0x1: Clears the port 2 port enable bit Note: This bit is set to 1 if the DR bit <a href="#">HCRHDESCRIPTORB[1]</a> is set to indicate a nonremovable device on port 2.	RW	0

**Table 23-1114. Register Call Summary for Register HCRHPORTSTATUS\_2**

High-Speed Multiport USB Host Subsystem

- [OHCI Register Summary: \[0\]](#)

### 23.11.6.6 EHCI Registers

#### 23.11.6.6.1 EHCI Register Summary

[Table 23-1115](#) summarizes the EHCI register mapping.

**Table 23-1115. EHCI Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
<a href="#">HCCAPBASE</a>	R	32	0x0000 0000	0x4A06 4C00
<a href="#">HCSPARAMS</a>	R	32	0x0000 0004	0x4A06 4C04
<a href="#">HCCPARAMS</a>	R	32	0x0000 0008	0x4A06 4C08

**Table 23-1115. EHCI Register Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
USBCMD	RW	32	0x0000 0010	0x4A06 4C10
USBSTS	RW	32	0x0000 0014	0x4A06 4C14
USBINTR	RW	32	0x0000 0018	0x4A06 4C18
FRINDEX	RW	32	0x0000 001C	0x4A06 4C1C
CTRLDSSEGMENT	R	32	0x0000 0020	0x4A06 4C20
PERIODICLISTBASE	RW	32	0x0000 0024	0x4A06 4C24
ASYNCLISTADDR	RW	32	0x0000 0028	0x4A06 4C28
CONFIGFLAG	RW	32	0x0000 0050	0x4A06 4C50
PORTSC <sub>i</sub> <sup>(1)</sup>	RW	32	0x0000 0054 + (0x4 * i)	0x4A06 4C54 + (0x4 * i)
INSNREG00	RW	32	0x0000 0090	0x4A06 4C90
INSNREG01	RW	32	0x0000 0094	0x4A06 4C94
INSNREG02	RW	32	0x0000 0098	0x4A06 4C98
INSNREG03	RW	32	0x0000 009C	0x4A06 4C9C
INSNREG04	RW	32	0x0000 00A0	0x4A06 4CA0
INSNREG05_UTMI	RW	32	0x0000 00A4	0x4A06 4CA4
INSNREG05_ULPI	RW	32	0x0000 00A4	0x4A06 4CA4
INSNREG06	RW	32	0x0000 00A8	0x4A06 4CA8
INSNREG07	R	32	0x0000 00AC	0x4A06 4CAC
INSNREG08	RW	32	0x0000 00B0	0x4A06 4CB0

<sup>(1)</sup> i = 0 to 2**CAUTION**

The high-speed USB registers are limited to 32-bit data accesses; 16-bit and 8-bit accesses are not allowed and can corrupt register content.

EHCI register descriptions conform to the EHCI USB standard: Enhanced Host controller Interface (EHCI) specification for USB, Release 1.0. For more information about these registers or for new specification releases, search EHCI on [www.usb.org](http://www.usb.org).

**23.11.6.6.2 EHCI Register Description**

through describe the EHCI registers.

**Table 23-1116. HCCAPBASE**

Address Offset	0x0000 0000	Instance	EHCI
Physical Address	See <a href="#">Table 23-1115</a> .		
Description	Host Controller Capability register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCVERSION								RESERVED								CAPLENGTH															

Bits	Field Name	Description	Type	Reset
31:16	HCVERSION	Interface version number It contains a BCD encoding of the EHCI revision number supported by this host controller.  [7:4] Major revision [3:0] Minor revision	R	0x0100

Bits	Field Name	Description	Type	Reset
15:8	RESERVED	Reserved	R	0x00
7:0	CAPLENGTH	Capability register length	R	0x10

**Table 23-1117. Register Call Summary for Register HCCAPBASE**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

**Table 23-1118. HCSPARAMS**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	Host Controller Structural Parameters		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED		P_INDICATOR		N_CC				N_PCC				PRR	RESERVED		PPC	N_PORTS							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x000
19:17	RESERVED	Reserved	R	0x0
16	P_INDICATOR	Port indicator support indication  This bit indicates whether the ports support port indicator control.  0x1: The port status and control registers include a read/write field for controlling the state of the port indicator.	R	0
15:12	N_CC	Number of Companion Controllers  This field indicates the number of companion controllers associated with this USB 2.0 host controller.  0x0: There are no companion host controllers. Port-ownership hand-off is not supported. Only high-speed devices are supported on the host controller root ports.  Others: there are companion USB 1.1 host controller(s). Port-ownership hand-off is supported. High-, full-, and low-speed devices are supported on the host controller root ports.	R	0x1
11:8	N_PCC	Number of Ports per Companion Controller  This field indicates the number of ports supported per companion host controller. It is used to indicate the port routing configuration to system software.  For example, if N_PORTS has a value of 6 and N_CC has a value of 2, then N_PCC can have a value of 3.  The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc.  The number in this field must be consistent with N_PORTS and N_CC.	R	0x3
7	PRR	Port Routing Rules	R	0

Bits	Field Name	Description	Type	Reset
		The first N_PCC ports are routed to the lowest-numbered function companion host controller, the next N_PCC ports are routed to the next lowest-function companion controller, and so on.		
6:5	RESERVED	Reserved	R	0x0
4	PPC	Port Power control  This field indicates whether the host controller implementation includes port power control. 0x0: The ports do not have port power switches. 0x1: The ports have port power switches.	R	1
3:0	N_PORTS	Number of downstream ports  This field specifies the number of physical downstream ports implemented on this host controller.	R	0x3

**Table 23-1119. Register Call Summary for Register HCSPARAMS**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description: \[1\] \[2\] \[3\]](#)

**Table 23-1120. HCCPARAMS**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	Host Controller Capability Parameters		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EECP				IST			RESERVED	ASPC	PFLF	BIT64AC					

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17	LPM	Link power management capability 0: Link power management not supported 1: Link power management supported	R	1
16	RESERVED	Reserved	R	0
15:8	EECP	EHCI Extended Capabilities Pointer  This field indicates the existence of a capabilities list. 0x0: No extended capabilities are implemented. Others: The offset in PCI configuration space of the first EHCI extended capability.	R	0x00
7:4	IST	Isochronous Scheduling Threshold  This field indicates where software can reliably update the isochronous schedule in relation to the current position of the executing host controller.  The host controller can hold 1 microframe of isochronous data structures before flushing the state.	R	0x1
3	RESERVED	Reserved	R	0
2	ASPC	Asynchronous Schedule Park Capability  0x1: The host controller supports the park feature for high-speed queue heads in the asynchronous schedule.	R	1



Bits	Field Name	Description	Type	Reset
1	PFLF	<p>The feature can be disabled or enabled and set to a specific level by using the USBHOST.USBCMD[11]ASPME and the USBHOST.USBCMD[9:8] ASPMC fields.</p> <p>Programmable Frame List Flag</p> <p>0x0: System software must use a frame list length of 1024 elements with this host controller.</p> <p>0x1: System software can specify and use a smaller frame list and configure the host controller via the USBHOST.USBCMD[3:2] FLS field. The frame list must always be aligned on a 4K-page boundary.</p>	R	1
0	BIT64AC	<p>64-bit addressing capability</p> <p>This field documents the addressing range capability of this implementation.</p> <p>0x0: Data structures using 32-bit address memory pointers</p> <p>0x1: Data structures using 64-bit address memory pointers</p>	R	0

**Table 23-1121. Register Call Summary for Register HCCPARAMS**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description: \[1\]](#)

**Table 23-1122. USBCMD**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	USB Command		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				HIRD				ITC				RESERVED				ASPME	RESERVED	ASPMC	LHCR	IAAD	ASE	PSE	FLS	HCR	RS						

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	R	0x0
27:24	HIRD	<p>Host-initiated resume duration.</p> <p>If LPM is enabled, this field is RW; otherwise, it is R.</p> <p>The minimum for K-state during resume from LPM:</p> <p>0x0: 50 μs</p> <p>Each increment adds 75 μs.</p>	RW	0x0
23:16	ITC	<p>Interrupt Threshold Control</p> <p>This field is used by the system software to select the maximum rate at which the host controller issues interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are undefined.</p> <p>0x00: Reserved</p> <p>0x01: 1 microframe</p> <p>0x02: 2 microframes</p> <p>0x04: 4 microframes</p>	RW	0x08

Bits	Field Name	Description	Type	Reset
		0x08: 8 microframes (default, equates to 1 ms) 0x10: 16 microframes (2 ms) 0x20: 32 microframes (4 ms) 0x40: 64 microframes (8 ms) Others: Undefined		
15:12	RESERVED	Reserved	R	0x0
11	ASPME	Asynchronous Schedule Park Mode Enable 0x0: Park mode is disabled. 0x1: Park mode is enabled.	RW	1
10	RESERVED	Reserved	R	0
9:8	ASPMC	Asynchronous Schedule Park Mode Count  It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the asynchronous schedule before continuing traversal of the asynchronous schedule.  Valid values are 0x1 to 0x3. The software must not write a 0 to this bit when Park Mode Enable is a 1 as this may result in undefined behavior.	RW	0x3
7	LHCR	Light Host Controller Reset  It allows the driver to reset the EHCI controller without affecting the state of the ports or the relationship to the companion host controllers.  Read 0x0: Light host controller reset is complete and it is safe for host software to reinitialize the host controller. Read 0x1: Light host controller reset is still ongoing.	RW	0
6	IAAD	Interrupt on Async Advance Doorbell  This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule.  Write 0x1: Ring the doorbell.  Software should not write a 1 to this bit when the asynchronous schedule is disabled. Doing so may yield undefined results.	RW	0
5	ASE	Asynchronous Schedule Enable  This bit controls whether the host controller skips processing the asynchronous schedule. 0x0: Do not process the asynchronous schedule 0x1: Use the USBHOST. <a href="#">ASYNCLISTADDR</a> register to access the asynchronous schedule.	RW	0
4	PSE	Periodic Schedule Enable  This bit controls whether the host controller skips processing the periodic schedule. 0x0: Do not process the periodic schedule 0x1: Use the USBHOST. <a href="#">PERIODICLISTBASE</a> register to access the periodic schedule.	RW	0
3:2	FLS	Frame List Size  This field specifies the size of the frame list. The size of the frame list controls which bits in the frame index register should be used for the frame list current index. 0x0: 1024 elements (4096 bytes) 0x1: 512 elements (2048 bytes) 0x2: 256 elements (1024 bytes), for resource-constrained environments 0x3: Reserved	RW	0

Bits	Field Name	Description	Type	Reset
1	HCR	Host Controller Reset  This control bit is used by software to reset the host controller. Write  0x1: Reset the host controller, the PCI configuration registers are not affected by this reset and all operational registers are set to their initial values.  This bit is set to 0 by the host controller when the reset process is complete.	W	0
0	RS	Run/stop  0x1: Run, the host controller proceeds with execution of the schedule. The host controller continues execution as long as this bit is set to 1.  0x0: Stop, the host controller completes the current and any actively pipelined transactions on the USB and then halts.	RW	0

**Table 23-1123. Register Call Summary for Register USBCMD**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description: \[1\] \[2\] \[3\] \[4\] \[5\]](#)

**Table 23-1124. USBSTS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	USB status		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ASS	PSS	REC	HCH	RESERVED				IAA	HSE	FLR	PCD	USBEI	USBI		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15	ASS	Asynchronous Schedule Status  The bit reports the current real status of the asynchronous schedule.  0x0: The status of the asynchronous schedule is disabled.  0x1: The status of the asynchronous schedule is enabled.	R	0
14	PSS	Periodic Schedule Status  The bit reports the current real status of the periodic schedule.  0x0: The status of the periodic schedule is disabled.  0x1: The status of the periodic schedule is enabled.	R	0
13	REC	Reclamation  It is used to detect an empty asynchronous schedule.	R	0
12	HCH	Host Controller Halted  This bit is a 0 whenever the USBHOST.USBCMD[0] RS bit is a 1. The host controller sets this bit to 1 after it has stopped executing as a result of the RS bit being set to 0, either by software or by the host controller hardware.	R	1
11:6	RESERVED	Reserved	R	0x00

Bits	Field Name	Description	Type	Reset
5	IAA	Interrupt on Async Advance  System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a 1 in the USBHOST.USBCMD[6] IAAD bit. This status bit indicates the assertion of that interrupt source.	RW	0
4	HSE	Host System Error  The host controller sets this bit to 1 when a serious error occurs during a host system access involving the host controller module.	RW	0
3	FLR	Frame List Rollover  The host controller sets this bit to 1 when the USBHOST.FRINDEX rolls over from its maximum value to 0. The exact value at which the rollover occurs depends on the frame list size.	RW	0
2	PCD	Port Change Detect  The host controller sets this bit to 1 when any port for which the USBHOST.PORTSC_i[13] PO bit is set to 0 has a change bit transition from a 0 to a 1 or a USBHOST.PORTSC_i[6] FPR bit transition from a 0 to a 1.  This bit is also set as a result of the USBHOST.PORTSC_i[1] CSC bit being set to 1 after system software has relinquished ownership of a connected port by writing a 1 to a USBHOST.PORTSC_i[13] PO bit.	RW	0
1	USBEI	USB Error Interrupt  The host controller sets this bit to 1 when completion of a USB transaction results in an error condition.	RW	0
0	USBI	USB Interrupt  The host controller sets this bit to 1 on completion of a USB transaction, which results in the retirement of a transfer descriptor that had its IOC bit set.  The host controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes).	RW	0

**Table 23-1125. Register Call Summary for Register USBSTS**

High-Speed Multiport USB Host Subsystem

- [HSUSB Controller Register Description: \[0\]](#)
- [EHCI Register Summary: \[1\]](#)
- [EHCI Register Description: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\]](#)

**Table 23-1126. USBINTR**

<b>Address Offset</b>	0x0000 0018																														
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .																														
<b>Description</b>	USB interrupt enable																														
<b>Type</b>	RW																														
RESERVED																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																IAAE	HSEE	FLRE	PCIE	USBEIE	USBIE										

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0000000
5	IAAE	Interrupt on Async Advance Enable 0x1: When the <a href="#">USBSTS[5]</a> IAA bit is 1, the host controller issues an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the <a href="#">USBSTS[5]</a> IAA bit.	RW	0
4	HSEE	Host System Error Enable 0x1: When the <a href="#">USBSTS[4]</a> HSE bit is 1, the host controller issues an interrupt. The interrupt is acknowledged by software clearing the <a href="#">USBSTS[4]</a> HSE bit.	RW	0
3	FLRE	Frame List Rollover Enable 0x1: When the <a href="#">USBSTS[3]</a> FLR bit is 1, the host controller issues an interrupt. The interrupt is acknowledged by software clearing the <a href="#">USBSTS[3]</a> FLR bit.	RW	0
2	PCIE	Port Change Interrupt Enable 0x1: When the <a href="#">USBSTS[2]</a> PCD bit is 1, the host controller issues an interrupt. The interrupt is acknowledged by software clearing the <a href="#">USBSTS[3]</a> FLR bit.	RW	0
1	USBEIE	USB Error Interrupt Enable 0x1: When the <a href="#">USBSTS[1]</a> USBEI bit is 1, the host controller issues an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the <a href="#">USBSTS[1]</a> USBEI bit.	RW	0
0	USBIE	USB Interrupt Enable 0x1: When the <a href="#">USBSTS[0]</a> USBI bit is 1, the host controller issues an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the <a href="#">USBSTS[0]</a> USBI bit.	RW	0

**Table 23-1127. Register Call Summary for Register USBINTR**

High-Speed Multiport USB Host Subsystem

- [HS Multiport USB Host Subsystem Global Initialization: \[0\]](#)
- [EHCI Register Summary: \[1\]](#)

**Table 23-1128. FRINDEX**

<b>Address Offset</b>	0x0000 001C	
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .	<b>Instance</b> EHCI
<b>Description</b>	USB frame index	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FI															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x00000
13:0	FI	Frame index The value in this register is incremented at the end of each time frame.	RW	0x0000

**Table 23-1129. Register Call Summary for Register FRINDEX**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description: \[1\]](#)

**Table 23-1130. CTRLDSSEGMENT**

<b>Address Offset</b>	0x0000 0020																																																																																														
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .																<b>Instance</b>	EHCI																																																																													
<b>Description</b>	4G segment selector																																																																																														
<b>Type</b>	R																																																																																														
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">CDSS</td> <td colspan="16"></td> </tr> </table>																																31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CDSS																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																
CDSS																																																																																															

Bits	Field Name	Description	Type	Reset
31:0	CDSS	This 32-bit register corresponds to the most significant address bits [63:32] for all EHCI data structures.	R	0x00000000

**Table 23-1131. Register Call Summary for Register CTRLDSSEGMENT**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

**Table 23-1132. PERIODICLISTBASE**

<b>Address Offset</b>	0x0000 0024																																																																																														
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .																<b>Instance</b>	EHCI																																																																													
<b>Description</b>	Frame list base address																																																																																														
<b>Type</b>	RW																																																																																														
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">BAL</td> <td colspan="16">RESERVED</td> </tr> </table>																																31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BAL																RESERVED															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																
BAL																RESERVED																																																																															

Bits	Field Name	Description	Type	Reset
31:12	BAL	Base address (low) These bits correspond to memory address signals.	RW	0x00000
11:0	RESERVED	Reserved	R	0x000

**Table 23-1133. Register Call Summary for Register PERIODICLISTBASE**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description: \[1\]](#)

**Table 23-1134. ASYNCLISTADDR**

<b>Address Offset</b>	0x0000 0028	
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .	<b>Instance</b> EHCI
<b>Description</b>	Next asynchronous list address	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPL																RESERVED															

Bits	Field Name	Description	Type	Reset
31:5	LPL	Link Pointer Low It contains the address of the next asynchronous queue head to be executed.	RW	0x00000000
4:0	RESERVED	Reserved	R	0x00

**Table 23-1135. Register Call Summary for Register ASYNCLISTADDR**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description: \[1\]](#)

**Table 23-1136. CONFIGFLAG**

<b>Address Offset</b>	0x0000 0050	
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .	<b>Instance</b> EHCI
<b>Description</b>	Configured flag register	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	CF														

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x00000000
0	CF	Configure Flag This bit controls the default port-routing control logic. 0x0: Port routing control logic default-routes each port to an implementation dependent classic host controller. 0x1: Port routing control logic default-routes all ports to this host controller.	RW	0

**Table 23-1137. Register Call Summary for Register CONFIGFLAG**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description: \[1\] \[2\]](#)

Table 23-1138. PORTSC\_i

<b>Address Offset</b>	0x0000 0054 + (0x4 * i)	<b>Index</b>	i = 0 to 2
<b>Physical Address</b>	See Table 23-1115.	<b>Instance</b>	EHCI
<b>Description</b>	Port Status / Control		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVICEADDRESS								SUSPENDSTATUS	RESERVED	WDE	WCE	PTC				PIC	PO	PP	LS	SUSPENDL1	PR	SUS	FPR	RESERVED	PEDC	PED	CSC	CCS			

Bits	Field Name	Description	Type	Reset
31:25	DEVICEADDRESS	The USB device address for the device attached to and immediately downstream from the associated root port. R/W only if LPM is enabled; otherwise, R.	RW	0x00
24:23	SUSPENDSTATUS	Addition for LPM support.  Indicates status of L1 suspend request. 0x0: Success 0x1: Not yet 0x2: Not supported 0x3: Time-out/error	R	0x0
22	RESERVED	Reserved	R	0
21	WDE	Wake on Disconnect Enable  This field is 0 if the PP bit is 0.  Write 0x1: Enables the port to be sensitive to device disconnects as wake-up events.	RW	0
20	WCE	Wake on Connect Enable  This field is 0 if the PP bit is 0.  Write 0x1: Enables the port to be sensitive to device connects as wake-up events.	RW	0
19:16	PTC	Port Test Control  The port is operating in specific test modes as indicated by the specific value. The encoding of the test mode bits are: 0x0: Test mode not enabled 0x1: Test J_STATE 0x2: Test K_STATE 0x3: Test SE0_NAK 0x4: Test Packet 0x5: Test FORCE_ENABLE Others: Reserved	RW	0x0
15:14	PIC	Port Indicator Control (not implemented)	R	0x0
13	PO	Port Owner  This bit unconditionally goes to a 0x0 when the USBHOST.CONFIGFLAG[0] CF bit makes a 0 to 1 transition. This bit unconditionally goes to 0 whenever the USBHOST.CONFIGFLAG[0] CF bit is 0.  0x1: A companion host controller owns and controls the port.	RW	1
12	PP	Port Power	RW	0



Bits	Field Name	Description	Type	Reset															
		<p>The function of this bit depends on the value of the USBHOST.HCSPARAMS[4] PPC bit. The behavior is as follows:</p> <table border="1"> <thead> <tr> <th>PPC</th> <th>PP</th> <th>Operation</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>0x1</td> <td>Host controller does not have port power control switches. Each port is hard-wired to power.</td> </tr> <tr> <td>0x1</td> <td>N/A</td> <td>Host controller has port power control switches. This bit represents the current setting of the switch (0 = Off, 1 = On).</td> </tr> </tbody> </table> <p>When an overcurrent condition is detected on a powered port and the USBHOST.HCSPARAMS[4] PPC bit is a 1, the PP bit in each affected port may be transitioned by the host controller from 1 to 0.</p>	PPC	PP	Operation	0x0	0x1	Host controller does not have port power control switches. Each port is hard-wired to power.	0x1	N/A	Host controller has port power control switches. This bit represents the current setting of the switch (0 = Off, 1 = On).								
PPC	PP	Operation																	
0x0	0x1	Host controller does not have port power control switches. Each port is hard-wired to power.																	
0x1	N/A	Host controller has port power control switches. This bit represents the current setting of the switch (0 = Off, 1 = On).																	
11:10	LS	<p>Line Status</p> <p>These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. This field is valid only when the port enable bit is 0 and the current connect status bit is set to 1. The encoding of the bits is:</p> <table border="1"> <thead> <tr> <th>Bits[11:10]</th> <th>USB State</th> <th>Interpretation</th> </tr> </thead> <tbody> <tr> <td>0x0</td> <td>SE0</td> <td>Not low-speed device, perform EHCI reset.</td> </tr> <tr> <td>0x2</td> <td>J-state</td> <td>Not low-speed device, perform EHCI reset.</td> </tr> <tr> <td>0x1</td> <td>K-state</td> <td>Low-speed device, release ownership of port.</td> </tr> <tr> <td>0x3</td> <td>Undefined</td> <td>Not low-speed device, perform EHCI reset.</td> </tr> </tbody> </table>	Bits[11:10]	USB State	Interpretation	0x0	SE0	Not low-speed device, perform EHCI reset.	0x2	J-state	Not low-speed device, perform EHCI reset.	0x1	K-state	Low-speed device, release ownership of port.	0x3	Undefined	Not low-speed device, perform EHCI reset.	R	0x0
Bits[11:10]	USB State	Interpretation																	
0x0	SE0	Not low-speed device, perform EHCI reset.																	
0x2	J-state	Not low-speed device, perform EHCI reset.																	
0x1	K-state	Low-speed device, release ownership of port.																	
0x3	Undefined	Not low-speed device, perform EHCI reset.																	
9	SUSPENDL1	<p>When this bit is set to 1, an LPM token is generated.</p> <p>0: Suspend using L2 1: Suspend using L1 (LPM)</p>	RW	0															
8	PR	<p>Port Reset</p> <p>This field is 0 if the PP bit is 0.</p> <p>0x0: Port is not in reset. 0x1: Port is in reset.</p> <p>Write 0x0: Terminate the bus reset sequence. Write 0x1 when at 0x0: The bus reset sequence is started.</p>	RW	0															
7	SUS	<p>Suspend</p> <p>This field is 0 if the PP bit is 0.</p> <p>0x0 when PED = 0x1: Port enabled 0x1 when PED = 0x1: Port in suspend state When PED = 0x0: Port disabled</p>	RW	0															
6	FPR	<p>Force Port Resume</p> <p>This field is 0 if the PP bit is 0.</p> <p>0x0: No resume (K-state) detected/driven on port 0x1: Resume detected/driven on port</p>	RW	0															
5:4	RESERVED	Reserved	R	0															
3	PEDC	<p>Port Enabled/Disabled Change</p> <p>This field is 0 if the PP bit is 0.</p> <p>Read 0x0: No change. Read 0x1: Port enabled/disabled status has changed. Write 0x1: Clears this bit to 0.</p>	RW	0															
2	PED	<p>Port Enabled/Disabled</p> <p>Software cannot enable a port by writing a 1 to this field. The host controller only sets this to 1 when the reset sequence determines that the attached device is a high-speed device.</p> <p>Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software.</p>	RW	0															

Bits	Field Name	Description	Type	Reset
		This field is 0 if the PP bit is 0. 0x0: Disable 0x1: Enable		
1	CSC	Connect Status Change Indicates a change has occurred in the port CCS bit. This field is 0 if the PP bit is 0. Read 0x0: No change Read 0x1: Change in current connect status Write 0x1: Clears this bit to 0	RW	0
0	CCS	Current Connect Status This value reflects the current state of the port, and may not correspond directly to the event that caused the CSC bit to be set. This field is 0 if the PP bit is 0. 0x0: No device is present. 0x1: Device is present on port.	R	0

**Table 23-1139. Register Call Summary for Register PORTSC\_i**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description: \[1\] \[2\] \[3\] \[4\]](#)

**Table 23-1140. INSNREG00**

<b>Address Offset</b>	0x0000 0090	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	Implementation-specific register 0		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																UFRAME_CNT											$\bar{Z}$				

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x00000
13:1	UFRAME_CNT	1-microframe length value, to reduce simulation time SIMULATIONS ONLY, NOT AN ACTUAL REGISTER	RW	0x0000
0	EN	Enable of this register	RW	0

**Table 23-1141. Register Call Summary for Register INSNREG00**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

**Table 23-1142. INSNREG01**

<b>Address Offset</b>	0x0000 0094	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	Implementation-specific register 1		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUT_THRESHOLD																IN_THRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	OUT_THRESHOLD	Programmable output packet buffer threshold, in 32-bit words	RW	0x0020
15:0	IN_THRESHOLD	Programmable input packet buffer threshold, in 32-bit words	RW	0x0020

**Table 23-1143. Register Call Summary for Register INSNREG01**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description: \[1\]](#)

**Table 23-1144. INSNREG02**

<b>Address Offset</b>	0x0000 0098	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	Implementation-specific register 2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUF_DEPTH															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x00000
11:0	BUF_DEPTH	Programmable packet buffer depth, in 32-bit words	RW	0x080

**Table 23-1145. Register Call Summary for Register INSNREG02**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

**Table 23-1146. INSNREG03**

<b>Address Offset</b>	0x0000 009C	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	Implementation-specific register 3		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												BRK_MEM_TRSF			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	BRK_MEM_TRSF	Break Memory Transfer, in conjunction with <a href="#">INSNREG01</a> 0x0: Disabled 0x1: Enabled	RW	1

**Table 23-1147. Register Call Summary for Register INSNREG03**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

**Table 23-1148. INSNREG04**

<b>Address Offset</b>	0x0000 00A0	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	Implementation-specific register 4		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																												NAK_FIX_DIS	RESERVED	SHORT_PORT_ENUM	HCCPARAMS_WRE	HCSPARAMS_WRE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4	NAK_FIX_DIS	Disable NAK fix (don't touch)	RW	0
3	RESERVED	Reserved	R	0
2	SHORT_PORT_ENUM	Scale down Port enumeration time (debug)	RW	0
1	HCCPARAMS_WRE	Make read-only <a href="#">HCCPARAMS</a> register writable (debug)	RW	0
0	HCSPARAMS_WRE	Make read-only <a href="#">HCSPARAMS</a> register writable (debug)	RW	0

**Table 23-1149. Register Call Summary for Register INSNREG04**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)
- [EHCI Register Description:](#)

**Table 23-1150. INSNREG05\_UTMI**

<b>Address Offset</b>	0x0000 00A4	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	Implementation-specific register 5 Register functionality for UTMI mode.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VBUSY	VPORT			VCONTROLLOADM	VCONTROL			VSTATUS							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17	VBUSY	Read 0x0: Vendor interface is done / inactive Read 0x1: Vendor interface is busy	R	0
16:13	VPORT	Vendor interface port selection 0x1: Port 1 vendor interface selected 0x2: Port 2 vendor interface selected	RW	0x0
12	VCONTROLLOADM	UTMI VcontrolLoadM output (active-low) 0x0: Load Vcontrol value into PHY 0x1: No Action	RW	0
11:8	VCONTROL	UTMI Vcontrol output, to be loaded into the PHY	RW	0x0
7:0	VSTATUS	UTMI Vstatus input image, from PHY	R	0x00

**Table 23-1151. Register Call Summary for Register INSNREG05\_UTMI**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

**Table 23-1152. INSNREG05\_ULPI**

<b>Address Offset</b>	0x0000 00A4	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	Implementation-specific register 5 Register functionality for ULPI mode		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONTROL	RESERVED			PORTSEL			OPSEL	REGADD			EXTREGADD			RDWRDATA																	

Bits	Field Name	Description	Type	Reset
31	CONTROL	Control/status of the ULPI register access Write 0x0: No effect 0x0: ULPI access done 0x1: Start ULPI access	RW	0
30:28	RESERVED		R	0x0
27:24	PORTSEL	0x1: Port 1 selected for register access 0x2: Port 2 selected for register access	RW	0x0
23:22	OPSEL	0x2: Register access is Write 0x3: Register access is Read	RW	0x0
21:16	REGADD	ULPI direct register address, for any value different than 0x2F. 0x2F: Triggers an extended address	RW	0x00
15:8	EXTREGADD	Address for extended register accesses. Don't care for direct accesses.	RW	0x00
7:0	RDWRDATA	Read/Write data of (resp. read/write) register access	RW	0x00

**Table 23-1153. Register Call Summary for Register INSNREG05\_ULPI**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

**Table 23-1154. INSNREG06**

<b>Address Offset</b>	0x0000 00A8	<b>Instance</b>	EHCI
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .		
<b>Description</b>	AHB Error Status		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRORCAP	RESERVED															HBURST			BEATSEXP				BEATSCOMP								

Bits	Field Name	Description	Type	Reset
31	ERRORCAP	Indicator that an AHB error was encountered and values were captured. Read 0x0: No error Write 0x0: Clear pending error Write 0x1: No action Read 0x1: Error pending	RW	0
30:12	RESERVED		R	0x00000
11:9	HBURST	HBURST Value of the control phase at which the AHB error occurred.	R	0x0
8:4	BEATSEXP	Number of beats expected in the burst at which the AHB error occurred. Valid values are 0 to 16.	R	0x00
3:0	BEATSCOMP	Number of successfully completed beats in the current burst before the AHB error occurred.	R	0x0

**Table 23-1155. Register Call Summary for Register INSNREG06**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

**Table 23-1156. INSNREG07**

<b>Address Offset</b>	0x0000 00AC	
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .	<b>Instance</b> EHCI
<b>Description</b>	AHB Master Error Address	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASTERERRADD																															

Bits	Field Name	Description	Type	Reset
31:0	MASTERERRADD	AHB Master Error Address	R	0x0000 0000

**Table 23-1157. Register Call Summary for Register INSNREG07**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

**Table 23-1158. INSNREG08**

<b>Address Offset</b>	0x0000 00B0	
<b>Physical Address</b>	See <a href="#">Table 23-1115</a> .	<b>Instance</b> EHCI
<b>Description</b>		
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												NEWBITFIELD1																			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	NEWBITFIELD1		RW	0x0000

**Table 23-1159. Register Call Summary for Register INSNREG08**

High-Speed Multiport USB Host Subsystem

- [EHCI Register Summary: \[0\]](#)

## 23.12 High-Speed USB OTG Controller

This section describes the high-speed USB On-The-Go (OTG) controller of the device.

---

**NOTE:** The high-speed USB OTG controller is an instantiation of the MUSBMHDC from Mentor Graphics Corporation.

This document contains materials that are ©2003-2007 Mentor Graphics Corporation.

Mentor Graphics is a registered trademarks of Mentor Graphics Corporation or its affiliated companies in the United States and other countries.

---

### 23.12.1 High-Speed USB OTG Controller Overview

The High-Speed USB controller is a high-speed, USB OTG dual-role-device (DRD) link controller supporting the following modes:

- USB 2.0 peripheral (function controller) in high/full speed (480/12 Mbps, respectively)
- USB 2.0 host in high/full/low speed (480/12/1.5 Mbps, respectively), with one downstream port but multipoint capability when a hub is connected to it (split transaction support, etc.)
- USB 2.0 OTG DRD in high/full speed, with HNP (OTG1.3) and SRP support.

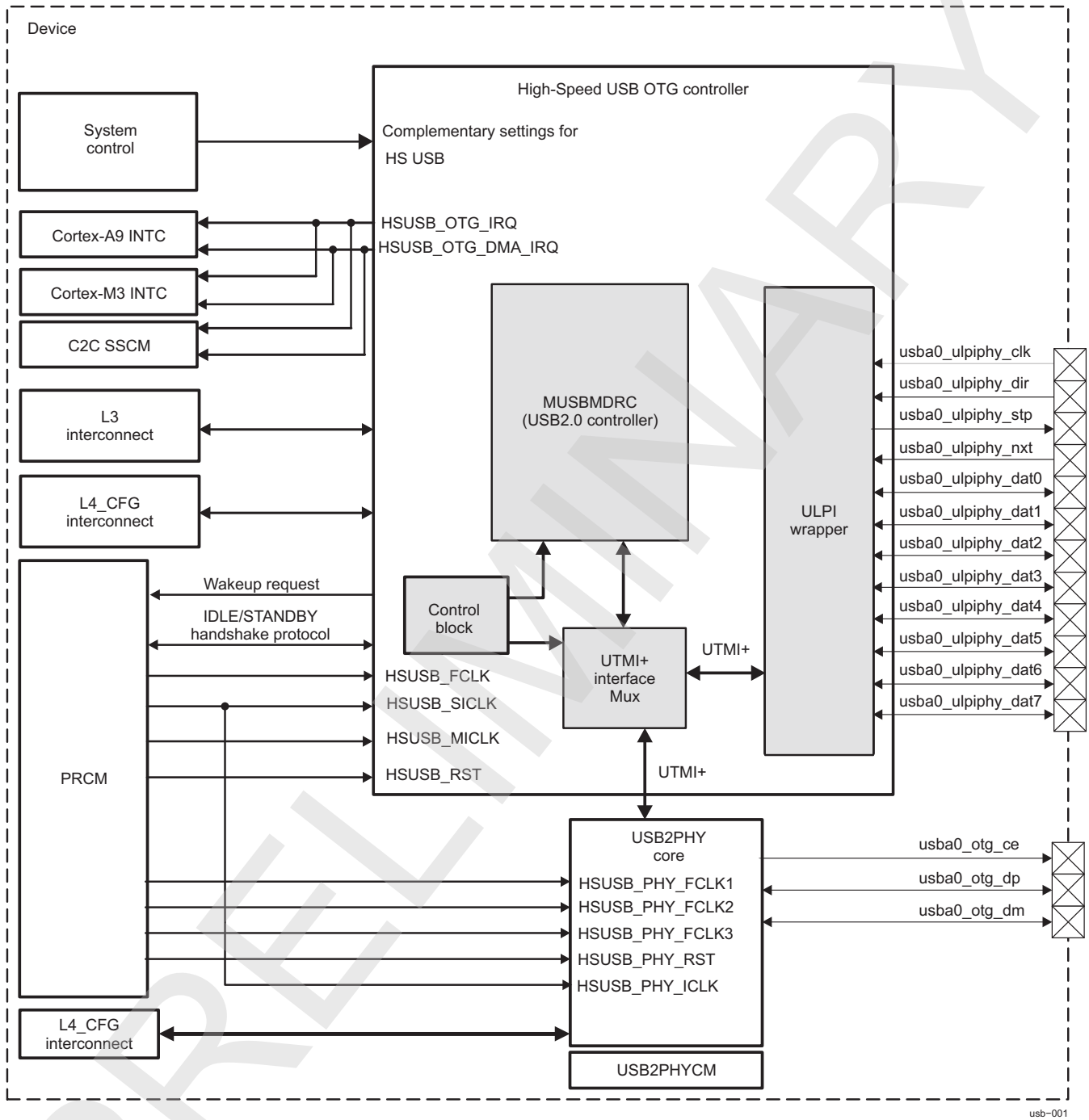
The high-speed USB OTG controller supports a single USB port, which uses the ULPI interface mode, to connect to an off-chip transceiver (12-pin/8-bit data SDR mode) and to connect on D+/D- (+ID) USB bus thanks to an embedded USB-HS OTG PHY.

It is connected to L3 master and L4 slave interfaces. It is clocked from independent functional clock from a PHY. This module operates either as the function controller of a high-/full-speed USB peripheral or as the host/peripheral in point-to-point or multi-point communications with other USB functions.

[Figure 23-258](#) highlights the high-speed USB controller.



Figure 23-258. High-Speed USB OTG Controller Highlight



### 23.12.1.1 Main Features

The high-speed USB controller features:

- L3 (master) interface supports:
  - 32-bit data bus width
  - 8/16/32 bit access supported
  - 32-bit address bus width
  - Burst supported
  - WNP not supported
- L4 (slave) interface supports:
  - 32-bit data bus width
  - 16/32 bit access supported
  - 12-bit address bus width
  - Burst not supported
  - WNP supported
- Complies with USB2.0 standard for high-speed (480 Mb/s) functions and with the OTG 1.3 (On-The-Go) supplement. Implement Link Power Management
- Contains embedded HS PHY with battery charger detection
- Contains PHY (physical layer) interface: ULPI interface (12-bit version)
- 15 transmit endpoints and 15 receive endpoints in addition to control endpoint 0
- Supports OTG communications with one or more high/full/low speed devices.
- Supports packet splitting / combining for bulk transfers
- Soft connect/disconnect option
- Performs all transaction scheduling in hardware
- Each endpoint can be configured for types (interrupt, bulk or isochronous), for size and double buffering operation within a limit of a 8KB of data and total RAM size of 16KB.
- Supports high-bandwidth isochronous and interrupt transfers.
- Supports suspend/resume and remote wake-up.
- Two interrupt lines.
- DMA access to endpoint FIFOs with internal DMA controller supporting 8 channels
- Supports host negotiation and session request protocols (HNP (OTG1.3) and SRP)
- CarKit feature implementation

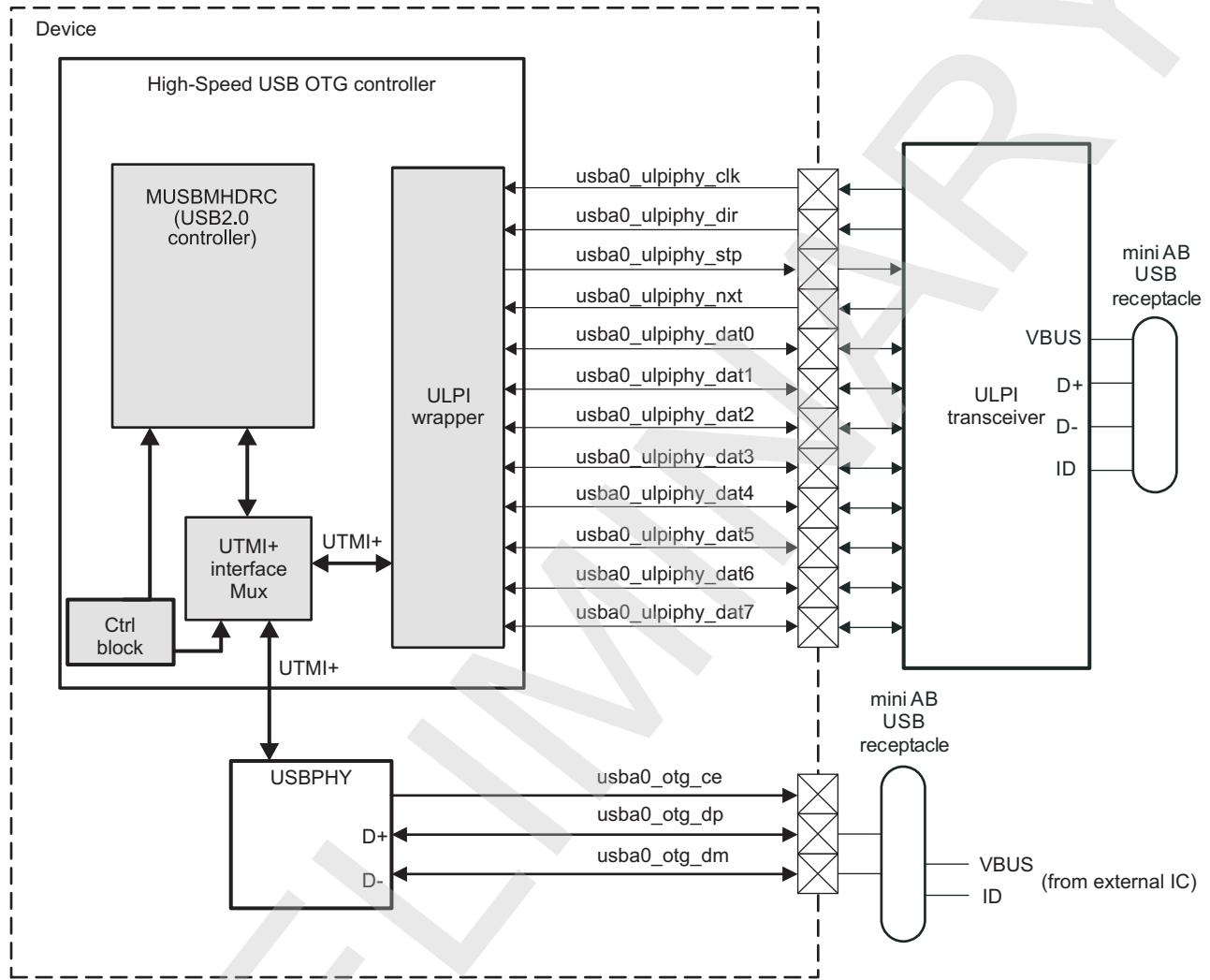
USB-HS OTG PHY (USBPHY). This embedded PHY is composed of two modules:

- USB2PHYCORE contains the USB functions, drivers, receivers and pads for correct D+/D– signalling and the battery charger detection (supporting dead-battery mode). Delivers a 3.3-V output signal (charger enable) to control external battery charger.
- USB2PHYCM contains LDO and bandgap for providing voltage reference to USB2PHYCORE module.

### 23.12.2 High-Speed USB OTG Controller Environment

Figure 23-259 shows a typical application using the high-speed USB controller.

Figure 23-259. High-Speed USB Controller Typical Application System



usb-002

**NOTE:** Only one interface can be active at a time.

The high-speed USB OTG controller supports a single USB port, which uses the ULPI interface mode, to connect to an off-chip transceiver (12-pin/8-bit data SDR mode). A mini A-B external receptacle allows the connection of an external device.

The 12-pin ULPI interface uses an 8-bit data bus with data synchronous to the rising edge of the PHY clock (SDR mode); the 8-pin ULPI uses a 4-bit data bus with data generated on both the rising and falling clock edge (DDR mode).

The current implementation of high-speed USB controller supports only the 12-pin, 8-bit version ULPI interface. The 12-pin/8-bit data SDR ULPI interface is selected through the `OTG_INTERFSEL[2:0]` PHYSEL field and must be set to 0x1.

Embedded PHY interface is 8-bit, UTMI+ and is selected through setting `OTG_INTERFSEL [2:0]` PHYSEL field to 0x0.

### 23.12.2.1 High-Speed USB OTG Controller Functional Interfaces

Table 23-1160 describes the I/O signals of the high-speed USB controller interfaces shown in Figure 23-259

**Table 23-1160. Input/Output Description**

Signal Name	I/O <sup>(1)</sup>	Description	Reset Value
usba0_ulpiphy_clk	I	Clock input from external transceiver <sup>(2)</sup>	HiZ
usba0_ulpiphy_dir	I	Data direction control from external transceiver	HiZ
usba0_ulpiphy_stp	O	Output to external transceiver to stop data stream	1
usba0_ulpiphy_next	I	Next signal control from external transceiver	HiZ
usba0_ulpiphy_dat0	I/O	Data 0 to/from external transceiver	HiZ
usba0_ulpiphy_dat1	I/O	Data 1 to/from external transceiver	HiZ
usba0_ulpiphy_dat2	I/O	Data 2 to/from external transceiver	HiZ
usba0_ulpiphy_dat3	I/O	Data 3 to/from external transceiver	HiZ
usba0_ulpiphy_dat4	I/O	Data 4 to/from external transceiver	HiZ
usba0_ulpiphy_dat5	I/O	Data 5 to/from external transceiver	HiZ
usba0_ulpiphy_dat6	I/O	Data 6 to/from external transceiver	HiZ
usba0_ulpiphy_dat7	I/O	Data 7 to/from external transceiver	HiZ
usba0_otg_ce	O	Charging Enable signal	0
usba0_otg_dp	I/O	USB Data +	HiZ
usba0_otg_dm	I/O	USB Data -	HiZ

<sup>(1)</sup> I = Input; O = Output

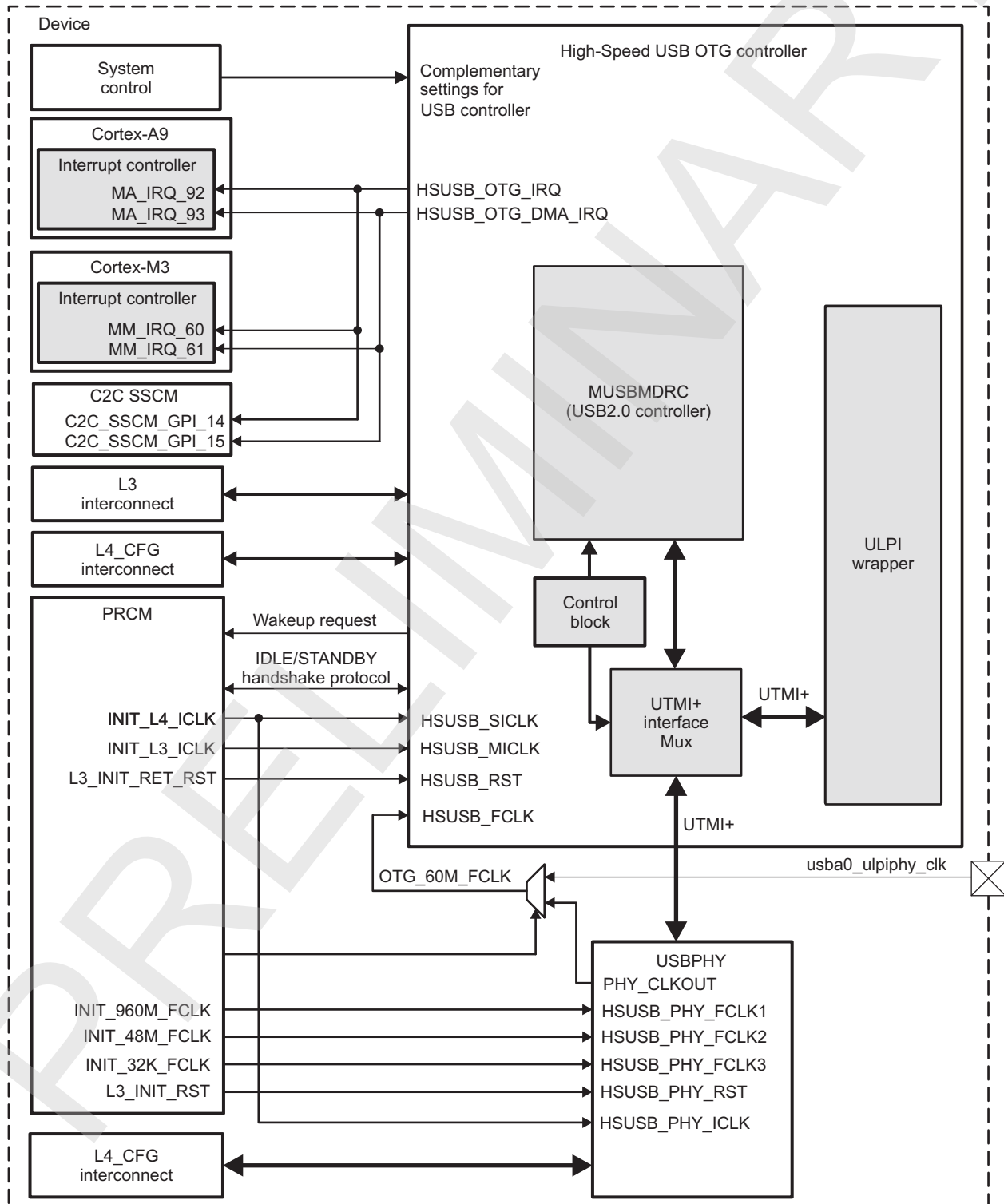
<sup>(2)</sup> This signal is also used as re-timing input.

### 23.12.3 High-Speed USB OTG Controller Integration

The L3 (master) interconnect generates data traffic within the device. The L4(slave) interconnect is a configuration port for register setting.

Figure 23-260 shows the high-speed USB controller integration in the device.

Figure 23-260. High-Speed USB Controller Integration



usb-003

Table 23-1161 through Table 23-1163 summarize the integration of the module in the device.

**Table 23-1161. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
HSUSBOTG	PD_L3_INIT	L3 L4_CFG
USBPHY	PD_L3_INIT	L4_CFG

Three clocks are provided to the high-speed USB controller, as listed in Table 23-1162

**Table 23-1162. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HSUSBOTG	HSUSB_FCLK	OTG_60M_FCLK	usb0_ulpiphy_clk/ USBPHY	Functional clock. Source is selected between external and embedded PHY. Section 3.6.3.2, <i>CM Clock Source</i> , in Chapter 3, <i>Power, Reset, and Clock Management</i> .
	HSUSB_SICLK	INIT_L4_ICLK	PRCM	L4 interconnect enable signal for the slave interface on L4 Section 3.6.3.2, <i>CM Clock Source</i> in Chapter 3, <i>Power, Reset and Clock Management</i>
	HSUSB_MICLK	INIT_L3_ICLK	PRCM	L3 interconnect clock, for the L3 master port interface Section 3.6.9, <i>CD_L3_INIT Clock Domain</i> in Chapter 3, <i>Power, Reset and Clock Management</i> .
USBPHY	HSUSB_PHY_FCLK1	INIT_960M_FCLK	PRCM	Functional clock
	HSUSB_PHY_FCLK2	INIT_48M_FCLK	PRCM	Optional functional clock
	HSUSB_PHY_FCLK3	INIT_32K_FCLK	PRCM	Optional functional clock
	HSUSB_PHY_ICLK	INIT_L4_ICLK	PRCM	Interface clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HSUSBOTG	HSUSB_RST	L3_INIT_RET_RST	PRCM	HSUSBOTG controller hardware reset
USBPHY	HSUSB_PHY_RST	L3_INIT_RST	PRCM	USBPHY module hardware reset

Table 23-1163 lists the interrupt lines that are driven out from the high-speed USB controller.

**Table 23-1163. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
HSUSBOTG	HSUSB_OTG_IRQ	MA_IRQ_92	Cortex™-A9 INTC	HSUSBOTG interrupt request
	HSUSB_OTG_DMA_IRQ	MA_IRQ_93	Cortex-A9 INTC	HSUSBOTG DMA interrupt request
	HSUSB_OTG_IRQ	MM_IRQ_60	Cortex™-M3 INTC	HSUSBOTG interrupt request
	HSUSB_OTG_DMA_IRQ	MM_IRQ_61	Cortex-M3 INTC	HSUSBOTG DMA interrupt request
	HSUSB_OTG_IRQ	C2C_SSCM_GPI_1_4	C2C SSCM	HSUSBOTG interrupt request
	HSUSB_OTG_DMA_IRQ	C2C_SSCM_GPI_1_5	C2C SSCM	HSUSBOTG DMA interrupt request

## **23.12.4 High-Speed USB OTG Controller Functional Description**

### **23.12.4.1 Software Reset**

The high-speed USB controller has a software reset through the [OTG\\_SYSCONFIG\[1\] SOFTRESET](#) bit. Writing 0x1 to this bit initiates reset of the module. The bit value 0x1 remains until the reset is complete. When the software reset is complete, the SOFTRESET bit is automatically cleared to 0x0 and has the same effect as the hardware reset. The status of the reset can be observed through the [OTG\\_SYSSTATUS\[0\] RESETDONE](#) bit.

### **23.12.4.2 Power Management**

#### **23.12.4.2.1 Overview**

To save dynamic power consumption, an efficient idle scheme in the device is based on the following:

- An efficient local autoclock gating for each module
- The implementation of control sideband signals between the PRCM module and each module

This enhanced idle control allows clocks to be activated/deactivated safely without complex software intervention. In both cases, the high-speed USB controller power management is applied only to the interface clock domain. The USB OTG functional clock, HSUSB\_FCLK, is controlled by the transceiver and is responsible only for a very small percentage of the module overall power consumption.

The high-speed USB controller has both master (initiator) and slave (target) interfaces.

- As an initiator, the high-speed USB OTG controller implements the standby handshake protocol to inform the PRCM module when it enters standby mode and does not generate traffic on the interconnect.
- As a target, the high-speed USB OTG controller implements the IDLE handshake protocol to allow the PRCM module requiring it to enter idle mode.

#### **23.12.4.2.2 System Power Management**

##### **23.12.4.2.2.1 Standby Handshake Protocol**

The high-speed USB OTG controller can choose to go to standby mode, in which case it stops generating transactions on the interconnect. The module standby leads the PRCM to disable the USB clocks to save power.

The high-speed USB OTG controller has a MSTANDBY handshake mechanism with the PRCM module

The module is ready to enter standby mode (indicated by the MSTANDBY signal to the PRCM asserted) when the USB is in suspend mode and the module is idle. It means the following:

- The module is committed not to start any new transaction on its master interface.
- The module is idle and, therefore, the power manager can start the procedure to turn off the interface clock, if needed. This procedure must be implemented using the slave power-management protocol.

The handshake mechanism lets the module to go to standby state based on the [OTG\\_SYSCONFIG\[13:12\] MIDDLEMODE](#) field.

- Smart-standby

The high-speed USB controller is configured in smart-standby mode ([OTG\\_SYSCONFIG\[13:12\] MIDDLEMODE](#) field = 0x2). The module is ready to enter standby mode (MSTANDBY is asserted) when there is no more activity on the USB master interface of the interconnect. MSTANDBY is asserted when the module is idle and deasserted when the module is activated by either an external USB event or an appropriate register access. The module then waits for MWAIT deassertion before a DMA transfer is started.

- Force-standby

– When the high-speed USB controller operates as a host: The USBOTG.POWER[1] SUSPENDMODE bit is set to 0x1 to bring the module to low-power mode (suspend mode). After this setting, the high-speed USB controller waits for its idle state. Software must set

**OTG\_SYSCONFIG**[13:12] MIDDLEMODE = 0x0. Hardware then asserts MSTANDBY. Similarly, to release the MSTANDBY signal, an appropriate register access must be applied, which can be either of the following two cases:

- Remote wakeup causes a RESUME interrupt
  - Set the USBOTG.POWER[3] RESET bit to 0x1
  - Write 0x2 to the **OTG\_SYSCONFIG**[13:12] MIDDLEMODE.  
OR
  - Set the USBOTG.POWER[3] RESET bit to 0x1
  - Write 0x2 to the **OTG\_SYSCONFIG**[13:12] MIDDLEMODE.
- When the high-speed USB controller operates as a peripheral: When the USB bus is idle for 3 ms, a SUSPEND interrupt is generated by the high-speed USB controller. Software must set **OTG\_SYSCONFIG**[13:12] MIDDLEMODE = 0x0. Hardware then asserts MSTANDBY. The high-speed USB controller then asserts MSTANDBY. Similarly, to release the MSTANDBY signal, an appropriate register access must be applied, which can be either of the following two cases:
- Set the USBOTG.POWER[2] RESUME bit to 0x1.
  - Write 0x2 to the **OTG\_SYSCONFIG**[13:12] MIDDLEMODE.  
OR
  - RESET interrupt is generated by the high-speed USB controller.
  - Write 0x2 to the **OTG\_SYSCONFIG**[13:12] MIDDLEMODE.

When MSTANDBY is deasserted as a consequence of the previous register access, the module waits for MWAIT deassertion before a DMA transfer is started.

- No-standby

The high-speed USB controller is configured in no-standby mode (**OTG\_SYSCONFIG**[13:12] MIDDLEMODE field = 0x1). The module never enters standby mode (that is, MSTANDBY is never asserted).

[Table 23-1164](#) describes the high-speed USB standby mode settings.

**Table 23-1164. HSUSBOTG Host Controller STANDBYMODE Settings**

<b>OTG_SYSCONFIG</b> [13:12] MIDDLEMODE Value	Selected Mode	Description
0x0	Force-standby	The HSUSBOTG host controller enters standby mode unconditionally
0x1	No-standby	The HSUSBOTG host controller never enters standby mode
0x2	Smart-standby	The HSUSBOTG host controller is ready to enter standby mode when there is no more activity on the USB master interface of the interconnect.

#### 23.12.4.2.2.2 IDLE Handshake Protocol

The high-speed USB controller can be configured through the **OTG\_SYSCONFIG**[4:3] SIDLEMODE field as one of the following acknowledgment modes:

- Smart-idle mode

When the high-speed USB controller receives an IDLE request from the PRCM module:

- The interface clock INIT\_L3\_ICLK is disabled, see [Section 3.6.9, CD\\_L3\\_INIT Clock Domain](#)
- L4 interface clock idle transitions:

Configured in smart-idle mode (**OTG\_SYSCONFIG**[4:3] SIDLEMODE field = 0x2), the high-speed USB controller checks for no ongoing activity. The idle acknowledge then is asserted and the module waits for active system clock gating by the PRCM module (this occurs only when all peripherals supplied by the same L3 clock domain are also ready for idle).

Once in smart idle mode (when the PRCM module gates the interface clock), the module has no activity, the interface clock paths are gated, no interrupt request can be generated, and the module is ready to issue a wake-up request. If a wake-up condition occurs, the high-speed USB controller exits from smart idle mode if the **OTG\_SYSCONFIG**[2] ENABLEWAKEUP bit is set to 0x1 (wake-up capability enabled).



- Smart-idle wake-up mode  
The high-speed USB OTG controller is configured in smart-idle wake-up mode (**OTG\_SYSCONFIG**[4:3] Smart Idle Wkp field = 0x3). The functionality of this mode is the same as smart-idle mode, additionally Wake up feature is enabled.
- Force-idle mode  
When the high-speed USB controller receives an IDLE request from the PRCM module:
  - The interface clock INIT\_L3\_ICLK is disabled, see [Section 3.6.9, CD\\_L3\\_INIT Clock Domain](#)
  - The L4 interface clock idle transitions:  
Configured in force-idle mode (**OTG\_SYSCONFIG**[4:3] SIDLEMODE field = 0x0), the high-speed USB controller waits unconditionally for active system clock gating by the PRCM module (this occurs only when all peripherals supplied by the same L3 clock domain are also ready for idle).  
Once in idle mode (when the PRCM module gates the interface clock), the module has no activity, the interface clock paths are gated, no interrupt request can be generated, and the wake-up feature is totally inhibited.
- No-idle mode  
When the high-speed USB controller receives an IDLE request from the PRCM module:
  - The interface clock INIT\_L3\_ICLK is disabled [Section 3.6.9, CD\\_L3\\_INIT Clock Domain](#)
  - L4 interface clock idle transitions  
Configured in no-idle mode (**OTG\_SYSCONFIG**[4:3] SIDLEMODE bit field = 0x1), the high-speed USB controller module does not go to idle mode and the idle acknowledge is never sent.

[Table 23-1165](#) describes the high-speed USB sidle mode settings. For more information, see [Chapter 3, Power, Reset, and Clock Management](#).

**Table 23-1165. HSUSBOTG Host Controller SIDLEMODE Settings**

<b>OTG_SYSCONFIG</b> [4:3] <b>SIDLEMODE Value</b>	<b>Selected Mode</b>	<b>Description</b>
0x0	Force-idle	The HSUSBOTG host controller acknowledges unconditionally the IDLE request from the PRCM, regardless of its internal operations.
0x1	No-idle	The HSUSBOTG host controller never acknowledges any IDLE request from the PRCM.
0x2	Smart-idle	The HSUSBOTG host controller acknowledges the IDLE request basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, IRQs or DMA requests are treated.
0x3	Smart-idle wake-up	The HSUSBOTG host controller enters smart mode with wakeup enabled.

#### 23.12.4.2.2.3 Wake-Up Request

Wake-up request is enabled by setting **OTG\_SYSCONFIG**[2] ENABLEWAKEUP to 0x1. The enable wake-up feature works only when smart-idle wakeup is enabled.

#### 23.12.4.2.3 Local Power Management

The high-speed USB OTG controller has local power management by internal clock gating features:

- Internal interface clock autogating: Clock for the L3 interconnect logic can be gated when the module is not accessed, if the **OTG\_SYSCONFIG**[0] AUTOIDLE bit is set. Otherwise, this logic is free-running on the interface clock. This bit is used to save power when the module is not used because of the multiplexing configuration selected at the chip level. This bit has precedence over all other internal configuration bits.

#### 23.12.4.3 Inventra MUSBMHDC

The core functionality of the module is provided by a third-party IP, the Inventra™ MUSBMHDC from Mentor Graphics® Corporation.

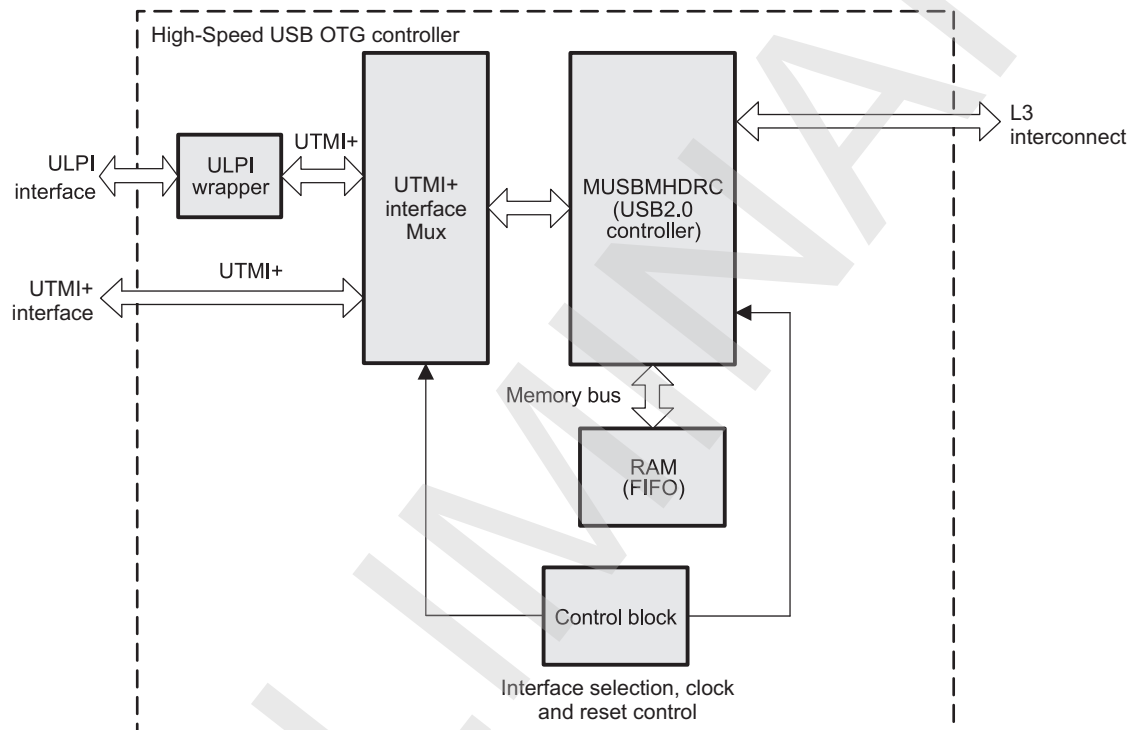
The high-speed USB OTG controller is basically the appropriate adaptation/integration of this IP to comply with TI requirements.

The high-speed USB OTG controller includes the following:

- MUSBMHDRC (USB 2.0 controller)
- Master and slave bridges for conversion AHB to L3 interconnect and L3 interconnect to AHB
- Built-in self-test (BIST) controller
- Two memories (total of 16KB; available for various endpoints)
- Reset and clock generation
- DFT compliance with regards to latest specs

Figure 23-261 shows the high-speed USB controller.

**Figure 23-261. High-Speed USB OTG Controller**



usb-004

#### 23.12.4.4 Configuration

The following options are implemented in the TI high-speed USB OTG controller:

- 8-bit internal data path processing width (mandated by the use of ULPI)
- 32-bit vcontrol/vstatus support
- External charge pump: Hacked
- Software connect/disconnect supported
- Little-endian and big-endian byte ordering
- Eight DMA channels (with internal DMA initiator implemented)
- Dynamic FIFO sizing enabled
- 16-KB RAM buffer
- 15 IN/OUT endpoints in addition to control endpoint 0
- IN/OUT bulk packet splitting enabled
- High-bandwidth IN/OUT isochronous support enabled (see *Universal Serial Bus Specification Revision 2.0*)

#### 23.12.4.5 Basic Operation

This section provides an overview of the module basic operation.

To implement the most time-critical functions in hardware, the module provides all of the encoding, decoding, and checking required to send and receive USB packets, interrupting the MPU only when endpoint data is successfully transferred. Generally, the following steps are performed.

##### 23.12.4.5.1 Module Initialization

First, the firmware must do the overall initialization of the module by configuring the interrupts, DMA controller, and individual endpoints.

The specific items of a configuration for each endpoint are:

- Direction TX/RX
- Speed: High/full/low
- Special host settings when used in host mode (function address/hub parameters, etc.)
- Transaction protocol: Control/isochronous/bulk/interrupt
- Maximum packet size, that can be transferred through the endpoint
- Whether DMA is required (DMA enable)
- DMA mode
- Start address of the endpoint FIFO within the RAM block
- Maximum packet size allowed
- Whether double-buffering is required. This and the previous option define the amount of space that must be allocated to the FIFO. In the case of double-buffering, the FIFO size is doubled; therefore, the maximum supported size is 8196 bytes for the endpoint FIFOs.

---

**NOTE:** On the system level, it must be ensured that the reset and the interface are selected before the functional clock (OTG\_60M\_FCLK) is applied to the module. Correct functionality is not ensured if a reset deassertion or a change in the interface selection occurs when the functional clock OTG\_60M\_FCLK is already running.

---

##### 23.12.4.5.2 Transaction Handling

Depending on whether the module is in the host or peripheral mode, an RX endpoint is assigned to IN or OUT transactions, respectively. Similarly, a TX endpoint is used for IN transactions in the peripheral mode and for OUT transactions in the host mode. When a transaction is complete, the appropriate endpoint interrupt is generated (if enabled).

When a transaction is handled through an RX endpoint, the received packet is placed in the RX FIFO, and

the appropriate RX endpoint interrupt is generated. The packet can then be unloaded from the FIFO either manually (that is, by the MPU) or by the DMA. Although an RX endpoint is always active for a peripheral, in host mode the transaction must first be initiated by setting the appropriate request flag (REQPKT). This indicates to the transaction scheduler that there is an active transaction on this endpoint, which requires an IN token to be sent to the target function.

When a transaction is handled through a TX endpoint, the packet to send is loaded into the TX FIFO either manually (that is, by the MPU) or by the DMA. The transaction then must be initiated by setting the appropriate ready flag (TXPKTRDY) to indicate the availability of the new packet. When the packet is successfully sent (scheduled according to the protocol rules), the appropriate TX interrupt is generated.

### 23.12.4.6 Optional Features

This section gives a quick overview of the optional features that are available, depending on the endpoint configuration.

#### 23.12.4.6.1 Double Packet Buffering

When double packet buffering is enabled (by setting the MSB of the FIFOSZ register) two data packets can be stored in the FIFO. This option can be set for both the TX and RX endpoints in both the peripheral and host mode. For a TX endpoint, double-buffering means that up to two packets can be loaded into the FIFO awaiting transmission; for an RX endpoint, one packet can be received while another is being read. Double packet buffering is especially advisable for isochronous transactions to avoid underrun or overrun errors.

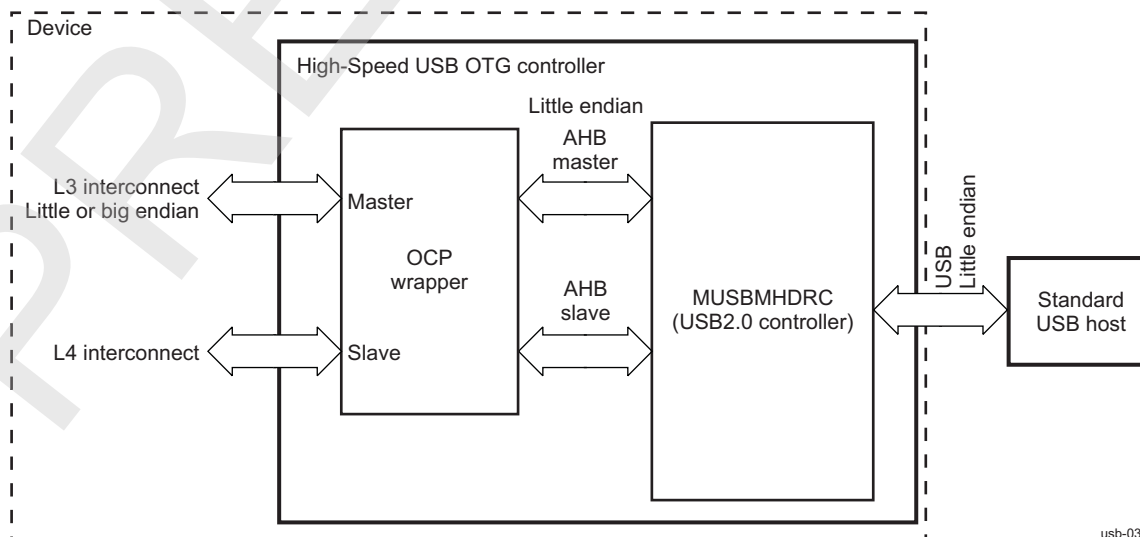
The high-speed USB controller provides dynamic FIFO sizing with an overall RAM size of 16KB, which can then be allocated to the different endpoints when the module is initialized. The maximum size of an endpoint FIFO is 4096 bytes for single packet buffering and 8192 bytes for double packet buffering. The firmware must ensure that a block of RAM is properly assigned to all TX and RX endpoints, considering the total RAM size and the maximum packet size set for the endpoint.

#### 23.12.4.6.2 USBOTGHS Support for Big Endian

In the example in [Figure 23-262](#), the high-speed USB OTG controller module is configured as a USB device and is connected to a standard USB host. The two interconnect interfaces of the high-speed USB OTG controller connect it to the system bus. The register read/write is performed through the interconnect slave interface (control path). The data transfer to/from system memory takes place through the interconnect master interface (data path).

The data transfer on USB is always in little-endian mode, but on interconnect it can be in little- or big-endian mode, as shown in [Figure 23-262](#).

**Figure 23-262. High-Speed USB OTG Controller Endianness**



usb-036

Big-endian support in the high-speed USB OTG controller may or may not require byte-swapping in the data path. Byte-swapping is performed based on the unit size of data. For example, if multiple bytes are packed into a single 32-bit word, these bytes must be swapped in a big-endian interconnect. On the other hand, if the data is simply a 32-bit word (for example, a 32-bit descriptor pointer in a memory buffer), the bytes in the word must not be swapped in big-endian mode.

In the high-speed USB OTG controller, the unit size of descriptors/descriptor pointers is 32 bits, and the unit size of USB data is 8 bits. Four 8-bit USB data are packed to form a 32-bit word. This 32-bit data requires byte-swapping, whereas the descriptors/descriptor pointers do not require byte-swapping.

---

**NOTE:** Byte-swapping is not required in the control path (slave interface).

---

The interconnect master interface (a submodule within the interconnect wrapper) supports big-endian conversion by setting the `OTG_BIGENDIAN[0]` `BIG_ENDIAN` bit to 0x1.

For interconnect write transactions, the data word is swapped, as shown in [Table 23-1166](#), before its output on the interconnect master interface. In case of an interconnect read transaction, the received data is swapped before its output on the AHB master interface of the MUSBMHDCR core. The interconnect `MByteEn` is swapped before the command is issued on the interconnect master interface. The interconnect `MAddr` is always aligned to the size of the interconnect data (32 bits). Hence, the interconnect `MAddr` is always word-aligned.

**Table 23-1166. Interconnect Data and MByteEn in Little- and Big-Endian Modes**

Unit Size of Data	AHB Size (of Packed/Nonpacked Data)	AHB Start Address Offset (Always in Little-Endian Mode)	Interconnect MByteEn in Little-Endian Mode	Interconnect MByteEn in Big-Endian Mode	Interconnect Data in Little-Endian Mode	Interconnect Data in Big-Endian Mode
Byte	Byte	0	0001	1000	xxx <b>b</b> 0	b0xxx
Byte	Byte	1	0010	0100	xxb0x	xb0xx
Byte	Byte	2	0100	0010	xb0xx	xxb0x
Byte	Byte	3	1000	0001	b0xxx	xxx <b>b</b> 0
Byte	Half-word	0	0011	1100	xxb1b0	b0b1xx
Byte	Half-word	2	1100	0011	b1b0xx	xxb0b1
Byte	Word	0	1111	1111	b3b2b1b0	b0b1b2b3
Half-word	Half-word <sup>(1)</sup>	0	0011	1100	xxb1b0	b1b0xx
Half-word	Half-word <sup>(1)</sup>	2	1100	0011	b1b0xx	xxb1b0
Half-word	Word <sup>(1)</sup>	0	1111	1111	b3b2b1b0	b1b0b3b2
Word	Word <sup>(2)</sup>	0	1111	1111	b3b2b1b0	b3b2b1b0

<sup>(1)</sup> Not a likely scenario for high-speed USB OTG controller

<sup>(2)</sup> Corresponds to descriptor and descriptor pointer. No bytes are packed to form a word; hence, no byte-swapping is done in this case.

### 23.12.4.6.3 DMA

The high-speed USB controller has an internal DMA controller for efficient loading/unloading of the endpoint FIFOs. The DMA controller is configured for 8 channels and it has its own block of control registers and its own interrupt controller

If the DMA is enabled for a TX endpoint, a DMA transfer occurs whenever the endpoint can accept another packet in its FIFO (that is, the DMA controller loads the packets into the FIFO without processor intervention).

If the DMA is enabled for an RX endpoint, a DMA transfer occurs whenever the endpoint has a packet in its FIFO (that is, the DMA CONTROLLER unloads the packets from the FIFO without processor intervention).

The high-speed USB controller supports two DMA request modes, which also affects the generation of endpoint interrupts.

- DMA request mode 0 is especially advisable for isochronous transfers but can also be used for bulk and interrupt transfers.
- DMA request mode 1 is mainly valuable for bulk transfers, where typically a large block of data is split into a series of packets of the maximum size.

#### 23.12.4.6.3.1 DMA Request Mode 0

For RX endpoints, the DMA transfer is initiated when a data packet is available in the endpoint FIFO. The appropriate endpoint interrupt is also generated after the packets are received.

For TX endpoints, the DMA transfer is initiated when the endpoint FIFO can accept a data packet. The appropriate endpoint interrupt is also generated to prompt the loading of the packets.

#### 23.12.4.6.3.2 DMA Request Mode 1

For RX endpoints the DMA transfer is initiated only when the received packet is the maximum packet size (as set in the RXMAXP register for the corresponding endpoint). The appropriate endpoint interrupt is generated only when the received packet is a short packet (that is, one less than the maximum packet size); otherwise, it is suppressed.

For TX endpoints, the DMA transfer is initiated when the endpoint FIFO can accept a data packet. The appropriate endpoint interrupt is suppressed.

#### 23.12.4.6.3.3 Internal DMA Controller

The high-speed USB controller has an internal DMA controller to perform DMA transfers acting as host on the L3 interconnect. The DMA controller has eight channels, which can be independently programmed based on the following configuration options:

- DMA enable
- Direction (DMA write/DMA read)
- DMA mode (transfer one or more packets)
- Interrupt enable
- Endpoint number
- L3 interconnect memory address (32 bits)
- Byte count

The DMA controller can issue single accesses (8-bit, 16-bit, or 32-bit) and also bursts (4 × 32-bit, 8 × 32-bit, or 16 × 32-bit) on the L3 interconnect. The start address provided to the DMA controller must be 32-bit aligned.

#### 23.12.4.7 Automatic Packet Splitting/Combining for Bulk Transfers

The high-speed USB controller offers the facility for bulk endpoints to store larger amounts of data in their FIFOs than can be transferred in a single USB operation. In other words, the module includes a configuration option that, if selected, allows larger data packets to be written to bulk TX endpoints, which are then split into packets of an appropriate size for transfer across the USB. A similar option exists for bulk RX endpoints, which, if selected, causes the module to combine the packet received across the USB into larger data packets before being read by the application software.

The firmware can enable these options by setting the corresponding bits (MPRXE and MPTXE) in the CONFIGDATA register. The necessary packet size information contains the payload size for one transaction and a multiplier defining the maximum number of USB packets. The payload is required to be either 8, 16, 32, 64, or (in case of high-speed transfers) 512 bytes; the multiplier can be any value up to 32.

#### 23.12.4.8 High-Bandwidth Isochronous Endpoints

In the high-speed mode, isochronous endpoints can transfer up to three USB packets in any microframe with a payload of up to 1024 bytes in each packet, corresponding to a data transfer rate of up to 3072 bytes per microframe (see *Universal Serial Bus Specification Revision 2.0*).

For TX endpoints, the high-speed USB controller supports this by allowing loading data packets of up to 3072 bytes (that is,  $3 \times 1024$  bytes) into the associated endpoint FIFO, which is then automatically split into USB packets of the maximum payload, or smaller to be transmitted in one microframe.

For RX endpoints, the module automatically combines all the USB packets received during a microframe into a single packet of up to 3072 bytes (that is,  $3 \times 1024$  bytes) within the RX FIFO.

The number of USB packets transferred per microframe and the maximum payload in each packet is defined through the appropriate registers (RXMAXP and TXMAXP).

#### 23.12.4.9 Carkit Implementation

Carkit feature implementation:

UART3 RX/TX muxing on the `usba0_ulpiphy_dat0` and `usba0_ulpiphy_dat1`. To support carkit interrupt through `USB0HS.DATA3`, three different methods can be used:

- External PHY interrupt if provided: `IRQ_SYS` interrupt
- UART3 interrupt: CTS edge detection
- GPIO interrupt: Edge detection on GPIO associated to `usba0_ulpiphy_dat3` line

#### 23.12.4.10 USB PHY

The device-embedded USB PHY is composed of the following modules:

- **USB2PHYCORE**: Contains the USB functions, drivers, receivers, and pads for correct D+/D– signaling, and the battery charger detection (supporting dead-battery mode). Delivers a 3.3-V output signal (charger enable [pin `usba0_otg_ce`]) to control an external battery charger.
- **USB2PHYCM**: Contains LDO and bandgap to provide voltage reference to the **USB2PHYCORE** module

The embedded PHY does not support the OTG features of USB-OTG standards (that is, ID pin detection and VBUS detection). This function is exported to an external companion chip (for example, the TWL6030). Software must retrieve the OTG HNP and SRP status from the TWL6030 (using the I2C1 interface) and configure the bits inside the control module that drive the related USBOTGHS UTMI+ interface signals. It must also read back the UTMI+ signals needed to configure the TWL6030 OTG module.

These signals are:

- Inputs of the USB controller driven by the control module:
  - `Avalid`: Indicates whether VBUS is above the A-Device session valid threshold
  - `Bvalid` (currently not used)
  - `Vbusvalid`: Indicates whether VBUS is above the threshold for normal operation
  - `Sessend`: Indicates whether VBUS is below the B-Device session end threshold
  - `Iddig`: Indicates the value of the ID pin of the USB connector
- Outputs of the USB controller readable in the control module:
  - `Idpullup`: Enables sampling of the ID pin of the USB connector
  - `Drv vbus`: Controls the driving of 5-V power on VBUS
  - `Chrg vbus`: Controls the charging of VBUS for initiating SRP
  - `Dischrg vbus`: Controls the discharging of VBUS for completing SRP

See the `CONTROL_USBOTGHS_CONTROL` register in [Chapter 18, Control Module](#), and the TWL6030 TRM.

The following UTMI signals are not connected:

- `Txbitstufferable` (no bit-stuffing out from the USB controller)
- `Txbitstufferableh` (because only 8 bits are used)

The following USBPHY signals are routed internally to the GPIO6 module pins to generate interrupts when any of the following events occur:

- **CHGDETECTED**: This signal indicates the output of the charger detection protocol.  
0: Charger not detected

1: Charger detected

The value is latched until the charger protocol is reset.

- **DPDM5VSHORT:** This signal is asserted high if a short-circuit-to-VBUS condition is detected on the USB DP or DM pins. This signal is valid only if all power supplies are present. The short is detected if any or even none of the supplies are present.

0: No short

1: VBUS short on DP or DM detected

For information about how to enable these interrupts, see [Chapter 25, General-Purpose Interface](#).

### 23.12.5 High-Speed USB OTG Controller Basic Programming Model

This section describes the low-level hardware programming sequences for configuration and use of the high-speed USB OTG controller.

#### 23.12.5.1 Global Initialization

##### 23.12.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the high-speed USB OTG controller is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the high-speed USB OTG controller.

**Table 23-1167. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	All required clocks must be enabled, see <a href="#">Table 23-1162</a> . Do not apply OTG_60M_FCLK clock before the interface is selected. Refer to the <a href="#">Section 3.6.3.2</a>
Cortex-A9 MPU INTC or Cortex-M3 MPU INTC	Cortex-A9 MPU or Cortex-M3 MPU interrupt controller must be configured to enable the interrupt request generation to the Cortex-A9 MPU or Cortex-M3 MPU subsystem when interrupt requests are generated by the High-Speed Multiport USB Host Subsystem. See the respective Functional Description in the Interrupt Controller chapter for the MPUs.
Interconnect (L3 and L4)	See <a href="#">Section 13.2, L3 Interconnect</a> , and <a href="#">Section 13.3, L4 Interconnect</a> , for more information.

##### 23.12.5.1.2 High-Speed USB OTG Controller Global Initialization

This procedure initializes the high-speed multiport USB host subsystem after a power-on or software reset. The module USBPHY is reset through boundary reset signal L3\_INIT\_RST coming from PRCM module.

**Table 23-1168. High-Speed USB OTG Software Reset**

Step	Register/Bit Field/Programming Model	Value
Perform a software reset on the USB OTG controller	OTG_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is finished	OTG_SYSSTATUS[0] RESETDONE	=0x1

**Table 23-1169. Power Management Configuration**

Step	Register/Bit Field/Programming Model	Value
Select master interface standby mode	OTG_SYSCONFIG[13:12] MIDLemode	0x-
Select slave interface idle mode	OTG_SYSCONFIG[4:3] SIDLemode	0x-
Set clock autogating	OTG_SYSCONFIG[0] AUTOIDLE	0x-



### 23.12.5.2 High-Speed USB OTG Controller Interface Modes

This procedure describes TI high-speed USB controller interface selection:

**Table 23-1170. High-Speed USB Controller Interface Selection**

Step	Register/Bit Field/Programming Model	Value
Select between embedded and external PHY interface	<a href="#">OTG_INTERFSEL</a> [2:0] PHYSEL	0x-
Enable functional PHY clock (OTG_60M_FCLK)	Refer to <a href="#">Section 3.6.3.2</a>	-

### 23.12.5.3 High-Speed USB OTG Controller Power Management Configuration

This section describes the settings for optimal high-speed USB controller power management, depending on the use of this module in the application.

#### 23.12.5.3.1 No Application Mode

The optimal configuration when the high-speed USB controller is not used by the application is as follows:

**Table 23-1171. Power Management Configuration - No Application Mode**

Step	Register/Bit Field/Programming Model	Value
Switch master interface to force standby mode	<a href="#">OTG_SYSCONFIG</a> [13:12] MIDDLEMODE	0x0
Switch slave interface to force-idle mode	<a href="#">OTG_SYSCONFIG</a> [4:3] SIDLEMODE	0x0
Set the clock to smart mode - clock is cut-off if no activity on the interface	<a href="#">OTG_SYSCONFIG</a> [0] AUTOIDLE	0x1

#### 23.12.5.3.2 Host or Peripheral Mode

When used as a host or in peripheral mode, the high-speed USB controller must be programmed as follows:

**Table 23-1172. Power Management Configuration - Host or Peripheral Mode**

Step	Register/Bit Field/Programming Model	Value
Switch Master interface to smart standby mode	<a href="#">OTG_SYSCONFIG</a> [13:12] MIDDLEMODE	0x2
Switch Slave interface to smart-idle mode	<a href="#">OTG_SYSCONFIG</a> [4:3] SIDLEMODE	0x2
Set the clock to smart mode - clock is cut-off if no activity on the interface	<a href="#">OTG_SYSCONFIG</a> [0] AUTOIDLE	0x1

#### 23.12.5.3.3 Switching Between Host and Peripheral Mode

When used as a host/peripheral, the high-speed USB OTG controller must be programmed as follows:

**Table 23-1173. Power Management Configuration - Switching Between Host and Peripheral Mode**

Step	Register/Bit Field/Programming Model	Value
Switch Master interface to force standby mode	<a href="#">OTG_SYSCONFIG</a> [13:12] MIDDLEMODE	0x0
Switch Slave interface to smart-idle mode	<a href="#">OTG_SYSCONFIG</a> [4:3] SIDLEMODE	0x2
Set the clock to smart mode - clock is cut-off if no activity on the interface	<a href="#">OTG_SYSCONFIG</a> [0] AUTOIDLE	0x1

### 23.12.6 High-Speed USB OTG Controller Register Manual

This section presents only the TI-specific registers of the high-speed USB controller.

[Table 23-1174](#) lists the base address and address space for the high-speed USB controller.

**Table 23-1174. High-Speed USB OTG Controller Instance Summary**

Module Name	Base Address	Size
HSUSBOTG	0x4A0A B000	4KB
USBPHY	0x4A0A D000	4KB

**CAUTION**

The high-speed USB registers are limited to 32-bit data accesses; 16-bit and 8-bit accesses are not allowed and can corrupt register content.

**23.12.6.1 HSUSBOTG Registers****23.12.6.1.1 HSUSBOTG Register Summary****Table 23-1175. HSUSBOTG Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
<a href="#">OTG_REVISION</a>	R	32	0x0000 0400	0x4A0A B400
<a href="#">OTG_SYSCONFIG</a>	RW	32	0x0000 0404	0x4A0A B404
<a href="#">OTG_SYSSTATUS</a>	R	32	0x0000 0408	0x4A0A B408
<a href="#">OTG_INTERFSEL</a>	RW	32	0x0000 040C	0x4A0A B40C
<a href="#">OTG_SIMENABLE</a>	RW	32	0x0000 0410	0x4A0A B410
<a href="#">OTG_FORCESTDBY</a>	RW	32	0x0000 0414	0x4A0A B414
<a href="#">OTG_BIGENDIAN</a>	RW	32	0x0000 0418	0x4A0A B418

**23.12.6.1.2 HSUSBOTG Register Description****Table 23-1176. OTG\_REVISION**

<b>Address Offset</b>	0x0000 0400		
<b>Physical Address</b>	See <a href="#">Table 23-1175</a>	<b>Instance</b>	HSUSBOTG
<b>Description</b>	OCP standard USB OTG HS core revision number		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	IP Revision	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 23-1177. Register Call Summary for Register OTG\_REVISION**

High-Speed USB OTG Controller

- [High-Speed USB Register Mapping Summary: \[0\]](#)

**Table 23-1178. OTG\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0404		
<b>Physical Address</b>	See <a href="#">Table 23-1175</a>	<b>Instance</b>	HSUSBOTG
<b>Description</b>	OCP standard configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MIDLEMODE		RESERVED						SIDLEMODE		ENABLEWAKEUP	SOFTRESET	AUTOIDLE			

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	reserved	R	0x00000
13:12	MIDLEMODE	Master interface power management control. Standby/wait control  0x0: Force standby mode. Mstandby asserted unconditionally 0x1: No standby mode. Mstandby never asserted. 0x2: Smart standby mode. Mstandby asserted when no more activity on the USB master.	RW	0x2
11:5	RESERVED	reserved	R	0x00
4:3	SIDLEMODE	Slave interface power management control. Req/ack control  0x0: Force Idle mode. Sidleack asserted after Midlereq assertion 0x1: No idle mode. Sidleack never asserted. 0x2: SmartIdle mode. Sidleack asserted after Midlereq assertion when no more activity on the USB. Swakeup is not asserted in this mode. 0x3: SmartIdle - Wakeup mode. Sidleack asserted after Midlereq assertion when no more activity on the USB. Swakeup is asserted in this mode if ENABLEWAKEUP is set.	RW	0x2
2	ENABLEWAKEUP	Enable wakeup capability. 0x0: Wakeup disabled 0x1: Wakeup enabled	RW	0
1	SOFTRESET	Software reset bit 0x1: Starts softreset sequence.	RW	0
0	AUTOIDLE	Autoidle bit 0x0: Clock always running 0x1: When no activity on OCP, clock is cut off.	RW	1

**Table 23-1179. Register Call Summary for Register OTG\_SYSCONFIG**

High-Speed USB OTG Controller

- [Software Reset: \[0\]](#)
- [System Power Management: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\]](#)
- [Local Power Management: \[19\]](#)
- [High-Speed USB OTG Controller Global Initialization: \[20\] \[21\] \[22\] \[23\]](#)
- [No Application Mode: \[24\] \[25\] \[26\]](#)
- [Host or Peripheral Mode: \[27\] \[28\] \[29\]](#)
- [Switching Between Host and Peripheral Mode: \[30\] \[31\] \[32\]](#)
- [High-Speed USB Register Mapping Summary: \[33\]](#)

**Table 23-1180. OTG\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0408	<b>Instance</b>	HSUSBOTG
<b>Physical Address</b>	See <a href="#">Table 23-1175</a>		
<b>Description</b>	OCP standard status		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	RESETDONE														

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	reserved	R	0x0000 0000
0	RESETDONE	Reset done Read 0x0: Reset is ongoing Read 0x1: Reset is finished.	R	1

**Table 23-1181. Register Call Summary for Register OTG\_SYSSTATUS**

High-Speed USB OTG Controller

- [Software Reset: \[0\]](#)
- [High-Speed USB OTG Controller Global Initialization: \[1\]](#)
- [High-Speed USB Register Mapping Summary: \[2\]](#)

**Table 23-1182. OTG\_INTERFSEL**

<b>Address Offset</b>	0x0000 040C	<b>Instance</b>	HSUSBOTG
<b>Physical Address</b>	See <a href="#">Table 23-1175</a>		
<b>Description</b>	USB OTG HS interface selection. The interface selection has to be done before the PHY is activated and is not allowed to change when the PHY clock is already running.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	PHYSEL														

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	reserved	R	0x0000 0000
2:0	PHYSEL	PHY interface selection 0x0: PHY interface is 8-bit, UTMI+ level 3 0x1: PHY interface is 12-pin, 8-bit SDR ULPI 0x2: PHY interface is 8-pin, 4-bit DDR ULPI (Non-functional mode in current implementation. Do not use.)	RW	0x1

**Table 23-1183. Register Call Summary for Register OTG\_INTERFSEL**

High-Speed USB OTG Controller

- [High-Speed USB OTG Controller Environment: \[0\] \[1\]](#)
- [High-Speed USB OTG Controller Interface Modes: \[2\]](#)
- [High-Speed USB Register Mapping Summary: \[3\]](#)

**Table 23-1184. OTG\_SIMENABLE**

<b>Address Offset</b>	0x0000 0410	<b>Instance</b>	HSUSBOTG
<b>Physical Address</b>	See <a href="#">Table 23-1175</a>		
<b>Description</b>	Enable simulation acceleration features. WARNING: For simulations only, since those features have an impact on USB protocol.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												TM1			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	reserved	R	0x0000 0000
0	TM1	Test Mode 1 enabling (timer shortcuts)	RW	0

**Table 23-1185. Register Call Summary for Register OTG\_SIMENABLE**

High-Speed USB OTG Controller

- [High-Speed USB Register Mapping Summary: \[0\]](#)

**Table 23-1186. OTG\_FORCESTDBY**

<b>Address Offset</b>	0x0000 0414	<b>Instance</b>	HSUSBOTG
<b>Physical Address</b>	See <a href="#">Table 23-1175</a>		
<b>Description</b>	Enabling MSTANDBY in FORCESTANDBY mode. Programming this register will impact SmartStandby functionality.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												ENABLEFORCE			

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	ENABLEFORCE	Enabling MSTANDBY to go high	RW	0

**Table 23-1187. Register Call Summary for Register OTG\_FORCESTDBY**

High-Speed USB OTG Controller

- [High-Speed USB Register Mapping Summary: \[0\]](#)

**Table 23-1188. OTG\_BIGENDIAN**

<b>Address Offset</b>	0x0000 0418		
<b>Physical Address</b>	See <a href="#">Table 23-1175</a>	<b>Instance</b>	HSUSBOTG
<b>Description</b>	Enable BIG ENDIANESS for OCP MASTER		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															BIG_ENDIAN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	BIG_ENDIAN	Enable BIG ENDIAN in OCP MASTER	RW	0

**Table 23-1189. Register Call Summary for Register OTG\_BIGENDIAN**

High-Speed USB OTG Controller

- [USBOTGHS Support for Big Endian: \[0\]](#)
- [High-Speed USB Register Mapping Summary: \[1\]](#)

## 23.12.6.2 USBPHY Registers

### 23.12.6.2.1 USBPHY Register Summary

**Table 23-1190. USBPHY Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	USBPHY L3 Physical Address
<a href="#">USBPHY_TERMINATION_CONTROL</a>	RW	32	0x0000 0000	0x4A0A D000
<a href="#">USBPHY_RX_CALIB</a>	RW	32	0x0000 0004	0x4A0A D004
<a href="#">USBPHY_DLLHS_2</a>	RW	32	0x0000 0008	0x4A0A D008
<a href="#">USBPHY_RX_TEST_2</a>	RW	32	0x0000 000C	0x4A0A D00C
<a href="#">USBPHY_TX_TEST_CHRG_DET</a>	RW	32	0x0000 0010	0x4A0A D010
<a href="#">USBPHY_CHRG_DET</a>	RW	32	0x0000 0014	0x4A0A D014
<a href="#">USBPHY_PWR_CNTL</a>	RW	32	0x0000 0018	0x4A0A D018
<a href="#">USBPHY_UTMI_INTERFACE_CNTL_1</a>	RW	32	0x0000 001C	0x4A0A D01C
<a href="#">USBPHY_UTMI_INTERFACE_CNTL_2</a>	RW	32	0x0000 0020	0x4A0A D020
<a href="#">USBPHY_BIST</a>	RW	32	0x0000 0024	0x4A0A D024
<a href="#">USBPHY_BIST_CRC</a>	RW	32	0x0000 0028	0x4A0A D028
<a href="#">USBPHY_CDR_BIST2</a>	RW	32	0x0000 002C	0x4A0A D02C

**Table 23-1190. USBPHY Registers Mapping Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	USBPHY L3 Physical Address
USBPHY_GPIO	RW	32	0x0000 0030	0x4A0A D030
USBPHY_DLLHS	RW	32	0x0000 0034	0x4A0A D034
USBPHY_USB2PHYCM_TRIM	RW	32	0x0000 0038	0x4A0A D038
USBPHY_USB2PHYCM_CONFIG	RW	32	0x0000 003C	0x4A0A D03C
USBPHY_USBOTG	RW	32	0x0000 0040	0x4A0A D040
USBPHY_AD_INTERFACE_REG1	RW	32	0x0000 0044	0x4A0A D044
USBPHY_AD_INTERFACE_REG2	RW	32	0x0000 0048	0x4A0A D048
USBPHY_AD_INTERFACE_REG3	RW	32	0x0000 004C	0x4A0A D04C
USBPHY_ANA_CONFIG1	RW	32	0x0000 0050	0x4A0A D050
USBPHY_ANA_CONFIG2	RW	32	0x0000 0054	0x4A0A D054

**23.12.6.2.2 USBPHY Register Description**

**Table 23-1191. USBPHY\_TERMINATION\_CONTROL**

<b>Address Offset</b>	0x0000 0000	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	Contains bits related to control of terminations in USBPHY		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		ALWAYS_UPDATE	RTERM_CAL_DONE	FS_CODE_SEL				RESERVED	USE_RTERM_RMX_REG	RTERM_RMX						HS_CODE_SEL		RTERM_COMP_OUT	RESTART_RTERM_CAL	DISABLE_TEMP_TRACK	USE_RTERM_CAL_REG	RTERM_CAL									

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	reserved	RW	0x0
29	ALWAYS_UPDATE	When set to 1, the calibration code is updated immediately after a code computation without waiting for idle periods.	RW	0
28	RTERM_CAL_DONE	Rterm calibration is done. First time cal is done this bit gets set and gets reset at a restart cal. Read value is valid only if VDDLDO is on.	R	0
27:24	FS_CODE_SEL	FS Code selection control	RW	0x1
23:22	RESERVED	reserved	RW	0x0
21	USE_RTERM_RMX_REG	Override termination resistor trim code with RTERM_RMX from this register	RW	0
20:14	RTERM_RMX	When read, this field returns the current Termination resistor trim code. Read value is valid only if VDDLDO is on. The value written to this field is used as Termination resistor trim code if bit 21 is set to 1	RW	0x00
13:11	HS_CODE_SEL	HS Code selection control	RW	0x1
10	RTERM_COMP_OUT	Master loop comparator output. Read value is valid only if VDDLDO is on.	R	0

Bits	Field Name	Description	Type	Reset
9	RESTART_RTERM_CAL	Restart the rterm calibration. the calibration restarts on any toggle 0->1 or 1->0 on this bit.	RW	0
8	DISABLE_TEMP_TRACK	Disables the temperature tracking function of the termination calibration	RW	0
7	USE_RTERM_CAL_REG	When 1, the rterm cal code is overridden by values in RTERM_CAL	RW	0
6:0	RTERM_CAL	When read this field returns the current rterm calibration code. Read value is valid only if VDDLDO is on. The value written to this field is used as rterm calibration code if the bit USE_RTERM_CAL_REG is 1.	RW	0x00

**Table 23-1192. Register Call Summary for Register USBPHY\_TERMINATION\_CONTROL**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1193. USBPHY\_RX\_CALIB**

<b>Address Offset</b>	0x0000 0004	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	Contains bits related to RX calibration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESTART_HSRX_CAL	USE_HS_OFF_REG	HS_OFF_CODE						HSRX_COMP_OUT	HSRX_CAL_DONE	USE_SQ_OFF_DAC1	SQ_OFF_CODE_DAC1						USE_SQ_OFF_DAC2	SQ_OFF_CODE_DAC2						USE_SQ_OFF_DAC3	SQ_OFF_CODE_DAC3						SQ_COMP_OUT	SQ_CAL_DONE	RESTART_SQ_CAL

Bits	Field Name	Description	Type	Reset
31	RESTART_HSRX_CAL	Restart the HSRX calibration state machine when this bit goes from 0 to 1.	RW	0
30	USE_HS_OFF_REG	Override HS offset correction with HS_OFF_CODE when set to 1	RW	0
29:24	HS_OFF_CODE	HS offset code, this code is forced when bit 30 is 1. Code is updated from calibration logic when bit 30 = 0.	RW	0x00
23	HSRX_COMP_OUT	The output of the HSRX comparator. Read value is valid only if VDDLDO is on.	R	0
22	HSRX_CAL_DONE	Signal that indicates that the HSRX calibration is done. This gets reset at every restart. Read value is valid only if VDDLDO is on.	R	0
21	USE_SQ_OFF_DAC1	Override Squelch offset DAC1 code when '1'	RW	0
20:15	SQ_OFF_CODE_DAC1	When read returns current Sq offset code for DAC1, if VDDLDO is on. When written this is used as Sq offset code for DAC1 when USE_SQ_OFF_DAC1 = 1	RW	0x00
14	USE_SQ_OFF_DAC2	Override Squelch offset DAC2 code when '1'	RW	0
13:9	SQ_OFF_CODE_DAC2	When read returns current Sq offset code for DAC2, if VDDLDO is on. When written this is used as Sq offset code for DAC2 when USE_SQ_OFF_DAC2 = 1	RW	0x00
8	USE_SQ_OFF_DAC3	Override Squelch offset DAC3 code when 1	RW	0



Bits	Field Name	Description	Type	Reset
7:3	SQ_OFF_CODE_DAC3	When read returns current Sq offset code for DAC3, if VDDLDO is on. When written this is used as Sq offset code for DAC3 when USE_SQ_OFF_DAC3 = 1	RW	0x00
2	SQ_COMP_OUT	Sq comp output. Read value is valid only if VDDLDO is on.	R	0
1	SQ_CAL_DONE	Sq calibration is done when this bit = 1. See RESTART_SQ_CAL for more description. Read value is valid only if VDDLDO is on.	R	0
0	RESTART_SQ_CAL	The squelch calibration continuously goes through restart cycles when this bit is 1.	RW	0

**Table 23-1194. Register Call Summary for Register USBPHY\_RX\_CALIB**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1195. USBPHY\_DLLHS\_2**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	Second DLLHS control register. Bits 4:0 are unrelated to the DLLHS and are linestate filter settings.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLLHS_CNTRL_LDO								DLLHS_STATUS_LDO								RESERVED								LINESTATE_DEBOUNCE_EN		LINESTATE_DEBOUNCE_CNTL					

Bits	Field Name	Description	Type	Reset
31:24	DLLHS_CNTRL_LDO		RW	0x00
23:16	DLLHS_STATUS_LDO		R	0x00
15:5	RESERVED	reserved	RW	0x000
4	LINESTATE_DEBOUNCE_EN	Enables the linestate debounce filter	RW	1
3:0	LINESTATE_DEBOUNCE_CNTL	Used for control of the linestate debounce filter when going from synchronous to async linestate.	RW	0xF

**Table 23-1196. Register Call Summary for Register USBPHY\_DLLHS\_2**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1197. USBPHY\_RX\_TEST\_2**

<b>Address Offset</b>	0x0000 000C	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	Second receiver test register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSOSREVERSAL	HSOSBITINVERSION	PHYCLKOUTINVERSION	RXPIDERR	USEINTDATAOUT	INTDATAOUTREG											RESERVED				CDR_TESTOUT											

Bits	Field Name	Description	Type	Reset
31	HSOSREVERSAL	Swaps the dataout from HSOS	RW	0
30	HSOSBITINVERSION	Inverts the HSOS bits	RW	0
29	PHYCLKOUTINVERSION	This inverts the phase for the PHYCLKOUT	RW	0
28	RXPIDERR	Flags if the RX data packet has PID error. NOT IMPLEMENTED.	R	0
27	USEINTDATAOUT	This will bypass the analog and will send data packet to controller incase of receiver (Faking the receive data). data used will be INTDATAOUTREG	RW	0
26:11	INTDATAOUTREG	This register will be loaded through OCP and this data will be given to the controller if USEINTDATAOUT is set to 1	RW	0x0000
10:8	RESERVED	reserved	RW	0x0
7:0	CDR_TESTOUT	CDR debug bits. Read value is valid only if VDDLDO is on.	R	0x00

**Table 23-1198. Register Call Summary for Register USBPHY\_RX\_TEST\_2**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1199. USBPHY\_TX\_TEST\_CHRG\_DET**

<b>Address Offset</b>	0x0000 0010	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	TX test register and also charger detect register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSYNCERR	UTMIDATATX														RESERVED	TXPIDERR	USE_CHGDET_DPDMWSW	CHGDET_DPSW0EN	CHGDET_DPSW1EN	CHGDET_DMSW0EN	CHGDET_DMSW1EN	RESERVED	RDPPDCHGDETEN	RDMPDCHGDETEN	RDPPUCHGDETEN	RDMPUCHGDETEN	USE_CHG_DET_PU_REG	USE_CHG_DET_PD_REG	RESERVED		

Bits	Field Name	Description	Type	Reset
31	TXSYNCERR	Sync error on TX data. NOT IMPLEMENTED.	R	0
30:15	UTMIDATATX	Stores Last 2 byte of transmit data coming from the controller. NOT IMPLEMENTED .	R	0x0000
14	RESERVED	reserved	RW	0
13	TXPIDERR	Flags if the TX packet has PID error. NOT IMPLEMENTED.	R	0
12	USE_CHGDET_DPDMSW	Use bits 11:8 as override bits	RW	0
11	CHGDET_DPSW0EN	Overrides the same named A/D interface signal for the charger detect block	RW	0
10	CHGDET_DPSW1EN	Overrides the same named A/D interface signal for the charger detect block. Read value is valid only if VCHGLDO is on.	RW	0
9	CHGDET_DMSW0EN	Overrides the same named A/D interface signal for the charger detect block. Read value is valid only if VCHGLDO is on.	RW	0
8	CHGDET_DMSW1EN	Overrides the same named A/D interface signal for the charger detect block..Read value is valid only if VCHGLDO is on.	RW	0
7	RESERVED	reserved	RW	0
6	RDPPDCHGDETEN	When set to 1 connects a 15K (+/- 30%) pulldown resistor on DP. Read value is valid only if VCHGLDO is on.	RW	0
5	RDMPDCHGDETEN	When set to 1 connects a 15K (+/- 30%) pulldown resistor on DM. Read value is valid only if VCHGLDO is on.	RW	0
4	RDPPUCHGDETEN	When set to 1 connects a 150K (+/- 30%) pullup resistor on DP. Read value is valid only if VCHGLDO is on.	RW	0
3	RDMPUCHGDETEN	When set to 1 connects a 150K (+/- 30%) pullup resistor on DM. Read value is valid only if VCHGLDO is on.	RW	0
2	USE_CHG_DET_PU_REG	Use bits 4:3 from this register	RW	0
1	USE_CHG_DET_PD_REG	Use bits 6:5 from this register.	RW	0
0	RESERVED	reserved	RW	0

**Table 23-1200. Register Call Summary for Register USBPHY\_TX\_TEST\_CHRG\_DET**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1201. USBPHY\_CHRG\_DET**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	This is the charger detect register. This register is not used in the dead battery case.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	USE_CHG_DET_REG	DIS_CHG_DET	SRC_ON_DM	SINK_ON_DP	CHG_DET_EXT_CTL	RESTART_CHG_DET	CHG_DET_DONE	CHG_DETECTED	DATA_DET	RESERVED	CHG_ISINK_EN	CHG_VSRC_EN	COMP_DP	COMP_DM	CHG_DET_OSC_CNTRL	CHG_DET_TIMER						RESERVED	CHG_DET_ICTRL	CHG_DET_VCTRL	FOR_CE						

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	reserved	RW	0x0
29	USE_CHG_DET_REG	Use bits 28:24 and 18:17 from this register	RW	0
28	DIS_CHG_DET	When read, returns current value of charger detect input. When USE_CHG_DET_REG = 1, the value written to this field overrides the corresponding charger detect input.	RW	0
27	SRC_ON_DM	When read, returns current value of charger detect input. When USE_CHG_DET_REG = 1, the value written to this field overrides the corresponding charger detect input.	RW	0
26	SINK_ON_DP	When read, returns current value of charger detect input. When USE_CHG_DET_REG = 1, the value written to this field overrides the corresponding charger detect input.	RW	0
25	CHG_DET_EXT_CTL	When read, returns current value of charger detect input. When USE_CHG_DET_REG = 1, the value written to this field overrides the corresponding charger detect input.	RW	0
24	RESTART_CHG_DET	Restart the charger detection protocol when this goes from 0 to 1	RW	0
23	CHG_DET_DONE	Charger detect protocol has completed	R	0
22	CHG_DETECTED	Same signal as CE pin	R	0
21	DATA_DET	Output of the data det comparator	R	0
20:19	RESERVED	reserved	RW	0x0
18	CHG_ISINK_EN	When read, returns current value of charger detect input. When USE_CHG_DET_REG = 1, the value written to this field overrides the corresponding charger detect input.	RW	0
17	CHG_VSRC_EN	When read, returns current value of charger detect input. When USE_CHG_DET_REG = 1, the value written to this field overrides the corresponding charger detect input.	RW	0
16	COMP_DP	Comparator on the DP line value	R	0
15	COMP_DM	Comparator on the DM line value	R	0
14:13	CHG_DET_OSC_CNTRL	Charger detect osc control	RW	0x0
12:7	CHG_DET_TIMER	Charger detect timer control.	RW	0x00
6:5	RESERVED	reserved	RW	0x0
4:3	CHG_DET_ICTRL	Charger detect current control	RW	0x0
2:1	CHG_DET_VCTRL	Charger detect voltage buffer control	RW	0x0
0	FOR_CE	Force CE = 1 when this bit is set	RW	0

**Table 23-1202. Register Call Summary for Register USBPHY\_CHRG\_DET**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1203. USBPHY\_PWR\_CNTRL**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	Includes all the power control bits		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESETDONETCLK	RESET_DONE_VMAIN	VMAIN_GLOBAL_RESET_DONE	RESETDONEMCLK	RESETDONE_CHGDET	LDOPWRCOUNTER												FORCEPLLSLOWCLK	FORCELDOON	FORCEPLLON	RESERVED	PLLLOCK	USEPLLLOCK	USE_DATAPOLARITYN_REG	DATAPOLARITYN	USE_PD_REG	PD	RESERVED				

Bits	Field Name	Description	Type	Reset
31	RESETDONETCLK	Goes high when the RESET is synchronized to TCLK	R	0
30	RESET_DONE_VMAIN		R	0
29	VMAIN_GLOBAL_RESET_DONE		R	0
28	RESETDONEMCLK	Goes high when the RESET is synchronized to MCLK	R	0
27	RESETDONE_CHGDET	Goes high when the RESET is synchronized to charger detect oscillator clock domain	R	0
26:12	LDOPWRCOUNTER	This is the value of the counter used for LDO power up. RESET to default.	RW	0x0400
11	FORCEPLLSLOWCLK	Forces the PLL to the slow clk mode	RW	0
10	FORCELDOON	Forces the LDO to be ON.	RW	0
9	FORCEPLLON	Forces the PLL to be ON.	RW	0
8:7	RESERVED	reserved	RW	0x0
6	PLLLOCK	Lock signal from the PLL	R	0
5	USEPLLLOCK	This signal is used to indicate to the Phy, not to do any clock related activity until PLLLOCK = 1. This is not the default option. 0 - do not use PLLLOCK. 1 - use PLLLOCK as a clock gate.	RW	0
4	USE_DATAPOLARITYN_REG	1 - use bit 3 as override for the DATAPOLARITYN signal.	RW	0
3	DATAPOLARITYN	Override value of datapolarityn	RW	0
2	USE_PD_REG	Use bit 1 from this register as PD override when set to 1	RW	0
1	PD	Override value for PD	RW	0
0	RESERVED	reserved	RW	0

**Table 23-1204. Register Call Summary for Register USBPHY\_PWR\_CNTL**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1205. USBPHY\_UTMI\_INTERFACE\_CNTL\_1**

Address Offset	0x0000 001C		
Physical Address	See <a href="#">Table 23-1190</a>	Instance	USBPHY
Description	register to override UTMI interface control pins.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USEUTMIDATAREG	UTMIDATAIN															RESERVED	USEDATABUSREG	DATABUS16OR8	USEOPMODEREG	OPMODE	OVERRIDESUSRESET	SUSPENDM	UTMIRESET	OVERRIDEXCVRSEL	XCVRSEL	USETXVALIDREG	TXVALID	TXVALIDH			

Bits	Field Name	Description	Type	Reset
31	USEUTMIDATAREG	Use datain from UTMI interface register	RW	0
30:15	UTMIDATAIN	Override value for the UTMIDATAIN	RW	0x0000
14	RESERVED	reserved	RW	0
13	USEDATABUSREG	When set to 1 use bit 12 from register instead of interface	RW	0
12	DATABUS16OR8	Override value for UTMI signal DATABUS16OR8	RW	0
11	USEOPMODEREG	When set to 1 use bits 10:9 from register instead of interface	RW	0
10:9	OPMODE	Override value for UTMI signal OPMODE[1:0]	RW	0x0
8	OVERRIDESUSRESET	Override the suspend and reset values. Use bits 6 and 7	RW	0
7	SUSPENDM	Override value for UTMI signal SUSPENDM	RW	0
6	UTMIRESET	Override value for UTMI signal UTMIRESET	RW	0
5	OVERRIDEXCVRSEL	When set to 1 use bits 4:3 from register instead of interface	RW	0
4:3	XCVRSEL	Override value for UTMI signal XCVRSEL[1:0]	RW	0x0
2	USETXVALIDREG	When set to 1 use bits 1:0 from register instead of interface	RW	0
1	TXVALID	Override value for UTMI signal TXVALID	RW	0
0	TXVALIDH	Override value for UTMI signal TXVALIDH	RW	0

**Table 23-1206. Register Call Summary for Register USBPHY\_UTMI\_INTERFACE\_CNTL\_1**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1207. USBPHY\_UTMI\_INTERFACE\_CNTL\_2**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	UTMI interface override and observe register 2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXRCV	RXDP	RXDM	HOSTDISCONNECT	LINESTATE	RXVALID	RXVALIDH	RXACTIVE	RXERROR	TXREADY	UTMIRESETDONE	USEBITSTUFFREG	TXBITSTUFFENABLE	TXBITSTUFFENABLEH	USETERMCONTROLREG	TERMSEL	DPPULLDOWN	DMPULLDOWN	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	USEREGSERIALMODE	TXSE0	TXDAT	FSLSSERIALMODE	TXENABLEN	RESERVED	RESERVED	SIG_BYPASS_SUSPENDMPULSE_INCR

Bits	Field Name	Description	Type	Reset
31	RXRCV	Read for UTMI signal. Read value is valid only if VDDLDO is on. Read value is valid only if VDDLDO is on.	R	0
30	RXDP	Read for UTMI signal. Read value is valid only if VDDLDO is on.	R	0
29	RXDM	Read for UTMI signal. Read value is valid only if VDDLDO is on.	R	0
28	HOSTDISCONNECT	Read for UTMI signal. Read value is valid only if VDDLDO is on.	R	0
27:26	LINESTATE	Read for UTMI signal. Read value is valid only if VDDLDO is on.	R	0x0
25	RXVALID	Read for UTMI signal. Read value is valid only if VDDLDO is on.	R	0
24	RXVALIDH	Read for UTMI signal. Read value is valid only if VDDLDO is on.	R	0
23	RXACTIVE	Read for UTMI signal. Read value is valid only if VDDLDO is on.	R	0
22	RXERROR	Read for UTMI signal. Read value is valid only if VDDLDO is on.	R	0
21	TXREADY	Read for UTMI signal. Read value is valid only if VDDLDO is on.	R	0
20	UTMIRESETDONE	Read for UTMIRESETDONE signal	R	0
19	USEBITSTUFFREG	When set to 1 use bits 18-17 from register instead of interface	RW	0
18	TXBITSTUFFENABLE	Override value for signal TXBITSTUFFENABLE	RW	0
17	TXBITSTUFFENABLEH	Override value for pin TXBITSTUFFENABLE	RW	0
16	USETERMCONTROLREG	When set to 1, bits 15:13 from register are used instead of interface	RW	0
15	TERMSEL	Override value for signal TERMSEL	RW	0
14	DPPULLDOWN	Override value for signal DPPULLDOWN	RW	0
13	DMPULLDOWN	Override value for signal DMPULLDOWN	RW	0
12:10	RESERVED	reserved	RW	0x0
9	USEREGSERIALMODE	When set to 1 use bits 8:5 from register instead of interface	RW	0
8	TXSE0	Override value for signal TXSE0	RW	0
7	TXDAT	Override value for signal TXDAT	RW	0
6	FSLSSERIALMODE	Override value for signal FSLSSERIALMODE	RW	0
5	TXENABLEN	Override value for signal TXENABLEN	RW	0
4:1	RESERVED	reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
0	SIG_BYPASS_SUSPENDMPULSE_INCR		RW	0

**Table 23-1208. Register Call Summary for Register USBPHY\_UTMI\_INTERFACE\_CNTL\_2**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1209. USBPHY\_BIST**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	Contains bits related to the built in self test of the phy		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIST_START	REDUCED_SWING	BIST_CRC_CALC_EN	BIST_PKT_LENGTH					LOOPBACK_EN	BIST_OP_PHASE_SEL			SWEEP_EN	SWEEP_MODE			BIST_PASS	BIST_BUSY	RESERVED		OP_CODE	RX_TEST_MODE	RESERVED	INTER_PKT_DELAY_TEST	HS_ALL_ONES_TEST	USE_BIST_TX_PHASES						

Bits	Field Name	Description	Type	Reset
31	BIST_START	When set to 1 the BIST mode is started.	RW	0
30	REDUCED_SWING	When 1 the TX swing is reduced in BIST mode	RW	0
29	BIST_CRC_CALC_EN	Enables CRC calculation during BIST when set to 1	RW	0
28:20	BIST_PKT_LENGTH	Address for which BIST to select	RW	0x000
19	LOOPBACK_EN	Enables the loopback mode	RW	0
18:16	BIST_OP_PHASE_SEL	Selects which phase to use for data transmission during BIST	RW	0x0
15	SWEEP_EN	Enables freq sweep on CDR	RW	0
14:12	SWEEP_MODE	Selects the freq sweep mode.	RW	0x0
11	BIST_PASS	Indicates that the BIST has passed. Read value is valid only if VDDLDO is on.	R	0
10	BIST_BUSY	Indicates that BIST is running. Read value is valid only if VDDLDO is on.	R	0
9:7	RESERVED	reserved	RW	0x0
6:5	OP_CODE		RW	0x0
4	RX_TEST_MODE		RW	0
3	RESERVED	reserved	RW	0
2	INTER_PKT_DELAY_TEST		RW	0
1	HS_ALL_ONES_TEST		RW	0
0	USE_BIST_TX_PHASES	When set to 1 bits 18:16 are activated for choosing the transmitting phase.	RW	0

**Table 23-1210. Register Call Summary for Register USBPHY\_BIST**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)



**Table 23-1211. USBPHY\_BIST\_CRC**

<b>Address Offset</b>	0x0000 0028		
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>	<b>Instance</b>	USBPHY
<b>Description</b>	CRC code for BIST test		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Bits	Field Name	Description	Type	Reset
31:0	BIST_CRC	The CRC value from the BIST.	RW	0x0000 0000

**Table 23-1212. Register Call Summary for Register USBPHY\_BIST\_CRC**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1213. USBPHY\_CDR\_BIST2**

<b>Address Offset</b>	0x0000 002C		
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>	<b>Instance</b>	USBPHY
<b>Description</b>	clock data recovery register and BIST register 2		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

CDR_EXE_EN	CDR_EXE_MODE	NUM_DECISIONS	CDR_CHOSEN_PHASE	FORCE_CDR_PHASE	DISABLE_CDR_FREQ_TRACK	CDR_CONFIGURE						FORCE_CDR_PHASE_EN	BIST_START_ADDR				BIST_END_ADDR			

Bits	Field Name	Description	Type	Reset
31	CDR_EXE_EN	CDR debug bits	RW	0
30:28	CDR_EXE_MODE	CDR debug bits	RW	0x0
27:25	NUM_DECISIONS	CDR debug bits	RW	0x0
24:12	RESERVED	reserved	R	0x000
11:6	BIST_START_ADDR		RW	0x00
5:0	BIST_END_ADDR		RW	0x00

**Table 23-1214. Register Call Summary for Register USBPHY\_CDR\_BIST2**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1215. USBPHY\_GPIO**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>	<b>Instance</b>	USBPHY
<b>Description</b>	GPIO mode configurations and reads		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USEGPIOMODEREG	GPIOMODE	DPGPIOGZ	DMGPIOGZ	DPGPIOA	DMGPIOA	DPGPIOY	DMGPIOY	1P8VCONFIG	GPIOCONFIG				DMGPIOIPD	DPGPIOIPD	RESERVED																

Bits	Field Name	Description	Type	Reset
31	USEGPIOMODEREG	When set to 1 use bits 31:24 from this register instead of primary inputs	RW	0
30	GPIOMODE	Overrides the corresponding primary input	RW	0
29	DPGPIOGZ	Overrides the corresponding primary input	RW	0
28	DMGPIOGZ	Overrides the corresponding primary input	RW	0
27	DPGPIOA	Overrides the corresponding primary input	RW	0
26	DMGPIOA	Overrides the corresponding primary input	RW	0
25	DPGPIOY	The GPIO Y output is stored here	R	0
24	DMGPIOY	The GPIO Y output is stored here	R	0
23	GPIO1P8VCONFIG	Overrides the corresponding primary input	RW	0
22:20	GPIOCONFIG	Used for configuring the GPIOs.	RW	0x0
19	DMGPIOIPD	GPIO mode DM pull-down enabled. Overrides the corresponding primary input	RW	0
18	DPGPIOIPD	GPIO mode DP pull-down enabled. Overrides the corresponding primary input.	RW	0
17:0	RESERVED	reserved	RW	0x00000

**Table 23-1216. Register Call Summary for Register USBPHY\_GPIO**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1217. USBPHY\_DLLHS**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>	<b>Instance</b>	USBPHY
<b>Description</b>	Bits for control and debug of the DLL inside the USBPHY		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
RESERVED								DLLHS_GENERATED_CODE								DLL_PHS0_8		DLL_FORCED_CODE								FORCE_DLL_CODE		DLL_RATE		DLL_FILTER		DLL_CDR_MODE		DLL_IDLE		DLL_FREEZE		RESERVED			
RESERVED								DLLHS_LOCK								DLL_SEL_CODE_PHS		DLL_LOCKCHK		DLL_SEL_COD								FORCE_DLL_CODE		DLL_RATE		DLL_FILTER		DLL_CDR_MODE		DLL_IDLE		DLL_FREEZE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	reserved	RW	0x0
28	DLLHS_LOCK	Read the AFE output by this name	R	0
27:22	DLLHS_GENERATED_CODE	Read the AFE output by this name. Read value is valid only if VDDLDO is on.	R	0x00
21	DLL_SEL_CODE_PHS	Connect to DLLHS_TEST_LDO[0] on AFE interface.	RW	0
20:19	DLL_LOCKCHK	Connect to DLLHS_TEST_LDO[2:1] on AFE interface.	RW	0x0
18:16	DLL_SEL_COD	Connect to DLLHS_TEST_LDO[5:3] on AFE interface.	RW	0x0
15	DLL_PHS0_8	Connect to DLLHS_TEST_LDO[6] on AFE interface.	RW	1
14:9	DLL_FORCED_CODE	Connect to the pin of this name on AFE interface.	RW	0x00
8	FORCE_DLL_CODE	Connect to DLLHS_TEST_LDO[11] on AFE interface.	RW	0
7:6	DLL_RATE	Connect to DLLHS_TEST_LDO[8:7] on AFE interface.	RW	0x0
5:4	DLL_FILTER	Connect to DLLHS_TEST_LDO[10:9] on AFE interface.	RW	0x0
3	DLL_CDR_MODE	Connect to the pin of this name on AFE interface.	RW	0
2	DLL_IDLE	Connect to DLLHS_TEST_LDO[12] on AFE interface.	RW	0
1	DLL_FREEZE	Connect to DLLHS_TEST_LDO[13] on AFE interface.	RW	0
0	RESERVED	reserved	RW	0

**Table 23-1218. Register Call Summary for Register USBPHY\_DLLHS**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1219. USBPHY\_USB2PHYCM\_TRIM**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	Contains trim bit overrides for the USBPHYCM		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USEBGTRIM								BGTRIM								USE_SW_TRIM		SWTRIM								USE_NWELLTRIM_REG		NWELLTRIM_CODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31	USEBGTRIM	When set to 1 bits 30:16 are used as the trim value for the USBPHYCM bandgap	RW	0
30:16	BGTRIM	Override value for the BGTRIM value	RW	0x0000
15	USE_SW_TRIM	Use bits 14:8 to override the switch cap trim value.	RW	0
14:8	SWTRIM	Override value for the switch cap trim value.	RW	0x00
7	USE_NWELLTRIM_REG	Override NWELL resistor trim using NWELLTRIM_CODE	RW	0
6:4	NWELLTRIM_CODE	NWELL resistor trim code.	RW	0x0
3:0	RESERVED	reserved	RW	0x0

**Table 23-1220. Register Call Summary for Register USBPHY\_USB2PHYCM\_TRIM**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1221. USBPHY\_USB2PHYCM\_CONFIG**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	Configuration and status register for the USBPHYCM and LDO		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONFIGURECM								CMSTATUS								LDOCONFIG								LDOSTATUS							

Bits	Field Name	Description	Type	Reset
31:24	CONFIGURECM	Connects to the CONFIGURECM pins.	RW	0x00
23:18	CMSTATUS	Reads the CMSTATUS bits.	R	0x00
17:2	LDOCONFIG	The LDOCONFIG bit settings.	RW	0x0000
1:0	LDOSTATUS	Reads the LDOSTATUS bits.	R	0x0

**Table 23-1222. Register Call Summary for Register USBPHY\_USB2PHYCM\_CONFIG**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1223. USBPHY\_USBOTG**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>			
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TESTOTGCONFIG								TESTOTGSTATUS								RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	TESTOTGCONFIG	Used to control the OTG module if used.	RW	0x0000
15:6	TESTOTGSTATUS	The OTG status bits (if OTG macro is used)	R	0x0000
5:0	RESERVED	reserved	R	0x00

**Table 23-1224. Register Call Summary for Register USBPHY\_USBOTG**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1225. USBPHY\_AD\_INTERFACE\_REG1**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	All bits (unless defined) are bypass bits for internal analog to digital interface pins with the same name. All the bits of this register, except the over-ride bits return a '0' on read, if VDDLDO is off.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USE_AD_DATA_REG	HS_TX_DATA	FS_TX_DATA	TEST_PRE_EN_CNTRL	SQ_PRE_EN	HS_TX_PRE_EN	HS_RX_PRE_EN	TEST_EN_CNTRL	HS_TX_EN	FS_RX_EN	RESERVED	SQ_EN	HS_RX_EN	TEST_HS_MODE	HS_HV_SW	HS_CHIRP	TEST_FS_MODE	FSTX_GZ	FSTX_PRE_EN	RESERVED	TEST_SQ_CAL_CONTROL	SQ_CAL_EN3	SQ_CAL_EN1	SQ_CAL_EN2	TEST_RTERM_CAL_CONTROL	RTERM_CAL_EN	DLL_RX_DATA	DISCON_DETECT	USE_LSHOST_REG	LSHOSTMODE	LSFS_RX_DATA	SQUELCH

Bits	Field Name	Description	Type	Reset
31	USE_AD_DATA_REG	Override for bits 30:29	RW	0
30	HS_TX_DATA		RW	0
29	FS_TX_DATA		RW	0
28	TEST_PRE_EN_CNTRL	Override for bits 27:25	RW	0
27	SQ_PRE_EN		RW	0
26	HS_TX_PRE_EN		RW	0
25	HS_RX_PRE_EN		RW	0
24	TEST_EN_CNTRL	Override for bits 23:19	RW	0
23	HS_TX_EN		RW	0
22	FS_RX_EN		RW	0
21	RESERVED		RW	0
20	SQ_EN		RW	0
19	HS_RX_EN		RW	0
18	TEST_HS_MODE	Override for bits 17:16	RW	0
17	HS_HV_SW		RW	0
16	HS_CHIRP		RW	0
15	TEST_FS_MODE	Override for bits 14:12	RW	0
14	FSTX_GZ		RW	0
13	FSTX_PRE_EN		RW	0
12	RESERVED		RW	0
11	TEST_SQ_CAL_CONTROL	Override for bits 10:8	RW	0
10	SQ_CAL_EN3		RW	0
9	SQ_CAL_EN1		RW	0
8	SQ_CAL_EN2		RW	0
7	TEST_RTERM_CAL_CONTROL	Override for bits 6	RW	0
6	RTERM_CAL_EN		RW	0

Bits	Field Name	Description	Type	Reset
5	DLL_RX_DATA		R	0
4	DISCON_DETECT		R	0
3	USE_LSHOST_REG	Use bit 2 for this reg	RW	0
2	LSHOSTMODE		RW	0
1	LSFS_RX_DATA		R	0
0	SQUELCH		R	0

**Table 23-1226. Register Call Summary for Register USBPHY\_AD\_INTERFACE\_REG1**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1227. USBPHY\_AD\_INTERFACE\_REG2**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	All bits (unless defined) are bypass bits for internal analog to digital interface pins with the same name. All the bits of this register, except the override bits return a '0' on read, if VDDLDO is off.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
USE_SUSP_DRV_REG	SUS_DRV_DP_DATA	SUS_DRV_DP_EN	SUS_DRV_DM_DATA	SUS_DRV_DM_EN	USE_DISCON_REG	DISCON_EN	DISCON_PRE_EN	RESERVED			SPARE_OUT_CORE			SERX_DP_CORE	SERX_DM_CORE	USE_HSRX_CAL_EN_REG	HSRX_CAL_EN	USE_RPU_RPD_REG	RPU_DP_SW1_EN_CORE	RPU_DP_SW2_EN_CORE	RPU_DM_SW1_EN_CORE	RPU_DM_SW2_EN_CORE	DP_PULLDOWN_EN_CORE	DM_PULLDOWN_EN_CORE	DP_DM_5V_SHORT									SPARE_IN_CORE		PORZ

Bits	Field Name	Description	Type	Reset
31	USE_SUSP_DRV_REG	Use bits 30:27 from this register as overrides	RW	0
30	SUS_DRV_DP_DATA		RW	0
29	SUS_DRV_DP_EN		RW	0
28	SUS_DRV_DM_DATA		RW	0
27	SUS_DRV_DM_EN		RW	0
26	USE_DISCON_REG	Use bits 25:24 from this register as override	RW	0
25	DISCON_EN		RW	0
24	DISCON_PRE_EN		RW	0
23	RESERVED	reserved	RW	0
22:18	SPARE_OUT_CORE		R	0x00
17	SERX_DP_CORE		R	0
16	SERX_DM_CORE		R	0
15	USE_HSRX_CAL_EN_REG	Use bit 14 from this register as override	RW	0
14	HSRX_CAL_EN		RW	0
13	USE_RPU_RPD_REG	Use override from bits 12:7	RW	0
12	RPU_DP_SW1_EN_CORE		RW	0
11	RPU_DP_SW2_EN_CORE		RW	0
10	RPU_DM_SW1_EN_CORE		RW	0
9	RPU_DM_SW2_EN_CORE		RW	0

Bits	Field Name	Description	Type	Reset
8	DP_PULLDOWN_EN_CORE		RW	0
7	DM_PULLDOWN_EN_CORE		RW	0
6	DP_DM_5V_SHORT		R	0
5:1	SPARE_IN_CORE		RW	0x00
0	PORZ	Read only bit - the PORZ generated from the digital registered on the A-D interface.	R	0

**Table 23-1228. Register Call Summary for Register USBPHY\_AD\_INTERFACE\_REG2**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1229. USBPHY\_AD\_INTERFACE\_REG3**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	USBPHY
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>		
<b>Description</b>	All bits (unless defined) are bypass bits for internal analog to digital interface pins with the same name. All the bits of this register, except the override bits return a '0' on read, if VDDLDO is off.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USE_HSOS_DATA_REG	HSOS_DATA							USE_FS_REG3	FSTX_MODE	FSTX_SE0	USE_HS_TERM_RES_REG	HS_TERM_RES	SPARE_IN_LDO					SPARE_OUT_LDO					USE_FARCORE_REG	FARCORE							

Bits	Field Name	Description	Type	Reset
31	USE_HSOS_DATA_REG	Use bits 30:23 in this register as bypass bits	RW	0
30:23	HSOS_DATA		RW	0x00
22	USE_FS_REG3	Use bits 21:20 as bypass bits	RW	0
21	FSTX_MODE		RW	0
20	FSTX_SE0		RW	0
19	USE_HS_TERM_RES_REG	Use bit 18 as override bit	RW	0
18	HS_TERM_RES		RW	0
17:10	SPARE_IN_LDO		RW	0x00
9:2	SPARE_OUT_LDO		R	0x00
1	USE_FARCORE_REG	Use bit 0 from this register as bypass	RW	0
0	FARCORE		RW	0

**Table 23-1230. Register Call Summary for Register USBPHY\_AD\_INTERFACE\_REG3**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1231. USBPHY\_ANA\_CONFIG1**

<b>Address Offset</b>	0x0000 0050		
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>	<b>Instance</b>	USBPHY
<b>Description</b>	Used to configure and debug the analog blocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ_CTRL_REG								FS_SLEW				HS_PRE_EMP_CNTRL		HSFSTX_TEST				PROTECT_TEST													

Bits	Field Name	Description	Type	Reset
31:17	SQ_CTRL_REG		RW	0x0000
16:14	FS_SLEW		RW	0x0
13:12	HS_PRE_EMP_CNTRL		RW	0x0
11:5	HSFSTX_TEST		RW	0x00
4:0	PROTECT_TEST		RW	0x00

**Table 23-1232. Register Call Summary for Register USBPHY\_ANA\_CONFIG1**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

**Table 23-1233. USBPHY\_ANA\_CONFIG2**

<b>Address Offset</b>	0x0000 0054		
<b>Physical Address</b>	See <a href="#">Table 23-1190</a>	<b>Instance</b>	USBPHY
<b>Description</b>	Used to configure and debug the analog blocks.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTERM_CAL_TEST		REF_GEN_TEST				FSRX_TEST				DISCON_TEST				HSRX_TEST				SERX_TEST		SERX_HYST_CNTRL		SQ_LPMODEZ									

Bits	Field Name	Description	Type	Reset
31:27	RTERM_CAL_TEST		RW	0x00
26:20	REF_GEN_TEST		RW	0x00
19:15	FSRX_TEST		RW	0x00
14:11	DISCON_TEST		RW	0x0
10:6	HSRX_TEST		RW	0x00
5:3	SERX_TEST		RW	0x0
2:1	SERX_HYST_CNTRL		RW	0x0



Bits	Field Name	Description	Type	Reset
0	SQ_LPMODEZ		RW	0

**Table 23-1234. Register Call Summary for Register USBPHY\_ANA\_CONFIG2**

High-Speed USB OTG Controller

- [USBPHY Register Summary: \[0\]](#)

## 23.13 Full-Speed USB Host Controller

This section describes the full-speed universal serial bus (FSUSB) host controller for the device

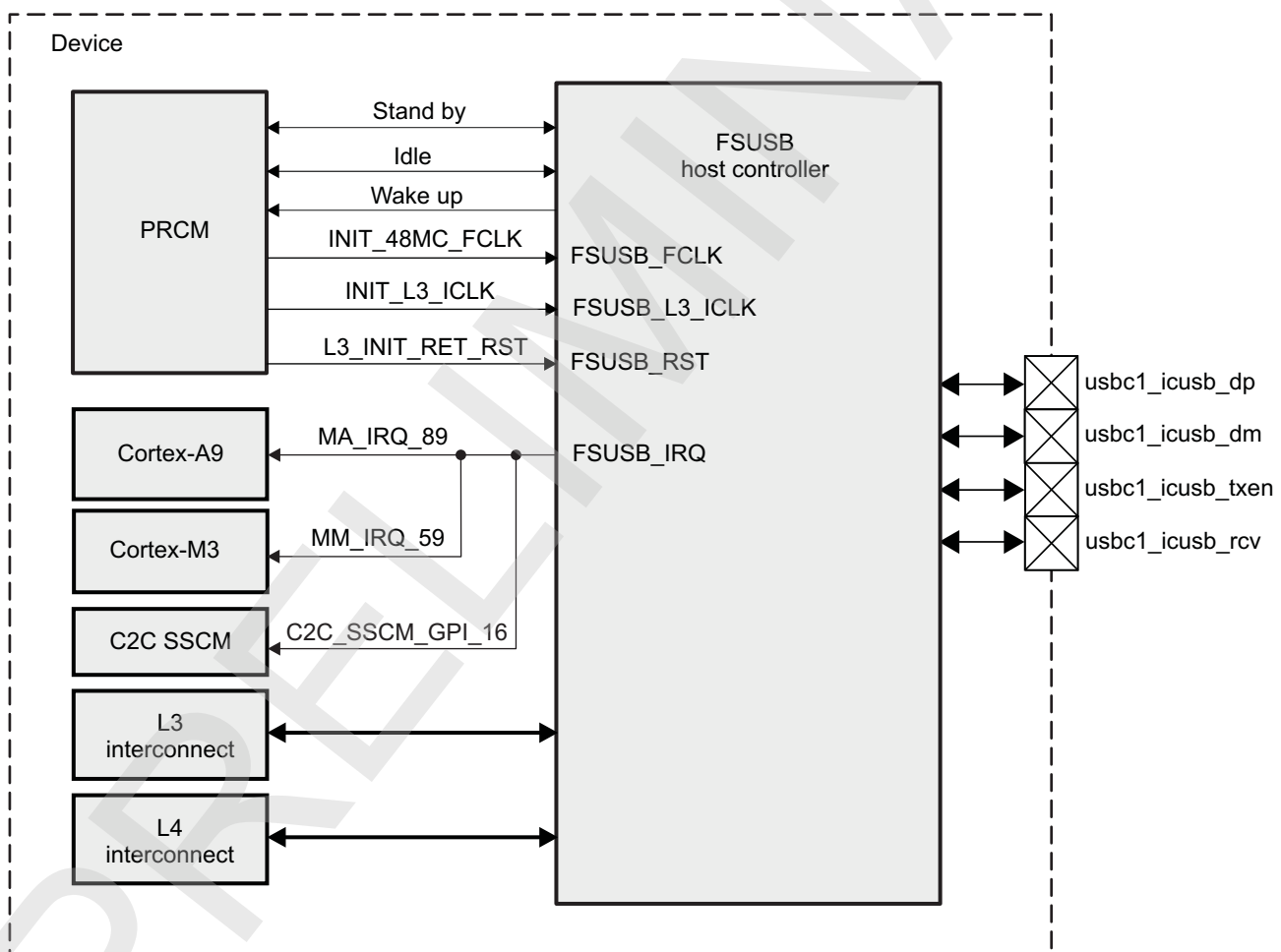
### 23.13.1 FSUSB Host Controller Overview

**NOTE:** Copyright ©2004, 2005, 2006, 2007, 2008 Synopsys, Inc. All rights reserved. Used with permission.

This module is a full-speed universal serial bus (FSUSB) host controller which complies with the USB 2.0 standard for full-speed functions. The FSUSB host controller supports a single universal serial bus (USB) host port with a USB interchip to interface with a multimedia subscriber-identity module (SIM) card, using 2-/4-pin bidirectional mode (host mode).

The FSUSB host controller has level 3 (L3) and level 4 (L4) interfaces with an independent functional clock.

**Figure 23-263. FSUSB Host Controller Overview**



fsusb-001

#### 23.13.1.1 Main Features

The main features of the FSUSB host controller include:

- USB host
  - Compatible with OHCI Rev 1.0

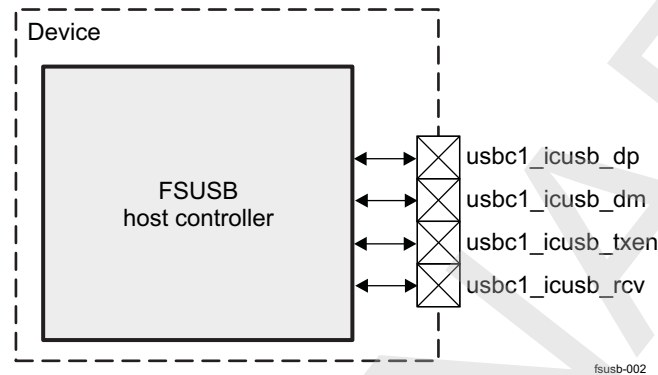
- USB 2.0 low-speed (1.5 Mbps) and full-speed (12 Mbps) operations
- Single port
- USB port signal pins interface:
  - Only supports the 2-/4-pin (bidirectional) mode
- Slave interface (L4) supporting:
  - 32-bit data bus width
  - 32-bit access
  - 32-bit address bus width
  - Wireless network portability (WNP)
  - Burst is not supported
- Master interface (L3) supporting:
  - 32-bit data bus width
  - 8-/32-bit access
  - 32-bit address bus width
  - Write nonposted (WRPN) interconnect transaction

### 23.13.2 FSUSB Host Controller Environment

This section describes the FSUSB host controller application fields from an environment point of view (external connections). It discusses FSUSB host controller connectivity options, lists all possible interfaces, and describes the protocol and data format used in each case.

Figure 23-264 shows the output pins of the FSUSB host controller.

**Figure 23-264. Overview of FSUSB Host Controller Environment**



#### 23.13.2.1 FSUSB Host Controller Functional Interfaces Description

Table 23-1235 and Table 23-1236 list the input/output (I/O) signals.

**Table 23-1235. Pin List for FS USB-Host (USB-Interchip Mode)**

Signal Name	I/O <sup>(1)</sup>	Description	Reset Value
usbc1_icusb_dp	I/O	Interchip FSUSB host controller D plus	Unknown
usbc1_icusb_dm	I/O	Interchip FSUSB host controller D minus	Unknown

<sup>(1)</sup> I = Input, O = Output

**Table 23-1236. 4-Pin Mode Delivered on Pin (for External Transceiver Connection)**

Signal Name	I/O <sup>(1)</sup>	Description	Reset Value
usbc1_icusb_dp	I/O	FSUSB host controller D plus	Unknown
usbc1_icusb_dm	I/O	FSUSB host controller D minus	Unknown
usbc1_icusb_txen	O	FSUSB host controller transmit enable	Unknown
usbc1_icusb_rcv	I	FSUSB host controller receive	Unknown

<sup>(1)</sup> I = Input, O = Output

### 23.13.3 FSUSB Host Controller Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 23-265 shows the integration of the FSUSB host controller in the device.

Figure 23-265. FSUSB Host Controller Integration

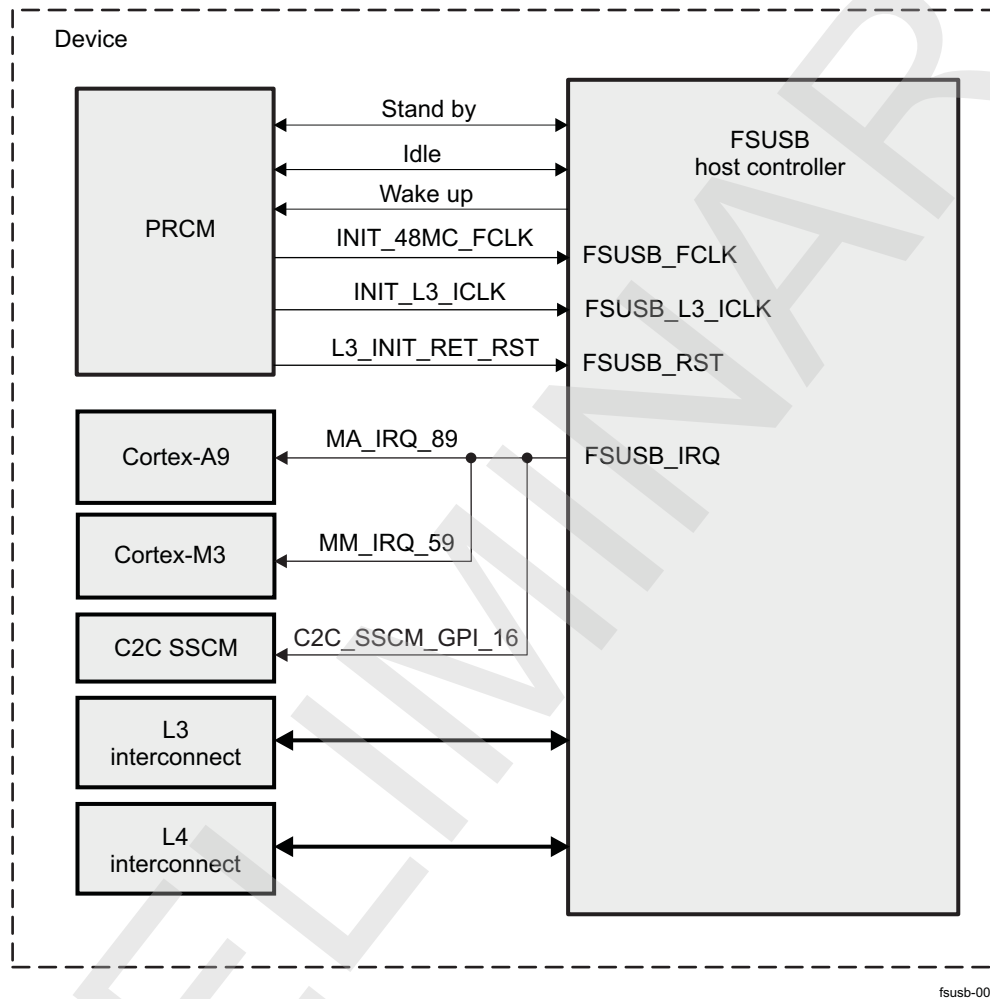


Table 23-1237 through Table 23-1239 summarize the integration of the module in the device.

Table 23-1237. Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
FSUSB host controller	PD_L3_INIT	L3 L4_CFG

Table 23-1238. Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
FSUSB host controller	FSUSB_FCLK	INIT_48MC_FCLK	PRCM	Functional clock (see Section 3.6.9, CD_L3_INIT Clock Domain in Chapter 3, Power, Reset, and Clock Management).

**Table 23-1238. Clocks and Resets (continued)**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
	FSUSB_L3_ICLK	INIT_L3_ICLK	PRCM	L3 interconnect clock for the L3 master port interface (see <a href="#">Section 3.6.9, CD_L3_INIT Clock Domain</a> in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> )
	FSUSB_L4_ICLK	INIT_L4_ICLK	PRCM	L4 interconnect-enable signal for the slave interface on L4 (see <a href="#">Section 3.6.3.2, CM Clock Source</a> in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> )
Resets				
FSUSB host controller	FSUSB_RST	L3_INIT_RET_RST	PRCM	This signal resets the FSUSB host controller module.

**Table 23-1239. Hardware Requests**

Interrupt Requests					
Module Instance		Source Signal Name	Destination Signal Name	Destination	Description
FSUSB controller	host	FSUSB_IRQ	MA_IRQ_89	Cortex-A9 INTC	Destination is Cortex-A9 interrupt controller (INTC)
		FSUSB_IRQ	MM_IRQ_59	Cortex-M3 INTC	Destination is Cortex-M3 INTC
		FSUSB_IRQ	C2C_SSCM_GPI_16	C2C INTC	Destination is C2C INTC

**NOTE:** For interrupt source information, see [Section 23.13.5.1.2, Interrupt Requests](#).

## 23.13.4 FSUSB Host Controller Functional Description

### 23.13.4.1 Resets

The different software reset functions of the FSUSB host controller are controlled by the following register bits:

- **HCOCPYSYS[1] SOFT\_RESET** bit: Writing 0x1 to this bit resets the entire module, except IDLE, STANDBY, and DISCONNECT logic. The bit value of 0x1 remains until the reset completes. When the software reset completes, the SOFT\_RESET bit is automatically set to 0x0.
- **HCCOMMANDSTATUS[0] HCR** bit: Writing 0x1 resets most of the open host controller interface (OHCI) registers of the FSUSB host controller. It is useful for error handling only. OHCI register accesses must not be attempted until a read of this register returns 0x0.

### 23.13.4.2 Power Management

#### 23.13.4.2.1 Features

The power management module has the following features:

- The FSUSB host controller has one instance of IDLE and standby interface.
- There are separate disconnect interfaces for master and slave ports.
- The PRCM module uses IDLE protocol to put the FSUSB host controller asynchronous into idle mode.
- The FSUSB host controller uses standby protocol for USB suspend event.
- Clock gating is done externally for IDLE and standby protocol.
- The IDLE and STANDBY interfaces are synchronous to the PRCM clock.

#### 23.13.4.2.2 IDLE Protocol

The IDLE protocol has the following features:

- There is one IDLE protocol for the FSUSB host controller.
- The **HCOCPYSYS[3:2] IDLE\_MODE** bit field controls the way USBFSHOST handles the IDLE protocol.
- The FSUSB host controller WakeupEnable mode is enabled by setting the **HCOCPYSYS[3:2] IDLE\_MODE** bit field to 0x11.
- Wakeup is generated asynchronously in the FSUSB host controller. To detect the USB remote wakeup, the FSUSB host controller should always be powered up (that is, the clocks can be off, but the power is needed for a wake-up detection to work).
- The FSUSB host controller interface asserts wakeup from suspend when the wake-up conditions occur (such as remote signaling on port, device connect/disconnect) and the FSUSB host controller is in idle mode. The FSUSB host controller interface deasserts the wake-up signal when the PRCM module has serviced the event (usually at completion of a resume sequence on the USB) and IDLE request is 0.
- The FSUSB host controller interface acknowledges and goes into IDLE only if the host controller is in suspend mode, or if force-idle mode is enabled.

Table 23-1240 lists the FSUSB host controller idle mode settings.

**Table 23-1240. FSUSB Host Controller Idle Mode Settings**

SIDLEMODE Value	Selected Mode	Description
0x0	Force-idle	The high-speed USB host controller unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations.
0x1	Not supported (no-idle)	
0x2	Smart-idle	The high-speed USB host controller acknowledges the IDLE request, basing its decision on its internal activity. Namely, the acknowledged signal is asserted only when all pending transactions, IRQs, or DMA requests are treated.
0x3	Smart-idle wakeup	The high-speed USB host controller enters smart mode with asynchronous wakeup enabled.

### 23.13.4.2.3 Standby Protocol

The Standby protocol is associated with FSUSB host controller suspend. Standby assertion is linked with Suspend, and Wakeup assertion is linked with remote wakeup. When the application wants to resume the USB, the power manager must initiate the wakeup using the Standby protocol.

- There is one Standby protocol for the FSUSB host controller.
- The `HCOCPSYS[5:4]` STANDBY\_MODE bit field controls the way USBFSHOST handles the Standby protocol.
- The FSUSB host controller wakes up with standby, and stays in standby until the host core becomes functional.

Table 23-1241 lists the STANDBY\_MODE settings for the FSUSB host controller.

**Table 23-1241. FSUSB Host Controller STANDBY\_MODE Settings**

MIDLEMODE Value	Selected Mode	Description
0x0	Force-standby	The high-speed USB host controller enters standby mode unconditionally (Mstandby is asserted unconditionally).
0x1	Not supported (no-standby)	
0x2	Smart-standby	The high-speed USB host controller is ready to enter standby mode when there is no more activity on the USB master interface of the interconnect.
0x3	Smart-standby wakeup	The high-speed USB host controller enters smart mode with asynchronous wakeup enabled.

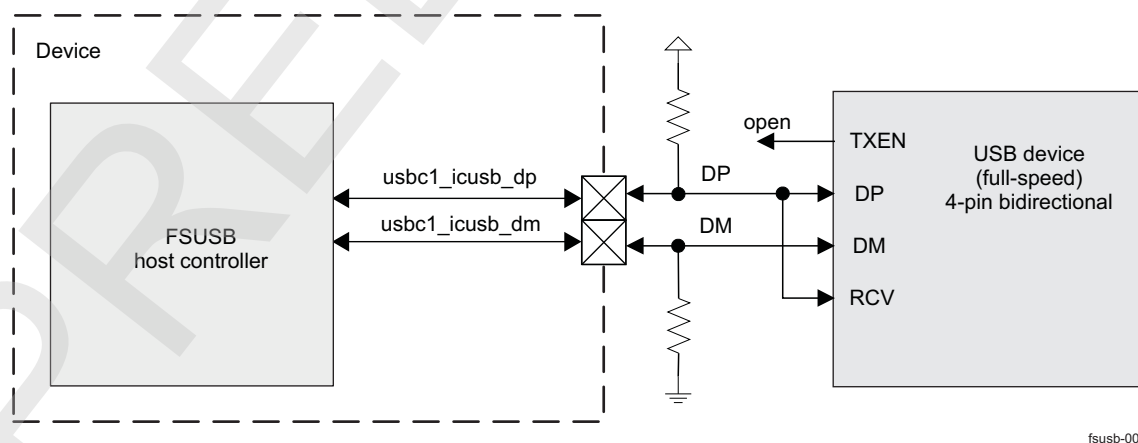
### 23.13.4.3 FSUSB Host Controller Bidirectional Interface Modes

Bidirectional SIM interface modes are implemented using existing bidirectional modes. Their goal is to reduce pin count with respect to 6-/5-pin modes.

Pullup/pulldown resistors on the bidirectional lines are compulsory, because the lines are not constantly driven from one side or the other of the interface. In this case, the pullup on one of the two lines triggers the connect signal from the device to the host. The pullup can be controlled (by the USB device side) or hardwired. In the first case, the USB host sees only the external device if the pullup is active; in the second case, the USB host sees the device as soon as software enables the port in question.

The device supports the 2-pin bidirectional SIM Interface mode with DP/DM encoding. Figure 23-266 shows a USB port using DP/DM encoding.

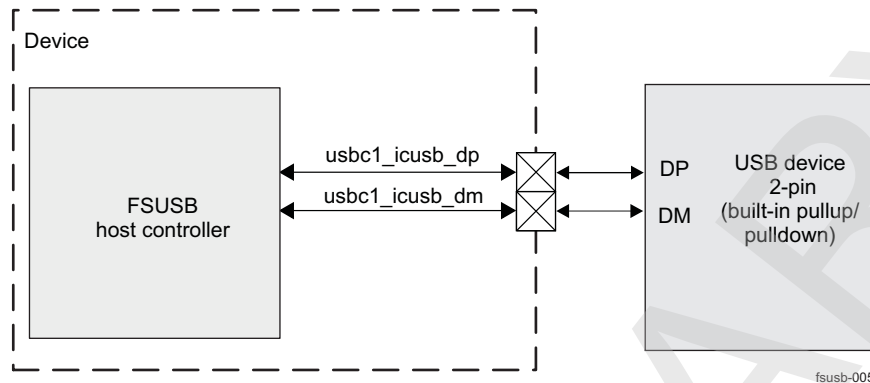
**Figure 23-266. 2-Pin Bidirectional Interface Using DP/DM Encoding With a Standard 4-Pin Bidirectional USB Device**



A specific 2-pin USB device can also be implemented with just the two bidirectional pins. Figure 23-267 shows an example in DP/DM encoding with built-in pullup/pulldown resistors.



**Figure 23-267. 2-Pin Bidirectional Interface Using DP/DM Encoding With a Specific 2-Pin Bidirectional USB Device**



### 23.13.5 FSUSB Host Controller Basic Programming Model

This section describes the low-level hardware programming sequences for the configuration and use of the FSUSB host controller.

#### 23.13.5.1 Global Initialization

##### 23.13.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the FSUSB host controller is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the FSUSB host controller.

Table 23-1242 describes the global initialization of the surrounding modules.

**Table 23-1242. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	All clocks must be enabled (see Table 23-1238). See Section 3.6.3.2, CM Clock Source in Chapter 3, Power, Reset and Clock Management.
Cortex-A9 MPU INTC or Cortex-M3 MPU INTC	The Cortex-A9 MPU or Cortex-M3 MPU INTC must be configured to enable the interrupt request generation to the Cortex-A9 MPU or Cortex-M3 MPU subsystem when interrupt requests are generated by the FSUSB host controller. See the respective functional description in Chapter 17, Interrupt Controllers, for the MPUs.
Interconnect (L3 and L4)	For more information, see Section 13.2, L3 Interconnect, and Section 13.3, L4 Interconnect.

**NOTE:** This module requires an active functional clock for L4 access to function.

##### 23.13.5.1.2 FSUSB Host Controller Global Initialization

This procedure initializes the FSUSB host controller after a power-on or software reset (see Table 23-1243).

**Table 23-1243. High-Speed Multiport USB Host Subsystem Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Perform a software reset on the FSUSB host controller module, except for IDLE, STANDBY, and DISCONNECT logics.	HCOCPSYS[1] SOFT_RESET	0x1

**Table 23-1243. High-Speed Multiport USB Host Subsystem Global Initialization (continued)**

Step	Register/Bit Field/Programming Model	Value
Wait until reset is finished.	HCOCPSYS[1] SOFT_RESET	0x0
Enable IRQ event for the FSUSB host controller.	HCINTERRUPTENABLE[31] MIE	0x1
Enable IRQ event for each function of the FSUSB host controller.	HCINTERRUPTENABLE[6:0] xxx	0x1

## 23.13.6 USBFSHOST Register Manual

### 23.13.6.1 USBFSHOST Instance Summary

**Table 23-1244. USBFSHOST Instance Summary**

Module Name	Base Address	Size
USBFSHOST	0x4A0A 9000	4KB

### 23.13.6.2 USBFSHOST Registers

#### 23.13.6.2.1 USBFSHOST Register Summary

**Table 23-1245. USBFSHOST Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	USBFSHOST L3 Base Address
HCREVISION	R	32	0x0000 0000	0x4A0A 9000
HCCONTROL	RW	32	0x0000 0004	0x4A0A 9004
HCCOMMANDSTATUS	RW	32	0x0000 0008	0x4A0A 9008
HCINTERRUPTSTATUS	RW	32	0x0000 000C	0x4A0A 900C
HCINTERRUPTENABLE	RW	32	0x0000 0010	0x4A0A 9010
HCINTERRUPTDISABLE	RW	32	0x0000 0014	0x4A0A 9014
HCHCCA	RW	32	0x0000 0018	0x4A0A 9018
HCPERIODCURRENTED	R	32	0x0000 001C	0x4A0A 901C
HCCONTROLHEADED	RW	32	0x0000 0020	0x4A0A 9020
HCCONTROLCURRENTED	RW	32	0x0000 0024	0x4A0A 9024
HCBULKHEADED	RW	32	0x0000 0028	0x4A0A 9028
HCBULKCURRENTED	RW	32	0x0000 002C	0x4A0A 902C
HCDONEHEAD	R	32	0x0000 0030	0x4A0A 9030
HCFMINTERVAL	RW	32	0x0000 0034	0x4A0A 9034
HCFMREMAINING	R	32	0x0000 0038	0x4A0A 9038
HCFMNUMBER	R	32	0x0000 003C	0x4A0A 903C
HCPERIODICSTART	RW	32	0x0000 0040	0x4A0A 9040
HCLSTHRESHOLD	RW	32	0x0000 0044	0x4A0A 9044
HCRHDESCRIPTORA	RW	32	0x0000 0048	0x4A0A 9048
HCRHDESCRIPTORB	RW	32	0x0000 004C	0x4A0A 904C
HCRHSTATUS	RW	32	0x0000 0050	0x4A0A 9050
HCRHPORTSTATUS	RW	32	0x0000 0054	0x4A0A 9054
HCOCPREV	R	32	0x0000 0200	0x4A0A 9200
HCOCPHWI	R	32	0x0000 0204	0x4A0A 9204
HCOCPSYS	RW	32	0x0000 0210	0x4A0A 9210

### 23.13.6.2.2 USBFSHOST Register Description

**Table 23-1246. HCREVISION**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .	<b>Instance</b>	USBFSHOST
<b>Description</b>	OHCI revision number		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	IP Revision	R	TI internal data

**Table 23-1247. Register Call Summary for Register HCREVISION**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1248. HCCONTROL**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .	<b>Instance</b>	USBFSHOST
<b>Description</b>	HC operating mode register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RWE	RWC	IR	HCFS	BLE	CLE	IE	PLE	CBSR

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x000000
10	RWE	Remote wake-up enable	RW	0
9	RWC	Remote wake up connected.	RW	0
8	IR	Interrupt routing.	RW	0
7:6	HCFS	Host controller functional state 0x0: HCFS: USB reset 0x1: HCFS: USB resume 0x2: HCFS: USB operational 0x3: HCFS: USB suspend	RW	0x0
5	BLE	Bulk list processing enable 0x0: The bulk ED list is not processed after the next SOF. 0x1: Enables processing of the bulk ED list in the next frame.	RW	0
4	CLE	Control list processing enable 0x0: Control ED list is not processed after the next SOF. 0x1: Enables processing of control ED list in the next frame.	RW	0
3	IE	Isochronous ED processing enabled by host controller driver. 0x0: Isochronous EDs are not processed. 0x1: Enables processing of isochronous EDs	RW	0

Bits	Field Name	Description	Type	Reset
2	PLE	Periodic list enable 0x0: Periodic ED lists are not processed after the next frame. 0x1: Enables processing of periodic ED lists in the next frame	RW	0
1:0	CBSR	Control/bulk service ratio. Specifies the ratio between control and bulk EDs processed in a frame. 0x0: One control ED per bulk ED 0x1: Two control ED per bulk ED 0x2: Three control ED per bulk ED 0x3: Four control ED per bulk ED	RW	0x0

**Table 23-1249. Register Call Summary for Register HCCONTROL**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1250. HCCOMMANDSTATUS**

<b>Address Offset</b>	0x0000 0008	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC Command and status		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SOC		RESERVED								OCR	BLF	CLF	HCR		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R	0x0000
17:16	SOC	Scheduling overrun count	R	0x0
15:4	RESERVED	Reserved	R	0x000
3	OCR	Ownership change request	RW	0
2	BLF	Bulk list filled	RW	0
1	CLF	Control list filled	RW	0
0	HCR	Host controller reset (software reset) Set this bit to initiate a USB host controller reset. This resets most USB host controller OHCI registers. OHCI register accesses must not be attempted until a read of this register returns a 0. 0x0: No effect 0x1: USB host controller is reset.	RW	0

**Table 23-1251. Register Call Summary for Register HCCOMMANDSTATUS**

FSUSB Host Controller Functional Description

- [Resets: \[0\]](#)

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[1\]](#)

**Table 23-1252. HCINTERRUPTSTATUS**

<b>Address Offset</b>	0x0000 000C		
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .	<b>Instance</b>	USBFSHOST
<b>Description</b>	HC Interrupt status		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	OC	RESERVED														RHSC	FNO	UE	RD	SF	WDH	SO									

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	OC	Ownership change	R	0
29:7	RESERVED	Reserved	R	0x000000
6	RHSC	Root hub status change When 0x1: A root hub status change has occurred. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
5	FNO	Frame number overflow When 0x1: A frame number overflow has occurred. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
4	UE	Unrecoverable error. When 0x1: An unrecoverable error has occurred. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
3	RD	Resume detected. When 0x1: A downstream device has issued a resume request. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
2	SF	Start of frame. When 0x1: A SOF has been issued. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
1	WDH	Write done head When 0x1: the USB host controller has updated the <a href="#">HCDONEHEAD</a> register. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
0	SO	Scheduling overrun When 0x1: A scheduling overrun has occurred. Write 0x0: No effect Write 0x1: Clears this bit	RW	0

**Table 23-1253. Register Call Summary for Register HCINTERRUPTSTATUS**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1254. HCINTERRUPTENABLE**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .	<b>Instance</b>	USBFSHOST
<b>Description</b>	HC Interrupt Enable		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIE	OC	RESERVED														RHSC	FNO	UE	RD	SF	WDH	SO									

Bits	Field Name	Description	Type	Reset
31	MIE	Master interrupt enable When 0x1: Allows other enabled OHCI interrupt sources to propagate to the device interrupt controller. When 0x0: OHCI interrupt sources are ignored. Write 0x0: No effect Write 0x1: Sets this bit	RW	0
30	OC	Ownership change	RW	0
29:7	RESERVED	Reserved	R	0x000000
6	RHSC	Root hub status change When 0x1 and MIE is 0x1: Allows root hub status change interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: root hub status change interrupts do not propagate. Write 0x0: No effect Write 0x1: Sets this bit	RW	0
5	FNO	Frame number overflow. When 0x1 and MIE is 0x1: Allows FNO interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: FNO interrupts do not propagate. Write 0x0: No effect Write 0x1: Sets this bit	RW	0
4	UE	Unrecoverable error. When 0x1 and MIE is 0x1: Allows UE interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: UE interrupts do not propagate. Write 0x0: No effect Write 0x1: Sets this bit	RW	0
3	RD	Resume detected. When 0x1 and MIE is 0x1: Allows RD interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: RD interrupts do not propagate. Write 0x0: No effect Write 0x1: Sets this bit	RW	0
2	SF	Start of frame When 0x1 and MIE is 0x1: Allows SF interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: SF interrupts do not propagate. Write 0x0: No effect Write 0x1: Sets this bit	RW	0
1	WDH	Write done head When 0x1 and MIE is 0x1: Allows WDH interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: WDH interrupts do not propagate. Write 0x0: No effect Write 0x1: Sets this bit	RW	0
0	SO	Scheduling overrun. When 0x1 and MIE is 0x1: Allows SO interrupts to propagate to the device interrupt controller. When 0x0 or MIE is 0x0: SO interrupts do not propagate. Write 0x0: No effect Write 0x1: Sets this bit	RW	0

**Table 23-1255. Register Call Summary for Register HCINTERRUPTENABLE**

FSUSB Host Controller Basic Programming Model

- [FSUSBHC Global Initialization: \[0\] \[1\]](#)

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[2\]](#)
- [USBFSHOST Register Description: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

**Table 23-1256. HCINTERRUPTDISABLE**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC Interrupt disable		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIE	OC	RESERVED														RHSC	FNO	UE	RD	SF	WDH	SO									

Bits	Field Name	Description	Type	Reset
31	MIE	Master interrupt enable Always reads 0x0 Write 0x0: No effect Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> MIE bit	RW	0
30	OC	Ownership change	RW	0
29:7	RESERVED	Reserved	R	0x000000
6	RHSC	Root hub status change Always reads 0x0 Write 0x0: No effect Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> RHSC bit	RW	0
5	FNO	Frame number overflow Always reads 0x0 Write 0x0: No effect Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> FNO bit	RW	0
4	UE	Unrecoverable error Always reads 0x0 Write 0x0: No effect Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> UE bit	RW	0
3	RD	Resume detected Always reads 0x0 Write 0x0: No effect Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> RD bit	RW	0
2	SF	Start of frame Always reads 0x0 Write 0x0: No effect Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> SF bit	RW	0
1	WDH	Write done head Always reads 0x0 Write 0x0: No effect Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> WDH bit	RW	0
0	SO	Scheduling overrun Always reads 0x0 Write 0x0: No effect Write 0x1: Clears the <a href="#">HCINTERRUPTENABLE</a> SO bit	RW	0

**Table 23-1257. Register Call Summary for Register HCINTERRUPTDISABLE**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)



**Table 23-1258. HCHCCA**

<b>Address Offset</b>	0x0000 0018		
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .	<b>Instance</b>	USBFSHOST
<b>Description</b>	HC HCCA address register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HCCA																RESERVED															

Bits	Field Name	Description	Type	Reset
31:8	HCCA	Physical address of the beginning of the HCCA	RW	0x000000
7:0	RESERVED	Reserved	R	0x00

**Table 23-1259. Register Call Summary for Register HCHCCA**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1260. HCPERIODCURRENTED**

<b>Address Offset</b>	0x0000 001C		
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .	<b>Instance</b>	USBFSHOST
<b>Description</b>	HC Current periodic register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	PCED	Physical address of current ED on the periodic ED list	R	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1261. Register Call Summary for Register HCPERIODCURRENTED**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1262. HCONTROLHEADED**

<b>Address Offset</b>	0x0000 0020		
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .	<b>Instance</b>	USBFSHOST
<b>Description</b>	HC head control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	CHED	Physical address of head ED on the control ED list	RW	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1263. Register Call Summary for Register HCCONTROLHEADED**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1264. HCCONTROLCURRENTED**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC current control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	CCED	Physical address of current ED on the control ED list	RW	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1265. Register Call Summary for Register HCCONTROLCURRENTED**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1266. HCBULKHEADED**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC head bulk register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BHED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	BHED	Physical address of head ED on the bulk ED list	RW	0x00000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1267. Register Call Summary for Register HCBULKHEADED**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1268. HCBULKCURRENTED**

<b>Address Offset</b>	0x0000 002C	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC current bulk register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCED																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	BCED	Physical address of current ED on the bulk ED list	RW	0x0000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1269. Register Call Summary for Register HCBULKCURRENTED**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1270. HCDONEHEAD**

<b>Address Offset</b>	0x0000 0030	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC head done register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DH																RESERVED															

Bits	Field Name	Description	Type	Reset
31:4	DH	Physical address of last TD that was added to the done queue	R	0x0000000
3:0	RESERVED	Reserved	R	0x0

**Table 23-1271. Register Call Summary for Register HCDONEHEAD**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)
- [USBFSHOST Register Description: \[1\]](#)

**Table 23-1272. HCFMINTERVAL**

<b>Address Offset</b>	0x0000 0034	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC Frame Interval register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIT	FSMPS											RESERVED	FI																		

Bits	Field Name	Description	Type	Reset
31	FIT	Frame interval toggle	RW	0
30:16	FSMPS	Largest data packet size for full-speed packets, bit times.	RW	0x0000
15:14	RESERVED	Reserved	R	0x0
13:0	FI	Frame interval Number of 12-MHz clocks in the USB frame. The nominal value is set to 11,999 to give a 1-ms frame.	RW	0x2EDF

**Table 23-1273. Register Call Summary for Register HCFMINTERVAL**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1274. HCFMREMAINING**

<b>Address Offset</b>	0x0000 0038	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC Frame-remaining register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FR															

Bits	Field Name	Description	Type	Reset
31	FRT	Frame remaining toggle	R	0
30:14	RESERVED	Reserved	R	0x00000
13:0	FR	Frame remaining	R	0x0000

**Table 23-1275. Register Call Summary for Register HCFMREMAINING**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1276. HCFMNUMBER**

<b>Address Offset</b>	0x0000 003C	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC Frame number register		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FN															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	FN	Frame number	R	0x0000

**Table 23-1277. Register Call Summary for Register HCFMNUMBER**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1278. HCPERIODICSTART**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC Periodic start register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PS															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x00000
13:0	PS	Periodic start The host controller driver must program this value to be about 10% less than the frame interval field value so that control and bulk EDs have priority for the first 10% of the frame; then periodic EDs have priority for the remaining 90% of the frame.	RW	0x0000

**Table 23-1279. Register Call Summary for Register HCPERIODICSTART**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1280. HCLSTHRESHOLD**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC low-speed threshold register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LST															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x000000
11:0	LST	Low-speed threshold.	RW	0x628

**Table 23-1281. Register Call Summary for Register HCLSTHRESHOLD**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1282. HCRHDESCRIPTORA**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC root hub A register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POTPG								RESERVED								NOCP	OCPM	DT	NPS	PSM	NDP										

Bits	Field Name	Description	Type	Reset
31:24	POTPG	Power-on to power-good time Defines the minimum length of time (2 ms * POTPG) between the USB host controller turning on power to a downstream port, and when the USB host can access the downstream device.	RW	0x0A
23:13	RESERVED	Reserved	R	0x000
12	NOCP	No overcurrent protection 0x0: Overcurrent status is reported collectively for all downstream ports. 0x1: the USB host controller does not implement overcurrent protection inputs.	RW	0
11	OCPM	Overcurrent protection mode	RW	0

Bits	Field Name	Description	Type	Reset
10	DT	Device type Always reads 0x0: Indicates that the USB host controller implemented is not a compound device	R	0
9	NPS	No power switching 0x0: VBUS power switching is supported, either per-port or all-port switched per the power. 0x1: VBUS power switching is not supported, power is available to all downstream ports.	RW	0
8	PSM	Power switching mode 0x0: Indicates that all ports are powered at the same time 0x1: Individual port power switching is supported	RW	0
7:0	NDP	Number of downstream ports.	R	0x01

**Table 23-1283. Register Call Summary for Register HCRHDESCRIPTORA**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1284. HCRHDESCRIPTORB**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC root hub B register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PPCM	RESERVED	RESERVED																DR	RESERVED				

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0000
17	PPCM	Port power control mask This bit defines whether downstream port #1 has port power controlled by the global power control. When set, the port power state is only affected by per-port power control. When cleared the port is controlled by the global power switch. This field is not valid if the device is configured to global switch mode.	RW	0
16	RESERVED		R	0
15:2	RESERVED		R	0x0000
1	DR	Device removable This bit defines whether the downstream port has a removable device. When cleared, the attached device is removable. When set, the attached device is not removable. bit 1: Device attached to port #1	RW	0
0	RESERVED		R	0

**Table 23-1285. Register Call Summary for Register HCRHDESCRIPTORB**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)
- [USBFSHOST Register Description: \[1\] \[2\]](#)

**Table 23-1286. HCRHSTATUS**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC root hub status register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
CRWE	RESERVED														OCIC	LPSC	DRWE	RESERVED														OCI	LPS

Bits	Field Name	Description	Type	Reset
31	CRWE	Clear remote wake-up enable Write 0x0: No effect Write 0x1: Clears the device remote wake-up enable bit	W	0
30:18	RESERVED	Reserved	R	0x0000
17	OCIC	Overcurrent indication change This bit is automatically set when the overcurrent indicator bit changes. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
16	LPSC	Local power status change Always reads 0x0: The root hub does not support the local power status feature. Write 0x0: No effect Write 0x1: Sets port power status bits for all ports, if power switching mode is 0. Sets port power status bits for ports with their corresponding port power control mask bits cleared if power switching mode is 1.	RW	0
15	DRWE	Device remote wake-up enable Enables a connect status change event as a resume event, causing a USB suspend to USB resume state transition and sets the resume detected interrupt status bit. Read 0x1: Connect status change is a remote wake-up event. Read 0x0: Connect status change is not a remote wake-up event. Write 0x0: No effect Write 0x1: Sets the device remote wake-up enable bit	RW	0
14:2	RESERVED	Reserved	R	0x0000
1	OCI	Overcurrent indicator. Reports global overcurrent indication if global overcurrent reporting is selected. If per-port overcurrent protection is implemented, this bit is always 0. Read 0x0: All power operations are normal. Read 0x1: An overcurrent condition exists.	R	0
0	LPS	Local power status. Always reads 0x0 Write 0x0: No effect Write 0x1: When in global power mode (power switching mode = 0), turns off power to all ports. If in per-port power mode (power switching mode = 1), turns of power to those ports whose corresponding port power control mask bit is 0.	RW	0

**Table 23-1287. Register Call Summary for Register HCRHSTATUS**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1288. HCRHPORTSTATUS**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	USBFSHOST
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .		
<b>Description</b>	HC Port 1 status and control register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRSC	OCIC	PSSC	PESC	CSC	RESERVED								LSDA_CPP	PPS_SPP	RESERVED	PRS_SPR	POCI_CSS	PSS_SPS	PES_SPE	CCS_CPE			

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved	R	0x000
20	PRSC	Port 1 reset status change This bit is set when the Port 1 port reset status bit has changed. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
19	OCIC	Port 1 overcurrent indicator change This bit is set when the Port 1 port overcurrent indicator has changed. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
18	PSSC	Port 1 suspend status change This bit is set when the Port1 port suspend status has changed. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
17	PESC	Port 1 enable status change This bit is set when the Port1 port enable status has changed. Write 0x0: No effect Write 0x1: Clears this bit	RW	0
16	CSC	Port 1 connect status change This bit is set when the Port1 port current connect status has changed due to a connect or disconnect event. If current connect status is 0 when a set port reset, set port enable, or set port suspend write occurs, this bit is set. Write 0x0: No effect Write 0x1: Clears this bit Note: If the DR bit <a href="#">HCRHDESCRIPTORB[1]</a> is set, this bit is set only after a root hub reset to inform the system that the device is attached.	RW	0
15:10	RESERVED	Reserved	R	0x00
9	LSDA_CPP	Port 1 low-speed device attached/clear port power. This bit is valid only when port 1 current connect status is 1. Read 0x0: A full-speed device is attached to port 1. Read 0x1: A low-speed device is attached to port 1. Write 0x0: No effect Write 0x1: Clears the port 1 port power status.	RW	0
8	PPS_SPP	Port 1 port power status/set port power. Read 0x0: Port 1 power is off. Read 0x1: Port 1 power is on. Write 0x0: No effect Write 0x1: Sets the port 1 port power status bit	RW	0
7:5	RESERVED	Reserved	R	0x0
4	PRS_SPR	Port 1 port reset status/set port reset. Read 0x0: USB reset is not being sent to port 1. Read 0x1: Port 1 is signaling the USB reset. Write 0x0: No effect Write 0x1: Sets the port 1 port reset status bit and causes the USB host controller to begin signaling USB reset to port 1	RW	0



Bits	Field Name	Description	Type	Reset
3	POCI_CSS	Port 1 port overcurrent indicator/clear suspend status Read 0x0: No port 1 port overcurrent condition has occurred. Read 0x1: A port 1 port overcurrent condition has occurred. Write 0x0: No effect Write 0x1: When port 1 port suspend status is 1, it causes resume signaling on port 1. When port 1 port suspend status is 0, it has no effect.	RW	0
2	PSS_SPS	Port 1 port suspend status/set port suspend This bit is cleared automatically at the end of the USB resume sequence and also at the end of the USB reset sequence. Write 0x0: No effect Read 0x0: Port 1 is not in the USB suspend state. Read 0x1: Port 1 is in the USB suspend state or is in the resume sequence. Write 0x1: If port 1 current connect status is 1, sets the port 1 port suspend status bit and places port 1 in USB suspend state. If current connect status is 0, it sets connect status instead change to inform the USB host controller driver in an attempt to suspend a disconnected port.	RW	0
1	PES_SPE	Port 1 port enable status/set port enable This bit is automatically set at completion of port 1 USB reset, if it was not already set before the USB reset completed, and is automatically set at the end of a USB suspend, if the port was not enabled when the USB resume completed. Read 0x0: Port 1 is not enabled. Read 0x1: Port 1 is enabled. Write 0x0: No effect Write 0x1: When port 1 current connect status is 1, sets the port 1 port enable status bit. When port 1 current status is 0, it has no effect.	RW	0
0	CCS_CPE	Port 1 current connection status/clear port enable Read 0x0: No USB device is attached to port 1. Read 0x1: Port 1 currently has a USB device attached. Write 0x0: No effect Write 0x1: Clears the port 1 port enable bit Note: This bit is set to 1 if the DR bit <a href="#">HCRHDESCRIPTORB[1]</a> is set to indicate a nonremovable device on port 1.	RW	0

**Table 23-1289. Register Call Summary for Register HCRHPORTSTATUS**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1290. HCOCPREV**

<b>Address Offset</b>	0x0000 0200	<b>Instance</b>	USBFSHOST																																																												
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .																																																														
<b>Description</b>																																																															
<b>Type</b>	R																																																														
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">REVISION</td> <td colspan="12"></td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	REVISION																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																
REVISION																																																															
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																											
31:0	REVISION	IP Revision	R	TI Internal data																																																											

**Table 23-1291. Register Call Summary for Register HCOCPREV**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1292. HCOCPHWI**

<b>Address Offset</b>	0x0000 0204	
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .	<b>Instance</b> USBFSHOST
<b>Description</b>	Hardware information register, maintains the IP modules hardware configuration. The fields can be set to different values through parameters.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INFO																															

Bits	Field Name	Description	Type	Reset
31:0	INFO	IP-module dependent. Information about the IP module hardware configuration, that is, typically the module HDL generics (if any).	R	0x0000 002A

**Table 23-1293. Register Call Summary for Register HCOCPHWI**

USBFSHOST Register Manual

- [USBFSHOST Register Summary: \[0\]](#)

**Table 23-1294. HCOCPSYS**

<b>Address Offset</b>	0x0000 0210	
<b>Physical Address</b>	See <a href="#">Table 23-1245</a> .	<b>Instance</b> USBFSHOST
<b>Description</b>	OCP system configuration register holds the OCP power down control fields that controls the clock management.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																APPLICATIONSTARTCLOCK	SIMULATIONSCALEDOWN	RESERVED	STANDBY_MODE	IDLE_MODE	SOFT_RESET	RESERVED									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	APPLICATIONSTARTCLOCK	1: RCFG_SUSPEND_O output port is masked and this output port is held low (1'b0) 0: RCFG_SUSPEND_O port functions normally, that is, it indicates the SUSPEND mode.	RW	0
8	SIMULATIONSCALEDOWN	1: Simulation runs in time scaled down mode. 0: Simulation runs in real time. NOTE: This bit should not be set to 1 in actual hardware.	RW	0
7:6	RESERVED		R	0x0
5:4	STANDBY_MODE	Standby mode, controls the way USBFSHOST handles the STANDBY protocol. The application can program this field in the following configurations 2'b00 : Force standby 2'b01: Not supported (No-Standby) 2'b10: Smart standby 2'b11: Smart standby wake up	RW	0x3

Bits	Field Name	Description	Type	Reset
3:2	IDLE_MODE	The idle mode controls the way USBFSHOST handles the IDLE protocol. The application can program this field in the following configurations 2'b00: Force Idle 2'b01: Not supported (No-Idle) 2'b10: Smart-Idle 2'b11: Smart-Idle wake up	RW	0x2
1	SOFT_RESET	Setting this bit to 1 resets the entire IP, except for OCP-IDLE, OCP-STANDBY, and OCP-DISCONNECT logic. This bit remains set until the IP comes out of soft reset.	RW	0
0	RESERVED		R	0

**Table 23-1295. Register Call Summary for Register HCOCPSYS**


---

**FSUSB Host Controller Functional Description**

- [Resets: \[0\]](#)
- [IDLE Protocol: \[1\] \[2\]](#)
- [STANDBY Protocol: \[3\]](#)

---

**FSUSB Host Controller Basic Programming Model**

- [FSUSBHC Global Initialization: \[4\] \[5\]](#)

---

**USBFSHOST Register Manual**

- [USBFSHOST Register Summary: \[6\]](#)
-

PRELIMINARY

## MMC/SD/SDIO

This chapter describes the features and functions of the multimedia card/SD card (MMC/SD/SDIO) interface of the device.

Topic	Page
24.1 MMC/SD/SDIO Overview .....	4958
24.2 MMC/SD/SDIO Environment .....	4964
24.3 MMC/SD/SDIO Integration .....	4972
24.4 MMC/SD/SDIO Functional Description .....	4975
24.5 MMC/SD/SDIO Programming Guide .....	5004
24.6 MMC/SD/SDIO Register Manual .....	5030

## 24.1 MMC/SD/SDIO Overview

The multimedia card high-speed/SD/SDIO (MMC/SD/SDIO) host controller provides an interface between a local host (LH) such as a microprocessor unit (MPU) or digital signal processor (DSP) and either MMC, SD memory cards, or SDIO cards and handles MMC/SD/SDIO transactions with minimal LH intervention.

Optionally, the controller is connected to the L3 interconnect to have a direct access to system memory. It also supports two DMA slave channels or a DMA master access (in this case, slave DMA channels are deactivated) depending on its integration. These controllers are embedded MMC/SD (eMMC/SD) controllers.

The application interface manages transaction semantics. The MMC/SD/SDIO host controller deals with MMC/SD/SDIO protocol at transmission level, data packing, adding cyclic redundancy checks (CRC), start/end bit, and checking for syntactical correctness.

The application interface can send every MMC/SD/SDIO command and poll for the status of the adapter or wait for an interrupt request, which is sent back in case of exceptions or to warn of end of operation.

The application interface can read card responses or flag registers. It can also mask individual interrupt sources. All these operations can be performed by reading and writing control registers. The MMC/SD/SDIO host controller also supports two DMA channels.

There are five MMC/SD/SDIO host controllers inside the device:

[Figure 24-1](#) gives an overview of the MMC/SD/SDIO1 controller instance (eMMC/SD).

[Figure 24-2](#) gives an overview of the MMC/SD/SDIO2 controller instance (eMMC/SD).

[Figure 24-3](#) gives an overview of the MMC/SD/SDIO3 through MMC/SD/SDIO5 controller instances.

Figure 24-1. MMC/SD/SDIO1 Overview (where i = 1)

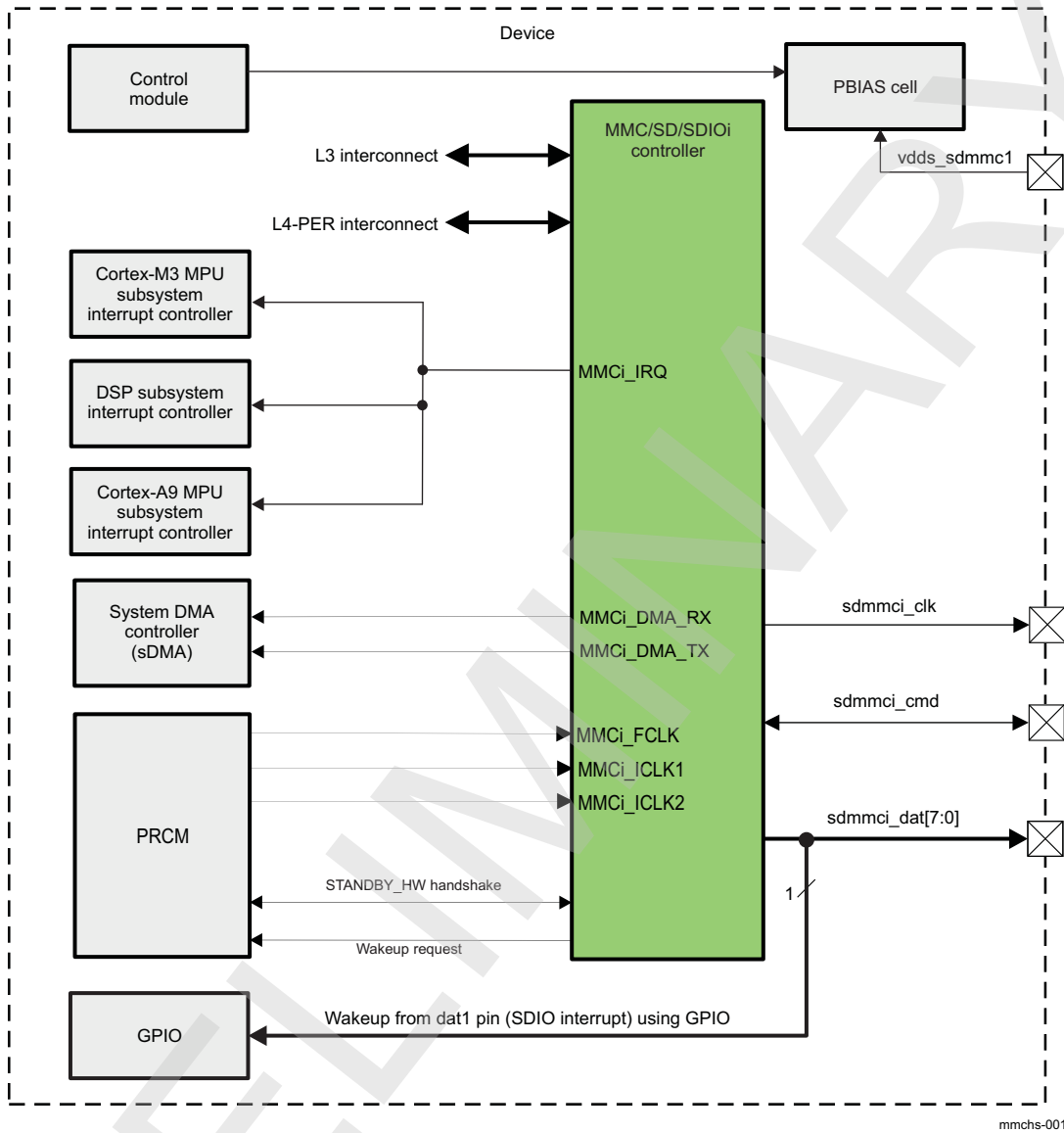


Figure 24-2. MMC/SD/SDIO2 Overview (where i = 2)

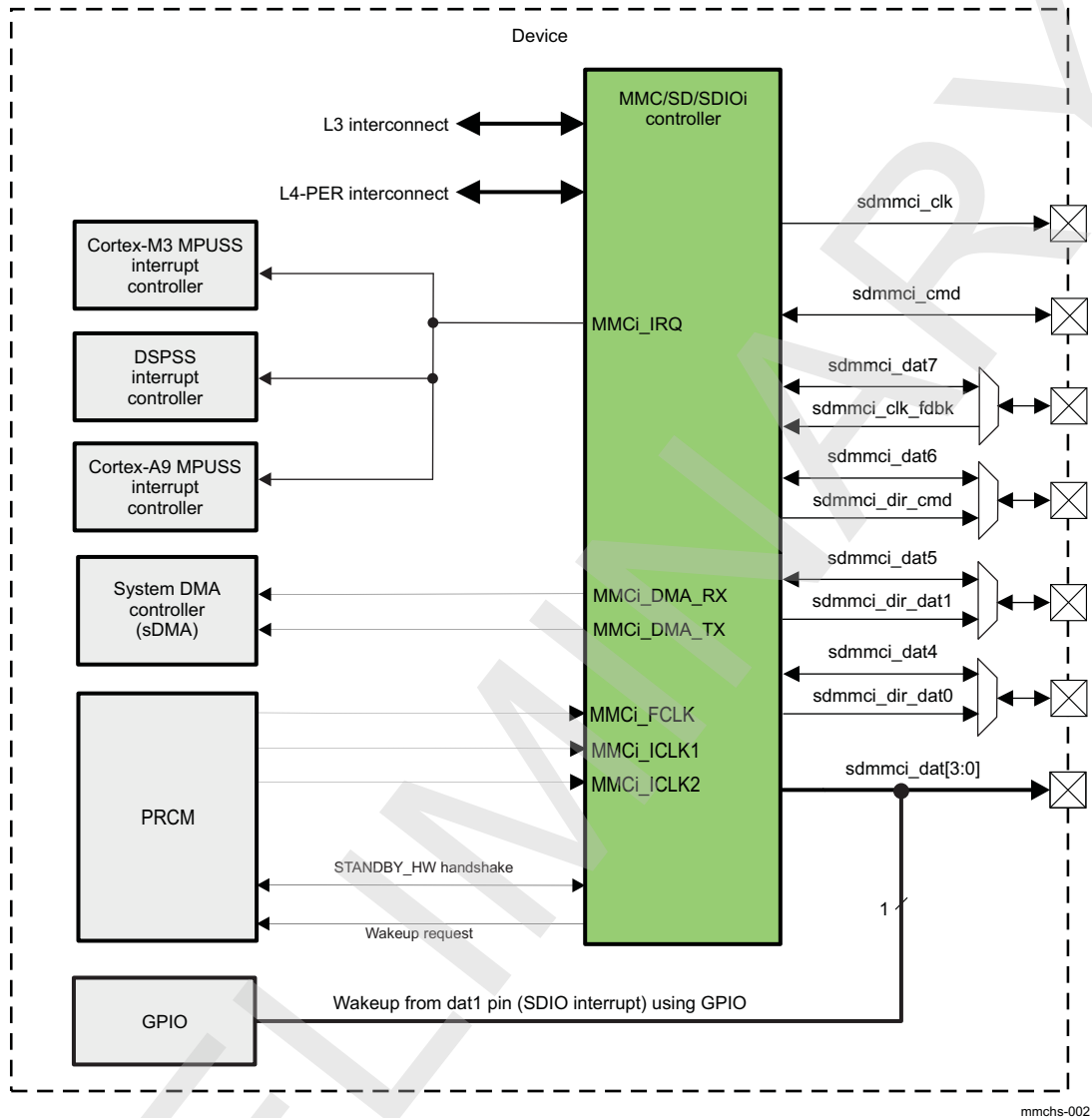
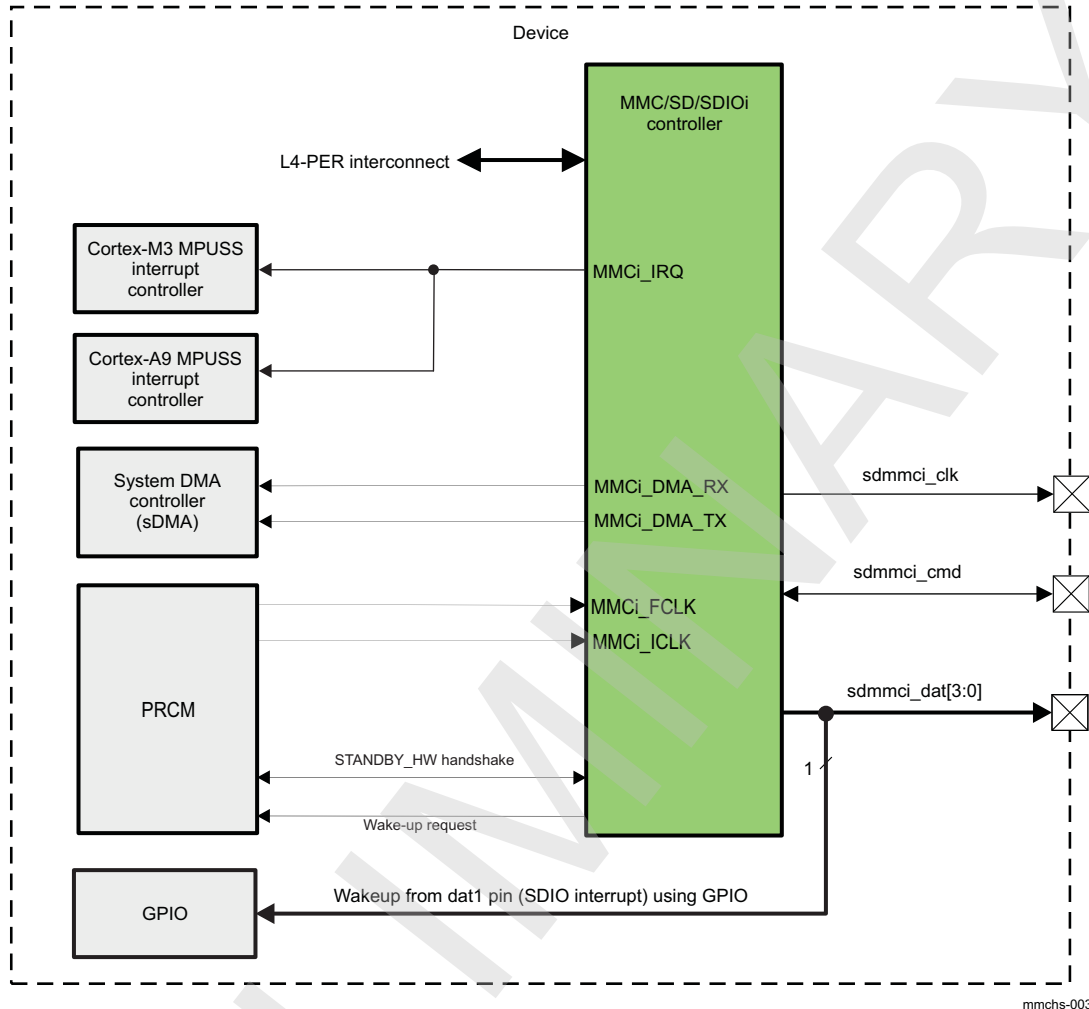




Figure 24-3. MMC/SD/SDIO3 Through MMC/SD/SDIO5 Overview (where i = 3, 4, or 5)



### 24.1.1 MMC/SD/SDIO Features

This section describes the features supplied by the MMC/SD/SDIO module.

Compliance with standards:

- Full compliance with MMC command/response sets as defined in the *Multimedia Card System Specification*, v4.4, including high-capacity (size > 2GB) cards HC MMC
- Full compliance with SD command/response sets as defined in the *SD Memory Card Specifications*, v3.0, including high-capacity SDHC cards up to 32GB
- Full compliance with SDIO command/response sets and interrupt/read-wait mode as defined in the *SDIO Card Specification, Part E1*, v2.00
- Compliance with sets as defined in the *SD Card Specification, Part A2, SD Host Controller Standard Specification*, v2.00
- Full compliance with MMC bus testing procedure as defined in the *Multimedia Card System Specification*, v4.4
- Full compliance with CE-ATA command/response sets as defined in the *CE-ATA Standard Specification*
- Full compliance with ATA for MMCA specification
- Support command completion signal (CCS) and command completion signal disable (CCSD) management as specified in the *CE-ATA Standard Specification*

Main features of the MMC/SD/SDIO host controller:

- Flexible architecture allowing support for new command structure
- Support:
  - 1- or 4-bit transfer mode specifications for SD and SDIO cards
  - 1-, 4-, or 8-bit transfer mode specifications for MMC cards
- Built-in buffer for read or write (up to 4096 bytes in single-buffering, 2048 bytes in double-buffering)
- 32-bit-wide access bus to maximize bus throughput
- Single interrupt line for multiple interrupt source events
- Two slave DMA channels (one for TX, one for RX)
- Designed for low power
- Programmable clock generation
- Support SDIO read wait and suspend/resume functions
- Support stop at block gap
- Support MMC4.4 JC 64.1 boot mode operations
- Support dual data rate transfers (DDR mode) as specified in *JEDEC JC64 Item #18.00 Bus Performance Proposal (February 2008)*
- Support SDA 2.0 Part A2 programming model

Optional features (depending on module integration):

- Master interface (level 3 [L3] interconnect)
- One master DMA (32-bit ADMA2), replacing the two slave DMA channels
- Retention mode is supported
- Support SDA 2.0 Part A2 DMA feature (ADMA2)

Supported data rates:

- Up to 768 Mbps (96 Mbps) in MMC mode, 8-bit DDR data transfer, 48-MHz card clock when functional clock is 48 MHz
- Up to 384 Mbps (48 Mbps) MMC mode, 8-bit SDR data transfer, 48-MHz card clock when functional clock is 48 MHz
- Up to 192 Mbps (24 Mbps) in high-speed SD mode 4-bit data transfer
- Up to 24 Mbps (3 Mbps) in default SD mode 1-bit data transfer

Supported MMC clock frequencies:

- Up to 64 MHz in standard mode and up to 48 MHz in DDR mode
- Up to 96 MHz in single data rate (SDR) mode for embedded devices

Known limitations:

- No built-in hardware support for error correction codes (ECC). For information about ECC, see the *Multimedia Card System Specification, v4.4*, and the *SD Memory Card Specifications, v3.0*.
- The maximum block size defined in the *SD Memory Card Specifications, v3.0*, which the host driver can read and write to the buffer in the host controller, is 2048 bytes. MMC supports a maximum block size of 1024 bytes. Up to 512 byte transfers, the buffer in MMC is considered as a double-buffering with ping-pong management; half of the buffer can be written while the other half is read. For 512 to 1024 byte transfers, the entire buffer is dedicated to the transfer (read only or write only).
- SPI transfers are not supported.
- Module does not support card insertion/removal sensing with pullup resistor on sdmmc\_i\_dat3 bus line, as specified in the *SD Memory Card Specifications – Part 1 Physical Layer Specification (SD Group, May 2006)*.

The differences between the MMC/SD/SDIO host controllers and a standard SD host controller defined by the *SD Card Specification, Part A2, SD Host Controller Standard Specification, v2.00* are:

- The MMC/SD/SDIO host controllers support MMC cards.
- The clock divider in the MMC/SD/SDIO host controller supports a wider range of frequency than specified in the *SD Memory Card Specifications, v2.0*. The MMC/SD/SDIO host controller supports odd and even clock ratio.

- The MMC/SD/SDIO host controller supports configurable busy time-out.
- sDMA and ADMA2 64-bit modes are not supported.
- No external LED control

---

**NOTE:** Only even ratios are supported in DDR mode

---

## 24.2 MMC/SD/SDIO Environment

One MMC/SD/SDIO host controller can support one MMC memory card, one SD memory card, or one SDIO card.

Other combinations (for example, two SD cards, one MMC card, and one SD card) are not supported through a single controller.

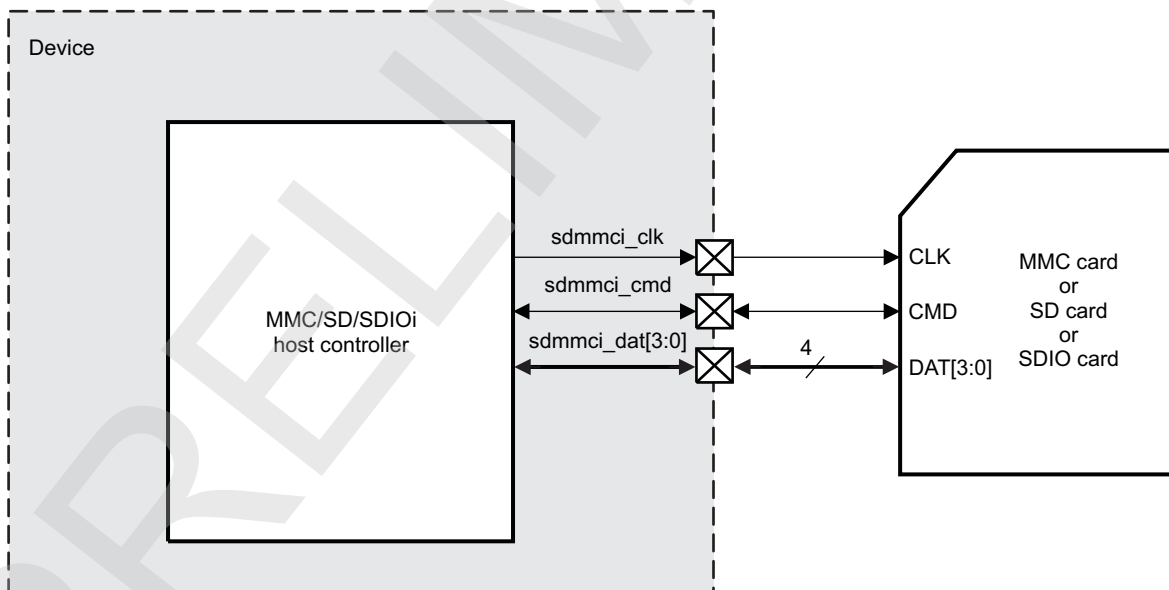
- The MMC/SD/SDIO1 controller integrates an internal transceiver that allows a direct connection to the MMC/SD/SDIO card (1.8 V and 3 V), without external transceiver. It is connected to the level 3 (L3) interconnect (internal DMA enabled) and supports 1-, 4-, and 8-bit data transfers.
- The MMC/SD/SDIO2 controller allows connecting MMC/SD/SDIO cards (only 1.8V cards) or an external device that uses the MMC/SD/SDIO interface (JC64, for example). The module is connected to the L3 interconnect (internal DMA enabled) and supports 1-, 4-, and 8-bit data transfers. The second instance also supports an external transceiver and provides direction signals for data and command. Using an external transceiver device precludes 8-bit transfer mode.
- The MMC/SD/SDIO3, MMC/SD/SDIO4, and MMC/SD/SDIO5 controllers allow connecting MMC/SD/SDIO cards (only 1.8V cards) or an external device that uses the MMC/SD/SDIO interface (wireless LAN, for example). The modules are connected to the L4 interconnect (internal DMA enabled) and support 1- and 4-bit data transfers.

### 24.2.1 MMC/SD/SDIO Functional Modes

#### 24.2.1.1 MMC/SD/SDIO Connected to an MMC, an SD, or an SDIO Card

Figure 24-4 shows the MMC/SD/SDIO<sub>i</sub> host controller (where *i* = 1 through 5) connected to an MMC, an SD, or an SDIO card and its related external connections.

**Figure 24-4. MMC/SD/SDIO<sub>i</sub> Controller Connected to an MMC, an SD, or an SDIO Card Without External Transceiver (where *i* = 1 through 5)**



mmchs-004

**NOTE:** Modules MMC/SD/SDIO1 and MMC/SD/SDIO2 support data transfers on 8 bits (and thus handle data on signals `sdmmc_i_dat[7:0]`)

Table 24-1 describes the MMC/SD/SDIO<sub>i</sub> inputs and outputs.

**Table 24-1. MMC/SD/SDIOi I/O Description Without External Transceiver (where i = 1 through 5)**

Signal Name	I/O <sup>(1)</sup>	Description	Reset Value
sdmmc_clk	O	External clock for MMC/SD/SDIO card <sup>(2)</sup>	0
sdmmc_cmd	I/O	Command signal	HiZ <sup>(3)</sup>
sdmmc_dat[3:0]	I/O	Data signals	HiZ <sup>(3)</sup>
sdmmc_dat[7:4]	I/O	Data signals (only for modules MMC/SD/SDIO1 and MMC/SD/SDIO2)	HiZ <sup>(3)</sup>

<sup>(1)</sup> I = Input; O = Output.

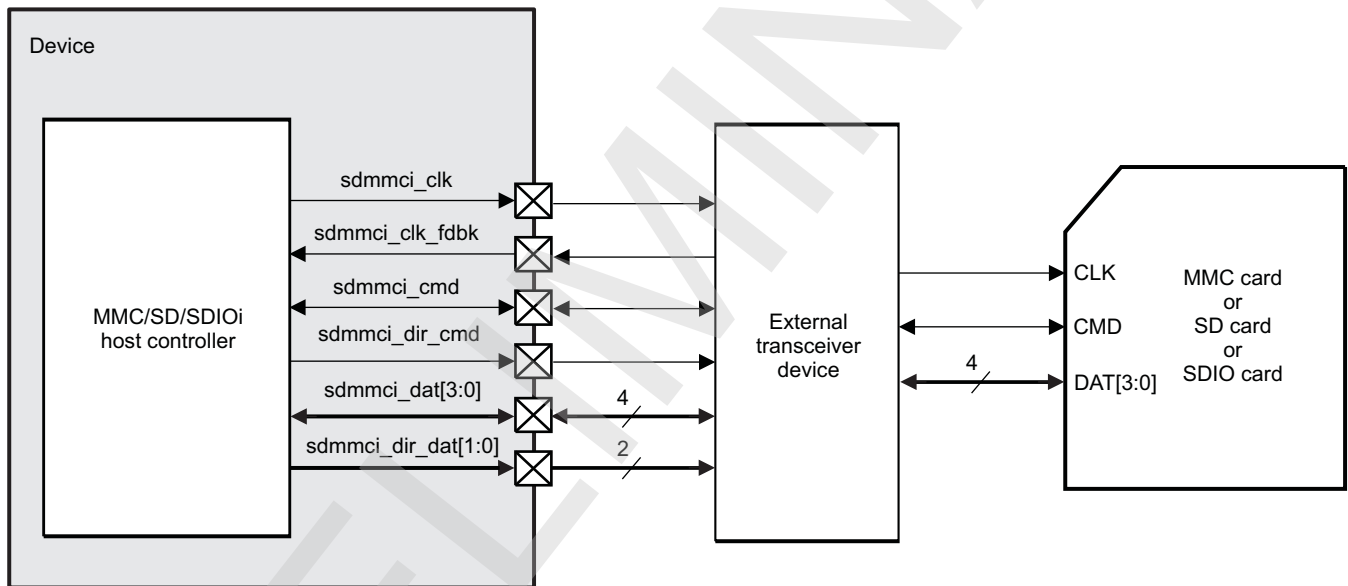
<sup>(2)</sup> This output signal is also used as retiming input.

<sup>(3)</sup> Initialized as input upon reset

**24.2.1.2 MMC/SD/SDIO Connected to an MMC, an SD, or an SDIO Card Through an External Transceiver Device**

Figure 24-5 shows the MMC/SD/SDIOi (where i = 2) host controller connected to an MMC, an SD, or an SDIO card and its related external connections.

**Figure 24-5. MMC/SD/SDIOi Controller Connected to an MMC, an SD, or an SDIO Card with External Transceiver (where i = 2)**



mmchs-005

Table 24-2 describes the MMC/SD/SDIOi inputs and outputs.

**Table 24-2. MMC/SD/SDIOi I/O Description With External Transceiver (where i = 2)**

Signal Name	I/O	Description	Reset Value
sdmmc_clk	O	External clock for MMC/SD/SDIO card	0
sdmmc_clk_fdbk	I	Input clock from MMC/SD/SDIO card	0
sdmmc_cmd	I/O	Command signal	HiZ <sup>(1)</sup>
sdmmc_dir_cmd	O	Direction control for mmc2_cmd signal when an external transceiver is used (high when transmit, low when receive)	0
sdmmc_dat[3:0]	I/O	Data signals	HiZ <sup>(1)</sup>
sdmmc_dir_dat0	O	Direction control for mmc2_dat0 signal when an external transceiver is used (high when transmit, low when receive)	0

<sup>(1)</sup> (1) Initialized as input upon reset

**Table 24-2. MMC/SD/SDIOi I/O Description With External Transceiver (where i = 2) (continued)**

Signal Name	I/O	Description	Reset Value
sdmmc_dir_dat1	O	Direction control for mmc2_dat1 signal when an external transceiver is used (high when transmit, low when receive)	0

**NOTE:** There is no signal for controlling individually the sdmmc\_dir\_dat2 and sdmmc\_dir\_dat3 direction (sdmmc\_dir\_dat2 and sdmmc\_dir\_dat3 are not delivered), this may limit the use of this interface with the SDIO card requesting the wait feature on the sdmmc\_dir\_dat2 line.

## 24.2.2 Protocol and Data Format

The bus protocol between the MMC/SD/SDIOi host controller and the card is message-based. Each message is represented by one of the following parts:

**Command:** A command starts an operation. The command is transferred serially from the MMC/SD/SDIO host controller to the card on the sdmmc\_cmd line.

**Response:** A response is an answer to a command. The response is sent from the card to the MMC/SD/SDIO host controller. It is transferred serially on the sdmmc\_cmd line.

**Data:** Data are transferred from the MMC/SD/SDIOi host controller to the card or from a card to the MMC/SD/SDIO host controller using the data lines.

**Busy:** The sdmmc\_dat0 signal is maintained low by the card as far as it is programming the data received.

**CRC status:** The CRC result is sent by the card through the sdmmc\_dat0 line when executing a write transfer. In the case of transmission error, occurring on any of the active data lines, the card sends a negative CRC status on sdmmc\_dat0. In the case of successful transmission, over all active data lines, the card sends a positive CRC status on sdmmc\_dat0 and starts the data programming procedure.

### 24.2.2.1 Protocol

There are two types of data transfer:

- Sequential operation
- Block-oriented operation

There are specific commands for each type of operation (sequential or block-oriented).

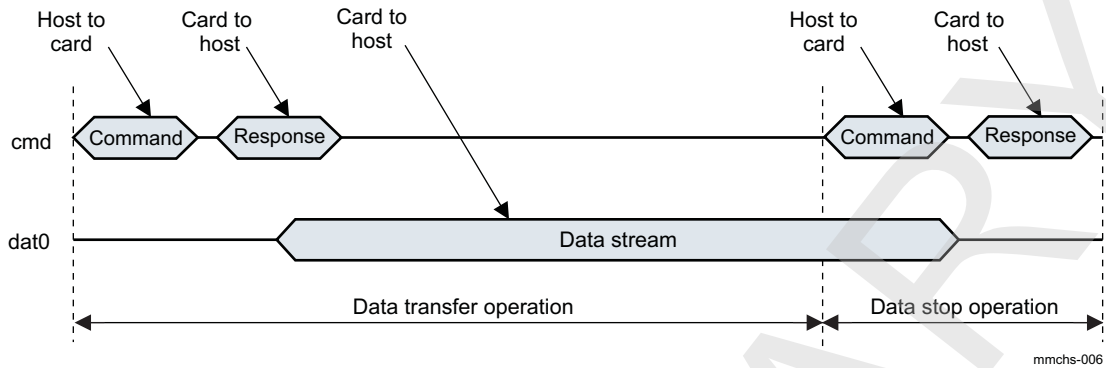
For information about commands and programming sequences supported by the MMC, SD, and SDIO cards, see the *Multimedia Card System Specification*, *SD Memory Card Specifications*, and *SDIO Card Specification (Part E1)*.

Figure 24-6 and Figure 24-7 show how sequential operations are defined. Sequential operation is only for 1-bit transfer and initiates a continuous data stream. The transfer terminates when a stop command follows on the sdmmc\_cmd line.

#### CAUTION

Stream commands are supported only by MMC cards.

**Figure 24-6. Sequential Read Operation (MMC Cards Only)**



**Figure 24-7. Sequential Write Operation (MMC Cards Only)**

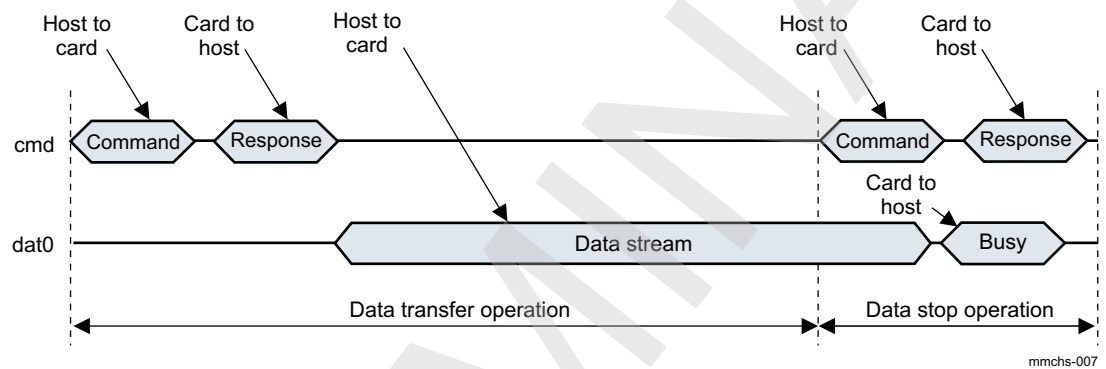
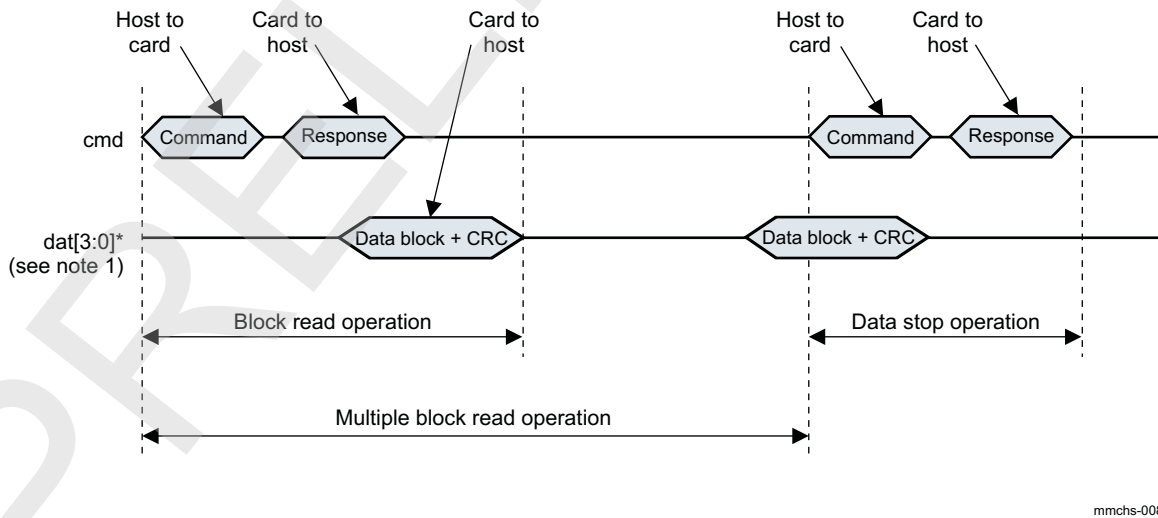
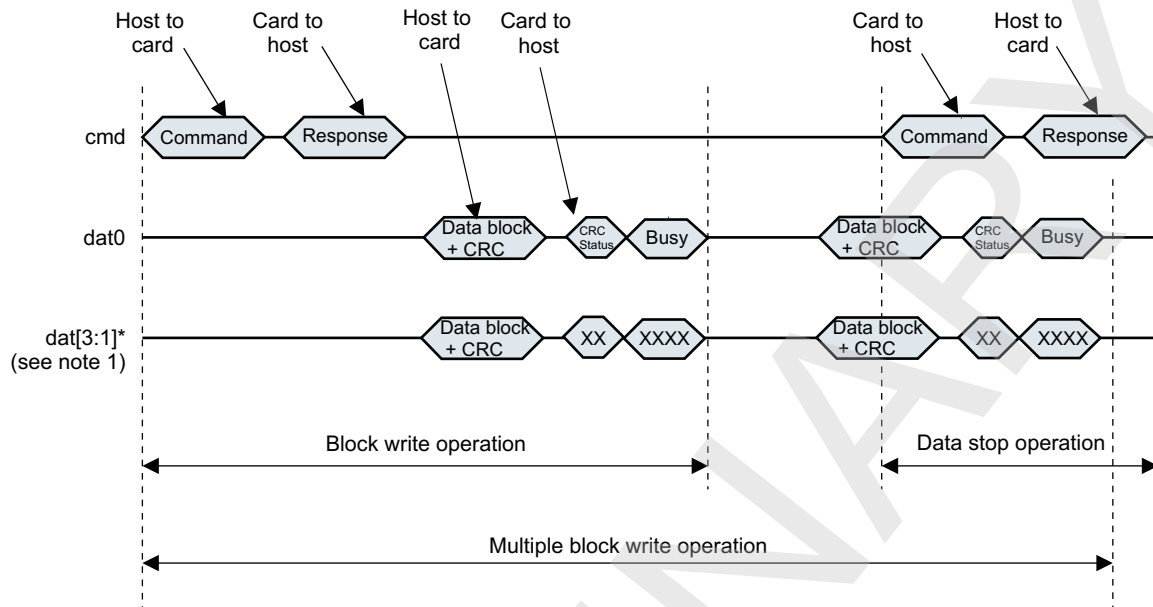


Figure 24-8 and Figure 24-9 show how multiple block-oriented operations are defined. A multiple block-oriented operation sends a data block plus CRC bits. The transfer terminates when a stop command follows on the sdmmc\_i\_cmd line. These operations are available for all kinds of cards.

**Figure 24-8. Multiple Block Read Operation**



**Figure 24-9. Multiple Block Write Operation With Card Busy Signal**

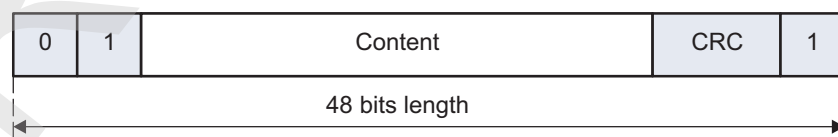
mmchs-009

**NOTE:**

1. The card busy signal is not always generated by the card; the previous examples show a particular case.
2. Software must perform a software reset (set MMCi.MMCHS\_SYSCTL[26] SRD bit to 0x1) after a data time-out to ensure sdmmc\_i\_clk is stopped.
3. For multiblock transfer, and especially for MMC cards, a transfer can be aborted without using a stop command. If a CMD23 is used before data transfer to define the number of blocks that will be transferred, then the transfer stops automatically after the last block (if the MMC card supports this feature).

**24.2.2.2 Data Format****Coding Scheme for Command Token**

Command tokens always start with 0 and end with 1. The second bit is a transmitter bit: 1 for a host command. The content is the command index (coded by 6 bits) and an argument (for example, an address), coded by 32 bits. The content is protected by 7-bit CRC checksum (see [Figure 24-10](#)).

**Figure 24-10. Command Token Format**

mmchs-010

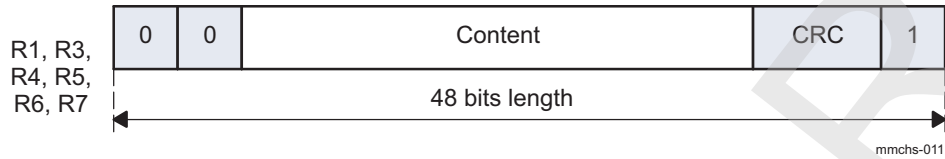
**Coding Scheme for Response Token**

Response tokens always start with 0 and end with 1. The second bit is a transmitter bit: 0 for a card response. The content is different for each type of response (R1, R2, R3, R4, and R5, R6, R7 [for SDIO])

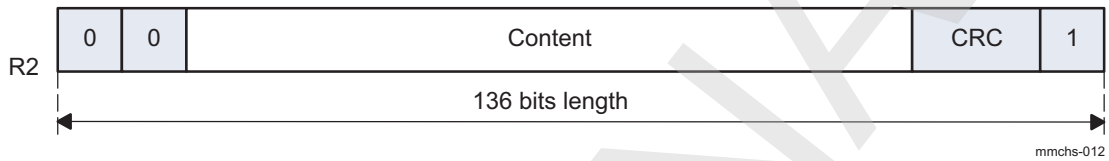


and the content is protected by 7-bit CRC checksum (see [Figure 24-11](#) and [Figure 24-12](#)). Depending on the type of commands sent to the card, the `MMCHS_CMD` register must be configured differently to avoid false CRC or index errors to be flagged on command response (see [Table 24-3](#)). For more information about response types, see the *Multimedia Card System Specification*, *SD Memory Card Specifications*, and *SDIO Card Specification*.

**Figure 24-11. Response Token Format (R1, R3, R4, R5, R6, R7)**



**Figure 24-12. Response Token Format (R2)**



**Table 24-3. Relationship Between Configuration and Name of Response Type**

Response Type MMCI.MMCHS_CMD[17:16] RSP_TYPE	Index Check Enable MMCI.MMCHS_CMD[20] CICE	CRC Check Enable MMCI.MMCHS_CMD[19] CCCE	Name of Response Type
00	0	0	No response
01	0	1	R2
10	0	0	R3 (R4 for SD cards)
10	1	1	R1, R6, R5, R7
11	1	1	R1b, R5b

**Coding Scheme for Data Token**

Data tokens always start with 0 and end with 1 (see [Figure 24-13](#) through [Figure 24-15](#)).

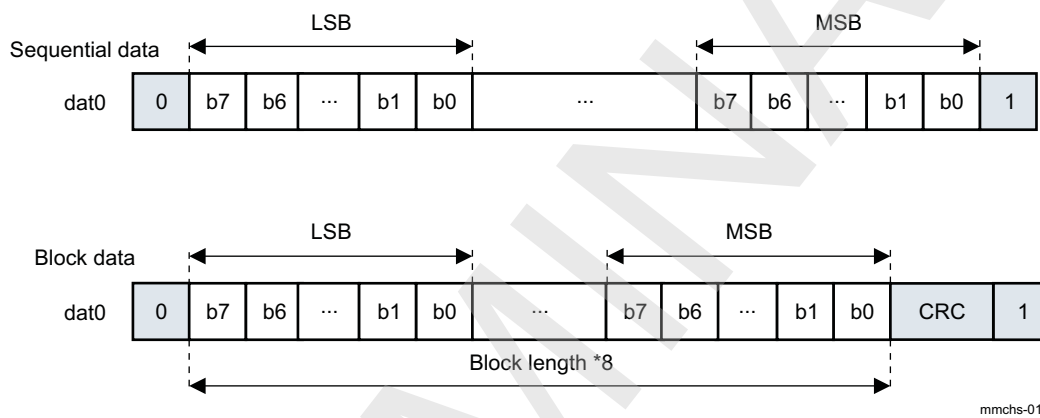
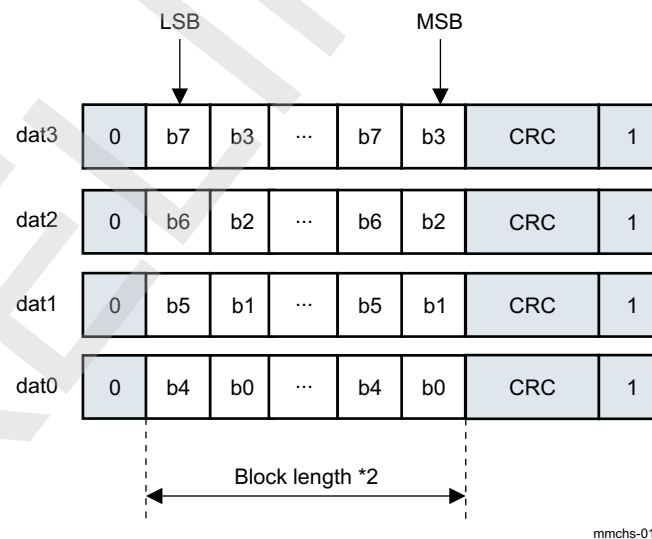
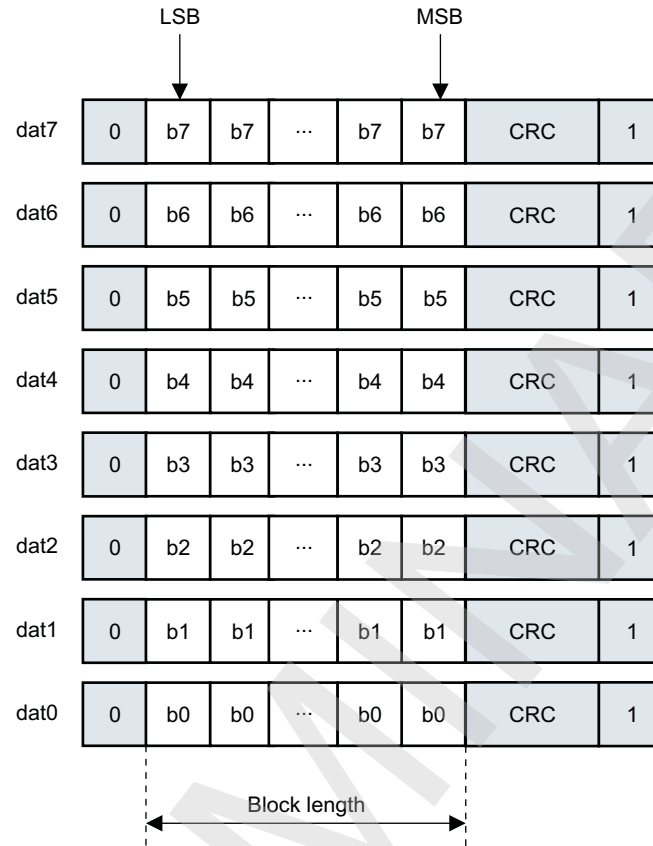
**Figure 24-13. Data Token Format for 1-Bit Transfers****Figure 24-14. Data Token Format for 4-Bit Transfers**

Figure 24-15. Data Token Format for 8-Bit Transfers



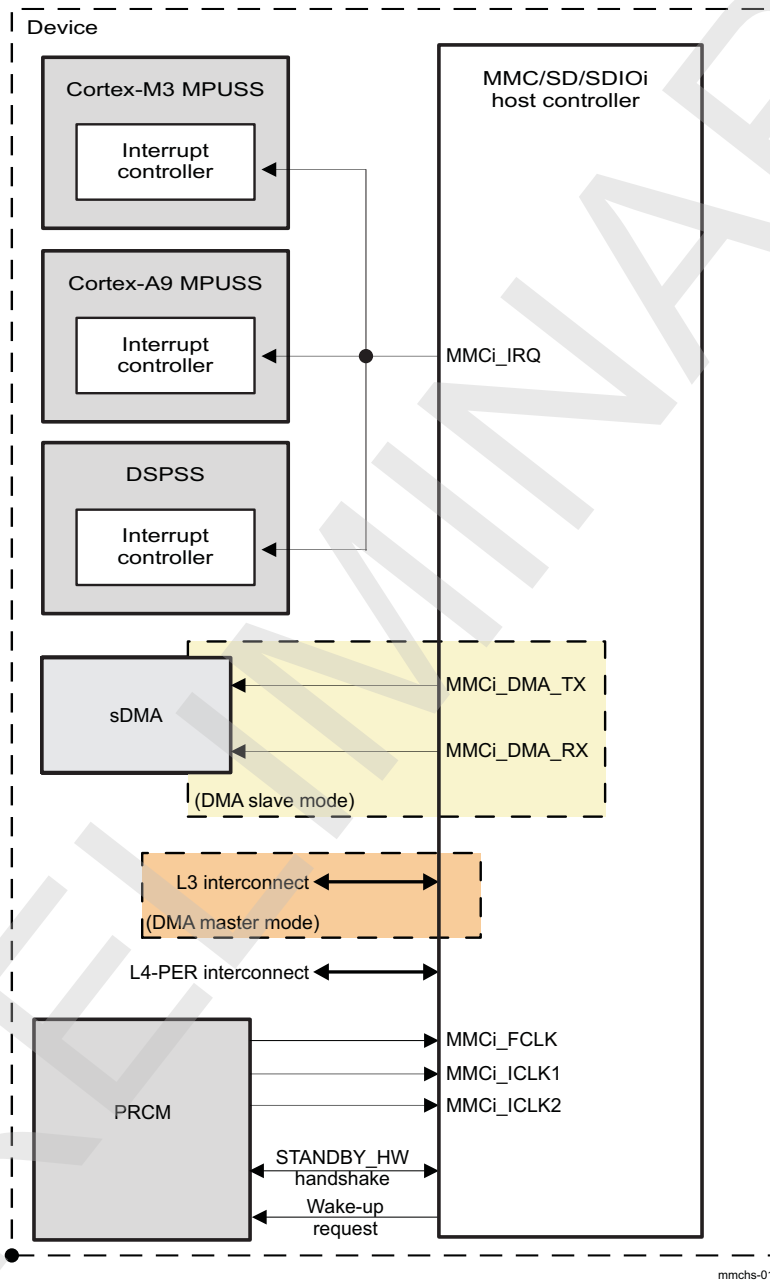
mmchs-015

### 24.3 MMC/SD/SDIO Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 24-16 shows the MMC/SD/SDIO integration.

**Figure 24-16. MMC/SD/SDIO Controllers Integration**



**NOTE:** For more information about the IDLE hardware handshake and the wake-up request, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

Table 24-4 through Table 24-6 summarize the integration of the module in the device.

**Table 24-4. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
HSMMC1	PD_L3_INIT	L3 L4_PER
HSMMC2	PD_L3_INIT	L3 L4_PER
HSMMC3	PD_L4_PER	L4_PER
HSMMC4	PD_L4_PER	L4_PER
HSMMC5	PD_L4_PER	L4_PER

**Table 24-5. Clocks and Resets**

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HSMMC1	MMC1_FCLK	MMC1_FCLK	PRCM	MMC1 function clock
	MMC1_ICLK2	INIT_L4_ICLK	PRCM	MMC1 interface clock
	MMC1_ICLK1	INIT_L3_ICLK	PRCM	MMC1 interface clock
HSMMC2	MMC2_FCLK	MMC2_FCLK	PRCM	MMC2 function clock
	MMC2_ICLK2	INIT_L4_ICLK	PRCM	MMC2 interface clock
	MMC2_ICLK1	INIT_L3_ICLK	PRCM	MMC2 interface clock
HSMMC3	MMC3_ICLK	PER_L4_ICLK	PRCM	MMC3 interface clock
	MMC3_FCLK	PER_48M_FCLK	PRCM	MMC3 function clock
HSMMC4	MMC4_ICLK	PER_L4_ICLK	PRCM	MMC4 interface clock
	MMC4_FCLK	PER_48M_FCLK	PRCM	MMC4 function clock
HSMMC5	MMC5_ICLK	PER_L4_ICLK	PRCM	MMC5 interface clock
	MMC5_FCLK	PER_48M_FCLK	PRCM	MMC5 function clock
Resets				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
HSMMC1	MMC1_RST	L3_INIT_RET_RST	PRCM	L3 reset to MMC1
HSMMC2	MMC2_RST	L3_INIT_RET_RST	PRCM	L3 reset to MMC2
HSMMC3	MMC3_RST	L4_PER_RST	PRCM	L4 reset to MMC3
HSMMC4	MMC4_RST	L4_PER_RST	PRCM	L4 reset to MMC4
HSMMC5	MMC5_RST	L4_PER_RST	PRCM	L4 reset to MMC5

**Table 24-6. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
HSMMC1	MMC1_IRQ	MM_IRQ_50	Cortex-M3 INTC	MMC1 interrupt to Cortex-M3
	MMC1_IRQ	D_IRQ_16	DSP INTC	MMC1 interrupt to DSP
	MMC1_IRQ	MA_IRQ_83	Cortex-A9 INTC	MMC1 interrupt to Cortex-A9
HSMMC2	MMC2_IRQ	MM_IRQ_51	Cortex-M3 INTC	MMC2 interrupt to Cortex-M3
	MMC2_IRQ	D_IRQ_17	DSP INTC	MMC2 interrupt to DSP
	MMC2_IRQ	MA_IRQ_86	Cortex-A9 INTC	MMC2 interrupt to Cortex-A9
HSMMC3	MMC3_IRQ	MM_IRQ_52	Cortex-M3 INTC	MMC3 interrupt to Cortex-M3
	MMC3_IRQ	MA_IRQ_94	Cortex-A9 INTC	MMC3 interrupt to Cortex-A9
HSMMC4	MMC4_IRQ	MM_IRQ_53	Cortex-M3 INTC	MMC4 interrupt to Cortex-M3
	MMC4_IRQ	MA_IRQ_96	Cortex-A9 INTC	MMC4 interrupt to Cortex-A9
HSMMC5	MMC5_IRQ	MM_IRQ_54	Cortex-M3 INTC	MMC5 interrupt to Cortex-M3
	MMC5_IRQ	MA_IRQ_59	Cortex-A9 INTC	MMC5 interrupt to Cortex-A9
DMA Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
HSMMC1	MMC1_DMA_TX	S_DMA_60	sDMA	MMC1 DMA TX
	MMC1_DMA_RX	S_DMA_61	sDMA	MMC1 DMA RX
HSMMC2	MMC2_DMA_TX	S_DMA_46	sDMA	MMC2 DMA TX
	MMC2_DMA_RX	S_DMA_47	sDMA	MMC2 DMA RX
HSMMC3	MMC3_DMA_TX	S_DMA_76	sDMA	MMC3 DMA TX
	MMC3_DMA_RX	S_DMA_77	sDMA	MMC3 DMA RX
HSMMC4	MMC4_DMA_TX	S_DMA_56	sDMA	MMC4 DMA TX
	MMC4_DMA_RX	S_DMA_57	sDMA	MMC4 DMA RX
HSMMC5	MMC5_DMA_TX	S_DMA_58	sDMA	MMC5 DMA TX
	MMC5_DMA_RX	S_DMA_59	sDMA	MMC5 DMA RX

**NOTE:**

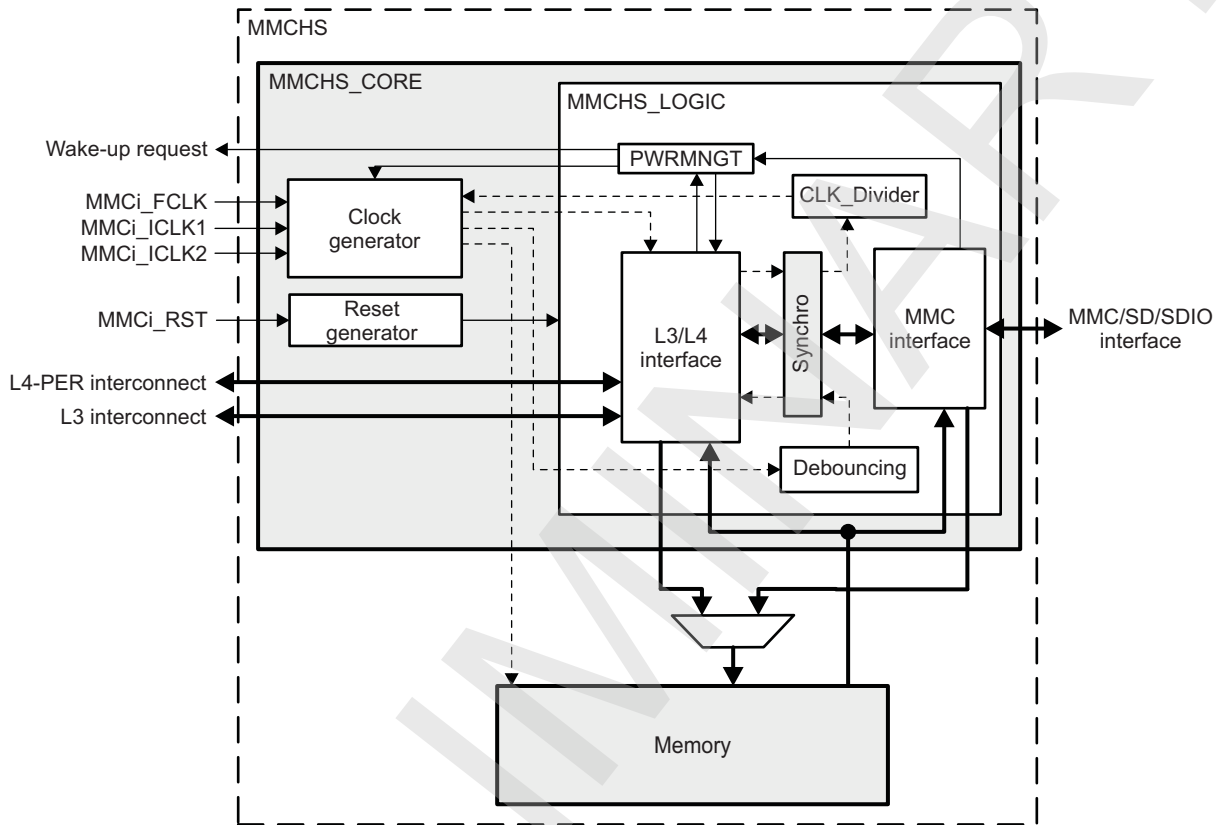
- For a description of the interrupt source, see [Section 24.4.4, Interrupt Requests](#).
- For a description of the DMA source, see [Section 24.4.5, DMA Requests](#).

## 24.4 MMC/SD/SDIO Functional Description

### 24.4.1 Block Diagram

Figure 24-17 is a block diagram of the MMC/SD/SDIOi host controller.

Figure 24-17. MMC/SD/SDIO Diagram



mmchs-017

### 24.4.2 Resets

#### 24.4.2.1 Hardware Reset

The module is reinitialized by the hardware (see Table 24-5 for more information about reset signals).

The MMCHS.MMCHS\_SYSSTATUS[0] RESETDONE bit can be monitored by software to check whether the module is ready to use after a hardware reset.

**NOTE:** The functional clock (MMCi\_FCLK) and interface clock (MMCi\_ICLK) must be provided to the module to allow the RESETDONE status bit to be set.

The debounce clock (MMCi\_32K) must be active to reset the module correctly.

This hardware reset signal has a global reset action on the module. All configuration registers and all state-machines are reset in all clock domains.

#### 24.4.2.2 Software Reset

The module is reinitialized by software through the MMCHS.MMCHS\_SYSCONFIG[1] SOFTRESET bit. This bit has the same action on the module logic as the hardware signal (MMCi\_RST), with the following exceptions:

- Debounce logic

- MMCI.MMCHS\_PSTATE, MMCI.MMCHS\_CAPA, and MMCI.MMCHS\_CUR\_CAPA registers (see the corresponding register description)

The SOFTRESET bit is active high. The bit is automatically reinitialized to 0 by the hardware. The MMCI.MMCHS\_SYSCTL[24] SRA bit has the same action on the design as the SOFTRESET bit.

The MMCI.MMCHS\_SYSSTATUS[0] RESETDONE bit can be monitored by software to check whether the module is ready to use after a software reset.

Moreover, two partial software reset bits are provided:

- MMCI.MMCHS\_SYSCTL[26] SRD bit
- MMCI.MMCHS\_SYSCTL[25] SRC bit

These two reset bits are useful to reinitialize data or command processes, respectively, in case of line conflict. When set to 1, a reset process is automatically released when the reset completes:

- The MMCI.MMCHS\_SYSCTL[26] SRD bit resets all finite state-machines and status management that handle data transfers on the interface and functional sides.
- The MMCI.MMCHS\_SYSCTL[25] SRC bit resets all finite state-machines and status management that handle command transfers on the interface and functional sides.

### 24.4.3 Power Management

The MMC/SD/SDIO host controller can enter into different modes and save power:

- Normal mode
- Idle mode

The two modes are mutually exclusive (the module can be in normal mode or in idle mode). The MMC/SD/SDIO host controller is compliant with the handshake protocol of the PRCM module.

When the MMC/SD/SDIO power domain is off, the only way to wake up the power domain and different MMC/SD/SDIO clocks is to monitor the state of the sdmmci\_dat1 input pin through a different GPIO line for each MMC/SD/SDIO interface (for more information, see [Section 25.1, GPIO Overview](#), in [Chapter 25, GPIO](#)).

#### Normal Mode

The autogating of interface and functional clocks occurs when the following conditions are met:

- The MMCI.MMCHS\_SYSCONFIG[0] AUTOIDLE bit is set to 1.
- There is no transaction on the MMC interface.

The autogating of interface and functional clocks stops when the following conditions are met:

- A register access occurs through the L4 interconnect.
- A wake-up event occurs (an interrupt from an SDIO card).
- A transaction on the MMC/SD/SIO interface starts.

Then the MMC/SD/SDIO host controller enters into low-power state (MMCI\_ICLK clock autogated) even if the MMCI.MMCHS\_SYSCONFIG[0] AUTOIDLE bit is set to 0.

The functional clock is internally switched off and only interconnect read and write accesses are allowed.

#### Idle Mode

The MMCI\_ICLK and MMCI\_FCLK clocks provided to MMC/SD/SDIO are switched off upon a PRCM module request. They are switched back upon module request.

The MMC/SD/SDIO host controller complies with the handshaking protocol of the PRCM module:

- Idle request from the system power manager
- Idle acknowledgment from the MMC/SD/SDIO host controller
- Wake-up request from the MMC/SD/SDIO host controller

The idle acknowledgment varies according to the MMCI.MMCHS\_SYSCONFIG[4:3] SIDLEMODE bit field:

- 0x0: Force-idle mode. The MMC/SD/SDIO host controller acknowledges the system power manager request unconditionally.



- 0x1: No-idle mode. The MMC/SD/SDIO host controller ignores the system power manager request and behaves normally as if the request was not asserted.
- 0x2: Smart-idle mode. The MMC/SD/SDIO host controller acknowledges the system power manager request according to its internal state.
- 0x3: Smart-idle wake-up-capable mode. The MMC/SD/SDIO host controller acknowledges the system power manager request according to its internal state. However, the module may generate wake-up events when in IDLE state ( related to IRQ or DMA requests)

During the smart-idle mode period, the MMC/SD/SDIO host controller acknowledges that the MMCi\_ICLK and MMCi\_FCLK clocks may be switched off, regardless of the value set in the MMCi.MMCHS\_SYSCONFIG[9:8] CLOCKACTIVITY bit field.

**Transition From Normal Mode to Smart-Idle Mode**

Smart-idle mode is enabled when the MMCi.MMCHS\_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x2 or 0x3.

The MMC/SD/SDIOi host controller goes into idle mode when the PRCM issues an idle request, according to its internal activity.

The MMC/SD/SDIO host controller acknowledges the idle request from the PRCM after ensuring the following:

- The current multi- or single-block transfer is complete.
- Any interrupt or DMA request is asserted.
- There is no card interrupt on the sdmmc\_i\_dat1 signal.

As long as the MMC/SD/SDIOi controller does not acknowledge the idle request, if an event occurs, the MMC/SD/SDIOi host controller can still generate an interrupt or a DMA request. In this case, the module ignores the idle request from the PRCM module.

As soon as the MMC/SD/SDIOi controller acknowledges the idle request from the PRCM module:

- If smart-idle mode: The module does not assert any new interrupt or DMA request.
- If smart-idle wake-up-capable mode: The module may generate wake-up events related to an interrupt or DMA request.

**Wake-Up Event in Smart-Idle Mode**

The wake-up feature is enabled when the following enable wake-up bits are set:

- The MMCi.MMCHS\_SYSCONFIG[2] ENAWAKEUP bit is set to 0x1.
- The MMCi.MMCHS\_HCTL[24] IWE bit is set to 0x1.
- The MMCi.MMCHS\_IE[8] CIRQ\_ENABLE bit is set to 0x1.

The wakeup is generated only in smart-idle mode, when the module is in idle mode.

Table 24-7 lists the supported cases in smart-idle mode.

**Table 24-7. Smart-Idle Mode and Wake-Up Capabilities**

Mode	MMCi_ICLK clock	MMCi_FCLK clock	Wake-Up Event
Card interrupt	May be switched off <sup>(1)</sup>	May be switched off <sup>(1)</sup>	The module sends an asynchronous wake-up request on detection of a card interrupt on sdmmc_i_dat1 signal.

<sup>(1)</sup> The MMC/SD/SDIOi host controller assumes that both clocks may be switched off, regardless of the value set in the MMCi.MMCHS\_SYSCONFIG[9:8] CLOCKACTIVITY bit field.

**Transition From Smart-Idle Mode to Normal Mode**

The MMC/SD/SDIO host controller detects the end of the idle period when the PRCM deasserts the idle request.

For the wake-up event, there is a corresponding interrupt status in the MMCi.MMCHS\_STAT register. The MMC/SD/SDIOi host controller operates the conversion between the wake-up and interrupt (or DMA request) upon exit from smart-idle mode, if the associated enable bit is set in the MMCi.MMCHS\_ISE register.

Interrupts and wake-up events have independent enable and disable controls, accessible through the MMCI.MMCHS\_HCTL and MMCI.MMCHS\_ISE registers. The overall consistency must be ensured by software.

The interrupt status register MMCI.MMCHS\_STAT is updated in the CIRQ bit with the event that caused the wakeup when the MMCI.MMCHS\_IE[8] CIRQ\_ENABLE associated bit is enabled.

Then, the wake-up event at the origin of the transition from smart-idle mode to normal mode is converted into its corresponding interrupt or DMA request. (The MMCI.MMCHS\_STAT register is updated and the status of the interrupt signal changes.)

When the idle request from the PRCM module is deasserted, the module switches back to normal mode. The module is fully operational.

### Force-Idle Mode

Force-idle mode is enabled when the MMCI.MMCHS\_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x0.

Force-idle mode is an idle mode in which the MMC/SD/SDIOi host controller responds unconditionally to the idle request from the PRCM module. Moreover, in this mode, the MMC/SD/SDIOi host controller unconditionally deasserts interrupts and DMA request lines asserted.

The transition from normal mode to force-idle mode does not affect the bits of the MMCI.MMCHS\_STAT register.

In force-idle mode, the interrupt and DMA request lines are deasserted. MMCI\_ICLK and MMCI\_FCLK can be switched off.

#### CAUTION

In force-idle mode, an idle request from the PRCM module during a command or a data transfer can lead to an unexpected and unpredictable result. When the module is idle, any access to the module generates an error as long as the MMCI\_ICLK clock is alive.

The module exits force-idle mode when the PRCM module deasserts the idle request. Then the module switches back to normal mode. The module is fully operational. Interrupt and DMA request lines are optionally asserted one clock cycle later.

### Local Power Management

Table 24-8 describes the power-management features available for the MMC/SD/SDIOi modules.

**NOTE:** For information about source clock gating and a description of the sleep/wake-up transitions, see Section 3.1.1.1, *Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

**Table 24-8. Local Power-Management Features**

Feature	Registers	Description
Clock auto gating	MMCHS_SYSCONFIG AUTOIDLE bit	This bit allows a local power optimization inside the module by gating the MMCI_ICLK clock upon the interface activity, or gating the MMCI_FCLK clock upon the internal activity.
Slave idle modes	MMCHS_SYSCONFIG SIDLEMODE bit field	Force-idle, no-idle and smart-idle modes are available.
Clock activity	MMCHS_SYSCONFIG CLOCKACTIVITY bit field	For configuration details, see Table 24-9.
Master standby modes	MMCHS_SYSCONFIG STANDBYMODE bit field	Force-idle, no-idle, smart-idle modes are available.
Global wake-up enable	MMCHS_SYSCONFIG ENAWAKEUP bit	This bit enables the wake-up feature at the module level.

**Table 24-8. Local Power-Management Features (continued)**

Feature	Registers	Description
Wake-up sources enable	MMCHS_HCTL register	This register holds one active-high enable bit per event source that is able to generate a wake-up signal.

**Table 24-9. Clock Activity Settings**

CLOCKACTIVITY Values	Clock State When Module is in IDLE State		Features Available When Module is in IDLE State	Wake-Up Events
	MMCi_ICLK	MMCi_FCLK		
00	OFF	OFF	None	Card interrupt
10	OFF	ON	None	
01	ON	OFF	None	
11	ON	ON	All	

**CAUTION**

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the CLOCKACTIVITY and MMCi clock PRCM control bits. For a description of the ClockActivity feature, see [Section 3.1.1.1.2, Module Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**24.4.4 Interrupt Requests**

Several internal module events can generate an interrupt. Each interrupt has a status bit, an interrupt enable bit, and a signal status enable:

- The status of each type of interrupt is automatically updated in the MMCi.MMCHS\_STAT register; it indicates which service is required.
- The interrupt status enable bits of the MMCi.MMCHS\_IE register enable or disable the automatic update of the MMCi.MMCHS\_STAT register on an event-by-event basis.
- The interrupt signal enable bits of the MMCi.MMCHS\_ISE register enable or disable the transmission of an interrupt request on the interrupt line MMCi\_IRQ (from the MMC/SD/SDIOi host controller to the MPU subsystem INTC) on an event-by-event basis.

If an interrupt status is disabled in the MMCi.MMCHS\_IE register, then the corresponding interrupt request is not transmitted, and the value of the corresponding interrupt signal enable in the MMCi.MMCHS\_ISE register is ignored.

When an interrupt event occurs, the corresponding status bit is automatically set to 0x1 (the MMC/SD/SDIOi host controller updates the status bit) in the MMCi.MMCHS\_STAT register. If later a mask is applied on the interrupt in the MMCi.MMCHS\_ISE register, the interrupt request is deactivated.

When the interrupt source has not been serviced, if the interrupt status is cleared in the MMCi.MMCHS\_STAT register and the corresponding mask is removed from the MMCi.MMCHS\_ISE register, the interrupt status is not asserted again in the MMCi.MMCHS\_STAT register and the MMC/SD/SDIOi host controller does not transmit an interrupt request.

**CAUTION**

If the buffer write ready (BWR) interrupt or the buffer read ready (BRR) only interrupt are not serviced and are cleared in the MMCi.MMCHS\_STAT register, and the corresponding mask is removed, then the MMC/SD/SDIOi host controller waits for the service of the interrupt without updating the status MMCi.MMCHS\_STAT register or transmitting an interrupt request.

[Table 24-10](#) lists the event flags, and their mask, that can cause module interrupts.

Table 24-10. Events

Event Flag	Event Mask	Map to	Description
MMCHS_STAT[29] BADA	MMCHS_IE[29] BADA_ENABLE	MMCi_IRQ	<p>Bad access to data space. This bit is set automatically to indicate a bad access to buffer when not allowed:</p> <p>This bit is set during a read access to the data register (MMCHS_DATA) while buffer reads are not allowed (MMCHS_PSTATE[11] BRE = 0)</p> <p>This bit is set during a write access to the data register (MMCHS_DATA) while buffer writes are not allowed (MMCHS_STATE[10] BWE = 0)</p>
MMCHS_STAT[28] CERR	MMCHS_IE[28] CERR_ENABLE	MMCi_IRQ	Card error. This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5, or R5b. Only bits referenced as type E (error) in the status field in the response can set a card status error. An error bit in the response is flagged only if the corresponding bit in the card status response error MMCHS_CSRE is set. There is no card error detection for the autoCMD12 command.
MMCHS_STAT[25] ADMAE	MMCHS_IE[25] ADMAE_ENABLE	MMCi_IRQ	ADMA error. This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA error status register. In addition, the host controller generates this interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state.
MMCHS_STAT[24] ACE	MMCHS_IE[24] ACE_ENABLE	MMCi_IRQ	Auto CMD12 error. This bit is set automatically when one of the bits in the auto CMD12 error status register changes from 0 to 1.
MMCHS_STAT[22] DEB	MMCHS_IE[22] DEB_ENABLE	MMCi_IRQ	Data end bit error. This bit is set automatically when detecting a 0 at the end bit position of read data on the DAT line or at the end position of the CRC status in write mode.
MMCHS_STAT[21] DCRC	MMCHS_IE[21] DCRC_ENABLE	MMCi_IRQ	Data CRC error. This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status different of a position 010 token during a block write command.
MMCHS_STAT[20] DTO	MMCHS_IE[20] DTO_ENABLE	MMCi_IRQ	Data time-out error. This bit is set automatically according to the following conditions: <ul style="list-style-type: none"> <li>– Busy time-out for R1b, R5b response type</li> <li>– Busy time-out after write CRC status</li> <li>– Write CRC status time-out</li> <li>– Read data time-out</li> </ul>
MMCHS_STAT[19] CIE	MMCHS_IE[19] CIE_ENABLE	MMCi_IRQ	Command index error. This bit is set automatically when the response index differs from the corresponding command index previously emitted. The check is enabled through the MMCHS_CMD[20] CICE bit.
MMCHS_STAT[18] CEB	MMCHS_IE[18] CEB_ENABLE	MMCi_IRQ	Command end bit error. This bit is set automatically when detecting a 0 at the end bit position of a command response.
MMCHS_STAT[17] CCRC	MMCHS_IE[17] CCRC_ENABLE	MMCi_IRQ	Command CRC error. This bit is set automatically when there is a CRC7 error in the command response. CRC check is enabled through the MMCHS_CMD[19] CCCE bit.
MMCHS_STAT[16] CTO	MMCHS_IE[16] CTO_ENABLE	MMCi_IRQ	Command time-out error. This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within five clock cycles, the time-out is still detected at 64 clock cycles.

**Table 24-10. Events (continued)**

Event Flag	Event Mask	Map to	Description
MMCHS_STAT[15] ERRI	MMCHS_IE[15] ERRI_ENABLE	MMCi_IRQ	Error interrupt. If any of the bits in the error interrupt status register (MMCHS_STAT[24:15]) are set, this bit is set to 1.
MMCHS_STAT[10] BSR	MMCHS_IE[10] BSR_ENABLE	MMCi_IRQ	Boot status received interrupt. This bit is set automatically when the MMCHS_CON[18] BOOT_CF0 bit is set to 0x1 or 0x2 and a boot status is received on the dat0 line. This interrupt is useful only for the MMC card.
MMCHS_STAT[8] CIRQ	MMCHS_IE[8] CIRQ_ENABLE	MMCi_IRQ	Card interrupt. This bit is used only for SD, SDIO, and CE-ATA cards. In 1-bit mode, the interrupt source is asynchronous (can be a source of asynchronous wakeup). In 4-bit mode, the interrupt source is sampled during the interrupt cycle. In CE-ATA mode, the interrupt source is detected when the card drives the CMD line to 0 during one cycle after data transmission end.
MMCHS_STAT[5] BRR	MMCHS_IE[5] BRR_ENABLE	MMCi_IRQ	Buffer read ready This bit is set automatically during a read operation to the card (see class 2 block-oriented read commands) when one block specified by the MMCHS_BLK[10:0] BLEN bit field is completely written in the buffer. It indicates that the memory card has filled out the buffer and that the LH needs to empty the buffer by reading it.
MMCHS_STAT[4] BWR	MMCHS_IE[4] BWR_ENABLE	MMCi_IRQ	Buffer write ready This bit is set automatically during a write operation to the card (see class 4 block-oriented write command) when the host can write a complete block as specified by the MMCHS_BLK[10:0] BLEN bit field. It indicates that the memory card has emptied one block from the buffer and that the LH can write one block of data into the buffer.
MMCHS_STAT[3] DMA	MMCHS_IE[3] DMA_ENABLE	MMCi_IRQ	DMA interrupt: This status is set when an interrupt is required in the ADMA instruction and after the data transfer completes.
MMCHS_STAT[2] BGE	MMCHS_IE[2] BGE_ENABLE	MMCi_IRQ	Block gap event. When a stop at block gap is requested (MMCHS_HCTL[16] SBGR), this bit is automatically set when transaction is stopped at the block gap during a read or write operation.
MMCHS_STAT[1] TC	MMCHS_IE[1] TC_ENABLE	MMCi_IRQ	Transfer completed. This bit is always set when a read/write transfer is complete or between two blocks when the transfer is stopped because of a stop at block gap request (MMCHS_HCTL[16] SBGR). - In read mode: This bit is automatically set on completion of a read transfer (MMCHS_PSTATE[9] RTA). - In write mode: This bit is automatically set on completion of the DAT line use (MMCHS_PSTATE[2] DLA).

**Table 24-10. Events (continued)**

Event Flag	Event Mask	Map to	Description
MMCHS_STAT[0] CC	MMCHS_IE[0] CC_ENABLE	MMCi_IRQ	Command complete This bit is set when a 1-to-0 transition occurs in the register command inhibit (MMCHS_PSTATE[0] CMDI). If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command time-out error (MMCHS_STAT[16] CTO) has higher priority than command complete (MMCHS_STAT[0] CC). If a response is expected but none is received, then a command time-out error is detected and signaled, instead of the command complete interrupt.

**NOTE:** To send an interrupt request to the MMCi\_IRQ line, the mask/unmask bit must be set in the MMCi.MMCHS\_IE and MMCi.MMCHS\_ISE registers.

The MMC/SD/SDIOi host controller supports interrupt-driven operation and polling.

#### 24.4.4.1 Interrupt-Driven Operation

An interrupt enable bit must be set in the MMCi.MMCHS\_IE register to enable the module internal source of interrupt.

When an interrupt event occurs, the single interrupt line is asserted and the LH must:

1. Read the MMCi.MMCHS\_STAT register to identify which event occurred.
2. Write 1 into the corresponding bit of the MMCi.MMCHS\_STAT register to clear the interrupt status and release the interrupt line (if a read is done after this write, this returns 0).

**NOTE:** In the MMCi.MMCHS\_STAT register, the card interrupt (CIRQ) and error interrupt (ERRI) bits cannot be cleared.

The MMCi.MMCHS\_STAT[8] CIRQ status bit must be masked by disabling the MMCi.MMCHS\_IE[8] CIRQ\_ENABLE bit (set to 0x0), and then the interrupt routine must clear the SDIO interrupt source in the SDIO card common control register (CCCR).

The MMCi.MMCHS\_STAT[15] ERRI bit is automatically cleared when all status bits in the MMCi.MMCHS\_STAT[31:16] bit field are cleared.

#### 24.4.4.2 Polling

When the interrupt capability of an event is disabled in the MMCi.MMCHS\_ISE register, the interrupt line is not asserted:

- Software can poll the status bit in the MMCi.MMCHS\_STAT register to detect when the corresponding event occurs.
- Writing 1 into the corresponding bit of the MMCi.MMCHS\_STAT register clears the interrupt status and does not affect the interrupt line state.

**NOTE:** Refer to the previous note concerning CIRQ and ERRI bits clearing.

#### 24.4.5 DMA Modes

Two DMA managements can be used to load data from memory to the internal buffer of the controller (or vice versa). These modes are exclusive and depend on the module integration.

- DMA master mode

DMA master mode is selected by setting the [MMCHS\\_CON\[20\] DMA\\_MNS](#) bit to 1. In this case, the controller has direct access to data using a specific algorithm called ADMA2 (prevents the system from being interrupted). Data are exchanged using the L3 master interface, which supports burst accesses to maximize throughput.

---

**NOTE:** This mode is supported only by modules connected to the L3 interconnect. See [Section 24.1](#) for more information and/or check the value of the [MMCHS\\_HL\\_HWINFO\[0\] MADMA\\_EN](#) bit.

This mode is available for modules MMC1 and MMC2.

---

- **DMA slave mode**

DMA slave mode is selected by setting the [MMCHS\\_CON\[20\] DMA\\_MNS](#) bit to 0. In this case, the controller is slave on the DMA transaction managed by two separated requests (MMCi\_DMA\_TX and MMcI\_DMA\_RX).

---

**NOTE:** This mode is implied by modules that are not connected to the L3 interconnect (regardless of the value of the [MMCHS\\_CON\[20\] DMA\\_MNS](#) bit). See [Section 24.1](#) for more information and/or to check the value of the [MMCHS\\_HL\\_HWINFO\[0\] MADMA\\_EN](#) bit.

This mode is available for modules MMC3, MMC4, and MMC5.

---

#### 24.4.5.1 MASTER DMA Operations

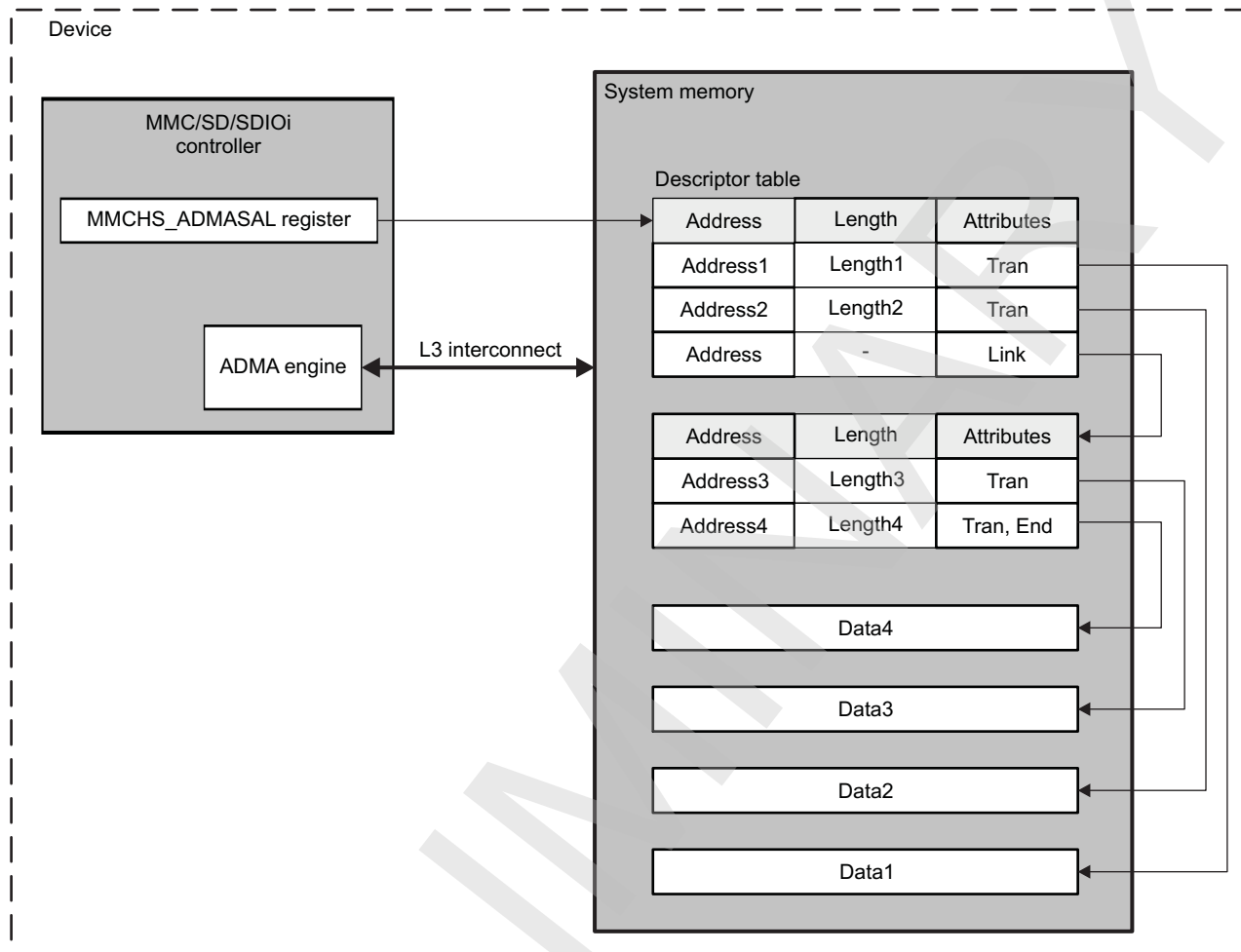
The MMC/SD/SDIOi host controller has direct access to the internal data. This feature is called advanced DMA (ADMA). It follows a specific algorithm (ADMA2) defined by an instruction in memory that starts at an address previously loaded in the [MMCHS\\_ADMASAL](#) register before any data command issued to the MMC card. Only 32-bit address spacing is supported by the controller for data storage.

---

**NOTE:** This mode is supported only by modules connected to the L3 interconnect. See [Section 24.1](#) for more information and/or to check the value of the [MMCHS\\_HL\\_HWINFO\[0\] MADMA\\_EN](#) bit.

---

These instructions must be loaded by software in a 32-bit-addressed descriptor table in system memory, as shown in [Figure 24-18](#). In this case the [MMCHS\\_ADMASAL](#) register is used as the program address pointer

**Figure 24-18. ADMA Block Diagram Overview**

mmchs-018

**24.4.5.1.1 Descriptor Table Description**

Each descriptor line contains an address, a length, and attributes fields. The attributes define which operation will be processed. Every descriptor line is a 64-bit width register that is fetched in the controller using the L3 master interface, and requires two 32-bit accesses to memory.

Table 24-11 shows the structure of a descriptor line.

**Table 24-11. Descriptor Line Overview**

Address Field		Length		Reserved		Attributes					
63	32	31	16	15	6	5	4	3	2	1	0
32-bit Address		16-bit Length		0x0		Act2	Act1	0	Int	Ent	Valid

The attribute of the descriptor line is divided into two parts:

- Attributes[5:4]: The action to be processed by the ADMA engine
- Attributes[3:0]: Additional parameters characterizing the behavior of the ADMA engine

Table 24-12 describes the available actions of a descriptor line.



**Table 24-12. Available Actions of a Descriptor Line**

Act2	Act1	Symbol	Comment	Operation
0	0	Nop	No operation	Do not execute the current line and go to the next line.
0	1	Rsv	Reserved	Reserved action. Behaves the same as the Nop command.
1	0	Tran	Transfer data	Transfer data of one descriptor line.
1	1	Link	Link descriptor	Link to another descriptor.

Table 24-13 describes the additional parameters of a descriptor line.

**Table 24-13. Additional Parameters of a Descriptor Line**

Bit	Description
Valid	Valid = 1 indicates that this descriptor line is effective. If Valid = 0, an ADMA error interrupt is generated and the ADMA is stopped. This prevents runaways.
End	End = 1 indicates the end of a descriptor. The transfer-complete interrupt is generated when the operation of the descriptor line is complete.
Int	Int = 1 generates a DMA interrupt when the operation of the descriptor line is complete.

#### 24.4.5.1.2 Requirements for Descriptors

The following sections discuss restrictions and tips on how to correctly configure the descriptors to be used by the ADMA engine.

##### 24.4.5.1.2.1 Data Length

There are three requirements to program descriptors:

- The minimum unit of address is 4 bytes.
- The maximum data length of each descriptor line is less than 64KB.
- Total length = Length<sub>1</sub> + Length<sub>2</sub> + Length<sub>3</sub> + ... + Length<sub>n</sub> must be a multiple of the block size.

If the total length of a descriptor is not a multiple of the block size, data transfer with the ADMA engine may not have been terminated. In this case, the controller returns a data time-out event and the transfer is aborted.

The block count register (the [MMCHS\\_BLK\[31:16\]](#) NBLK bit field) is defined as 16 bits and limits data transfers to a maximum of 65,535 blocks. If the ADMA data transfer size is less than or equal to the 65,535-block transfer, the block count register can be used. In this case, the total length of the descriptor table must be equivalent to "block size" by "block count." If the ADMA data transfer is greater than 65,535 blocks, the block count register must be disabled by setting the block count enable bit (the [MMCHS\\_CMD\[1\]](#) BCE bit) to 0. In this case, the length of the data transfer is not designated by the block count but by the descriptor table.

---

**NOTE:** The timing for detecting the last block on the SD bus may differ, which affects control of the read transfer active ([MMCHS\\_PSTATE\[9\]](#) RTA), write transfer active ([MMCHS\\_PSTATE\[8\]](#) WTA), and DAT line active ([MMCHS\\_PSTATE\[2\]](#) DLA) bits. In case of a read operation, more blocks than required may be read from the card. The host driver must ignore an out-of-range error if the read operation is for the last block of the memory area.

---

##### 24.4.5.1.2.2 Supported Features

The ADMA engine does not support the suspend/resume function. However, the stop and continue functions are available.

When the stop-at-block-gap-request ([MMCHS\\_HCTL\[16\]](#) SBGR bit) is set during the ADMA operation, the block gap event interrupt is generated when the ADMA is stopped at block gap ([MMCHS\\_STAT\[2\]](#) BGE bit). The host controller must stop the ADMA read operation by using read wait or by stopping the SD clock. While stopping ADMA, SD commands cannot be issued.

### 24.4.5.1.2.3 Error Generation

When an error occurs during an ADMA transfer, the ADMA operation is stopped and the ADMA error interrupt is generated. The ADMA error state field (the `MMCHS_ADMAES[1:0]` AES bit field) holds the state of the ADMA when it is stopped. Software can identify the erroneous descriptor line by using the following method:

- if ADMA stopped at `ST_FDS` state, the ADMA system address register (`MMCHS_ADMASAL`) points to the erroneous descriptor line
- If ADMA stopped at `ST_TFR` or `ST_STOP` state, the ADMA system address register points to the descriptor line following the erroneous one.

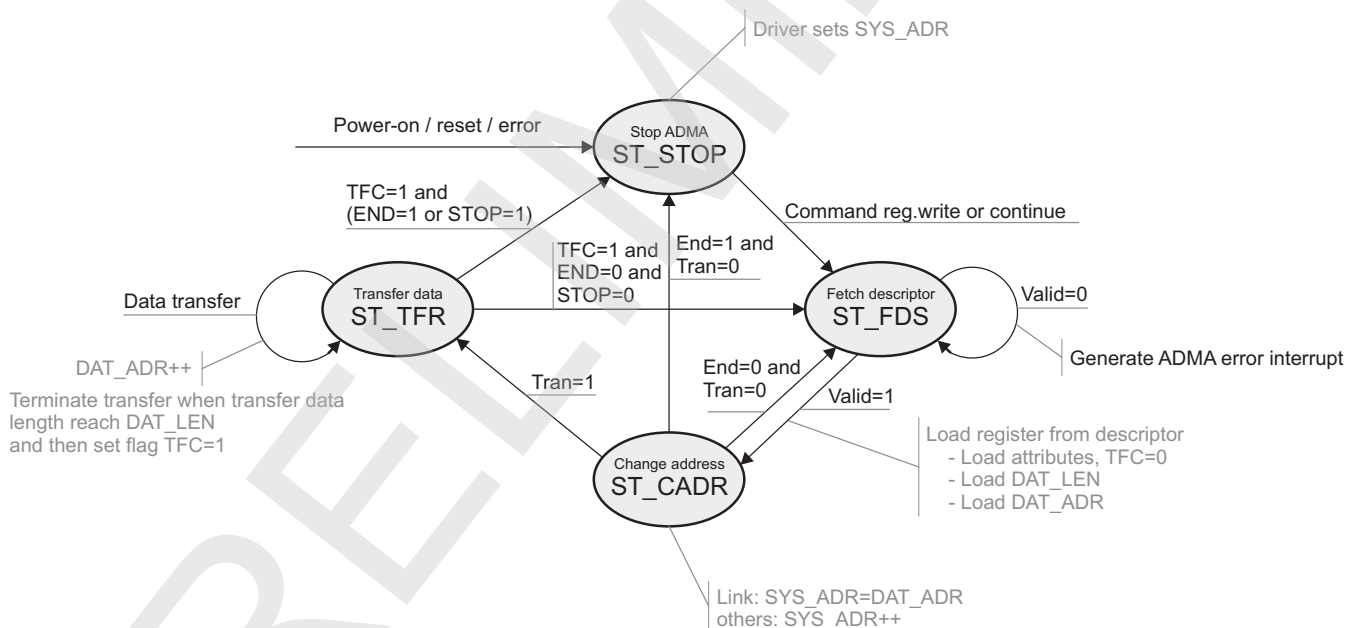
### 24.4.5.1.3 Advanced DMA Description

The ADMA is a DMA controller embedded in each eMMC controller. It can be seen as a small sequencer that fetches a descriptor line and executes the corresponding action. The base address of the descriptor table is stored in the `MMCHS_ADMASAL` register.

**NOTE:** Software must write the base address of the descriptor table in the `MMCHS_ADMASAL` register before the first use of the ADMA engine.

The ADMA program is executed according to descriptor attributes (see [Section 24.4.5.1.1, Descriptor Table Description](#)) and a finite state-machine (FSM), as shown in [Figure 24-19](#).

**Figure 24-19. ADMA Finite State-Machine**



mmchs-019

[Table 24-14](#) describes each state of the ADMA FSM.

**Table 24-14. ADMA2 States Description**

State Name	Operation
<code>ST_FDS</code> (fetch descriptor)	ADMA2 fetches a descriptor line and sets parameters in internal registers. It then goes to the <code>ST_CADR</code> state.
<code>ST_CADR</code> (change address)	Link operation loads another descriptor address to the ADMA system address register ( <code>MMCHS_ADMASAL</code> ). In other operations, the ADMA system address register is incremented to point to the next descriptor line. If <code>End = 0</code> , go to <code>ST_FDS</code> state. <b>NOTE:</b> ADMA2 is not stopped at this state if some errors occur.

**Table 24-14. ADMA2 States Description (continued)**

State Name	Operation
ST_TFR (transfer data)	Data transfer of one descriptor line is executed between system memory and the SD card: <ul style="list-style-type: none"> <li>• if data transfer continues (End = 0) go to ST_FDS state.</li> <li>• if data transfer completes, go to ST_STOP state.</li> </ul>
ST_STOP (stop DMA)	ADMA2 stays in this state in the following cases: <ul style="list-style-type: none"> <li>• After power on reset (POR) or software reset</li> <li>• All descriptor data transfers are complete.</li> </ul> If a new ADMA operation is stated by writing the command register, go to ST_FDS state.

Table 24-15 gives the description of each symbol used in the ADMA FSM flow chart (see Figure 24-19).

**Table 24-15. ADMA FSM Symbol Definition**

Symbol	Definition
SYS_ADR	ADMA system address register
SYS_ADR++	Point to next descriptor line
DAT_ADR	Data address register (internal)
DAT_LEN	Data length register (internal)
TFC	Transfer complete flag (internal)
STOP	Stop-at-block-gap request

#### 24.4.5.2 SLAVE DMA Operations

The MMC/SD/SDIOi host controller can be interfaced with a DMA controller. At the system level, the advantage is to discharge the LH of the data transfers. The module does not support wide DMA access (above 1024 bytes) for SD cards, as specified in the *SD Card Specification* and *SD Host Controller Standard Specification*.

**NOTE:** This mode is implied by modules that are not connected to the L3 interconnect (regardless of the value of the `MMCHS_CON[20]` `DMA_MNS` bit). See Section 24.4.5.1, *MASTER DMA Operations*, for more information and/or to check the value of the `MMCHS_HL_HWINFO[0]` `MADMA_EN` bit.

The DMA request is issued if the three following conditions are met:

- The `MMCi.MMCHS_CMD[0]` `DE` bit is set to 1 to trigger the initial DMA request (the write must be done when running the data transfer command).
- A command was emitted on the `sdmmc_i_cmd` line.
- There is enough space in the buffer of the MMC/SD/SDIOi host controller to write an entire block (BLEN writes).

##### 24.4.5.2.1 DMA Receive Mode

In a DMA block read operation (single or multiple), the request signal `MMCi_DMA_RX` is asserted to its active level when a complete block is written in the buffer. The block size transfer is specified in the `MMCi.MMCHS_BLK[10:0]` `BLEN` bit field.

The `MMCi_DMA_RX` signal is deasserted to its inactive level when the sDMA has read one word from the buffer.

Only one request is sent per block; the DMA controller can make a 1-shot read access or several DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to the `BLEN` field block size.

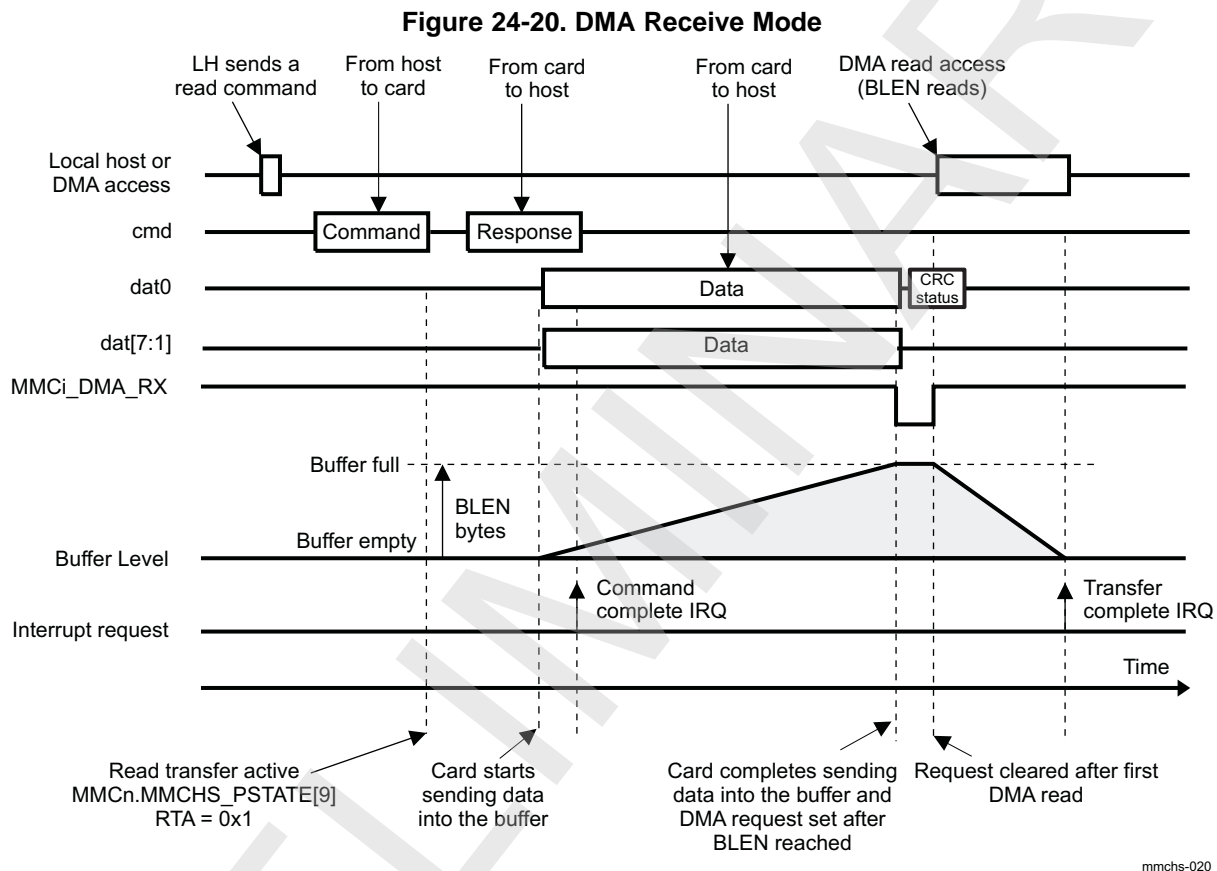
New DMA requests are internally masked if the sDMA has not read exactly `BLEN` bytes and a new complete block is not ready. Because DMA accesses are 32-bit accesses, the number of sDMA read is  $\text{Integer}(\text{BLEN}/4) + 1$ .

The receive buffer never overflows. In multiple block transfers for block sizes above 512 bytes, when the buffer becomes full, the `sdmmc_clk` clock signal (provided to the card) is momentarily stopped until the sDMA or the MPU performs a read access, which reads a complete block in the buffer.

To summarize:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 24-20 shows DMA receive mode.



#### 24.4.5.2.2 DMA Transmit Mode

In a DMA block write operation (single or multiple), the request signal `MMCi_DMA_TX` is asserted to its active level when a complete block is to be written to the buffer. The block size transfer is specified in the `MMChs.MMCHS_BLK[10:0]` BLEN bit field.

The `MMCi_DMA_TX` signal is deasserted to its inactive level when the sDMA has written one word to the buffer.

Only one request is sent per block; the DMA controller can make a one-shot write access or multiple write DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to the BLEN field block size.

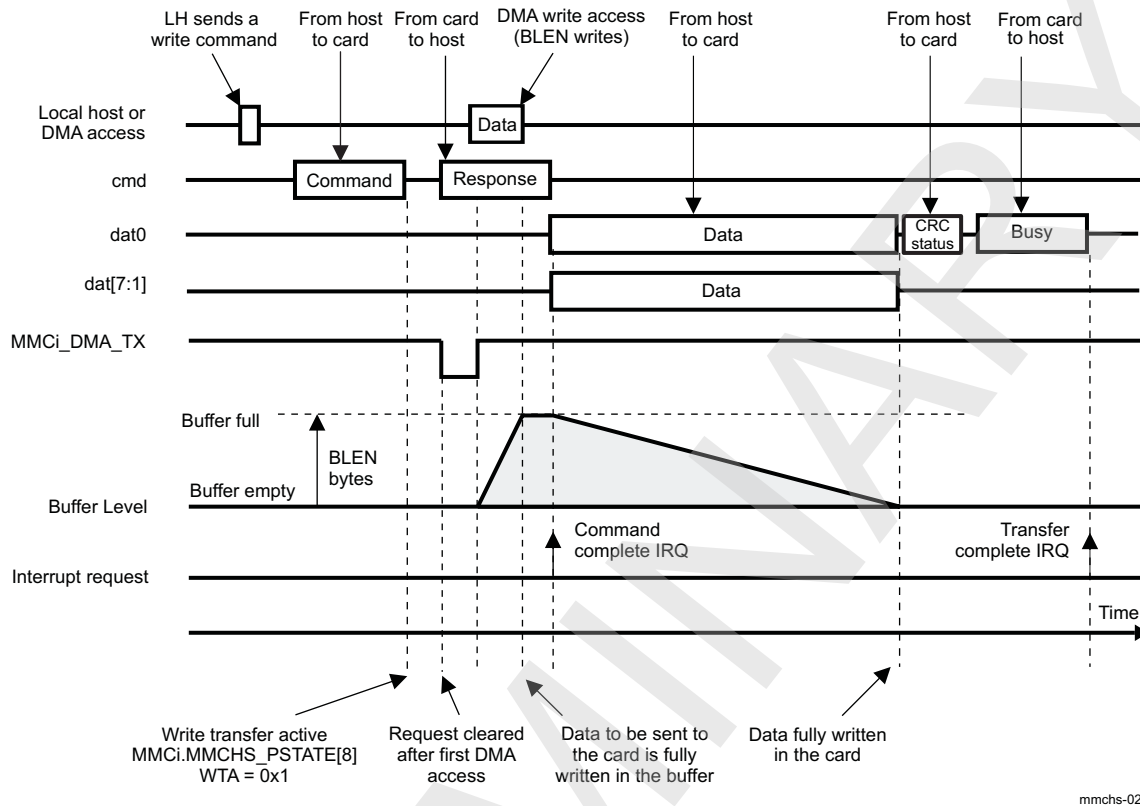
New DMA requests are internally masked if the sDMA has not written exactly BLEN bytes (because DMA accesses are 32-bit accesses, the number of sDMA read is  $\text{Integer}(\text{BLEN}/4) + 1$ ) and if there is not enough memory space to write a complete block in the buffer.

To summarize:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 24-21 shows DMA transmit mode.

Figure 24-21. DMA Transmit Mode



### 24.4.6 Mode Selection

The MMC/SD/SDIO host controller can be used in two modes: MMC and SD/SDIO. It has been designed to be the most transparent with the type of card.

The type of the card connected is differentiated by the software initialization procedure. Software identifies the type of card connected during software initialization. For each given card type, there are corresponding commands. Some commands are not supported by all cards. For more information, see the *Multimedia Card System Specification*, *SD Memory Card Specifications*, and *SDIO Card Specification, Part E1*.

The purpose of the module is to transfer commands and data to whatever card is connected, respecting the protocol of the connected card.

Writes and reads to the card must respect the appropriate protocol of that card.

### 24.4.7 Buffer Management

#### 24.4.7.1 Data Buffer

The MMC/SD/SDIOi host controller uses a data buffer. This buffer transfers data from one data bus (Interconnect) to another data bus (SD SDIO or MMC card bus) and vice versa.

The buffer is the heart of the interface and ensures the transfer between the two interfaces (Interconnect and the card).

To enhance performance, the data buffer is completed by a prefetch register and a post-write buffer that are not accessible by the host controller.

The read access time of the prefetch register is faster than that of the data buffer. The prefetch register allows data to be read from the data buffer at an increased speed by preloading data into the prefetch register.

The entry point of the data buffer, prefetch buffer, and post-write buffer is the 32-bit register `MMCi.MMCHS_DATA` register. A write access to the `MMCi.MMCHS_DATA` register followed by a read access from the `MMCi.MMCHS_DATA` register corresponds to a write access to the post-write buffer followed by a read access to the prefetch buffer. As a consequence, it is normal that the data of the write access to the `MMCi.MMCHS_DATA` register and the data of the read access to the `MMCi.MMCHS_DATA` register are different.

The number of 32-bit accesses to the `MMCi.MMCHS_DATA` register that are needed to read (or write) a data block with a size of the `MMCi.MMCHS_BLK[10:0]` `BLEN` bit field equals the rounded up result of `BLEN` divided by 4.

The maximum block size supported by the host controller is hard-coded in the register `MMCi.MMCHS_CAPA[17:16]` `MBL` bit field and cannot be changed.

A read access to the `MMCi.MMCHS_DATA` register is allowed only when the buffer read enable status is set to 1 (the `MMCi.MMCHS_PSTATE[11]` `BRE` bit); otherwise, a bad access (the `MMCi.MMCHS_STAT[29]` `BADA` bit) is signaled.

A write access to the `MMCi.MMCHS_DATA` register is allowed only when the buffer write enable status is set to 1 (the `MMCi.MMCHS_PSTATE[10]` `BWE` bit); otherwise, a bad access (the `MMCi.MMCHS_STAT[29]` `BADA` bit) is signaled and the data is not written.

The data buffer has two modes of operation to store and read of the first and second portions of the data buffer:

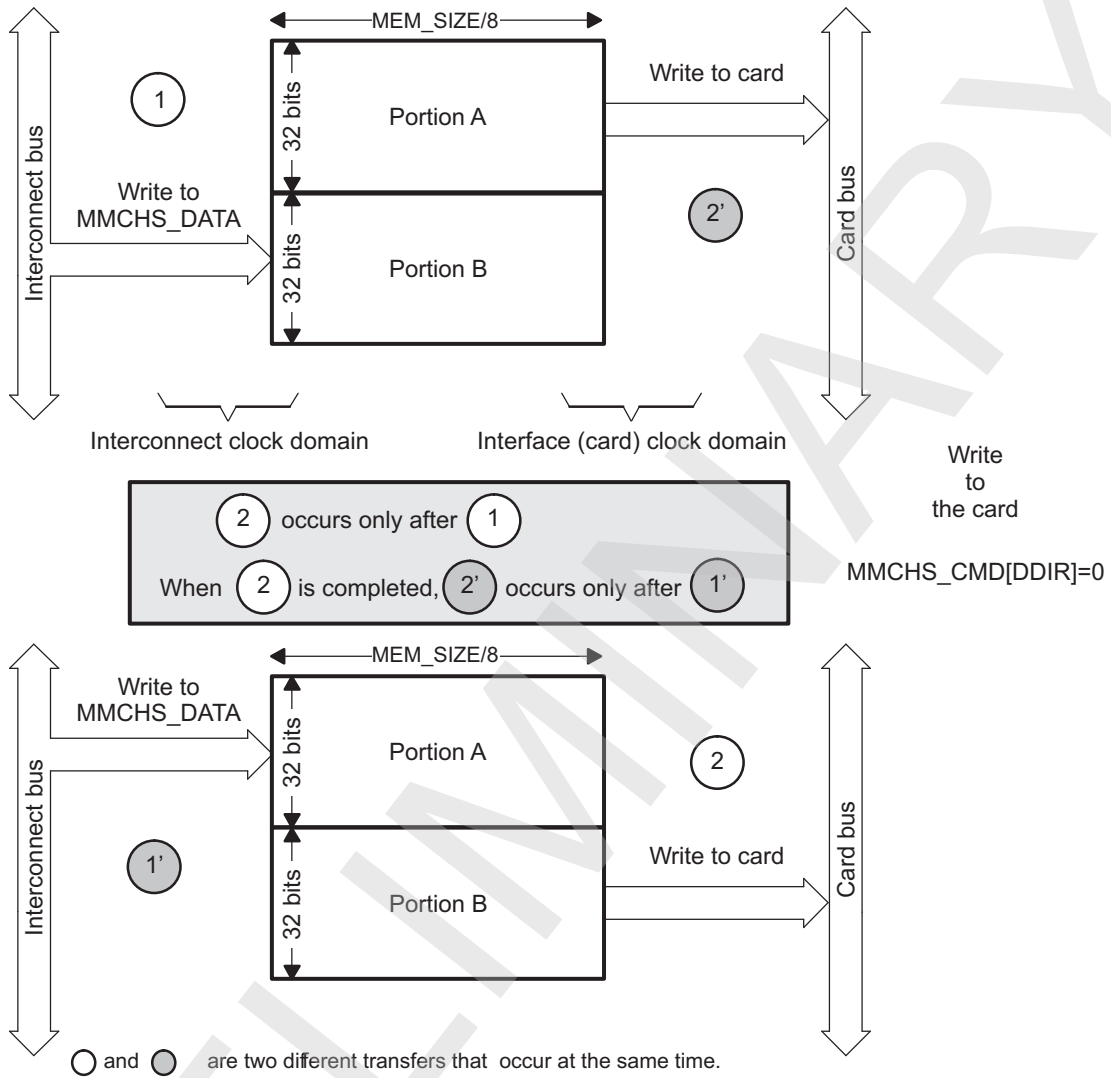
- When the size of the data block to transfer is less than or equal to `MEM_SIZE/2` (in double-buffering), two data transfers can occur at the same time from one data bus to the other data bus and vice versa. The MMC/SD/SDIO host controller uses the two portions of the data buffer in a ping-pong manner so that storing and reading the first and second portions of the data buffer are automatically interchanged from time to time so that data may be read from one portion (for instance, through a DMA read access on the interconnect bus) while data (for instance, from the card) is being stored into the other portion, and vice versa. When `BLEN` is less than or equal to `0x200` (that is, less than or equal to 512 bytes), each of the two portions of the buffer that can be used have a size of `BLEN` (that is, 32 bits  $\times$  `BLEN` divided by 4). No more than this total size of  $2 \times 32 \text{ bits} \times \text{BLEN} \text{ divided by } 4$  can be used.

#### CAUTION

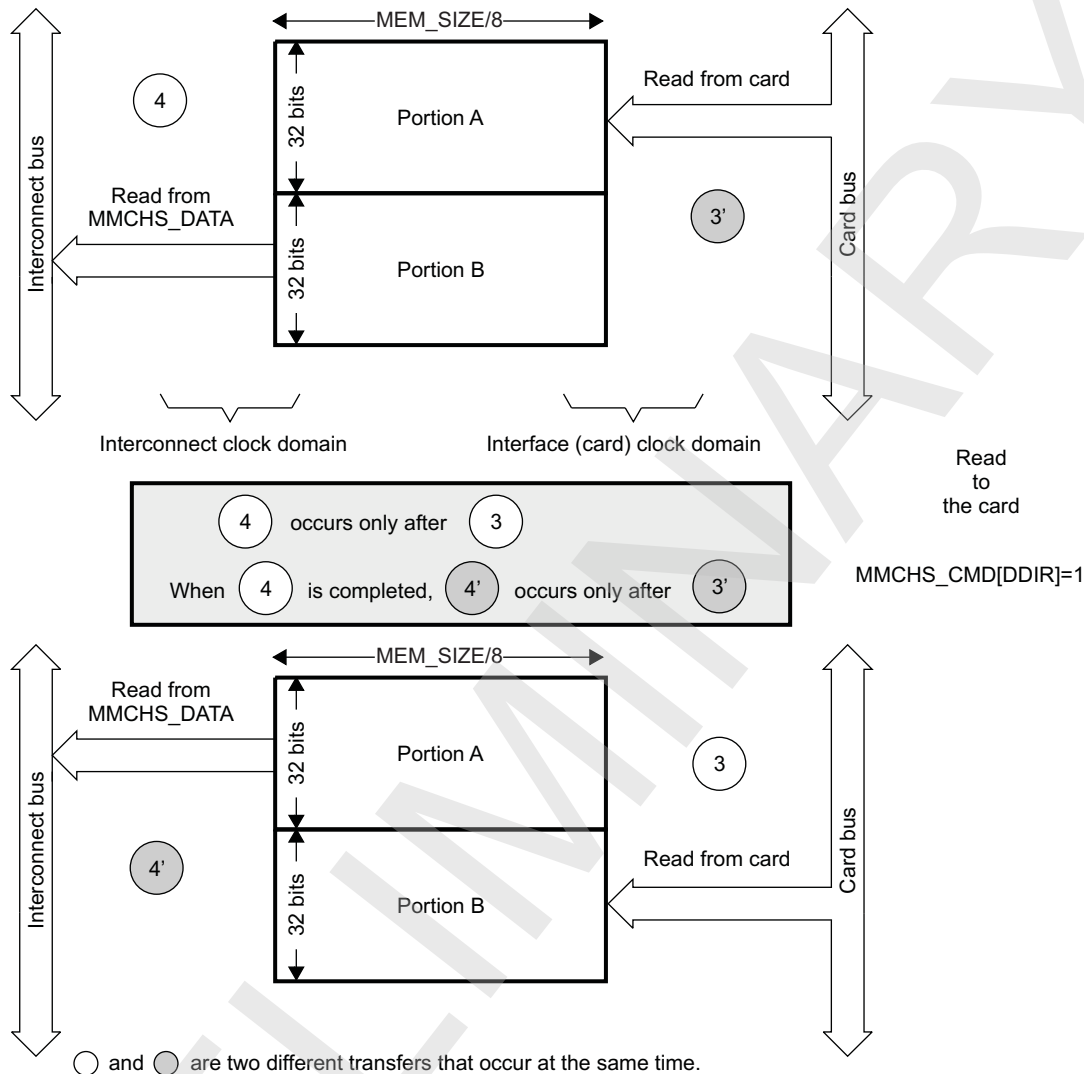
The `MMCi.MMCHS_CMD[4]` `DDIR` bit must be configured before a transfer to indicate the direction of the transfer.

Figure 24-22 and Figure 24-23 show the buffer management for a write and for a read, respectively.

Figure 24-22. Buffer Management for a Write



mmchs-022

**Figure 24-23. Buffer Management for a Read**

mmchs-023

- When the size of the data block to transfer is larger than  $\text{MEM\_SIZE}/2$ , only one data transfer can occur at a time from one data bus to the other data bus. The MMC/SD/SDIOi host controller uses the entire data buffer as a single portion. In this mode, a bad access (the MMCi.MMCHS\_STAT[29] BADA bit) is signaled when two data transfers occur at the same time from one data bus to the other data bus, and vice versa.

#### 24.4.7.1.1 Memory Size, Block Length, and Buffer-Management Relationship

The maximum block length and buffer management that can be targeted by the system depend on the memory depth setting (see [Table 24-16](#)).

**NOTE:** Double-buffering is always the buffer management for larger memory depth ( $\text{MEM\_SIZE} = 4096$ ).



**Table 24-16. Memory Size, BLEN, and Buffer Relationship**

Memory Size (MMCHS_HL_HWINFO[5:2] MEM_SIZE in bytes)	512	1024	2048	4096
Maximum block length supported	512	1024	2048	2048
Double-buffering for maximum block length	N/A	BLEN<=512	BLEN<=1024	BLEN<=2048
Single-buffering for block length	BLEN<=512	512<BLEN<=1024	1024<BLEN<=2048	N/A

**NOTE:** For single-buffering management, throughput on the MMC bus interface deteriorates in multiblock transfers, because the controller must wait for the filling or emptying of the buffer between each block transfer on the MMC bus. The clock is maintained on write MMC transfers (the MMCHS\_CMD[3] DDIR bit is 0) and halted on read MMC transfers (the MMCHS\_CMD[3] DDIR bit is 1).

#### 24.4.7.1.2 Data Buffer Status

The data buffer status is defined in the following interrupt status register and status register:

- Interrupt status registers:
  - MMCh.MMCHS\_STAT[29] BADA: Bad access to data space
  - MMCh.MMCHS\_STAT[5] BRR: Buffer read ready
  - MMCh.MMCHS\_STAT[4] BWR: Buffer write ready
- Status registers:
  - MMCh.MMCHS\_PSTATE[11] BRE: Buffer read enable
  - MMCh.MMCHS\_PSTATE[10] BWE: Buffer write enable

#### 24.4.8 Transfer Process

The process of a transfer is dependent on the type of command. It can be with or without a response, and with or without data.

##### 24.4.8.1 Different Types of Commands

Different types of commands are specific to the MMC, SD, and SDIO cards. For more information, see the *Multimedia Card System Specification*, *SD Memory Card Specifications*, *SDIO Card Specification, Part E1*, or *SD Card Specification, Part A2*, *SD Host Controller Standard Specification*.

##### 24.4.8.2 Different Types of Responses

Different types of responses are specific to the MMC, SD, and SDIO cards. For more information, see the *Multimedia Card System Specification*, *SD Memory Card Specifications*, *SDIO Card Specification, Part E1*, or the *SD Card Specification, Part A2*, *SD Host Controller Standard Specification*.

Table 24-17 describes how the MMC, SD, and SDIO responses are stored in the MMCHS\_RSPxx registers.

**Table 24-17. MMC, SD, SDIO Responses in the MMCHS\_RSPxx Registers**

Kind of Response	Response Field	Response Register
R1, R1b (normal response), R3, R4, R5, R5b, R6, R7	RESP[39:8] <sup>(1)</sup>	MMCHS_RSP10[31:0]
R1b (Auto CMD12 response)	RESP[39:8] <sup>(1)</sup>	MMCHS_RSP76[31:0]
R2	RESP[127:0] <sup>(1)</sup>	MMCHS_RSP76[31:0] MMCHS_RSP54[31:0] MMCHS_RSP32[31:0] MMCHS_RSP10[31:0]

<sup>(1)</sup> RESP refers to the command response format described in the specifications mentioned.

When the host controller modifies part of the MMCHS\_RSPxx registers, it preserves the unmodified bits.

The host controller stores the Auto CMD12 response in the MMCHS\_RSP76[31:0] register because the host controller may have a multiple block data DAT line transfer executing concurrently with a command. This allows the host controller to avoid overwriting the Auto CMD12 response with the command response stored in the MMCHS\_RSP10 register, and vice versa.

#### 24.4.9 Transfer or Command Status and Errors Reporting

Flags in the MMC/SD/SDIOi host controller show the status of communication with the card:

- A time-out (of a command, data, or response)
- A CRC error

Error conditions generate interrupts. For more information, see [Table 24-18](#) and the register description.

**Table 24-18. CC and TC Values Upon Error Detected**

Error Hold in MMCi.MMCHS_STAT	CC	TC	Comments
29 BADA			No dependency with CC or TC BADA is related to the MMCHS_DATA register accesses. Its assertion is not dependent on the ongoing transfer.
28 CERR	1		CC is set upon CERR.
22 DEB		1	TC is set upon DEB.
21 DCRC		1	TC is set upon DCRC.
20 DTO			DTO and TC are mutually exclusive. DCRC and DEB cannot occur with DTO.
19 CIE	1		CC is set upon CIE.
18 CEB	1		CC is set upon CEB.
17 CCRC	1		CC can be set upon CCRC. See CTO comment.
16 CTO			CTO and CC are mutually exclusive. CIE, CEB and CERR cannot occur with CTO. CTO can occur at the same time as CCRC: It indicates a command abort due to contention on the CMD line. In this case no CC appears.

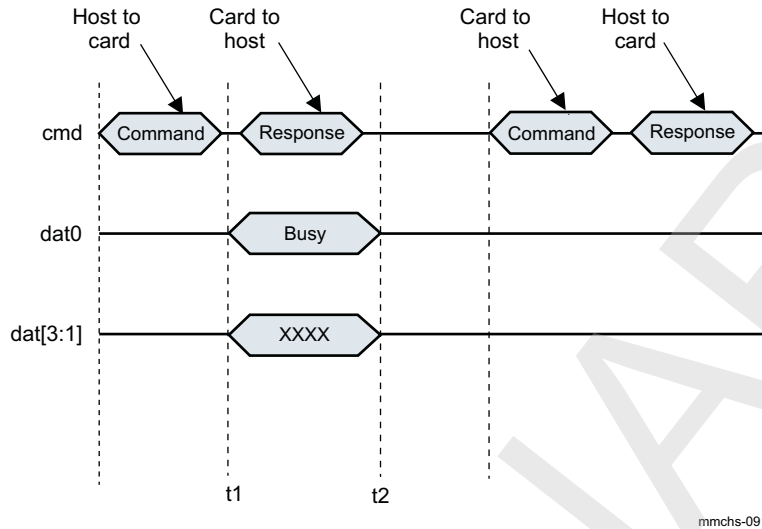
A MMCHS\_STAT[21] DCRC event can be asserted in the following conditions:

- Busy time-out for R1b, R5b response type
- Busy time-out after write CRC status
- Write CRC status time-out
- Read data time-out
- Boot acknowledge time-out

##### 24.4.9.1 Busy Time-Out for R1b, R5b Response Type

[Figure 24-24](#) shows the DCRC event condition asserted when there is busy time-out for Rb1, R5b response.

**Figure 24-24. Busy Time-Out for R1b, R5b Response Type**

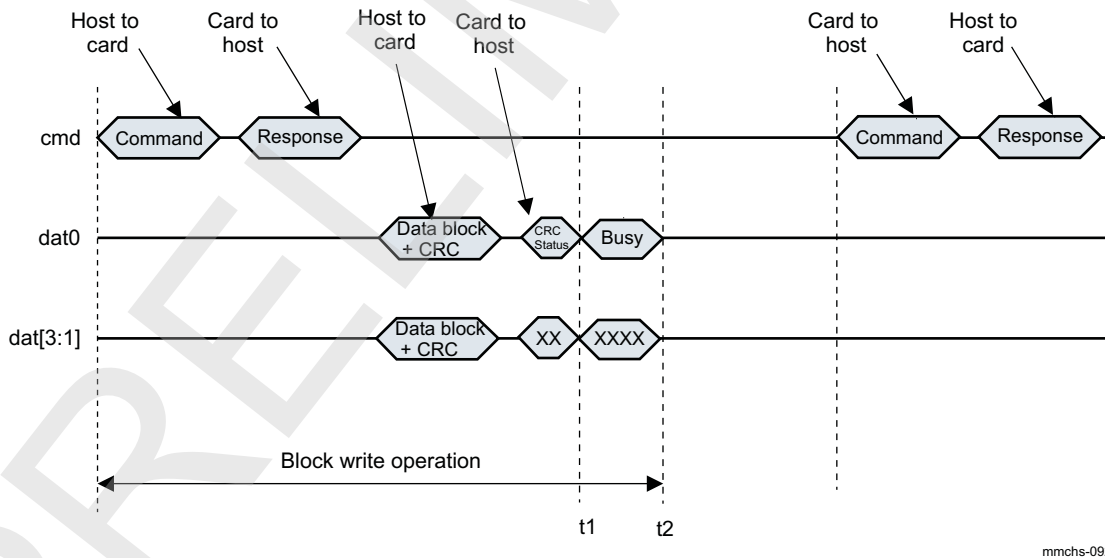


t1 – Data time-out counter is loaded and starts after R1b, R5b response type.  
t2 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.

**24.4.9.2 Busy Time-Out After Write CRC Status**

Figure 24-25 shows the DCRC event condition asserted when there is busy time-out after write CRC status.

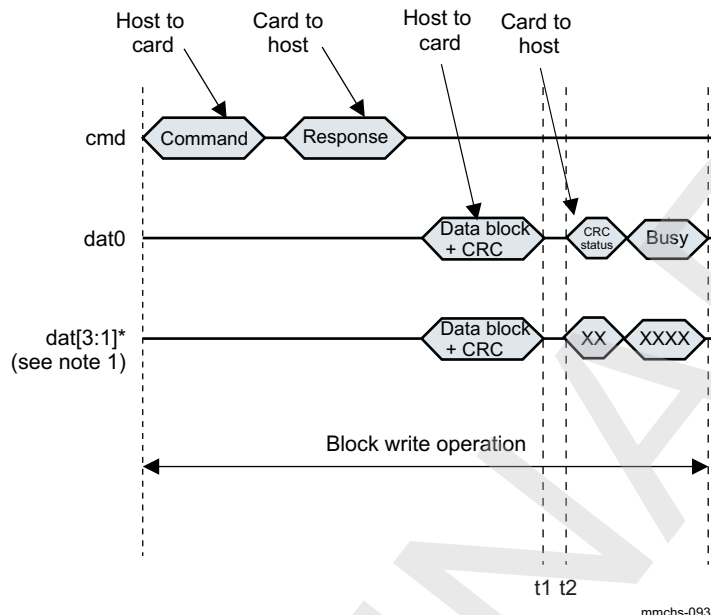
**Figure 24-25. Busy Time-Out After Write CRC Status**



t1 – Data time-out counter is loaded and starts after CRC status.  
t2 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.

**24.4.9.3 Write CRC Status Time-Out**

Figure 24-26 shows the DCRC event condition asserted when there is write CRC status time-out.

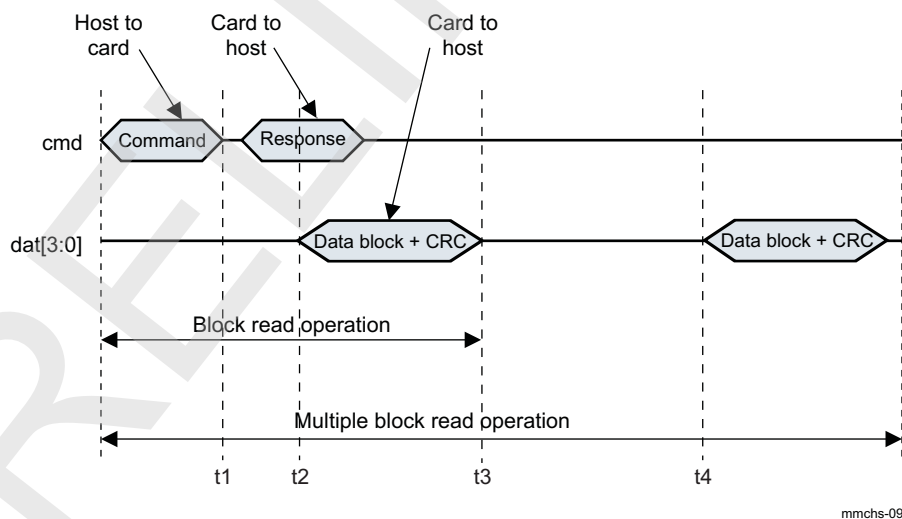
**Figure 24-26. Write CRC Status Time-Out**

t1 – Data time-out counter is loaded and starts after data block + CRC.

t2 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.

#### 24.4.9.4 Read Data Time-Out

Figure 24-27 shows the DCRC event condition asserted when there is read data time-out.

**Figure 24-27. Read Data Time-Out**

t1 – Data time-out counter is loaded and starts after command transmission.

t2 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.

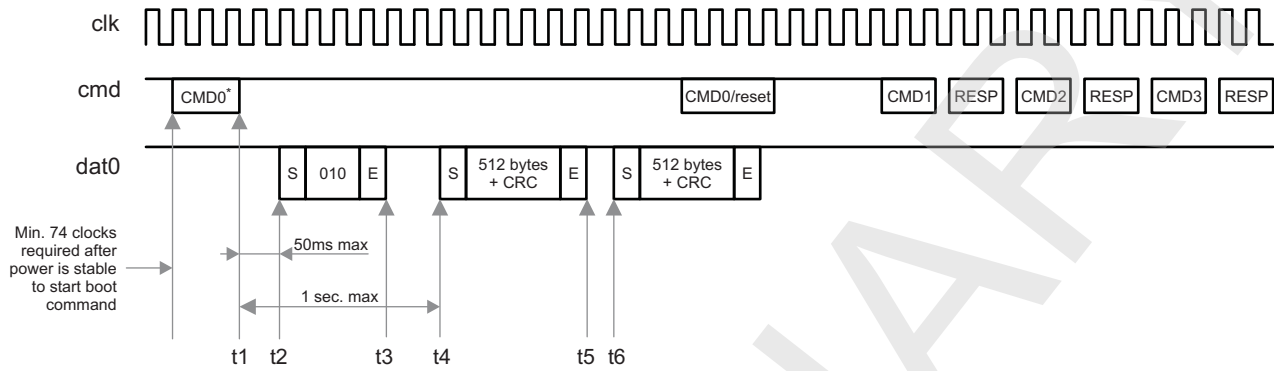
t3 – Data time-out counter is loaded and starts after data block + CRC transmission.

t4 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.

24.4.9.5 Boot Acknowledge Time-Out

Figure 24-28 shows the DCRC event condition asserted when there is a boot acknowledge time-out and CMD0 is used.

Figure 24-28. Boot Acknowledge Time-Out When Using CMD0



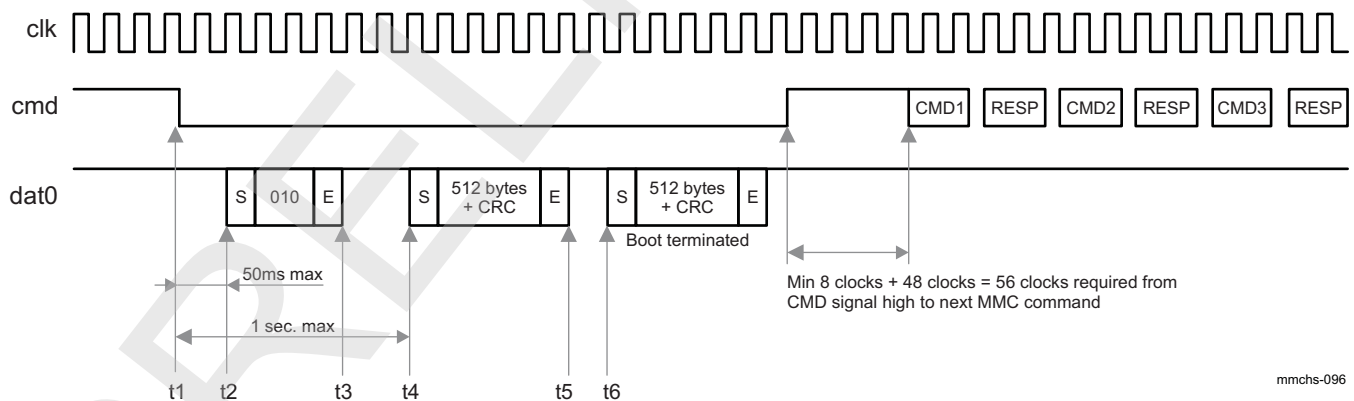
\* Refer to MMC specification for correct Argument

mmchs-095

- t1 – Data time-out counter is loaded and starts after CMD0.
- t2 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.
- t3 – Data time-out counter is loaded and starts.
- t4 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.
- t5 – Data time-out counter is loaded and starts after data + CRC transmission.
- t6 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.

Figure 24-28 shows the DCRC event condition asserted when there is boot acknowledge time-out and CMD0 is used.

Figure 24-29. Boot Acknowledge Time-Out When CMD Line Tied to 0



mmchs-096

- t1 – Data time-out counter is loaded and starts after the command line is tied to 0.
- t2 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.
- t3 – Data time-out counter is loaded and starts.
- t4 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.
- t5 – Data time-out counter is loaded and starts after data + CRC transmission.
- t6 – Data time-out counter stops and if it is 0, the [MMCHS\\_STAT\[21\]](#) DCRC bit is generated.

### 24.4.10 Auto Command 12 Timings

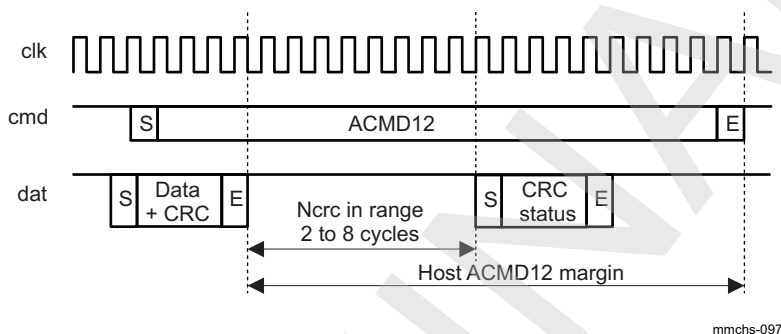
With the UHS definition of SD cards with higher frequency for MMC clock up to 208, the SD standard imposes a specific timing for the arrival of the auto command 12 (ACMD12) end bit.

#### 24.4.10.1 ACMD12 Timings During Write Transfer

A margin named *Nrc* in the range of two to eight cycles has been defined for SDR50 and SDR104 card components for write data transfers, because the ACMD12 end bit must arrive after the CRC status end bit.

Figure 24-30 shows the ACMD12 timings during write transfer.

**Figure 24-30. ACMD12 Timings During Write Transfer**



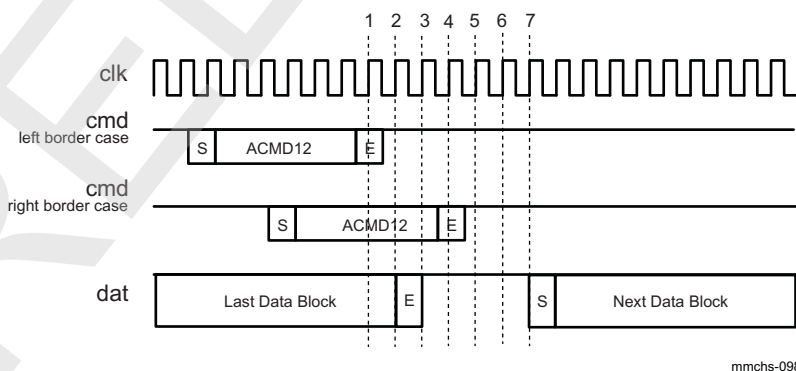
The host controller has a margin of 18 clock cycles to ensure that the ACMD12 end bit arrives after the CRC status. This margin does not depend on the MMC bus configuration, DDR, or standard transfer, 1-, 4-, or 8-bit bus width.

#### 24.4.10.2 ACMD12 Timings During Read Transfer

With UHS cards, the gap timing between two successive cards has been extended from two cycles to four cycles. It provides more flexibility for the host ACMD12 arrival to receive the last complete and reliable block. The MMCHS controller follows only the Left Border Case defined by the SD UHS specification.

Figure 24-31 shows ACMD12 timings during read transfer.

**Figure 24-31. ACMD12 Timings During Read Transfer**



The ACMD12 arrival sent by the host controller is not sensitive to the MMC bus configuration, whether it is a DDR or standard transfer and whether it is a 1-, 4-, or 8-bit bus width transfer.

### 24.4.11 Transfer Stop

Whenever a transfer is initiated, the transmission may be willed to stop while it is still not finished. Several cases are possible, depending on the transfer type:

- Multiple-block-oriented transfers (transfer length is known)

- Continuous stream transfers (transfer has an infinite length)

**NOTE:** Because the MMC/SD/SDIOi controller manages transfers based on a block granularity, the buffer accepts a block only if there is enough space to store it completely. Consequently, if a block is pending in the buffer, no command is sent to the card because the card clock will be shut off by the controller.

The MMC/SD/SDIOi controller includes two features that make a transfer stop more convenient and easier to manage:

- **ACMD12 (for MMC and SD only):**  
This feature is enabled by setting the MMCI.MMCHS\_CMD[2] ACEN bit to 0x1 (this setting is relevant for a MMC/SD transfer with a known number of blocks to transfer). When the ACMD12 feature is enabled, the MMC/SD/SDIOi controller automatically issues a CMD12 command when the expected number of blocks is exchanged.
- **Stop at block gap:**  
This feature is enabled by setting the MMCI.MMCHS\_HCTL[16] SBGR bit to 0x1. When enabled, this capability holds the transfer on until the end of a block boundary. If a stop transmission is needed, software can use this pause to send a CMD12 to the card.
- **ADMA mode:**  
For ADMA-capable modules (MMC1 and MMC2) (see Section 24.4.5, DMA Modes, for more information), the last instruction can stop the transfer (the END bit is enabled in the descriptor line).

**NOTE:** For MMC and SD cards, the stop-at-block-gap feature is not supported in read mode.

For SDIO cards, this setting can be supported in read mode if the card has a read-wait capability.

Table 24-19 shows the common way to stop a transfer, indicating the command to send and the features to enable.

**Table 24-19. MMC/SD/SDIOi Controller Transfer Stop Command Summary**

		Write Transfer		Read Transfer	
		SD/MMC	SDIO	SD/MMC	SDIO
Single block		Transfer ends automatically. Wait TC.	Transfer ends automatically. Wait TC.	Transfer ends automatically. Wait TC.	Transfer ends automatically. Wait TC.
Multiblocks (finite or infinite)	Before the programmed block boundary	Send CMD12. Wait TC.	Send CMD52. Wait TC.	Send CMD12. Wait TC.	Send CMD52. Wait TC.
	Stop at the end of the transfer (finite transfer only)	ACMD12 active Transfer ends automatically. Wait TC.	Set MMCI.MMCHS_HCTL[16] SBGR bit to 0x1. Send CMD52. Wait TC.	ACMD12 active Transfer ends automatically. Wait TC.	<b>If READ_WAIT supported</b> Stop at block gap. Wait TC.  <b>If READ_WAIT not supported</b> Send CMD52. Wait TC.

**NOTE:** The MMC/SD/SDIOi controller sends the stop command to the card on a block boundary, regardless of when the command was written to the controller registers.

### 24.4.12 Retention Mode

This mode is enabled if generic parameter MMCHS\_HL\_HWINFO[6] RETMODE is set to 1 during module integration. The controller can enter into retention mode when the module was previously set in idle mode. When the module goes into retention mode all clocks are left inactive at low level except for registers that

are driven by a falling-edge clock, which are kept as free during retention mode. This also means that all divided functional clock subdomains are forced to low when the module enters into smart-idle mode. In force-idle mode, the system must ensure that all divided functional clocks are set inactive or in bypass mode by configuring the `MMCHS_SYSCTL` register correctly before entering into idle mode (under software responsibility).

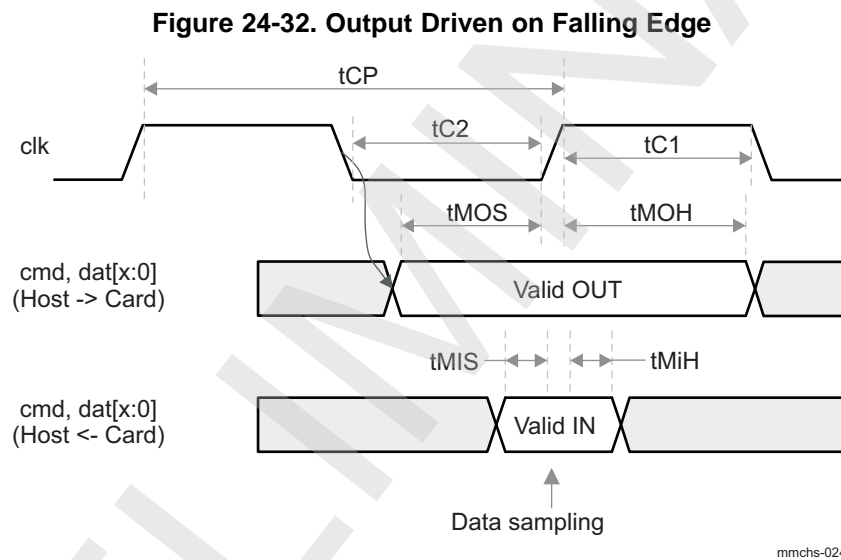
### 24.4.13 Output Signals Generation

The MMC/SD/SDIO output signals can be driven on the falling edge or rising edge, depending on the `MMCHS_HCTL[2]` HSPE bit. This feature allows reaching better timing performance, and thus increasing data transfer frequency.

#### 24.4.13.1 Generation on Falling Edge of MMC Clock

The controller is by default in this mode to maximize hold timings. In this case, the `MMCHS_HCTL[2]` HSPE bit is set to 0.

Figure 24-32 shows the output signals of the module when generating from the falling edge of the MMC clock.



#### 24.4.13.2 Generation on Rising Edge of MMC Clock

This mode increases setup timings and allows reaching higher bus frequency. This feature is activated by setting the `MMCHS_HCTL[2]` HSPE bit to 1. The controller must be set in this mode to support SDR transfers.

---

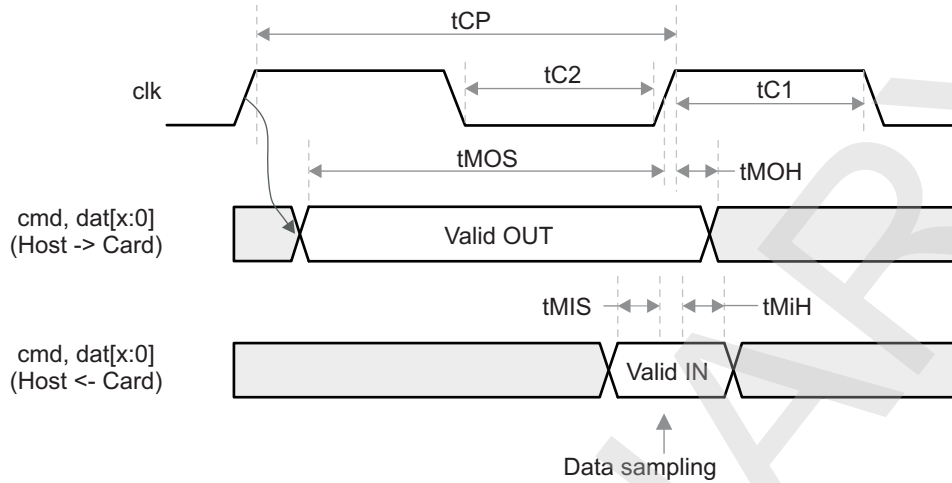
**NOTE:** Do not use this feature in DDR mode (when the `MMCHS_CON[19]` DDR bit is set to 1).

---

Figure 24-33 shows the output signals of the module when generating from the rising edge of the MMC clock.



Figure 24-33. Output Driven on Rising Edge



mmchs-025

#### 24.4.14 Card Boot Mode Management

Boot operation mode allows the MMC/SD/SDIOi host controller to read boot data from the connected slave (MMC device) by keeping CMD line low after power on (or sending CMD0 with specific argument) before issuing CMD1. The data can be read from the boot area or user area, depending on the register setting.

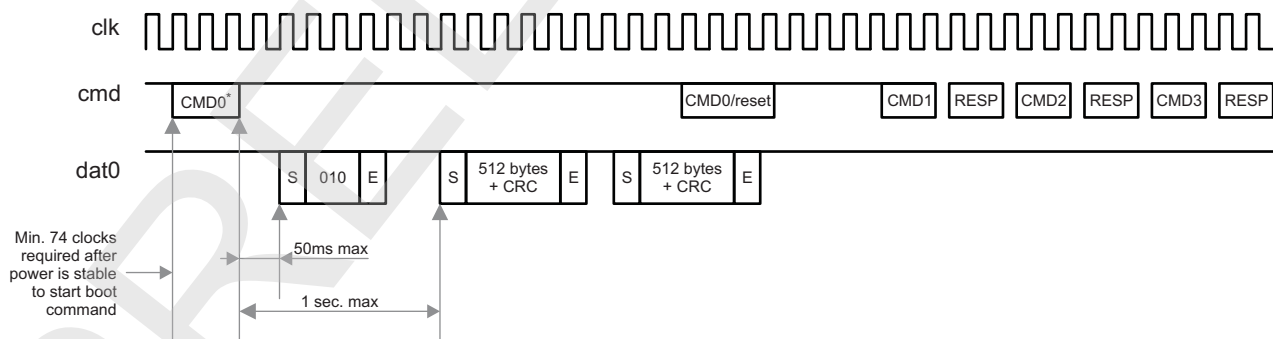
Power-on boot defines a way for the boot code to be accessed by the MMC/SD/SDIOi host controller without an upper-level software driver, speeding the time it takes for a controller to access the boot code.

The two possible ways to issue a boot command (issuing a CMD0 or driving the CMD line to 0 during the whole boot phase) are described in the following sections.

##### 24.4.14.1 Boot Mode Using CMD0

Figure 24-34 shows the timing diagram of a boot sequence using CMD0.

Figure 24-34. Boot Mode Using the CMD0 Timing Diagram



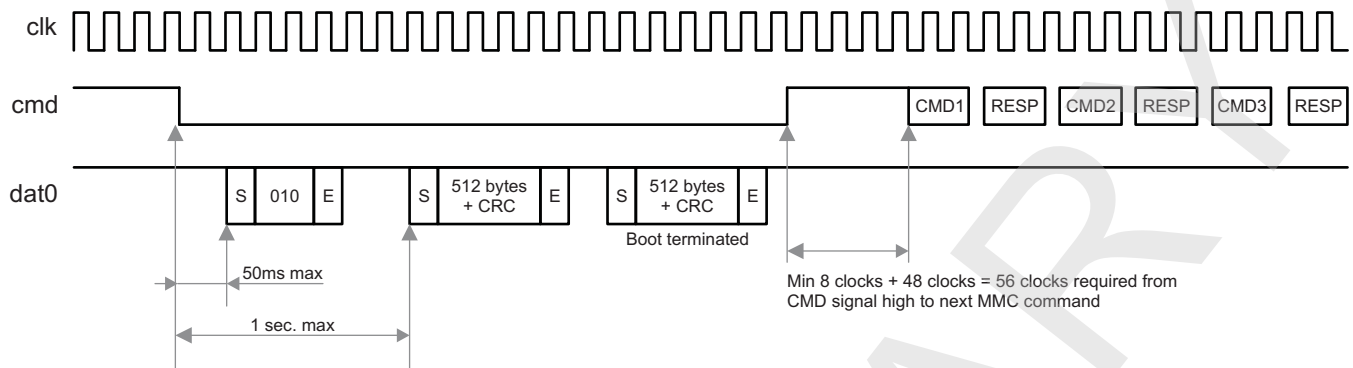
\* Refer to MMC specification for correct Argument

mmchs-026

For more information about how to configure the MMC/SD/SDIO host controller, see [Section 24.5.1.2.3.1, Boot Using the CMD0](#).

##### 24.4.14.2 Boot Mode With CMD Line Tied to 0

Figure 24-35 shows the timing diagram of a boot sequence with CMD line tied to 0.

**Figure 24-35. Boot Mode With CMD Line Tied to 0 Timing Diagram**

mmchs-027

For more information about how to configure the MMC/SD/SDIO host controller, see [Section 24.5.1.2.3.2, Boot With CMD Line Tied to 0](#).

#### 24.4.15 MMC CE-ATA Command Completion Disable Management

The MMC/SD/SDIOi host controller supports CE-ATA features, in particular the detection of command completion token. When a command that requires a command completion signal (the [MMCHS\\_CON\[12\]](#) CEATA and [MMCHS\\_CMD\[2\]](#) ACEN bits set to 1) is launched, the host system is no longer allowed to emit a new command in parallel to the data transfer unless it is a command completion disable token.

The settings to emit a command completion disable token are:

- The [MMCHS\\_CON\[12\]](#) CEATA is set to 1.
- The [MMCHS\\_CON\[2\]](#) HR set to 1.
- Clear the [MMCHS\\_ARG](#) register.
- Write into the [MMCHS\\_CMD](#) register with value 0x00000000.

When a command completion disable token was emitted (that is, the [MMCHS\\_STAT\[0\]](#) CC bit received), the host system is again allowed to emit another type of command (for example, a CMD12 to abort transfer).

A critical case can be encountered when command completion signal disable (CCSD) is emitted during the last data block transfer, and the sequence on the command line is sent close to the command completion signal (CCS) token sent by the card.

Three possible cases are:

- CCS is received immediately before CCSD is emitted:  
An interrupt CIRQ is generated with CCS detection, CCSD is transmitted to the card, and then an interrupt CC is generated when CCSD ends. In this case, the card considers the CCSD sequence.
- CCS is not generated or is generated during the CCSD transfer:  
The CCS bit cannot be detected (conflict is not possible because they drive the same level on the command line, and no CIRQ interrupt is generated; a CC interrupt is generated when CCSD ends).
- CCS is generated without CCSD token required:  
Only the interrupt CIRQ is generated when CCS is detected.

#### 24.4.16 Test Registers

Test registers are available to comply with the *SD Host Controller Specification*. This feature is useful to generate interrupts manually for driver debugging.

The force event register ([MMCHS\\_FE](#)) is used to control the error interrupt status and ACMD12 error status.

The system test register ([MMCHS\\_SYSTEST](#)) is used to control the signals that connect to I/O pins when the module is configured in the system test mode (the [MMCHS\\_CON\[4\]](#) MODE bit = 1) for boundary connectivity verification.

### 24.4.17 MMC/SD/SDIO Hardware Status Features

This section discusses the MMC/SD/SDIO hardware status features.

**Table 24-20. MMC/SD/SDIO Hardware Status Features**

Feature	Type	Register/Bit Field/Observability Control	Description
Interrupt flags		See <a href="#">Section 24.4.4</a> , <i>Interrupt Requests</i> .	
CMD line signal level	Status	<a href="#">MMCHS_PSTATE</a> [24] CLEV	Indicates the level of the cmd line
DAT lines signal level	Status	<a href="#">MMCHS_PSTATE</a> [23:20] DLEV	Indicates the level of the data lines
Buffer read enable	Status	<a href="#">MMCHS_PSTATE</a> [11] BRE	Readable data exists in the buffer.
Buffer write enable	Status	<a href="#">MMCHS_PSTATE</a> [10] BWE	Indicates whether there is enough space in the buffer to write BLEN bytes of data
Read transfer active	Status	<a href="#">MMCHS_PSTATE</a> [9] RTA	This status is used for detecting completion of a read transfer.
Write transfer active	Status	<a href="#">MMCHS_PSTATE</a> [8] WTA	This status indicates a write transfer active.
Data line active	Status	<a href="#">MMCHS_PSTATE</a> [2] DLA	Indicates whether the data lines are active
Command Inhibit (data lines)	Status	<a href="#">MMCHS_PSTATE</a> [1] DATI	Indicates whether issuing of command using data lines is allowed
Command inhibit (CMD line)	Status	<a href="#">MMCHS_PSTATE</a> [0] CMDI	Indicates whether issuing of command using CMD line is allowed

## 24.5 MMC/SD/SDIO Programming Guide

### 24.5.1 Low-Level Programming Models

#### 24.5.1.1 Global Initialization

##### 24.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the module has to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the MMC/SD/SDIO modules. For more information, see [Section 24.3, MMC/SD/SDIO Integration](#), and [Section 24.2, MMC/SD/SDIO Environment](#).

**Table 24-21. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. See "module clocks enable programming model" in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Control module	Module-specific pad muxing and configuration must be set in the control module. See <a href="#">Chapter 18, Control Module</a> .
(optional) MPU INTC (or DSP INTC)	MPU INTC configuration must be done to enable the interrupts from the MMCHS module. See <a href="#">Chapter 17, Interrupt Controllers</a> .
(optional) sDMA (or dDMA)	DMA configuration must be done to enable the module DMA channel requests. See <a href="#">Chapter 16, SDMA</a> .
(optional) Interconnect	For more information about the interconnect configuration, see <a href="#">Chapter 13, Interconnect</a> .

**NOTE:** The MPU/DSP INTC and the sDMA/dDMA configurations are necessary if the interrupt and DMA-based communication modes are used.

##### 24.5.1.1.2 MMC/SD/SDIO Host Controller Initialization Flow

[Table 24-22](#) shows the general boot process.

**Table 24-22. MMC/SD/SDIO Controller Meta Initialization Steps**

Step	Access Type	Register/Bit Field/Programming Model	Value
Initialize clocks.		See <a href="#">Section 24.5.1.1.2.1</a> .	
Software reset of the controller		See <a href="#">Section 24.5.1.1.2.2</a> .	
Set module hardware capabilities.		See <a href="#">Section 24.5.1.1.2.3</a> .	
Set module idle and wake-up modes.		See <a href="#">Section 24.5.1.1.2.4</a> .	

##### 24.5.1.1.2.1 Enable Interface and Functional clock for MMC Controller

Before any MMCHS register access the MMCHS interface clock and functional clock in the PRCM module registers must be enabled. See [Section 3.6.4.4, Clock Domain Module Attributes](#), in [Chapter 3, Power, Reset, and Clock Management](#).

##### 24.5.1.1.2.2 MMCHS Soft Reset Flow

[Figure 24-36](#) shows the soft reset process of the MMCHS controller.

Figure 24-36. MMC/SD/SDIO Controller Software Reset Flow

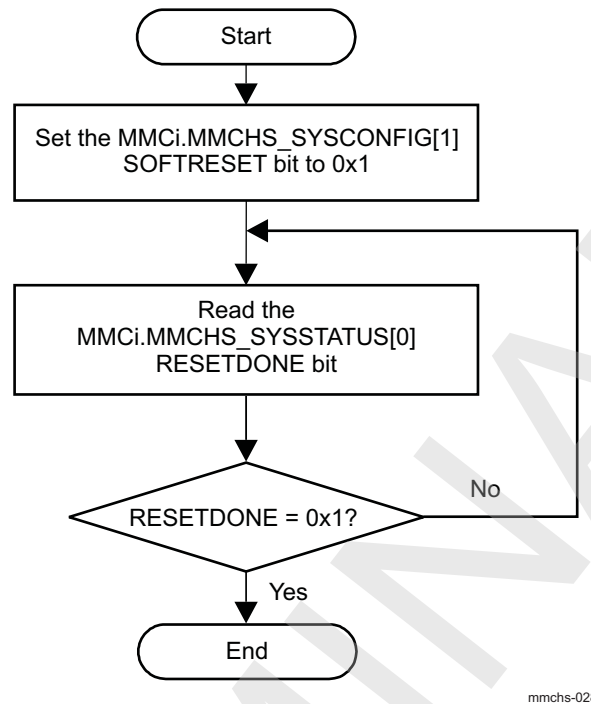


Table 24-23. Register Call Summary for Main Sequence – Software Reset Flow

Register Name	Register Name
MMCHS_SYSCONFIG	MMCHS_SYSSTATUS

24.5.1.1.2.3 Set MMCHS Default Capabilities

Software must read capabilities (in boot ROM, for example) and is allowed to set (write) the MMCi.MMCHS\_CAPA[26:24] and MMCi.MMCHS\_CUR\_CAPA[23:0] bit fields before the MMC/SD/SDIO host driver is started.

24.5.1.1.2.4 Wake-Up Configuration

Table 24-24 describes the MMCHS controller wake-up configuration.

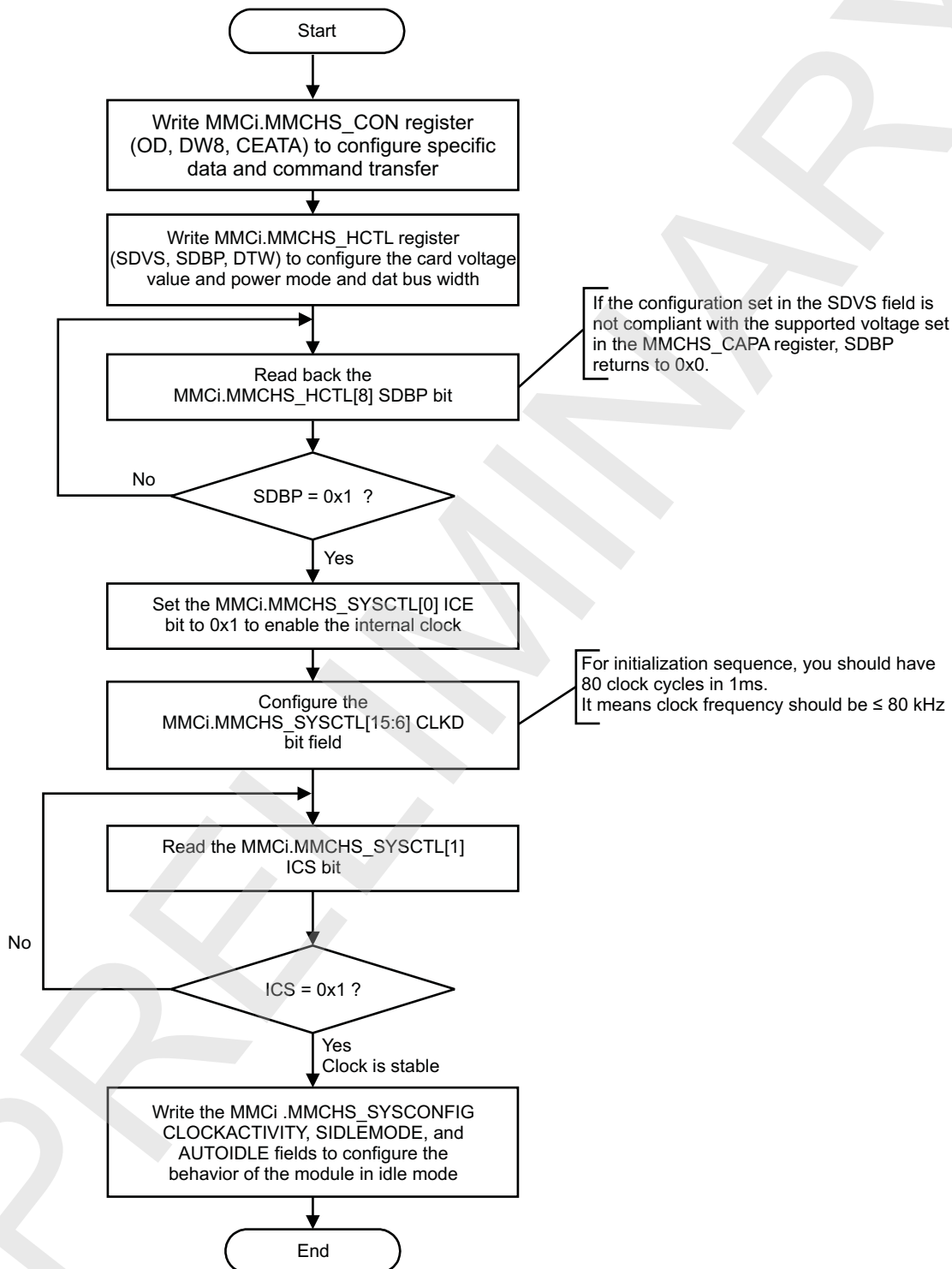
Table 24-24. MMC/SD/SDIO Controller Wake-Up Configuration

Step	Access Type	Register/Bit Field/Programming Model	Value
Configure wake-up bit (if necessary).	W	MMCi.MMCHS_SYSCONFIG[2] ENAWAKEUP	0x1
Enable wake-up events on SD card interrupt (if necessary).	W	MMCi.MMCHS_HCTL[24] IWE	0x1
<b>SDIO Card only:</b> Enable card interrupt (if necessary).	W	MMCi.MMCHS_IE[8] CIRQ_ENABLE	0x1

### 24.5.1.1.2.5 MMC Host and Bus Configuration

Figure 24-37 shows the MMC bus configuration process.

**Figure 24-37. MMC/SD/SDIO Controller Bus Configuration**



mmchs-029

**Table 24-25. Register Call Summary for Main Sequence – Bus Configuration**

Register Name	Register Name
MMCHS_CON	MMCHS_HCTL
MMCHS_SYSCONFIG	MMCHS_SYSCTL

### 24.5.1.2 Operational Modes Configuration

#### 24.5.1.2.1 Basic Operations for MMC/SD/SDIO Host Controller

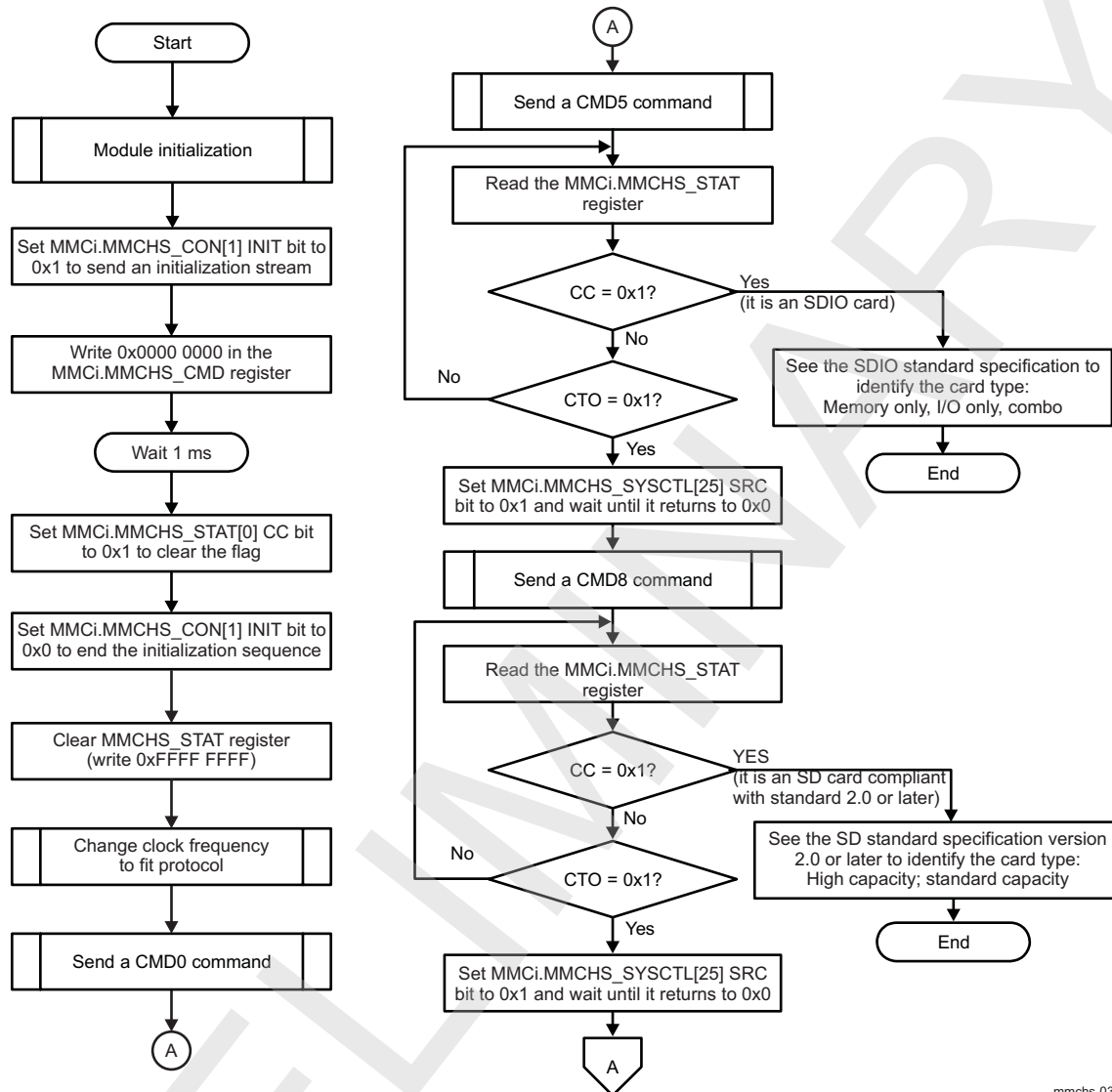
The MMC/SD/SDIO host controller performs data transfers: data to card (referred to as write transfers) and data from card (referred to as read transfers).

The host controller requires transfers to run on a block-by-block basis rather than on a DMA burst size basis. A single DMA request (or block request interrupt) is signaled for each block. Pipelining is supported as long as the block size is less than one half of the memory buffer size.

##### 24.5.1.2.1.1 Card Detection, Identification, and Selection

[Figure 24-38](#) and [Figure 24-39](#) show the card identification and selection process.

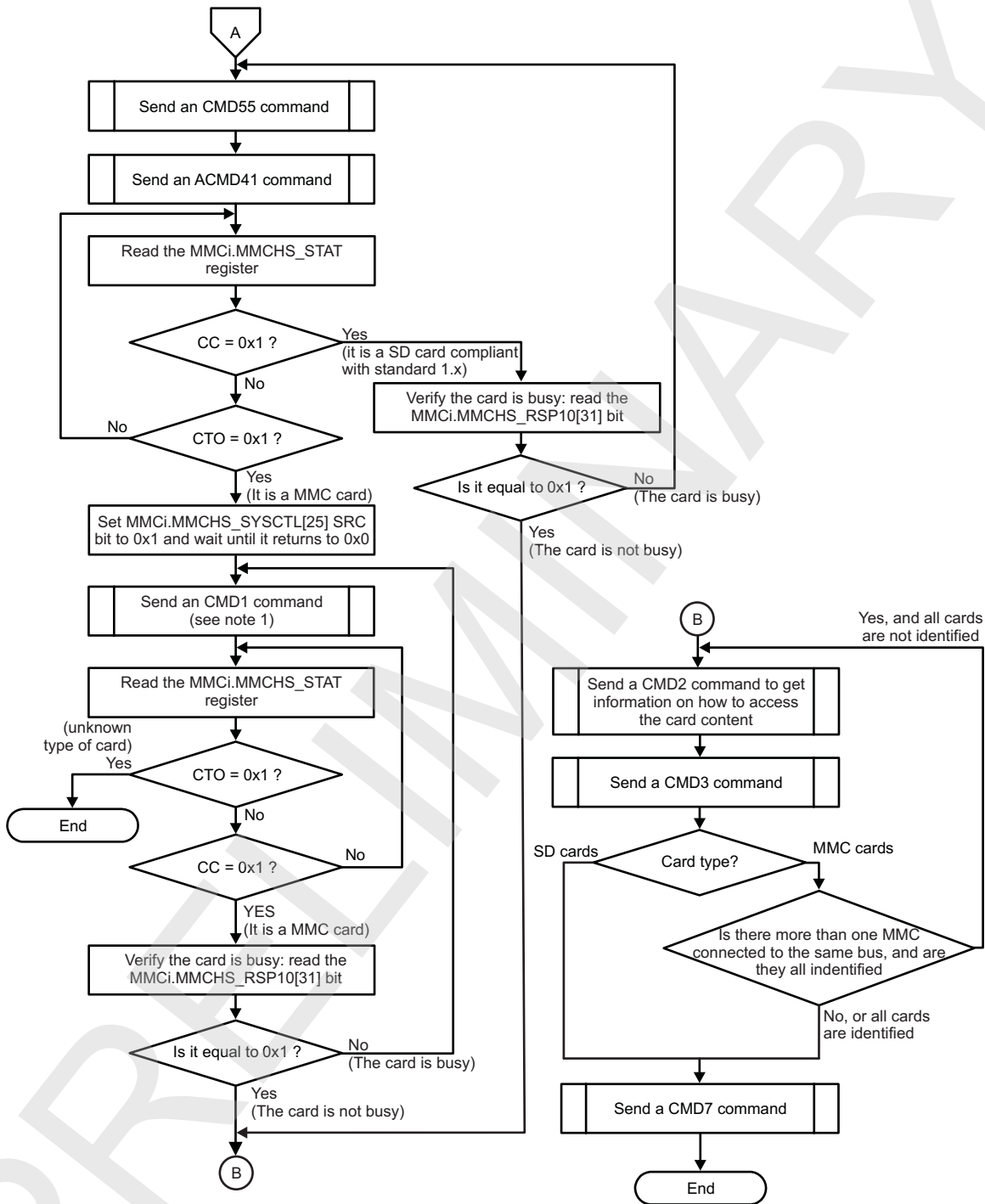
Figure 24-38. MMC/SD/SDIO Controller Card Identification and Selection – Part 1



mmchs-030



Figure 24-39. MMC/SD/SDIO Controller Card Identification and Selection – Part 2



mmchs-031

Table 24-26. Register Call Summary for Main Sequence – Card Identification and Selection

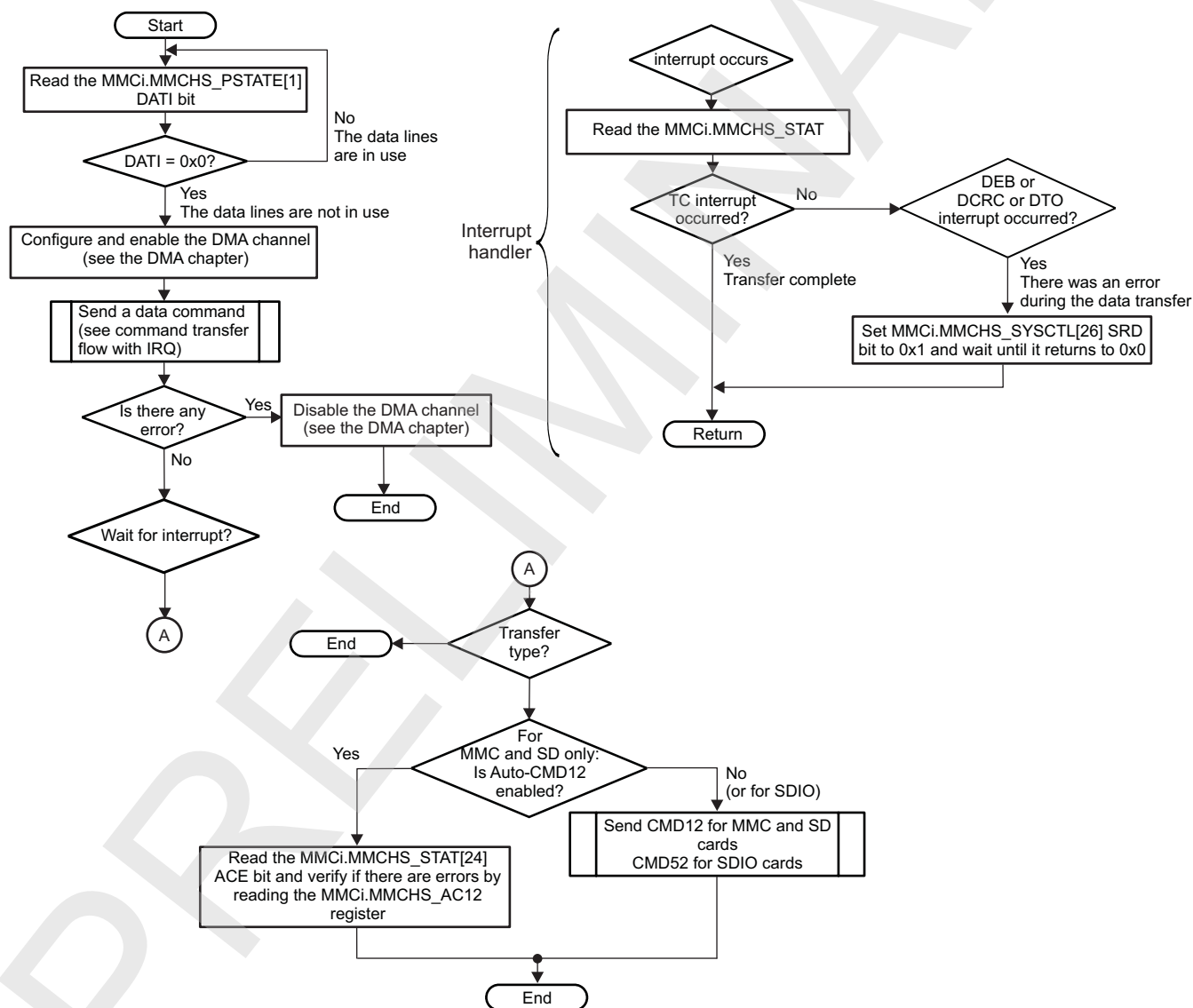
Register Name	Register Name	Register Name
MMCHS_CON	MMCHS_CMD	MMCHS_STAT
MMCHS_SYSCTL	MMCHS_RSP10	

**Table 24-27. Subprocess Call Summary for Main Sequence – Card Identification and Selection**

Subprocess Name	Cross-Reference
Initialize module.	<a href="#">Section 24.5.1.1.2</a>
Change clock frequency to fit protocol.	<a href="#">Section 24.5.1.2.1.7.2</a>
Send a command.	<a href="#">Section 24.5.1.2.1.7.1</a>

### 24.5.1.2.1.2 Read/Write Transfer Flow in DMA Mode With Interrupt

Figure 24-40 shows the read and write protocol in DMA slave mode with interrupt signaling. For more information about the DMA settings, see [Section 16.5, sDMA Basic Programming Model](#), in [Chapter 16, SDMA](#).

**Figure 24-40. MMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Slave Mode With interrupt**

mmchs-032

**Table 24-28. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With interrupt**

Register Name	Register Name
<a href="#">MMCHS_PSTATE</a>	<a href="#">MMCHS_STAT</a>

**Table 24-28. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With interrupt (continued)**

Register Name	Register Name
MMCHS_SYSCTL	MMCHS_CMD

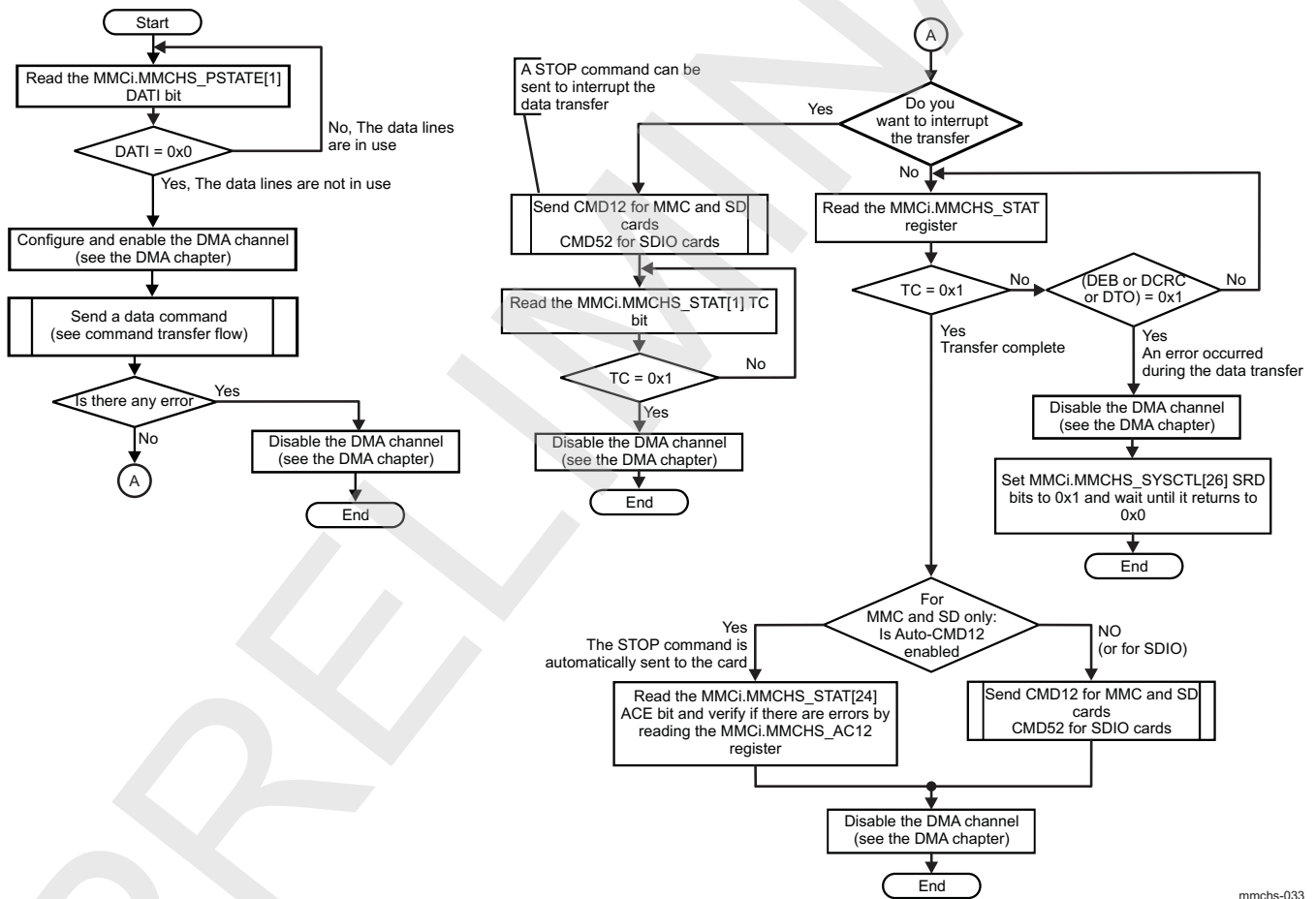
**Table 24-29. Sub-process call summary for Main Sequence - MMC/SD/SDIO Controller Read/Write Transfer Flow in DMA mode with interrupt**

Subprocess Name	Cross-Reference
Send a data command.	Figure 24-47

**24.5.1.2.1.3 Read/Write Transfer Flow in DMA Mode With Polling**

Figure 24-41 shows the read and write protocol in DMA mode. For more information about the DMA settings, see Section 16.5, sDMA Basic Programming Model in Chapter 16, SDMA.

**Figure 24-41. MMC/SD/SDIO Controller Read/Write Transfer Flow in DMA Mode With Polling**



**Table 24-30. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling**

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_STAT	MMCHS_SYSCTL
MMCHS_CMD	MMCHS_AC12	

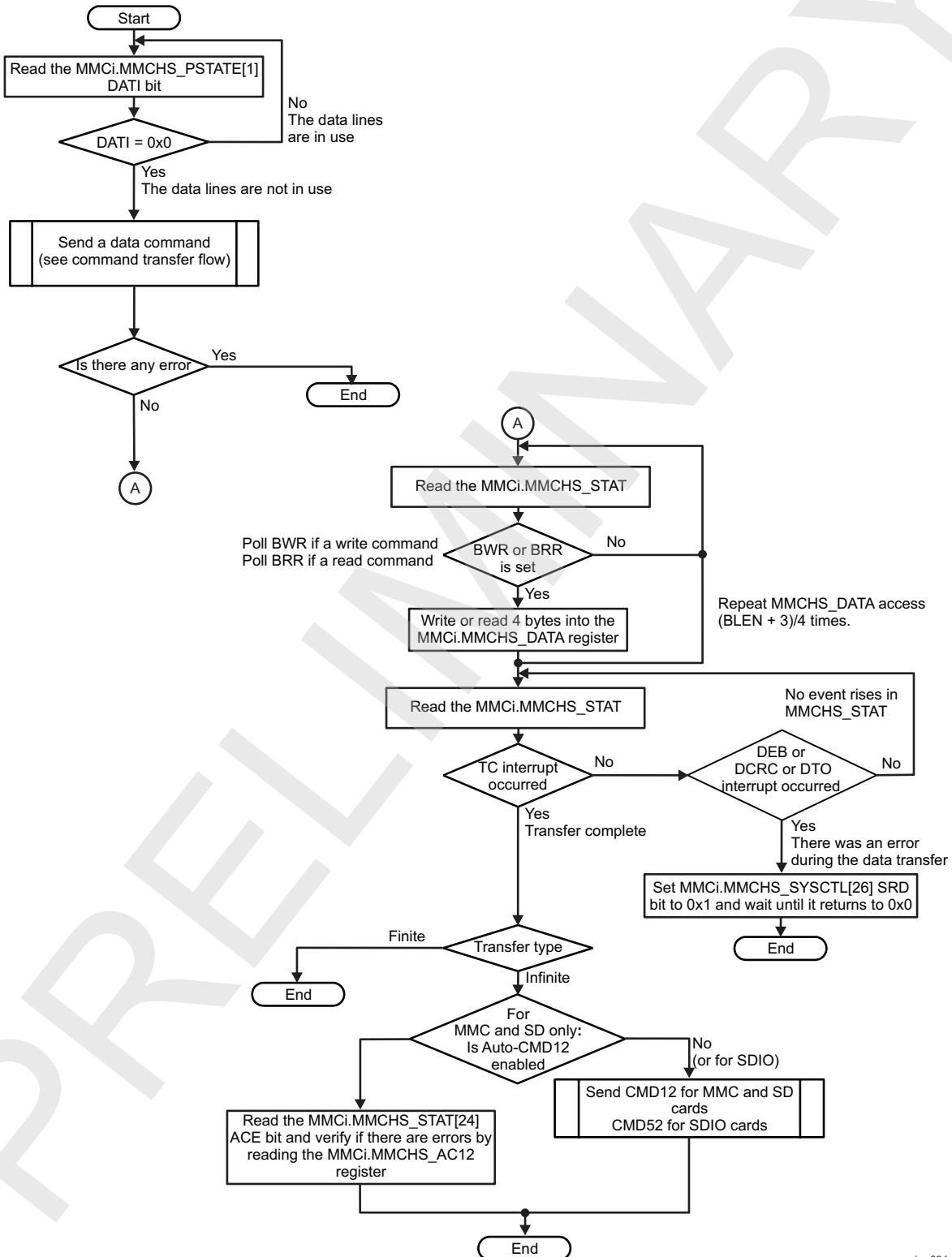
**Table 24-31. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling**

Sub-process Name	Cross-Reference
Send command	<a href="#">Section 24.5.1.2.1.7.1</a>

24.5.1.2.1.4 Read/Write Transfer Flow Without DMA With Polling

Figure 24-42 shows a read/write transfer without using the DMA and with polling.

Figure 24-42. MMC/SD/SDIO Controller Read/Write Transfer Flow Without DMA With Polling



mmchs-034

**Table 24-32. Register Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling**

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_DATA	MMCHS_STAT
MMCHS_SYSCTL	MMCHS_CMD	MMCHS_AC12

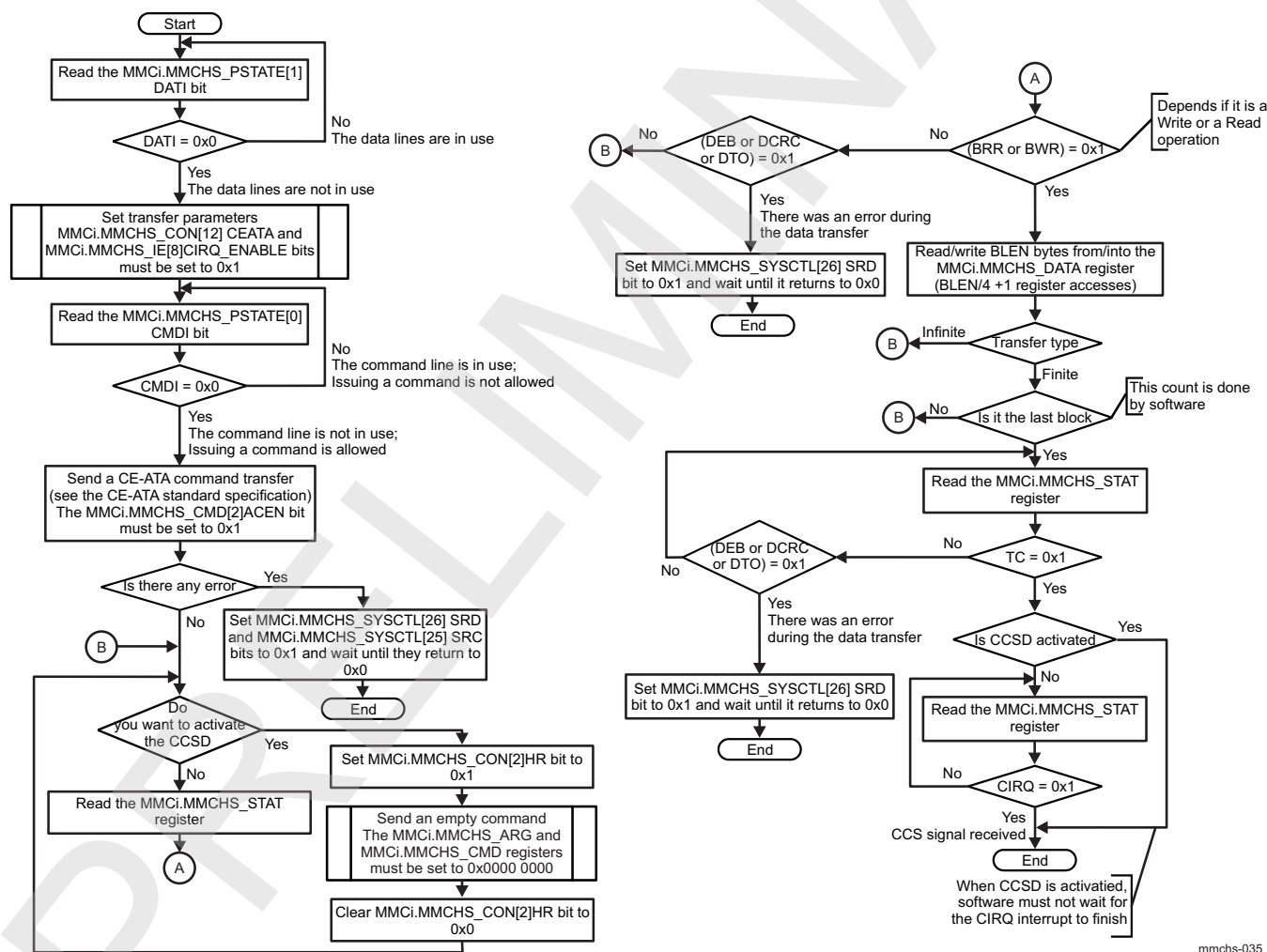
**Table 24-33. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling**

Subprocess Name	Cross-Reference
Send data command	Section 24.5.1.2.1.7.1

**24.5.1.2.1.5 Read/Write Transfer Flow in CE-ATA Mode**

Figure 24-43 shows the read and write CE-ATA protocol when in polling mode.

**Figure 24-43. MMC/SD/SDIO Controller Read/Write in CE-ATA Mode**



**Table 24-34. Register Call Summary for Main Sequence – Read/Write in CE-ATA Mode**

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_CON	MMCHS_IE
MMCHS_CMD	MMCHS_STAT	
MMCHS_ARG	MMCHS_SYSCTL	

**CAUTION**

CE-ATA protocol is supported only by MMC cards.

In CE-ATA mode, issuing a command during the transfer (except a CCSD command) is not allowed.

In CE-ATA mode, infinite transfers are not allowed; only finite transfers are permitted.

**24.5.1.2.1.6 Suspend-Resume Flow**

The suspend and resume feature is supported only by SDIO cards.

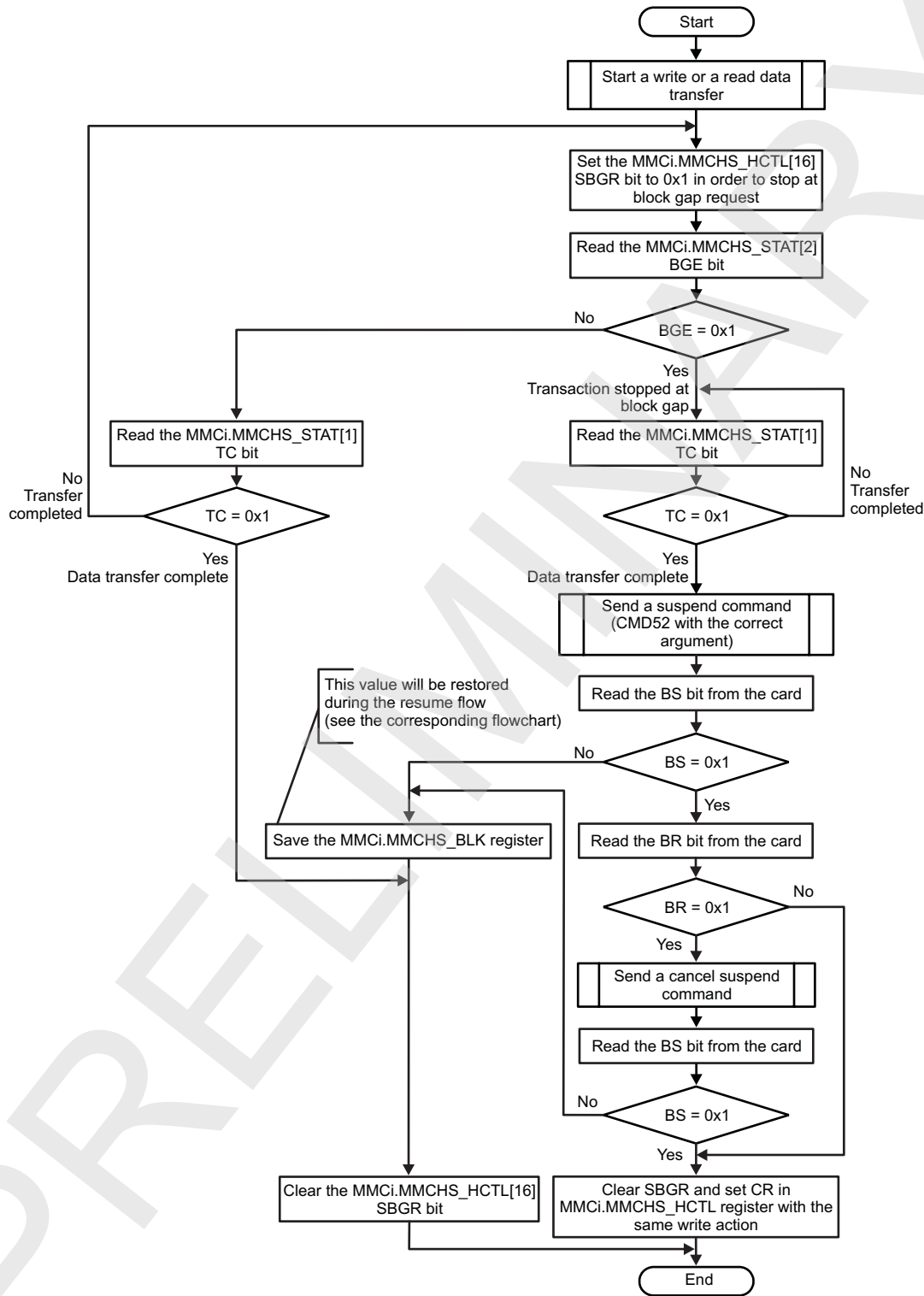
**24.5.1.2.1.6.1 Suspend Flow**

[Figure 24-44](#) shows the suspend flow for SDIO cards.

PRELIMINARY



Figure 24-44. MMC/SD/SDIO Controller Suspend Flow



mmchs-036

Table 24-35. Register Call Summary for Main Sequence – Suspend Flow

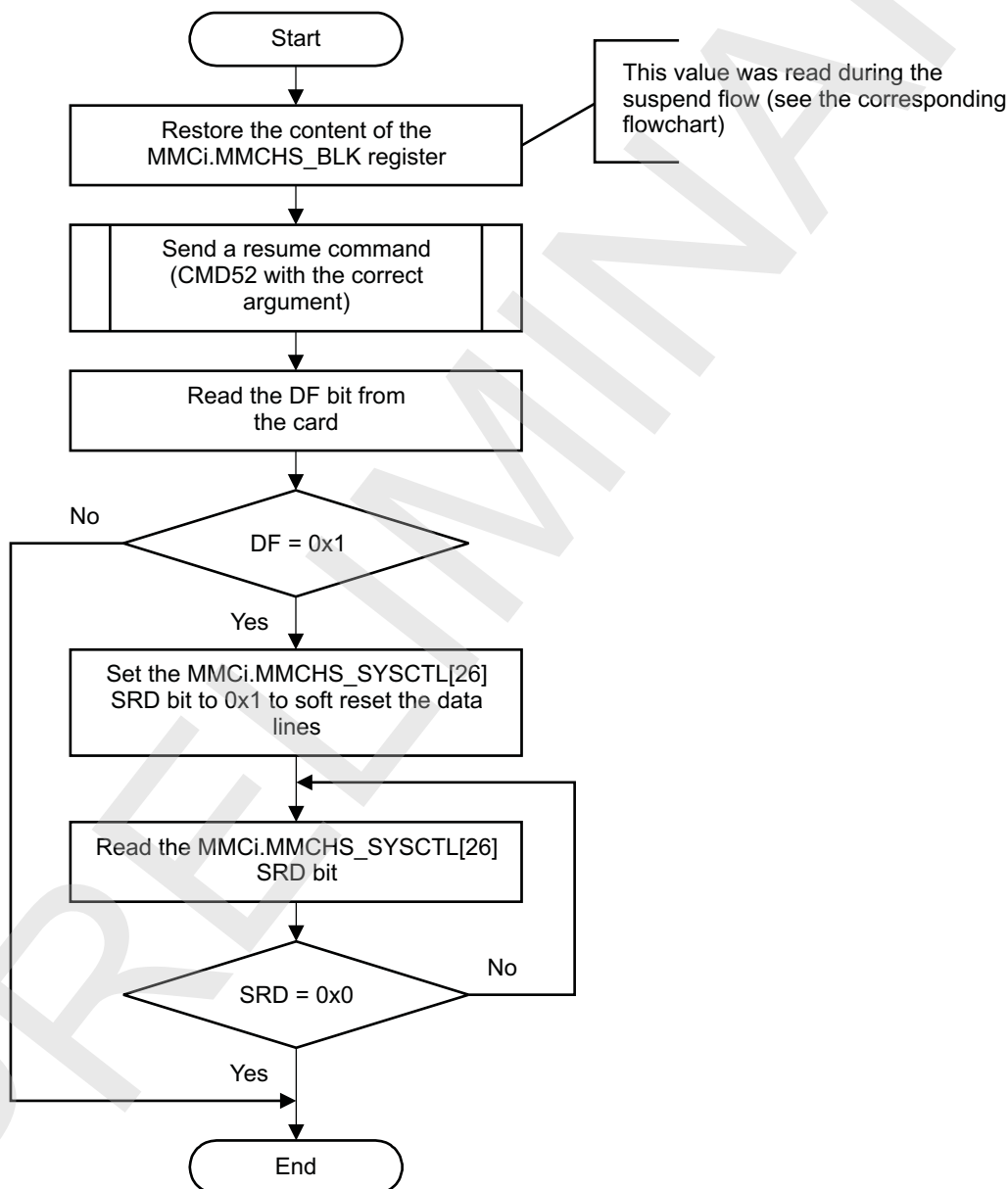
Register Name	Register Name	Register Name
MMCHS_HCTL	MMCHS_STAT	MMCHS_BLK

**Table 24-36. Subprocess Call Summary for Main Sequence – Suspend Flow**

Subprocess Name	Cross-Reference
Start a write or a read data transfer.	<a href="#">Section 24.5.1.2.1</a>
Send a suspend command (CMD52 with the correct argument).	<a href="#">Section 24.5.1.2.1.7.1</a>
Send a cancel suspend command.	<a href="#">Section 24.5.1.2.1.7.1</a>

**24.5.1.2.1.6.2 Resume Flow**

Figure 24-45 shows the resume flow for SDIO cards.

**Figure 24-45. MMC/SD/SDIO Controller Resume Flow**

mmchs-037

**Table 24-37. Register Call Summary for Main Sequence - Resume Flow**

Register Name	Register Name
<a href="#">MMCHS_BLK</a>	<a href="#">MMCHS_SYSCTL</a>

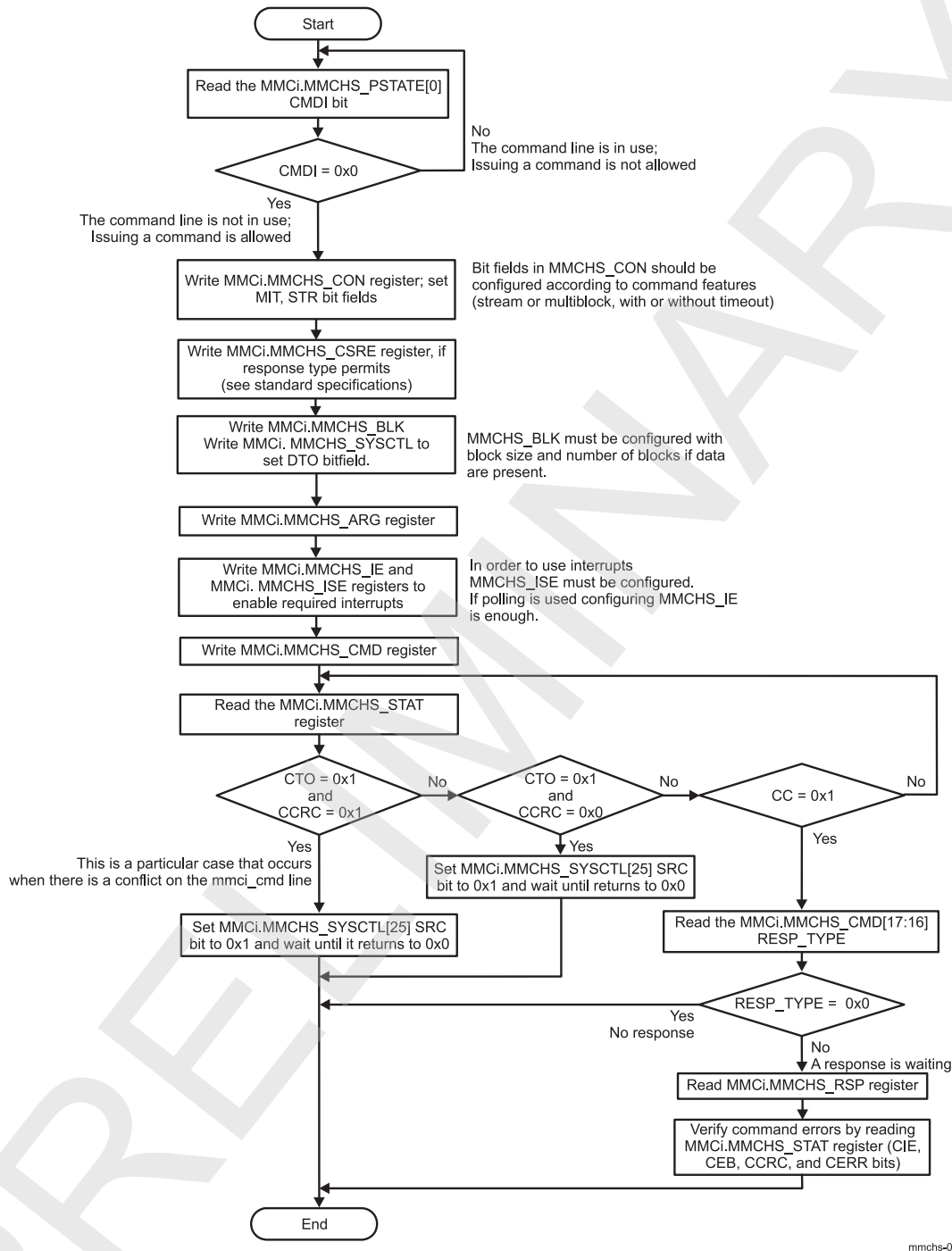
**Table 24-38. Subprocess Call Summary for Main Sequence - Resume Flow**

Subprocess Name	Cross-Reference
Send a resume command (CMD52 with the correct argument).	<a href="#">Section 24.5.1.2.1.7.1</a>

### **24.5.1.2.1.7 Basic Operations – Steps Detailed**

#### **24.5.1.2.1.7.1 Command Transfer Flow**

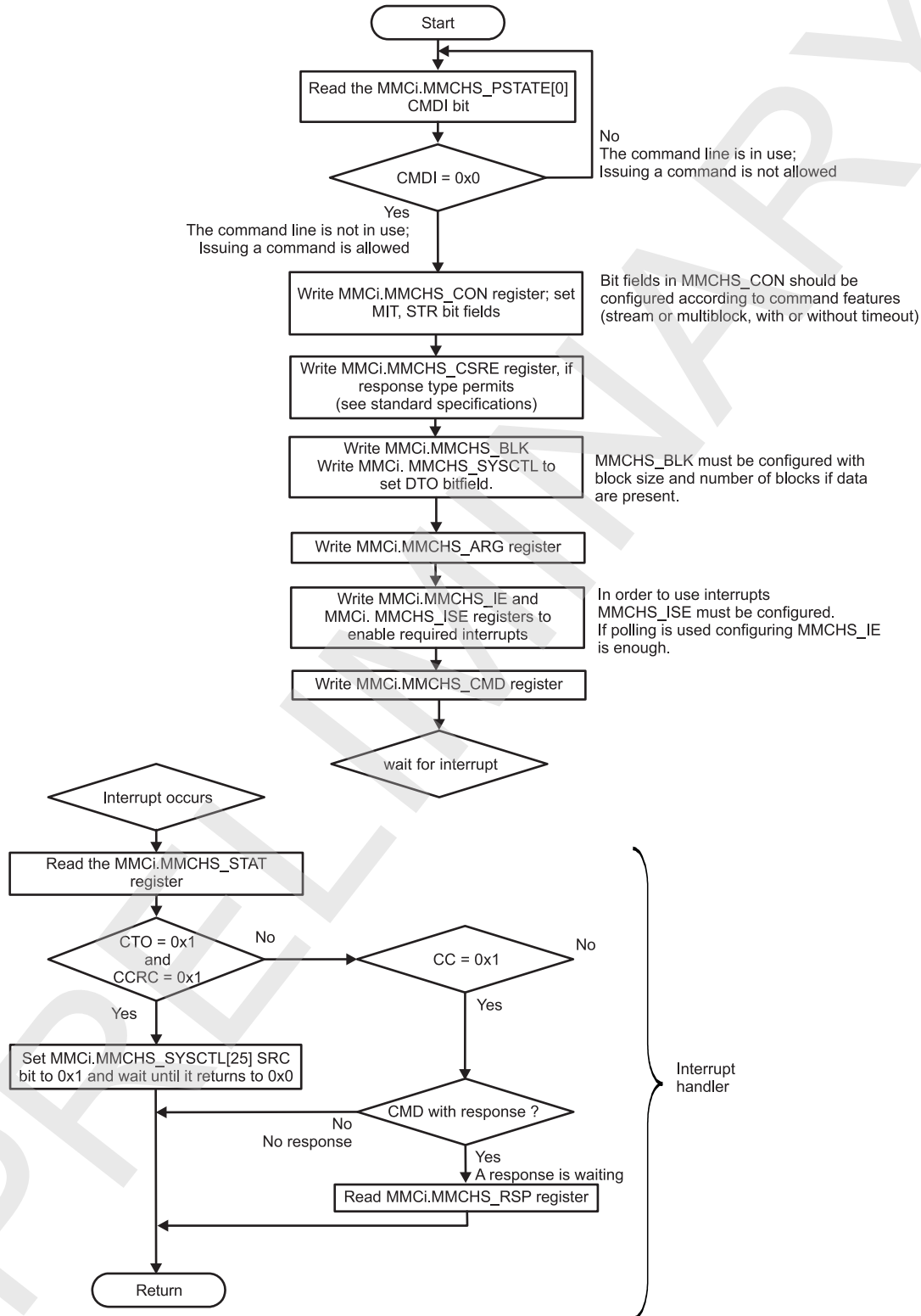
[Figure 24-46](#) shows how to send a command to the card using polling instead of interrupts for event signaling.

**Figure 24-46. MMC/SD/SDIO Controller Command Transfer Flow With Polling****Table 24-39. Register Call Summary for Main Sequence – Command Transfer Flow With Polling**

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_CON	MMCHS_CSRE
MMCHS_STAT	MMCHS_BLK	MMCHS_SYSTL
MMCHS_ARG	MMCHS_IE	MMCHS_CMD
MMCHS_RSP10	MMCHS_RSP32	MMCHS_RSP54
MMCHS_RSP76		

Figure 24-47 shows how to send a command to the card using interrupts for event signaling.

**Figure 24-47. MMC/SD/SDIO Controller Command Transfer Flow With interrupts**



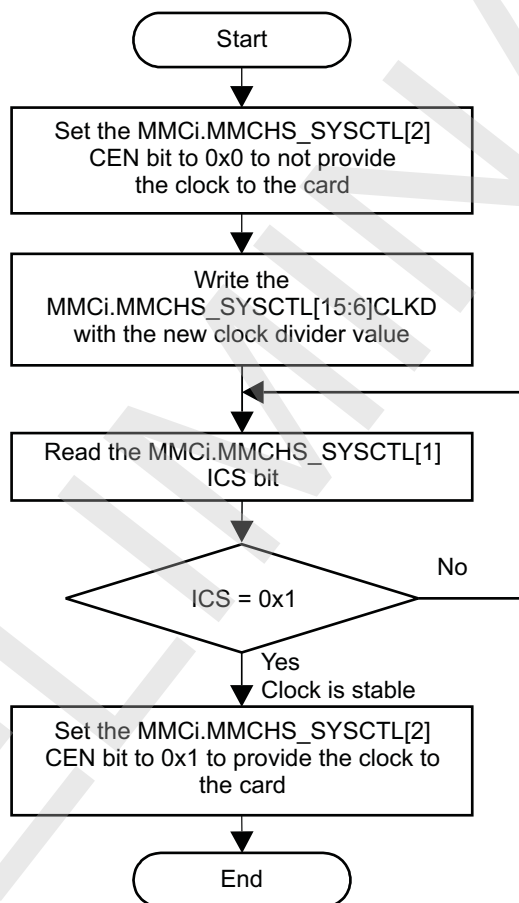
mmchs-039

**Table 24-40. Register Call Summary for Main Sequence – Command Transfer Flow With Interrupts**

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_CON	MMCHS_CSRE
MMCHS_STAT	MMCHS_BLK	MMCHS_SYSCTL
MMCHS_ARG	MMCHS_IE	MMCi.MMCHS_ISE
MMCHS_RSP10	MMCHS_RSP32	MMCHS_RSP54
MMCHS_RSP76	MMCHS_CMD	

#### 24.5.1.2.1.7.2 MMCHS Clock Frequency Change

Figure 24-48 shows the different steps that allow changing the MMC/SD/SDIO output clock frequency.

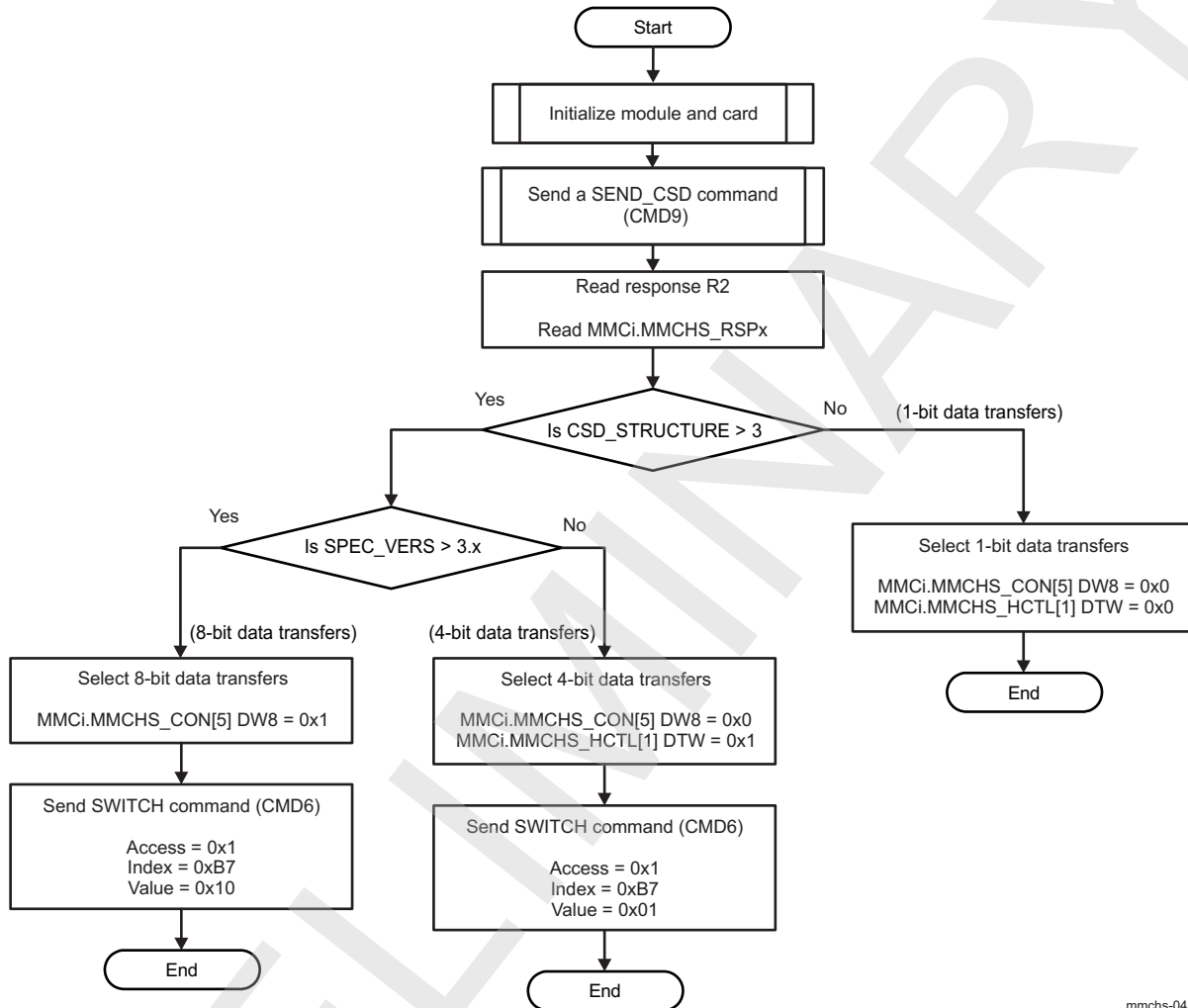
**Figure 24-48. MMC/SD/SDIO Controller Clock Frequency Change Flow****Table 24-41. Register Call Summary for Main Sequence – Clock Frequency Change Flow**

Register Name
MMCHS_SYSCTL

24.5.1.2.1.7.3 Bus Width Selection

Figure 24-49 shows the different steps that allow changing the MMC/SD/SDIO bus width.

Figure 24-49. MMC/SD/SDIO Controller Bus Width Configuration Flow



mmchs-041

Table 24-42. Register Call Summary for Main Sequence – Bus Width Configuration Flow

Register Name	Register Name	Register Name
MMCHS_RSP10	MMCHS_RSP32	MMCHS_RSP54
MMCHS_RSP76	MMCHS_CON	MMCHS_HCTL

Table 24-43. Subprocess Call Summary for Main Sequence – Bus Width Configuration Flow

Subprocess Name	Cross-Reference
Initialize module and card.	Section 24.5.1.1.2 Section 24.5.1.2.1.1
Send a SEND_CSD command (CMD9).	Section 24.5.1.2.1.7.1
Send SWITCH command (CMD6).	Section 24.5.1.2.1.7.1

### 24.5.1.2.2 Bus Voltage Selection

The MMC/SD/SDIO1 controller can operate with two types of card voltages: 1.8 V and 3.0 V. For this reason, dual voltage pads are implemented on this interface. For technological concerns, those pads must have an internal bias voltage reference to operate. The PBIAS module supplies this bias voltage, depending on the CONTROL.CONTROL\_PBIAS\_LITE register settings.

For more information about the PBIAS\_LITE cell, see [Section 18.4.9](#), *Extended-Drain I/O and PBIAS Cell*, in [Chapter 18](#), *Control Module*.

For the steps involved in transitioning from 1.8 V to 3.0 V and from 3.0 V to 1.8 V, applicable to the MMC/SD/SDIO1 controller, see [Section 18.5.1.2.2](#), *Extended-Drain I/Os and PBIAS Cell Programming Guide*, in [Chapter 18](#), *Control Module*.

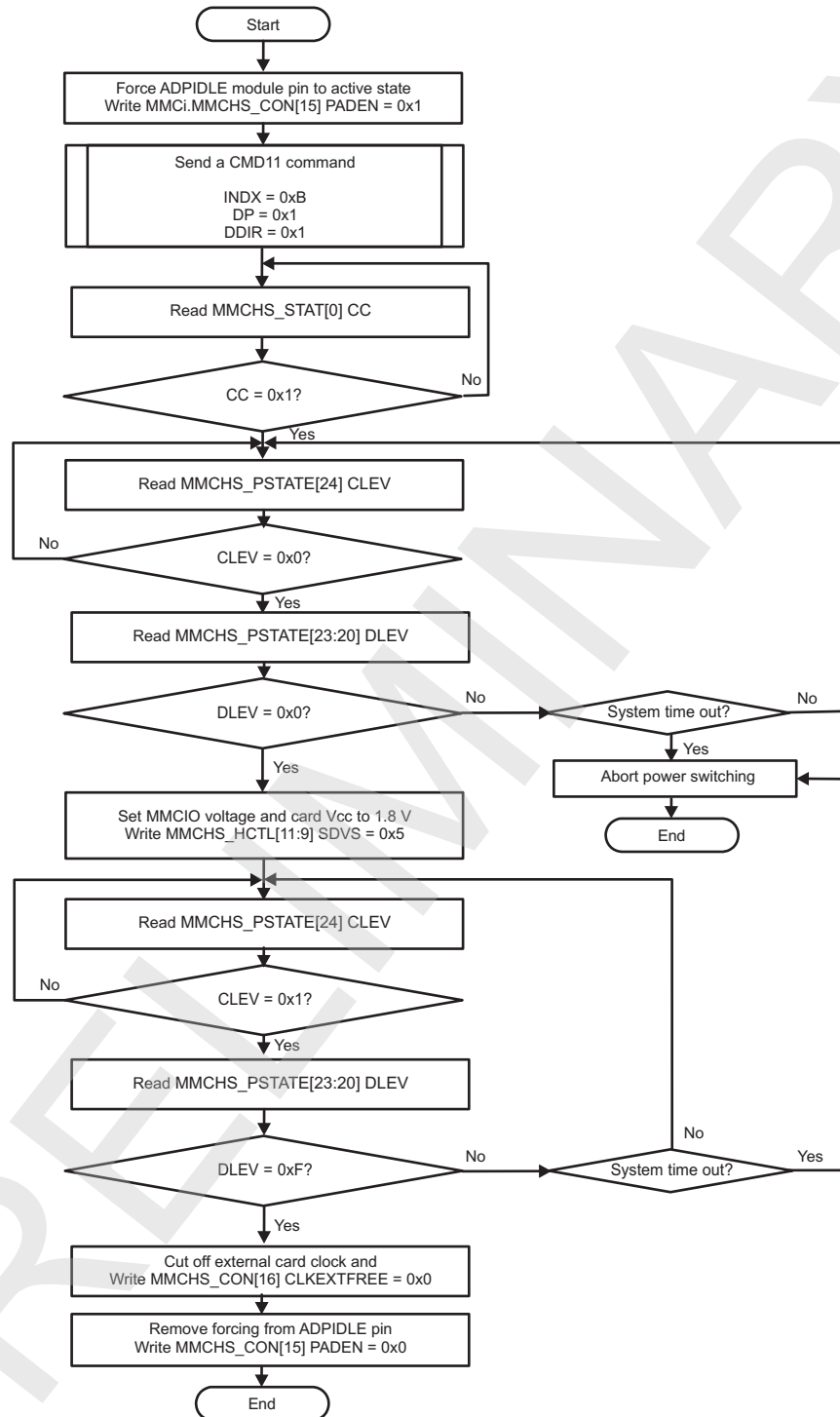
#### CAUTION

The BIAS voltage must be set using the procedure described in [Section 18.5.1.2.2](#), *Extended-Drain I/Os and PBIAS Cell Programming Guide*, in [Chapter 18](#), *Control Module*. Failure to follow this procedure can damage the MMCHS interface.

[Figure 24-50](#) shows how to configure the MMCHS controller to fit with power switching sequence.



Figure 24-50. MMC/SD/SDIO Power Switching Procedure



mmchs-042

Table 24-44. Register Call Summary for Main Sequence – Power Switching Procedure

Register Name	Register Name
MMCHS_STAT	MMCHS_PSTATE
MMCHS_CMD	MMCHS_CON

**Table 24-45. Subprocess Call Summary for Main Sequence – Power Switching Procedure**

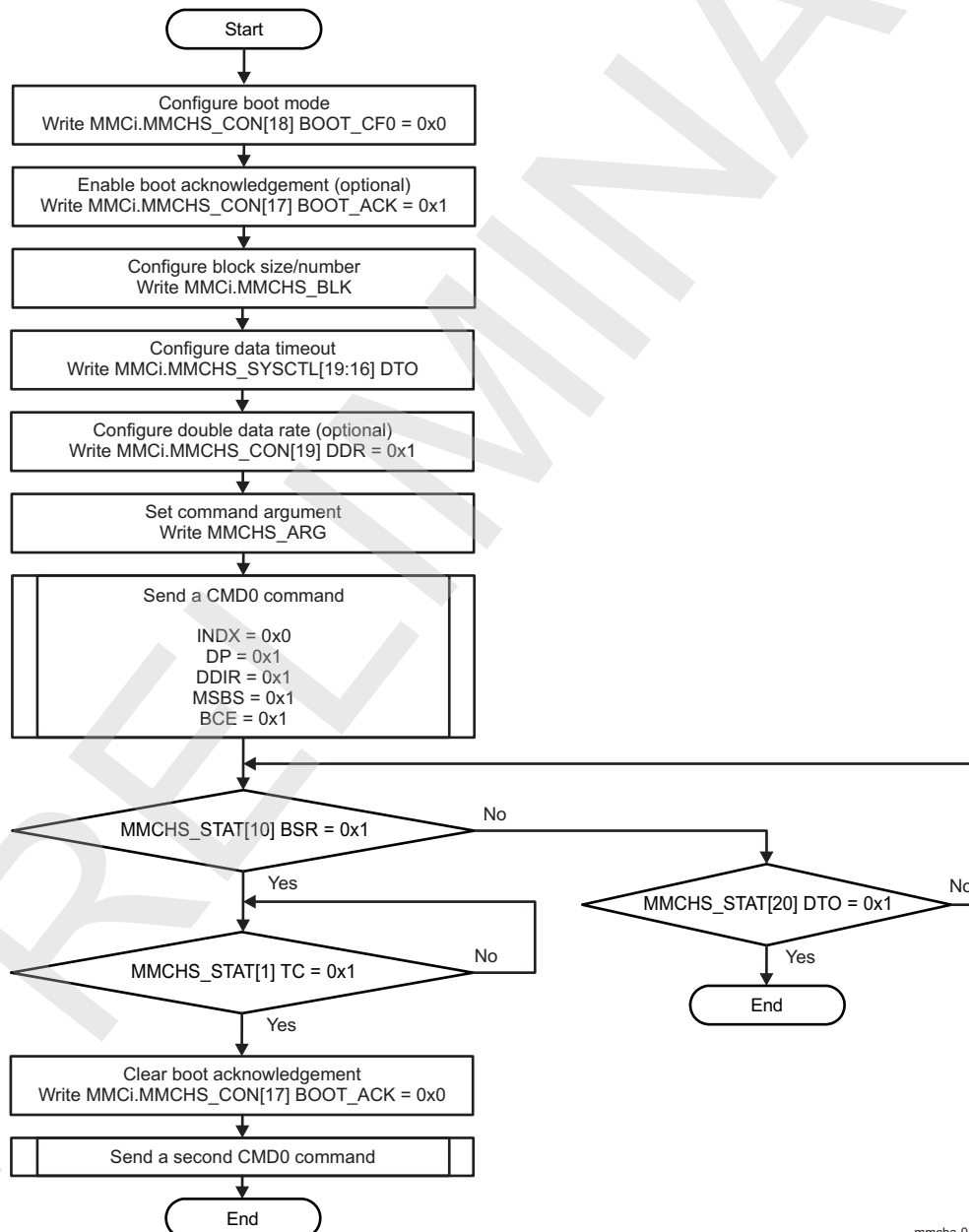
Subprocess Name	Cross-Reference
Send a READ_DAT_UNTIL_STOP command (CMD11).	<a href="#">Section 24.5.1.2.1.7.1</a>

### 24.5.1.2.3 Boot Mode Configuration

The following sections describe the two possible ways to issue a boot command: issue a CMD0 or drive the CMD line to 0 during the whole boot phase.

#### 24.5.1.2.3.1 Boot Using CMD0

Figure 24-51 shows the necessary steps to configure the controller boot mode using CMD0.

**Figure 24-51. MMC/SD/SDIO Controller Boot Using CMD0**

mmchs-043

To abort a boot sequence, the system must issue a CMD0 with [MMCHS\\_CMD\[23:22\]](#) CMD\_TYPE bit field set to 0x3 (the [MMCHS\\_CON\[17\]](#) BOOT\_ACK bit previously cleared to 0x0) during the transfer to abort the transfer and enable the card to exit from boot state.

**Table 24-46. Register Call Summary for Main Sequence – Boot Using CMD0**

Register Name	Register Name	Register Name
<a href="#">MMCHS_CON</a>	<a href="#">MMCHS_BLK</a>	<a href="#">MMCHS_SYSCTL</a>
<a href="#">MMCHS_ARG</a>	<a href="#">MMCHS_STAT</a>	

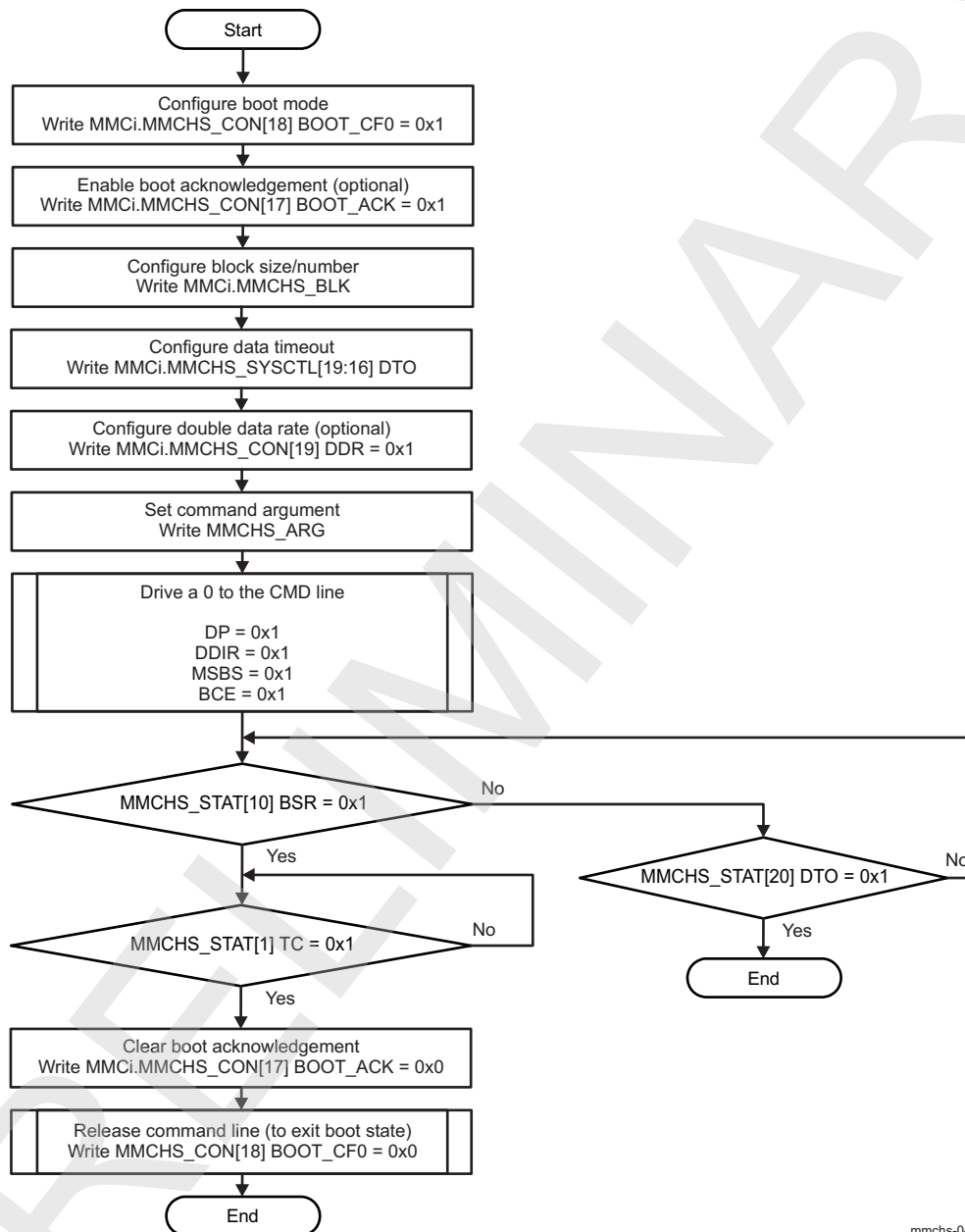
**Table 24-47. Subprocess Call Summary for Main Sequence – Boot Using CMD0**

Subprocess Name	Cross-Reference
Send a CMD0 command.	<a href="#">Section 24.5.1.2.1.7.1</a>

### 24.5.1.2.3.2 Boot With CMD Line Tied to 0

Figure 24-52 shows the necessary steps to configure the controller in this mode; the driver must follow this sequence.

**Figure 24-52. MMC/SD/SDIO Controller Boot With CMD Line Tied to 0**



mmchs-044

To abort the boot sequence, the system must clear the [MMCHS\\_CON\[18\] BOOT\\_CF0](#) bit to 0x0 during the transfer to abort the transfer and enable the card to exit from boot state.

**Table 24-48. Register Call Summary for Main Sequence – Boot Using CMD0**

Register Name	Register Name	Register Name
<a href="#">MMCHS_CON</a>	<a href="#">MMCHS_BLK</a>	<a href="#">MMCHS_SYSCCTL</a>
<a href="#">MMCHS_ARG</a>	<a href="#">MMCHS_STAT</a>	<a href="#">MMCHS_CMD</a>

**Table 24-49. Subprocess Call Summary for Main Sequence – Boot Using CMD0**

<b>Subprocess Name</b>	<b>Cross-Reference</b>
Send a CMD0 command.	<a href="#">Section 24.5.1.2.1.7.1</a>

PRELIMINARY

## 24.6 MMC/SD/SDIO Register Manual

### 24.6.1 MMC/SD/SDIO Instance Summary

**Table 24-50. MMC/SD/SDIO Instance Summary**

Module Name	Base Address	Size
MMCHS1	0x4809 C000	4KB
MMCHS2	0x480B 4000	4KB
MMCHS3	0x480A D000	4KB
MMCHS4	0x480D 1000	4KB
MMCHS5	0x480D 5000	4KB

### 24.6.2 MMC/SD/SDIO Registers

#### 24.6.2.1 MMC/SD/SDIO Register Summary

**Table 24-51. MMC/SD/SDIO Registers Mapping Summary 1**

Register Name	Type	Register Width (Bits)	Address Offset	MMCHS1 Physical Address	MMCHS2 Physical Address	MMCHS3 Physical Address
MMCHS_HL_REV	R	32	0x0000 0000	0x4809 C000	0x480B 4000	0x480A D000
MMCHS_HL_HWINFO	R	32	0x0000 0004	0x4809 C004	0x480B 4004	0x480A D004
MMCHS_HL_SYSCONFIG	RW	32	0x0000 0010	0x4809 C010	0x480B 4010	0x480A D010
MMCHS_SYSCONFIG	RW	32	0x0000 0110	0x4809 C110	0x480B 4110	0x480A D110
MMCHS_SYSSTATUS	R	32	0x0000 0114	0x4809 C114	0x480B 4114	0x480A D114
MMCHS_CSRE	RW	32	0x0000 0124	0x4809 C124	0x480B 4124	0x480A D124
MMCHS_SYSTEST	RW	32	0x0000 0128	0x4809 C128	0x480B 4128	0x480A D128
MMCHS_CON	RW	32	0x0000 012C	0x4809 C12C	0x480B 412C	0x480A D12C
MMCHS_PWCNT	RW	32	0x0000 0130	0x4809 C130	0x480B 4130	0x480A D130
RESERVED	R	32	0x0000 0200	0x4809 C200	0x480B 4200	0x480A D200
MMCHS_BLK	RW	32	0x0000 0204	0x4809 C204	0x480B 4204	0x480A D204
MMCHS_ARG	RW	32	0x0000 0208	0x4809 C208	0x480B 4208	0x480A D208
MMCHS_CMD	RW	32	0x0000 020C	0x4809 C20C	0x480B 420C	0x480A D20C
MMCHS_RSP10	R	32	0x0000 0210	0x4809 C210	0x480B 4210	0x480A D210
MMCHS_RSP32	R	32	0x0000 0214	0x4809 C214	0x480B 4214	0x480A D214
MMCHS_RSP54	R	32	0x0000 0218	0x4809 C218	0x480B 4218	0x480A D218
MMCHS_RSP76	R	32	0x0000 021C	0x4809 C21C	0x480B 421C	0x480A D21C
MMCHS_DATA	RW	32	0x0000 0220	0x4809 C220	0x480B 4220	0x480A D220
MMCHS_PSTATE	R	32	0x0000 0224	0x4809 C224	0x480B 4224	0x480A D224
MMCHS_HCTL	RW	32	0x0000 0228	0x4809 C228	0x480B 4228	0x480A D228
MMCHS_SYSCTL	RW	32	0x0000 022C	0x4809 C22C	0x480B 422C	0x480A D22C
MMCHS_STAT	RW	32	0x0000 0230	0x4809 C230	0x480B 4230	0x480A D230
MMCHS_IE	RW	32	0x0000 0234	0x4809 C234	0x480B 4234	0x480A D234
MMCHS_ISE	RW	32	0x0000 0238	0x4809 C238	0x480B 4238	0x480A D238
MMCHS_AC12	R	32	0x0000 023C	0x4809 C23C	0x480B 423C	0x480A D23C
MMCHS_CAPA	RW	32	0x0000 0240	0x4809 C240	0x480B 4240	0x480A D240
MMCHS_CUR_CAPA	RW	32	0x0000 0248	0x4809 C248	0x480B 4248	0x480A D248
MMCHS_FE	RW	32	0x0000 0250	0x4809 C250	0x480B 4250	0x480A D250
MMCHS_ADMAES	RW	32	0x0000 0254	0x4809 C254	0x480B 4254	N/A
MMCHS_ADMASAL	RW	32	0x0000 0258	0x4809 C258	0x480B 4258	N/A

**Table 24-51. MMC/SD/SDIO Registers Mapping Summary 1 (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	MMCHS1 Physical Address	MMCHS2 Physical Address	MMCHS3 Physical Address
RESERVED	R	32	0x0000 025C	0x4809 C25C	0x480B 425C	0x480A D25C
MMCHS_REV	R	32	0x0000 02FC	0x4809 C2FC	0x480B 42FC	0x480A D2FC

**Table 24-52. MMC/SD/SDIO Registers Mapping Summary 2**

Register Name	Type	Register Width (Bits)	Address Offset	MMCHS4 Physical Address	MMCHS5 Physical Address
MMCHS_HL_REV	R	32	0x0000 0000	0x480D 1000	0x480D 5000
MMCHS_HL_HWINFO	R	32	0x0000 0004	0x480D 1004	0x480D 5004
MMCHS_HL_SYSCONFIG	RW	32	0x0000 0010	0x480D 1010	0x480D 5010
MMCHS_SYSCONFIG	RW	32	0x0000 0110	0x480D 1110	0x480D 5110
MMCHS_SYSSTATUS	R	32	0x0000 0114	0x480D 1114	0x480D 5114
MMCHS_CSRE	RW	32	0x0000 0124	0x480D 1124	0x480D 5124
MMCHS_SYSTEST	RW	32	0x0000 0128	0x480D 1128	0x480D 5128
MMCHS_CON	RW	32	0x0000 012C	0x480D 112C	0x480D 512C
MMCHS_PWCNT	RW	32	0x0000 0130	0x480D 1130	0x480D 5130
RESERVED	R	32	0x0000 0200	0x480D 1200	0x480D 5200
MMCHS_BLK	RW	32	0x0000 0204	0x480D 1204	0x480D 5204
MMCHS_ARG	RW	32	0x0000 0208	0x480D 1208	0x480D 5208
MMCHS_CMD	RW	32	0x0000 020C	0x480D 120C	0x480D 520C
MMCHS_RSP10	R	32	0x0000 0210	0x480D 1210	0x480D 5210
MMCHS_RSP32	R	32	0x0000 0214	0x480D 1214	0x480D 5214
MMCHS_RSP54	R	32	0x0000 0218	0x480D 1218	0x480D 5218
MMCHS_RSP76	R	32	0x0000 021C	0x480D 121C	0x480D 521C
MMCHS_DATA	RW	32	0x0000 0220	0x480D 1220	0x480D 5220
MMCHS_PSTATE	R	32	0x0000 0224	0x480D 1224	0x480D 5224
MMCHS_HCTL	RW	32	0x0000 0228	0x480D 1228	0x480D 5228
MMCHS_SYSCTL	RW	32	0x0000 022C	0x480D 122C	0x480D 522C
MMCHS_STAT	RW	32	0x0000 0230	0x480D 1230	0x480D 5230
MMCHS_IE	RW	32	0x0000 0234	0x480D 1234	0x480D 5234
MMCHS_ISE	RW	32	0x0000 0238	0x480D 1238	0x480D 5238
MMCHS_AC12	R	32	0x0000 023C	0x480D 123C	0x480D 523C
MMCHS_CAPA	RW	32	0x0000 0240	0x480D 1240	0x480D 5240
MMCHS_CUR_CAPA	RW	32	0x0000 0248	0x480D 1248	0x480D 5248
MMCHS_FE	RW	32	0x0000 0250	0x480D 1250	0x480D 5250
MMCHS_ADMAES	R	32	0x0000 0254	N/A	N/A
MMCHS_ADMASAL	RW	32	0x0000 0258	N/A	N/A
RESERVED	R	32	0x0000 025C	0x480D 125C	0x480D 525C
MMCHS_REV	R	32	0x0000 02FC	0x480D 12FC	0x480D 52FC

### 24.6.2.2 MMC/SD/SDIO Register Description

**Table 24-53. MMCHS\_HL\_REV**

<b>Address Offset</b>	0x0000 0000		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision.	R	See <sup>(1)</sup>

<sup>(1)</sup> TI internal data.

**Table 24-54. Register Call Summary for Register MMCHS\_HL\_REV**

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[0\] \[1\]](#)

**Table 24-55. MMCHS\_HL\_HWINFO**

<b>Address Offset</b>	0x0000 0004		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	Information about the IP module's hardware configuration, that is, typically the module's HDL generics (if any). Actual field format and encoding is up to the module's designer to decide.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RETMODE	MEM_SIZE				MERGE_MEM	MADMA_EN									

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x00000000
6	RETMODE	Retention mode generic parameter This bit field indicates whether the retention mode is supported using the pin PIRFFRET.  0x0: Retention mode disabled 0x1: Retention mode enabled	R	See <a href="#">Table 24-57</a> .
5:2	MEM_SIZE	Memory size for FIFO buffer:  Read 0x1: Memory of 512 bytes, max block length is 512 bytes Read 0x2: Memory of 1024 bytes, max block length is 1024 bytes Read 0x4: Memory of 2048 bytes, max block length is 2048 bytes Read 0x8: Memory of 4096 bytes, max block length is 2048 bytes	R	0x2



Bits	Field Name	Description	Type	Reset
1	MERGE_MEM	Memory merged for FIFO buffer: This register defines the configuration of FIFO buffer architecture. If the bit is set STA and DFT shall support clock multiplexing and balancing.  Read 0x0: 2 memories instantiated, one per data transfer direction.  Read 0x1: A single memory is used with multiplexed addresses, data and clocks.	R	0
0	MADMA_EN	Master DMA enabled generic parameter: This register defines the configuration of the controller to know if it supports the master DMA management called ADMA.  Read 0x0: No Master DMA (ADMA) management supported  Read 0x1: Controller supports ADMA	R	Please refer to <a href="#">Table 24-58</a>

**Table 24-56. Register Call Summary for Register MMCHS\_HL\_HWINFO**

MMC/SD/SDIO Functional Description

- [DMA Modes: \[0\] \[1\]](#)
- [MASTER DMA Operations: \[2\]](#)
- [SLAVE DMA Operations: \[3\]](#)
- [Memory size, Block Length and Buffer Management Relationship: \[4\]](#)
- [Retention Mode: \[5\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[6\] \[7\]](#)
- [MMC/SD/SDIO Register Description: \[8\] \[9\] \[10\] \[11\] \[12\]](#)

**Table 24-57. Reset value for MMCHS\_HL\_HWINFO**

Instance	RETMODE
MMCHS1	1
MMCHS2	1
MMCHS3	0
MMCHS4	0
MMCHS5	0

**Table 24-58. Reset value for MADMA\_EN**

Instance	RETMODE
MMCHS1	1
MMCHS2	1
MMCHS3	0
MMCHS4	0
MMCHS5	0

**Table 24-59. MMCHS\_HL\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	Clock management configuration		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							STANDBYMODE	IDLEMODE	FREEEMU	SOFTRESET					

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0000000
5:4	STANDBYMODE	<p>Configuration of the local initiator state management mode.</p> <p>By definition, initiator may generate read/write transaction as long as it is out of STANDBY state.</p> <p>0x0: Force-standby mode: local initiator is unconditionally placed in standby state.Backup mode, for debug only.</p> <p>0x1: No-standby mode: local initiator is unconditionally placed out of standby state.Backup mode, for debug only.</p> <p>0x2: Smart-standby mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator.IP module shall not generate (initiator-related) wakeup events.</p> <p>0x3: Smart-Standby wakeup-capable mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator. IP module may generate (master-related) wakeup events when in standby state.Mode is only relevant if the appropriate IP module "mwakeup" output is implemented.</p>	RW	0x2
3:2	IDLEMODE	<p>Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.</p> <p>0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements.Backup mode, for debug only.</p> <p>0x1: No-idle mode: local target never enters idle state.Backup mode, for debug only.</p> <p>0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements.IP module shall not generate (IRQ- or DMA-request-related) wakeup events.</p> <p>0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements.IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.</p>	RW	0x2
1	FREEEMU	<p>Sensitivity to emulation (debug) suspend input signal. Functionality NOT implemented in MMCHS.</p> <p>0x0: IP module is sensitive to emulation suspend</p> <p>0x1: IP module is not sensitive to emulation suspend</p>	RW	0

Bits	Field Name	Description	Type	Reset
0	SOFTRESET	Software reset. (Optional) Write 0x0: No action Read 0x0: Reset done, no pending action Read 0x1: Reset (software or other) ongoing Write 0x1: Initiate software reset	RW	0

**Table 24-60. Register Call Summary for Register MMCHS\_HL\_SYSCONFIG**

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[0\] \[1\]](#)

**Table 24-61. MMCHS\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0110	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	System Configuration Register This register allows controlling various parameters of the Interconnect interface.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STANDBYMODE	RESERVED	CLOCKACTIVITY	RESERVED	RESERVED	RESERVED	SIDLEMODE	ENAWAKEUP	SOFTRESET	AUTOIDLE						

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored. Reads return 0.	R	0x00000
13:12	STANDBYMODE	Master interface power Management, standby/wait control. The bit field is only useful when generic parameter <a href="#">MMCHS_HL_HWINFO[0] MADMA_EN</a> (Master ADMA enable) is set as active, otherwise it is a read only register read a 0. 0x0: Force-standby. Mstandby is forced unconditionally. 0x1: No-standby. Mstandby is never asserted. 0x2: Smart-standby mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator. IP module shall not generate (initiator-related) wakeup events. 0x3: Smart-Standby wakeup-capable mode: local initiator standby status depends on local conditions, that is, the module's functional requirement from the initiator. IP module may generate (master-related) wakeup events when in standby state. Mode is only relevant if the appropriate IP module "mwakeup" output is implemented.	RW	0x0
11:10	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored. Reads return 0.	R	0x0

Bits	Field Name	Description	Type	Reset
9:8	CLOCKACTIVITY	<p>Clocks activity during wake up mode period.</p> <p>Bit8: Interface clock Bit9: Functional clock</p> <p>0x0: Interface and Functional clock may be switched off. 0x1: Interface clock is maintained. Functional clock may be switched-off. 0x2: Functional clock is maintained. Interface clock may be switched-off. 0x3: Interface and Functional clocks are maintained.</p>	RW	0x0
7	RESERVED	<p>Reserved This bit is initialized to zero, and writes to it are ignored. Reads return 0.</p>	R	0
6	RESERVED	<p>Reserved - EmuSoft not supported This bit is initialized to zero, and writes to it are ignored. Reads return 0.</p>	R	0
5	RESERVED	<p>Reserved - EmuFree not supported This bit is initialized to zero, and writes to it are ignored. Reads return 0.</p>	R	0
4:3	SIDLEMODE	<p>Power management</p> <p>0x0: If an idle request is detected, the MMCHS acknowledges it unconditionally and goes in Inactive mode. Interrupt and DMA requests are unconditionally de-asserted. 0x1: If an idle request is detected, the request is ignored and the module keeps on behaving normally. 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events. 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.</p>	RW	0x2
2	ENAWAKEUP	<p>Wakeup feature control</p> <p>0x0: Wakeup capability is disabled 0x1: Wakeup capability is enabled</p>	RW	1
1	SOFTRESET	<p>Software reset. The bit is automatically reset by the hardware. During reset, it always returns 0.</p> <p>Read 0x0: Normal mode Write 0x0: No effect. Write 0x1: Trigger a module reset. Read 0x1: The module is reset.</p>	RW	0
0	AUTOIDLE	<p>Internal Clock gating strategy</p> <p>0x0: Clocks are free-running 0x1: Automatic clock gating strategy is applied, based on the Interconnect and MMC interface activity</p>	RW	1

**Table 24-62. Register Call Summary for Register MMCHS\_SYSCONFIG**

MMC/SD/SDIO Functional Description

- [Software Reset: \[0\]](#)
- [Power Management: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\]](#)

**Table 24-62. Register Call Summary for Register MMCHS\_SYSCONFIG (continued)**

MMC/SD/SDIO Programming Guide

- [MMCHS Soft Reset Flow: \[13\]](#)
- [Wakeup Configuration: \[14\]](#)
- [MMC Host and Bus Configuration: \[15\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[16\] \[17\]](#)
- [MMC/SD/SDIO Register Description: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\]](#)

**Table 24-63. MMCHS\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0114	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	System Status Register This register provides status information about the module excluding the interrupt status information		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESETDONE															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored. Reads return 0.	R	0x0000 0000
0	RESETDONE	Internal Reset Monitoring Note: the debounce clock , the system clock (Interface) and the functional clock shall be provided to the MMC/SD/SDIO host controller to allow the internal reset monitoring. Read 0x0: Internal module reset is on-going Read 0x1: Reset completed.	R	0

**Table 24-64. Register Call Summary for Register MMCHS\_SYSSTATUS**

MMC/SD/SDIO Functional Description

- [Hardware Reset: \[0\]](#)
- [Software Reset: \[1\]](#)

MMC/SD/SDIO Programming Guide

- [MMCHS Soft Reset Flow: \[2\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[3\] \[4\]](#)

**Table 24-65. MMCHS\_CSRE**

<b>Address Offset</b>	0x0000 0124		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	<p>Card status response error</p> <p>This register enables the host controller to detect card status errors of response type R1, R1b for all cards and of R5, R5b and R6 response for cards types SD or SDIO.</p> <p>When a bit <a href="#">MMCHS_CSRE[i]</a> is set to 1, if the corresponding bit at the same position in the response <a href="#">MMCHS_RSP10[i]</a> is set to 1, the host controller indicates a card error (<a href="#">MMCHS_STAT[CERR]</a>) interrupt status to avoid the host driver reading the response register (<a href="#">MMCHS_RSP10</a>).</p> <p>Note: No automatic card error detection for autoCMD12 is implemented; the host system has to check autoCMD12 response register (<a href="#">MMCHS_RSP76</a>) for possible card errors.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSRE																															

Bits	Field Name	Description	Type	Reset
31:0	CSRE	Card status response error	RW	0x0000 0000

**Table 24-66. Register Call Summary for Register MMCHS\_CSRE**

MMC/SD/SDIO Functional Description

- [Interrupt Requests: \[0\]](#)

MMC/SD/SDIO Programming Guide

- [Basic Operations - Steps Detailed: \[1\] \[2\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[3\] \[4\]](#)
- [MMC/SD/SDIO Register Description: \[5\] \[6\]](#)

**Table 24-67. MMCHS\_SYSTEST**

<b>Address Offset</b>	0x0000 0128		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	<p>System Test register</p> <p>This register is used to control the signals that connect to I/O pins when the module is configured in system test (SYSTEST) mode for boundary connectivity verification.</p> <p>Note: In SYSTEST mode, a write into <a href="#">MMCHS_CMD</a> register will not start a transfer. The buffer behaves as a stack accessible only by the local host (push and pop operations). In this mode, the Transfer Block Size (<a href="#">MMCHS_BLK[11:0]</a> BLEN) and the Blocks count for current transfer (<a href="#">MMCHS_BLK[31:16]</a> NBLK) are needed to generate a Buffer write ready interrupt (<a href="#">MMCHS_STAT[4]</a> BWR) or a Buffer read ready interrupt (<a href="#">MMCHS_STAT[5]</a> BRR) and DMA requests if enabled.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED																OBI	SDCD	SDWP	WAKD	SSB	D7D	D6D	D5D	D4D	D3D	D2D	D1D	D0D	DDIR	CDAT	CDIR	MCKD

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored. Reads return 0.	R	0x0000
16	OBI	Out-Of-Band Interrupt (OBI) data value  Read 0x0: The Out-of-Band Interrupt pin is driven low. Read 0x1: The Out-of-Band Interrupt pin is driven high.	RW	0

Bits	Field Name	Description	Type	Reset
15	SDCD	Card detect input signal (SDCD) data value Read 0x0: The card detect pin is driven low. Read 0x1: The card detect pin is driven high.	R	0
14	SDWP	Write protect input signal (SDWP) data value Read 0x0: The write protect pin SDWP is driven low. Read 0x1: The write protect pin SDWP is driven high.	R	0
13	WAKD	Wake request output signal data value Write 0x0: The pin SWAKEUP is driven low. Read 0x0: No action. Returns 0. Write 0x1: The pin SWAKEUP is driven high. Read 0x1: No action. Returns 1.	RW	0
12	SSB	Set status bit This bit must be cleared prior attempting to clear a status bit of the interrupt status register ( <a href="#">MMCHS_STAT</a> ). Write 0x0: Clear this SSB bit field. Writing 0 does not clear already set status bits; Read 0x0: No action. Returns 0. Read 0x1: No action. Returns 1. Write 0x1: Force to 1 all status bits of the interrupt status register ( <a href="#">MMCHS_STAT</a> ) only if the corresponding bit field in the Interrupt signal enable register ( <a href="#">MMCHS_ISE</a> ) is set.	RW	0
11	D7D	DAT7 input/output signal data value Read 0x0: If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 1 (input mode direction), returns the value on the DAT7 line (low). If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 0 (output mode direction), returns 0 Write 0x0: If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 0 (output mode direction), the DAT7 line is driven low. If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 1 (input mode direction), no effect. Read 0x1: If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 1 (input mode direction), returns the value on the DAT7 line (high) If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 0 (output mode direction), returns 1 Write 0x1: If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 0 (output mode direction), the DAT7 line is driven high. If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 1 (input mode direction), no effect.	RW	0
10	D6D	DAT6 input/output signal data value Read 0x0: If <a href="#">SYSTEST</a> [DDIR] = 1 (input mode direction), returns the value on the DAT6 line (low). If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 0 (output mode direction), returns 0 Write 0x0: If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 0 (output mode direction), the DAT6 line is driven low. If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 1 (input mode direction), no effect. Read 0x1: If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 1 (input mode direction), returns the value on the DAT6 line (high) If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 0 (output mode direction), returns 1 Write 0x1: If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 0 (output mode direction), the DAT6 line is driven high. If <a href="#">MMCHS_SYSTEST</a> [3] DDIR = 1 (input mode direction), no effect.	RW	0

Bits	Field Name	Description	Type	Reset
9	D5D	<p>DAT5 input/output signal data value</p> <p>Read 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT5 line (low). If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 0</p> <p>Write 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT5 line is driven low. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p> <p>Read 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT5 line (high) If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 1</p> <p>Write 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT5 line is driven high. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p>	RW	0
8	D4D	<p>DAT4 input/output signal data value</p> <p>Write 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT4 line is driven low. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p> <p>Read 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT4 line (low). If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 0</p> <p>Write 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT4 line is driven high. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p> <p>Read 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT4 line (high) If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 1</p>	RW	0
7	D3D	<p>DAT3 input/output signal data value</p> <p>Write 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT3 line is driven low. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p> <p>Read 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT3 line (low). If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 0</p> <p>Read 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT3 line (high) If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 1</p> <p>Write 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT3 line is driven high. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p>	RW	0



Bits	Field Name	Description	Type	Reset
6	D2D	<p>DAT2 input/output signal data value</p> <p>Write 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT2 line is driven low. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p> <p>Read 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT2 line (low). If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 0</p> <p>Write 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT2 line is driven high. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p> <p>Read 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT2 line (high) If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 1</p>	RW	0
5	D1D	<p>DAT1 input/output signal data value</p> <p>Read 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT1 line (low). If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 0</p> <p>Write 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT1 line is driven low. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p> <p>Read 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT1 line (high) If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 1</p> <p>Write 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT1 line is driven high. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p>	RW	0
4	D0D	<p>DAT0 input/output signal data value</p> <p>Write 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT0 line is driven low. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p> <p>Read 0x0: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT0 line (low). If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 0</p> <p>Write 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), the DAT0 line is driven high. If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), no effect.</p> <p>Read 0x1: If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 1 (input mode direction), returns the value on the DAT0 line (high) If <a href="#">MMCHS_SYSTEST[3]</a> DDIR = 0 (output mode direction), returns 1</p>	RW	0
3	DDIR	<p>Control of the DAT[7:0] pins direction.</p> <p>Write 0x0: The DAT lines are outputs (host to card)</p> <p>Read 0x0: No action. Returns 0.</p> <p>Read 0x1: No action. Returns 1.</p> <p>Write 0x1: The DAT lines are inputs (card to host)</p>	RW	0

Bits	Field Name	Description	Type	Reset
2	CDAT	<p>CMD input/output signal data value</p> <p>Write 0x0: If <a href="#">MMCHS_SYSTEST[1]</a> CDIR = 0 (output mode direction), the CMD line is driven low. If <a href="#">MMCHS_SYSTEST[1]</a> CDIR = 1 (input mode direction), no effect.</p> <p>Read 0x0: If <a href="#">MMCHS_SYSTEST[1]</a> CDIR = 1 (input mode direction), returns the value on the CMD line (low). If <a href="#">MMCHS_SYSTEST[1]</a> CDIR = 0 (output mode direction), returns 0</p> <p>Write 0x1: If <a href="#">MMCHS_SYSTEST[1]</a> CDIR = 0 (output mode direction), the CMD line is driven high. If <a href="#">MMCHS_SYSTEST[1]</a> CDIR = 1 (input mode direction), no effect.</p> <p>Read 0x1: If <a href="#">MMCHS_SYSTEST[1]</a> CDIR = 1 (input mode direction), returns the value on the CMD line (high) If <a href="#">MMCHS_SYSTEST[1]</a> CDIR = 0 (output mode direction), returns 1</p>	RW	0
1	CDIR	<p>Control of the CMD pin direction.</p> <p>Read 0x0: No action. Returns 0. Write 0x0: The CMD line is an output (host to card)</p> <p>Read 0x1: No action. Returns 1. Write 0x1: The CMD line is an input (card to host)</p>	RW	0
0	MCKD	<p>MMC clock output signal data value</p> <p>Read 0x0: No action. Returns 0. Write 0x0: The output clock is driven low.</p> <p>Write 0x1: The output clock is driven high. Read 0x1: No action. Returns 1.</p>	RW	0

**Table 24-68. Register Call Summary for Register MMCHS\_SYSTEST**

MMC/SD/SDIO Functional Description

- [Test Registers: \[0\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[1\] \[2\]](#)
- [MMC/SD/SDIO Register Description: \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\] \[44\] \[45\] \[46\] \[47\] \[48\] \[49\] \[50\] \[51\] \[52\] \[53\] \[54\] \[55\] \[56\] \[57\] \[58\] \[59\] \[60\] \[61\] \[62\] \[63\] \[64\] \[65\] \[66\] \[67\] \[68\] \[69\] \[70\] \[71\] \[72\] \[73\]](#)

**Table 24-69. MMCHS\_CON**

<b>Address Offset</b>	0x0000 012C	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	<p>Configuration register</p> <p>This register is used:</p> <ul style="list-style-type: none"> <li>- to select the functional mode or the SYSTEST mode for any card.</li> <li>- to send an initialization sequence to any card.</li> <li>- to enable the detection on DAT[1] of a card interrupt for SDIO cards only.</li> </ul> <p>and also to configure :</p> <ul style="list-style-type: none"> <li>- specific data and command transfers for MMC cards only.</li> <li>- the parameters related to the card detect and write protect input signals.</li> </ul>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED								SDMA_LINE	DMA_MNS	DDR	BOOT_CFO	BOOT_ACK	CLKEXTFREE	PADEN	OBIE	OBIP	CEATA	CTPL	DVAL	WPP	CDP	MIT	DW8	MODE	STR	HR	INIT	OD				

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored. Reads return 0.	R	0x000
21	SDMA_LNE	Slave DMA Level/Edge Request: The waveform of the DMA request can be configured either edge sensitive with early de-assertion on first access to <a href="#">MMCHS_DATA</a> register or late de-assertion, request remains active until last allowed data written into <a href="#">MMCHS_DATA</a> .  0x0: Slave DMA edge sensitive, Early DMA de-assertion 0x1: Slave DMA level sensitive, Late DMA de-assertion	RW	0
20	DMA_MNS	DMA Master or Slave selection: When this bit is set and the controller is configured to use the DMA, Interconnect master interface is used to get datas from system using ADMA2 procedure (direct access to the memory). This option is only available if generic parameter <a href="#">MMCHS_HL_HWINFO[0]</a> <a href="#">MADMA_EN</a> is asserted to 1.  0x0: The controller is slave on data transfers with system. 0x1: The controller is master on data exchange with system, controller must be configured as using DMA.	RW	0
19	DDR	Dual Data Rate mode: When this register is set, the controller uses both clock edge to emit or receive data. Odd bytes are transmitted on falling edges and even bytes are transmitted on rise edges. It only applies on Data bytes and CRC, Start, end bits and CRC status are kept full cycle. This bit field is only meaningful and active for even clock divider ratio of <a href="#">MMCHS_SYSCTL[15:6]</a> <a href="#">CLKD</a> , it is insensitive to <a href="#">MMCHS_HCTL[2]</a> <a href="#">HSPE</a> setting.  0x0: Standard mode : data are transmitted on a single edge depending on <a href="#">MMCHS_HCTL[2]</a> <a href="#">HSPE</a> . 0x1: Data Bytes and CRC are transmitted on both edge.	RW	0
18	BOOT_CFO	Boot status supported: This register is set when the CMD line need to be forced to 0 for a boot sequence. CMD line is driven to 0 after writing in <a href="#">MMCHS_CMD</a> . The line is released when this bit field is de-asserted and abort data transfer in case of a pending transaction.  Write 0x0: CMD line is released when it was previously forced to 0 by a boot sequence. Read 0x0: CMD line not forced Read 0x1: CMD line forced to 0 is enabled Write 0x1: CMD line forced to 0 is enabled and will be active after writing into <a href="#">MMCHS_CMD</a>	RW	0
17	BOOT_ACK	Book acknowledge received: When this bit is set the controller should receive a boot status on DAT0 line after next command issued. If no status is received a data timeout will be generated.  0x0: No acknowledge to be received 0x1: A boot status will be received on DAT0 line after issuing a command.	RW	0

Bits	Field Name	Description	Type	Reset
16	CLKEXTFREE	<p>External clock free running: This register is used to maintain card clock out of transfer transaction to enable slave module for example to generate a synchronous interrupt on DAT[1]. The Clock will be maintain only if <a href="#">MMCHS_SYSCTL[2]</a> CEN is set.</p> <p>0x0: External card clock is cut off outside active transaction period.</p> <p>0x1: External card clock is maintain even out of active transaction period only if <a href="#">MMCHS_SYSCTL[2]</a> CEN is set.</p>	RW	0
15	PADEN	<p>Control Power for MMC Lines: This register is only useful when MMC PADS contain power saving mechanism to minimize its leakage power. It works as a GPIO that directly control the ACTIVE pin of PADS. Excepted for DAT[1], the signal is also combine outside the module with the dedicated power control <a href="#">MMCHS_CON[11]</a> CTPL bit.</p> <p>0x0: ADPIDLE module pin is not forced, it is automatically generated by the MMC fsms.</p> <p>0x1: ADPIDLE module pin is forced to active state.</p>	RW	0
14	OBIE	<p>Out-of-Band Interrupt Enable MMC cards only: This bit enables the detection of Out-of-Band Interrupt on MMCOBI input pin. The usage of the Out-of-Band signal (OBI) is optional and depends on the system integration.</p> <p>0x0: Out-of-Band interrupt detection disabled</p> <p>0x1: Out-of-Band interrupt detection enabled</p>	RW	0
13	OBIP	<p>Out-of-Band Interrupt Polarity MMC cards only: This bit selects the active level of the out-of-band interrupt coming from MMC cards. The usage of the Out-of-Band signal (OBI) is optional and depends on the system integration.</p> <p>0x0: active high level</p> <p>0x1: active low level</p>	RW	0
12	CEATA	<p>CE-ATA control mode MMC cards compliant with CE-ATA:By default, this bit is set to 0. It is use to indicate that next commands are considered as specific CE-ATA commands that potentially use 'command completion' features.</p> <p>0x0: Standard MMC/SD/SDIO mode.</p> <p>0x1: CE-ATA mode next commands are considered as CE-ATA commands.</p>	RW	0
11	CTPL	<p>Control Power for DAT[1] line MMC and SD cards: By default, this bit is set to 0 and the host controller automatically disables all the input buffers outside of a transaction to minimize the leakage current. SDIO cards: When this bit is set to 1, the host controller automatically disables all the input buffers except the buffer of DAT[1] outside of a transaction in order to detect asynchronous card interrupt on DAT[1] line and minimize the leakage current of the buffers.</p> <p>0x0: Disable all the input buffers outside of a transaction.</p> <p>0x1: Disable all the input buffers except the buffer of DAT[1] outside of a transaction.</p>	RW	0

Bits	Field Name	Description	Type	Reset
10:9	DVAL	<p>Debounce filter value All cards This register is used to define a debounce period to filter the card detect input signal (SDCD). The usage of the card detect input signal (SDCD) is optional and depends on the system integration and the type of the connector housing that accommodates the card.</p> <p>0x0: 33 us debounce period 0x1: 231 us debounce period 0x2: 1 ms debounce period 0x3: 8,4 ms debounce period</p>	RW	0x3
8	WPP	<p>Write protect polarity For SD and SDIO cards only This bit selects the active level of the write protect input signal (SDWP). The usage of the write protect input signal (SDWP) is optional and depends on the system integration and the type of the connector housing that accommodates the card.</p> <p>0x0: active high level 0x1: active low level</p>	RW	0
7	CDP	<p>Card detect polarity All cards This bit selects the active level of the card detect input signal (SDCD). The usage of the card detect input signal (SDCD) is optional and depends on the system integration and the type of the connector housing that accommodates the card.</p> <p>0x0: active high level 0x1: active low level</p>	RW	0
6	MIT	<p>MMC interrupt command Only for MMC cards. This bit must be set to 1, when the next write access to the command register (<a href="#">MMCHS_CMD</a>) is for writing a MMC interrupt command (CMD40) requiring the command timeout detection to be disabled for the command response.</p> <p>0x0: Command timeout enabled 0x1: Command timeout disabled</p>	RW	0
5	DW8	<p>8-bit mode MMC select For SD/SDIO cards, this bit must be set to 0. For MMC card, this bit must be set following a valid SWITCH command (CMD6) with the correct value and extend CSD index written in the argument. Prior to this command, the MMC card configuration register (CSD and EXT_CSD) must be verified for compliancy with MMC standard specification 4.x (see section 3.6).</p> <p>0x0: 1-bit or 4-bit Data width (DAT[0] used, MMC, SD cards) 0x1: 8-bit Data width (DAT[7:0] used, MMC cards)</p>	RW	0

Bits	Field Name	Description	Type	Reset
4	MODE	<p>Mode select All cards These bits select between Functional mode and SYSTEST mode.</p> <p>0x0: Functional mode. Transfers to the MMC/SD/SDIO cards follow the card protocol. MMC clock is enabled. MMC/SD transfers are operated under the control of the CMD register.</p> <p>0x1: SYSTEST mode The signal pins are configured as general-purpose input/output and the 1024-byte buffer is configured as a stack memory accessible only by the local host or system DMA. The pins retain their default type (input, output or in-out). SYSTEST mode is operated under the control of the SYSTEST register.</p>	RW	0
3	STR	<p>Stream command Only for MMC cards. This bit must be set to 1 only for the stream data transfers (read or write) of the adtc commands. Stream read is a class 1 command (CMD11: READ_DAT_UNTIL_STOP). Stream write is a class 3 command (CMD20: WRITE_DAT_UNTIL_STOP).</p> <p>0x0: Block oriented data transfer 0x1: Stream oriented data transfer</p>	RW	0
2	HR	<p>Broadcast host response Only for MMC cards. This register is used to force the host to generate a 48-bit response for bc command type. It can be used to terminate the interrupt mode by generating a CMD40 response by the core (see section 4.3, "Interrupt Mode", in the MMC [1] specification). In order to have the host response to be generated in open drain mode, the register <a href="#">MMCHS_CON[0]</a> OD must be set to 1. When <a href="#">MMCHS_CON[12]</a> CEATA is set to 1 and <a href="#">MMCHS_ARG</a> set to 0x00000000 when writing 0x00000000 into <a href="#">MMCHS_CMD</a> register, the host controller performs a 'command completion signal disable' token that is, CMD line held to 0 during 47 cycles followed by a 1.</p> <p>0x0: The host does not generate a 48-bit response instead of a command. 0x1: The host generates a 48-bit response instead of a command or a command completion signal disable token.</p>	RW	0
1	INIT	<p>Send initialization stream All cards. When this bit is set to 1, and the card is idle, an initialization sequence is sent to the card. An initialization sequence consists of setting the CMD line to 1 during 80 clock cycles. The initialisation sequence is mandatory - but it is not required to do it through this bit - this bit makes it easier. Clock divider (<a href="#">MMCHS_SYSCTL[15:6]</a> CLKD) should be set to ensure that 80 clock periods are greater than 1ms. (see section 9.3, "Power-Up", in the MMC card specification [1], or section 6.4 in the SD card specification [2]). Note: in this mode, there is no command sent to the card and no response is expected</p> <p>0x0: The host does not send an initialization sequence. 0x1: The host sends an initialization sequence.</p>	RW	0

Bits	Field Name	Description	Type	Reset
0	OD	<p>Card open drain mode. Only for MMC cards. This bit must be set to 1 for MMC card commands 1, 2, 3 and 40, and if the MMC card bus is operating in open-drain mode during the response phase to the command sent. Typically, during card identification mode when the card is either in idle, ready or ident state. It is also necessary to set this bit to 1, for a broadcast host response (see Broadcast host response register <a href="#">MMCHS_CON[2]</a> HR)</p> <p>0x0: No Open Drain 0x1: Open Drain or Broadcast host response</p>	RW	0

**Table 24-70. Register Call Summary for Register MMCHS\_CON**

MMC/SD/SDIO Functional Description

- [Interrupt Requests: \[0\]](#)
- [DMA Modes: \[1\] \[2\] \[3\]](#)
- [SLAVE DMA Operations: \[4\]](#)
- [Generation on Rising Edge of MMC Clock: \[5\]](#)
- [MMC CE-ATA Command Completion Disable Management: \[6\] \[7\] \[8\]](#)
- [Test Registers: \[9\]](#)

MMC/SD/SDIO Programming Guide

- [MMC Host and Bus Configuration: \[10\]](#)
- [Card Detection, Identification, and Selection: \[11\]](#)
- [Read/Write Transfer Flow in CE-ATA Mode: \[12\]](#)
- [Basic Operations - Steps Detailed: \[13\] \[14\] \[15\]](#)
- [Bus Voltage Selection: \[16\]](#)
- [Boot using the CMD0: \[17\] \[18\]](#)
- [Boot with CMD line tied to 0: \[19\] \[20\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[21\] \[22\]](#)
- [MMC/SD/SDIO Register Description: \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\]](#)

**Table 24-71. MMCHS\_PWCNT**

<b>Address Offset</b>	0x0000 0130	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	<p>Power counter register This register is used to program a mmc counter to delay command transfers after activating the PAD power, this value depends on PAD characteristics and voltage.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWCNT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	<p>Reserved These bits are initialized to zero, and writes to them are ignored. Reads return 0.</p>	R	0x0000

Bits	Field Name	Description	Type	Reset
15:0	PWRCNT	Power counter register. This register is used to introduce a delay between the PAD ACTIVE pin assertion and the command issued.  0x0: No additional delay added 0x1: TCF delay (card clock period) 0x2: TCF x 2 delay (card clock period) 0xFFFE: TCF x 65534 delay (card clock period) 0xFFFF: TCF x 65535 delay (card clock period)	RW	0x0000

**Table 24-72. Register Call Summary for Register MMCHS\_PWCNT**

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[0\] \[1\]](#)

**Table 24-73. MMCHS\_SDMA SA**

<b>Address Offset</b>	0x0000 0200		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	sDMA System address: This register is used to program a mmc counter to delay command transfers after activating the PAD power, this value depends on PAD characteristics and voltage.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWRCNT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored. Reads return 0.	R	0x0000
15:0	PWRCNT	Power counter register. This register is used to introduce a delay between the PAD ACTIVE pin assertion and the command issued.  0x0: No additional delay added 0x1: TCF delay (card clock period) 0x2: TCF x 2 delay (card clock period) 0xFFFE: TCF x 65534 delay (card clock period) 0xFFFF: TCF x 65535 delay (card clock period)	RW	0x0000

**Table 24-74. MMCHS\_BLK**

<b>Address Offset</b>	0x0000 0204		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	Transfer Length Configuration register <a href="#">MMCHS_BLK[11:0]</a> BLEN is the block size register. <a href="#">MMCHS_BLK[31:16]</a> NBLK is the block count register. This register shall be used for any card.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NBLK																RESERVED				BLEN											



Bits	Field Name	Description	Type	Reset
31:16	NBLK	<p>Blocks count for current transfer</p> <p>This register is enabled when Block count Enable (<a href="#">MMCHS_CMD[1] BCE</a>) is set to 1 and is valid only for multiple block transfers. Setting the block count to 0 results no data blocks being transferred.</p> <p>Note: The host controller decrements the block count after each block transfer and stops when the count reaches zero.</p> <p>This register can be accessed only if no transaction is executing (i.e, after a transaction has stopped). Read operations during transfers may return an invalid value and write operation will be ignored.</p> <p>In suspend context, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, The local host shall restore the previously saved block count.</p> <p>0x0: Stop count                      0x1: 1 block                      0x2: 2 blocks                      0xFFFF: 65535 blocks</p>	RW	0x0000
15:12	RESERVED	<p>Reserved.</p> <p>Host DMA Buffer Boundary not supported. These bits are initialized to zero, and writes to them are ignored.</p>	R	0x0
11:0	BLEN	<p>Transfer Block Size.</p> <p>This register specifies the block size for block data transfers.</p> <p>Read operations during transfers may return an invalid value, and write operations are ignored.</p> <p>When a CMD12 command is issued to stop the transfer, a read of the BLEN field after transfer completion (<a href="#">MMCHS_STAT[1] TC</a> set to 1) will not return the true byte number of data length while the stop occurs but the value written in this register before transfer is launched.</p> <p>0x0: No data transfer                      0x1: 1 byte block length                      0x2: 2 bytes block length                      0x3: 3 bytes block length                      0x1FF: 511 bytes block length                      0x200: 512 bytes block length                      0x7FF: 2047 bytes block length                      0x800: 2048 bytes block length</p>	RW	0x000

**Table 24-75. Register Call Summary for Register MMCHS\_BLK**

MMC/SD/SDIO Functional Description

- [Interrupt Requests: \[0\] \[1\]](#)
- [Data Length: \[2\]](#)
- [DMA Receive Mode: \[3\]](#)
- [DMA Transmit Mode: \[4\]](#)
- [Data Buffer: \[5\]](#)

MMC/SD/SDIO Programming Guide

- [Suspend-Resume Flow: \[6\] \[7\]](#)
- [Basic Operations - Steps Detailed: \[8\] \[9\]](#)
- [Boot using the CMD0: \[10\]](#)
- [Boot with CMD line tied to 0: \[11\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[12\] \[13\]](#)
- [MMC/SD/SDIO Register Description: \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\]](#)

**Table 24-76. MMCHS\_ARG**

<b>Address Offset</b>	0x0000 0208	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	Command argument Register This register contains command argument specified as bit 39-8 of Command-Format These registers must be initialized prior to sending the command itself to the card (write action into the register <a href="#">MMCHS_CMD</a> register). Only exception is for a command index specifying stuff bits in arguments, making a write unnecessary.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARG																															

Bits	Field Name	Description	Type	Reset
31:0	ARG	Command argument bits [31:0]	RW	0x0000 0000

**Table 24-77. Register Call Summary for Register MMCHS\_ARG**

MMC/SD/SDIO Functional Description

- [MMC CE-ATA Command Completion Disable Management: \[0\]](#)

MMC/SD/SDIO Programming Guide

- [Read/Write Transfer Flow in CE-ATA Mode: \[1\]](#)
- [Basic Operations - Steps Detailed: \[2\] \[3\]](#)
- [Boot using the CMD0: \[4\]](#)
- [Boot with CMD line tied to 0: \[5\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[6\] \[7\]](#)
- [MMC/SD/SDIO Register Description: \[8\]](#)

**Table 24-78. MMCHS\_CMD**

<b>Address Offset</b>	0x0000 020C	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	Command and transfer mode register <a href="#">MMCHS_CMD</a> [31:16] = the command register <a href="#">MMCHS_CMD</a> [15:0] = the transfer mode. This register configures the data and command transfers. A write into the most significant byte send the command. A write into <a href="#">MMCHS_CMD</a> [15:0] registers during data transfer has no effect. This register shall be used for any card. Note: In SYSTEST mode, a write into <a href="#">MMCHS_CMD</a> register will not start a transfer.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	INDX							CMD_TYPE	DP	CICE	CCCE	RESERVED	RSP_TYPE	RESERVED										MSBS	DDIR	RESERVED	ACEN	BCE	DE		

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0
29:24	INDX	Command index Binary encoded value from 0 to 63 specifying the command number send to card 0x0: CMD0 or ACMD0	RW	0x00

Bits	Field Name	Description	Type	Reset
		0x1: CMD1 or ACMD1		
		0x2: CMD2 or ACMD2		
		0x3: CMD3 or ACMD3		
		0x4: CMD4 or ACMD4		
		0x5: CMD5 or ACMD5		
		0x6: CMD6 or ACMD6		
		0x7: CMD7 or ACMD7		
		0x8: CMD8 or ACMD8		
		0x9: CMD9 or ACMD9		
		0xA: CMD10 or ACMD10		
		0xB: CMD11 or ACMD11		
		0xC: CMD12 or ACMD12		
		0xD: CMD13 or ACMD13		
		0xE: CMD14 or ACMD14		
		0xF: CMD15 or ACMD15		
		0x10: CMD16 or ACMD16		
		0x11: CMD17 or ACMD17		
		0x12: CMD18 or ACMD18		
		0x13: CMD19 or ACMD19		
		0x14: CMD20 or ACMD20		
		0x15: CMD21 or ACMD21		
		0x16: CMD22 or ACMD22		
		0x17: CMD23 or ACMD23		
		0x18: CMD24 or ACMD24		
		0x19: CMD25 or ACMD25		
		0x1A: CMD26 or ACMD26		
		0x1B: CMD27 or ACMD27		
		0x1C: CMD28 or ACMD28		
		0x1D: CMD29 or ACMD29		
		0x1E: CMD30 or ACMD30		
		0x1F: CMD31 or ACMD31		
		0x20: CMD32 or ACMD32		
		0x21: CMD33 or ACMD33		
		0x22: CMD34 or ACMD34		
		0x23: CMD35 or ACMD35		
		0x24: CMD36 or ACMD36		
		0x25: CMD37 or ACMD37		
		0x26: CMD38 or ACMD38		
		0x27: CMD39 or ACMD39		
		0x28: CMD40 or ACMD40		
		0x29: CMD41 or ACMD41		
		0x2A: CMD42 or ACMD42		
		0x2B: CMD43 or ACMD43		
		0x2C: CMD44 or ACMD44		
		0x2D: CMD45 or ACMD45		
		0x2E: CMD46 or ACMD46		
		0x2F: CMD47 or ACMD47		
		0x30: CMD48 or ACMD48		
		0x31: CMD49 or ACMD49		

Bits	Field Name	Description	Type	Reset
		0x32: CMD50 or ACMD50 0x33: CMD51 or ACMD51 0x34: CMD52 or ACMD52 0x35: CMD53 or ACMD53 0x36: CMD54 or ACMD54 0x37: CMD55 or ACMD55 0x38: CMD56 or ACMD56 0x39: CMD57 or ACMD57 0x3A: CMD58 or ACMD58 0x3B: CMD59 or ACMD59 0x3C: CMD60 or ACMD60 0x3D: CMD61 or ACMD61 0x3E: CMD62 or ACMD62 0x3F: CMD63 or ACMD63		
23:22	CMD_TYPE	Command type This register specifies three types of special command: Suspend, Resume and Abort. These bits shall be set to 00b for all other commands. 0x0: Others Commands 0x1: CMD52 for writing "Bus Suspend" in CCCR 0x2: CMD52 for writing "Function Select" in CCCR 0x3: Abort command CMD12, CMD52 for writing " I/O Abort" in CCCR	RW	0x0
21	DP	Data present select This register indicates that data is present and DAT line shall be used. It must be set to 0 in the following conditions: - command using only CMD line - command with no data transfer but using busy signal on DAT[0] - Resume command 0x0: Command with no data transfer 0x1: Command with data transfer	RW	0
20	CICE	Command Index check enable This bit must be set to 1 to enable index check on command response to compare the index field in the response against the index of the command. If the index is not the same in the response as in the command, it is reported as a command index error ( <a href="#">MMCHS_STAT[19]</a> CIE set to 1) Note: The register CICE cannot be configured for an Auto CMD12, then index check is automatically checked when this command is issued. 0x0: Index check disable 0x1: Index check enable	RW	0
19	CCCE	Command CRC check enable This bit must be set to 1 to enable CRC7 check on command response to protect the response against transmission errors on the bus. If an error is detected, it is reported as a command CRC error ( <a href="#">MMCHS_STAT[17]</a> CCRC set to 1). Note: The register CCCE cannot be configured for an Auto CMD12, and then CRC check is automatically checked when this command is issued. 0x0: CRC7 check disable 0x1: CRC7 check enable	RW	0
18	RESERVED	Reserved. This bit is initialized to zero, and writes to it are ignored.	R	0

Bits	Field Name	Description	Type	Reset
17:16	RSP_TYPE	Response type This bits defines the response type of the command 0x0: No response 0x1: Response Length 136 bits 0x2: Response Length 48 bits 0x3: Response Length 48 bits with busy after response	RW	0x0
15:6	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x000
5	MSBS	Multi/Single block select This bit must be set to 1 for data transfer in case of multi block command. For any others command this bit shall be set to 0. 0x0: Single block. If this bit is 0, it is not necessary to set the register <a href="#">MMCHS_BLK[31:16]</a> NBLK. 0x1: Multi block. When Block Count is disabled ( <a href="#">MMCHS_CMD[1]</a> BCE is set to 0) in Multiple block transfers ( <a href="#">MMCHS_CMD[5]</a> MSBS is set to 1), the module can perform infinite transfer.	RW	0
4	DDIR	Data transfer Direction Select This bit defines either data transfer will be a read or a write. 0x0: Data Write (host to card) 0x1: Data Read (card to host)	RW	0
3	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0
2	ACEN	Auto CMD12 Enable SD card only. When this bit is set to 1, the host controller issues a CMD12 automatically after the transfer completion of the last block. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop data transfer. In particular, safe commands do not require CMD12. 0x0: Auto CMD12 disable 0x1: Auto CMD12 enable or CCS detection enabled.	RW	0
1	BCE	Block Count Enable Multiple block transfers only. This bit is used to enable the block count register ( <a href="#">MMCHS_BLK[31:16]</a> NBLK). When Block Count is disabled ( <a href="#">MMCHS_CMD[1]</a> BCE is set to 0) in Multiple block transfers ( <a href="#">MMCHS_CMD[5]</a> MSBS is set to 1), the module can perform infinite transfer. 0x0: Block count disabled for infinite transfer. 0x1: Block count enabled for multiple block transfer with known number of blocks	RW	0
0	DE	DMA Enable This bit is used to enable DMA mode for host data access. 0x0: DMA mode disable 0x1: DMA mode enable	RW	0

**Table 24-79. Register Call Summary for Register MMCHS\_CMD**

MMC/SD/SDIO Environment
<ul style="list-style-type: none"> <li>Data Format: [0] [1] [2] [3]</li> </ul>
MMC/SD/SDIO Functional Description
<ul style="list-style-type: none"> <li>Interrupt Requests: [4] [5]</li> <li>Data Length: [6]</li> <li>SLAVE DMA Operations: [7]</li> <li>Data Buffer: [8]</li> <li>Memory size, Block Length and Buffer Management Relationship: [9] [10]</li> <li>Transfer Stop: [11]</li> <li>MMC CE-ATA Command Completion Disable Management: [12] [13]</li> </ul>
MMC/SD/SDIO Programming Guide
<ul style="list-style-type: none"> <li>Card Detection, Identification, and Selection: [14]</li> <li>Read/Write Transfer Flow in DMA Mode with Interrupt: [15]</li> <li>Read/Write Transfer Flow in DMA Mode with Polling: [16]</li> <li>Read/Write Transfer Flow without DMA with Polling: [17]</li> <li>Read/Write Transfer Flow in CE-ATA Mode: [18]</li> <li>Basic Operations - Steps Detailed: [19] [20]</li> <li>Bus Voltage Selection: [21]</li> <li>Boot using the CMD0: [22]</li> <li>Boot with CMD line tied to 0: [23]</li> </ul>
MMC/SD/SDIO Register Manual
<ul style="list-style-type: none"> <li>MMC/SD/SDIO Register Summary: [24] [25]</li> <li>MMC/SD/SDIO Register Description: [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47]</li> </ul>

**Table 24-80. MMCHS\_RSP10**

<b>Address Offset</b>	0x0000 0210	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See Table 24-51 to Table 24-52		
<b>Description</b>	Command response[31:0] Register This 32-bit register holds bits positions [31:0] of command response type R1/R1b/R2/R3/R4/R5/R5b/R6		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP1								RSP0																							

Bits	Field Name	Description	Type	Reset
31:16	RSP1	Command Response [31:16]	R	0x0000
15:0	RSP0	Command Response [15:0]	R	0x0000

**Table 24-81. Register Call Summary for Register MMCHS\_RSP10**

MMC/SD/SDIO Functional Description
<ul style="list-style-type: none"> <li>Different Types of Responses: [0] [1] [2]</li> </ul>
MMC/SD/SDIO Programming Guide
<ul style="list-style-type: none"> <li>Card Detection, Identification, and Selection: [3]</li> <li>Basic Operations - Steps Detailed: [4] [5] [6]</li> </ul>
MMC/SD/SDIO Register Manual
<ul style="list-style-type: none"> <li>MMC/SD/SDIO Register Summary: [7] [8]</li> <li>MMC/SD/SDIO Register Description: [9] [10]</li> </ul>

**Table 24-82. MMCHS\_RSP32**

<b>Address Offset</b>	0x0000 0214	
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b> MMCHS
<b>Description</b>	Command response[63:32] Register This 32-bit register holds bits positions [63:32] of command response type R2	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP3																RSP2															

Bits	Field Name	Description	Type	Reset
31:16	RSP3	Command Response [63:48]	R	0x0000
15:0	RSP2	Command Response [47:32]	R	0x0000

**Table 24-83. Register Call Summary for Register MMCHS\_RSP32**

MMC/SD/SDIO Functional Description
<ul style="list-style-type: none"> <li><a href="#">Different Types of Responses: [0]</a></li> </ul>
MMC/SD/SDIO Programming Guide
<ul style="list-style-type: none"> <li><a href="#">Basic Operations - Steps Detailed: [1] [2] [3]</a></li> </ul>
MMC/SD/SDIO Register Manual
<ul style="list-style-type: none"> <li><a href="#">MMC/SD/SDIO Register Summary: [4] [5]</a></li> </ul>

**Table 24-84. MMCHS\_RSP54**

<b>Address Offset</b>	0x0000 0218	
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b> MMCHS
<b>Description</b>	Command response[95:64] Register This 32-bit register holds bits positions [95:64] of command response type R2	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP5																RSP4															

Bits	Field Name	Description	Type	Reset
31:16	RSP5	Command Response [95:80]	R	0x0000
15:0	RSP4	Command Response [79:64]	R	0x0000

**Table 24-85. Register Call Summary for Register MMCHS\_RSP54**

MMC/SD/SDIO Functional Description
<ul style="list-style-type: none"> <li><a href="#">Different Types of Responses: [0]</a></li> </ul>
MMC/SD/SDIO Programming Guide
<ul style="list-style-type: none"> <li><a href="#">Basic Operations - Steps Detailed: [1] [2] [3]</a></li> </ul>
MMC/SD/SDIO Register Manual
<ul style="list-style-type: none"> <li><a href="#">MMC/SD/SDIO Register Summary: [4] [5]</a></li> </ul>

**Table 24-86. MMCHS\_RSP76**

<b>Address Offset</b>	0x0000 021C	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	Command response[127:96] Register This 32-bit register holds bits positions [127:96] of command response type R2		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP7																RSP6															

Bits	Field Name	Description	Type	Reset
31:16	RSP7	Command Response [127:112]	R	0x0000
15:0	RSP6	Command Response [111:96]	R	0x0000

**Table 24-87. Register Call Summary for Register MMCHS\_RSP76**

MMC/SD/SDIO Functional Description

- [Different Types of Responses: \[0\] \[1\] \[2\]](#)

MMC/SD/SDIO Programming Guide

- [Basic Operations - Steps Detailed: \[3\] \[4\] \[5\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[6\] \[7\]](#)
- [MMC/SD/SDIO Register Description: \[8\] \[9\]](#)

**Table 24-88. MMCHS\_DATA**

<b>Address Offset</b>	0x0000 0220	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	<p>Data Register</p> <p>This register is the 32-bit entry point of the buffer for read or write data transfers. The buffer size is 32bits x256(1024 bytes). Bytes within a word are stored and read in little endian format. This buffer can be used as two 512 byte buffers to transfer data efficiently without reducing the throughput.</p> <p>Sequential and contiguous access is necessary to increment the pointer correctly. Random or skipped access is not allowed. In little endian, if the local host accesses this register byte-wise or 16bit-wise, the least significant byte (bits [7:0]) must always be written/read first. The update of the buffer address is done on the most significant byte write for full 32-bit DATA register or on the most significant byte of the last word of block transfer.</p> <p>Example 1: Byte or 16-bit access  Mbyteen[3:0]=0001 (1-byte) =&gt; Mbyteen[3:0]=0010 (1-byte) =&gt; Mbyteen[3:0]=1100 (2-bytes) OK  Mbyteen[3:0]=0001 (1-byte) =&gt; Mbyteen[3:0]=0010 (1-byte) =&gt; Mbyteen[3:0]=0100 (1-byte) OK  Mbyteen[3:0]=0001 (1-byte) =&gt; Mbyteen[3:0]=0010 (1-byte) =&gt; Mbyteen[3:0]=1000 (1-byte) Bad</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															



Bits	Field Name	Description	Type	Reset
31:0	DATA	Data Register [31:0] In functional mode ( <a href="#">MMCHS_CON[4] MODE</a> set to the default value 0) , A read access to this register is allowed only when the buffer read enable status is set to 1 ( <a href="#">MMCHS_PSTATE[11] BRE</a> ), otherwise a bad access ( <a href="#">MMCHS_STAT[29] BADA</a> ) is signaled. A write access to this register is allowed only when the buffer write enable status is set to 1( <a href="#">MMCHS_PSTATE[10] BWE</a> ), otherwise a bad access ( <a href="#">MMCHS_STAT[29] BADA</a> ) is signaled and the data is not written.	RW	0x0000 0000

**Table 24-89. Register Call Summary for Register MMCHS\_DATA**

MMC/SD/SDIO Functional Description

- [Interrupt Requests: \[0\] \[1\]](#)
- [Data Buffer: \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [Transfer or Command Status and Errors Reporting: \[10\]](#)

MMC/SD/SDIO Programming Guide

- [Read/Write Transfer Flow without DMA with Polling: \[11\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[12\] \[13\]](#)
- [MMC/SD/SDIO Register Description: \[14\] \[15\] \[16\] \[17\] \[18\] \[19\]](#)

**Table 24-90. MMCHS\_PSTATE**

<b>Address Offset</b>	0x0000 0224	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	Present state register The Host can get status of the Host Controller from this 32-bit read only register.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CLEV	DLEV			WP	CDPL	CSS	CINS	RESERVED					BRE	BWE	RTA	WTA	RESERVED				DLA	DATI	CMDI

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x00
24	CLEV	CMD line signal level This status is used to check the CMD line level to recover from errors, and for debugging. The value of this register after reset depends on the CMD line level at that time. Read 0x0: The CMD line level is 0. Read 0x1: The CMD line level is 1.	R	-
23:20	DLEV	DAT[3:0] line signal level DAT[3] => bit 23 DAT[2] => bit 22 DAT[1] => bit 21 DAT[0] => bit 20 This status is used to check DAT line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DAT[0]. The value of these registers after reset depends on the DAT lines level at that time.	R	0x-

Bits	Field Name	Description	Type	Reset
19	WP	<p>Write protect switch pin level For SDIO cards only. This bit reflects the write protect input pin (SDWP) level. The value of this register after reset depends on the protect input pin (SDWP) level at that time.</p> <p>Read 0x0: If <a href="#">MMCHS_CON[8]</a> WPP is set to 0 (default), the card is write protected, otherwise the card is not protected.</p> <p>Read 0x1: If <a href="#">MMCHS_CON[8]</a> WPP is set to 0 (default), the card is not write protected, otherwise the card is protected.</p>	R	-
18	CDPL	<p>Card detect pin level This bit reflects the inverse value of the card detect input pin (SDCD), debouncing is not performed on this bit and bit is valid only when Card State Stable (<a href="#">MMCHS_PSTATE[17]</a> CSS) is set to 1. Use of this bit is limited to testing since it must be debounced by software. The value of this register after reset depends on the card detect input pin (SDCD) level at that time.</p> <p>Read 0x0: The value of the card detect input pin (SDCD) is 1</p> <p>Read 0x1: The value of the card detect input pin (SDCD) is 0</p>	R	-
17	CSS	<p>Card State Stable This bit is used for testing. It is set to 1 only when Card Detect Pin Level is stable (<a href="#">MMCHS_PSTATE[18]</a> CDPL). Debouncing is performed on the card detect input pin (SDCD) to detect card stability. This bit is not affected by a software reset.</p> <p>Read 0x0: Reset or Debouncing</p> <p>Read 0x1: No card or card inserted</p>	R	0
16	CINS	<p>Card inserted This bit is the debounced value of the card detect input pin (SDCD). An inactive to active transition of the card detect input pin (SDCD) will generate a card insertion interrupt (<a href="#">MMCHS_STAT[6]</a> CINS). A active to inactive transition of the card detect input pin (SDCD) will generate a card removal interrupt (<a href="#">MMCHS_STAT[7]</a> CREM). This bit is not affected by a software reset.</p> <p>Read 0x0: If <a href="#">MMCHS_CON[7]</a> CDP is set to 0 (default), no card is detected. The card may have been removed from the card slot. If <a href="#">MMCHS_CON[7]</a> CDP is set to 1, the card has been inserted.</p> <p>Read 0x1: If <a href="#">MMCHS_CON[7]</a> CDP is set to 0 (default), the card has been inserted from the card slot. If <a href="#">MMCHS_CON[7]</a> CDP is set to 1, no card is detected. The card may have been removed from the card slot.</p>	R	0
15:12	RESERVED	<p>Reserved. These bits are initialized to zero, and writes to them are ignored.</p>	R	0x0

Bits	Field Name	Description	Type	Reset
11	BRE	<p>Buffer read enable This bit is used for non-DMA read transfers. It indicates that a complete block specified by <a href="#">MMCHS_BLK[11:0]</a> BLEN has been written in the buffer and is ready to be read. It is set to 0 when the entire block is read from the buffer. It is set to 1 when a block data is ready in the buffer and generates the Buffer read ready status of interrupt (<a href="#">MMCHS_STAT[5]</a> BRR).</p> <p>Read 0x0: Read BLEN bytes disable Read 0x1: Read BLEN bytes enable. Readable data exists in the buffer.</p>	R	0
10	BWE	<p>Buffer Write enable This status is used for non-DMA write transfers. It indicates if space is available for write data.</p> <p>Read 0x0: There is no room left in the buffer to write BLEN bytes of data. Read 0x1: There is enough space in the buffer to write BLEN bytes of data.</p>	R	0
9	RTA	<p>Read transfer active This status is used for detecting completion of a read transfer. It is set to 1 after the end bit of read command or by activating a continue request (<a href="#">MMCHS_HCTL[17]</a> CR) following a stop at block gap request. This bit is set to 0 when all data have been read by the local host after last block or after a stop at block gap request.</p> <p>Read 0x0: No valid data on the DAT lines. Read 0x1: read data transfer on going.</p>	R	0
8	WTA	<p>Write transfer active This status indicates a write transfer active. It is set to 1 after the end bit of write command or by activating a continue request (<a href="#">MMCHS_HCTL[17]</a> CR) following a stop at block gap request. This bit is set to 0 when CRC status has been received after last block or after a stop at block gap request.</p> <p>Read 0x0: No valid data on the DAT lines. Read 0x1: Write data transfer on going.</p>	R	0
7:3	RESERVED	<p>Reserved. These bits are initialized to zero, and writes to them are ignored.</p>	R	0x00
2	DLA	<p>DAT line active This status bit indicates whether one of the DAT line is in use. In the case of read transactions (card to host): This bit is set to 1 after the end bit of read command or by activating continue request <a href="#">MMCHS_HCTL[17]</a> CR. This bit is set to 0 when the host controller received the end bit of the last data block or at the beginning of the read wait mode. In the case of write transactions (host to card): This bit is set to 1 after the end bit of write command or by activating continue request <a href="#">MMCHS_HCTL[17]</a> CR. This bit is set to 0 on the end of busy event for the last block; host controller must wait 8 clock cycles with line not busy to really consider not "busy state" or after the busy block as a result of a stop at gap request.</p> <p>Read 0x0: DAT Line inactive Read 0x1: DAT Line active</p>	R	0

Bits	Field Name	Description	Type	Reset
1	DATI	<p>Command inhibit(DAT)</p> <p>This status bit is generated if either DAT line is active (<a href="#">MMCHS_PSTATE[2]</a> DLA) or Read transfer is active (<a href="#">MMCHS_PSTATE[9]</a> RTA) or when a command with busy is issued. This bit prevents the local host to issue a command.</p> <p>A change of this bit from 1 to 0 generates a transfer complete interrupt (<a href="#">MMCHS_STAT[1]</a> TC).</p> <p>Read 0x0: Issuing of command using the DAT lines is allowed</p> <p>Read 0x1: Issuing of command using DAT lines is not allowed</p>	R	0
0	CMDI	<p>Command inhibit(CMD)</p> <p>This status bit indicates that the CMD line is in use. This bit is set to 0 when the most significant byte is written into the command register. This bit is not set when Auto CMD12 is transmitted.</p> <p>This bit is set to 0 in either the following cases:</p> <ul style="list-style-type: none"> <li>- After the end bit of the command response, excepted if there is a command conflict error (<a href="#">MMCHS_STAT[17]</a> CCRC or <a href="#">MMCHS_STAT[18]</a> CEB set to 1) or a Auto CMD12 is not executed (<a href="#">MMCHS_AC12[0]</a> ACNE).</li> <li>- After the end bit of the command without response (<a href="#">MMCHS_CMD[17:16]</a> RSP_TYPE set to "00")</li> </ul> <p>In case of a command data error is detected (<a href="#">MMCHS_STAT[16]</a> CTO set to 1), this register is not automatically cleared.</p> <p>Read 0x0: Issuing of command using CMD line is allowed</p> <p>Read 0x1: Issuing of command using CMD line is not allowed</p>	R	0

**Table 24-91. Register Call Summary for Register MMCHS\_PSTATE**

MMC/SD/SDIO Functional Description

- [Software Reset: \[0\]](#)
- [Interrupt Requests: \[1\] \[2\] \[3\] \[4\]](#)
- [Data Length: \[5\] \[6\] \[7\]](#)
- [Data Buffer: \[8\] \[9\]](#)
- [Data Buffer Status: \[10\] \[11\]](#)
- [MMC/SD/SDIO Hardware Status Features: \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\]](#)

MMC/SD/SDIO Programming Guide

- [Read/Write Transfer Flow in DMA Mode with Interrupt: \[21\]](#)
- [Read/Write Transfer Flow in DMA Mode with Polling: \[22\]](#)
- [Read/Write Transfer Flow without DMA with Polling: \[23\]](#)
- [Read/Write Transfer Flow in CE-ATA Mode: \[24\]](#)
- [Basic Operations - Steps Detailed: \[25\] \[26\]](#)
- [Bus Voltage Selection: \[27\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[28\] \[29\]](#)
- [MMC/SD/SDIO Register Description: \[30\] \[31\] \[32\] \[33\] \[34\] \[35\] \[36\] \[37\] \[38\] \[39\] \[40\] \[41\] \[42\] \[43\] \[44\] \[45\] \[46\] \[47\] \[48\] \[49\] \[50\] \[51\] \[52\] \[53\] \[54\]](#)

Table 24-92. MMCHS\_HCTL

<b>Address Offset</b>	0x0000 0228	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See Table 24-51 to Table 24-52		
<b>Description</b>	Control register This register defines the host controls to set power, wakeup and transfer parameters. <a href="#">MMCHS_HCTL[31:24]</a> = Wakeup control <a href="#">MMCHS_HCTL[23:16]</a> = Block gap control <a href="#">MMCHS_HCTL[15:8]</a> = Power control <a href="#">MMCHS_HCTL[7:0]</a> = Host control		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				OBWE	REM	INS	IWE	RESERVED				IBG	RWC	CR	SBGR	RESERVED				SDVS	SDBP	CDSS	CDTL	RESERVED	DMAS	HSPE	DTW	LED			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0
27	OBWE	Wakeup event enable for 'Out-of-Band' Interrupt. This bit enables wakeup events for 'Out-of-Band' assertion. Wakeup is generated if the wakeup feature is enabled ( <a href="#">MMCHS_SYSCONFIG[2]</a> ENAWAKEUP). The write to this register is ignored when <a href="#">MMCHS_CON[14]</a> OBIE is not set. 0x0: Disable wakeup on 'Out-of-Band' Interrupt 0x1: Enable wakeup on 'Out-of-Band' Interrupt	RW	0
26	REM	Wakeup event enable on SD card removal This bit enables wakeup events for card removal assertion. Wakeup is generated if the wakeup feature is enabled ( <a href="#">MMCHS_SYSCONFIG[2]</a> ENAWAKEUP). 0x0: Disable wakeup on card removal 0x1: Enable wakeup on card removal	RW	0
25	INS	Wakeup event enable on SD card insertion This bit enables wakeup events for card insertion assertion. Wakeup is generated if the wakeup feature is enabled ( <a href="#">MMCHS_SYSCONFIG[2]</a> ENAWAKEUP). 0x0: Disable wakeup on card insertion 0x1: Enable wakeup on card insertion	RW	0
24	IWE	Wakeup event enable on SD card interrupt This bit enables wakeup events for card interrupt assertion. Wakeup is generated if the wakeup feature is enabled ( <a href="#">MMCHS_SYSCONFIG[2]</a> ENAWAKEUP). 0x0: Disable wakeup on card interrupt 0x1: Enable wakeup on card interrupt	RW	0
23:20	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0

Bits	Field Name	Description	Type	Reset
19	IBG	<p>Interrupt block at gap</p> <p>This bit is valid only in 4-bit mode of SDIO card to enable interrupt detection in the interrupt cycle at block gap for a multiple block transfer. For MMC cards and for SD card this bit should be set to 0.</p> <p>0x0: Disable interrupt detection at the block gap in 4-bit mode</p> <p>0x1: Enable interrupt detection at the block gap in 4-bit mode</p>	RW	0
18	RWC	<p>Read wait control</p> <p>The read wait function is optional only for SDIO cards. If the card supports read wait, this bit must be enabled, then requesting a stop at block gap (<a href="#">MMCHS_HCTL[16]</a> SBGR) generates a read wait period after the current end of block. Be careful, if read wait is not supported it may cause a conflict on DAT line.</p> <p>0x0: Disable Read Wait Control. Suspend/Resume cannot be supported.</p> <p>0x1: Enable Read Wait Control</p>	RW	0
17	CR	<p>Continue request</p> <p>This bit is used to restart a transaction that was stopped by requesting a stop at block gap (<a href="#">MMCHS_HCTL[16]</a> SBGR). Set this bit to 1 restarts the transfer. The bit is automatically set to 0 by the host controller when transfer has restarted i.e DAT line is active (<a href="#">MMCHS_PSTATE[2]</a> DLA) or transferring data (<a href="#">MMCHS_PSTATE[8]</a> WTA). The Stop at block gap request must be disabled (<a href="#">MMCHS_HCTL[16]</a> SBGR = 0) before setting this bit.</p> <p>0x0: No affect</p> <p>0x1: transfer restart</p>	RW	0
16	SBGR	<p>Stop at block gap request</p> <p>This bit is used to stop executing a transaction at the next block gap. The transfer can restart with a continue request (<a href="#">MMCHS_HCTL[17]</a> CR) or during a suspend/resume sequence.</p> <p>In case of read transfer, the card must support read wait control.</p> <p>In case of write transfer, the host driver shall set this bit after all block data written.</p> <p>Until the transfer completion (<a href="#">MMCHS_STAT[1]</a> TC set to 1), the host driver shall leave this bit set to 1.</p> <p>If this bit is set, the local host shall not write to the data register (<a href="#">MMCHS_DATA</a>).</p> <p>0x0: Transfer mode</p> <p>0x1: Stop at block gap</p>	RW	0
15:12	RESERVED	<p>Reserved.</p> <p>These bits are initialized to zero, and writes to them are ignored.</p>	R	0x0
11:9	SDVS	<p>SD bus voltage select</p> <p>All cards.</p> <p>The host driver should set to these bits to select the voltage level for the card according to the voltage supported by the system (<a href="#">MMCHS_CAPA[26:24]</a>) before starting a transfer.</p> <p>0x5: 1.8V (Typical)</p> <p>0x6: 3.0V (Typical)</p> <p>0x7: 3.3V (Typical)</p>	RW	0x0

Bits	Field Name	Description	Type	Reset
8	SDBP	<p>SD bus power</p> <p>Before setting this bit, the host driver shall select the SD bus voltage (<a href="#">MMCHS_HCTL[11:9] SDVS</a>). If the host controller detects the No card state, this bit is automatically set to 0. If the module is power off, a write in the command register (<a href="#">MMCHS_CMD</a>) will not start the transfer. A write to this bit has no effect if the selected SD bus voltage is not supported according to capability register (<a href="#">MMCHS_CAPA[26:24]</a>).</p> <p>0x0: Power off</p> <p>0x1: Power on</p>	RW	0
7	CDSS	<p>Card Detect Signal Selection</p> <p>This bit selects source for the card detection. When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupt being caused by the glitch. The Interrupt Status/Signal Enable should be disabled during over the period of debouncing.</p> <p>0x0: SDCD# is selected (for normal use)</p> <p>0x1: The Card Detect Test Level is selected (for test purpose)</p>	RW	0
6	CDTL	<p>Card Detect Test Level:</p> <p>This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not.</p> <p>0x0: No Card</p> <p>0x1: Card Inserted</p>	RW	0
5	RESERVED	<p>Reserved.</p> <p>These bits are initialized to zero, and writes to them are ignored.</p>	R	0
4:3	DMAS	<p>DMA Select Mode:</p> <p>One of supported DMA modes can be selected. The host driver shall check support of DMA modes by referring the Capabilities register. Use of selected DMA is determined by DMA Enable of the Transfer Mode register. This register is only meaningful when <a href="#">MMCHS_HL_HWINFO[0] MADMA_EN</a> is set to 1. When <a href="#">MMCHS_HL_HWINFO[0] MADMA_EN</a> is set to 0 the bit field is read only and returned value is 0.</p> <p>0x0: Reserved</p> <p>0x1: Reserved</p> <p>0x2: 32-bit Address ADMA2 is selected</p> <p>0x3: Reserved</p>	RW	0x0
2	HSPE	<p>High Speed Enable:</p> <p>Before setting this bit, the Host Driver shall check the High Speed Support in the Capabilities register. If this bit is set to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock. If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock. This bit shall not be set when dual data rate mode is activated in <a href="#">MMCHS_CON[19] DDR</a>.</p> <p>0x0: Normal speed mode</p> <p>0x1: High speed mode</p>	RW	0

Bits	Field Name	Description	Type	Reset
1	DTW	<p>Data transfer width</p> <p>For MMC card, this bit must be set following a valid SWITCH command (CMD6) with the correct value and extend CSD index written in the argument. Prior to this command, the MMC card configuration register (CSD and EXT_CSD) must be verified for compliance with MMC standard specification 4.x (see section 3.6).</p> <p>This register has no effect when the MMC 8-bit mode is selected (register <a href="#">MMCHS_CON[5]</a> DW8 set to 1 ),</p> <p>For SD/SDIO cards, this bit must be set following a valid SET_BUS_WIDTH command (ACMD6) with the value written in bit 1 of the argument. Prior to this command, the SD card configuration register (SCR) must be verified for the supported bus width by the SD card.</p> <p>0x0: 1-bit Data width (DAT[0] used)</p> <p>0x1: 4-bit Data width (DAT[3:0] used)</p>	RW	0
0	LED	<p>Reserved bit.</p> <p>LED control feature is not supported</p> <p>This bit is initialized to zero, and writes to it are ignored.</p>	R	0

**Table 24-93. Register Call Summary for Register MMCHS\_HCTL**

## MMC/SD/SDIO Functional Description

- [Power Management: \[0\] \[1\] \[2\]](#)
- [Interrupt Requests: \[3\] \[4\]](#)
- [Supported Features: \[5\]](#)
- [Transfer Stop: \[6\] \[7\]](#)
- [Output Signals Generation: \[8\]](#)
- [Generation on Falling Edge of MMC Clock: \[9\]](#)
- [Generation on Rising Edge of MMC Clock: \[10\]](#)

## MMC/SD/SDIO Programming Guide

- [Wakeup Configuration: \[11\]](#)
- [MMC Host and Bus Configuration: \[12\]](#)
- [Suspend-Resume Flow: \[13\]](#)
- [Basic Operations - Steps Detailed: \[14\]](#)

## MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[15\] \[16\]](#)
- [MMC/SD/SDIO Register Description: \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\] \[34\]](#)

**Table 24-94. MMCHS\_SYSCTL**

<b>Address Offset</b>	0x0000 022C																														
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>										<b>Instance</b>	MMCHS																			
<b>Description</b>	<p>SD system control register</p> <p>This register defines the system controls to set software resets, clock frequency management and data timeout.</p> <p><a href="#">MMCHS_SYSCTL[31:24]</a> = Software resets</p> <p><a href="#">MMCHS_SYSCTL[23:16]</a> = Timeout control</p> <p><a href="#">MMCHS_SYSCTL[15:0]</a> = Clock control</p>																														
<b>Type</b>	RW																														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SRD	SRC	SRA	RESERVED				DTO				CLKD				RESERVED				CEN	ICS	ICE						



Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x00
26	SRD	Software reset for DAT line This bit is set to 1 for reset and released to 0 when completed. DAT finite state machine in both clock domain are also reset. Here below are the registers cleared by <a href="#">MMCHS_SYSCTL[26]</a> SRD: - <a href="#">MMCHS_DATA</a> - <a href="#">MMCHS_PSTATE[11]</a> BRE, <a href="#">MMCHS_PSTATE[10]</a> BWE, <a href="#">MMCHS_PSTATE[9]</a> RTA, <a href="#">MMCHS_PSTATE[8]</a> WTA, <a href="#">MMCHS_PSTATE[2]</a> DLA, <a href="#">MMCHS_PSTATE[1]</a> DAT1 - <a href="#">MMCHS_HCTL</a> : SBGR and CR - <a href="#">MMCHS_STAT</a> : <a href="#">MMCHS_STAT[5]</a> BRR, <a href="#">MMCHS_STAT[4]</a> BWR, <a href="#">MMCHS_STAT[2]</a> BGE and <a href="#">MMCHS_STAT[1]</a> TC Interconnect and MMC buffer data management is reinitialized.  0x0: Reset completed 0x1: Software reset for DAT line	RW	0
25	SRC	Software reset for CMD line This bit is set to 1 for reset and released to 0 when completed. CMD finite state machine in both clock domain are also reset. Here below are the registers cleared by <a href="#">MMCHS_SYSCTL[25]</a> SRC: - <a href="#">MMCHS_PSTATE[0]</a> CMDI - <a href="#">MMCHS_STAT[0]</a> CC Interconnect and MMC command status management is reinitialized.  0x0: Reset completed 0x1: Software reset for CMD line	RW	0
24	SRA	Software reset for all This bit is set to 1 for reset , and released to 0 when completed. This reset affects the entire host controller except for the card detection circuit and capabilities registers.  0x0: Reset completed 0x1: Software reset for all the design	RW	0
23:20	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0

Bits	Field Name	Description	Type	Reset
19:16	DTO	<p>Data timeout counter value and busy timeout. This value determines the interval by which DAT lines timeouts are detected.</p> <p>The host driver needs to set this bit field based on</p> <ul style="list-style-type: none"> <li>- the maximum read access time (NAC) (Refer to the SD Specification Part1 Physical Layer),</li> <li>- the data read access time values (TAAC and NSAC) in the card specific data register (CSD) of the card,</li> <li>- the timeout clock base frequency (<a href="#">MMCHS_CAPA[5:0]</a> TCF).</li> </ul> <p>If the card does not respond within the specified number of cycles, a data timeout error occurs (<a href="#">MMCHS_STAT[20]</a> DTO).</p> <p>The <a href="#">MMCHS_SYSCTL[19:16]</a> DTO register is also used to check busy duration, to generate busy timeout for commands with busy response or for busy programming during a write command. Timeout on CRC status is generated if no CRC token is present after a block write.</p> <p>0x0: <math>TCF \times 2^{13}</math>            0x1: <math>TCF \times 2^{14}</math>            0xE: <math>TCF \times 2^{27}</math>            0xF: Reserved</p>	RW	0x0
15:6	CLKD	<p>Clock frequency select</p> <p>These bits define the ratio between a reference clock frequency (system dependant) and the output clock frequency on the CLK pin of either the memory card (MMC, SD or SDIO).</p> <p>0x0: Clock Ref bypass            0x1: Clock Ref bypass            0x2: Clock Ref / 2            0x3: Clock Ref / 3            0x3FF: Clock Ref / 1023</p>	RW	0x000
5:3	RESERVED	<p>Reserved.</p> <p>These bits are initialized to zero, and writes to them are ignored.</p>	R	0x0
2	CEN	<p>Clock enable</p> <p>This bit controls if the clock is provided to the card or not.</p> <p>0x0: The clock is not provided to the card . Clock frequency can be changed .</p> <p>0x1: The clock is provided to the card and can be automatically gated when <a href="#">MMCHS_SYSCONFIG[0]</a> AUTOIDLE is set to 1 (default value) .</p> <p>The host driver shall wait to set this bit to 1 until the Internal clock is stable (<a href="#">MMCHS_SYSCTL[1]</a> ICS).</p>	RW	0
1	ICS	<p>Internal clock stable (status)</p> <p>This bit indicates either the internal clock is stable or not.</p> <p>Read 0x0: The internal clock is not stable.</p> <p>Read 0x1: The internal clock is stable after enabling the clock (<a href="#">MMCHS_SYSCTL[1]</a> ICE) or after changing the clock ratio (<a href="#">MMCHS_SYSCTL[15:6]</a> CLKD).</p>	R	0
0	ICE	<p>Internal clock enable</p> <p>This register controls the internal clock activity. In very low power state, the internal clock is stopped.</p> <p>Note: The activity of the debounce clock (used for wakeup events) and the interface clock (used for reads and writes to the module register map) are not affected by this register.</p> <p>0x0: The internal clock is stopped (very low power state).</p> <p>0x1: The internal clock oscillates and can be automatically gated when <a href="#">MMCHS_SYSCONFIG[0]</a> AUTOIDLE is set to 1 (default value) .</p>	RW	0

**Table 24-95. Register Call Summary for Register MMCHS\_SYSCTL**

MMC/SD/SDIO Environment
<ul style="list-style-type: none"> <li>Protocol: [0]</li> </ul>
MMC/SD/SDIO Functional Description
<ul style="list-style-type: none"> <li>Software Reset: [1] [2] [3] [4] [5]</li> <li>Retention Mode: [6]</li> </ul>
MMC/SD/SDIO Programming Guide
<ul style="list-style-type: none"> <li>MMC Host and Bus Configuration: [7]</li> <li>Card Detection, Identification, and Selection: [8]</li> <li>Read/Write Transfer Flow in DMA Mode with Interrupt: [9]</li> <li>Read/Write Transfer Flow in DMA Mode with Polling: [10]</li> <li>Read/Write Transfer Flow without DMA with Polling: [11]</li> <li>Read/Write Transfer Flow in CE-ATA Mode: [12]</li> <li>Suspend-Resume Flow: [13]</li> <li>Basic Operations - Steps Detailed: [14] [15] [16]</li> <li>Boot using the CMD0: [17]</li> <li>Boot with CMD line tied to 0: [18]</li> </ul>
MMC/SD/SDIO Register Manual
<ul style="list-style-type: none"> <li>MMC/SD/SDIO Register Summary: [19] [20]</li> <li>MMC/SD/SDIO Register Description: [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34]</li> </ul>

**Table 24-96. MMCHS\_STAT**

<b>Address Offset</b>	0x0000 0230	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See Table 24-51 to Table 24-52		
<b>Description</b>	Interrupt status register The interrupt status regroups all the status of the module internal events that can generate an interrupt. <a href="#">MMCHS_STAT[31:16]</a> = Error Interrupt Status <a href="#">MMCHS_STAT[15:0]</a> = Normal Interrupt Status		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	BADA	CERR	RESERVED	ADMAE	ACE	CLE	DEB	DCRC	DTO	CIE	CEB	CCRC	CTO	ERRI	RESERVED	BSR	OBI	CIRQ	CREM	CINS	BRR	BWR	DMA	BGE	TC	CC					

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0
29	BADA	Bad access to data space This bit is set automatically to indicate a bad access to buffer when not allowed: -This bit is set during a read access to the data register ( <a href="#">MMCHS_DATA</a> ) while buffer reads are not allowed ( <a href="#">MMCHS_PSTATE[11]</a> BRE = 0) -This bit is set during a write access to the data register ( <a href="#">MMCHS_DATA</a> ) while buffer writes are not allowed ( <a href="#">MMCHS_PSTATE[10]</a> BWE = 0)  Write 0x0: Status bit unchanged Read 0x0: No Interrupt. Read 0x1: Bad Access Write 0x1: Status is cleared	RW	0

Bits	Field Name	Description	Type	Reset
28	CERR	<p>Card error</p> <p>This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5 or R5b. Only bits referenced as type E(error) in status field in the response can set a card status error. An error bit in the response is flagged only if corresponding bit in card status response error <a href="#">MMCHS_CSRE</a> in set.</p> <p>There is no card error detection for autoCMD12 command. The host driver shall read <a href="#">MMCHS_RSP76</a> register to detect error bits in the command response.</p> <p>Write 0x0: Status bit unchanged</p> <p>Read 0x0: No Error</p> <p>Write 0x1: Status is cleared</p> <p>Read 0x1: Card error</p>	RW	0
27:26	RESERVED	<p>Reserved.</p> <p>These bits are initialized to zero, and writes to them are ignored.</p>	R	0x0
25	ADMAE	<p>ADMA Error:</p> <p>This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA Error Status Register. In addition, the Host Controller generates this interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor.</p> <p>Write 0x0: Status bit unchanged</p> <p>Read 0x0: No Interrupt.</p> <p>Write 0x1: Status is cleared</p> <p>Read 0x1: ADMA error</p>	RW	0
24	ACE	<p>Auto CMD12 error</p> <p>This bit is set automatically when one of the bits in Auto CMD12 Error status register has changed from 0 to 1.</p> <p>Write 0x0: Status bit unchanged</p> <p>Read 0x0: No Error.</p> <p>Read 0x1: AutoCMD12 error</p> <p>Write 0x1: Status is cleared</p>	RW	0
23	CLE	<p>Reserved.</p> <p>Current limit error is not supported.</p> <p>These bits are initialized to zero, and writes to them are ignored.</p>	R	0
22	DEB	<p>Data End Bit error</p> <p>This bit is set automatically when detecting a 0 at the end bit position of read data on DAT line or at the end position of the CRC status in write mode.</p> <p>Read 0x0: No Error</p> <p>Write 0x0: Status bit unchanged</p> <p>Write 0x1: Status is cleared</p> <p>Read 0x1: Data end bit error</p>	RW	0
21	DCRC	<p>Data CRC Error</p> <p>This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status different of a position "010" token during a block write command.</p> <p>Read 0x0: No Error.</p> <p>Write 0x0: Status bit unchanged</p> <p>Read 0x1: Data CRC error</p> <p>Write 0x1: Status is cleared</p>	RW	0

Bits	Field Name	Description	Type	Reset
20	DTO	<p>Data timeout error</p> <p>This bit is set automatically according to the following conditions:</p> <ul style="list-style-type: none"> <li>- busy timeout for R1b, R5b response type</li> <li>- busy timeout after write CRC status</li> <li>- write CRC status timeout</li> <li>- read data timeout</li> </ul> <p>Read 0x0: No error.</p> <p>Write 0x0: Status bit unchanged</p> <p>Write 0x1: Status is cleared</p> <p>Read 0x1: Time out</p>	RW	0
19	CIE	<p>Command index error</p> <p>This bit is set automatically when response index differs from corresponding command index previously emitted. It depends on the enable in <a href="#">MMCHS_CMD[20]</a> CICE register.</p> <p>Read 0x0: No error.</p> <p>Write 0x0: Status bit unchanged</p> <p>Read 0x1: Command index error</p> <p>Write 0x1: Status is cleared</p>	RW	0
18	CEB	<p>Command end bit error</p> <p>This bit is set automatically when detecting a 0 at the end bit position of a command response.</p> <p>Write 0x0: Status bit unchanged</p> <p>Read 0x0: No error.</p> <p>Read 0x1: Command end bit error</p> <p>Write 0x1: Status is cleared</p>	RW	0
17	CCRC	<p>Command CRC Error</p> <p>This bit is set automatically when there is a CRC7 error in the command response depending on the enable in <a href="#">MMCHS_CMD[19]</a> CCCE register.</p> <p>Read 0x0: No Error.</p> <p>Write 0x0: Status bit unchanged</p> <p>Write 0x1: Status is cleared</p> <p>Read 0x1: Command CRC error</p>	RW	0
16	CTO	<p>Command Timeout Error</p> <p>This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within 5 clock cycles - the timeout is still detected at 64 clock cycles.</p> <p>Write 0x0: Status bit unchanged</p> <p>Read 0x0: No error</p> <p>Read 0x1: Time Out</p> <p>Write 0x1: Status is cleared</p>	RW	0
15	ERRI	<p>Error Interrupt</p> <p>If any of the bits in the Error Interrupt Status register (<a href="#">MMCHS_STAT[24:15]</a>) are set, then this bit is set to 1. Therefore the host driver can efficiently test for an error by checking this bit first. Writes to this bit are ignored.</p> <p>Read 0x0: No Interrupt.</p> <p>Read 0x1: Error interrupt event(s) occurred</p>	R	0
14:11	RESERVED	<p>Reserved.</p> <p>These bits are initialized to zero, and writes to them are ignored.</p>	R	0x0

Bits	Field Name	Description	Type	Reset
10	BSR	<p>Boot status received interrupt This bit is set automatically when <a href="#">MMCHS_CON[18]</a> BOOT_CF0 is set 0x0 or 0x1 and a boot status is received on DAT[0] line. This interrupt is only useful for MMC card.</p> <p>Write 0x0: Status bit unchanged Read 0x0: No Interrupt. Write 0x1: Status is cleared Read 0x1: Boot status received interrupt.</p>	RW	0
9	OBI	<p>Out-Of-Band interrupt This bit is set automatically when <a href="#">MMCHS_CON[14]</a> OBIE is set and an Out-of-Band interrupt occurs on OBI pin. The interrupt detection depends on polarity controlled by <a href="#">MMCHS_CON[13]</a> OBIP. This interrupt is only useful for MMC card. The Out-of-Band interrupt signal is a system specific feature for future use, this signal is not required for existing specification implementation.</p> <p>Read 0x0: No Out-Of-Band interrupt. Write 0x0: Status bit unchanged Read 0x1: Interrupt Out-Of-Band occurs Write 0x1: Status is cleared</p>	RW	0
8	CIRQ	<p>Card interrupt This bit is only used for SD and SDIO and CE-ATA cards. In 1-bit mode, interrupt source is asynchronous (can be a source of asynchronous wakeup). In 4-bit mode, interrupt source is sampled during the interrupt cycle. In CE-ATA mode, interrupt source is detected when the card drives CMD line to zero during one cycle after data transmission end. All modes above are fully exclusive. The controller interrupt must be clear by setting <a href="#">MMCHS_IE[8]</a> CIRQ_ENABLE to 0, then the host driver must start the interrupt service with card (clearing card interrupt status) to remove card interrupt source. Otherwise the Controller interrupt will be reasserted as soon as <a href="#">MMCHS_IE[8]</a> CIRQ_ENABLE is set to 1. Writes to this bit are ignored.</p> <p>Read 0x0: No card interrupt Read 0x1: Generate card interrupt</p>	R	0
7	CREM	<p>Card removal This bit is set automatically when <a href="#">MMCHS_PSTATE[16]</a> CINS changes from 1 to 0. A clear of this bit doesn't affect Card inserted present state (<a href="#">MMCHS_PSTATE[16]</a> CINS).</p> <p>Write 0x0: Status bit unchanged Read 0x0: Card state stable or Debouncing Read 0x1: Card removed Write 0x1: Status is cleared</p>	RW	0
6	CINS	<p>Card insertion This bit is set automatically when <a href="#">MMCHS_PSTATE[16]</a> changes from 0 to 1. A clear of this bit doesn't affect Card inserted present state (<a href="#">MMCHS_PSTATE[16]</a> CINS).</p> <p>Read 0x0: Card state stable or debouncing Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Card inserted</p>	RW	0

Bits	Field Name	Description	Type	Reset
5	BRR	<p>Buffer read ready</p> <p>This bit is set automatically during a read operation to the card (see class 2 - block oriented read commands) when one block specified by <a href="#">MMCHS_BLK[11:0]</a> BLEN is completely written in the buffer. It indicates that the memory card has filled out the buffer and that the local host needs to empty the buffer by reading it. Note: If the DMA receive-mode is enabled, this bit is never set; instead a DMA receive request to the main DMA controller of the system is generated.</p> <p>Read 0x0: Not Ready to read buffer</p> <p>Write 0x0: Status bit unchanged</p> <p>Read 0x1: Ready to read buffer</p> <p>Write 0x1: Status is cleared</p>	RW	0
4	BWR	<p>Buffer write ready</p> <p>This bit is set automatically during a write operation to the card (see class 4 - block oriented write command) when the host can write a complete block as specified by <a href="#">MMCHS_BLK[11:0]</a> BLEN. It indicates that the memory card has emptied one block from the buffer and that the local host is able to write one block of data into the buffer. Note: If the DMA transmit mode is enabled, this bit is never set; instead, a DMA transmit request to the main DMA controller of the system is generated.</p> <p>Write 0x0: Status bit unchanged</p> <p>Read 0x0: Not Ready to write buffer</p> <p>Read 0x1: Ready to write buffer</p> <p>Write 0x1: Status is cleared</p>	RW	0
3	DMA	<p>DMA interrupt :</p> <p>This status is set when an interrupt is required in the ADMA instruction and after the data transfer completion.</p> <p>Read 0x0: Dma interrupt detected</p> <p>Write 0x0: Status bit unchanged</p> <p>Write 0x1: Status is cleared</p> <p>Read 0x1: No dma interrupt</p>	RW	0
2	BGE	<p>Block gap event</p> <p>When a stop at block gap is requested (<a href="#">MMCHS_HCTL[16]</a> SBGR), this bit is automatically set when transaction is stopped at the block gap during a read or write operation.</p> <p>This event does not occur when the stop at block gap is requested on the last block.</p> <p>In read mode, a 1-to-0 transition of the DAT Line active status (<a href="#">MMCHS_PSTATE[2]</a> DLA) between data blocks generates a Block gap event interrupt.</p> <p>Read 0x0: No block gap event</p> <p>Write 0x0: Status bit unchanged</p> <p>Write 0x1: Status is cleared</p> <p>Read 0x1: Transaction stopped at block gap</p>	RW	0

Bits	Field Name	Description	Type	Reset
1	TC	<p>Transfer completed</p> <p>This bit is always set when a read/write transfer is completed or between two blocks when the transfer is stopped due to a stop at block gap request (<a href="#">MMCHS_HCTL[16]</a> SBGR).</p> <p>In Read mode: This bit is automatically set on completion of a read transfer (<a href="#">MMCHS_PSTATE[9]</a> RTA).</p> <p>In write mode: This bit is set automatically on completion of the DAT line use (<a href="#">MMCHS_PSTATE[2]</a> DLA).</p> <p>Write 0x0: Status bit unchanged Read 0x0: No transfer complete Write 0x1: Status is cleared Read 0x1: Data transfer complete</p>	RW	0
0	CC	<p>Command complete</p> <p>This bit is set when a 1-to-0 transition occurs in the register command inhibit (<a href="#">MMCHS_PSTATE[0]</a> CMDI)</p> <p>If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command.</p> <p>A command timeout error (<a href="#">MMCHS_STAT[16]</a> CTO) has higher priority than command complete (<a href="#">MMCHS_STAT[0]</a> CC).</p> <p>If a response is expected but none is received, then a command timeout error is detected and signaled instead of the command complete interrupt.</p> <p>Write 0x0: Status bit unchanged Read 0x0: No Command complete Write 0x1: Status is cleared Read 0x1: Command complete</p>	RW	0

**Table 24-97. Register Call Summary for Register MMCHS\_STAT**

MMC/SD/SDIO Functional Description

- [Power Management: \[0\] \[1\] \[2\] \[3\]](#)
- [Interrupt Requests: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\] \[11\] \[12\] \[13\] \[14\] \[15\] \[16\] \[17\] \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\] \[25\] \[26\] \[27\] \[28\] \[29\] \[30\] \[31\] \[32\] \[33\]](#)
- [Interrupt-Driven Operation: \[34\] \[35\] \[36\] \[37\] \[38\] \[39\]](#)
- [Polling: \[40\] \[41\]](#)
- [Supported Features: \[42\]](#)
- [Data Buffer: \[43\] \[44\] \[45\]](#)
- [Data Buffer Status: \[46\] \[47\] \[48\]](#)
- [Transfer or Command Status and Errors Reporting: \[49\] \[50\]](#)
- [Busy Timeout For R1b, R5b Response Type: \[51\]](#)
- [Busy Timeout After Write CRC Status: \[52\]](#)
- [Write CRC Status Timeout: \[53\]](#)
- [Read Data Timeout: \[54\] \[55\]](#)
- [Boot Acknowledge Timeout: \[56\] \[57\] \[58\] \[59\] \[60\] \[61\]](#)
- [MMC CE-ATA Command Completion Disable Management: \[62\]](#)



**Table 24-97. Register Call Summary for Register MMCHS\_STAT (continued)**

MMC/SD/SDIO Programming Guide

- [Card Detection, Identification, and Selection: \[63\]](#)
- [Read/Write Transfer Flow in DMA Mode with Interrupt: \[64\]](#)
- [Read/Write Transfer Flow in DMA Mode with Polling: \[65\]](#)
- [Read/Write Transfer Flow without DMA with Polling: \[66\]](#)
- [Read/Write Transfer Flow in CE-ATA Mode: \[67\]](#)
- [Suspend-Resume Flow: \[68\]](#)
- [Basic Operations - Steps Detailed: \[69\] \[70\]](#)
- [Bus Voltage Selection: \[71\]](#)
- [Boot using the CMD0: \[72\]](#)
- [Boot with CMD line tied to 0: \[73\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[74\] \[75\]](#)
- [MMC/SD/SDIO Register Description: \[76\] \[77\] \[78\] \[79\] \[80\] \[81\] \[82\] \[83\] \[84\] \[85\] \[86\] \[87\] \[88\] \[89\] \[90\] \[91\] \[92\] \[93\] \[94\] \[95\] \[96\] \[97\] \[98\] \[99\] \[100\] \[101\] \[102\] \[103\] \[104\] \[105\] \[106\] \[107\] \[108\]](#)

**Table 24-98. MMCHS\_IE**

<b>Address Offset</b>	0x0000 0234	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	Interrupt SD enable register This register allows to enable/disable the module to set status bits, on an event-by-event basis. <a href="#">MMCHS_IE[31:16]</a> = Error Interrupt Status Enable <a href="#">MMCHS_IE[15:0]</a> = Normal Interrupt Status Enable		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RESERVED	BADA_ENABLE	CERR_ENABLE	RESERVED	ADMAE_ENABLE	ACE_ENABLE	CLE	DEB_ENABLE	DCRC_ENABLE	DTO_ENABLE	CIE_ENABLE	CEB_ENABLE	CCRC_ENABLE	CTO_ENABLE	NULL	RESERVED	BSR_ENABLE	OBI_ENABLE	CIRQ_ENABLE	CREM_ENABLE	CINS_ENABLE	BRR_ENABLE	BWR_ENABLE	DMA_ENABLE	BGE_ENABLE	TC_ENABLE	CC_ENABLE						

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	BADA_ENABLE	Bad access to data space Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
28	CERR_ENABLE	Card error interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
27:26	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0
25	ADMAE_ENABLE	ADMA error Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
24	ACE_ENABLE	Auto CMD12 error Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
23	CLE	Reserved bit. Current limit error is not supported. These bits are initialized to zero, and writes to them are ignored.	R	0
22	DEB_ENABLE	Data end bit error Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
21	DCRC_ENABLE	Data CRC error Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
20	DTO_ENABLE	Data timeout error Interrupt Enable 0x0: The data timeout detection is deactivated. The host controller provides the clock to the card until the card sends the data or the transfer is aborted. 0x1: The data timeout detection is enabled.	RW	0
19	CIE_ENABLE	Command index error Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
18	CEB_ENABLE	Command end bit error Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
17	CCRC_ENABLE	Command CRC error Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
16	CTO_ENABLE	Command timeout error Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
15	NULL	Fixed to 0 The host driver shall control error interrupts using the Error Interrupt Signal Enable register. Writes to this bit are ignored	R	0
14:11	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0
10	BSR_ENABLE	Boot status interrupt Enable A write to this register when <a href="#">MMCHS_CON[17]</a> BOOT_ACK is set to 0x0 is ignored. 0x0: Masked 0x1: Enabled	RW	0
9	OBI_ENABLE	Out-of-Band interrupt Enable A write to this register when <a href="#">MMCHS_CON[14]</a> OBIE is set to 0 is ignored. 0x0: Masked 0x1: Enabled	RW	0
8	CIRQ_ENABLE	Card interrupt Enable A clear of this bit also clears the corresponding status bit. During 1-bit mode, if the interrupt routine doesn't remove the source of a card interrupt in the SDIO card, the status bit is reasserted when this bit is set to 1. 0x0: Masked 0x1: Enabled	RW	0
7	CREM_ENABLE	Card removal Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
6	CINS_ENABLE	Card insertion Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
5	BRR_ENABLE	Buffer Read Ready Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
4	BWR_ENABLE	Buffer Write Ready Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
3	DMA_ENABLE	DMA interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
2	BGE_ENABLE	Block Gap Event Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
1	TC_ENABLE	Transfer completed Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0
0	CC_ENABLE	Command completed Interrupt Enable 0x0: Masked 0x1: Enabled	RW	0

**Table 24-99. Register Call Summary for Register MMCHS\_IE**

MMC/SD/SDIO Functional Description

- [Power Management](#): [0] [1]
- [Interrupt Requests](#): [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24]
- [Interrupt-Driven Operation](#): [25] [26]

MMC/SD/SDIO Programming Guide

- [Wakeup Configuration](#): [27]
- [Read/Write Transfer Flow in CE-ATA Mode](#): [28]
- [Basic Operations - Steps Detailed](#): [29] [30]

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary](#): [31] [32]
- [MMC/SD/SDIO Register Description](#): [33] [34] [35] [36]

**Table 24-100. MMCHS\_ISE**

<b>Address Offset</b>	0x0000 0238	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	Interrupt signal enable register This register allows to enable/disable the module internal sources of status, on an event-by-event basis. <a href="#">MMCHS_ISE[31:16]</a> = Error Interrupt Signal Enable <a href="#">MMCHS_ISE[15:0]</a> = Normal Interrupt Signal Enable		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	BADA_SIGEN	CERR_SIGEN	RESERVED	ADMAE_SIGEN	ACE_SIGEN	CLE	DEB_SIGEN	DCRC_SIGEN	DTO_SIGEN	CIE_SIGEN	CEB_SIGEN	CCRC_SIGEN	CTO_SIGEN	NULL	RESERVED	BSR_SIGEN	OBI_SIGEN	CIRQ_SIGEN	CREM_SIGEN	CINS_SIGEN	BRR_SIGEN	BWR_SIGEN	DMA_SIGEN	BGE_SIGEN	TC_SIGEN	CC_SIGEN					

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	BADA_SIGEN	Bad access to data space signal status Enable 0x0: Masked 0x1: Enabled	RW	0
28	CERR_SIGEN	Card error interrupt signal status Enable 0x0: Masked 0x1: Enabled	RW	0
27:26	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0
25	ADMAE_SIGEN	ADMA error signal status Enable 0x0: Masked 0x1: Enabled	RW	0
24	ACE_SIGEN	Auto CMD12 error signal status Enable 0x0: Masked 0x1: Enabled	RW	0
23	CLE	Reserved bit. Current limit error is not supported. These bits are initialized to zero, and writes to them are ignored.	R	0
22	DEB_SIGEN	Data end bit error signal status Enable 0x0: Masked 0x1: Enabled	RW	0
21	DCRC_SIGEN	Data CRC error signal status Enable 0x0: Masked 0x1: Enabled	RW	0
20	DTO_SIGEN	Data timeout error signal status Enable 0x0: Masked 0x1: Enabled	RW	0
19	CIE_SIGEN	Command index error signal status Enable 0x0: Masked 0x1: Enabled	RW	0
18	CEB_SIGEN	Command end bit error signal status Enable 0x0: Masked 0x1: Enabled	RW	0
17	CCRC_SIGEN	Command CRC error signal status Enable 0x0: Masked 0x1: Enabled	RW	0
16	CTO_SIGEN	Command timeout error signal status Enable 0x0: Masked 0x1: Enabled	RW	0
15	NULL	Fixed to 0 The host driver shall control error interrupts using the Error Interrupt Signal Enable register. Writes to this bit are ignored	R	0
14:11	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0
10	BSR_SIGEN	Boot status signal status Enable A write to this register when <code>MMCHS_CON[17]</code> <code>BOOT_ACK</code> is set to 0x0 is ignored. 0x0: Masked 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
9	OBI_SIGEN	Out-Of-Band Interrupt signal status Enable A write to this register when <a href="#">MMCHS_CON</a> [14] OBIE is set to 0 is ignored. 0x0: Masked 0x1: Enabled	RW	0
8	CIRQ_SIGEN	Card interrupt signal status Enable 0x0: Masked 0x1: Enabled	RW	0
7	CREM_SIGEN	Card removal signal status Enable 0x0: Masked 0x1: Enabled	RW	0
6	CINS_SIGEN	Card insertion signal status Enable 0x0: Masked 0x1: Enabled	RW	0
5	BRR_SIGEN	Buffer Read Ready signal status Enable 0x0: Masked 0x1: Enabled	RW	0
4	BWR_SIGEN	Buffer Write Ready signal status Enable 0x0: Masked 0x1: Enabled	RW	0
3	DMA_SIGEN	DMA interrupt Signal status enable 0x0: Masked 0x1: Enabled	RW	0
2	BGE_SIGEN	Black Gap Event signal status Enable 0x0: Masked 0x1: Enabled	RW	0
1	TC_SIGEN	Transfer completed signal status Enable 0x0: Masked 0x1: Enabled	RW	0
0	CC_SIGEN	Command completed signal status Enable 0x0: Masked 0x1: Enabled	RW	0

**Table 24-101. Register Call Summary for Register MMCHS\_ISE**

MMC/SD/SDIO Functional Description

- [Power Management: \[0\] \[1\]](#)
- [Interrupt Requests: \[2\] \[3\] \[4\] \[5\] \[6\]](#)
- [Polling: \[7\]](#)

MMC/SD/SDIO Programming Guide

- [Basic Operations - Steps Detailed: \[8\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[9\] \[10\]](#)
- [MMC/SD/SDIO Register Description: \[11\] \[12\] \[13\]](#)

**Table 24-102. MMCHS\_AC12**

<b>Address Offset</b>	0x0000 023C	
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b> MMCHS
<b>Description</b>	Auto CMD12 Error Status Register The host driver may determine which of the errors cases related to Auto CMD12 has occurred by checking this <a href="#">MMCHS_AC12</a> register when an Auto CMD12 Error interrupt occurs. This register is valid only when Auto CMD12 is enabled ( <a href="#">MMCHS_CMD[2]</a> ACEN) and Auto CMD12Error ( <a href="#">MMCHS_STAT[24]</a> ACE) is set to 1. Note: These bits are automatically reset when starting a new adtc command with data.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CNI	RESERVED	ACIE	ACEB	ACCE	ACTO	ACNE									

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored.	R	0x000000
7	CNI	Command not issue by Auto CMD12 error If this bit is set to 1, it means that pending command is not executed due to Auto CMD12 error : ACEB, ACCE, ACTO or ACNE.  Read 0x0: Not error Read 0x1: Command not issued	R	0
6:5	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored.	R	0x0
4	ACIE	Auto CMD12 index error This bit is a set to 1 when response index differs from corresponding command auto CMD12 index previously emitted. This bit depends on the command index check enable ( <a href="#">MMCHS_CMD[20]</a> CICE).  Read 0x0: No error Read 0x1: Auto CMD12 Index Error	R	0
3	ACEB	Auto CMD12 end bit error This bit is set to 1 when detecting a 0 at the end bit position of auto CMD12 command response.  Read 0x0: No error Read 0x1: AutoCMD12 End bit Error	R	0
2	ACCE	Auto CMD12 CRC error This bit is automatically set to 1 when a CRC7 error is detected in the auto CMD12 command response depending on the enable in <a href="#">MMCHS_CMD[19]</a> CCCE register.  Read 0x0: No error Read 0x1: Auto CMD12 CRC Error	R	0
1	ACTO	Auto CMD12 timeout error This bit is set to 1 if no response is received within 64 clock cycles from the end bit of the auto CMD12 command.  Read 0x0: No error Read 0x1: Auto CMD12 Time Out	R	0

Bits	Field Name	Description	Type	Reset
0	ACNE	Auto CMD12 not executed This bit is set to 1 if multiple block data transfer command has started and if an error occurs in command before Auto CMD12 starts. Read 0x0: Auto CMD12 Executed Read 0x1: Auto CMD12 Not Executed	R	0

**Table 24-103. Register Call Summary for Register MMCHS\_AC12**

MMC/SD/SDIO Programming Guide

- [Read/Write Transfer Flow in DMA Mode with Polling: \[0\]](#)
- [Read/Write Transfer Flow without DMA with Polling: \[1\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[2\] \[3\]](#)
- [MMC/SD/SDIO Register Description: \[4\] \[5\]](#)

**Table 24-104. MMCHS\_CAPA**

<b>Address Offset</b>	0x0000 0240	
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b> MMCHS
<b>Description</b>	Capabilities register This register lists the capabilities of the MMC/SD/SDIO host controller.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	BIT64	RESERVED	VS18	VS30	VS33	SRS	DS	HSS	RESERVED	AD2S	RESERVED	MBL	RESERVED	BCF	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	R	0x0
28	BIT64	64 Bit System Bus Support: Setting 1 to this bit indicates that the Host Controller supports 64-bit address descriptor mode and is connected to 64-bit address system bus. Read 0x0: 32 bit System bus address Read 0x1: 64 bit System bus address	R	0
27	RESERVED		R	0
26	VS18	Voltage support 1.8V Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via RESETN signal) Write 0x0: 1.8V Not supported Read 0x0: 1.8V Not Supported Read 0x1: 1.8V Supported Write 0x1: 1.8V Supported	RW	0

Bits	Field Name	Description	Type	Reset
25	VS30	<p>Voltage support 3.0V</p> <p>Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via RESETN signal)</p> <p>Read 0x0: 3.0V Not Supported</p> <p>Write 0x0: 3.0V Not supported</p> <p>Write 0x1: 3.0V Supported</p> <p>Read 0x1: 3.0V Supported</p>	RW	0
24	VS33	<p>Voltage support 3.3V</p> <p>Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via RESETN signal)</p> <p>Read 0x0: 3.3V Not Supported</p> <p>Write 0x0: 3.3V Not supported</p> <p>Write 0x1: 3.3V Supported</p> <p>Read 0x1: 3.3V Supported</p>	RW	0
23	SRS	<p>Suspend/Resume support (SDIO cards only)</p> <p>This bit indicates whether the host controller supports Suspend/Resume functionality.</p> <p>Read 0x0: The Host controller does not Suspend/Resume functionality.</p> <p>Read 0x1: The Host controller supports Suspend/Resume functionality.</p>	R	1
22	DS	<p>DMA support</p> <p>This bit indicates that the Host Controller is able to use DMA to transfer data between system memory and the Host Controller directly.</p> <p>Read 0x0: DMA Not Supported</p> <p>Read 0x1: DMA Supported</p>	R	1
21	HSS	<p>High speed support</p> <p>This bit indicates that the host controller supports high speed operations and can supply an up-to maximum card frequency.</p> <p>Read 0x0: High Speed Not Supported</p> <p>Read 0x1: High Speed Supported</p>	R	1
20	RESERVED	<p>Reserved.</p> <p>This bit is initialized to zero, and writes to it are ignored.</p>	R	0
19	AD2S	<p>ADMA2 Support:</p> <p>This bit indicates whether the Host Controller is capable of using ADMA2. It depends on setting of generic parameter <code>MMCHS_HL_HWINFO[0] MADMA_EN</code></p> <p>Read 0x0: ADMA2 Supported</p> <p>Read 0x1: ADMA2 Supported</p>	R	1
18	RESERVED	<p>Reserved.</p> <p>This bit is initialized to zero, and writes to it are ignored.</p>	R	0



Bits	Field Name	Description	Type	Reset
17:16	MBL	<p>Maximum block length</p> <p>This value indicates the maximum block size that the host driver can read and write to the buffer in the host controller.</p> <p>This value depends on definition of generic parameter with a max value of 2048 bytes.</p> <p>The host controller supports 512 bytes and 1024 bytes block transfers.</p> <p>Read 0x0: 512 bytes</p> <p>Read 0x1: 1024 bytes</p> <p>Read 0x2: 2048 bytes</p>	R	0x1
15:14	RESERVED	<p>Reserved.</p> <p>These bits are initialized to zero, and writes to them are ignored.</p>	R	0x0
13:8	BCF	<p>Base clock frequency for clock provided to the card.</p> <p>Read 0x0: The value indicating the base (maximum) frequency for the output clock provided to the card is system dependent and is not available in this register. Get the information via another method.</p>	R	0x00
7	TCU	<p>Timeout clock unit</p> <p>This bit shows the unit of base clock frequency used to detect Data Timeout Error (<a href="#">MMCHS_STAT[20] DTO</a>).</p> <p>Read 0x0: KHz</p> <p>Read 0x1: MHz</p>	R	1
6	RESERVED	<p>Reserved.</p> <p>This bit is initialized to zero, and writes to it are ignored.</p>	R	0
5:0	TCF	<p>Timeout clock frequency</p> <p>The timeout clock frequency is used to detect Data Timeout Error (<a href="#">MMCHS_STAT[20] DTO</a>).</p> <p>Read 0x0: The timeout clock frequency depends on the frequency of the clock provided to the card. The value of the timeout clock frequency is not available in this register.</p>	R	0x00

**Table 24-105. Register Call Summary for Register MMCHS\_CAPA**

MMC/SD/SDIO Functional Description

- [Software Reset: \[0\]](#)
- [Data Buffer: \[1\]](#)

MMC/SD/SDIO Programming Guide

- [Set MMCHS Default Capabilities: \[2\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[3\] \[4\]](#)
- [MMC/SD/SDIO Register Description: \[5\] \[6\] \[7\] \[8\]](#)

**Table 24-106. MMCHS\_CUR\_CAPA**

<b>Address Offset</b>	0x0000 0248		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	<p>Maximum current capabilities Register</p> <p>This register indicates the maximum current capability for each voltage. The value is meaningful if the voltage support is set in the capabilities register (<a href="#">MMCHS_CAPA</a>).</p> <p>Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization.</p> <p>This register is only reinitialized by a hard reset (via RESETN signal)</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CUR_1V8								CUR_3V0								CUR_3V3							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved. This bit is initialized to zero, and writes to it are ignored.	R	0x00
23:16	CUR_1V8	Maximum current for 1.8V  Read 0x0: The maximum current capability for this voltage is not available. Feature not implemented.	RW	0x00
15:8	CUR_3V0	Maximum current for 3.0V  Read 0x0: The maximum current capability for this voltage is not available. Feature not implemented.	RW	0x00
7:0	CUR_3V3	Maximum current for 3.3V  Read 0x0: The maximum current capability for this voltage is not available. Feature not implemented.	RW	0x00

**Table 24-107. Register Call Summary for Register MMCHS\_CUR\_CAPA**

MMC/SD/SDIO Functional Description

- [Software Reset: \[0\]](#)

MMC/SD/SDIO Programming Guide

- [Set MMCHS Default Capabilities: \[1\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[2\] \[3\]](#)

**Table 24-108. MMCHS\_FE**

<b>Address Offset</b>	0x0000 0250		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	<p>Force Event Register for Error Interrupt status</p> <p>The force Event Register is not a physically implemented register. Rather, it is an address at which the Error Interrupt Status register can be written. The effect of a write to this address will be reflected in the Error Interrupt Status Register, if corresponding bit of the Error Interrupt Status Enable Register is set.</p>		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	FE_BADA	FE_CERR	RESERVED	FE_ADMAE	FE_ACE	FE_CLE	FE_DEB	FE_DCRC	FE_DTO	FE_CIE	FE_CEB	FE_CCRC	FE_CTO	RESERVED								FE_CNI	RESERVED	FE_ACIE	FE_ACEB	FE_ACCE	FE_ACTO	FE_ACNE			

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	N/A <sup>(1)</sup>	0x0
29	FE_BADA	Force Event Bad access to data space Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
28	FE_CERR	Force Event Card error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
27:26	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	N/A <sup>(1)</sup>	0x0
25	FE_ADMAE	Force Event ADMA Error: Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
24	FE_ACE	Force Event Auto CMD12 error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
23	FE_CLE	Reserved. Current limit error is not supported. These bits are initialized to zero, and writes to them are ignored.	N/A <sup>(1)</sup>	0
22	FE_DEB	Force Event Data End Bit error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
21	FE_DCRC	Force Event Data CRC Error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
20	FE_DTO	Force Event Data timeout error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
19	FE_CIE	Force Event Command index error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
18	FE_CEB	Force Event Command end bit error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
17	FE_CCRC	Force Event Command CRC Error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
16	FE_CTO	Command Timeout Error  This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command.  For commands that reply within 5 clock cycles - the timeout is still detected at 64 clock cycles.  Write 0x0: Status bit unchanged Write 0x1: Status is cleared	W	0
15:8	RESERVED	Reserved. These bits are initialized to zero, and writes to them are ignored.	N/A <sup>(1)</sup>	0x00

<sup>(1)</sup> N/A = Not accessible

Bits	Field Name	Description	Type	Reset
7	FE_CNI	Force Event Command not issue by Auto CMD12 error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
6:5	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored.	N/A <sup>(1)</sup>	0x0
4	FE_ACIE	Force Event Auto CMD12 index error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
3	FE_ACEB	Force Event Auto CMD12 end bit error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
2	FE_ACCE	Force Event Auto CMD12 CRC error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
1	FE_ACTO	Force Event Auto CMD12 timeout error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
0	FE_ACNE	Force Event Auto CMD12 not executed Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0

**Table 24-109. Register Call Summary for Register MMCHS\_FE**

MMC/SD/SDIO Functional Description

- [Test Registers: \[0\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[1\] \[2\]](#)

**Table 24-110. MMCHS\_ADMAES**

<b>Address Offset</b>	0x0000 0254																																																																
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>										<b>Instance</b>	MMCHS																																																					
<b>Description</b>	<p>ADMA Error Status Register</p> <p>When ADMA Error Interrupt is occurred, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address Register holds the address around the error descriptor. For recovering the error, the Host Driver requires the ADMA state to identify the error descriptor address as follows:</p> <p>ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address</p> <p>ST_FDS: Current location set in the ADMA System Address register is the error descriptor address</p> <p>ST_CADR: This state is never set because do not generate ADMA error in this state.</p> <p>ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address</p> <p>In case of write operation, the Host Driver should use ACMD22 to get the number of written block rather than using this information, since unwritten data may exist in the Host Controller. The Host Controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. In this case, ADMA Error State indicates that an error occurs at ST_FDS state. The Host Driver may find that the Valid bit is not set in the error descriptor.</p>																																																																
<b>Type</b>	RW																																																																
<table border="1" style="width:100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 2.5%;">31</th><th style="width: 2.5%;">30</th><th style="width: 2.5%;">29</th><th style="width: 2.5%;">28</th><th style="width: 2.5%;">27</th><th style="width: 2.5%;">26</th><th style="width: 2.5%;">25</th><th style="width: 2.5%;">24</th><th style="width: 2.5%;">23</th><th style="width: 2.5%;">22</th><th style="width: 2.5%;">21</th><th style="width: 2.5%;">20</th><th style="width: 2.5%;">19</th><th style="width: 2.5%;">18</th><th style="width: 2.5%;">17</th><th style="width: 2.5%;">16</th><th style="width: 2.5%;">15</th><th style="width: 2.5%;">14</th><th style="width: 2.5%;">13</th><th style="width: 2.5%;">12</th><th style="width: 2.5%;">11</th><th style="width: 2.5%;">10</th><th style="width: 2.5%;">9</th><th style="width: 2.5%;">8</th><th style="width: 2.5%;">7</th><th style="width: 2.5%;">6</th><th style="width: 2.5%;">5</th><th style="width: 2.5%;">4</th><th style="width: 2.5%;">3</th><th style="width: 2.5%;">2</th><th style="width: 2.5%;">1</th><th style="width: 2.5%;">0</th> </tr> </thead> <tbody> <tr> <td colspan="16" style="text-align: center;">RESERVED</td> <td style="text-align: center;">LME</td> <td style="text-align: center;">AES</td> </tr> </tbody> </table>																31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																LME	AES
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																		
RESERVED																LME	AES																																																

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	LME	ADMA Length Mismatch Error: (1) While Block Count Enable being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. (2) Total data length can not be divided by the block length.  0x0: No Error 0x1: Error	RW	0
1:0	AES	ADMA Error State his field indicates the state of ADMA when error is occurred during ADMA data transfer. This field never indicates "10" because ADMA never stops in this state.  0x0: ST_STOP (Stop DMA)Contents of SYS_SDR register 0x1: ST_STOP (Stop DMA)Points the error descriptor 0x2: Never set this state(Not used) 0x3: ST_TFR (Transfer Data)Points the next of the error descriptor	RW	0x0

**Table 24-111. Register Call Summary for Register MMCHS\_ADMAES**

MMC/SD/SDIO Functional Description

- [Error Generation: \[0\]](#)

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[1\] \[2\]](#)

**Table 24-112. MMCHS\_ADMASAL**

<b>Address Offset</b>	0x0000 0258	<b>Instance</b>	MMCHS
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>		
<b>Description</b>	ADMA System address Low bits		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADMA_A32B																															

Bits	Field Name	Description	Type	Reset
31:0	ADMA_A32B	ADMA System address 32 bits.This register holds byte address of executing command of the Descriptor table. 32-bit Address Descriptor uses lower 32-bit of this register. At the start of ADMA, the Host Driver shall set start address of the Descriptor table. The ADMA increments this register address, which points to next line, when every fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register shall hold valid Descriptor address depending on the ADMA state. The Host Driver shall program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores lower 2-bit of this register and assumes it to be 00b.	RW	0x0000 0000

**Table 24-113. Register Call Summary for Register MMCHS\_ADMASAL**

MMC/SD/SDIO Functional Description

- [MASTER DMA Operations: \[0\] \[1\]](#)
- [Error Generation: \[2\]](#)
- [Advanced DMA Description: \[3\] \[4\] \[5\]](#)

**Table 24-113. Register Call Summary for Register MMCHS\_ADMASAL (continued)**

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[6\] \[7\]](#)

**Table 24-114. MMCHS\_REV**

<b>Address Offset</b>	0x0000 02FC		
<b>Physical Address</b>	See <a href="#">Table 24-51</a> to <a href="#">Table 24-52</a>	<b>Instance</b>	MMCHS
<b>Description</b>	Versions Register This register contains the hard coded RTL vendor revision number, the version number of SD specification compliancy and a slot status bit. <a href="#">MMCHS_REV[31:16]</a> = Host controller version <a href="#">MMCHS_REV[15:0]</a> = Slot Interrupt Status		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VREV								SREV								RESERVED								SIS							

Bits	Field Name	Description	Type	Reset
31:24	VREV	Vendor Version Number: IP revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0 0x21 for 2.1	R	0x--
23:16	SREV	Specification Version Number This status indicates the Standard SD Host Controller Specification Version. The upper and lower 4-bits indicate the version. Read 0x0: SD Host Specification Version 1.0 Read 0x1: SD Host Specification Version 2.0	R	0x01
15:1	RESERVED	Reserved These bits are initialized to zero, and writes to them are ignored.	R	0x0000
0	SIS	Slot Interrupt Status This status bit indicates the inverted state of interrupt signal for the module. By a power on reset or by setting a software reset for all ( <a href="#">MMCHS_SYSCTL[24]</a> SRA), the interrupt signal shall be de-asserted and this status shall read 0.	R	0

**Table 24-115. Register Call Summary for Register MMCHS\_REV**

MMC/SD/SDIO Register Manual

- [MMC/SD/SDIO Register Summary: \[0\] \[1\]](#)
- [MMC/SD/SDIO Register Description: \[2\] \[3\]](#)

## General-Purpose Interface

This chapter describes the general-purpose interface on the device.

Topic	Page
25.1 General-Purpose Interface Overview .....	5088
25.2 General-Purpose Interface Environment .....	5091
25.3 General-Purpose Interface Integration .....	5095
25.4 General-Purpose Interface Functional Description .....	5099
25.5 General-Purpose Interface Programming Guide .....	5115
25.6 General-Purpose Interface Register Manual .....	5117

## 25.1 General-Purpose Interface Overview

The general-purpose interface combines six general-purpose input/output (GPIO) banks.

Each GPIO module provides 32 dedicated general-purpose pins with input and output capabilities; thus, the general-purpose interface supports up to 192 (6 × 32) pins.

These pins can be configured for the following applications:

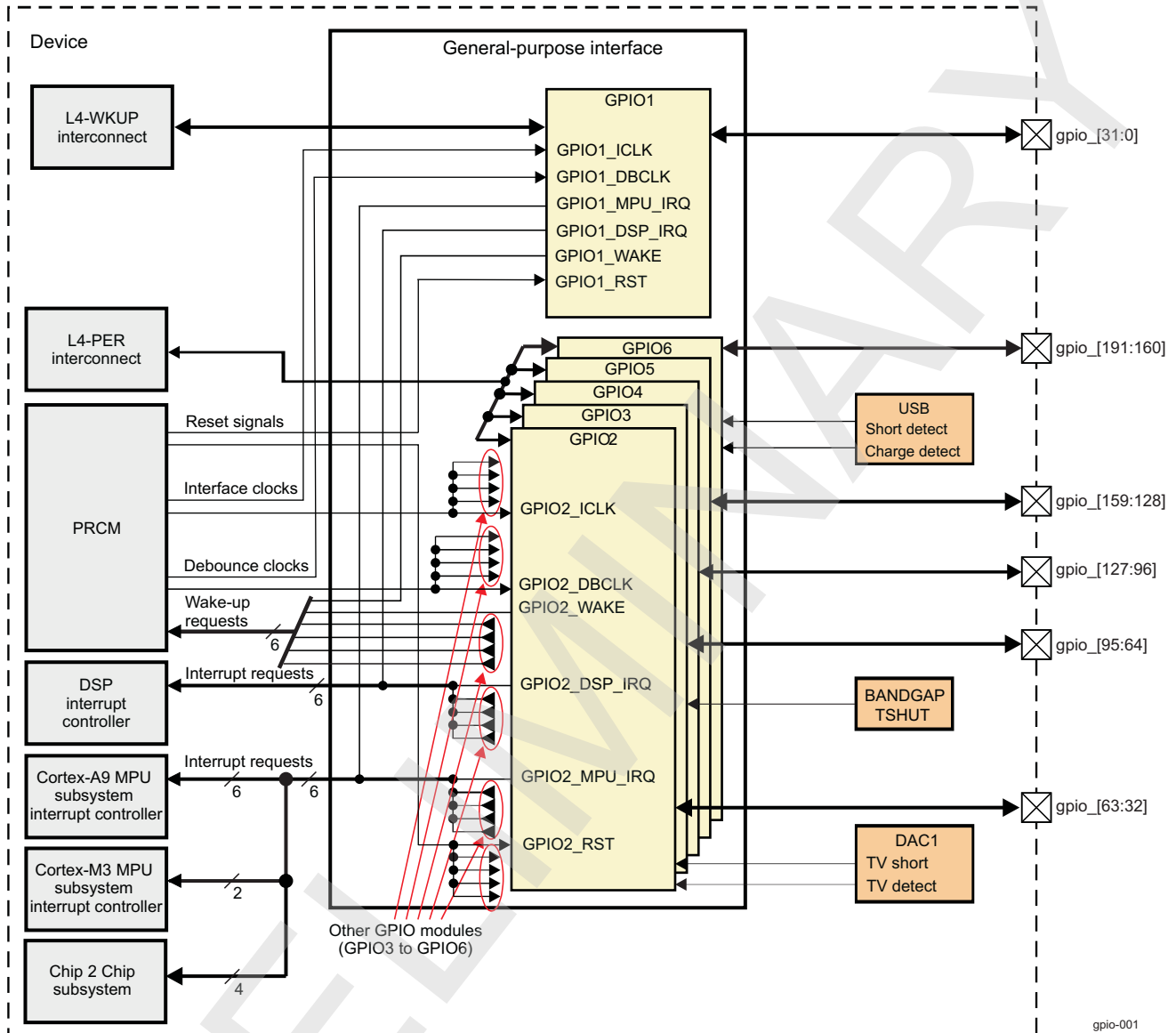
- Data input (capture)/output (drive)
- Keyboard interface with a debounce cell
- Interrupt generation in active mode upon the detection of external events. Detected events are processed by two parallel independent interrupt-generation submodules to support biprocessor operations.
- Wake-up request generation in idle mode upon the detection of external events

These modules do not include pad control (pullup/pulldown control, open-drain feature). For more information, see [Section 18.4.8, Pad Functional Multiplexing and Configuration](#), in [Chapter 18, Control Module](#).

[Figure 25-1](#) shows an overview of the general-purpose interface.



Figure 25-1. General-Purpose Interface Overview



The GPIO modules include the following global features:

- Synchronous interrupt requests in active mode from each channel are processed by two identical interrupt generation submodules used independently by the digital signal processor (DSP) and microprocessor unit (MPU) Cortex™-A9 and Cortex™-M3 subsystems. One of these interrupts is mapped on the digital signal processor (DSP) subsystem interrupt controller and the other on the MPU Cortex-A9 and Cortex-M3 subsystem interrupt controllers (INTCs).
- Asynchronous wake-up requests in idle mode from input channels are merged together to issue one wake-up signal per GPIO module.
- Data input (capture)/output (drive)
- Power management support

The general-purpose interface has 12 interrupt lines (two interrupt lines per GPIO module instance).

Each GPIO module produces a wake-up request signal to the power, reset, and clock management (PRCM) module.

Each channel in the GPIO modules has the following features:

- The GPIOi.GPIO\_OE register controls the output capability for each pin.

- The output line level reflects the value written in the GPIOi.GPIO\_DATAOUT register through the L4 interconnect.
- The input line can be fed to the GPIO module through an optional and configurable debounce cell. (Because the debouncing time value is global for all ports of one GPIO module, up to five different debouncing time values are possible.)
- The input line value is sampled into the GPIOi.GPIO\_DATAIN register and can be read through the L4 interconnect.
- In active mode, the input line can be used through level and edge detectors to trigger synchronous interrupts. The edge (rising, falling, or both) or the level used (logical 0, logical 1, or both) can be configured.
- In idle mode, the input line can be used to activate the asynchronous wake-up request (on edge detection: rising edge, falling edge, or both).

The module provides an alternative to the atomic test and set operations for the following registers:

- GPIOi.GPIO\_DATAOUT
- GPIOi.GPIO\_IRQSTATUS\_SET\_0
- GPIOi.GPIO\_IRQSTATUS\_SET\_1
- GPIOi.GPIO\_IRQSTATUS\_CLR\_0
- GPIOi.GPIO\_IRQSTATUS\_CLR\_1
- GPIOi.GPIO\_IRQWAKEN\_0
- GPIOi.GPIO\_IRQWAKEN\_1

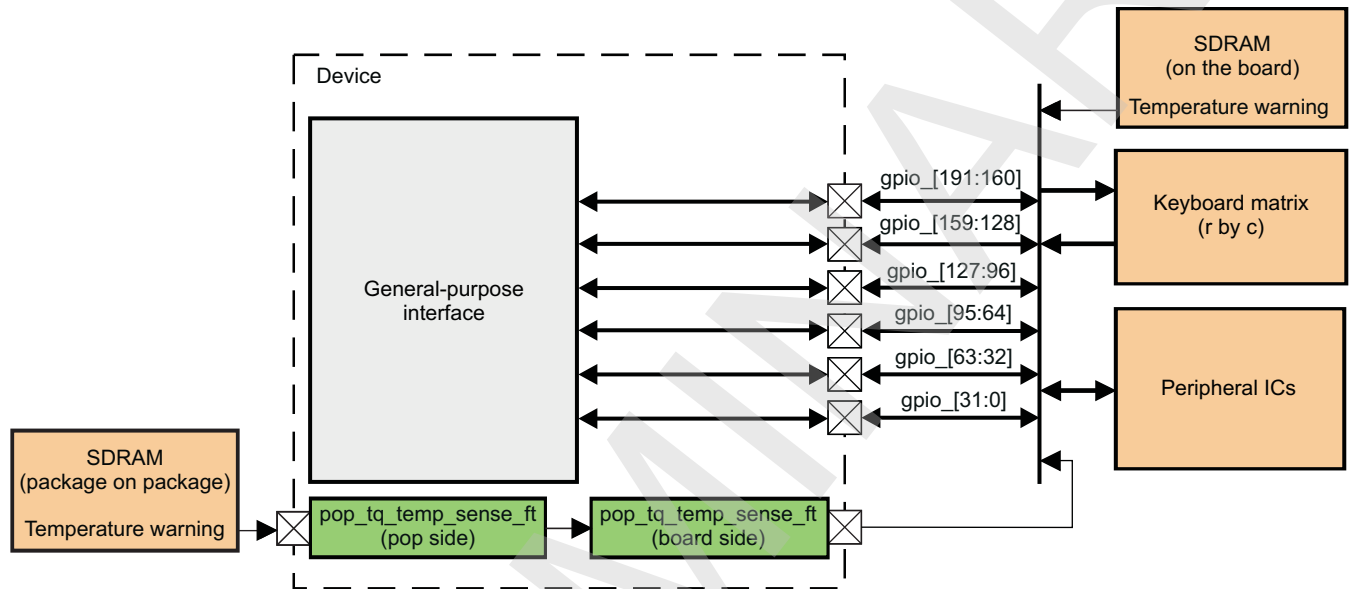
For these registers, the modules implement the set-and-clear protocol register update (see [Section 25.4.9, General-Purpose Interface Set-and-Clear Protocol](#)).

## 25.2 General-Purpose Interface Environment

The general-purpose interface combines six GPIO modules for a flexible, user-programmable, GPIO controller. The general-purpose interface implements functions that are not implemented with the dedicated controllers in the device and require simple input and/or output software-controlled signals. The general-purpose interface allows a variety of custom connections and expands the I/O capabilities of the system to the real world.

Figure 25-2 shows a typical application using the general-purpose interface.

Figure 25-2. General-Purpose Interface Typical Application



gpio-002

**NOTE:** Most memories provide a temperature sensor to control the autorefresh duty cycle. The device monitors the temperature of the external memory using the pop\_tq\_temp\_sense\_ft ball and a GPIO input. To do this, pop\_tq\_temp\_sense\_ft is connected to a GPIO through the customer board. This feature is application-dependent.

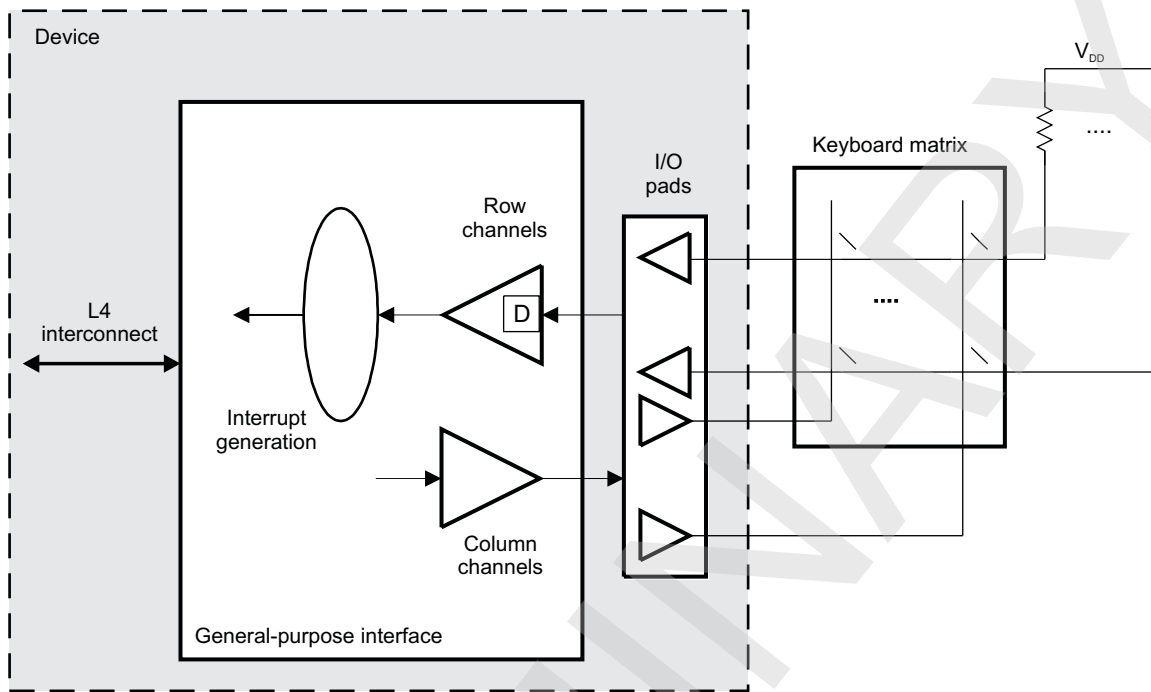
### CAUTION

Because of buffer strength, an external serial resistor must be connected to the balls where gpio\_120 through gpio\_129 are muxed with MMC signals.

The general-purpose interface can physically connect the device to a keyboard matrix and peripheral integrated circuits (ICs).

### 25.2.1 General-Purpose Interface as a Keyboard Interface

The general-purpose interface can be used as a keyboard interface. Channels can be dedicated based on the keyboard matrix (r \* c). Figure 25-3 shows row channels configured as inputs with the input debounce feature enabled. The row channels are driven high with an external pullup. Column channels are configured as outputs and drive a low level.

**Figure 25-3. General-Purpose Interface Used as a Keyboard Interface**

gpio-003

When a keyboard matrix key is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row channel. This generates an interrupt based on the proper configuration (see [Section 25.4.6, Interrupt and Wake-up Requests](#)).

When the keyboard interrupt is received, the processor (MPU and/or DSP subsystem) can disable the keyboard interrupt and scan the column channels for the key coordinates.

- The scanning sequence has as many states as column channels: For each step in the sequence, the processor drives one column channel low and the others high.
- The processor reads the values of the row channels and thus detects which keys in the column are pressed.

At the end of the scanning sequence, the processor establishes which keys are pressed. The keyboard interface can then be reconfigured in the interrupt waiting state.

## 25.2.2 General-Purpose Interface Signals

[Table 25-1](#) describes the module signals.

**Table 25-1. I/O**

Signal	I/O <sup>(1)</sup>	Description	Reset Value
gpio_[31:0]	I/O	GPIO	HiZ
gpio_[63:32]	I/O	GPIO	HiZ
gpio_[95:64]	I/O	GPIO	HiZ
gpio_[127:96]	I/O	GPIO	HiZ
gpio_[159:128]	I/O	GPIO	HiZ
gpio_[191:160]	I/O	GPIO	HiZ

<sup>(1)</sup> I = Input; O = Output

Figure 25-4 shows the signal connections of GPIO module 1.

Figure 25-4. GPIO1 Signal Connections

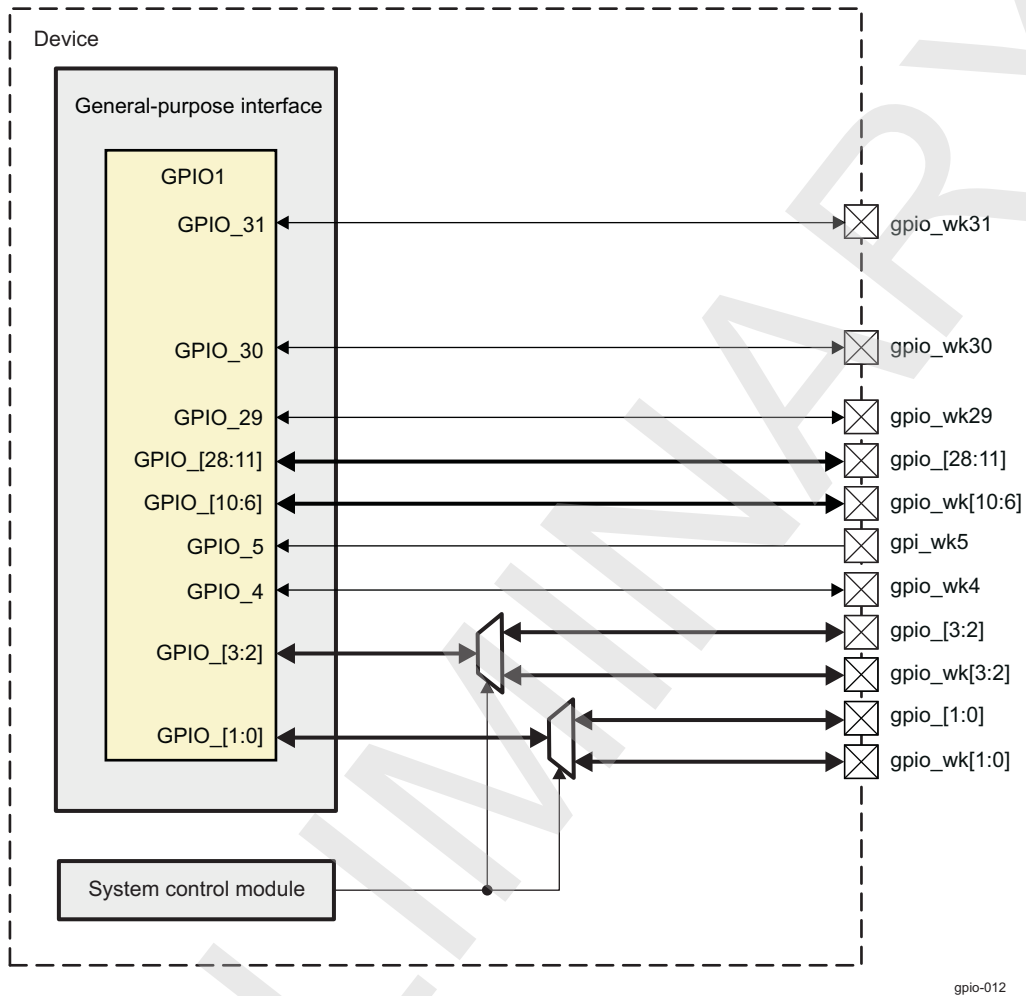
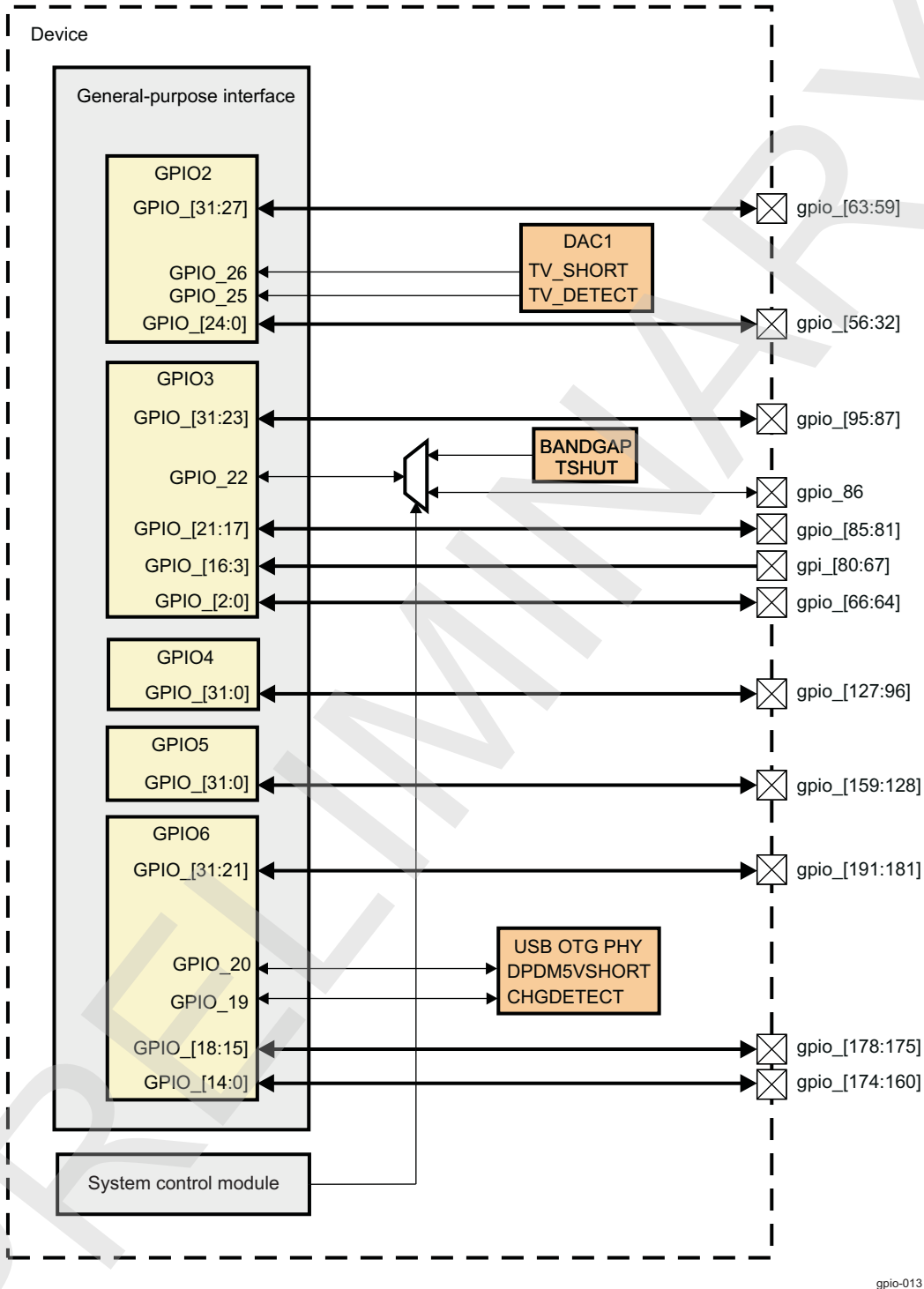


Figure 25-5 shows the signal connections of GPIO modules 2 through 6.

Figure 25-5. GPIO2 through GPIO6 Signal Connections



**NOTE:** For more information about the GPIO1 through GPIO6 signals and channel descriptions, see [Section 25.4.7, General-Purpose Interface Channels Description](#).

**NOTE:** For more information about GPIO signal multiplexing, see [Section 18.4.8, Pad Functional Multiplexing and Configuration](#), in [Chapter 18, Control Module](#).

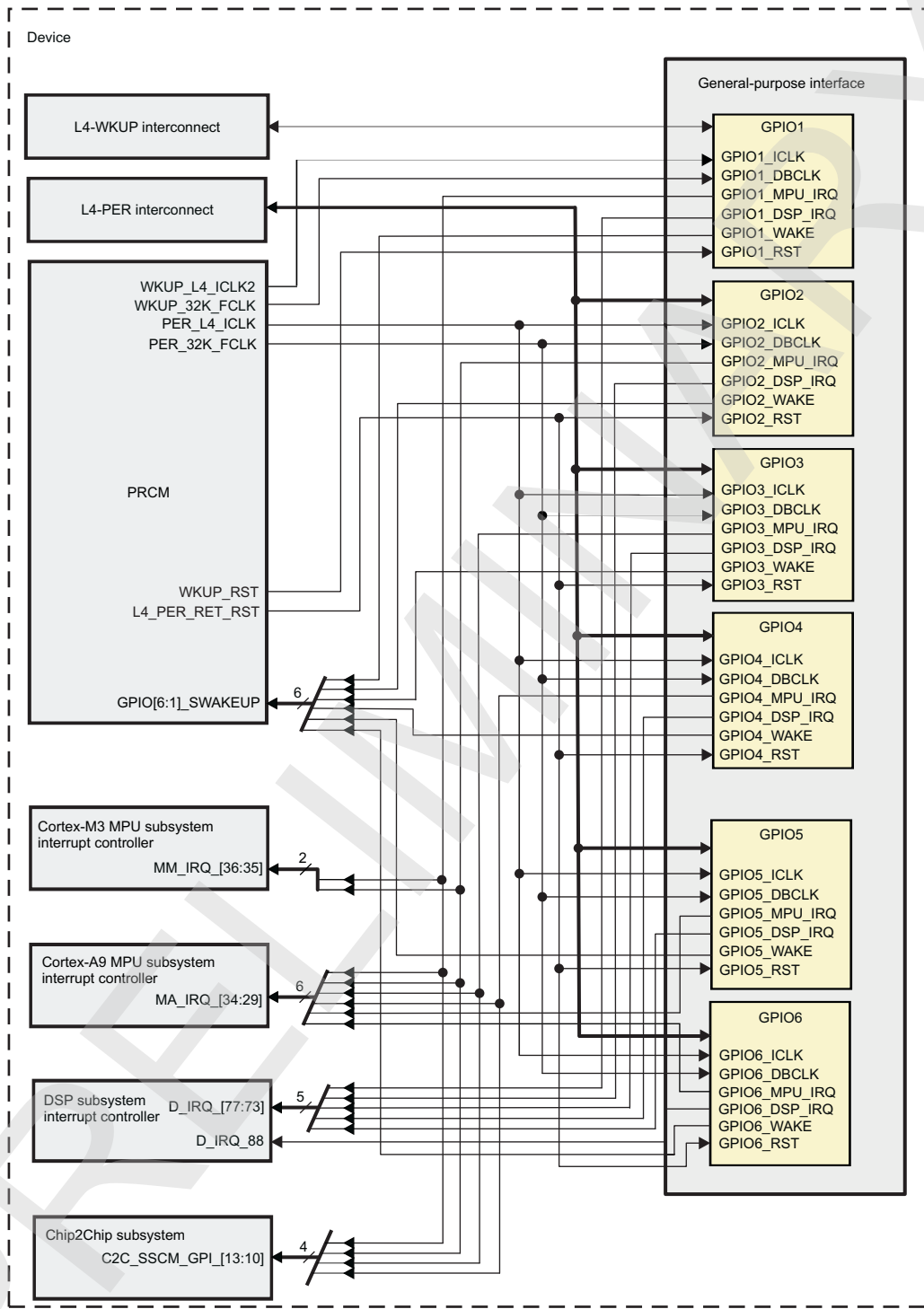
---

### 25.3 General-Purpose Interface Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

[Figure 25-6](#) shows this module integration.

Figure 25-6. GPIO Integration



gpio-004

**NOTE:** For more information about the IDLE hardware handshake and the wake-up request, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

Table 25-2 through Table 25-4 summarize the integration of the module in the device.



**Table 25-2. Integration Attributes**

Module Instance	Attributes	
	Power Domain	Interconnect
GPIO1	PD_WKUP	L4-WKUP
GPIOi, where i = 2 to 6	PD_L4_PER	L4-PER

**Table 25-3. Clocks and Resets**

Module Instance	Destination Signal Name	Clocks		
		Source Signal Name	Source	Description
GPIO1	GPIO1_ICLK	WKUP_L4_ICLK	PRCM	GPIO interface clock
GPIO1	GPIO1_DBCLK	WKUP_32K_FCLK	PRCM	GPIO debounce clock
GPIO2	GPIO2_ICLK	PER_L4_ICLK	PRCM	GPIO interface clock
GPIO2	GPIO2_DBCLK	PER_32K_FCLK	PRCM	GPIO debounce clock
GPIO3	GPIO3_ICLK	PER_L4_ICLK	PRCM	GPIO interface clock
GPIO3	GPIO3_DBCLK	PER_32K_FCLK	PRCM	GPIO debounce clock
GPIO4	GPIO4_ICLK	PER_L4_ICLK	PRCM	GPIO interface clock
GPIO4	GPIO4_DBCLK	PER_32K_FCLK	PRCM	GPIO debounce clock
GPIO5	GPIO5_ICLK	PER_L4_ICLK	PRCM	GPIO interface clock
GPIO5	GPIO5_DBCLK	PER_32K_FCLK	PRCM	GPIO debounce clock
GPIO6	GPIO6_ICLK	PER_L4_ICLK	PRCM	GPIO interface clock
GPIO6	GPIO6_DBCLK	PER_32K_FCLK	PRCM	GPIO debounce clock
<b>Resets</b>				
GPIO1	GPIO1_RST	WKUP_RST	PRCM	GPIO reset signal
GPIO2	GPIO2_RST	L4_PER_RET_RST	PRCM	GPIO reset signal
GPIO3	GPIO3_RST	L4_PER_RET_RST	PRCM	GPIO reset signal
GPIO4	GPIO4_RST	L4_PER_RET_RST	PRCM	GPIO reset signal
GPIO5	GPIO5_RST	L4_PER_RET_RST	PRCM	GPIO reset signal
GPIO6	GPIO6_RST	L4_PER_RET_RST	PRCM	GPIO reset signal

**Table 25-4. Hardware Requests**

Module Instance	Source Signal Name	Destination Signal Name	Interrupt Requests	
			Destination	Description
GPIO1	GPIO1_DSP_IRQ	D_IRQ_73	DSP	An interrupt to DSP (second interrupt)
GPIO1	GPIO1_MPU_IRQ	MA_IRQ_29	Cortex-A9	An interrupt to MPU subsystem INTC (first interrupt)
GPIO1	GPIO1_MPU_IRQ	MM_IRQ_35	Cortex-M3	An interrupt to MPU subsystem INTC (first interrupt)
GPIO1	GPIO1_MPU_IRQ	C2C_SSCM_GPI_10	C2C	An interrupt to distant device through C2C subsystem (first interrupt)
GPIO2	GPIO2_DSP_IRQ	D_IRQ_74	DSP	An interrupt to DSP (second interrupt)
GPIO2	GPIO2_MPU_IRQ	MA_IRQ_30	Cortex-A9	An interrupt to MPU subsystem INTC (first interrupt)
GPIO2	GPIO2_MPU_IRQ	MM_IRQ_36	Cortex-M3	An interrupt to MPU subsystem INTC (first interrupt)
GPIO2	GPIO2_MPU_IRQ	C2C_SSCM_GPI_11	C2C	An interrupt to distant device through C2C subsystem (first interrupt)
GPIO3	GPIO3_DSP_IRQ	D_IRQ_75	DSP	An interrupt to DSP (second interrupt)
GPIO3	GPIO3_MPU_IRQ	MA_IRQ_31	Cortex-A9	An interrupt to MPU subsystem INTC (first interrupt)
GPIO3	GPIO3_MPU_IRQ	C2C_SSCM_GPI_12	C2C	An interrupt to distant device through C2C subsystem (first interrupt)

**Table 25-4. Hardware Requests (continued)**

GPIO4	GPIO4_DSP_IRQ	D_IRQ_76	DSP	An interrupt to DSP (second interrupt)
GPIO4	GPIO4_MPU_IRQ	MA_IRQ_32	Cortex-A9	An interrupt to MPU subsystem INTC (first interrupt)
GPIO4	GPIO4_MPU_IRQ	C2C_SSCM_GPI_13	C2C	An interrupt to distant device through C2C subsystem (first interrupt)
GPIO5	GPIO5_DSP_IRQ	D_IRQ_77	DSP	An interrupt to DSP (second interrupt)
GPIO5	GPIO5_MPU_IRQ	MA_IRQ_33	Cortex-A9	An interrupt to MPU subsystem INTC (first interrupt)
GPIO6	GPIO6_DSP_IRQ	D_IRQ_88	DSP	An interrupt to DSP (second interrupt)
GPIO6	GPIO6_MPU_IRQ	MA_IRQ_34	Cortex-A9	An interrupt to MPU subsystem INTC (first interrupt)

**NOTE:** For a description of interrupt source, see [Section 25.4.6, Interrupt and Wake-Up Requests](#).

## 25.4 General-Purpose Interface Functional Description

### 25.4.1 General-Purpose Interface Block Diagram

Figure 25-7 shows the general-purpose interface.

Figure 25-7. General-Purpose Interface Description

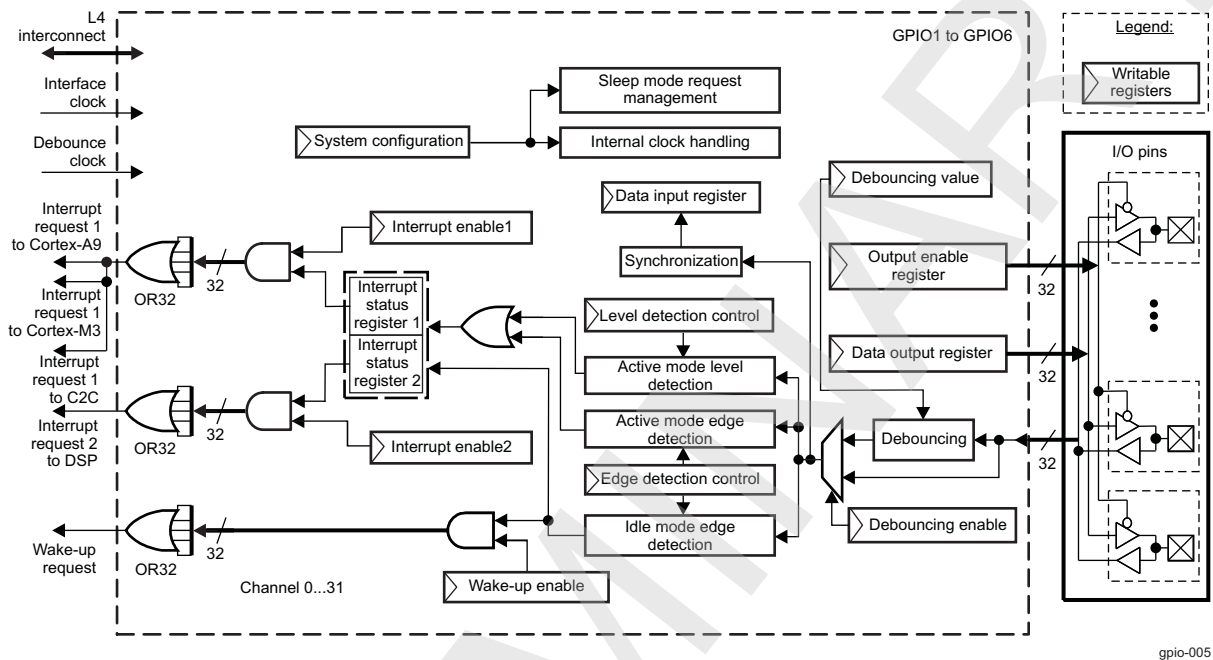
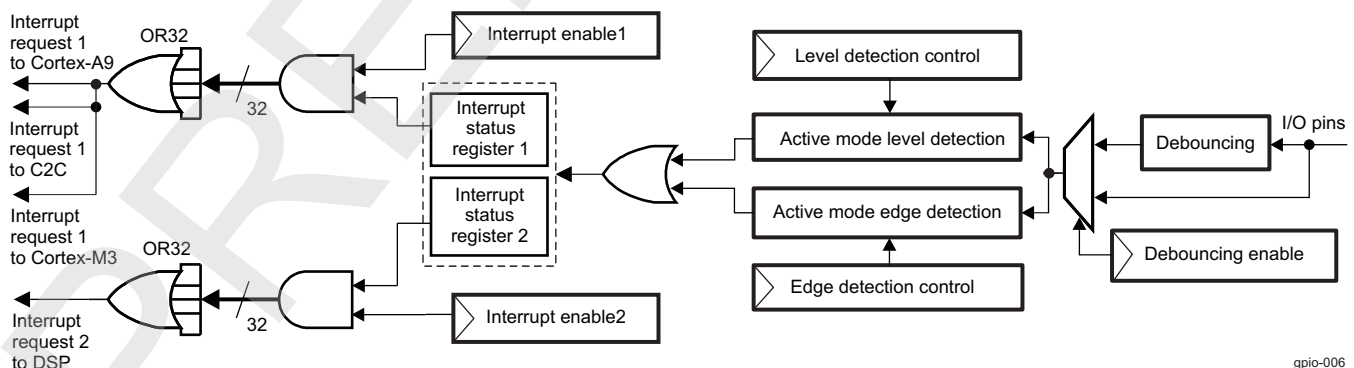


Figure 25-7 shows the details of the GPIO modules in the general-purpose interface block diagram, including their configuration registers and main functional paths:

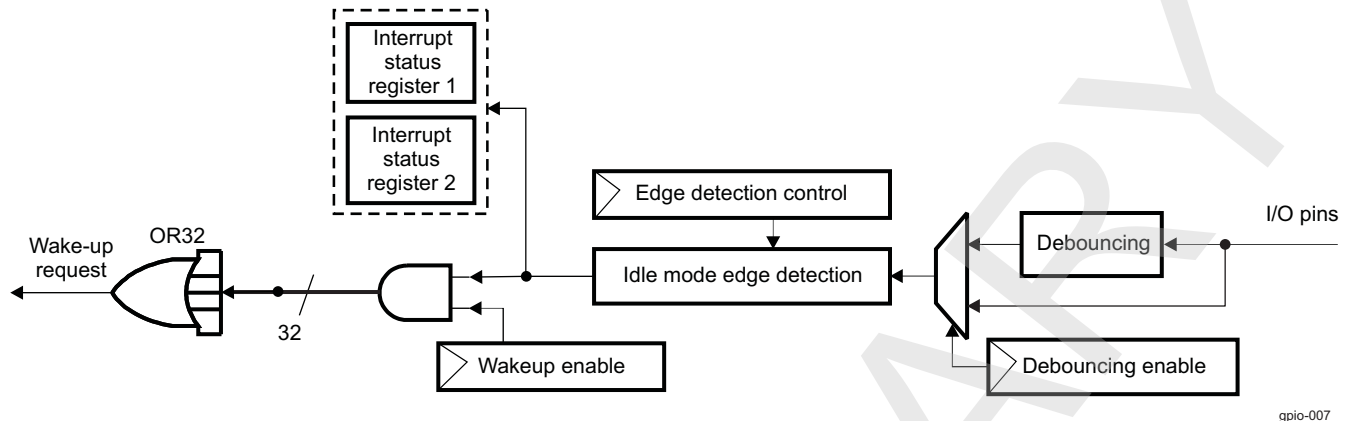
- The synchronous path (for active mode operation) used to generate a synchronous interrupt request on expected event detection on any input GPIO. Synchronous interrupt request lines 1 and 2 are active based on their respective interrupt enable 1 and 2 registers (GPIOi.GPIO\_IRQSTATUS\_SET\_0, GPIOi.GPIO\_IRQSTATUS\_SET\_1, GPIOi.GPIO\_IRQSTATUS\_CLR\_0, and GPIOi.GPIO\_IRQSTATUS\_CLR\_1). See Figure 25-8.

Figure 25-8. Synchronous Path



- The asynchronous path (for idle mode operation) used to generate an asynchronous wake-up request on the expected edge detection on any input GPIO. The asynchronous wake-up request line is active based on the wake-up-enable register. See Figure 25-9.

Figure 25-9. Asynchronous Path



- The blocks handling the internal clock (clock gating) and managing the sleep mode request/acknowledge protocol (enabling the synchronous path in active mode and the asynchronous path in idle mode)

## 25.4.2 General-Purpose Interface Interrupt and Wake-Up Features

### 25.4.2.1 Synchronous Path: Interrupt Request Generation

The general-purpose interface has 12 interrupt lines (two interrupt lines per GPIO module instance). The 12 interrupt signals are GPIOi\_MPU\_IRQ (used by the MPU and C2C subsystems) and GPIOi\_DSP\_IRQ (used by the DSP subsystem), where  $i = 1, 2, 3, 4, 5, \text{ or } 6$ .

Synchronous interrupt requests from each channel are processed by two identical interrupt generation submodules used independently by the DSP subsystem on one side and by the MPU and C2C subsystems on the other side. Each submodule controls its own synchronous interrupt request line and has its own interrupt enable (GPIOi.GPIO\_IRQSTATUS\_SET\_0, GPIOi.GPIO\_IRQSTATUS\_SET\_1, GPIOi.GPIO\_IRQSTATUS\_CLR\_0, or GPIOi.GPIO\_IRQSTATUS\_CLR\_1), and interrupt status (GPIOi.GPIO\_IRQSTATUS\_0 or GPIOi.GPIO\_IRQSTATUS\_1) registers. The interrupt-enable register selects the channel(s) considered for the interrupt request generation, and the interrupt status register determines which channel(s) activate the interrupt request. Event detection on GPIO channels is reflected in the interrupt status registers independent of the content of the interrupt-enable registers.

In active mode, when the GPIO configuration registers are set to enable the interrupt generation (see Section 25.4.6, *Interrupt and Wake-up Requests*), a synchronous path samples the transitions and levels on the input GPIO with the internally gated interface clock (see Section 25.4.5.2.4, *Module Power Saving*). When an event matches the programmed settings (see Section 25.4.6, *Interrupt and Wake-up Requests*), the corresponding bit in the interrupt status register is set to 1 and, on the following interface clock cycle, interrupt lines 1 and/or 2 are activated (depending on the interrupt-enable registers).

Because of the sampling operation, the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the internally gated interface clock period (the internally gated interface clock period equals  $N$  times the interface clock period (see Section 25.4.5.2.4, *Module Power Saving*)). This minimum pulse width must be met before and after any expected level transition detection. Level detection requires the selected level to be stable for at least two times the internally gated interface clock period to trigger a synchronous interrupt.

Because the module is synchronous, latency is minimal between the expected event occurrence and the activation of the interrupt line(s). This latency must not exceed four internally gated interface clock cycles + one interface clock cycle when the debounce feature is not used.

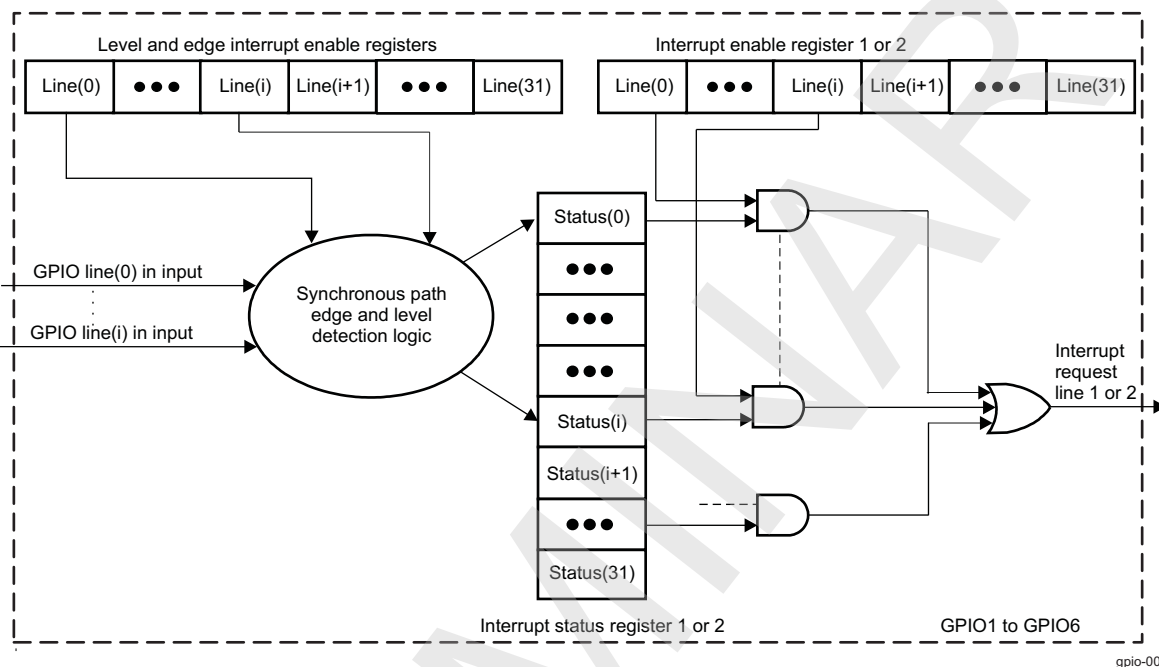
When the debounce feature is active, the latency depends on the value of the debouncing time register (GPIOi.GPIO\_DEBOUNCINGTIME) (see Section 25.4.3, *General-Purpose Interface Clock Configuration*) and is less than three internally gated interface clock cycles + two interface clock cycles + GPIOi.GPIO\_DEBOUNCINGTIME register value debounce clock cycles + three debounce clock cycles.

Synchronous interrupt request line 1 is mapped on the Cortex-A9 MPU, Cortex-M3 MPU and C2C subsystem.

Synchronous interrupt request line 2 is mapped on the DSP INTC.

Figure 25-10 is an overview of the interrupt request generation.

Figure 25-10. Interrupt Request Generation



### 25.4.2.2 Asynchronous Path: Wake-Up Request Generation

The general-purpose interface has six wake-up lines (one wake-up line per GPIO module instance) connected to the PRCM module.

Asynchronous wake-up requests from input channels are merged to issue one wake-up signal to the system per GPIO module. The wake-up-enable registers (GPIOi.GPIO\_IRQWAKEN\_0 and GPIOi.GPIO\_IRQWAKEN\_1) select the channel(s) considered for the wake-up request generation. The asynchronous wake-up request is reflected into the synchronous interrupt status registers (GPIOi.GPIO\_IRQSTATUS\_0 and GPIOi.GPIO\_IRQSTATUS\_1).

In idle mode (the interface clock is shut down and the GPIO configuration registers are programmed; see Section 25.4.6, *Interrupt and Wake-Up Requests*), an asynchronous path detects the expected transition(s) on a GPIO input (based on register programming) and activates an asynchronous wake-up request by the sideband signal (GPIOi\_SWAKEUP, where i = 1, 2, 3, 4, 5, or 6), if the wake-up-enable register is set.

When the system is awakened, the interface clock is restarted and synchronously set to 1 based on the input GPIO pin triggering the wake-up request and the corresponding bit in the interrupt status registers (GPIOi.GPIO\_IRQSTATUS\_0 and GPIOi.GPIO\_IRQSTATUS\_1). On the following internal clock cycle, interrupt lines 1 and/or 2 are active (active low) when the corresponding bits are set in the interrupt-enable registers (GPIOi.GPIO\_IRQSTATUS\_SET\_0, GPIOi.GPIO\_IRQSTATUS\_SET\_1, GPIOi.GPIO\_IRQSTATUS\_CLR\_0 and GPIOi.GPIO\_IRQSTATUS\_CLR\_1).

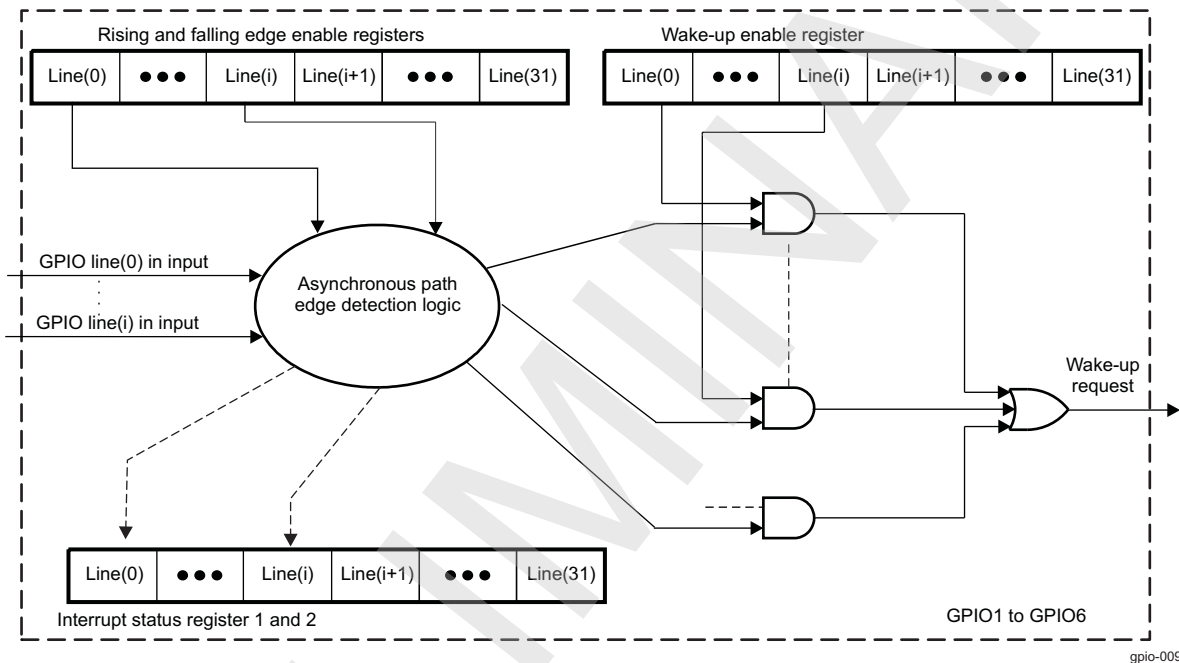
**NOTE:** When debouncing is not enabled, a minimum input pulse width does not trigger the wake-up request because there is no sampling operation.

When debouncing is enabled, the minimum pulse width is set by the specified debouncing time.

The GPIOi.GPIO\_SYSCONFIG[2] ENAWAKEUP bit enables or disables the GPIO wake-up feature globally. If the bit is 0, the wake-up-enable registers (GPIOi.GPIO\_IRQWAKEN\_0 and GPIOi.GPIO\_IRQWAKEN\_1) have no effect.

Figure 25-11 is an overview of the wake-up request generation.

**Figure 25-11. Wake-Up Request Generation**



### 25.4.2.3 Wake-Up Event Conditions During Transition To and From IDLE State

Figure 25-12 shows the wake-up event conditions.

In phase A, only the synchronous path is enabled. A synchronous interrupt request (see [Section 25.4.2.1, Synchronous Path: Interrupt Request Generation](#)) activates the interrupt line(s) and prevents the GPIO from transitioning into IDLE state until the interrupt is cleared.

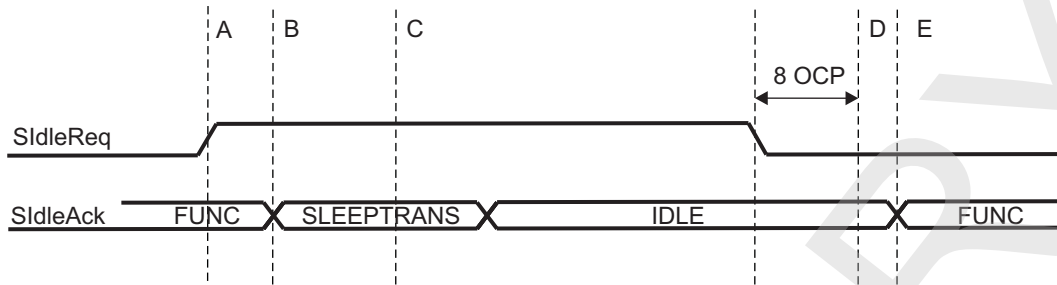
In phase B, the asynchronous path and synchronous path are enabled during the first five functional clock cycles of the SLEEPTRANS state. During this period a synchronous interrupt request (see [Section 25.4.2.1, Synchronous Path: Interrupt Request Generation](#)) prevents the GPIO from transitioning into IDLE state. A shorter pulse puts the module into IDLE state but triggers a wakeup once in IDLE.

In phase C, only the asynchronous path is enabled. A wake-up request (see [Section 25.4.2.2, Asynchronous Path: Wake-Up Request Generation](#)) triggers a wake-up request from the GPIO and when the module is awakened an interrupt is generated. If debouncing is not enabled, there is no minimum input pulse width to trigger the wake-up request.

In phase D, eight OCP clock cycles occur until the module is in FUNCT state, the synchronous path is enabled, and an event that fulfills the pulse width requirements (see [Section 25.4.2.1, Synchronous Path: Interrupt Request Generation](#)) activates the interrupt line(s).

In phase E, only the synchronous path is enabled. A synchronous interrupt request (see [Section 25.4.2.1, Synchronous Path: Interrupt Request Generation](#)) activates the interrupt line(s).

Figure 25-12. Wake-Up Event Conditions



gpio-014

#### 25.4.2.4 Interrupt (or Wake-Up) Line Release

When the host processor (the MPU and/or DSP subsystem in the device) receives an interrupt request issued by the GPIO module, it reads the corresponding interrupt status register (GPIOi.GPIO\_IRQSTATUS\_0 or GPIOi.GPIO\_IRQSTATUS\_1) to determine which GPIO input triggered the interrupt (or the wake-up request).

After servicing the interrupt (or acknowledging the wake-up request), the processor resets the status bit and releases the interrupt line by writing 1 in the corresponding bit of the interrupt status register. If there is still a pending interrupt request to serve (all bits in the interrupt status register that are not masked by the interrupt-enable register are not cleared), the interrupt line is reasserted.

---

**NOTE:** The status bit must be reset to re-enter idle mode.

---

### 25.4.3 General-Purpose Interface Clock Configuration

#### 25.4.3.1 Clocking

Each GPIO module uses two clocks:

- **Debounce clock:** The 32-kHz debounce clock, GPIOi\_DBCLK, (where i = 1, 2, 3, 4, 5, or 6, with one debounce clock per module), comes from the PRCM module and is used for the debounce cell logic (without the corresponding configuration registers). This cell can sample the input line and filters the input level using a programmed delay.

The debouncing value register (GPIOi.GPIO\_DEBOUNCINGTIME) is used to set the debouncing time for all input lines in the GPIO module. The value is global for all the ports of one GPIO module, so up to six different debouncing values are possible. The debounce cell is running with the debounce clock (32 kHz). This register represents the number of clock cycle(s) to be used (one cycle is 31 μs long).

The following formula describes the required input stable time to be propagated to the debounced output:

Required input line stable = (GPIOi.GPIO\_DEBOUNCINGTIME[7:0] DEBOUNCVL bit field value + 1) × 31,

where the value of the GPIOi.GPIO\_DEBOUNCINGTIME[7:0] DEBOUNCVL bit field is from 0 to 255.

For more information, see [Section 3.6.8, CD\\_L4\\_PER\\_Clock Domain](#), and [Section 3.6.4, CD\\_WKUP\\_Clock Domain](#), in [Chapter 3, Power, Reset, and Clock Management](#).

- **Interface clock:** The interface clock, GPIOi\_ICLK (where i = 1, 2, 3, 4, 5, or 6), comes from the PRCM module and is used throughout the GPIO module (except within the debounce cell logic). GPIOi\_ICLK clocks the data exchanges between the L4 interconnect and the internal logic. The clock-gating features allow module power consumption to be adapted to the activity.

For more information, see [Section 3.6.8, CD\\_L4\\_PER\\_Clock Domain](#), and [Section 3.6.4, CD\\_WKUP\\_Clock Domain](#), in [Chapter 3, Power, Reset, and Clock Management](#).

[Table 25-3](#) describes the GPIO module clocks.

[Table 25-5](#) summarizes the functional clock configuration.

**Table 25-5. Functional Clock Configuration**

Interface Clock	GPIO_CTRL[2:1] GATINGRATIO	Functional Clock
GPIOi_ICLK where i = 1 to 6	00	GPIOi_ICLK /1, where i = 1 to 6
GPIOi_ICLK where i = 1 to 6	01	GPIOi_ICLK /2, where i = 1 to 6
GPIOi_ICLK where i = 1 to 6	10	GPIOi_ICLK /4, where i = 1 to 6
GPIOi_ICLK where i = 1 to 6	11	GPIOi_ICLK /8, where i = 1 to 6

### 25.4.4 General-Purpose Interface Hardware and Software Reset

The GPIO can be reset by using the domain reset (hardware reset) or by setting a dedicated configuration bit (software reset) in each GPIO module.

- **Hardware reset:** The GPIO2 to GPIO6 modules are attached to the PER\_RST reset domain. The GPIO1 module is attached to the WKUP\_RST reset domain.

The hardware reset has a global reset action on the GPIO modules of the general-purpose interface. All configuration registers and internal logic are reset when it is active (low level). In each GPIO module, the GPIOi.GPIO\_SYSTATUS[0] RESETDONE bit monitors the internal reset status; it is set when the reset completes. For more information, see [Section 3.5.4, Reset Domains](#), in [Chapter 3, Power, Reset, and Clock Management](#).

- **Software reset:** Each GPIO module has its own software reset using the GPIOi.GPIO\_SYSCONFIG[1] SOFTRESET bit (where i = 1, 2, 3, 4, 5, or 6). The software reset has the same effect as the hardware reset signal, but this reset can be applied on one or more modules.

Writing 1 to the GPIOi.GPIO\_SYSCONFIG[1] SOFTRESET bit, (where i = 1, 2, 3, 4, 5, or 6) resets the module. A bit value of 1 remains until the reset is complete. When the software reset is complete, the GPIOi.GPIO\_SYSCONFIG[1] SOFTRESET bit is automatically reset to 0 and has the same effect as a hardware reset. The GPIOi.GPIO\_SYSTATUS[0] RESETDONE bit is cleared during a software reset.



This bit is set to 1 when the software reset is complete.

## **25.4.5 General-Purpose Interface Power Management**

### **25.4.5.1 Power Domain**

The GPIO1 module is attached to the WKUP power domain (see [Section 3.1.1.2, Power Management, in Chapter 3, Power, Reset, and Clock Management](#)). This domain is composed of the logic permanently supplied to manage domain power state transitions and detect wake-up events. The WKUP power domain is continuously active. Modules GPIO2 to GPIO6 are attached to the PER power domain (see [Section 3.1.1.2, Power Management, in Chapter 3, Power, Reset, and Clock Management](#)). The PER power domain is not active continuously.

### **25.4.5.2 Power Management**

#### **25.4.5.2.1 Idle Scheme**

To reduce dynamic consumption, an efficient idle scheme is based on the following:

- An efficient local autclock gating for each module
- The implementation of control sideband signals between the PRCM module and each module

This enhanced idle control allows clocks to be activated and deactivated safely without requiring complex software management.

The idle mode request, idle acknowledge, and wake-up request (GPIOi\_WAKEUP, where i = 1, 2, 3, 4, 5, or 6) are sideband signals between the PRCM module and the general-purpose interface (see [Section 25.4.6.2, Wake-Up Request Generation](#)).

#### **25.4.5.2.2 Operating Modes**

The following four operating modes are defined for the modules:

- **Active mode:** The module runs synchronously on the interface clock; interrupts can be generated based on the configuration and external signals.
- **Idle mode:** Power-saving mode with the module in a waiting state. The interface clock can be stopped, an interrupt cannot be generated, and a wake-up signal can be generated based on the configuration and external signals.  
If the debounce clock provided by the PRCM module is active, the debounce cell can sample and filter the input to generate a wake-up event. If the debounce clock is inactive, the debounce cell gates all input signals and thus cannot be used.
- **Inactive mode:** The module has no activity. The interface clock can be stopped, an interrupt cannot be generated, and the wake-up feature is inhibited.
- **Disabled mode:** The module is not used. The internal clock paths are gated, and an interrupt or wake-up request cannot be generated.

The idle and inactive modes are configured within the module and activated on request by the PRCM module (see [Section 3.1.1.2, Power Management in Chapter 3, Power, Reset, and Clock Management](#)) through sideband signals (see [Section 25.4.5.2.3, System Power Management and Wakeup](#)).

The disabled mode is set by software through a dedicated configuration bit, the GPIOi.GPIO\_CTRL[0] DISABLEMODULE bit (0: The module is enabled and clocks are not gated; 1: The module is disabled and clocks are gated). It unconditionally gates the internal clock paths that are not used for the L4 interconnect.

#### **25.4.5.2.3 System Power Management and Wakeup**

The PRCM module can require the GPIO modules to be idled for power-saving purposes.

The general-purpose interface has six identical idle mode request/acknowledge (handshake) mechanisms with the PRCM module (see [Section 25.4.6.2, Wake-Up Request Generation](#)): one per GPIO module. The general-purpose interface allows the GPIO modules to enter idle mode based on the GPIOi.GPIO\_SYSCONFIG[4:3] IDLEMODE bit field.

Idle acknowledge depends on the configuration and activity of each GPIO module:

- Smart-idle mode

When the GPIO module is configured in smart-idle mode, it checks for more activity (capture of the input GPIO pins in the GPIOi.GPIO\_DATAIN register is complete with no pending interrupt; all interrupt status bits are cleared), and there is no access to the GPIO.GPIO\_DEBOUNCINGTIME register, which is waiting to be synchronized.

Idle acknowledge is then asserted and the module enters into idle-mode. It waits for active system clock gating by the PRCM module (when all peripherals supplied by the same L4 interface clock domain are also ready for idle).

In idle mode (that is, when the PRCM module gates the interface clock), no interrupt occurs and the module is ready to issue a wake-up request.

When the expected transition occurs on an enabled GPIO input pin, the GPIO module exits from idle mode if the GPIOi.GPIO\_SYSCONFIG[2] ENAWAKEUP bit is set to 1 (wake-up capability enabled).

- Smart-idle wake-up mode

If the GPIOi.GPIO\_SYSCONFIG[4:3] IDLEMODE bit field selects smart-idle or smart-idle wake-up mode, the GPIO module evaluates its internal capability to switch off the interface clock. Once all internal activity ceases (the DATA INPUT REGISTER completed to capture the input GPIO pins, there is no pending interrupt, all interrupt status bits are cleared, and there is no write access to the GPIO\_DEBOUNCINGTIME register pending to be synchronized), the Idle acknowledge is asserted and the GPIO enters into idle mode, ready to issue a wake-up request when the expected transition occurs on an enabled GPIO input pin. This wake-up request is effectively sent only if the GPIOi.GPIO\_SYSCONFIG[2] ENAWAKEUP bit of the system configuration register enables the GPIO wake-up capability (see [Section 25.6.2.2, System Configuration Register](#)). When the system is awake, the idle request goes inactive, the idle acknowledge and wake-up request (if the GPIO triggered the systems wakeup) signals are immediately deasserted, and the asynchronous wake-up request (if existing) is reflected into the synchronous interrupt status registers.

- Force-idle mode

When the GPIO module is configured in force-idle mode (the GPIOi.GPIO\_SYSCONFIG[4:3] IDLEMODE bit field = 2b00) and receives an idle request from the PRCM module, the GPIO module waits unconditionally for active system clock gating by the PRCM module. (This occurs only when all peripherals supplied by the same L4 interface clock domain are also ready for idle.)

When in idle mode (that is, when the PRCM module gates the interface clock), the module (in inactive mode) has no activity, the interface clock paths are gated, an interrupt cannot be generated, and the wake-up feature is totally inhibited.

- No-idle mode

When the GPIO module is configured in no-idle mode (the GPIOi.GPIO\_SYSCONFIG[4:3] IDLEMODE bit field = 2b01) and receives an idle request from the PRCM module, the GPIO module does not go into idle mode and the idle acknowledge is never sent.

---

**NOTE:** For more information about idle modes, see [Section 3.1.1.2, Power Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

---

#### 25.4.5.2.4 Module Power Saving

The GPIO module has local power management by internal clock-gating features:

- Internal interface clock gating: The clock for the L4 interconnect logic can be gated when the module is not accessed, if the GPIOi.GPIO\_SYSCONFIG[0] AUTOIDLE bit is set. Otherwise, this logic is free-running on the interface clock.
- Clock gating for the input data sample logic: Clock for the input data sample logic can be gated when the data in the GPIOi.GPIO\_DATAIN register is not accessed.
- Clock gating for the event detection logic: Each GPIO module implements four clock groups used for

the logic in the synchronous event detection. Each group of eight input GPIO pins has a separate enable signal depending on the edge/level detection register setting (because the input is 32 bits, four groups of eight inputs are defined for each GPIO module). If a group requires no detection, the corresponding clock is gated.

Grouping the edge/level enable can save the power consumption of the module.

If any of the following registers is set to 0x01 01 01 01, then all clocks are active (power consumption is high). If they are set to 0x00 00 00 FF, then a single clock is active (power saving).

- GPIOi.GPIO\_LEVELDETECT0
- GPIOi.GPIO\_LEVELDETECT1
- GPIOi.GPIO\_RISINGDETECT
- GPIOi.GPIO\_FALLINGDETECT

**NOTE:** When the clocks are enabled by writing to the following registers, the detection starts after five clock cycles. This period is required to clean the synchronization edge/level detection pipeline.

- GPIOi.GPIO\_LEVELDETECT0
- GPIOi.GPIO\_LEVELDETECT1
- GPIOi.GPIO\_RISINGDETECT
- GPIOi.GPIO\_FALLINGDETECT

The mechanism is independent of each clock group. If the clock has been started before and a new setting is performed, the following is recommended:

1. Set the new detection required.
2. Disable the previous setting (if necessary).

In this way, the corresponding clock is not gated and the detection starts immediately.

All channels are also gated using a one-out-of-N scheme. N is the GPIOi.GPIO\_CTRL[2:1] GATINGRATIO bit field and can take the values 1 (2b00), 2 (2b01), 4 (2b10), or 8 (2b11). The interface clock is enabled for this logic one cycle every N cycles. When N is equal to 1, there is no gating and this logic is free-running on the interface clock. When N is 2, 4, or 8, this logic is running at the equivalent frequency of interface clock frequency divided by N.

- Inactive mode: In inactive mode, all internal clock paths are gated.
- Disabled mode: All internal clock paths not used for the L4 interconnect are gated. The GPIOi.GPIO\_CTRL[0] DISABLEMODULE bit controls a clock-gating feature at the module level. When set to 1, this bit forces clock gating for all internal clock paths. Internal activity in the module is suspended. The L4 interconnect is not affected by this bit.

The interface clock gating is controlled with the GPIOi.GPIO\_SYSCONFIG[0] AUTOIDLE bit, which is used to save power when the module is not used because of the multiplexing configuration selected at the chip level. This bit has precedence over all other internal configuration bits.

Table 25-6 describes the power-management features available for the general-purpose interface module.

**NOTE:** For information about source clock gating and a description of sleep/wake-up transitions, see Section 3.1.1.1.2, *Module Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

For descriptions of the EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see Section 3.1.1.1.2, *Module Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

**Table 25-6. Local Power-Management Features**

Feature	Registers	Description
Clock autogating	GPIOi.GPIO_SYSCONFIG[0] AUTOIDLE bit	Sets the clock gating strategy for the OCP interface block

**Table 25-6. Local Power-Management Features (continued)**

Feature	Registers	Description
Slave idle modes	GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE bit field	Force-idle, no-idle, and smart-idle wake-up-capable modes are available.
Clock activity	GPIOi.GPIO_CTRL[0] DISABLEMODULE bit field	Enable and disable the module.
Debouncing enable	GPIOi.GPIO_DEBOUNCENABLE[31:0] DEBOUNCEENABLE bit field	Debouncing mode is available.
Global wake-up enable	GPIOi.GPIO_SYSCONFIG[2] ENAWAKEUP bit	This bit enables the wake-up feature at module level.
Wake-up sources enable	GPIOi.GPIO_IRQWAKEN_0[31:0] INTLINE register GPIOi.GPIO_IRQWAKEN_1[31:0] INTLINE register	This register enables or disables a specific IRQ request source to generate a wake-up signal.

Table 25-7 describes the clock activity settings.

**Table 25-7. Clock Activity Settings**

GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE	Selected Mode	Description	Wake-Up Events
00	Force-idle	The GPIO module goes into inactive mode independently of the internal module state, and the Idle acknowledge is never sent.	No
01	No-idle	The GPIO module does not go into the idle mode and the idle acknowledge is never sent.	No
10	Smart-idle	The GPIO module evaluates its internal capability to switch off the interface clock. If there is no more internal activity, the idle acknowledge is asserted and the GPIO enters into idle mode.	No
11	Smart-idle wake-up	The GPIO module evaluates its internal capability to have the interface clock switched off. If there is no more internal activity, the idle acknowledge is asserted and the GPIO enters into idle mode.	Yes

## 25.4.6 General-Purpose Interface Interrupt and Wake-Up Requests

### 25.4.6.1 Interrupt Request Generation

All interrupt sources (the 32 GPIO input channels) are merged to issue two synchronous interrupt requests in each GPIO module. Thus, the general-purpose interface has 12 interrupt lines (two interrupt lines per GPIO module instance).

- Synchronous interrupt request line 1 is mapped on the Cortex-A9 and Cortex-M3 MPU INTCs, and on the Chip2Chip subsystem.
- Synchronous interrupt request line 2 is mapped on the DSP INTC.

Table 25-8 lists the event flags, and their mask, that can cause module interrupts.

**Table 25-8. Events**

Event Flag	Event Mask	Synchronous	Sensitivity	Map to	Description
GPIOi.GPIO_IRQSTATUS1[31:0] INTLINE	GPIOi.GPIO_IRQENABLE1[31:0] IRQENABLE1	Yes	Edge/level	MA_IRQ_n MM_IRQ_n <sup>(1)</sup> C2C_SSCM_GPI_n <sup>(2)</sup>	Corresponds to the first line of interrupt
GPIOi.GPIO_IRQSTATUS2[31:0] INTLINE	GPIOi.GPIO_IRQENABLE2[31:0] IRQENABLE2	Yes	Edge/level	D_IRQ_n	Corresponds to the second line of interrupt
GPIOi.GPIO_IRQSTATUS_0[31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_S_SET_0[31:0] INTLINE	Yes	Edge/level	MA_IRQ_n MM_IRQ_n <sup>(1)</sup> C2C_SSCM_GPI_n <sup>(2)</sup>	Corresponds to the first line of interrupt

<sup>(1)</sup> For GPIO1 and GPIO2 only

<sup>(2)</sup> For GPIO1 to GPIO4 only

**Table 25-8. Events (continued)**

Event Flag	Event Mask	Synchronous	Sensitivity	Map to	Description
GPIOi.GPIO_IRQSTATUS_1[31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_SET_1[31:0] INTLINE	Yes	Edge/level	D_IRQ_n	Corresponds to the second line of interrupt
GPIOi.GPIO_IRQSTATUS_0 [31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_CLR_0[31:0] INTLINE	Yes	Edge/level	MA_IRQ_n MM_IRQ_n <sup>(1)</sup> C2C_SSCM_GPI_n <sup>(2)</sup>	Corresponds to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1[31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_CLR_1[31:0] INTLINE	Yes	Edge/level	D_IRQ_n	Corresponds to the second line of interrupt
GPIOi.GPIO_IRQSTATUS_0[31:0] INTLINE	GPIOi.GPIO_IRQWAKEN_0[31:0] INTLINE	No	Edge/level	MA_IRQ_n MM_IRQ_n <sup>(1)</sup> C2C_SSCM_GPI_n <sup>(2)</sup>	Corresponds to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1[31:0] INTLINE	GPIOi.GPIO_IRQWAKEN_1[31:0] INTLINE	No	Edge/level	D_IRQ_n	Corresponds to the second line of interrupt

Synchronous interrupt request lines 1 and 2 are active depending on their respective interrupt enable 1 and 2 registers (GPIOi.GPIO\_IRQSTATUS\_SET\_0, GPIOi.GPIO\_IRQSTATUS\_SET\_1, GPIOi.GPIO\_IRQSTATUS\_CLR\_0 and GPIOi.GPIO\_IRQSTATUS\_CLR\_1).

- Interrupt enable registers (GPIOi.GPIO\_IRQSTATUS\_SET\_0 and GPIOi.GPIO\_IRQSTATUS\_SET\_1)  
The interrupt enable1 (or interrupt enable2) register allows masking of the expected transition on input GPIO to prevent the generation of an interrupt request on line1 (or line2). The interrupt-enable registers are programmed synchronously with the interface clock.  
These registers can be accessed with direct read/write operations or using the alternate set and clear protocol feature for register update. This feature allows to set or clear explicit bits of these registers with a single write access (see Section 25.4.9, *General-Purpose Interface Set-and-Clear Protocol*).
- Interrupt status registers (GPIOi.GPIO\_IRQSTATUS\_0 and GPIOi.GPIO\_IRQSTATUS\_1)  
The interrupt status1 (or interrupt status2) register determines which of the input GPIO pins triggered the interrupt line1 (or interrupt line2) request (or the wake-up line).  
When a bit in this register is set to 1, it indicates that the corresponding GPIO pin is requesting the interrupt (or the wake-up). To reset a bit in this register, write 1 to the appropriate bit. However, an interrupt cannot be generated by writing 1 to the interrupt status1 (or interrupt status2) register.  
If 0 is written to a bit in this register, the value remains unchanged. The interrupt status1 (or interrupt status2) register is synchronous with the interface clock. In idle mode, the event is detected through an asynchronous path, and the corresponding bit in the interrupt status1 and interrupt status2 registers are set when the GPIO module is awoken.

**CAUTION**

After servicing the interrupt, the status bit in the interrupt status register (GPIOi.GPIO\_IRQSTATUS\_0 or GPIOi.GPIO\_IRQSTATUS\_1) must be reset and the interrupt line released (by writing 1 in the corresponding bit of the interrupt status register) before enabling an interrupt for the GPIO channel in the interrupt-enable register (GPIOi.GPIO\_IRQSTATUS\_SET\_0 or GPIOi.GPIO\_IRQSTATUS\_SET\_1) to prevent the occurrence of unexpected interrupts when enabling an interrupt for the GPIO channel.

**25.4.6.2 Wake-Up Request Generation**

The GPIO1 module of the general-purpose interface is attached to the WKUP power domain (see Section 3.1.1.2, *Power Management*, in Chapter 3, *Power, Reset, and Clock Management*, and can wake up the system.

**NOTE:** The GPIO2 to GPIO6 modules belong to the PER power domain and thus their wake-up capabilities are operational only when the PER power domain is active.

All wake-up sources (the 32 input GPIO channels) are merged together to issue a single asynchronous wake-up request in each GPIO module following the expected transition(s) (based on register programming). Each GPIO module generates a wake-up signal to the PRCM module.

**NOTE:** Only gpio\_wk[10:0] and gpio\_wk[31:29] can be used to generate a direct wake-up event.

The asynchronous wake-up request line is active based on the GPIOi.GPIO\_IRQWAKEN\_0 and GPIO\_IRQWAKEN\_1 wake-up-enable registers (where i = 1, 2, 3, 4, 5, or 6).

The wake-up-enable register allows masking of the expected transition on input GPIO to prevent the generation of a wake-up request. The wake-up-enable register is programmed synchronously with the interface clock before any idle mode request coming from the host processor.

This register can be accessed with direct read/write operations or by using the alternate set and clear protocol register update feature. This feature allows setting or clearing specific bits of this register with a single write access to the set wake-up-enable register (or to the clear wake-up-enable register) address (see Section 25.4.9, *General-Purpose Interface Set-and-Clear Protocol*).

**NOTE:** There must be a correlation between the wake-up enable and interrupt-enable registers. If a GPIO pin has a wake-up configured on it, it must also have the corresponding interrupt enabled (on one of the two interrupt lines). Otherwise, it is possible to have a wake-up event, but after exiting the IDLE state, no interrupt is generated; thus, the corresponding bit from the interrupt status register is not cleared, and the module does not acknowledge a future idle request.

Table 25-9 lists the mapping of the wake-up signals.

**Table 25-9. Wake-Up Signals**

Name	Mapping	Comments
GPIOi_WAKE	GPIOi_SWAKEUP	Where i = 1, 2, 3, 4, 5, and 6. Destination is the PRCM module.

### 25.4.7 General-Purpose Interface Channel Description

Table 25-10 describes the GPIO channels.

**Table 25-10. GPIO Channel Description**

Channel Number	Type <sup>(1)</sup>	Mapping	Wake-Up Feature	Comments
<b>GPIO1 Module</b>				
[31]	I/O	gpio_wk31	Yes	Wake-up path
[30]	I/O	gpio_wk30	Yes	Wake-up path
[29]	I/O	gpio_wk29	Yes	Wake-up path
[28:11]	I/O	gpio_[28:11]	Yes	GPIO
[10:6]	I/O	gpio_wk[10:6]	Yes	Wake-up path
[5]	I	gpi_wk5	Yes	Wake-up path
[4]	I/O	gpio_wk4	Yes	Wake-up path
[3:2]	I/O	gpio_[3:2]	Yes	GPIO
	I/O	gpio_wk[3:2]	Yes	Wake-up path
[1:0]	I/O	gpio_[1:0]	Yes	GPIO
	I/O	gpio_wk[1:0]	Yes	Wake-up path
<b>GPIO2 Module</b>				

<sup>(1)</sup> I = Input; O = Output

**Table 25-10. GPIO Channel Description (continued)**

Channel Number	Type <sup>(1)</sup>	Mapping	Wake-Up Feature	Comments
[31:27]	I/O	gpio_[63:59]	Yes <sup>(2)</sup>	GPIO
[26] <sup>(3)</sup>	I	TV_SHORT	Yes <sup>(2)</sup>	Internal TV short signal from video DAC1
[25] <sup>(3)</sup>	I	TV_DETECT	Yes <sup>(2)</sup>	Internal TV detection signal from video DAC1
[24:0]	I/O	gpio_[56:32]	Yes <sup>(2)</sup>	GPIO
<b>GPIO3 Module</b>				
[31:17]	I/O	gpio_[95:81]	Yes <sup>(2)</sup>	GPIO
[22] <sup>(4)</sup>	I/O	gpio_86	Yes <sup>(2)</sup>	GPIO
	I	TSHUT	Yes <sup>(2)</sup>	Bandgap internal TSHUT
[21:17]	I/O	gpio_[85:81]	Yes <sup>(2)</sup>	GPIO
[16:3]	I	gpi_[80:67]	Yes <sup>(2)</sup>	GPIO inputs only
[2:0]	I/O	gpio_[66:64]	Yes <sup>(2)</sup>	GPIO
<b>GPIO4 Module</b>				
[31:0]	I/O	gpio_[127:96]	Yes <sup>(5)</sup>	GPIO
<b>GPIO5 Module</b>				
[31:0]	I/O	gpio_[159:128]	Yes <sup>(5)</sup>	GPIO
<b>GPIO6 Module</b>				
[31:21]	I/O	gpio_[191:181]	Yes <sup>(5)</sup>	GPIO
[20] <sup>(6)</sup>	I	DPDM5VSHORT	Yes <sup>(5)</sup>	USB OTG PHY short detect (DPDM5VSHORT)
[19] <sup>(6)</sup>	I	CHGDETECT	Yes <sup>(5)</sup>	USB OTG PHY charge detect (CHGDETECT)
[18:15]	I/O	gpio_[178:175]	Yes <sup>(5)</sup>	GPIO
[14:0]	I/O	gpio_[174:160]	Yes <sup>(5)</sup>	GPIO

<sup>(2)</sup> Only when the PER power domain is active

<sup>(3)</sup> The GPIO channel is not available on device pads. Reserved for internal use only (see the Comments column).

<sup>(4)</sup> The GPIO channel provides additional internal functionality (see the Comments column), besides its GPIO capabilities, when it is not connected to a pad of the device. That is, the corresponding pad is configured in a mode different from multiplexing Mode 3. For more information about pad configuration and multiplexing, see [Section 18.4.8, Pad Functional Multiplexing and Configuration](#), in [Chapter 18, Control Module](#).

<sup>(5)</sup> Only when the PER power domain is active

<sup>(6)</sup> The GPIO channel is not available on device pads. Reserved for internal use only (see the Comments column).

**NOTE:** The thermal shutdown comparator output signal (TSHUT) is an output from the BANDGAP module. This signal is low during normal operation and goes high during a thermal shutdown event. When channel 22 of the GPIO3 is not connected to a ball of the device (the corresponding pin is configured in a mode different from configuration mode 4; see [Section 18.4.8, Pad Functional Multiplexing and Configuration](#), in [Chapter 18, Control Module](#), for more information about pin configuration), TSHUT is connected to channel 22 of the GPIO3. An interrupt can be generated when a low-to-high transition occurs on TSHUT whether or not the interrupt generation for channel 22 of the GPIO3 is correctly configured.

**NOTE:** Channels 20 and 19 of GPIO6 are internally connected inside the device to the short detection (DPDM5VSHORT) and charger detection (CHGDETECT) outputs, respectively, of the USB OTG PHY module. This allows interrupt generation on the corresponding GPIO channel, if either of the mentioned events occurs.

### 25.4.8 General-Purpose Interface Data Input/Output Capabilities

The output-enable register (GPIOi.GPIO\_OE) controls the output/input capability of each pin. At reset, all the GPIO-related pins are configured as inputs, and their output capabilities are disabled. This register is not used within the module. Its only function is to carry the pad configuration.

When configured as an output (the desired bit reset in the GPIOi.GPIO\_OE register), the value of the corresponding bit in the GPIOi.GPIO\_DATAOUT register is driven on the corresponding GPIO pin. Data is written to the data-output register synchronously with the interface clock. This register can be accessed with read/write operations or by using the alternate set and clear protocol register update feature. This feature gives the possibility to set or clear specific bits of this register with a single write access to the set-output data register (GPIOi.GPIO\_SETDATAOUT) or to the clear-output data register (GPIOi.GPIO\_CLEARDATAOUT) address (see Section 25.4.9, *General-Purpose Interface Set-and-Clear Protocol*). If the application uses a pin as an output and does not want interrupt/wake-up generation from this pin, the application must properly configure the wake-up enable (GPIOi.GPIO\_WAKEUPENABLE) and the interrupt enable (GPIOi.GPIO\_IRQSTATUS\_SET\_0 and GPIOi.GPIO\_IRQSTATUS\_SET\_1) registers.

When configured as an input (the desired bit is set to 1 in the GPIOi.GPIO\_OE register), the state of the input can be read from the corresponding bit in the GPIOi.GPIO\_DATAIN register. The input data is sampled synchronously with the interface clock and then captured in the data input register synchronously with the interface clock. When the GPIO pin levels change, they are captured into this register after two interface clock cycles (the required cycles to synchronize and to write data). If the application uses a pin as an input, the application must properly configure the wake-up enable (GPIOi.GPIO\_IRQWAKEN\_0 and GPIOi.GPIO\_IRQWAKEN\_1) and the interrupt enable (GPIOi.GPIO\_IRQSTATUS\_SET\_0 and GPIOi.GPIO\_IRQSTATUS\_SET\_0) registers to the interrupt and wake-up feature as needed. For using the alternate set and clear protocol, see Section 25.4.9, *General-Purpose Interface Set-and-Clear Protocol*.

### 25.4.9 General-Purpose Interface Set-and-Clear Protocol

#### 25.4.9.1 Description

The GPIO module implements the set-and-clear protocol register update for the following registers:

- GPIOi.GPIO\_DATAOUT
- GPIOi.GPIO\_IRQSTATUS\_CLR\_0
- GPIOi.GPIO\_IRQSTATUS\_CLR\_1
- GPIOi.GPIO\_IRQSTATUS\_SET\_0
- GPIOi.GPIO\_IRQSTATUS\_SET\_1
- GPIOi.GPIO\_IRQWAKEN\_0
- GPIOi.GPIO\_IRQWAKEN\_1

This protocol is an alternative to the atomic test and set operations and consists of writing operations at dedicated addresses (one address for setting bit[s] and one address for clearing bit[s]). The data to write is 1 at bit position(s) to clear (or to set) and 0 at unaffected bit(s). Registers can be accessed in two ways:

- Standard: Full register read and write operations at the primary register address
- Set and clear: Separate addresses are provided to set (and clear) bits in registers. Writing 1 at these addresses sets (or clears) the corresponding bit into the equivalent register; writing a 0 has no effect.

Therefore, for these registers, three addresses are defined for one unique physical register. Reading these addresses has the same effect and returns the register value.



### 25.4.9.2 Clear Instruction

#### 25.4.9.2.1 Clear Register Addresses

- Clear interrupt-enable registers (GPIOi.GPIO\_IRQSTATUS\_CLR\_0 and GPIOi.GPIO\_IRQSTATUS\_CLR\_1)  
A write operation in the GPIO\_IRQSTATUS\_CLR\_0 (or GPIO\_IRQSTATUS\_CLR\_1) register clears the corresponding bit in the same register when the written bit is 1; a written bit at 0 has no effect.  
A read of the clear interrupt enable0 (or enable1) register returns the value of the GPIO\_IRQSTATUS\_CLR\_0 (or GPIO\_IRQSTATUS\_CLR\_1) register
- Clear wake-up-enable registers (GPIOi.GPIO\_IRQWAKEN\_0 and GPIOi.GPIO\_IRQWAKEN\_1)  
A write operation in the clear wake-up-enable register clears the corresponding bit in the wake-up-enable register when the written bit is 1; a written bit at 0 has no effect.  
A read of the clear wake-up-enable register returns the value of the wake-up-enable register.
- Clear data-output register (GPIOi.GPIO\_CLEARDATAOUT)  
A write operation in the clear data-output register clears the corresponding bit in the data-output register when the written bit is 1; a written bit at 0 has no effect.  
A read of the clear data-output register returns the value of the data-output register.

#### 25.4.9.2.2 Clear Instruction Example

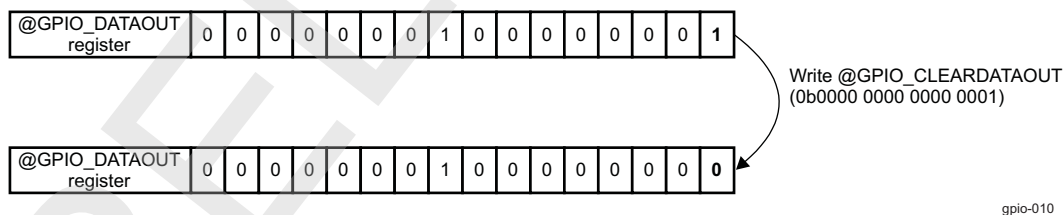
Assume the data-output register (or one of the interrupt/wake-up-enable register) contains the binary value, 0b0000 0001 0000 0001, and bit 0 is to be clear.

With the clear instruction feature, write 0b0000 0000 0000 0001 at the address of the clear data-output register (or at the address of the clear interrupt/wake-up-enable register). After this write operation, a reading of the data-output register (or the interrupt/wake-up-enable register) returns 0b0000 0001 0000 0000; bit 0 is cleared.

**NOTE:** Although the general-purpose interface registers are 32 bits wide, only the least-significant 16 bits are represented in this example.

Figure 25-13 shows an example of a clear instruction.

Figure 25-13. GPIO\_CLEARDATAOUT Register Example



### 25.4.9.3 Set Instruction

#### 25.4.9.3.1 Set Register Addresses

- Set interrupt-enable registers (GPIOi.GPIO\_IRQSTATUS\_SET\_0 and GPIOi.GPIO\_IRQSTATUS\_SET\_1)  
A write operation in the GPIOi.GPIO\_IRQSTATUS\_SET\_0 (or GPIOi.GPIO\_IRQSTATUS\_SET\_1) register sets the corresponding bit in the same register when the written bit is 1; a written bit at 0 has no effect.  
A read of the set interrupt enable0 (or enable1) register returns the value of the interrupt GPIOi.GPIO\_IRQSTATUS\_SET\_0 (or GPIOi.GPIO\_IRQSTATUS\_SET\_1) register.
- Set wake-up-enable registers (GPIOi.GPIO\_IRQWAKEN\_0 and GPIOi.GPIO\_IRQWAKEN\_1)  
A write operation in the set wake-up-enable register sets the corresponding bit in the wake-up-enable register when the written bit is 1; a written bit at 0 has no effect.

A read of the set wake-up-enable register returns the value of the wake-up-enable register.

- Set data-output register (GPIOi.GPIO\_SETDATAOUT)

A write operation in the set data-output register sets the corresponding bit in the data-output register when the written bit is 1; a written bit at 0 has no effect.

A read of the set data-output register returns the value of the data-output register.

### 25.4.9.3.2 Set Instruction Example

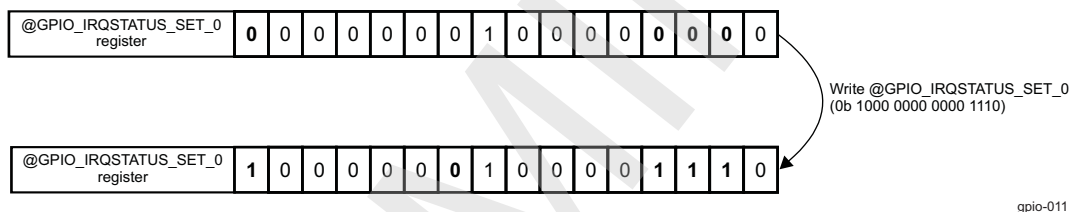
Assume the interrupt enable1 (or enable2) register (or the data-output register) contains the binary value, 0b0000 0001 0000 0000, and bits 15, 3, 2, and 1 are to be set.

With the set instruction feature, write 0b1000 0000 0000 1110 at the address of the set interrupt enable1 (or enable2) register (or at the address of the set data-output register). After this write operation, a reading of the interrupt enable1 (or enable2) register (or the data-output register) returns 0b1000 0001 0000 1110; bits 15, 3, 2, and 1 are set.

**NOTE:** Although the general-purpose interface registers are 32 bits wide, only the less-significant 16 bits are represented in this example.

Figure 25-14 shows an example of a set instruction.

**Figure 25-14. Write in GPIO\_IRQSTATUS\_SET\_0 Register Example**



gpio-011

The set wake-up-enable register offers the same feature with the wake-up-enable register.

## 25.5 General-Purpose Interface Programming Guide

### 25.5.1 General-Purpose Interface Low-Level Programming Models

#### 25.5.1.1 Global Initialization

##### 25.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the general-purpose interface module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the general-purpose interface. For more information, see [Section 25.3, General-Purpose Interface Integration](#), and [Section 25.2, General-Purpose Interface Environment](#).

**Table 25-11. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. For more information about the module configuration, see <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
Control module	Module specific pad muxing must be set in the control module. For more information about the module configuration, see <a href="#">Chapter 18, Control Module</a> .
MPU INTC	MPU INTC configuration must be done to enable the interrupts from the general-purpose interface module. See <a href="#">Chapter 17, Interrupt Controllers</a> .
DSP INTC	DSP INTC configuration must be done to enable the interrupts from the general-purpose interface module. See <a href="#">Chapter 17, Interrupt Controllers</a> .

##### 25.5.1.1.2 General-Purpose Interface Module Global Initialization

This procedure initializes the general-purpose interface module after a POR or software reset.

**Table 25-12. General-Purpose Interface Global Initialization**

Step	Register /Bit Field / Programming Model	Value
Execute software reset.	GPIO_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset completed?	GPIO_SYSSTATUS[0] RESETDONE	= 0x1
Configure idle mode.	GPIO_SYSCONFIG[4:3] IDLEMODE	0x–
Configure interface clock gating.	GPIO_SYSCONFIG[0] AUTOIDLE	0x–
Set clock gating power saving.	GPIO_LEVELDETECT0[31:0] LEVELDETECT0 and GPIO_LEVELDETECT1[31:0] LEVELDETECT1 and GPIO_RISINGDETECT[31:0] RISINGDETECT and GPIO_FALLINGDETECT[31:0] FALLINGDETECT	0x0000 00FF
Set clock gating ratio.	GPIO_CTRL[2:1] GATINGRATIO	0x–
Configure GPIO channels as input or output.	GPIO_OE[31:0] OUTPUTEN	0x–
Set debounce time value.	GPIO_DEBOUNCINGTIME[7:0] DEBOUNCETIME	0x–
Enable/disable debouncing for desired input line.	GPIO_DEBOUNCENABLE[7:0] DEBOUNCENABLE	0x–
<b>Interrupt and Wake-Up Request Configuration</b>		
Enable/disable interrupts.	GPIO_IRQSTATUS_SET_0[31:0] INTLINE and/or GPIO_IRQSTATUS_SET_1[31:0] INTLINE	0x–
Enable/disable wakeup for GPIO channels.	GPIO_IRQWAKEN_0[31:0] INTLINE and/or GPIO_IRQWAKEN_1[31:0] INTLINE	0x–
Enable/disable wake-up generation	GPIO_SYSCONFIG[2] ENAWAKEUP	0x1

## 25.5.1.2 General-Purpose Interface Operational Modes Configuration

### 25.5.1.2.1 General-Purpose Interface Read Input Register

**Table 25-13. General-Purpose Interface Read Input Register**

Step	Register /Bit Field / Programming Model	Value
Read input register value	GPIO_DATAIN[31:0] DATAIN	0x-

### 25.5.1.2.2 General-Purpose Interface Write Output Register

**Table 25-14. General-Purpose Interface Write Output Register**

Step	Register /Bit Field / Programming Model	Value
Write value to output register	GPIO_DATAOUT[31:0] DATAOUT	0x-

### 25.5.1.2.3 General-Purpose Interface Set Bit Function

**Table 25-15. General-Purpose Interface Set Bit Function**

Step	Register /Bit Field / Programming Model	Value
Write to set desired bit(s) in register	GPIO_SETDATAOUT[31:0] INTLINE	0x1
	GPIO_IRQSTATUS_SET_0[31:0] INTLINE	
	GPIO_IRQSTATUS_SET_1[31:0] INTLINE	
	GPIO_IRQWAKEN_0[31:0] INTLINE	
	GPIO_IRQWAKEN_1[31:0] INTLINE	

### 25.5.1.2.4 General-Purpose Interface Clear Bit Function

**Table 25-16. General-Purpose Interface Clear Bit Function**

Step	Register / Bit Field / Programming Model	Value
Write to clear desired bit(s) in register	GPIO_CLEARDATAOUT[31:0] INTLINE	0x1
	GPIO_IRQSTATUS_CLR_0[31:0] INTLINE	
	GPIO_IRQSTATUS_CLR_1[31:0] INTLINE	
	GPIO_IRQWAKEN_0[31:0] INTLINE	
	GPIO_IRQWAKEN_1[31:0] INTLINE	

## 25.6 General-Purpose Interface Register Manual

### 25.6.1 General-Purpose Interface Instance Summary

Table 25-17 summarizes the general-purpose interface instance.

**Table 25-17. Instance Summary**

Module Name	Base Address	Size
GPIO1	0x4A31 0000	4KB
GPIO2	0x4805 5000	4KB
GPIO3	0x4805 7000	4KB
GPIO4	0x4805 9000	4KB
GPIO5	0x4805 B000	4KB
GPIO6	0x4805 D000	4KB

### 25.6.2 General-Purpose Interface Registers

#### 25.6.2.1 General-Purpose Interface Register Summary

Table 25-18 summarizes the general-purpose interface GPIO1 to GPIO3 registers.

**Table 25-18. General-Purpose Interface GPIO1 to GPIO3 Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPIO1 L4 Physical Address	GPIO2 L4 Physical Address	GPIO3 L4 Physical Address
GPIO_REVISION	R	32	0x0000 0000	0x4A31 0000	0x4805 5000	0x4805 7000
GPIO_SYSCONFIG	RW	32	0x0000 0010	0x4A31 0010	0x4805 5010	0x4805 7010
RESERVED			0x0000 0020	0x4A31 0020	0x4805 5020	0x4805 7020
GPIO_IRQSTATUS_RAW_0	RW	32	0x0000 0024	0x4A31 0024	0x4805 5024	0x4805 7024
GPIO_IRQSTATUS_RAW_1	RW	32	0x0000 0028	0x4A31 0028	0x4805 5028	0x4805 7028
GPIO_IRQSTATUS_0	RW	32	0x0000 002C	0x4A31 002C	0x4805 502C	0x4805 702C
GPIO_IRQSTATUS_1	RW	32	0x0000 0030	0x4A31 0030	0x4805 5030	0x4805 7030
GPIO_IRQSTATUS_SET_0	RW	32	0x0000 0034	0x4A31 0034	0x4805 5034	0x4805 7034
GPIO_IRQSTATUS_SET_1	RW	32	0x0000 0038	0x4A31 0038	0x4805 5038	0x4805 7038
GPIO_IRQSTATUS_CLR_0	RW	32	0x0000 003C	0x4A31 003C	0x4805 503C	0x4805 703C
GPIO_IRQSTATUS_CLR_1	RW	32	0x0000 0040	0x4A31 0040	0x4805 5040	0x4805 7040
GPIO_IRQWAKEN_0	RW	32	0x0000 0044	0x4A31 0044	0x4805 5044	0x4805 7044
GPIO_IRQWAKEN_1	RW	32	0x0000 0048	0x4A31 0048	0x4805 5048	0x4805 7048
GPIO_SYSSTATUS	R	32	0x0000 0114	0x4A31 0114	0x4805 5114	0x4805 7114
GPIO_IRQSTATUS1	RW	32	0x0000 0118	0x4A31 0118	0x4805 5118	0x4805 7118
GPIO_IRQENABLE1	RW	32	0x0000 011C	0x4A31 011C	0x4805 511C	0x4805 711C
GPIO_WAKEUPENABLE	RW	32	0x0000 0120	0x4A31 0120	0x4805 5120	0x4805 7120
GPIO_IRQSTATUS2	RW	32	0x0000 0128	0x4A31 0128	0x4805 5128	0x4805 7128
GPIO_IRQENABLE2	RW	32	0x0000 012C	0x4A31 012C	0x4805 512C	0x4805 712C
GPIO_CTRL	RW	32	0x0000 0130	0x4A31 0130	0x4805 5130	0x4805 7130
GPIO_OE	RW	32	0x0000 0134	0x4A31 0134	0x4805 5134	0x4805 7134
GPIO_DATAIN	R	32	0x0000 0138	0x4A31 0138	0x4805 5138	0x4805 7138
GPIO_DATAOUT	RW	32	0x0000 013C	0x4A31 013C	0x4805 513C	0x4805 713C
GPIO_LEVELDETECT0	RW	32	0x0000 0140	0x4A31 0140	0x4805 5140	0x4805 7140
GPIO_LEVELDETECT1	RW	32	0x0000 0144	0x4A31 0144	0x4805 5144	0x4805 7144
GPIO_RISINGDETECT	RW	32	0x0000 0148	0x4A31 0148	0x4805 5148	0x4805 7148
GPIO_FALLINGDETECT	RW	32	0x0000 014C	0x4A31 014C	0x4805 514C	0x4805 714C

**Table 25-18. General-Purpose Interface GPIO1 to GPIO3 Registers Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	GPIO1 L4 Physical Address	GPIO2 L4 Physical Address	GPIO3 L4 Physical Address
GPIO_DEBOUNCENABLE	RW	32	0x0000 0150	0x4A31 0150	0x4805 5150	0x4805 7150
GPIO_DEBOUNCINGTIME	RW	32	0x0000 0154	0x4A31 0154	0x4805 5154	0x4805 7154
GPIO_CLEARIRQENABLE1	RW	32	0x0000 0160	0x4A31 0160	0x4805 5160	0x4805 7160
GPIO_SETIRQENABLE1	RW	32	0x0000 0164	0x4A31 0164	0x4805 5164	0x4805 7164
GPIO_CLEARIRQENABLE2	RW	32	0x0000 0170	0x4A31 0170	0x4805 5170	0x4805 7170
GPIO_SETIRQENABLE2	RW	32	0x0000 0174	0x4A31 0174	0x4805 5174	0x4805 7174
GPIO_CLEARWKUPENA	RW	32	0x0000 0180	0x4A31 0180	0x4805 5180	0x4805 7180
GPIO_SETWKUPENA	RW	32	0x0000 0184	0x4A31 0184	0x4805 5184	0x4805 7184
GPIO_CLEARDATAOUT	RW	32	0x0000 0190	0x4A31 0190	0x4805 5190	0x4805 7190
GPIO_SETDATAOUT	RW	32	0x0000 0194	0x4A31 0194	0x4805 5194	0x4805 7194

Table 25-19 summarizes the general-purpose interface GPIO4 to GPIO6 registers.

**Table 25-19. General-Purpose Interface GPIO4 to GPIO6 Registers Summary**

Register Name	Type	Register Width (Bits)	Address Offset	GPIO4 L4 Physical Address	GPIO5 L4 Physical Address	GPIO6 L4 Physical Address
GPIO_REVISION	R	32	0x0000 0000	0x4805 9000	0x4805 B000	0x4805 D000
GPIO_SYSCONFIG	RW	32	0x0000 0010	0x4805 9010	0x4805 B010	0x4805 D010
RESERVED			0x0000 0020	0x4805 9020	0x4805 B020	0x4805 D020
GPIO_IRQSTATUS_RAW_0	RW	32	0x0000 0024	0x4805 9024	0x4805 B024	0x4805 D024
GPIO_IRQSTATUS_RAW_1	RW	32	0x0000 0028	0x4805 9028	0x4805 B028	0x4805 D028
GPIO_IRQSTATUS_0	RW	32	0x0000 002C	0x4805 902C	0x4805 B02C	0x4805 D02C
GPIO_IRQSTATUS_1	RW	32	0x0000 0030	0x4805 9030	0x4805 B030	0x4805 D030
GPIO_IRQSTATUS_SET_0	RW	32	0x0000 0034	0x4805 9034	0x4805 B034	0x4805 D034
GPIO_IRQSTATUS_SET_1	RW	32	0x0000 0038	0x4805 9038	0x4805 B038	0x4805 D038
GPIO_IRQSTATUS_CLR_0	RW	32	0x0000 003C	0x4805 903C	0x4805 B03C	0x4805 D03C
GPIO_IRQSTATUS_CLR_1	RW	32	0x0000 0040	0x4805 9040	0x4805 B040	0x4805 D040
GPIO_IRQWAKEN_0	RW	32	0x0000 0044	0x4805 9044	0x4805 B044	0x4805 D044
GPIO_IRQWAKEN_1	RW	32	0x0000 0048	0x4805 9048	0x4805 B048	0x4805 D048
GPIO_SYSSTATUS	R	32	0x0000 0114	0x4805 9114	0x4805 B114	0x4805 D114
GPIO_IRQSTATUS1	RW	32	0x0000 0118	0x4805 9118	0x4805 B118	0x4805 D118
GPIO_IRQENABLE1	RW	32	0x0000 011C	0x4805 911C	0x4805 B11C	0x4805 D11C
GPIO_WAKEUPENABLE	RW	32	0x0000 0120	0x4805 9120	0x4805 B120	0x4805 D120
GPIO_IRQSTATUS2	RW	32	0x0000 0128	0x4805 9128	0x4805 B128	0x4805 D128
GPIO_IRQENABLE2	RW	32	0x0000 012C	0x4805 912C	0x4805 B12C	0x4805 D12C
GPIO_CTRL	RW	32	0x0000 0130	0x4805 9130	0x4805 B130	0x4805 D130
GPIO_OE	RW	32	0x0000 0134	0x4805 9134	0x4805 B134	0x4805 D134
GPIO_DATAIN	R	32	0x0000 0138	0x4805 9138	0x4805 B138	0x4805 D138
GPIO_DATAOUT	RW	32	0x0000 013C	0x4805 913C	0x4805 B13C	0x4805 D13C
GPIO_LEVELDETECT0	RW	32	0x0000 0140	0x4805 9140	0x4805 B140	0x4805 D140
GPIO_LEVELDETECT1	RW	32	0x0000 0144	0x4805 9144	0x4805 B144	0x4805 D144
GPIO_RISINGDETECT	RW	32	0x0000 0148	0x4805 9148	0x4805 B148	0x4805 D148
GPIO_FALLINGDETECT	RW	32	0x0000 014C	0x4805 914C	0x4805 B14C	0x4805 D14C
GPIO_DEBOUNCENABLE	RW	32	0x0000 0150	0x4805 9150	0x4805 B150	0x4805 D150
GPIO_DEBOUNCINGTIME	RW	32	0x0000 0154	0x4805 9154	0x4805 B154	0x4805 D154
GPIO_CLEARIRQENABLE1	RW	32	0x0000 0160	0x4805 9160	0x4805 B160	0x4805 D160
GPIO_SETIRQENABLE1	RW	32	0x0000 0164	0x4805 9164	0x4805 B164	0x4805 D164

**Table 25-19. General-Purpose Interface GPIO4 to GPIO6 Registers Summary (continued)**

Register Name	Type	Register Width (Bits)	Address Offset	GPIO4 L4 Physical Address	GPIO5 L4 Physical Address	GPIO6 L4 Physical Address
<a href="#">GPIO_CLEARIRQENABLE2</a>	RW	32	0x0000 0170	0x4805 9170	0x4805 B170	0x4805 D170
<a href="#">GPIO_SETIRQENABLE2</a>	RW	32	0x0000 0174	0x4805 9174	0x4805 B174	0x4805 D174
<a href="#">GPIO_CLEARWKUPENA</a>	RW	32	0x0000 0180	0x4805 9180	0x4805 B180	0x4805 D180
<a href="#">GPIO_SETWKUENA</a>	RW	32	0x0000 0184	0x4805 9184	0x4805 B184	0x4805 D184
<a href="#">GPIO_CLEARDATAOUT</a>	RW	32	0x0000 0190	0x4805 9190	0x4805 B190	0x4805 D190
<a href="#">GPIO_SETDATAOUT</a>	RW	32	0x0000 0194	0x4805 9194	0x4805 B194	0x4805 D194

**25.6.2.2 General-Purpose Interface Register Description**

Table 25-20 through Table 25-90 describe the individual general-purpose interface registers.

**Table 25-20. GPIO\_REVISION**

<b>Address Offset</b>	0x0000 0000																																																																		
<b>Physical Address</b>	See Table 25-18	<b>Instance</b>																																																																	
		GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																	
<b>Description</b>	IP Revision Identifier (X.Y.R)																																																																		
<b>Type</b>	R																																																																		
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">REVISION</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	REVISION																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
REVISION																																																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																															
31:0	REVISION	IP Revision	R	See <sup>(1)</sup>																																																															

<sup>(1)</sup> TI Internal Data

**Table 25-21. Register Call Summary for Register GPIO\_REVISION**

- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)

**Table 25-22. GPIO\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010																																																					
<b>Physical Address</b>	See Table 25-18	<b>Instance</b>																																																				
		GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																				
<b>Description</b>	System configuration register																																																					
<b>Type</b>	R																																																					
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td>IDLEMODE</td><td>ENAWAKEUP</td><td>SOFTRESET</td><td>AUTOIDLE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																IDLEMODE	ENAWAKEUP	SOFTRESET	AUTOIDLE
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																							
RESERVED																IDLEMODE	ENAWAKEUP	SOFTRESET	AUTOIDLE																																			

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x00000000
4:3	IDLEMODE	0x0: Force Idle: An idle request is acknowledged unconditionally. 0x1: No Idle: An idle request is never acknowledged. 0x2: Smart Idle: The acknowledgment to an idle request is given based on the internal activity (see 4.1.2). 0x3: Smart Idle Wake-up	RW	0x0
2	ENAWAKEUP	Wake-up control. 0x0: Wake-up generation is disabled. 0x1: Wake-up capability is enabled upon expected transition on input GPIO pin	RW	0
1	SOFTRESET	Software Reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0x0: Normal Mode 0x1: The module is reset.	RW	0
0	AUTOIDLE	OCP clock gating control. 0x0: Internal Interface OCP clock is free-running 0x1: Automatic internal OCP clock gating, based on the OCP interface activity	RW	0

**Table 25-23. Register Call Summary for Register GPIO\_SYSCONFIG**

General-Purpose Interface Functional Description

- [Asynchronous Path: Wake-Up Request Generation: \[0\]](#)
- [General-Purpose Interface Hardware and Software Reset: \[1\] \[2\] \[3\]](#)
- [System Power Management and Wake-Up: \[4\] \[5\] \[6\] \[7\] \[8\] \[9\]](#)
- [Module Power Saving: \[10\] \[11\] \[12\] \[13\] \[14\] \[15\]](#)

General-Purpose Interface Programming Guide

- [General-Purpose Interface Module Global Initialization: \[16\] \[17\] \[18\] \[19\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[20\] \[21\]](#)

**Table 25-24. GPIO\_IRQSTATUS\_RAW\_0**

<b>Address Offset</b>	0x0000 0024																																																																		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Description</b>	Per-event raw interrupt status vector (corresponding to first line of interrupt)																																																																		
<b>Type</b>	RW																																																																		
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">INTLINE</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	INTLINE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
INTLINE																																																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																															
31:0	INTLINE	Status Raw for Interrupt Line	RW	0x0000 0000																																																															

**Table 25-25. Register Call Summary for Register GPIO\_IRQSTATUS\_RAW\_0**

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)



**Table 25-26. GPIO\_IRQSTATUS\_RAW\_1**

<b>Address Offset</b>	0x0000 0028	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b> GPIO1
<b>Description</b>	Per-event raw interrupt status vector (corresponding to second line of interrupt)	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status Raw for Interrupt Line	RW	0x0000 0000

**Table 25-27. Register Call Summary for Register GPIO\_IRQSTATUS\_RAW\_1**

- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)

**Table 25-28. GPIO\_IRQSTATUS\_0**

<b>Address Offset</b>	0x0000 002C	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b> GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Per-event interrupt status vector (enabled) (corresponding to first line of interrupt)	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status for Interrupt Line	RW	0x0000 0000

**Table 25-29. Register Call Summary for Register GPIO\_IRQSTATUS\_0**

- General-Purpose Interface Functional Description
- [Synchronous Path: Interrupt Request Generation: \[0\]](#)
  - [Asynchronous Path: Wake-Up Request Generation: \[1\] \[2\]](#)
  - [Interrupt \(or Wake-Up\) Line Release: \[3\]](#)
  - [Interrupt Requests Generation: \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[9\] \[10\]](#)

**Table 25-30. GPIO\_IRQSTATUS\_1**

<b>Address Offset</b>	0x0000 0030		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Per-event enabled interrupt status vector (corresponding to second line of interrupt)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status for Interrupt Line	RW	0x0000 0000

**Table 25-31. Register Call Summary for Register GPIO\_IRQSTATUS\_1**

General-Purpose Interface Functional Description

- [Synchronous Path: Interrupt Request Generation: \[0\]](#)
- [Asynchronous Path: Wake-Up Request Generation: \[1\] \[2\]](#)
- [Interrupt \(or Wake-Up\) Line Release: \[3\]](#)
- [Interrupt Requests Generation: \[4\] \[5\] \[6\] \[7\] \[8\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[9\] \[10\]](#)

**Table 25-32. GPIO\_IRQSTATUS\_SET\_0**

<b>Address Offset</b>	0x0000 0034		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Per-event interrupt enable set vector (corresponding to first line of interrupt)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status Set for Interrupt Line	RW	0x0000 0000

**Table 25-33. Register Call Summary for Register GPIO\_IRQSTATUS\_SET\_0**

General-Purpose Interface Overview

- [General-Purpose Interface Overview: \[0\]](#)

General-Purpose Interface Functional Description

- [General-Purpose Interface Block Diagram: \[1\]](#)
- [Synchronous Path: Interrupt Request Generation: \[2\]](#)
- [Asynchronous Path: Wake-Up Request Generation: \[3\] \[4\]](#)
- [Interrupt Requests Generation: \[5\] \[6\] \[7\] \[8\]](#)
- [General-Purpose Interface Data Input/Output Capabilities: \[9\] \[10\] \[11\]](#)
- [Description: \[12\]](#)
- [Set Register Addresses: \[13\] \[14\] \[15\]](#)

**Table 25-33. Register Call Summary for Register GPIO\_IRQSTATUS\_SET\_0 (continued)**

- General-Purpose Interface Programming Guide
- [General-Purpose Interface Module Global Initialization: \[16\]](#)
  - [General-Purpose Interface Set Bit Function: \[17\]](#)
- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[18\] \[19\]](#)

**Table 25-34. GPIO\_IRQSTATUS\_SET\_1**

<b>Address Offset</b>	0x0000 0038																																																																	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>																																																																
		GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Description</b>	Per-event enable set interrupt vector (corresponding to second line of interrupt)																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">INTLINE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	INTLINE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
INTLINE																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	INTLINE	Status Set for Interrupt Line	RW	0x0000 0000																																																														

**Table 25-35. Register Call Summary for Register GPIO\_IRQSTATUS\_SET\_1**

- General-Purpose Interface Overview
- [General-Purpose Interface Overview: \[0\]](#)
- General-Purpose Interface Functional Description
- [General-Purpose Interface Block Diagram: \[1\]](#)
  - [Synchronous Path: Interrupt Request Generation: \[2\]](#)
  - [Interrupt Requests Generation: \[3\] \[4\] \[5\] \[6\]](#)
  - [General-Purpose Interface Data Input/Output Capabilities: \[7\]](#)
  - [Description: \[8\]](#)
  - [Set Register Addresses: \[9\] \[10\] \[11\]](#)
- General-Purpose Interface Programming Guide
- [General-Purpose Interface Module Global Initialization: \[12\]](#)
  - [General-Purpose Interface Set Bit Function: \[13\]](#)
- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[14\] \[15\]](#)

**Table 25-36. GPIO\_IRQSTATUS\_CLR\_0**

<b>Address Offset</b>	0x0000 003C																																																																	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>																																																																
		GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Description</b>	Per-event interrupt enable clear vector (corresponding to first line of interrupt)																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td style="background-color:yellow;">23</td><td style="background-color:yellow;">22</td><td style="background-color:yellow;">21</td><td style="background-color:yellow;">20</td><td style="background-color:yellow;">19</td><td style="background-color:yellow;">18</td><td style="background-color:yellow;">17</td><td style="background-color:yellow;">16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td style="background-color:yellow;">7</td><td style="background-color:yellow;">6</td><td style="background-color:yellow;">5</td><td style="background-color:yellow;">4</td><td style="background-color:yellow;">3</td><td style="background-color:yellow;">2</td><td style="background-color:yellow;">1</td><td style="background-color:yellow;">0</td> </tr> <tr> <td colspan="32">INTLINE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	INTLINE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
INTLINE																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	INTLINE	Status Clear for Interrupt Line	RW	0x0000 0000																																																														

**Table 25-37. Register Call Summary for Register GPIO\_IRQSTATUS\_CLR\_0**

General-Purpose Interface Overview
<ul style="list-style-type: none"> <li><a href="#">General-Purpose Interface Overview: [0]</a></li> </ul>
General-Purpose Interface Functional Description
<ul style="list-style-type: none"> <li><a href="#">General-Purpose Interface Block Diagram: [1]</a></li> <li><a href="#">Synchronous Path: Interrupt Request Generation: [2]</a></li> <li><a href="#">Asynchronous Path: Wake-Up Request Generation: [3]</a></li> <li><a href="#">Interrupt Requests Generation: [4] [5]</a></li> <li><a href="#">Description: [6]</a></li> <li><a href="#">Clear Register Addresses: [7] [8] [9]</a></li> </ul>
General-Purpose Interface Programming Guide
<ul style="list-style-type: none"> <li><a href="#">General-Purpose Interface Clear Bit Function: [10]</a></li> </ul>
General-Purpose Interface Register Manual
<ul style="list-style-type: none"> <li><a href="#">General-Purpose Interface Register Summary: [11] [12]</a></li> </ul>

**Table 25-38. GPIO\_IRQSTATUS\_CLR\_1**

<b>Address Offset</b>	0x0000 0040																																																																	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>																																																																
		GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Description</b>	Per-event enable clear interrupt vector (corresponding to second line of interrupt)																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td style="background-color:yellow;">23</td><td style="background-color:yellow;">22</td><td style="background-color:yellow;">21</td><td style="background-color:yellow;">20</td><td style="background-color:yellow;">19</td><td style="background-color:yellow;">18</td><td style="background-color:yellow;">17</td><td style="background-color:yellow;">16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td style="background-color:yellow;">7</td><td style="background-color:yellow;">6</td><td style="background-color:yellow;">5</td><td style="background-color:yellow;">4</td><td style="background-color:yellow;">3</td><td style="background-color:yellow;">2</td><td style="background-color:yellow;">1</td><td style="background-color:yellow;">0</td> </tr> <tr> <td colspan="32">INTLINE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	INTLINE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
INTLINE																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	INTLINE	Status Clear for Interrupt Line	RW	0x0000 0000																																																														

**Table 25-39. Register Call Summary for Register GPIO\_IRQSTATUS\_CLR\_1**

General-Purpose Interface Overview
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Overview: [0]</a></li> </ul>
General-Purpose Interface Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Block Diagram: [1]</a></li> <li>• <a href="#">Synchronous Path: Interrupt Request Generation: [2]</a></li> <li>• <a href="#">Asynchronous Path: Wake-Up Request Generation: [3]</a></li> <li>• <a href="#">Interrupt Requests Generation: [4] [5]</a></li> <li>• <a href="#">Description: [6]</a></li> <li>• <a href="#">Clear Register Addresses: [7] [8] [9]</a></li> </ul>
General-Purpose Interface Programming Guide
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Clear Bit Function: [10]</a></li> </ul>
General-Purpose Interface Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Register Summary: [11] [12]</a></li> </ul>

**Table 25-40. GPIO\_IRQWAKEN\_0**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Physical Address</b>	See <a href="#">Table 25-18</a>																																																																		
<b>Description</b>	Per-event wake-up enable set vector (corresponding to first line of interrupt)																																																																		
<b>Type</b>	RW																																																																		
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">INTLINE</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	INTLINE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
INTLINE																																																																			
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																															
31:0	INTLINE	Wake-up Set for Interrupt Line	RW	0x0000 0000																																																															

**Table 25-41. Register Call Summary for Register GPIO\_IRQWAKEN\_0**

General-Purpose Interface Overview
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Overview: [0]</a></li> </ul>
General-Purpose Interface Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">Asynchronous Path: Wake-Up Request Generation: [1] [2]</a></li> <li>• <a href="#">Module Power Saving: [3]</a></li> <li>• <a href="#">Interrupt Requests Generation: [4]</a></li> <li>• <a href="#">Wake-Up Requests Generation: [5]</a></li> <li>• <a href="#">General-Purpose Interface Data Input/Output Capabilities: [6]</a></li> <li>• <a href="#">Description: [7]</a></li> <li>• <a href="#">Clear Register Addresses: [8]</a></li> <li>• <a href="#">Set Register Addresses: [9]</a></li> </ul>
General-Purpose Interface Programming Guide
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Module Global Initialization: [10]</a></li> <li>• <a href="#">General-Purpose Interface Set Bit Function: [11]</a></li> <li>• <a href="#">General-Purpose Interface Clear Bit Function: [12]</a></li> </ul>
General-Purpose Interface Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Register Summary: [13] [14]</a></li> </ul>

**Table 25-42. GPIO\_IRQWAKEN\_1**

<b>Address Offset</b>	0x0000 0048		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Per-event wake-up enable set vector (corresponding to second line of interrupt)		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Wake-up Set for Interrupt Line	RW	0x0000 0000

**Table 25-43. Register Call Summary for Register GPIO\_IRQWAKEN\_1**

## General-Purpose Interface Overview

- [General-Purpose Interface Overview: \[0\]](#)

## General-Purpose Interface Functional Description

- [Asynchronous Path: Wake-Up Request Generation: \[1\] \[2\]](#)
- [Module Power Saving: \[3\]](#)
- [Interrupt Requests Generation: \[4\]](#)
- [Wake-Up Requests Generation: \[5\]](#)
- [General-Purpose Interface Data Input/Output Capabilities: \[6\]](#)
- [Description: \[7\]](#)
- [Clear Register Addresses: \[8\]](#)
- [Set Register Addresses: \[9\]](#)

## General-Purpose Interface Programming Guide

- [General-Purpose Interface Module Global Initialization: \[10\]](#)
- [General-Purpose Interface Set Bit Function: \[11\]](#)
- [General-Purpose Interface Clear Bit Function: \[12\]](#)

## General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[13\] \[14\]](#)

**Table 25-44. GPIO\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0114		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	System Status register.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Read 0x0: Internal Reset is on-going Read 0x1: Reset completed	R	0

**Table 25-45. Register Call Summary for Register GPIO\_SYSSTATUS**

General-Purpose Interface Functional Description

- [General-Purpose Interface Hardware and Software Reset: \[0\] \[1\]](#)

General-Purpose Interface Programming Guide

- [General-Purpose Interface Module Global Initialization: \[2\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[3\] \[4\]](#)

**Table 25-46. GPIO\_IRQSTATUS1**

<b>Address Offset</b>	0x0000 0118	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Physical Address</b>	See <a href="#">Table 25-18</a>																																																																		
<b>Description</b>	Interrupt Status Register (legacy) for first line of interrupt.																																																																		
<b>Type</b>	RW																																																																		
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td style="background-color:yellow;">23</td><td style="background-color:yellow;">22</td><td style="background-color:yellow;">21</td><td style="background-color:yellow;">20</td><td style="background-color:yellow;">19</td><td style="background-color:yellow;">18</td><td style="background-color:yellow;">17</td><td style="background-color:yellow;">16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td style="background-color:yellow;">7</td><td style="background-color:yellow;">6</td><td style="background-color:yellow;">5</td><td style="background-color:yellow;">4</td><td style="background-color:yellow;">3</td><td style="background-color:yellow;">2</td><td style="background-color:yellow;">1</td><td style="background-color:yellow;">0</td> </tr> <tr> <td colspan="32">INTLINE</td> </tr> </table>				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	INTLINE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																				
INTLINE																																																																			

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status for Interrupt Line 0x0: No interrupt set 0x1: Interrupt set	RW	0x0000 0000

**Table 25-47. Register Call Summary for Register GPIO\_IRQSTATUS1**

General-Purpose Interface Functional Description

- [Interrupt Requests Generation: \[0\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[1\] \[2\]](#)

**Table 25-48. GPIO\_IRQENABLE1**

<b>Address Offset</b>	0x0000 011C		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Interrupt Enable Register (legacy) for first line of interrupt. 0 Disabled 1 Enabled		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status for Interrupt Line 0x0: Interrupt Disabled 0x1: Interrupt enabled	RW	0x0000 0000

**Table 25-49. Register Call Summary for Register GPIO\_IRQENABLE1**

General-Purpose Interface Functional Description

- [Interrupt Requests Generation: \[0\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[1\] \[2\]](#)

**Table 25-50. GPIO\_WAKEUPENABLE**

<b>Address Offset</b>	0x0000 0120		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Wake-up Enable Register (legacy) for first line of interrupt.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Wake-up Enable for Interrupt Line 0 Wake-up generation is disabled 1 Wake-up generation is enabled 0x0: Wake-up generation is disabled 0x1: Wake-up generation is enabled	RW	0x0000 0000

**Table 25-51. Register Call Summary for Register GPIO\_WAKEUPENABLE**

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)



**Table 25-52. GPIO\_IRQSTATUS2**

<b>Address Offset</b>	0x0000 0128																																																																	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b> GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Description</b>	Interrupt Status Register (legacy) for second line of interrupt. 0 No interrupt set 1 Interrupt set																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">INTLINE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	INTLINE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
INTLINE																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	INTLINE	Status for Interrupt Line 0x0: No interrupt set 0x1: Interrupt set	RW	0x0000 0000																																																														

**Table 25-53. Register Call Summary for Register GPIO\_IRQSTATUS2**

General-Purpose Interface Functional Description

- [Interrupt Requests Generation: \[0\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[1\] \[2\]](#)

**Table 25-54. GPIO\_IRQENABLE2**

<b>Address Offset</b>	0x0000 012C																																																																	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b> GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Description</b>	Interrupt Enable Register (legacy) for second line of interrupt. 0 Disabled 1 Enabled																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td colspan="32">INTLINE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	INTLINE																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
INTLINE																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	INTLINE	Enable for Interrupt Line 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0000 0000																																																														

**Table 25-55. Register Call Summary for Register GPIO\_IRQENABLE2**

General-Purpose Interface Functional Description

- [Interrupt Requests Generation: \[0\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[1\] \[2\]](#)

**Table 25-56. GPIO\_CTRL**

<b>Address Offset</b>	0x0000 0130		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	GPIO Control register.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											GATINGRATIO	DISABLEMODULE			

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:1	GATINGRATIO	Clock gating ratio for event detection 0x0: N = 1 0x1: N = 2 0x2: N = 4 0x3: N = 8	RW	0x1
0	DISABLEMODULE	0x0: Module is enabled, clocks are not gated. 0x1: Module is disabled, internal clocks are gated	RW	0

**Table 25-57. Register Call Summary for Register GPIO\_CTRL**

## General-Purpose Interface Functional Description

- [Clocking: \[0\]](#)
- [Operating Modes: \[1\]](#)
- [Module Power Saving: \[2\] \[3\] \[4\]](#)

## General-Purpose Interface Programming Guide

- [General-Purpose Interface Module Global Initialization: \[5\]](#)

## General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[6\] \[7\]](#)

**Table 25-58. GPIO\_OE**

<b>Address Offset</b>	0x0000 0134	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>
		GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Output Enable Register 0 Output Enabled 1 Output disabled	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUTEN																															

Bits	Field Name	Description	Type	Reset
31:0	OUTPUTEN	Output Enable 0x0: Output Enabled 0x1: Output Disabled	RW	0xFFFF FFFF

**Table 25-59. Register Call Summary for Register GPIO\_OE**

General-Purpose Interface Overview
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Overview: [0]</a></li> </ul>
General-Purpose Interface Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Data Input/Output Capabilities: [1] [2] [3]</a></li> </ul>
General-Purpose Interface Programming Guide
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Module Global Initialization: [4]</a></li> </ul>
General-Purpose Interface Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Register Summary: [5] [6]</a></li> </ul>

**Table 25-60. GPIO\_DATAIN**

<b>Address Offset</b>	0x0000 0138	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>
		GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Data Input Register (with sampled input data).	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN																															

Bits	Field Name	Description	Type	Reset
31:0	DATAIN	Sampled input data	R	0x0000 0000

**Table 25-61. Register Call Summary for Register GPIO\_DATAIN**

General-Purpose Interface Overview
<ul style="list-style-type: none"> <li>• <a href="#">General-Purpose Interface Overview: [0]</a></li> </ul>
General-Purpose Interface Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">System Power Management and Wake-Up: [1]</a></li> <li>• <a href="#">Module Power Saving: [2]</a></li> <li>• <a href="#">General-Purpose Interface Data Input/Output Capabilities: [3]</a></li> </ul>

**Table 25-61. Register Call Summary for Register GPIO\_DATAIN (continued)**

General-Purpose Interface Programming Guide

- [General-Purpose Interface Read Input Register: \[4\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[5\] \[6\]](#)

**Table 25-62. GPIO\_DATAOUT**

<b>Address Offset</b>	0x0000 013C																																																																	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>																																																																
		GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Description</b>	Data Output Register (data to set on output pins).																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%;">31</td><td style="width: 5%;">30</td><td style="width: 5%;">29</td><td style="width: 5%;">28</td><td style="width: 5%;">27</td><td style="width: 5%;">26</td><td style="width: 5%;">25</td><td style="width: 5%;">24</td><td style="width: 5%;">23</td><td style="width: 5%;">22</td><td style="width: 5%;">21</td><td style="width: 5%;">20</td><td style="width: 5%;">19</td><td style="width: 5%;">18</td><td style="width: 5%;">17</td><td style="width: 5%;">16</td><td style="width: 5%;">15</td><td style="width: 5%;">14</td><td style="width: 5%;">13</td><td style="width: 5%;">12</td><td style="width: 5%;">11</td><td style="width: 5%;">10</td><td style="width: 5%;">9</td><td style="width: 5%;">8</td><td style="width: 5%;">7</td><td style="width: 5%;">6</td><td style="width: 5%;">5</td><td style="width: 5%;">4</td><td style="width: 5%;">3</td><td style="width: 5%;">2</td><td style="width: 5%;">1</td><td style="width: 5%;">0</td> </tr> <tr> <td colspan="32" style="text-align: center;">DATAOUT</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DATAOUT																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
DATAOUT																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	DATAOUT	Data to set on output pins	RW	0x0000 0000																																																														

**Table 25-63. Register Call Summary for Register GPIO\_DATAOUT**

General-Purpose Interface Overview

- [General-Purpose Interface Overview: \[0\] \[1\]](#)

General-Purpose Interface Functional Description

- [General-Purpose Interface Data Input/Output Capabilities: \[2\]](#)
- [Description: \[3\]](#)

General-Purpose Interface Programming Guide

- [General-Purpose Interface Write Output Register: \[4\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[5\] \[6\]](#)

**Table 25-64. GPIO\_LEVELDETECT0**

<b>Address Offset</b>	0x0000 0140																																																																	
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>																																																																
		GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6																																																																
<b>Description</b>	Detect Low Level Register 0 Low level detection disabled 1 Low level detection enabled																																																																	
<b>Type</b>	RW																																																																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%;">31</td><td style="width: 5%;">30</td><td style="width: 5%;">29</td><td style="width: 5%;">28</td><td style="width: 5%;">27</td><td style="width: 5%;">26</td><td style="width: 5%;">25</td><td style="width: 5%;">24</td><td style="width: 5%;">23</td><td style="width: 5%;">22</td><td style="width: 5%;">21</td><td style="width: 5%;">20</td><td style="width: 5%;">19</td><td style="width: 5%;">18</td><td style="width: 5%;">17</td><td style="width: 5%;">16</td><td style="width: 5%;">15</td><td style="width: 5%;">14</td><td style="width: 5%;">13</td><td style="width: 5%;">12</td><td style="width: 5%;">11</td><td style="width: 5%;">10</td><td style="width: 5%;">9</td><td style="width: 5%;">8</td><td style="width: 5%;">7</td><td style="width: 5%;">6</td><td style="width: 5%;">5</td><td style="width: 5%;">4</td><td style="width: 5%;">3</td><td style="width: 5%;">2</td><td style="width: 5%;">1</td><td style="width: 5%;">0</td> </tr> <tr> <td colspan="32" style="text-align: center;">LEVELDETECT0</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LEVELDETECT0																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																			
LEVELDETECT0																																																																		
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																														
31:0	LEVELDETECT0	Low Level detection 0x0: Low level detection disabled 0x1: Low level detection enabled	RW	0x0000 0000																																																														

**Table 25-65. Register Call Summary for Register GPIO\_LEVELDETECT0**

General-Purpose Interface Functional Description

- [Module Power Saving: \[0\] \[1\]](#)

General-Purpose Interface Programming Guide

- [General-Purpose Interface Module Global Initialization: \[2\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[3\] \[4\]](#)

**Table 25-66. GPIO\_LEVELDETECT1**

<b>Address Offset</b>	0x0000 0144	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Physical Address</b>	See <a href="#">Table 25-18</a>		
<b>Description</b>	Detect High Level Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVELDETECT1																															

Bits	Field Name	Description	Type	Reset
31:0	LEVELDETECT1	0x0: High level detection disabled 0x1: High level detection enabled	RW	0x0000 0000

**Table 25-67. Register Call Summary for Register GPIO\_LEVELDETECT1**

General-Purpose Interface Functional Description

- [Module Power Saving: \[0\] \[1\]](#)

General-Purpose Interface Programming Guide

- [General-Purpose Interface Module Global Initialization: \[2\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[3\] \[4\]](#)

**Table 25-68. GPIO\_RISINGDETECT**

<b>Address Offset</b>	0x0000 0148	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Physical Address</b>	See <a href="#">Table 25-18</a>		
<b>Description</b>	Detect Rising Edge Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RISINGDETECT																															

Bits	Field Name	Description	Type	Reset
31:0	RISINGDETECT	0x0: Rising edge detection disabled 0x1: Rising edge detection enabled	RW	0x0000 0000

**Table 25-69. Register Call Summary for Register GPIO\_RISINGDETECT**

General-Purpose Interface Functional Description

- [Module Power Saving: \[0\] \[1\]](#)

General-Purpose Interface Programming Guide

- [General-Purpose Interface Module Global Initialization: \[2\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[3\] \[4\]](#)

**Table 25-70. GPIO\_FALLINGDETECT**

<b>Address Offset</b>	0x0000 014C	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Physical Address</b>	See <a href="#">Table 25-18</a>		
<b>Description</b>	Detect Falling Edge Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FALLINGDETECT																															

Bits	Field Name	Description	Type	Reset
31:0	FALLINGDETECT	0x0: Falling edge detection disabled 0x1: Falling edge detection enabled	RW	0x0000 0000

**Table 25-71. Register Call Summary for Register GPIO\_FALLINGDETECT**

General-Purpose Interface Functional Description

- [Module Power Saving: \[0\] \[1\]](#)

General-Purpose Interface Programming Guide

- [General-Purpose Interface Module Global Initialization: \[2\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[3\] \[4\]](#)

**Table 25-72. GPIO\_DEBOUNCENABLE**

<b>Address Offset</b>	0x0000 0150	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Physical Address</b>	See <a href="#">Table 25-18</a>		
<b>Description</b>	Debouncing Enable Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEBOUNCEENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	DEBOUNCEENABLE	0x0: No debouncing 0x1: Debouncing Activated	RW	0x0000 0000

**Table 25-73. Register Call Summary for Register GPIO\_DEBOUNCENABLE**

- General-Purpose Interface Functional Description
- [Module Power Saving: \[0\]](#)
- General-Purpose Interface Programming Guide
- [General-Purpose Interface Module Global Initialization: \[1\]](#)
- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[2\] \[3\]](#)

**Table 25-74. GPIO\_DEBOUNCINGTIME**

<b>Address Offset</b>	0x0000 0154	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Physical Address</b>	See <a href="#">Table 25-18</a>		
<b>Description</b>	Debouncing Value Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DEBOUNCETIME															

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved.	R	0x000000
7:0	DEBOUNCETIME	8-bits values specifying the debouncing time in 31us steps	RW	0x00

**Table 25-75. Register Call Summary for Register GPIO\_DEBOUNCINGTIME**

- General-Purpose Interface Functional Description
- [Synchronous Path: Interrupt Request Generation: \[0\] \[1\]](#)
  - [Clocking: \[2\] \[3\] \[4\]](#)
  - [System Power Management and Wake-Up: \[5\] \[6\]](#)
- General-Purpose Interface Programming Guide
- [General-Purpose Interface Module Global Initialization: \[7\]](#)
- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[8\] \[9\]](#)

**Table 25-76. GPIO\_CLEARIRQENABLE1**

<b>Address Offset</b>	0x0000 0160	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Physical Address</b>	See <a href="#">Table 25-18</a>		
<b>Description</b>	Clear Interrupt Enable Register - Legacy Mode		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Clear the corresponding bit in the relevant Interrupt Enable Register	RW	0x0000 0000

**Table 25-77. Register Call Summary for Register GPIO\_CLEARIRQENABLE1**

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)

**Table 25-78. GPIO\_SETIRQENABLE1**

<b>Address Offset</b>	0x0000 0164		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Set Interrupt Enable Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Set the corresponding bit in the relevant Interrupt Enable Register	RW	0x0000 0000

**Table 25-79. Register Call Summary for Register GPIO\_SETIRQENABLE1**

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)

**Table 25-80. GPIO\_CLEARIRQENABLE2**

<b>Address Offset</b>	0x0000 0170		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Clear Interrupt Enable Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect. 0x1: Clear the corresponding bit in the relevant Interrupt Enable Register	RW	0x0000 0000



**Table 25-81. Register Call Summary for Register GPIO\_CLEARIRQENABLE2**

- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)

**Table 25-82. GPIO\_SETIRQENABLE2**

<b>Address Offset</b>	0x0000 0174	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Physical Address</b>	See <a href="#">Table 25-18</a>		
<b>Description</b>	Set Interrupt Enable Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Set the corresponding bit in the relevant Interrupt Enable Register	RW	0x0000 0000

**Table 25-83. Register Call Summary for Register GPIO\_SETIRQENABLE2**

- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)

**Table 25-84. GPIO\_CLEARWKUPENA**

<b>Address Offset</b>	0x0000 0180	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Physical Address</b>	See <a href="#">Table 25-18</a>		
<b>Description</b>	Clear Wake-up Enable Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0 No effect 1 Clear the corresponding bit in the Wake-up Enable Register 0x0: No effect 0x1: Clear the corresponding bit in the Wake-up Enable Register	RW	0x0000 0000

**Table 25-85. Register Call Summary for Register GPIO\_CLEARWKUPENA**

- General-Purpose Interface Register Manual
- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)

**Table 25-86. GPIO\_SETWKUENA**

<b>Address Offset</b>	0x0000 0184		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Set Wake-up Enable Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Set the corresponding bit in the Wake-up Enable Register	RW	0x0000 0000

**Table 25-87. Register Call Summary for Register GPIO\_SETWKUENA**

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[0\] \[1\]](#)

**Table 25-88. GPIO\_CLEARDATAOUT**

<b>Address Offset</b>	0x0000 0190		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Clear Data Output Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Clear the corresponding bit in the Data Output Register	RW	0x0000 0000

**Table 25-89. Register Call Summary for Register GPIO\_CLEARDATAOUT**

General-Purpose Interface Functional Description

- [General-Purpose Interface Data Input/Output Capabilities: \[0\]](#)
- [Clear Register Addresses: \[1\]](#)

General-Purpose Interface Programming Guide

- [General-Purpose Interface Clear Bit Function: \[2\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[3\] \[4\]](#)

**Table 25-90. GPIO\_SETDATAOUT**

<b>Address Offset</b>	0x0000 0194		
<b>Physical Address</b>	See <a href="#">Table 25-18</a>	<b>Instance</b>	GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6
<b>Description</b>	Set Data Output Register		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Set the corresponding bit in the Data Output Register	RW	0x0000 0000

**Table 25-91. Register Call Summary for Register GPIO\_SETDATAOUT**

General-Purpose Interface Functional Description

- [General-Purpose Interface Data Input/Output Capabilities: \[0\]](#)
- [Set Register Addresses: \[1\]](#)

General-Purpose Interface Programming Guide

- [General-Purpose Interface Set Bit Function: \[2\]](#)

General-Purpose Interface Register Manual

- [General-Purpose Interface Register Summary: \[3\] \[4\]](#)

PRELIMINARY

## Keyboard Controller

---

---

---

This chapter describes the keyboard module of the device.

Topic	Page
<b>26.1 Keyboard Controller Overview .....</b>	<b>5142</b>
<b>26.2 Keyboard Controller Environment .....</b>	<b>5144</b>
<b>26.3 Keyboard Controller Integration .....</b>	<b>5147</b>
<b>26.4 Keyboard Controller Functional Description .....</b>	<b>5149</b>
<b>26.5 Keyboard Controller Programming Guide .....</b>	<b>5159</b>
<b>26.6 Keyboard Controller Register Manual .....</b>	<b>5163</b>

## 26.1 Keyboard Controller Overview

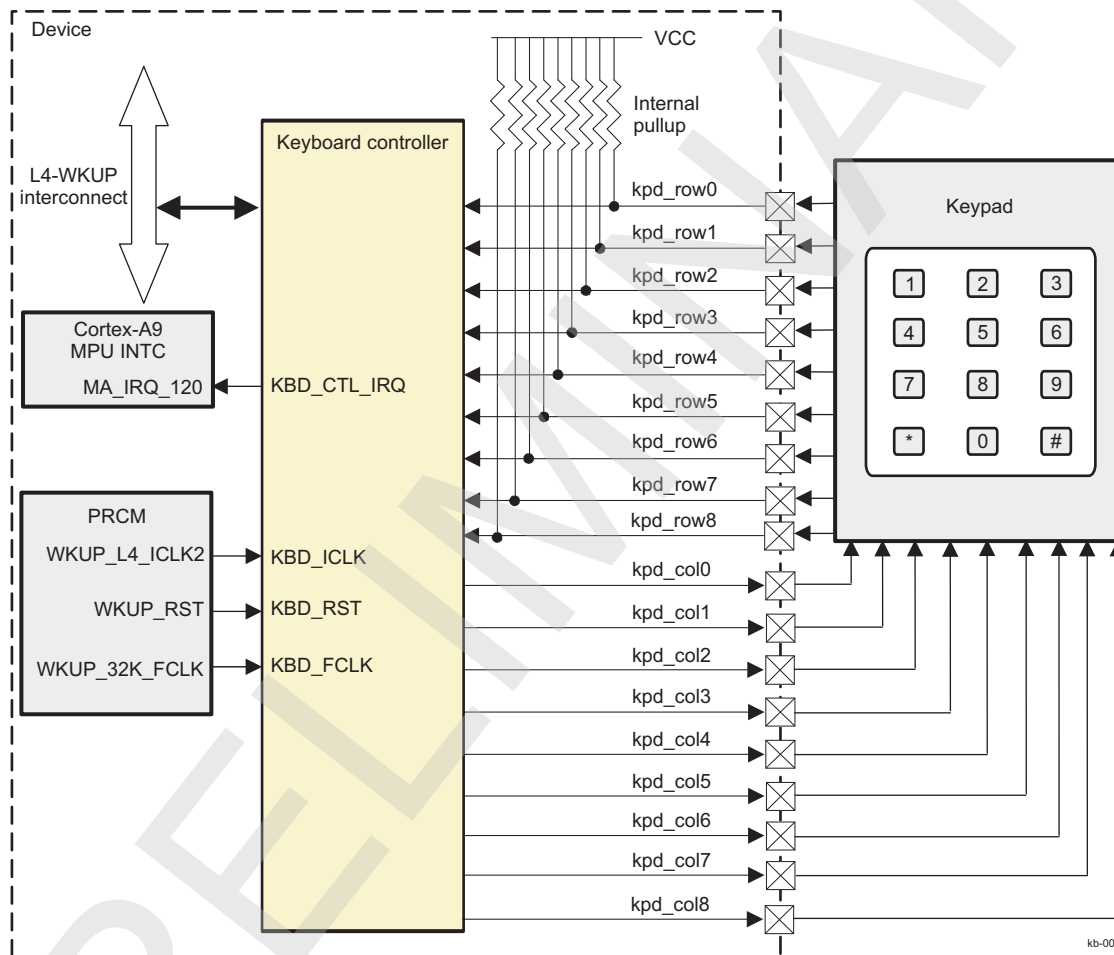
The keyboard controller implements a built-in scanning algorithm for hardware-based key-press decoding and reduces overhead in the microprocessor unit (MPU) software.

The keyboard controller includes a debouncing feature to ensure that only one key combination can be registered in the programmed time.

The keyboard controller can handle up to 9 × 9 keys, works on a 32-kHz clock, and can generate wake-up events when the chip is in sleep mode.

Figure 26-1 shows the keyboard controller.

**Figure 26-1. Keyboard Controller Overview**



The keyboard controller includes the following main features:

- Support of multiconfiguration keyboards up to 9 rows × 9 columns
- Each key coded on 1 bit in two 32-bit registers
- Long-key value or repeat timing reconfigurable on the fly
- Event detection on key press and key release
- Multikey-press detection and decoding
- Long-key detection on prolonged key press
- Integrated timer with four programmable comparison values
- Programmable time-out on permanent key press or after keyboard release
- Programmable interrupt generation on key events
- Software reset capability

- Read/write-posted register access modes
- 8-/16-/32-bit access supported on the level 4 (L4) interface
- 32-bit data bus
- 7-bit address bus

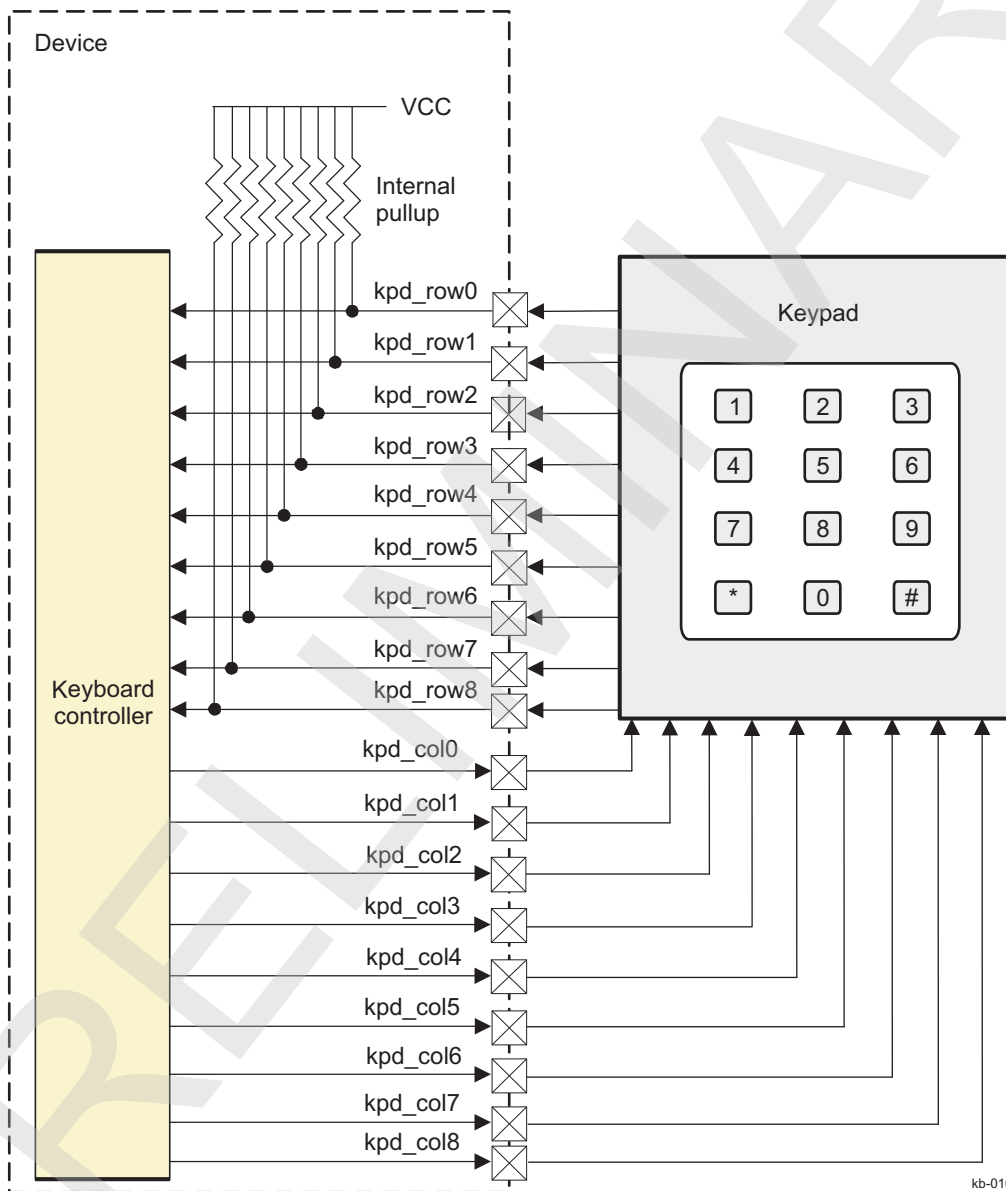
PRELIMINARY

## 26.2 Keyboard Controller Environment

The keyboard controller external interface pins map on the device pads when the device general-purpose input/output (GPIO) cells are configured and multiplexed for the keypad function by the control module. For more information, see [Section 18.4.8, PAD Functional Multiplexing and Configuration](#), and [Section 25.2.1, General-Purpose Interface as a Keyboard interface](#).

The external keypad typically connects directly to the keyboard controller of the device (see [Figure 26-2](#)).

**Figure 26-2. Typical Keyboard Environment**



### 26.2.1 Keyboard Controller Functions/Modes

The keyboard controller executes two functions:

- Key pad scanning and decoding for input of the external key presses. For more information, see [Section 26.4.6.4, Software Mode](#), and [Section 26.4.5, Keyboard Controller Hardware Decoding Modes](#).
- Device wake-up by interrupt request to the processor when a key is pressed and the device is in idle or sleep mode. For more information, see [Section 26.4.6, Keyboard Controller Interrupt Generation](#).



### 26.2.2 Keyboard Controller Signals

Table 26-1 describes the module signals and specifies their links to functions.

**Table 26-1. I/O External Keyboard Signals**

Signal	I/O <sup>(1)</sup>	Description	Reset Value	Function	Wake-Up Capability
kpd_row0	I	Keypad row 0 feed	HiZ (pulled up)	Key reading	Yes
kpd_row1	I	Keypad row 1 feed	HiZ (pulled up)	Key reading	Yes
kpd_row2	I	Keypad row 2 feed	HiZ (pulled up)	Key reading	Yes
kpd_row3	I	Keypad row 3 feed	HiZ (pulled up)	Key reading	Yes
kpd_row4	I	Keypad row 4 feed	HiZ (pulled up)	Key reading	Yes
kpd_row5	I	Keypad row 5 feed	HiZ (pulled up)	Key reading	Yes
kpd_row6	I	Keypad row 6 feed	HiZ (pulled up)	Key reading	Yes
kpd_row7	I	Keypad row 7 feed	HiZ (pulled up)	Key reading	Yes
kpd_row8	I	Keypad row 8 feed	HiZ (pulled up)	Key reading	Yes
kpd_col0	O	Keypad column 0 feed, active low	HiZ	Key reading	No
kpd_col1	O	Keypad column 1 feed, active low	HiZ	Key reading	No
kpd_col2	O	Keypad column 2 feed, active low	HiZ	Key reading	No
kpd_col3	O	Keypad column 3 feed, active low	HiZ	Key reading	No
kpd_col4	O	Keypad column 4 feed, active low	HiZ	Key reading	No
kpd_col5	O	Keypad column 5 feed, active low	HiZ	Key reading	No
kpd_col6	O	Keypad column 6 feed, active low	HiZ	Key reading	No
kpd_col7	O	Keypad column 7 feed, active low	HiZ	Key reading	No
kpd_col8	O	Keypad column 8 feed, active low	HiZ	Key reading	No

<sup>(1)</sup> I = Input; O = Output

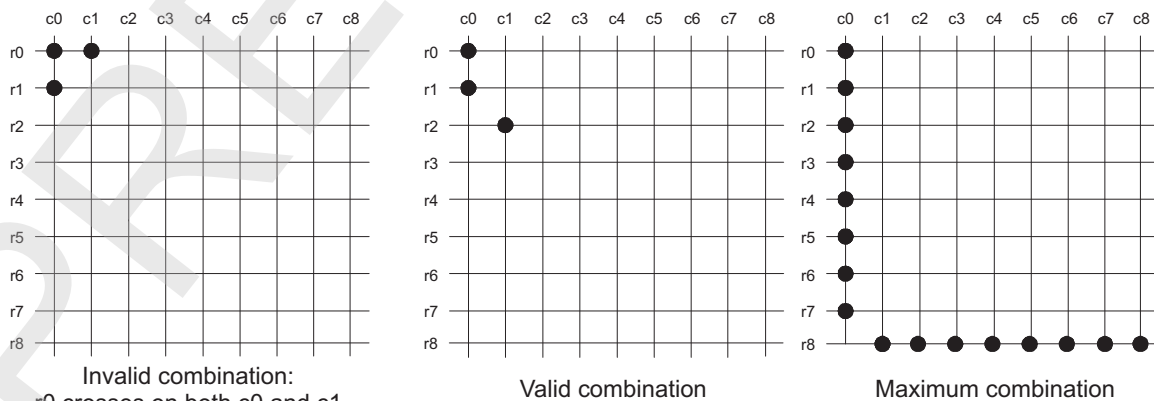
### 26.2.3 Protocols and Data Formats

The keyboard controller detects and decodes multikey combinations using the following rules:

- Any 2-key combination is valid and can be decoded.
- Combinations using more than two keys are valid only if the rows and columns used do not cross over on another key to be detected. This is caused by equipotent propagation on a row/column (multikey limitations).

Figure 26-3 shows an example of multikey limitation.

**Figure 26-3. Multikey Limitation Example**



kb-007

**NOTE:** When using the keyboard controller with a smaller keypad (for example, 5 × 5), unused rows must be tied high to prevent disturbing the scanning process. Normally, all rows must be pulled up internally at the I/O cells of the device.

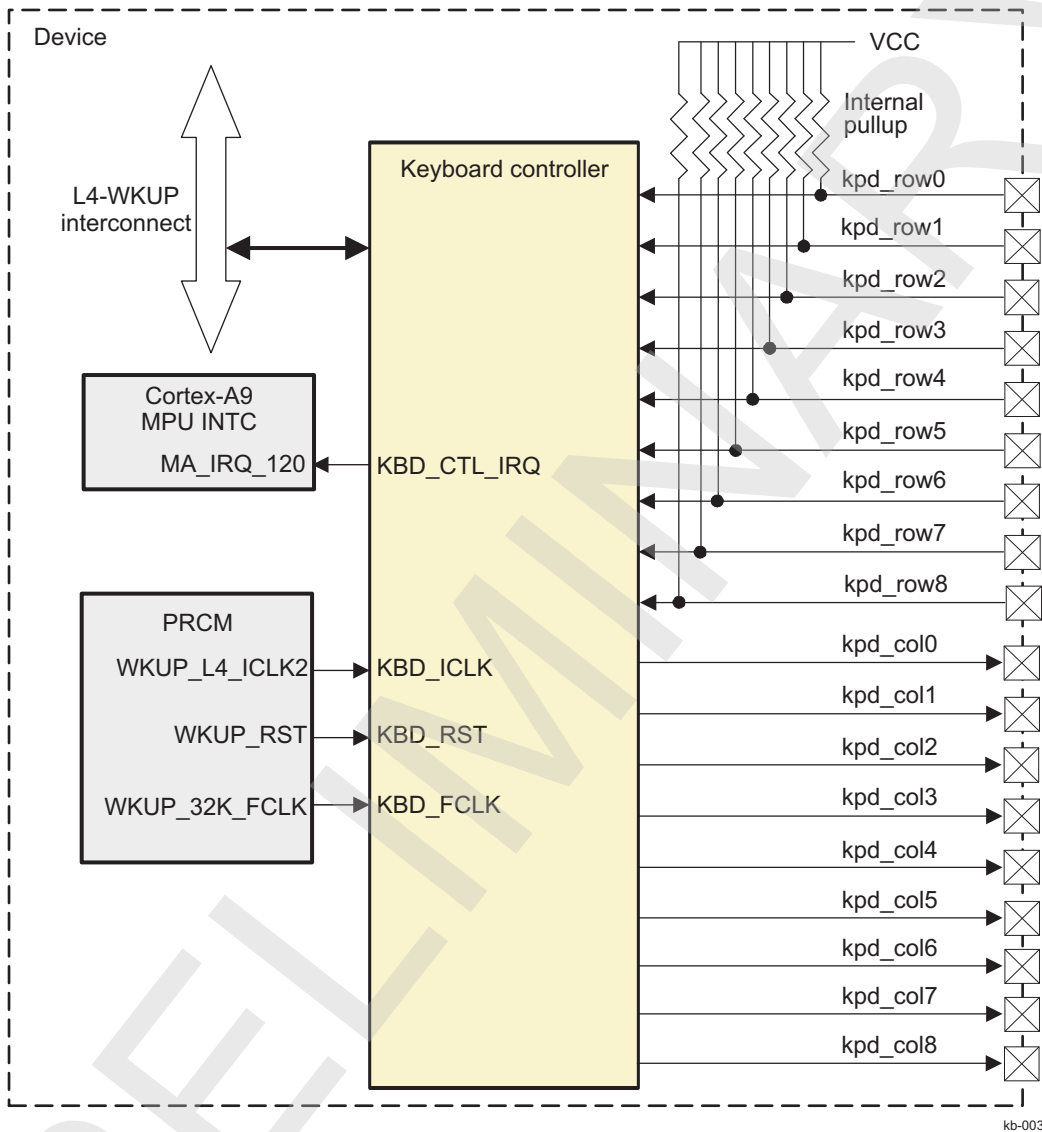
---

PRELIMINARY

### 26.3 Keyboard Controller Integration

Figure 26-4 shows keyboard controller integration.

Figure 26-4. Keyboard Controller Integration



The control module must enable the internal pullups of the GPIO cells for all keypad rows (for more information about configuration, see [Chapter 18, Control Module](#)).

[Table 26-2](#) through [Table 26-4](#) summarize the integration of the module in the device.

Table 26-2. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
KEYBOARD CONTROLLER	PD_WKUP	Yes	L4-WKUP

Table 26-3. Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description

**Table 26-3. Clocks and Resets (continued)**

KEYBOARD CONTROLLER	KBD_FCLK	WKUP_32K_FCLK	PRCM	32-kHz functional clock. For information about power, reset, clock management (PRCM) clock gating and management, see <a href="#">Section 3.1.1.1.4, Clock Domain Level Clock Management</a> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
KEYBOARD CONTROLLER	KBD_ICLK	WKUP_L4_ICLK2	PRCM	L4-Interconnect interface clock. For information about PRCM clock gating and management, see <a href="#">Section 3.1.1.1.4, Clock Domain Level Clock Management</a> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .
<b>Resets</b>				
KEYBOARD CONTROLLER	KBD_RST	WKUP_RST	PRCM	Active low asynchronous general reset. For information about PRCM reset sources and distribution, see <a href="#">Section 3.1.1.1.4, Clock Domain Level Clock Management</a> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a> .

**Table 26-4. Hardware Requests**

Interrupt Requests				
Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
KEYBOARD CONTROLLER	KBD_CTL_IRQ	MA_IRQ_120	Cortex-A9 INTC	Interrupt signal to the interrupt controller (INTC) of the MPU. For information about INTC interrupt control, see <a href="#">Chapter 17, Interrupt Controllers</a> .

**NOTE:**

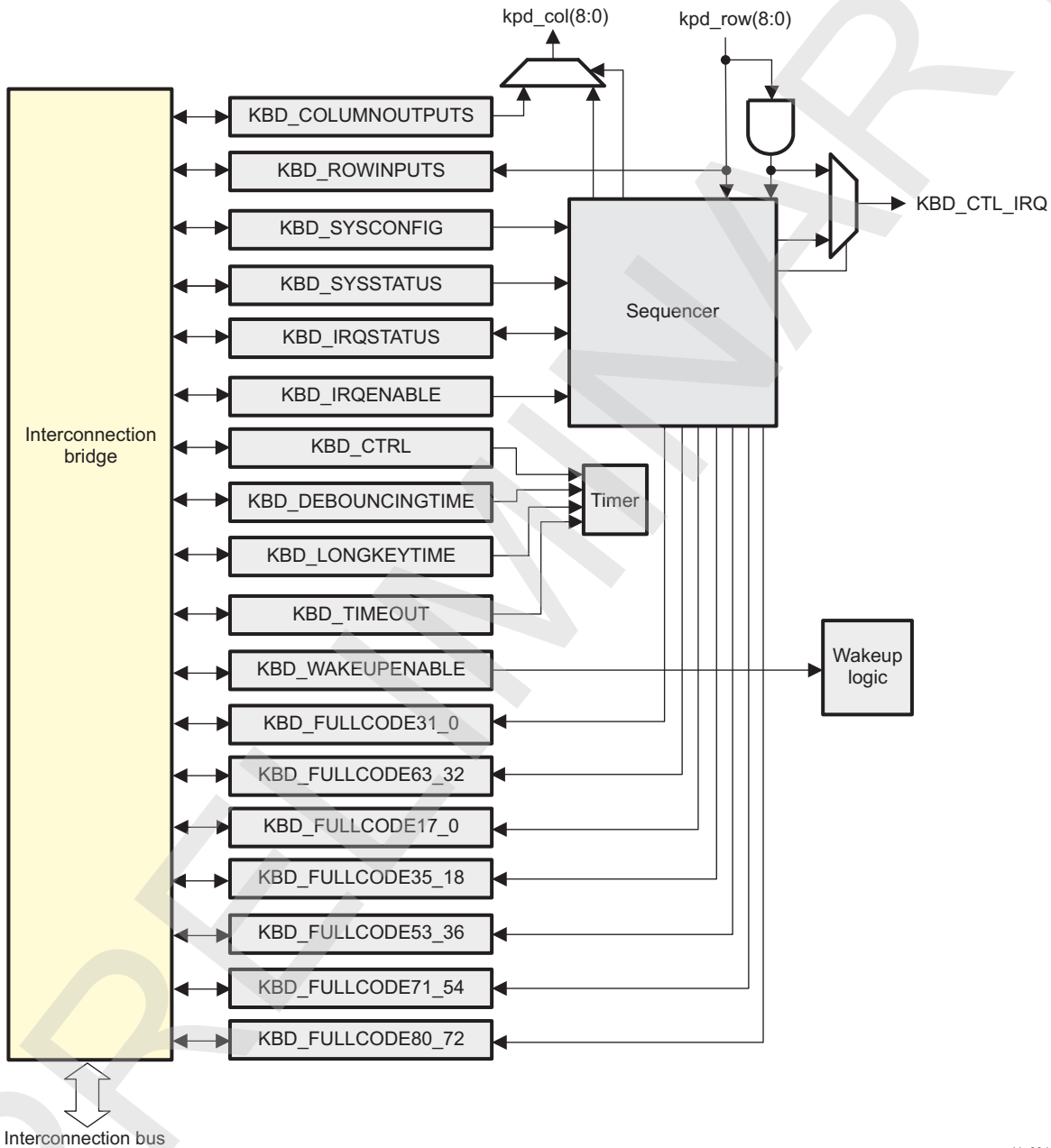
- For the description of the interrupt source, see [Section 26.4.4, Interrupt Requests](#).

## 26.4 Keyboard Controller Functional Description

### 26.4.1 Keyboard Controller Block Diagram

Figure 26-5 shows the functional specification block diagram of the keyboard controller.

Figure 26-5. Keyboard Controller Block Diagram



kb-004

The keyboard controller detects events issued on any key of the connected keyboard and generates an interrupt to alert the host processor. The built-in hardware-scan algorithm decodes the pressed keys, including multikey combinations.

To reduce MPU software overhead, the hardware performs detecting and decoding in the keyboard controller state-machine. However, hardware decoding can be deactivated so that software handles the scanning algorithm.

The value of the columns output is determined in the [KBD\\_COLUMNOUTPUTS\[8:0\]](#) KBC\_REG bit field. To activate a key pad row-column connection, the corresponding bit must be 0b0.

The following sections describe subfunctions and subfunction interactions (control, data paths).

### 26.4.2 Keyboard Controller Software Reset

To perform a software reset, write 1 to the [KBD\\_SYSCONFIG\[1\]](#) SOFTRESET bit. The [KBD\\_SYSSTATUS\[0\]](#) RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the [KBD\\_SYSCONFIG\[1\]](#) SOFTRESET bit is automatically reset. The software must ensure that the software reset completes before performing mailbox operations.

### 26.4.3 Keyboard Controller Power Management

[Table 26-5](#) describes the power-management features available for the keyboard controller module.

**NOTE:** For information about source clock gating and sleep/wake-up transitions description, see [Section 3.1.1.1.4, Clock Domain Level Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**Table 26-5. Local Power-Management Features**

Feature	Registers	Description
Clock autogating	<a href="#">KBD_SYSCONFIG[0]</a> AUTOGATING	This bit allows local power optimization in the module by gating the KBD_ICLK clock upon interface activity, or gating KBD_FCLK upon internal activity.
Slave idle modes	<a href="#">KBD_SYSCONFIG[4:3]</a> IDLEMODE	Force-idle, no-idle, and smart-idle modes are available.
Clock activity	<a href="#">KBD_SYSCONFIG[9:8]</a> CLOCKACTIVITY	For configuration details, see <a href="#">Table 26-6</a> .
Master standby modes	N/A	N/A
Global wake-up enable	<a href="#">KBD_SYSCONFIG[2]</a> ENAWAKEUP	This bit enables the wake-up feature at the module level. If smart-idle mode is selected, this bit provides the option of smart-idle wake-up-capable mode.
Wake-up sources enable	<a href="#">KBD_WAKEUPENABLE</a>	This register holds one active-high enable bit per event source able to generate a wake-up signal.

[Table 26-6](#) lists the clock activity settings.

**Table 26-6. Clock Activity Settings**

<a href="#">KBD_SYSCONFIG[9:8]</a> CLOCKACTIVITY Values	Clock State When Module is in IDLE State		Features Available When Module is in IDLE State	Wake-Up Events
	KBD_ICLK	KBD_FCLK		
00	OFF	OFF	None	The wake-up capability of the module is disabled.
10	OFF	ON	All	Asserting of wake-up signal
01	ON	OFF	None	The wake-up capability of the module is disabled.
11	ON	ON	All	Asserting of wake-up signal

**CAUTION**

The PRCM module has no hardware means of reading the CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the keyboard controller CLOCKACTIVITY and keyboard controller clock PRCM control bits. For a description of the ClockActivity feature, see [Section 3.1.1.1, Clock Management](#), in [Chapter 3, Power, Reset, and Clock Management](#).

**26.4.4 Keyboard Controller Interrupt Requests**

[Table 26-7](#) lists the event flags, and their mask, that can cause module interrupts.

**Table 26-7. Events**

Event Flag	Event Mask	Description
KBD_IRQSTATUS[3] MISS_EVENT	–	A miss event occurs.
KBD_IRQSTATUS[2] IT_TIMEOUT	KBD_IRQENABLE[2] IT_TIMEOUT_EN	A time-out event is detected.
KBD_IRQSTATUS[1] IT_LONG_KEY	KBD_IRQENABLE[1] IT_LONG_KEY_EN	A long-key event is detected.
KBD_IRQSTATUS[0] IT_EVENT	KBD_IRQENABLE[0] IT_EVENT_EN	An event is detected.

**26.4.5 Keyboard Controller Software Mode**

The KBD\_CTRL[1] NSOFTWARE\_MODE bit selects software mode when it is set to 0.

In software mode, the keyboard controller internal sequencer, which performs automatic scanning and decoding, is disabled. Consequently, the software must manually perform the scanning algorithm.

The scanning sequence is managed using the keyboard controller column outputs register (KBD\_COLUMNOUTPUTS) and the keyboard controller row inputs register (KBD\_ROWINPUTS). In this configuration, the keyboard interrupt is a logical ANDing of all bits of the keyboard controller row input register (KBD\_ROWINPUTS).

For more information about the software scan, see [Section 26.5.1.2.2, Keyboard Controller Basic Programming Model](#).

**26.4.6 Keyboard Controller Hardware Decoding Modes**

**26.4.6.1 Functional Modes**

When running in hardware(default) decoding mode (the KBD\_CTRL[1] NSOFTWARE\_MODE bit is set to 1), the keyboard controller offers several functional modes; these modes are summarized in [Table 26-8](#). The keyboard interrupt depends on the configuration in the keyboard controller interrupt-enable register (KBD\_IRQENABLE). If the event is enabled (bit 0 at 1), an interrupt is generated when the sequencer detects an event. Even if this interrupt is disabled, the flag status of the keyboard controller interrupt-status register (KBD\_IRQSTATUS) is updated.

**Table 26-8. Keyboard Controller Functional Modes**

Functional Mode	Associated Interrupt	Associated Timer Value	Description	Control
Keyboard event	Event interrupt	Debouncing value	Occurs when a key is pressed or released Always enabled	KBD_CTRL[8:5] bits must be set to 0 to disable all the other features.
Long key	Long-key interrupt	Long-key value	Used to detect a key that is pressed for a long time Should be associated with the long-key time-out functionality or repeat mode	KBD_CTRL[5] LONG_KEY bit

**Table 26-8. Keyboard Controller Functional Modes (continued)**

Functional Mode	Associated Interrupt	Associated Timer Value	Description	Control
Repeat key	Long-key interrupt	Long-key value	Generates an interrupt every long-key delay No time-out can be associated.	<a href="#">KBD_CTRL</a> [8] REPEAT_MODE bit
Empty time-out	Time-out interrupt	Empty time-out value	Interrupt generated if no key is pressed during an empty time-out period.	<a href="#">KBD_CTRL</a> [6] TIMEOUT_EMPTY bit
Long-key time-out	Time-out interrupt	Long-key time-out value	Associated with the long-key function Generated after a long-key interrupt if no event occurs during a long-key time-out period	<a href="#">KBD_CTRL</a> [7] TIMEOUT_LONG_KEY bit

Each mode can be activated/deactivated by setting the corresponding bits (5, 6, 7, and 8) in the [KBD\\_CTRL](#) register with the appropriate values (for more information, see [Section 26.6, Register Manual](#)).

### 26.4.6.2 Keyboard Controller Timer

As described in the previous section, each functional mode is associated with a timer value. Depending on the selected mode, the keyboard controller timer is loaded with the corresponding value as set in the related registers:

- [KBD\\_DEBOUNCINGTIME](#)
- [KBD\\_KEYLONGTIME](#)
- [KBD\\_TIMEOUT](#)

[Table 26-9](#) summarizes the keyboard controller timer values.

**Table 26-9. Keyboard Controller Timer Values**

Timer Value	Associated Register Field	Description
Debouncing time	<a href="#">KBD_DEBOUNCINGTIME</a> [5:0] DEBOUNCING_VALUE	To remove the effects of glitches when an event occurs on the keyboard, the controller waits for a debouncing period before taking a snapshot of the current state on the keyboard matrix. The timer is loaded with the debouncing time value after each detected event on the keyboard matrix. An event interrupt is generated after this delay.
Long-key time	<a href="#">KBD_KEYLONGTIME</a> [11:0] LONG_KEY_VALUE	This is the delay before generating a long-key interrupt after an event interrupt. If the long-key mode is selected, the timer is loaded with the long-key time value after an event interrupt is generated.  In repeat mode, the timer is reloaded with the same value after a long-key interrupt, and starts to count down again.
Long-key time-out	<a href="#">KBD_TIMEOUT</a> [15:0] TIMEOUT_VALUE	The timer is loaded with the time-out value and then a long-key interrupt is generated and starts to count down. When it reaches 0, a time-out interrupt is generated and the keyboard controller comes back to its IDLE state. This long-key time-out does not work in repeat mode.
Empty key time-out	<a href="#">KBD_TIMEOUT</a> [15:0] TIMEOUT_VALUE	The time-out interrupt occurs if no key is pressed during this delay. The keyboard controller then goes back to IDLE state.

The timer countdown period depends on three factors:

- The loaded value as set in:
  - [KBD\\_DEBOUNCINGTIME](#)
  - [KBD\\_KEYLONGTIME](#)
  - [KBD\\_TIMEOUT](#)
- The prescale clock timer value as set in the [KBD\\_CTRL](#)[4:2] PTV bit field. This programmable clock divider allows the reduction of the clock frequency used by the timer.
- The keyboard controller functional clock frequency (32 kHz)

The period is calculated as follows:



$$T_{\text{period}} = (T_{\text{value}} + 1) \times 2^{\text{PTV} + 1} \times T_{\text{clk}}$$

Where:

$T_{\text{value}}$  is the value stored in the [KBD\\_DEBOUNCINGTIME](#), [KBD\\_KEYLONGTIME](#), or [KBD\\_TIMEOUT](#) register.

PTV is the value of the [KBD\\_CTRL\[4:2\]](#) register PTV bit field.

$T_{\text{clk}}$  is the period of the 32-kHz functional clock; that is, 31.25  $\mu\text{s}$ .

The [KBD\\_CTRL\[4:2\]](#) PTV bit field determines the division factor of the timer clock. [Table 26-10](#) lists the divider rates.

**Table 26-10. Timer Prescale Values**

<a href="#">KBD_CTRL[4:2]</a> PTV Value	Divisor
0	2
1	4
2	8
3	16
4	32
5	64
6	128
7	256

**NOTE:** The timer minimum period is 62.5  $\mu\text{s}$ ; its maximum period is 524.288 s.

**CAUTION**

To prevent undefined results, the [KBD\\_CTRL\[4:2\]](#) PTV bit field must not be changed when the timer is running.

The timer value registers ([KBD\\_DEBOUNCINGTIME](#), [KBD\\_KEYLONGTIME](#), and [KBD\\_TIMEOUT](#)) can be updated at any time, whether or not the timer is running.. Nevertheless, the timer is updated only on the fly for the long-key time value. The new debouncing and time-out values are loaded only on the next load. Depending on the updated register, two cases can occur:

- The [KBD\\_KEYLONGTIME](#) register is updated; the new value stored in the [KBD\\_KEYLONGTIME](#) register is loaded into the timer when the register is written. If the timer is already counting down when [KBD\\_KEYLONGTIME](#) is updated, it counts down from the new value loaded in [KBD\\_KEYLONGTIME](#).
- The [KBD\\_DEBOUNCINGTIME](#) or [KBD\\_TIMEOUT](#) register is updated; the new value is considered only at the next timer load. If the timer is counting down when the registers are updated, the timer continues counting down from the previous value and is loaded with the new one on the next load.

Regardless of the timer state (stopped, counting down any of the values previously described, etc.), when a new event occurs on the keyboard, the timer is stopped and loaded with the debouncing time value. It then starts counting down.

**26.4.6.3 State-Machine Status**

To facilitate debugging, each state of the state-machine is coded in a register that indicates the current state of the machine. [Table 26-11](#) lists the corresponding codes.

**Table 26-11. State-Machine Values**

<a href="#">KBD_STATEMACHINE[3:0]</a> Value	Description
0x0	Idle
0x1	Scanning

**Table 26-11. State-Machine Values (continued)**

<b>KBD_STATEMACHINE[3:0] Value</b>	<b>Description</b>
0x2	Load timer debouncing
0x3	Test timer debouncing
0x4	Generated interrupt event
0x6	Load timer long key
0x7	Test timer long key
0x8	Generated interrupt long key
0x9	Load timer time-out
0xA	Test timer time-out
0xB	Generated interrupt time-out
0xF	Other

#### 26.4.6.4 Keyboard Controller Interrupt Generation

##### 26.4.6.4.1 Interrupt-Generation Scheme

The keyboard controller generates one interrupt signal to external INTC KBD\_CTL\_IRQ. Each functional mode generates dedicated interrupt events (logged in the [KBD\\_IRQSTATUS](#) register) that can be masked or unmasked using the [KBD\\_IRQENABLE](#) register.

The [KBD\\_IRQSTATUS](#) register is updated when the selected functional mode generates an interrupt event. However, the KBD\_CTL\_IRQ signal is asserted on the related event only when the corresponding bit is set in the [KBD\\_IRQENABLE](#) register.

---

**NOTE:** To reset the interrupt status bit, 1 must be written to the appropriate bit of the [KBD\\_IRQSTATUS](#) read/write register.

---

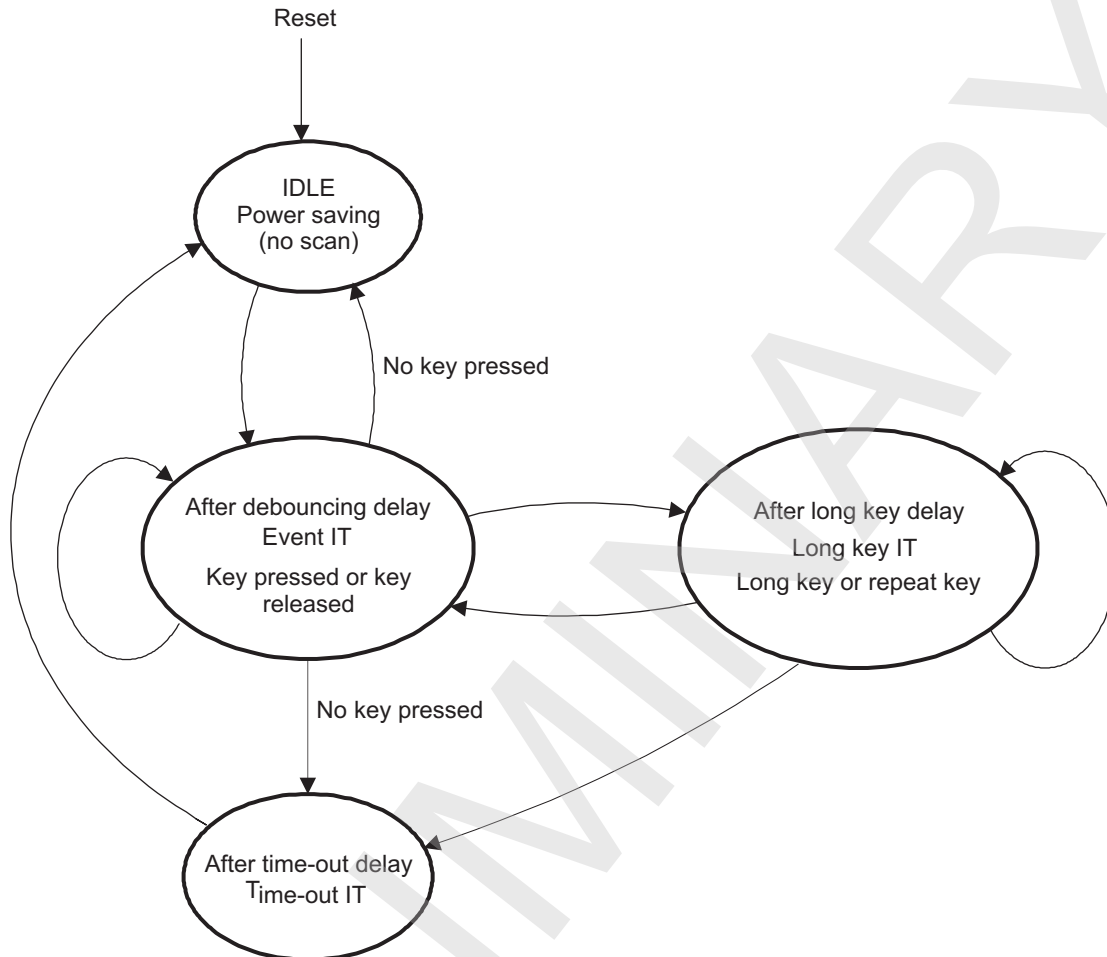
[Figure 26-6](#) shows the different interrupt events generated in each keyboard controller functional mode and details the relationships between them.

---

**NOTE:** Depending on the selected mode, some interrupt events cannot be generated.

---

Figure 26-6. Functional Modes and Related Interrupt Events



kb-005

While running in hardware-decoding mode, the keyboard controller performs automatic scans when not in IDLE state. When a key-press event occurs on the keyboard matrix, the keyboard controller leaves IDLE state and an interrupt event (the `KBD_IRQSTATUS[0]` IT\_EVENT bit) is set after the timer counts down the debouncing delay. An IT\_EVENT is generated when a key is pressed or released.

**NOTE:** An IT\_EVENT is generated regardless of the selected functional mode. If no time-out is set and no more keys are pressed, the keyboard controller goes back to IDLE state.

If long-key detection mode is set when the timer counts down the long-key delay, an interrupt long key (the `KBD_IRQSTATUS[1]` IT\_LONG\_KEY bit) is generated. If the repeat mode is set, the IT\_LONG\_KEY interrupt is generated periodically every long-key delay.

A time-out can also be set in event detection or long-key detection mode. In this case, a time-out interrupt (the `KBD_IRQSTATUS[2]` IT\_TIMEOUT bit) is generated after the time-out delay timer expires. After such an interrupt, the keyboard controller always goes back to idle.

**NOTE:** No time-out can be set in repeat mode. Only a keyboard event can stop the periodic interrupt generation.

### 26.4.6.4.2 Keyboard Buffer and Missed Events (Overrun Feature)

The keyboard controller has an overrun feature: A dedicated buffer allows the keyboard controller to memorize two successive events. If two successive events occur before a read is performed, the second event is stored and a second interrupt is generated when the first interrupt is cleared, allowing two key events to be received in a row.

If a third event occurs before the first event is treated, a missed event interrupt ([KBD\\_IRQSTATUS\[3\] MISS\\_EVENT](#) bit) is generated to report the lost event.

---

**NOTE:** The `MISS_EVENT` interrupt is not routed to the `KBD_CTL_IRQ` signal; the software must check the [KBD\\_IRQSTATUS\[3\] MISS\\_EVENT](#) bit to detect any missed events.

---

### 26.4.7 Keyboard Controller Key Coding Registers

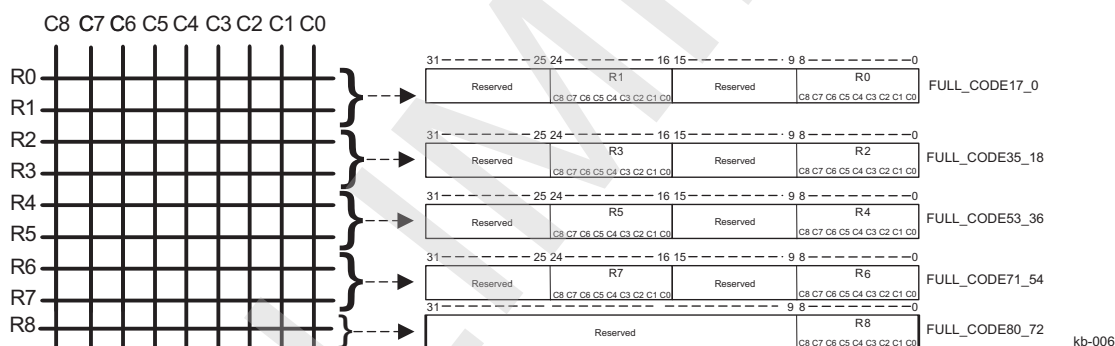
The keyboard controller matrix pressed keys state (indicating which columns/rows are connected) is reflected in the [KBD\\_FULLCODE17\\_0](#) up to [KBD\\_FULLCODE80\\_72](#) registers, as shown in [Figure 26-7](#). These registers are updated only when interrupt event status is inactive to prevent a lost event.

---

**NOTE:** The [KBD\\_FULLCODE31\\_0](#) and [KBD\\_FULLCODE63\\_32](#) registers, which are limited to supporting keyboards of a maximum 8 × 8 array size, can be used for back-to-back software compatibility.

---

**Figure 26-7. Key Coding Registers**



Each of the 9 × 9 keyboard size supporting registers ([KBD\\_FULLCODE17\\_0](#) up to [KBD\\_FULLCODE71\\_54](#)) stores the state of two keyboard rows. The [KBD\\_FULLCODE80\\_72](#) register stores the state of the last row, R8:

- The [KBD\\_FULLCODE17\\_0](#) register code rows 0, 1 (row 0 is coded between bits 0 and 8, row 1 is coded between bits 16 and 24)
- The [KBD\\_FULLCODE35\\_18](#) register code rows 2, 3 (row 2 is coded between bits 0 and 8, row 3 is coded between bits 16 and 24)
- The [KBD\\_FULLCODE53\\_36](#) register code rows 4, 5 (row 4 is coded between bits 0 and 8, row 5 is coded between bits 16 and 24)
- The [KBD\\_FULLCODE71\\_54](#) register code rows 6, 7 (row 6 is coded between bits 0 and 8, row 7 is coded between bits 16 and 24)
- The [KBD\\_FULLCODE80\\_72](#) register code row 8 (row 8 is coded between bits 0 and 8)

In each of these registers (excluding [KBD\\_FULLCODE80\\_72](#)):

- Bit 0 corresponds to column(0) – row (i) key, ..., bit 8 corresponds to column(8) – row(i) key.
- Bits from 9 to 15 are reserved.
- Bit 16 corresponds to column(0) – row(i+1) key, ..., bit 24 corresponds to column(8) – row(i+1) key.
- Bits 31 to 25 are reserved, where  $i = 0, 2, 4, 6$ .

In register [KBD\\_FULLCODE80\\_72](#) bit 0 corresponds to column(0) – row(8) key, ..., the bit 8 corresponds to column(8) – row(8) key, and the remaining bits (from 9 to 31) are reserved.

Each of the 8 × 8-keyboard size supporting registers ([KBD\\_FULLCODE31\\_0](#) and [KBD\\_FULLCODE63\\_32](#)) stores the state of four keyboard rows:

- The [KBD\\_FULLCODE31\\_0](#) register code rows 0, 1, 2, and 3 (row 0 is coded between bits 0 and 7, row 1 is coded between bits 8 and 15, row 2 is coded between bits 16 and 23, and row 3 is coded between bits 24 and 31)
- The [KBD\\_FULLCODE63\\_32](#) register code rows 4, 5, 6, and 7 (row 4 is coded between bits 0 and 7, row 5 is coded between bits 8 and 15, row 6 is coded between bits 16 and 23, and row 7 is coded between bits 24 and 31)

In these registers:

- [KBD\\_FULLCODE31\\_0](#)[0] corresponds to column(0) – row(0) key,..., [KBD\\_FULLCODE31\\_0](#)[7] corresponds to column(7) – row(0).
- [KBD\\_FULLCODE31\\_0](#)[8] corresponds to column(0) – row(1) key,..., [KBD\\_FULLCODE31\\_0](#)[15] corresponds to column(7) – row(1).
- [KBD\\_FULLCODE31\\_0](#)[16] corresponds to column(0) – row(2) key,..., [KBD\\_FULLCODE31\\_0](#)[23] corresponds to column(7) – row(2).
- [KBD\\_FULLCODE31\\_0](#)[24] corresponds to column(0) – row(3) key,..., [KBD\\_FULLCODE31\\_0](#)[31] corresponds to column(7) – row(3).
- [KBD\\_FULLCODE63\\_32](#)[0] corresponds to column(0) – row(4) key,..., [KBD\\_FULLCODE63\\_32](#)[7] corresponds to column(7) – row(4).
- [KBD\\_FULLCODE63\\_32](#)[8] corresponds to column(0) – row(5) key,..., [KBD\\_FULLCODE63\\_32](#)[15] corresponds to column(7) – row(5).
- [KBD\\_FULLCODE63\\_32](#)[16] corresponds to column(0) – row(6) key,..., [KBD\\_FULLCODE63\\_32](#)[23] corresponds to column(7) – row(6).
- [KBD\\_FULLCODE63\\_32](#)[24] corresponds to column(0) – row(7) key,..., [KBD\\_FULLCODE63\\_32](#)[31] corresponds to column(7) – row(7).

---

**NOTE:** The keyboard fullcode registers are not updated in software mode.

---



---

**NOTE:** When using a smaller keyboard (for example, 5 × 5 or 4 × 4), the bits of the unused columns/rows are not used either.

---

## 26.4.8 Keyboard Controller Register Access

### 26.4.8.1 Write Registers Access

The keyboard module uses a posted-write scheme to update any internal registers. This means the write transaction is immediately acknowledged on the L4-interface, although the effective write operation occurs later due to a resynchronization in the functional clock domain. This has the advantage of not stalling the interconnect system or the CPU that requested the write transaction. For each functional register, a pending bit is provided that is set if there is a pending write access to this register. The pending bits are accessible in the keyboard pending write register ([KBD\\_PENDING](#)).

In this mode, it is mandatory that the CPU checks the pending bits before any write access in the functional registers. In case a write is attempted to a register with a previous access pending, the previous access is discarded without notice (this can also lead to unexpected results).

A register read following a posted write (on the same register) may not read the previous write value if the write posted process is not complete. Software synchronization must be used to avoid noncoherent read.

This posted period is defined as the interval between the posted write access request and the reset of the pending bit in the [KBD\\_PENDING](#) register, and can be quantified:

$$T (\text{reset posted bit maximum}) = 3 * T_{\text{iclk}} + 3 * T_{\text{funcclk}}$$

The time it takes to accomplish the writing is:

$$T (\text{write accomplish maximum}) = 1 * T_{\text{iclk}} + 3 * T_{\text{funcclk}}$$

where:

Tickl is the L4-Interface clock period, and Tfuncclk is the functional clock period.

#### 26.4.8.2 Read Registers Access

The keyboard module uses a posted-read scheme for reading any internal register. The read transaction is immediately acknowledged on the L4-Interface. The value of the functional register to be read must be previously synchronized. This has the advantage of not stalling the interconnect system or the CPU that requested the read transaction.

The posted-read scheme can be used only if  $\text{Freq}(\text{KBD\_FCLK}) < \text{Freq}(\text{KBD\_ICLK})/4$ .

## 26.5 Keyboard Controller Programming Guide

### 26.5.1 Keyboard Controller Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the module.

#### 26.5.1.1 Global Initialization

##### 26.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the keyboard module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the keyboard. For more information, see [Section 26.2, Keyboard Controller Environment](#), and [Section 26.3, Keyboard Controller Integration](#).

A description of the global initialization of surrounding modules is presented in [Table 26-12](#).

**Table 26-12. Global Initialization of Surrounding Modules**

Surrounding Modules	Comments
Power, reset, and clock management (PRCM)	The module interface (WKUP_L4_ICLK2) and functional (WKUP_32K_FCLK) clocks must be enabled. See <i>Clock Management Global Initialization</i> , in <a href="#">Chapter 3, Power, Reset, and Clock Management</a>
Interrupt controller of the Cortex-A9 MPU (INTC)	Interrupt request KBD_CTL_IRQ from the keyboard must be unmasked. For more information, see <a href="#">Chapter 17, Interrupt Controllers</a> .
Control module	The PAD CONFIG registers must be configured in to map the keyboard interface signals on the device pins through the multiplexers (mux mode 0 must be switched on) to determine the signal directions and to enable the internal pullups of the GPIO cells for all the keyboard rows. For more information about configuration, see <a href="#">Section 18.4.8, Pad Functional Multiplexing and Configuration</a> , and <a href="#">Section 25.2.1, General-Purpose Interface as a Keyboard interface</a> .

##### 26.5.1.1.2 Keyboard Controller Global Initialization

###### 26.5.1.1.2.1 Main Sequence – Keyboard Controller Global Initialization

This procedure initializes the keyboard controller after a POR or software reset (see [Table 26-13](#)).

**Table 26-13. Keyboard Controller Global Initialization**

Step	Register/Bit Field/Programming Model	Value
Configure the parameters of the L4-Interface and enable the clock activity.	<a href="#">KBD_SYSCONFIG</a>	0x00000–
Configure the debouncing time of filtering the glitches on pressing or releasing key.	<a href="#">KBD_DEBOUNCINGTIME[5:0]</a> DEBOUNCING_VALUE	0x–
Configure the duration of a key press to allow shortcut detection (desired value of the long key interrupt or repeat mode value).	<a href="#">KBD_KEYLONGTIME[11:0]</a> LONG_KEY_VALUE	0x–
Configure the period of the long inactivity on the keyboard (desired value of the time-out interrupt).	<a href="#">KBD_TIMEOUT[15:0]</a> TIMEOUT_VALUE	0x–
Define the logical value of the column outputs (KEYPAD configuration) (logical 0 bit = active column-row).	<a href="#">KBD_COLUMNOUTPUTS[8:0]</a> KBC_REG	0x–
Perform the functional configuration of the keyboard module and the prescale clock timer value.	<a href="#">KBD_CTRL</a>	0x00000–
Clear the interrupt-status register.	<a href="#">KBD_IRQSTATUS</a>	0x0000000F
Enable (0b1)/disable (0b0) certain keyboard events for generating an Interrupt request.	<a href="#">KBD_IRQENABLE[2:0]</a> IT_..._EN	0x–
Unmask (0b1)/mask (0b0) the expected source of wake-up event that generates a wake-up request.	<a href="#">KBD_WAKEUPENABLE[2:0]</a> WUP_..._ENA	0x–

## 26.5.1.2 Operational Modes Configuration

### 26.5.1.2.1 Keyboard Controller in Hardware Decoding Mode (Default Mode)

#### 26.5.1.2.1.1 Main Sequence – Keyboard Controller Hardware Mode

After reset, all available functional modes are disabled, except detect-event mode, which is always active. [Table 26-14](#) describes the keyboard controller hardware mode.

**Table 26-14. Keyboard Controller Hardware Mode**

Step	Register/Bit Field/Programming Model	Value
Activate the internal keyboard controller sequencer by setting the bit.	<a href="#">KBD_CTRL</a> [1] NSOFTWARE_MODE	0b1
Select the functional mode by setting its corresponding bit to 1.	<a href="#">KBD_CTRL</a> [8:5]	0x–
Configure the duration of a key press to allow shortcut detection (desired value of the long key interrupt or repeat mode value).	<a href="#">KBD_KEYLONGTIME</a> [11:0] LONG_KEY_VALUE	0x–
Configure the period of the long inactivity on the keyboard (desired value of the time-out interrupt).	<a href="#">KBD_TIMEOUT</a> [15:0] TIMEOUT_VALUE	0x–
Configure the debouncing time of filtering the glitches on pressing or releasing key.	<a href="#">KBD_DEBOUNCINGTIME</a> [5:0] DEBOUNCING_VALUE	0x–
Clear the interrupt-status register.	<a href="#">KBD_IRQSTATUS</a>	0x0000000F
Enable (by writing 1)/disable (by writing 0) certain keyboard event for generating an Interrupt request	<a href="#">KBD_IRQENABLE</a> [2:0] IT_..._EN	0x–
Unmask (0b1)/mask (0b0) the expected source of wake-up event that generates a wake-up request.	<a href="#">KBD_WAKEUPENABLE</a> [2:0] WUP_..._ENA	0x–
Wait for the <a href="#">KBD_CTL_IRQ</a> interrupt signal assertion.		
Read the interrupt-status register to determine which event caused the interrupt.	<a href="#">KBD_IRQSTATUS</a>	
Read the <a href="#">KBD_FULLCODE17_0</a> to <a href="#">KBD_FULLCODE80_72</a> registers (or 8 × 8-keyboard-size-supporting <a href="#">KBD_FULLCODE31_0</a> and <a href="#">KBD_FULLCODE63_32</a> registers) to determine which key matrix combination was pressed.	<a href="#">KBD_FULLCODE17_0</a> [ROWi bits] (where i = 0 to 1) up to <a href="#">KBD_FULLCODE71_54</a> [ROWi bits] (where i = 6 to 7); <a href="#">KBD_FULLCODE80_72</a> [ROWi bits] (where i = 8); (or <a href="#">KBD_FULLCODE31_0</a> [j] FULL_CODE_31_0 and <a href="#">KBD_FULLCODE63_32</a> [k] FULL_CODE_63_32 bits (where j = 0 to 31, k = 32 to 63)	
Clear the corresponding bit(s) in the interrupt-status register by writing logical 1.	<a href="#">KBD_IRQSTATUS</a>	0x1

**NOTE:** The long-key detection mode and the repeat mode cannot be used simultaneously, because they share the same interrupt status bit and are mutually exclusive. Software must ensure that only one of these modes is selected at a time.

**NOTE:** All interrupts are disabled on reset

**NOTE:** When two events occur successively before the first event is read, the second interrupt is generated when the first interrupt is cleared. When more than two events in a row occur, the [KBD\\_IRQSTATUS](#)[3] MISS\_EVENT bit is set. Software must check this bit, which is not reflected on the [KBD\\_CTL\\_IRQ](#) line.

**NOTE:** The keyboard controller uses a posted-write scheme to update any internal register. Software must read the pending write status bits to ensure that the next write access is not discarded because of ongoing write synchronization. For more information, see [Section 26.4.8.1, Write Registers Access](#).



### 26.5.1.2.2 Keyboard Controller Software Scanning Mode

#### 26.5.1.2.2.1 Main Sequence – Keyboard Controller Software Mode

Table 26-15 contains a description of the keyboard controller software mode.

**Table 26-15. Keyboard Controller Software Mode**

Step	Register/Bit Field/Programming Model	Value D
Deactivate the internal keyboard controller sequencer by clearing the bit.	<a href="#">KBD_CTRL[1]</a> NSOFTWARE_MODE	0b0
Enable the event interrupt by setting the bit.	<a href="#">KBD_IRQENABLE[0]</a> IT_EVENT_EN	0b1
Wait for the KBD_CTL_IRQ interrupt signal assertion. Begin the software scan when the interrupt signal is asserted:		
1) Disable all columns to drive a logical 0 on the kpd_col[8:0] output by writing 0xFF. (logical 1 bit = inactive column-row)	<a href="#">KBD_COLUMNOUTPUTS[8:0]</a> KBC_REG	0xFF
2) Drive kpd_col(i) output, (where i = 0 to 8) , to 0 to capture a pressed-key-event, at the corresponding row input.	<a href="#">KBD_COLUMNOUTPUTS[i]</a> KBC_REG (where i = 0 to 8)	0b0
3) Read the KBR_LATCH bit field of the <a href="#">KBD_ROWINPUTS</a> register to determinate which is the pressed key. IF: KBR_LATCH k-bit = 0 , where k = 0 to 8 Then, the corresponding k-row is connected to the column being enabled. END IF	<a href="#">KBD_ROWINPUTS[8:0]</a> KBR_LATCH	
Repeat steps 2) and 3) for all existing columns.	<a href="#">KBD_WAKEUPENABLE[2:0]</a> WUP_..._ENA	0x–

**NOTE:** In software mode, during the manual keyboard scan, an interrupt is generated when a pressed key is detected.

#### 26.5.1.2.3 Using the Timer

For information about programming the keyboard controller timer, see [Section 26.4.6.2, Keyboard Controller Timer](#).

#### 26.5.1.2.4 State-Machine Status Register

To see the state of the state-machine, see [Section 26.4.6.3, State-Machine Status](#).

### 26.5.1.3 Keyboard Controller Events Servicing

Table 26-16 lists the keyboard controller event servicing.

**Table 26-16. Keyboard Controller Event Servicing**

Step	Register/Bit Field/Programming Model	Value
Clear all eventual previous indications of treated interrupts; write the IRQSTATUS register:	<a href="#">KBD_IRQSTATUS</a>	0x0000000F
Enable the desired sources of interrupt by writing in:	<a href="#">KBD_IRQENABLE</a>	0x0000000–
Unmask (enable) the desired sources of interrupt by writing logical 1 in the desired bit fields.	<a href="#">KBD_WAKEUPENABLE</a>	0x0000000–
When event occurs:		
Read the status register.	<a href="#">KBD_IRQSTATUS</a>	
IF: <a href="#">KBD_IRQSTATUS[3]</a> MISS_EVENT = 1	Start the interrupt handler for missed event.	

**Table 26-16. Keyboard Controller Event Servicing (continued)**

Step	Register/Bit Field/Programming Model	Value
ELSIF: <a href="#">KBD_IRQSTATUS[2]</a> IT_TIMEOUT = 1	Start the interrupt handler for timeout event.	
ELSIF: <a href="#">KBD_IRQSTATUS[1]</a> IT_LONG_KEY = 1	Start the interrupt handler for long key pressed event.	
ELSIF: <a href="#">KBD_IRQSTATUS[0]</a> IT_EVENT = 1	Start the interrupt handler for pressed key event.	
ENDIF		
Wait for the execution of the corresponding interrupt handler.	Interrupt controller of the MPU	
Clear the corresponding IRQ status register bits by writing of 1 there.	<a href="#">KBD_IRQSTATUS</a>	0x0000000-
Disable the corresponding sources of Interrupt if needed by writing of 0 at the corresponding bit fields.	<a href="#">KBD_IRQENABLE</a> or <a href="#">KBD_WAKEUPENABLE</a>	0x0000000-

## 26.6 Keyboard Controller Register Manual

### 26.6.1 Keyboard Controller Instance Summary

Table 26-17 summarizes the Keyboard Controller instances.

**Table 26-17. Keyboard Controller Instance Summary**

Module Name	Base Address	Size
Keyboard controller	0x4A31 C000	4KB

### 26.6.2 Keyboard Controller Registers

#### 26.6.2.1 Keyboard Controller Register Summary

Table 26-18 summarizes the keyboard controller register mapping.

**Table 26-18. Keyboard Controller Register Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset	Keyboard Controller L4-WKUP Base Address
KBD_REVISION	R	32	0x0000 0000	0x4A31 C000
KBD_SYSCONFIG	RW	32	0x0000 0010	0x4A31 C010
KBD_SYSSTATUS	R	32	0x0000 0014	0x4A31 C014
KBD_IRQSTATUS	RW	32	0x0000 0018	0x4A31 C018
KBD_IRQENABLE	RW	32	0x0000 001C	0x4A31 C01C
KBD_WAKEUPENABLE	RW	32	0x0000 0020	0x4A31 C020
KBD_PENDING	R	32	0x0000 0024	0x4A31 C024
KBD_CTRL	RW	32	0x0000 0028	0x4A31 C028
KBD_DEBOUNCINGTIME	RW	32	0x0000 002C	0x4A31 C02C
KBD_KEYLONGTIME	RW	32	0x0000 0030	0x4A31 C030
KBD_TIMEOUT	RW	32	0x0000 0034	0x4A31 C034
KBD_STATEMACHINE	R	32	0x0000 0038	0x4A31 C038
KBD_ROWINPUTS	R	32	0x0000 003C	0x4A31 C03C
KBD_COLUMNOUTPUTS	RW	32	0x0000 0040	0x4A31 C040
KBD_FULLCODE31_0	R	32	0x0000 0044	0x4A31 C044
KBD_FULLCODE63_32	R	32	0x0000 0048	0x4A31 C048
KBD_FULLCODE17_0	R	32	0x0000 004C	0x4A31 C04C
KBD_FULLCODE35_18	R	32	0x0000 0050	0x4A31 C050
KBD_FULLCODE53_36	R	32	0x0000 0054	0x4A31 C054
KBD_FULLCODE71_54	R	32	0x0000 0058	0x4A31 C058
KBD_FULLCODE80_72	R	32	0x0000 005C	0x4A31 C05C

#### 26.6.2.2 Keyboard Controller Register Description

Table 26-19 through Table 26-59 describe the individual keyboard module registers.

**Table 26-19. KBD\_REVISION**

<b>Address Offset</b>	0x0000 0000	
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .	<b>Instance</b> Keyboard Controller
<b>Description</b>	This register contains the IP revision code.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x----- <sup>(1)</sup>

<sup>(1)</sup> TI internal data

**Table 26-20. Register Call Summary for Register KBD\_REVISION**

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[0\]](#)

**Table 26-21. KBD\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010	
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .	<b>Instance</b> Keyboard Controller
<b>Description</b>	This register controls the various parameters of the L4-Interface.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CLOCKACTIVITY	RESERVED	EMUFREE	IDLEMODE	ENAWAKEUP	SOFTRESET	AUTOGATING									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reads return 0.	RW	0x000000
9:8	CLOCKACTIVITY	<p>Clock(s) activity during wake-up mode period.</p> <p>0x0: Functional clock can be switched-off; L4-Interface clock can be switched-off.</p> <p>0x1: Functional clock can be switched-off; L4-Interface clock is maintained during wake-up period.</p> <p>0x2: Functional clock is maintained during wake-up period; L4-Interface clock can be switched-off.</p> <p>0x3: Functional clock is maintained during wake-up period; L4-Interface clock is maintained during wake-up period.</p>	RW	0x0
7:6	RESERVED	Write 0s for future compatibility. Reads return 0.	RW	0x0
5	EMUFREE	<p>Emulation mode</p> <p>0x0: The keyboard module is frozen in emulation mode (PINSUSPENDN signal active).</p> <p>0x1: The keyboard module runs free, regardless of PINSUSPENDN value.</p>	RW	0

Bits	Field Name	Description	Type	Reset
4:3	IDLEMODE	Power management, req/ack control 0x0: Force-idle. An idle request is acknowledged unconditionally. 0x1: No-idle. An idle request is never acknowledged. 0x2: Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module. 0x3: Reserved. Do not use.	RW	0x0
2	ENAWAKEUP	Wake-up feature global control 0x0: No wake-up line assertion in idle mode 0x1: Wake-up line assertion enabled in smart-idle mode	RW	0
1	SOFTRESET	Software reset. This bit is automatically reset by the hardware. During reads, it always return 0. 0x0: Normal mode 0x1: The module is reset.	RW	0
0	AUTOGATING	Internal L4-Interface clock gating strategy 0x0: L4-Interface clock is free-running. 0x1: Automatic L4-Interface clock gating strategy is applied, based on the L4-Interface activity.	RW	1

**Table 26-22. Register Call Summary for Register KBD\_SYSCONFIG**

Keyboard Controller Functional Description

- [Keyboard Controller Software Reset: \[0\] \[1\]](#)
- [Keyboard Controller Power Management: \[2\] \[3\] \[4\] \[5\] \[6\]](#)

Keyboard Controller Programming Guide

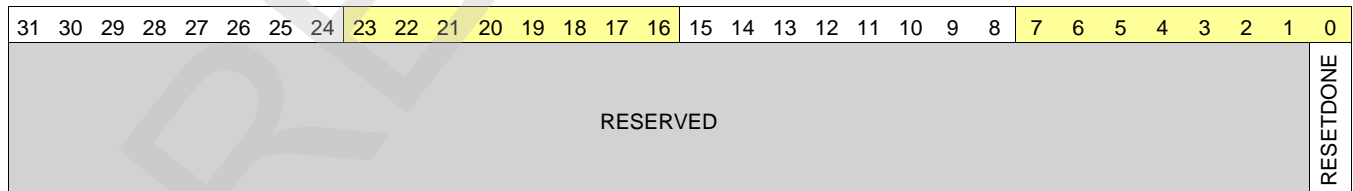
- [Main Sequence - Keyboard Global Initialization: \[7\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[8\]](#)

**Table 26-23. KBD\_SYSSTATUS**

<b>Address Offset</b>	0x0000 0014	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	This register controls optional features specific to the timer functionality.		
<b>Type</b>	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0. Reserved for L4-Interface socket status information.	R	0x0000 0000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is on-going. Read 0x1: Reset completed	R	0

**Table 26-24. Register Call Summary for Register KBD\_SYSSTATUS**

Keyboard Controller Functional Description

- [Keyboard Controller Software Reset: \[0\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[1\]](#)

**Table 26-25. KBD\_IRQSTATUS**

<b>Address Offset</b>	0x0000 0018	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The keyboard interrupt-status register is used to determine which of the keyboard events requested an interrupt.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MISS_EVENT	IT_TIMEOUT	IT_LONG_KEY	IT_EVENT

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads return 0.	RW	0x00000000
3	MISS_EVENT	Indicates when a miss event occurs. 0x0: No miss event 0x1: A miss event occurs.	RW	0
2	IT_TIMEOUT	Indicates when a timeout event is detected. 0x0: No timeout event 0x1: A timeout event occurs.	RW	0
1	IT_LONG_KEY	Indicates when a long key event is detected. 0x0: No long key event 0x1: A long key event occurs.	RW	0
0	IT_EVENT	Indicates when an event is detected. 0x0: No event. 0x1: An event occurs.	RW	0

**Table 26-26. Register Call Summary for Register KBD\_IRQSTATUS**

Keyboard Controller Functional Description

- [Keyboard Controller Interrupt Requests: \[0\] \[1\] \[2\] \[3\]](#)
- [Functional Modes: \[4\]](#)
- [Interrupt-Generation Scheme: \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- [Keyboard Buffer and Missed Events \(Overrun Feature\): \[11\] \[12\]](#)

Keyboard Controller Programming Guide

- [Main Sequence - Keyboard Global Initialization: \[13\]](#)
- [Main Sequence - Keyboard Hardware Mode: \[14\] \[15\] \[16\] \[17\]](#)
- [Keyboard Events Servicing: \[18\] \[19\] \[20\] \[21\] \[22\] \[23\] \[24\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[25\]](#)

**Table 26-27. KBD\_IRQENABLE**

<b>Address Offset</b>	0x0000 001C		
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .	<b>Instance</b>	Keyboard Controller
<b>Description</b>	The keyboard Interrupt-enable register allows the user to enable certain keyboard event for generating an interrupt request.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IT_TIMEOUT_EN	IT_LONG_KEY_EN	IT_EVENT_EN													

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads return 0	RW	0x0000 0000
2	IT_TIMEOUT_EN	Timeout interrupt enable 0x0: Timeout interrupt disabled 0x1: Timeout interrupt enabled	RW	0
1	IT_LONG_KEY_EN	Long key interrupt enable 0x0: Long Key interrupt disabled 0x1: Long Key interrupt enabled	RW	0
0	IT_EVENT_EN	Enable event interrupt 0x0: Event interrupt disabled 0x1: Event interrupt enabled	RW	0

**Table 26-28. Register Call Summary for Register KBD\_IRQENABLE**

Keyboard Controller Functional Description

- [Keyboard Controller Interrupt Requests: \[0\] \[1\] \[2\]](#)
- [Functional Modes: \[3\]](#)
- [Interrupt-Generation Scheme: \[4\] \[5\]](#)

Keyboard Controller Programming Guide

- [Main Sequence - Keyboard Global Initialization: \[6\]](#)
- [Main Sequence - Keyboard Hardware Mode: \[7\]](#)
- [Main Sequence - Keyboard Software Mode: \[8\]](#)
- [Keyboard Events Servicing: \[9\] \[10\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[11\]](#)

**Table 26-29. KBD\_WAKEUPENABLE**

<b>Address Offset</b>	0x0000 0020	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The keyboard wake-up enable register allows the user to mask the expected source of wake-up event that generates a wake-up request.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WUP_TIMEOUT_ENA		WUP_LONG_KEY_ENA		WUP_EVENT_ENA											

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reads return 0.	RW	0x0000 0000
2	WUP_TIMEOUT_ENA	Timeout wake-up enable 0x0: Timeout wake-up generation disabled 0x1: Timeout wake-up generation enabled	RW	0
1	WUP_LONG_KEY_ENA	Long key wake-up enable 0x0: Long key wake-up generation disabled 0x1: Long key wake-up generation enabled	RW	0
0	WUP_EVENT_ENA	Event wake-up enable 0x0: Event wake-up generation disabled 0x1: Event wake-up generation enabled	RW	0

**Table 26-30. Register Call Summary for Register KBD\_WAKEUPENABLE**

Keyboard Controller Functional Description

- [Keyboard Controller Power Management: \[0\]](#)

Keyboard Controller Programming Guide

- [Main Sequence - Keyboard Global Initialization: \[1\]](#)
- [Main Sequence - Keyboard Hardware Mode: \[2\]](#)
- [Main Sequence - Keyboard Software Mode: \[3\]](#)
- [Keyboard Events Servicing: \[4\] \[5\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[6\]](#)

**Table 26-31. KBD\_PENDING**

<b>Address Offset</b>	0x0000 0024	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The software must read the pending write bits to insure that following write access is not discarded due to ongoing write synchronization process.		
<b>Type</b>	R		



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												PEND_TIMEOUT	PEND_LONG_KEY	PEND_DEBOUNCING	PEND_CTRL

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads return 0.	R	0x00000000
3	PEND_TIMEOUT	Write-pending-bit for <a href="#">KBD_TIMEOUT</a> register Read 0x0: No write pending to the <a href="#">KBD_TIMEOUT</a> register Read 0x1: A write is pending to the <a href="#">KBD_TIMEOUT</a> register	R	0
2	PEND_LONG_KEY	Write-pending-bit for <a href="#">KBD_KEYLONGTIME</a> register Read 0x0: No write pending to the <a href="#">KBD_KEYLONGTIME</a> register. Read 0x1: A write is pending to the <a href="#">KBD_KEYLONGTIME</a> register	R	0
1	PEND_DEBOUNCING	Write-pending-bit for <a href="#">KBD_DEBOUNCINGTIME</a> register Read 0x0: No write pending to the <a href="#">KBD_DEBOUNCINGTIME</a> register Read 0x1: A write is pending to the <a href="#">KBD_DEBOUNCINGTIME</a> register	R	0
0	PEND_CTRL	Write pending-bit for <a href="#">KBD_CTRL</a> register Read 0x0: No write pending to the <a href="#">KBD_CTRL</a> register Read 0x1: A write is pending to the <a href="#">KBD_CTRL</a> register	R	0

**Table 26-32. Register Call Summary for Register KBD\_PENDING**

Keyboard Controller Functional Description

- [Write Registers Access: \[0\] \[1\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[2\]](#)

**Table 26-33. KBD\_CTRL**

<b>Address Offset</b>	0x0000 0028	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	This register sets the functional configuration of the module.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
RESERVED																												REPEAT_MODE	TIMEOUT_LONG_KEY	TIMEOUT_EMPTY	LONG_KEY	PTV	NSOFTWARE_MODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reads return 0.	RW	0x000000
8	REPEAT_MODE	Repeat mode enable 0x0: Repeat mode detection disabled. 0x1: Repeat mode detection enabled	RW	0
7	TIMEOUT_LONG_KEY	Timeout long key mode enable 0x0: Timeout long key mode disabled 0x1: Timeout long key mode enabled	RW	0
6	TIMEOUT_EMPTY	Timeout empty mode enable 0x0: Timeout long key mode disabled 0x1: Timeout long key mode enabled	RW	0
5	LONG_KEY	Long key mode enable. 0x0: Long key mode disabled 0x1: Long key mode enabled	RW	0
4:2	PTV	Pre-scale clock timer value	RW	0x7
1	NSOFTWARE_MODE	Select hardware or software mode for key decoding. 0x0: Enable software mode. 0x1: Enable hardware decoding using internal sequencer.	RW	1
0	RESERVED	Reads return 0.	RW	0

**Table 26-34. Register Call Summary for Register KBD\_CTRL**

## Keyboard Controller Functional Description

- [Keyboard Controller Software Mode: \[0\]](#)
- [Functional Modes: \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\]](#)
- [Keyboard Controller Timer: \[8\] \[9\] \[10\] \[11\] \[12\]](#)

## Keyboard Controller Programming Guide

- [Main Sequence - Keyboard Global Initialization: \[13\]](#)
- [Main Sequence - Keyboard Hardware Mode: \[14\] \[15\]](#)
- [Main Sequence - Keyboard Software Mode: \[16\]](#)

## Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[17\]](#)
- [Keyboard Controller Register Description: \[18\] \[19\] \[20\]](#)

**Table 26-35. KBD\_DEBOUNCINGTIME**

<b>Address Offset</b>	0x0000 002C																																																													
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .	<b>Instance</b> Keyboard Controller																																																												
<b>Description</b>	This register is used to filter glitches on the press key or release key.																																																													
<b>Type</b>	RW																																																													
<table border="1" style="width:100%; text-align:center;"> <tr> <td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td> <td style="background-color:yellow;">23</td><td style="background-color:yellow;">22</td><td style="background-color:yellow;">21</td><td style="background-color:yellow;">20</td><td style="background-color:yellow;">19</td><td style="background-color:yellow;">18</td><td style="background-color:yellow;">17</td><td style="background-color:yellow;">16</td> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td> <td style="background-color:yellow;">7</td><td style="background-color:yellow;">6</td><td style="background-color:yellow;">5</td><td style="background-color:yellow;">4</td><td style="background-color:yellow;">3</td><td style="background-color:yellow;">2</td><td style="background-color:yellow;">1</td><td style="background-color:yellow;">0</td> </tr> <tr> <td colspan="16">RESERVED</td> <td colspan="12">DEBOUNCING_VALUE</td> </tr> </table>			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RESERVED																DEBOUNCING_VALUE											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																															
RESERVED																DEBOUNCING_VALUE																																														
<b>Bits</b>	<b>Field Name</b>	<b>Description</b>	<b>Type</b>	<b>Reset</b>																																																										
31:6	RESERVED	Reads return 0.	RW	0x00000000																																																										
5:0	DEBOUNCING_VALUE	This value corresponds to the desired value of debouncing time.	RW	0x00																																																										

**Table 26-36. Register Call Summary for Register KBD\_DEBOUNCINGTIME**

- Keyboard Controller Functional Description
- [Keyboard Controller Timer: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\]](#)
- Keyboard Controller Programming Guide
- [Main Sequence - Keyboard Global Initialization: \[6\]](#)
  - [Main Sequence - Keyboard Hardware Mode: \[7\]](#)
- Keyboard Controller Register Manual
- [Keyboard Controller Register Summary: \[8\]](#)
  - [Keyboard Controller Register Description: \[9\] \[10\] \[11\]](#)

**Table 26-37. KBD\_KEYLONGTIME**

<b>Address Offset</b>	0x0000 0030	
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .	<b>Instance</b> Keyboard Controller
<b>Description</b>	This register is used to measure duration of a key press to allow shortcut detection.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LONG_KEY_VALUE															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reads return 0.	RW	0x00000
11:0	LONG_KEY_VALUE	This value corresponds to the desired value of the long key interrupt or repeat mode value.	RW	0x000

**Table 26-38. Register Call Summary for Register KBD\_KEYLONGTIME**

- Keyboard Controller Functional Description
- [Keyboard Controller Timer: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\]](#)
- Keyboard Controller Programming Guide
- [Main Sequence - Keyboard Global Initialization: \[9\]](#)
  - [Main Sequence - Keyboard Hardware Mode: \[10\]](#)
- Keyboard Controller Register Manual
- [Keyboard Controller Register Summary: \[11\]](#)
  - [Keyboard Controller Register Description: \[12\] \[13\] \[14\]](#)

**Table 26-39. KBD\_TIMEOUT**

<b>Address Offset</b>	0x0000 0034	
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .	<b>Instance</b> Keyboard Controller
<b>Description</b>	This register is used to detect a long inactivity on the keyboard.	
<b>Type</b>	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TIMEOUT_VALUE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reads return 0.	RW	0x0000
15:0	TIMEOUT_VALUE	This value corresponds to the desired value of the time-out interrupt.	RW	0x0000

**Table 26-40. Register Call Summary for Register KBD\_TIMEOUT**

Keyboard Controller Functional Description

- [Keyboard Controller Timer: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\]](#)

Keyboard Controller Programming Guide

- [Main Sequence - Keyboard Global Initialization: \[7\]](#)
- [Main Sequence - Keyboard Hardware Mode: \[8\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[9\]](#)
- [Keyboard Controller Register Description: \[10\] \[11\] \[12\]](#)

**Table 26-41. KBD\_STATEMACHINE**

<b>Address Offset</b>	0x0000 0038	
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .	<b>Instance</b> Keyboard Controller
<b>Description</b>	This register indicates the state of the sequencer.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STATE_MACHINE															

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reads return 0	R	0x00000000
3:0	STATE_MACHINE	The state of internal state machine. See <a href="#">Section 26.5.1.2.4, State-Machine Status Register</a> , for details.	R	0x0

**Table 26-42. Register Call Summary for Register KBD\_STATEMACHINE**

Keyboard Controller Functional Description

- [State-Machine Status: \[0\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[1\]](#)

**Table 26-43. KBD\_ROWINPUTS**

<b>Address Offset</b>	0x0000 003C	
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .	<b>Instance</b> Keyboard Controller
<b>Description</b>	This register stores the value of the row inputs.	
<b>Type</b>	R	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																KBR_LATCH															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reads return 0.	R	0x00000000
8:0	KBR_LATCH	The value of the rows input.	R	0x00

**Table 26-44. Register Call Summary for Register KBD\_ROWINPUTS**

- Keyboard Controller Functional Description
- [Keyboard Controller Software Mode: \[0\] \[1\]](#)
- Keyboard Controller Programming Guide
- [Main Sequence - Keyboard Software Mode: \[2\] \[3\]](#)
- Keyboard Controller Register Manual
- [Keyboard Controller Register Summary: \[4\]](#)

**Table 26-45. KBD\_COLUMNOUTPUTS**

<b>Address Offset</b>	0x0000 0040	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	This register holds the value of the columns output.		
<b>Type</b>	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																KBC_REG															

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reads return 0.	RW	0x000000
8:0	KBC_REG	The value of the columns output.	RW	0x00

**Table 26-46. Register Call Summary for Register KBD\_COLUMNOUTPUTS**

- Keyboard Controller Functional Description
- [Keyboard Controller Block Diagram: \[0\]](#)
  - [Keyboard Controller Software Mode: \[1\]](#)
- Keyboard Controller Programming Guide
- [Main Sequence - Keyboard Global Initialization: \[2\]](#)
  - [Main Sequence - Keyboard Software Mode: \[3\] \[4\]](#)
- Keyboard Controller Register Manual
- [Keyboard Controller Register Summary: \[5\]](#)

**Table 26-47. KBD\_FULLCODE31\_0**

<b>Address Offset</b>	0x0000 0044	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The <a href="#">KBD_FULLCODE31_0</a> register codes the row 0, row 1, row 2, and row 3.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FULL_CODE_31_0																															

Bits	Field Name	Description	Type	Reset
31:0	FULL_CODE_31_0	A bit at one indicates that the corresponding key is pressed.	R	0x0000 0000

**Table 26-48. Register Call Summary for Register KBD\_FULLCODE31\_0**

- Keyboard Controller Functional Description
- [Keyboard Controller Key Coding Registers: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)
- Keyboard Controller Programming Guide
- [Main Sequence - Keyboard Hardware Mode: \[11\] \[12\]](#)

**Table 26-48. Register Call Summary for Register KBD\_FULLCODE31\_0 (continued)**

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[13\]](#)
- [Keyboard Controller Register Description: \[14\]](#)

**Table 26-49. KBD\_FULLCODE63\_32**

<b>Address Offset</b>	0x0000 0048	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The <a href="#">KBD_FULLCODE63_32</a> register codes rows 4, 5, 6, and 7.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FULL_CODE_63_32																															

Bits	Field Name	Description	Type	Reset
31:0	FULL_CODE_63_32	A bit at one indicates that the corresponding key is pressed.	R	0x0000 0000

**Table 26-50. Register Call Summary for Register KBD\_FULLCODE63\_32**

Keyboard Controller Functional Description

- [Keyboard Controller Key Coding Registers: \[0\] \[1\] \[2\] \[3\] \[4\] \[5\] \[6\] \[7\] \[8\] \[9\] \[10\]](#)

Keyboard Controller Programming Guide

- [Main Sequence - Keyboard Hardware Mode: \[11\] \[12\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[13\]](#)
- [Keyboard Controller Register Description: \[14\]](#)

**Table 26-51. KBD\_FULLCODE17\_0**

<b>Address Offset</b>	0x0000 004C	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The <a href="#">KBD_FULLCODE17_0</a> register codes the row 0 and row 1. The row 0 is coded between bit 0 and 8, the row 1 is coded between bit 24 and 16.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ROW1								RESERVED								ROW0							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved. Read returns 0.	RO	0x00
24:16	ROW1	A bit at 1 indicates that the corresponding key is pressed.	R	0x000
15:9	RESERVED	Reserved	R	0x00
8:0	ROW0	A bit at 1 indicates that the corresponding key is pressed.	R	0x000

**Table 26-52. Register Call Summary for Register KBD\_FULLCODE17\_0**

Keyboard Controller Functional Description

- [Keyboard Controller Key Coding Registers: \[0\] \[1\] \[2\]](#)

Keyboard Controller Programming Guide

- [Main Sequence - Keyboard Hardware Mode: \[3\] \[4\]](#)

**Table 26-52. Register Call Summary for Register KBD\_FULLCODE17\_0 (continued)**

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[5\]](#)
- [Keyboard Controller Register Description: \[6\]](#)

**Table 26-53. KBD\_FULLCODE35\_18**

<b>Address Offset</b>	0x0000 0050	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The <a href="#">KBD_FULLCODE35_18</a> register codes the row 2 and row 3. The row 2 is coded between bit 0 and 8, the row 3 is coded between bit 24 and 16		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ROW3								RESERVED								ROW2							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved. Read returns 0.	RO	0x00
24:16	ROW3	A bit at 1 indicates that the corresponding key is pressed.	R	0x000
15:9	RESERVED	Reserved	R	0x00
8:0	ROW2	A bit at 1 indicates that the corresponding key is pressed.	R	0x000

**Table 26-54. Register Call Summary for Register KBD\_FULLCODE35\_18**

Keyboard Controller Functional Description

- [Keyboard Controller Key Coding Registers: \[0\]](#)

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[1\]](#)
- [Keyboard Controller Register Description: \[2\]](#)

**Table 26-55. KBD\_FULLCODE53\_36**

<b>Address Offset</b>	0x0000 0054	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The <a href="#">KBD_FULLCODE53_36</a> register codes the row 4 and row 5. The row 4 is coded between bit 0 and 8, the row 5 is coded between bit 24 and 16.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ROW5								RESERVED								ROW4							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved. Read returns 0.	RO	0x00
24:16	ROW5	A bit at 1 indicates that the corresponding key is pressed.	R	0x000
15:9	RESERVED	Reserved	R	0x00
8:0	ROW4	A bit at 1 indicates that the corresponding key is pressed.	R	0x000

**Table 26-56. Register Call Summary for Register KBD\_FULLLCODE53\_36**

Keyboard Controller Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">Keyboard Controller Key Coding Registers: [0]</a></li> </ul>
Keyboard Controller Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">Keyboard Controller Register Summary: [1]</a></li> <li>• <a href="#">Keyboard Controller Register Description: [2]</a></li> </ul>

**Table 26-57. KBD\_FULLLCODE71\_54**

<b>Address Offset</b>	0x0000 0058	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The <a href="#">KBD_FULLLCODE71_54</a> register codes the row 6 and row 7. The row 6 is coded between bit 0 and 8, the row 7 is coded between bit 24 and 16.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ROW7								RESERVED								ROW6							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved. Read returns 0.	RO	0x00
24:16	ROW7	A bit at 1 indicates that the corresponding key is pressed.	R	0x000
15:9	RESERVED	Reserved	R	0x00
8:0	ROW6	A bit at 1 indicates that the corresponding key is pressed.	R	0x000

**Table 26-58. Register Call Summary for Register KBD\_FULLLCODE71\_54**

Keyboard Controller Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">Keyboard Controller Key Coding Registers: [0] [1]</a></li> </ul>
Keyboard Controller Programming Guide
<ul style="list-style-type: none"> <li>• <a href="#">Main Sequence - Keyboard Hardware Mode: [2]</a></li> </ul>
Keyboard Controller Register Manual
<ul style="list-style-type: none"> <li>• <a href="#">Keyboard Controller Register Summary: [3]</a></li> <li>• <a href="#">Keyboard Controller Register Description: [4]</a></li> </ul>

**Table 26-59. KBD\_FULLLCODE80\_72**

<b>Address Offset</b>	0x0000 005C	<b>Instance</b>	Keyboard Controller
<b>Physical Address</b>	See <a href="#">Table 26-18</a> .		
<b>Description</b>	The <a href="#">KBD_FULLLCODE80_72</a> register codes the row 8. The row 8 is coded between bit 0 and 8.		
<b>Type</b>	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ROW8							

Bits	Field Name	Description	Type	Reset
31: 9	RESERVED	Reserved. Read returns 0.	RO	0x000000
8:0	ROW8	A bit at 1 indicates that the corresponding key is pressed.	R	0x000

**Table 26-60. Register Call Summary for Register KBD\_FULLLCODE80\_72**

Keyboard Controller Functional Description
<ul style="list-style-type: none"> <li>• <a href="#">Keyboard Controller Key Coding Registers: [0] [1] [2] [3] [4]</a></li> </ul>
Keyboard Controller Programming Guide
<ul style="list-style-type: none"> <li>• <a href="#">Main Sequence - Keyboard Hardware Mode: [5] [6]</a></li> </ul>



---

**Table 26-60. Register Call Summary for Register KBD\_FULLCODE80\_72 (continued)**

---

Keyboard Controller Register Manual

- [Keyboard Controller Register Summary: \[7\]](#)
  - [Keyboard Controller Register Description: \[8\]](#)
-



## Initialization

This chapter introduces the steps in the general-purpose (GP) device initialization.

Topic	Page
<b>27.1 Initialization Overview</b> .....	<b>5180</b>
<b>27.2 Preinitialization</b> .....	<b>5182</b>
<b>27.3 Reset Sequences</b> .....	<b>5197</b>
<b>27.4 Device Initialization by ROM Code</b> .....	<b>5198</b>
<b>27.5 Services for HLOS Support</b> .....	<b>5264</b>

## 27.1 Initialization Overview

This chapter provides an overview of the requirements for initializing the device from power-on to OS/application execution. It provides an overview of the overall initialization process, including hardware- and software-related steps, a general overview of the ROM code operational requirements, and the behavior expectations.

### 27.1.1 Terminology

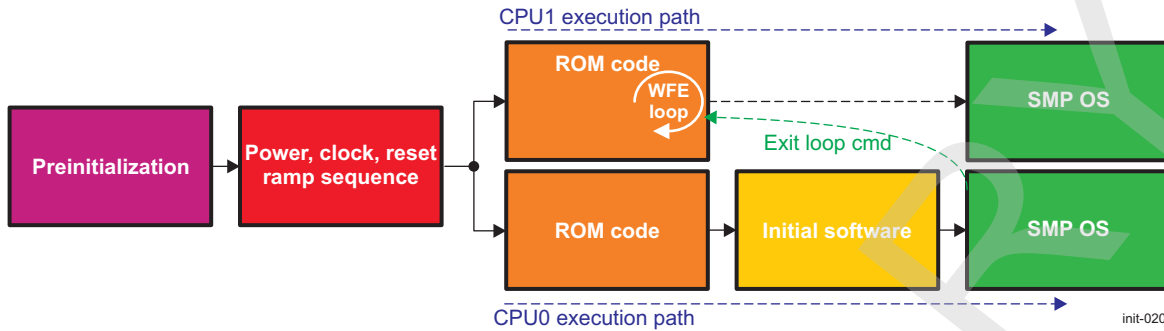
- **Bootstrap:** Initial software launched by the ROM code during the memory booting phase
- **Configuration Header (CH):** Optional structure, that precedes the initial software and allows the redefinition of the ROM code default settings.
- **Downloaded software:** Initial software downloaded into the internal static RAM (SRAM) by the ROM code during the peripheral booting phase
- **eFuse:** A one-time programmable memory location usually set at the factory
- **Flash loader:** Downloaded software launched by the ROM code in preflashing. It also programs an image in external memories.
- **GP device:** General-purpose device
- **Initial software:** Software executed by any of the ROM code mechanisms (memory booting or peripheral booting). Initial software is a generic term for bootstrap and downloaded software.
- **Memory booting:** ROM code mechanism that consists of executing initial software from external memory
- **Master CPU:** The ARM Cortex™-A9 MPCore™ CPU for which CPU-ID is 0. It configures the multicore platform and starts the ROM code to ensure device booting from a mass storage memory (memory booting) or a peripheral interface (peripheral booting).
- **Peripheral booting:** ROM code mechanism that consists of polling selected interfaces, downloading, and executing initial software (in this case, downloaded software) in the internal RAM.
- **Permanent booting device:** Memory device containing, by default, the image to be executed during the booting sequence. It is the default memory booting device. The permanent booting device is used after warm reset if no software booting configuration is programmed.
- **Preflashing:** A specific case of peripheral booting where the ROM code mechanism is used to program the external flash memory
- **ROM Code:** The on-chip software in OMAP™ ROM that implements booting
- **Save-And-Restore (SAR RAM) memory:** On-chip RAM memory that is not cleared after warm resets or wake-ups from low-power modes
- **Slave CPU:** The ARM Cortex-A9 MPCore CPU for which CPU-ID is 1. It is brought to the wait-for-event (WFE) state by the ROM code, waiting to be woken up by the master CPU.
- **Software booting configuration:** A logical structure stored in the SAR memory that allows the redefinition of the ROM code default settings when booting after a warm reset

### 27.1.2 Initialization Process

Figure 27-1 is an overview of the initialization process and its steps:

- **Preinitialization:** Power, clock, and control connections must be present, and the boot configuration pins must be held at the desired logical levels.
- **Power/clock/reset ramp sequence:** Specific sequence that is applied by the power-management chip
- **ROM code:** Responsible for finding/downloading and for executing the initial software by using the master CPU
- **Initial software:** Software that prepares and passes control to application software or to the high-level operating system (HLOS)
- **Symmetric multiprocessing (SMP)-capable HLOS** or application (primarily for diagnostics)

Figure 27-1. Initialization Process



The first two steps in the initialization process are hardware-oriented; however, they require understanding of the process of configuring these system interface pins (balls on the device), which have software-configurable functionality. This configuration is an essential part of the chip configuration and is application-dependent. This chapter discusses these pins and the associated configuration registers that are vital to the correct initialization of the device.

## 27.2 Preinitialization

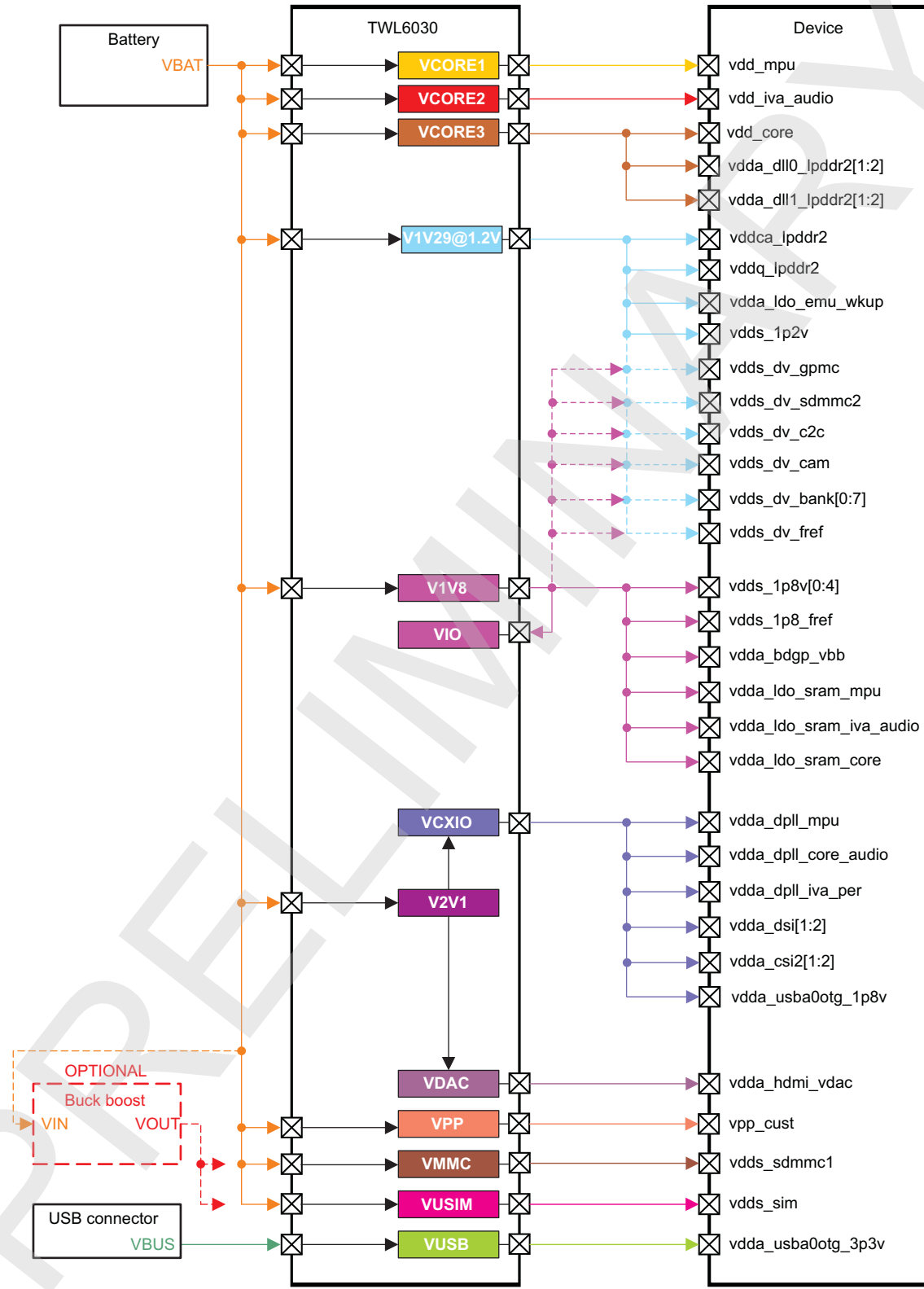
To accomplish a successful boot-up operation with a general-purpose (GP) device, certain hardware configuration settings must be in place. Clock, reset, and power connections, as well as pins involved in setting the boot memory space for the master CPU, must be connected and driven correctly for successful initialization of the device. The following sections describe the specific requirements for the preinitialization stage.

### 27.2.1 Power Requirements

The device can be supplied by an external power-management integrated circuit (IC). TI provides a global solution with the device connected to the TWL6030 power-management/TWL6040 audio codec IC.

[Figure 27-2](#) shows the power connections between the OMAP device and the TWL6030 power IC.

Figure 27-2. Power Supply Connections



init-002

**NOTE:** Figure 27-2 is an example of power connections between the device and the TWL6030. These connections depend on the actual application.

Table 27-1 describes the device power pins.

**Table 27-1. OMAP Power Pins <sup>(1)</sup>**

Voltage Pin Name	Domain
vdd_core	<ul style="list-style-type: none"> <li>Core</li> </ul>
vdd_mpu	<ul style="list-style-type: none"> <li>Dual MPU core</li> <li>NEON</li> </ul>
vdd_iva_audio	<ul style="list-style-type: none"> <li>IVA HD</li> <li>DSP</li> <li>Audio back-end (ABE)</li> <li>Wake-up logic</li> </ul>
vdda_ldo_emu_wkup	<ul style="list-style-type: none"> <li>Wakeup</li> <li>Emulation subsystem</li> </ul>
vddq_lpddr2	LPDDR2 DQ I/Os (OMAP side only)
vddca_lpddr2	LPDDR2 CA I/Os (OMAP side only)
vddq_vref_lpddr21 vddq_vref_lpddr21	Supplies VDDQ_VREF generator
vddca_vref_lpddr21 vddca_vref_lpddr21	Supplies VDDCA_VREF generator
vdda_dli0_lpddr21 vdda_dli0_lpddr22 vdda_dli1_lpddr21 vdda_dli1_lpddr22	<ul style="list-style-type: none"> <li>LPDDR21 DLL</li> <li>LPDDR22 DLL</li> </ul>
vdds_1p8v[0:4]	1.8-V I/O: <ul style="list-style-type: none"> <li>HDMI CEC/DDC/HPD</li> <li>I2C2 and HDQ</li> <li>Sys</li> <li>JTAG</li> <li>DPM</li> </ul>
vdds_1p2v	1.2-V I/O: USB3/4_HSIC
vdda_dsi1	DSI1 I/Os + ADPLL
vdda_dsi2	DSI2 I/Os + ADPLL
vdda_csi21	CSI21 I/Os
vdda_csi22	CSI22 I/Os
vdda_hdmi_vdac	HDMI PHY + ADPLL or TV-out video DAC
vdda_usba_otg_1p8v	USB PHY common module + ADPLL
vdda_usba0_otg_3p3v	USB PHY core module
vpp_cust	Customer eFuse program module
vdds_sim	SIM I/Os
vdds_sdmmc1	SD/MMC1 I/Os
vdda_dppll_iva_per	<ul style="list-style-type: none"> <li>ADPLL PER and ADPLL IVA</li> </ul>
vdda_dppll_core_audio	<ul style="list-style-type: none"> <li>ADPLL CORE and ADPLL ABE</li> </ul>
vdda_dppll_mpu	<ul style="list-style-type: none"> <li>DPLL MPU</li> </ul>
vdda_ldo_sram_mpu	Internal MPU SRAM LDO
vdda_ldo_sram_iva_audio	Internal IVA and ABE SRAM LDOs
vdda_ldo_sram_core	Internal Core SRAM LDO

<sup>(1)</sup> The package-on-package (POP) device provides feedthroughs from the bottom of the package to the POP interface. Among these feedthroughs (FEEDTHROUGH balls), several provide power to the top memory device. The correct power supply to the feedthroughs must be provided based on memory requirements.



**Table 27-1. OMAP Power Pins <sup>(1)</sup> (continued)**

Voltage Pin Name	Domain
vdda_bdgp_vbb	Bandgap and Bodybias LDO
vdds_dv_gpmc	GPMC I/Os
vdds_dv_sdmmc2	SD/MMC2 I/Os
vdds_dv_cam	Camera I/Os: <ul style="list-style-type: none"> <li>• fref_clk1_out/fref_clk2_out</li> <li>• strobe /shutter/global reset</li> <li>• I2C3</li> </ul>
vdds_dv_bank0	Bank 0 of I/Os: usb3_ulpitll, HSI, McBSP4/3 I/Os (modem connectivity)
vdds_dv_bank1	Bank 1 of I/Os: MCBSP1/2, UART2 I/Os (typically used for BT/GPS connectivity)
vdds_dv_bank2	Bank 2 of I/Os: MCPDM, DMIC, I2C1, SmartReflex™ I/Os (TWL6030/TWL6040 connectivity)
vdds_dv_bank3	Bank 3 of I/Os: MCSP11 I/Os (low bandwidth peripheral connectivity)
vdds_dv_bank4	Bank 4 of I/Os: SDMMC5 I/Os
vdds_dv_bank5	Bank 5 of I/Os: I2C4, MCSP14, UART4 I/Os
vdds_dv_bank6	Bank 6 of I/Os: USB4 ULPI TLL, SDMMC3, SDMMC4 I/Os
vdds_dv_bank7	Bank 7 of I/Os: KEYBOARD I/Os
vdds_dv_c2c	Chip-to-chip (C2C) I/Os
vdds_dv_fref	fref dual voltage I/Os: fref_clk[3:4]_out/fref_clk[3:4]_req
vdds_1p8_fref	Oscillator, slicer, fref_clk0_out, and fref_clk_ioreq

For more information about power management, see [Section 3.7 Power Management Functional Description](#), in [Chapter 3, Power, Reset, and Clock Management](#).

The OMAP4430 power supply structure is based on the following constraints:

- Because most of the I/Os must be powered before the core, they are supplied on a specific rail (1.2 or 1.8 V).
- MPU, IVA/ABE, and Core are mapped on dedicated SMPS to allow independent SmartReflex dynamic frequency and voltage scaling for each of them.
- Because the DPLL, DSI, CSI, and USB OTG interfaces are noise-sensitive power domains, they are supplied by a dedicated rail to avoid other power sinks from affecting their power rail.
- Because bandgap, BodyBias LDOs, and SRAM LDOs must be supplied at 1.8 V from the beginning of the power-up sequence up to the end of power down, they are supplied by V1V8. This also applies to WKUP LDO at 1.2 V.
- Video DAC, eFuse VPP, SDMMC, and USB I/Os have specific requirements (sequencing, dual voltage, noise immunity, etc.) that require each of them be supplied by a dedicated power source.

## 27.2.2 Access to Power-Management IC

The ROM code accesses the power-management chip for the following purposes:

- SD/MMC1 powering:
  - The VMMC LDO is switched on by the ROM code when booting from the SD/MMC1 interface.

---

**NOTE:** The power supplies required for SD/MMC interface 2 are not controlled by the ROM code. It is assumed that the power-management chip (or the system hardware) properly powers the SD/MMC module attached to this interface.

---

- USB peripheral booting (internal or external transceiver):
  - PrevStartOk mechanism

This mechanism provides the ability to bypass the USB connection if a specific flag is set in the TWL6030 RTC domain (retained when the system is switched off).
- USB peripheral booting (internal transceiver):

- USB transceiver powering  
The TWL6030 VUSB LDO is turned on by the ROM code when selecting boot from the internal transceiver.
- VBUS detection  
The VBUS power pad is localized on the TWL6030 power IC. As a result, the ROM code must read from a TWL6030 register to get the status reflecting the device attachment to the USB link.
- Gating the TWL6030 USB charging circuitry during a suspend scenario  
If this event occurs, the ROM code should detect a suspend event and gate the charge circuitry inside the TWL6030.

### 27.2.2.1 Power-Management IC Detection

When booting from an SD/MMC card or USB internal transceiver (and only for these two booting options) the ROM code attempts to access the power device through the inter-integrated circuit (I<sup>2</sup>C™). The first access is a read transaction to obtain the TWL6030 version register. If this read access is successful (that is, the I<sup>2</sup>C transaction is acknowledged), the ROM code considers that a power-management device is attached, assuming it is TWL6030. The version value read is not used in any computation or compared to any hardcoded value.

If the first read transaction fails (that is, the I<sup>2</sup>C transaction times out or is not acknowledged), the ROM code logs it into a flag and does not try any further transaction until the next reset cycle. In this case, the nondetection of the companion chip does not block the boot process. However, the MMC card and USB boot procedures do not work as defined in this document:

- The ROM code does not ensure that the SD/MMC1 interface is powered when accessing the memory contents.
- The PrevStartOk mechanism is not functional (the specific flag is not checked).
- The peripheral booting from the internal USB transceiver is not possible, because the VBUS state read transaction is not performed. The USB (internal) device is skipped, and ROM code proceeds to the next device in the booting list.

### 27.2.2.2 Summary of the I<sup>2</sup>C Transactions

Nothing prevents the use of another power companion device together with the OMAP4430. However, special care must be taken if considering the two previously mentioned booting options (MMC/SD1 or USB internal transceiver). As a workaround solution, the alternative companion device can match (or emulate) the same register set and functionality as per TWL6030 for the following registers:

- JTAGVERNUM: Power-management IC detection. There must be a register at this location; however, its value is not used further by ROM code.
- PHOENIX\_SEQ\_CFG: The PrevStartOk flag is checked when performing USB peripheral boot (internal or external transceiver). If the flag is set, the USB peripheral booting procedure is skipped. In any case the full register is cleared (write 0x00) after the read transaction.
- VMMC\_CFG\_VOLTAGE: For the sake of powering the SDRAM1 card cage, the VMMC\_CFG\_VOLTAGE and VMMC\_CFG\_STATE registers are configured to set the VMMC LDO to 3 V. The value 0x15 is written to the VMMC\_CFG\_VOLTAGE register (selects 3-V output).
- VMMC\_CFG\_STATE: The value 0xE1 is written to the VMMC\_CFG\_STATE register (applies to all register groups, LDO state switched on).
- VUSB\_CFG\_VOLTAGE: The USB OMAP internal transceiver must be powered with a 3.3-V source to ensure its correct functionality. Three registers are used for this purpose. The VUSB\_CFG\_VOLTAGE register is written the value 0x18 for selecting 3.3 V.
- MISC2: The MISC2 register is written the value 0x08 for selecting the VBUS source.
- VUSB\_CFG\_STATE: The VUSB\_CFG\_STATE register is written the value 0xE1 for switching the VUSB LDO on for all register groups.
- CONTROLLER\_STAT1: VBUS detection is performed by reading the VBUS\_DET bit from the CONTROLLER\_STAT1 register. If this bit is set it reflects that a USB cable is plugged.
- CHARGERUSB\_CTRL1: This register is written the value 0x80 (SUSPEND\_BOOT = 1) for gating the charge circuitry when getting a USB SUSPEND event. If this bit is not set, the USB boot is aborted.

The complete description of TWL6030 register map can be found in TWL6030 documentation.

**NOTE:** No specific action is performed regarding the TWL6030 primary watchdog feature. In particular, if it is active (configurable in EEPROM) in the companion device, a system reset may occur before ROM code completes. The elapse value for the reset must be appropriately tuned so that the ROM code is not interrupted before the full platform start up (that is, before the initial software can take a specific action).

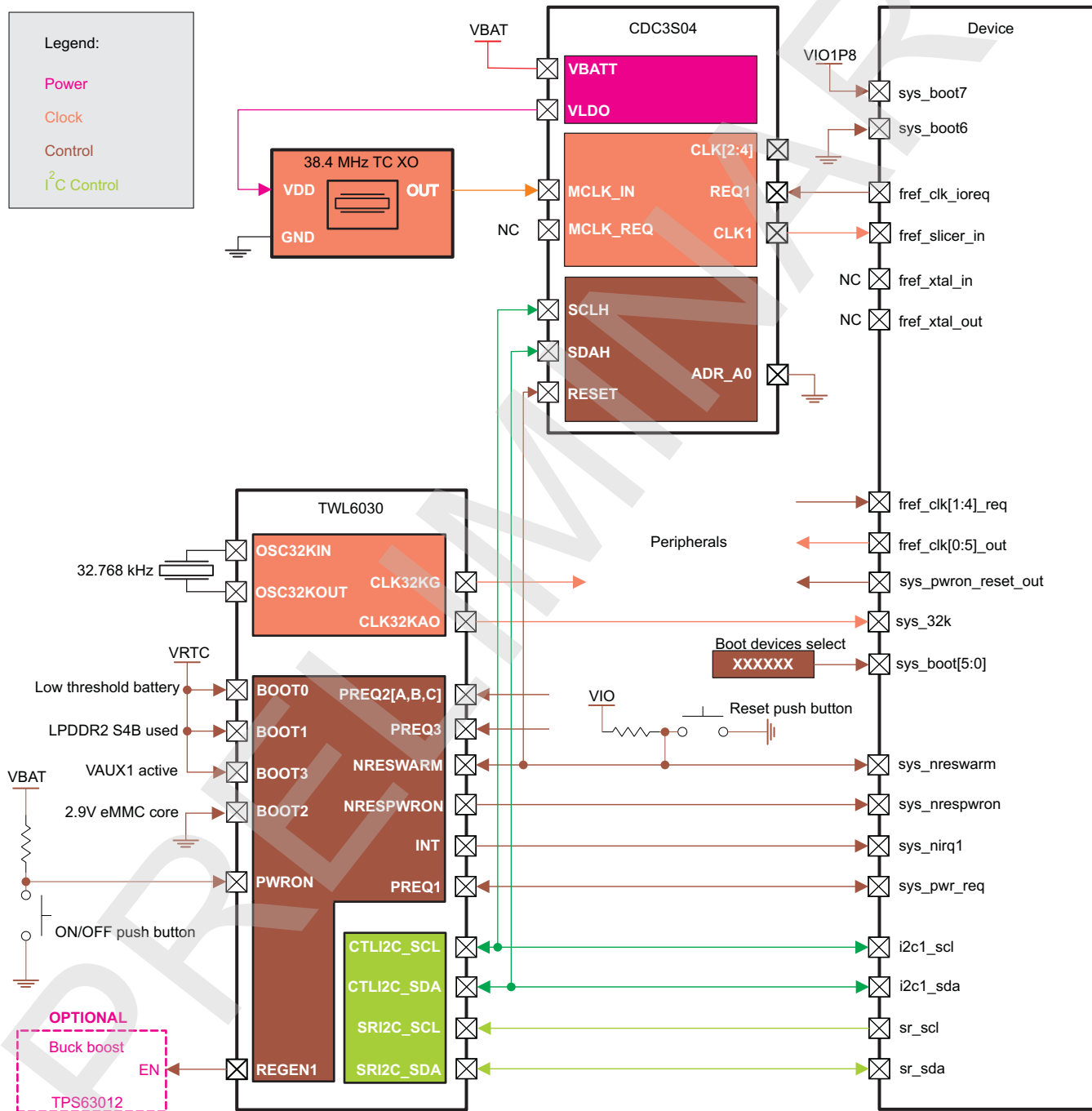
---

## 27.2.3 Clock, Reset, and Control

### 27.2.3.1 Overview

Figure 27-3 shows the clock and reset environment that gathers the clocks and reset-related signals at the system level, and the system-expansion signals.

Figure 27-3. Clock, Reset and Control Environment



init-004

The main features of the system interface are:

- A clock request output to an external clock source

- 12-, 16.8-, 19.2-, 26-, or 38.4-MHz reference clock input from external oscillator or clock supplied using internal crystal oscillator (up to 20 MHz)
- 32-kHz CMOS clock input
- Six configurable output clocks
- Eight input signals to define the boot mode and device system clock source
- Two reset sources
  - Power-on reset (POR) (cold reset)
  - Bidirectional warm reset

### 27.2.3.2 Clocking Scheme

#### 27.2.3.2.1 Required System Input Clocks

The device operation requires external input clocks, as follows:

- 32k-clock: A 32.768-kHz crystal is connected to the TWL6030 power device that embeds the 32k oscillator (a square CMOS clock can also be delivered on the 32k\_xtalin pin if the 32k crystal is connected on another device of the system). The resulting 32k-clock is delivered to the entire system on three outputs:
  - CLK32KAO\_is always on (meaning that when TWL6030 power VIO is on; that is, it is in WAIT-ON or ACTIVE state). It is used to deliver a 32k-clock to always-on features, such as the OMAP4430 and modem in the described system.
  - CLK32KG is software-controlled and is delivered when requested by the OMAP4430 (this input is on by default at platform boot).
  - CLK32KAUDIO is software-controlled and is delivered to the TWL6040 audio device (not shown in [Figure 27-3](#)) when requested by the OMAP4430 (this input is on by default at platform boot).
- High-speed clock: The platform high-speed (HS) low-jitter sine clock is provided by the clock driver (CDC3S04) to the TWL6040 audio IC, OMAP, and other peripherals requesting low-jitter sine clock (up to three outputs are available for peripherals). The OMAP device delivers digital clocks to other peripherals. The OMAP4430 supports different flavors of clock sources at several frequencies:
  - Crystal source. An internal oscillator embedded in the OMAP4430 is used. In this mode, the input clock is limited to 20 MHz by oscillator construction. The `fref_clk_ioreq` is then set as an input to be used by an external device to request the OMAP4430 to deliver the clock on `fref_clk0_out`. Pin `fref_slicer_in` can then be used to input an auxiliary clock (mainly for debugging purposes).
  - External oscillator type source. The internal OMAP4430 oscillator is disabled. The external source delivers a sine (slicer active) or square (slicer bypassed) single-ended clock on `fref_slicer_in` pin. The external source is enabled by the `fref_clk_ioreq` output. `fref_xtal_in` can then be used to input an auxiliary clock (oscillator-bypass mode).
- Audio clock: The TWL6040 audio device builds its own clock, either from the 32k clock or the HS clock (depending on the audio scenario). This internal clock is delivered to the OMAP4430 ABE subsystem (`abe_clks` input) for audio flow synchronization.

Clock input modes are selected by the `sysboot` pin (`sys_boot[6:7]`) to allow the system to wake up in the proper mode, even after a cold boot (selection of the relevant input pin (`fref_slicer_in` or `fref_xtal_in/out`) and slicer/oscillator mode setting (on or bypass)).

The OMAP4430 provides the HS clock to other devices through six `fref_clk[0:5]_out` pins.

TWL6040 audio device receives the HS clock on the MCLK pin when the audio scenario requires it (some low-power audio player scenarios can be performed only with the 32k-clock).

[Table 27-2](#) lists the mapping for the OMAP clock input sources. [Table 27-3](#) lists the TWL6030 clock requirements.

**Table 27-2. Mapping for Input Sources**

Input Source	Mapping	Frequency Range/List	Type
Internal oscillator	<code>fref_xtal_in/fref_xtal_out</code>	12, 16.8, or 19.2 MHz	Crystal connection balls
External HS clock	<code>fref_slicer_in</code>	12, 16.8, 19.2, 26, or 38.4 MHz	Sine or square single ended

**Table 27-2. Mapping for Input Sources (continued)**

Input Source	Mapping	Frequency Range/List	Type
External 32k-clock	sys_32k	32.768 kHz (nom.)	Square
TWL6040 audio IC	abe_clks	< 19.2 MHz	Square

**Table 27-3. TWL6030 Clock Requirements**

Input Source	Mapping	Frequencies Range/List	Type
Internal oscillator	OSC32KIN/OSC32KOUT	32.768 kHz (nom.)	Crystal connection pins

**CAUTION**

Clock configurations depend on core voltage, and maximum clock frequencies may not be applicable to production.

**27.2.3.2.2 Optional System Output Clocks**

The OMAP device can output six alternate clocks, `fref_clk[0:5]_out`. Output clocks 1 through 4 can be gated or enabled through associated clock-request signals, `fref_clk[1:4]_req`.

The clocks can be managed by software with the appropriate register in the power, reset, and clock management (PRCM) module. For more information, see [Section 3.3, PRCM Subsystem Environment](#).

**27.2.3.3 Reset Configuration****27.2.3.3.1 ON/OFF Interconnect and Power-On-Reset**

The entire system is typically awakened by an ON/OFF push button connected to the TWL6030 power IC. This signal belongs to the VBAT power domain and is active low (the TWL6030 internal pullup ties it to VBAT). Power up is propagated to the OMAP4430 through NRESPWRON/sys\_nrespwron signals when the TWL6030 power-up sequence is achieved. Because this signal is also provided to the TWL6040 audio IC, it resets its state-machine and gets some available functions even if V2V1 power supply is not provided.

**27.2.3.3.2 Warm Reset**

A warm reset can be asserted by the OMAP4430, by an external button (typically for development platform), or by any other device connected to it. It is mainly used to propagate the OMAP4430 internal reset to the whole platform to set it to a known state.

The OMAP4430 releases `sys_nreswarm` after NRESPWRON is deasserted.

**27.2.3.3.3 Peripheral Reset by sys\_pwron\_reset\_out**

`Sys_pwron_reset_out` is provided to the system by the OMAP4430 to power on and/or deactivate the reset of system peripherals. This signal is activated by SCRMP during the power-up sequence a few milliseconds after the default-on clocks (OMAP4430 – `fref_clk4_out` and CDC3S04 – `CLK[1:4]`) are stabilized. This allows switching on peripherals that require synchronous reset.

`Sys_pwron_reset_out` is controllable by software when the system is on. It is not affected by a warm reset.

**27.2.3.3.4 Peripheral Reset by GPIO**

Most of the peripherals are reset/power on/off by GPIO. By default, under POR, most OMAP4430 signals are in safe mode with a default value driven by the I/O cell. The value is driven by an internal pullup or pulldown. Depending on the peripheral reset active level, users must select one GPIO or another (according to the reset value).

Once POR is released, the value on the pad is driven by the default configuration of control module. Most of time, this configuration is aligned with the default value selected on the I/O cell.

The next step is application-dependent: Users must configure the OMAP4430 registers to validate GPIO use and the default configuration of the control module.

### 27.2.3.3.5 Warm Reset Impact on GPIOs

When a warm reset event occurs:

- The GPIO controller is reset. Consequently, the GPIO is automatically turned in input mode.
- The control module, which is part of the wake-up domain, is not reset. Information related to signal multiplexing mode and pullup or pulldown configuration is still valid.

Therefore, when a warm reset event happens, the output buffer is disabled and the selected pull is activated. Consequently, two different behaviors can be defined with regard to what is expected by the platform:

- GPIO sensitive to warm reset:  
To avoid getting a floating signal during (and after) a warm reset event, users need to select a pullup or pulldown (depending on the reset value of the peripheral) and enable the pull. Pull activation does not affect overall leakage when the GPIO buffer is driving because the pull is disconnected at that time.
- GPIO not sensitive to warm reset:  
To avoid getting a floating signal during (and after) a warm reset event and to keep the same value that was driven before the reset, users must align the pull value with the drive value each time there is an access to a dedicated GPIO register. This activation of the pull does not affect the overall leakage when the GPIO buffer is driving because the pull is disconnected at that time.

---

**NOTE:**

- For more information about device reset management, see [Section 3.5, Reset Management Functional Description](#).
  - To determine the cause of the last reset, see [Section 3.5.5, Reset Logging](#), in [Chapter 3, Power, Reset, and Clock Management](#).
- 

### 27.2.3.4 TWL Power/Audio Devices Control

- I<sup>2</sup>C:  
The TWL6030 power and TWL6040 audio devices are controlled by the OMAP4430 I2C1 (HS I<sup>2</sup>C). Some internal pullup resistors are provided by the OMAP4430 (different values can be selected to adapt the I/O buffer strength to the bus topology). The I<sup>2</sup>C links must first be started in fast-speed mode. Therefore, the buffer default pullup value must support this mode.
- SmartReflex control:  
Some of the TWL6030 power resources are controlled through SmartReflex link (I<sup>2</sup>C-like). This control allows the host to drive the dynamic-frequency voltage scaling (DVFS) and adaptive voltage scaling (AVS) operations of these resources. As with its I<sup>2</sup>C interfaces, some internal pullup resistors are provided on the OMAP4430 SmartReflex I/Os. The link must be run at the highest speed allowed by the voltage controller of the OMAP4430 (lower than I<sup>2</sup>C HS mode frequency). For example, this frequency must be 1.82 MHz with a 38.4-MHz system clock.
- INT:  
Both devices can activate the INT signal at any time when they require the host device to monitor their activity. When receiving such an interruption, the OMAP4430 checks, through the I<sup>2</sup>C, to determine the source of the interrupt. These INT pins are active low.
- Sys\_pwr\_req:  
When exiting off mode, the OMAP4430 ties up the sys\_pwr\_req pin to wake up the TWL6030 power application group.

### 27.2.3.5 TWL6030 Power Request Signals

The TWL6030 power-management IC drives three external enable-output signals. This allows switching on some external resources at different stages of the power-up sequence:

- REGEN1/2 are driven high at the beginning of the power-up sequence, before any internal power source is turned on. They belong to the VBAT power domain. REGEN1 can typically be used for buck boost control.
- SYSEN is driven high immediately after the VMEM power output is turned on. It allows switching on an additional regulator in case the VMEM power regulator does not support the current requirements of memory. SYSEN belongs to the VIO power domain.

TWL6030 power also receives five power resource requests: PREQ[1, 2A, 2B, 2C, 3]. These pins allow an external device to request TWL6030 power internal resources. TWL6030 power behavior when pins are activated must be programmed after the first boot (resources and PREQ pins allocated by group, resource behavior upon group activation, etc.). The typical PREQ allocation per group is:

- PREQ1: Application group. Connected to OMAP.
- PREQ2A, PREQ2B, and PREQ2C: Peripheral group. Connected to peripherals A, B, and C (for instance, BT/FM/GPS/WLAN device, Wimax device, W-USB device)
- PREQ3: Modem group

---

**NOTE:** Powering the clock driver requires PREQ assertion. Therefore, it is advised to connect any peripheral receiving a clock from CDC3S04 to one PREQ.

---

The use of PREQ[1:3] is especially required when TWL6030 power is in sleep mode, because it cannot handle an I<sup>2</sup>C command. Any device that requires the TWL6030 resource to wake up must first activate its associated PREQ. All TWL6030 power regulators are off or in sleep mode when TWL6030 power is in sleep mode.

PREQ1 is used by OMAP when waking up from off mode to wake up the TWL6030.

### 27.2.4 Boot Configuration

Six external pins (sys\_boot[5:0]) are used to select interfaces or devices for the booting list. The sys\_boot[7:6] pins select the OMAP clock source configuration.

All eight pins are sampled and latched onto the SYSCTRL\_GENERAL\_CORE.CONTROL\_STATUS[7:0] SYS\_BOOT register bit field after POR. After booting, these pins can be used for other functions such as GPIOs, and the associated register bit field is not updated by the new functionality. See [Section 18.4.8, Pad Functional Multiplexing and Configuration](#), for more information about pad multiplexing configuration.

---

**NOTE:**

- If used as GPIOs, sys\_boot[7:0] pins must be used only in output mode to ensure that the input values are selected by the pullups and pulldowns after a POR/warm reset.
  - Care must be taken when choosing to use sys\_boot[7:6] pins as GPIOs because these pins have no internal pullup or pulldown capability.
- 

#### 27.2.4.1 Clock Source Selection

OMAP4430 clock source settings:

- sys\_boot[7:6] = 0b00:
  - The system clock source is the on-chip oscillator.
  - The slicer is powered down and its input can be used to provide the OMAP with an alternate clock.
  - Fref\_clk\_ioreq is a clock request input associated to fref\_clk0\_out by default.
- sys\_boot[7:6] = 0b01:
  - The system clock source is external. The internal oscillator is bypassed to inject the system clock.
  - The slicer is powered down and the slicer input can be used to provide the chip with an alternate clock.
  - The Fref\_clk\_ioreq signal is an output that requests the clock to the external clock supplier



- `sys_boot[7:6] = 0b10:`
  - The system clock source is the on-chip slicer, supplied by an external oscillator. The internal oscillator is powered down and the input can be configured to supply the alternate system clock to OMAP.
  - The slicer is active.
  - The `Fref_clk_ioreq` signal is an output that requests the clock from the external clock supplier.
- `sys_boot[7:6] = 0b11:`
  - The system clock source is external. The internal oscillator is powered down and the input can be configured to supply the alternate system clock to the OMAP device.
  - The slicer is bypassed to inject the system clock.
  - The `Fref_clk_ioreq` signal is an output that requests the clock from the external clock supplier.

#### 27.2.4.2 Booting Device Order Selection

The ROM code creates the device list (order) based on information gathered from two locations.

- The first location is the software booting configuration stored in the nonvolatile save-and-restore (SAR) memory, and used after warm resets.
- The second location is the `sys_boot[5:0]` external configuration pins sensed in the control module. The SYSBOOT pins are used to index a booting device list from a table with possible booting scenarios. The order of examined booting devices is from the first to the fourth devices.

The following names are used in the tables:

- Memory types:
  - Execute in place (XIP): XIP memory without wait-signal monitoring enabled (NOR flash memory or other XIP device)
  - XIP wait: XIP memory with wait-signal monitoring
  - NAND: NAND flash memories (non-XIP)
  - OneNAND: OneNAND and Flex-OneNAND flash memories
  - EMIF: LPDDR2-NVM device
  - MMC1: MMC/SD card or eMMC/eSD™ device (SD/MMC port 1)
  - MMC2(1): eMMC or eSD memory device, first option (SD/MMC port 2). See [Section 27.2.4.3, Boot Peripheral Pin Multiplexing](#).
  - MMC2(2): eMMC or eSD memory device, second option (SD/MMC port 2). See [Section 27.2.4.3, Boot Peripheral Pin Multiplexing](#).
- Peripheral interfaces:
  - USB: USB A0 from OMAP internal transceiver (USBPHY)
  - USB-ULPI: USB A0 from external transceiver through ULPI interface
  - Universal asynchronous receiver/transmitter (UART): UART3 interface
- Permanent booting devices listed in [Table 27-4](#) are in **bold**.

---

**NOTE:** Only permanent booting devices (in bold) are put on in the list in case of a warm reset, and the device list is taken from the `sys_boot` pins. As a result, peripheral booting devices are skipped when the OMAP4430 restarts from warm resets.

---

Users can force the execution of the peripheral booting procedure after warm reset by using the software booting configuration feature.

[Table 27-4](#) lists the booting device order when it is preferred to boot from a memory type device. This table is selected by ROM code when `sys_boot5` pin is tied high (= 1) after reset and before booting completion. If fast XIP booting is selected, the `sys_boot5` pin controls the XIP wait-signal monitoring (that is, a logical high level enables wait-signal monitoring).

**Table 27-4. Memory Preferred Booting**

sys_boot[5:0]	Booting Devices Order			
	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
0b100000	<b>MMC2(1)</b>	USB		
0b100001	<b>XIP</b>	USB		
0b100010	<b>XIPWAIT</b>	USB		
0b100011	<b>NAND</b>	USB		
0b100100	<b>EMIF</b>	USB		
0b100101	<b>MMC1</b>	USB		
0b100110	<b>OneNAND</b>	USB		
0b100111	<b>OneNAND</b>	<b>MMC2(1)</b>	USB	
0b101000	<b>MMC2(1)</b>	UART		
0b101001	<b>XIP</b>	UART		
0b101010	<b>XIPWAIT</b>	UART		
0b101011	<b>NAND</b>	UART		
0b101100	<b>EMIF</b>	UART		
0b101101	<b>MMC1</b>	UART		
0b101110	<b>OneNAND</b>	UART		
0b101111	<b>MMC2(1)</b>	USB-ULPI		
0b110000	<b>XIP</b>	USB-ULPI		
0b110001	<b>XIPWAIT</b>	USB-ULPI		
0b110010	<b>NAND</b>	USB-ULPI		
0b110011	<b>EMIF</b>	USB-ULPI		
0b110100	<b>MMC1</b>	USB-ULPI		
0b110101	<b>OneNAND</b>	USB-ULPI		
0b110110	<b>MMC2(1)</b>	USB	UART	MMC1
0b110111	<b>XIP</b>	USB	UART	MMC1
0b111000	<b>MMC2(2)</b>	USB-ULPI	UART	
0b111001	<b>NAND</b>	USB	UART	MMC1
0b111010	<b>MMC2(2)</b>	UART		
0b111011	<b>MMC1</b>	USB	UART	
0b111100	<b>MMC2(2)</b>	USB		
0b111101			Reserved	
0b111110			Reserved	
0b111111 <sup>(1)</sup>	<b>Fast XIP booting. Wait monitoring ON</b>		USB	UART

<sup>(1)</sup> Only on GP devices

[Table 27-5](#) lists the booting device order when it is preferred to boot from a peripheral-type device. This table is selected by ROM code when the sys\_boot5 pin is tied low (= 0). If fast XIP booting is selected, the sys\_boot5 pin controls the XIP wait-signal monitoring (that is, a logical low level disables wait-signal monitoring).

**Table 27-5. Peripheral Preferred Booting**

sys_boot[5:0]	Booting Devices Order			
	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
0b000000	USB	MMC2(1)		
0b000001	USB	XIP		
0b000010	USB	XIPWAIT		
0b000011	USB	NAND		
0b000100	USB	EMIF		
0b000101	USB	MMC1		
0b000110	USB	OneNAND		
0b000111	USB	OneNAND	MMC2(1)	
0b001000	UART	MMC2(1)		
0b001001	UART	XIP		
0b001010	UART	XIPWAIT		
0b001011	UART	NAND		
0b001100	UART	EMIF		
0b001101	UART	MMC1		
0b001110	UART	OneNAND		
0b001111	USB-ULPI	MMC2(1)		
0b010000	USB-ULPI	XIP		
0b010001	USB-ULPI	XIPWAIT		
0b010010	USB-ULPI	NAND		
0b010011	USB-ULPI	EMIF		
0b010100	USB-ULPI	MMC1		
0b010101	USB-ULPI	OneNAND		
0b010110	USB	UART	MMC1	MMC2(1)
0b010111	USB	UART	MMC1	XIP
0b011000	USB-ULPI	UART	MMC2(2)	
0b011001	USB	UART	MMC1	NAND
0b011010	UART	MMC2(2)		
0b011011	USB	UART	MMC1	
0b011100	USB	MMC2(2)		
0b011101			Reserved	
0b011110			Reserved	
0b011111 <sup>(1)</sup>	Fast XIP booting. Wait monitoring OFF		USB	UART

<sup>(1)</sup> Only on GP devices

### 27.2.4.3 Boot Peripheral Pin Multiplexing

Table 27-6 lists the code pin multiplexing configuration supported by ROM according to boot peripheral.

**Table 27-6. Pin Multiplexing According to Boot Peripheral**

Boot Device	Boot Interface	Pins
NAND/OneNAND	GPMC	GPMC pins in mux mode 0, CS0, WAIT0
XIP memory	GPMC	GPMC pins in mux mode 0, CS0, WAIT0
MMC/SD/eMMC/eSD	SDMMC1	SDMMC1 pins in mux mode 0
eMMC/eSD	SDMMC2 (option 1) SDMMC2 (option 2)	GPMC pins in mux mode 1 SDMMC5 + FREF_CLK pins, all in mux mode 5
LPDDR2-NVM	EMIF1	LPDDR21 pins in mux mode 0, CS1
UART	UART3	UART3 pins in mux mode 0

**Table 27-6. Pin Multiplexing According to Boot Peripheral (continued)**

Boot Device	Boot Interface	Pins
USB from internal transceiver	HSUSBOTG (USBA0-USBPHY)	USBA0_OTG pins in mux mode 0
USB from external transceiver	HSUSBOTG (USBA0-ULPI)	DPM_EMU[2:13] pins in mux mode 1

**NOTE:** The ROM code examines the interfaces that are selected to be searched until a valid bootable interface/device is found. The activities on the pins of the failed interfaces must be considered in case they are connected to any other peripherals for any other purposes (for example, an LED connected to a GPMC pin muxed internally to a GPIO).

### 27.3 Reset Sequences

For the description of the reset sequences, see [Section 3.5.6, Reset Sequences](#), in [Section 3.5, Reset Management Functional Description](#).

## 27.4 Device Initialization by ROM Code

This section describes high-level booting concepts and provides basic knowledge for booting on the device.

### 27.4.1 Booting Overview

#### 27.4.1.1 Booting Types

Bootting is the process of starting a bootstrap from one of the booting devices.

The ROM code has two functions for booting: Peripheral booting and memory booting.

- In peripheral booting, the ROM code polls a selected communication interface such as UART or USB, downloads the executable code over the interface, and executes it in internal RAM. Downloaded software from an external host can be used to program flash memories connected to the device. This special case of peripheral booting is called preflashing; software downloaded for preflashing is called the flash loader. The flash loader burns a new client application image in external flash memory. Initial software is a generic term for bootstrap, downloaded software, and flash loader. After the image is burnt, a software (warm) reset can be performed.
- In memory booting, the ROM code finds the bootstrap in permanent memories such as flash memory or memory cards and executes it. This process is normally performed after a cold or warm device reset.

The ROM code detects whether the device should download software from a peripheral interface (USB or UART) by using the `sys_boot[5:0]` pin configuration. This mechanism encompasses initial flashing in production (external memory is empty) and reflashing in service (external memory is already programmed).

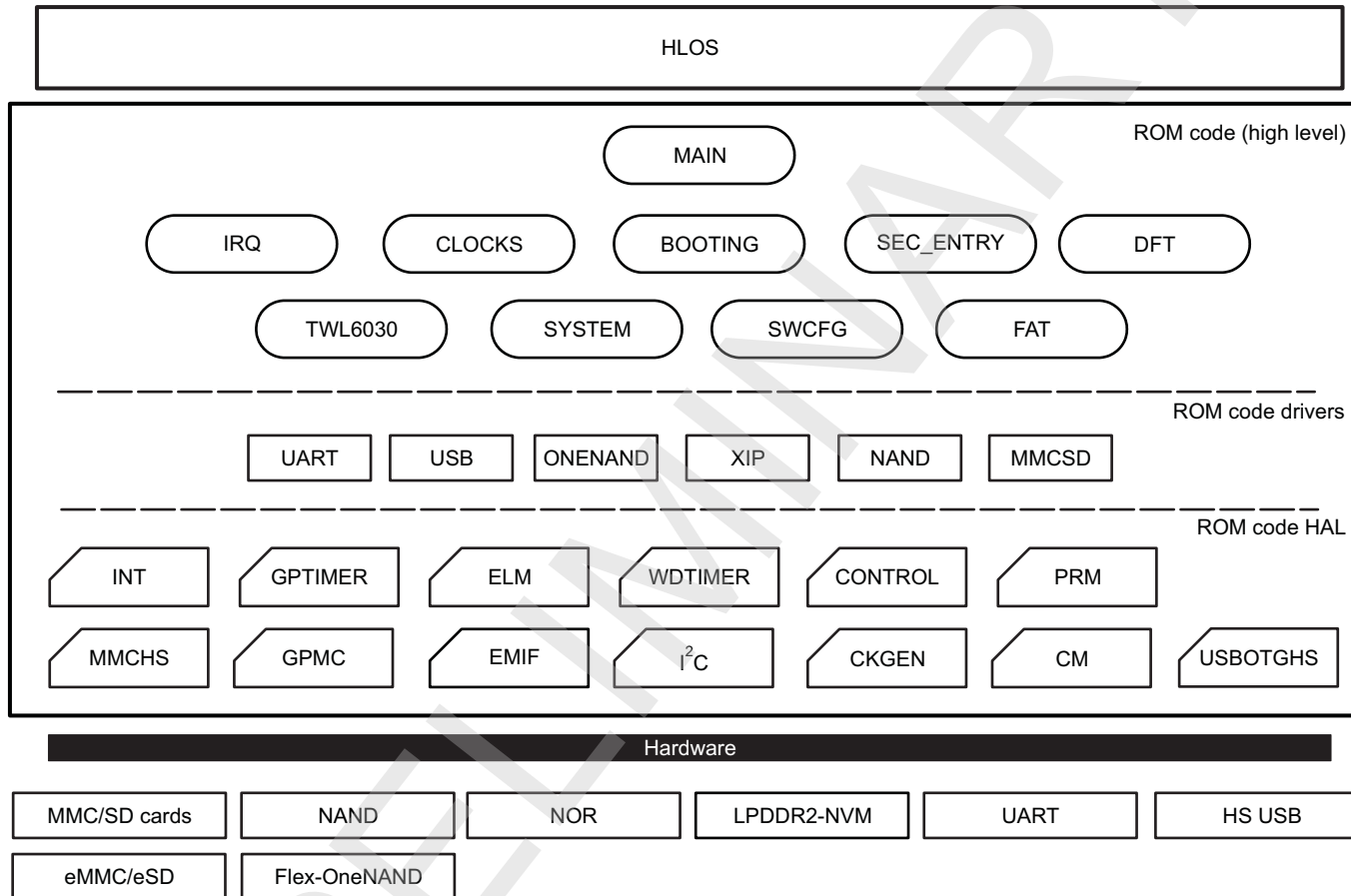
#### 27.4.1.2 ROM Code Architecture

Figure 27-4 shows the ROM code architecture. It is split into three main layers with a top-down approach: high-level, drivers, and hardware abstraction layer (HAL). One layer communicates with a lower-level layer through a unified interface.

- The high-level layer performs the main tasks of the public ROM code: multicore startup, watchdog and clock configurations, interrupt management, interaction with the power-management IC, and main booting routine.
- The driver layer implements the logical and communication protocols for any booting device in accordance with the interface specification.
- The HAL implements the lowest level code for interacting with the hardware infrastructure IPs. End booting devices are attached to the device I/O pads.

Figure 27-4 shows the three layers with their modules.

Figure 27-4. ROM Code Architecture



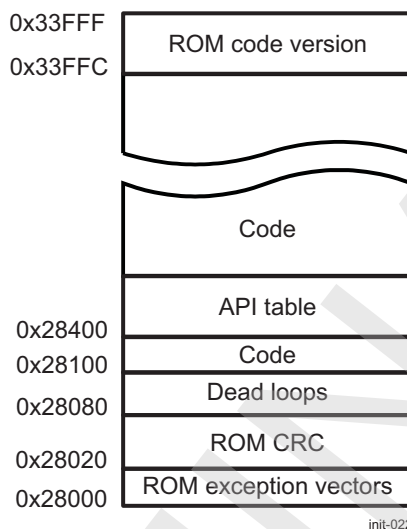
init-006

## 27.4.2 Memory Maps

### 27.4.2.1 ROM Memory Map

Figure 27-5 shows the 48-KB ROM memory map.

**Figure 27-5. ROM Memory Map**



- ROM exception vectors  
Exceptions are redirected to ROM exception vectors (see [Table 27-7](#)). The reset exception is redirected to the public ROM code startup. Other exceptions are redirected to RAM handlers by loading appropriate addresses to the PC register.

**Table 27-7. ROM Exception Vectors**

Address	Exception	Content
0x28000	Reset	Branch to the ROM code startup
0x28004	Undefined	PC = 0x4030 D004
0x28008	Software interrupt (SWI)	PC = 0x4030 D008
0x2800C	Prefetch abort	PC = 0x4030 D00C
0x28010	Data abort	PC = 0x4030 D010
0x28014	Unused	PC = 0x4030 D014
0x28018	IRQ	PC = 0x4030 D018
0x2801C	FIQ	PC = 0x4030 D01C

- ROM code cyclic redundancy check (CRC)  
The ROM code CRC is calculated as 32-bit CRC code (CRC-32-IEEE 802.3) for the address range 0x28000 - 0x33FFF. The 4-byte CRC code is stored at location 0x28020.
- Dead loops  
Dead loops are branch instructions coded in ARM™ mode. They have multiple purposes (see [Table 27-8](#)).



**Table 27-8. Dead Loops**

Address	Purpose
0x28080	Undefined exception default handler
0x28084	SWI exception default handler
0x28088	Prefetch abort exception default handler
0x2808C	Data abort exception default handler
0x28090	Unused exception default handler
0x28094	IRQ exception default handler
0x28098	FIQ exception default handler

The fixed location of these dead loops facilitates debugging and testing. The first seven dead loops are default exception handlers linked to RAM exception vectors.

Dead loops can be called directly from code, but there is also a special function called from ROM code to execute a dead loop. This function is at address 0x280C0. The function is assembly code in ARM mode, which takes the dead loop address from the R0 register. The main purpose of the function is to issue a global software reset before going to a dead loop. In addition, the function clears the global cold reset status before issuing the global software reset.

- Code

This space is used to hold code and constant data.

- API table

The purpose of this table is to allow external code access to system maintenance, utility, and device driver functions which are used for ensuring the ROM code boot functionality. These functions can be reused at run time by calling a fixed address hardcoded in this table.

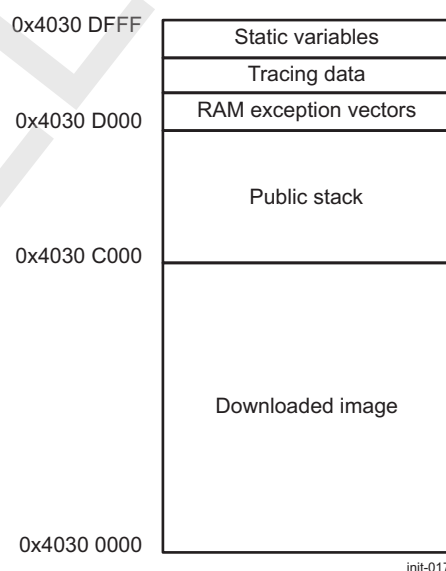
- ROM code version

The ROM code version consists of two decimal numbers: major and minor. It can be used to identify the ROM code release version burnt in a given IC (for example, useful during a debugging session). The ROM code version is a 32-bit hexadecimal value located at address 0x33FFC.

### 27.4.2.2 RAM Memory Map

The partitioning of the on-chip SRAM (L3 OCM RAM) shown in Figure 27-6 is used during the booting process. Tracing and static variables areas can also be accessed when calling API functions.

**Figure 27-6. RAM Memory Map**



- Downloaded image

This space is used by the public ROM code to store a downloaded booting image. It can be up to 48KB.

- Public stack  
This space is reserved for stack.
- RAM exception vectors  
The RAM exception vectors provide an easy way to redirect exceptions to the custom handler. [Table 27-9](#) lists the contents of the RAM space reserved for RAM vectors. The first eight addresses are ARM instructions that load the value in the subsequent eight addresses into the PC. These instructions are executed when an exception occurs because they are called from ROM exception vectors. Undefined, SWI, unused, and FIQ exceptions are redirected to a hardcoded dead loop. Prefetch abort, data abort, and IRQ exception are redirected to predefined ROM handlers. Users can redirect an exception to another handler by writing its address to the appropriate location from 0x4030 D024 to 0x4030 D03C, or by overriding the branch (load into PC) instruction between addresses from 0x4030 D004 to 0x4030 D01C.

**Table 27-9. RAM Exception Vectors**

Address	Exception	Content
0x4030 D000	Reserved	Reserved
0x4030 D004	Undefined	PC = [0x4030D024]
0x4030 D008	SWI	PC = [0x4030D028]
0x4030 D00C	Prefetch abort	PC = [0x4030D02C]
0x4030 D010	Data abort	PC = [0x4030D030]
0x4030 D014	Unused	PC = [0x4030D034]
0x4030 D018	Interrupt request (IRQ)	PC = [0x4030D038]
0x4030 D01C	Fast interrupt request (FIQ)	PC = [0x4030D03C]
0x4030 D020	Reserved	0x28090
0x4030 D024	Undefined	0x28080
0x4030 D028	SWI	0x28084
0x4030 D02C	Prefetch abort	Address of default prefetch abort handler <sup>(1)</sup>
0x4030 D030	Data abort	Address of default data abort handler <sup>(1)</sup>
0x4030 D034	Unused	0x28090
0x4030 D038	IRQ	Address of default IRQ handler
0x4030 D03C	FIQ	0x28098

<sup>(1)</sup> The default handlers for prefetch and data abort perform reads from CP15 debug registers to retrieve the reason for the abort:

- In case of prefetch abort: the IFAR register is read from CP15 and stored into R0. The IFSR register is read and stored into the R1 register. Then the ROM code jumps to the prefetch abort dead loop (20088h).
  - In case of data abort: the DFAR register is read from CP15 and stored into R0. The DFSR register is read and stored into the R1 register. Then the ROM code jumps to the data abort dead loop (2008Ch).
- Tracing data  
This area contains trace vectors reflecting the execution path of the ROM Code. The public ROM code tracing data is described in [Table 27-10](#). For more information about ROM code tracing, see [Section 27.4.9, Tracing](#).

**Table 27-10. Tracing Data**

Address	Size (Bytes)	Description
0x4030 D040	4	Current tracing vector, word 1
0x4030 D044	4	Current tracing vector, word 2
0x4030 D048	4	Current tracing vector, word 3
0x4030 D04C	4	Cold reset run tracing vector, word 1
0x4030 D050	4	Cold reset run tracing vector, word 2
0x4030 D054	4	Cold reset run tracing vector, word 3
0x4030 D058	4	Current copy of the PRM_RSTST register (reset reasons)

**Table 27-10. Tracing Data (continued)**

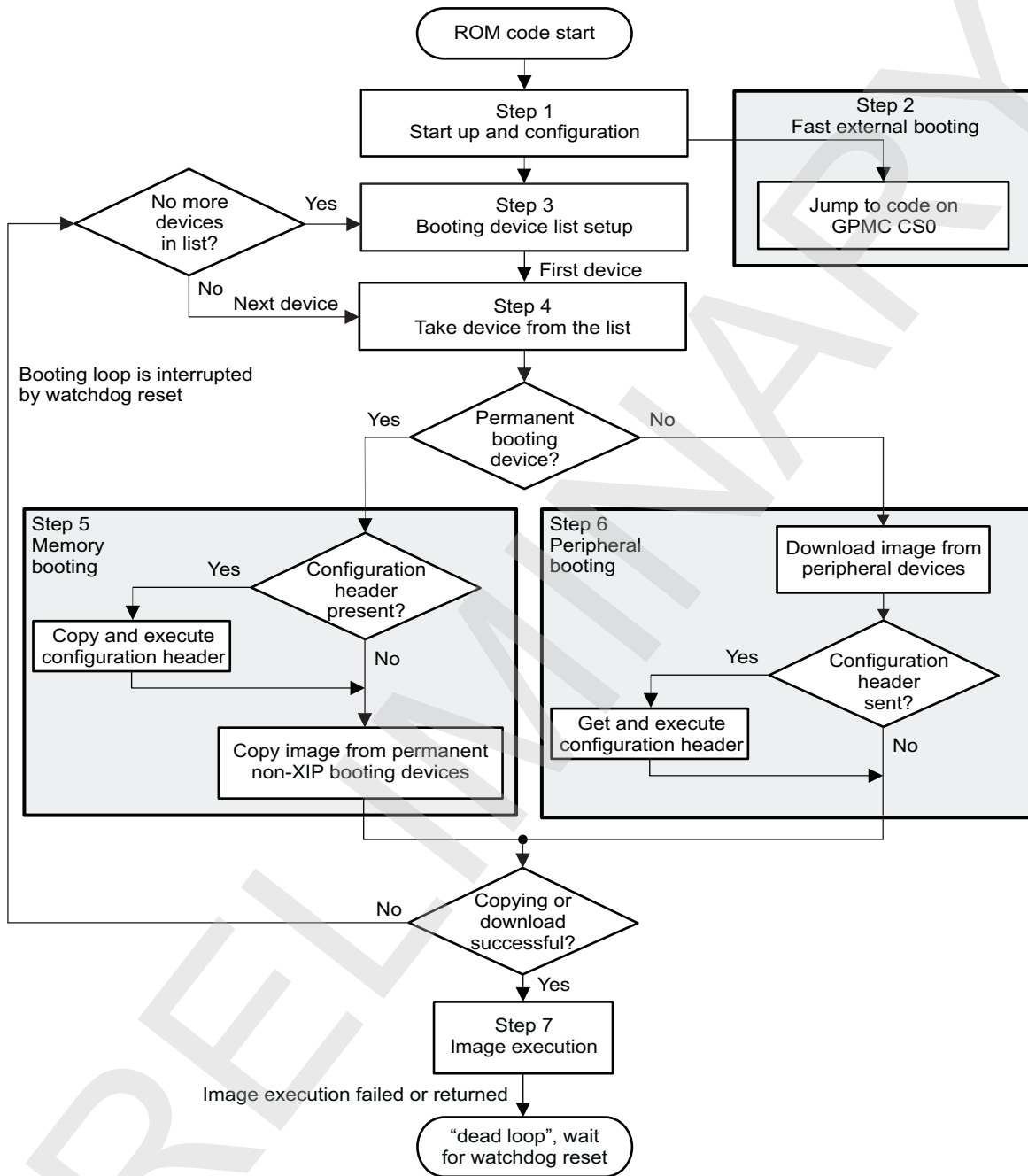
<b>Address</b>	<b>Size (Bytes)</b>	<b>Description</b>
0x4030 D05C	4	Reserved
0x4030 D060	4	Reserved
0x4030 D064	4	Reserved

- **Static variables**  
ROM code static variables used at boot time (and possibly during run time if calling the ROM API functions).

### **27.4.3 Overall Booting Sequence**

Figure 27-7 shows the ROM code flow chart.

Figure 27-7. Overall Booting Sequence



init-007

The main loop of the booting module goes through the booting device list and tries to get an image from the currently selected booting device. The ROM code performs the following steps:

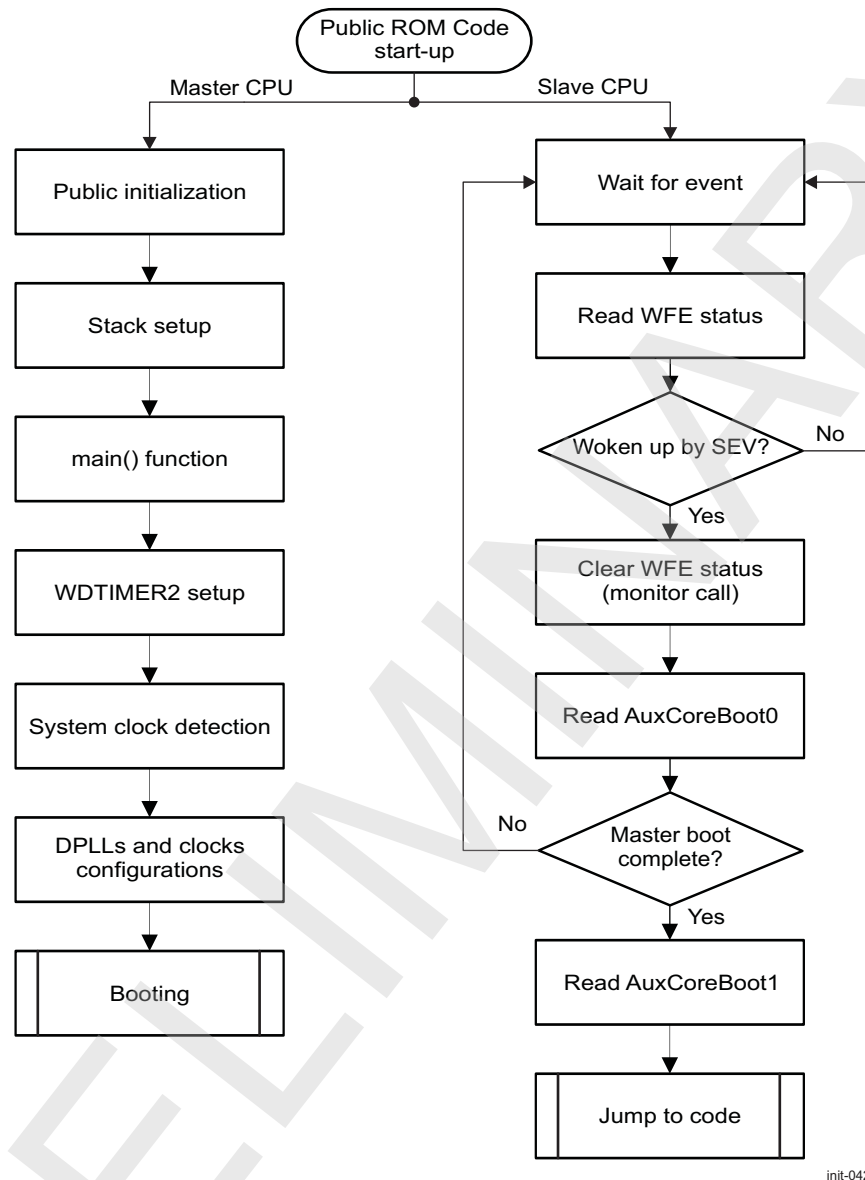
- Step 1. Basic configuration and initialization. Reading of SYSBOOT pins or software booting configuration
- Step 2. The path named fast external boot is a special low-latency boot mode. It consists of a blind jump to an external addressable memory. See [Section 27.4.6, Fast External Booting](#).
- Step 3. A booting device list is created (see [Section 27.4.4.3, Booting Device List Setup](#)). The list consists of all devices to be searched for a booting image. The list is created based on the sys\_boot pins or the software booting configuration described in [Section 27.4.4.4, Software Booting Configuration](#). The software booting configuration structure is in the SAR memory and can be written by software before executing a software reset. After a software reset, the software booting configuration has priority over the sys\_boot pin configuration.
- Step 4. The main loop of the booting procedure goes through the booting device list and tries to search for an image from the currently selected booting device. This loop is exited if a valid booting image is found and successfully executed or upon watchdog expiration. If an image is found, ROM code executes memory booting or peripheral booting, depending on the current booting device type:
  - Memory booting is executed when the booting device is XIP memory, NAND, OneNAND/Flex-OneNAND, LPDDR2-NVM, or eMMC/eSD/MMC/SD.
  - Peripheral booting is executed when the device is a UART or USB.
- Step 5. Memory booting reads data from memory type devices. Memory booting is described in detail in [Section 27.4.7, Memory Booting](#).
- Step 6. Peripheral booting downloads data from communication interfaces. Peripheral booting is described in [Section 27.4.5, Peripheral Booting](#).
- Step 7. For the GP device, the image is automatically started.

An additional feature of the booting module is the execution of the configuration header (CH). The CH configures the system for faster and more flexible booting from the selected permanent device. The CH, which is optional, is described in [Section 27.4.8.2, Configuration Header](#).

## **27.4.4 Startup and Configuration**

### **27.4.4.1 Startup**

[Figure 27-8](#) shows the ROM code start-up sequence.

**Figure 27-8. ROM Code Multiprocessor Start-Up Sequence**

init-042

The master CPU L1 instruction cache and branch prediction mechanisms are activated as part of the public boot process. The public vector base address is configured to the reset vector of ROM code (0x28000). The MMU remains switched off during boot (thus, L1 data cache is off). The master CPU performs the public-side basic initialization and stack setup (compiler auto-generated C- initialization or *scatter loading*). Next, it configures watchdog timer 2 (set to 3 minutes), performs system clock detection, and configures the system clock. Finally, it jumps to the booting routine.

No specific configuration is performed for the slave CPU, which keeps its default configuration after reset (L1 instruction and data caches off, branch prediction off, MMU off, no remap of public-vector-base address). The slave CPU is rapidly held in wait-for-event state. It stays in this state while the master CPU completes the public boot process and until jumping to the external software (for example, HLOS). At this stage, the external software can wake up the slave CPU by executing an SEV command.

Two internal memory-mapped registers are available to the OS for communicating start-up information. The AuxCoreBoot 0 and AuxCoreBoot 1 registers are located in the MPU WakeupGen domain.

- AuxCoreBoot0 is used as a status register to signal the slave CPU that it must wake up after the send event operation initiated by the master CPU.

- The AuxCoreBoot1 register contains the address location to which the slave CPU must jump after wakeup. These registers can be accessed by the master CPU (HLOS) using with the appropriate service calls (see [Section 27.5](#), *Services for HLOS Support*).

#### 27.4.4.2 Clocking Configuration

The ROM code detects the system input clock frequency. The supported frequencies are:

- 12 MHz
- 16.8 MHz
- 19.2 MHz
- 26 MHz
- 38.4 MHz

After detecting the input clock, the ROM code configures the clocks and DPLLs required for ROM code execution.

The configured DPLLs are:

- Peripheral DPLL: Set to provide 192, 128, 96, 48, 24, and 12 MHz for peripheral blocks
- Core DPLL: Set to provide L3/L4 interconnect and MPU and EMIF clocks
- MPU DPLL: Left in bypass, and the MPU internal clock is derived from the core DPLL
- USB DPLL: Locked only in case of USB peripheral booting from internal transceiver. It is left unused in other cases (for example, memory boot or peripheral boot from UART or USB ULPI) to save power.

The multipliers and dividers of the DPLLs are set to values that depend on the input clock detected.

[Table 27-11](#) summarizes the ROM code default settings for key clocks.

**Table 27-11. ROM Code Default Clock Settings**

Clock	Frequency [MHz]	Source
<b>FCORE_DPLL</b>	400	Derived from SYS_CLK
CORE_X2_CLK	200	
CORE_CLK	200	CORE_X2_CLK/1
L3_ICLK	100	CORE_CLK/2
L4_ICLK	50	L3_ICLK/2
<b>FPER_DPLL</b>	384	Derived from SYS_CLK
FUNC_192M_CLK	192	
FUNC_128M_CLK	128	
FUNC_96M_FCLK	96	
FUNC_48M_FCLK	48	
<b>FMPU_DPLL</b> (in bypass)	200	CORE_X2_CLK
MPU_DPLL_CLK	200	FMPU_DPLL/1
<b>FUSB_DPLL</b> <sup>(1)</sup>	960	Derived from SYS_CLK
INIT_960M	960	FUSB_DPLL/1
INIT_480M	480	INIT_960M/2
INIT_60M	60	INIT_480M/8

<sup>(1)</sup> Locked only in case of USB peripheral booting from internal PHY

There are three ways to change DPLLs and all related clock divider, gating, and multiplexer configurations during the boot:

- ROM code default settings, described in this paragraph. They are always applied at any reset. [Table 27-11](#)
- Software booting configuration after a software reset, described in [Section 27.4.4.4](#), *Software Booting Configuration*
- Configuration header (CH), described in [Section 27.4.8.2](#), *Configuration Header*. The CH lets users

have a known configuration (about GPMC and clock registers) after memory booting. This configuration can be blocked by the software booting configuration.

### 27.4.4.3 Booting Device List Setup

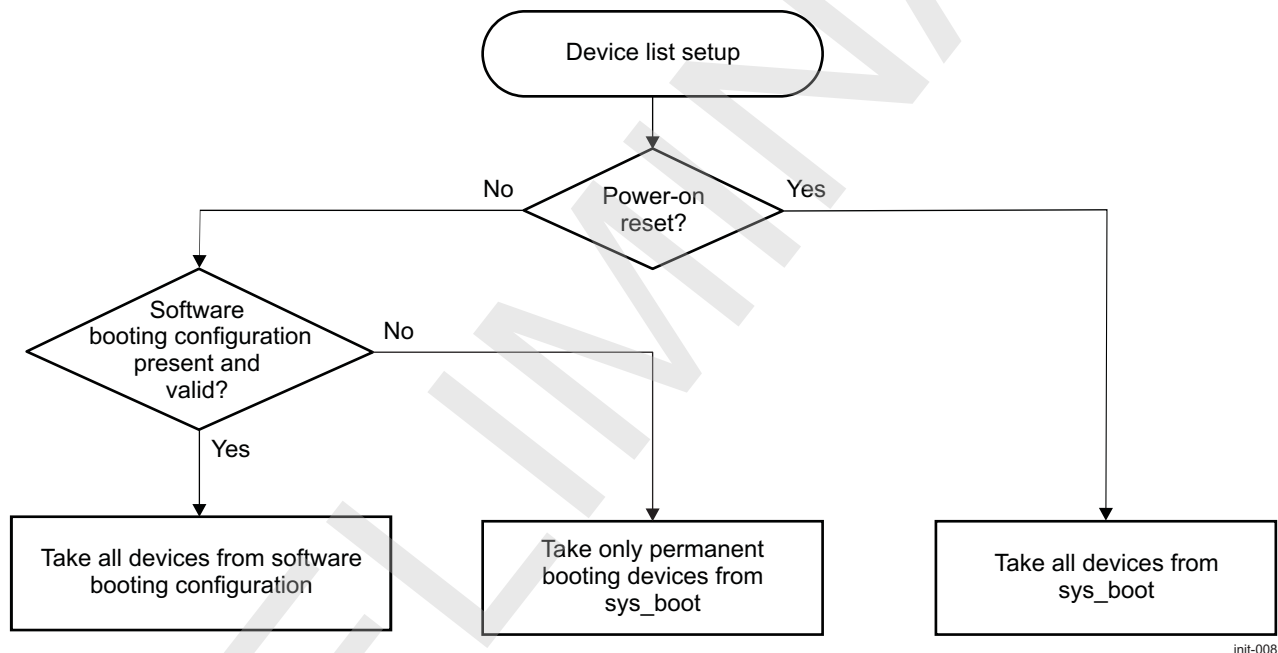
The ROM code creates a device list based on two sources:

- The software booting configuration is stored in nonvolatile RAM SAR memory.
- The sys\_boot[5:0] pins sensed in the control module are used to index the device table from which the list of devices is extracted.

Figure 27-9 shows how the ROM code sets up the device list depending on the reset source.

**NOTE:** Only permanent booting devices are put on the list when reset is not power on and devices are taken from the sys\_boot pins. Users can force peripheral booting after software reset using the software booting configuration.

**Figure 27-9. Device List Setup**



### 27.4.4.4 Software Booting Configuration

The software booting configuration is a logical structure stored in SAR RAM memory, which is not cleared after warm resets or wakeups from low-power modes.

#### 27.4.4.4.1 Public Use of SAR RAM

At system level, the OMAP4430 SAR RAM memory is divided into four banks. The public ROM code uses only the first bank, which is always public-accessible. More specifically, the software booting configuration structure must be located in the upper 1.5KB of the first bank.

The public ROM code offers some flexibility about the location of the software booting configuration structure. The PUBLIC\_SW\_BOOT\_CFG\_ADDR pointer defines the start address of the structure within the SAR RAM bank (see Table 27-12).



**Table 27-12. Public Use of SAR RAM**

Logical name	Address	Size (Bytes)	Description
PUBLIC_SW_BOOT_CFG_ADDR	0x4A326A00	4	Public software booting configuration pointer. The word value at this location is: - Reset (zero) on a cold reset. - Read on a warm reset.
CPU0_WAKEUP_NS_PA_ADDR	0x4A326A04	4	CPU0 nonsecure wake-up restoration pointer <sup>(1)</sup>
CPU1_WAKEUP_NS_PA_ADDR	0x4A326A08	4	CPU1 Nonsecure wake-up restoration pointer <sup>(1)</sup>
PUBLIC_SAR_RAM_1_FREE	0x4A326A0C	...	Recommended address for start of software booting configuration structure described in <a href="#">Section 27.4.4.4.2, Software Booting Configuration Structure</a> .

<sup>(1)</sup> This pointer is used as part of the restoration procedure from low-power modes.

As mentioned previously, the software booting configuration feature is optional. Hence, the public ROM code decides to use the feature based on the value read on a warm reset at the address pointed to by the PUBLIC\_SW\_BOOT\_CFG\_ADDR pointer. If the value matches the range 0x4A326A00 – 0x4A326FFF, the ROM code tries to extract the structure located at that address. The value pointed to by PUBLIC\_SW\_BOOT\_CFG\_ADDR is always overwritten to zero on a cold reset.

The recommended address for storing the software booting configuration structure described hereafter is defined as PUBLIC\_SAR\_RAM\_1\_FREE. It is, however, possible to locate the structure at any location within the 1.5-KB range.

It is moreover possible to use the public SAR RAM area for any other purpose, such as storing traces for HLOS use. Obviously, care must be taken not to overwrite the locations used for low-power modes and/or software booting configuration if used.

#### **27.4.4.4.2 Software Booting Configuration Structure**

**Table 27-13** describes the software booting configuration structure. It offers three levels of flexibility for redefining the ROM code defaults after a warm reset:

- Redefining the default device booting list (from ROM code defaults given by SYSBOOT pins configuration)
- Redefining the default clock settings
- Redefining timeouts linked to the peripheral booting mechanism

---

**NOTE:** The sections are provided as a linked list; therefore, the order and number of items is not relevant.

---

The ROM code searches for the next section at the location based on the size filled in the previous section. The clock configuration from software booting configuration overwrites the CH settings.

**Table 27-13. Software Booting Configuration Structure**

Field	Size (Bytes)	Description
<b>Booting Configuration</b>		
Section 1 key	4	Synchronization key for section 1: 0xCF00AA01
Section 1 size	4	Size of section 1: 0x0000000C (12)
Flags	2	Bits [4:1]: Mask the CH; when one of these 4 bits is set to 1, the CH item is not analyzed: [1]: SETTINGS [2]: RAM [3]: FLASH (GPMC) [4]: MMCSD Others: Reserved
1 <sup>st</sup> device	2	Devices to be put into the device list 0x00: Void, no device 0x01: XIP memory 0x02: XIPWAIT memory (wait-signal monitoring on)
2 <sup>nd</sup> device	2	0x03: NAND 0x04: OneNAND 0x05: MMC1 0x06: MMC2(1)
3 <sup>rd</sup> device	2	0x07: MMC2(2) 0x08: EMIF 0x43: UART 0x45: USB (internal transceiver)
4 <sup>th</sup> device	2	0x46: USB-ULPI Others: Reserved
Padding	2	
<b>Clock Settings</b>		
Section 2 key	4	Synchronization key for section 2: 0xCF00AA02
Section 2 size	4	Size of section 2: 0x00000064 (100)
Flags	4	Bit mask of various switches, active when set to 1: Bit [0]: Clock configuration defined in this structure is applied. Bit [1]: Reserved Bit [2]: Apply general clock settings. Bit [3]: Set and lock PER DPLL. Bit [4]: Set and lock MPU DPLL. Bit [5]: Set and lock Core DPLL. Bit [6]: Set and lock USB DPLL. Bit [7]: Bypass PER DPLL before setting clocks. Bit [8]: Bypass MPU DPLL before setting clocks. Bit [9]: Bypass Core DPLL before setting clocks. Bit [10]: Bypass USB DPLL before setting clocks. Bit [11]: Perform DLL control configuration. Bits [31:12]: Reserved
<b>General Clock Settings</b>		
CM_CLKSEL_CORE	4	Register value
CM_DLL_CTRL	4	Register value
<b>MPU DPLL Settings</b>		
CM_AUTOIDLE_DPLL_MPU	4	Register value
CM_CLKSEL_DPLL_MPU	4	Register value
CM_DIV_M2_DPLL_MPU	4	Register value

**Table 27-13. Software Booting Configuration Structure (continued)**

Field	Size (Bytes)	Description
<b>Core DPLL Settings</b>		
CM_AUTOIDLE_DPLL_CORE	4	Register value
CM_CLKSEL_DPLL_CORE	4	Register value
CM_DIV_M2_DPLL_CORE	4	Register value
CM_DIV_M3_DPLL_CORE	4	Register value
CM_DIV_M4_DPLL_CORE	4	Register value
CM_DIV_M5_DPLL_CORE	4	Register value
CM_DIV_M6_DPLL_CORE	4	Register value
CM_DIV_M7_DPLL_CORE	4	Register value
<b>PER DPLL Settings</b>		
CM_AUTOIDLE_DPLL_PER	4	Register value
CM_CLKSEL_DPLL_PER	4	Register value
CM_DIV_M2_DPLL_PER	4	Register value
CM_DIV_M3_DPLL_PER	4	Register value
CM_DIV_M4_DPLL_PER	4	Register value
CM_DIV_M5_DPLL_PER	4	Register value
CM_DIV_M6_DPLL_PER	4	Register value
CM_DIV_M7_DPLL_PER	4	Register value
<b>USB DPLL Settings</b>		
CM_AUTOIDLE_DPLL_USB	4	Register value
CM_CLKSEL_DPLL_USB	4	Register value
CM_DIV_M2_DPLL_USB	4	Register value
<b>Peripheral Booting Timeout Configuration</b>		
Section 3 key	4	Synchronization key for section 3: 0xCF00AA03
Section 3 size	4	Size of the section 3: 0x00000008 (8)
Flags	2	Bit [0]: If cleared (= 0), the timeout USB field is ignored. Bit [1]: If cleared (= 0), the timeout boot message field is ignored. Other bits: Reserved
Timeout USB	2	Maximum time allowed for the host to complete the USB enumeration. <ul style="list-style-type: none"> <li>• 0x0000: OMAP device will wait indefinitely for the host to complete USB enumeration.</li> <li>• Others: The value expressed in milliseconds defines the maximum time the OMAP device will wait for the host to complete the USB enumeration (maximum 65.5 s).</li> </ul>
Timeout boot message	2	Maximum time allowed for the host to for the host to send a booting message. <ul style="list-style-type: none"> <li>• 0x0000: OMAP device will wait indefinitely for the host to send a booting message.</li> <li>• Others: The value expressed in milliseconds defines the maximum time the OMAP device will wait for the host to send a booting message (maximum 65.5 s).</li> </ul>
Padding	2	

## 27.4.5 Peripheral Booting

### 27.4.5.1 Description

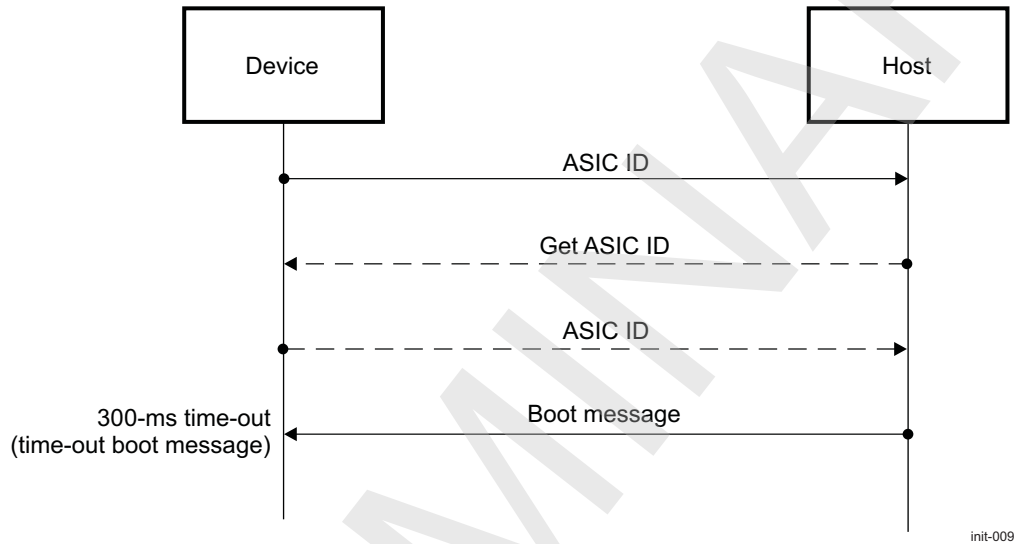
The ROM code can boot from two different peripherals:

- USB: High/full speed USB from internal or external transceiver
- UART3: 115.2 Kbps, 8 bits, even parity, 1 stop-bit

The purpose of booting from a peripheral is to download a flash loader code from an external host. This booting method is used primarily for programming flash memories connected to the OMAP device (for example, in the case of initial flashing, firmware update or servicing). The overall peripheral booting procedure is shown in [Figure 27-12](#). It consists of a synchronization phase (handshake between host and OMAP device) and a transfer phase. The synchronization phase is similar for UART and USB boots. Both transfer phases use the same procedure.

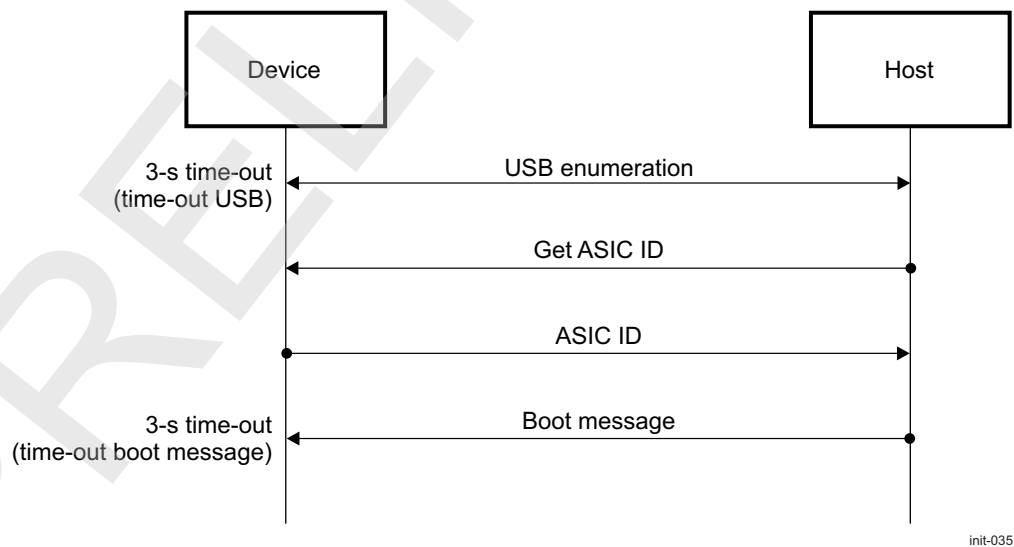
When booting from the UART, the ROM code first initializes the UART3 interface. Then the ROM code sends an ASIC ID block of data. From there, it expects to receive a boot message from the host within 300 ms, by default. [Figure 27-10](#) shows this procedure.

**Figure 27-10. Synchronization Phase for UART**



[Figure 27-11](#) shows the procedure when booting from the USB.

**Figure 27-11. Synchronization Phase for USB**



During the synchronization phase ([Figure 27-10](#) and [Figure 27-11](#)), the OMAP device can provide a small packet of data called the ASIC ID (described in [Table 27-14](#)). It is a simple structure that contains different kinds of information, such as ROM version, checksums, and ID.

The default time-outs involved during the synchronization phase can be redefined after a warm reset by means of the software booting configuration (described in [Section 27.4.4.4, Software Booting Configuration](#)). The host can decide the desired operation by providing a booting message ([Table 27-18](#)). This message can be: Get ASIC ID, peripheral boot, change device, or next device. If the OMAP device receives the Get ASIC ID boot message, it sends back the ASIC ID contents.

If the change device or next device message is received, the ROM code stops the current peripheral booting procedure and returns to the main booting, which decides about the next booting device according to the boot message received.

If the peripheral boot message is received without a time-out, the OMAP device is entering the transfer phase. From there, the flash loader image size (as a 32-bit word) and the flash loader image itself are expected to be received. The ROM code waits up to 1 minute for completion of image size reception, and up to 1 more minute to download the image. If the download procedure does not complete before this time, the peripheral booting returns with a TIMEOUT status. ROM code continues to examine the devices included in the booting device list. If the download procedure passes, the peripheral booting returns with a SUCCESS status and the image can be executed.

The USB or UART connection is left open at the end of the transfer phase and once exiting the ROM code for the initial software to take over. It means the initial software can “reuse” the currently established connection. In case of USB, the endpoints can be reused as such, without closing the connection and performing a full enumeration again.

**Table 27-14. ASIC ID Structure**

ASIC ID Item	Size (Bytes)	Description
Items	1	Number of subblocks
ID subblock	7	Device identification information
Reserved subblock	4	This subblock is transmitted, but does not contain useful information in case of GP-device.
Reserved subblock	23	This subblock is transmitted, but does not contain useful information in case of GP-device.
Reserved subblock	35	This subblock is transmitted, but does not contain useful information in case of GP-device.
Checksum subblock	11	CRC (4 bytes)

**Table 27-15. Items**

Offset	Size (Bytes)	Description
0x00	1	0x05: Number of subblocks USB 0x04: Number of subblocks UART <sup>(1)</sup>

<sup>(1)</sup> The checksum subblock is not transmitted over the UART.

**Table 27-16. ID Subblock**

Offset	Size (Bytes)	Description
0x01	1	0x01: Subblock ID
0x02	1	0x05: Subblock size
0x03	1	0x01: Fixed value
0x04	2	0x44, 0x30: OMAP4430 device
0x06	1	0x07: CH enabled (read from eFuse) 0x17: CH disabled (read from eFuse)
0x07	1	ROM revision

**Table 27-17. Checksum Subblock<sup>(1)</sup>**

Offset	Size (Bytes)	Description
0x46	1	0x15: Subblock ID
0x47	1	0x09: Subblock size

<sup>(1)</sup> The checksum subblock is not transmitted over the UART.

**Table 27-17. Checksum Subblock<sup>(1)</sup> (continued)**

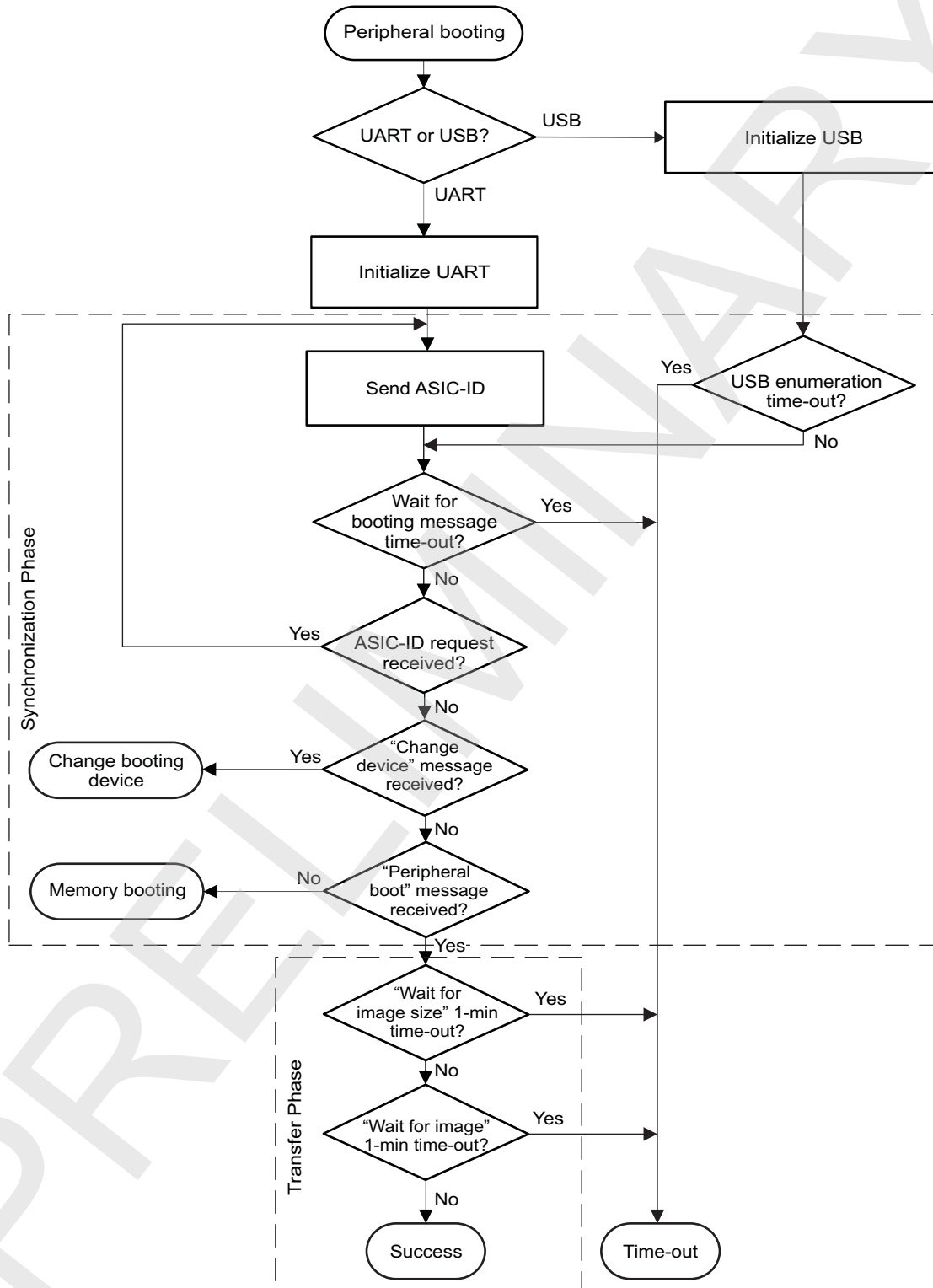
Offset	Size (Bytes)	Description
0x48	1	0x01: Fixed value
0x49	4	ROM CRC
0x4D	4	0x00000000: For GP device

**Table 27-18. Booting Messages**

Message Name	Value	Description
Peripheral boot	0xF0030002	Continue peripheral booting.
Get ASIC ID	0xF0030003	ASIC ID request. The Get ASIC ID request message is optional. If received, the ROM code sends its ASIC ID data to the host in return. The host can issue the Get ASIC ID message multiple times if required. The structure of the ASIC ID is described in <a href="#">Table 27-14</a> .
Change device	0xF003xx06	Skip current peripheral booting and continue booting from device type indicated by xx: 0x00: Void, no device 0x01: XIP memory 0x02: XIPWAIT memory (wait-signal monitoring on) 0x03: NAND 0x04: OneNAND 0x05: MMC1 0x06: MMC2(1) 0x07: MMC2(2) 0x08: EMIF 0x43: UART 3 0x45: USB (internal transceiver) 0x46: USB-ULPI Others: Reserved
Next device	0xFFFFFFFF	Skip current device and move to the next device on the device list.
Memory booting	Others	Skip current peripheral booting and move to the first device for memory booting.

Figure 27-12 shows the peripheral booting procedure.

Figure 27-12. Peripheral Booting Procedure



init-010

### 27.4.5.2 UART

The ROM code supports booting from a UART interface with the following characteristics:

- UART interface 3
- Communication parameters set to 115.2 Kbps, 8 bits, even parity, 1 stop-bit
- Two-pin interface: RX/TX
- The boot message default time-out is 300 ms (time-out boot message)

### 27.4.5.3 USB

The ROM code supports booting from a USB interface with the following characteristics:

- HSUSBOTG interface (USBA0)
- OMAP integrated transceiver (USBPHY) or external transceiver through the ULPI
- Enumeration default time-out is 3 s (timeout USB)
- The boot message default time-out is 3 s (timeout boot message).

---

**NOTE:** Even though using an OTG capable hardware, the ROM code does not handle any OTG-specific features.

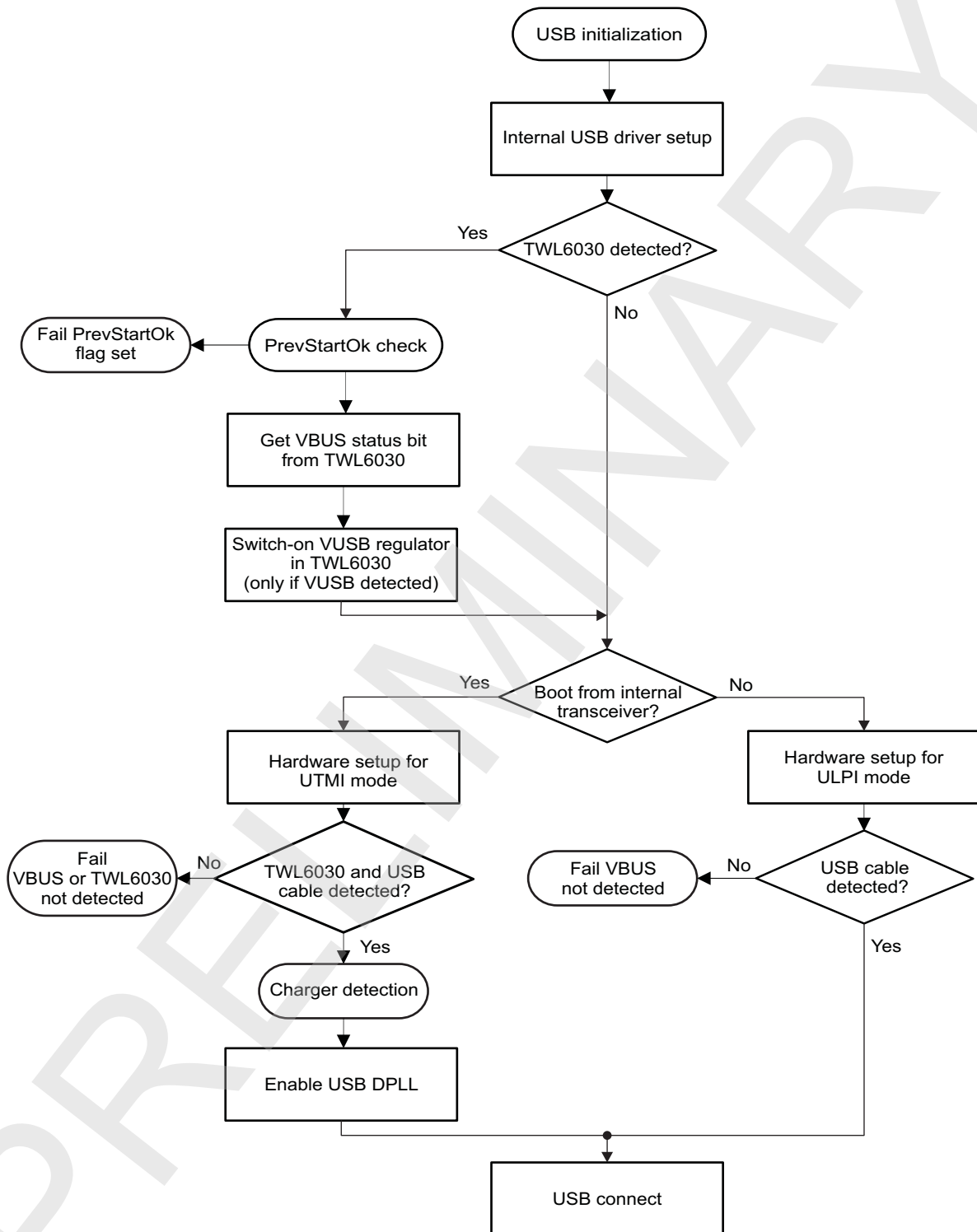
---

#### 27.4.5.3.1 Overview

[Figure 27-13](#) shows the USB initialization procedure.



Figure 27-13. USB Initialization Procedure



init-037

After internal USB module driver setup, the TWL6030 power IC is checked for presence. If found, the following actions are performed using I<sup>2</sup>C transactions through the I2C1 interface:

- PrevStartOk status flag check, as described in [Section 27.4.5.3.2](#).
- Get the VBUS status reflecting presence (or not) of the USB cable
- If VBUS was detected, then the VUSB regulator is enabled to power the internal OMAP transceiver.

In case of boot from external transceiver in accordance with the chosen SYSBOOT configuration:

- The HSUSBOTG hardware and PRCM clocks are configured for ULPI mode.
- The ROM code continues with the USB procedure only if the USB cable is detected (that is, VBUS is detected at external transceiver level and communicated as such through the ULPI traffic). If not, the initialization procedure is aborted.

---

**NOTE:** The OMAP USB DPLL is not set up in case of ULPI mode, for which the 60-MHz clock is driven by the external transceiver.

---

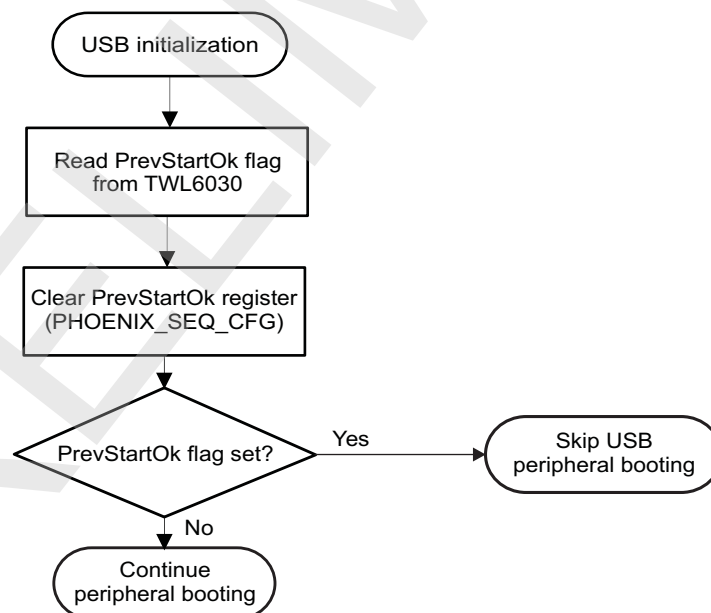
In case of boot from internal transceiver in accordance with the chosen SYSBOOT configuration:

- The USBOTGHS hardware and PRCM clocks are configured for UTMI mode and internal PHY clocks are enabled.
- The ROM code continues with the USB procedure only if TWL6030 (or emulator) is detected and the USB cable is detected (that is, VBUS is detected at the TWL6030 level and read as such with a register read through I<sup>2</sup>C). If not, the initialization procedure is aborted.
- The ROM code proceeds with charger detection, as described in [Section 27.4.5.3.2](#).
- It enables the USB DPLL for feeding the appropriate 60-MHz clock needed by the internal transceiver.

### 27.4.5.3.2 PrevStartOk Status Flag Check

The ROM code supports a mechanism for avoiding the boot time spent because of the enumeration time-out (default 3 s) in case the USB cable is attached and the USB option is present in the selected boot list but is not intended to initiate a peripheral booting procedure (for example, firmware update or flashing). This mechanism is jointly handled with the support of the TWL6030 device (not supported if TWL6030 is not detected). This mechanism is supported only for USB and not UART (see [Figure 27-14, PrevStartOk Mechanism](#)).

**Figure 27-14. PrevStartOk Mechanism**



init-038

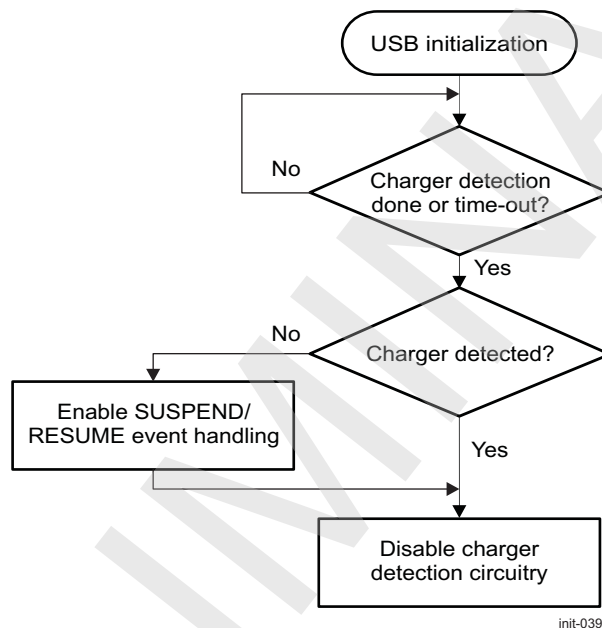
The ROM code first checks the value of a register in the TWL6030 RTC domain (hence, retained by the backup battery when the platform is switched off) by means of an I<sup>2</sup>C read access. If the PrevStartOk bit in this register is set, the USB peripheral booting is simply skipped. In any case, the bit is reset (zeroed) by

the ROM code, which means that the USB peripheral boot is re-enabled on the next cold reset (or warm reset if suitable software booting configuration is present). The initial software of the user must set this bit again by an I<sup>2</sup>C write access to disable USB peripheral booting on a subsequent reset. For more information about the accesses made towards PrevStartOk in the PHOENIX\_SEQ\_CFG register, see TWL6030 documentation.

### 27.4.5.3.3 Charger Detection

The ROM code proceeds with charger detection as part of the USB peripheral booting procedure and in compliance with the *Battery Charging Specification Revision 1.1*. It uses the specific charger detection hardware embedded in the OMAP device. Figure 27-15 shows the high-level procedure.

Figure 27-15. Charger Detection



The hardware charger detection mechanism should end up within the 500-ms time-out. The main goal of such detection is to gate the charger circuitry in TWL6030 in case a charger is not detected and a suspend event is detected on the OMAP side, for which the power consumption must be reduced.

### 27.4.5.3.4 SUSPEND and RESUME Handling

The SUSPEND event is detected by the HSUSBOTG module when activity on the bus stops for at least 3 ms. During peripheral booting time, a SUSPEND event is detected by the ROM code and processed by initiating a write into the TWL6030 to gate the charging circuitry. The ROM code releases the TWL6030 charger circuitry upon getting a RESUME event. During run time, SUSPEND and RESUME interrupts can be routed to the initial software for defining custom handling.

### 27.4.5.3.5 USB Driver Descriptors

USB devices report their attributes using descriptors. A descriptor is a data structure with a defined format. Each descriptor begins with a byte-wide field that contains the total number of bytes in the descriptor followed by a byte-wide field that identifies the descriptor type. Using descriptors allows concise storage of the attributes of individual configurations so that each configuration can reuse descriptors or portions of descriptors from other configurations that have the same characteristics. Where appropriate, descriptors contain references to string descriptors. String descriptors contain displayable, human-readable description information. These descriptor details can be used for tool development or debugging:

- Device descriptor

A device descriptor contains general information about a USB device, including global information that applies to the device and all device configurations. A USB device has only one device descriptor. A device-qualifier descriptor is required because the ROM code uses the HS feature of the USB core. [Table 27-19](#) lists the device descriptors.

**Table 27-19. Device Descriptor**

Field	Value	Description
bLength	0x12	Size of this descriptor in bytes
bDescriptorType	0x01	Device descriptor type
bcdUSB	0x0210	USB specification release number in binary coded decimal (BCD) format
bDeviceClass	Vendor-specific (0xFF)	Class code
bDeviceSubClass	Vendor-specific (0xFF)	Subclass code
bDeviceProtocol	Vendor-specific (0xFF)	Protocol code
bMaxPacketSize0	0x40	Maximum packet size for endpoint 0
idVendor	0x0451	Vendor ID (Texas Instruments), TI default value
idProduct	0xD00F	Product ID (OMAP4430), TI default value
bcdDevice	0x0000	Device release number
iManufacturer	See <a href="#">Section 27.4.5.3.6</a> .	Index of string descriptor describing manufacturer
iProduct	See <a href="#">Section 27.4.5.3.6</a> .	Index of string descriptor describing product
iSerialNumber	See <a href="#">Section 27.4.5.3.6</a> .	Index of string descriptor describing device serial number
bNumConfigurations	0x01	Number of possible configurations

- Device-qualifier descriptor

The device-qualifier descriptor contains information about a HS-capable device that changes if the device operates at its other speed. This descriptor is retrieved by the host using the GetDescriptor() request (standard device request). [Table 27-20](#) describes a device-qualifier descriptor.

**Table 27-20. Device-Qualifier Descriptor**

Field	Value	Description
bLength	0x0a	Size of this descriptor in bytes
bDescriptorType	0x06	Device-qualifier descriptor type
bcdUSB	0x0210	USB specification release number in BCD
bDeviceClass	0xFF	Class code
bDeviceSubClass	0xFF	Subclass code
bDeviceProtocol	0xFF	Protocol code
bMaxPacketSize0	0x40	Maximum packet size for endpoint 0
bNumConfigurations	0x01	Number of possible configurations
bReserved	0x00	Reserved for future use

- Configuration descriptor

This descriptor gives information about a specific device configuration. It describes the number of interfaces supported by the configuration (see [Table 27-21](#)).

**Table 27-21. Configuration Descriptor**

Field	Value	Description
bLength	0x09	Size of this descriptor in bytes
bDescriptorType	0x02	Configuration descriptor type
wTotalLength	–	Combined length of all descriptors
bNumInterfaces	0x01	Number of interfaces supported
bConfigurationValue	0x01	Value to use as an argument for the SetConfiguration() request
iConfiguration	Index	Index of string descriptor describing this configuration
bmAttributes	0xC0	Power setting and remote wakeup

**Table 27-21. Configuration Descriptor (continued)**

Field	Value	Description
bMaxPower	0x32	Maximum power consumption of the USB device

- Other speed configuration descriptor  
This descriptor describes the configuration of a HS-capable device if it operates at its other possible speed (see [Table 27-22](#)).

**Table 27-22. Other Speed Configuration Descriptor**

Field	Value	Description
bLength	0x09	Size of this descriptor in bytes
bDescriptorType	0x07	Other speed configuration descriptor type
wTotalLength	–	Combined length of all descriptors
bNumInterfaces	0x01	Number of interfaces supported
bConfigurationValue	0x01	Value to use as an argument for the SetConfiguration() request
iConfiguration	Index	Index of string descriptor describing this configuration
bmAttributes	0xC0	Power setting and remote wakeup
bMaxPower	0x32	Maximum power consumption of the USB device

- Interface descriptor  
This descriptor describes a specific interface in a configuration (see [Table 27-23](#)).

**Table 27-23. Interface Descriptor**

Field	Value	Description
bLength	0x09	Size of this descriptor in bytes
bDescriptorType	0x04	Interface descriptor type
bInterfaceNumber	0x00	Number of this descriptor
bAlternateSetting	0x00	Value to select the alternate setting
bNumEndpoints	0x02	Number of endpoints used for this interface
bInterfaceClass	0xFF	Class code
bInterfaceSubClass	0xFF	Subclass code
bInterfaceProtocol	0xFF	Protocol code
iInterface	Index	Index of string descriptor describing this interface

- Endpoint descriptor  
Each endpoint used for an interface has its own descriptor. This descriptor contains information required by the host to determine the bandwidth requirements of each endpoint. This descriptor is returned as part of the GetDescriptor(Configuration) request. (see [Table 27-24](#) and [Table 27-25](#)).

**Table 27-24. BULK IN Endpoint Descriptor**

Field	Value	Description
bLength	0x07	Size of this descriptor in bytes
bDescriptorType	0x05	Endpoint descriptor type
bEndpointAddress	0x81 (1 IN)	Address of the endpoint on the USB device
bmAttributes	0x02 (Bulk)	Type of transfer
wMaxPacketSize	See <sup>(1)</sup> .	Number of endpoints used for this interface
bInterval	0x00	Maximum NAK rate

<sup>(1)</sup> The maximum size is 0x0200 (512 bytes) for HS bulk endpoint and 0x0040 (64 bytes) for FS bulk endpoint.

**Table 27-25. BULK OUT Endpoint Descriptor**

Field	Value	Description
bLength	0x07	Size of this descriptor in bytes
bDescriptorType	0x05	Endpoint descriptor type
bEndpointAddress	0x01 (1 OUT)	Address of the endpoint on the USB device
bmAttributes	0x02 (Bulk)	Type of transfer
wMaxPacketSize	See <sup>(1)</sup> .	Number of endpoints used for this interface
bInterval	0x00	Maximum NAK rate

<sup>(1)</sup> The maximum size is 0x0200 (512 bytes) for HS bulk endpoint and 0x0040 (64 bytes) for FS bulk endpoint.

- String descriptors

String descriptors use UNICODE encoding. The strings in a USB device can support multiple languages. When requesting a string descriptor, the requester specifies the desired language using a 16-bit language ID (LANGID) defined by the USB interface. String index 0 for all languages returns a string descriptor that contains an array of 2-byte LANGID codes supported by the device.

For the description of the different string descriptors, see:

- The language ID string descriptor ([Table 27-26](#))
- The manufacturer ID string descriptor ([Table 27-27](#))
- The product ID string descriptor ([Table 27-28](#))
- The configuration string descriptor ([Table 27-29](#))
- The interface string descriptor ([Table 27-30](#))

**Table 27-26. Language ID String Descriptor**

Field	Value	Description
bLength	0x04	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type
wLangId	0x0409 (US English)	Language ID code

**Table 27-27. Manufacturer ID String Descriptor**

Field	Value	Description
bLength	0x24	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type
bString	Texas Instruments	Manufacturer string

**Table 27-28. Product ID String Descriptor**

Field	Value	Description
bLength	0x12	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type
bString	OMAP4430	Product string

**Table 27-29. Configuration String Descriptor**

Field	Value	Description
bLength	0x08	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type
bString	psc	Configuration string

**Table 27-30. Interface String Descriptor**

Field	Value	Description
bLength	0x08	Size of this descriptor in bytes
bDescriptorType	0x03	String descriptor type

**Table 27-30. Interface String Descriptor (continued)**

Field	Value	Description
bString	pbi	Interface string

**27.4.5.3.6 USB Customized Descriptors**

There are two parameters in USB descriptors that customers can define after the chip is created: vendor ID (VID) and product ID (PID). The ROM code uses dedicated eFuses that hold VID and PID values. Other parameters can also be changed based on the VID value. The ROM code has an encoded set of parameters for customers who have defined their requirements before the ROM code has been done. [Table 27-31](#) lists the parameters that depend on the VID value.

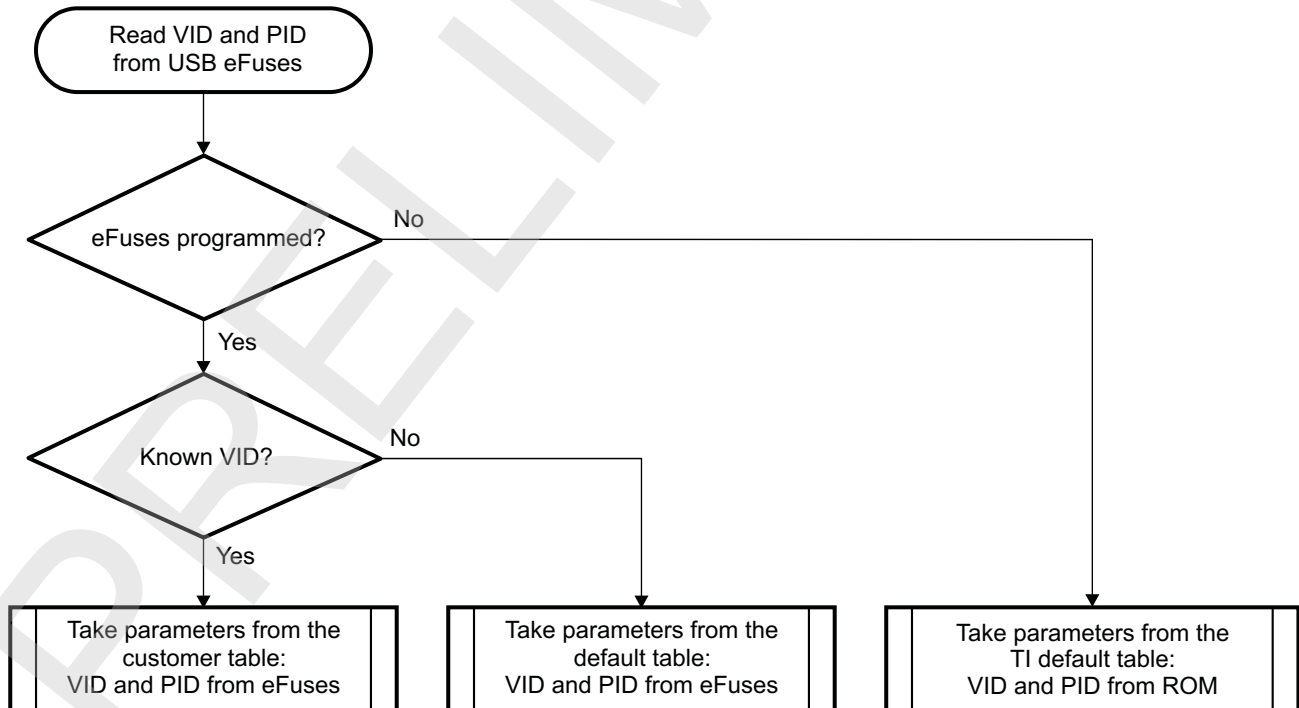
**Table 27-31. Customized Descriptor Parameters**

Parameter	Size (Bytes)	Default Values (From eFuses)	TI Default Values (From ROM)
Device ID code	2	0x0000	0x0000
Device class	1	0xFF	0xFF
Device subclass	1	0xFF	0xFF
Device protocol	1	0xFF	0xFF
Manufacturer	String	N/A	Texas Instruments
Product	String	OMAP4430	OMAP4430
Serial number	String	See <sup>(1)</sup>	See <sup>(1)</sup>

<sup>(1)</sup> The standard device descriptor indicates that the device has no serial number.

[Figure 27-16](#) shows an additional customer parameter selection method. It is based on the VID burned in the USB eFuses.

**Figure 27-16. Customer USB Descriptor Selection**

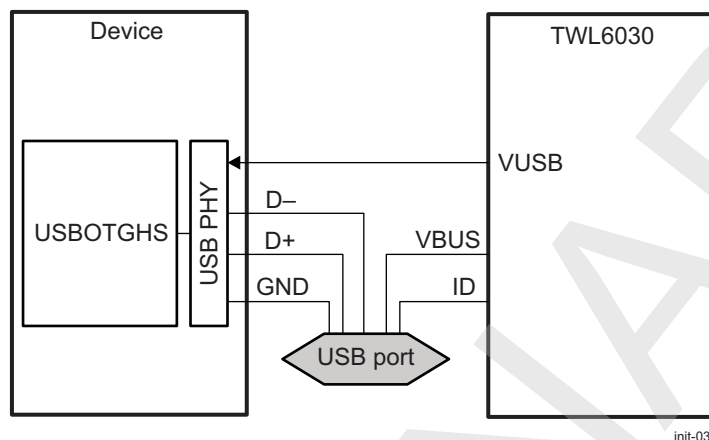


init-011

### 27.4.5.3.7 System Interconnection – Integrated Transceiver

When SYSBOOT selects the embedded PHY, the USB power/data path can be described as shown in Figure 27-17.

**Figure 27-17. USB System Using Internal Transceiver**



### 27.4.5.3.8 System Interconnection - External Transceiver

The USBOTGHS module uses the ultralow-pin-count interface (ULPI) to interconnect with an external USB transceiver. In this case, the ROM code does not interact with the transceiver. Thus, users must ensure that the transceiver power supply and clock are present and the transceiver is out of reset.

### 27.4.5.3.9 USB Driver Functionality

- Transactions supported:
  - Control transactions: Used for standard device requests
  - Bulk transactions: Used for data transfer in the image downloading stage. The ASIC ID is sent to the Bulk IN endpoint and the image is transferred from the host over the Bulk OUT endpoint.

The OMAP USB device first attaches to the host as an FS device. In the reset mechanism, the USB core requests HS operation. If the HS negotiation in the reset phase is successful, further transactions are at HS; otherwise, they are at FS. After reset, the USB driver checks for the speed of the device, whether it is FS or HS. Depending on the speed configured by the host, the standard USB device requests are answered with the corresponding descriptors.

- Standard device request restrictions:
 

Some standard device requests are not supported by the driver because the USB driver is dedicated for use by the ROM code for peripheral booting. Table 27-32 lists the standard device requests supported by the driver.

**Table 27-32. Standard Device Requests Supported**

Request	Description	Support
CLEAR_FEATURE	Sets/clears a specific feature	Supported only for ENDPOINT_HALT feature
GET_CONFIGURATION	Returns the current device configuration value	Yes
GET_DESCRIPTOR	Returns the specified descriptor	Yes
GET_INTERFACE	Returns the selected alternate setting for the specified interface	Yes
GET_STATUS	Returns the status for the specified recipient	Yes
SET_ADDRESS	Sets the device address	Yes
SET_CONFIGURATION	Sets the device configuration	Yes
SET_DESCRIPTOR	Updates existing descriptors or adds new descriptors	Runtime updating of descriptors is not supported.
SET_FEATURE	Sets or enables a specific feature	Supported only for ENDPOINT_HALT feature



**Table 27-32. Standard Device Requests Supported (continued)**

<b>Request</b>	<b>Description</b>	<b>Support</b>
SET_INTERFACE	Selects an alternate setting in an interface	Runtime setting of alternate features is not supported.
SYNCH_FRAME	Sets and reports an endpoint synchronization frame	No, because isochronous transfers are not used

## **27.4.6 Fast External Booting**

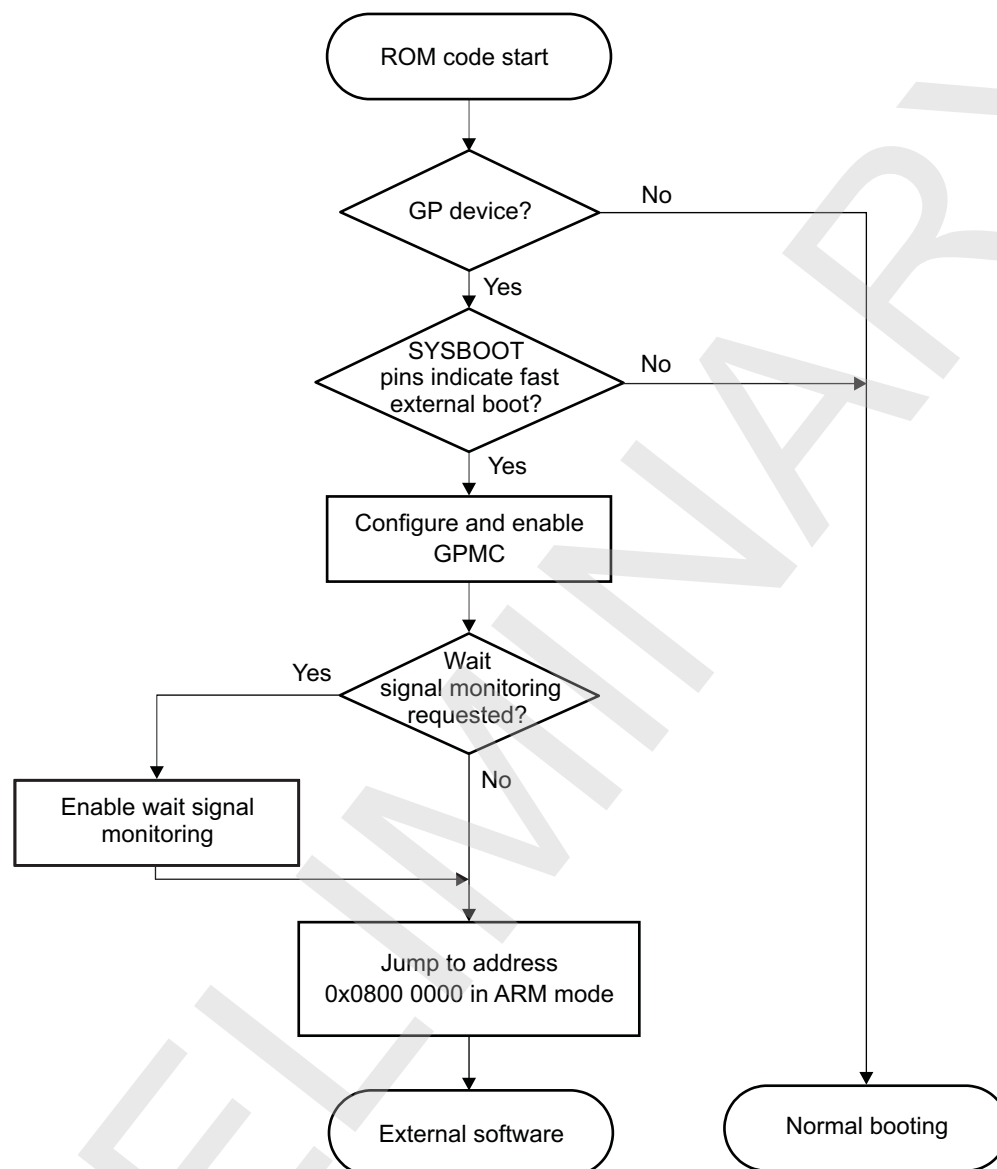
### **27.4.6.1 Overview**

The fast external boot is a special memory booting mode, possible only on GP devices. It consists of a blind jump to a code in an external XIP memory device connected to GPMC CS0. Fast external booting is set up by means of the SYSBOOT configuration pins and lets customers create their own booting code.

The jump is performed with minimum on-chip ROM code execution.

### **27.4.6.2 Fast External Booting Procedure**

[Figure 27-18](#) shows the fast external boot procedure. The code is at the beginning and is written in assembly. The code does not use any RAM and is designed for fast execution.

**Figure 27-18. Fast External Boot Procedure**

init-013

## 27.4.7 Memory Booting

### 27.4.7.1 Overview

The memory booting process starts an external code in memory devices. Only memory type of booting devices can be used by ROM code as permanent booting devices (that is, devices examined after both cold (POR) and warm resets). Temporary booting devices are examined only after cold resets. The supported permanent booting devices are:

- NOR flash devices
- NAND flash devices
- OneNAND/Flex-OneNAND devices
- eMMC/eSD embedded flash memories
- SD/MMC flash cards
- LPDDR2-NVM

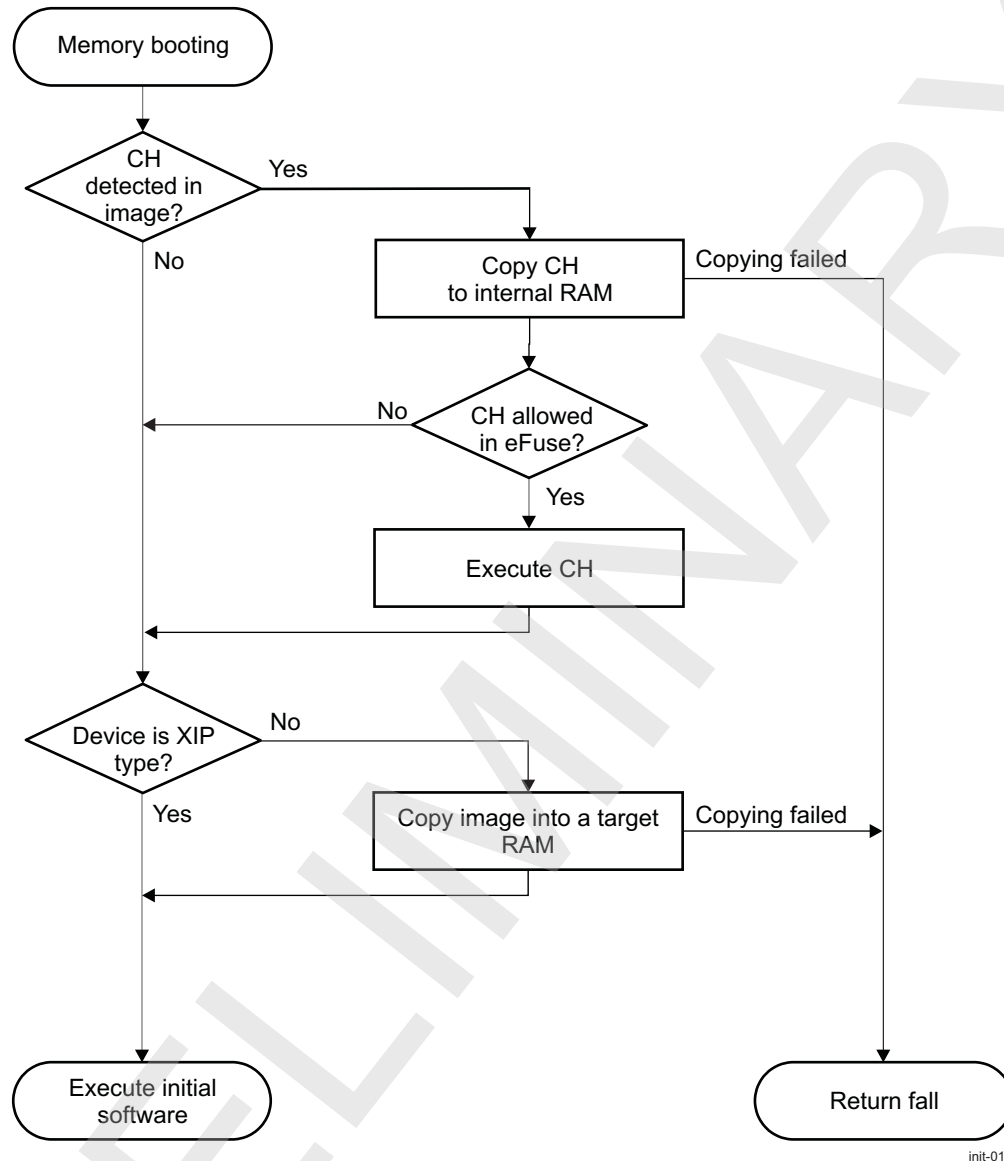
Two main groups of permanent booting devices are distinguished by code shadowing. Code shadowing means copying code from a nondirectly addressable device (non-XIP) to RAM, where the code can be executed. Directly addressable devices are XIP devices.

Figure 27-19 shows the general memory booting procedure common to all types of devices. First, CH is copied to internal RAM. It is copied even for XIP devices, because the OMAP device can temporarily lose a connection with XIP memory during CH execution. The second step is to shadow the image, if the device is not XIP. The last step is image execution and any return from image results in a dead loop.

If CH copying or shadowing fails, memory booting returns to the main booting procedure, which selects the next device for booting.

During the first read sector (512 bytes) call, sectors are copied to a temporary RAM buffer. Once the image is found and the destination address is known, the content of the temporary buffer is moved to the target RAM location so it is required to reread the first image sector. On a GP device, the GP header is discarded; therefore, only executable code is located in RAM, with the first executable instruction located at the destination address.

eMMC/eSD embedded memories, MMC/SD cards, NAND, and OneNAND/Flex-OneNAND devices can hold up to four copies of the booting image. Therefore, the ROM code searches for one valid image out of the four, if present, by walking over the first blocks of mass storage space. Other XIP devices (NOR) use only one copy of the booting image.

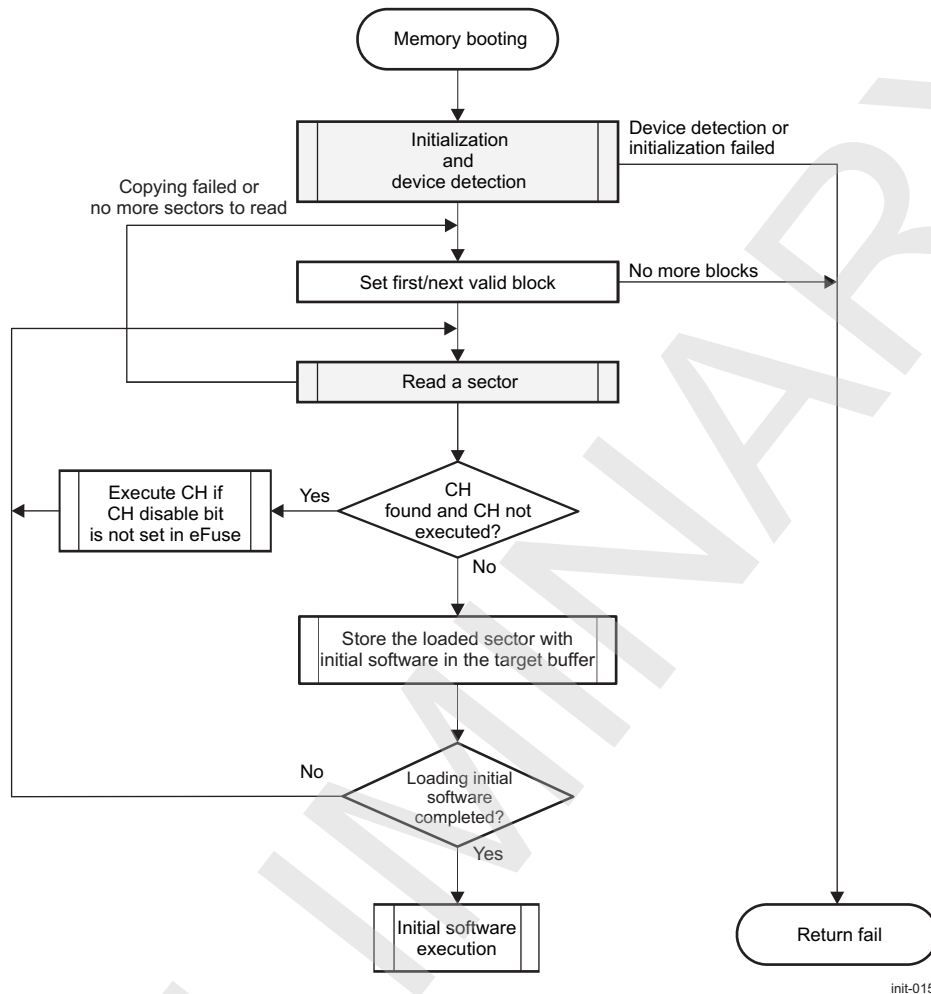
**Figure 27-19. Memory Booting Procedure**

### 27.4.7.2 Non-XIP Memory

Figure 27-20 shows the procedure used when memory booting runs with non-XIP devices. The grayed procedures are specific to each device. NAND and OneNAND/Flex-OneNAND devices use up to four copies of the image in the first four physical blocks. Therefore, the ROM code searches for the image in the first four physical blocks of these devices. Other devices use only one copy of the image and the block loop runs only once.

During image shadowing on a GP device, the CH is expected to be in a separate sector before the initial software.

Figure 27-20. Image Shadowing on GP Device



init-015

Table 27-33 summarizes the number of blocks and sectors searched during memory booting from devices requiring image shadowing. A sector is assumed to be 512 bytes long. NANDs and OneNAND/Flex-OneNAND are organized with blocks, which are erasable units. MMC/SD/eMMC/eSD + FAT12/16/32 booting consists of reading a file.

Table 27-33. Blocks and Sectors Searched on Non-XIP Memories

Memory	Maximum Number of Checked Blocks	Number of Sectors Searched
NAND	First 4	Number of sectors in a block <sup>(1)</sup>
OneNAND/Flex-OneNAND	First 4	8
eMMC/eSD raw	First 4	1
MMC/SD raw	First 4	1
MMC/SD/eMMC/eSD + FAT12/16/32	1 File	

<sup>(1)</sup> Depends on NAND geometry

For more information about the GPMC module, see Section 15.4, GPMC Overview.

The following sections describe the supported device types.

### 27.4.7.3 XIP Memory

The ROM code can boot directly from XIP devices, such as NOR flash memories, that have the following characteristics:

- The GPMC is the communication interface.
- Memories up to 1 Gb (128MB) can be connected.
- x16 data bus width only
- Asynchronous protocol and address/data multiplexed mode
- The GPMC clock is 50 MHz.
- The device is connected to CS0 mapped to address 0x0800 0000.
- The wait pin signal gpmc\_wait0 is monitored according to the sys\_boot configuration pins.

Depending on the SYSBOOT option, the GPMC can be configured to use the wait signal connected to the gpmc\_wait0 pin. Wait pin polarity is set to stall accessing memory when gpmc\_wait0 is low. Wait-signal monitoring is used with memories that require a long time for initialization after reset, or that must pause while reading data.

For an XIP memory booting, no user intervention is required; the following debugging steps are described. Only CH, which is not mandatory, lets users change clock settings and GPMC parameters. Failure in CH copying causes a return to the main booting procedure, which selects the next device.

Bootting from an XIP device consists of the following steps:

1. Configure the GPMC for XIP device access.
2. Verify that the CH is present at address 0x0800 0000. If it is, copy the entire sector (512 bytes) to internal RAM and execute CH.
3. Set the image location:
  - 0x0800 0000 if the CH is not found
  - 0x0800 0200 if the CH is found
4. Verify that a bootable image is at the image location.
5. Execute the image if it is found.
6. If the image is not found, return from XIP booting to the main booting loop.

#### 27.4.7.3.1 GPMC Initialization

Table 27-34 lists the timing settings of the GPMC when set for XIP and other address-data accessible devices, such as OneNAND/Flex-OneNAND. Table 27-34 is included for debug information.

**Table 27-34. XIP Timing Parameters**

Parameter	Value [Clock Cycles] (1)	Register Initialization (where i = 0–7)	Reset Value
Write cycle period	17	The GPMC_CONFIG5_i[12:8] WRCYCLETIME bit field is set to 0x11.	0x11
Read cycle period	17	The GPMC_CONFIG5_i[4:0] RDCYCLETIME bit field is set to 0x11.	0x11
CS low time	1	The GPMC_CONFIG2_i[3:0] CSONTIME bit field is set to 0x1.	0x1
CS high time	16	The GPMC_CONFIG2_i[12:8] CSRDOFFTIME bit field is set to 0x10.	0x10
ADV low time	1	The GPMC_CONFIG3_i[3:0] ADVONTIME bit field is set to 0x1.	0x1
ADV high time	2	The GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field is set to 0x2.	0x2
OE low time	3	The GPMC_CONFIG4_i[3:0] OEONTIME bit field is set to 0x3.	0x6
OE high time	16	The GPMC_CONFIG4_i[12:8] OEOFFTIME bit field is set to 0x10.	0x10
WE low time	3	The GPMC_CONFIG4_i[19:16] WEONTIME bit field is set to 0x3.	0x05
WE high time	15	The GPMC_CONFIG4_i[28:24] WEOFFTIME bit field is set to 0xF.	0x10
Data latch time	15	The GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field is set to 0xF.	0x0F

(1) The one clock cycle is 20 ns, which corresponds to 50-MHz frequency.

There is no specific identification routine executed before booting from an XIP device.

#### 27.4.7.4 NAND

NAND flash memory is not an XIP device; it requires shadowing before the code can be executed. ROM code support for the NAND flash devices has the following characteristics:

- The GPMC is the communication interface.
- Device from 512 Mbit (64MB)
- x8 and x16 bus width
- Support for large page size (2048 bytes + 64 spare bytes) or very large page size 4096 bytes + 128/218 spare bytes)
- Chip enable (CE) don't care devices only
- Single-level cell (SLC) and multilevel cell (MLC) devices
- Device identification based on ONFI or ROM table
- ECC correction: 8 bits/sector for most devices (16 bits/sector for devices with large spare area)
- GPMC timings are adjusted for NAND access.
- The GPMC clock is 50 MHz.
- The device is connected to CS0.
- The wait pin signal gpmc\_wait0 is connected to the NAND BUSY output.
- Four physical blocks are searched for an image. Block size depends on the device.

For NAND memory booting, no user intervention is needed; the information in the following subsections is included for debugging. Only the CH, which is not mandatory, lets the user change clock settings and GPMC parameters. Failure in CH copying causes a return to the main booting procedure, which selects the next device for booting.

#### 27.4.7.4.1 Initialization and NAND Detection

The initialization routine for NAND consists of three parts: GPMC initialization, device detection with parameter determination, and bad block detection.

- GPMC initialization

The GPMC interface is configured so that it can access NAND. Because NAND memories do not need the address bus, it is released. The data bus width is initially set to 8 bits. If necessary, it is changed to 16 bits after the device parameters are determined. [Table 27-35](#) shows the GPMC configuration used during NAND boot. [Table 27-35](#) is included for debug information.

**Table 27-35. NAND Timing Parameters**

Parameter	Value (Clock Cycles)	Register Initialization (where i = 0–7)	Reset Value
Write cycle time	20	GPMC_CONFIG5_i[12:8] WRCYCLETIME = 0x14	0x11
Read cycle time	20	GPMC_CONFIG5_i[4:0] RDCYCLETIME = 0x14	0x11
CS low time	0	GPMC_CONFIG2_i[3:0] CSONTIME = 0x0	0x1
CS low to OE low time	5	GPMC_CONFIG4_i[3:0] OEONTIME = 0x5	0x6
CS low to OE high time	16	GPMC_CONFIG4_i[12:8] OEOFFTIME = 0x10	0x10
CS low to WE low time	1	GPMC_CONFIG4_i[19:16] WEONTIME = 0x1	0x5
CS low to WE high time	15	GPMC_CONFIG4_i[28:24] WEOFFTIME = 0xF	0x10
CS low to data latch time	14	GPMC_CONFIG5_i[20:16] RDACCESSTIME = 0xE	0xF

- Device detection and parameters

The ROM code first performs an initial wait for device auto initialization (with 250-ms time-out) with polling of the ready information. Then, it needs to identify the NAND type connected to the GPMC interface. The GPMC is initialized using 8 bits, asynchronous mode. The NAND device is reset (command FFh) and its status is polled until ready for operation (with 100-ms time-out). The ONFI Read ID (command 90h/address 20h) is sent to the NAND device. If it replies with the ONFI signature (4 bytes) then a Read parameters page (command ECh) is sent. The information provided in [Table 27-36](#) is then extracted: page size, spare area size, number of pages per block, and the addressing mode. The remaining data bytes from the parameter page stream are simply ignored.

**Table 27-36. ONFI Parameters Page Description**

Offset	Description	Size (Bytes)
6	Features supported	2
80	Number of data bytes per page	4
84	Number of spare bytes per page	2
92	Number of pages per block	4
101	Number of address cycles	1

If the ONFI Read ID command fails (it will be the case with any device not supporting ONFI) then the device is reset again with polling for device to be ready (with 100ms timeout). Then, the standard Read ID (command 90h / address 00h) is sent. If the Device ID (2nd byte of the ID byte stream) is recognized as being a supported device then the device parameters are extracted from an internal ROM Code table. [Table 27-37](#) lists the supported devices.

**Table 27-37. Supported NAND Devices**

Capacity	Device ID	Bus Width	Page Size in Bytes
512 Mb	F0h	8	2048
512 Mb	C0h	16	2048
512 Mb	A0h	8	2048
512 Mb	B0h	16	2048
512 Mb	F2h	8	2048
512 Mb	C2h	16	2048
512 Mb	A2h	8	2048
512 Mb	B2h	16	2048
1 Gb	F1h	8	2048
1 Gb	C1h	16	2048
1 Gb	A1h	8	2048
1 Gb	B1h	16	2048
2 Gb	DAh	8	2048
2 Gb	CAh	16	2048
2 Gb	AAh	8	2048
2 Gb	BAh	16	2048
2 Gb	83h	8	2048
2 Gb	93h	16	2048
4 Gb	DCh	8	2048
4 Gb	CCh	16	2048
4 Gb	ACh	8	2048/4096
4 Gb	BCh	16	2048/4096
4 Gb	84h	8	2048
4 Gb	94h	16	2048
8 Gb	D3h	8	2048/4096
8 Gb	C3h	16	2048/4096
8 Gb	A3h	8	2048/4096
8 Gb	B3h	16	2048/4096
8 Gb	85h	8	2048
8 Gb	95h	16	2048
16 Gb	D5h	8	2048/4096
16 Gb	C5h	16	2048/4096
16 Gb	A5h	8	2048/4096
16 Gb	B5h	16	2048/4096
16 Gb	86h	8	2048



**Table 27-37. Supported NAND Devices (continued)**

Capacity	Device ID	Bus Width	Page Size in Bytes
16 Gb	96h	16	2048
32 Gb	D7h	8	2048/4096
32 Gb	C7h	16	2048/4096
32 Gb	A7h	8	2048/4096
32 Gb	B7h	16	2048/4096
32 Gb	87h	8	2048
32 Gb	97h	16	2048
64 Gb	DEh	8	2048/4096
64 Gb	CEh	16	2048/4096
64 Gb	AEh	8	2048/4096
64 Gb	BEh	16	2048/4096

After retrieving parameters from the table, the page size and block size are updated based on the fourth byte of the NAND ID data. Because of inconsistency among manufacturers, only devices recognized to be at least 2Gb have these parameters updated. Therefore, the ROM code supports 4-KB page devices, but only if their size, according to the table, is at least 2Gb. Devices that are smaller than 2Gb have the block size parameter set to 128KB (when the page size is 2KB).

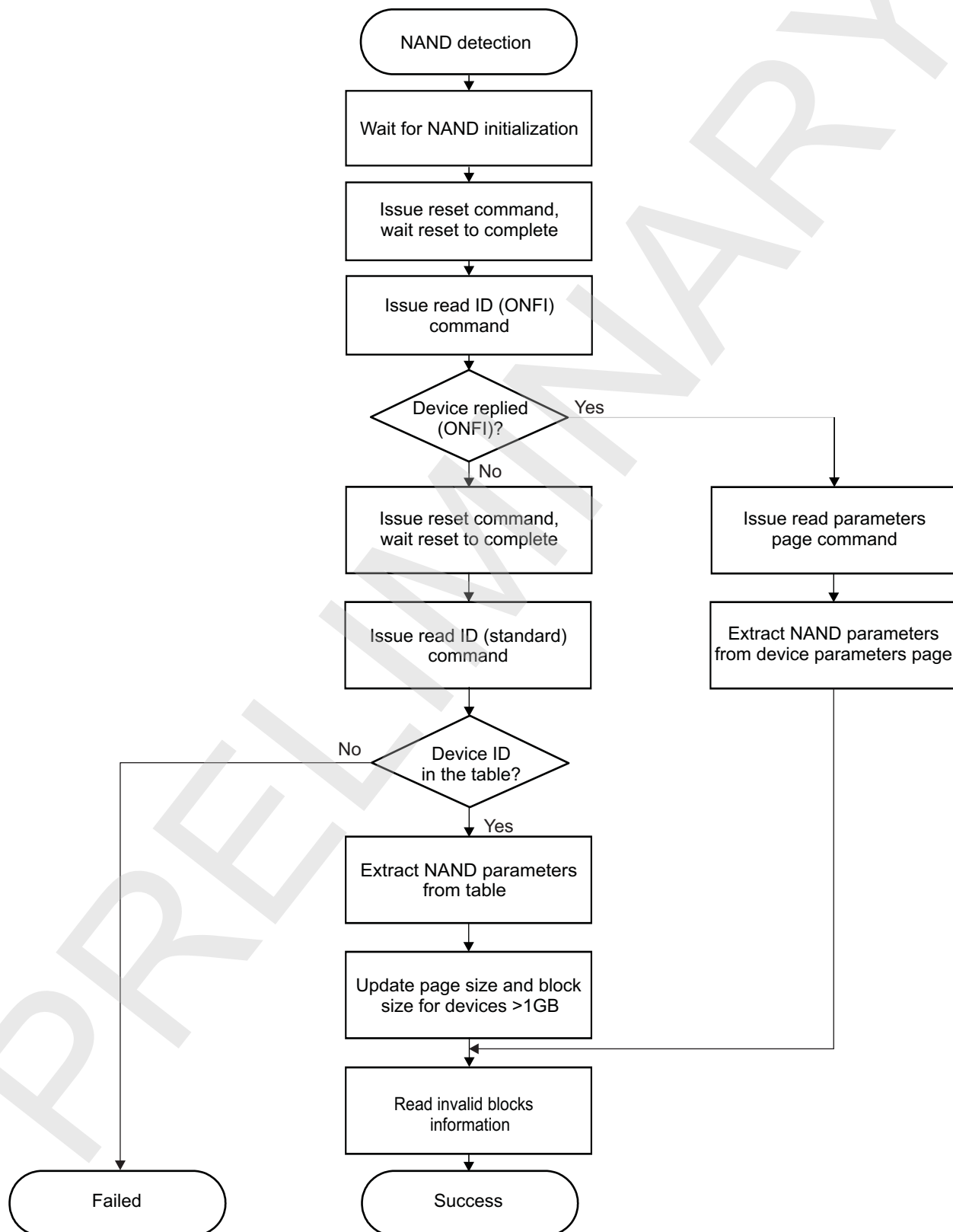
[Table 27-38](#) shows the fourth ID data byte encoding used in the ROM code.

**Table 27-38. Fourth NAND ID Data Byte**

Item	Description	I/O Number							
		7	6	5	4	3	2	1	0
<b>Page Size</b>	1KB							0	0
	2KB							0	1
	4KB							1	0
	8KB							1	1
<b>Cell type</b>	2 levels						0	0	
	4 levels						0	1	
	8 levels						1	0	
	16 levels						1	1	
<b>Block Size</b>	64KB				0	0			
	128KB				0	1			
	256KB				1	0			
	512KB				1	1			

Figure 27-21 shows the detection procedure. Once the NAND device is successfully detected, the ROM code changes the GPMC to 16-bit bus width, if necessary.

**Figure 27-21. NAND Device Detection**

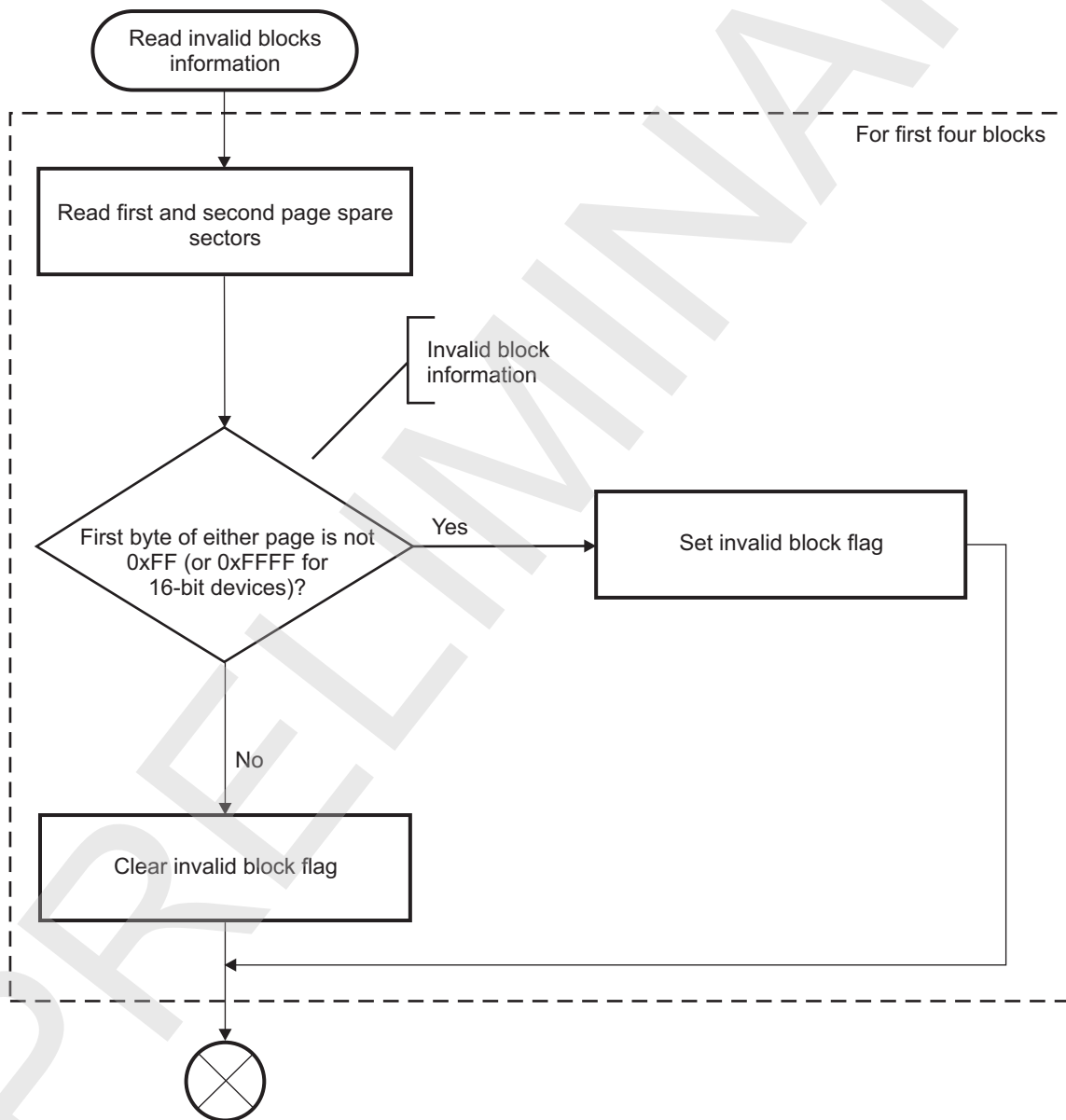


init-023

- **Bad block verification**  
 Invalid blocks contain invalid bits whose reliability cannot be ensured by the manufacturer. These bits are identified in the factory or during the programming and reported in the initial invalid block information in the spare area on the first and second page of each block. Because the ROM code looks for an image in the first four blocks, it detects the validity status of these blocks. Blocks detected as invalid are not accessed later. Block validity status is coded in the spare areas of the first two pages of a block (first byte equal to FFh in the first and second pages for an 8-bit device/first word equal to FFFFh in the first and second pages for a 16-bit device).

Figure 27-22 shows the invalid block detection routine. The routine consists in reading spare areas and checking the validity data pattern.

**Figure 27-22. Bad NAND Invalid Block Detection**



init-033

**27.4.7.4.2 Read Sector Procedure**

During the booting procedure, the ROM code reads 512-byte sectors from the NAND device. The reading fails in two cases:

- The accessed sector is in a block marked as invalid.
- The accessed sector contains an error that cannot be corrected with ECC.

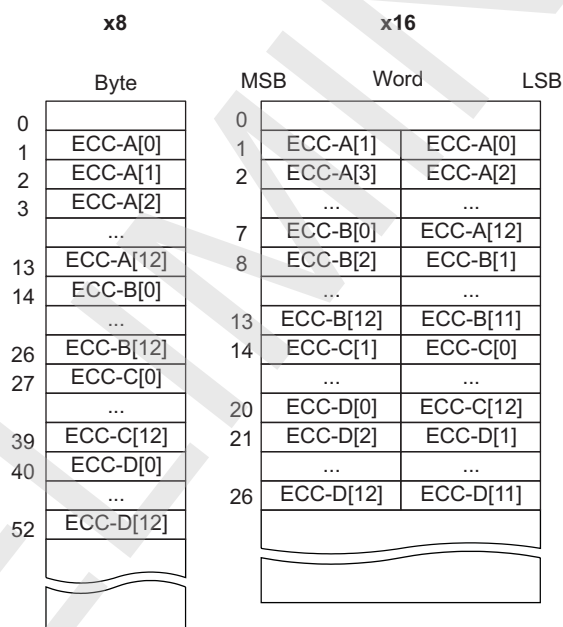
The ROM code uses normal read (command 00h 30h) for reading NAND page data.

Page data can contain errors caused by memory alteration. The ROM code uses an ECC correction algorithm to detect and possibly correct those errors. The default ECC correction applied is BCH 8b/sector using the GPMC and ELM hardware.

For device ID codes D3h, C3h, D5h, C5h, D7h, C7h, DEh, CEh when the manufacturer code (first ID byte) is 98h, the cell type information is checked in the 4th byte of ID data. If it is equal to 10b, the ECC correction applied is BCH 16b/sector.

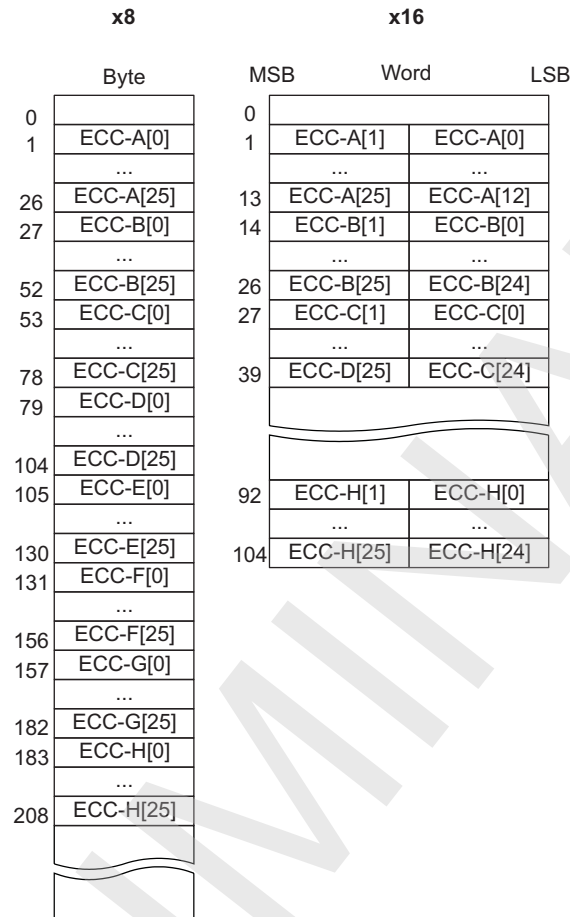
The BCH data is automatically calculated by the GPMC on reading each 512-byte sector. The computed ECC is compared against the ECC stored in the spare area for the corresponding page. Depending on the page size, the amount of ECC data bytes stored in the corresponding spare area is different. [Figure 27-23](#) and [Figure 27-24](#) show the mapping of ECC data inside the spare area for 2KB-page and 4KB- page devices, respectively. If both ECC data are equal, the read sector function returns the read 512-byte sector without error. Otherwise, the ROM code tries to correct error(s) in the corresponding sector (this procedure is assisted by the ELM hardware) and returns the data if successful. If errors are uncorrectable, the function returns with FAIL.

**Figure 27-23. ECC Data Mapping for 2-KB Page and 8b BCH Encoding**



init-040

Figure 27-24. ECC Data Mapping for 4-KB Page and 16b BCH Encoding



init-041

### 27.4.7.5 OneNAND/Flex-OneNAND

The ROM code support for OneNAND/Flex-OneNAND devices has the following characteristics:

- Devices from 512 Mbits
- The GPMC is the communication interface.
- x16 data bus width only
- Asynchronous protocol and address/data multiplexed mode
- GPMC reset default timings are used.
- The GPMC clock is 50 MHz.
- The device is connected to GPMC CS0 mapped to address 0x0800 0000.
- The wait-pin signal gpmc\_wait0 is not monitored.
- Four physical blocks are searched for image. The block size is 128KB (typically).
- Data correction is fully handled by the OneNAND/FlexOneNAND device

The OneNAND/Flex-OneNAND device is a NAND matrix coupled with RAM buffers and a NOR-type interface. ECC correction handling is done automatically by the internal state-machine. The page to be accessed is first loaded in the RAM buffer using memory-mapped registers. Then, the page is read directly from the buffer using a NOR-type interface.

For OneNAND/Flex-OneNAND memory booting, no user intervention is required. The information in the following sections is included for debugging. Only the CH, which is not mandatory, lets users change clock settings and GPMC parameters. Failure in CH copying causes a return to the main booting procedure, which selects the next device for booting.

#### 27.4.7.5.1 Initialization and OneNAND/Flex-OneNAND Detection

The initialization routine for OneNAND/Flex-OneNAND consists of two parts: GPMC initialization and device detection with parameter determination.

- GPMC initialization

The ROM code first initializes the GPMC interface in XIP mode (that is, asynchronous 16-bit multiplexed mode). Wait signal monitoring is disabled.

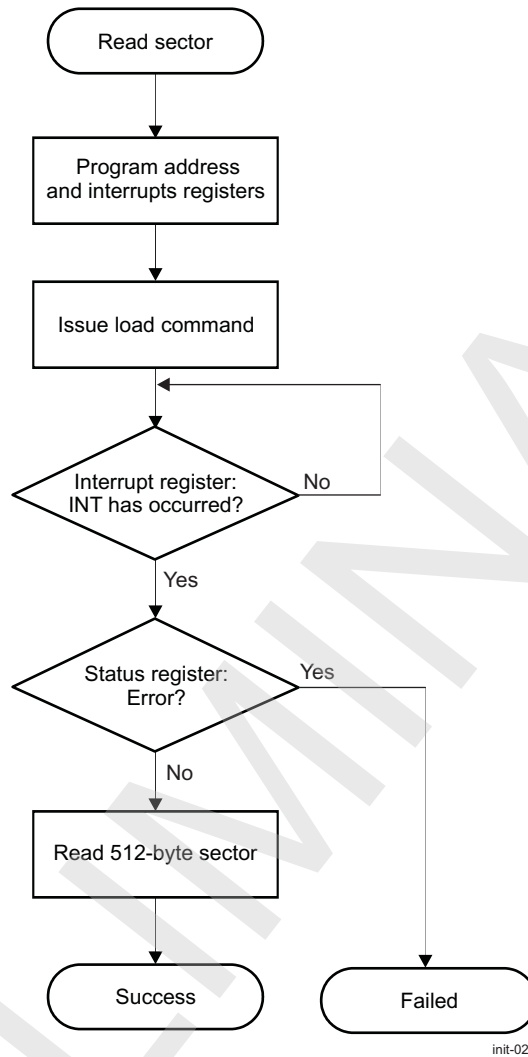
- Device detection and parameters

The ROM code identifies a OneNAND/Flex-OneNAND device by reading the device identification data. There are two ways to read identification data: using serial commands and reading from fixed memory-mapped registers. The ROM code reads identification data using both methods and compares the results. When the comparison passes, the ROM code assumes that the OneNAND/Flex-OneNAND device is connected. If the device is successfully recognized, the ROM code reads the device configuration (number and size of data buffers) and configures it for asynchronous mode (default).

#### 27.4.7.5.2 OneNAND/Flex-OneNAND Read Sector Procedure

When booting requests a sector from the OneNAND/Flex-OneNAND device, the ROM code performs the load operation, which transfers the content of the requested sector to the data buffer RAM. The ROM code waits until the operation completes, polling the OneNAND/Flex-OneNAND interrupt register. The status register is then checked and the ROM code returns FAIL if the operation completes with an error. Otherwise, the data buffer RAM is copied to the destination buffer. [Figure 27-25](#) shows this procedure.

Figure 27-25. OneNAND/Flex-OneNAND Read Sector



**27.4.7.5.3 OneNAND/Flex-OneNAND Support Limitations**

As described in [Section 27.4.7.5.1, Initialization and OneNAND/Flex-OneNAND Detection](#), the ROM code checks only the coherency of the device ID obtained by two different methods. There is no table of supported Flex-OneNANDs in ROM code. This removes any dependency of the ROM code on device IDs from different manufacturers. However, the driver works with the following assumption:

- Page size is assumed to be 2048 bytes, divided into four sectors of 512 bytes. In case of OneNAND/Flex-OneNAND, with pages of 4096 bytes in size, divided into eight sectors, users must program only the first sectors (sectors 0, 1, 2, 3) of each page. Sectors 4, 5, 6, and 7 of each page are not considered by the driver.

**27.4.7.6 eMMC/eSD Embedded Memories and MMC/SD Cards**

The OMAP4430 device allows booting from eMMC/eSD embedded memories or MMC/SD cards connected to SD/MMC1 or SD/MMC2. The booting interface is selected by configuration of the sys\_boot pins.

The HS MMC/SD/SDIO host controllers (MMCHS) handle the physical layer, while the ROM code handles the simplified logical protocol layer (read-only protocol). A limited range of commands is implemented in the ROM code.

The MMC interface 1 supports 1.8V/3V digital I/Os. The selection on I/O voltage level is done using the PBIAS circuitry. The PBIAS reference level is sensed by the ROM code and PBIAS cell voltage setup appropriately before any communication with the device. The MMC interface 2 supports only 1.8-V digital I/Os.

The OMAP4430 device supports two booting modes:

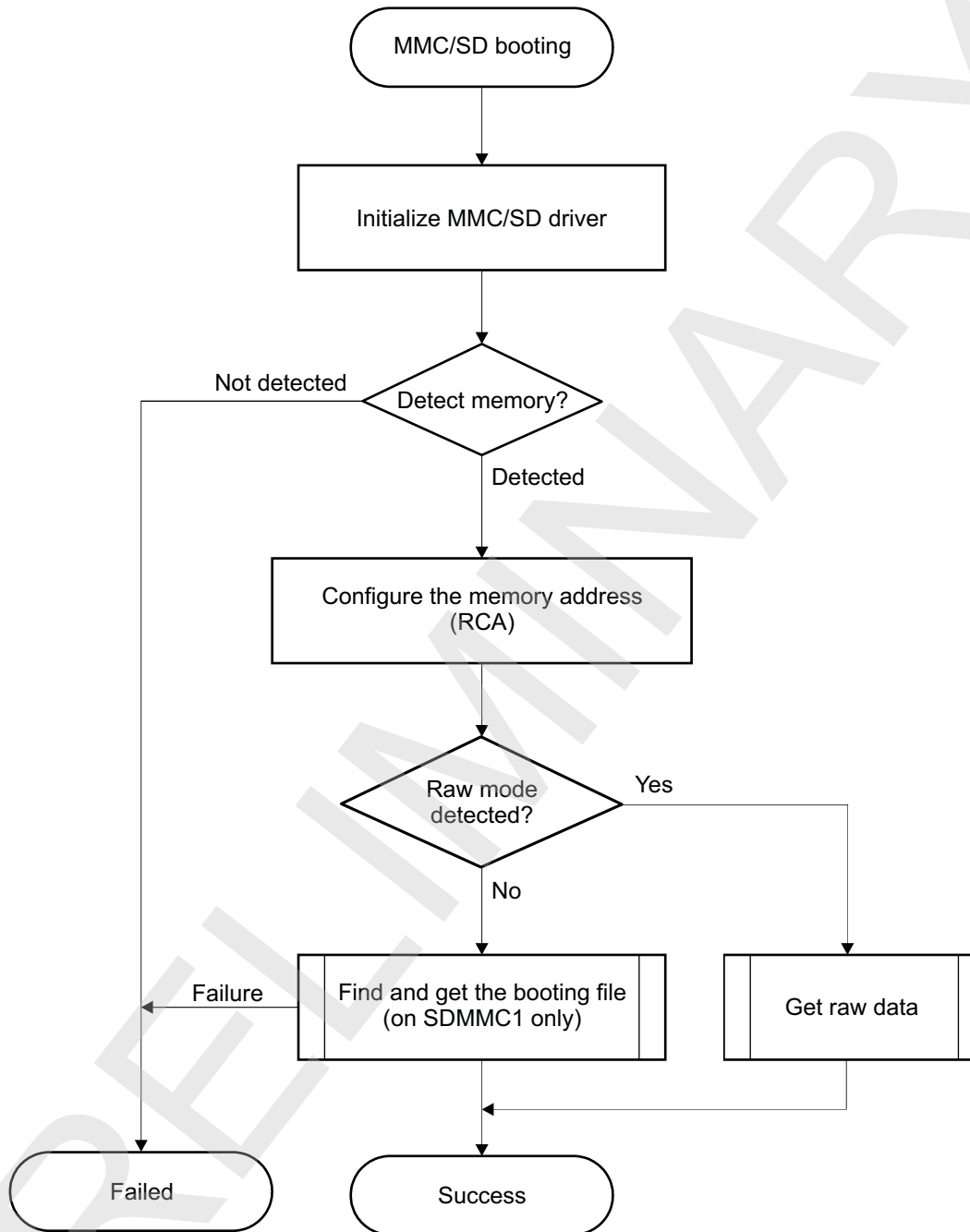
- Raw: The booting image data is read directly from sectors in the user area.
- File system (FAT12/16/32 supported with or without master boot record): The image data is read from a booting file.

When using the SDMMC1 interface, raw and file system modes are supported. When using the SDMMC2 interface, only raw mode is supported. The purpose of this approach is to avoid the boot time penalty of searching for a file system hierarchy when it is not always necessary.

[Figure 27-26](#) shows the complete procedure.



Figure 27-26. eMMC/eSD and MMC/SD Booting



init-016

### 27.4.7.6.1 eMMC/eSD Overview

The ROM code supports eMMC/eSD cards, with the following conditions:

- eMMC devices compliant with *Embedded MultiMediaCard (eMMC) eMMC/Card Product Standard, High Capacity, including Reliable Write Boot, and Sleep Modes, Dual Data Rate, Multiple Partitions Supports and Security Enhancement v4.4* from the MMCA Technical Committee. The exception is the hardware reset feature. If the user requires hardware reset, it can be accomplished with a GPIO.
- eSD devices compliant with *SD Specification Part 1 eSD (Embedded SD) Specification Addendum v2.1* from the SD Card Association
- The eMMC/eSD memory device is powered externally by a power-management IC or other power

supply.

- Initial 1-bit MMC mode, optional 4-bit or 8-bit modes.
- Clock frequency:
  - Identification mode: 400 kHz
  - Data transfer mode: Up to 10 MHz, optionally up to 19.2 MHz
- Only one memory connected to the bus

#### 27.4.7.6.1.1 System Conditions and Limitations

The ROM code does not support the specific details of the partition management and boot operations modes as defined in the eMMC Standard sections *Partition Management* and *Boot Operation Mode*. The booting image is assumed to be located in the User Data Area (no use of boot and/or RPMB partitions). Customers must set up the eMMC device appropriately so that the booting image is located in a reliable area within the User Data Area (possibly with the Enhanced User Data Area property).

The ROM code does not provide a bus direction control signaling in case of using MMC interface 2 with level shifters.

The ROM code expects that the eMMC device is powered externally and supplies are set and stable upon entering the booting procedure. The ROM code does not perform any software action to the companion device with respect to powering the eMMC/eSD device.

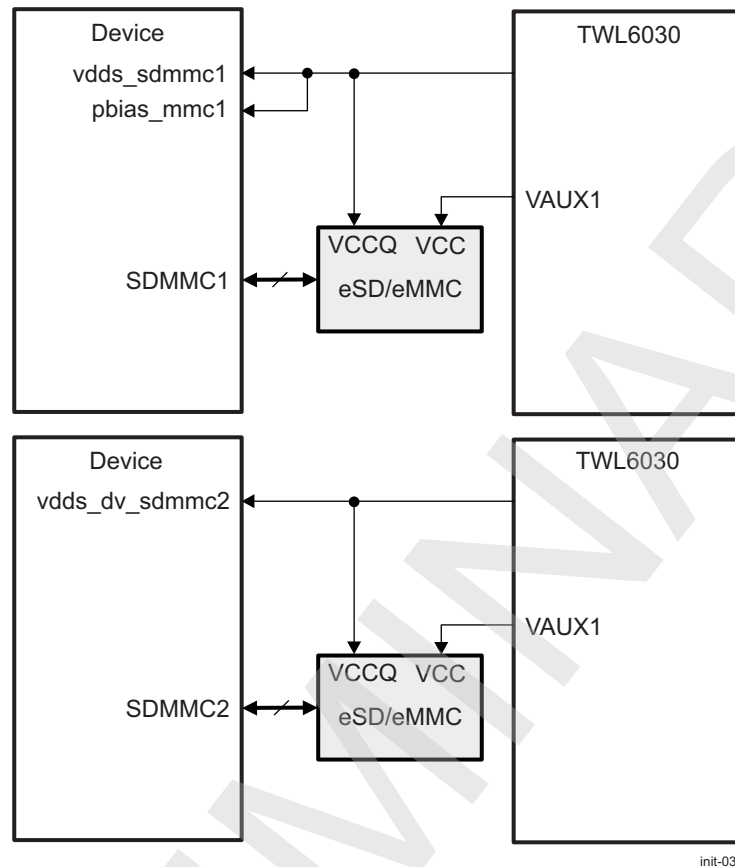
#### 27.4.7.6.1.2 eMMC/eSD Memory Connection

An eMMC/eSD device typically requires two supplies: one for the core memory array (selectable 1.8/2.9 V at TWL6030 BOOT2 pin level) (see TWL6030 documentation), and one for the device interface I/Os and controller (typical VCCQ = 1.8 V). Both memory device supply pins can possibly be merged into one, thus requiring only one power supply.

The SD/MMC2 interface shares the same balls as the GPMC interface. Therefore, the ROM code performs the necessary pin mux configuration to route the MMC signal through the GPMC balls.

[Figure 27-27](#) shows an example of the system connection between the power-management IC, memory device, and OMAP device.

Figure 27-27. eMMC/eSD Connection



The ROM code supports the muxing of the MMC interface 2 through two paths:

- GPMC pins in configuration mode 1 (Mux1)
- SDMMC5/FREF\_CLKx pins in Mux5

The ROM code performs the necessary I/O pin muxing configuration to route the MMC2 signals through the appropriate pins depending on the selected SYSBOOT configuration.

#### 27.4.7.6.2 MMC/SD Overview

The ROM code supports booting from MMC/SD cards under the following conditions:

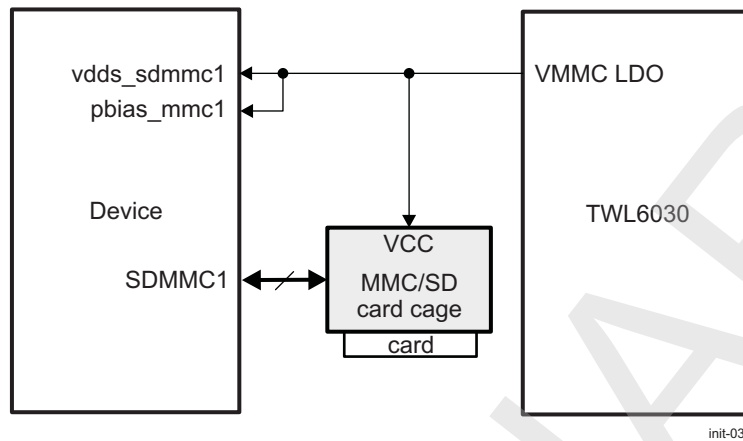
- MMC/SD cards compliant with *The MultiMediaCard System Specification* from the MMCA Technical Committee and the *SD Memory Card Specifications, File System Specification* from the SD Association. These Include low- and high-capacity cards
- MMC/SD cards connected to SDMMC interface 1
- 3-V VCC power supply, support for 3-V or 1.8-V I/O voltages
- Initial 1-bit MMC mode, optional 4-bit or 8-bit modes
- Clock frequency:
  - Identification mode: 400 kHz
  - Data transfer mode: Up to 10 MHz (optionally up to 19.2 MHz)
- Only one card connected to the bus

##### 27.4.7.6.2.1 MMC/SD Card Connection

An SD/MMC card can be connected to the MMC1 interface, typically through a card cage. The ROM code sends the appropriate I<sup>2</sup>C sequence to the power-management IC to set the VMMC LDO to 3 V and switch it on.

Figure 27-28 shows the typical connection between the power IC, the card, and the OMAP device .

**Figure 27-28. MMC/SD Card Connection**



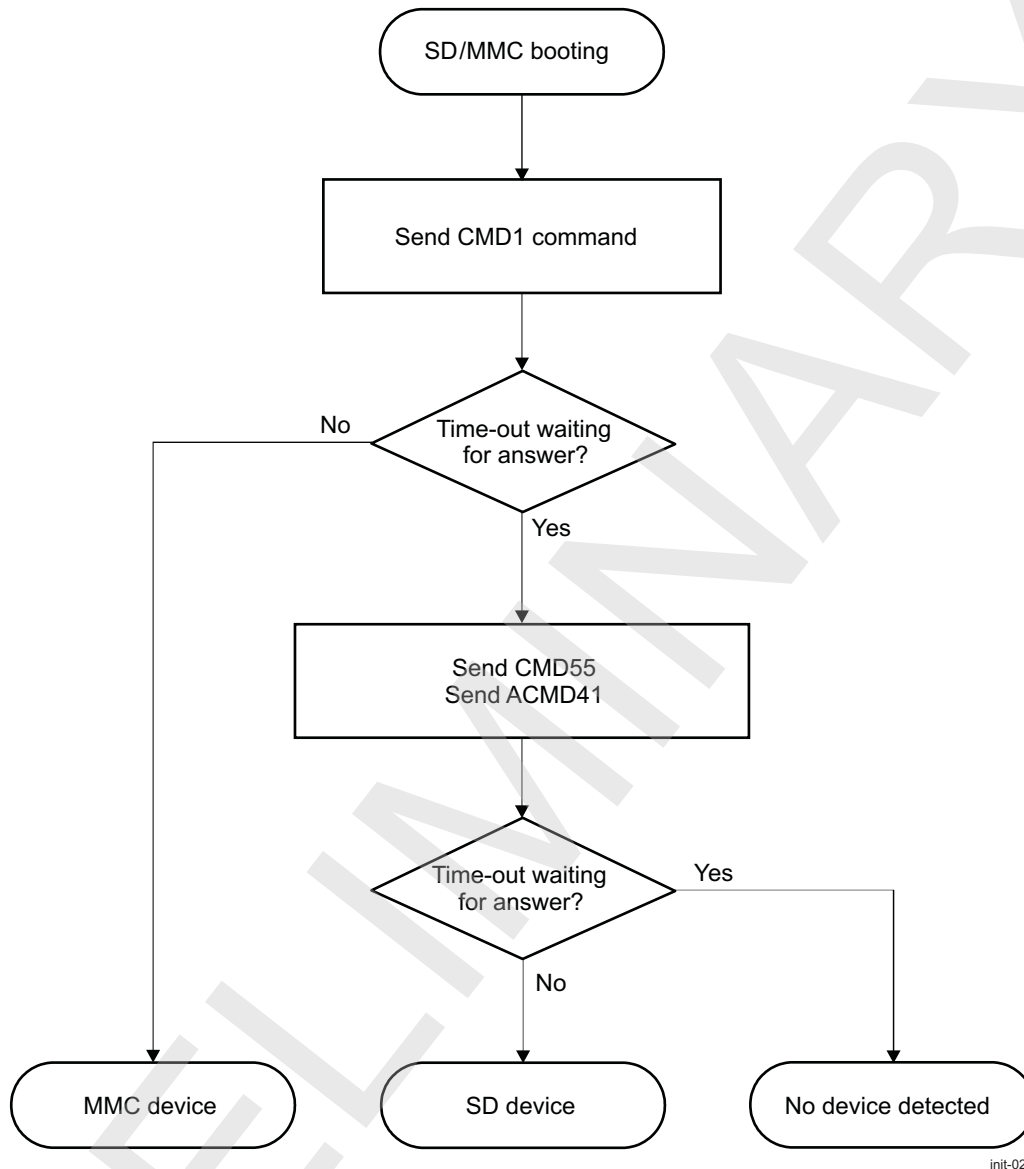
**NOTE:** The ROM code does not handle the card detection feature on the card cage.

#### 27.4.7.6.3 Initialization and Detection

The ROM code initializes the memory device or card connected on the interface using the standard high-voltage range (3.0 V). If neither a card or memory device is detected, the ROM code moves on to the next booting device. The standard identification process and relative card address (RCA) assignment are used. However, the ROM code assumes that only one memory or card is connected on the bus. This is done using the CMD line common to the SD and MMC memory devices. The MMC and SD standards describe this phase as the initialization phase. They differ in the first commands involved, CMD1 and ACMD41. The ROM code uses this command difference to discriminate between MMC and SD cards; that is, CMD1 is only supported by the MMC standard and ACMD41 is only supported by the SD standard. The ROM code first sends a CMD1 to the device and gets an answer only if an MMC device is connected. If no response is received, ACMD41 (a combination of CMD55 and ACMD41) is sent, and an response is expected from an SD device. If no response is received, no devices are connected and the ROM code exits MMC/SD booting with FAIL.

Figure 27-29 shows the MMC/SD detection procedure.

Figure 27-29. MMC/SD Detection Procedure



init-026

#### 27.4.7.6.4 Read Sector Procedure

- Raw mode

In raw mode, an image can be located at one of the four consecutive locations in the main area: offset 0x0 (0KB)/0x20000 (128KB)/0x40000 (256KB)/0x60000 (384KB). For this reason, the size of a booting image must not exceed 128KB. However, a device with an image greater than 128KB can be flash starting at one of the aforementioned locations. Therefore, the ROM code does not check the image size. The only drawback is that the image crosses the subsequent image boundary. Raw mode is detected by reading sectors 0, 256, 512, and 768. The content of these sectors is verified for the presence of a TOC structure. For a GP device, a CH (if used) must be located in the first sector followed by a GP header. The CH can be void (containing only a CHSETTINGS item for which the Valid field is 0), as described in [Section 27.4.8.2, Configuration Header](#). Image data is read directly from continuous sectors of a card. If raw mode is not detected, file system mode is assumed.

- File system handling

File system support is available only for SDMMC1 interface. The sector read procedure uses the standard MMC/SD read data procedure. The sector address is generated based on the booting

memory file map collected during initialization. Thus, the ROM code can freely address sectors in the booting file space.

#### 27.4.7.6.5 File System Handling

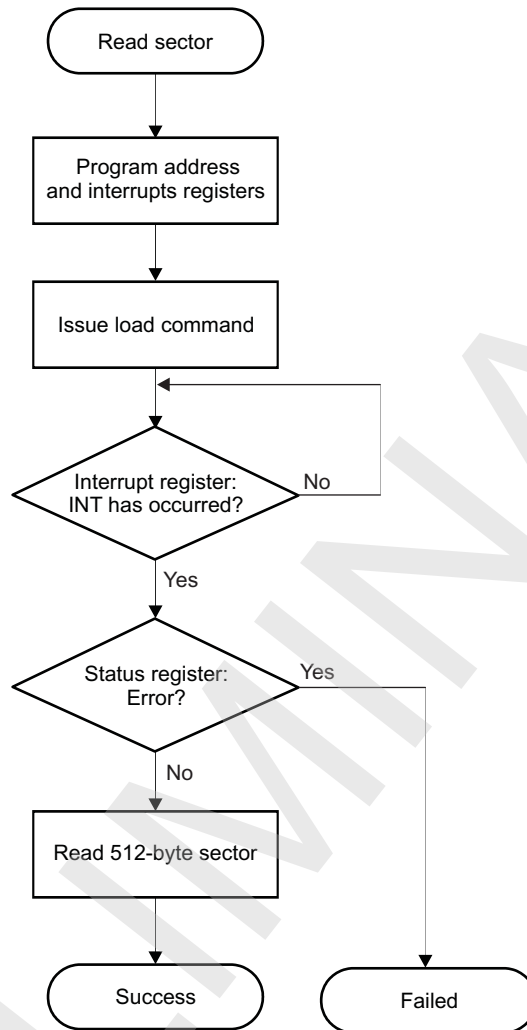
The MMC/SD cards can hold a file system that the ROM code reads. The image used by the booting procedure is taken from a booting file named MLO. This file must be in the root directory on an **active** primary partition of type FAT12/16 or FAT32.

An MMC/SD card can be configured as floppy-like or hard-drive-like:

- When acting like a floppy, the content of the card is a single FAT12/16/32 file system without an MBR holding a partition table.
- When acting like a hard drive, an MBR is present in the first sector of the card. This MBR holds a table of partitions, one of which must be FAT12/16/32, primary, and active.

According to the *MultiMediaCard FAT16 File System Specification* from the MMCA Technical Committee, the card must always hold an MBR, except for MMC cards using a floppy-like file system. However, depending on the operating system used, the MMC/SD card is formatted with or without partition(s) (using an MBR). The ROM code supports both types: floppy-like or hard-drive-like. The ROM code retrieves a map of the booting file from the FAT table. The booting file map is a collection of all FAT table entries related to the booting file (a FAT entry points to a cluster holding part of the file). The booting procedure uses this map to access any 512-byte sector in the booting file without involving the ROM code FAT module. [Figure 27-30](#) shows the complete process.

Figure 27-30. SD/MMC Get Booting File



init-025

### 27.4.7.6.5.1 MBR and FAT File System

This paragraph describes functions used by the ROM code to recognize whether an MBR with a FAT is used. It is not intended to fully describe the MBR and the FAT file system detection and reading procedure. The ROM code can detect FAT12/16/32 allocation table types. It cannot boot on devices with NTFS or Linux® FS partitions. Some memory devices that support file systems can be formatted with or without MBR; therefore, the first task of the ROM code is to detect whether the device is holding an MBR in the first sector.

The MBR is the first sector of a memory device. It consists of executable code, four partition entries, and one signature. The aim of such a structure is to divide the hard disk in partitions used primarily to boot different systems (for instance, Microsoft Windows™). This structure is described in [Table 27-39](#), and the partition table entry is described in [Table 27-40](#).

**Table 27-39. Master Boot Record Structure**

Offset	Length (Bytes)	Entry Description
0000h	446	Optional code
01BEh	16	Partition table entry
01CEh	16	Partition table entry
01DEh	16	Partition table entry
01EEh	16	Partition table entry
01FEh	2	Signature (0xAA55)

**Table 27-40. Partition Table Entry**

Offset	Length [Bytes]	Entry Description	Value
0000h	1	Partition state	00h: Inactive 80h: Active
0001h	1	Partition start head	Hs
0002h	2	Partition start cylinder and sector	Cs[7:0]-Cs[9:8]-Ss[5:0]
0004h	1	Partition type	01h: FAT12 04h, 06h, 0Eh: FAT16 0Bh, 0Ch, 0Fh: FAT32
0005h	16	Partition end head	He
0006h	2	Partition end cylinder and sector	Ce[7:0]-Ce[9:8]-Se[5:0]
0008h	4	First sector position relative to the beginning of media	LBAs = Cs.H.S+ Hs.S+ Ss-1
000Ch	4	Number of sectors in partition	LBAe = Ce.H.S+ He.S+ Se-1 Nb s= LBAe-LBAs + 1

SD/MMC booting consists of the following steps:

#### Step 1. MBR detection

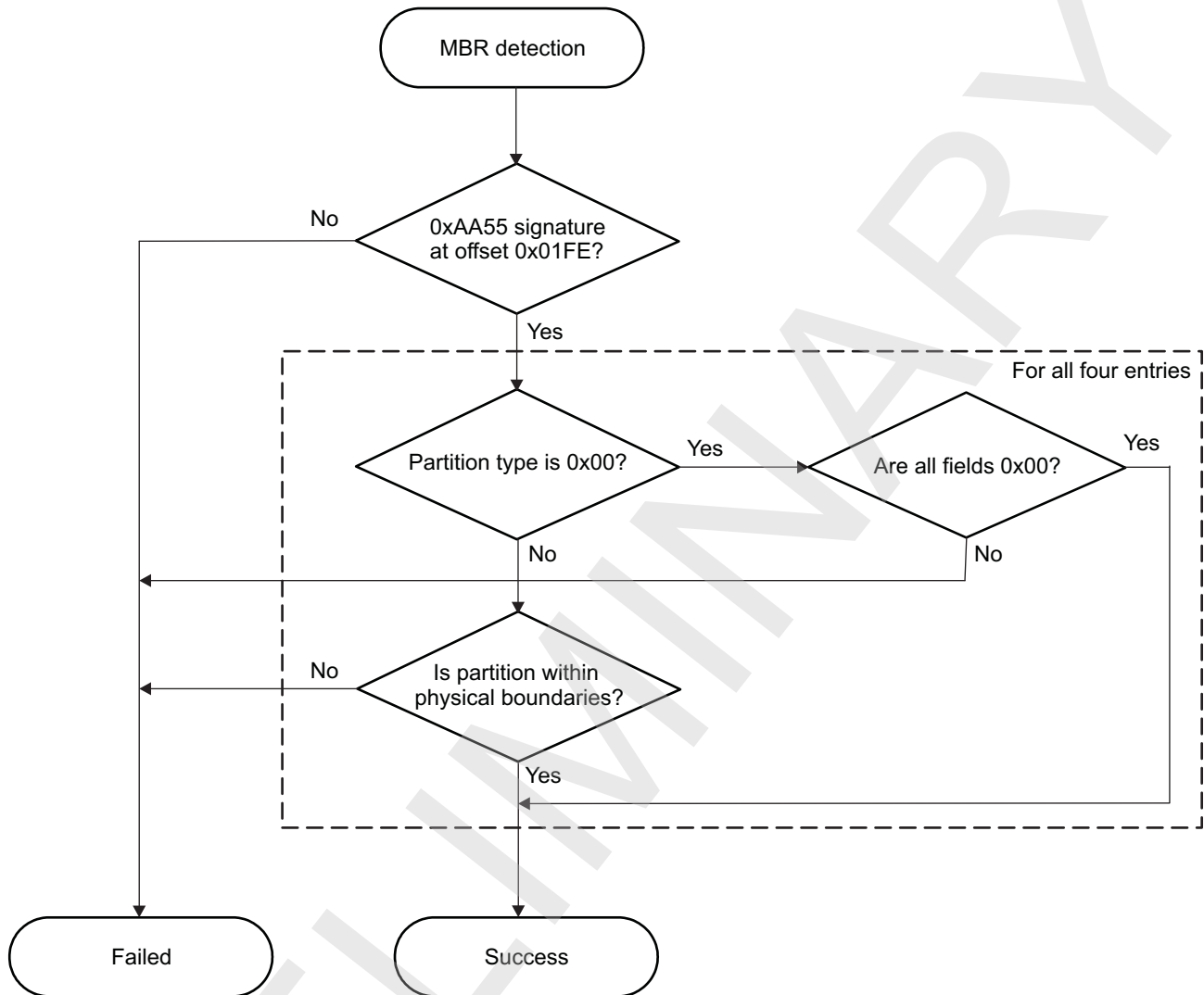
The ROM code first checks whether the MBR signature is present, and then it searches an active FAT12/16/32 partition in all four MBR partition entries, based on the type field. If the MBR entries are not valid, or if no usable partition is found, the ROM code returns to the booting procedure with FAIL. The extended partitions are not checked; the booting file must reside in a primary partition. Each partition entry is checked:

- (a) If its type is set to 00h, all fields in the entry must be 00h.
- (b) The partition is checked to be within physical boundaries (that is, the partition is inside and it fits the total physical sectors).

See [Figure 27-31](#) for more information about MBR detection.



Figure 27-31. MBR Detection Procedure



init-028

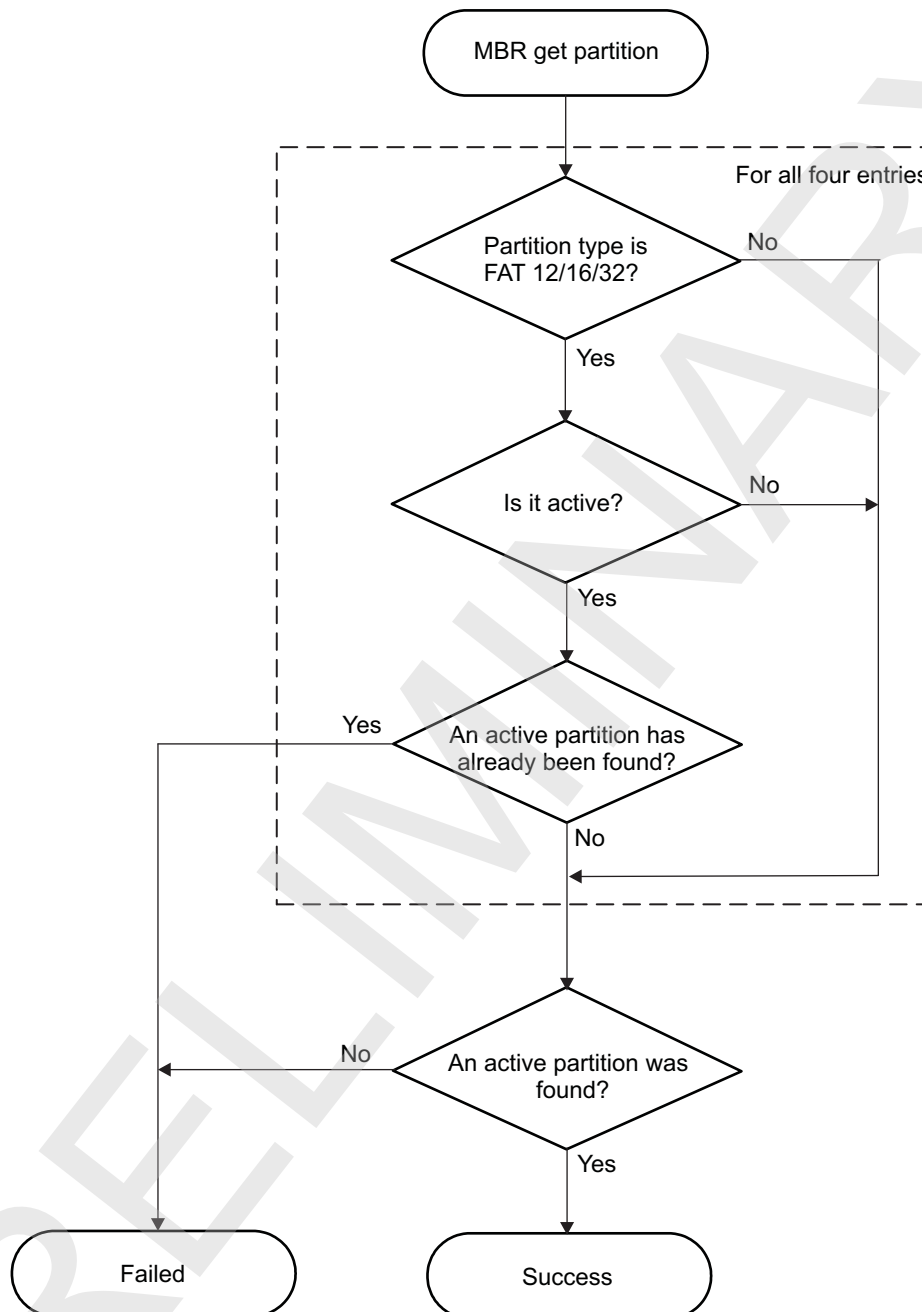
Step 2. Get the MBR partition.

Once identified, the ROM code gets the partition using the procedure described in Figure 27-31. The partition type is checked to be FAT12/16 or FAT32. Its state must be 00h (inactive) or 80h (active.) The ROM code returns with FAIL if no active primary FAT12/16/32 is found, or the test fails if there is more than one active partition. If an active partition is found, its first sector is read and used later. If no MBR is present (in case of a floppy-like system), the first sector of the device is read and used later. The read sector is checked to be a valid FAT12/16 or FAT32 partition. If this fails, the ROM code returns with FAIL if another partition type is used (for instance, Linux FS) or if the partition is not valid.

The FAT file system consists of several parts:

- Boot sector, which holds the BIOS parameter block (BPB). Not all are used by the ROM code.
- FAT, which describes the use of each cluster of the partition
- Data area, which holds the files, directories, and root directory (for FAT12/16, the root directory has a specific fixed location)

To check whether a sector holds a valid FAT12/16/32 partition, many fields of the boot sector (used by all FAT types) that must have specific values are checked. Figure 27-32 shows more about getting the partition.

**Figure 27-32. MBR, Get Partition**

init-029

**Step 3. Find the booting file.**

When a partition is found, the root directory entries are searched for a booting file named MLO in the root directory of the FAT12/16/32 file system. The file is not searched in any other location. For a FAT12/16 file system, the root directory has a fixed location, which is cluster 0. For a FAT32 file system, its cluster location is given by BPB\_RootClus. The formula to find the sector number (relative to device sector 0, not partition sector 0) of a cluster is given by the following equation:

$$\text{Cluster}_{\text{sector}} = \text{BPB\_HiddSec} + \text{BPB\_RsvdSecCnt} + \text{BPB\_NumFATs} \cdot \text{BPB\_FATSz} + \text{Cluster} \cdot \text{BPB\_SecPerCLus}$$

init-E001

**NOTE:** BPB\_FatSz is BPB\_FatSz16 for FAT12/16, or BPB\_FatSz32 for FAT32.

**NOTE:** The BPB\_HiddSec field can contain 0, even though the FAT file system is somewhere other than on sector 0 (floppy-like). The ROM code uses the partition offset taken from the MBR instead of this field, which can be wrong. If no MBR is found (floppy-like), the value 0 is used.

Each entry in the root directory is 32 bytes and holds information about the file (the filename, date of creation, rights, cluster location, etc.). (See [Table 27-41](#).)

**Table 27-41. FAT Directory Entry**

Offset	Length [Bytes]	Name	Description
0000h	11	DIR_Name	Short Name (8 + 3)
000Bh	1	DIR_Attr	File Attributes: 01h - ATTR_READ_ONLY 02h - ATTR_HIDDEN 04h - ATTR_SYSTEM 08h - ATTR_VOLUME_ID 10h - ATTR_DIRECTORY 20h - ATTR_ARCHIVE 0Fh - ATTR_LONG_NAME
000Ch	1	DIR_NTRes	Reserved. Set to 00h
000Dh	1	DIR_CrtTimeTenth	Millisecond stamp at file creation
000Eh	2	DIR_CrtTime	Time file was created.
0010h	2	DIR_CrtDate	Date file was created.
0012h	2	DIR_LstAccDate	Last access date
0014h	2	DIR_FstClusHi	High word of the first cluster number of this entry
0016h	2	DIR_WrtTime	Time of last write
0018h	2	DIR_WrtDate	Date of last write
001Ah	2	DIR_FstClusLo	Low word of the first cluster number of this entry
001Ch	4	DIR_FileSize	File size in bytes

The ROM code checks each entry in the root directory until either the booting file is found or the entry is empty (first byte is 00h), or when the end of the root directory is reached. Entries with the ATTR\_LONG\_NAME attribute (LFN) and first byte at E5h (erased file) are ignored. When found, the first cluster offset of the file is read from the DIR\_FstClusHi/DIR\_FstClusLo fields. There is a slight difference between FAT12/16 and FAT32 when handling the root directory. On FAT12/16, this directory has a fixed location (see above) and length fixed by BPB\_RootEntCnt, which is the total of 32-byte entries. Handling this directory is therefore straightforward. On FAT32, the root directory is like a standard file. The FAT must be used to retrieve each sector of the directory. The way in which the FAT is handled is described in the following paragraph.

#### Step 4. Buffer FAT entries in the FAT buffer

When the booting file is found, the ROM code reads the FAT and buffers the singly-linked chain of clusters in a FAT buffer used by booting to access the file directly, sector by sector. For FAT12/16 and for FAT32, multiple copies of the FAT exist (ROM code supports only two copies), after the boot sector.

$$FATn_{sector} = BPB\_HiddSec + BPB\_RsvdSecCnt + BP\_FatSz \cdot n$$

init-E002

The size of the FAT buffer is given by BPB\_FATsZ16 or BPB\_FATsZ32. The ROM code checks each copy of the FAT if they are identical. If the values are different, the ROM code uses the value from the last FAT copy. With the FAT32 file system, the copy system can be disabled according to a flag in BPB\_ExtFlags[7]. If this flag is set, the FAT BPB\_ExtFlags[3:0] bit field is used. In this case, no verification is made by the ROM code with other copies of FAT.

The FAT is a simple array of values, each referring to a cluster located in the data area. One entry of the array is 12, 16, or 32 bits, depending on the file system in use. The value in an entry defines whether the cluster is being used or not, and if another cluster must be considered. This creates a

singly-linked chain of clusters defining the file. The meaning of an entry is described in [Table 27-42](#).

**NOTE:** For compatibility, cluster 0 and cluster 1 are not used for files, and these entries must contain:

- FF8h and FFFh (for FAT12)
- FFF8h and FFFFh (for FAT16)
- 0FFFFFFF8h and 0FFFFFFFh (for FAT32)

**Table 27-42. FAT Entry Description**

FAT12	FAT16	FAT32	Description
000h	0000h	00000000h	Free cluster
001h	0001h	00000001h	Reserved cluster
002h-FEFh	0002h- FFEFh	00000002h- 0FFFFFFEFh	Used cluster; value points to next cluster
FF0h-FF6h	FFF0h- FFF6h	0FFFFFFF0h- 0FFFFFFF6h	Reserved values
FF7h	FFF7h	0FFFFFFF7h	Bad cluster
FF8h-FFFh	FFF8h- FFFh	0FFFFFFF8h- 0FFFFFFFh	Last cluster in file

**NOTE:** FAT32 uses only bits [27:0]; the upper 4 bits are usually 0 and must be left untouched.

When accessing the root directory for FAT32, the ROM code starts from the root directory cluster entry and follows the linked chain to retrieve the clusters.

When the booting file has been found, the ROM code buffers each FAT entry corresponding to the file in a sector way. This means each cluster is translated to one or several sectors, depending on how many sectors are in a cluster (BPB\_SecPerClus). This buffer is used later by the booting procedure to access the file.

#### 27.4.7.7 LPDDR2-NVM

The OMAP4430 device allows booting from a LPDDR2-NVM device connected to the EMIF interface. After setup, the NVM device can be accessed in XIP mode and instructions fetched directly through the EMIF interface (similarly as the XIP/NOR-like interfaces described in [Section 27.4.7.3, XIP Memory](#)).

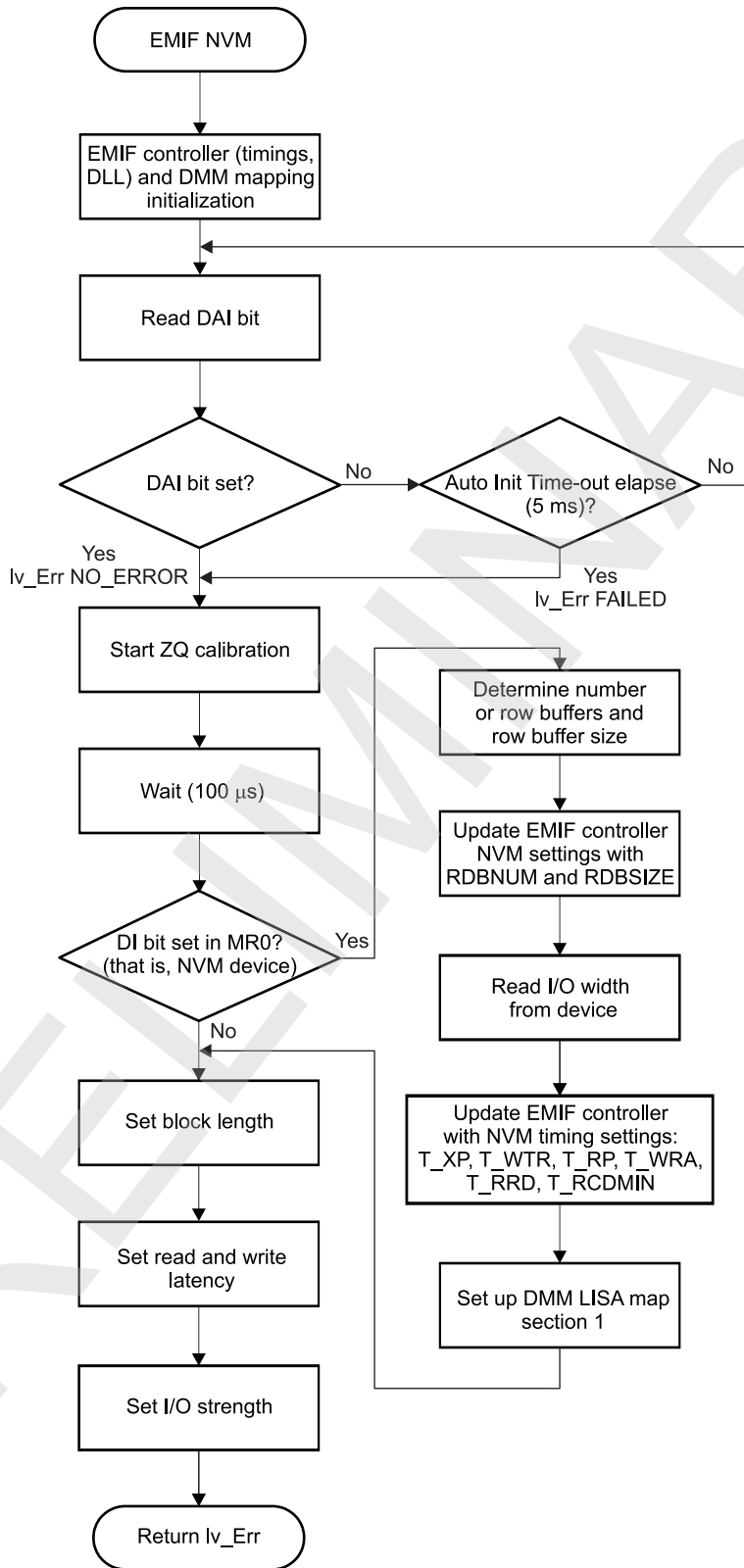
Features:

- XIP addressable
- Uses EMIF as the communication interface
- Up to 2-GB memories can be connected
- x32 data bus width only
- Supports 4/8 row data buffers of 32/64 bytes
- EMIF DDR clock frequency at boot time is 50 MHz
- Device connected to EMIF1/CS1, mapped at address C000000h

##### 27.4.7.7.1 Initialization and Detection

[Figure 27-25](#) shows the initialization and detection procedure.

Figure 27-33. EMIF LPDDR2-NVM Detection and Initialization



init-043

The ROM code first performs basic initialization of the EMIF controller and prepares for communication through EMIF1/CS1. The clock frequency at device boundary (ck pin) has been setup during the clocks setup (see [Section 27.4.4.2, Clocking Configuration](#)) to be 50 MHz (as defined as the upper boundary of the JEDEC specification for boot).

The device auto-initialization (DAI) status bit in Mode Register 0 (MR0) is polled for completion of device auto-initialization. This must complete within a 5-ms timeframe. If not, the initialization procedure returns with FAIL and the ROM code moves to the next device in the booting list. The MR0 device information (DI) bit reflects whether the attached device is an NVM device.

Then, the ROM code sends a ZQ calibration command (mode register write to MR10) and waits for completion (simple wait for 100  $\mu$ s). If the DI bit is set (it has been detected during the auto-initialization sequence), the ROM code determines the number and size of row data buffers and sets up the EMIF controller accordingly. Additionally, the NVM registers of the EMIF controller are updated with appropriate NVM timings.

The DMM (LISA map section 1) is set up to open a 2-GB window from system address C0000000h. This leaves the possibility for using a CHRAM configuration header to enable further SDRAM access.

The last steps consist in sending three configuration commands for:

- Block length setup, burst sequence, wrapping, and nWR (MR1)
- Read/write latency (MR2)
- I/O strength (MR3)

[Table 27-43](#) summarizes the list of mode registers used during NVM device configuration.

**Table 27-43. Summary of Mode Register Commands**

Mode Register (JEDEC Name)	Read/Write	Use
MR0 (Device_info)	MRR	Polling for auto-initialization and NVM device type determination
MR1 (Device_Feature1)	MRW	Block length setup (BL = 8), sequential burst (default), wrap enable (default), nWR = 3 (default)
MR2 (Device_Feature2)	MRW	Read/Write latency (RL = 3 / WL = 1, default)
MR3 (IO_Config1)	MRW	I/O strength (40 $\Omega$ , default)
MR8 (Basic_Config4)	MRR	I/O width determination (not used currently) <sup>(1)</sup>
MR10 (IO_Calibration)	MRW	ZQ calibration (FFh, calibration command after initialization)
MR20 (NVM_Geometry)	MRR	Determination of number and size of row data buffers

<sup>(1)</sup> The MR8 register is read for I/O width determination; however, no specific action is performed with this information. The device is assumed to be 32 bits wide.

## 27.4.8 Image Format

### 27.4.8.1 Overview

An image has two major parts:

- An optional CH
- Software to execute

The CH can contain several parameters set by users to speed up booting. It is further described in [Section 27.4.8.2, Configuration Header](#).

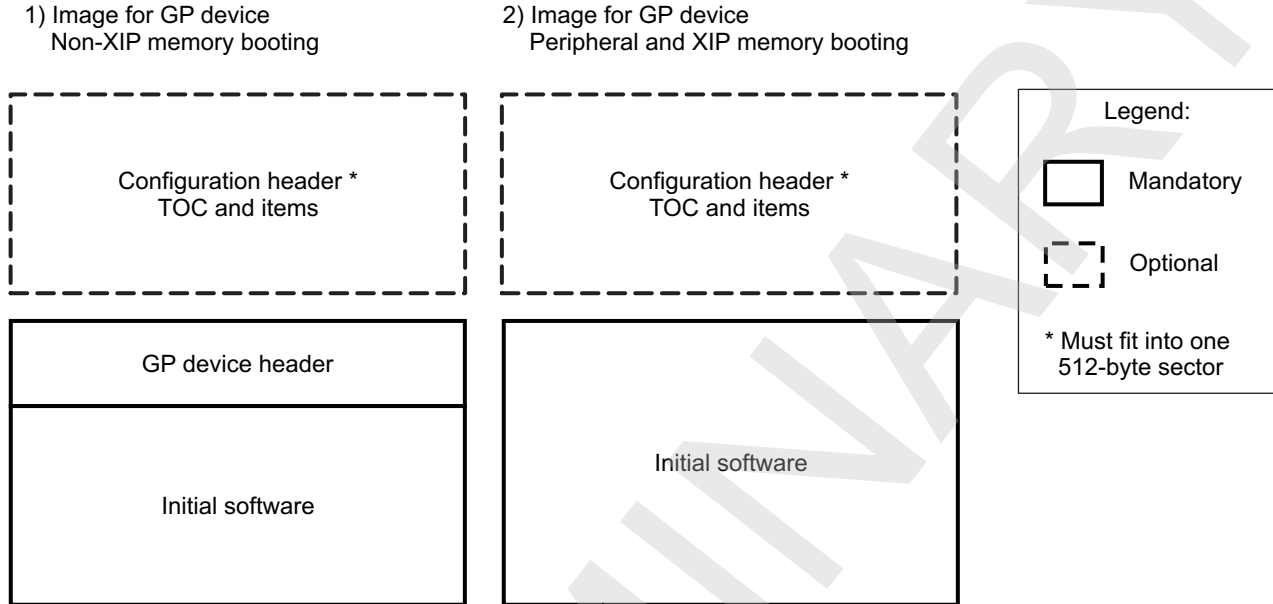
The mandatory part contains the software that is loaded into the memory and executed.

[Figure 27-34](#) is an overview of the boot image formats. There are two image types:

1. GP non-XIP memory booting: This image type is used for memories that require shadowing (for example, MMC). An image can contain a CH that includes at least a CHSETTINGS item entry. Next, there must be a small header (referred to as a GP header) that contains information about the size and the destination address.

- When the memory device is of XIP type (for example, NOR), the GP header is not required, and the image can contain code for direct execution. Optionally, the first sector can contain a CH. The same image format is used for peripheral booting (where the code is transferred to internal RAM).

**Figure 27-34. Image Formats**



init-018

### 27.4.8.2 Configuration Header

The CH is optional and is required only if the customer wants to use settings different from the ROM code defaults (for example, clock frequencies, EMIF, GPMC, or MMCHS interfaces).

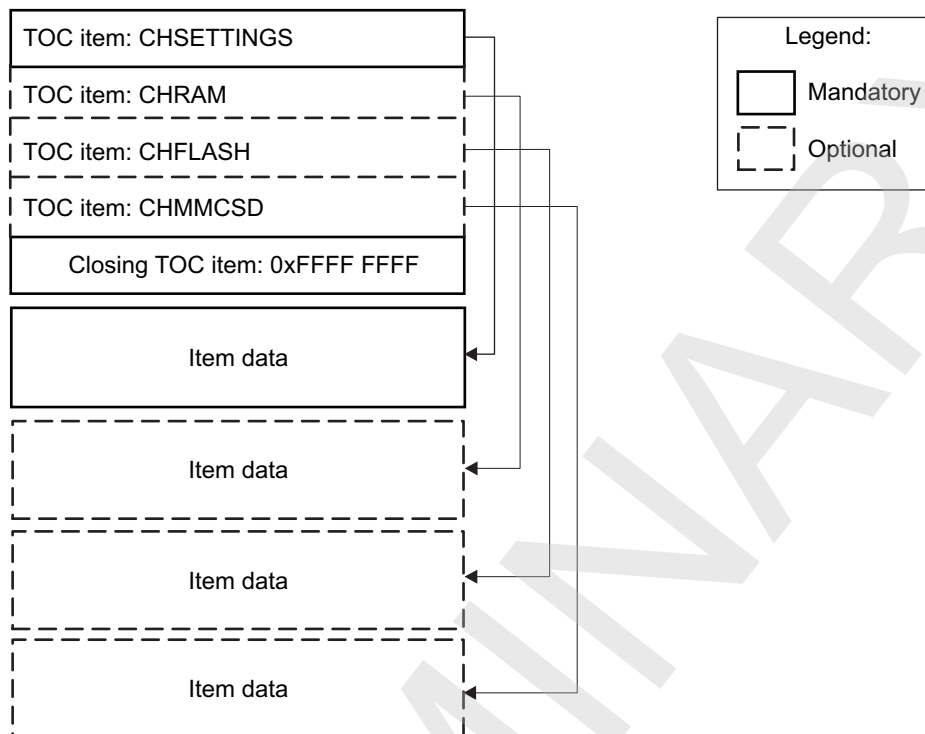
The CH can contain the following parts:

- Settings: Various clock settings (mandatory)
- RAM: EMIF settings
- FLASH: Flash interface (GPMC) settings
- MMC/SD: MMCHS interface settings

The beginning of the CH is a table of contents (TOC), which points to each item. This is described in [Figure 27-35](#). Each TOC item is a simple structure described in [Table 27-44](#). The complete CH (CH TOC and items) should fit in a 512-byte sector.

The ROM code identifies the presence of a CH by reading the first TOC item if it contains a known string (CHSETTINGS, CHRAM, etc.). Next, the TOC is identified and searched until a 0xFFFFFFFF offset is found. The CH is read and parameters are executed sequentially.

For the sake of simplicity, each field represents the content of a register to be modified. Only fields required for the configuration are used; fields for status, for instance, are not modified and therefore are not shown in the tables.

**Figure 27-35. CH Format**

init-019

**Table 27-44. CH TOC Item**

Offset	Field	Size (Bytes)	Description
0x0000	Start	4	Offset from the start address of the TOC to the actual address of item contents
0x0004	Size	4	Size of item
0x0008	Reserved	4	Unused
0x000C	Reserved	4	Unused
0x0010	Reserved	4	Unused
0x0014	Filename	12	12-character name of a item, including the zero (0) termination character

### 27.4.8.2.1 CHSETTINGS Item

The CHSETTINGS configuration header contains settings specific to the clock system. The ROM code configures the OMAP clocking to some default settings as described in [Section 27.4.4.2, Clocking Configuration](#). The CH CHSETTINGS section contains a method to override the ROM code default clock settings.

[Table 27-45](#) describes the fields. The clocking procedure and the clocking setting structure are described in [Section 27.4.4.4, Software Booting Configuration](#).

**Table 27-45. CHSETTINGS Item**

Offset	Field	Description
0000h	Section key	Key used for item verification: C0C0C0C1h
0004h	Valid	Enables/disables the section 00h: Disable Others: Enable



**Table 27-45. CHSETTINGS Item (continued)**

Offset	Field	Description
0005h	Version	Configuration header version 0001h Others: Reserved
0006h	Reserved	
0008h	Clocking settings	Described in <a href="#">Table 27-13</a> , <i>Software Booting Configuration Structure</i> , starting from the FLAGS field.

### 27.4.8.2.2 CHRAM Item

The CHRAM configuration header contains settings specific to the SDRAM memory controller (EMIF). [Table 27-46](#) describes the fields. The ROM code does not configure the EMIF by default, because it cannot assume any external SDRAM type. Therefore, if the ROM code is needed to configure the SDRAM access, the user must provide the appropriate timing details in the CHRAM structure. For more information, see [Section 15.3](#), *EMIF Controller*, and [Section 15.3.6](#), *EMIF Register Manual*.

**Table 27-46. CHRAM Item**

Offset	Field	Description
0000h	Section key	Key used for section verification: C0C0C0C2h.
0004h	Valid	Enables/disables the section: 00h: Disable Other: Enable
0005h	Reserved	
0008h	SdramConfig (EMIF1)	SDRAM configuration register
000Ch	SdramRefresh (EMIF1)	SDRAM Refresh control register
0010h	SdramTim1 (EMIF1)	SDRAM timing 1 register
0014h	SdramTim2 (EMIF1)	SDRAM timing 2 register
0018h	SdramTim3 (EMIF1)	SDRAM timing 3 register
001Ch	PwrMgtCtrl (EMIF1)	Power management control register
0020h	DdrPhyCtrl1 (EMIF1)	DDR PHY control 1 register
0024h	DdrPhyCtrl2 (EMIF1)	DDR PHY control 2 register
0028h	ModeReg1 (EMIF1)	Mode register MR1
0029h	ModeReg2 (EMIF1)	Mode register MR2
002Ah	ModeReg3 (EMIF1)	Mode register MR3
002Bh	Reserved	
002Ch	SdramConfig (EMIF2)	SDRAM configuration register
0030h	SdramRefresh (EMIF2)	SDRAM refresh control register
0034h	SdramTim1 (EMIF2)	SDRAM timing 1 register
0038h	SdramTim2 (EMIF2)	SDRAM timing 2 register
003Ch	SdramTim3 (EMIF2)	SDRAM timing 3 register
0040h	PwrMgtCtrl (EMIF2)	Power management control register
0044h	DdrPhyCtrl1 (EMIF2)	DDR PHY control 1 register
0048h	DdrPhyCtrl2 (EMIF2)	DDR PHY control 2 register
004Ch	ModeReg1 (EMIF2)	Mode register MR1
004Dh	ModeReg2 (EMIF2)	Mode register MR2
004Eh	ModeReg3 (EMIF2)	Mode register MR3
004Fh	Reserved	
0050h	DMM LISA Map 0	DMM LISA section 0 mapping (see <a href="#">Section 15.2</a> for more information about the DMM module architecture)

**Table 27-46. CHRAM Item (continued)**

Offset	Field	Description
0054h	Flags	Setting bits 0 to 3 requests ROM code to configure the appropriate channel/chip select: Bit[0]: Configure EMIF1 CS0 Bit[1]: Configure EMIF1 CS1 Bit[2]: Configure EMIF2 CS0 Bit[3]: Configure EMIF2 CS1 Other bits: Reserved

**NOTE:**

- The ROM code supports only the LPDDR2 memory type (as set in the SDRAM configuration register).
- It is assumed that the same kind of SDRAM (same type and timings) is attached to both CSs when requesting the configuration of both CSs for a given EMIF (for example, if bit 0 and 1 of the flags field are set, it is assumed that both EMIF1 CSs are attached to the same kind of memory).

**27.4.8.2.3 CHFLASH Item**

The CHFLASH configuration header contains settings specific to the general-purpose memory controller (GPMC). For more information, see [Section 15.4, GPMC Overview](#). [Table 27-47](#) describes the fields.

**Table 27-47. CHFLASH Item**

Offset	Field	Description
0000h	Section Key	Key used for section verification: C0C0C0C3h.
0004h	Valid	Enables/disables the section: 00h: Disable Others: Enable
0005h	Reserved	
0008h	GPMC_SYSCONFIG (LSB)	Register values
000Ah	GPMC_IRQENABLE (LSB)	
000Ch	GPMC_TIMEOUT_CONTROL (LSB)	
000Eh	GPMC_CONFIG (LSB)	
0010h	GPMC_CONFIG1_0	
0014h	GPMC_CONFIG2_0	
0018h	GPMC_CONFIG3_0	
001Ch	GPMC_CONFIG4_0	
0020h	GPMC_CONFIG5_0	
0024h	GPMC_CONFIG6_0	
0028h	GPMC_CONFIG7_0	
002Ch	GPMC_PREFETCH_CONFIG1	
0030h	GPMC_PREFETCH_CONFIG2 (LSB)	
0032h	GPMC_PREFETCH_CONTROL (LSB)	
0034h	GPMC_ECC_CONFIG	
0036h	GPMC_ECC_CONTROL	
0038h	GPMC_ECC_SIZE_CONFIG (LSB)	
003Ch	Enable_A1_A10	0: Do not change A1– A10 pads Others: Enable A1 – A10 pads

#### 27.4.8.2.4 CHMMCS D Item

The CHMMCS D configuration header contains settings specific to the high-speed MMC/SD/SDIO host controller (MMCHS). For more information, see [Chapter 24, MMC/SD/SDIO](#). [Table 27-48](#) describes the fields.

**Table 27-48. CHMMCS D Item**

Offset	Register Modified	Description
0000h	Section key	Key used for section verification C0C0C0C4h
0004h	Valid	Enables/disables the section 00h: Disable Other: Enable
0005h	Reserved	
0008h	MMCHS_SYSCTRL(MSW)	Update MMCHS_SYSCTRL interface register with the value specified in these fields. The register is not updated if the value is 0xFFFFFFFF.
000Ah	MMCHS_SYSCTRL(LSW)	
000Ch	MMCHS interface bus width	1: 1 bit 2: 4 bits 4: 8 bits 0xFFFF FFFF: Do not update bus width. Others: Reserved

The ROM code provides a booting parameter structure to the Initial software (see [Section 27.4.8.4, Image Execution](#)). This structure contains a field that indicates whether the configuration header items have been correctly processed. For a CHMMCS D item, if the MMCHS\_SYSCTRL and bus width fields are set to 0xFFFF FFFF, the booting parameters report that the CHMMCS D section has not been executed, regardless of the value of the Valid field.

#### 27.4.8.3 GP Header

When the booting memory device is non-XIP (for example, MMC) the image must contain a small header, located before the executable code, and having the size of the software to load and the destination address of where to store it. [Table 27-49](#) describes the image format. The GP header is not required when booting from an XIP memory device (for example, NOR) or in case of peripheral booting. In this case, the peripheral or memory booting image starts directly with executable code.

**Table 27-49. GP Header Format**

Field	Non-XIP Device Offset	XIP Device Offset	Size (Bytes)	Description
Size	0x0000	-	4	Size of the image
Destination	0x0004	-	4	Address where to store the image or code entry point
Image	0x0008	0x0000	x	Executable code

**NOTE:** The destination address field stands for:

- Target address for the image copy from the non-XIP storage to the target XIP location (for example, internal RAM or SDRAM)
- Entry point for image code

Users must take care to locate the code entry point to the target address for image copy.

#### 27.4.8.4 Image Execution

The image is executed at the time the ROM code performs the branch to the first executable instruction in the initial software. For a GP device in non-XIP, the execution address is the first word after the GP header. The branch is performed in public ARM supervisor mode. The R0 register points to the booting parameter structure that contains information about booting execution. [Table 27-50](#) shows the booting parameter structure.

**Table 27-50. Booting Parameter Structure**

Offset	Field	Size (Bytes)	Description
0x00	Booting message	4	Last received booting message
0x04	Memory booting device descriptor	4	Pointer to the memory device descriptor that has been used during the memory booting process
0x08	Current booting device	1	Code of device used for booting: 0x00: void, no device 0x02: XIP memory 0x02: XIPWAIT memory (wait-signal monitoring on) 0x03: NAND 0x04: OneNAND 0x05: MMC1 0x06: MMC2(1) 0x07: MMC2(2) 0x08: EMIF 0x43: UART3 0x45: USB (internal transceiver) 0x46: USB-ULPI Others: Reserved
0x09	Reset reason	1	Current reset reason bit mask (bit = 1, event present): [0]: Power-on reset [1]: Global software reset [3]: Violation reset [4]: MPU watchdog reset [5]: Watchdog reset [6]: External warm reset Other bits: Reserved
0x0A	CH flags	1	Configuration header items flag. Each item is described by 1 bit. A set bit indicates that the item was executed: [0]: CHSETTINGS [1]: CHRAM [2]: CHFLASH [3]: CHMMCS Other bits: Reserved

#### 27.4.9 Tracing

Tracing in the public ROM code consists in three 32-bit vectors for which each bit corresponds to a particular way point in the ROM code execution sequence. [Table 27-51](#) through [Table 27-53](#) list the organization of the tracing data in RAM. Tracing vectors are initialized at the beginning of the start-up phase and are updated all along the boot process.

There are two sets of tracing vectors ([Table 27-10](#)). The first set is the current trace information (after cold or warm reset). The second set holds a copy of trace vectors collected at the first ROM code run after cold reset. As a result, after a warm reset it is possible to have visibility on the boot scenario that occurred during cold reset.

[Table 27-51](#) lists the organization of tracing vector 1.

**Table 27-51. Tracing Vector 1**

Bit	Group	Meaning
0	General	Passed the public reset vector
1	General	Entered main function
2	General	Running after the cold reset
3	Boot	Main booting routine entered
4	Memory boot	Memory booting started
5	Peripheral boot	Peripheral booting started
6	Boot	Booting loop reached last device
7	Boot	GP header found
8	Boot	Booting message Skip peripheral booting received
9	Boot	Booting message Change device received
10	Peripheral boot	Booting message Peripheral booting received
11	Peripheral boot	Booting message Get ASIC ID received
12	Peripheral boot	Device initialized
13	Peripheral boot	ASIC ID sent
14	Peripheral boot	Image received
15	Peripheral boot	Peripheral booting failed
16	Peripheral boot	Booting message not received (time-out)
17	Peripheral boot	Image size not received (time-out)
18	Peripheral boot	Image not received (time-out)
19	Reserved	
20	Configuration header	CHSETTINGS found
21	Configuration header	CHSETTINGS executed
22	Configuration header	CHRAM executed
23	Configuration header	CHFLASH executed
24	Configuration header	CHMMCSDBUS clocks executed
25	Configuration header	CHMMCSDBUS bus width executed
26	Reserved	
27	Reserved	
28	Software booting configuration	SWCFG general detected
29	Software booting configuration	SWCFG clocks detected
30	Software booting configuration	SWCFG timeout detected
31	Reserved	

[Table 27-52](#) lists the organization of tracing vector 2.

**Table 27-52. Tracing Vector 2**

Bit	Group	Meaning
0	Power-management IC	TWL6030 detected
1	Power-management IC	VBUS detected
2	Power-management IC	VMMC switched on
3	Power-management IC	VUSB switched on

**Table 27-52. Tracing Vector 2 (continued)**

Bit	Group	Meaning
4	USB	USB connect
5	USB	USB configured state
6	USB	USB VBUS valid
7	USB	USB session valid
8	USB	PrevStartOk set
9	USB	Suspend event
10	USB	Resume event
11	USB	Charger detected
12	Memory boot	Memory booting trial 0
13	Memory boot	Memory booting trial 1
14	Memory boot	Memory booting trial 2
15	Memory boot	Memory booting trial 3
16	Memory and peripheral boot	Execute GP image
17	Reserved	
18	Reserved	
19	Reserved	
20	Reserved	
21	Reserved	
22	Reserved	
23	Reserved	
24	Reserved	
25	Reserved	
26	Reserved	
27	Reserved	
28	Reserved	
29	Reserved	
30	Reserved	
31	MMCSD	MMCSD1 PBIAS configuration is 1.8 V.

Table 27-53 lists the organization of tracing vector 3.

**Table 27-53. Tracing Vector 3**

Bit	Group	Meaning
0	Memory boot	Memory booting device NULL
1	Memory boot	Memory booting device XIP
2	Memory boot	Memory booting device XIPWAIT
3	Memory boot	Memory booting device NAND
4	Memory boot	Memory booting device OneNAND
5	Memory boot	Memory booting device MMC1
6	Reserved	
7	Memory boot	Memory booting device MMC2(1)
8	Reserved	
9	Reserved	
10	Memory boot	Memory booting device EMIF
11	Memory boot	Memory booting device MMC2(2)
12	Reserved	

**Table 27-53. Tracing Vector 3 (continued)**

<b>Bit</b>	<b>Group</b>	<b>Meaning</b>
13	Reserved	
14	Reserved	
15	Reserved	
16	Reserved	
17	Reserved	
18	Peripheral boot	Peripheral booting device UART3
19	Reserved	
20	Peripheral boot	Peripheral booting device USB
21	Peripheral boot	Peripheral booting device USB ULPI
22	Peripheral boot	Peripheral booting device NULL
23	Reserved	
24	Reserved	
25	Reserved	
26	Reserved	
27	Reserved	
28	Reserved	
29	Reserved	
30	Reserved	
31	Reserved	

## 27.5 Services for HLOS Support

The ROM code provides different services that can be called on GP devices for L2 cache maintenance, wake up of slave CPU(s), etc. These services are implemented in monitor mode (the service must be called by writing the function ID into the R12 register and using the SMC instruction) and do not use any resources like RAM/stack or hardware outside the master CPU. The caller must ensure the save and restore of the processor registers before and after calling the monitor service.

The following code example shows how the monitor ROM code functions can be accessed by an application running in public mode:

```

;-----;
FUNCTION: PL310_SetDebugL2Cache
;
; DESCRIPTION: Function calls the Monitor Service to configure the debug
;              control register of the L2 cache.
;
; INPUTS: r0 Value to set in the debug control register
;
; RETURN:
;
;-----;
PL310_SetDebugL2Cache FUNCTION
    PUSH {R1-R12, LR}
    LDR R12, =API_MONITOR_L2CACHE_SETDEBUG_INDEX
    SMI 0x1 POP {R1-R12, PC}
    ENDFUNC

```

### 27.5.1 L2 Cache Maintenance

**Table 27-54. L2 Cache Write Debug Register**

Function ID	Description	
R12 = 0x100	Set the input parameter value in the L2 Cache Debug register	
Parameters		
Type	Location	Description
Input	R0	Debug register value to set
Output		
Return		

**Table 27-55. L2 Cache Clean and Invalidate Range of Physical Address**

Function ID	Description	
R12 = 0x101	This function cleans and invalidates a range Physical Addresses. This is a workaround of the clean and invalidate L2 cache.	
Parameters		
Type	Location	Description
Input	R0 R1	Physical Start Address of the range Size of the range to invalidate
Output		
Return		

**Table 27-56. Enable/Disable L2 Cache**

Function ID	Description	
R12 = 0x102	This function sets the given input value in the L2 Cache Control Register (enable/disable)	
Parameters		
Type	Location	Description
Input	R0	Control register value to set



**Table 27-56. Enable/Disable L2 Cache (continued)**

Function ID	Description
Output	
Return	

**Table 27-57. Write Auxiliary Control Register**

Function ID	Description	
R12 = 0x109	This function writes the PL310 Auxiliary Control Register with the input given value in R0.	
Parameters		
Type	Location	Description
Input	R0	Value to set in the register
Output		
Return		

**Table 27-58. Write Tag and Data RAM Latency Control Register**

Function ID	Description	
R12 = 0x112	This function writes the PL310 Tag and Data RAM Latency Control Register with the input given value in R0 and R1.	
Parameters		
Type	Location	Description
Input	R0 R1	Tag RAM latency to set Data RAM latency to set
Output		
Return		

**27.5.2 Multicore Infrastructure Maintenance**

**Table 27-59. Read AuxCoreBoot 0 and AuxCoreBoot 1 Register**

Function ID	Description	
R12 = 0x103	Read both AuxCoreBoot registers. This function is not necessary. It is kept for compatibility with ES1.0.	
Parameters		
Type	Location	Description
Input		
Output	R0 R1	AuxCoreBoot 0 register value AuxCoreBoot 1 register value
Return		

**Table 27-60. Modify AuxCoreBoot 0**

Function ID	Description	
R12 = 0x104	Modify the AuxCoreBoot 0 register used as a status bit. The function sets and clears the bit given in the parameter and returns the new register value after operation. This function is not necessary. It is kept for compatibility with ES1.0.	
Parameters		
Type	Location	Description
Input	R0 R1	Bit mask to set Bit mask to clear
Output		

**Table 27-60. Modify AuxCoreBoot 0 (continued)**

Function ID	Description	
Return	R0	AuxCoreBoot 0 register new value

**Table 27-61. Write AuxCoreBoot 1**

Function ID	Description	
R12 = 0x105	Set the value of the AuxCoreBoot 1 register value used to give the CPU1 boot address. The given address can be virtual or physical. CPU 1 MMU must be enabled before jumping to this address. This function is not necessary. It is kept for compatibility with ES1.0.	
Parameters		
Type	Location	Description
Input	R0	Value to set in the register
Output		
Return		

**Table 27-62. Read Control and Status Register**

Function ID	Description	
R12 = 0x106	Function returns the value of the WakeUpGen Control and Status register CPU 0 or CPU 1 depending on the input parameter value. This function is not necessary. It is kept for compatibility with ES1.0.	
Parameters		
Type	Location	Description
Input	R0	CPU ID
Output		
Return	R0	Value of the control and status register of the CPU ID

**Table 27-63. Clear Control and Status Register**

Function ID	Description	
R12 = 0x107	This function clears the given bit mask of the WakeUpGen Control and Status register of the given CPU ID.	
Parameters		
Type	Location	Description
Input	R0 R1	CPU ID Bit Mask to clear
Output		
Return		

### 27.5.3 Snoop Control Unit (SCU)

**Table 27-64. Set Power Status Register**

Function ID	Description	
R12 = 0x108	This function sets the power status register of the CPU and cleans the full CPU L1 data cache, if the cache will be switched OFF when CPU enters into low-power state.	
Parameters		
Type	Location	Description
Input	R0 R1	CPU power state value to set (0x2: CSWRET, 0x3: OFF) L1 data cache state (0x00: RET, 0xFF: OFF)

**Table 27-64. Set Power Status Register (continued)**

Function ID	Description
Output	
Return	

### 27.5.4 Lockdown Translation Lookaside Buffers (TLBs)

**Table 27-65. Select the TLB Entry for Read**

Function ID	Description	
R12 = 0x10A	This function writes the Select Lockdown TLB Entry for Read register.	
<b>Parameters</b>		
Type	Location	Description
Input	R0	TLB entry
Output		
Return		

**Table 27-66. Select the TLB Entry for Write**

Function ID	Description	
R12 = 0x10B	This function writes the Select Lockdown TLB Entry for Write register.	
<b>Parameters</b>		
Type	Location	Description
Input	R0	TLB entry
Output		
Return		

**Table 27-67. Read the TLB VA Entry**

Function ID	Description	
R12 = 0x10C	This function reads the Lockdown TLB VA register (Selected TLB Entry for Read).	
<b>Parameters</b>		
Type	Location	Description
Input		
Output		
Return	R0	Value of the register

**Table 27-68. Write the TLB VA Entry**

Function ID	Description	
R12 = 0x10D	This function writes the Lockdown TLB VA register (Selected TLB Entry for Write). The NS bit is forced to 1.	
<b>Parameters</b>		
Type	Location	Description
Input	R0	Value to write
Output		
Return		

**Table 27-69. Read the TLB PA Entry**

Function ID	Description	
R12 = 0x10E	This function reads the Lockdown TLB PA register (Selected TLB Entry for Read).	
Parameters		
Type	Location	Description
Input		
Output		
Return	R0	Value of the register

**Table 27-70. Write the TLB PA Entry**

Function ID	Description	
R12 = 0x10F	This function writes the Lockdown TLB PA register (Selected TLB Entry for Write).	
Parameters		
Type	Location	Description
Input	R0	Value to write
Output		
Return		

**Table 27-71. Read the TLB Attributes Entry**

Function ID	Description	
R12 = 0x110	This function reads the Lockdown TLB Attributes register (Selected TLB Entry for Read).	
Parameters		
Type	Location	Description
Input		
Output		
Return	R0	Value of the register

**Table 27-72. Write the TLB Attributes Entry**

Function ID	Description	
R12 = 0x111	This function writes the Lockdown TLB Attributes register (Selected TLB Entry for Write). The NS bit is forced to 1.	
Parameters		
Type	Location	Description
Input	R0	Value to write
Output		
Return		

## On-Chip Debug Support

This chapter describes the on-chip debug support.

**NOTE:** The L3 interconnect is instantiation of the NoC interconnect from Arteris, Inc. Arteris is a registered trademark of Arteris, Inc.

This document contains materials that are ©Arteris, Inc., and that constitute proprietary information of Arteris, Inc.

All materials and trademarks are used under license from Arteris, Inc. For additional information, see the Arteris Reference manuals, or contact Arteris, Inc.

NoC is an abbreviation for Network On Chip.

Topic	Page
<b>28.1 Introduction</b> .....	<b>5270</b>
<b>28.2 Debug Ports</b> .....	<b>5271</b>
<b>28.3 Debugger Connection</b> .....	<b>5275</b>
<b>28.4 Primary Debug Support</b> .....	<b>5278</b>
<b>28.5 Power, Reset, and Clock Management Debug Support</b> .....	<b>5283</b>
<b>28.6 Performance Monitoring</b> .....	<b>5284</b>
<b>28.7 Processor Trace</b> .....	<b>5288</b>
<b>28.8 System Instrumentation</b> .....	<b>5289</b>
<b>28.9 Concurrent Debug Modes</b> .....	<b>5297</b>
<b>28.10 Memory Mapping</b> .....	<b>5299</b>

## 28.1 Introduction

Debugging a system containing an embedded processor involves an environment that connects high-level debugging software, executing on a host computer, to a low-level debug interface supported by the target device. In between these levels is a debug and trace controller (DTC) that facilitates communication between the host debugger and the debug support logic on the target chip.

A combination of hardware and software that connects the host debugger to the target system, the DTC uses one or more hardware interfaces and/or protocols to convert actions dictated by the debugger user to JTAG commands and scans that execute the core hardware.

The debug software and hardware components let the user control multiple central processing unit (CPU) cores embedded in the device in a global or local manner. This environment provides:

- Synchronized global starting and stopping of multiple processors
- Starting and stopping of an individual processor
- Each processor can generate triggers that can be used to alter the execution flow of other processors.

System topics include but are not limited to:

- System clocking and power-down issues
- Interconnection of multiple devices
- Trigger channels

For easy integration into applications, a set of libraries (APIs) for debug-IP programming and software message library are being provided. More information about the APIs, download files, and other useful links for available libraries can be found on the CToolsLib Wiki site:

<http://processors.wiki.ti.com/index.php/CToolsLib>

CToolsLib is a collection of embedded target APIs/library focused to enable easy access to the chip tools (CTools), which are system-level debug facilities included in the debug subsystem capabilities of TI devices.

## 28.2 Debug Ports

### 28.2.1 IEEE1149.1

The target debug interface has the following signals:

- Five standard IEEE1149.1 JTAG signals: nTRST, TCK, TMS, TDI, and TDO
- A return clock (RTCK) due to the clocking requirements of the ARM968™ processor
- Two EMU [1:0] or five EMU [4:0] TI extensions, depending on the pin count (14 pins or 20 pins) in the JTAG header of the device.

Table 28-1 describes the IEEE1149.1 signals.

**Table 28-1. IEEE1149.1 Signals**

Pin Name	Internal Signal Name	Type <sup>(1)</sup>	Function	Description
jtag_nrst	nTRST	I	Test reset	When asserted (active low), causes all test and debug logic in the device to be reset along with the IEEE1149.1 interface
jtag_tck	TCK	I	Test clock	This is the test clock used to drive an IEEE1149.1 test access port (TAP) state-machine and logic. This is a free-running clock or a gated clock, depending on the DTC attached to the device and the RTCK monitoring.
jtag_rtck	RTCK	O	Returned (synchronized) test clock	Depending on the DTC attached to the device, either the JTAG signals are clocked from RTCK or the RTCK is monitored by the DTC to the gate TCK.
jtag_tms_tmisc	TMS	I	Test mode select	Directs the next state of the IEEE1149.1 TAP state-machine
jtag_tdi	TDI	I	Test data input	Scans data input to the device
jtag_tdo	TDO	O	Test data output	Scans data output by the device
dpm_emu0	EMU0	IO	Emulation 0	Channel 0 trigger OR boot mode OR trace port
dpm_emu1	EMU1	IO	Emulation 1	Channel 1 trigger OR boot mode OR trace port
dpm_emu2 <sup>(2)</sup>	EMU2 <sup>(2)</sup>	O	Emulation 2	Trace port
dpm_emu3 <sup>(2)</sup>	EMU3 <sup>(2)</sup>	O	Emulation 3	Trace port
dpm_emu4 <sup>(2)</sup>	EMU4 <sup>(2)</sup>	O	Emulation 4	Trace port

(1) I = Input; O = Output

(2) 20-pin JTAG header only

Table 28-2 specifies the OMAP4430 JTAG ID code, which can be accessed through the ICEPick module embedded in the debug subsystem.

**Table 28-2. OMAP4430 JTAG ID Code**

Device	F#	JTAG ID Code
OMAP4430	F78182	0xB852

### 28.2.2 IEEE1149.7

Besides the standard (legacy) JTAG mode of operation, the target debug interface can also be switched to a compressed JTAG (cJTAG) mode of operation, commonly referred to as IEEE1149.7 standard. An IEEE1149.7 adapter module runs a 2-pin communication protocol on top of an IEEE1149.1 JTAG TAP. The debug-IP logic serializes the IEEE1149.1 transactions, using a variety of compression formats, to reduce the number of pins needed to implement a JTAG debug port. The OMAP4430 device implements only a subset of the IEEE1149.7 protocol (see Table 28-4).

**NOTE:** At power-on-reset (POR), the default debug interface mode is IEEE1149.1. See the IEEE1149.7 specification for the scan sequence required to switch modes.

The IEEE1149.7 communication protocol can switch between serial and parallel formats. The parallel

format consists of the IEEE1149.1 signals, TCK, TMS, TDI, TDO, nTRST, and a RTCK return clock, if stalls are required. The serial format uses two signals, TCK and TMSC. The values of the TMS, TDI, TDO, and RTCK signals are multiplexed on the bidirectional pin, jtag\_tms\_tmsc, by the Debug-IP logic. Each TCK of the DTC is serialized into a packet, which is transmitted to the IEEE1149.7 adapter, where it is converted back to parallel. The TDO and RTCK information are returned during the packet transmission.

**NOTE:** Only MScan, Oscan0, and Oscan4 formats can be used if the device requires stalls.

Table 28-3 describes the IEEE1149.7 signals.

**Table 28-3. IEEE1149.7 Signals**

Pin Name	Internal Signal Name	Type <sup>(1)</sup>	Function	Description
jtag_tck	TCK	I	Test clock	This is the test clock used to drive an IEEE1149.1 TAP state-machine and logic. This is a free-running clock or a gated clock, depending on the DTC attached to the device and the TMSC RDY field monitoring.
jtag_tms_tmsc	TMSC	I/O	Test mode control and data scan	Compressed JTAG packet. The TMSC signal is driven only when the TCK signal is low. The signal level must be maintained by a keeper while the TCK signal is high.

(1) I = Input; O = Output

Table 28-4 summarizes the IEEE1149.7 features subset supported by the OMAP4430 device.

**Table 28-4. IEEE1149.7 Features Subset**

	IEEE1149.7 Feature	Device Support	Comment
<b>Configuration</b>	Class 4 TAP	✓	Supports 2-pin operation
	Class 5 TAP	–	No BDx or CDx channel
<b>Optional Components</b>	FRST	–	Functional reset (done via ICEPick)
	TRST	–	Test reset
	RDBK capability	✓	Read back of register data
	Aux Pin Functions	–	Reuse of TDI and TDO pins
	TCKWID	✓	Programmable TCK width
<b>Scan Formats</b>	JScan0	✓	Parallel mode
	JScan1	✓	Parallel with firewall
	JScan2	✓	Parallel with super-bypass select
	JScan3	✓	Parallel with register select
	SScan0	–	Segmented scan
	SScan1	–	Segmented scan + stalls
	SScan2	–	Segmented scan
	SScan3	–	Segmented scan + stalls
	Oscan0	✓	Support stalls
	Oscan1	✓	Nonstall mode
	Oscan2	✓	Bidirectional transfers
	Oscan3	✓	Host to target only
	Oscan4	✓	Support stalls
	Oscan5	✓	Pipelined
	Oscan6	✓	Bidirectional transfers
Oscan7	✓	Host to target only	
Mscan	✓	Multi devices mode + stalls	
<b>Power Control</b>	Power down logic	–	Handled by ICEMelter



### 28.2.3 Trace Port

On-chip debug and trace events can be exported to external equipment through the trace port in the device. The following exportable debug events and trace sources are supported:

- Debug events
  - Triggers. For more information about triggers, see [Section 28.4.2, Cross-Triggering](#).
- Trace sources
  - Processor trace: Cortex-A9 MPU trace supported by program trace macrocell (PTM) and trace port interface unit (TPIU) modules. For more information about processor trace, see [Section 28.7, Processor Trace](#).
  - System trace: Trace coming from various system Instrumentation modules and supported by system trace module (STM). For more information about system trace, see [Section 28.8, System Instrumentation](#).

Note that not all debug and trace features can be supported concurrently due to limited amount of pins allocated to debug. Thus, a multiplexing among debug and trace sources is implemented. The configuration and the debug/trace source selection are done through the debug resource manager (DRM) module embedded in the debug subsystem.

[Table 28-5](#) describes the trace port signals.

**Table 28-5. Trace Port Signals**

Pin Name	Internal Signal Name	I/O	Description
dpm_emu19	EMU19	O	Debug resource manager pin 19
dpm_emu18	EMU18	O	Debug resource manager pin 18
dpm_emu17	EMU17	O	Debug resource manager pin 17
dpm_emu16	EMU16	O	Debug resource manager pin 16
dpm_emu15	EMU15	O	Debug resource manager pin 15
dpm_emu14	EMU14	O	Debug resource manager pin 14
dpm_emu13	EMU13	O	Debug resource manager pin 13
dpm_emu12	EMU12	O	Debug resource manager pin 12
dpm_emu11	EMU11	O	Debug resource manager pin 11
dpm_emu10	EMU10	O	Debug resource manager pin 10
dpm_emu9	EMU9	O	Debug resource manager pin 9
dpm_emu8	EMU8	O	Debug resource manager pin 8
dpm_emu7	EMU7	O	Debug resource manager pin 7
dpm_emu6	EMU6	O	Debug resource manager pin 6
dpm_emu5	EMU5	O	Debug resource manager pin 5
dpm_emu4	EMU4	O	Debug resource manager pin 4
dpm_emu3	EMU3	O	Debug resource manager pin 3
dpm_emu2	EMU2	O	Debug resource manager pin 2
dpm_emu1	EMU1	I/O	Debug resource manager pin 1
dpm_emu0	EMU0	I/O	Debug resource manager pin 0

**NOTE:** The dpm\_emu[19:0] pins are shared with other functional (application) pins on the device boundary. To use the dpm\_emu[19:0] pins, user must program the device application pin manager (control module) appropriately. For more information, see [Chapter 18, Control Module](#).

For more information about DRM multiplexing and concurrent debug modes, see [Section 28.9, Concurrent Debug Modes](#).

### 28.2.4 End Product

For an end product phone, there are usually severe connectivity constraints and the standard debug ports may not be available at board level. This could make it difficult to trace and debug software.

To ease the process of tracing and debugging, a Narrow Interface for Debug and Test (NIDnT) concept defined by MIPI is used. NIDnT is based on IEEE1149.7 and STM interfaces and re-uses application interface pins. In the context of the OMAP4430 device, the application interface that can serve as a NIDnT port is the MMC1 interface. To use the NIDnT port, the relevant MMC1 pins must be set to the appropriate MUX mode through control module.

Table 28-6 shows the NIDnT port re-using MMC interface pins along with the corresponding MUX modes.

**Table 28-6. NIDnT Port Reusing MMC Interface Pins**

Module	Control Module Multiplexing		
	MUX0 (MMC)	MUX2 (STM)	MUX4 (IEEE1149.7)
Extension card (MMC)	sdmmc1_clk	dpm_emu19	–
	sdmmc1_cmd	uart1_rx	–
	sdmmc1_dat0	dpm_emu18	–
	sdmmc1_dat1	dpm_emu17	–
	sdmmc1_dat2	dpm_emu16	jtag_tms_tmssc
	sdmmc1_dat3	dpm_emu15	jtag_tck

**NOTE:** The nTRST signal must be driven high for the MMC1 interface to be used for IEEE1149.7 protocol.

**NOTE:** Uart1\_rx is an optional return channel. Depending on the trace receiver, the return channel is implemented by IEEE1149.1/IEEE1149.7.

For more information about how to configure the proper MUX mode, see [Chapter 18, Control Module](#).

For more information about the NIDnT concept, see *MIPI Alliance Test and Debug – NIDnT-Port* specification.

## 28.3 Debugger Connection

### 28.3.1 ICEPick Module

The debugger connects to the OMAP4430 device through its JTAG interface. The first level of debug interface seen by the debugger is the IEEE1149.7 adapter connected to the ICEPick module embedded in the debug subsystem.

---

**NOTE:** ICEPick version D (ICEPick-D) is used in the OMAP4430.

---

System-on-chip (SoC) designs typically have multiple processors, each having a JTAG test access port (TAP) embedded in the processor. The ICEPick module manages these TAPs and the power, reset, and clock controls for modules that have TAPs.

The ICEPick module is visible only from the debugger point of view, and thus cannot be programmed by application software. The debugger can configure ICEPick through its own TAP controller. The ICEPick TAP has an instruction length of 6 bits and is the primary TAP. It is always visible in the scan chain and is used to control and monitor the other secondary TAPs.

ICEPick provides the following debug capabilities:

- Debug connect logic for enabling or disabling most ICEPick instructions
- Dynamic TAP insertion
  - Serially linking up to 16 TAP controllers
  - Individually selecting one or more of the TAPs for scan without disrupting the instruction register (IR) state of other TAPs
- Power, reset, and clock management
  - Provides the power and clock states of each domain
  - Provides debugger control of the power domain of a processor. Can force the domain power and clocks on, and prohibit the domain from being clock-gated or powered down while a debugger is connected.
  - Applies system reset
  - Provides wait-in-reset (WIR) boot mode
  - Provides global and local WIR release
  - Provides global and local reset blocking

The ICEPick module implements a connect register, which must be configured with a predefined key to enable the full set of JTAG instructions. Once the debug connect key has been properly programmed, ICEPick signals and subsystems emulation logics should be turned on.

For more information about ICEPick dynamic TAP insertion, see [Section 28.3.3, Dynamic TAP Insertion](#).

For more information about ICEPick power, reset, and clock management features, see [Section 28.5, Power, Reset, and Clock Management Debug Support](#).

### 28.3.2 Boot Modes

The initial configuration of ICEPick is determined by the level of the dpm\_emu0 and dpm\_emu1 pins upon POR release. At POR, dpm\_emu0 and dpm\_emu1 are automatically configured as inputs. The dpm\_emu0 and dpm\_emu1 pins are free once POR has been released.

[Table 28-7](#) summarizes the ICEPick boot modes.

**Table 28-7. ICEPick Boot Modes Upon POR**

dpm_emu1	dpm_emu0	TAPs in the TDI → TDO Path	Other Effects/Comments
0	0	None	Reserved (do not use)
0	1	None	Reserved (do not use)
1	0	ICEPick	TAP only + wait-in-reset mode
1	1	ICEPick	TAP only (default mode)

### 28.3.2.1 Default Boot Mode

In ICEPick-only configuration, none of the secondary TAPs are selected. The ICEPick TAP is the only TAP between device-level TDI and TDO pins. This is the recommended boot mode.

### 28.3.2.2 Wait-In-Reset

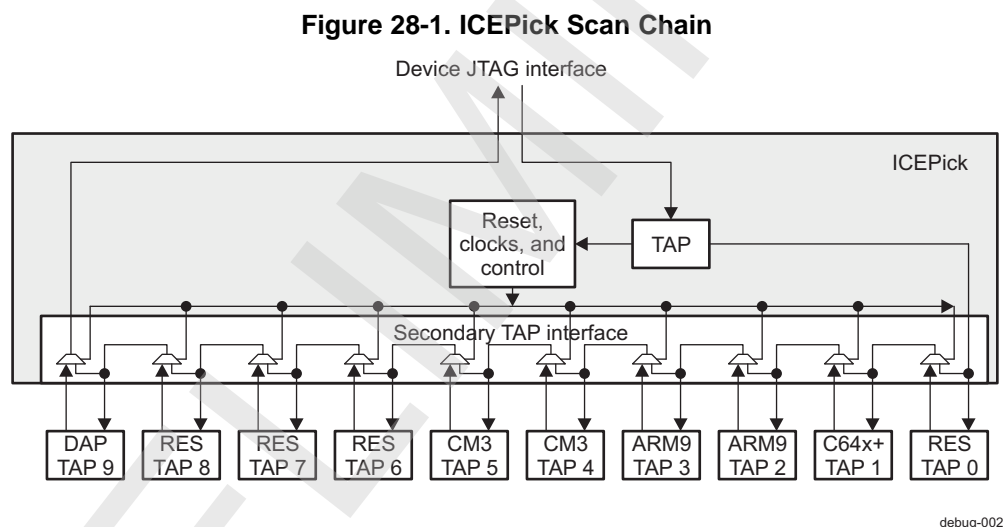
The device can boot to invoke WIR mode. If the device is booted in this mode, all processors within the device that support a TAP through ICEPick are held in reset until released. Individual processors may be released from reset (local), or all processors held in the reset state may be released at the same time (global).

## 28.3.3 Dynamic TAP Insertion

### 28.3.3.1 ICEPick Secondary TAPs

To include more or fewer secondary TAPs in the scan chain, the debugger must use the ICEPick TAP router to program the TAPs. At its root, ICEPick is a scan-path linker that lets the DTC selectively choose which subsystem TAPs are accessible through the device-level debug interface. Each secondary TAP can be dynamically included in or excluded from the scan path. From external JTAG interface point of view, secondary TAPs that are not selected appear not to exist.

Figure 28-1 shows the TAPs in the ICEPick scan chain.



**NOTE:** RES = Reserved/not used

CM3 = Cortex™-M3

DAP = Debug access port

Table 28-8 shows the secondary debug TAPs connected to the ICEPick scan chain along with the modules that can be accessed. The TAP number shows the position of the TAP in the scan chain.

**Table 28-8. ICEPick Secondary Debug TAP Mapping**

Secondary JTAG Port	CoreSight	TAP Number	Modules Accessed Through That JTAG Port
Reserved	No	0	–
DSP subsystem	No	1	C64x+™/ICEMaker
IVA-HD ICONT1	No	2	ARM968/ICECrusher™-9
IVA-HD ICONT2 (vDMA)	No	3	ARM968/ICECrusher-9

**Table 28-8. ICEPick Secondary Debug TAP Mapping (continued)**

Secondary JTAG Port	CoreSight	TAP Number	Modules Accessed Through That JTAG Port
Dual Cortex-M3 MPU subsystem	No	4	Cortex-M3 RTOS/ICECrusher-CS
	No	5	Cortex-M3 ISS CTL/ICECrusher-CS
Reserved	No	6	–
Reserved	No	7	–
Reserved	No	8	–
DAP APB-AP	Yes	9	Cortex-A9 x 2
	Yes		CS-PTM x 2
	Yes		CS-CTI x 2
	Yes		DAP-PC
	Yes		CS-ETB
	Yes		CS-TF x 2 <sup>(1)</sup>
	Yes		CS-TPIU
	No		DRM
	Yes		STM
DAP AHB-AP	No	9	IVA-HD SMSET
	No		IVA-HD hardware accelerators
	No		Statistics collectors (three instances)
	No		OCP-WP
	No		CM1 instrumentation
	No		CM2 instrumentation
	No		PM instrumentation
Reserved	No	1015	Reserved

(1) One instance in Cortex-A9 MPU subsystem; one instance in debug subsystem

Besides secondary debug TAPs, ICEPick supports also power, reset, and clock controls for non-JTAG debug cores. The debug cores are accessible through DAP.

Table 28-9 summarizes the ICEPick debug core mapping.

**Table 28-9. ICEPick Debug Core Mapping**

Debug Core #	Module
0	Cortex-A9 MPU subsystem
1	IVA-HD iLF3
2	IVA-HD iME3
3	IVA-HD CALC3
4	IVA-HD iPE3
5	IVA-HD MC3
6	IVA-HD ECD3

## 28.4 Primary Debug Support

### 28.4.1 Processor Native Debug Support

#### 28.4.1.1 Cortex-A9 MPU

The dual Cortex-A9 processor supports the following native debug features:

- Halt mode and monitor mode debugging
- Six breakpoints and four watch points
- Asynchronous aborts
- Performance monitoring
- Cross-triggering: Allows stopping one CPU upon debug event (for example, breakpoint) detection in the other CPU

For more information about Cortex-A9 native debug support features, see the *Cortex-A9 Technical Reference Manual*.

#### 28.4.1.2 Cortex-M3 MPU

The Cortex-M3 processor supports the following native debug features:

- Program halt and stepping
- Hardware breakpoints, breakpoint instruction
- Data watch point on access to data add, add range, and data value
- Register value accesses
- Debug monitor exception
- Memories accesses

For more information about Cortex-M3 native debug support features, see the *Cortex-M3 Technical Reference Manual*.

#### 28.4.1.3 DSP

This information is not available in the public domain.

#### 28.4.1.4 IVA-HD ARM968

The ARM968E-S processor supports the following native debug features through its EmbeddedICE-RT logic:

- Two hardware watch points/breakpoints
- Halt mode and monitor mode debugging
- Debug control and status registers
- Debug communications channel

For more information about ARM968 native debug support features, see the *ARM968E-S Technical Reference Manual*.

#### 28.4.1.5 IVA-HD Hardware Accelerators

Hardware accelerators offer the following debug capabilities through their embedded SyncBox module:

- Manual halt: Halt occurs at the SyncBox task boundary, when requested by the debugger
- Single-step execution at SyncBox task level (from 1 to  $n$  macroblocks depending on the user software)
- Cross-triggering: Hardware accelerator can halt, after the execution of current SyncBox task, based on trigger event detection
- Global run

[Table 28-10](#) summarizes the debug capabilities of the hardware accelerator SyncBox.

**Table 28-10. Hardware Accelerator Debug Capability Options**

Feature	iME3	iPE3	MC3	CALC3	iLF3	ECD3
Core reset	-	-	-	-	-	-
Execution request	-	-	-	-	-	-
Trigger output	-	-	-	-	-	-
Trigger input	✓	✓	✓	✓	✓	✓
Number of trigger channels	1	1	1	1	1	1
Number of counters	0	0	0	0	0	0
Number of watch points	0	0	0	0	0	0
Number of hardware breakpoints	0	0	0	0	0	0

### 28.4.2 Cross-Triggering

The device supports a cross-triggering feature, which provides a way to propagate debug (trigger) events from one processor subsystem/module to another. For example, a given *subsystem A* can be programmed to generate a debug event, which can then be exported as a global trigger across the device. Another *subsystem B* can be programmed to be sensitive to the trigger line input and to generate an action upon trigger detection.

Examples of debug events are: processor entering debug state, watch point match, PTM trigger, ETB full, ETB acquisition complete, etc.

Examples of debug actions are: debug request generation, restart (Cortex-A9 synchronous run), interrupt request generation, start/stop trace, etc.

Subsystems cross-triggering is consolidated at the device level by the XTRIGGER module, which is embedded in the debug subsystem.

---

**NOTE:** XTRIGGER is not programmatically visible from the JTAG interface or any device processor. Thus, cross-triggering is programmed at the subsystem level.

---

### 28.4.2.1 SOC Level Cross-Triggering

Table 28-11 summarizes the device cross-triggering capabilities.

**Table 28-11. XTRIGGER**

Subsystem	Module		Cortex-A9	PTM/E TB	Cortex-M3	Cortex-M3	Target Core	IVA3- C64x	IVA-HD ARM96 8	IVA-HD ARM96 8	HWA	PM Instr	CM Instr	OCP- WP	EMU0/ EMU1
		<b>Trigger Input / Trigger Source</b>	✓	✓	✓	✓	-	✓	✓	✓	-	✓	✓	✓	✓
Cortex-A9 MPU subsystem	Cortex-A9	✓		✓	✓	✓	-	✓	✓	✓	-	✓	✓	✓	
	PTM / ETB	✓	✓		✓	✓	-	-	-	-	-	✓	✓	✓	-
Dual Cortex-M3 MPU subsystem	Cortex-M3	✓	✓	-		✓	-	✓	✓	✓	-	✓	✓	✓	-
	Cortex-M3	✓	✓	-	✓		-	✓	✓	✓	-	✓	✓	✓	-
ABE subsystem	Target Core	-	-	-	-	-		-	-	-	-	-	-	-	-
DSP subsystem	C64x	✓	✓	-	✓	✓	-		✓	✓	✓	✓	✓	✓	✓
IVA-HD subsystem	ARM968 ICONT	✓	✓	-	✓	✓	-	✓		✓	✓	✓	✓	✓	✓
	ARM968 vDMA	✓	✓	-	✓	✓	-	✓	✓		✓	✓	✓	✓	✓
	HWA	-	-	-	-	-	-	-	-	-		-	-	-	-
Power manager	Instrumenta tion	-	-	-	-	-	-	-	-	-	-		-	-	-
Clock manager	Instrumenta tion	-	-	-	-	-	-	-	-	-	-	-		-	-
Device	OCP-WP	✓	✓	✓	✓	✓	-	✓	✓	✓	-	✓	✓		✓
	EMU0, EMU1	✓	✓	✓	✓	✓	-	✓	✓	✓	-	✓	✓	✓	

**NOTE:** The following modules support only trigger inputs:

- IVA-HD hardware accelerators
- IVA-HD SMSET
- PM instrumentation
- CM instrumentation



### 28.4.3 Suspend

The device supports a suspend feature, which provides a way to stop a "closely coupled" hardware process running on a peripheral-IP when the host processor enters debug state. The suspend mechanism is important for debug to ensure that peripheral-IPs operate in a lock-step manner with a host controller processor.

An entry is provided for each peripheral-IP that must consider the suspend signals from a number of processors (MPU or DSP). For each peripheral-IP, sensitivity to the suspend signals is defined within two possibilities (and so coded using 1bit):

- Peripheral-IP is sensitive to the suspend line request.
- Peripheral-IP ignores the suspend line request.

For more information about how to program the sensitivity, see the peripheral-IP TRM chapter.

#### 28.4.3.1 Debug Aware Peripherals and Host Processors

Table 28-12 lists the mapping of the device processors to the suspend control input lines.

**Table 28-12. Debug Subsystem Suspend Input Lines**

Suspend Input Line	Host Processor
0	Reserved
1	DSP C64x
2	IVAHD ICONT1
3	IVAHD ICONT2
4	Cortex-M3 RTOS
5	Cortex-M3 ISS CTL
6	Cortex-A9 CPU0
7	Cortex-A9 CPU1

Table 28-13 lists the mapping of the device peripheral-IPs to the suspend control output lines.

**Table 28-13. Debug Subsystem Suspend Output Lines**

Suspend Output Line	Peripheral-IP Module
0	Keyboard controller
1	MCBSP1 (ABE)
2	MCBSP2 (ABE)
3	MCBSP3 (ABE)
4	WDT3 (ABE)
5	GPTIMER5 (ABE)
6	GPTIMER6 (ABE)
7	GPTIMER7 (ABE)
8	GPTIMER8 (ABE)
9	Reserved for non-GP device
10	Reserved (not used)
11	Reserved for non-GP device
12	Reserved (not used)
13	Reserved for non-GP device
14	Reserved (not used)
15	Reserved for non-GP device
16	GPTIMER1
17	GPTIMER2
18	GPTIMER3
19	GPTIMER4

**Table 28-13. Debug Subsystem Suspend Output Lines (continued)**

Suspend Output Line	Peripheral-IP Module
20	GPTIMER9
21	GPTIMER10
22	GPTIMER11
23	Reserved for non-GP device
24	HSI
25	I2C1
26	I2C2
27	I2C3
28	I2C4
29	Reserved for non-GP device
30	sDMA
31	32-kHz synchronized timer
32	Reserved (not used)
33	Reserved (not used)
34	Reserved (not used)
35	Reserved for non-GP device
36	WDT1
37	WDT2

## 28.5 Power, Reset, and Clock Management Debug Support

The global PRCM module implements facilities to support debug across power and clock domain cycles. The debugger can control or get the status of each power and clock domain associated with an ICEPick secondary TAP.

ICEPick provides a set of directives allowing the debugger to:

- Get visibility on the associated power and clock domains state. This includes:
  - Current power setting indicating whether the power domain is on or off
  - Loss of power detected since the software last checked the status
  - Current clock setting indicating whether the clock domain is on or off
  - Sleep desired (PM and CM indicate that the debug settings in ICEPick are changing the application state. If it were not for the ICEPick controls, the power or clock would be turned off.)
  - Subsystem reset state
  - Subsystem has entered a debug state that requires the attention of the host debug software.
- Override power/clock control settings to wake up a power or clock domain or to prevent a power or clock domain from going to sleep once it is in ACTIVE state
- Assert/block/extend reset; release from extended reset (WIR)

ICEPick handles debug power management at the device level. In addition, the OMAP4430 implements a second level of debug power management through the DAP-PC module integrated in the Cortex-A9 MPU subsystem, which provides the same capabilities as ICEPick for the two Cortex-A9 CPUs. Each Cortex-A9 CPU core is in its own independent power domain, which DAP-PC manages. The hierarchical debug power-management approach provided by DAP-PC at the subsystem level and by ICEPick at the device level ensures a theoretically independent debug capability for each Cortex-A9 CPU core.

---

**NOTE:** The debugger must properly sequence global ICEPick and local DAP-PC commands:

- To force clocks/power:
    1. The debugger must first force it for the Cortex-A9 MPU subsystem at the ICEPick level.
    2. The debugger must then force it for a specific Cortex-A9 CPU at the DAP-PC level.
  - To release clocks/power:
    1. The debugger must first release it for a specific Cortex-A9 CPU at the DAP-PC level.
    2. The debugger must then release the Cortex-A9 MPU subsystem controls at the ICEPick level.
-

## 28.6 Performance Monitoring

### 28.6.1 Cortex-A9 MPU Subsystem Performance Monitoring

#### 28.6.1.1 Performance Monitoring Unit

The Cortex-A9 processor includes a performance monitoring unit (PMU) that enables events, such as cache misses and instructions executed, to be counted over a period of time. The PMU provides six counters to gather statistics about the operation of the processor and memory system. Each counter can count any of the events available in Cortex-A9.

The Cortex-A9 PMU outputs 29 events (PMUEVENT[28:0]) to PTM (see [Table 28-14](#)).

**Table 28-14. PMU Events**

PMU Event	Description
0	Software increment
1	Instruction fetch that causes a refill
2	Instruction fetch that causes a TLB refill
3	Data read or write operation that causes a refill
4	Data read or write operation that causes a cache access
5	Data read or write operation that causes a TLB refill
6	Data read architecturally executed
7	Data write architecturally executed
8	Number of instructions decoded
9	Exception taken
10	Exception return architecturally executed
11	Change to context ID retired
12	Software change of PC
13	Immediate branch architecturally executed
14	Number of predictable function returns
15	Unaligned access architecturally executed
16	Branch mispredicted or not predicted
17	Branches or other change in program flow
18	Java byte code executed
19	Software Java byte code executed
20	Jazelle backward branches executed
21	Coherent line fill request that misses in other uniprocessors
22	Request for coherent line fill that hits in other uniprocessors
23	Instruction cache dependent stalls
24	Data cache dependent stalls
25	TLB miss dependent stalls
26	STREX passed
27	STREX failed
28	Data eviction

For more information about Cortex-A9 PMU, see the *ARM Cortex-A9 Technical Reference Manual*.

#### 28.6.1.2 Snoop Control Unit

The individual CPU event monitors can be configured to gather statistics about the operation of the snoop control unit (SCU).

### 28.6.1.3 L2 Cache (PL310)

The Cortex-A9 MPU subsystem includes 512KB of L2 cache and uses PL310 as the L2 cache controller. The PL310 includes logic to support cache event monitoring.

Table 28-15 summarizes the PL310 L2 cache events.

**Table 28-15. PL310 L2 Cache Events**

Event	Event Description
0	Eviction of a line from the L2 cache
1	Data read hit in the L2 cache
2	Data read lookup to the L2 cache. Subsequently results in a hit or miss.
3	Data write hit in the L2 cache
4	Data write lookup to the L2 cache. Subsequently results in a hit or miss.
5	Data write lookup to the L2 cache with Write-Through attribute. Subsequently results in a hit or miss.
6	Instruction read hit in the L2 cache
7	Instruction read lookup to the L2 cache. Subsequently results in a hit or miss.
8	Prefetch line-fill sent to L3
9	Allocation into the L2 cache caused by a write (with Write-Allocate attribute) miss

The PL310 implements two 32-bit event counters. The PL310 can be configured to generate interrupts on error conditions or event counter overflow or increment. The PL310 interrupt is routed to MA\_IRQ\_0. When an interrupt occurs, software can look at the relevant interrupt status register to determine the source of the interrupt.

For more information about PL310, see the ARM *PrimeCell Level 2 Cache Controller (PL310) Technical Reference Manual*.

## 28.6.2 Cortex-M3 MPU Subsystem Performance Monitoring

### 28.6.2.1 Subsystem Counter Timer Module

The dual Cortex-M3 MPU subsystem includes a subsystem counter timer module (SCTM), which is embedded in shared cache and provides additional data to the user timing or profiling capability. SCTM integrates eight profiling counters that collect:

- Shared cache events
- Sleep and deep sleep events from each Cortex-M3

Table 28-16 describes the configuration of the Cortex-M3 SCTM counters.

**Table 28-16. Cortex-M3 SCTM Counters Configuration**

Counters	Features
0 – 1	Timer and event
2 – 3	64-bit chained + shadowing
4 – 5	64-bit chained + shadowing
6 – 7	Event

### 28.6.2.2 Cache Events

Table 28-17 summarizes the SCTM events for the Cortex-M3 MPU subsystem.

**Table 28-17. SCTM Events For Cortex-M3 MPU Subsystem**

Event	Event Description
0	Cache hit
1	Nonregistered tag hit information

**Table 28-17. SCTM Events For Cortex-M3 MPU Subsystem (continued)**

Event	Event Description
2	Cache miss
3	Read event
4	Write event
5	Fill buffer hit
6	Write buffer hit
7	Allocation ongoing
8	Bank miss
9	Stall due to busy ocp
10	Stall due to maintenance
11	Line evicted
12	Line replaced
13	Allocation failed due to locked lines

**NOTE:** These cache metrics can be exported as STP software messages through STM.

### 28.6.3 DSP Subsystem Performance Monitoring

#### 28.6.3.1 Subsystem Counter Timer Module

The DSP subsystem includes an SCTM, which is embedded in the DSP megamodule and provides additional data to the user timing or profiling capability. SCTM integrates eight profiling counters that collect L1 and L2 events.

[Table 28-16](#) describes the configuration of the DSP SCTM counters.

**Table 28-18. DSP SCTM Counters Configuration**

Counters	Features
0 – 1	Timer and event
2 – 7	Event (no chaining)

#### 28.6.3.2 L1 and L2 Events

[Table 28-19](#) summarizes the SCTM events for DSP subsystem.

**Table 28-19. SCTM Events For DSP Subsystem**

Event	Event Description
0	L1 Slave0 (program) tag hit
1	L1 Slave1(data) tag hit
2	L1 Slave0 (program) tag miss
3	L1 Slave1(data) tag miss
4	L1 Slave0 (program) fill buffer hit
5	L1 Slave1(data) fill buffer hit
6	L1 Slave1(data) write buffer hit
7	L1 Allocation FIFO
8	L1 Slave0 (program) bank conflict
9	L1 Slave1(data) bank conflict
10	L1 Slave0 (program) cached request
11	L1 Slave1(data) cached request
12	L1 Slave0 (program) waiting for fill buffer

**Table 28-19. SCTM Events For DSP Subsystem (continued)**

<b>Event</b>	<b>Event Description</b>
13	L1 Slave1(data) waiting for fill buffer
14	L1 Slave0 (program) waiting for maintenance
15	L1 Slave1(data) waiting for maintenance
16	L1 Eviction occurred
17	L1 Replacement occurred
18	L1 Allocation failed due to locked lines occurred
19	L2 tag hit
20	L2 tag miss
21	L2 fill buffer hit
22	L2 write buffer hit
23	L2 allocation FIFO
24	L2 cached request
25	L2 waiting for fill buffer
26	L2 waiting for maintenance
27	L2 eviction occurred
28	L2 replacement occurred
29	L2 allocation failed due to locked lines occurred

---

**NOTE:** These cache metrics can be exported as STP software messages through STM.

---

## 28.7 Processor Trace

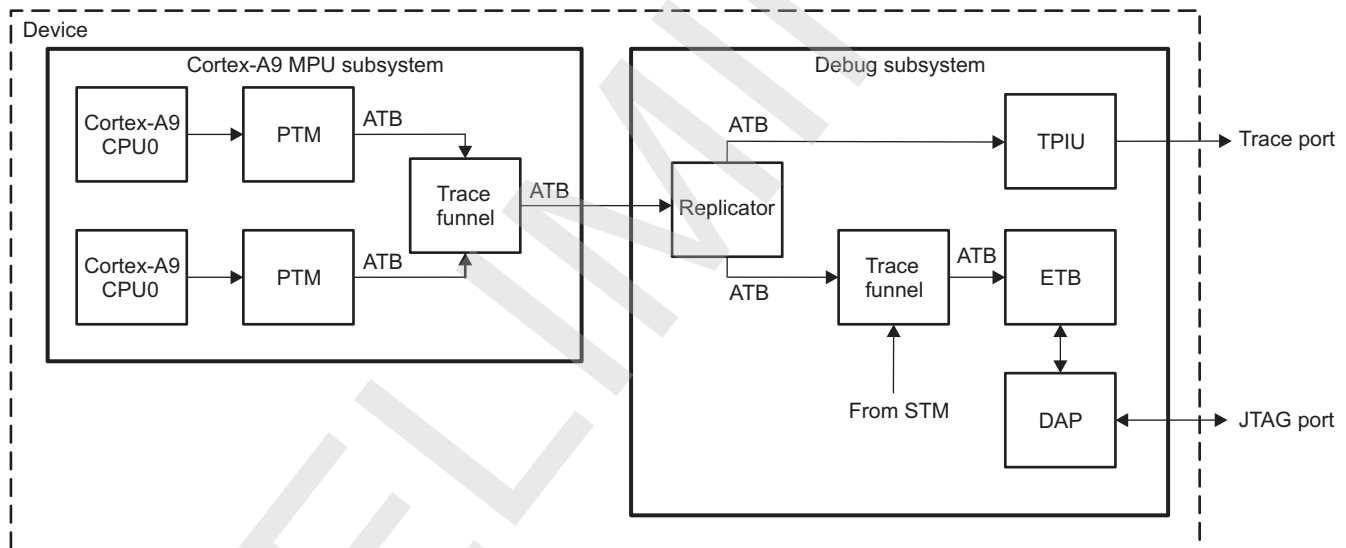
The device supports only Cortex-A9 processor trace. The main Cortex-A9 processor trace characteristics are:

- Program trace only (no data trace)
- Separate PTM for each Cortex-A9 CPU
- Trace can be:
  - Exported off-chip through TPIU in the debug subsystem (maximum number of trace data pins = 18)
  - Stored on-chip to ETB through CS-TF in the debug subsystem
- Trace can be optionally:
  - Cycle accurate useful for profiling sections of code
  - Interleaved dynamically between the two Cortex-A9 CPUs by a CS-TF embedded in the Cortex-A9 MPU subsystem

**NOTE:** Depending on use case, the device may not be able to interleave two cycle accurate traces without overflow (bandwidth restriction).

Figure 28-2 shows an overview of the processor trace flow.

**Figure 28-2. Processor Trace Flow**



debug-004



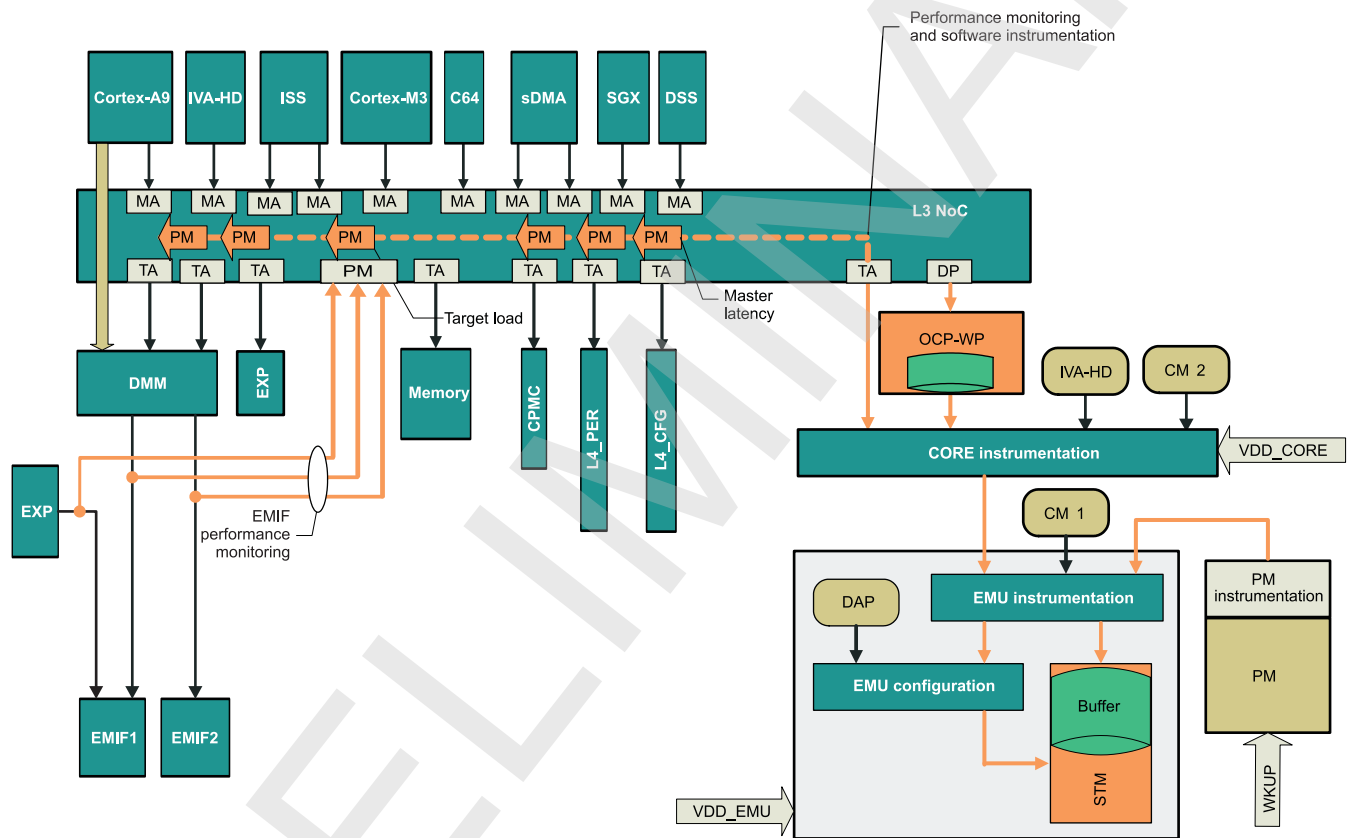
## 28.8 System Instrumentation

The device supports the following system instrumentation features:

- Real-time software trace (see [Section 28.8.2, Software Instrumentation](#))
- OCP target traffic monitoring (see [Section 28.8.3, OCP Watch-Point](#))
- OCP target load and master latency monitoring (see [Section 28.8.5, NoC Statistics Collector](#))
- HWA load monitoring (see [Section 28.8.4, IVA-HD Pipeline](#))
- PM events trace (see [Section 28.8.6, PM Instrumentation](#))
- CM events trace (see [Section 28.8.7, CM Instrumentation](#))

Figure 28-3 is an overview of the device system instrumentation framework.

Figure 28-3. System Instrumentation Framework



debug-005

### 28.8.1 MIPI STM

MIPI-STM is a trace module that aids in software debugging. The main features of this module are:

- Implements MIPI STP protocol (rev 1.0) with the following characteristics:
  - Highly optimized for software-generated traces
  - Automatic timestamping of messages
  - Support for 8-, 16-, and 32-bit data types
- Collects the following information:
  - Software messages
  - Hardware instrumentation trace from hardware agents:
    - OCP-WP
    - PM profiler
    - CM profiler

- SMSET
- L3 NoC statistics collectors
- Exports trace data to:
  - External trace receiver
  - On-chip ETB buffer
- Available in 1-, 2-, or 4-pin mode with single- or dual-edge clock, depending on the trace bandwidth requirements and characteristics of the trace receiver
- Dedicated 128 × 32-bit FIFO buffer

A maximum of 255 different bus masters can be connected to the STM trace port through a bus arbiter. STP recognizes two distinct modes of tracing (software and hardware types), which use slightly different message combinations to output different types of data. The bus masters can be configured for either type to optimize the system for the different types of trace data.

### 28.8.2 Software Instrumentation

The device provides support for real-time software trace through user-defined message writes to specific memory mapped register (MMR) locations. Software masters can transmit trace data from the operating system (OS) processes or tasks on 256 different channels, with each channel being defined by the software protocol implemented. The different channels can be used to group different types of data logically so that it is easy to filter out the data irrelevant to the on-going debugging task. The message structures in STP are optimized to provide an efficient transport for software data through the STM module.

The software masters are:

- Cortex-A9 MPU subsystem
- DAP (for testing purpose)
- DSP subsystem
- Dual Cortex-M3 MPU subsystem
- sDMA controller write port (SDMA\_WR)
- IVA-HD subsystem

Each software master has a master-ID assigned to it.

Software messages can be interleaved with other hardware messages.

Software messages are intrusive and use both processor cycles and memory.

### 28.8.3 OCP Watch-Point

The L3 interconnect provides five functional probes embedded and attached to the following L3 targets:

- GPMC
- L4-PER
- L4-CFG
- DMM1 (DMM target port 1)
- DMM2 (DMM target port 2)

The probes output are muxed together and then sent to the L3 interconnect debug port. A component called OCP-WP is used to collect data from functional probes and then transmit captured data to the STM module. The OCP-WP drives a Probe-ID signal to the L3 interconnect for probe selection. The probe selection is exclusive, meaning that interleaving is not possible.

The OCP-WP provides the following main features:

- Monitoring the OCP traffic originated by all initiators that can access the selected target where the probe is attached
- Filtering OCP monitored bus traffic by:
  - Address range
  - Initiator-ID (see [Section 13.2, L3 Interconnect](#), for initiator-ID mapping)

- Transaction type
- Transaction qualifier
- Generating a trigger upon watch-point match
- Starting and stopping OCP traffic monitoring upon:
  - WP address match
  - External trigger
- OCP-WP messages can be interleaved with software messages
- Programming from:
  - Debugger
  - Application

**NOTE:** The OCP-WP is restricted to monitor request flow.

Table 28-20 summarizes the OCP targets that can be monitored by the OCP-WP and their respective probe-ID.

**Table 28-20. L3 Interconnect Functional Probe Mapping**

Probe-ID	L3 OCP Target
000	Reserved
001	GPMC
010	L4-PER
011	L4-CFG
100	DMM1 (DMM target port 1)
101	DMM2 (DMM target port 2)
110	Reserved
111	Reserved

### 28.8.4 IVA-HD Pipeline

The device takes advantage of the system trace infrastructure to provide visibility to the user regarding IVA-HD micro-task sequencing. This is supported through a SMSET module instantiated in the IVA-HD subsystem. The micro-task boundaries are handled as generic events and encapsulated in STP messages with an event-ID and local timestamp and exported through the MIPI-STM module.

The IVA-HD instrumentation scheme allows the user to understand micro-task dependencies, hardware accelerators load balancing, and potential bottlenecks. DMA transfer boundaries are reported as IVA-HD events. Software messages from ARM968 execution can be interleaved with IVA-HD events.

### 28.8.5 NoC Statistics Collector

The L3 interconnect supports a built-in performance monitoring feature by implementing a statistics collector (STATCOLL) component, which computes traffic statistics within a user-defined window and periodically reports to the user through the MIPI-STM interface. Three STATCOLL instances are instantiated in the device:

- One statistics collector dedicated to SDRAM load monitoring – STATCOLL\_SDRAM (see [Section 28.8.5.1, OCP Target Load Monitoring](#), for more information)
- Two statistics collectors dedicated to master latency monitoring – STATCOLL\_LAT0 and STATCOLL\_LAT1 (see [Section 28.8.5.2, OCP Master Latency Monitoring](#), for more information)

Statistics collectors (SDRAM and LAT0/1) can report:

- Average burst length in bytes/packet per sampling window
- Average throughput in bytes/cycle
- % Link occupancy on the request link (for store transactions) during a sampling window
- % Link occupancy on the response link (for load transactions) during a sampling window

- % Arbitration conflict cycles on the request link
- % Initiator busy cycles on the response link
- Histogram of payload length in bytes (for example, 0–16 , 16–32, 32–128) each sampling window.
- Histogram of quality of service (QoS) metric for IVA-HD initiator (for example, low priority, high priority) each sampling window.

The performance metrics are interleaved with software instrumentation data at the L3 interconnect level.

The performance monitoring probes implement three main functions:

- Events detection
- Transactions filtering
- Aggregation

The probes can be configured to detect the events summarized in [Table 28-21](#).

**Table 28-21. Performance Monitoring Events Detection**

Event	Definition
NONE	No event selected
ANY	Any clock cycles
TRANSFER	Word has been accepted by the receiver.
WAIT	Transfer has been initiated but the transmitter currently has no data to send.
BUSY	Receiver applies flow control
PKT	Transfer of a new packet header
DATA	Transfer of a payload word
IDLES	No communication over the link
LATENCY	Debug bit detection

The probes can be configured to filter the traffic based on the criteria summarized in [Table 28-22](#).

**Table 28-22. Performance Filtering Options**

Probe Filtering Options	Comment
Master address	Mask and match
Slave address	
UserInfo	
Read	Opcode is a load.
Write	Opcode is a store.
Error	–

The probes implement a user-defined set of counters that aggregate the events sampled by the detector and filtered according to the user setup.

**NOTE:** Statistics collectors counter values are not accessible by application software.

[Table 28-23](#) summarizes the performance probe aggregation modes.

**Table 28-23. Aggregation Modes**

Aggregation Mode	Description
FILTER_HIT	The counter increments by 1 when the filter hits.
MIN_MAX_HIT	The counter increments by 1 when the filter hits and the selected event information is within range. <ul style="list-style-type: none"> <li>- Payload length (bytes)</li> <li>- Pressure value</li> <li>- Request/response latency (clock cycles)</li> </ul>
EVT_INFO	The selected event information is added to the counter value when the filter hits.

**Table 28-23. Aggregation Modes (continued)**

Aggregation Mode	Description
	Payload length (bytes) Pressure value Request/response latency (clock cycles)
AND_FILTER	The counter increments by 1 when all unit filters hit.
OR_FILTER	The counter increments by 1 when at least one unit filter hits.
SUM_REQ_EVT	The counter sums the events from any request port.
SUM_RSP_EVT	The counter sums the events from any response port.
SUM_ALL_EVT	The counter sums the events from any port.
EXT_EVT	The counter increments by 1 when selected external event input signal is sampled high.

### 28.8.5.1 OCP Target Load Monitoring

The L3 interconnect implements three performance monitoring probes on LPDDR2 memory channels. The OCP traffic statistics are computed within a user-defined window and periodically reported to the user through the MIPI-STM interface.

STATCOLL\_SDRAM supports the following main features:

- Three probe inputs
  - Probe 0 – EMIF1 128-bit memory channel
  - Probe 1 – EMIF2 128-bit memory channel
  - Probe 2 – EMIF 64-bit EXP interface – allows monitoring traffic from C2C (shared memory)
- Five 16-bit counters shared concurrently
  - Counter 0 with two filters
  - Counter 1 with one filter
  - Counter 2 with two filters
  - Counter 3 with one filter
  - Counter 4 without any filter
- Simple (with one element) or complex (with several elements) filters available
- Filtering according to:
  - Initiator of traffic
  - Access priorities
- No latency counter. Only bandwidth measurement on this collector
- Dump Identifier is 0x0 (tie-off value)
- Dumps frames at L3 interconnect slave address 0x19 (debug subsystem)

Table 28-24 summarizes the STATCOLL\_SDRAM configuration.

**Table 28-24. STATCOLL\_SDRAM Configuration**

Counters	Min Max	Filter Elements	L3 Target		
			EMIF1	EMIF2	EXP
Counter 0	Yes	2	EMIF1	EMIF2	EXP
Counter 1	Yes	1	EMIF1	EMIF2	EXP
Counter 2	Yes	2	EMIF1	EMIF2	EXP
Counter 3	Yes	1	EMIF1	EMIF2	EXP
Counter 4	Yes	No	No	No	EXP

### 28.8.5.2 OCP Master Latency Monitoring

The device L3 interconnect implements nine performance monitoring probes on the main L3 initiators:

- Cortex-A9 MPU Subsystem

- DSP
- sDMA controller read port (SDMA\_RD)
- sDMA controller write port (SDMA\_WR)
- IVA-HD
- DSS
- ISS
- SGX
- Dual Cortex-M3 MPU subsystem

The OCP master latency statistics are computed within a user-defined window and periodically reported to the user through the MIPI-STM interface.

The probes can be configured to filter latencies in four classes and report to the user a latency distribution along execution.

Because the performance metrics and the software events are exported through a unified export channel, it is possible to correlate latency trends with on-going execution and system context.

Because computing latency requires maintaining the state between request and response ports, the probe cannot compute latency statistics on 100 percent of the initiator traffic. Hence, latency histograms must be extracted on large execution windows to be accurate.

#### 28.8.5.2.1 STATCOLL\_LAT0 Configuration

STATCOLL\_LAT0 supports the following main features:

- Six probe inputs:
  - Probe 0: Cortex-A9 MPU subsystem
  - Probe 1: DSP subsystem
  - Probe 2: sDMA controller RD port
  - Probe 3: sDMA controller WR port
  - Probe 4: DSS
  - Probe 5: ISS
- Four 20-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- Identifier is 0x1 (tie-off value)
- Dumps frames at slave address 0x19 (DEBUGSS)

[Table 28-25](#) summarizes the STATCOLL\_LAT0 configuration.

**Table 28-25. STATCOLL\_LAT0 Configuration**

Counters	Min Max	Filter Elements	L3 Master					
			Cortex-A9 MPU subsystem	DSP	sDMA read	sDMA write	DSS	ISS
Counter 0	Yes	1	Cortex-A9 MPU subsystem	DSP	sDMA read	sDMA write	DSS	ISS
Counter 1	Yes	1	Cortex-A9 MPU subsystem	DSP	sDMA read	sDMA write	DSS	ISS
Counter 2	Yes	1	Cortex-A9 MPU subsystem	DSP	sDMA read	sDMA write	DSS	ISS

**Table 28-25. STATCOLL\_LAT0 Configuration (continued)**

Counter 3	Yes	1	Cortex-A9 MPU subsystem	DSP	sDMA read	sDMA write	DSS	ISS
-----------	-----	---	-------------------------	-----	-----------	------------	-----	-----

### 28.8.5.2.2 STATCOLL\_LAT1 Configuration

STATCOLL\_LAT1 supports the following main features:

- Five probe inputs:
  - Probe 0: IVA-HD
  - Probe 1: Cortex-A9 MPU subsystem
  - Probe 2: sDMA controller WR port
  - Probe 3: SGX
  - Probe 4: Dual Cortex-M3 MPU subsystem
- Four 20-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- Identifier is 0x2 (tie-off value)
- Dumps frames at slave address 0x19 (DEBUGSS)

Table 28-26 summarizes the STATCOLL\_LAT1 configuration.

**Table 28-26. STATCOLL\_LAT1 Configuration**

Counters	Min Max	Filter Elements	L3 Master				
			IVA-HD	Cortex-A9 MPU subsystem	sDMA write	SGX	Dual Cortex-M3 MPU subsystem
Counter 0	Yes	1	IVA-HD	Cortex-A9 MPU subsystem	sDMA write	SGX	Dual Cortex-M3 MPU subsystem
Counter 1	Yes	1	IVA-HD	Cortex-A9 MPU subsystem	sDMA write	SGX	Dual Cortex-M3 MPU subsystem
Counter 2	Yes	1	IVA-HD	Cortex-A9 MPU subsystem	sDMA write	SGX	Dual Cortex-M3 MPU subsystem
Counter 3	Yes	1	IVA-HD	Cortex-A9 MPU subsystem	sDMA write	SGX	Dual Cortex-M3 MPU subsystem

### 28.8.6 PM Instrumentation

The device takes advantage of the system trace infrastructure to provide visibility to the user about the major power management events. This is supported through a PM profiler module instantiated in the PRM module. The PRM state changes are handled as generic events and encapsulated in STP hardware messages and exported through the MIPI-STM module. The nature of the PM events does not require accurate timestamping and thus, timestamping is handled at STM or trace receiver level.

The PM events are organized by class. Any PM state change from a specific class refreshes the entire instrumentation frame associated with that class. The STP message structure includes a PM event-ID indicating the class of the PM events.

The PM event classes supported are:

- Logic voltage domain OPP change
- Memory voltage domain OPP change
- Logic power domain state change

- Memory power domain state change

The PM profiler has a unique hardware master-ID assigned to it.

The PM profiler supports the possibility to report on activity in different event classes in the same sampling window. The user has the flexibility to size the capture sampling window.

Software events from the PM routines instrumentation can be interleaved with the PM hardware events. The user can take advantage of that feature to understand latencies for a specific power-management scenario or strategy.

### 28.8.7 CM Instrumentation

The device instantiates two CM profilers, one in CM1, and one in CM2. The CM1 and CM2 profilers are instantiations of the same debug-IP and can operate concurrently. Each of them has a separate unique hardware master-ID assigned to it.

The CM events profiling is similar to the PM instrumentation. Two exclusive instrumentation modes are supported:

- Clock activity:
  - Exposes to the user a snapshot of the state of all the clock domains derived from the same DPLL when CM detects a state change in the clock domain
  - Exposes to the user a snapshot of the DPLL settings when the CM signals a DPLL programming
- Module activity:
  - Exposes to the user periodically the active cycles count of the target modules
  - Exposes to the user periodically the active cycles count of the initiator modules

It provides visibility to the user about the state of the major clock domains along the application code execution. The STP message reports the effective state of the clock domain and therefore can highlight scenarios where a particular dependency is preventing the clock domain from being switched off.

The CM event classes supported are:

- Events capture mode – four classes:
  - Clock domain state
  - DPLL settings update
  - Clock frequency divider ratio
  - Clock source selection update
- Module activity collection mode – two classes:
  - Target module activity
  - Initiator module activity

When in events capture mode, the CM profiler supports the possibility of reporting on activity in different event classes in the same sampling window. The user has the flexibility to size the capture sampling window.



## 28.9 Concurrent Debug Modes

The debugger or application software can program the DRM to route a specific debug function to each device debug port pin.

Because of the limited number of pins allocated to debug, debug and trace source signals are multiplexed.

[Table 28-27](#) summarizes the trace port configuration.

**Table 28-27. Trace Port Configuration**

Pin Name	Internal Signal Name	I/O	Trigger	JTAG 20-Pin Header	TPIU PTM (ARM Trace)		STM (System Trace)		
					16-Bit Mode	18-Bit Mode	Option 1	Option 2	Option 3
dpm_emu19	EMU19	O		EMU4	TRACEDATA[15]	TRACEDATA[17]		STM_CLK	
dpm_emu18	EMU18	O		EMU3	TRACEDATA[14]	TRACEDATA[16]		STM_DATA[0]	
dpm_emu17	EMU17	O		EMU2	TRACEDATA[13]	TRACEDATA[15]		STM_DATA[1]	
dpm_emu16	EMU16	O			TRACEDATA[12]	TRACEDATA[14]		STM_DATA[2]	
dpm_emu15	EMU15	O			TRACEDATA[11]	TRACEDATA[13]		STM_DATA[3]	
dpm_emu14	EMU14	O			TRACEDATA[10]	TRACEDATA[12]		UART1_RX	
dpm_emu13	EMU13	O			TRACEDATA[9]	TRACEDATA[11]			
dpm_emu12	EMU12	O			TRACEDATA[8]	TRACEDATA[10]			
dpm_emu11	EMU11	O			TRACEDATA[7]	TRACEDATA[9]			
dpm_emu10	EMU10	O			TRACEDATA[6]	TRACEDATA[8]			
dpm_emu9	EMU9	O			TRACEDATA[5]	TRACEDATA[7]			
dpm_emu8	EMU8	O			TRACEDATA[4]	TRACEDATA[6]			
dpm_emu7	EMU7	O			TRACEDATA[3]	TRACEDATA[5]			
dpm_emu6	EMU6	O			TRACEDATA[2]	TRACEDATA[4]			
dpm_emu5	EMU5	O			TRACEDATA[1]	TRACEDATA[3]			
dpm_emu4	EMU4	O			TRACEDATA[0]	TRACEDATA[2]	STM_DATA[3]		
dpm_emu3	EMU3	O			TRACECTL	TRACECTL	STM_DATA[2]		
dpm_emu2	EMU2	O			TRACECLK	TRACECLK	STM_CLK		
dpm_emu1	EMU1	I/O	Trigger 1	EMU1		TRACEDATA[1]	STM_DATA[1]		STM_CLK
dpm_emu0	EMU0	I/O	Trigger 0	EMU0		TRACEDATA[0]	STM_DATA[0]		STM_DATA[0]

**NOTE:** The configuration of the trace port must comply with [Table 28-27](#); otherwise, it will be ignored by DRM hardware. For example, if Trigger0 is programmed on EMU10 and Trigger1 is programmed on EMU11, this configuration will be ignored.

[Table 28-28](#) summarizes the concurrent debug and trace in the device.

**Table 28-28. Concurrent Debug and Trace**

Debug Use Case	Concurrent Debug Flows	Debug Pins	Trace Pins		
		Triggers	Data	Control	Clock
0	PTM		18	1	1
	STM				
	Triggers				
1	PTM		16	1	1
	STM				
	Triggers	2			
2	PTM		16	1	1
	STM		1	–	1
	Triggers				
3	PTM		8	1	1
	STM		4	–	1
	Triggers	2			
4	PTM		11	1	1
	STM		4	–	1
	Triggers	2			

## 28.10 Memory Mapping

Table 28-29 summarizes the memory mapping of the debug modules.

**Table 28-29. Debug Modules Memory Mapping**

Memory Space	Module Name	Start Address (hex)	End Address (hex)	Size
L3_EMU	MIPI-STM (256 x 4K channels) (address space 0)	0x5400 0000	0x540F FFFF	1MB
	MIPI-STM (256 x 1K channels) (address space 1)	0x5410 0000	0x5413 FFFF	256KB
	Cortex-A9 CPU0 debug/PMU	0x5414 0000	0x5414 1FFF	8KB
	Cortex-A9 CPU1 debug/PMU	0x5414 2000	0x5414 3FFF	8KB
	CTI0 component	0x5414 8000	0x5414 8FFF	4KB
	CTI1 component	0x5414 9000	0x5414 9FFF	4KB
	PTM0 component	0x5414 C000	0x5414 CFFF	4KB
	PTM1 component	0x5414 D000	0x5414 DFFF	4KB
	CS-TF <sup>(1)</sup>	0x5415 8000	0x5415 8FFF	4KB
	DAP-PC	0x5415 9000	0x5415 9FFF	4KB
	DRM	0x5416 0000	0x5416 0FFF	4KB
	MIPI-STM	0x5416 1000	0x5416 1FFF	4KB
	CS-ETB	0x5416 2000	0x5416 2FFF	4KB
	CS-TPIU	0x5416 3000	0x5416 3FFF	4KB
	CS-TF <sup>(2)</sup>	0x5416 4000	0x5416 4FFF	4KB
L4_CFG	PMI-CM1	0x4A00 4F00	0x4A00 4FFF	256B
	PMI-CM2	0x4A00 9F00	0x4A00 9FFF	256B
	OCF-WP	0x4A10 2000	0x4A10 2FFF	4KB
L4_WKUP	PMI-PM	0x4A30 7F00	0x4A30 7FFF	256B
IVA-HD configuration	ICECrusher-9 (ICONT1)	0x5A04 0000	0x5A04 07FF	2KB
	ICECrusher-9 (ICONT2)	0x5A04 0800	0x5A04 0FFF	2KB
	SMSET	0x5A04 1000	0x5A04 1FFF	4KB
L3 configuration	STATCOLL_SDRAM	0x4500 0400	0x4500 05FF	512B
	STATCOLL_LAT0	0x4500 0600	0x4500 07FF	512B
	STATCOLL_LAT1	0x4500 0800	0x4500 09FF	512B
DSP configuration	SCTM	0x01C3 0400	0x01C3 05FF	512B
Cortex-M3 MPU subsystem configuration	SCTM	0x5508 0400	0x5508 07FF	1KB
	ICECrusher-CS x 2 <sup>(3)</sup>	0xE004 2000	0xE004 2FFF	4KB

(1) Embedded in Cortex-A9 MPU subsystem

(2) Embedded in debug subsystem

(3) Private memory access per Cortex-M3 core



---



---

## Glossary

---

**3**

- 3D**— Three Dimensional  
**3G**— 3rd Generation of mobile communication systems

**A**

- A/D**— Analog to Digital Converter  
**ABB**— Adaptive Body Bias  
**ABE**— Audio Back End  
**ABEFW**— Audio Back End Firewall  
**ACB**— ac-bias frequency  
**ACBI**— ac-bias line transitions per interrupt  
**ACE**— ASIC Compiler Environment.  
**ACK**— Acknowledge  
**ADC**— Analog-to-Digital Converter/Conversion.  
**ADMA**— Audio DMA  
**ADPLL**— All Digital Phase Locked Loop. A closed loop frequency control system whose function is based on the phase-sensitive detection of the phase difference between the input signal and the output signal of the controlled oscillator (CO).  
**AE**— Audio Engine. The AESS own processor.  
**AES**— Advanced Encryption Standard  
**AESS**— Audio Engine Subsystem  
**AEW**— Auto Exposure and Auto White Balance  
**AF**— Auto Focus  
**AFE**— Analog Front End  
**AHB**— Advanced High-performance Bus  
**ALE**— Address Latch Enable  
**AMMU**— Attribute Memory Management Unit  
**ANSI**— American National Standards Institute  
**AP**— Address Protection  
**APB**— Advanced Peripheral Bus

**APE**— Application Engine

**API**— Application Programming Interface

**AR**— Automatic Reload

**ARGB**— Alpha Red Green Blue

**ARM**— Advanced RISC Machine

**ASCII**— American Standard Code for Information Interchange

**ASIC**— Application-Specific Integrated Circuit. A chip built for a particular application. In the context of this document, this refers to the FPGA that resides on the EVM board.

**ASP**— Application-Specific Peripheral

**ATA**— Interface standard for the connection of storage devices

**ATB**— AMBA Advanced Trace Bus

**ATC**— Audio Traffic Controller

**AVDAC**— Audio-Video DAC

**AVS**— Adaptive Voltage Scaling

**AWB**— Auto White Balance

**B**

**B**— Byte, 8 bits

**BAYER**— Bayer Filter mosaic. A color filter array (CFA) for arranging RGB color filters on a square grid of photosensors.

**BB**— Bus Busy

**BCD**— Binary-Coded Decimal.

**BCH**— Bose-Chaudhuri-Hocquenghem

**BE**— Big Endian.

**BGA**— Ball Grid Array

**BGAP**— Band Gap

**BIOS**— Built-In Operating System

**BIST**— Built-In Self-Test

**BL**— Buffer Logic

**BOF**— Beginning of Frame

**BPP**— Bits Per Pixel

**BS**— Block Synchronization

**BSC**— Boundary Scan Chain

**BSMEM**— Bit Stream Memory

**BTA**— Bus Turn Around

**BTE**— Burst Translation Engine

**BW**— Band Width

**Bluetooth**— A short-range radio technology aimed at simplifying communications among network devices and between devices and the Internet. It also aims to simplify data synchronization between network devices and other computers.

## C

**C2C**— Chip-to-Chip

**CABAC**— Context-Adaptive Binary Arithmetic Coding

**CALC3**— Transform and Quantization Calculation Engine

**CAS**— Column Address Strobe

**CAVLC**— Context-Adaptive Variable Length Coder

**CB**— Copy Back

**CBP**— Coded Block Pattern

**CBUFF**— Circular Buffer

**CCCR**— Card Common Control Registers (SDIO)

**CCP**— Compact Camera Port

**CDP**— Coprocessor Data Operation

**CDR**— Clock Data Recovery

**CE**— Chip Enable

**CEA-861-D**— HDMI 1.3 video standard. It defines the video timing requirements, discovery structures, and data transfer structure.

**CEC**— Consumer Electronics Control

**CFA**— Color Filter Array

**CH**— Configuration Header. To use different settings than ROM Code defaults, that is, clock frequencies, SDRAM/DDRAM settings, GPMC settings if the customer wants.

**CID**— Card Identification Number

**CIR**— Consumer Infra Red

**CLE**— Command Latch Enable

**CLK**— Clock

**CLUT**— Color Look-Up Table

**CM**— Clock Management

**CMEM**— Coefficient Memory

**CMOS**— Complimentary Metal Oxide Semiconductor

**CODEC**— Coder/Decoder or Compression/Decompression. A device that codes in one direction of transmission and decodes in another direction of transmission.

**CP15**— Coprocessor 15. This coprocessor controls the operation and configuration of the TI925T.

**CPR**— Clock, Power, Reset

- CPU**— Central Processing Unit.
- CRC**— Cyclic Redundancy Check
- CS**— Chip-Select
- CS-ETB**— CoreSight Embedded Trace Buffer
- CS-ETM**— CoreSight Embedded Trace Macrocell
- CS-TPIU**— CoreSight Trace Port Interface Unit
- CSI**— Camera Serial Interface
- CSW**— Control Space Width
- CSWR**— Closed switch retention
- CTM**— Counter-Timer Module
- CTRL**— Control
- CTS**— Clear to Send
- CVBS**— Composite Video Broadcast Signal
- ConnID**— Connection Identifier. An Initiator Module Identifier. A ConnID is transmitted in-band with the request and is used for protection and error logging mechanism.

**D**

- D2D**— Die-to-Die
- DAC**— Digital to Analog Converter
- DBB**— Digital Baseband
- DBI**— Display Buffer Interface
- DCD**— Data Carrier Detect
- DCO**— Digitally Controlled Oscillator
- DCS**— Display Command Set
- DCT**— Discrete Cosine Transform.
- DDC**— Display Data Channel
- DDMA**— DSP Subsystem Direct Memory Access module
- DDR**— Double Data Rate
- DE**— Data Enable
- DFF**— Digital Flip-Flop
- DFT**— Design For Test
- DI**— Data In
- DISPC**— Display Controller
- DL**— Data Length
- DLB**— Data Loopback.
- DLL**— Delay-Locked Loop



- DMA**— Direct Memory Access.
- DMC**— Data Memory Controller
- DMEM**— Data Memory
- DMIC**— Digital Microphone Controller
- DMM**— Dynamic Memory Management
- DNL**— Differential nonlinearity. It describes the deviation between two analog values corresponding to adjacent input digital values for a DAC.
- DO**— Data Out
- DPC**— Defect Pixel Correction
- DPCM**— Differential Pulse Code Modulation
- DPI**— Display Parallel Interface
- DPLL**— Digital Phase-Locked Loop. Digital implementation of PLL.
- DPS**— Digital Power Switching
- DRD**— Dual-Role Device (a type of OTG USB device)
- DRDY**— Data Ready
- DRM**— Digital Rights Management
- DRP**— Digital Radio Processor
- DSI**— Display Serial Interface
- DSP**— Digital Signal Processor.
- DSR**— Data Set Ready
- DSS**— Display Sub-System. Also DISS
- DT**— Data Type
- DTBC**— Data Buffer Controller
- DTCM**— Data Tightly Coupled Memory
- DTR**— Data Transmit Ready
- DVFS**— Dynamic Voltage and Frequency Scaling
- DVI**— Digital Video Interface

**E**

- EA**— Enumeration Address
- EAV**— End of Active Video
- ECC**— Error Checking and Correction. Also Error Correction Code.
- ECD3**— Entropy Coder/Decoder
- ED**— Endpoint Descriptor
- EFUSE**— Electrical Fuse. A one-time programmable memory location usually set at the factory
- EHCI**— Enhanced Host Controller Interface

**EMC**— External Memory Controller  
**EMI**— Electromagnetic interference  
**EMIF**— External Memory Interface  
**EOB**— End of Block  
**EOF**— End of Frame  
**EOL**— End of Line  
**EOT**— End of Transfer  
**EPM**— Emulation Pin Manager  
**ES**— Erase Status  
**ETB**— Embedded Trace Bus  
**ETM**— Embedded Trace Macrocell

**F**

**FAT**— File Allocation Table  
**FBB**— Forward Body-Bias  
**FCLK**— Functional Clock  
**FCS**— Frame Check Sequence  
**FD**— Face Detect  
**FE**— Framing Error.  
**FEC**— Frame End Code  
**FF**— Flip-Flop  
**FIFO**— First In First Out.  
**FIQ**— Fast Interrupt Request. See ISR.  
**FIR**— Fast Infrared  
**FM**— Frequency Modulate  
**FPS**— Frame per Second  
**FROM**— eFuse ROM  
**FS**— Full-Speed  
**FSC**— Frame Start Code. Also Frame Start Count.  
**FSM**— Finite State Machine.  
**FSR**— Fault Status Register  
**FV**— Focus Value  
**FW**— Firewall

**G**

**GBC**— Green Balance Correction

- GDP**— Generic Dot Product
- GHB**— Global History Buffer
- GPIO**— Genreal Purpose Input Output
- GPMC**— General Purpose Memory Controller

**H**

- H.263**— Video Codec Standart
- H.264**— Video Codec Standart
- H/W**— Hardware
- HAL**— Hardware Abstraction Layer
- HBP**— Horizontal Back Porch
- HC**— Host Controller
- HD**— High Definition
- HDCP**— High-bandwidth Digital Content Protection
- HDMI**— High-Definition Multimedia Interface
- HDQ**— Single-wire communication interface
- HDTV**— High-Definition Television
- HFP**— Horizontal Front Porch
- HLOS**— High-Level Operating System
- HNP**— Host Negotiation Protocol (OTG feature)
- HPF**— High-Pass Filter
- HPI**— Host Port Interface
- HS**— High-Speed
- HSEC**— Horizontal Sync End Code
- HSI**— High Speed Synchronous Serial Interface
- HSSC**— Horizontal Sync Start Code
- HSSCLL**— High-Speed Serial Control Channel
- HSW**— Horizontal Synchronization Pulse Width
- HSYNC**— Horizontal Synchronization.
- HW**— Hardware
- HWA**— Hardware Accelerators.
- HWOBS**— Hardware Observability
- HWSEQ**— Hardware Sequencer

**I**

- I/F**— Interface

- I/O**— Input/Output
- I2C**— Inter-Integrated Circuit.
- I2S**— Inter-IC Sound.
- IA**— Identifier Address
- ICE**— In-Circuit Emulation
- ICEPICK**— Generic TAP for emulation control
- ICLK**— Interface Clock
- ICONT**— Imaging Controller
- ICR**— Intersystem Communication Registers
- ID**— Identification
- IDCT**— Inverse Discrete Cosine Transform. See DCT.
- ILF3**— Improved Loop Filter engine
- IM**— Initiator Module. A module is an initiator whenever it is able to initiate read and write requests to the chip interconnect (typically: processors , DMA...).
- IME3**— Improved Motion Estimation engine
- IMX**— Image Extension coprocessor
- INL**— Integral Nonlinearity. It describes the maximum deviation between the ideal output of a DAC and the actual output level.
- INT**— Interrupt .
- INTC**— Interrupt Controller
- IP**— Intellectual Property
- IPC**— Interprocessor Communication. (also referred to as mailbox on occasion)
- IPE3**— Intra Prediction Estimation engine
- IPIPE**— Image Pipe
- IPIPEIF**— Image Pipe Interface
- IQ**— Inverse Quantization
- IR**— Incremental Redundancy Buffer
- IRQ**— Interrupt Request.
- ISA**— Instruction Set Architecture
- ISIF**— Image Sensor Interface
- ISP**— Image Signal Processor
- ISR**— Interrupt Service Routine.
- ISS**— Image SubSystem
- IST**— Interrupt Service Thread.
- ITCM**— Instruction Tightly Coupled Memory

- IV**— Initialization Vector
- IVA**— Image and Video Accelerator
- IVA-HD**— High Definition Image and Video Accelerator
- IrDA**— Infrared Data Association.

**J**

- JEDEC**— Joint Electronic Devices Engineering Council
- JPEG**— Joint Photographics Experts Group
- JTAG**— Joint Test Action Group.

**K**

- KB**— Kilobyte, 1024 B
- KBD**— Keyboard
- Kbps**— Kilobits per second

**L**

- L1**— Level 1 cache/memory
- L2**— Level 2 cache/memory
- L3**— First level of interconnect in OMAP platform
- L4**— Second level of interconnect in OMAP platform
- LA**— Logical Address
- LAN**— Local Area Network
- LC**— Logical Channel
- LCD**— Liquid Crystal Display.
- LCM**— Logical Channel to Memory
- LCh**— Logical DMA Channel. Also LCH
- LDC**— Load (from memory) to Coprocessor
- LDC2**— Lens Distortion Correction
- LDM**— Load Multiple
- LDO**— Low Dropout
- LE**— Little Endian.
- LEC**— Line End Code
- LED**— Light Emitting Diode.
- LF**— Loop Filter
- LFB**— Line-Fill-Buffer
- LFN**— Long File Name
- LH**— Local Host

**LINK**— Link Layer Device  
**LLP**— Low-Level Protocol  
**LP**— Low-Power, operation mode for PHY  
**LPCM**— Linear Pulse Code Modulation  
**LPDDR**— Low Power Double Data Rate  
**LPF**— Loop Filter  
**LPM**— Low-Power Mode  
**LPP**— Lines Per Panel  
**LRB**— Line-Read-Buffer  
**LS**— Low-Speed  
**LSB**— Least Significant Bit  
**LSC**— Line Start Code  
**LSR**— Linear Shift Register  
**LSW**— Least Significant Word  
**LUT**— Look-up Table

**M**

**M2**— Micro Memory  
**MAC**— Message Authentication Code  
**MB**— Megabyte, 1024 KB  
**MBAFF**— MB-Level Adaptive Frame/Field  
**MBR**— Master Boot Record  
**MC3**— Motion Compensation Engine  
**MCSPi**— Multichannel Serial Port Interface  
**MCU**— Microcontroller Unit. Refers to the MPU.  
**ME**— Motion Estimation  
**MEMIF**— Memory Interface  
**MIF**— Memory InterFace  
**MIPI**— Mobile Industry Processor Interface  
**MIR**— Medium Infrared  
**MJPEG**— Motion JPEG  
**MMC**— Multimedia Card  
**MMC/SD**— Multimedia Card/SD  
**MMU**— Memory Management Unit.  
**MP3**— MPEG Layer 3.

**MPEG**— Motion Pictures Expert Group.  
**MPEG1**— The first MPEG compression scheme specification.  
**MPU**— Microprocessor Unit.  
**MS**— Memory Stick  
**MSB**— Most Significant Bit  
**MSGIF**— Message Interface  
**Mb**— Megabit  
**Mbps**— Mega bits per second  
**McBSP**— Multichannel Buffered Serial Port.  
**Modem**— Modulator Demodulator

**N**

**N/A**— Not Applicable  
**NAC**— Network Access Control  
**NAK**— Not Acknowledged  
**NAND**— NAND Flash memory.  
**NC**— Not Connected  
**NIU**— Network Interface Unit  
**NMI**— Nonmaskable Interrupt. An interrupt that can be neither masked nor disabled.  
**NOP**— No Operation (DSP/CPU instruction)  
**NOR**— A type of flash memory  
**NRZI**— Non-Return-to-Zero Inverted  
**NSF**— Noise Filter  
**NTSC**— National Television System Committee. Television broadcast system.  
**NVB**— Number of Valid Bytes  
**NVIC**— Nested Vectored Interrupt Controller  
**NVM**— Non-volatile Memory

**O**

**OCM**— On-chip Memory  
**OCMC**— On-chip Memory Controller  
**OCP**— Open-Core Protocol  
**OCPI**— Open-Core Protocol Interface  
**OE**— Output Enable  
**OHCI**— Open Host Controller Interface. This is an industry standard USB Host Controller Interface.  
**OMAP**— Open Multimedia Application Platform

- ON2**— Video Codec Standart
- OPP**— Operating Performance Point
- OS**— Operating System
- OSI**— (OSI model) Open Systems Interconnection Basic Reference Model
- OTG**— On-The-Go (USB 2.0 specification)

**P**

- PA**— Program Address
- PBIAS**— PMOS Bias transistor to provide the bias voltage to extended drain IOs
- PC**— Program Counter
- PCB**— Printed Circuit Board
- PCI**— Peripheral Component Interconnect.
- PCLK**— Pixel Clock
- PCM**— Pulse Code Modulation.
- PD**— Program Data
- PDA**— Personal Digital Assistant
- PDC**— Power-down Controller
- PDM**— Pulse Density Modulation
- PDU**— Protocol Data Unit
- PE**— Parity Error
- PF**— Packet Footer
- PFPW**— Prefetch and Prewrite posting engine
- PH**— Packet Header
- PHY**— Physical Layer Device
- PI**— Pixel Interpolation
- PID**— Protocol Identifier. The PID register is used in Windows CE mode only.
- PLL**— Phase-Locked Loop.
- PM**— Programming Model
- PMC**— Program Memory Controller
- PMEM**— Program Memory
- PMP**— Power Management Port
- POR**— Power-On Reset
- PPA**— Primary Protected Application.
- PPC**— Palm-size PC
- PPI**— Physical Layer Protocol Interface



- PPL**— Pixels per Line
- PRCM**— Power, Reset, Clock Management module
- PRM**— Power and Reset manager
- PS**— Packet Start
- PSC**— Prescaler Counter
- PSS**— Program Suspend Status
- PT**— Packet Type
- PTM**— Program Trace Macrocell
- PTV**— Prescale Clock Timer Value. Sets the value of the divisor used in scaling the clock.
- PU/PD**— Pull-Up / Pull-Down
- PWL**— Pulse Width Light (modulator). A 4096-bit randomsequence generator that provides control of the LCD backlighting and keypad.
- PWM**— Pulse Width Modulation
- PWR**— Power

**Q**

- QIQ**— Quantization and Inverse Quantization
- QMEM**— Quantizer Memory
- QMR**— Quantizer Matrix
- QVGA**— Quarter Video Graphics Array. One-fourth the resolution of VGA.

**R**

- R/W**— Read/Write. Also RW.
- R5**— Release 5 of 3GPP specifications on IMS and HSDPA standards
- R6**— Release 6 of 3GPP specifications on Wireless LAN networks, HSUPA, MBMS and enhancements to IMS standards
- RAM**— Random Access Memory. A memory element that can be written to, as well as read.
- RBB**— Reverse Body-Bias
- RCA**— Relative Card Address
- RDR**— Receive Data Register
- RE**— Read Enable
- REQ**— Request
- RF**— Radio Frequency
- RFB**— Remote Frame Buffer
- RFBI**— Remote Frame Buffer Interface
- RFF**— Retention Flip-Flop
- RGB**— Red Green Blue

**RGBA**— Red Green Blue Alpha

**RI**— Ring Indicator

**RM**— Reed-Muller code

**RO**— Read Only

**ROM**— Read Only Memory. A semiconductor storage element containing permanent data that cannot be changed.

**RST**— Reset

**RT**— Real-Time

**RTA**— Retention Till Access

**RTC**— Real-Time Clock. A clock that keeps track of the time even when the device is turned off.

**RTL**— Register Transfer Level

**RTOS**— Real-Time Operating System

**RTS**— Request to Send

**RVLC**— Reversible Variable Length Coder

**RX**— Receive/Receiver

**RXD**— Receive Data

**S**

**S/PDIF**— Sony/Philips Digital Interface

**S/W**— Software

**SAM**— Signal Amplitude Modulation

**SAR**— Save and Restore. Hardware context saving for power saving.

**SAV**— Start of Active Video

**SB**— Silicon Backplane (Trade Mark)

**SBC**— Stream Buffer Controller

**SCCB**— Serial Camera Control Interface. 3-wire and 2-wire serial bus defined and deployed by Omnivision Technologies, Inc.

**SCL**— Serial Clock. Programmable serial clock used in the I2C interface. Also SCLK.

**SCM**— System Control Module

**SCP**— Serial Configuration Port

**SCR**— SDIO Configuration Registers

**SCRM**— System Clock and Reset Manager

**SCTM**— System Counter Timer Module

**SCU**— Snoop Control Unit

**SD**— SD card. A non-volatile memory card.

**SDA**— Serial Data. Serial data bus in the I2C interface.

- SDHC**— SD High Capacity card
- SDIO**— SD Input/Output
- SDMA**— System Direct Memory Access module
- SDP**— Software Development Platform
- SDR**— Single Data Rate
- SDRAM**— Synchronous Dynamic Random Access Memory
- SDRC**— SDRAM Controller.
- SE**— Safe Environment. Execution environment inside a device, which is protected against tampering
- SFL**— Subframe Length
- SGX**— Acronym for Graphics Accelerator
- SIM**— Subscriber Identity Module
- SIMCOP**— Still image coprocessor
- SIMD**— Single Instruction-Stream, Multiple Data-Stream
- SIR**— Slow Infrared
- SL2**— Shared Level 2 (memory/interface)
- SLM**— Static Leakage Management
- SLVS**— Scalable Low Voltage Signaling
- SMC**— Shared Message Channel
- SMEM**— Sample Memory
- SMI**— Safe Monitor Interrupt
- SMP**— Symmetric Multiprocessor Platform
- SMPS**— Switch Mode Power Supply
- SMSET**— Software Message and System Event Trace module
- SNR**— Signal-to-Noise Ratio
- SOC**— System-On-a-Chip
- SOF**— Start Of Frame
- SP**— Serial Port or Small Page
- SPI**— Serial Port Interface. A signaling protocol for exchanging serial data.
- SR**— SmartReflex
- SRAM**— Static Random Access Memory
- SRC**— Sample Rate Conversion
- SRG**— Sample Rate Generator
- SRP**— Session Request Protocol (OTG feature)
- SS**— Subsystem

- SSC**— Spread Spectrum Clocking
- SSI**— Serial Synchronous Interface
- ST**— Start Timer
- STC**— Store from Coprocessor (to memory) or System Time Clock, which is the master clock in an MPEG-2 encoder or decoder system.
- STM**— Synchronous Transfer Mode or Store Multiple.
- STN**— Super-Twist Nematic. A technique for improving LCD display screens by twisting light rays.
- SW**— Software
- SWI**— Software Interrupt
- SXGA**— Super eXtended Graphics Array
- SYSC**— System Control Module
- SmartReflex**— Dynamic voltage sensing module that generates the voltage error signal proportional to the difference in desired voltage and the current voltage

**T**

- TA**— Target Agent
- TAP**— Test Access Port
- TC**— Traffic Controller. Allows asynchronous operation among the external memory interface, the MPU, and the DSP.
- TCK**— Test Clock
- TCM**— Tightly Coupled Memory
- TD**— Transfer Descriptor
- TDI**— Test Data Input
- TDM**— Time Division Multiplex/Multiplexing
- TDO**— Test Data Output
- TFT**— Thin Film Transistor. A type of LCD flat panel display screen in which each pixel is controlled by one to four transistors.
- TI**— Texas Instruments
- TL**— Transmission Line
- TLB**— Translation Lookaside Buffer. A cache that contains entries for virtual-to-physical address translation and access permission checking.
- TLL**— Transceiver Less Link. This is logic which allows the user to connect two USB transceiver interfaces together directly without the use of differential transceivers.
- TM**— Target Module. A target module cannot generate read/write requests to the chip interconnects, but respond to these requests. However it may generate interrupts or DMA request to the system (typically: peripherals, memory controllers).
- TMDS**— Transition Minimized Differential Signaling. A technology for transmitting high-speed serial data and is used by the DVI and HDMI video interfaces.
- TMS**— Test Mode Select

**TOC**— Table of Contents

**TP**— Tiny Page

**TRM**— Technical Reference Manual

**TRST**— Test Reset

**TRX**— USB Transceiver. The USB analog driver/receiver.

**TSHUT**— Temperature Shutdown.

**TTB**— Translation Tabbe Base. It points to the base of a table in physical memory that contains section and page table descriptors.

**TTH**— Translation Table Hierarchy

**TTL**— Transistor Transistor Logic

**TWL**— Table Walking Logic

**TX**— Transmit/Transmitter

## U

**UART**— Universal Asynchronous Receiver/Transmitter. Another name for the asynchronous serial port.

**UE**— Unrecoverable Error

**UHS**— Definition of SD cards with higher frequency

**ULPI**— UTMI+ Low Pin Interface (12-pin interface standard for connecting USB core logic to a USB transceiver)

**ULPM**— Ultralow-Power Mode

**ULPS**— Ultralow-Power State

**UMC**— Unified Memory Controller

**USB**— Universal Serial Bus. An external bus standard that supports data transfer rates of 12M bps (12 million bits per second). A single USB port can be used to connect up to 127 peripheral devices.

**USSE**— Universal Scalable Shader Engine

**UTMI**— USB 2.0 Transceiver Macrocell Interface

## V

**VA**— Volt-Amps. A form of power management. A VA rating is the volts rating multiplied by the amps (current) rating, used to indicate the output capacity of an uninterruptible power supply (UPS) or other power source.

**VBP**— Vertical Back Porch

**VC**— Virtual Channel

**VC-1**— Video Codec Standart

**VDMA**— Video Direct Memory Access module

**VENC**— Video Encoder

**VESA**— Video Electronics Standards Association

**VFP**— Vertical Front Porch

**VGA**— Video Graphics Array. An industry standard for video cards.

**VLC**— Variable Length Decoder

**VLCD**— Variable Length Coding and Decoding coprocessor

**VLCDJ**— Variable-Length Coder/Decoder for JPEG

**VLD**— Variable Length Coder

**VLIW**— Very Long Instruction Word

**VMODE**— Bi-level voltage control interface

**VP**— Video Port

**VRFB**— Virtual Rotation Fixed Buffer

**VS**— Vertical Synchronization

**VSEC**— Vertical Sync End Code

**VSSC**— Vertical Sync Start Code

**VSW**— Vertical Synchronization Pulse Width

**VSYNC**— Vertical Synchronization. A bidirectional vertical timing signal occurring once per frame with a pulse-width defined as an integral number of lines (half-lines for interlaced mode). Also VS.

## W

**WB**— Write Buffer

**WC**— Word Count

**WD**— Watchdog. A timer that requires the user program or OS periodically write to the count register before the counter underflows.

**WDT**— Watchdog Timer

**WE**— Write Enable

**WFI**— Wait For Interrupt

**WNP**— Write Non-Posted

**WP**— Write Protect

**WSS**— Wide-Screen Signaling

**WT**— Write Through

**WTBU**— Wireless Terminal Business Unit

**WUGEN**— Wake-Up Generator

**Word16**— 16 bits word

## X

**XGA**— eXtended Graphics Array. Also XVGA.

**XIP**— eXecution In Place

## Y

**YUV**— Luminance-Bandwidth-Chrominance

**e**

**eMMC**— Embedded Multimedia Card

**eSD**— Embedded SD

PRELIMINARY

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
RF/IF and ZigBee® Solutions	<a href="http://www.ti.com/lprf">www.ti.com/lprf</a>

### Applications

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics & Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
Wireless	<a href="http://www.ti.com/wireless-apps">www.ti.com/wireless-apps</a>